Kneron Inc

Document Name: **Kneron KL520 Driver API**

KL520 Driver API
Kneron Inc

Engineering Design Document

Kneron Confidential

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Kneron Confidential

Kneron Confidential

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Kneron Confidential

Kneron Confidential

Kneron Confidential

Kneron Confidential

Kneron Confidential

Kneron Confidential

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Kneron Confidential

Kneron Confidential

Kneron Confidential

Kneron Confidential

Chapter 4

Module Documentation

4.1 KDRV_ADC

Kneron generic adc driver.

Data Structures

- struct kdrv_adc_resource_t
- struct kdrv_adc_regs_t

Macros

- #define HTHR_EN (1<<31)
- #define HTHR(x) (((x)&0xFFF)<<16)
- #define LTHR_EN (1<<15)
- #define LTHR(x) (((x)&0xFFF)<<0)
- #define SCAN_NUM(x) (x<<16)
- #define SCANMODE_CONT (1<<9)
- #define SCANMODE_SGL (1<<8)
- #define SWSTART (1<<4)
- #define ADC_EN (1<<0)
- #define CHDONE_INTEN(x) (1<<((x)+8))
- #define TS_OVR_INTREN (1<<3)
- #define TS_UDR_INTREN (1<<2)
- #define STOP_INTEN (1<<1)
- #define DONE_INTEN (1<<0)
- #define CH_INTRSTS(x) (1<<((x)+8))
- #define TS_THDOD_INTRSTS (1<<3)
- #define TS_THDUD_INTRSTS (1<<2)
- #define ADC_STOP_INTRSTS (1<<1)
- #define ADC_DONE_INTSTS (1<<0)

Kneron Confidential

Functions

- kdrv_status_t kdrv_adc_initialize (void)

  ADC driver initialization.
- kdrv_status_t kdrv_adc_uninitialize (kdrv_adc_resource_t *res)

  ADC driver uninitialization.
- kdrv_status_t kdrv_adc_rest (kdrv_adc_resource_t *res)

  ADC reset control.
- kdrv_status_t kdrv_adc_enable (kdrv_adc_resource_t *res, int mode)

  ADC enable control.
- int kdrv_adc_read (int id)

  ADC data read control.

Variables

- int kdrv_adc_resource_t::io_base
- int kdrv_adc_resource_t::irq
- uint32_t kdrv_adc_regs_t::data [8]
- uint32_t kdrv_adc_regs_t::reserve [24]
- uint32_t kdrv_adc_regs_t::thrhold [8]
- uint32_t kdrv_adc_regs_t::reserve1 [24]
- uint32_t kdrv_adc_regs_t::ctrl
- uint32_t kdrv_adc_regs_t::trim
- uint32_t kdrv_adc_regs_t::inten
- uint32_t kdrv_adc_regs_t::intst
- uint32_t kdrv_adc_regs_t::tparam
- uint32_t kdrv_adc_regs_t::smpr
- uint32_t kdrv_adc_regs_t::reserve2
- uint32_t kdrv_adc_regs_t::prescal
- uint32_t kdrv_adc_regs_t::sqr

4.1.1 Detailed Description

Kneron generic adc driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.1.2 Macro Definition Documentation

4.1.2.1 ADC_DONE_INTSTS

```
#define ADC_DONE_INTSTS (1<<0)
```

Kneron Confidential

4.1.2.2 ADC_EN

```
#define ADC_EN (1<<0)
```

4.1.2.3 ADC_STOP_INTRSTS

```
#define ADC_STOP_INTRSTS (1<<1)
```

4.1.2.4 CH_INTRSTS

```
#define CH_INTRSTS(
              x ) (1<<((x)+8))
```

4.1.2.5 CHDONE_INTEN

```
#define CHDONE_INTEN(
              x ) (1<<((x)+8))
```

4.1.2.6 DONE_INTEN

```
#define DONE_INTEN (1<<0)
```

4.1.2.7 HTHR

```
#define HTHR(
              x ) (((x)&0xFFF)<<16)
```

4.1.2.8 HTHR_EN

```
#define HTHR_EN (1<<31)
```

Kneron Confidential

4.1.2.9 LTHR

```
#define LTHR(
             x ) (((x)&0xFFF)<<0)
```

4.1.2.10 LTHR_EN

```
#define LTHR_EN (1<<15)
```

4.1.2.11 SCAN_NUM

```
#define SCAN_NUM(
             x ) (x<<16)
```

4.1.2.12 SCANMODE_CONT

```
#define SCANMODE_CONT (1<<9)
```

4.1.2.13 SCANMODE_SGL

```
#define SCANMODE_SGL (1<<8)
```

4.1.2.14 STOP_INTEN

```
#define STOP_INTEN (1<<1)
```

4.1.2.15 SWSTART

```
#define SWSTART (1<<4)
```

Kneron Confidential

4.1.2.16 TS_OVR_INTREN

```
#define TS_OVR_INTREN (1<<3)
```

4.1.2.17 TS_THDOD_INTRSTS

```
#define TS_THDOD_INTRSTS (1<<3)
```

4.1.2.18 TS_THDUD_INTRSTS

```
#define TS_THDUD_INTRSTS (1<<2)
```

4.1.2.19 TS_UDR_INTREN

```
#define TS_UDR_INTREN (1<<2)
```

4.1.3 Function Documentation

4.1.3.1 kdrv_adc_enable()

```
kdrv_status_t kdrv_adc_enable (
            kdrv_adc_resource_t * res,
            int mode )
```

ADC enable control.

Parameters

in	*res	a handle of a ADC resource mode which mode to be enabled

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.1.3.2 kdrv_adc_initialize()

```
kdrv_status_t kdrv_adc_initialize (
            void  )
```

ADC driver initialization.

Returns

kdrv_status_t see kdrv_status_t

4.1.3.3 kdrv_adc_read()

```
int kdrv_adc_read (
            int id )
```

ADC data read control.

Parameters

in	*id*	which adc channel to be enabled

Returns

int return ADC data

4.1.3.4 kdrv_adc_rest()

```
kdrv_status_t kdrv_adc_rest (
            kdrv_adc_resource_t * res )
```

ADC reset control.

Parameters

in	**res*	a handle of a ADC resource

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.1.3.5 kdrv_adc_uninitialize()

```
kdrv_status_t kdrv_adc_uninitialize (
            kdrv_adc_resource_t * res )
```

ADC driver uninitialization.

Parameters

in	*res	a handle of a ADC resource

Returns

kdrv_status_t see kdrv_status_t

4.1.4 Variable Documentation

4.1.4.1 ctrl

```
uint32_t kdrv_adc_regs_t::ctrl
```

4.1.4.2 data

```
uint32_t kdrv_adc_regs_t::data[8]
```

4.1.4.3 inten

```
uint32_t kdrv_adc_regs_t::inten
```

4.1.4.4 intst

```
uint32_t kdrv_adc_regs_t::intst
```

4.1.4.5 io_base

```
int kdrv_adc_resource_t::io_base
```

Kneron Confidential

4.1.4.6 irq

```
int kdrv_adc_resource_t::irq
```

4.1.4.7 prescal

```
uint32_t kdrv_adc_regs_t::prescal
```

4.1.4.8 reserve

```
uint32_t kdrv_adc_regs_t::reserve[24]
```

4.1.4.9 reserve1

```
uint32_t kdrv_adc_regs_t::reserve1[24]
```

4.1.4.10 reserve2

```
uint32_t kdrv_adc_regs_t::reserve2
```

4.1.4.11 smpr

```
uint32_t kdrv_adc_regs_t::smpr
```

4.1.4.12 sqr

```
uint32_t kdrv_adc_regs_t::sqr
```

4.1.4.13 thrhold

```
uint32_t kdrv_adc_regs_t::thrhold[8]
```

4.1.4.14 tparam

```
uint32_t kdrv_adc_regs_t::tparam
```

4.1.4.15 trim

```
uint32_t kdrv_adc_regs_t::trim
```

Kneron Confidential

4.2 KDRV_CLOCK

Kneron generic clock driver.

Data Structures

- struct kdrv_clock_value
- struct kdrv_clock_list
- struct kdrv_clock_node

Macros

- #define CLOCK_MUXSEL_NCPU_TRACECLK_DEFAULT 0x10000000
- #define CLOCK_MUXSEL_NCPU_TRACECLK_FROM_SCPU_TRACECLK 0x20000000
- #define CLOCK_MUXSEL_NCPU_TRACECLK_MASK 0x30000000
- #define CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV3 0x01000000
- #define CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV2 0x02000000
- #define CLOCK_MUXSEL_SCPU_TRACECLK_MASK 0x03000000
- #define CLOCK_MUXSEL_CSIRX1_CLK_PLL5 0x00100000
- #define CLOCK_MUXSEL_CSIRX1_CLK_PLL3 0x00200000
- #define CLOCK_MUXSEL_CSIRX1_CLK_MASK 0x00300000
- #define CLOCK_MUXSEL_NPU_CLK_PLL4 0x00010000
- #define CLOCK_MUXSEL_NPU_CLK_PLL5 0x00020000
- #define CLOCK_MUXSEL_NPU_CLK_PLL0 0x00040000
- #define CLOCK_MUXSEL_NPU_CLK_MASK 0x00070000
- #define CLOCK_MUXSEL_PLL4_FREF_PLL0DIV 0x00001000
- #define CLOCK_MUXSEL_PLL4_FREF_OSC 0x00002000
- #define CLOCK_MUXSEL_PLL4_MASK 0x00003000
- #define CLOCK_MUXSEL_UART_0_IRDA_UCLK_UART 0x00000100
- #define CLOCK_MUXSEL_UART_0_IRDA_UCLK_IRDA 0x00000200
- #define CLOCK_MUXSEL_UART_0_IRDA_UCLK_MASK 0x00000300

Typedefs

- typedef int(∗ fn_set) (struct kdrv_clock_node ∗, struct kdrv_clock_value ∗)

Enumerations

- enum clk {
  CLK_PLL1 = 1, CLK_PLL1_OUT, CLK_PLL2, CLK_PLL2_OUT,
  CLK_PLL3, CLK_PLL3_OUT1, CLK_PLL3_OUT2, CLK_PLL4,
  CLK_PLL4_OUT, CLK_PLL5, CLK_PLL5_OUT1, CLK_PLL5_OUT2,
  CLK_FCS_PLL2 = 20, CLK_FCS_DLL, CLK_PLL4_FREF_PLL0, CLK_BUS_SAHB = 30,
  CLK_BUS_NAHB, CLK_BUS_PAHB1, CLK_BUS_PAHB2, CLK_BUS_APB0,
  CLK_BUS_APB1, CLK_SCPU = 50, CLK_SCPU_TRACE, CLK_NCPU = 60,
  CLK_NCPU_TRACE, CLK_NPU, CLK_SPI_CLK = 100, CLK_ADC_CLK,
  CLK_WDT_EXT_CLK, CLK_SD_CLK, CLK_MIPI_TXHSPLLREF_CLK, CLK_MIPI_TX_ESC_CLK,
  CLK_MIPI_CSITX_DSI_CLK, CLK_MIPI_CSITX_CSI_CLK, CLK_MIPI_CSIRX1_TXESC_CLK, CLK_MIPI_CSIRX1_CSI_CLK
  CLK_MIPI_CSIRX1_VC0_CLK, CLK_MIPI_CSIRX0_TXESC_CLK, CLK_MIPI_CSIRX0_CSI_CLK, CLK_MIPI_CSIRX0_VC0_
  CLK_LC_SCALER, CLK_LC_CLK, CLK_TMR1_EXTCLK3, CLK_TMR1_EXTCLK2,
  CLK_TMR1_EXTCLK1, CLK_TMR0_EXTCLK3, CLK_TMR0_EXTCLK2, CLK_TMR0_EXTCLK1,
  CLK_PWM_EXTCLK6, CLK_PWM_EXTCLK5, CLK_PWM_EXTCLK4, CLK_PWM_EXTCLK3,
  CLK_PWM_EXTCLK2, CLK_PWM_EXTCLK1, CLK_UART1_3_FREF, CLK_UART1_2_FREF,
  CLK_UART1_1_FREF, CLK_UART1_0_FREF, CLK_UART0_FREF, CLK_SSP1_1_SSPCLK,
  CLK_SSP1_0_SSPCLK, CLK_SSP0_1_SSPCLK, CLK_SSP0_0_SSPCLK }

Kneron Confidential

- enum pll_id {
  pll_1 = 0, pll_2, pll_3, pll_4,
  pll_5 }
- enum scuclkin_type { scuclkin_osc = 0, scuclkin_rtcosc, scuclkin_pll0div3, scuclkin_pll0div4 }

Functions

- void kdrv_clock_mgr_init (void)
- void kdrv_clock_mgr_open (struct kdrv_clock_node *node, struct kdrv_clock_value *clock_val)
- void kdrv_clock_mgr_close (struct kdrv_clock_node *node)
- void kdrv_clock_mgr_set_scuclkin (enum scuclkin_type type, bool enable)
- void kdrv_clock_mgr_set_muxsel (uint32_t flags)
- uint32_t kdrv_clock_mgr_calculate_clockout (enum pll_id id, uint16_t ms, uint16_t ns, uint16_t F_ps)
- void kdrv_clock_mgr_open_pll1 (void)
- void kdrv_clock_mgr_open_pll2 (void)
- void kdrv_clock_mgr_open_pll3 (void)
- void kdrv_clock_mgr_open_pll4 (void)
- void kdrv_clock_mgr_open_pll5 (void)
- void kdrv_clock_mgr_close_pll1 (void)
- void kdrv_clock_mgr_close_pll2 (void)
- void kdrv_clock_mgr_close_pll4 (void)
- void kdrv_clock_mgr_change_pll3_clock (uint32_t ms, uint32_t ns, uint32_t ps, uint32_t csi0_txesc, uint32_t csi0_csi, uint32_t csi0_vc0, uint32_t csi1_txesc, uint32_t csi1_csi, uint32_t csi1_vc0)
- void kdrv_clock_mgr_change_pll5_clock (uint32_t ms, uint32_t ns, uint32_t ps)
- void kdrv_debug_pll_clock (void)
- void kdrv_clock_set_csiclk (uint32_t cam_idx, uint32_t enable)
- void kdrv_clock_enable (enum clk clk)
- void kdrv_clock_disable (enum clk clk)
- void kdrv_delay_us (uint32_t usec)

Variables

- uint16_t kdrv_clock_value::ms
- uint16_t kdrv_clock_value::ns
- uint16_t kdrv_clock_value::ps
- uint8_t kdrv_clock_value::div
- uint8_t kdrv_clock_value::enable
- struct kdrv_clock_list * kdrv_clock_list::next
- struct kdrv_clock_node * kdrv_clock_node::parent
- struct kdrv_clock_node * kdrv_clock_node::child_head
- struct kdrv_clock_node * kdrv_clock_node::child_next
- fn_set kdrv_clock_node::set
- uint8_t kdrv_clock_node::is_enabled
- char kdrv_clock_node::name [15]
- struct kdrv_clock_node clock_node_pll1_out

4.2.1 Detailed Description

Kneron generic clock driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.2.2 Macro Definition Documentation

4.2.2.1 CLOCK_MUXSEL_CSIRX1_CLK_MASK

```
#define CLOCK_MUXSEL_CSIRX1_CLK_MASK 0x00300000
```

4.2.2.2 CLOCK_MUXSEL_CSIRX1_CLK_PLL3

```
#define CLOCK_MUXSEL_CSIRX1_CLK_PLL3 0x00200000
```

4.2.2.3 CLOCK_MUXSEL_CSIRX1_CLK_PLL5

```
#define CLOCK_MUXSEL_CSIRX1_CLK_PLL5 0x00100000
```

4.2.2.4 CLOCK_MUXSEL_NCPU_TRACECLK_DEFAULT

```
#define CLOCK_MUXSEL_NCPU_TRACECLK_DEFAULT 0x10000000
```

4.2.2.5 CLOCK_MUXSEL_NCPU_TRACECLK_FROM_SCPU_TRACECLK

```
#define CLOCK_MUXSEL_NCPU_TRACECLK_FROM_SCPU_TRACECLK 0x20000000
```

4.2.2.6 CLOCK_MUXSEL_NCPU_TRACECLK_MASK

```
#define CLOCK_MUXSEL_NCPU_TRACECLK_MASK 0x30000000
```

4.2.2.7 CLOCK_MUXSEL_NPU_CLK_MASK

```
#define CLOCK_MUXSEL_NPU_CLK_MASK 0x00070000
```

Kneron Confidential

4.2.2.8 CLOCK_MUXSEL_NPU_CLK_PLL0

```
#define CLOCK_MUXSEL_NPU_CLK_PLL0 0x00040000
```

4.2.2.9 CLOCK_MUXSEL_NPU_CLK_PLL4

```
#define CLOCK_MUXSEL_NPU_CLK_PLL4 0x00010000
```

4.2.2.10 CLOCK_MUXSEL_NPU_CLK_PLL5

```
#define CLOCK_MUXSEL_NPU_CLK_PLL5 0x00020000
```

4.2.2.11 CLOCK_MUXSEL_PLL4_FREF_OSC

```
#define CLOCK_MUXSEL_PLL4_FREF_OSC 0x00002000
```

4.2.2.12 CLOCK_MUXSEL_PLL4_FREF_PLL0DIV

```
#define CLOCK_MUXSEL_PLL4_FREF_PLL0DIV 0x00001000
```

4.2.2.13 CLOCK_MUXSEL_PLL4_MASK

```
#define CLOCK_MUXSEL_PLL4_MASK 0x00003000
```

4.2.2.14 CLOCK_MUXSEL_SCPU_TRACECLK_MASK

```
#define CLOCK_MUXSEL_SCPU_TRACECLK_MASK 0x03000000
```

4.2.2.15 CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV2

```
#define CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV2 0x02000000
```

Kneron Confidential

4.2.2.16 CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV3

```
#define CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV3 0x01000000
```

4.2.2.17 CLOCK_MUXSEL_UART_0_IRDA_UCLK_IRDA

```
#define CLOCK_MUXSEL_UART_0_IRDA_UCLK_IRDA 0x00000200
```

4.2.2.18 CLOCK_MUXSEL_UART_0_IRDA_UCLK_MASK

```
#define CLOCK_MUXSEL_UART_0_IRDA_UCLK_MASK 0x00000300
```

4.2.2.19 CLOCK_MUXSEL_UART_0_IRDA_UCLK_UART

```
#define CLOCK_MUXSEL_UART_0_IRDA_UCLK_UART 0x00000100
```

4.2.3 Typedef Documentation

4.2.3.1 fn_set

```
typedef int(* fn_set) (struct kdrv_clock_node *, struct kdrv_clock_value *)
```

4.2.4 Enumeration Type Documentation

Kneron Confidential

Enumerator

4.2.4.1 clk

`enum clk`

Enumerator

CLK_PLL1	
CLK_PLL1_OUT	
CLK_PLL2	
CLK_PLL2_OUT	
CLK_PLL3	
CLK_PLL3_OUT1	
CLK_PLL3_OUT2	
CLK_PLL4	
CLK_PLL4_OUT	
CLK_PLL5	
CLK_PLL5_OUT1	
CLK_PLL5_OUT2	
CLK_FCS_PLL2	
CLK_FCS_DLL	
CLK_PLL4_FREF_PLL0	
CLK_BUS_SAHB	
CLK_BUS_NAHB	
CLK_BUS_PAHB1	
CLK_BUS_PAHB2	
CLK_BUS_APB0	
CLK_BUS_APB1	
CLK_SCPU	
CLK_SCPU_TRACE	
CLK_NCPU	
CLK_NCPU_TRACE	
CLK_NPU	
CLK_SPI_CLK	
CLK_ADC_CLK	
CLK_WDT_EXT_CLK	
CLK_SD_CLK	
CLK_MIPI_TXHSPLLREF_CLK	
CLK_MIPI_TX_ESC_CLK	
CLK_MIPI_CSITX_DSI_CLK	
CLK_MIPI_CSITX_CSI_CLK	
CLK_MIPI_CSIRX1_TXESC_CLK	
CLK_MIPI_CSIRX1_CSI_CLK	
CLK_MIPI_CSIRX1_VC0_CLK	
CLK_MIPI_CSIRX0_TXESC_CLK	
CLK_MIPI_CSIRX0_CSI_CLK	
CLK_MIPI_CSIRX0_VC0_CLK	
CLK_LC_SCALER	
CLK_LC_CLK	

Kneron Confidential

Enumerator

CLK_TMR1_EXTCLK3	
CLK_TMR1_EXTCLK2	
CLK_TMR1_EXTCLK1	
CLK_TMR0_EXTCLK3	
CLK_TMR0_EXTCLK2	
CLK_TMR0_EXTCLK1	
CLK_PWM_EXTCLK6	
CLK_PWM_EXTCLK5	
CLK_PWM_EXTCLK4	
CLK_PWM_EXTCLK3	
CLK_PWM_EXTCLK2	
CLK_PWM_EXTCLK1	
CLK_UART1_3_FREF	
CLK_UART1_2_FREF	
CLK_UART1_1_FREF	
CLK_UART1_0_FREF	
CLK_UART0_FREF	
CLK_SSP1_1_SSPCLK	
CLK_SSP1_0_SSPCLK	
CLK_SSP0_1_SSPCLK	
CLK_SSP0_0_SSPCLK	

4.2.4.2 pll_id

```
enum pll_id
```

Enumerator

pll_1	
pll_2	
pll_3	
pll_4	
pll_5	

4.2.4.3 scuclkin_type

```
enum scuclkin_type
```

Kneron Confidential

Enumerator

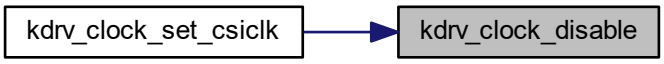
scuclkin_osc	
scuclkin_rtcosc	
scuclkin_pll0div3	
scuclkin_pll0div4	

4.2.5 Function Documentation

4.2.5.1 kdrv_clock_disable()

```
void kdrv_clock_disable (
            enum clk clk )
```

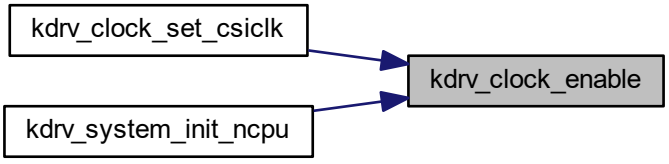
Here is the caller graph for this function:



4.2.5.2 kdrv_clock_enable()

```
void kdrv_clock_enable (
            enum clk clk )
```

Here is the caller graph for this function:

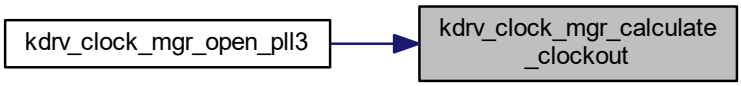


Kneron Confidential

4.2.5.3 kdrv_clock_mgr_calculate_clockout()

```
uint32_t kdrv_clock_mgr_calculate_clockout (
            enum pll_id id,
            uint16_t ms,
            uint16_t ns,
            uint16_t F_ps )
```

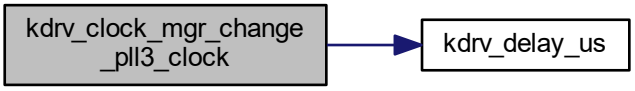
Here is the caller graph for this function:



4.2.5.4 kdrv_clock_mgr_change_pll3_clock()

```
void kdrv_clock_mgr_change_pll3_clock (
            uint32_t ms,
            uint32_t ns,
            uint32_t ps,
            uint32_t csi0_txesc,
            uint32_t csi0_csi,
            uint32_t csi0_vc0,
            uint32_t csi1_txesc,
            uint32_t csi1_csi,
            uint32_t csi1_vc0 )
```

Here is the call graph for this function:

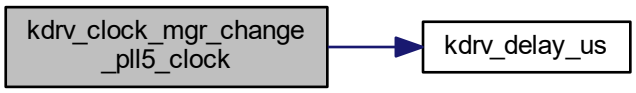


Kneron Confidential

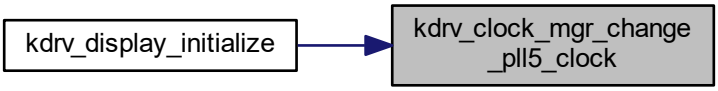
4.2.5.5 kdrv_clock_mgr_change_pll5_clock()

```
void kdrv_clock_mgr_change_pll5_clock (
            uint32_t ms,
            uint32_t ns,
            uint32_t ps )
```

Here is the call graph for this function:



Here is the caller graph for this function:



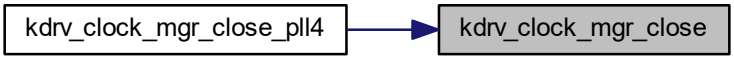
4.2.5.6 kdrv_clock_mgr_close()

```
void kdrv_clock_mgr_close (
            struct kdrv_clock_node * node )
```

Here is the call graph for this function:

Kneron Confidential

Here is the caller graph for this function:



4.2.5.7 kdrv_clock_mgr_close_pll1()

```
void kdrv_clock_mgr_close_pll1 (
            void  )
```

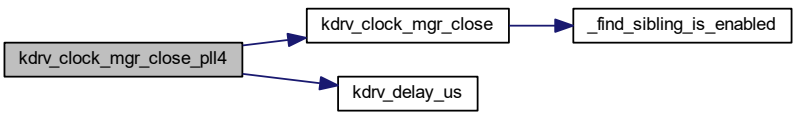
4.2.5.8 kdrv_clock_mgr_close_pll2()

```
void kdrv_clock_mgr_close_pll2 (
            void  )
```

4.2.5.9 kdrv_clock_mgr_close_pll4()

```
void kdrv_clock_mgr_close_pll4 (
            void  )
```

Here is the call graph for this function:



Kneron Confidential

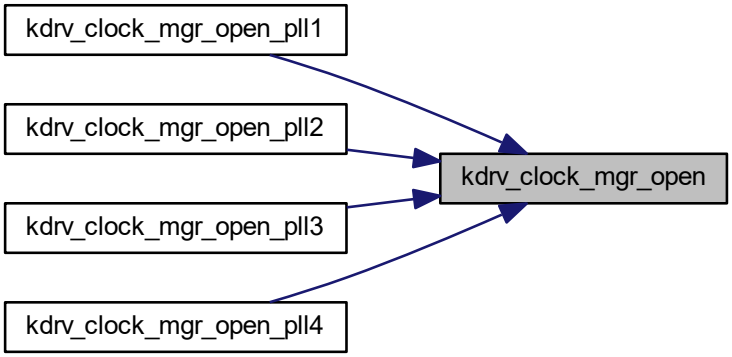
4.2.5.10 kdrv_clock_mgr_init()

```
void kdrv_clock_mgr_init (
            void  )
```

4.2.5.11 kdrv_clock_mgr_open()

```
void kdrv_clock_mgr_open (
            struct kdrv_clock_node * node,
            struct kdrv_clock_value * clock_val )
```

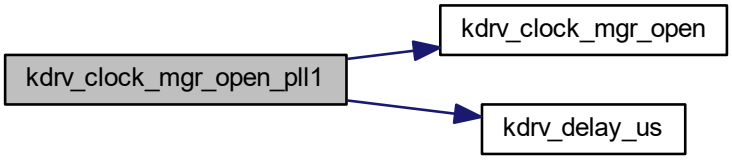
Here is the caller graph for this function:



4.2.5.12 kdrv_clock_mgr_open_pll1()

```
void kdrv_clock_mgr_open_pll1 (
            void )
```

Here is the call graph for this function:

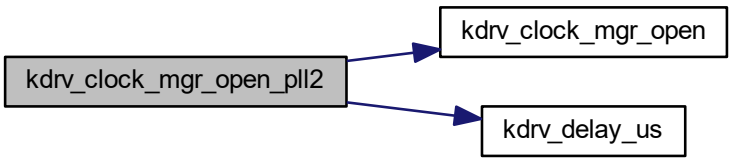


Kneron Confidential

4.2.5.13 kdrv_clock_mgr_open_pll2()

```
void kdrv_clock_mgr_open_pll2 (
            void  )
```

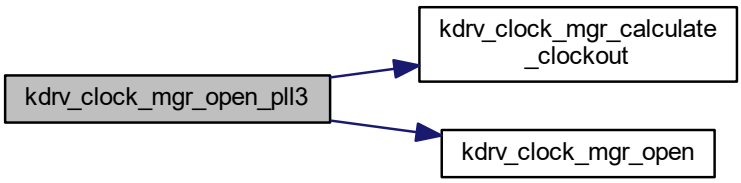
Here is the call graph for this function:



4.2.5.14 kdrv_clock_mgr_open_pll3()

```
void kdrv_clock_mgr_open_pll3 (
            void  )
```

Here is the call graph for this function:

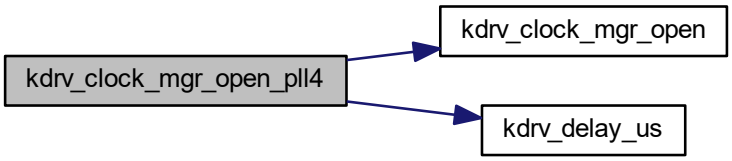


4.2.5.15 kdrv_clock_mgr_open_pll4()

```
void kdrv_clock_mgr_open_pll4 (
            void  )
```

Kneron Confidential

Here is the call graph for this function:



4.2.5.16 kdrv_clock_mgr_open_pll5()

```
void kdrv_clock_mgr_open_pll5 (
            void  )
```

4.2.5.17 kdrv_clock_mgr_set_muxsel()

```
void kdrv_clock_mgr_set_muxsel (
            uint32_t flags )
```

4.2.5.18 kdrv_clock_mgr_set_scuclkin()

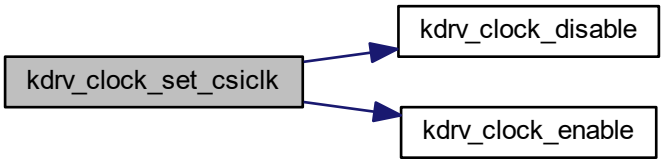
```
void kdrv_clock_mgr_set_scuclkin (
            enum scuclkin_type type,
            bool enable )
```

4.2.5.19 kdrv_clock_set_csiclk()

```
void kdrv_clock_set_csiclk (
            uint32_t cam_idx,
            uint32_t enable )
```

Kneron Confidential

Here is the call graph for this function:



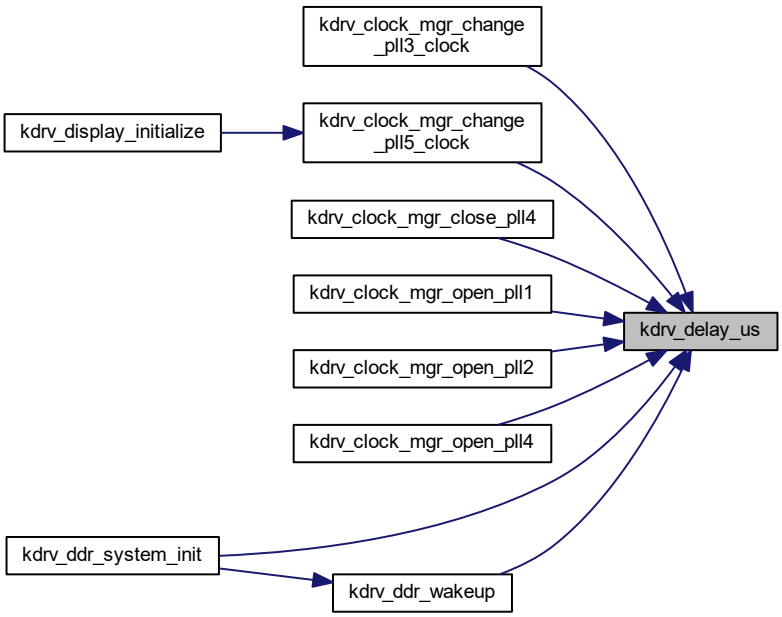
4.2.5.20 kdrv_debug_pll_clock()

```
void kdrv_debug_pll_clock (
            void  )
```

4.2.5.21 kdrv_delay_us()

```
void kdrv_delay_us (
            uint32_t usec )
```

Here is the caller graph for this function:



Kneron Confidential

4.2.6 Variable Documentation

4.2.6.1 child_head

```
struct kdrv_clock_node* kdrv_clock_node::child_head
```

4.2.6.2 child_next

```
struct kdrv_clock_node* kdrv_clock_node::child_next
```

4.2.6.3 clock_node_pll1_out

```
struct kdrv_clock_node clock_node_pll1_out
```

4.2.6.4 div

```
uint8_t kdrv_clock_value::div
```

4.2.6.5 enable

```
uint8_t kdrv_clock_value::enable
```

4.2.6.6 is_enabled

```
uint8_t kdrv_clock_node::is_enabled
```

4.2.6.7 ms

```
uint16_t kdrv_clock_value::ms
```

Kneron Confidential

4.2.6.8 name

```
char kdrv_clock_node::name[15]
```

4.2.6.9 next

```
struct kdrv_clock_list* kdrv_clock_list::next
```

4.2.6.10 ns

```
uint16_t kdrv_clock_value::ns
```

4.2.6.11 parent

```
struct kdrv_clock_node* kdrv_clock_node::parent
```

4.2.6.12 ps

```
uint16_t kdrv_clock_value::ps
```

4.2.6.13 set

```
fn_set kdrv_clock_node::set
```

Kneron Confidential

4.3 KDRV_DDR

Kneron generic DDR driver.

Enumerations

- enum kdrv_ddr_init_mode { DDR_INIT_WAKEUP_ONLY = 0, DDR_INIT_ALL, DDR_INIT_ALL_EXIT_SELF_REFRESH }

  Enumeration of ddr init mode.

Functions

- void kdrv_ddr_wakeup (void)

  DDR wakeup and de-assert reset of DDR controller.
- void kdrv_ddr_system_init (enum kdrv_ddr_init_mode mode)

  DDR initialize.
- void kdrv_ddr_self_refresh_enter (void)

  DDR enter self-refresh mode to save power.
- void kdrv_ddr_self_refresh_exit (void)

  DDR exit self-refresh mode to normal mode.

4.3.1 Detailed Description

Kneron generic DDR driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.3.2 Enumeration Type Documentation

4.3.2.1 kdrv_ddr_init_mode

`enum kdrv_ddr_init_mode`

Enumeration of ddr init mode.

Enumerator

DDR_INIT_WAKEUP_ONLY	Wake up DDR controller only.
DDR_INIT_ALL	DDRx SDRAM enters into normal mode, In the normal mode, all operations run at full speed.
DDR_INIT_ALL_EXIT_SELF_REFRESH	To exit from the self refresh mode by software, users need to ensure that DDRx SDRAM is in the self refresh mode before issuing an existing self refresh command by software(kdrv_ddr_system_init()).

Kneron Confidential

4.3.3 Function Documentation

4.3.3.1 kdrv_ddr_self_refresh_enter()

```
void kdrv_ddr_self_refresh_enter (
            void  )
```

DDR enter self-refresh mode to save power.

Returns

N/A

4.3.3.2 kdrv_ddr_self_refresh_exit()

```
void kdrv_ddr_self_refresh_exit (
            void  )
```

DDR exit self-refresh mode to normal mode.

Returns

N/A

4.3.3.3 kdrv_ddr_system_init()

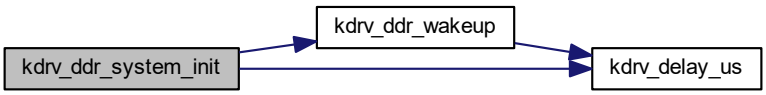
```
void kdrv_ddr_system_init (
            enum kdrv_ddr_init_mode mode )
```

DDR initialize.

Returns

N/A

Here is the call graph for this function:



Kneron Confidential

4.3.3.4 kdrv_ddr_wakeup()

```
void kdrv_ddr_wakeup (
             void  )
```

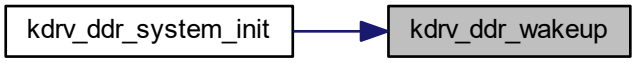
DDR wakeup and de-assert reset of DDR controller.

Returns

N/A

Here is the call graph for this function:

Here is the caller graph for this function:



Kneron Confidential

4.4 KDRV_DISPLAY

Kneron display interface for LCDC and LCM driver.

Data Structures

- struct kdrv_display_pen_info_t

  Enumerations of display pen setting.
- struct kdrv_display_t

  Enumerations of display driver setting.

Macros

- #define FRAME_SIZE_RGB(xres, yres, mbpp) ((xres) ∗ (yres) ∗ (mbpp) / 8)

Functions

- int kdrv_display_cal_framesize (unsigned short width, unsigned short height, unsigned int input_fmt)

  inline function to calculate frame size for display
- kdrv_display_t ∗ kdrv_display_initialize (void)

  Initialize display driver.
- kdrv_status_t kdrv_display_buffer_initialize (struct video_input_params ∗params)

  Initialize display frame buffer.
- uint32_t kdrv_display_get_buffer (void)
- kdrv_status_t kdrv_display_set_params (kdrv_display_t ∗display_drv, struct video_input_params ∗params)

  Set display parameters.
- kdrv_status_t kdrv_display_get_params (kdrv_display_t ∗display_drv, struct video_input_params ∗params)

  Get display parameters.
- kdrv_status_t kdrv_display_set_camera (kdrv_display_t ∗display_drv, uint8_t cam_idx)

  Set camera source which will be preview on display.
- kdrv_status_t kdrv_display_start (kdrv_display_t ∗display_drv)

  Start display image preview.
- kdrv_status_t kdrv_display_stop (kdrv_display_t ∗display_drv)

  Stop display image preview.
- kdrv_status_t kdrv_display_set_pen (kdrv_display_t ∗display_drv, uint16_t color, uint32_t width)

  Set pen setting.
- kdrv_status_t kdrv_display_update_draw_fb (kdrv_display_t ∗display_drv, uint32_t addr, uint8_t cam_idx)

  Update frame buffer which be used to draw something on display.
- kdrv_status_t kdrv_display_draw_static_rect (kdrv_display_t ∗display_drv, uint32_t org_x, uint32_t org_↩ y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_rect (kdrv_display_t ∗display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_line (kdrv_display_t ∗display_drv, uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- kdrv_status_t kdrv_display_fill_rect (kdrv_display_t ∗display_drv, uint32_t org_x, uint32_t org_y, uint32_↩ t width, uint32_t height)

Kneron Confidential

Draw rectangle with filling color on display.

- kdrv_status_t kdrv_display_draw_bitmap (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height, void *pBuf)

  Draw bitmap on display.

- kdrv_status_t kdrv_display_test_pattern_gen (kdrv_display_t *display_drv, bool pat_gen)

  Set display backlight.

- kdrv_status_t kdrv_display_set_backlight (kdrv_display_t *display_drv, int light)

  Set display backlight.

- kdrv_status_t kdrv_display_set_frame_margin_len (uint16_t margin_len)
- uint16_t kdrv_display_get_frame_margin_len (void)
- kdrv_status_t kdrv_display_set_review_snapshot_en (bool enable)
- bool kdrv_display_get_review_snapshot_en (void)
- kdrv_status_t kdrv_display_set_snapshot_preview_en (bool enable)
- bool kdrv_display_get_snapshot_preview_en (void)

4.4.1 Detailed Description

Kneron display interface for LCDC and LCM driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.4.2 Macro Definition Documentation

4.4.2.1 FRAME_SIZE_RGB

```
#define FRAME_SIZE_RGB(
            xres,
            yres,
            mbpp ) ((xres) * (yres) * (mbpp) / 8)
```

4.4.3 Function Documentation

4.4.3.1 kdrv_display_buffer_initialize()

```
kdrv_status_t kdrv_display_buffer_initialize (
            struct video_input_params * params )
```

Initialize display frame buffer.

Kneron Confidential

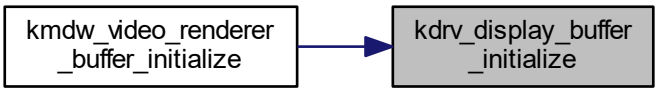
Parameters

in	*params*	see video_input_params

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.4.3.2 kdrv_display_cal_framesize()

```
int kdrv_display_cal_framesize (
            unsigned short width,
            unsigned short height,
            unsigned int input_fmt )  [inline]
```

inline function to calculate frame size for display

Parameters

in	*width*	Width of image
in	*height*	Height of image
in	*input_fmt*	Format of image

Returns

$>$ 0 Frame size 0 Wrong input image format

4.4.3.3 kdrv_display_draw_bitmap()

```
kdrv_status_t kdrv_display_draw_bitmap (
            kdrv_display_t * display_drv,
```

Kneron Confidential

```
            uint32_t org_x,
            uint32_t org_y,
            uint32_t width,
            uint32_t height,
            void * pBuf )
```

Draw bitmap on display.

Parameters

in	*display_drv*	see kdrv_display_t
in	*org_x*	Start x-axis of bitmap on display to draw
in	*org_y*	Start y-axis of bitmap on display to draw
in	*width*	Width of bitmap
in	*height*	Height of bitmap
in	*pBuf*	Bitmap of target

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.4.3.4 kdrv_display_draw_line()

```
kdrv_status_t kdrv_display_draw_line (
            kdrv_display_t * display_drv,
            uint32_t xs,
            uint32_t ys,
            uint32_t xe,
            uint32_t ye )
```

Draw line on display.

Parameters

in	*display_drv*	see kdrv_display_t
in	*xs*	Start x-axis of line on display to draw
in	*xe*	End x-axis of line on display to draw
in	*ys*	Start y-axis of line on display to draw
in	*ye*	End y-axis of line on display to draw

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.4.3.5 kdrv_display_draw_rect()

```
kdrv_status_t kdrv_display_draw_rect (
            kdrv_display_t * display_drv,
            uint32_t org_x,
            uint32_t org_y,
            uint32_t width,
            uint32_t height )
```

Draw rectangle without filling color on display.

Note

Mainly be used to draw fdfr result boundingbox on display

Parameters

in	*display_drv*	see kdrv_display_t
in	*org_x*	Start x-axis of rectangle on display to draw
in	*org_y*	Start y-axis of rectangle on display to draw
in	*width*	Width of rectangle
in	*height*	Height of rectangle

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.4.3.6 kdrv_display_draw_static_rect()

```
kdrv_status_t kdrv_display_draw_static_rect (
            kdrv_display_t * display_drv,
            uint32_t org_x,
            uint32_t org_y,
            uint32_t width,
            uint32_t height )
```

Draw rectangle without filling color on display.

Note

Mainly be used to draw hint boundingbox on display

Parameters

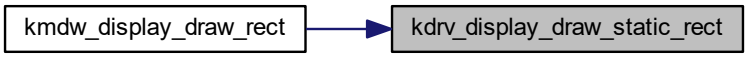
in	*display_drv*	see kdrv_display_t
in	*org_x*	Start x-axis of rectangle on display to draw
in	*org_y*	Start y-axis of rectangle on display to draw
in	*width*	Width of rectangle
in	*height*	Height of rectangle

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.4.3.7 kdrv_display_fill_rect()

```
kdrv_status_t kdrv_display_fill_rect (
            kdrv_display_t * display_drv,
            uint32_t org_x,
            uint32_t org_y,
            uint32_t width,
            uint32_t height )
```

Draw rectangle with filling color on display.

Parameters

in	*display_drv*	see kdrv_display_t
in	*org_x*	Start x-axis of rectangle on display to draw
in	*org_y*	Start y-axis of rectangle on display to draw
in	*width*	Width of rectangle
in	*height*	Height of rectangle

Returns

kdrv_status_t see kdrv_status_t

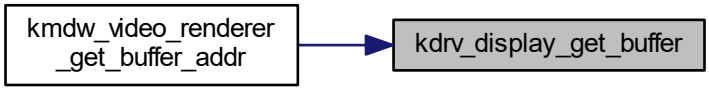
Here is the caller graph for this function:

Kneron Confidential

4.4.3.8 kdrv_display_get_buffer()

```
uint32_t kdrv_display_get_buffer (
            void  )
```

Here is the caller graph for this function:



4.4.3.9 kdrv_display_get_frame_margin_len()

```
uint16_t kdrv_display_get_frame_margin_len (
            void  )
```

4.4.3.10 kdrv_display_get_params()

```
kdrv_status_t kdrv_display_get_params (
            kdrv_display_t * display_drv,
            struct video_input_params * params )
```

Get display parameters.

Parameters

in	*display_drv*	see kdrv_display_t
out	*params*	see video_input_params

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.4.3.11 kdrv_display_get_review_snapshot_en()

```
bool kdrv_display_get_review_snapshot_en (
            void  )
```

4.4.3.12 kdrv_display_get_snapshot_preview_en()

```
bool kdrv_display_get_snapshot_preview_en (
            void  )
```

4.4.3.13 kdrv_display_initialize()

```
kdrv_display_t* kdrv_display_initialize (
            void  )
```

Initialize display driver.

Parameters

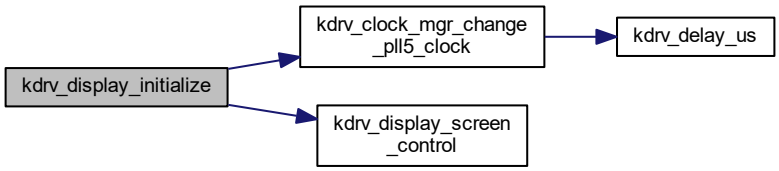
in	*N/A*	

Kneron Confidential

Returns

kdrv_display_t* see kdrv_display_t

Here is the call graph for this function:



Here is the caller graph for this function:

4.4.3.14 kdrv_display_set_backlight()

```
kdrv_status_t kdrv_display_set_backlight (
            kdrv_display_t * display_drv,
            int light )
```

Set display backlight.

Parameters

in	*display_drv*	see kdrv_display_t

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.4.3.15 kdrv_display_set_camera()

```
kdrv_status_t kdrv_display_set_camera (
            kdrv_display_t * display_drv,
            uint8_t cam_idx )
```

Set camera source which will be preview on display.

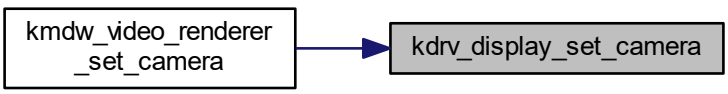
Parameters

in	*display_drv*	see kdrv_display_t
in	*cam_idx*	Camera source index

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



Kneron Confidential

4.4.3.16 kdrv_display_set_frame_margin_len()

```
kdrv_status_t kdrv_display_set_frame_margin_len (
            uint16_t margin_len )
```

4.4.3.17 kdrv_display_set_params()

```
kdrv_status_t kdrv_display_set_params (
            kdrv_display_t * display_drv,
            struct video_input_params * params )
```

Set display parameters.

Parameters

in	*display_drv*	see kdrv_display_t
out	*params*	see video_input_params

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.4.3.18 kdrv_display_set_pen()

```
kdrv_status_t kdrv_display_set_pen (
            kdrv_display_t * display_drv,
            uint16_t color,
            uint32_t width )
```

Set pen setting.

Parameters

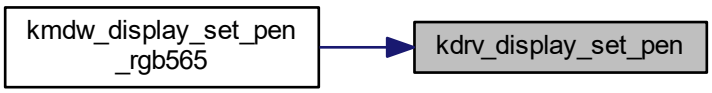
in	*display_drv*	see kdrv_display_t
in	*color*	Color of pen setting
in	*width*	Width of pen setting

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.4.3.19 kdrv_display_set_review_snapshot_en()

```
kdrv_status_t kdrv_display_set_review_snapshot_en (
            bool enable )
```

4.4.3.20 kdrv_display_set_snapshot_preview_en()

```
kdrv_status_t kdrv_display_set_snapshot_preview_en (
            bool enable )
```

4.4.3.21 kdrv_display_start()

```
kdrv_status_t kdrv_display_start (
            kdrv_display_t * display_drv )
```

Start display image preview.

Parameters

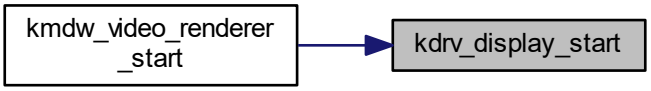
in	*display_drv*	see kdrv_display_t

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.4.3.22 kdrv_display_stop()

```
kdrv_status_t kdrv_display_stop (
            kdrv_display_t * display_drv )
```

Stop display image preview.

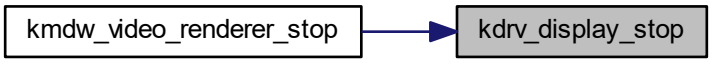
Parameters

in	*display_drv*	see kdrv_display_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



Kneron Confidential

4.4.3.23 kdrv_display_test_pattern_gen()

```
kdrv_status_t kdrv_display_test_pattern_gen (
            kdrv_display_t * display_drv,
            bool pat_gen )
```

Set display backlight.

Kneron Confidential

Parameters

in	*display_drv*	see kdrv_display_t
in	*pat_gen*	yes or no to generate displayu test pattern

Returns

kdrv_status_t see kdrv_status_t

Note

Exmpale:
kdrv_display_test_pattern_gen(true);
kmdw_video_renderer_open(params);

Here is the caller graph for this function:

4.4.3.24 kdrv_display_update_draw_fb()

```
kdrv_status_t kdrv_display_update_draw_fb (
            kdrv_display_t * display_drv,
            uint32_t addr,
            uint8_t cam_idx )
```

Update frame buffer which be used to draw something on display.

Parameters

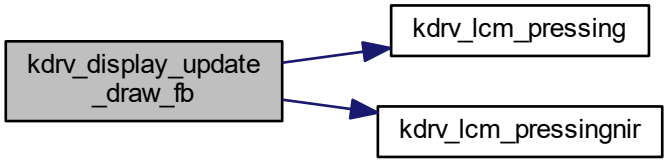
in	*display_drv*	see kdrv_display_t
in	*addr*	Frame buffer address
in	*cam_idx*	Camera source index

Kneron Confidential

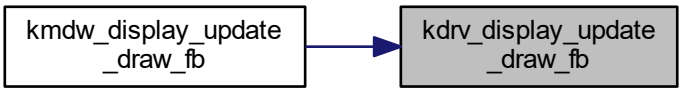
Returns

kdrv_status_t see kdrv_status_t

Here is the call graph for this function:



Here is the caller graph for this function:



Kneron Confidential

4.5 KDRV_DPI2AHB

Kneron dpi2ahb driver.

Macros

- #define TILE_BLOCK_MAX_W 10
- #define TILE_BLOCK_MAX_H 6
- #define TILE_BLOCKS_MAX (TILE_BLOCK_MAX_W * TILE_BLOCK_MAX_H)
- #define TILE_REGS_MAX (TILE_BLOCKS_MAX / 4)

Enumerations

- enum { D2A_CAM_0, D2A_CAM_1, D2A_CAM_NUM }

Functions

- kdrv_status_t kdrv_dpi2ahb_enable (uint32_t cam_idx, struct cam_format *fmt)

  kdrv_dpi2ahb_enable Enable dpi2ahb IP,
- kdrv_status_t kdrv_dpi2ahb_stop (uint32_t cam_idx)

  kdrv_dpi2ahb_stop Stop dpi2ahb interrup, disable IRQ.
- kdrv_status_t kdrv_dpi2ahb_start (uint32_t cam_idx)

  kdrv_dpi2ahb_start Start dpi2ahb interrup, enable IRQ.
- kdrv_status_t kdrv_dpi2ahb_buf_init (uint32_t cam_idx)

  kdrv_dpi2ahb_buf_init Set dpi2ahb page buffer default address.
- kdrv_status_t kdrv_dpi2ahb_initialize (uint32_t cam_idx)

  kdrv_dpi2ahb_initialize Init dpi2ahb IRQ and reset IP.

4.5.1 Detailed Description

Kneron dpi2ahb driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.5.2 Macro Definition Documentation

4.5.2.1 TILE_BLOCK_MAX_H

```
#define TILE_BLOCK_MAX_H 6
```

Kneron Confidential

4.5.2.2 TILE_BLOCK_MAX_W

```
#define TILE_BLOCK_MAX_W 10
```

4.5.2.3 TILE_BLOCKS_MAX

```
#define TILE_BLOCKS_MAX (TILE_BLOCK_MAX_W * TILE_BLOCK_MAX_H)
```

4.5.2.4 TILE_REGS_MAX

```
#define TILE_REGS_MAX (TILE_BLOCKS_MAX / 4)
```

4.5.3 Enumeration Type Documentation

4.5.3.1 anonymous enum

```
anonymous enum
```

Enumerator

D2A_CAM_0	
D2A_CAM_1	
D2A_CAM_NUM	

4.5.4 Function Documentation

4.5.4.1 kdrv_dpi2ahb_buf_init()

```
kdrv_status_t kdrv_dpi2ahb_buf_init (
            uint32_t cam_idx )
```

kdrv_dpi2ahb_buf_init Set dpi2ahb page buffer default address.

Parameters

in	*cam_idx*	cam_idx

Kneron Confidential

Returns

kdrv_status_t

4.5.4.2 kdrv_dpi2ahb_enable()

```
kdrv_status_t kdrv_dpi2ahb_enable (
            uint32_t cam_idx,
            struct cam_format * fmt )
```

kdrv_dpi2ahb_enable Enable dpi2ahb IP,

Parameters

in	*cam_idx*	cam_idx
in	*fmt*	camera related format setting

Returns

kdrv_status_t

4.5.4.3 kdrv_dpi2ahb_initialize()

```
kdrv_status_t kdrv_dpi2ahb_initialize (
            uint32_t cam_idx )
```

kdrv_dpi2ahb_initialize Init dpi2ahb IRQ and reset IP.

Parameters

in	*cam_idx*	cam_idx

Returns

kdrv_status_t

4.5.4.4 kdrv_dpi2ahb_start()

```
kdrv_status_t kdrv_dpi2ahb_start (
            uint32_t cam_idx )
```

kdrv_dpi2ahb_start Start dpi2ahb interrup, enable IRQ.

Kneron Confidential

Parameters

in	*cam_idx*	cam_idx

Returns

kdrv_status_t

4.5.4.5 kdrv_dpi2ahb_stop()

```
kdrv_status_t kdrv_dpi2ahb_stop (
            uint32_t cam_idx )
```

kdrv_dpi2ahb_stop Stop dpi2ahb interrup, disable IRQ.

Parameters

in	*cam_idx*	cam_idx

Returns

kdrv_status_t

Kneron Confidential

4.6 KDRV_FB_MGR

Kneron mipicsirx driver.

Macros

- #define MAX_FRAME_BUFFER 7
- #define DISPLAY_SRC_READ 0
- #define DISPLAY_SRC_INF 1
- #define DISPLAY_SRC_WRITE 2

Typedefs

- typedef int(∗ fb_write_done_notify) (int cam_idx, int write_idx)

Functions

- int kdrv_fb_mgr_init (int cam_idx, uint32_t buf_size, int buf_num, int frame_info_size)

  To initialize frame buffer pool for camera.
- uint32_t kdrv_fb_mgr_next_write (int cam_idx, int ∗write_idx)

  To get next available buffer for camera to write.
- int kdrv_fb_mgr_write_done (int cam_idx, int write_idx)

  To notify the finish of a buffer written by camera.
- void kdrv_fb_mgr_free_write_buf (int cam_idx)

  To free the use of frame buffers for camera.
- uint32_t kdrv_fb_mgr_get_buf (int cam_idx, int idx)

  To get a buffer of an index.
- uint32_t kdrv_fb_mgr_get_frame_info_buf (int cam_idx, int idx)

  To get a buffer of an index.
- uint32_t kdrv_fb_mgr_get_buf_seq_num (int cam_idx, int idx)

  To get the (camera write) sequence number of a buffer.
- uint32_t kdrv_fb_mgr_get_current_buf_seq_num (int cam_idx)

  To get current buffer sequence number.
- uint32_t kdrv_fb_mgr_next_read (int cam_idx, int ∗read_idx)

  To get the next buffer for drawing.
- int kdrv_fb_mgr_read_done (int cam_idx, int read_idx)

  To notify the finish of a buffer done for drawing.
- void kdrv_fb_mgr_free_read_buf (int cam_idx)

  To free the use of frame buffers for drawing.
- uint32_t kdrv_fb_mgr_next_inf (int cam_idx, int ∗inf_idx)

  To get the next buffer for inference.
- int kdrv_fb_mgr_inf_done (int cam_idx, int inf_idx)

  To notify the finish of a buffer done for inference.
- void kdrv_fb_mgr_free_inf_buf (int cam_idx)

  To free the use of frame buffers for inference.
- uint32_t kdrv_fb_mgr_next_display (int cam_idx, int ∗disp_idx)

  To get the buffer for display.
- int kdrv_fb_mgr_display_set_src (int cam_idx, int disp_src)

  To set display buffer source.
- int kdrv_fb_mgr_notifier_register (int cam_idx, fb_write_done_notify callback)

  To register a callback for a finished frame by camera.

Kneron Confidential

4.6.1 Detailed Description

Kneron mipicsirx driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.6.2 Macro Definition Documentation

4.6.2.1 DISPLAY_SRC_INF

```
#define DISPLAY_SRC_INF 1
```

4.6.2.2 DISPLAY_SRC_READ

```
#define DISPLAY_SRC_READ 0
```

4.6.2.3 DISPLAY_SRC_WRITE

```
#define DISPLAY_SRC_WRITE 2
```

4.6.2.4 MAX_FRAME_BUFFER

```
#define MAX_FRAME_BUFFER 7
```

4.6.3 Typedef Documentation

4.6.3.1 fb_write_done_notify

```
typedef int(* fb_write_done_notify) (int cam_idx, int write_idx)
```

4.6.4 Function Documentation

4.6.4.1 kdrv_fb_mgr_display_set_src()

```
int kdrv_fb_mgr_display_set_src (
            int cam_idx,
            int disp_src )
```

To set display buffer source.

Kneron Confidential

Parameters

in	*cam_idx*	camera index
in	*disp_src*	one of display sources from read(drawing)/inference/write

Returns

0 for success, or -1 for error

4.6.4.2 kdrv_fb_mgr_free_inf_buf()

```
void kdrv_fb_mgr_free_inf_buf (
            int cam_idx )
```

To free the use of frame buffers for inference.

Parameters

in	*cam_idx*	camera index

Returns

0 for success, or -1 for error

4.6.4.3 kdrv_fb_mgr_free_read_buf()

```
void kdrv_fb_mgr_free_read_buf (
            int cam_idx )
```

To free the use of frame buffers for drawing.

Parameters

in	*cam_idx*	camera index

Returns

0 for success, or -1 for error

4.6.4.4 kdrv_fb_mgr_free_write_buf()

```
void kdrv_fb_mgr_free_write_buf (
            int cam_idx )
```

Kneron Confidential

To free the use of frame buffers for camera.

Parameters

in	*cam_idx*	camera index

Returns

0 for success, or -1 for error

4.6.4.5 kdrv_fb_mgr_get_buf()

```
uint32_t kdrv_fb_mgr_get_buf (
            int cam_idx,
            int idx )
```

To get a buffer of an index.

Parameters

in	*cam_idx*	camera index
in	*idx*	a buffer index

Returns

the buffer address of the idx

4.6.4.6 kdrv_fb_mgr_get_buf_seq_num()

```
uint32_t kdrv_fb_mgr_get_buf_seq_num (
            int cam_idx,
            int idx )
```

To get the (camera write) sequence number of a buffer.

Parameters

in	*cam_idx*	camera index
in	*idx*	a buffer index

Returns

the sequence number starting from 0

Kneron Confidential

4.6.4.7 kdrv_fb_mgr_get_current_buf_seq_num()

```
uint32_t kdrv_fb_mgr_get_current_buf_seq_num (
            int cam_idx )
```

To get current buffer sequence number.

Parameters

in	*cam_idx*	camera index

Returns

the sequence number starting from 0

4.6.4.8 kdrv_fb_mgr_get_frame_info_buf()

```
uint32_t kdrv_fb_mgr_get_frame_info_buf (
            int cam_idx,
            int idx )
```

To get a buffer of an index.

Parameters

in	*cam_idx*	camera index
in	*idx*	a buffer index

Returns

the frame info buffer address of the idx

4.6.4.9 kdrv_fb_mgr_inf_done()

```
int kdrv_fb_mgr_inf_done (
            int cam_idx,
            int inf_idx )
```

To notify the finish of a buffer done for inference.

Parameters

in	*cam_idx*	camera index
in	*inf_idx*	the buffer index done for inference

Kneron Confidential

Returns

0 for success, or -1 for error

4.6.4.10 kdrv_fb_mgr_init()

```
int kdrv_fb_mgr_init (
            int cam_idx,
            uint32_t buf_size,
            int buf_num,
            int frame_info_size )
```

To initialize frame buffer pool for camera.

Parameters

in	*cam_idx*	camera index
in	*buf_size*	buffer size
in	*buf_num*	buffer number
in	*frame_info_size*	frame info size (e.g. tile average)

Returns

0 for success, or -1 for error

4.6.4.11 kdrv_fb_mgr_next_display()

```
uint32_t kdrv_fb_mgr_next_display (
            int cam_idx,
            int * disp_idx )
```

To get the buffer for display.

Parameters

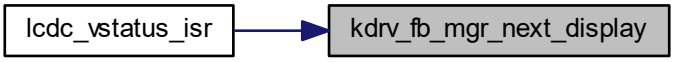
in	*cam_idx*	camera index
in	*disp_idx*	an address/pointer to get the buffer index

Kneron Confidential

Returns

the buffer address

Here is the caller graph for this function:



4.6.4.12 kdrv_fb_mgr_next_inf()

```
uint32_t kdrv_fb_mgr_next_inf (
            int cam_idx,
            int * inf_idx )
```

To get the next buffer for inference.

Parameters

in	*cam_idx*	camera index
in	*inf_idx*	an address/pointer to get the buffer index

Returns

the buffer address

4.6.4.13 kdrv_fb_mgr_next_read()

```
uint32_t kdrv_fb_mgr_next_read (
            int cam_idx,
            int * read_idx )
```

To get the next buffer for drawing.

Parameters

in	*cam_idx*	camera index
in	*read_idx*	an address/pointer to get the buffer index

Kneron Confidential

Returns

the buffer address

4.6.4.14 kdrv_fb_mgr_next_write()

```
uint32_t kdrv_fb_mgr_next_write (
            int cam_idx,
            int * write_idx )
```

To get next available buffer for camera to write.

Parameters

in	*cam_idx*	camera index
in	*write_idx*	an address/pointer to get the buffer index

Returns

the address of available buffer

4.6.4.15 kdrv_fb_mgr_notifier_register()

```
int kdrv_fb_mgr_notifier_register (
            int cam_idx,
            fb_write_done_notify callback )
```

To register a callback for a finished frame by camera.

Parameters

in	*cam_idx*	camera index
in	*callback*	the callback function when a frame buffer is done written

Returns

0 for success, or -1 for error

4.6.4.16 kdrv_fb_mgr_read_done()

```
int kdrv_fb_mgr_read_done (
            int cam_idx,
            int read_idx )
```

To notify the finish of a buffer done for drawing.

Kneron Confidential

Parameters

in	*cam_idx*	camera index
in	*read_idx*	the buffer index done for drawing

Returns

0 for success, or -1 for error

4.6.4.17 kdrv_fb_mgr_write_done()

```
int kdrv_fb_mgr_write_done (
            int cam_idx,
            int write_idx )
```

To notify the finish of a buffer written by camera.

Parameters

in	*cam_idx*	camera index
in	*write_idx*	the buffer index of write-done

Returns

0 for success, or -1 for error

Kneron Confidential

4.7 KDRV_GDMA

Kneron generic DMA driver.

Data Structures

- struct gdma_setting_t

  Structure of GDMA advanced settings for a specified DMA handle (channel)

Typedefs

- typedef int32_t kdrv_gdma_handle_t

  GDMA handle type which represents for a DMA channel and related DMA operations.
- typedef void(∗ gdma_xfer_callback_t) (kdrv_status_t status, void ∗arg)

  GDMA user callback function with transfer status notification. Note that this is callback form ISR context.

Enumerations

- enum gdma_transfer_width_t { GDMA_TXFER_WIDTH_8_BITS = 0x0, GDMA_TXFER_WIDTH_16_BITS, GDMA_TXFER_WIDTH_32_BITS }

  Enumeration of GDMA transfer size: 8/16/32 bits, this is about byte-alignment.
- enum gdma_burst_size_t { GDMA_BURST_SIZE_1 = 0x0, GDMA_BURST_SIZE_4, GDMA_BURST_SIZE_8, GDMA_BURST_SIZE_16, GDMA_BURST_SIZE_32, GDMA_BURST_SIZE_64, GDMA_BURST_SIZE_128, GDMA_BURST_SIZE_256 }

  Enumeration of GDMA transfer burst : 1/4/8/16/32/64/128/256, this is about performance.
- enum gdma_address_control_t { GDMA_INCREMENT_ADDRESS = 0x0, GDMA_DECREMENT_ADDRESS, GDMA_FIXED_ADDRESS }

  Enumeration of DMA address control, auto-increasing/descreading or fixed.
- enum gdma_work_mode_t { GDMA_NORMAL_MODE = 0x0, GDMA_HW_HANDSHAKE_MODE }

  Enumeration of DMA working mode, can be normal or hardware handshake mode.

Functions

- kdrv_status_t kdrv_gdma_initialize (void)

  GDMA driver initialization.
- kdrv_status_t kdrv_gdma_uninitialize (void)

  GDMA driver uninitialization.
- kdrv_status_t kdrv_gdma_acquire_handle (kdrv_gdma_handle_t ∗handle)

  Acquire a GDMA handle.
- kdrv_status_t kdrv_gdma_configure_setting (kdrv_gdma_handle_t handle, gdma_setting_t ∗dma_setting)

  Configure the DMA working behavior on specified DMA handle with specified dma settings.
- kdrv_status_t kdrv_gdma_release_handle (kdrv_gdma_handle_t handle)

  Release the DMA handle.
- kdrv_status_t kdrv_gdma_transfer_async (kdrv_gdma_handle_t handle, uint32_t dst_addr, uint32_t src_↩ addr, uint32_t num_bytes, gdma_xfer_callback_t xfer_isr_cb, void ∗usr_arg)

  Start DMA transfer with specified DMA handle running in asynchronous (non-blocking) mode.
- kdrv_status_t kdrv_gdma_transfer (kdrv_gdma_handle_t handle, uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes)

  Start DMA transfer with specified DMA handle running in synchronous (blocking) mode.
- kdrv_status_t kdrv_gdma_memcpy_async (uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes, gdma_xfer_callback_t xfer_isr_cb, void ∗usr_arg)

  Start DMA transfer with automatic DMA handle running in asynchronous (non-blocking) mode.
- kdrv_status_t kdrv_gdma_memcpy (uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes)

  Start DMA transfer with automatic DMA handle running in synchronous (blocking) mode.

Kneron Confidential

Variables

- gdma_transfer_width_t gdma_setting_t::dst_width
- gdma_transfer_width_t gdma_setting_t::src_width
- gdma_burst_size_t gdma_setting_t::burst_size
- gdma_address_control_t gdma_setting_t::dst_addr_ctrl
- gdma_address_control_t gdma_setting_t::src_addr_ctrl
- gdma_work_mode_t gdma_setting_t::dma_mode
- uint32_t gdma_setting_t::dma_dst_req
- uint32_t gdma_setting_t::dma_src_req

4.7.1 Detailed Description

Kneron generic DMA driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.7.2 Typedef Documentation

4.7.2.1 gdma_xfer_callback_t

```
typedef void(* gdma_xfer_callback_t) (kdrv_status_t status, void *arg)
```

GDMA user callback function with transfer status notification. Note that this is callback form ISR context.

4.7.2.2 kdrv_gdma_handle_t

```
typedef int32_t kdrv_gdma_handle_t
```

GDMA handle type which represents for a DMA channel and related DMA operations.

4.7.3 Enumeration Type Documentation

4.7.3.1 gdma_address_control_t

```
enum gdma_address_control_t
```

Enumeration of DMA address control, auto-increasing/descreading or fixed.

Kneron Confidential

Enumerator

GDMA_INCREMENT_ADDRESS	DMA address control, auto-increasing, default value
GDMA_DECREMENT_ADDRESS	DMA address control, auto-descreading
GDMA_FIXED_ADDRESS	DMA address control, fixed

4.7.3.2 gdma_burst_size_t

`enum gdma_burst_size_t`

Enumeration of GDMA transfer burst : 1/4/8/16/32/64/128/256, this is about performance.

Enumerator

GDMA_BURST_SIZE_1	GDMA transfer burst size: 1
GDMA_BURST_SIZE_4	GDMA transfer burst size: 4
GDMA_BURST_SIZE_8	GDMA transfer burst size: 8
GDMA_BURST_SIZE_16	GDMA transfer burst size: 16, default value
GDMA_BURST_SIZE_32	GDMA transfer burst size: 32
GDMA_BURST_SIZE_64	GDMA transfer burst size: 64
GDMA_BURST_SIZE_128	GDMA transfer burst size: 128
GDMA_BURST_SIZE_256	GDMA transfer burst size: 256

4.7.3.3 gdma_transfer_width_t

`enum gdma_transfer_width_t`

Enumeration of GDMA transfer size: 8/16/32 bits, this is about byte-alignment.

Enumerator

GDMA_TXFER_WIDTH_8_BITS	GDMA transfer size: 8 bits
GDMA_TXFER_WIDTH_16_BITS	GDMA transfer size: 16 bits
GDMA_TXFER_WIDTH_32_BITS	GDMA transfer size: 32 bits, default value

4.7.3.4 gdma_work_mode_t

`enum gdma_work_mode_t`

Enumeration of DMA working mode, can be normal or hardware handshake mode.

Kneron Confidential

Enumerator

GDMA_NORMAL_MODE	DMA working mode, normal mode , default value
GDMA_HW_HANDSHAKE_MODE	DMA working mode, hardware handshake mode

4.7.4 Function Documentation

4.7.4.1 kdrv_gdma_acquire_handle()

```
kdrv_status_t kdrv_gdma_acquire_handle (
            kdrv_gdma_handle_t * handle )
```

Acquire a GDMA handle.

Parameters

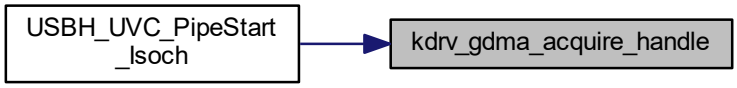
out	*handle*	a handle of a DMA channel, see kdrv_gdma_handle_t

Returns

kdrv_status_t see kdrv_status_t

```
Example:\n
kdrv_gdma_handle_t dma_handle;
kdrv_status_t sts = kdrv_gdma_acquire_handle(&dma_handle);
if(sts == KDRV_STATUS_OK) printf("Succeeds to get valid dma_handle");
```

Here is the caller graph for this function:



4.7.4.2 kdrv_gdma_configure_setting()

```
kdrv_status_t kdrv_gdma_configure_setting (
            kdrv_gdma_handle_t handle,
            gdma_setting_t * dma_setting )
```

Configure the DMA working behavior on specified DMA handle with specified dma settings.

Kneron Confidential

Parameters

in	*handle*	a handle of a DMA channel, see kdrv_gdma_handle_t
in	*dma_setting*	pointer of dma_setting, see gdma_setting_t

Returns

kdrv_status_t

Note

Before call this API, you should get a valid dma_handle via kdrv_gdma_acquire_handle() firstly.

4.7.4.3 kdrv_gdma_initialize()

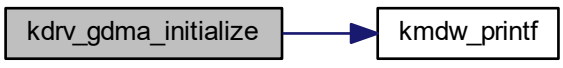
```
kdrv_status_t kdrv_gdma_initialize (
            void  )
```

GDMA driver initialization.

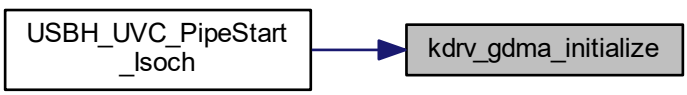
Returns

kdrv_status_t see kdrv_status_t

Here is the call graph for this function:



Here is the caller graph for this function:



Kneron Confidential

4.7.4.4 kdrv_gdma_memcpy()

```
kdrv_status_t kdrv_gdma_memcpy (
            uint32_t dst_addr,
            uint32_t src_addr,
            uint32_t num_bytes )
```

Start DMA transfer with automatic DMA handle running in synchronous (blocking) mode.

Parameters

in	*dst_addr*	destination address
in	*src_addr*	source address
in	*num_bytes*	number of bytes to be transfered

Returns

kdrv_status_t

4.7.4.5 kdrv_gdma_memcpy_async()

```
kdrv_status_t kdrv_gdma_memcpy_async (
            uint32_t dst_addr,
            uint32_t src_addr,
            uint32_t num_bytes,
            gdma_xfer_callback_t xfer_isr_cb,
            void * usr_arg )
```

Start DMA transfer with automatic DMA handle running in asynchronous (non-blocking) mode.

Parameters

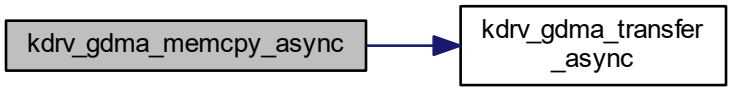
in	*dst_addr*	destination address
in	*src_addr*	source address
in	*num_bytes*	number of bytes to be transfered
in	*xfer_isr_cb*	user callback function, see gdma_xfer_callback_t
in	*usr_arg*	user's own argument

Kneron Confidential

Returns

kdrv_status_t

Here is the call graph for this function:



4.7.4.6 kdrv_gdma_release_handle()

```
kdrv_status_t kdrv_gdma_release_handle (
            kdrv_gdma_handle_t handle )
```

Release the DMA handle.

Parameters

in	*handle*	a handle of a DMA channel, see kdrv_gdma_handle_t

Returns

kdrv_status_t

Here is the caller graph for this function:

4.7.4.7 kdrv_gdma_transfer()

```
kdrv_status_t kdrv_gdma_transfer (
            kdrv_gdma_handle_t handle,
```

Kneron Confidential

```
            uint32_t dst_addr,
            uint32_t src_addr,
            uint32_t num_bytes )
```

Start DMA transfer with specified DMA handle running in synchronous (blocking) mode.

Parameters

in	*handle*	a handle of a DMA channel, see kdrv_gdma_handle_t
in	*dst_addr*	destination address
in	*src_addr*	source address
in	*num_bytes*	number of bytes to be transfered

Returns

kdrv_status_t

4.7.4.8 kdrv_gdma_transfer_async()

```
kdrv_status_t kdrv_gdma_transfer_async (
            kdrv_gdma_handle_t handle,
            uint32_t dst_addr,
            uint32_t src_addr,
            uint32_t num_bytes,
            gdma_xfer_callback_t xfer_isr_cb,
            void * usr_arg )
```

Start DMA transfer with specified DMA handle running in asynchronous (non-blocking) mode.

Parameters

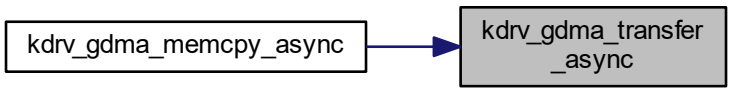
in	*handle*	a handle of a DMA channel, see kdrv_gdma_handle_t
in	*dst_addr*	destination address
in	*src_addr*	source address
in	*num_bytes*	number of bytes to be transfered
in	*xfer_isr_cb*	user callback function, see gdma_xfer_callback_t
in	*usr_arg*	user's argument

Kneron Confidential

Returns

kdrv_status_t

Here is the caller graph for this function:



4.7.4.9 kdrv_gdma_uninitialize()

```
kdrv_status_t kdrv_gdma_uninitialize (
            void  )
```

GDMA driver uninitialization.

Returns

kdrv_status_t see kdrv_status_t

4.7.5 Variable Documentation

4.7.5.1 burst_size

```
gdma_burst_size_t gdma_setting_t::burst_size
```

see gdma_burst_size_t

4.7.5.2 dma_dst_req

```
uint32_t gdma_setting_t::dma_dst_req
```

for HW handshake mode, refer to kneron_mozart.h XXX_DMA_REQ

4.7.5.3 dma_mode

```
gdma_work_mode_t gdma_setting_t::dma_mode
```

see gdma_work_mode_t

Kneron Confidential

4.7.5.4 dma_src_req

```
uint32_t gdma_setting_t::dma_src_req
```

for HW handshake mode, refer to kneron_mozart.h XXX_DMA_REQ

4.7.5.5 dst_addr_ctrl

```
gdma_address_control_t gdma_setting_t::dst_addr_ctrl
```

see gdma_address_control_t

4.7.5.6 dst_width

```
gdma_transfer_width_t gdma_setting_t::dst_width
```

see gdma_transfer_width_t

4.7.5.7 src_addr_ctrl

```
gdma_address_control_t gdma_setting_t::src_addr_ctrl
```

see gdma_address_control_t

4.7.5.8 src_width

```
gdma_transfer_width_t gdma_setting_t::src_width
```

see gdma_transfer_width_t

Kneron Confidential

4.8 KDRV_GPIO

Kneron GPIO driver.

Typedefs

- typedef void(∗ gpio_interrupt_callback_t) (kdrv_gpio_pin_t pin, void ∗arg)

Enumerations

- enum kdrv_gpio_attribute_t {
  GPIO_DIR_INPUT = 0x1, GPIO_DIR_OUTPUT = 0x2, GPIO_INT_EDGE_RISING = 0x4, GPIO_INT_EDGE_FALLING = 0x8,
  GPIO_INT_EDGE_BOTH = 0x10, GPIO_INT_LEVEL_HIGH = 0x20, GPIO_INT_LEVEL_LOW = 0x40 }

  Enumerations of GPIO pin attributes, input or output, interrupt trigger settings.
- enum kdrv_gpio_pin_t {
  GPIO_PIN_0 = 0, GPIO_PIN_1, GPIO_PIN_2, GPIO_PIN_3,
  GPIO_PIN_4, GPIO_PIN_5, GPIO_PIN_6, GPIO_PIN_7,
  GPIO_PIN_8, GPIO_PIN_9, GPIO_PIN_10, GPIO_PIN_11,
  GPIO_PIN_12, GPIO_PIN_13, GPIO_PIN_14, GPIO_PIN_15,
  GPIO_PIN_16, GPIO_PIN_17, GPIO_PIN_18, GPIO_PIN_19,
  GPIO_PIN_20, GPIO_PIN_21, GPIO_PIN_22, GPIO_PIN_23,
  GPIO_PIN_24, GPIO_PIN_25, GPIO_PIN_26, GPIO_PIN_27,
  GPIO_PIN_28, GPIO_PIN_29, GPIO_PIN_30, GPIO_PIN_31 }

  Enumerations of GPIO pin ID, there 32 GPIO pins.

Functions

- kdrv_status_t kdrv_gpio_initialize (void)

  GPIO driver initialization, this must be invoked once before any GPIO manipulations.
- kdrv_status_t kdrv_gpio_uninitialize (void)

  GPIO driver uninitialization.
- kdrv_status_t kdrv_gpio_set_attribute (kdrv_gpio_pin_t pin, uint32_t attributes)

  set pin attributes for a specified GPIO pin
- kdrv_status_t kdrv_gpio_register_callback (gpio_interrupt_callback_t gpio_isr_cb, void ∗usr_arg)

  register user callback with user argument for GPIO interrupt in this callback can get interrupts for all GPIO pins
- kdrv_status_t kdrv_gpio_set_interrupt (kdrv_gpio_pin_t pin, bool isEnable)

  set interrupt enable/disable for a specified GPIO pin
- kdrv_status_t kdrv_gpio_set_debounce (kdrv_gpio_pin_t pin, bool isEnable, uint32_t debounce_clock)

  set debounce enable/disable with clock setting in Hz
- kdrv_status_t kdrv_gpio_write_pin (kdrv_gpio_pin_t pin, bool value)

  write GPIO digitial pin value
- kdrv_status_t kdrv_gpio_read_pin (kdrv_gpio_pin_t pin, bool ∗pValue)

  read GPIO digitial pin value

4.8.1 Detailed Description

Kneron GPIO driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.8.2 Typedef Documentation

4.8.2.1 gpio_interrupt_callback_t

```
typedef void(* gpio_interrupt_callback_t) (kdrv_gpio_pin_t pin, void *arg)
```

GPIO user callback function with specified GPIO pin. Note that this is callback form ISR context.

4.8.3 Enumeration Type Documentation

4.8.3.1 kdrv_gpio_attribute_t

```
enum kdrv_gpio_attribute_t
```

Enumerations of GPIO pin attributes, input or output, interrupt trigger settings.

Enumerator

GPIO_DIR_INPUT	pin direction digital input
GPIO_DIR_OUTPUT	pin direction digital output
GPIO_INT_EDGE_RISING	indicate pin interrupt triggered when at rising edge
GPIO_INT_EDGE_FALLING	indicate pin interrupt triggered when at falling edge
GPIO_INT_EDGE_BOTH	indicate pin interrupt triggered when at both edge rsising or falling
GPIO_INT_LEVEL_HIGH	indicate pin interrupt triggered when at high voltage level
GPIO_INT_LEVEL_LOW	indicate pin interrupt triggered when at low voltage level

4.8.3.2 kdrv_gpio_pin_t

```
enum kdrv_gpio_pin_t
```

Enumerations of GPIO pin ID, there 32 GPIO pins.

Enumerator

GPIO_PIN_0	GPIO pin ID 0
GPIO_PIN_1	GPIO pin ID 1
GPIO_PIN_2	GPIO pin ID 2
GPIO_PIN_3	GPIO pin ID 3
GPIO_PIN_4	GPIO pin ID 4
GPIO_PIN_5	GPIO pin ID 5

Kneron Confidential

Enumerator

GPIO_PIN_6	GPIO pin ID 6
GPIO_PIN_7	GPIO pin ID 7
GPIO_PIN_8	GPIO pin ID 8
GPIO_PIN_9	GPIO pin ID 9
GPIO_PIN_10	GPIO pin ID 10
GPIO_PIN_11	GPIO pin ID 11
GPIO_PIN_12	GPIO pin ID 12
GPIO_PIN_13	GPIO pin ID 13
GPIO_PIN_14	GPIO pin ID 14
GPIO_PIN_15	GPIO pin ID 15
GPIO_PIN_16	GPIO pin ID 16
GPIO_PIN_17	GPIO pin ID 17
GPIO_PIN_18	GPIO pin ID 18
GPIO_PIN_19	GPIO pin ID 19
GPIO_PIN_20	GPIO pin ID 20
GPIO_PIN_21	GPIO pin ID 21
GPIO_PIN_22	GPIO pin ID 22
GPIO_PIN_23	GPIO pin ID 23
GPIO_PIN_24	GPIO pin ID 24
GPIO_PIN_25	GPIO pin ID 25
GPIO_PIN_26	GPIO pin ID 26
GPIO_PIN_27	GPIO pin ID 27
GPIO_PIN_28	GPIO pin ID 28
GPIO_PIN_29	GPIO pin ID 29
GPIO_PIN_30	GPIO pin ID 30
GPIO_PIN_31	GPIO pin ID 31

4.8.4 Function Documentation

4.8.4.1 kdrv_gpio_initialize()

```
kdrv_status_t kdrv_gpio_initialize (
            void  )
```

GPIO driver initialization, this must be invoked once before any GPIO manipulations.

Returns

KDRV_STATUS_OK only

Kneron Confidential

4.8.4.2 kdrv_gpio_read_pin()

```
kdrv_status_t kdrv_gpio_read_pin (
            kdrv_gpio_pin_t pin,
            bool * pValue )
```

read GPIO digitial pin value

This function read a high or low value from a digital pin.
The specified pin must be configured as digital input and not in interrupt mode.

Parameters

in	*pin*	GPIO pin ID, see kdrv_gpio_pin_t
out	*pValue*	pointer to a value to read out GPIO voltage level

Returns

KDRV_STATUS_OK only

4.8.4.3 kdrv_gpio_register_callback()

```
kdrv_status_t kdrv_gpio_register_callback (
            gpio_interrupt_callback_t gpio_isr_cb,
            void * usr_arg )
```

register user callback with user argument for GPIO interrupt in this callback can get interrupts for all GPIO pins

Parameters

in	*gpio_isr_cb*	user callback function for GPIO interrupts, see gpio_interrupt_callback_t
in	*usr_arg*	user's argument

Returns

KDRV_STATUS_OK only

4.8.4.4 kdrv_gpio_set_attribute()

```
kdrv_status_t kdrv_gpio_set_attribute (
            kdrv_gpio_pin_t pin,
            uint32_t attributes )
```

set pin attributes for a specified GPIO pin

it must be well set up before GPIO pin to be used.

Kneron Confidential

Parameters

in	*pin*	After configuring the desired pin as a GPIO pin, the corresponding GPIO pin name should be used as kdp_gpio_pin_e indicated
in	*attributes*	This is to specify the function of specified GPIO pin, for digital output, set only DIR_OUTPUT, for digital input for read, set only DIR_INPUT, for interrupt usage, set DIR_INPUT and one of EDGE or LEVEL trigger attributes, this implies pin is used as an interrupt input

Returns

KDRV_STATUS_OK only

4.8.4.5 kdrv_gpio_set_debounce()

```
kdrv_status_t kdrv_gpio_set_debounce (
            kdrv_gpio_pin_t pin,
            bool isEnable,
            uint32_t debounce_clock )
```

set debounce enable/disable with clock setting in Hz

This can enable internal debouncing hardware for interrupt mode to eliminate the switch bounce.
It is very useful for connecting devices like a switch button or a keypad thing.

Parameters

in	*pin*	GPIO pin ID, see kdrv_gpio_pin_t
in	*isEnable*	enable/disable
in	*debounce_clock*	The debouncing clock frequency in Hz

Returns

KDRV_STATUS_OK only

4.8.4.6 kdrv_gpio_set_interrupt()

```
kdrv_status_t kdrv_gpio_set_interrupt (
            kdrv_gpio_pin_t pin,
            bool isEnable )
```

set interrupt enable/disable for a specified GPIO pin

Kneron Confidential

Parameters

in	*pin*	GPIO pin ID, see kdrv_gpio_pin_t
in	*isEnable*	enable/disable

Returns

KDRV_STATUS_OK only

4.8.4.7 kdrv_gpio_uninitialize()

```
kdrv_status_t kdrv_gpio_uninitialize (
            void  )
```

GPIO driver uninitialization.

This function disables the corresponding clock and frees resources allocated for GPIO operations.

Returns

KDRV_STATUS_OK only

4.8.4.8 kdrv_gpio_write_pin()

```
kdrv_status_t kdrv_gpio_write_pin (
            kdrv_gpio_pin_t pin,
            bool value )
```

write GPIO digitial pin value

This function writes a high or low value to a digital pin.
The specified pin must be configured as digital output.

Parameters

in	*pin*	GPIO pin ID, see kdrv_gpio_pin_t
in	*value*	Output value as digital high or digital low

Returns

KDRV_STATUS_OK only

Kneron Confidential

4.9 KDRV_I2C

Kneron I2C driver.

Enumerations

- enum kdrv_i2c_bus_speed_t { KDRV_I2C_SPEED_100K = 0, KDRV_I2C_SPEED_200K, KDRV_I2C_SPEED_400K, KDRV_I2C_SPEED_1M }

  Enumerations of I2C bus speed.
- enum kdrv_i2c_ctrl_t { KDRV_I2C_CTRL_0 = 0, KDRV_I2C_CTRL_1, KDRV_I2C_CTRL_2, KDRV_I2C_CTRL_3, TOTAL_KDRV_I2C_CTRL }

  Enumerations of I2C controller instances.

Functions

- kdrv_status_t kdrv_i2c_initialize (kdrv_i2c_ctrl_t ctrl_id, kdrv_i2c_bus_speed_t bus_speed)

  Initializes Kdrv I2C driver (as master) and configures it for the specified speed.
- kdrv_status_t kdrv_i2c_uninitialize (kdrv_i2c_ctrl_t ctrl_id)

  Uninitializes Kdrv I2C driver.
- kdrv_status_t kdp_i2c_transmit (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint8_t *data, uint32_t num, bool with_STOP)

  transmit data to a specified slave address, the STOP condition can be optionally not generated.
- kdrv_status_t kdp_i2c_receive (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint8_t *data, uint32_t num, bool with_STOP)

  receive data from a specified slave address, the STOP condition can be optionally not generated.
- kdrv_status_t kdrv_i2c_write_register (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint16_t reg, uint16_t reg_size, uint16_t len, uint8_t *data)

  specialized function to write to the register of slave device, register address can be 1 or 2 bytes.
- kdrv_status_t kdrv_i2c_read_register (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint16_t reg, uint16_t reg_size, uint16_t len, uint8_t *data)

  specialized function to read from the register of slave device, register address can be 1 or 2 bytes.

4.9.1 Detailed Description

Kneron I2C driver.

Here are the design highlight points:

- At present it supports only 7-bit slave address
- It is designed in polling way instead of interrupt way, user should be aware of this

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.9.2 Enumeration Type Documentation

4.9.2.1 kdrv_i2c_bus_speed_t

```
enum kdrv_i2c_bus_speed_t
```

Enumerations of I2C bus speed.

Enumerator

KDRV_I2C_SPEED_100K	Kdrv I2C bus speed 100KHz, standard mode
KDRV_I2C_SPEED_200K	Kdrv I2C bus speed 200KHz
KDRV_I2C_SPEED_400K	Kdrv I2C bus speed 400KHz, fast mode
KDRV_I2C_SPEED_1M	Kdrv I2C bus speed 1MHz, fast plus mode

4.9.2.2 kdrv_i2c_ctrl_t

```
enum kdrv_i2c_ctrl_t
```

Enumerations of I2C controller instances.

Enumerator

KDRV_I2C_CTRL_0	Kdrv I2C controller 0
KDRV_I2C_CTRL_1	Kdrv I2C controller 1
KDRV_I2C_CTRL_2	Kdrv I2C controller 2
KDRV_I2C_CTRL_3	Kdrv I2C controller 3
TOTAL_KDRV_I2C_CTRL	Total Kdrv I2C controllers

4.9.3 Function Documentation

4.9.3.1 kdp_i2c_receive()

```
kdrv_status_t kdp_i2c_receive (
            kdrv_i2c_ctrl_t ctrl_id,
            uint16_t slave_addr,
            uint8_t * data,
            uint32_t num,
            bool with_STOP )
```

Kneron Confidential

receive data from a specified slave address, the STOP condition can be optionally not generated.

This function will first set START condition then send slave address for write operations;
if 9th bit is NACK, it returns DEV_NACK error, and if it is ACK,
controller will continue to send out all data with specified number of bytes,
once it is done it will set STOP condition while the 'with_STOP' is KDP_BOOL_TRUE.
For every byte transmission, it returns DEV_NACK error while encountering NACK at 9th bit

Parameters

in	*ctrl_id*	see kdrv_i2c_ctrl_t
in	*slave_addr*	Address of the slave(7-bit by default)
out	*data*	data buffer address
in	*num*	Length of data to be written (in bytes)
in	*with_STOP*	STOP condition will be generated or not

Returns

kdrv_status_t see kdrv_status_t

4.9.3.2 kdp_i2c_transmit()

```
kdrv_status_t kdp_i2c_transmit (
            kdrv_i2c_ctrl_t ctrl_id,
            uint16_t slave_addr,
            uint8_t * data,
            uint32_t num,
            bool with_STOP )
```

transmit data to a specified slave address, the STOP condition can be optionally not generated.

This function will first set START condition then send slave address for write operations;
if 9th bit is NACK, it returns DEV_NACK error, and if it is ACK,
controller will continue to send out all data with specified number of bytes,
once it is done it will set STOP condition while the 'with_STOP' is KDP_BOOL_TRUE.
For every byte transmission, it returns DEV_NACK error while encountering NACK at 9th bit.

Parameters

in	*ctrl_id*	see kdrv_i2c_ctrl_t
in	*slave_addr*	Address of the slave(7-bit by default)
in	*data*	data buffer address
in	*num*	Length of data to be written (in bytes)
in	*with_STOP*	STOP condition will be generated or not

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.9.3.3 kdrv_i2c_initialize()

```
kdrv_status_t kdrv_i2c_initialize (
            kdrv_i2c_ctrl_t ctrl_id,
            kdrv_i2c_bus_speed_t bus_speed )
```

Initializes Kdrv I2C driver (as master) and configures it for the specified speed.

Parameters

in	*ctrl_id*	see kdrv_i2c_ctrl_t
in	*bus_speed*	see kdrv_i2c_bus_speed_t

Returns

kdrv_status_t see kdrv_status_t

Note

This API MUST be called before using the Read/write APIs for I2C.

4.9.3.4 kdrv_i2c_read_register()

```
kdrv_status_t kdrv_i2c_read_register (
            kdrv_i2c_ctrl_t ctrl_id,
            uint16_t slave_addr,
            uint16_t reg,
            uint16_t reg_size,
            uint16_t len,
            uint8_t * data )
```

specialized function to read from the register of slave device, register address can be 1 or 2 bytes.

Parameters

in	*ctrl_id*	see kdrv_i2c_ctrl_t
in	*slave_addr*	Address of the slave(7-bit by default)
in	*reg*	Register address
in	*reg_size*	Length of register address
in	*len*	Length of data to be read (in bytes).
out	*data*	data buffer to read register value

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.9.3.5 kdrv_i2c_uninitialize()

```
kdrv_status_t kdrv_i2c_uninitialize (
            kdrv_i2c_ctrl_t ctrl_id )
```

Uninitializes Kdrv I2C driver.

Parameters

in	*ctrl_id*	see kdrv_i2c_ctrl_t

Returns

kdrv_status_t see kdrv_status_t

4.9.3.6 kdrv_i2c_write_register()

```
kdrv_status_t kdrv_i2c_write_register (
            kdrv_i2c_ctrl_t ctrl_id,
            uint16_t slave_addr,
            uint16_t reg,
            uint16_t reg_size,
            uint16_t len,
            uint8_t * data )
```

specialized function to write to the register of slave device, register address can be 1 or 2 bytes.

Parameters

in	*ctrl_id*	see kdrv_i2c_ctrl_t
in	*slave_addr*	Address of the slave(7-bit by default)
in	*reg*	Register address
in	*reg_size*	Length of register address
in	*len*	Length of data to be written (in bytes).
in	*data*	data write register value

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.10 KDRV_LCDC

Kneron LCDC driver.

Macros

- #define FLAGS_KDP520_LCDC_START_DRAW_RECT_EVT BIT0
- #define FLAGS_KDP520_LCDC_STOP_DRAW_RECT_EVT BIT1
- #define MAX_FRAME_NUM (1)
- #define LCDC_HINT_BOUNDINGBOX_MARGIN_LEN (30)

Enumerations

- enum kdrv_lcdc_screen_ctrl_t { KDRV_LCDC_SCREEN_OFF = 0, KDRV_LCDC_SCREEN_ON }
- enum kdrv_lcdc_img_pixfmt_t { KDRV_LCDC_IMG_PIXFMT_1BPP = 0, KDRV_LCDC_IMG_PIXFMT_2BPP, KDRV_LCDC_IMG_PIXFMT_4BPP, KDRV_LCDC_IMG_PIXFMT_8BPP, KDRV_LCDC_IMG_PIXFMT_16BPP, KDRV_LCDC_IMG_PIXFMT_24BPP, KDRV_LCDC_IMG_PIXFMT_ARGB8888, KDRV_LCDC_IMG_PIXFMT_ARGB1555 }
- enum kdrv_lcdc_panel_type_t { KDRV_LCDC_6BIT_PER_CHANNEL = 0, KDRV_LCDC_8BIT_PER_CHANNEL }
- enum kdrv_lcdc_output_fmt_t { KDRV_LCDC_OUTPUT_FMT_RGB = 0, KDRV_LCDC_OUTPUT_FMT_BGR }
- enum kdrv_lcdc_serial_pix_sr_t { KDRV_LCDC_SERIAL_PIX_RSR = 0, KDRV_LCDC_SERIAL_PIX_LSR }
- enum kdrv_lcdc_serial_pix_colorseq_t { KDRV_LCDC_SERIAL_PIX_COLORSEQ_RGB = 0, KDRV_LCDC_SERIAL_PIX_COL KDRV_LCDC_SERIAL_PIX_COLORSEQ_GBR }
- enum kdrv_lcdc_serial_pix_delta_type_t { KDRV_LCDC_SERIAL_PIX_DELTA_TYPE_SAME_SEQ = 0, KDRV_LCDC_SERIAL_PIX_DELTA_TYPE_DIFF_SEQ }
- enum kdrv_lcdc_serial_pix_output_mode_t { KDRV_LCDC_SERIAL_PIX_RGB_PARALLEL_OUTPUT = 0, KDRV_LCDC_SERIAL_PIX_RGB_SERIAL_OUTPUT }
- enum kdrv_lcdc_fb_data_endianness_t { KDRV_LCDC_FB_DATA_ENDIAN_LBLP = 0, KDRV_LCDC_FB_DATA_ENDIAN_BE KDRV_LCDC_FB_DATA_ENDIAN_LBBP }
- enum kdrv_lcdc_pat_gen_t { KDRV_LCDC_PAT_GEN_DISABLE = 0, KDRV_LCDC_PAT_GEN_ENABLE }
- enum kdrv_lcdc_auo052_mode_t { KDRV_LCDC_AUO052_OFF = 0, KDRV_LCDC_AUO052_ON }

Functions

- kdrv_status_t kdrv_display_screen_control (kdrv_lcdc_screen_ctrl_t ctrl)

  Control display screen ON/OFF.
- kdrv_status_t kdrv_lcdc_set_panel_type (kdrv_lcdc_panel_type_t type)

  Set TFT panel color depth selection of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_bgrsw (kdrv_lcdc_output_fmt_t format)

  Set output format selection of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_sr (kdrv_lcdc_serial_pix_sr_t rotate)

  Set odd line shift rotate of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_colorseq (kdrv_lcdc_serial_pix_colorseq_t color)

  Set color sequence of odd line of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_delta_type (kdrv_lcdc_serial_pix_delta_type_t type)

  Set delta type arrangement of color filter of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_serial_mode (kdrv_lcdc_serial_pix_output_mode_t mode)

Kneron Confidential

Set RGB serial output mode of LCD serial panel pixel parameter.

- kdrv_status_t kdrv_lcdc_set_endian (kdrv_lcdc_fb_data_endianness_t endian_type)

  Set data endian.
- kdrv_status_t kdrv_lcdc_set_auo052_mode (kdrv_lcdc_auo052_mode_t mode)

  Set data endian.
- kdrv_status_t kdrv_lcdc_down_scale (uint16_t hor_no_in, uint16_t hor_no_out, uint16_t ver_no_in, uint16_t ver_no_out)

  Set image down scale.
- kdrv_status_t kdrv_lcdc_set_framerate (int framerate, int width, int height)

  Set frame rate of lcdc vsync.
- kdrv_status_t kdrv_lcdc_set_image_color_params (uint32_t color0, uint32_t color1, uint32_t color2, uint32_t color3)

  Set LCD color management parameter.
- kdrv_status_t kdrv_lcdc_set_bus_bandwidth_ctrl (uint32_t ctrl)

  Set frame rate of lcdc vsync.
- kdrv_status_t kdrv_lcdc_set_frame_buffer (uint32_t img_scal_down)

  Set frame buffer parameter.

4.10.1 Detailed Description

Kneron LCDC driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.10.2 Macro Definition Documentation

4.10.2.1 FLAGS_KDP520_LCDC_START_DRAW_RECT_EVT

```
#define FLAGS_KDP520_LCDC_START_DRAW_RECT_EVT BIT0
```

4.10.2.2 FLAGS_KDP520_LCDC_STOP_DRAW_RECT_EVT

```
#define FLAGS_KDP520_LCDC_STOP_DRAW_RECT_EVT BIT1
```

4.10.2.3 LCDC_HINT_BOUNDINGBOX_MARGIN_LEN

```
#define LCDC_HINT_BOUNDINGBOX_MARGIN_LEN (30)
```

Kneron Confidential

4.10.2.4 MAX_FRAME_NUM

```
#define MAX_FRAME_NUM (1)
```

4.10.3 Enumeration Type Documentation

4.10.3.1 kdrv_lcdc_auo052_mode_t

```
enum kdrv_lcdc_auo052_mode_t
```

Enumerations of lcdc panel pixel parameter, AUO052 mode

Enumerator

KDRV_LCDC_AUO052_OFF	0: Turn off the AUO052 mode
KDRV_LCDC_AUO052_ON	1: Turn on the AUO052 mode

4.10.3.2 kdrv_lcdc_fb_data_endianness_t

```
enum kdrv_lcdc_fb_data_endianness_t
```

Enumerations of lcdc image format parameter, endian control

Enumerator

KDRV_LCDC_FB_DATA_ENDIAN_LBLP	00: Little-endian byte little-endian pixel
KDRV_LCDC_FB_DATA_ENDIAN_BBBP	01: Big-endian byte big-endian pixel
KDRV_LCDC_FB_DATA_ENDIAN_LBBP	10: Little-endian byte big-endian pixel (WinCE)

4.10.3.3 kdrv_lcdc_img_pixfmt_t

```
enum kdrv_lcdc_img_pixfmt_t
```

Enumerations of lcdc panel pixel parameter, image pixel format in FIFO

Enumerator

KDRV_LCDC_IMG_PIXFMT_1BPP	000: 1 bpp
KDRV_LCDC_IMG_PIXFMT_2BPP	001: 2 bpp
KDRV_LCDC_IMG_PIXFMT_4BPP	010: 4 bpp

Kneron Confidential

Enumerator

KDRV_LCDC_IMG_PIXFMT_8BPP	011: 8 bpp
KDRV_LCDC_IMG_PIXFMT_16BPP	100: 16 bpp
KDRV_LCDC_IMG_PIXFMT_24BPP	101: 24 bpp
KDRV_LCDC_IMG_PIXFMT_ARGB8888	110: ARGB8888
KDRV_LCDC_IMG_PIXFMT_ARGB1555	111: ARGB1555

4.10.3.4 kdrv_lcdc_output_fmt_t

`enum kdrv_lcdc_output_fmt_t`

Enumerations of lcdc panel pixel parameter, output format selection

Enumerator

KDRV_LCDC_OUTPUT_FMT_RGB	RGB normal output
KDRV_LCDC_OUTPUT_FMT_BGR	BGR red and blue swapped output

4.10.3.5 kdrv_lcdc_panel_type_t

`enum kdrv_lcdc_panel_type_t`

Enumerations of lcdc panel pixel parameter, TFT panel color depth selection

Enumerator

KDRV_LCDC_6BIT_PER_CHANNEL	6 bits per channel with a 18-bit panel interface
KDRV_LCDC_8BIT_PER_CHANNEL	8 bits per channel with a 24-bit panel interface

4.10.3.6 kdrv_lcdc_pat_gen_t

`enum kdrv_lcdc_pat_gen_t`

Enumerations of lcdc function enable parameter, test pattern generator

Enumerator

KDRV_LCDC_PAT_GEN_DISABLE	Turn-off pattern generator
KDRV_LCDC_PAT_GEN_ENABLE	Turn-on pattern generator

Kneron Confidential

4.10.3.7 kdrv_lcdc_screen_ctrl_t

enum kdrv_lcdc_screen_ctrl_t

Enumerations of lcdc screen control

Enumerator

KDRV_LCDC_SCREEN_OFF	LCDC screen control off
KDRV_LCDC_SCREEN_ON	LCDC screen control on

4.10.3.8 kdrv_lcdc_serial_pix_colorseq_t

enum kdrv_lcdc_serial_pix_colorseq_t

Enumerations of lcdc serial panel pixel parameter, color sequence of odd line

Enumerator

KDRV_LCDC_SERIAL_PIX_COLORSEQ_RGB	RGB decides the sub-pixel sequence of the odd line
KDRV_LCDC_SERIAL_PIX_COLORSEQ_BRG	BRG decides the sub-pixel sequence of the odd line
KDRV_LCDC_SERIAL_PIX_COLORSEQ_GBR	GBR decides the sub-pixel sequence of the odd line

4.10.3.9 kdrv_lcdc_serial_pix_delta_type_t

enum kdrv_lcdc_serial_pix_delta_type_t

Enumerations of lcdc serial panel pixel parameter, delta type arrangement of color filter

Enumerator

KDRV_LCDC_SERIAL_PIX_DELTA_TYPE_SAM↩ E_SEQ	Odd line and even line have the same data sequence
KDRV_LCDC_SERIAL_PIX_DELTA_TYPE_DIFF_↩ SEQ	Odd line and even line have the difference data sequence

4.10.3.10 kdrv_lcdc_serial_pix_output_mode_t

enum kdrv_lcdc_serial_pix_output_mode_t

Kneron Confidential

Enumerations of lcdc serial panel pixel parameter, RGB serial output mode

Enumerator

KDRV_LCDC_SERIAL_PIX_RGB_PARALLEL_OUTPUT	RGB parallel format output
KDRV_LCDC_SERIAL_PIX_RGB_SERIAL_OUTPUT	RGB serial format output

4.10.3.11 kdrv_lcdc_serial_pix_sr_t

```
enum kdrv_lcdc_serial_pix_sr_t
```

Enumerations of lcdc serial panel pixel parameter, shift rotate

Enumerator

KDRV_LCDC_SERIAL_PIX_RSR	Even line sequence from through the odd line rotating right
KDRV_LCDC_SERIAL_PIX_LSR	Even line sequence from through the odd line rotating left

4.10.4 Function Documentation

4.10.4.1 kdrv_display_screen_control()

```
kdrv_status_t kdrv_display_screen_control (
            kdrv_lcdc_screen_ctrl_t ctrl )
```

Control display screen ON/OFF.

Parameters

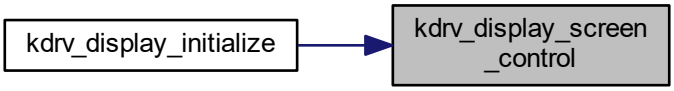
in	*ctrl*	see kdrv_lcdc_screen_ctrl_t

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.10.4.2 kdrv_lcdc_down_scale()

```
kdrv_status_t kdrv_lcdc_down_scale (
            uint16_t hor_no_in,
            uint16_t hor_no_out,
            uint16_t ver_no_in,
            uint16_t ver_no_out )
```

Set image down scale.

Parameters

in	*hor_no_in*	Width of input image source
in	*hor_no_out*	Height of input image source
in	*ver_no_in*	Width of output image source
in	*ver_no_out*	Height of output image source

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

Kneron Confidential

4.10.4.3 kdrv_lcdc_set_auo052_mode()

```
kdrv_status_t kdrv_lcdc_set_auo052_mode (
            kdrv_lcdc_auo052_mode_t mode )
```

Set data endian.

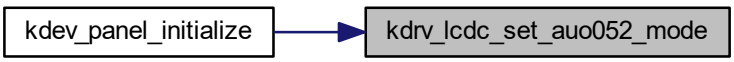
Parameters

in	*mode*	see kdrv_lcdc_auo052_mode_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.10.4.4 kdrv_lcdc_set_bgrsw()

```
kdrv_status_t kdrv_lcdc_set_bgrsw (
            kdrv_lcdc_output_fmt_t format )
```

Set output format selection of LCD serial panel pixel parameter.

Parameters

in	*type*	see kdrv_lcdc_output_fmt_t

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.10.4.5 kdrv_lcdc_set_bus_bandwidth_ctrl()

```
kdrv_status_t kdrv_lcdc_set_bus_bandwidth_ctrl (
            uint32_t ctrl )
```

Set frame rate of lcdc vsync.

Parameters

in	*ctrl*	[BIT9] Enable the LOCK command 0:Disable the LOCK command 1:Enable the LOCK command [BIT8] Enable the bus bandwidth ratio 0:Disable the bus bandwidth ratio 1:Enable the bus bandwidth ratio [BIT7:BIT6] Bus bandwidth control ratio for the Image3 Frame buffer 00:Ratio 1 01:Ratio 2 10:Ratio 4 [BIT5:BIT4] Bus bandwidth control ratio for the Image2 Frame buffer 00:Ratio 1 01:Ratio 2 10:Ratio 4 [BIT3:BIT2] Bus bandwidth control ratio for the Image1 Frame buffer 00:Ratio 1 01:Ratio 2 10:Ratio 4 [BIT1:BIT0] Bus bandwidth control ratio for the Image0 Frame buffer 00:Ratio 1 01:Ratio 2 10:Ratio 4

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

Here is the caller graph for this function:

4.10.4.6 kdrv_lcdc_set_endian()

```
kdrv_status_t kdrv_lcdc_set_endian (
            kdrv_lcdc_fb_data_endianness_t endian_type )
```

Set data endian.

Parameters

in	*mode*	see kdrv_lcdc_fb_data_endianness_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

Kneron Confidential

4.10.4.7 kdrv_lcdc_set_frame_buffer()

```
kdrv_status_t kdrv_lcdc_set_frame_buffer (
            uint32_t img_scal_down )
```

Set frame buffer parameter.

Parameters

in	*img_scal_down*	[BIT15:BIT14] Scaling down for image3 The image from LCDImage3FrameBase can be scaled down depending on the value. 00: Disable 01:Image3 will be scaling down to 1/2 x 1/2 10:Image3 will be scaling down to 1/2 x 1 [BIT13:BIT12] Scaling down for image2 The image from LCDImage2FrameBase can be scaled down depending on the value. 00: Disable 01:Image2 will be scaling down to 1/2 x 1/2 10:Image2 will be scaling down to 1/2 x 1 [BIT11:BIT10] Scaling down for image1 The image from LCDImage1FrameBase can be scaled down depending on the value. 00: Disable 01:Image1 will be scaling down to 1/2 x 1/2 10:Image1 will be scaling down to 1/2 x 1 [BIT9:BIT8] Scaling down for image0 The image from LCDImage0FrameBase can be scaled down depending on the value. 00: Disable 01:Image0 will be scaling down to 1/2 x 1/2 10:Image0 will be scaling down to 1/2 x 1

Note

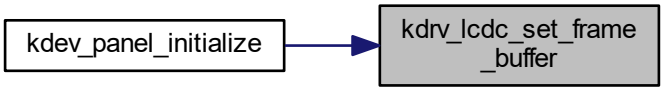
Please note that these filed values can only be set to '00' under the folloing conditions:

- VirtualScreenEn or LCM_En is set
- PiP has a chance to be turned-on when TV is enabled.

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



Kneron Confidential

4.10.4.8 kdrv_lcdc_set_framerate()

```
kdrv_status_t kdrv_lcdc_set_framerate (
            int framerate,
            int width,
            int height )
```

Set frame rate of lcdc vsync.

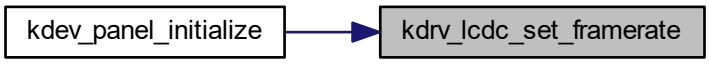
Parameters

in	*framerate*	Frame rate
in	*framerate*	Image width
in	*framerate*	Image height

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.10.4.9 kdrv_lcdc_set_image_color_params()

```
kdrv_status_t kdrv_lcdc_set_image_color_params (
            uint32_t color0,
            uint32_t color1,
            uint32_t color2,
            uint32_t color3 )
```

Set LCD color management parameter.

Kneron Confidential

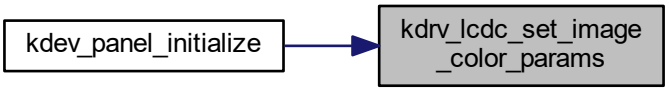
Parameters

in	*color0*	[BIT13_BIT8] Saturation value. Cb(sat) = Cb(org) ∗ (SatValue/32). Cr(sat) = Cr(org) ∗ (SatValue/32). [BIT7] Sign bit of brightness value. 0: The value of brightness is positive. 1: The value of brightness is negative. [BIT6:BIT0] Brightness level The range of the brightness level is from 0 to 127.
in	*color1*	This register value defines the coefficient of the hur operation [BIT14] Sign bit of HuCosValue. 0: The value of HuCosValue is positive. 1: The value of HuCosValue is negative. [BIT13:BIT8] Hue value of coefficient Cos -180∼180 degree. [BIT6] Sigh bit of HuSinValue 0: The value of HuSinValue is positive. 1: The value of HuSinValue is negative. [BIT5:BIT0] Hue value of coefficient Sin -180∼180 degree.
in	*color2*	This register value defines the coefficient of the sharpness operation [BIT23:BIT20] Sharpness weight value 1. The value determines the second weight of sharpness. [BIT19:BIT16] Sharpness weight value 0. The value determines the first weight of sharpness. [BIT15:BIT8] Sharpness threshold value 1. The value determines the second threshold of sharpness. [BIT7:BIT0] Sharpness threshold value 0. The value determines the second threshold of sharpness.
in	*color3*	This register value defines the coefficient of the contast operation [BIT20:BIT16] Contrast cure slope. The value determines the slope of contrast cure. The actual slope is the value devided by 4. Note: This value cannot be programmed to 0. [BIT12] Contrast offset sign 1: (Contr_slope x 128) > 512. 0: (Contr_slope x 128) < 512. [BIT11:BIT0] Contrast offset value. The value is defined as absolute of "Contr_slope x 128 - 512".

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



Kneron Confidential

4.10.4.10 kdrv_lcdc_set_panel_type()

```
kdrv_status_t kdrv_lcdc_set_panel_type (
            kdrv_lcdc_panel_type_t type )
```

Set TFT panel color depth selection of LCD serial panel pixel parameter.

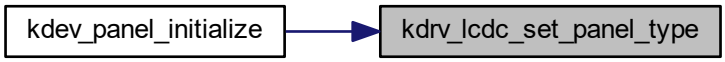
Parameters

in	*type*	see kdrv_lcdc_panel_type_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.10.4.11 kdrv_lcdc_set_pixel_colorseq()

```
kdrv_status_t kdrv_lcdc_set_pixel_colorseq (
          kdrv_lcdc_serial_pix_colorseq_t color )
```

Set color sequence of odd line of LCD serial panel pixel parameter.

Parameters

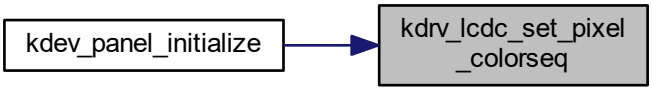
in	*type*	see kdrv_lcdc_serial_pix_colorseq_t

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.10.4.12 kdrv_lcdc_set_pixel_delta_type()

```
kdrv_status_t kdrv_lcdc_set_pixel_delta_type (
            kdrv_lcdc_serial_pix_delta_type_t type )
```

Set delta type arrangement of color filter of LCD serial panel pixel parameter.

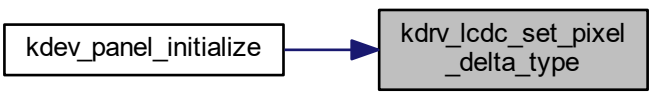
Parameters

in	*type*	see kdrv_lcdc_serial_pix_delta_type_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



Kneron Confidential

4.10.4.13 kdrv_lcdc_set_pixel_serial_mode()

```
kdrv_status_t kdrv_lcdc_set_pixel_serial_mode (
            kdrv_lcdc_serial_pix_output_mode_t mode )
```

Set RGB serial output mode of LCD serial panel pixel parameter.

Kneron Confidential

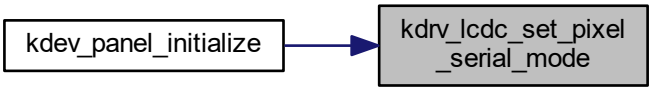
Parameters

in	*mode*	see kdrv_lcdc_serial_pix_output_mode_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.10.4.14 kdrv_lcdc_set_pixel_sr()

```
kdrv_status_t kdrv_lcdc_set_pixel_sr (
            kdrv_lcdc_serial_pix_sr_t rotate )
```

Set odd line shift rotate of LCD serial panel pixel parameter.

Parameters

in	*type*	see kdrv_lcdc_serial_pix_sr_t

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

Kneron Confidential

4.11 KDRV_LCM

Kneron LCM driver.

Functions

- u8 kdrv_lcm_get_backlight (void)
- kdrv_status_t kdrv_lcm_pressing (kdrv_display_t *display_drv, u32 addr)
- kdrv_status_t kdrv_lcm_pressingnir (kdrv_display_t *display_drv, u32 addr)
- kdrv_status_t kdrv_lcm_write_cmd (uint32_t base, unsigned char data)
- kdrv_status_t kdrv_lcm_write_data (uint32_t base, unsigned char data)
- unsigned int kdrv_lcm_read_data (uint32_t base)
- uint32_t kdrv_lcm_get_db_frame (void)

4.11.1 Detailed Description

Kneron LCM driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.11.2 Function Documentation

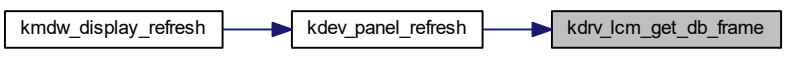
4.11.2.1 kdrv_lcm_get_backlight()

```
u8 kdrv_lcm_get_backlight (
            void  )
```

4.11.2.2 kdrv_lcm_get_db_frame()

```
uint32_t kdrv_lcm_get_db_frame (
            void  )
```

Here is the caller graph for this function:

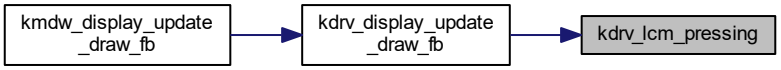


Kneron Confidential

4.11.2.3 kdrv_lcm_pressing()

```
kdrv_status_t kdrv_lcm_pressing (
            kdrv_display_t * display_drv,
            u32 addr )
```

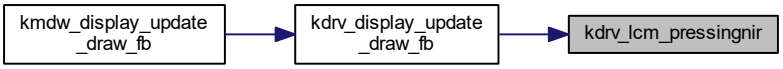
Here is the caller graph for this function:



4.11.2.4 kdrv_lcm_pressingnir()

```
kdrv_status_t kdrv_lcm_pressingnir (
            kdrv_display_t * display_drv,
            u32 addr )
```

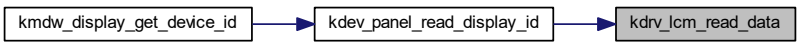
Here is the caller graph for this function:



4.11.2.5 kdrv_lcm_read_data()

```
unsigned int kdrv_lcm_read_data (
            uint32_t base )
```

Here is the caller graph for this function:

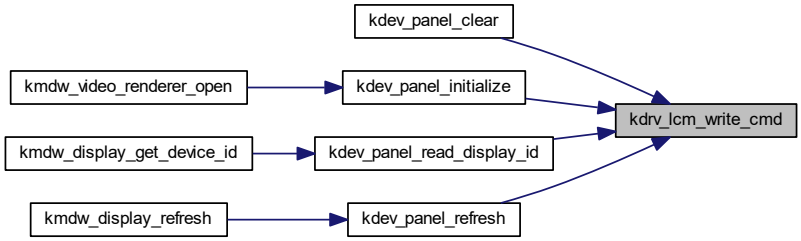


Kneron Confidential

4.11.2.6 kdrv_lcm_write_cmd()

```
kdrv_status_t kdrv_lcm_write_cmd (
            uint32_t base,
            unsigned char data )
```

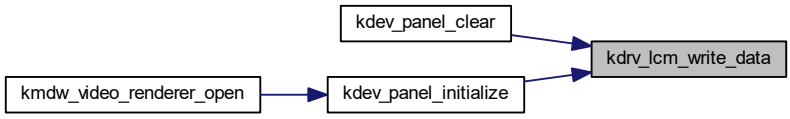
Here is the caller graph for this function:



4.11.2.7 kdrv_lcm_write_data()

```
kdrv_status_t kdrv_lcm_write_data (
            uint32_t base,
            unsigned char data )
```

Here is the caller graph for this function:



Kneron Confidential

4.12 KDRV_MIPICSIRX

Kneron mipicsirx driver.

Enumerations

- enum { CSI2RX_CAM_0, CSI2RX_CAM_1, CSI2RX_CAM_NUM }

Functions

- kdrv_status_t kdrv_csi2rx_initialize (uint32_t cam_idx)

  kdrv_csi2rx_initialize Initiclize mipicsirx related variable.
- kdrv_status_t kdrv_csi2rx_enable (uint32_t input_type, uint32_t cam_idx, uint32_t sensor_idx, struct cam_format *fmt)

  kdrv_csi2rx_enable Set mipicsirx related register for IP enable.
- kdrv_status_t kdrv_csi2rx_start (uint32_t input_type, uint32_t cam_idx)

  kdrv_csi2rx_start Set mipicsirx related register for IP start.
- kdrv_status_t kdrv_csi2rx_set_power (uint32_t cam_idx, uint32_t on)

  kdrv_csi2rx_set_power Set mipicsirx power related register.
- kdrv_status_t kdrv_csi2rx_reset (uint32_t cam_idx, uint32_t sensor_idx)

  kdrv_csi2rx_reset Reset mipicsirx.

4.12.1 Detailed Description

Kneron mipicsirx driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.12.2 Enumeration Type Documentation

4.12.2.1 anonymous enum

```
anonymous enum
```

Enumerator

CSI2RX_CAM_0	
CSI2RX_CAM_1	
CSI2RX_CAM_NUM	

Kneron Confidential

4.12.3 Function Documentation

4.12.3.1 kdrv_csi2rx_enable()

```
kdrv_status_t kdrv_csi2rx_enable (
            uint32_t input_type,
            uint32_t cam_idx,
            uint32_t sensor_idx,
            struct cam_format * fmt )
```

kdrv_csi2rx_enable Set mipicsirx related register for IP enable.

Parameters

in	*input_type*	sensor input type
in	*cam_idx*	cam idx
in	*sensor_idx*	sensor idx
in	*fmt*	camera related format setting

Returns

kdrv_status_t

4.12.3.2 kdrv_csi2rx_initialize()

```
kdrv_status_t kdrv_csi2rx_initialize (
            uint32_t cam_idx )
```

kdrv_csi2rx_initialize Initiclize mipicsirx related variable.

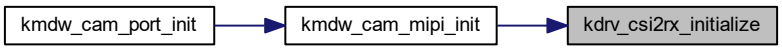
Parameters

in	*cam_idx*	cam idx

Returns

kdrv_status_t

Here is the caller graph for this function:



Kneron Confidential

4.12.3.3 kdrv_csi2rx_reset()

```
kdrv_status_t kdrv_csi2rx_reset (
            uint32_t cam_idx,
            uint32_t sensor_idx )
```

kdrv_csi2rx_reset Reset mipicsirx.

Parameters

in	*cam_idx*	cam_idx
in	*on*	csirx power status, 1: ON, 0:Off

Returns

kdrv_status_t

4.12.3.4 kdrv_csi2rx_set_power()

```
kdrv_status_t kdrv_csi2rx_set_power (
            uint32_t cam_idx,
            uint32_t on )
```

kdrv_csi2rx_set_power Set mipicsirx power related register.

Parameters

in	*cam_idx*	cam_idx
in	*on*	csirx power status, 1: ON, 0:Off

Returns

kdrv_status_t

4.12.3.5 kdrv_csi2rx_start()

```
kdrv_status_t kdrv_csi2rx_start (
            uint32_t input_type,
            uint32_t cam_idx )
```

kdrv_csi2rx_start Set mipicsirx related register for IP start.

Kneron Confidential

Parameters

in	*input_type*	sensor input type
in	*cam_idx*	cam_idx

Returns

kdrv_status_t

Kneron Confidential

4.13 KDRV_MPU

Kneron MPU driver.

Functions

- void kdrv_mpu_config (void)
  config memorty protect space, siram + niram
- void kdrv_mpu_niram_enable (void)
  mpu protect enable for niram memory space
- void kdrv_mpu_niram_disable (void)
  mpu protect disable for niram memory space

4.13.1 Detailed Description

Kneron MPU driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.13.2 Function Documentation

4.13.2.1 kdrv_mpu_config()

```
void kdrv_mpu_config (
            void  )
```

config memorty protect space, siram + niram

Returns

N/A

4.13.2.2 kdrv_mpu_niram_disable()

```
void kdrv_mpu_niram_disable (
            void  )
```

mpu protect disable for niram memory space

Returns

N/A

Kneron Confidential

4.13.2.3 kdrv_mpu_niram_enable()

```
void kdrv_mpu_niram_enable (
            void  )
```

mpu protect enable for niram memory space

Returns

N/A

Kneron Confidential

4.14 KDRV_NPU

Kneron NPU driver.

Typedefs

- typedef void(∗ ipc_handler_t) (int ipc_idx, int state)

Functions

- void kdrv_ncpu_initialize (ipc_handler_t ipc_handler)
  Initialize NPU functionality.
- void kdrv_ncpu_set_model (struct kdp_model_s ∗model_info_addr, uint32_t info_idx, int32_t slot_idx)
  Set model information.
- int kdrv_ncpu_get_avail_com (void)
  Get available COM.
- int kdrv_ncpu_set_model_active (uint32_t index)
  Set active model index.
- int kdrv_ncpu_get_model_active (void)
  Get active model index.
- int kdrv_ncpu_set_image_active (uint32_t index)
  Set active image index.
- int kdrv_ncpu_get_image_active (void)
  Get active image index.
- void kdrv_ncpu_set_scpu_debug_lvl (uint32_t lvl)
  Set SCPU debug level.
- void kdrv_ncpu_set_ncpu_debug_lvl (uint32_t lvl)
  Set NCPU debug level.
- void kdrv_ncpu_trigger_int (void)
  Trigger NCPU interrupt.
- struct ncpu_to_scpu_s ∗ kdrv_ncpu_get_input (void)
  Get ncpu_to_scpu_s.
- struct scpu_to_ncpu_s ∗ kdrv_ncpu_get_output (void)
  Get scpu_to_ncpu_s.

4.14.1 Detailed Description

Kneron NPU driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.14.2 Typedef Documentation

Kneron Confidential

4.14.2.1 ipc_handler_t

```
typedef void(* ipc_handler_t) (int ipc_idx, int state)
```

4.14.3 Function Documentation

4.14.3.1 kdrv_ncpu_get_avail_com()

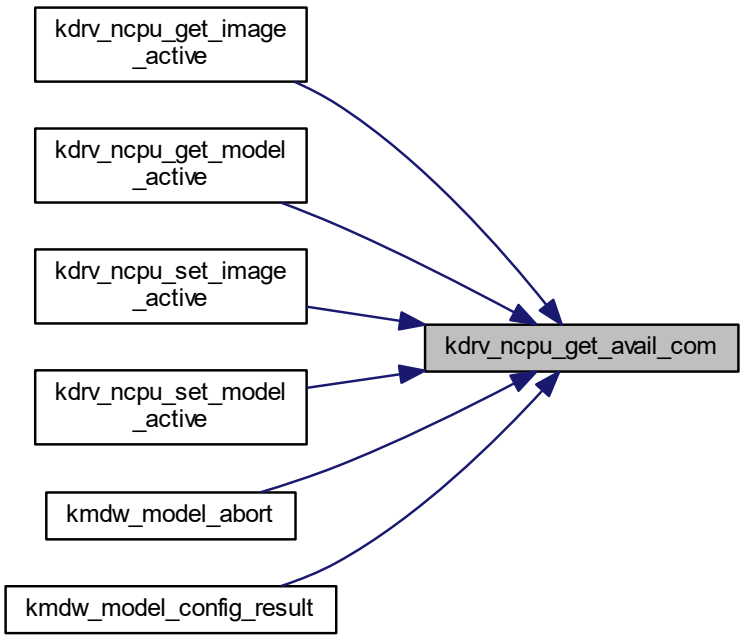
```
int kdrv_ncpu_get_avail_com (
            void  )
```

Get available COM.

Returns

COM index

Here is the caller graph for this function:



Kneron Confidential

4.14.3.2 kdrv_ncpu_get_image_active()

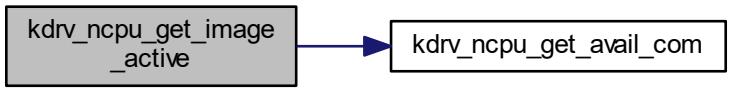
```
int kdrv_ncpu_get_image_active (
            void  )
```

Get active image index.

Returns

index

Here is the call graph for this function:



4.14.3.3 kdrv_ncpu_get_input()

```
struct ncpu_to_scpu_s* kdrv_ncpu_get_input (
            void  )
```

Get ncpu_to_scpu_s.

Returns

IPC struct

4.14.3.4 kdrv_ncpu_get_model_active()

```
int kdrv_ncpu_get_model_active (
            void  )
```

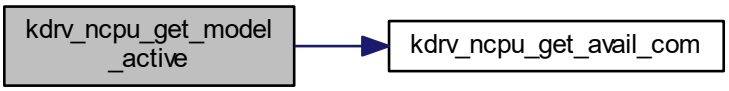
Get active model index.

Kneron Confidential

Returns

index

Here is the call graph for this function:



4.14.3.5 kdrv_ncpu_get_output()

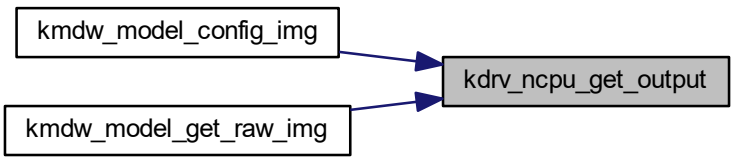
```
struct scpu_to_ncpu_s* kdrv_ncpu_get_output (
            void  )
```

Get scpu_to_ncpu_s.

Returns

IPC struct

Here is the caller graph for this function:



4.14.3.6 kdrv_ncpu_initialize()

```
void kdrv_ncpu_initialize (
            ipc_handler_t ipc_handler )
```

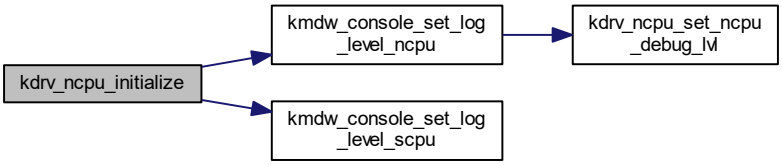
Initialize NPU functionality.

Kneron Confidential

Parameters

ipc_handler	IPC callback

Here is the call graph for this function:



Here is the caller graph for this function:

4.14.3.7 kdrv_ncpu_set_image_active()

```
int kdrv_ncpu_set_image_active (
            uint32_t index )
```

Set active image index.

Parameters

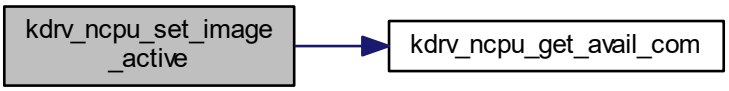
index	image index

Kneron Confidential

Returns

available COM

Here is the call graph for this function:



4.14.3.8 kdrv_ncpu_set_model()

```
void kdrv_ncpu_set_model (
             struct kdp_model_s * model_info_addr,
             uint32_t info_idx,
             int32_t slot_idx )
```

Set model information.

Parameters

model_info_addr	model information address
info_idx	information index
slot_idx	slot index

4.14.3.9 kdrv_ncpu_set_model_active()

```
int kdrv_ncpu_set_model_active (
             uint32_t index )
```

Set active model index.

Parameters

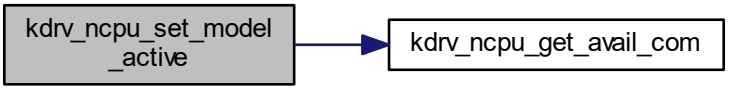
index	model slot index

Returns

available COM

Kneron Confidential

Here is the call graph for this function:



4.14.3.10 kdrv_ncpu_set_ncpu_debug_lvl()

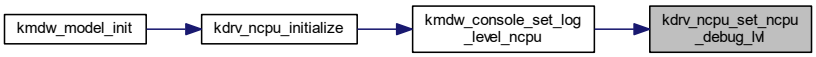
```
void kdrv_ncpu_set_ncpu_debug_lvl (
            uint32_t lvl )
```

Set NCPU debug level.

Parameters

lvl	level

Here is the caller graph for this function:



4.14.3.11 kdrv_ncpu_set_scpu_debug_lvl()

```
void kdrv_ncpu_set_scpu_debug_lvl (
            uint32_t lvl )
```

Set SCPU debug level.

Parameters

lvl	level

Kneron Confidential

4.14.3.12 kdrv_ncpu_trigger_int()

```
void kdrv_ncpu_trigger_int (
            void  )
```

Trigger NCPU interrupt.

Kneron Confidential

4.15 KDRV_PINMUX_CONFIG

Kneron pinmux config driver.

Enumerations

- enum kdrv_pin_name {
  KDRV_PIN_SPI_WP_N = 0, KDRV_PIN_SPI_HOLD_N, KDRV_PIN_JTAG_TRST_N, KDRV_PIN_JTAG_TDI,
  KDRV_PIN_JTAG_SWDITMS, KDRV_PIN_JTAG_SWCLKTCK, KDRV_PIN_JTAG_TDO, KDRV_PIN_LC_PCLK,
  KDRV_PIN_LC_VS, KDRV_PIN_LC_HS, KDRV_PIN_LC_DE, KDRV_PIN_LC_DATA_0,
  KDRV_PIN_LC_DATA_1, KDRV_PIN_LC_DATA_2, KDRV_PIN_LC_DATA_3, KDRV_PIN_LC_DATA_4,
  KDRV_PIN_LC_DATA_5, KDRV_PIN_LC_DATA_6, KDRV_PIN_LC_DATA_7, KDRV_PIN_LC_DATA_8,
  KDRV_PIN_LC_DATA_9, KDRV_PIN_LC_DATA_10, KDRV_PIN_LC_DATA_11, KDRV_PIN_LC_DATA_12,
  KDRV_PIN_LC_DATA_13, KDRV_PIN_LC_DATA_14, KDRV_PIN_LC_DATA_15, KDRV_PIN_SD_CLK,
  KDRV_PIN_SD_CMD, KDRV_PIN_SD_DAT_0, KDRV_PIN_SD_DAT_1, KDRV_PIN_SD_DAT_2,
  KDRV_PIN_SD_DAT_3, KDRV_PIN_UART0_RX, KDRV_PIN_UART0_TX, KDRV_PIN_I2C0_SCL,
  KDRV_PIN_I2C0_SDA, KDRV_PIN_PWM0 }

  Enumerations of KDP520 all configurable pins.
- enum kdrv_pinmux_mode {
  PIN_MODE_0 = 0, PIN_MODE_1, PIN_MODE_2, PIN_MODE_3,
  PIN_MODE_4, PIN_MODE_5, PIN_MODE_6, PIN_MODE_7 }

  Enumerations of KDP520 pinmux modes.
- enum kdrv_pin_pull { PIN_PULL_NONE, PIN_PULL_UP, PIN_PULL_DOWN }

  Enumerations of KDP520 pull status.
- enum kdrv_pin_driving {
  PIN_DRIVING_NONE, PIN_DRIVING_4MA, PIN_DRIVING_8MA, PIN_DRIVING_12MA,
  PIN_DRIVING_16MA }

  Enumerations of KDP520 output driving capability.

Functions

- void kdrv_pinmux_config (kdrv_pin_name pin, kdrv_pinmux_mode mode, kdrv_pin_pull pull_type, kdrv_pin_driving driving)

  Pinmux configure.

4.15.1 Detailed Description

Kneron pinmux config driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.15.2 Enumeration Type Documentation

4.15.2.1 kdrv_pin_driving

`enum kdrv_pin_driving`

Enumerations of KDP520 output driving capability.

Kneron Confidential

Enumerator

PIN_DRIVING_NONE	None, for GPIO input
PIN_DRIVING_4MA	4mA - 00
PIN_DRIVING_8MA	8mA - 01
PIN_DRIVING_12MA	12mA - 10
PIN_DRIVING_16MA	16mA - 11

4.15.2.2 kdrv_pin_name

`enum kdrv_pin_name`

Enumerations of KDP520 all configurable pins.

Enumerator

KDRV_PIN_SPI_WP_N	PAD name X_SPI_WP_N, default PIN_MODE_0
KDRV_PIN_SPI_HOLD_N	PAD name X_SPI_HOLD_N, default PIN_MODE_0
KDRV_PIN_JTAG_TRST_N	PAD name X_JTAG_TRST_N, default PIN_MODE_0
KDRV_PIN_JTAG_TDI	PAD name X_JTAG_TDI, default PIN_MODE_0
KDRV_PIN_JTAG_SWDITMS	PAD name X_JTAG_SWDITMS, default PIN_MODE_0
KDRV_PIN_JTAG_SWCLKTCK	PAD name X_JTAG_SWCLKTCK, default PIN_MODE_0
KDRV_PIN_JTAG_TDO	PAD name X_JTAG_TDO, default PIN_MODE_0
KDRV_PIN_LC_PCLK	PAD name X_LC_PCLK, default PIN_MODE_0
KDRV_PIN_LC_VS	PAD name X_LC_VS, default PIN_MODE_0
KDRV_PIN_LC_HS	PAD name X_LC_HS, default PIN_MODE_0
KDRV_PIN_LC_DE	PAD name X_LC_DE, default PIN_MODE_0
KDRV_PIN_LC_DATA_0	PAD name X_LC_DATA_0, default PIN_MODE_0
KDRV_PIN_LC_DATA_1	PAD name X_LC_DATA_1, default PIN_MODE_0
KDRV_PIN_LC_DATA_2	PAD name X_LC_DATA_2, default PIN_MODE_0
KDRV_PIN_LC_DATA_3	PAD name X_LC_DATA_3, default PIN_MODE_0
KDRV_PIN_LC_DATA_4	PAD name X_LC_DATA_4, default PIN_MODE_0
KDRV_PIN_LC_DATA_5	PAD name X_LC_DATA_5, default PIN_MODE_0
KDRV_PIN_LC_DATA_6	PAD name X_LC_DATA_6, default PIN_MODE_0
KDRV_PIN_LC_DATA_7	PAD name X_LC_DATA_7, default PIN_MODE_0
KDRV_PIN_LC_DATA_8	PAD name X_LC_DATA_8, default PIN_MODE_0
KDRV_PIN_LC_DATA_9	PAD name X_LC_DATA_9, default PIN_MODE_0
KDRV_PIN_LC_DATA_10	PAD name X_LC_DATA_10, default PIN_MODE_0
KDRV_PIN_LC_DATA_11	PAD name X_LC_DATA_11, default PIN_MODE_0
KDRV_PIN_LC_DATA_12	PAD name X_LC_DATA_12, default PIN_MODE_0
KDRV_PIN_LC_DATA_13	PAD name X_LC_DATA_13, default PIN_MODE_0
KDRV_PIN_LC_DATA_14	PAD name X_LC_DATA_14, default PIN_MODE_0
KDRV_PIN_LC_DATA_15	PAD name X_LC_DATA_15, default PIN_MODE_0
KDRV_PIN_SD_CLK	PAD name X_SD_CLK, default PIN_MODE_0
KDRV_PIN_SD_CMD	PAD name X_SD_CMD, default PIN_MODE_0
KDRV_PIN_SD_DAT_0	PAD name X_SD_DAT_0, default PIN_MODE_0
KDRV_PIN_SD_DAT_1	PAD name X_SD_DAT_1, default PIN_MODE_0

Kneron Confidential

Enumerator

KDRV_PIN_SD_DAT_2	PAD name X_SD_DAT_2, default PIN_MODE_0
KDRV_PIN_SD_DAT_3	PAD name X_SD_DAT_3, default PIN_MODE_0
KDRV_PIN_UART0_RX	PAD name X_UART0_RX, default PIN_MODE_0
KDRV_PIN_UART0_TX	PAD name X_UART0_TX, default PIN_MODE_0
KDRV_PIN_I2C0_SCL	PAD name X_I2C0_SCL, default PIN_MODE_0
KDRV_PIN_I2C0_SDA	PAD name X_I2C0_SDA, default PIN_MODE_0
KDRV_PIN_PWM0	PAD name X_PWM0, default PIN_MODE_0

4.15.2.3 kdrv_pin_pull

`enum kdrv_pin_pull`

Enumerations of KDP520 pull status.

Enumerator

PIN_PULL_NONE	Pin none
PIN_PULL_UP	Pin pull up
PIN_PULL_DOWN	Pin pull down

4.15.2.4 kdrv_pinmux_mode

`enum kdrv_pinmux_mode`

Enumerations of KDP520 pinmux modes.

Enumerator

PIN_MODE_0	Pimux mode 0
PIN_MODE_1	Pimux mode 1
PIN_MODE_2	Pimux mode 2
PIN_MODE_3	Pimux mode 3, for GPIO mode only
PIN_MODE_4	Pimux mode 4
PIN_MODE_5	Pimux mode 5
PIN_MODE_6	Pimux mode 6
PIN_MODE_7	Pimux mode 7

Kneron Confidential

4.15.3 Function Documentation

4.15.3.1 kdrv_pinmux_config()

```
void kdrv_pinmux_config (
            kdrv_pin_name pin,
            kdrv_pinmux_mode mode,
            kdrv_pin_pull pull_type,
            kdrv_pin_driving driving )
```

Pinmux configure.

Parameters

in	*pin*	see kdrv_pin_name
in	*mode*	see kdrv_pinmux_mode
in	*pull_type*	see kdrv_pin_pull
in	*driving*	see kdrv_pin_driving

Returns

N/A

Kneron Confidential

4.16 KDRV_POWER

Kneron power driver.

Enumerations

- enum kdrv_power_domain_t { POWER_DOMAIN_DEFAULT = 1, POWER_DOMAIN_NPU, POWER_DOMAIN_DDRCK }

  Enumerations of kl520 power domains.
- enum kdrv_power_ops_t { POWER_OPS_FCS = 0, POWER_OPS_CHANGE_BUS_SPEED, POWER_OPS_PLL_UPDATE, POWER_OPS_SLEEPING }

  Enumerations of kl520 power operations.
- enum kdrv_power_mode_t { POWER_MODE_RTC = 0, POWER_MODE_ALWAYSON, POWER_MODE_FULL, POWER_MODE_RETENTION, POWER_MODE_DEEP_RETENTION }

  Enumerations of kl520 power modes.

Functions

- void kdrv_power_sw_reset (void)

  Watchdog reset.
- kdrv_status_t kdrv_power_ops (kdrv_power_ops_t ops)

  Power operation.
- kdrv_status_t kdrv_power_set_domain (kdrv_power_domain_t domain, int enable)

  Set power domain.
- kdrv_status_t kdrv_power_softoff (kdrv_power_mode_t mode)

  Shutdown the power supply to all blocks, except the logic in the RTC domain and DDR memory is in self-refresh state.

4.16.1 Detailed Description

Kneron power driver.

KL520 is a system with one NPU, two CPUs, and peripherals.
One CPU handles system requests such as host communication, camera video and display, and peripherals.
Another CPU assists NPU to do works like input image preprocessing and postprocessing.
Two CPUs use shared memory and interrupt for their communication (IPC).

Upon power-on, default power domain will be turned on and the initial bootloader code (IPL) in ROM starts to run on SCPU.
Once the secondary bootloader (SPL) is loaded to system internal SRAM by IPL, SPL will load SCPU OS and NCPU OS in SRAM and
pass over the execution SCPU OS.
Once NCPU OS is started (by SCPU OS), it will stay in idle thread and listen to commands from SCPU.
SCPU will also stay in idle thread and listen to host commands in companion mode or listen to user commands in standalone mode.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.16.2 Enumeration Type Documentation

4.16.2.1 kdrv_power_domain_t

enum kdrv_power_domain_t

Enumerations of kl520 power domains.

Enumerator

POWER_DOMAIN_DEFAULT	Power to Default power domain triggered by wake-up events
POWER_DOMAIN_NPU	Power to NPU power domain controlled by software
POWER_DOMAIN_DDRCK	Power to DDRCK power domain controlled by software

4.16.2.2 kdrv_power_mode_t

enum kdrv_power_mode_t

Enumerations of kl520 power modes.

Enumerator

POWER_MODE_RTC	RTC
POWER_MODE_ALWAYSON	RTC + Default
POWER_MODE_FULL	RTC + Default + DDR + NPU
POWER_MODE_RETENTION	RTC + Default + DDR(Self-refresh)
POWER_MODE_DEEP_RETENTION	RTC + DDR(Self-refresh)

4.16.2.3 kdrv_power_ops_t

enum kdrv_power_ops_t

Enumerations of kl520 power operations.

Enumerator

POWER_OPS_FCS	
POWER_OPS_CHANGE_BUS_SPEED	
POWER_OPS_PLL_UPDATE	
POWER_OPS_SLEEPING	

Kneron Confidential

4.16.3 Function Documentation

4.16.3.1 kdrv_power_ops()

```
kdrv_status_t kdrv_power_ops (
            kdrv_power_ops_t ops )
```

Power operation.

Parameters

in	*ops*	see kdrv_power_ops_t

Returns

kdrv_status_t see kdrv_status_t

4.16.3.2 kdrv_power_set_domain()

```
kdrv_status_t kdrv_power_set_domain (
            kdrv_power_domain_t domain,
            int enable )
```

Set power domain.

There are three powe domain in Kneron kl520 chip, see kdrv_power_domain_t

Parameters

in	*domain*	see kdrv_power_domain_t
in	*enable*	Enable the power domain

Returns

kdrv_status_t see kdrv_status_t

4.16.3.3 kdrv_power_softoff()

```
kdrv_status_t kdrv_power_softoff (
            kdrv_power_mode_t mode )
```

Shutdown the power supply to all blocks, except the logic in the RTC domain and DDR memory is in self-refresh state.

Kneron Confidential

Parameters

in	*mode*	see kdrv_power_mode_t

Returns

kdrv_status_t see kdrv_status_t

4.16.3.4 kdrv_power_sw_reset()

```
void kdrv_power_sw_reset (
            void  )
```

Watchdog reset.

Parameters

in	*N/A*	

Returns

N/A

Kneron Confidential

4.17 KDRV_PWM

Kneron PWM timer driver.

Macros

- #define APB_CLK APB_CLOCK
- #define PWMTMR_1000MSEC_PERIOD (uint32_t)(APB_CLK)
- #define PWMTMR_1000MSEC_PERIOD (uint32_t)(APB_CLK)
- #define PWMTMR_5000MSEC_PERIOD (uint32_t)(APB_CLK∗5)
- #define PWMTMR_100MSEC_PERIOD (uint32_t)(APB_CLK/10)
- #define PWMTMR_20MSEC_PERIOD (uint32_t)(APB_CLK/50)
- #define PWMTMR_15MSEC_PERIOD (uint32_t)(((APB_CLK/100)∗3)/2)
- #define PWMTMR_10MSEC_PERIOD (uint32_t)(APB_CLK/100)
- #define PWMTMR_1MSEC_PERIOD (uint32_t)(APB_CLK/1000)
- #define PWMTMR_01MSEC_PERIOD (uint32_t)(APB_CLK/10000)

Typedefs

- typedef enum Timer_IoType timeriotype

  Enumerations of kl520 power domains.

Enumerations

- enum pwmtimer {
  PWMTIMER1 =1, PWMTIMER2 =2, PWMTIMER3 =3, PWMTIMER4 =4,
  PWMTIMER5 =5, PWMTIMER6 =6 }

  Enumerations of all timer callback event return status.
- enum Timer_IoType { IO_TIMER_RESETALL, IO_TIMER_GETTICK, IO_TIMER_SETTICK, IO_TIMER_SETCLKSRC }

  Enumerations of kl520 power domains.
- enum pwmpolarity { PWM_POLARITY_NORMAL = 0, PWM_POLARITY_INVERSED }

  Enumerations of polarity of a PWM signal.

Functions

- uint32_t kdrv_current_t1_tick (void)

  Get t1 tick.
- uint32_t kdrv_current_t2_tick (void)

  Get t2 tick.
- uint32_t kdrv_current_t3_tick (void)

  Get t3 tick.
- uint32_t kdrv_current_t4_tick (void)

  Get t4 tick.
- uint32_t kdrv_current_t5_tick (void)

  Get t5 tick.
- uint32_t kdrv_current_t6_tick (void)

  Get t6 tick.
- kdrv_status_t kdrv_pwmtimer_initialize (pwmtimer timer, uint32_t tick)

Kneron Confidential

Initialize specific timer id and give tick.

- kdrv_status_t kdrv_pwmtimer_close (pwmtimer timer)

  Close specifiec pwm timer id.

- kdrv_status_t kdrv_pwmtimer_tick_reset (pwmtimer timer)

  Reset pwm timer tick.

- kdrv_status_t kdrv_pwmtimer_delay_ms (uint32_t msec)

  Use pwm timer to delay for certain time interval.

- kdrv_status_t kdrv_pwm_config (pwmtimer timer, pwmpolarity polarity, uint32_t duty, uint32_t period, bool ns2clkcnt)

  kdrv_pwm_config

- kdrv_status_t kdrv_pwm_enable (pwmtimer timer)

  kdrv_pwm_enable

- kdrv_status_t kdrv_pwm_disable (pwmtimer timer)

  kdrv_pwm_disable

4.17.1 Detailed Description

Kneron PWM timer driver.

Note

If you don't want to use pwm timer, you can refer to KDRV_TIMER

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.17.2 Macro Definition Documentation

4.17.2.1 APB_CLK

```
#define APB_CLK APB_CLOCK
```

4.17.2.2 PWMTMR_01MSEC_PERIOD

```
#define PWMTMR_01MSEC_PERIOD (uint32_t)(APB_CLK/10000)
```

4.17.2.3 PWMTMR_1000MSEC_PERIOD [1/2]

```
#define PWMTMR_1000MSEC_PERIOD (uint32_t)(APB_CLK)
```

Kneron Confidential

4.17.2.4 PWMTMR_1000MSEC_PERIOD [2/2]

```
#define PWMTMR_1000MSEC_PERIOD (uint32_t)(APB_CLK)
```

4.17.2.5 PWMTMR_100MSEC_PERIOD

```
#define PWMTMR_100MSEC_PERIOD (uint32_t)(APB_CLK/10)
```

4.17.2.6 PWMTMR_10MSEC_PERIOD

```
#define PWMTMR_10MSEC_PERIOD (uint32_t)(APB_CLK/100)
```

4.17.2.7 PWMTMR_15MSEC_PERIOD

```
#define PWMTMR_15MSEC_PERIOD (uint32_t)(((APB_CLK/100)*3)/2)
```

4.17.2.8 PWMTMR_1MSEC_PERIOD

```
#define PWMTMR_1MSEC_PERIOD (uint32_t)(APB_CLK/1000)
```

4.17.2.9 PWMTMR_20MSEC_PERIOD

```
#define PWMTMR_20MSEC_PERIOD (uint32_t)(APB_CLK/50)
```

4.17.2.10 PWMTMR_5000MSEC_PERIOD

```
#define PWMTMR_5000MSEC_PERIOD (uint32_t)(APB_CLK*5)
```

4.17.3 Typedef Documentation

Kneron Confidential

4.17.3.1 timeriotype

```
typedef enum Timer_IoType timeriotype
```

Enumerations of kl520 power domains.

4.17.4 Enumeration Type Documentation

4.17.4.1 pwmpolarity

```
enum pwmpolarity
```

Enumerations of polarity of a PWM signal.

Enumerator

PWM_POLARITY_NORMAL	A high signal for the duration of the duty-cycle, followed by a low signal for the remainder of the pulse period
PWM_POLARITY_INVERSED	A low signal for the duration of the duty-cycle, followed by a high signal for the remainder of the pulse period

4.17.4.2 pwmtimer

```
enum pwmtimer
```

Enumerations of all timer callback event return status.

Enumerator

PWMTIMER1	PWM timer callback instance 1
PWMTIMER2	PWM timer callback instance 2
PWMTIMER3	PWM timer callback instance 3
PWMTIMER4	PWM timer callback instance 4
PWMTIMER5	PWM timer callback instance 5
PWMTIMER6	PWM timer callback instance 6

4.17.4.3 Timer_IoType

```
enum Timer_IoType
```

Enumerations of kl520 power domains.

Kneron Confidential

Enumerator

IO_TIMER_RESETALL	
IO_TIMER_GETTICK	
IO_TIMER_SETTICK	
IO_TIMER_SETCLKSRC	

4.17.5 Function Documentation

4.17.5.1 kdrv_current_t1_tick()

```
uint32_t kdrv_current_t1_tick (
            void  )
```

Get t1 tick.

Parameters

in	*N/A*	

Returns

t1_tick

4.17.5.2 kdrv_current_t2_tick()

```
uint32_t kdrv_current_t2_tick (
            void  )
```

Get t2 tick.

Parameters

in	*N/A*	

Returns

t2_tick

Kneron Confidential

4.17.5.3 kdrv_current_t3_tick()

```
uint32_t kdrv_current_t3_tick (
            void  )
```

Get t3 tick.

Parameters

in	*N/A*	

Returns

t3_tick

4.17.5.4 kdrv_current_t4_tick()

```
uint32_t kdrv_current_t4_tick (
            void  )
```

Get t4 tick.

Parameters

in	*N/A*	

Returns

t4_tick

4.17.5.5 kdrv_current_t5_tick()

```
uint32_t kdrv_current_t5_tick (
            void  )
```

Get t5 tick.

Parameters

in	*N/A*	

Returns

t5_tick

Kneron Confidential

4.17.5.6 kdrv_current_t6_tick()

```
uint32_t kdrv_current_t6_tick (
            void  )
```

Get t6 tick.

Parameters

in	*N/A*	

Returns

t6_tick

4.17.5.7 kdrv_pwm_config()

```
kdrv_status_t kdrv_pwm_config (
            pwmtimer timer,
            pwmpolarity polarity,
            uint32_t duty,
            uint32_t period,
            bool ns2clkcnt )
```

kdrv_pwm_config

After config pwm timer via this API, you should call kdrv_pwm_enable() to let pwm timer work well.

Parameters

in	*timer*	pwm timer id, see pwmtimer
in	*polarity*	polarity, see pwmpolarity
in	*duty_ms*	duty cycle(ms)
in	*period_ms*	period(ms)

Returns

kdrv_status_t see kdrv_status_t

Note

Example:
kdrv_pwm_config(PWMTIMER1, PWM_POLARITY_NORMAL, duty, PWM0_FREQ_CNT, 0);
kdrv_pwm_enable(PWMTIMER1);

Kneron Confidential

4.17.5.8 kdrv_pwm_disable()

```
kdrv_status_t kdrv_pwm_disable (
            pwmtimer timer )
```

kdrv_pwm_disable

Parameters

in	*timer*	pwm timer id, see pwmtimer

Returns

kdrv_status_t see kdrv_status_t

4.17.5.9 kdrv_pwm_enable()

```
kdrv_status_t kdrv_pwm_enable (
            pwmtimer timer )
```

kdrv_pwm_enable

Parameters

in	*timer*	pwm timer id, see pwmtimer

Returns

kdrv_status_t see kdrv_status_t

4.17.5.10 kdrv_pwmtimer_close()

```
kdrv_status_t kdrv_pwmtimer_close (
            pwmtimer timer )
```

Close specifiec pwm timer id.

Parameters

in	*Timer↩ Id*	pwm timer id, see pwmtimer

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

4.17.5.11 kdrv_pwmtimer_delay_ms()

```
kdrv_status_t kdrv_pwmtimer_delay_ms (
            uint32_t msec )
```

Use pwm timer to delay for certain time interval.

Parameters

in	*Timer↩ Id*	timer id

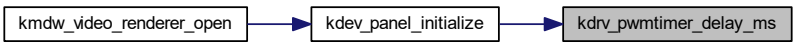
Returns

kdrv_status_t see kdrv_status_t

Note

This API uses PWMTIMER1 to run pwm timer tick.
You should avoid to use the duplicated pwmtimer.
If you don't want to occupy one PWM timer instance, please refer to use kdrv_timer_delay_ms()

Here is the caller graph for this function:



4.17.5.12 kdrv_pwmtimer_initialize()

```
kdrv_status_t kdrv_pwmtimer_initialize (
            pwmtimer timer,
            uint32_t tick )
```

Initialize specific timer id and give tick.

Kneron Confidential

Parameters

in	*Timer↩ Id*	pwm timer id, see pwmtimer
in	*tick*	tick number

Returns

kdrv_status_t see kdrv_status_t

4.17.5.13 kdrv_pwmtimer_tick_reset()

```
kdrv_status_t kdrv_pwmtimer_tick_reset (
            pwmtimer timer )
```

Reset pwm timer tick.

Parameters

in	*Timer↩ Id*	pwm timer id, see pwmtimer

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.18 KDRV_SDC

Kneron sdc sd/emmc driver.

Data Structures

- struct kdrv_sdc_reg_t
- struct kdrv_sdc_flow_info_t
- struct kdrv_sd_status_t
- struct kdrv_sdc_csd_v1_t
- struct kdrv_sdc_csd_v2_t
- struct kdrv_sdc_sd_scr_t
- struct kdrv_sdc_sdcard_info_t
- struct kdrv_sdc_sd_host_t
- struct kdrv_sdc_adma2desc_table_t
- struct kdrv_sdc_res_t
- struct kdrv_sdc_mmc_csd_t
- struct kdrv_sdc_mmc_ext_csd_t

Macros

- #define ADMA2_NUM_OF_LINES 64
- #define CARD_TYPE_UNKNOWN 0
- #define MEMORY_CARD_TYPE_SD 1
- #define MEMORY_CARD_TYPE_MMC 2
- #define SDIO_TYPE_CARD 3
- #define MEMORY_SDIO_COMBO 4
- #define SCR_LENGTH 8
- #define SD_STATUS_LENGTH 64
- #define EXT_CSD_LENGTH 512
- #define SDHCI_SCR_SUPPORT_4BIT_BUS 0x4
- #define SDHCI_SCR_SUPPORT_1BIT_BUS 0x1
- #define WAIT_CMD_COMPLETE BIT(0)
- #define WAIT_TRANS_COMPLETE BIT(1)
- #define WAIT_DMA_INTR BIT(2)
- #define WAIT_BLOCK_GAP BIT(3)
- #define KDRV_SDC_BASE SDC_FTSDC021_PA_BASE
- #define SDHCI_TXMODE_DMA_EN BIT(0)
- #define SDHCI_TXMODE_BLKCNT_EN BIT(1)
- #define SDHCI_TXMODE_AUTOCMD12_EN BIT(2)
- #define SDHCI_TXMODE_AUTOCMD23_EN (2 << 2)
- #define SDHCI_TXMODE_READ_DIRECTION BIT(4)
- #define SDHCI_TXMODE_WRITE_DIRECTION (0 << 4)
- #define SDHCI_TXMODE_MULTI_SEL BIT(5)
- #define SDHCI_CMD_IDX_SHIFT 0x08
- #define SDHCI_CMD_TYPE_SHIFT 0x06
- #define SDHCI_CMD_DATA_PRESEL_SHIFT 0x05
- #define SDHCI_CMD_NO_RESPONSE 0x00
- #define SDHCI_CMD_RTYPE_R2 0x09
- #define SDHCI_CMD_RTYPE_R3R4 0x02
- #define SDHCI_CMD_RTYPE_R1R5R6R7 0x1A
- #define SDHCI_CMD_RTYPE_R1BR5B 0x1B

Kneron Confidential

- #define SDHCI_CMD_TYPE_NORMAL 0x00
- #define SDHCI_CMD_TYPE_SUSPEND 0x01
- #define SDHCI_CMD_TYPE_RESUME 0x02
- #define SDHCI_CMD_TYPE_ABORT 0x03
- #define SDHCI_CMD_DATA_PRESENT 0x01
- #define SDHCI_REG_DATA_PORT 0x20
- #define SDHCI_REG_PRE_STATE 0x24
- #define SDHCI_STS_CMD_INHIBIT BIT(0)
- #define SDHCI_STS_CMD_DAT_INHIBIT BIT(1)
- #define SDHCI_STS_DAT_LINE_ACT BIT(2)
- #define SDHCI_STS_WRITE_TRAN_ACT BIT(8)
- #define SDHCI_STS_READ_TRAN_ACT BIT(9)
- #define SDHCI_STS_BUFF_WRITE BIT(10)
- #define SDHCI_STS_BUFF_READ BIT(11)
- #define SDHCI_STS_CARD_INSERT BIT(16)
- #define SDHCI_STS_CARD_STABLE BIT(17)
- #define SDHCI_STS_CARD_WP BIT(19)
- #define SDHCI_STS_DAT_LINE_LEVEL (0xF << 20)
- #define SDHCI_STS_CMD_LINE_LEVEL BIT(24)
- #define SDHCI_REG_HC 0x28
- #define SDHCI_HC_LED_ON BIT(0)
- #define SDHCI_HC_BUS_WIDTH_4BIT BIT(1)
- #define SDHCI_HC_HI_SPEED BIT(2)
- #define SDHCI_HC_USE_ADMA2 BIT(3)
- #define SDHCI_HC_BUS_WIDTH_8BIT BIT(5)
- #define SDHCI_HC_CARD_DETECT_TEST BIT(6)
- #define SDHCI_HC_CARD_DETECT_SIGNAL BIT(7)
- #define SDHCI_POWER_ON BIT(0)
- #define SDHCI_POWER_180 (5 << 1)
- #define SDHCI_POWER_300 (6 << 1)
- #define SDHCI_POWER_330 (7 << 1)
- #define SDHCI_STOP_AT_BLOCK_GAP_REQ BIT(0)
- #define SDHCI_CONTINUE_REQ BIT(1)
- #define SDHCI_READ_WAIT_CTL BIT(2)
- #define SDHCI_INT_AT_BLOCK_GAP BIT(3)
- #define SDHCI_REG_CLK_CTRL 0x2C
- #define SDHCI_CLK_CTRL_LOW_CLK_SEL_SHIFT 8
- #define SDHCI_CLK_CTRL_UP_CLK_SEL_SHIFT 6
- #define SDHCI_CLK_CTRL_INTERNALCLK_EN BIT(0)
- #define SDHCI_CLK_CTRL_INTERNALCLK_STABLE BIT(1)
- #define SDHCI_CLK_CTRL_SDCLK_EN BIT(2)
- #define SDHCI_CLK_CTRL_CLK_GEN_SEL_PRO BIT(5)
- #define SDHCI_SOFTRST_ALL BIT(0)
- #define SDHCI_SOFTRST_CMD BIT(1)
- #define SDHCI_SOFTRST_DAT BIT(2)
- #define SDHCI_REG_INTR_STATE 0x30
- #define SDHCI_INTR_STS_ERR BIT(15)
- #define SDHCI_INTR_STS_CARD_INTR BIT(8)
- #define SDHCI_INTR_STS_CARD_REMOVE BIT(7)
- #define SDHCI_INTR_STS_CARD_INSERT BIT(6)
- #define SDHCI_INTR_STS_BUFF_READ_READY BIT(5)
- #define SDHCI_INTR_STS_BUFF_WRITE_READY BIT(4)
- #define SDHCI_INTR_STS_DMA BIT(3)
- #define SDHCI_INTR_STS_BLKGAP BIT(2)
- #define SDHCI_INTR_STS_TXR_COMPLETE BIT(1)

Kneron Confidential

- #define SDHCI_INTR_STS_CMD_COMPLETE BIT(0) /* CMD completed, CMD12/CMD23 will not generate this command */
- #define SDHCI_INTR_ERR_TUNING BIT(10)
- #define SDHCI_INTR_ERR_ADMA BIT(9)
- #define SDHCI_INTR_ERR_AUTOCMD BIT(8)
- #define SDHCI_INTR_ERR_CURR_LIMIT BIT(7)
- #define SDHCI_INTR_ERR_DATA_ENDBIT BIT(6)
- #define SDHCI_INTR_ERR_DATA_CRC BIT(5)
- #define SDHCI_INTR_ERR_DATA_TIMEOUT BIT(4)
- #define SDHCI_INTR_ERR_CMD_INDEX BIT(3)
- #define SDHCI_INTR_ERR_CMD_ENDBIT BIT(2)
- #define SDHCI_INTR_ERR_CMD_CRC BIT(1)
- #define SDHCI_INTR_ERR_CMD_TIMEOUT BIT(0)
- #define SDHCI_INTR_ERR_CMD_LINE (SDHCI_INTR_ERR_CMD_INDEX | SDHCI_INTR_ERR_CMD_ENDBIT | SDHCI_INTR_ERR_CMD_CRC | SDHCI_INTR_ERR_CMD_TIMEOUT)
- #define SDHCI_INTR_ERR_DAT_LINE (SDHCI_INTR_ERR_DATA_ENDBIT | SDHCI_INTR_ERR_DATA_CRC | SDHCI_INTR_ERR_DATA_TIMEOUT)
- #define SDHCI_INTR_EN_ALL (0x10FF)
- #define SDHCI_ERR_EN_ALL (0xF7FF)
- #define SDHCI_INTR_SIG_EN (SDHCI_INTR_STS_CARD_REMOVE | SDHCI_INTR_STS_CARD_INSERT | SDHCI_INTR_STS_CMD_COMPLETE | SDHCI_INTR_STS_TXR_COMPLETE)
- #define SDHCI_INTR_SIGN_EN_SDMA (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_DMA | SDHCI_INTR_STS_BLKGAP)
- #define SDHCI_INTR_SIGN_EN_ADMA (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_DMA)
- #define SDHCI_INTR_SIGN_EN_PIO (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_BLKGAP)
- #define SDHCI_ERR_SIG_EN_ALL (0xF3FF)
- #define SDHCI_AUTOCMD12_ERR_NOT_EXECUTED BIT(0)
- #define SDHCI_AUTOCMD12_ERR_TIMEOUT BIT(1)
- #define SDHCI_AUTOCMD12_ERR_CRC BIT(2)
- #define SDHCI_AUTOCMD12_ERR_END_BIT BIT(3)
- #define SDHCI_AUTOCMD12_ERR_INDEX BIT(4)
- #define SDHCI_AUTOCMD12_ERR_CMD_NOT_ISSUE BIT(7)
- #define SDHCI_REG_HOST_CTRL2 0x3E
- #define SDHCI_PRESET_VAL_EN BIT(15)
- #define SDHCI_ASYNC_INT_EN BIT(14)
- #define SDHCI_SMPL_CLCK_SELECT BIT(7)
- #define SDHCI_EXECUTE_TUNING BIT(6) /* Write 1 Auto clear */
- #define SDHCI_DRV_TYPE_MASK BIT(4)
- #define SDHCI_DRV_TYPE_SHIFT 4
- #define SDHCI_DRV_TYPEB 0
- #define SDHCI_DRV_TYPEA 1
- #define SDHCI_DRV_TYPEC 2
- #define SDHCI_DRV_TYPED 3
- #define SDHCI_18V_SIGNAL BIT(3)
- #define SDHCI_UHS_MODE_MASK (7 << 0)
- #define SDHCI_SDR12 0
- #define SDHCI_SDR25 1
- #define SDHCI_SDR50 2
- #define SDHCI_SDR104 3
- #define SDHCI_DDR50 4
- #define SDHCI_CAP_VOLTAGE_33V BIT(24)
- #define SDHCI_CAP_VOLTAGE_30V BIT(25)
- #define SDHCI_CAP_VOLTAGE_18V BIT(26)
- #define SDHCI_CAP_FIFO_DEPTH_16BYTE (0 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_32BYTE (1 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_64BYTE (2 << 29)

Kneron Confidential

- #define SDHCI_CAP_FIFO_DEPTH_512BYTE (3 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_1024BYTE (4 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_2048BYTE (5 << 29)
- #define SDHCI_SUPPORT_SDR50 BIT(0)
- #define SDHCI_SUPPORT_SDR104 BIT(1)
- #define SDHCI_SUPPORT_DDR50 BIT(2)
- #define SDHCI_SUPPORT_DRV_TYPEA BIT(4)
- #define SDHCI_SUPPORT_DRV_TYPEC BIT(5)
- #define SDHCI_SUPPORT_DRV_TYPED BIT(6)
- #define SDHCI_RETUNING_TIME_MAS 0xF
- #define SDHCI_RETUNING_TIME_SHIFT 8
- #define SDHCI_SDR50_TUNING BIT(13)
- #define SDCHI_RETUNING_MODE_MASK 0x3
- #define SDHCI_RETUNING_MODE_SHIFT 14
- #define MMC_BOOT_ACK BIT(2)
- #define MMC_BUS_TEST_MODE 0x3
- #define MMC_ALTERNATIVE_BOOT_MODE 0x2
- #define MMC_BOOT_MODE 0x1
- #define NORMAL_MODE 0x0
- #define SDHCI_CMD0_GO_IDLE_STATE 0
- #define SDHCI_CMD1_MMC_SEND_OP_COND 1
- #define SDHCI_CMD2_SEND_ALL_CID 2
- #define SDHCI_CMD3_SEND_RELATIVE_ADDR 3
- #define SDHCI_CMD5_IO_SEND_OP_COND 5
- #define SDHCI_CMD6_SWITCH_FUNC 6
- #define SDHCI_CMD6_SET_BUS_WIDTH 6
- #define SDHCI_CMD7_SELECT_CARD 7
- #define SDHCI_CMD8_SEND_IF_COND 8
- #define SDHCI_CMD8_SEND_EXT_CSD 8
- #define SDHCI_CMD9_SEND_CSD 9
- #define SDHCI_CMD10_SEND_CID 10
- #define SDHCI_CMD11_VOLTAGE_SWITCH 11
- #define SDHCI_CMD12_STOP_TRANS 12
- #define SDHCI_CMD13_SEND_STATUS 13
- #define SDHCI_CMD13_SD_STATUS 13
- #define SDHCI_CMD16_SET_BLOCKLEN 16
- #define SDHCI_CMD17_READ_SINGLE_BLOCK 17
- #define SDHCI_CMD18_READ_MULTI_BLOCK 18
- #define SDHCI_CMD19_SEND_TUNE_BLOCK 19
- #define SDHCI_CMD23_SET_WR_BLOCK_CNT 23
- #define SDHCI_CMD24_WRITE_BLOCK 24
- #define SDHCI_CMD25_WRITE_MULTI_BLOCK 25
- #define SDHCI_CMD32_ERASE_WR_BLK_START 32
- #define SDHCI_CMD33_ERASE_WR_BLK_END 33
- #define SDHCI_CMD35_ERASE_GROUP_START 35
- #define SDHCI_CMD36_ERASE_GROUP_END 36
- #define SDHCI_CMD38_ERASE 38
- #define SDHCI_CMD41_SD_SEND_OP_COND 41
- #define SDHCI_CMD43_GET_MKB 43
- #define SDHCI_CMD44_GET_MID 44
- #define SDHCI_CMD45_CER_RN1 45
- #define SDHCI_CMD46_CER_RN2 46
- #define SDHCI_CMD47_CER_RES2 47
- #define SDHCI_CMD48_CER_RES1 48
- #define SDHCI_CMD51_SEND_SCR 51

Kneron Confidential

- #define SDHCI_CMD52_IO_RW_DIRECT 52
- #define SDHCI_CMD53_IO_RW_EXTENDED 53
- #define SDHCI_CMD55_APP 55
- #define SDHCI_CMD56_GEN 56
- #define SDHCI_CMD8_SEND_IF_COND_ARGU 0x1AA
- #define SDHCI_CMD41_SD_SEND_OP_COND_HCS_ARGU 0xC0FF8000
- #define SDHCI_CMD41_SD_SEND_OP_COND_ARGU 0x00FF8000
- #define SDHCI_CMD1_MMC_SEND_OP_COND_BYTE_MODE 0x80FF8000
- #define SDHCI_CMD1_MMC_SEND_OP_COND_SECTOR_MODE 0xC0FF8000
- #define CMD_RETRY_CNT 5
- #define SDHCI_TIMEOUT 0xFFF
- #define SD_CMD52_RW_in_W 0x80000000
- #define SD_CMD52_RW_in_R 0x00000000
- #define SD_CMD52_RAW 0x08000000
- #define SD_CMD52_no_RAW 0x00000000
- #define SD_CMD52_FUNC(Num) (Num << 28)
- #define SD_CMD52_Reg_Addr(Addr) (Addr << 9)
- #define SD_CMD53_RW_in_W 0x80000000
- #define SD_CMD53_RW_in_R 0x00000000
- #define SD_CMD53_FUNC(Num) (Num << 28)
- #define SD_CMD53_Block_Mode 0x08000000
- #define SD_CMD53_Byte_Mode 0x00000000
- #define SD_CMD53_OP_inc 0x04000000
- #define SD_CMD53_OP_fix 0x00000000
- #define SD_CMD53_Reg_Addr(Addr) (Addr << 9)
- #define SD_STATUS_OUT_OF_RANGE 0x80000000
- #define SD_STATUS_ADDRESS_ERROR BIT(30)
- #define SD_STATUS_BLOCK_LEN_ERROR BIT(29)
- #define SD_STATUS_ERASE_SEQ_ERROR BIT(28)
- #define SD_STATUS_ERASE_PARAM BIT(27)
- #define SD_STATUS_WP_VIOLATION BIT(26)
- #define SD_STATUS_CARD_IS_LOCK BIT(25)
- #define SD_STATUS_LOCK_UNLOCK_FAILED BIT(24)
- #define SD_STATUS_COM_CRC_ERROR BIT(23)
- #define SD_STATUS_ILLEGAL_COMMAND BIT(22)
- #define SD_STATUS_CARD_ECC_FAILED BIT(21)
- #define SD_STATUS_CC_ERROR BIT(20)
- #define SD_STATUS_ERROR BIT(19)
- #define SD_STATUS_UNDERRUN BIT(18)
- #define SD_STATUS_OVERRUN BIT(17)
- #define SD_STATUS_CSD_OVERWRITE BIT(16)
- #define SD_STATUS_WP_ERASE_SKIP BIT(15)
- #define SD_STATUS_CARD_ECC_DISABLE BIT(14)
- #define SD_STATUS_ERASE_RESET BIT(13)
- #define SD_STATUS_CURRENT_STATE (0xF << 9)
- #define SD_STATUS_READY_FOR_DATA BIT(8)
- #define MMC_STATUS_SWITCH_ERROR BIT(7)
- #define SD_STATUS_APP_CMD BIT(5)
- #define SD_STATUS_AKE_SEQ_ERROR BIT(3)
- #define SD_STATUS_ERROR_BITS
- #define SDHCI_1BIT_BUS_WIDTH 0x0
- #define SDHCI_4BIT_BUS_WIDTH 0x2
- #define ADMA2_ENTRY_VALID BIT(0)
- #define ADMA2_ENTRY_END BIT(1)
- #define ADMA2_ENTRY_INT BIT(2)

Kneron confidential

- #define ADMA2_NOP (0 << 4)
- #define ADMA2_SET (1 << 4)
- #define ADMA2_TRAN (2 << 4)
- #define ADMA2_LINK (3 << 4)
- #define SDHCI_MMC_SWITCH 6
- #define SDHCI_MMC_VENDOR_CMD 62
- #define EXT_CSD_CMD_SET_NORMAL (1<<0)
- #define EXT_CSD_CMD_SET_SECURE (1<<1)
- #define EXT_CSD_CMD_SET_CPSECURE (1<<2)
- #define EXT_CSD_PARTITION_SETTING_COMPLETED 156
- #define EXT_CSD_PARTITION_CONF 179
- #define EXT_CSD_BUS_WIDTH 183 /* R/W */
- #define EXT_CSD_HS_TIMING 185 /* R/W */
- #define EXT_CSD_CARD_TYPE 196 /* RO */
- #define EXT_CSD_SEC_CNT 212 /* RO, 4 bytes */
- #define EXT_CSD_BOOT_SIZE_MULT 226
- #define EXT_CSD_CMD_SET 0x0
- #define EXT_CSD_SET_BIT 0x1
- #define EXT_CSD_CLR_BYTE 0x2
- #define EXT_CSD_WRITE_BYTE 0x3
- #define EXT_CSD_BUS_8BIT 0x2
- #define EXT_CSD_BUS_4BIT 0x1
- #define EXT_CSD_BUS_1BIT 0x0
- #define MMC_CMD6_ACCESS_MODE(x) (uint32_t)(x << 24)
- #define MMC_CMD6_INDEX(x) (uint32_t)(x << 16)
- #define MMC_CMD6_VALUE(x) (uint32_t)(x << 8)
- #define MMC_CMD6_CMD_SET(x) (uint32_t)(x)
- #define MMC_CARD_BUSY 0x80000000 /* Card Power up status bit */

Enumerations

- enum kdrv_sdc_infinite_test_e { INFINITE_NO = 0, INFINITE_MODE_1, INFINITE_MODE_2 }
- enum kdrv_sdc_transfer_act_e { WRITE = 0, READ }
- enum kdrv_sdc_transfer_type_e {
  ADMA = 0, SDMA, PIO, EDMA,
  TRANS_UNKNOWN }
- enum kdrv_sdc_abort_type_e { ABORT_ASYNCHRONOUS = 0, ABORT_SYNCHRONOUS, ABORT_UNDEFINED
  }
- enum kdrv_sdc_cprm_test_e { CPRM_PROTECT_RW, CPRM_FILESYS, CPRM_UNKNOWN }
- enum kdrv_sdc_bus_speed_e {
  SPEED_DEFAULT = 0, SPEED_SDR25, SPEED_SDR50, SPEED_SDR104,
  SPEED_DDR50, SPEED_RSRV }
- enum kdrv_sdc_card_state_e {
  CUR_STATE_IDLE = 0, CUR_STATE_READY, CUR_STATE_IDENT, CUR_STATE_STBY,
  CUR_STATE_TRAN, CUR_STATE_DATA, CUR_STATE_RCV, CUR_STATE_PRG,
  CUR_STATE_DIS, CUR_STATE_RSV }

Kneron Confidential

Functions

- kdrv_status_t kdrv_sdc_initialize (void)

  kdrv_sdc_initialize, initail sd/emmc card interface
- kdrv_status_t kdrv_sdc_uninitialize (void)

  kdrv_sdc_uninitialize, uninitail sd/emmc card interface and resource
- kdrv_status_t kdrv_sdc_dev_scan (void)

  kdrv_sdc_dev_scan() scan sd/mmc memory card
- kdrv_sdc_res_t * kdrv_sdc_get_dev (void)

  kdrv_sdc_get_dev() get device structure
- kdrv_status_t kdrv_sdc_read (uint8_t *buf, uint32_t sd_offset, uint32_t size)

  kdrv_sdc_read read data from sd/mmc card
- kdrv_status_t kdrv_sdc_write (uint8_t *buf, uint32_t sd_offset, uint32_t size)

  kdrv_sdc_write write data from sd/mmc card

Variables

- uint32_t kdrv_sdc_reg_t::sdma_addr
- uint16_t kdrv_sdc_reg_t::blk_size
- uint16_t kdrv_sdc_reg_t::blk_cnt
- uint32_t kdrv_sdc_reg_t::cmd_argu
- uint16_t kdrv_sdc_reg_t::txmode
- uint16_t kdrv_sdc_reg_t::cmd_reg
- uint64_t kdrv_sdc_reg_t::cmd_resplo
- uint64_t kdrv_sdc_reg_t::cmd_resphi
- uint32_t kdrv_sdc_reg_t::buf_data
- uint32_t kdrv_sdc_reg_t::present_state
- uint8_t kdrv_sdc_reg_t::hcreg
- uint8_t kdrv_sdc_reg_t::pwr_ctl
- uint8_t kdrv_sdc_reg_t::blk_gap_ctl
- uint8_t kdrv_sdc_reg_t::wakeup_ctl
- uint16_t kdrv_sdc_reg_t::clk_ctl
- uint8_t kdrv_sdc_reg_t::timeout_ctl
- uint8_t kdrv_sdc_reg_t::softrst
- uint16_t kdrv_sdc_reg_t::intr_sts
- uint16_t kdrv_sdc_reg_t::err_sts
- uint16_t kdrv_sdc_reg_t::intr_en
- uint16_t kdrv_sdc_reg_t::err_en
- uint16_t kdrv_sdc_reg_t::intr_sig_en
- uint16_t kdrv_sdc_reg_t::err_sig_en
- uint16_t kdrv_sdc_reg_t::auto_cmd_err
- uint16_t kdrv_sdc_reg_t::host_ctl2
- uint32_t kdrv_sdc_reg_t::cap_reg
- uint32_t kdrv_sdc_reg_t::cap_reg2
- uint64_t kdrv_sdc_reg_t::max_curr
- uint16_t kdrv_sdc_reg_t::cmd12_force_evt
- uint16_t kdrv_sdc_reg_t::force_evt
- uint32_t kdrv_sdc_reg_t::adma_err_sts
- uint64_t kdrv_sdc_reg_t::adma_addr
- uint16_t kdrv_sdc_reg_t::preset_val_init
- uint16_t kdrv_sdc_reg_t::preset_val_ds
- uint16_t kdrv_sdc_reg_t::preset_val_hs
- uint16_t kdrv_sdc_reg_t::preset_val_sdr12

Kneron Confidential

- uint16_t kdrv_sdc_reg_t::preset_val_sdr25
- uint16_t kdrv_sdc_reg_t::preset_val_sdr50
- uint16_t kdrv_sdc_reg_t::preset_val_sdr104
- uint16_t kdrv_sdc_reg_t::preset_val_ddr50
- uint32_t kdrv_sdc_reg_t::reserved [28]
- uint32_t kdrv_sdc_reg_t::share_bus_ctl
- uint32_t kdrv_sdc_reg_t::reserved2 [6]
- uint16_t kdrv_sdc_reg_t::slt_intr_sts
- uint16_t kdrv_sdc_reg_t::hcver
- uint32_t kdrv_sdc_reg_t::vendor_reg0
- uint32_t kdrv_sdc_reg_t::vendor_reg1
- uint32_t kdrv_sdc_reg_t::vendor_reg2
- uint32_t kdrv_sdc_reg_t::vendor_reg3
- uint32_t kdrv_sdc_reg_t::vendor_reg4
- uint32_t kdrv_sdc_reg_t::vendor_reg5
- uint32_t kdrv_sdc_reg_t::vendor_reg6
- uint32_t kdrv_sdc_reg_t::ahb_err_sts
- uint32_t kdrv_sdc_reg_t::ahb_err_en
- uint32_t kdrv_sdc_reg_t::ahb_err_sig_en
- uint32_t kdrv_sdc_reg_t::dma_hndshk
- uint32_t kdrv_sdc_reg_t::reserved4 [19]
- uint32_t kdrv_sdc_reg_t::hw_attr
- uint32_t kdrv_sdc_reg_t::ip_ver
- uint32_t kdrv_sdc_reg_t::ciph_m_ctl
- uint32_t kdrv_sdc_reg_t::ciph_m_sts
- uint16_t kdrv_sdc_reg_t::ciph_m_sts_en
- uint16_t kdrv_sdc_reg_t::ciph_m_sig_en
- uint32_t kdrv_sdc_reg_t::in_data_lo
- uint32_t kdrv_sdc_reg_t::in_data_hi
- uint32_t kdrv_sdc_reg_t::in_key_lo
- uint32_t kdrv_sdc_reg_t::in_key_hi
- uint32_t kdrv_sdc_reg_t::out_data_lo
- uint32_t kdrv_sdc_reg_t::out_data_hi
- uint32_t kdrv_sdc_reg_t::secr_table_port
- kdrv_sdc_transfer_type_e kdrv_sdc_flow_info_t::use_dma
- uint16_t kdrv_sdc_flow_info_t::line_bound
- uint16_t kdrv_sdc_flow_info_t::adma2rand
- kdrv_sdc_abort_type_e kdrv_sdc_flow_info_t::sync_abort
- uint8_t kdrv_sdc_flow_info_t::erasing
- uint8_t kdrv_sdc_flow_info_t::auto_cmd
- uint8_t kdrv_sdc_flow_info_t::reserved
- uint32_t kdrv_sd_status_t::reserved1:5
- uint32_t kdrv_sd_status_t::secured_mode:1
- uint32_t kdrv_sd_status_t::dat_bus_width:2
- uint32_t kdrv_sd_status_t::sd_card_type_hi:8
- uint32_t kdrv_sd_status_t::reserved2:8
- uint32_t kdrv_sd_status_t::sd_card_type_lo:8
- uint32_t kdrv_sd_status_t::size_of_protected_area
- uint8_t kdrv_sd_status_t::speed_class
- uint8_t kdrv_sd_status_t::performance_move
- uint32_t kdrv_sd_status_t::reserved3:4
- uint32_t kdrv_sd_status_t::au_size:4
- uint8_t kdrv_sd_status_t::erase_size [2]
- uint32_t kdrv_sd_status_t::erase_offset:2
- uint32_t kdrv_sd_status_t::erase_timeout:6

Kneron Confidential

- uint8_t kdrv_sd_status_t::reserved4 [11]
- uint8_t kdrv_sd_status_t::reserved5 [39]
- uint32_t kdrv_sdc_csd_v1_t::csd_structure:2
- uint32_t kdrv_sdc_csd_v1_t::reserved1:6
- uint8_t kdrv_sdc_csd_v1_t::taac
- uint8_t kdrv_sdc_csd_v1_t::nsac
- uint8_t kdrv_sdc_csd_v1_t::tran_speed
- uint32_t kdrv_sdc_csd_v1_t::ccc:12
- uint32_t kdrv_sdc_csd_v1_t::read_bl_len:4
- uint32_t kdrv_sdc_csd_v1_t::read_bl_partial:1
- uint32_t kdrv_sdc_csd_v1_t::write_blk_misalign:1
- uint32_t kdrv_sdc_csd_v1_t::read_blk_misalign:1
- uint32_t kdrv_sdc_csd_v1_t::dsr_imp:1
- uint32_t kdrv_sdc_csd_v1_t::reserved2:2
- uint32_t kdrv_sdc_csd_v1_t::c_size:12
- uint32_t kdrv_sdc_csd_v1_t::vdd_r_curr_min:3
- uint32_t kdrv_sdc_csd_v1_t::vdd_r_curr_max:3
- uint32_t kdrv_sdc_csd_v1_t::vdd_w_curr_min:3
- uint32_t kdrv_sdc_csd_v1_t::vdd_w_curr_max:3
- uint32_t kdrv_sdc_csd_v1_t::c_size_mult:3
- uint32_t kdrv_sdc_csd_v1_t::erase_blk_en:1
- uint32_t kdrv_sdc_csd_v1_t::sector_size:7
- uint32_t kdrv_sdc_csd_v1_t::wp_grp_size:7
- uint32_t kdrv_sdc_csd_v1_t::wp_grp_enable:1
- uint32_t kdrv_sdc_csd_v1_t::reserved3:2
- uint32_t kdrv_sdc_csd_v1_t::r2w_factor:3
- uint32_t kdrv_sdc_csd_v1_t::write_bl_len:4
- uint32_t kdrv_sdc_csd_v1_t::write_bl_partial:1
- uint32_t kdrv_sdc_csd_v1_t::reserved4:5
- uint32_t kdrv_sdc_csd_v1_t::file_format_grp:1
- uint32_t kdrv_sdc_csd_v1_t::copy:1
- uint32_t kdrv_sdc_csd_v1_t::perm_write_protect:1
- uint32_t kdrv_sdc_csd_v1_t::tmp_write_protect:1
- uint32_t kdrv_sdc_csd_v1_t::file_format:2
- uint32_t kdrv_sdc_csd_v1_t::Reserver5:2
- uint32_t kdrv_sdc_csd_v2_t::csd_structure:2
- uint32_t kdrv_sdc_csd_v2_t::reserved1:6
- uint8_t kdrv_sdc_csd_v2_t::taac
- uint8_t kdrv_sdc_csd_v2_t::nsac
- uint8_t kdrv_sdc_csd_v2_t::tran_speed
- uint32_t kdrv_sdc_csd_v2_t::ccc:12
- uint32_t kdrv_sdc_csd_v2_t::read_bl_len:4
- uint32_t kdrv_sdc_csd_v2_t::read_bl_partial:1
- uint32_t kdrv_sdc_csd_v2_t::write_blk_misalign:1
- uint32_t kdrv_sdc_csd_v2_t::read_blk_misalign:1
- uint32_t kdrv_sdc_csd_v2_t::dsr_imp:1
- uint32_t kdrv_sdc_csd_v2_t::reserved2:6
- uint32_t kdrv_sdc_csd_v2_t::c_size:22
- uint32_t kdrv_sdc_csd_v2_t::reserved3:1
- uint32_t kdrv_sdc_csd_v2_t::erase_blk_en:1
- uint32_t kdrv_sdc_csd_v2_t::sector_size:7
- uint32_t kdrv_sdc_csd_v2_t::wp_grp_size:7
- uint32_t kdrv_sdc_csd_v2_t::wp_grp_enable:1
- uint32_t kdrv_sdc_csd_v2_t::reserved4:2
- uint32_t kdrv_sdc_csd_v2_t::r2w_factor:3

Kneron Confidential

- uint32_t kdrv_sdc_csd_v2_t::write_bl_len:4
- uint32_t kdrv_sdc_csd_v2_t::write_bl_partial:1
- uint32_t kdrv_sdc_csd_v2_t::reserved5:5
- uint32_t kdrv_sdc_csd_v2_t::file_format_grp:1
- uint32_t kdrv_sdc_csd_v2_t::copy:1
- uint32_t kdrv_sdc_csd_v2_t::perm_write_protect:1
- uint32_t kdrv_sdc_csd_v2_t::tmp_write_protect:1
- uint32_t kdrv_sdc_csd_v2_t::file_format:2
- uint32_t kdrv_sdc_csd_v2_t::reserver6:2
- uint32_t kdrv_sdc_sd_scr_t::sd_spec:4
- uint32_t kdrv_sdc_sd_scr_t::scr_structure:4
- uint32_t kdrv_sdc_sd_scr_t::sd_bus_widths:4
- uint32_t kdrv_sdc_sd_scr_t::sd_security:3
- uint32_t kdrv_sdc_sd_scr_t::data_stat_after_erase:1
- uint32_t kdrv_sdc_sd_scr_t::reserved1:7
- uint32_t kdrv_sdc_sd_scr_t::sd_spec3:1
- uint32_t kdrv_sdc_sd_scr_t::cmd20_support:1
- uint32_t kdrv_sdc_sd_scr_t::cmd23_support:1
- uint32_t kdrv_sdc_sd_scr_t::reserverd2:6
- uint32_t kdrv_sdc_sd_scr_t::reserverd3
- uint32_t kdrv_sdc_sdcard_info_t::card_insert
- kdrv_sdc_flow_info_t kdrv_sdc_sdcard_info_t::flow_set
- uint16_t kdrv_sdc_sdcard_info_t::rca
- uint16_t kdrv_sdc_sdcard_info_t::dsr
- kdrv_sdc_sd_scr_t kdrv_sdc_sdcard_info_t::scr
- uint32_t kdrv_sdc_sdcard_info_t::ocr
- uint64_t kdrv_sdc_sdcard_info_t::csd_lo
- uint64_t kdrv_sdc_sdcard_info_t::csd_hi
- kdrv_sdc_csd_v1_t kdrv_sdc_sdcard_info_t::csd_ver1
- kdrv_sdc_csd_v2_t kdrv_sdc_sdcard_info_t::csd_ver2
- uint64_t kdrv_sdc_sdcard_info_t::cid_lo
- uint64_t kdrv_sdc_sdcard_info_t::cid_hi
- uint64_t kdrv_sdc_sdcard_info_t::resp_lo
- uint64_t kdrv_sdc_sdcard_info_t::resp_hi
- uint32_t kdrv_sdc_sdcard_info_t::num_of_blks
- uint8_t kdrv_sdc_sdcard_info_t::switch_sts [64]
- volatile uint16_t kdrv_sdc_sdcard_info_t::err_sts
- volatile uint16_t kdrv_sdc_sdcard_info_t::auto_err
- volatile uint8_t kdrv_sdc_sdcard_info_t::cmpl_mask
- kdrv_sd_status_t kdrv_sdc_sdcard_info_t::sd_sts
- uint16_t kdrv_sdc_sdcard_info_t::bs_mode
- uint8_t kdrv_sdc_sdcard_info_t::bus_width
- uint8_t kdrv_sdc_sdcard_info_t::already_init
- uint8_t kdrv_sdc_sdcard_info_t::blk_addr
- uint32_t kdrv_sdc_sdcard_info_t::max_dtr
- kdrv_sdc_bus_speed_e kdrv_sdc_sdcard_info_t::speed
- uint32_t kdrv_sdc_sdcard_info_t::fifo_depth
- kdrv_sdc_mmc_csd_t kdrv_sdc_sdcard_info_t::csd_mmc
- kdrv_sdc_mmc_ext_csd_t kdrv_sdc_sdcard_info_t::ext_csd_mmc
- uint32_t kdrv_sdc_sdcard_info_t::num_of_boot_blks
- uint8_t kdrv_sdc_sdcard_info_t::u8_sdma_lock
- uint8_t kdrv_sdc_sdcard_info_t::sdma_intr
- uint32_t kdrv_sdc_sdcard_info_t::card_type
- uint8_t kdrv_sdc_sdcard_info_t::num_io_func
- uint8_t kdrv_sdc_sdcard_info_t::mem_present

Kneron Confidential

- uint32_t kdrv_sdc_sdcard_info_t::drive
- uint32_t kdrv_sdc_sdcard_info_t::sys_freq
- uint32_t kdrv_sdc_sdcard_info_t::protected_drive
- uint32_t kdrv_sdc_sdcard_info_t::cprm_init
- uint32_t kdrv_sdc_sdcard_info_t::kmu_lo
- uint32_t kdrv_sdc_sdcard_info_t::kmu_hi
- uint32_t kdrv_sdc_sdcard_info_t::auto_cbc
- uint32_t kdrv_sdc_sd_host_t::max_clk
- uint32_t kdrv_sdc_sd_host_t::min_clk
- uint32_t kdrv_sdc_sd_host_t::clock
- uint8_t kdrv_sdc_sd_host_t::power
- uint32_t kdrv_sdc_sd_host_t::ocr_avail
- uint16_t kdrv_sdc_adma2desc_table_t::attr
- uint16_t kdrv_sdc_adma2desc_table_t::lgth
- uint32_t kdrv_sdc_adma2desc_table_t::addr
- volatile kdrv_sdc_reg_t * kdrv_sdc_res_t::sdc_reg
- volatile kdrv_sdc_sdcard_info_t * kdrv_sdc_res_t::card_info
- kdrv_sdc_sd_host_t * kdrv_sdc_res_t::host
- kdrv_sdc_infinite_test_e kdrv_sdc_res_t::infinite_mode
- uint16_t kdrv_sdc_res_t::fifo_depth
- uint32_t kdrv_sdc_res_t::timeout_ms
- uint32_t kdrv_sdc_res_t::data_present
- uint32_t kdrv_sdc_res_t::adma2_use_interrupt
- uint32_t kdrv_sdc_res_t::adma2_insert_nop
- uint32_t kdrv_sdc_res_t::response_type
- uint32_t kdrv_sdc_res_t::inhibit_datchk
- uint32_t kdrv_sdc_mmc_csd_t::csd_structure:2
- uint32_t kdrv_sdc_mmc_csd_t::spec_vers:4
- uint32_t kdrv_sdc_mmc_csd_t::reserved1:2
- uint8_t kdrv_sdc_mmc_csd_t::taac
- uint8_t kdrv_sdc_mmc_csd_t::nsac
- uint8_t kdrv_sdc_mmc_csd_t::tran_speed
- uint32_t kdrv_sdc_mmc_csd_t::ccc:12
- uint32_t kdrv_sdc_mmc_csd_t::read_bl_len:4
- uint32_t kdrv_sdc_mmc_csd_t::read_bl_partial:1
- uint32_t kdrv_sdc_mmc_csd_t::write_blk_misalign:1
- uint32_t kdrv_sdc_mmc_csd_t::read_blk_misalign:1
- uint32_t kdrv_sdc_mmc_csd_t::dsr_imp:1
- uint32_t kdrv_sdc_mmc_csd_t::reserved2:2
- uint32_t kdrv_sdc_mmc_csd_t::c_size:12
- uint32_t kdrv_sdc_mmc_csd_t::vdd_r_curr_min:3
- uint32_t kdrv_sdc_mmc_csd_t::vdd_r_curr_max:3
- uint32_t kdrv_sdc_mmc_csd_t::vdd_w_curr_min:3
- uint32_t kdrv_sdc_mmc_csd_t::vdd_w_curr_max:3
- uint32_t kdrv_sdc_mmc_csd_t::c_size_mult:3
- uint32_t kdrv_sdc_mmc_csd_t::erase_grp_size:5
- uint32_t kdrv_sdc_mmc_csd_t::erase_grp_mult:5
- uint32_t kdrv_sdc_mmc_csd_t::wp_grp_size:5
- uint32_t kdrv_sdc_mmc_csd_t::wp_grp_enable:1
- uint32_t kdrv_sdc_mmc_csd_t::default_ecc:2
- uint32_t kdrv_sdc_mmc_csd_t::r2w_factor:3
- uint32_t kdrv_sdc_mmc_csd_t::write_bl_len:4
- uint32_t kdrv_sdc_mmc_csd_t::write_bl_partial:1
- uint32_t kdrv_sdc_mmc_csd_t::reserved3:4
- uint32_t kdrv_sdc_mmc_csd_t::content_prot_app:1

Kneron Confidential

- uint32_t kdrv_sdc_mmc_csd_t::file_format_grp:1
- uint32_t kdrv_sdc_mmc_csd_t::COPY:1
- uint32_t kdrv_sdc_mmc_csd_t::perm_write_protect:1
- uint32_t kdrv_sdc_mmc_csd_t::tmp_write_protect:1
- uint32_t kdrv_sdc_mmc_csd_t::file_format:2
- uint32_t kdrv_sdc_mmc_csd_t::ecc:2
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved27 [134]
- uint8_t kdrv_sdc_mmc_ext_csd_t::sec_bad_blk_mgmnt
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved26
- uint8_t kdrv_sdc_mmc_ext_csd_t::enh_start_addr [4]
- uint8_t kdrv_sdc_mmc_ext_csd_t::enh_size_mult [3]
- uint8_t kdrv_sdc_mmc_ext_csd_t::gp_size_mult [12]
- uint8_t kdrv_sdc_mmc_ext_csd_t::partition_setting_completed
- uint8_t kdrv_sdc_mmc_ext_csd_t::partitioning_attribute
- uint8_t kdrv_sdc_mmc_ext_csd_t::max_enh_size_mult [3]
- uint8_t kdrv_sdc_mmc_ext_csd_t::partitioning_support
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved25
- uint8_t kdrv_sdc_mmc_ext_csd_t::rst_n_function
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved24 [5]
- uint8_t kdrv_sdc_mmc_ext_csd_t::rpmb_size_mult
- uint8_t kdrv_sdc_mmc_ext_csd_t::fw_config
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved23
- uint8_t kdrv_sdc_mmc_ext_csd_t::user_wp
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved22
- uint8_t kdrv_sdc_mmc_ext_csd_t::boot_wp
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved21
- uint8_t kdrv_sdc_mmc_ext_csd_t::erase_group_def
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved20
- uint8_t kdrv_sdc_mmc_ext_csd_t::boot_bus_width
- uint8_t kdrv_sdc_mmc_ext_csd_t::boot_config_prot
- uint8_t kdrv_sdc_mmc_ext_csd_t::partition_conf
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved19
- uint8_t kdrv_sdc_mmc_ext_csd_t::erased_mem_cont
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved18
- uint8_t kdrv_sdc_mmc_ext_csd_t::bus_width
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved17
- uint8_t kdrv_sdc_mmc_ext_csd_t::hs_timing
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved16
- uint8_t kdrv_sdc_mmc_ext_csd_t::power_class
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved15
- uint8_t kdrv_sdc_mmc_ext_csd_t::cmd_set_rev
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved14
- uint8_t kdrv_sdc_mmc_ext_csd_t::cmd_set
- uint8_t kdrv_sdc_mmc_ext_csd_t::ext_csd_rev
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved13
- uint8_t kdrv_sdc_mmc_ext_csd_t::csd_structure
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved12
- uint8_t kdrv_sdc_mmc_ext_csd_t::cardtype
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved11 [3]
- uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_52_195
- uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_26_195
- uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_52_360
- uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_26_360
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved10
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_r_4_26

Kneron Confidential

- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_w_4_26
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_r_8_26_4_52
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_w_8_26_4_52
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_r_8_52
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_w_8_52
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved9
- uint32_t kdrv_sdc_mmc_ext_csd_t::sec_count
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved8
- uint8_t kdrv_sdc_mmc_ext_csd_t::s_a_timeout
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved7
- uint8_t kdrv_sdc_mmc_ext_csd_t::s_c_vccq
- uint8_t kdrv_sdc_mmc_ext_csd_t::s_c_vcc
- uint8_t kdrv_sdc_mmc_ext_csd_t::hc_wp_grp_size
- uint8_t kdrv_sdc_mmc_ext_csd_t::ref_wr_sec_c
- uint8_t kdrv_sdc_mmc_ext_csd_t::erase_timeout_mult
- uint8_t kdrv_sdc_mmc_ext_csd_t::hc_erase_grp_size
- uint8_t kdrv_sdc_mmc_ext_csd_t::acc_size
- uint8_t kdrv_sdc_mmc_ext_csd_t::boot_size_mult
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved6
- uint8_t kdrv_sdc_mmc_ext_csd_t::boot_info
- uint8_t kdrv_sdc_mmc_ext_csd_t::sec_trim_mult
- uint8_t kdrv_sdc_mmc_ext_csd_t::sec_erase_mult
- uint8_t kdrv_sdc_mmc_ext_csd_t::sec_feature_support
- uint8_t kdrv_sdc_mmc_ext_csd_t::trim_mult
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved5
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_ddr_r_8_52
- uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_ddr_w_8_52_8_52
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved4 [2]
- uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_ddr_52_195
- uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_ddr_52_360
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved3
- uint8_t kdrv_sdc_mmc_ext_csd_t::ini_timeout_ap
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved2 [262]
- uint8_t kdrv_sdc_mmc_ext_csd_t::s_cmd_set
- uint8_t kdrv_sdc_mmc_ext_csd_t::reserved1 [7]

4.18.1 Detailed Description

Kneron sdc sd/emmc driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.18.2 Macro Definition Documentation

4.18.2.1 ADMA2_ENTRY_END

```
#define ADMA2_ENTRY_END BIT(1)
```


Kneron Confidential

4.18.2.2 ADMA2_ENTRY_INT

```
#define ADMA2_ENTRY_INT BIT(2)
```

4.18.2.3 ADMA2_ENTRY_VALID

```
#define ADMA2_ENTRY_VALID BIT(0)
```

4.18.2.4 ADMA2_LINK

```
#define ADMA2_LINK (3 << 4)
```

4.18.2.5 ADMA2_NOP

```
#define ADMA2_NOP (0 << 4)
```

4.18.2.6 ADMA2_NUM_OF_LINES

```
#define ADMA2_NUM_OF_LINES 64
```

4.18.2.7 ADMA2_SET

```
#define ADMA2_SET (1 << 4)
```

4.18.2.8 ADMA2_TRAN

```
#define ADMA2_TRAN (2 << 4)
```

4.18.2.9 CARD_TYPE_UNKNOWN

```
#define CARD_TYPE_UNKNOWN 0
```

Kneron Confidential

4.18.2.10 CMD_RETRY_CNT

```
#define CMD_RETRY_CNT 5
```

4.18.2.11 EXT_CSD_BOOT_SIZE_MULT

```
#define EXT_CSD_BOOT_SIZE_MULT 226
```

4.18.2.12 EXT_CSD_BUS_1BIT

```
#define EXT_CSD_BUS_1BIT 0x0
```

4.18.2.13 EXT_CSD_BUS_4BIT

```
#define EXT_CSD_BUS_4BIT 0x1
```

4.18.2.14 EXT_CSD_BUS_8BIT

```
#define EXT_CSD_BUS_8BIT 0x2
```

4.18.2.15 EXT_CSD_BUS_WIDTH

```
#define EXT_CSD_BUS_WIDTH 183 /* R/W */
```

4.18.2.16 EXT_CSD_CARD_TYPE

```
#define EXT_CSD_CARD_TYPE 196 /* RO */
```

4.18.2.17 EXT_CSD_CLR_BYTE

```
#define EXT_CSD_CLR_BYTE 0x2
```

Kneron Confidential

4.18.2.18 EXT_CSD_CMD_SET

```
#define EXT_CSD_CMD_SET 0x0
```

4.18.2.19 EXT_CSD_CMD_SET_CPSECURE

```
#define EXT_CSD_CMD_SET_CPSECURE (1<<2)
```

4.18.2.20 EXT_CSD_CMD_SET_NORMAL

```
#define EXT_CSD_CMD_SET_NORMAL (1<<0)
```

4.18.2.21 EXT_CSD_CMD_SET_SECURE

```
#define EXT_CSD_CMD_SET_SECURE (1<<1)
```

4.18.2.22 EXT_CSD_HS_TIMING

```
#define EXT_CSD_HS_TIMING 185 /* R/W */
```

4.18.2.23 EXT_CSD_LENGTH

```
#define EXT_CSD_LENGTH 512
```

4.18.2.24 EXT_CSD_PARTITION_CONF

```
#define EXT_CSD_PARTITION_CONF 179
```

4.18.2.25 EXT_CSD_PARTITION_SETTING_COMPLETED

```
#define EXT_CSD_PARTITION_SETTING_COMPLETED 156
```

Kneron Confidential

4.18.2.26 EXT_CSD_SEC_CNT

```
#define EXT_CSD_SEC_CNT 212 /* RO, 4 bytes */
```

4.18.2.27 EXT_CSD_SET_BIT

```
#define EXT_CSD_SET_BIT 0x1
```

4.18.2.28 EXT_CSD_WRITE_BYTE

```
#define EXT_CSD_WRITE_BYTE 0x3
```

4.18.2.29 KDRV_SDC_BASE

```
#define KDRV_SDC_BASE SDC_FTSDC021_PA_BASE
```

4.18.2.30 MEMORY_CARD_TYPE_MMC

```
#define MEMORY_CARD_TYPE_MMC 2
```

4.18.2.31 MEMORY_CARD_TYPE_SD

```
#define MEMORY_CARD_TYPE_SD 1
```

4.18.2.32 MEMORY_SDIO_COMBO

```
#define MEMORY_SDIO_COMBO 4
```

4.18.2.33 MMC_ALTERNATIVE_BOOT_MODE

```
#define MMC_ALTERNATIVE_BOOT_MODE 0x2
```

Kneron Confidential

4.18.2.34 MMC_BOOT_ACK

```
#define MMC_BOOT_ACK BIT(2)
```

4.18.2.35 MMC_BOOT_MODE

```
#define MMC_BOOT_MODE 0x1
```

4.18.2.36 MMC_BUS_TEST_MODE

```
#define MMC_BUS_TEST_MODE 0x3
```

4.18.2.37 MMC_CARD_BUSY

```
#define MMC_CARD_BUSY 0x80000000 /* Card Power up status bit */
```

4.18.2.38 MMC_CMD6_ACCESS_MODE

```
#define MMC_CMD6_ACCESS_MODE(
             x ) (uint32_t)( x << 24)
```

4.18.2.39 MMC_CMD6_CMD_SET

```
#define MMC_CMD6_CMD_SET(
             x ) (uint32_t)( x )
```

4.18.2.40 MMC_CMD6_INDEX

```
#define MMC_CMD6_INDEX(
             x ) (uint32_t)( x << 16)
```

Kneron Confidential

4.18.2.41 MMC_CMD6_VALUE

```
#define MMC_CMD6_VALUE(
             x ) (uint32_t)( x << 8)
```

4.18.2.42 MMC_STATUS_SWITCH_ERROR

```
#define MMC_STATUS_SWITCH_ERROR BIT(7)
```

4.18.2.43 NORMAL_MODE

```
#define NORMAL_MODE 0x0
```

4.18.2.44 SCR_LENGTH

```
#define SCR_LENGTH 8
```

4.18.2.45 SD_CMD52_FUNC

```
#define SD_CMD52_FUNC(
             Num ) (Num << 28)
```

4.18.2.46 SD_CMD52_no_RAW

```
#define SD_CMD52_no_RAW 0x00000000
```

4.18.2.47 SD_CMD52_RAW

```
#define SD_CMD52_RAW 0x08000000
```

Kneron Confidential

4.18.2.48 SD_CMD52_Reg_Addr

```
#define SD_CMD52_Reg_Addr(
              Addr ) (Addr << 9)
```

4.18.2.49 SD_CMD52_RW_in_R

```
#define SD_CMD52_RW_in_R 0x00000000
```

4.18.2.50 SD_CMD52_RW_in_W

```
#define SD_CMD52_RW_in_W 0x80000000
```

4.18.2.51 SD_CMD53_Block_Mode

```
#define SD_CMD53_Block_Mode 0x08000000
```

4.18.2.52 SD_CMD53_Byte_Mode

```
#define SD_CMD53_Byte_Mode 0x00000000
```

4.18.2.53 SD_CMD53_FUNC

```
#define SD_CMD53_FUNC(
              Num ) (Num << 28)
```

4.18.2.54 SD_CMD53_OP_fix

```
#define SD_CMD53_OP_fix 0x00000000
```

Kneron Confidential

4.18.2.55 SD_CMD53_OP_inc

```
#define SD_CMD53_OP_inc 0x04000000
```

4.18.2.56 SD_CMD53_Reg_Addr

```
#define SD_CMD53_Reg_Addr(
             Addr ) (Addr << 9)
```

4.18.2.57 SD_CMD53_RW_in_R

```
#define SD_CMD53_RW_in_R 0x00000000
```

4.18.2.58 SD_CMD53_RW_in_W

```
#define SD_CMD53_RW_in_W 0x80000000
```

4.18.2.59 SD_STATUS_ADDRESS_ERROR

```
#define SD_STATUS_ADDRESS_ERROR BIT(30)
```

4.18.2.60 SD_STATUS_AKE_SEQ_ERROR

```
#define SD_STATUS_AKE_SEQ_ERROR BIT(3)
```

4.18.2.61 SD_STATUS_APP_CMD

```
#define SD_STATUS_APP_CMD BIT(5)
```

Kneron Confidential

4.18.2.62 SD_STATUS_BLOCK_LEN_ERROR

```
#define SD_STATUS_BLOCK_LEN_ERROR BIT(29)
```

4.18.2.63 SD_STATUS_CARD_ECC_DISABLE

```
#define SD_STATUS_CARD_ECC_DISABLE BIT(14)
```

4.18.2.64 SD_STATUS_CARD_ECC_FAILED

```
#define SD_STATUS_CARD_ECC_FAILED BIT(21)
```

4.18.2.65 SD_STATUS_CARD_IS_LOCK

```
#define SD_STATUS_CARD_IS_LOCK BIT(25)
```

4.18.2.66 SD_STATUS_CC_ERROR

```
#define SD_STATUS_CC_ERROR BIT(20)
```

4.18.2.67 SD_STATUS_COM_CRC_ERROR

```
#define SD_STATUS_COM_CRC_ERROR BIT(23)
```

4.18.2.68 SD_STATUS_CSD_OVERWRITE

```
#define SD_STATUS_CSD_OVERWRITE BIT(16)
```

4.18.2.69 SD_STATUS_CURRENT_STATE

```
#define SD_STATUS_CURRENT_STATE (0xF << 9)
```

Kneron Confidential

4.18.2.70 SD_STATUS_ERASE_PARAM

```
#define SD_STATUS_ERASE_PARAM BIT(27)
```

4.18.2.71 SD_STATUS_ERASE_RESET

```
#define SD_STATUS_ERASE_RESET BIT(13)
```

4.18.2.72 SD_STATUS_ERASE_SEQ_ERROR

```
#define SD_STATUS_ERASE_SEQ_ERROR BIT(28)
```

4.18.2.73 SD_STATUS_ERROR

```
#define SD_STATUS_ERROR BIT(19)
```

4.18.2.74 SD_STATUS_ERROR_BITS

```
#define SD_STATUS_ERROR_BITS
```

Value:

```
(SD_STATUS_OUT_OF_RANGE | SD_STATUS_ADDRESS_ERROR | \
                                SD_STATUS_BLOCK_LEN_ERROR | SD_STATUS_ERASE_SEQ_ERROR | \
                                SD_STATUS_ERASE_PARAM | SD_STATUS_WP_VIOLATION | \
                                SD_STATUS_LOCK_UNLOCK_FAILED | SD_STATUS_CARD_ECC_FAILED | \
                                SD_STATUS_CC_ERROR | SD_STATUS_ERROR | \
                                SD_STATUS_UNDERRUN | SD_STATUS_OVERRUN | \
                                SD_STATUS_CSD_OVERWRITE | SD_STATUS_WP_ERASE_SKIP | \
                                SD_STATUS_AKE_SEQ_ERROR | MMC_STATUS_SWITCH_ERROR)
```

4.18.2.75 SD_STATUS_ILLEGAL_COMMAND

```
#define SD_STATUS_ILLEGAL_COMMAND BIT(22)
```

4.18.2.76 SD_STATUS_LENGTH

```
#define SD_STATUS_LENGTH 64
```

Kneron Confidential

4.18.2.77 SD_STATUS_LOCK_UNLOCK_FAILED

```
#define SD_STATUS_LOCK_UNLOCK_FAILED BIT(24)
```

4.18.2.78 SD_STATUS_OUT_OF_RANGE

```
#define SD_STATUS_OUT_OF_RANGE 0x80000000
```

Card status return from R1 response format. Or use CMD13 to get this status

4.18.2.79 SD_STATUS_OVERRUN

```
#define SD_STATUS_OVERRUN BIT(17)
```

4.18.2.80 SD_STATUS_READY_FOR_DATA

```
#define SD_STATUS_READY_FOR_DATA BIT(8)
```

4.18.2.81 SD_STATUS_UNDERRUN

```
#define SD_STATUS_UNDERRUN BIT(18)
```

4.18.2.82 SD_STATUS_WP_ERASE_SKIP

```
#define SD_STATUS_WP_ERASE_SKIP BIT(15)
```

4.18.2.83 SD_STATUS_WP_VIOLATION

```
#define SD_STATUS_WP_VIOLATION BIT(26)
```

4.18.2.84 SDCHI_RETUNING_MODE_MASK

```
#define SDCHI_RETUNING_MODE_MASK 0x3
```

Kneron Confidential

4.18.2.85 SDHCI_18V_SIGNAL

```
#define SDHCI_18V_SIGNAL BIT(3)
```

4.18.2.86 SDHCI_1BIT_BUS_WIDTH

```
#define SDHCI_1BIT_BUS_WIDTH 0x0
```

4.18.2.87 SDHCI_4BIT_BUS_WIDTH

```
#define SDHCI_4BIT_BUS_WIDTH 0x2
```

4.18.2.88 SDHCI_ASYNC_INT_EN

```
#define SDHCI_ASYNC_INT_EN BIT(14)
```

4.18.2.89 SDHCI_AUTOCMD12_ERR_CMD_NOT_ISSUE

```
#define SDHCI_AUTOCMD12_ERR_CMD_NOT_ISSUE BIT(7)
```

4.18.2.90 SDHCI_AUTOCMD12_ERR_CRC

```
#define SDHCI_AUTOCMD12_ERR_CRC BIT(2)
```

4.18.2.91 SDHCI_AUTOCMD12_ERR_END_BIT

```
#define SDHCI_AUTOCMD12_ERR_END_BIT BIT(3)
```

4.18.2.92 SDHCI_AUTOCMD12_ERR_INDEX

```
#define SDHCI_AUTOCMD12_ERR_INDEX BIT(4)
```

Kneron Confidential

4.18.2.93 SDHCI_AUTOCMD12_ERR_NOT_EXECUTED

```
#define SDHCI_AUTOCMD12_ERR_NOT_EXECUTED BIT(0)
```

4.18.2.94 SDHCI_AUTOCMD12_ERR_TIMEOUT

```
#define SDHCI_AUTOCMD12_ERR_TIMEOUT BIT(1)
```

4.18.2.95 SDHCI_CAP_FIFO_DEPTH_1024BYTE

```
#define SDHCI_CAP_FIFO_DEPTH_1024BYTE (4 << 29)
```

4.18.2.96 SDHCI_CAP_FIFO_DEPTH_16BYTE

```
#define SDHCI_CAP_FIFO_DEPTH_16BYTE (0 << 29)
```

4.18.2.97 SDHCI_CAP_FIFO_DEPTH_2048BYTE

```
#define SDHCI_CAP_FIFO_DEPTH_2048BYTE (5 << 29)
```

4.18.2.98 SDHCI_CAP_FIFO_DEPTH_32BYTE

```
#define SDHCI_CAP_FIFO_DEPTH_32BYTE (1 << 29)
```

4.18.2.99 SDHCI_CAP_FIFO_DEPTH_512BYTE

```
#define SDHCI_CAP_FIFO_DEPTH_512BYTE (3 << 29)
```

4.18.2.100 SDHCI_CAP_FIFO_DEPTH_64BYTE

```
#define SDHCI_CAP_FIFO_DEPTH_64BYTE (2 << 29)
```

Kneron Confidential

4.18.2.101 SDHCI_CAP_VOLTAGE_18V

```
#define SDHCI_CAP_VOLTAGE_18V BIT(26)
```

4.18.2.102 SDHCI_CAP_VOLTAGE_30V

```
#define SDHCI_CAP_VOLTAGE_30V BIT(25)
```

4.18.2.103 SDHCI_CAP_VOLTAGE_33V

```
#define SDHCI_CAP_VOLTAGE_33V BIT(24)
```

4.18.2.104 SDHCI_CLK_CTRL_CLK_GEN_SEL_PRO

```
#define SDHCI_CLK_CTRL_CLK_GEN_SEL_PRO BIT(5)
```

4.18.2.105 SDHCI_CLK_CTRL_INTERNALCLK_EN

```
#define SDHCI_CLK_CTRL_INTERNALCLK_EN BIT(0)
```

4.18.2.106 SDHCI_CLK_CTRL_INTERNALCLK_STABLE

```
#define SDHCI_CLK_CTRL_INTERNALCLK_STABLE BIT(1)
```

4.18.2.107 SDHCI_CLK_CTRL_LOW_CLK_SEL_SHIFT

```
#define SDHCI_CLK_CTRL_LOW_CLK_SEL_SHIFT 8
```

4.18.2.108 SDHCI_CLK_CTRL_SDCLK_EN

```
#define SDHCI_CLK_CTRL_SDCLK_EN BIT(2)
```

Kneron Confidential

4.18.2.109 SDHCI_CLK_CTRL_UP_CLK_SEL_SHIFT

```
#define SDHCI_CLK_CTRL_UP_CLK_SEL_SHIFT 6
```

4.18.2.110 SDHCI_CMD0_GO_IDLE_STATE

```
#define SDHCI_CMD0_GO_IDLE_STATE 0
```

4.18.2.111 SDHCI_CMD10_SEND_CID

```
#define SDHCI_CMD10_SEND_CID 10
```

4.18.2.112 SDHCI_CMD11_VOLTAGE_SWITCH

```
#define SDHCI_CMD11_VOLTAGE_SWITCH 11
```

4.18.2.113 SDHCI_CMD12_STOP_TRANS

```
#define SDHCI_CMD12_STOP_TRANS 12
```

4.18.2.114 SDHCI_CMD13_SD_STATUS

```
#define SDHCI_CMD13_SD_STATUS 13
```

4.18.2.115 SDHCI_CMD13_SEND_STATUS

```
#define SDHCI_CMD13_SEND_STATUS 13
```

4.18.2.116 SDHCI_CMD16_SET_BLOCKLEN

```
#define SDHCI_CMD16_SET_BLOCKLEN 16
```

Kneron Confidential

4.18.2.117 SDHCI_CMD17_READ_SINGLE_BLOCK

```
#define SDHCI_CMD17_READ_SINGLE_BLOCK 17
```

4.18.2.118 SDHCI_CMD18_READ_MULTI_BLOCK

```
#define SDHCI_CMD18_READ_MULTI_BLOCK 18
```

4.18.2.119 SDHCI_CMD19_SEND_TUNE_BLOCK

```
#define SDHCI_CMD19_SEND_TUNE_BLOCK 19
```

4.18.2.120 SDHCI_CMD1_MMC_SEND_OP_COND

```
#define SDHCI_CMD1_MMC_SEND_OP_COND 1
```

4.18.2.121 SDHCI_CMD1_MMC_SEND_OP_COND_BYTE_MODE

```
#define SDHCI_CMD1_MMC_SEND_OP_COND_BYTE_MODE 0x80FF8000
```

4.18.2.122 SDHCI_CMD1_MMC_SEND_OP_COND_SECTOR_MODE

```
#define SDHCI_CMD1_MMC_SEND_OP_COND_SECTOR_MODE 0xC0FF8000
```

4.18.2.123 SDHCI_CMD23_SET_WR_BLOCK_CNT

```
#define SDHCI_CMD23_SET_WR_BLOCK_CNT 23
```

4.18.2.124 SDHCI_CMD24_WRITE_BLOCK

```
#define SDHCI_CMD24_WRITE_BLOCK 24
```

Kneron Confidential

4.18.2.125 SDHCI_CMD25_WRITE_MULTI_BLOCK

```
#define SDHCI_CMD25_WRITE_MULTI_BLOCK 25
```

4.18.2.126 SDHCI_CMD2_SEND_ALL_CID

```
#define SDHCI_CMD2_SEND_ALL_CID 2
```

4.18.2.127 SDHCI_CMD32_ERASE_WR_BLK_START

```
#define SDHCI_CMD32_ERASE_WR_BLK_START 32
```

4.18.2.128 SDHCI_CMD33_ERASE_WR_BLK_END

```
#define SDHCI_CMD33_ERASE_WR_BLK_END 33
```

4.18.2.129 SDHCI_CMD35_ERASE_GROUP_START

```
#define SDHCI_CMD35_ERASE_GROUP_START 35
```

4.18.2.130 SDHCI_CMD36_ERASE_GROUP_END

```
#define SDHCI_CMD36_ERASE_GROUP_END 36
```

4.18.2.131 SDHCI_CMD38_ERASE

```
#define SDHCI_CMD38_ERASE 38
```

4.18.2.132 SDHCI_CMD3_SEND_RELATIVE_ADDR

```
#define SDHCI_CMD3_SEND_RELATIVE_ADDR 3
```

Kneron Confidential

4.18.2.133 SDHCI_CMD41_SD_SEND_OP_COND

```
#define SDHCI_CMD41_SD_SEND_OP_COND 41
```

4.18.2.134 SDHCI_CMD41_SD_SEND_OP_COND_ARGU

```
#define SDHCI_CMD41_SD_SEND_OP_COND_ARGU 0x00FF8000
```

4.18.2.135 SDHCI_CMD41_SD_SEND_OP_COND_HCS_ARGU

```
#define SDHCI_CMD41_SD_SEND_OP_COND_HCS_ARGU 0xC0FF8000
```

4.18.2.136 SDHCI_CMD43_GET_MKB

```
#define SDHCI_CMD43_GET_MKB 43
```

4.18.2.137 SDHCI_CMD44_GET_MID

```
#define SDHCI_CMD44_GET_MID 44
```

4.18.2.138 SDHCI_CMD45_CER_RN1

```
#define SDHCI_CMD45_CER_RN1 45
```

4.18.2.139 SDHCI_CMD46_CER_RN2

```
#define SDHCI_CMD46_CER_RN2 46
```

4.18.2.140 SDHCI_CMD47_CER_RES2

```
#define SDHCI_CMD47_CER_RES2 47
```

Kneron Confidential

4.18.2.141 SDHCI_CMD48_CER_RES1

```
#define SDHCI_CMD48_CER_RES1 48
```

4.18.2.142 SDHCI_CMD51_SEND_SCR

```
#define SDHCI_CMD51_SEND_SCR 51
```

4.18.2.143 SDHCI_CMD52_IO_RW_DIRECT

```
#define SDHCI_CMD52_IO_RW_DIRECT 52
```

4.18.2.144 SDHCI_CMD53_IO_RW_EXTENDED

```
#define SDHCI_CMD53_IO_RW_EXTENDED 53
```

4.18.2.145 SDHCI_CMD55_APP

```
#define SDHCI_CMD55_APP 55
```

4.18.2.146 SDHCI_CMD56_GEN

```
#define SDHCI_CMD56_GEN 56
```

4.18.2.147 SDHCI_CMD5_IO_SEND_OP_COND

```
#define SDHCI_CMD5_IO_SEND_OP_COND 5
```

4.18.2.148 SDHCI_CMD6_SET_BUS_WIDTH

```
#define SDHCI_CMD6_SET_BUS_WIDTH 6
```

Kneron Confidential

4.18.2.149 SDHCI_CMD6_SWITCH_FUNC

```
#define SDHCI_CMD6_SWITCH_FUNC 6
```

4.18.2.150 SDHCI_CMD7_SELECT_CARD

```
#define SDHCI_CMD7_SELECT_CARD 7
```

4.18.2.151 SDHCI_CMD8_SEND_EXT_CSD

```
#define SDHCI_CMD8_SEND_EXT_CSD 8
```

4.18.2.152 SDHCI_CMD8_SEND_IF_COND

```
#define SDHCI_CMD8_SEND_IF_COND 8
```

4.18.2.153 SDHCI_CMD8_SEND_IF_COND_ARGU

```
#define SDHCI_CMD8_SEND_IF_COND_ARGU 0x1AA
```

4.18.2.154 SDHCI_CMD9_SEND_CSD

```
#define SDHCI_CMD9_SEND_CSD 9
```

4.18.2.155 SDHCI_CMD_DATA_PRESEL_SHIFT

```
#define SDHCI_CMD_DATA_PRESEL_SHIFT 0x05
```

4.18.2.156 SDHCI_CMD_DATA_PRESENT

```
#define SDHCI_CMD_DATA_PRESENT 0x01
```

Kneron Confidential

4.18.2.157 SDHCI_CMD_IDX_SHIFT

```
#define SDHCI_CMD_IDX_SHIFT 0x08
```

4.18.2.158 SDHCI_CMD_NO_RESPONSE

```
#define SDHCI_CMD_NO_RESPONSE 0x00
```

4.18.2.159 SDHCI_CMD_RTYPE_R1BR5B

```
#define SDHCI_CMD_RTYPE_R1BR5B 0x1B
```

4.18.2.160 SDHCI_CMD_RTYPE_R1R5R6R7

```
#define SDHCI_CMD_RTYPE_R1R5R6R7 0x1A
```

4.18.2.161 SDHCI_CMD_RTYPE_R2

```
#define SDHCI_CMD_RTYPE_R2 0x09
```

4.18.2.162 SDHCI_CMD_RTYPE_R3R4

```
#define SDHCI_CMD_RTYPE_R3R4 0x02
```

4.18.2.163 SDHCI_CMD_TYPE_ABORT

```
#define SDHCI_CMD_TYPE_ABORT 0x03
```

4.18.2.164 SDHCI_CMD_TYPE_NORMAL

```
#define SDHCI_CMD_TYPE_NORMAL 0x00
```

Kneron Confidential

4.18.2.165 SDHCI_CMD_TYPE_RESUME

```
#define SDHCI_CMD_TYPE_RESUME 0x02
```

4.18.2.166 SDHCI_CMD_TYPE_SHIFT

```
#define SDHCI_CMD_TYPE_SHIFT 0x06
```

4.18.2.167 SDHCI_CMD_TYPE_SUSPEND

```
#define SDHCI_CMD_TYPE_SUSPEND 0x01
```

4.18.2.168 SDHCI_CONTINUE_REQ

```
#define SDHCI_CONTINUE_REQ BIT(1)
```

4.18.2.169 SDHCI_DDR50

```
#define SDHCI_DDR50 4
```

4.18.2.170 SDHCI_DRV_TYPE_MASK

```
#define SDHCI_DRV_TYPE_MASK BIT(4)
```

4.18.2.171 SDHCI_DRV_TYPE_SHIFT

```
#define SDHCI_DRV_TYPE_SHIFT 4
```

4.18.2.172 SDHCI_DRV_TYPEA

```
#define SDHCI_DRV_TYPEA 1
```

Kneron Confidential

4.18.2.173 SDHCI_DRV_TYPEB

```
#define SDHCI_DRV_TYPEB 0
```

4.18.2.174 SDHCI_DRV_TYPEC

```
#define SDHCI_DRV_TYPEC 2
```

4.18.2.175 SDHCI_DRV_TYPED

```
#define SDHCI_DRV_TYPED 3
```

4.18.2.176 SDHCI_ERR_EN_ALL

```
#define SDHCI_ERR_EN_ALL (0xF7FF)
```

4.18.2.177 SDHCI_ERR_SIG_EN_ALL

```
#define SDHCI_ERR_SIG_EN_ALL (0xF3FF)
```

4.18.2.178 SDHCI_EXECUTE_TUNING

```
#define SDHCI_EXECUTE_TUNING BIT(6) /* Write 1 Auto clear */
```

4.18.2.179 SDHCI_HC_BUS_WIDTH_4BIT

```
#define SDHCI_HC_BUS_WIDTH_4BIT BIT(1)
```

4.18.2.180 SDHCI_HC_BUS_WIDTH_8BIT

```
#define SDHCI_HC_BUS_WIDTH_8BIT BIT(5)
```

Kneron Confidential

4.18.2.181 SDHCI_HC_CARD_DETECT_SIGNAL

```
#define SDHCI_HC_CARD_DETECT_SIGNAL BIT(7)
```

4.18.2.182 SDHCI_HC_CARD_DETECT_TEST

```
#define SDHCI_HC_CARD_DETECT_TEST BIT(6)
```

4.18.2.183 SDHCI_HC_HI_SPEED

```
#define SDHCI_HC_HI_SPEED BIT(2)
```

4.18.2.184 SDHCI_HC_LED_ON

```
#define SDHCI_HC_LED_ON BIT(0)
```

4.18.2.185 SDHCI_HC_USE_ADMA2

```
#define SDHCI_HC_USE_ADMA2 BIT(3)
```

4.18.2.186 SDHCI_INT_AT_BLOCK_GAP

```
#define SDHCI_INT_AT_BLOCK_GAP BIT(3)
```

4.18.2.187 SDHCI_INTR_EN_ALL

```
#define SDHCI_INTR_EN_ALL (0x10FF)
```

4.18.2.188 SDHCI_INTR_ERR_ADMA

```
#define SDHCI_INTR_ERR_ADMA BIT(9)
```

Kneron Confidential

4.18.2.189 SDHCI_INTR_ERR_AUTOCMD

```
#define SDHCI_INTR_ERR_AUTOCMD BIT(8)
```

4.18.2.190 SDHCI_INTR_ERR_CMD_CRC

```
#define SDHCI_INTR_ERR_CMD_CRC BIT(1)
```

4.18.2.191 SDHCI_INTR_ERR_CMD_ENDBIT

```
#define SDHCI_INTR_ERR_CMD_ENDBIT BIT(2)
```

4.18.2.192 SDHCI_INTR_ERR_CMD_INDEX

```
#define SDHCI_INTR_ERR_CMD_INDEX BIT(3)
```

4.18.2.193 SDHCI_INTR_ERR_CMD_LINE

```
#define SDHCI_INTR_ERR_CMD_LINE (SDHCI_INTR_ERR_CMD_INDEX | SDHCI_INTR_ERR_CMD_ENDBIT | SDHCI_INTR_ERR_CMD_CRC
| SDHCI_INTR_ERR_CMD_TIMEOUT)
```

4.18.2.194 SDHCI_INTR_ERR_CMD_TIMEOUT

```
#define SDHCI_INTR_ERR_CMD_TIMEOUT BIT(0)
```

4.18.2.195 SDHCI_INTR_ERR_CURR_LIMIT

```
#define SDHCI_INTR_ERR_CURR_LIMIT BIT(7)
```

Kneron Confidential

4.18.2.196 SDHCI_INTR_ERR_DAT_LINE

```
#define SDHCI_INTR_ERR_DAT_LINE (SDHCI_INTR_ERR_DATA_ENDBIT | SDHCI_INTR_ERR_DATA_CRC | SDHCI_INTR_ERR_DATA_TIM
```

4.18.2.197 SDHCI_INTR_ERR_DATA_CRC

```
#define SDHCI_INTR_ERR_DATA_CRC BIT(5)
```

4.18.2.198 SDHCI_INTR_ERR_DATA_ENDBIT

```
#define SDHCI_INTR_ERR_DATA_ENDBIT BIT(6)
```

4.18.2.199 SDHCI_INTR_ERR_DATA_TIMEOUT

```
#define SDHCI_INTR_ERR_DATA_TIMEOUT BIT(4)
```

4.18.2.200 SDHCI_INTR_ERR_TUNING

```
#define SDHCI_INTR_ERR_TUNING BIT(10)
```

4.18.2.201 SDHCI_INTR_SIG_EN

```
#define SDHCI_INTR_SIG_EN (SDHCI_INTR_STS_CARD_REMOVE | SDHCI_INTR_STS_CARD_INSERT | SDHCI_INTR_STS_CMD_COMPLET
| SDHCI_INTR_STS_TXR_COMPLETE)
```

4.18.2.202 SDHCI_INTR_SIGN_EN_ADMA

```
#define SDHCI_INTR_SIGN_EN_ADMA (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_DMA)
```

Kneron Confidential

4.18.2.203 SDHCI_INTR_SIGN_EN_PIO

```
#define SDHCI_INTR_SIGN_EN_PIO (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_BLKGAP)
```

4.18.2.204 SDHCI_INTR_SIGN_EN_SDMA

```
#define SDHCI_INTR_SIGN_EN_SDMA (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_DMA | SDHCI_INTR_STS_BLKGAP)
```

4.18.2.205 SDHCI_INTR_STS_BLKGAP

```
#define SDHCI_INTR_STS_BLKGAP BIT(2)
```

4.18.2.206 SDHCI_INTR_STS_BUFF_READ_READY

```
#define SDHCI_INTR_STS_BUFF_READ_READY BIT(5)
```

4.18.2.207 SDHCI_INTR_STS_BUFF_WRITE_READY

```
#define SDHCI_INTR_STS_BUFF_WRITE_READY BIT(4)
```

4.18.2.208 SDHCI_INTR_STS_CARD_INSERT

```
#define SDHCI_INTR_STS_CARD_INSERT BIT(6)
```

4.18.2.209 SDHCI_INTR_STS_CARD_INTR

```
#define SDHCI_INTR_STS_CARD_INTR BIT(8)
```

4.18.2.210 SDHCI_INTR_STS_CARD_REMOVE

```
#define SDHCI_INTR_STS_CARD_REMOVE BIT(7)
```

Kneron Confidential

4.18.2.211 SDHCI_INTR_STS_CMD_COMPLETE

```
#define SDHCI_INTR_STS_CMD_COMPLETE BIT(0) /* CMD completed, CMD12/CMD23 will not generate
this command */
```

4.18.2.212 SDHCI_INTR_STS_DMA

```
#define SDHCI_INTR_STS_DMA BIT(3)
```

4.18.2.213 SDHCI_INTR_STS_ERR

```
#define SDHCI_INTR_STS_ERR BIT(15)
```

4.18.2.214 SDHCI_INTR_STS_TXR_COMPLETE

```
#define SDHCI_INTR_STS_TXR_COMPLETE BIT(1)
```

4.18.2.215 SDHCI_MMC_SWITCH

```
#define SDHCI_MMC_SWITCH 6
```

4.18.2.216 SDHCI_MMC_VENDOR_CMD

```
#define SDHCI_MMC_VENDOR_CMD 62
```

4.18.2.217 SDHCI_POWER_180

```
#define SDHCI_POWER_180 (5 << 1)
```

Kneron Confidential

4.18.2.218 SDHCI_POWER_300

```
#define SDHCI_POWER_300 (6 << 1)
```

4.18.2.219 SDHCI_POWER_330

```
#define SDHCI_POWER_330 (7 << 1)
```

4.18.2.220 SDHCI_POWER_ON

```
#define SDHCI_POWER_ON BIT(0)
```

4.18.2.221 SDHCI_PRESET_VAL_EN

```
#define SDHCI_PRESET_VAL_EN BIT(15)
```

4.18.2.222 SDHCI_READ_WAIT_CTL

```
#define SDHCI_READ_WAIT_CTL BIT(2)
```

4.18.2.223 SDHCI_REG_CLK_CTRL

```
#define SDHCI_REG_CLK_CTRL 0x2C
```

4.18.2.224 SDHCI_REG_DATA_PORT

```
#define SDHCI_REG_DATA_PORT 0x20
```

4.18.2.225 SDHCI_REG_HC

```
#define SDHCI_REG_HC 0x28
```

Kneron Confidential

4.18.2.226 SDHCI_REG_HOST_CTRL2

```
#define SDHCI_REG_HOST_CTRL2 0x3E
```

4.18.2.227 SDHCI_REG_INTR_STATE

```
#define SDHCI_REG_INTR_STATE 0x30
```

4.18.2.228 SDHCI_REG_PRE_STATE

```
#define SDHCI_REG_PRE_STATE 0x24
```

4.18.2.229 SDHCI_RETUNING_MODE_SHIFT

```
#define SDHCI_RETUNING_MODE_SHIFT 14
```

4.18.2.230 SDHCI_RETUNING_TIME_MAS

```
#define SDHCI_RETUNING_TIME_MAS 0xF
```

4.18.2.231 SDHCI_RETUNING_TIME_SHIFT

```
#define SDHCI_RETUNING_TIME_SHIFT 8
```

4.18.2.232 SDHCI_SCR_SUPPORT_1BIT_BUS

```
#define SDHCI_SCR_SUPPORT_1BIT_BUS 0x1
```

4.18.2.233 SDHCI_SCR_SUPPORT_4BIT_BUS

```
#define SDHCI_SCR_SUPPORT_4BIT_BUS 0x4
```

Kneron Confidential

4.18.2.234 SDHCI_SDR104

```
#define SDHCI_SDR104 3
```

4.18.2.235 SDHCI_SDR12

```
#define SDHCI_SDR12 0
```

4.18.2.236 SDHCI_SDR25

```
#define SDHCI_SDR25 1
```

4.18.2.237 SDHCI_SDR50

```
#define SDHCI_SDR50 2
```

4.18.2.238 SDHCI_SDR50_TUNING

```
#define SDHCI_SDR50_TUNING BIT(13)
```

4.18.2.239 SDHCI_SMPL_CLCK_SELECT

```
#define SDHCI_SMPL_CLCK_SELECT BIT(7)
```

4.18.2.240 SDHCI_SOFTRST_ALL

```
#define SDHCI_SOFTRST_ALL BIT(0)
```

4.18.2.241 SDHCI_SOFTRST_CMD

```
#define SDHCI_SOFTRST_CMD BIT(1)
```

Kneron Confidential

4.18.2.242 SDHCI_SOFTRST_DAT

```
#define SDHCI_SOFTRST_DAT BIT(2)
```

4.18.2.243 SDHCI_STOP_AT_BLOCK_GAP_REQ

```
#define SDHCI_STOP_AT_BLOCK_GAP_REQ BIT(0)
```

4.18.2.244 SDHCI_STS_BUFF_READ

```
#define SDHCI_STS_BUFF_READ BIT(11)
```

4.18.2.245 SDHCI_STS_BUFF_WRITE

```
#define SDHCI_STS_BUFF_WRITE BIT(10)
```

4.18.2.246 SDHCI_STS_CARD_INSERT

```
#define SDHCI_STS_CARD_INSERT BIT(16)
```

4.18.2.247 SDHCI_STS_CARD_STABLE

```
#define SDHCI_STS_CARD_STABLE BIT(17)
```

4.18.2.248 SDHCI_STS_CARD_WP

```
#define SDHCI_STS_CARD_WP BIT(19)
```

4.18.2.249 SDHCI_STS_CMD_DAT_INHIBIT

```
#define SDHCI_STS_CMD_DAT_INHIBIT BIT(1)
```

Kneron Confidential

4.18.2.250 SDHCI_STS_CMD_INHIBIT

```
#define SDHCI_STS_CMD_INHIBIT BIT(0)
```

4.18.2.251 SDHCI_STS_CMD_LINE_LEVEL

```
#define SDHCI_STS_CMD_LINE_LEVEL BIT(24)
```

4.18.2.252 SDHCI_STS_DAT_LINE_ACT

```
#define SDHCI_STS_DAT_LINE_ACT BIT(2)
```

4.18.2.253 SDHCI_STS_DAT_LINE_LEVEL

```
#define SDHCI_STS_DAT_LINE_LEVEL (0xF << 20)
```

4.18.2.254 SDHCI_STS_READ_TRAN_ACT

```
#define SDHCI_STS_READ_TRAN_ACT BIT(9)
```

4.18.2.255 SDHCI_STS_WRITE_TRAN_ACT

```
#define SDHCI_STS_WRITE_TRAN_ACT BIT(8)
```

4.18.2.256 SDHCI_SUPPORT_DDR50

```
#define SDHCI_SUPPORT_DDR50 BIT(2)
```

4.18.2.257 SDHCI_SUPPORT_DRV_TYPEA

```
#define SDHCI_SUPPORT_DRV_TYPEA BIT(4)
```

Kneron Confidential

4.18.2.258 SDHCI_SUPPORT_DRV_TYPEC

```
#define SDHCI_SUPPORT_DRV_TYPEC BIT(5)
```

4.18.2.259 SDHCI_SUPPORT_DRV_TYPED

```
#define SDHCI_SUPPORT_DRV_TYPED BIT(6)
```

4.18.2.260 SDHCI_SUPPORT_SDR104

```
#define SDHCI_SUPPORT_SDR104 BIT(1)
```

4.18.2.261 SDHCI_SUPPORT_SDR50

```
#define SDHCI_SUPPORT_SDR50 BIT(0)
```

4.18.2.262 SDHCI_TIMEOUT

```
#define SDHCI_TIMEOUT 0xFFF
```

4.18.2.263 SDHCI_TXMODE_AUTOCMD12_EN

```
#define SDHCI_TXMODE_AUTOCMD12_EN BIT(2)
```

4.18.2.264 SDHCI_TXMODE_AUTOCMD23_EN

```
#define SDHCI_TXMODE_AUTOCMD23_EN (2 << 2)
```

4.18.2.265 SDHCI_TXMODE_BLKCNT_EN

```
#define SDHCI_TXMODE_BLKCNT_EN BIT(1)
```

Kneron Confidential

4.18.2.266 SDHCI_TXMODE_DMA_EN

```
#define SDHCI_TXMODE_DMA_EN BIT(0)
```

4.18.2.267 SDHCI_TXMODE_MULTI_SEL

```
#define SDHCI_TXMODE_MULTI_SEL BIT(5)
```

4.18.2.268 SDHCI_TXMODE_READ_DIRECTION

```
#define SDHCI_TXMODE_READ_DIRECTION BIT(4)
```

4.18.2.269 SDHCI_TXMODE_WRITE_DIRECTION

```
#define SDHCI_TXMODE_WRITE_DIRECTION (0 << 4)
```

4.18.2.270 SDHCI_UHS_MODE_MASK

```
#define SDHCI_UHS_MODE_MASK (7 << 0)
```

4.18.2.271 SDIO_TYPE_CARD

```
#define SDIO_TYPE_CARD 3
```

4.18.2.272 WAIT_BLOCK_GAP

```
#define WAIT_BLOCK_GAP BIT(3)
```

4.18.2.273 WAIT_CMD_COMPLETE

```
#define WAIT_CMD_COMPLETE BIT(0)
```

Kneron Confidential

4.18.2.274 WAIT_DMA_INTR

```
#define WAIT_DMA_INTR BIT(2)
```

4.18.2.275 WAIT_TRANS_COMPLETE

```
#define WAIT_TRANS_COMPLETE BIT(1)
```

4.18.3 Enumeration Type Documentation

4.18.3.1 kdrv_sdc_abort_type_e

```
enum kdrv_sdc_abort_type_e
```

Enumerator

ABORT_ASYNCHRONOUS	
ABORT_SYNCHRONOUS	
ABORT_UNDEFINED	

4.18.3.2 kdrv_sdc_bus_speed_e

```
enum kdrv_sdc_bus_speed_e
```

Enumerator

SPEED_DEFAULT	
SPEED_SDR25	
SPEED_SDR50	
SPEED_SDR104	
SPEED_DDR50	
SPEED_RSRV	

4.18.3.3 kdrv_sdc_card_state_e

```
enum kdrv_sdc_card_state_e
```

Kneron Confidential

Enumerator

CUR_STATE_IDLE	
CUR_STATE_READY	
CUR_STATE_IDENT	
CUR_STATE_STBY	
CUR_STATE_TRAN	
CUR_STATE_DATA	
CUR_STATE_RCV	
CUR_STATE_PRG	
CUR_STATE_DIS	
CUR_STATE_RSV	

4.18.3.4 kdrv_sdc_cprm_test_e

`enum kdrv_sdc_cprm_test_e`

Enumerator

CPRM_PROTECT_RW	
CPRM_FILESYS	
CPRM_UNKNOWN	

4.18.3.5 kdrv_sdc_infinite_test_e

`enum kdrv_sdc_infinite_test_e`

Enumerator

INFINITE_NO	
INFINITE_MODE_1	
INFINITE_MODE_2	

4.18.3.6 kdrv_sdc_transfer_act_e

`enum kdrv_sdc_transfer_act_e`

Kneron Confidential

Enumerator

WRITE	
READ	

4.18.3.7 kdrv_sdc_transfer_type_e

```
enum kdrv_sdc_transfer_type_e
```

Enumerator

ADMA	
SDMA	
PIO	
EDMA	
TRANS_UNKNOWN	

4.18.4 Function Documentation

4.18.4.1 kdrv_sdc_dev_scan()

```
kdrv_status_t kdrv_sdc_dev_scan (
            void  )
```

kdrv_sdc_dev_scan() scan sd/mmc memory card

Returns

kdrv_status_t

4.18.4.2 kdrv_sdc_get_dev()

```
kdrv_sdc_res_t* kdrv_sdc_get_dev (
            void  )
```

kdrv_sdc_get_dev() get device structure

Returns

kdrv_status_t

kdrv_sdc_get_dev() get device structure

Returns

kdrv_status_t

Kneron Confidential

4.18.4.3 kdrv_sdc_initialize()

```
kdrv_status_t kdrv_sdc_initialize (
            void  )
```

kdrv_sdc_initialize, initail sd/emmc card interface

1. reset sdc status
2. allocate resource and initail driving
3. turn on sdc clock

Returns

kdrv_status_t

4.18.4.4 kdrv_sdc_read()

```
kdrv_status_t kdrv_sdc_read (
            uint8_t * buf,
            uint32_t sd_offset,
            uint32_t size )
```

kdrv_sdc_read read data from sd/mmc card

Parameters

in	*buf*	buffer to write.
in	*sd_offset*	sd/mmc offset address
in	*size*	read size(Multiple of 512, 1sector=512Bytes)

Returns

kdrv_status_t

4.18.4.5 kdrv_sdc_uninitialize()

```
kdrv_status_t kdrv_sdc_uninitialize (
            void  )
```

kdrv_sdc_uninitialize, uninitail sd/emmc card interface and resource

1. reset sdc status and initail driven
2. turn on sdc clock

Kneron Confidential

Returns

kdrv_status_t

kdrv_sdc_uninitialize, uninitail sd/emmc card interface and resource

1. reset sdc status and initail driven
2. turn on sdc clock

4.18.4.6 kdrv_sdc_write()

```
kdrv_status_t kdrv_sdc_write (
            uint8_t * buf,
            uint32_t sd_offset,
            uint32_t size )
```

kdrv_sdc_write write data from sd/mmc card

Parameters

in	*buf*	buffer to write.
in	*sd_offset*	sd/mmc offset address
in	*size*	write size(Multiple of 512, 1sector=512Bytes)

Returns

kdrv_status_t

4.18.5 Variable Documentation

4.18.5.1 acc_size

```
uint8_t kdrv_sdc_mmc_ext_csd_t::acc_size
```

4.18.5.2 addr

```
uint32_t kdrv_sdc_adma2desc_table_t::addr
```

Kneron Confidential

4.18.5.3 adma2_insert_nop

uint32_t kdrv_sdc_res_t::adma2_insert_nop

4.18.5.4 adma2_use_interrupt

uint32_t kdrv_sdc_res_t::adma2_use_interrupt

4.18.5.5 adma2rand

uint16_t kdrv_sdc_flow_info_t::adma2rand

4.18.5.6 adma_addr

uint64_t kdrv_sdc_reg_t::adma_addr

4.18.5.7 adma_err_sts

uint32_t kdrv_sdc_reg_t::adma_err_sts

4.18.5.8 ahb_err_en

uint32_t kdrv_sdc_reg_t::ahb_err_en

4.18.5.9 ahb_err_sig_en

uint32_t kdrv_sdc_reg_t::ahb_err_sig_en

4.18.5.10 ahb_err_sts

uint32_t kdrv_sdc_reg_t::ahb_err_sts

Kneron Confidential

4.18.5.11 already_init

```
uint8_t kdrv_sdc_sdcard_info_t::already_init
```

4.18.5.12 attr

```
uint16_t kdrv_sdc_adma2desc_table_t::attr
```

4.18.5.13 au_size

```
uint32_t kdrv_sd_status_t::au_size
```

4.18.5.14 auto_cbc

```
uint32_t kdrv_sdc_sdcard_info_t::auto_cbc
```

4.18.5.15 auto_cmd

```
uint8_t kdrv_sdc_flow_info_t::auto_cmd
```

4.18.5.16 auto_cmd_err

```
uint16_t kdrv_sdc_reg_t::auto_cmd_err
```

4.18.5.17 auto_err

```
volatile uint16_t kdrv_sdc_sdcard_info_t::auto_err
```

4.18.5.18 blk_addr

```
uint8_t kdrv_sdc_sdcard_info_t::blk_addr
```

Kneron Confidential

4.18.5.19 blk_cnt

```
uint16_t kdrv_sdc_reg_t::blk_cnt
```

4.18.5.20 blk_gap_ctl

```
uint8_t kdrv_sdc_reg_t::blk_gap_ctl
```

4.18.5.21 blk_size

```
uint16_t kdrv_sdc_reg_t::blk_size
```

4.18.5.22 boot_bus_width

```
uint8_t kdrv_sdc_mmc_ext_csd_t::boot_bus_width
```

4.18.5.23 boot_config_prot

```
uint8_t kdrv_sdc_mmc_ext_csd_t::boot_config_prot
```

4.18.5.24 boot_info

```
uint8_t kdrv_sdc_mmc_ext_csd_t::boot_info
```

4.18.5.25 boot_size_mult

```
uint8_t kdrv_sdc_mmc_ext_csd_t::boot_size_mult
```

4.18.5.26 boot_wp

```
uint8_t kdrv_sdc_mmc_ext_csd_t::boot_wp
```

Kneron Confidential

4.18.5.27 bs_mode

```
uint16_t kdrv_sdc_sdcard_info_t::bs_mode
```

4.18.5.28 buf_data

```
uint32_t kdrv_sdc_reg_t::buf_data
```

4.18.5.29 bus_width [1/2]

```
uint8_t kdrv_sdc_mmc_ext_csd_t::bus_width
```

4.18.5.30 bus_width [2/2]

```
uint8_t kdrv_sdc_sdcard_info_t::bus_width
```

4.18.5.31 c_size [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::c_size
```

4.18.5.32 c_size [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::c_size
```

4.18.5.33 c_size [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::c_size
```

4.18.5.34 c_size_mult [1/2]

```
uint32_t kdrv_sdc_mmc_csd_t::c_size_mult
```

Kneron Confidential

4.18.5.35 c_size_mult [2/2]

```
uint32_t kdrv_sdc_csd_v1_t::c_size_mult
```

4.18.5.36 cap_reg

```
uint32_t kdrv_sdc_reg_t::cap_reg
```

4.18.5.37 cap_reg2

```
uint32_t kdrv_sdc_reg_t::cap_reg2
```

4.18.5.38 card_info

```
volatile kdrv_sdc_sdcard_info_t* kdrv_sdc_res_t::card_info
```

4.18.5.39 card_insert

```
uint32_t kdrv_sdc_sdcard_info_t::card_insert
```

4.18.5.40 card_type

```
uint32_t kdrv_sdc_sdcard_info_t::card_type
```

4.18.5.41 cardtype

```
uint8_t kdrv_sdc_mmc_ext_csd_t::cardtype
```

4.18.5.42 ccc [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::ccc
```

Kneron Confidential

4.18.5.43 ccc [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::ccc
```

4.18.5.44 ccc [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::ccc
```

4.18.5.45 cid_hi

```
uint64_t kdrv_sdc_sdcard_info_t::cid_hi
```

4.18.5.46 cid_lo

```
uint64_t kdrv_sdc_sdcard_info_t::cid_lo
```

4.18.5.47 ciph_m_ctl

```
uint32_t kdrv_sdc_reg_t::ciph_m_ctl
```

4.18.5.48 ciph_m_sig_en

```
uint16_t kdrv_sdc_reg_t::ciph_m_sig_en
```

4.18.5.49 ciph_m_sts

```
uint32_t kdrv_sdc_reg_t::ciph_m_sts
```

4.18.5.50 ciph_m_sts_en

```
uint16_t kdrv_sdc_reg_t::ciph_m_sts_en
```

Kneron Confidential

4.18.5.51 clk_ctl

```
uint16_t kdrv_sdc_reg_t::clk_ctl
```

4.18.5.52 clock

```
uint32_t kdrv_sdc_sd_host_t::clock
```

4.18.5.53 cmd12_force_evt

```
uint16_t kdrv_sdc_reg_t::cmd12_force_evt
```

4.18.5.54 cmd20_support

```
uint32_t kdrv_sdc_sd_scr_t::cmd20_support
```

4.18.5.55 cmd23_support

```
uint32_t kdrv_sdc_sd_scr_t::cmd23_support
```

4.18.5.56 cmd_argu

```
uint32_t kdrv_sdc_reg_t::cmd_argu
```

4.18.5.57 cmd_reg

```
uint16_t kdrv_sdc_reg_t::cmd_reg
```

4.18.5.58 cmd_resphi

```
uint64_t kdrv_sdc_reg_t::cmd_resphi
```

Kneron Confidential

4.18.5.59 cmd_resplo

```
uint64_t kdrv_sdc_reg_t::cmd_resplo
```

4.18.5.60 cmd_set

```
uint8_t kdrv_sdc_mmc_ext_csd_t::cmd_set
```

4.18.5.61 cmd_set_rev

```
uint8_t kdrv_sdc_mmc_ext_csd_t::cmd_set_rev
```

4.18.5.62 cmpl_mask

```
volatile uint8_t kdrv_sdc_sdcard_info_t::cmpl_mask
```

4.18.5.63 content_prot_app

```
uint32_t kdrv_sdc_mmc_csd_t::content_prot_app
```

4.18.5.64 COPY

```
uint32_t kdrv_sdc_mmc_csd_t::COPY
```

4.18.5.65 copy [1/2]

```
uint32_t kdrv_sdc_csd_v1_t::copy
```

4.18.5.66 copy [2/2]

```
uint32_t kdrv_sdc_csd_v2_t::copy
```

Kneron Confidential

4.18.5.67 cprm_init

```
uint32_t kdrv_sdc_sdcard_info_t::cprm_init
```

4.18.5.68 csd_hi

```
uint64_t kdrv_sdc_sdcard_info_t::csd_hi
```

4.18.5.69 csd_lo

```
uint64_t kdrv_sdc_sdcard_info_t::csd_lo
```

4.18.5.70 csd_mmc

```
kdrv_sdc_mmc_csd_t kdrv_sdc_sdcard_info_t::csd_mmc
```

4.18.5.71 csd_structure [1/4]

```
uint32_t kdrv_sdc_mmc_csd_t::csd_structure
```

4.18.5.72 csd_structure [2/4]

```
uint8_t kdrv_sdc_mmc_ext_csd_t::csd_structure
```

4.18.5.73 csd_structure [3/4]

```
uint32_t kdrv_sdc_csd_v1_t::csd_structure
```

4.18.5.74 csd_structure [4/4]

```
uint32_t kdrv_sdc_csd_v2_t::csd_structure
```

Kneron Confidential

4.18.5.75 csd_ver1

kdrv_sdc_csd_v1_t kdrv_sdc_sdcard_info_t::csd_ver1

4.18.5.76 csd_ver2

kdrv_sdc_csd_v2_t kdrv_sdc_sdcard_info_t::csd_ver2

4.18.5.77 dat_bus_width

uint32_t kdrv_sd_status_t::dat_bus_width

4.18.5.78 data_present

uint32_t kdrv_sdc_res_t::data_present

4.18.5.79 data_stat_after_erase

uint32_t kdrv_sdc_sd_scr_t::data_stat_after_erase

4.18.5.80 default_ecc

uint32_t kdrv_sdc_mmc_csd_t::default_ecc

4.18.5.81 dma_hndshk

uint32_t kdrv_sdc_reg_t::dma_hndshk

4.18.5.82 drive

uint32_t kdrv_sdc_sdcard_info_t::drive

Kneron Confidential

4.18.5.83 dsr

uint16_t kdrv_sdc_sdcard_info_t::dsr

4.18.5.84 dsr_imp [1/3]

uint32_t kdrv_sdc_mmc_csd_t::dsr_imp

4.18.5.85 dsr_imp [2/3]

uint32_t kdrv_sdc_csd_v1_t::dsr_imp

4.18.5.86 dsr_imp [3/3]

uint32_t kdrv_sdc_csd_v2_t::dsr_imp

4.18.5.87 ecc

uint32_t kdrv_sdc_mmc_csd_t::ecc

4.18.5.88 enh_size_mult

uint8_t kdrv_sdc_mmc_ext_csd_t::enh_size_mult[3]

4.18.5.89 enh_start_addr

uint8_t kdrv_sdc_mmc_ext_csd_t::enh_start_addr[4]

4.18.5.90 erase_blk_en [1/2]

uint32_t kdrv_sdc_csd_v1_t::erase_blk_en

Kneron Confidential

4.18.5.91 erase_blk_en [2/2]

uint32_t kdrv_sdc_csd_v2_t::erase_blk_en

4.18.5.92 erase_group_def

uint8_t kdrv_sdc_mmc_ext_csd_t::erase_group_def

4.18.5.93 erase_grp_mult

uint32_t kdrv_sdc_mmc_csd_t::erase_grp_mult

4.18.5.94 erase_grp_size

uint32_t kdrv_sdc_mmc_csd_t::erase_grp_size

4.18.5.95 erase_offset

uint32_t kdrv_sd_status_t::erase_offset

4.18.5.96 erase_size

uint8_t kdrv_sd_status_t::erase_size[2]

4.18.5.97 erase_timeout

uint32_t kdrv_sd_status_t::erase_timeout

4.18.5.98 erase_timeout_mult

uint8_t kdrv_sdc_mmc_ext_csd_t::erase_timeout_mult

Kneron Confidential

4.18.5.99 erased_mem_cont

```
uint8_t kdrv_sdc_mmc_ext_csd_t::erased_mem_cont
```

4.18.5.100 erasing

```
uint8_t kdrv_sdc_flow_info_t::erasing
```

4.18.5.101 err_en

```
uint16_t kdrv_sdc_reg_t::err_en
```

4.18.5.102 err_sig_en

```
uint16_t kdrv_sdc_reg_t::err_sig_en
```

4.18.5.103 err_sts [1/2]

```
uint16_t kdrv_sdc_reg_t::err_sts
```

4.18.5.104 err_sts [2/2]

```
volatile uint16_t kdrv_sdc_sdcard_info_t::err_sts
```

4.18.5.105 ext_csd_mmc

```
kdrv_sdc_mmc_ext_csd_t kdrv_sdc_sdcard_info_t::ext_csd_mmc
```

4.18.5.106 ext_csd_rev

```
uint8_t kdrv_sdc_mmc_ext_csd_t::ext_csd_rev
```

Kneron Confidential

4.18.5.107 fifo_depth [1/2]

```
uint32_t kdrv_sdc_sdcard_info_t::fifo_depth
```

4.18.5.108 fifo_depth [2/2]

```
uint16_t kdrv_sdc_res_t::fifo_depth
```

4.18.5.109 file_format [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::file_format
```

4.18.5.110 file_format [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::file_format
```

4.18.5.111 file_format [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::file_format
```

4.18.5.112 file_format_grp [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::file_format_grp
```

4.18.5.113 file_format_grp [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::file_format_grp
```

4.18.5.114 file_format_grp [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::file_format_grp
```

Kneron Confidential

4.18.5.115 flow_set

kdrv_sdc_flow_info_t kdrv_sdc_sdcard_info_t::flow_set

4.18.5.116 force_evt

uint16_t kdrv_sdc_reg_t::force_evt

4.18.5.117 fw_config

uint8_t kdrv_sdc_mmc_ext_csd_t::fw_config

4.18.5.118 gp_size_mult

uint8_t kdrv_sdc_mmc_ext_csd_t::gp_size_mult[12]

4.18.5.119 hc_erase_grp_size

uint8_t kdrv_sdc_mmc_ext_csd_t::hc_erase_grp_size

4.18.5.120 hc_wp_grp_size

uint8_t kdrv_sdc_mmc_ext_csd_t::hc_wp_grp_size

4.18.5.121 hcreg

uint8_t kdrv_sdc_reg_t::hcreg

4.18.5.122 hcver

uint16_t kdrv_sdc_reg_t::hcver

Kneron Confidential

4.18.5.123 host

```
kdrv_sdc_sd_host_t* kdrv_sdc_res_t::host
```

4.18.5.124 host_ctl2

```
uint16_t kdrv_sdc_reg_t::host_ctl2
```

4.18.5.125 hs_timing

```
uint8_t kdrv_sdc_mmc_ext_csd_t::hs_timing
```

4.18.5.126 hw_attr

```
uint32_t kdrv_sdc_reg_t::hw_attr
```

4.18.5.127 in_data_hi

```
uint32_t kdrv_sdc_reg_t::in_data_hi
```

4.18.5.128 in_data_lo

```
uint32_t kdrv_sdc_reg_t::in_data_lo
```

4.18.5.129 in_key_hi

```
uint32_t kdrv_sdc_reg_t::in_key_hi
```

4.18.5.130 in_key_lo

```
uint32_t kdrv_sdc_reg_t::in_key_lo
```

Kneron Confidential

4.18.5.131 infinite_mode

kdrv_sdc_infinite_test_e kdrv_sdc_res_t::infinite_mode

4.18.5.132 inhibit_datchk

uint32_t kdrv_sdc_res_t::inhibit_datchk

4.18.5.133 ini_timeout_ap

uint8_t kdrv_sdc_mmc_ext_csd_t::ini_timeout_ap

4.18.5.134 intr_en

uint16_t kdrv_sdc_reg_t::intr_en

4.18.5.135 intr_sig_en

uint16_t kdrv_sdc_reg_t::intr_sig_en

4.18.5.136 intr_sts

uint16_t kdrv_sdc_reg_t::intr_sts

4.18.5.137 ip_ver

uint32_t kdrv_sdc_reg_t::ip_ver

4.18.5.138 kmu_hi

uint32_t kdrv_sdc_sdcard_info_t::kmu_hi

Kneron Confidential

4.18.5.139 kmu_lo

```
uint32_t kdrv_sdc_sdcard_info_t::kmu_lo
```

4.18.5.140 lgth

```
uint16_t kdrv_sdc_adma2desc_table_t::lgth
```

4.18.5.141 line_bound

```
uint16_t kdrv_sdc_flow_info_t::line_bound
```

4.18.5.142 max_clk

```
uint32_t kdrv_sdc_sd_host_t::max_clk
```

4.18.5.143 max_curr

```
uint64_t kdrv_sdc_reg_t::max_curr
```

4.18.5.144 max_dtr

```
uint32_t kdrv_sdc_sdcard_info_t::max_dtr
```

4.18.5.145 max_enh_size_mult

```
uint8_t kdrv_sdc_mmc_ext_csd_t::max_enh_size_mult[3]
```

4.18.5.146 mem_present

```
uint8_t kdrv_sdc_sdcard_info_t::mem_present
```

Kneron Confidential

4.18.5.147 min_clk

```
uint32_t kdrv_sdc_sd_host_t::min_clk
```

4.18.5.148 min_perf_ddr_r_8_52

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_ddr_r_8_52
```

4.18.5.149 min_perf_ddr_w_8_52_8_52

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_ddr_w_8_52_8_52
```

4.18.5.150 min_perf_r_4_26

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_r_4_26
```

4.18.5.151 min_perf_r_8_26_4_52

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_r_8_26_4_52
```

4.18.5.152 min_perf_r_8_52

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_r_8_52
```

4.18.5.153 min_perf_w_4_26

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_w_4_26
```

4.18.5.154 min_perf_w_8_26_4_52

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_w_8_26_4_52
```

Kneron Confidential

4.18.5.155 min_perf_w_8_52

```
uint8_t kdrv_sdc_mmc_ext_csd_t::min_perf_w_8_52
```

4.18.5.156 nsac [1/3]

```
uint8_t kdrv_sdc_mmc_csd_t::nsac
```

4.18.5.157 nsac [2/3]

```
uint8_t kdrv_sdc_csd_v1_t::nsac
```

4.18.5.158 nsac [3/3]

```
uint8_t kdrv_sdc_csd_v2_t::nsac
```

4.18.5.159 num_io_func

```
uint8_t kdrv_sdc_sdcard_info_t::num_io_func
```

4.18.5.160 num_of_blks

```
uint32_t kdrv_sdc_sdcard_info_t::num_of_blks
```

4.18.5.161 num_of_boot_blks

```
uint32_t kdrv_sdc_sdcard_info_t::num_of_boot_blks
```

4.18.5.162 ocr

```
uint32_t kdrv_sdc_sdcard_info_t::ocr
```

Kneron Confidential

4.18.5.163 ocr_avail

```
uint32_t kdrv_sdc_sd_host_t::ocr_avail
```

4.18.5.164 out_data_hi

```
uint32_t kdrv_sdc_reg_t::out_data_hi
```

4.18.5.165 out_data_lo

```
uint32_t kdrv_sdc_reg_t::out_data_lo
```

4.18.5.166 partition_conf

```
uint8_t kdrv_sdc_mmc_ext_csd_t::partition_conf
```

4.18.5.167 partition_setting_completed

```
uint8_t kdrv_sdc_mmc_ext_csd_t::partition_setting_completed
```

4.18.5.168 partitioning_attribute

```
uint8_t kdrv_sdc_mmc_ext_csd_t::partitioning_attribute
```

4.18.5.169 partitioning_support

```
uint8_t kdrv_sdc_mmc_ext_csd_t::partitioning_support
```

4.18.5.170 performance_move

```
uint8_t kdrv_sd_status_t::performance_move
```

Kneron Confidential

4.18.5.171 perm_write_protect [1/3]

uint32_t kdrv_sdc_mmc_csd_t::perm_write_protect

4.18.5.172 perm_write_protect [2/3]

uint32_t kdrv_sdc_csd_v1_t::perm_write_protect

4.18.5.173 perm_write_protect [3/3]

uint32_t kdrv_sdc_csd_v2_t::perm_write_protect

4.18.5.174 power

uint8_t kdrv_sdc_sd_host_t::power

4.18.5.175 power_class

uint8_t kdrv_sdc_mmc_ext_csd_t::power_class

4.18.5.176 present_state

uint32_t kdrv_sdc_reg_t::present_state

4.18.5.177 preset_val_ddr50

uint16_t kdrv_sdc_reg_t::preset_val_ddr50

4.18.5.178 preset_val_ds

uint16_t kdrv_sdc_reg_t::preset_val_ds

Kneron Confidential

4.18.5.179 preset_val_hs

```
uint16_t kdrv_sdc_reg_t::preset_val_hs
```

4.18.5.180 preset_val_init

```
uint16_t kdrv_sdc_reg_t::preset_val_init
```

4.18.5.181 preset_val_sdr104

```
uint16_t kdrv_sdc_reg_t::preset_val_sdr104
```

4.18.5.182 preset_val_sdr12

```
uint16_t kdrv_sdc_reg_t::preset_val_sdr12
```

4.18.5.183 preset_val_sdr25

```
uint16_t kdrv_sdc_reg_t::preset_val_sdr25
```

4.18.5.184 preset_val_sdr50

```
uint16_t kdrv_sdc_reg_t::preset_val_sdr50
```

4.18.5.185 protected_drive

```
uint32_t kdrv_sdc_sdcard_info_t::protected_drive
```

4.18.5.186 pwr_cl_26_195

```
uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_26_195
```

Kneron Confidential

4.18.5.187 pwr_cl_26_360

```
uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_26_360
```

4.18.5.188 pwr_cl_52_195

```
uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_52_195
```

4.18.5.189 pwr_cl_52_360

```
uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_52_360
```

4.18.5.190 pwr_cl_ddr_52_195

```
uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_ddr_52_195
```

4.18.5.191 pwr_cl_ddr_52_360

```
uint8_t kdrv_sdc_mmc_ext_csd_t::pwr_cl_ddr_52_360
```

4.18.5.192 pwr_ctl

```
uint8_t kdrv_sdc_reg_t::pwr_ctl
```

4.18.5.193 r2w_factor [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::r2w_factor
```

4.18.5.194 r2w_factor [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::r2w_factor
```

Kneron Confidential

4.18.5.195 r2w_factor [3/3]

uint32_t kdrv_sdc_csd_v2_t::r2w_factor

4.18.5.196 rca

uint16_t kdrv_sdc_sdcard_info_t::rca

4.18.5.197 read_bl_len [1/3]

uint32_t kdrv_sdc_mmc_csd_t::read_bl_len

4.18.5.198 read_bl_len [2/3]

uint32_t kdrv_sdc_csd_v1_t::read_bl_len

4.18.5.199 read_bl_len [3/3]

uint32_t kdrv_sdc_csd_v2_t::read_bl_len

4.18.5.200 read_bl_partial [1/3]

uint32_t kdrv_sdc_mmc_csd_t::read_bl_partial

4.18.5.201 read_bl_partial [2/3]

uint32_t kdrv_sdc_csd_v1_t::read_bl_partial

4.18.5.202 read_bl_partial [3/3]

uint32_t kdrv_sdc_csd_v2_t::read_bl_partial

Kneron Confidential

4.18.5.203 read_blk_misalign [1/3]

uint32_t kdrv_sdc_mmc_csd_t::read_blk_misalign

4.18.5.204 read_blk_misalign [2/3]

uint32_t kdrv_sdc_csd_v1_t::read_blk_misalign

4.18.5.205 read_blk_misalign [3/3]

uint32_t kdrv_sdc_csd_v2_t::read_blk_misalign

4.18.5.206 ref_wr_sec_c

uint8_t kdrv_sdc_mmc_ext_csd_t::ref_wr_sec_c

4.18.5.207 reserved [1/2]

uint32_t kdrv_sdc_reg_t::reserved[28]

4.18.5.208 reserved [2/2]

uint8_t kdrv_sdc_flow_info_t::reserved

4.18.5.209 reserved1 [1/6]

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved1[7]

4.18.5.210 reserved1 [2/6]

uint32_t kdrv_sdc_mmc_csd_t::reserved1

Kneron Confidential

4.18.5.211 reserved1 [3/6]

uint32_t kdrv_sd_status_t::reserved1

4.18.5.212 reserved1 [4/6]

uint32_t kdrv_sdc_csd_v1_t::reserved1

4.18.5.213 reserved1 [5/6]

uint32_t kdrv_sdc_csd_v2_t::reserved1

4.18.5.214 reserved1 [6/6]

uint32_t kdrv_sdc_sd_scr_t::reserved1

4.18.5.215 reserved10

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved10

4.18.5.216 reserved11

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved11[3]

4.18.5.217 reserved12

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved12

4.18.5.218 reserved13

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved13

Kneron Confidential

4.18.5.219 reserved14

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved14

4.18.5.220 reserved15

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved15

4.18.5.221 reserved16

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved16

4.18.5.222 reserved17

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved17

4.18.5.223 reserved18

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved18

4.18.5.224 reserved19

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved19

4.18.5.225 reserved2 [1/6]

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved2[262]

4.18.5.226 reserved2 [2/6]

uint32_t kdrv_sdc_reg_t::reserved2[6]

Kneron Confidential

4.18.5.227 reserved2 [3/6]

uint32_t kdrv_sdc_mmc_csd_t::reserved2

4.18.5.228 reserved2 [4/6]

uint32_t kdrv_sd_status_t::reserved2

4.18.5.229 reserved2 [5/6]

uint32_t kdrv_sdc_csd_v1_t::reserved2

4.18.5.230 reserved2 [6/6]

uint32_t kdrv_sdc_csd_v2_t::reserved2

4.18.5.231 reserved20

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved20

4.18.5.232 reserved21

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved21

4.18.5.233 reserved22

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved22

4.18.5.234 reserved23

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved23

Kneron Confidential

4.18.5.235 reserved24

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved24[5]

4.18.5.236 reserved25

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved25

4.18.5.237 reserved26

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved26

4.18.5.238 reserved27

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved27[134]

4.18.5.239 reserved3 [1/5]

uint32_t kdrv_sdc_mmc_csd_t::reserved3

4.18.5.240 reserved3 [2/5]

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved3

4.18.5.241 reserved3 [3/5]

uint32_t kdrv_sd_status_t::reserved3

4.18.5.242 reserved3 [4/5]

uint32_t kdrv_sdc_csd_v1_t::reserved3

Kneron Confidential

4.18.5.243 reserved3 [5/5]

uint32_t kdrv_sdc_csd_v2_t::reserved3

4.18.5.244 reserved4 [1/5]

uint8_t kdrv_sd_status_t::reserved4[11]

4.18.5.245 reserved4 [2/5]

uint32_t kdrv_sdc_reg_t::reserved4[19]

4.18.5.246 reserved4 [3/5]

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved4[2]

4.18.5.247 reserved4 [4/5]

uint32_t kdrv_sdc_csd_v1_t::reserved4

4.18.5.248 reserved4 [5/5]

uint32_t kdrv_sdc_csd_v2_t::reserved4

4.18.5.249 reserved5 [1/3]

uint8_t kdrv_sd_status_t::reserved5[39]

4.18.5.250 reserved5 [2/3]

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved5

Kneron Confidential

4.18.5.251 reserved5 [3/3]

uint32_t kdrv_sdc_csd_v2_t::reserved5

4.18.5.252 reserved6

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved6

4.18.5.253 reserved7

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved7

4.18.5.254 reserved8

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved8

4.18.5.255 reserved9

uint8_t kdrv_sdc_mmc_ext_csd_t::reserved9

4.18.5.256 Reserver5

uint32_t kdrv_sdc_csd_v1_t::Reserver5

4.18.5.257 reserver6

uint32_t kdrv_sdc_csd_v2_t::reserver6

4.18.5.258 reserverd2

uint32_t kdrv_sdc_sd_scr_t::reserverd2

Kneron Confidential

4.18.5.259 reserverd3

```
uint32_t kdrv_sdc_sd_scr_t::reserverd3
```

4.18.5.260 resp_hi

```
uint64_t kdrv_sdc_sdcard_info_t::resp_hi
```

4.18.5.261 resp_lo

```
uint64_t kdrv_sdc_sdcard_info_t::resp_lo
```

4.18.5.262 response_type

```
uint32_t kdrv_sdc_res_t::response_type
```

4.18.5.263 rpmb_size_mult

```
uint8_t kdrv_sdc_mmc_ext_csd_t::rpmb_size_mult
```

4.18.5.264 rst_n_function

```
uint8_t kdrv_sdc_mmc_ext_csd_t::rst_n_function
```

4.18.5.265 s_a_timeout

```
uint8_t kdrv_sdc_mmc_ext_csd_t::s_a_timeout
```

4.18.5.266 s_c_vcc

```
uint8_t kdrv_sdc_mmc_ext_csd_t::s_c_vcc
```

Kneron Confidential

4.18.5.267 s_c_vccq

```
uint8_t kdrv_sdc_mmc_ext_csd_t::s_c_vccq
```

4.18.5.268 s_cmd_set

```
uint8_t kdrv_sdc_mmc_ext_csd_t::s_cmd_set
```

4.18.5.269 scr

```
kdrv_sdc_sd_scr_t kdrv_sdc_sdcard_info_t::scr
```

4.18.5.270 scr_structure

```
uint32_t kdrv_sdc_sd_scr_t::scr_structure
```

4.18.5.271 sd_bus_widths

```
uint32_t kdrv_sdc_sd_scr_t::sd_bus_widths
```

4.18.5.272 sd_card_type_hi

```
uint32_t kdrv_sd_status_t::sd_card_type_hi
```

4.18.5.273 sd_card_type_lo

```
uint32_t kdrv_sd_status_t::sd_card_type_lo
```

4.18.5.274 sd_security

```
uint32_t kdrv_sdc_sd_scr_t::sd_security
```

Kneron Confidential

4.18.5.275 sd_spec

```
uint32_t kdrv_sdc_sd_scr_t::sd_spec
```

4.18.5.276 sd_spec3

```
uint32_t kdrv_sdc_sd_scr_t::sd_spec3
```

4.18.5.277 sd_sts

```
kdrv_sd_status_t kdrv_sdc_sdcard_info_t::sd_sts
```

4.18.5.278 sdc_reg

```
volatile kdrv_sdc_reg_t* kdrv_sdc_res_t::sdc_reg
```

4.18.5.279 sdma_addr

```
uint32_t kdrv_sdc_reg_t::sdma_addr
```

4.18.5.280 sdma_intr

```
uint8_t kdrv_sdc_sdcard_info_t::sdma_intr
```

4.18.5.281 sec_bad_blk_mgmnt

```
uint8_t kdrv_sdc_mmc_ext_csd_t::sec_bad_blk_mgmnt
```

4.18.5.282 sec_count

```
uint32_t kdrv_sdc_mmc_ext_csd_t::sec_count
```

Kneron Confidential

4.18.5.283 sec_erase_mult

uint8_t kdrv_sdc_mmc_ext_csd_t::sec_erase_mult

4.18.5.284 sec_feature_support

uint8_t kdrv_sdc_mmc_ext_csd_t::sec_feature_support

4.18.5.285 sec_trim_mult

uint8_t kdrv_sdc_mmc_ext_csd_t::sec_trim_mult

4.18.5.286 secr_table_port

uint32_t kdrv_sdc_reg_t::secr_table_port

4.18.5.287 sector_size [1/2]

uint32_t kdrv_sdc_csd_v1_t::sector_size

4.18.5.288 sector_size [2/2]

uint32_t kdrv_sdc_csd_v2_t::sector_size

4.18.5.289 secured_mode

uint32_t kdrv_sd_status_t::secured_mode

4.18.5.290 share_bus_ctl

uint32_t kdrv_sdc_reg_t::share_bus_ctl

Kneron Confidential

4.18.5.291 size_of_protected_area

uint32_t kdrv_sd_status_t::size_of_protected_area

4.18.5.292 slt_intr_sts

uint16_t kdrv_sdc_reg_t::slt_intr_sts

4.18.5.293 softrst

uint8_t kdrv_sdc_reg_t::softrst

4.18.5.294 spec_vers

uint32_t kdrv_sdc_mmc_csd_t::spec_vers

4.18.5.295 speed

kdrv_sdc_bus_speed_e kdrv_sdc_sdcard_info_t::speed

4.18.5.296 speed_class

uint8_t kdrv_sd_status_t::speed_class

4.18.5.297 switch_sts

uint8_t kdrv_sdc_sdcard_info_t::switch_sts[64]

4.18.5.298 sync_abort

kdrv_sdc_abort_type_e kdrv_sdc_flow_info_t::sync_abort

Kneron Confidential

4.18.5.299 sys_freq

uint32_t kdrv_sdc_sdcard_info_t::sys_freq

4.18.5.300 taac [1/3]

uint8_t kdrv_sdc_mmc_csd_t::taac

4.18.5.301 taac [2/3]

uint8_t kdrv_sdc_csd_v1_t::taac

4.18.5.302 taac [3/3]

uint8_t kdrv_sdc_csd_v2_t::taac

4.18.5.303 timeout_ctl

uint8_t kdrv_sdc_reg_t::timeout_ctl

4.18.5.304 timeout_ms

uint32_t kdrv_sdc_res_t::timeout_ms

4.18.5.305 tmp_write_protect [1/3]

uint32_t kdrv_sdc_mmc_csd_t::tmp_write_protect

4.18.5.306 tmp_write_protect [2/3]

uint32_t kdrv_sdc_csd_v1_t::tmp_write_protect

Kneron Confidential

4.18.5.307 tmp_write_protect [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::tmp_write_protect
```

4.18.5.308 tran_speed [1/3]

```
uint8_t kdrv_sdc_mmc_csd_t::tran_speed
```

4.18.5.309 tran_speed [2/3]

```
uint8_t kdrv_sdc_csd_v1_t::tran_speed
```

4.18.5.310 tran_speed [3/3]

```
uint8_t kdrv_sdc_csd_v2_t::tran_speed
```

4.18.5.311 trim_mult

```
uint8_t kdrv_sdc_mmc_ext_csd_t::trim_mult
```

4.18.5.312 txmode

```
uint16_t kdrv_sdc_reg_t::txmode
```

4.18.5.313 u8_sdma_lock

```
uint8_t kdrv_sdc_sdcard_info_t::u8_sdma_lock
```

4.18.5.314 use_dma

```
kdrv_sdc_transfer_type_e kdrv_sdc_flow_info_t::use_dma
```

Kneron Confidential

4.18.5.315 user_wp

```
uint8_t kdrv_sdc_mmc_ext_csd_t::user_wp
```

4.18.5.316 vdd_r_curr_max [1/2]

```
uint32_t kdrv_sdc_mmc_csd_t::vdd_r_curr_max
```

4.18.5.317 vdd_r_curr_max [2/2]

```
uint32_t kdrv_sdc_csd_v1_t::vdd_r_curr_max
```

4.18.5.318 vdd_r_curr_min [1/2]

```
uint32_t kdrv_sdc_mmc_csd_t::vdd_r_curr_min
```

4.18.5.319 vdd_r_curr_min [2/2]

```
uint32_t kdrv_sdc_csd_v1_t::vdd_r_curr_min
```

4.18.5.320 vdd_w_curr_max [1/2]

```
uint32_t kdrv_sdc_mmc_csd_t::vdd_w_curr_max
```

4.18.5.321 vdd_w_curr_max [2/2]

```
uint32_t kdrv_sdc_csd_v1_t::vdd_w_curr_max
```

4.18.5.322 vdd_w_curr_min [1/2]

```
uint32_t kdrv_sdc_mmc_csd_t::vdd_w_curr_min
```

Kneron Confidential

4.18.5.323 vdd_w_curr_min [2/2]

```
uint32_t kdrv_sdc_csd_v1_t::vdd_w_curr_min
```

4.18.5.324 vendor_reg0

```
uint32_t kdrv_sdc_reg_t::vendor_reg0
```

4.18.5.325 vendor_reg1

```
uint32_t kdrv_sdc_reg_t::vendor_reg1
```

4.18.5.326 vendor_reg2

```
uint32_t kdrv_sdc_reg_t::vendor_reg2
```

4.18.5.327 vendor_reg3

```
uint32_t kdrv_sdc_reg_t::vendor_reg3
```

4.18.5.328 vendor_reg4

```
uint32_t kdrv_sdc_reg_t::vendor_reg4
```

4.18.5.329 vendor_reg5

```
uint32_t kdrv_sdc_reg_t::vendor_reg5
```

4.18.5.330 vendor_reg6

```
uint32_t kdrv_sdc_reg_t::vendor_reg6
```

Kneron Confidential

4.18.5.331 wakeup_ctl

```
uint8_t kdrv_sdc_reg_t::wakeup_ctl
```

4.18.5.332 wp_grp_enable [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::wp_grp_enable
```

4.18.5.333 wp_grp_enable [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::wp_grp_enable
```

4.18.5.334 wp_grp_enable [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::wp_grp_enable
```

4.18.5.335 wp_grp_size [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::wp_grp_size
```

4.18.5.336 wp_grp_size [2/3]

```
uint32_t kdrv_sdc_csd_v1_t::wp_grp_size
```

4.18.5.337 wp_grp_size [3/3]

```
uint32_t kdrv_sdc_csd_v2_t::wp_grp_size
```

4.18.5.338 write_bl_len [1/3]

```
uint32_t kdrv_sdc_mmc_csd_t::write_bl_len
```

Kneron Confidential

4.18.5.339 write_bl_len [2/3]

uint32_t kdrv_sdc_csd_v1_t::write_bl_len

4.18.5.340 write_bl_len [3/3]

uint32_t kdrv_sdc_csd_v2_t::write_bl_len

4.18.5.341 write_bl_partial [1/3]

uint32_t kdrv_sdc_mmc_csd_t::write_bl_partial

4.18.5.342 write_bl_partial [2/3]

uint32_t kdrv_sdc_csd_v1_t::write_bl_partial

4.18.5.343 write_bl_partial [3/3]

uint32_t kdrv_sdc_csd_v2_t::write_bl_partial

4.18.5.344 write_blk_misalign [1/3]

uint32_t kdrv_sdc_mmc_csd_t::write_blk_misalign

4.18.5.345 write_blk_misalign [2/3]

uint32_t kdrv_sdc_csd_v1_t::write_blk_misalign

4.18.5.346 write_blk_misalign [3/3]

uint32_t kdrv_sdc_csd_v2_t::write_blk_misalign

Kneron Confidential

4.19 KDRV_SPIF

Kneron spi flash driver.

Functions

- void kdrv_spif_initialize (void)
  Initialize spi flash include hardware setting, operation frequency, and flash status check.
- void kdrv_spif_memxfer_initialize (uint8_t flash_mode, uint8_t mem_mode)
  Initialize spi flash for memxfer include hardware setting, operation frequency, and flash status check.
- kdrv_status_t kdrv_spif_uninitialize (void)
  Uninitialize spi flash and clear related variables.
- void kdrv_spif_set_commands (uint32_t cmd0, uint32_t cmd1, uint32_t cmd2, uint32_t cmd3)
  set spi communication commands including read/write by 3/4bytes address, dummy byte size, operation mode, etc
- void kdrv_spif_wait_command_complete (void)
  Check status bit to wait until command completed.
- void kdrv_spif_wait_rx_full (void)
  Wait until the RX FIFO is full so ready to read.
- void kdrv_spif_wait_tx_empty (void)
  Wait until the TX FIFO is empty so ready to write.
- uint32_t kdrv_spif_rxfifo_depth (void)
  Check the RX FIFO size, unit in byte.
- uint32_t kdrv_spif_txfifo_depth (void)
  Check the TX FIFO size, unit in byte.
- void kdrv_spif_read_data (uint32_t *buf, uint32_t length)
  read data from specific index in spi flash
- void kdrv_spif_write_data (uint8_t *buf, uint32_t length)
  write data to specific index in spi flash
- void kdrv_spif_read_Rx_FIFO (uint32_t *buf_word, uint16_t *buf_word_index, uint32_t target_byte)
  read Rx FIFO data
- void kdrv_spif_check_status_till_ready_2 (void)
  check status till the progress is done and ready for next step
- void kdrv_spif_check_status_till_ready (void)
  wait command completed and check status till it's ready
- void kdrv_spif_check_quad_status_till_ready (void)
  wait quad read command completed and check status till ready

4.19.1 Detailed Description

Kneron spi flash driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.19.2 Function Documentation

Kneron Confidential

4.19.2.1 kdrv_spif_check_quad_status_till_ready()

```
void kdrv_spif_check_quad_status_till_ready (
            void  )
```

wait quad read command completed and check status till ready

Kneron Confidential

Parameters

in	*N/A*	

Returns

N/A

4.19.2.2 kdrv_spif_check_status_till_ready()

```
void kdrv_spif_check_status_till_ready (
            void  )
```

wait command completed and check status till it's ready

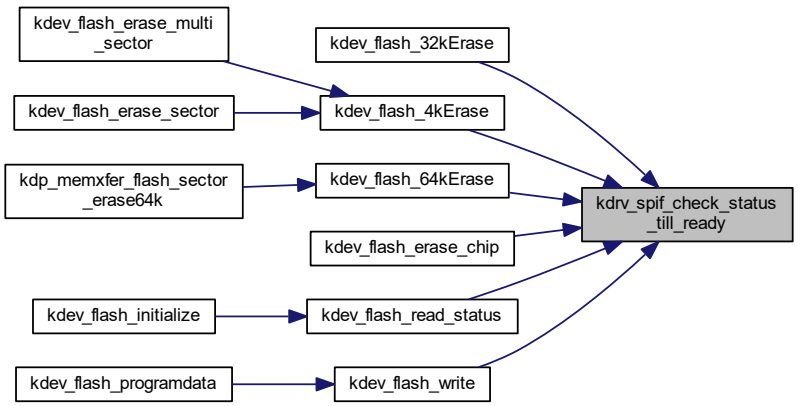
Parameters

in	*N/A*	

Returns

N/A

Here is the caller graph for this function:



4.19.2.3 kdrv_spif_check_status_till_ready_2()

```
void kdrv_spif_check_status_till_ready_2 (
            void  )
```

check status till the progress is done and ready for next step

Kneron Confidential

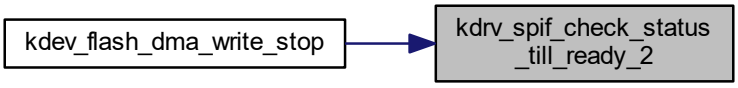
Parameters

in	*N/A*	

Returns

N/A

Here is the caller graph for this function:



4.19.2.4 kdrv_spif_initialize()

```
void kdrv_spif_initialize (
            void  )
```

Initialize spi flash include hardware setting, operation frequency, and flash status check.

Parameters

in	*N/A*	

Returns

N/A

Note

This API MUST be called before using the Read/write APIs for spi flash.

Here is the caller graph for this function:

Kneron Confidential

4.19.2.5 kdrv_spif_memxfer_initialize()

```
void kdrv_spif_memxfer_initialize (
            uint8_t flash_mode,
            uint8_t mem_mode )
```

Initialize spi flash for memxfer include hardware setting, operation frequency, and flash status check.

Parameters

in	*flash_mode*	flash operating mode mem_mode memory operating mode

Returns

N/A

4.19.2.6 kdrv_spif_read_data()

```
void kdrv_spif_read_data (
            uint32_t * buf,
            uint32_t length )
```

read data from specific index in spi flash

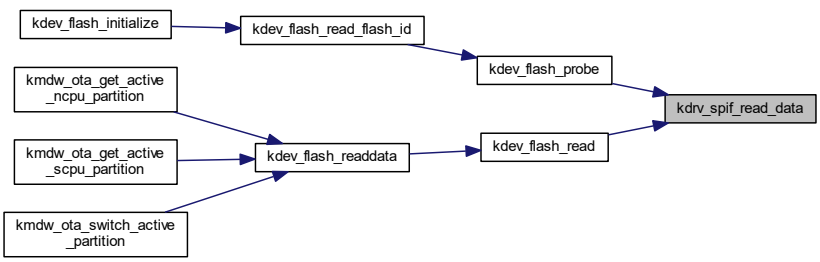
Parameters

in	**buf*	buffer for the data read from flash length data size

Returns

N/A

Here is the caller graph for this function:



Kneron Confidential

4.19.2.7 kdrv_spif_read_Rx_FIFO()

```
void kdrv_spif_read_Rx_FIFO (
            uint32_t * buf_word,
            uint16_t * buf_word_index,
            uint32_t target_byte )
```

read Rx FIFO data

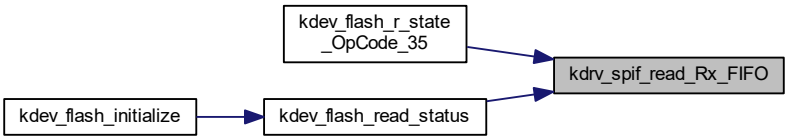
Parameters

in	*∗buf_word*	buffer for the data read from flash ∗buf_word_index start from specific flash index target_byte data size

Returns

N/A

Here is the caller graph for this function:



4.19.2.8 kdrv_spif_rxfifo_depth()

```
uint32_t kdrv_spif_rxfifo_depth (
            void  )
```

Check the RX FIFO size, unit in byte.

Parameters

in	*N/A*	

Returns

>0 RX FIFO depth

Kneron Confidential

4.19.2.9 kdrv_spif_set_commands()

```
void kdrv_spif_set_commands (
            uint32_t cmd0,
            uint32_t cmd1,
            uint32_t cmd2,
            uint32_t cmd3 )
```

set spi communication commands including read/write by 3/4bytes address, dummy byte size, operation mode, etc

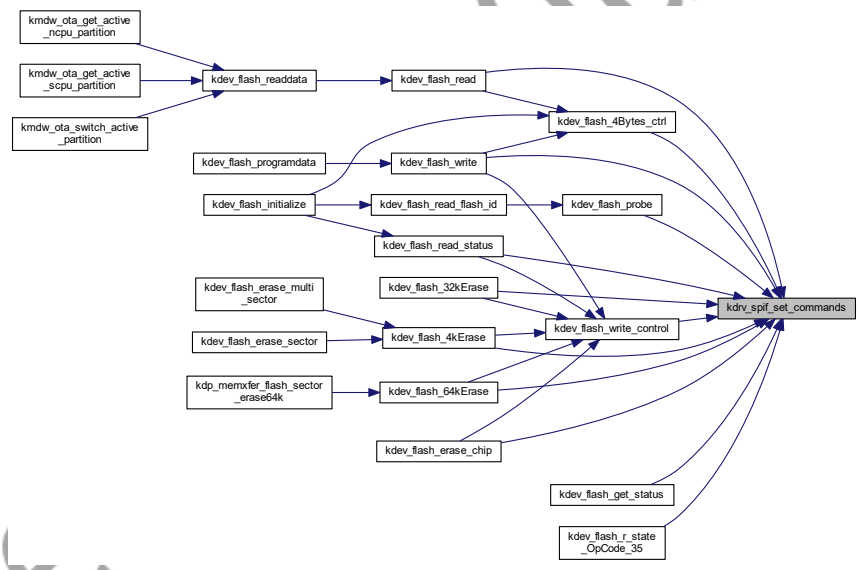
Parameters

in	*cmd0*	~ 3

Returns

N/A

Here is the caller graph for this function:



4.19.2.10 kdrv_spif_txfifo_depth()

```
uint32_t kdrv_spif_txfifo_depth (
            void  )
```

Check the TX FIFO size, unit in byte.

Parameters

in	*N/A*	

Returns

>0 TX FIFO depth

4.19.2.11 kdrv_spif_uninitialize()

```
kdrv_status_t kdrv_spif_uninitialize (
            void  )
```

Uninitialize spi flash and clear related variables.

Parameters

in	*N/A*	

Returns

kdrv_status_t

4.19.2.12 kdrv_spif_wait_command_complete()

```
void kdrv_spif_wait_command_complete (
            void  )
```

Check status bit to wait until command completed.

Parameters

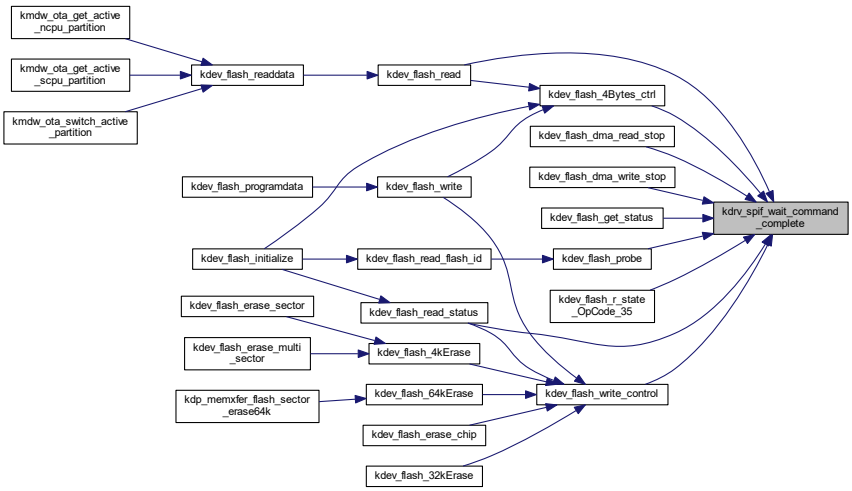
in	*N/A*	

Kneron Confidential

Returns

N/A

Here is the caller graph for this function:



4.19.2.13 kdrv_spif_wait_rx_full()

```
void kdrv_spif_wait_rx_full (
            void  )
```

Wait until the RX FIFO is full so ready to read.

Parameters

in	*N/A*	

Returns

N/A

4.19.2.14 kdrv_spif_wait_tx_empty()

```
void kdrv_spif_wait_tx_empty (
            void  )
```

Wait until the TX FIFO is empty so ready to write.

Kneron Confidential

Parameters

in	*N/A*	

Returns

N/A

4.19.2.15 kdrv_spif_write_data()

```
void kdrv_spif_write_data (
            uint8_t * buf,
            uint32_t length )
```

write data to specific index in spi flash

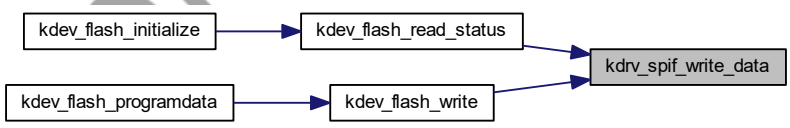
Parameters

in	**buf*	buffer for the data to write to flash length data size

Returns

N/A

Here is the caller graph for this function:



Kneron Confidential

4.20 KDRV_SYSTEM

Kneron system driver.

Macros

- #define FLAGS_SOURCE_READY_EVT 0x91ad
- #define SCPU_FW 1
- #define NCPU_FW 2

Enumerations

- enum { SUBSYS_NPU = 1, SUBSYS_PD_NPU, SUBSYS_LCDC, SUBSYS_NCPU }

  Enumeration of system reset.

Functions

- void kdrv_system_init (void)

  System initialize.
- void kdrv_system_init_ncpu (void)

  NCPU system initialize.
- void kdrv_system_reset (int32_t subsystem)

  System reset.

4.20.1 Detailed Description

Kneron system driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.20.2 Macro Definition Documentation

4.20.2.1 FLAGS_SOURCE_READY_EVT

```
#define FLAGS_SOURCE_READY_EVT 0x91ad
```

Kneron Confidential

4.20.2.2 NCPU_FW

```
#define NCPU_FW 2
```

4.20.2.3 SCPU_FW

```
#define SCPU_FW 1
```

4.20.3 Enumeration Type Documentation

4.20.3.1 anonymous enum

```
anonymous enum
```

Enumeration of system reset.

Enumerator

SUBSYS_NPU	Software reset for NPU
SUBSYS_PD_NPU	Software reset for whole NPU domain
SUBSYS_LCDC	Software reset for LCDC
SUBSYS_NCPU	The signal controls SYSRESETn of NCPU

4.20.4 Function Documentation

4.20.4.1 kdrv_system_init()

```
void kdrv_system_init (
            void  )
```

System initialize.

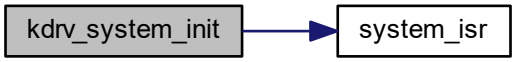
Turn on NPU/DDR power domain and enable some main clock PLL .

Kneron Confidential

Returns

N/A

Here is the call graph for this function:



4.20.4.2 kdrv_system_init_ncpu()

```
void kdrv_system_init_ncpu (
            void  )
```

NCPU system initialize.

Enable NCPU/NPU and some main PLL clock .

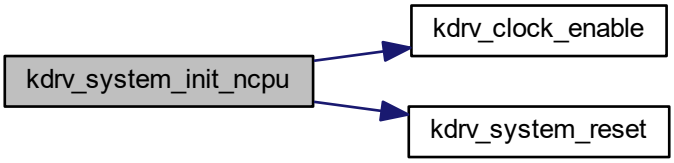
Returns

N/A

Note

This API should be called after kdrv_system_init() to make sure NPU/DDR power domain is powered on.

Here is the call graph for this function:



4.20.4.3 kdrv_system_reset()

```
void kdrv_system_reset (
            int32_t subsystem )
```

System reset.

Kneron Confidential

Parameters

in	*subsystem*	subsystem reset id

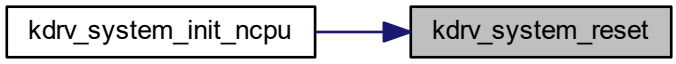
Note

SUBSYS_NPU: reset NPU SUBSYS_PD_NPU: reset whole NPU domain(clk+ddr phy) SUBSYS_LCDC: reset LCDC SUBSYS_NCPU: reset NCPU

Returns

N/A

Here is the caller graph for this function:



Kneron Confidential

4.21 KDRV_TIMER

Kneron timer driver.

Typedefs

- typedef void(∗ timer_cb_fr_isr_t) (cb_event_t argument, void ∗arg)

  Enumerations of all timer status for kdrv_timer_set.

Enumerations

- enum cb_event_t { TIMER_M1_TIMEOUT, TIMER_M2_TIMEOUT, TIMER_OF_TIMEOUT }

  Enumerations of all timer call back even return status.
- enum timer_stat_t { TIMER_PAUSE, TIMER_START, TIMER_STAT_DEFAULT }

  Enumerations of all timer status for kdrv_timer_set.

Functions

- kdrv_status_t kdrv_timer_initialize (void)

  Enable clock, init timer ip, register IRQ/ISR function.
- kdrv_status_t kdrv_timer_uninitialize (void)

  Disable clock, and timer IRQ.
- kdrv_status_t kdrv_timer_open (uint32_t ∗TimerId, timer_cb_fr_isr_t event_cb, void ∗arg)

  Request one timer id for further usage.
- kdrv_status_t kdrv_timer_close (uint32_t ∗TimerId)

  Close specific timer id.
- kdrv_status_t kdrv_timer_set (uint32_t ∗TimerId, uint32_t Intval, timer_stat_t State)

  Set specific timer with interval and status.
- kdrv_status_t kdrv_timer_perf_open (uint32_t ∗TimerId)

  Open a timer with specific timer id for performance measurement.
- kdrv_status_t kdrv_timer_perf_set (uint32_t ∗TimerId)

  Set specific timer for performance measurment usage.
- kdrv_status_t kdrv_timer_perf_get_instant (uint32_t ∗TimerId, uint32_t ∗instant, uint32_t ∗time)

  Get time consumption.
- kdrv_status_t kdrv_timer_perf_reset (uint32_t ∗TimerId)

  Reset performance timer.
- kdrv_status_t kdrv_timer_delay_ms (uint32_t msec)

  Let system delay ms.
- kdrv_status_t kdrv_timer_delay_us (uint32_t usec)

  Let system delay us.
- kdrv_status_t kdrv_timer_perf_measure_start (void)

  Start to use performance measurement function.
- kdrv_status_t kdrv_timer_perf_measure_get (uint32_t ∗instant, uint32_t ∗time)

  Get time interval.

Kneron Confidential

4.21.1 Detailed Description

Kneron timer driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.21.2 Typedef Documentation

4.21.2.1 timer_cb_fr_isr_t

```
typedef void(* timer_cb_fr_isr_t) (cb_event_t argument, void *arg)
```

Enumerations of all timer status for kdrv_timer_set.

4.21.3 Enumeration Type Documentation

4.21.3.1 cb_event_t

```
enum cb_event_t
```

Enumerations of all timer call back even return status.

Enumerator

TIMER_M1_TIMEOUT	reach timer M1 level
TIMER_M2_TIMEOUT	reach timer M2 level
TIMER_OF_TIMEOUT	timer overfloor

4.21.3.2 timer_stat_t

```
enum timer_stat_t
```

Enumerations of all timer status for kdrv_timer_set.

Enumerator

TIMER_PAUSE	
TIMER_START	
TIMER_STAT_DEFAULT	

Kneron Confidential

4.21.4 Function Documentation

4.21.4.1 kdrv_timer_close()

```
kdrv_status_t kdrv_timer_close (
            uint32_t * TimerId )
```

Close specific timer id.

Parameters

in	*TimerId*	pointer of timer id

Returns

kdrv_status_t see kdrv_status_t

4.21.4.2 kdrv_timer_delay_ms()

```
kdrv_status_t kdrv_timer_delay_ms (
            uint32_t msec )
```

Let system delay ms.

Parameters

in	*usec*	time interval(ms).

Returns

kdrv_status_t see kdrv_status_t

4.21.4.3 kdrv_timer_delay_us()

```
kdrv_status_t kdrv_timer_delay_us (
            uint32_t usec )
```

Let system delay us.

Kneron Confidential

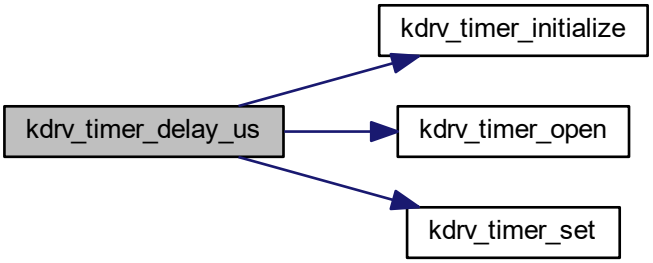
Parameters

in	*usec*	time interval(us).

Returns

kdrv_status_t see kdrv_status_t

Here is the call graph for this function:



4.21.4.4 kdrv_timer_initialize()

```
kdrv_status_t kdrv_timer_initialize (
            void  )
```

Enable clock, init timer ip, register IRQ/ISR function.

Parameters

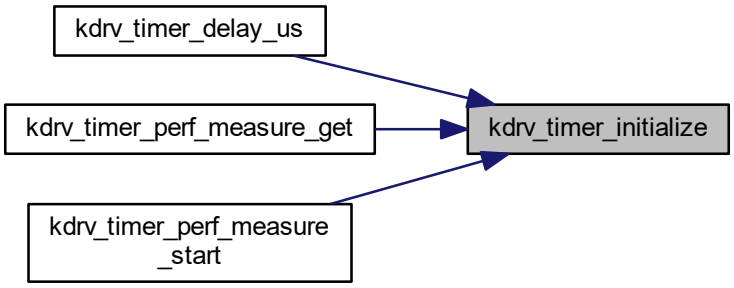
in	*N/A*	

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.21.4.5 kdrv_timer_open()

```
kdrv_status_t kdrv_timer_open (
            uint32_t * TimerId,
            timer_cb_fr_isr_t event_cb,
            void * arg )
```

Request one timer id for further usage.

Parameters

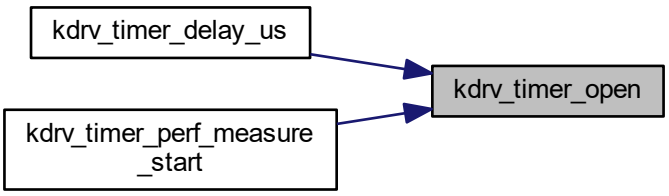
out	*TimerId*	pointer of timer id.
in	*event_cb*	timer_cb_fr_isr_t, see timer_cb_fr_isr_t
in	*arg*	user define argument

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.21.4.6 kdrv_timer_perf_get_instant()

```
kdrv_status_t kdrv_timer_perf_get_instant (
            uint32_t * TimerId,
            uint32_t * instant,
            uint32_t * time )
```

Get time consumption.

Parameters

in	*TimerId*	pointer of timer id
out	*instant*	pointer of time instant register

Returns

Time cunsumption

4.21.4.7 kdrv_timer_perf_measure_get()

```
kdrv_status_t kdrv_timer_perf_measure_get (
            uint32_t * instant,
            uint32_t * time )
```

Get time interval.

Kneron Confidential

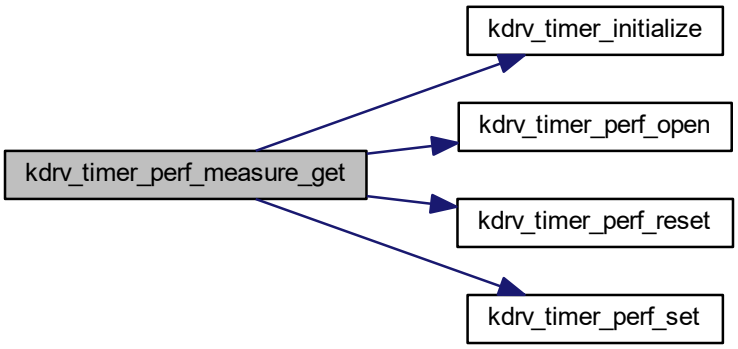
Parameters

in	*instant*	Difference time interval compare to last time instant.
in	*time*	Current time.

Returns

kdrv_status_t see kdrv_status_t

Here is the call graph for this function:



4.21.4.8 kdrv_timer_perf_measure_start()

```
kdrv_status_t kdrv_timer_perf_measure_start (
            void  )
```

Start to use performance measurement function.

Parameters

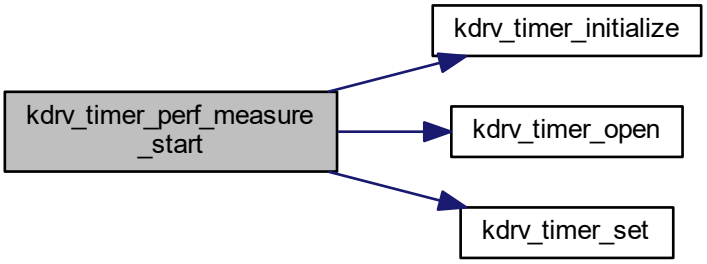
in	*N/A*	

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the call graph for this function:



4.21.4.9 kdrv_timer_perf_open()

```
kdrv_status_t kdrv_timer_perf_open (
            uint32_t * TimerId )
```

Open a timer with specific timer id for performance measurement.

Note

Need use kdrv_timer_perf_set() to start timing measurement.

Parameters

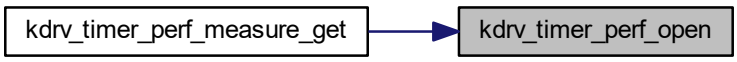
out	*TimerId*	pointer of timer id

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.21.4.10 kdrv_timer_perf_reset()

```
kdrv_status_t kdrv_timer_perf_reset (
            uint32_t * TimerId )
```

Reset performance timer.

Parameters

in	*TimerId*	pointer of timer id

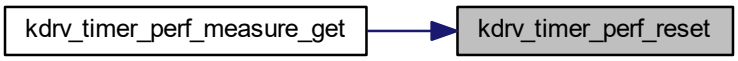
Returns

kdrv_status_t see kdrv_status_t

Note

After call kdrv_timer_perf_open(), you should reset this timer first.
Example:
uint32_t perftimerid; kdrv_timer_perf_open(&pftimerid);
kdrv_timer_perf_reset(&pftimerid);

Here is the caller graph for this function:



Kneron Confidential

4.21.4.11 kdrv_timer_perf_set()

```
kdrv_status_t kdrv_timer_perf_set (
            uint32_t * TimerId )
```

Set specific timer for performance measurment usage.

Parameters

in	*TimerId*	pointer of timer id

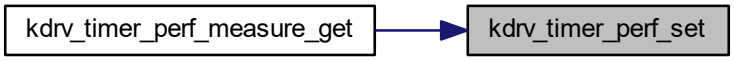
Returns

kdrv_status_t see kdrv_status_t

Note

You should call kdrv_timer_perf_open() and kdrv_timer_perf_reset() firstly before call this API.
Example:
uint32_t perftimerid;
kdrv_timer_perf_open(&pftimerid);
kdrv_timer_perf_reset(&pftimerid);
kdrv_timer_perf_set(&perftimerid);

Here is the caller graph for this function:



4.21.4.12 kdrv_timer_set()

```
kdrv_status_t kdrv_timer_set (
            uint32_t * TimerId,
            uint32_t Intval,
            timer_stat_t State )
```

Set specific timer with interval and status.

Parameters

in	*TimerId*	pointer of timer id
in	*Interval*	set timer interval
in	*timer_stat*	see timer_stat, see timer_stat_t

Kneron Confidential

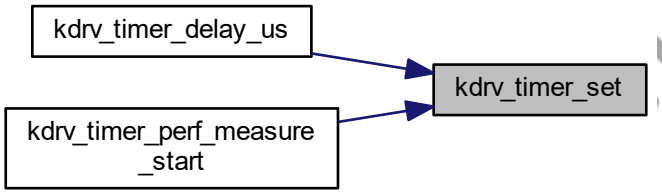
Returns

kdrv_status_t see kdrv_status_t

Note

This API should be called after kdrv_timer_open()
Example:
uint32_t timerid;
kdrv_timer_open(&timerid, NULL, NULL);
kdrv_timer_set(&timerid, 5000000, TIMER_START);

Here is the caller graph for this function:



4.21.4.13 kdrv_timer_uninitialize()

```
kdrv_status_t kdrv_timer_uninitialize (
            void  )
```

Disable clock, and timer IRQ.

Parameters

in	*N/A*

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.22 KDRV_UART

Kneron UART driver.

Data Structures

- struct kdrv_uart_config_t

  The structure of UART configuration parameters.
- struct kdrv_uart_fifo_config_t

  The structure of UART FIFO configuration parameters.

Macros

- #define BAUD_921600 (UART_CLOCK / 14745600)

  Enumerations of UART baud rate.
- #define BAUD_460800 (UART_CLOCK / 7372800)
- #define BAUD_115200 (UART_CLOCK / 1843200)
- #define BAUD_57600 (UART_CLOCK / 921600)
- #define BAUD_38400 (UART_CLOCK / 614400)
- #define BAUD_19200 (UART_CLOCK / 307200)
- #define BAUD_14400 (UART_CLOCK / 230400)
- #define BAUD_9600 (UART_CLOCK / 153600)
- #define BAUD_4800 (UART_CLOCK / 76800)
- #define BAUD_2400 (UART_CLOCK / 38400)
- #define BAUD_1200 (UART_CLOCK / 19200)
- #define PARITY_NONE 0

  The definition of UART parity.
- #define PARITY_ODD 1
- #define PARITY_EVEN 2
- #define PARITY_MARK 3
- #define PARITY_SPACE 4

Typedefs

- typedef int32_t kdrv_uart_handle_t
- typedef void(∗ kdrv_uart_callback_t) (uint32_t event)

Enumerations

- enum kdrv_uart_mode_t { UART_MODE_ASYN_RX = 0x1, UART_MODE_ASYN_TX = 0x2, UART_MODE_SYNC_RX = 0x4, UART_MODE_SYNC_TX = 0x8 }

  Enumerations of UART mode parameters.
- enum kdrv_uart_dev_id_t {
  UART0_DEV = 0, UART1_DEV, UART2_DEV, UART3_DEV,
  UART4_DEV, TOTAL_UART_DEV }

  Enumerations of UART device instance parameters.
- enum DRVUART_PORT {
  DRVUART_PORT0 = 0, DRVUART_PORT1 = 1, DRVUART_PORT2 = 2, DRVUART_PORT3 = 3,
  DRVUART_PORT4 = 4 }

  Enumerations of UART port parameters.
- enum kdrv_uart_control_t {
  UART_CTRL_CONFIG, UART_CTRL_FIFO_RX, UART_CTRL_FIFO_TX, UART_CTRL_LOOPBACK,
  UART_CTRL_TX_EN, UART_CTRL_RX_EN, UART_CTRL_ABORT_TX, UART_CTRL_ABORT_RX,
  UART_CTRL_TIMEOUT_RX, UART_CTRL_TIMEOUT_TX }

  Enumerations of UART control hardware signals.

Kneron Confidential

Functions

- kdrv_status_t kdrv_uart_initialize (void)

  UART driver initialization, it shall be called once in lifecycle.
- kdrv_status_t kdrv_uart_uninitialize (void)

  UART driver uninitialization.
- kdrv_status_t kdrv_uart_open (kdrv_uart_handle_t *handle, uint8_t com_port, uint32_t mode, kdrv_uart_callback_t callback)

  Open one UART port and acquire a uart port handle.
- kdrv_status_t kdrv_uart_configure (kdrv_uart_handle_t handle, kdrv_uart_control_t prop, uint8_t *val)

  set control for the opened UART port
- kdrv_status_t kdrv_uart_write (kdrv_uart_handle_t hdl, uint8_t *buf, uint32_t len)

  write data to uart port, such as command, parameters, but not suitable for chunk data
- kdrv_status_t kdrv_uart_get_char (kdrv_uart_handle_t handle, char *ch)

  read character data from UART port
- kdrv_status_t kdrv_uart_read (kdrv_uart_handle_t handle, uint8_t *buf, uint32_t len)

  read data from the UART port
- kdrv_status_t kdrv_uart_close (kdrv_uart_handle_t handle)

  close the UART port
- uint32_t kdrv_uart_get_rx_count (kdrv_uart_handle_t handle)

  get char number in RX buffer
- uint32_t kdrv_uart_get_tx_count (kdrv_uart_handle_t handle)

  get char number in TX buffer

Variables

- uint32_t kdrv_uart_config_t::baudrate
- uint8_t kdrv_uart_config_t::data_bits
- uint8_t kdrv_uart_config_t::frame_length
- uint8_t kdrv_uart_config_t::stop_bits
- uint8_t kdrv_uart_config_t::parity_mode
- bool kdrv_uart_config_t::fifo_en
- bool kdrv_uart_fifo_config_t::bEnFifo
- uint8_t kdrv_uart_fifo_config_t::fifo_trig_level

4.22.1 Detailed Description

Kneron UART driver.

Here are the design highlight points:

- The architecture adopts a lightweight non-thread design
- ISR driven architecture.
- Can support both synchronous and asynchronous mode
- Utilizes FIFO advantage to reduce interrupts and improve robust to accommodate more latency than normal.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.22.2 Macro Definition Documentation

4.22.2.1 BAUD_115200

```
#define BAUD_115200 (UART_CLOCK / 1843200)
```

UART baud rate: 115200.

4.22.2.2 BAUD_1200

```
#define BAUD_1200 (UART_CLOCK / 19200)
```

UART baud rate: 1200.

4.22.2.3 BAUD_14400

```
#define BAUD_14400 (UART_CLOCK / 230400)
```

UART baud rate: 14400.

4.22.2.4 BAUD_19200

```
#define BAUD_19200 (UART_CLOCK / 307200)
```

UART baud rate: 19200.

4.22.2.5 BAUD_2400

```
#define BAUD_2400 (UART_CLOCK / 38400)
```

UART baud rate: 2400.

4.22.2.6 BAUD_38400

```
#define BAUD_38400 (UART_CLOCK / 614400)
```

UART baud rate: 38400.

4.22.2.7 BAUD_460800

```
#define BAUD_460800 (UART_CLOCK / 7372800)
```

UART baud rate: 460800.

Kneron Confidential

4.22.2.8 BAUD_4800

```
#define BAUD_4800 (UART_CLOCK / 76800)
```

UART baud rate: 4800.

4.22.2.9 BAUD_57600

```
#define BAUD_57600 (UART_CLOCK / 921600)
```

UART baud rate: 57600.

4.22.2.10 BAUD_921600

```
#define BAUD_921600 (UART_CLOCK / 14745600)
```

Enumerations of UART baud rate.

UART baud rate: 921600.

4.22.2.11 BAUD_9600

```
#define BAUD_9600 (UART_CLOCK / 153600)
```

UART baud rate: 9600.

4.22.2.12 PARITY_EVEN

```
#define PARITY_EVEN 2
```

Even Parity

4.22.2.13 PARITY_MARK

```
#define PARITY_MARK 3
```

Stick odd Parity

4.22.2.14 PARITY_NONE

```
#define PARITY_NONE 0
```

The definition of UART parity.

Disable Parity

Kneron Confidential

4.22.2.15 PARITY_ODD

```
#define PARITY_ODD 1
```

Odd Parity

4.22.2.16 PARITY_SPACE

```
#define PARITY_SPACE 4
```

Stick even Parity

4.22.3 Typedef Documentation

4.22.3.1 kdrv_uart_callback_t

```
typedef void(* kdrv_uart_callback_t) (uint32_t event)
```

4.22.3.2 kdrv_uart_handle_t

```
typedef int32_t kdrv_uart_handle_t
```

4.22.4 Enumeration Type Documentation

4.22.4.1 DRVUART_PORT

```
enum DRVUART_PORT
```

Enumerations of UART port parameters.

Enumerator

DRVUART_PORT0	UART port 0
DRVUART_PORT1	UART port 1
DRVUART_PORT2	UART port 2
DRVUART_PORT3	UART port 3
DRVUART_PORT4	UART port 4

Kneron Confidential

4.22.4.2 kdrv_uart_control_t

enum kdrv_uart_control_t

Enumerations of UART control hardware signals.

Enumerator

UART_CTRL_CONFIG	set kdrv_uart_config_t
UART_CTRL_FIFO_RX	set kdrv_uart_fifo_config_t
UART_CTRL_FIFO_TX	set kdrv_uart_fifo_config_t
UART_CTRL_LOOPBACK	UART loopback enable
UART_CTRL_TX_EN	UART transmitter enable
UART_CTRL_RX_EN	UART receiver enable
UART_CTRL_ABORT_TX	UART abort transmitter
UART_CTRL_ABORT_RX	UART abort receiver
UART_CTRL_TIMEOUT_RX	UART receiver timeout value
UART_CTRL_TIMEOUT_TX	UART transmitter timeout value

4.22.4.3 kdrv_uart_dev_id_t

enum kdrv_uart_dev_id_t

Enumerations of UART device instance parameters.

Enumerator

UART0_DEV	UART device instance 0
UART1_DEV	UART device instance 1
UART2_DEV	UART device instance 2
UART3_DEV	UART device instance 3
UART4_DEV	UART device instance 4
TOTAL_UART_DEV	Total UART device instances

4.22.4.4 kdrv_uart_mode_t

enum kdrv_uart_mode_t

Enumerations of UART mode parameters.

Kneron Confidential

Enumerator

UART_MODE_ASYN_RX	UART asynchronous receiver mode.
UART_MODE_ASYN_TX	UART asynchronous transmitter mode.
UART_MODE_SYNC_RX	UART synchronous receiver mode.
UART_MODE_SYNC_TX	UART synchronous transmitter mode.

4.22.5 Function Documentation

4.22.5.1 kdrv_uart_close()

```
kdrv_status_t kdrv_uart_close (
            kdrv_uart_handle_t handle )
```

close the UART port

Parameters

in	*handle*	device handle for an UART port

Returns

kdrv_status_t see kdrv_status_t

4.22.5.2 kdrv_uart_configure()

```
kdrv_status_t kdrv_uart_configure (
            kdrv_uart_handle_t handle,
            kdrv_uart_control_t prop,
            uint8_t * val )
```

set control for the opened UART port

Parameters

in	*handle*	device handle for an UART port
in	*prop*	control enumeration
in	*val*	pointer to control value/structure

Returns

kdrv_status_t see kdrv_status_t

Kneron Confidential

4.22.5.3 kdrv_uart_get_char()

```
kdrv_status_t kdrv_uart_get_char (
            kdrv_uart_handle_t handle,
            char * ch )
```

read character data from UART port

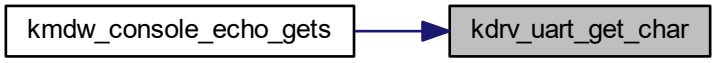
Parameters

in	*handle*	device handle for an UART port
out	*ch*	character data

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.22.5.4 kdrv_uart_get_rx_count()

```
uint32_t kdrv_uart_get_rx_count (
            kdrv_uart_handle_t handle )
```

get char number in RX buffer

Parameters

in	*handle*	device handle for an UART port

Returns

number of RX count in the buffer

Kneron Confidential

4.22.5.5 kdrv_uart_get_tx_count()

```
uint32_t kdrv_uart_get_tx_count (
            kdrv_uart_handle_t handle )
```

get char number in TX buffer

Parameters

in	*handle*	device handle for an UART port

Returns

number of TX count in the buffer

4.22.5.6 kdrv_uart_initialize()

```
kdrv_status_t kdrv_uart_initialize (
            void  )
```

UART driver initialization, it shall be called once in lifecycle.

Returns

kdrv_status_t see kdrv_status_t

4.22.5.7 kdrv_uart_open()

```
kdrv_status_t kdrv_uart_open (
            kdrv_uart_handle_t * handle,
            uint8_t com_port,
            uint32_t mode,
            kdrv_uart_callback_t callback )
```

Open one UART port and acquire a uart port handle.

This API will open a UART device (com_port: 0-5) for use.
It will return a UART device handle for future device reference.
The client can choose work mode: asynchronization or synchronization.
Synchronization mode will poll the hardware status to determine send/receiving point,
it will consume more power and introduce more delay to system execution.
But in the case of non-thread light weight environment, such as message log function, this mode is easy and suitable.
Asynchronization mode lets the driver interrupt driven, save more system power and more efficient,
the client needs to have a thread to listen/wait for the event/signal sent from callback function.
Callback function parameter 'callback' will be registered with this device which is mandatory for async mode,
will be invoked whenever Tx/Rx complete or timeout occur.
This callback function should be very thin, can only be used to set flag or send signals

Kneron Confidential

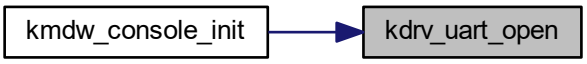
Parameters

out	*handle*	a handle of an UART port
in	*com_port*	UART port id
in	*mode*	bit combination of kdrv_uart_mode_t
in	*callback*	user callback function

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.22.5.8 kdrv_uart_read()

```
kdrv_status_t kdrv_uart_read (
            kdrv_uart_handle_t handle,
            uint8_t * data,
            uint32_t len )
```

read data from the UART port

The client can call this API to receive UART data from remote side.
Depending on the work mode, a little bit different behavior exists there.
In synchronous mode, the API call will not return until all data was received physically.
In asynchronous mode, the API call shall return immediately with UART_API_RX_BUSY.
When enough bytes are received or timeout occurs, the client registered callback function will be invoked.
The client shall have a very thin code there to set flags/signals. The client thread shall be listening the signal after this API call.
The client shall allocate the receiving buffer with max possible receiving length.
When one frame is sent out, after 4 chars transmission time, a timeout interrupt will be generated.

Parameters

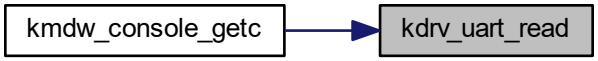
in	*handle*	device handle for an UART port
out	*buf*	data buffer
in	*len*	data buffer length

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Read data from UART receiver. Input data: buffer for receving data len: size Data buffer size in bytes handle: Driver handle return driver status Here is the caller graph for this function:



4.22.5.9 kdrv_uart_uninitialize()

```
kdrv_status_t kdrv_uart_uninitialize (
            void  )
```

UART driver uninitialization.

Returns

kdrv_status_t see kdrv_status_t

4.22.5.10 kdrv_uart_write()

```
kdrv_status_t kdrv_uart_write (
            kdrv_uart_handle_t hdl,
            uint8_t * buf,
            uint32_t len )
```

write data to uart port, such as command, parameters, but not suitable for chunk data

The client calls this API to send data out to remote side.
Depending on the work mode, a little bit different behavior exists there.
In synchronous mode, the API call will not return until all data was sent out physically;
In asynchronous mode, the API call shall return immediately with UART_API_TX_BUSY.
When all the buffer data is sent out, the client registered callback function will be invoked.
The client shall have a very thin code there to set flags/signals. The client thread shall be listening the signal after this API call.

Kneron Confidential

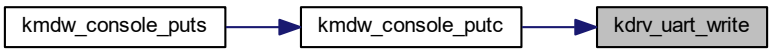
Parameters

in	*handle*	device handle for an UART port
in	*buf*	data buffer
in	*len*	data buffer length

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.22.6 Variable Documentation

4.22.6.1 baudrate

```
uint32_t kdrv_uart_config_t::baudrate
```

UART baud rate.

4.22.6.2 bEnFifo

```
bool kdrv_uart_fifo_config_t::bEnFifo
```

4.22.6.3 data_bits

```
uint8_t kdrv_uart_config_t::data_bits
```

UART data bits, a data character contains 5∼8 data bits.

4.22.6.4 fifo_en

```
bool kdrv_uart_config_t::fifo_en
```

UART fifo mode.

Kneron Confidential

4.22.6.5 fifo_trig_level

`uint8_t kdrv_uart_fifo_config_t::fifo_trig_level`

4.22.6.6 frame_length

`uint8_t kdrv_uart_config_t::frame_length`

UART frame length, non-zero value for FIR mode

4.22.6.7 parity_mode

`uint8_t kdrv_uart_config_t::parity_mode`

UART partity mode, see UART_PARITY_DEF

4.22.6.8 stop_bits

`uint8_t kdrv_uart_config_t::stop_bits`

UART stop bit, a data character is proceded by a start bit
and is followed by an optional parity bit and a stop bit.

Kneron Confidential

4.23 KDRV_USBD

Kneron USB device mode driver.

Data Structures

- struct kdrv_usbd_event_t

  USB event, it includes kdrv_usbd_event_name_t and related data.

Macros

- #define MAX_USBD_CONFIG 1
- #define MAX_USBD_INTERFACE 1
- #define MAX_USBD_ENDPOINT 4

Enumerations

- enum kdrv_usbd_speed_t { KDRV_USBD_HIGH_SPEED, KDRV_USBD_FULL_SPEED }

  Enumeration of connection speed.
- enum kdrv_usbd_event_name_t {
  KDRV_USBD_EVENT_BUS_RESET = 1, KDRV_USBD_EVENT_BUS_SUSPEND, KDRV_USBD_EVENT_BUS_RESUME,
  KDRV_USBD_EVENT_SETUP_PACKET,
  KDRV_USBD_EVENT_DEV_CONFIGURED, KDRV_USBD_EVENT_TRANSFER_BUF_FULL, KDRV_USBD_EVENT_TRAN
  KDRV_USBD_EVENT_TRANSFER_OUT,
  KDRV_USBD_EVENT_TRANSFER_TERMINATED, KDRV_USBD_EVENT_DMA_ERROR }

  Enumeration of USB event name type.
- enum kdrv_usbd_status_respond_t { KDRV_USBD_RESPOND_OK, KDRV_USBD_RESPOND_ERROR }

  Enumeration of code for response to host in control transfer.

Functions

- struct __attribute__ ((__packed__))

  8-byte setup packet struct
- kdrv_status_t kdrv_usbd_initialize (void)

  USB device mode driver initialization.
- kdrv_status_t kdrv_usbd_uninitialize (void)

  USB device mode driver uninitialization.
- kdrv_status_t kdrv_usbd_reset_device (void)

  reset device and then it can be re-enumerated by host
- kdrv_status_t kdrv_usbd_set_device_descriptor (kdrv_usbd_speed_t speed, kdrv_usbd_device_descriptor_t *dev_desc)

  configure device descriptor including configuration, interface and all endpoints descriptors
- kdrv_status_t kdrv_usbd_set_device_qualifier_descriptor (kdrv_usbd_speed_t speed, kdrv_usbd_device_qualifier_descriptor_t *dev_qual_desc)

  configure device qualifier descriptor, this is optional
- kdrv_status_t kdrv_usbd_register_thread_notification (osThreadId_t tid, uint32_t tflag)

  register user thread ID and thread flag for notifications including events or transfer completion/errors
- kdrv_status_t kdrv_usbd_set_enable (bool enable)

Kneron Confidential

set enable/disabale of USB device mode, host can enumerate this device only if device is enabled

- bool kdrv_usbd_is_dev_configured (void)

  check if device is enumerated and configured by a host
- kdrv_status_t kdrv_usbd_get_event (kdrv_usbd_event_t *uevent)

  get a usbd event, this is a blocking function for sync mode usage of USBD APIs
- kdrv_status_t kdrv_usbd_control_send (uint8_t *buf, uint32_t size, uint32_t timeout_ms)

  Control-IN transfer, send data to host through the control endpont.
- kdrv_status_t kdrv_usbd_control_receive (uint8_t *buf, uint32_t *size, uint32_t timeout_ms)

  Control-OUT transfer, receive data from host through the control endpont.
- kdrv_status_t kdrv_usbd_control_respond (kdrv_usbd_status_respond_t status)

  respond to host through control transfer in the status stage
- kdrv_status_t kdrv_usbd_reset_endpoint (uint32_t endpoint)

  reset specified endpoint
- kdrv_status_t kdrv_usbd_bulk_send (uint32_t endpoint, uint32_t *buf, uint32_t txLen, uint32_t timeout_ms)

  Bulk-IN transfser, send data to host through a bulk-in endpont in blocking mode.
- kdrv_status_t kdrv_usbd_bulk_send_async (uint32_t endpoint, uint32_t *buf, uint32_t txLen)

  Bulk-IN transfser, send data to host through a bulk-in endpont in non-blocking mode.
- kdrv_status_t kdrv_usbd_bulk_receive (uint32_t endpoint, uint32_t *buf, uint32_t *blen, uint32_t timeout_ms)

  Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in blocking mode.
- kdrv_status_t kdrv_usbd_bulk_receive_async (uint32_t endpoint, uint32_t *buf, uint32_t blen)

  Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in non-blocking mode.
- kdrv_status_t kdrv_usbd_interrupt_send (uint32_t endpoint, uint32_t *buf, uint32_t txLen, uint32_t timeout_ms)

  Interrupt-IN transfer in blocking mode.
- kdrv_status_t kdrv_usbd_interrupt_receive (uint32_t endpoint, uint32_t *buf, uint32_t *rxLen, uint32_t timeout_ms)

  Interrupt-OUT transfer in blocking mode.

Variables

- kdrv_usbd_setup_packet_t
- kdrv_usbd_endpoint_descriptor_t
- kdrv_usbd_interface_descriptor_t
- kdrv_usbd_config_descriptor_t
- kdrv_usbd_device_descriptor_t
- kdrv_usbd_device_qualifier_descriptor_t
- kdrv_usbd_event_name_t kdrv_usbd_event_t::ename
- kdrv_usbd_setup_packet_t kdrv_usbd_event_t::setup
- uint32_t kdrv_usbd_event_t::data1
- uint32_t kdrv_usbd_event_t::data2
- struct {
  uint32_t kdrv_usbd_event_t::data1
  uint32_t kdrv_usbd_event_t::data2
  }

- union {
  kdrv_usbd_setup_packet_t kdrv_usbd_event_t::setup
  struct {
  uint32_t kdrv_usbd_event_t::data1
  uint32_t kdrv_usbd_event_t::data2
  }
  };

Kneron Confidential

4.23.1 Detailed Description

Kneron USB device mode driver.

This USBD driver API implementation is based on an event-driven architecture.
For async mode API usage, to get notified of specific USB events,
user of USBD API needs to create a user thread to listen events by waiting for a specified thread flag (CMSIS-RTOS v2)
which is registered at early time.

Listening events is optional for sync mode usage by not setting notification for events and use synchronous mode API to perform transfers.
Once user is notified with the specified thread flag, a get-event API can be used to retrieve the exact USB event and take a corresponding action for it.
USBD handles hardware interrupts directly in ISR context, based on USB protocol to accomplish USB events and transfer work.

There are two layers of software for a complete USB device mode driver (software layer block diagram is shown as below),
one is USBD driver itself which provides a set of generic APIs with prefix "kdrv_usbd" them,
another is the function driver which can leverage USBD API to implement.

At present there is none of class drivers like MSC or CDC come with the USBD implementation, however users can implement their own function
driver for custom use cases.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.23.2 Macro Definition Documentation

4.23.2.1 MAX_USBD_CONFIG

```
#define MAX_USBD_CONFIG 1
```

maximum number of configuration descriptor

4.23.2.2 MAX_USBD_ENDPOINT

```
#define MAX_USBD_ENDPOINT 4
```

maximum number of endpoint descriptor

4.23.2.3 MAX_USBD_INTERFACE

```
#define MAX_USBD_INTERFACE 1
```

maximum number of interface descriptor

Kneron Confidential

4.23.3 Enumeration Type Documentation

4.23.3.1 kdrv_usbd_event_name_t

`enum kdrv_usbd_event_name_t`

Enumeration of USB event name type.

Enumerator

KDRV_USBD_EVENT_BUS_RESET	USBD event of bus reset
KDRV_USBD_EVENT_BUS_SUSPEND	USBD event of bus suspend
KDRV_USBD_EVENT_BUS_RESUME	USBD event of bus resume
KDRV_USBD_EVENT_SETUP_PACKET	USBD event of setup packet
KDRV_USBD_EVENT_DEV_CONFIGURED	USBD event of device configuration
KDRV_USBD_EVENT_TRANSFER_BUF_FULL	USBD event of transfer buffer full
KDRV_USBD_EVENT_TRANSFER_DONE	USBD event of transfer done
KDRV_USBD_EVENT_TRANSFER_OUT	USBD event of transfer out
KDRV_USBD_EVENT_TRANSFER_TERMINATED	USBD event of transfer terminated
KDRV_USBD_EVENT_DMA_ERROR	USBD event of DMA error

4.23.3.2 kdrv_usbd_speed_t

`enum kdrv_usbd_speed_t`

Enumeration of connection speed.

Enumerator

KDRV_USBD_HIGH_SPEED	USB high speed
KDRV_USBD_FULL_SPEED	USB full speed, not supported yet

4.23.3.3 kdrv_usbd_status_respond_t

`enum kdrv_usbd_status_respond_t`

Enumeration of code for response to host in control transfer.

Enumerator

KDRV_USBD_RESPOND_OK	send ACK in the status stage
KDRV_USBD_RESPOND_ERROR	send STALL in the status stage

Kneron Confidential

4.23.4 Function Documentation

4.23.4.1 __attribute__()

```
struct __attribute__ (
            (__packed__)  )
```

8-byte setup packet struct

Device qualifier descriptor struct.

Device descriptor struct.

Configuration descriptor struct.

Interface descriptor struct.

Endpoint descriptor struct.

4.23.4.2 kdrv_usbd_bulk_receive()

```
kdrv_status_t kdrv_usbd_bulk_receive (
            uint32_t endpoint,
            uint32_t * buf,
            uint32_t * blen,
            uint32_t timeout_ms )
```

Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in blocking mode.

Parameters

in	*endpoint*	a bulk-out endpoint address, should be the value from bEndpointAddress
out	*buf*	buffer for receiving data
in,out	*blen*	buffer length for input, actual transfered length for output
in	*timeout_ms*	timeout in millisecond

Returns

kdrv_status_t see kdrv_status_t

4.23.4.3 kdrv_usbd_bulk_receive_async()

```
kdrv_status_t kdrv_usbd_bulk_receive_async (
            uint32_t endpoint,
```

Kneron Confidential

```
            uint32_t * buf,
            uint32_t blen )
```

Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in non-blocking mode.

this works with kdrv_usbd_get_event(), when receiving a 'KDRV_USBD_EVENT_TRANSFER_OUT' event, user should commit a buffer for Bulk Out transfer through this function. when transfer is done by usbd, eihter a 'KDRV_USBD_EVENT_TRANSFER_DONE' or 'KDRV_USBD_EVENT_TRANSFER_BUF_FULL' event will be sent to user.

Parameters

in	*endpoint*	a bulk-out endpoint address, should be the value from bEndpointAddress
in	*buf*	buffer for receiving data
in	*blen*	buffer length

Returns

kdrv_status_t see kdrv_status_t

4.23.4.4 kdrv_usbd_bulk_send()

```
kdrv_status_t kdrv_usbd_bulk_send (
            uint32_t endpoint,
            uint32_t * buf,
            uint32_t txLen,
            uint32_t timeout_ms )
```

Bulk-IN transfser, send data to host through a bulk-in endpont in blocking mode.

Parameters

in	*endpoint*	a bulk-in endpoint address, should be the value from bEndpointAddress
in	*buf*	data to be sent to host
in	*txLen*	number of bytes to be transfered
in	*timeout_ms*	timeout in millisecond

Returns

kdrv_status_t see kdrv_status_t

4.23.4.5 kdrv_usbd_bulk_send_async()

```
kdrv_status_t kdrv_usbd_bulk_send_async (
            uint32_t endpoint,
```

Kneron Confidential

```
            uint32_t * buf,
            uint32_t txLen )
```

Bulk-IN transfser, send data to host through a bulk-in endpont in non-blocking mode.

user can commit a buffer for Bulk In transfer, and then wait for KDRV_USBD_EVENT_TRANSFER_DONE to be notified that the transfer is done or some error code if failed. This function works with kdrv_usbd_get_event().

Parameters

in	*endpoint*	a bulk-in endpoint address, should be the value from bEndpointAddress
in	*buf*	data to be sent to host
in	*txLen*	number of bytes to be transfered

Returns

kdrv_status_t see kdrv_status_t

4.23.4.6 kdrv_usbd_control_receive()

```
kdrv_status_t kdrv_usbd_control_receive (
            uint8_t * buf,
            uint32_t * size,
            uint32_t timeout_ms )
```

Control-OUT transfer, receive data from host through the control endpont.

for a user-defined vendor reqeust & control OUT & wLength $>$ 0, user should use this function to receive data from host, or respond an error via kdrv_usbd_control_respond(KDRV_USBD_RESPOND_ERROR) to claim STALL

Parameters

out	*buf*	buffer for receiving data
in	*size*	buffer length
in	*timeout_ms*	timeout in millisecond

Returns

kdrv_status_t see kdrv_status_t

4.23.4.7 kdrv_usbd_control_respond()

```
kdrv_status_t kdrv_usbd_control_respond (
            kdrv_usbd_status_respond_t status )
```

respond to host through control transfer in the status stage

this function is used as response function to report status for a user-defined vendor request

Kneron Confidential

Parameters

in	*status*	status, see kdrv_usbd_status_respond_t

Returns

kdrv_status_t see kdrv_status_t

4.23.4.8 kdrv_usbd_control_send()

```
kdrv_status_t kdrv_usbd_control_send (
            uint8_t * buf,
            uint32_t size,
            uint32_t timeout_ms )
```

Control-IN transfer, send data to host through the control endpont.

for a user-defined vendor reqeust & control IN & wLength $>$ 0, user should use this function to send data to host, or respond an error via kdrv_usbd_control_respond(KDRV_USBD_RESPOND_ERROR) to claim STALL

Parameters

in	*buf*	data to be sent to host
in	*size*	number of bytes to be transfered
in	*timeout_ms*	timeout in millisecond

Returns

kdrv_status_t see kdrv_status_t

4.23.4.9 kdrv_usbd_get_event()

```
kdrv_status_t kdrv_usbd_get_event (
            kdrv_usbd_event_t * uevent )
```

get a usbd event, this is a blocking function for sync mode usage of USBD APIs

@kdrv_usbd_get_event() when awake from osThreadFlagsWait( ) due to USBD notification,
users can use this function to retrieve which event is appearing and then take the corresponding action.
While performing transfers, user can also get notified through this call such as bulk-in notification or transfer complete notifications.

Parameters

in	*uevent*	usbd event to be notified, see kdrv_usbd_event_t

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

4.23.4.10 kdrv_usbd_initialize()

```
kdrv_status_t kdrv_usbd_initialize (
            void  )
```

USB device mode driver initialization.

This API should be the first call for USBD driver initialization and to invoke the driver thread.

Returns

kdrv_status_t see kdrv_status_t

4.23.4.11 kdrv_usbd_interrupt_receive()

```
kdrv_status_t kdrv_usbd_interrupt_receive (
            uint32_t endpoint,
            uint32_t * buf,
            uint32_t * rxLen,
            uint32_t timeout_ms )
```

Interrupt-OUT transfer in blocking mode.

Parameters

in	*endpoint*	a interrupt-out endpoint address, should be the value from bEndpointAddress
out	*buf*	buffer for receiving data
in,out	*rxLen*	buffer length for input, actual transfered length for output, should be less than MaxPacketSize
in	*timeout_ms*	timeout in millisecond

Returns

kdrv_status_t see kdrv_status_t

4.23.4.12 kdrv_usbd_interrupt_send()

```
kdrv_status_t kdrv_usbd_interrupt_send (
            uint32_t endpoint,
```

Kneron Confidential

```
            uint32_t * buf,
            uint32_t txLen,
            uint32_t timeout_ms )
```

Interrupt-IN transfer in blocking mode.

Immediately write data to the FIFO buffer for periodic interrupt-in transfer. Note even while the old data is not yet read by host, this function will overwrite it.

Parameters

in	*endpoint*	a interrupt-in endpoint address, should be the value from bEndpointAddress
in	*buf*	data to be sent to host
in	*txLen*	transfer length, shoudl be less then MaxPacketSize
in	*timeout_ms*	timeout in millisecond

Returns

kdrv_status_t see kdrv_status_t

4.23.4.13 kdrv_usbd_is_dev_configured()

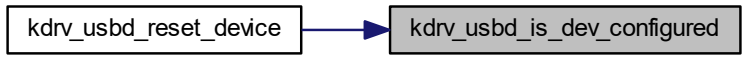
```
bool kdrv_usbd_is_dev_configured (
            void  )
```

check if device is enumerated and configured by a host

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:



4.23.4.14 kdrv_usbd_register_thread_notification()

```
kdrv_status_t kdrv_usbd_register_thread_notification (
            osThreadId_t tid,
            uint32_t tflag )
```

register user thread ID and thread flag for notifications including events or transfer completion/errors

Kneron Confidential

Parameters

in	*tid*	CMSIS-RTOS v2 thread ID
in	*tflag*	user defined thread flag to be notified by osThreadFlagsSet()

Returns

kdrv_status_t see kdrv_status_t

4.23.4.15 kdrv_usbd_reset_device()

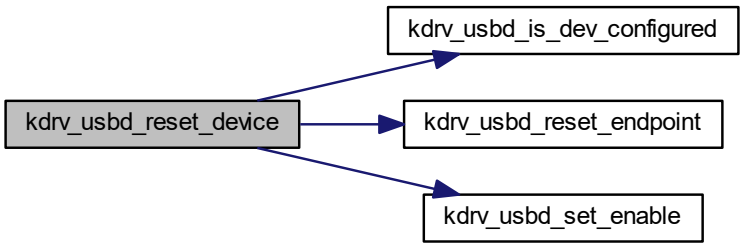
```
kdrv_status_t kdrv_usbd_reset_device (
            void  )
```

reset device and then it can be re-enumerated by host

Returns

kdrv_status_t see kdrv_status_t

Here is the call graph for this function:



Kneron confidential

4.23.4.16 kdrv_usbd_reset_endpoint()

```
kdrv_status_t kdrv_usbd_reset_endpoint (
            uint32_t endpoint )
```

reset specified endpoint

Parameters

in	*status*	status

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.23.4.17 kdrv_usbd_set_device_descriptor()

```
kdrv_status_t kdrv_usbd_set_device_descriptor (
            kdrv_usbd_speed_t speed,
            kdrv_usbd_device_descriptor_t * dev_desc )
```

configure device descriptor including configuration, interface and all endpoints descriptors

USBD driver API provides specific data structs for these descriptors, users must statically declare instances of these descriptors in memory which will be used when being enumerated by a USB host.
At present some limitations should be noted:

1. Support only one configuration descriptor, one interface descriptor and 4 endpoint descriptors.
2. Isochronous transfer is not supported yet.
3. If enabling log message through USB then one endpoint must be reserved for USBD internal use.

Parameters

in	*speed*	speed want to run, now support only High-Speed
in	*dev_desc*	user crated device descriptor, this must be kept during device enumeration

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

4.23.4.18 kdrv_usbd_set_device_qualifier_descriptor()

```
kdrv_status_t kdrv_usbd_set_device_qualifier_descriptor (
            kdrv_usbd_speed_t speed,
            kdrv_usbd_device_qualifier_descriptor_t * dev_qual_desc )
```

configure device qualifier descriptor, this is optional

This API is to set other speed when acting in high-speed, users can set a meaningful content in this descriptor.

Parameters

in	*speed*	speed want to run, now support only High-Speed
in	*dev_qual_desc*	user crated device qualifier descriptor, this must be kept during device enumeration

Returns

kdrv_status_t see kdrv_status_t

4.23.4.19 kdrv_usbd_set_enable()

```
kdrv_status_t kdrv_usbd_set_enable (
            bool enable )
```

set enable/disabale of USB device mode, host can enumerate this device only if device is enabled

Once above calls are done properly, users can invoke this function to enable the device and after that it can start to be seen and be enumerated by a USB host.
Once device is enabled and enumerated by a host, some USB events may start appearing,
user must start to wait for a specified thread flag to be notified of USB events through the osThreadFlagsWait(), events will be introduced in next section.

Parameters

in	*enable*	true to enable, false to disable

Kneron Confidential

Returns

kdrv_status_t see kdrv_status_t

Here is the caller graph for this function:

4.23.4.20 kdrv_usbd_uninitialize()

```
kdrv_status_t kdrv_usbd_uninitialize (
            void  )
```

USB device mode driver uninitialization.

Returns

kdrv_status_t see kdrv_status_t

4.23.5 Variable Documentation

4.23.5.1 "@5

```
union { ...  }
```

4.23.5.2 "@7

```
struct { ...  }
```

4.23.5.3 data1 [1/2]

```
uint32_t { ...  } ::data1
```

Kneron Confidential

4.23.5.4 data1 [2/2]

```
uint32_t kdrv_usbd_event_t::data1
```

4.23.5.5 data2 [1/2]

```
uint32_t { ... } ::data2
```

4.23.5.6 data2 [2/2]

```
uint32_t kdrv_usbd_event_t::data2
```

4.23.5.7 ename

```
kdrv_usbd_event_name_t kdrv_usbd_event_t::ename
```

see kdrv_usbd_event_name_t

4.23.5.8 kdrv_usbd_config_descriptor_t

```
kdrv_usbd_config_descriptor_t
```

4.23.5.9 kdrv_usbd_device_descriptor_t

```
kdrv_usbd_device_descriptor_t
```

4.23.5.10 kdrv_usbd_device_qualifier_descriptor_t

```
kdrv_usbd_device_qualifier_descriptor_t
```

4.23.5.11 kdrv_usbd_endpoint_descriptor_t

```
kdrv_usbd_endpoint_descriptor_t
```

Kneron Confidential

4.23.5.12 kdrv_usbd_interface_descriptor_t

```
kdrv_usbd_interface_descriptor_t
```

4.23.5.13 kdrv_usbd_setup_packet_t

```
kdrv_usbd_setup_packet_t
```

4.23.5.14 setup [1/2]

```
kdrv_usbd_setup_packet_t kdrv_usbd_event_t::setup
```

see kdrv_usbd_setup_packet_t

4.23.5.15 setup [2/2]

```
kdrv_usbd_setup_packet_t { ... } ::setup
```

see kdrv_usbd_setup_packet_t

Kneron Confidential

4.24 KDRV_WDT

Kneron WDT device mode driver.

Macros

- #define KDRV_WDT_BASE WDT_FTWDT010_PA_BASE
- #define REG_WDT_CNT (KDRV_WDT_BASE + 0x00) /* wdt timer counter */
- #define REG_WDT_LOAD (KDRV_WDT_BASE + 0x04) /* auto reload register */
- #define REG_WDT_RST (KDRV_WDT_BASE + 0x08) /* restart register */
- #define REG_WDT_CR (KDRV_WDT_BASE + 0x0C) /* control register */
- #define REG_WDT_STS (KDRV_WDT_BASE + 0x10) /* wdt status register */
- #define REG_WDT_CLR (KDRV_WDT_BASE + 0x14) /* wdt time cleared register */
- #define REG_WDT_INTR_LEN (KDRV_WDT_BASE + 0x18) /* wdt intr length register */
- #define REG_WDT_REV (KDRV_WDT_BASE + 0x1C) /* wdt revision */
- #define WDT_CR_EN BIT(0) /* WDT enable bit, 0: disable, 1: enable */
- #define WDT_CR_RST_EN BIT(1) /* WDT reset bit, 0: disable, 1: enable */
- #define WDT_CR_INT_EN BIT(2) /* WDT int enable bit, 0: disable, 1: enable */
- #define WDT_CR_EXT_EN BIT(3) /* WDT extclk enable bit, 0:disable, 1:enable */
- #define WDT_CR_EXTCLK BIT(4) /* WDT clock source bit, 0:PCLK, 1:EXTCLK */
- #define WDT_RST_AUTO_RELOAD_KEY 0x5AB9

Functions

- void kdrv_wdt_enable (void)

  watchdog enable
- void kdrv_wdt_disable (void)

  watchdog disable
- void kdrv_wdt_reset (void)

  watchdog reset
- void kdrv_wdt_set_auto_reload (uint32_t value)

  watchdog reload
- void kdrv_wdt_sys_int_enable (void)

  watchdog interrupt enable
- void kdrv_wdt_sys_int_disable (void)

  watchdog interrupt disable
- void kdrv_wdt_sys_reset_enable (void)

  watchdog reset enable
- void kdrv_wdt_sys_reset_disable (void)

  watchdog reset disable
- uint32_t kdrv_wdt_read_counter (void)

  watchdog read counter
- void kdrv_wdt_set_clear_status (void)

  watchdog status clear
- void kdrv_wdt_set_int_counter (uint8_t counter)

  watchdog set interrupt counter
- bool kdrv_wdt_is_counter_zero (void)

  watchdog, is counter zero

Kneron Confidential

4.24.1 Detailed Description

Kneron WDT device mode driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.24.2 Macro Definition Documentation

4.24.2.1 KDRV_WDT_BASE

```
#define KDRV_WDT_BASE WDT_FTWDT010_PA_BASE
```

4.24.2.2 REG_WDT_CLR

```
#define REG_WDT_CLR (KDRV_WDT_BASE + 0x14) /* wdt time cleared register */
```

4.24.2.3 REG_WDT_CNT

```
#define REG_WDT_CNT (KDRV_WDT_BASE + 0x00) /* wdt timer counter */
```

4.24.2.4 REG_WDT_CR

```
#define REG_WDT_CR (KDRV_WDT_BASE + 0x0C) /* control register */
```

4.24.2.5 REG_WDT_INTR_LEN

```
#define REG_WDT_INTR_LEN (KDRV_WDT_BASE + 0x18) /* wdt intr length register */
```

4.24.2.6 REG_WDT_LOAD

```
#define REG_WDT_LOAD (KDRV_WDT_BASE + 0x04) /* auto reload register */
```

Kneron Confidential

4.24.2.7 REG_WDT_REV

```
#define REG_WDT_REV (KDRV_WDT_BASE + 0x1C) /* wdt revision */
```

4.24.2.8 REG_WDT_RST

```
#define REG_WDT_RST (KDRV_WDT_BASE + 0x08) /* restart register */
```

4.24.2.9 REG_WDT_STS

```
#define REG_WDT_STS (KDRV_WDT_BASE + 0x10) /* wdt status register */
```

4.24.2.10 WDT_CR_EN

```
#define WDT_CR_EN BIT(0) /* WDT enable bit, 0:  disable, 1:  enable */
```

4.24.2.11 WDT_CR_EXT_EN

```
#define WDT_CR_EXT_EN BIT(3) /* WDT extclk enable bit, 0:disable, 1:enable */
```

4.24.2.12 WDT_CR_EXTCLK

```
#define WDT_CR_EXTCLK BIT(4) /* WDT clock source bit, 0:PCLK, 1:EXTCLK */
```

4.24.2.13 WDT_CR_INT_EN

```
#define WDT_CR_INT_EN BIT(2) /* WDT int enable bit, 0:  disable, 1:  enable */
```

4.24.2.14 WDT_CR_RST_EN

```
#define WDT_CR_RST_EN BIT(1) /* WDT reset bit, 0:  disable, 1:  enable */
```

Kneron Confidential

4.24.2.15 WDT_RST_AUTO_RELOAD_KEY

```
#define WDT_RST_AUTO_RELOAD_KEY 0x5AB9
```

4.24.3 Function Documentation

4.24.3.1 kdrv_wdt_disable()

```
void kdrv_wdt_disable (
            void  )
```

watchdog disable

Returns

N/A

4.24.3.2 kdrv_wdt_enable()

```
void kdrv_wdt_enable (
            void  )
```

watchdog enable

Returns

N/A

4.24.3.3 kdrv_wdt_is_counter_zero()

```
bool kdrv_wdt_is_counter_zero (
            void  )
```

watchdog, is counter zero

Returns

bool

Kneron Confidential

4.24.3.4 kdrv_wdt_read_counter()

```
uint32_t kdrv_wdt_read_counter (
            void  )
```

watchdog read counter

Returns

counter value

4.24.3.5 kdrv_wdt_reset()

```
void kdrv_wdt_reset (
            void  )
```

watchdog reset

Returns

N/A

4.24.3.6 kdrv_wdt_set_auto_reload()

```
void kdrv_wdt_set_auto_reload (
            uint32_t value )
```

watchdog reload

Parameters

in	*value*	watchdog reload value

Returns

N/A

4.24.3.7 kdrv_wdt_set_clear_status()

```
void kdrv_wdt_set_clear_status (
            void  )
```

watchdog status clear

Kneron Confidential

Returns

N/A

4.24.3.8 kdrv_wdt_set_int_counter()

```
void kdrv_wdt_set_int_counter (
            uint8_t counter )
```

watchdog set interrupt counter

Parameters

[in[	counter set the duration of assertion of wd_intr, the default value is 0xFF. which means that the default assertion duration is 256 clock cycles(PCLK)

Returns

N/A

4.24.3.9 kdrv_wdt_sys_int_disable()

```
void kdrv_wdt_sys_int_disable (
            void  )
```

watchdog interrupt disable

Returns

N/A

4.24.3.10 kdrv_wdt_sys_int_enable()

```
void kdrv_wdt_sys_int_enable (
            void  )
```

watchdog interrupt enable

Returns

N/A

Kneron Confidential

4.24.3.11 kdrv_wdt_sys_reset_disable()

```
void kdrv_wdt_sys_reset_disable (
            void  )
```

watchdog reset disable

Returns

N/A

4.24.3.12 kdrv_wdt_sys_reset_enable()

```
void kdrv_wdt_sys_reset_enable (
            void  )
```

watchdog reset enable

Returns

N/A

Kneron Confidential

4.25 KDEV_FLASH

Kneron flash device.

Data Structures

- struct kdev_flash_sector_t

  Flash Sector index struct.
- struct kdev_flash_info_t

  Flash information struct.
- struct kdev_flash_status_t

  Flash Status struct.

Macros

- #define SPI020_SECTOR_SIZE 4096
- #define SPI020_BLOCK_64SIZE 65536

Functions

- uint32_t kdev_flash_probe (spi_flash_t *flash)
- kdev_status_t kdev_flash_initialize (void)

  Initialize spi flash interface include hardware setting, get flash information and set to 4byte address if flash size is bigger than 16Mbytes.
- kdev_status_t kdev_flash_uninitialize (void)

  Uinitialize the spi flash interface.
- kdev_status_t kdev_flash_power_control (ARM_POWER_STATE state)

  Power handling for spi flasg.
- kdev_status_t kdev_flash_readdata (uint32_t addr, void *data, uint32_t cnt)

  Read data from specific index of spi flash.
- kdev_status_t kdev_flash_programdata (uint32_t addr, const void *data, uint32_t cnt)

  Program data to specific index in spi flash.
- kdev_status_t kdev_flash_erase_sector (uint32_t addr)

  Erase Flash by Sector(4k bytes).
- kdev_status_t kdev_flash_erase_multi_sector (uint16_t start_addr, uint16_t end_addr)

  Erase multiple Flash Sectors(continuously).
- kdev_status_t kdev_flash_erase_chip (void)

  Erase whole Flash at once. Optional function for faster full chip erase.
- kdev_flash_status_t kdev_flash_get_status (void)

  Get Flash status.
- kdev_flash_info_t * kdev_flash_get_info (void)

  Get Flash information.

4.25.1 Detailed Description

Kneron flash device.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.25.2 Macro Definition Documentation

4.25.2.1 SPI020_BLOCK_64SIZE

```
#define SPI020_BLOCK_64SIZE 65536
```

4.25.2.2 SPI020_SECTOR_SIZE

```
#define SPI020_SECTOR_SIZE 4096
```

4.25.3 Function Documentation

4.25.3.1 kdev_flash_erase_chip()

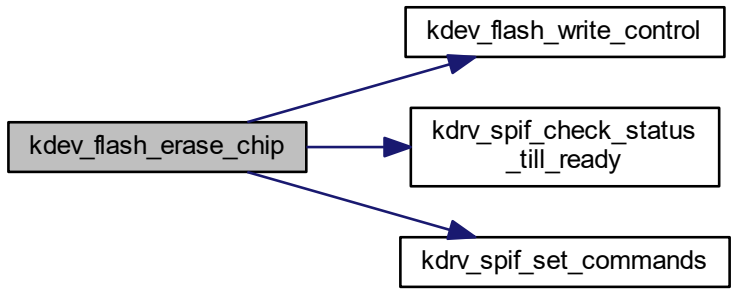
```
kdev_status_t kdev_flash_erase_chip (
            void  )
```

Erase whole Flash at once. Optional function for faster full chip erase.

Returns

kdev_status_t

Here is the call graph for this function:



Kneron Confidential

4.25.3.2 kdev_flash_erase_multi_sector()

```
kdev_status_t kdev_flash_erase_multi_sector (
            uint16_t start_addr,
            uint16_t end_addr )
```

Erase multiple Flash Sectors(continuously).

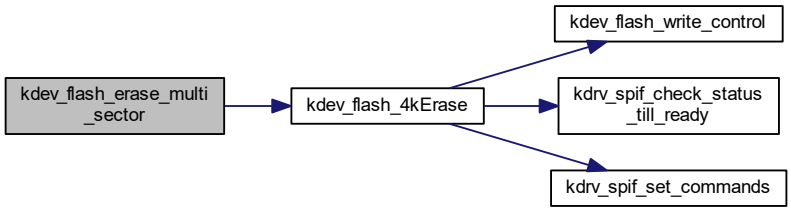
Parameters

in	*addr*	Sector start address
in	*addr*	Sector end address

Returns

kdev_status_t

Here is the call graph for this function:



4.25.3.3 kdev_flash_erase_sector()

```
kdev_status_t kdev_flash_erase_sector (
            uint32_t addr )
```

Erase Flash by Sector(4k bytes).

Parameters

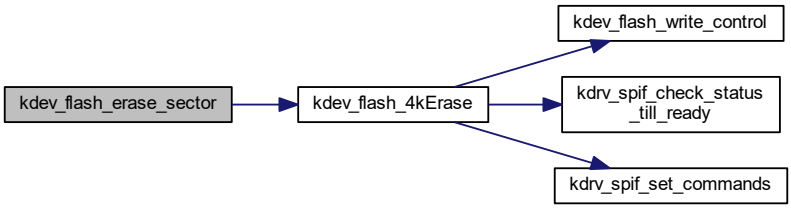
in	*addr*	Sector address

Kneron Confidential

Returns

kdev_status_t

Here is the call graph for this function:



4.25.3.4 kdev_flash_get_info()

```
kdev_flash_info_t* kdev_flash_get_info (
            void  )
```

Get Flash information.

Returns

Pointer to Flash information kdev_flash_info_t

4.25.3.5 kdev_flash_get_status()

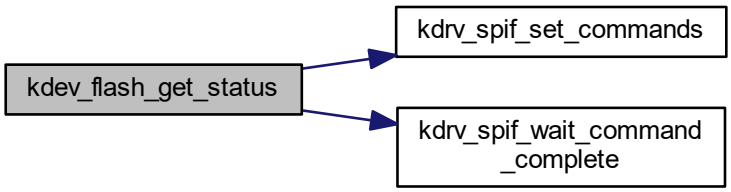
```
kdev_flash_status_t kdev_flash_get_status (
            void  )
```

Get Flash status.

Returns

Flash status kdev_flash_status_t

Here is the call graph for this function:



Kneron Confidential

4.25.3.6 kdev_flash_initialize()

```
kdev_status_t kdev_flash_initialize (
            void  )
```

Initialize spi flash interface include hardware setting, get flash information and set to 4byte address if flash size is bigger than 16Mbytes.

Parameters

in	*N/A*	

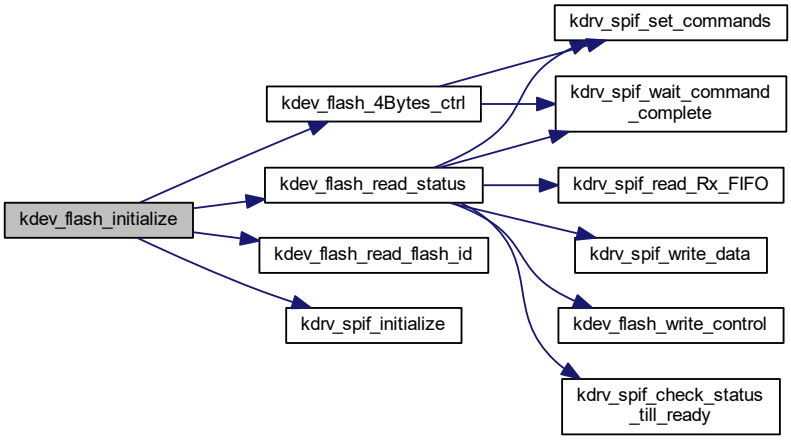
Returns

kdrv_status_t

Note

This API MUST be called before using the Read/write APIs for spi flash.

Here is the call graph for this function:



4.25.3.7 kdev_flash_power_control()

```
kdev_status_t kdev_flash_power_control (
            ARM_POWER_STATE state )
```

Power handling for spi flasg.

Parameters

in	*state*	Power state

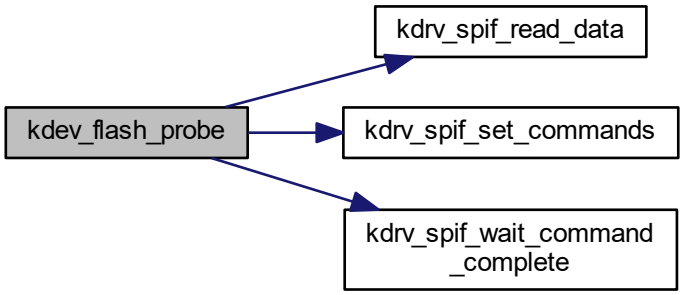
Returns

kdev_status_t

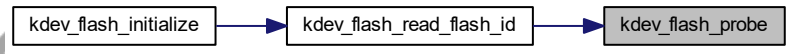
4.25.3.8 kdev_flash_probe()

```
uint32_t kdev_flash_probe (
            spi_flash_t * flash )
```

Here is the call graph for this function:



Here is the caller graph for this function:



4.25.3.9 kdev_flash_programdata()

```
kdev_status_t kdev_flash_programdata (
            uint32_t addr,
            const void * data,
            uint32_t cnt )
```

Program data to specific index in spi flash.

Kneron Confidential

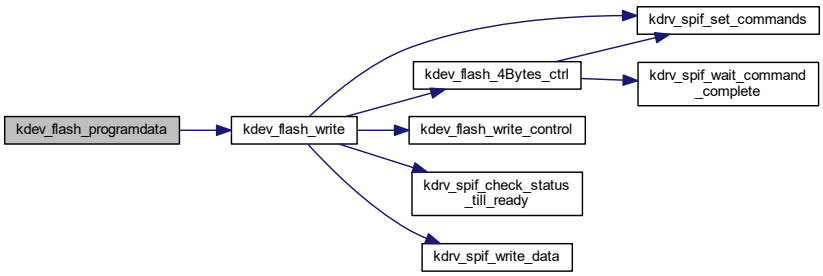
Parameters

in	*addr*	Data address.
in	*data*	Pointer to a buffer containing the data to be programmed to Flash.
in	*cnt*	Number of data items to program.

Returns

number of data items programmed or kdev_status_t

Here is the call graph for this function:



4.25.3.10 kdev_flash_readdata()

```
kdev_status_t kdev_flash_readdata (
            uint32_t addr,
            void * data,
            uint32_t cnt )
```

Read data from specific index of spi flash.

Parameters

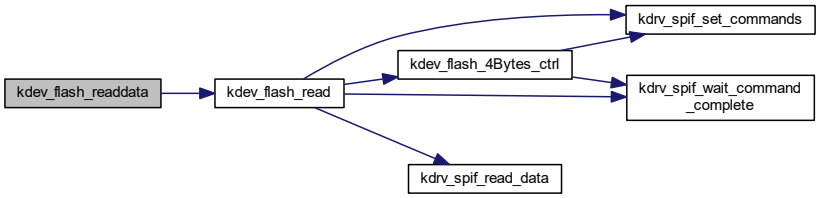
in	*addr*	Data address.
out	*data*	Pointer to a buffer storing the data read from Flash.
in	*cnt*	Number of data items to read.

Kneron Confidential

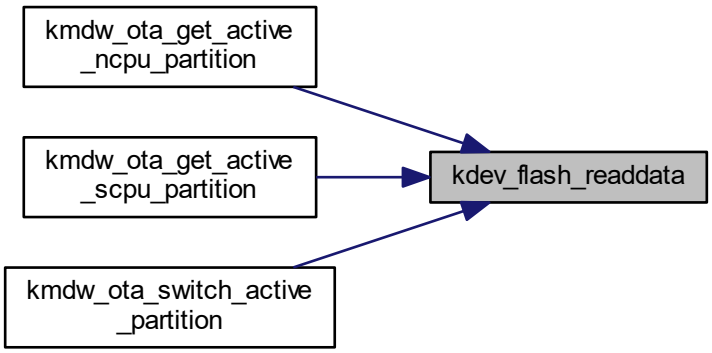
Returns

number of data items read or kdev_status_t

Here is the call graph for this function:



Here is the caller graph for this function:



Kneron Confidential

4.25.3.11 kdev_flash_uninitialize()

```
kdev_status_t kdev_flash_uninitialize (
            void  )
```

Uinitialize the spi flash interface.

Returns

kdrv_status_t

4.26 KDEV_PANEL

Kneron panel device interface for MZT_480x272 and ST778_240x320 driver.

Functions

- kdev_status_t kdev_panel_initialize (kdrv_display_t *display_drv)
  Initializes kdev panel driver.
- kdev_status_t kdev_panel_clear (kdrv_display_t *display_drv, u32 color)
- uint16_t kdev_panel_read_display_id (kdrv_display_t *display_drv)
- kdev_status_t kdev_panel_refresh (kdrv_display_t *display_drv)

4.26.1 Detailed Description

Kneron panel device interface for MZT_480x272 and ST778_240x320 driver.

Copyright

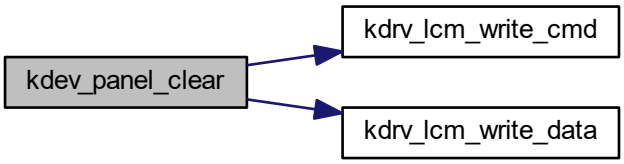
Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.26.2 Function Documentation

4.26.2.1 kdev_panel_clear()

```
kdev_status_t kdev_panel_clear (
            kdrv_display_t * display_drv,
            u32 color )
```

Here is the call graph for this function:



4.26.2.2 kdev_panel_initialize()

```
kdev_status_t kdev_panel_initialize (
            kdrv_display_t * display_drv )
```

Initializes kdev panel driver.

Kneron Confidential

Parameters

in	*display_drv*	see kdrv_display_t

Returns

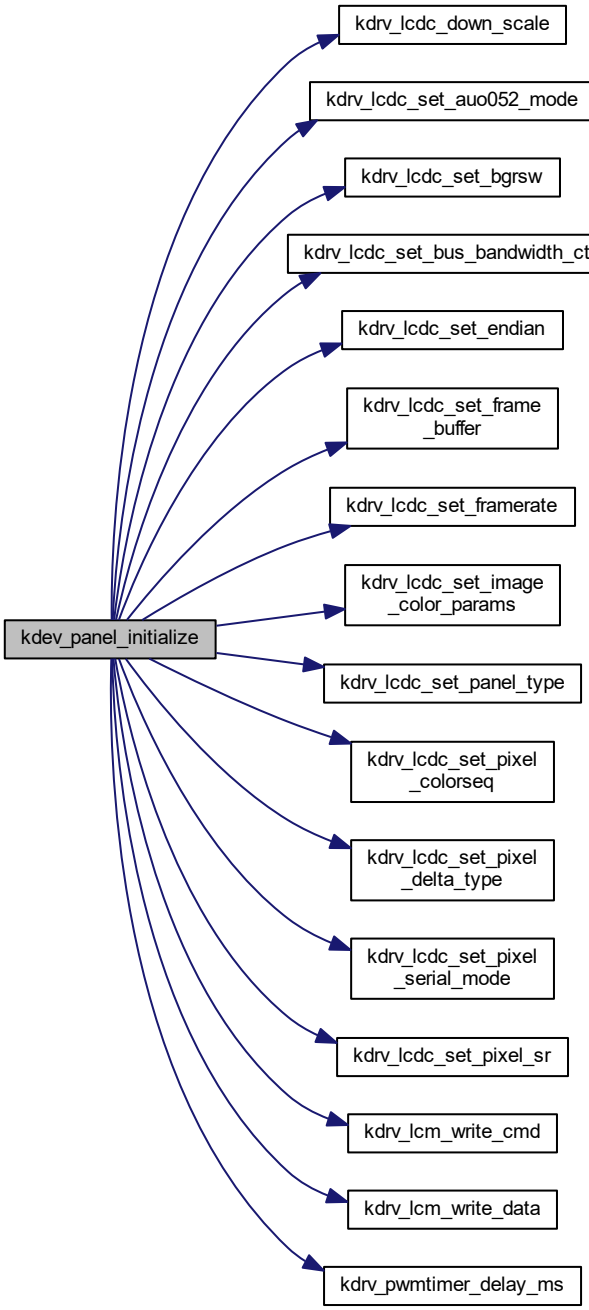
kdrv_status_t see kdrv_status_t

Note

This API MUST be called before using the Read/write APIs for I2C.

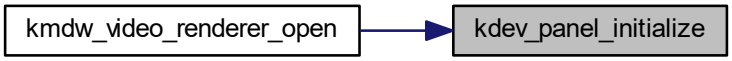
Kneron Confidential

Here is the call graph for this function:



Kneron Confidential

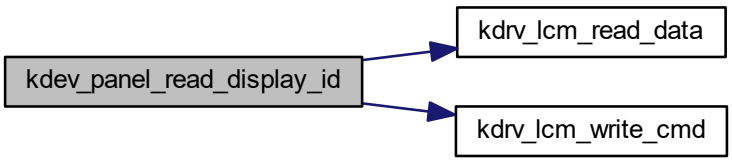
Here is the caller graph for this function:



4.26.2.3 kdev_panel_read_display_id()

```
uint16_t kdev_panel_read_display_id (
            kdrv_display_t * display_drv )
```

Here is the call graph for this function:



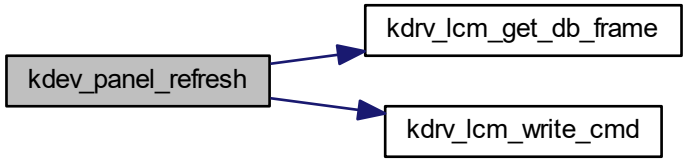
Here is the caller graph for this function:

Kneron Confidential

4.26.2.4 kdev_panel_refresh()

```
kdev_status_t kdev_panel_refresh (
            kdrv_display_t * display_drv )
```

Here is the call graph for this function:



Here is the caller graph for this function:

Kneron Confidential

4.27 KDEV_SENSOR

Kneron sensor device interface.

Data Structures

- struct sensor_ops

4.27.1 Detailed Description

Kneron sensor device interface.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

Kneron Confidential

4.28 KMDW_CAMERA

Kneron camera middleware for camera driver.

Data Structures

- struct cam_format
- struct cam_capability
- struct cam_sensor_aec
- struct cam_ops

Macros

- #define CAP_VIDEO_CAPTURE 0x00000001 /∗ Is a video capture device ∗/
- #define CAP_STREAMING 0x00000002 /∗ can stream on/off ∗/
- #define CAP_DEVICE_CAPS 0x00000004 /∗ can query capabilities ∗/

Enumerations

- enum {
  CID_SCANNING_MODE = 0x1, CID_AUTO_EXPOSURE_MODE, CID_AUTO_EXPOSURE_PRIORITY,
  CID_EXPOSURE_TIME_ABSOLUTE,
  CID_EXPOSURE_TIME_RELATIVE, CID_FOCUS_ABSOLUTE, CID_FOCUS_RELATIVE, CID_IRIS_ABSOLUTE,
  CID_IRIS_RELATIVE, CID_ZOOM_ABSOLUTE, CID_ZOOM_RELATIVE, CID_PANTILT_ABSOLUTE,
  CID_PANTILT_RELATIVE, CID_ROLL_ABSOLUTE, CID_ROLL_RELATIVE, CID_FOCUS_AUTO,
  CID_PRIVACY, CID_FOCUS_SIMPLE, CID_DIGITAL_WINDOW, CID_REGION_OF_INTEREST,
  CID_BRIGHTNESS, CID_CONTRAST, CID_HUE, CID_SATURATION,
  CID_SHARPNESS, CID_GAMMA, CID_WHITE_BALANCE_TEMPERATURE, CID_WHITE_BALANCE_COMPONENT,
  CID_BACKLIGHT_COMPENSATION, CID_GAIN, CID_POWER_LINE_FREQUENCY, CID_HUE_AUTO,
  CID_WHITE_BALANCE_TEMPERATURE_AUTO, CID_WHITE_BALANCE_COMPONENT_AUTO, CID_DIGITAL_MULTIPLIE
  CID_DIGITAL_MULTIPLIER_LIMIT,
  CID_CONTRAST_AUTO, CID_LIST_ALL = 0xFF }
- enum { KDP_CAM_0, KDP_CAM_1, KDP_CAM_NUM }
- enum camera_state {
  CAMERA_STATE_IDLE = 0, CAMERA_STATE_INITED, CAMERA_STATE_RUNNING, CAMERA_STATE_IN_FDR_INFERE
  CAMERA_STATE_IN_FDR_REGISTRATION, CAMERA_STATE_IN_FDR_AUTO_REGISTRATION,
  CAMERA_STATE_IN_FDR_REGISTRATION_CONFIRM, CAMERA_STATE_IN_FDR_BOTH_REGISTRATION,
  CAMERA_STATE_IN_FDR_BOTH_REGISTRATION_CONFIRM, CAMERA_STATE_IN_FDR_BOTH_INFERENCE,
  CAMERA_STATE_IN_FDR_REGISTRATION_POSE_JUSTIFY }

Functions

- kmdw_status_t kmdw_camera_init (void)

  Initializes camera setting.
- kmdw_status_t kmdw_camera_open (uint32_t cam_idx)

  camera open function
- kmdw_status_t kmdw_camera_close (uint32_t cam_idx)

  camera close function
- kmdw_status_t kmdw_camera_get_device_info (uint32_t cam_idx, struct cam_capability ∗cap)

  camera get device information function

Kneron Confidential

- kmdw_status_t kmdw_camera_set_frame_format (uint32_t cam_idx, struct cam_format *format)

  camera set frame format function
- kmdw_status_t kmdw_camera_get_frame_format (uint32_t cam_idx, struct cam_format *format)

  camera get frame format function
- kmdw_status_t kmdw_camera_buffer_init (uint32_t cam_idx)

  camera buffer init function
- kmdw_status_t kmdw_camera_start (uint32_t cam_idx)

  camera start function
- kmdw_status_t kmdw_camera_stop (uint32_t cam_idx)

  camera stop function
- kmdw_status_t kmdw_camera_buffer_prepare (uint32_t cam_idx)

  camera buffer prepare function
- kmdw_status_t kmdw_camera_buffer_capture (uint32_t cam_idx, uint32_t *addr, uint32_t *size)

  camera buffer capture function
- kmdw_status_t kmdw_camera_stream_on (uint32_t cam_idx)

  camera streaming on function
- kmdw_status_t kmdw_camera_stream_off (uint32_t cam_idx)

  camera streaming off function
- kmdw_status_t kmdw_camera_set_gain (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  camera set gain function
- kmdw_status_t kmdw_camera_set_aec (uint32_t cam_idx, struct cam_sensor_aec *aec_p)

  camera set ae controller ROI area function
- kmdw_status_t kmdw_camera_set_exp_time (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  camera set exposure time function
- kmdw_status_t kmdw_camera_get_lux (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain, uint8_t *global_gain, uint8_t *y_average)

  camera get lum and other parameter function
- kmdw_status_t kmdw_camera_led_switch (uint32_t cam_idx, uint32_t on)

  camera set nir led on/off function
- kmdw_status_t kmdw_camera_ioctl (uint32_t cam_idx, uint32_t cid, void *data, uint16_t len)

  register specific cam ops with cam_idx
- kmdw_status_t kmdw_camera_controller_register (uint32_t cam_idx, struct cam_ops *cam_ops_p)
- kmdw_status_t kmdw_camera_controller_unregister (uint32_t cam_idx, struct cam_ops *cam_ops_p)

  unregister specific cam ops with cam_idx

Variables

- uint32_t cam_format::width
- uint32_t cam_format::height
- uint32_t cam_format::pixelformat
- uint32_t cam_format::field
- uint32_t cam_format::bytesperline
- uint32_t cam_format::sizeimage
- uint32_t cam_format::colorspace
- char cam_capability::driver [16]
- char cam_capability::desc [16]
- uint32_t cam_capability::version
- uint32_t cam_capability::capabilities
- uint8_t cam_sensor_aec::x1
- uint8_t cam_sensor_aec::x2
- uint8_t cam_sensor_aec::y1

Kneron Confidential

- uint8_t cam_sensor_aec::y2
- uint8_t cam_sensor_aec::center_x1
- uint8_t cam_sensor_aec::center_x2
- uint8_t cam_sensor_aec::center_y1
- uint8_t cam_sensor_aec::center_y2
- kmdw_status_t(∗ cam_ops::open)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::close)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::set_format)(uint32_t cam_idx, struct cam_format ∗format)
- kmdw_status_t(∗ cam_ops::get_format)(uint32_t cam_idx, struct cam_format ∗format)
- kmdw_status_t(∗ cam_ops::buffer_init)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::start_capture)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::stop_capture)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::buffer_prepare)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::buffer_capture)(uint32_t cam_idx, uint32_t ∗addr, uint32_t ∗size)
- kmdw_status_t(∗ cam_ops::stream_on)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::stream_off)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::query_capability)(uint32_t cam_idx, struct cam_capability ∗cap)
- kmdw_status_t(∗ cam_ops::set_gain)(uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
- kmdw_status_t(∗ cam_ops::set_aec)(uint32_t cam_idx, struct cam_sensor_aec ∗aec_p)
- kmdw_status_t(∗ cam_ops::set_exp_time)(uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
- kmdw_status_t(∗ cam_ops::get_lux)(uint32_t cam_idx, uint16_t ∗expo, uint8_t ∗pre_gain, uint8_t ∗post_↩ gain, uint8_t ∗global_gain, uint8_t ∗y_average)
- kmdw_status_t(∗ cam_ops::led_switch)(uint32_t cam_idx, uint32_t on)
- kmdw_status_t(∗ cam_ops::set_mirror)(uint32_t cam_idx, uint32_t enable)
- kmdw_status_t(∗ cam_ops::set_flip)(uint32_t cam_idx, uint32_t enable)
- uint32_t(∗ cam_ops::get_device_id)(uint32_t cam_idx)
- kmdw_status_t(∗ cam_ops::ioctl)(uint32_t cam_idx, uint32_t cid, void ∗data, uint16_t len)

4.28.1 Detailed Description

Kneron camera middleware for camera driver.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.28.2 Macro Definition Documentation

4.28.2.1 CAP_DEVICE_CAPS

```
#define CAP_DEVICE_CAPS 0x00000004 /* can query capabilities */
```

4.28.2.2 CAP_STREAMING

```
#define CAP_STREAMING 0x00000002 /* can stream on/off */
```

Kneron Confidential

4.28.2.3 CAP_VIDEO_CAPTURE

```
#define CAP_VIDEO_CAPTURE 0x00000001 /* Is a video capture device */
```

4.28.3 Enumeration Type Documentation

4.28.3.1 anonymous enum

```
anonymous enum
```

Enumerator

CID_SCANNING_MODE	
CID_AUTO_EXPOSURE_MODE	
CID_AUTO_EXPOSURE_PRIORITY	
CID_EXPOSURE_TIME_ABSOLUTE	
CID_EXPOSURE_TIME_RELATIVE	
CID_FOCUS_ABSOLUTE	
CID_FOCUS_RELATIVE	
CID_IRIS_ABSOLUTE	
CID_IRIS_RELATIVE	
CID_ZOOM_ABSOLUTE	
CID_ZOOM_RELATIVE	
CID_PANTILT_ABSOLUTE	
CID_PANTILT_RELATIVE	
CID_ROLL_ABSOLUTE	
CID_ROLL_RELATIVE	
CID_FOCUS_AUTO	
CID_PRIVACY	
CID_FOCUS_SIMPLE	
CID_DIGITAL_WINDOW	
CID_REGION_OF_INTEREST	
CID_BRIGHTNESS	
CID_CONTRAST	
CID_HUE	
CID_SATURATION	
CID_SHARPNESS	
CID_GAMMA	
CID_WHITE_BALANCE_TEMPERATURE	
CID_WHITE_BALANCE_COMPONENT	
CID_BACKLIGHT_COMPENSATION	
CID_GAIN	
CID_POWER_LINE_FREQUENCY	
CID_HUE_AUTO	
CID_WHITE_BALANCE_TEMPERATURE_AUTO	
CID_WHITE_BALANCE_COMPONENT_AUTO	
CID_DIGITAL_MULTIPLIER	
CID_DIGITAL_MULTIPLIER_LIMIT	
CID_CONTRAST_AUTO	
CID_LIST_ALL	

Kneron Confidential

4.28.3.2 anonymous enum

```
anonymous enum
```

Enumerator

KDP_CAM_0	
KDP_CAM_1	
KDP_CAM_NUM	

4.28.3.3 camera_state

```
enum camera_state
```

Enumerator

CAMERA_STATE_IDLE	
CAMERA_STATE_INITED	
CAMERA_STATE_RUNNING	
CAMERA_STATE_IN_FDR_INFERENCE	
CAMERA_STATE_IN_FDR_REGISTRATION	
CAMERA_STATE_IN_FDR_AUTO_REGISTRATION	
CAMERA_STATE_IN_FDR_REGISTRATION_CONFIRM	
CAMERA_STATE_IN_FDR_BOTH_REGISTRATION	
CAMERA_STATE_IN_FDR_BOTH_REGISTRATION_CONFIRM	
CAMERA_STATE_IN_FDR_BOTH_INFERENCE	
CAMERA_STATE_IN_FDR_REGISTRATION_POSE_JUSTIFY	

4.28.4 Function Documentation

4.28.4.1 kmdw_camera_buffer_capture()

```
kmdw_status_t kmdw_camera_buffer_capture (
            uint32_t cam_idx,
            uint32_t * addr,
            uint32_t * size )
```

camera buffer capture function

Kneron Confidential

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.2 kmdw_camera_buffer_init()

```
kmdw_status_t kmdw_camera_buffer_init (
            uint32_t cam_idx )
```

camera buffer init function

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.3 kmdw_camera_buffer_prepare()

```
kmdw_status_t kmdw_camera_buffer_prepare (
            uint32_t cam_idx )
```

camera buffer prepare function

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.4 kmdw_camera_close()

```
kmdw_status_t kmdw_camera_close (
            uint32_t cam_idx )
```

camera close function

Kneron Confidential

Parameters

in	*cam_idx*	camera id

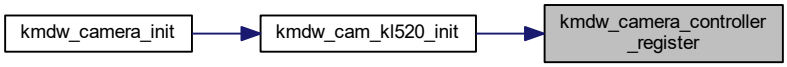
Returns

kmdw_status_t see kmdw_status_t

4.28.4.5 kmdw_camera_controller_register()

```
kmdw_status_t kmdw_camera_controller_register (
            uint32_t cam_idx,
            struct cam_ops * cam_ops_p )
```

Here is the caller graph for this function:



4.28.4.6 kmdw_camera_controller_unregister()

```
kmdw_status_t kmdw_camera_controller_unregister (
            uint32_t cam_idx,
            struct cam_ops * cam_ops_p )
```

unregister specific cam ops with cam_idx

Parameters

in	*cam_idx*	camera id
in	*cam_ops↩ _p*	incidence for each cam_idx

Returns

kmdw_status_t see kmdw_status_t

Kneron Confidential

4.28.4.7 kmdw_camera_get_device_info()

```
kmdw_status_t kmdw_camera_get_device_info (
            uint32_t cam_idx,
            struct cam_capability * cap )
```

camera get device information function

Parameters

in	*cam_idx*	camera id
out	*cap*	point of camera capability information.

Returns

kmdw_status_t see kmdw_status_t

4.28.4.8 kmdw_camera_get_frame_format()

```
kmdw_status_t kmdw_camera_get_frame_format (
            uint32_t cam_idx,
            struct cam_format * format )
```

camera get frame format function

Parameters

in	*cam_idx*	camera id
out	*cap*	point of format information.

Returns

kmdw_status_t see kmdw_status_t

4.28.4.9 kmdw_camera_get_lux()

```
kmdw_status_t kmdw_camera_get_lux (
            uint32_t cam_idx,
            uint16_t * expo,
            uint8_t * pre_gain,
            uint8_t * post_gain,
            uint8_t * global_gain,
            uint8_t * y_average )
```

camera get lum and other parameter function

Kneron Confidential

Parameters

in	*cam_idx*	camera id
out	*expo*	exposure time parameter
out	*pre_gain*	exposure time parameter
out	*post_gain*	exposure time parameter
out	*global_gain*	exposure time parameter
in	*y_average*	exposure time parameter 2

Returns

kmdw_status_t see kmdw_status_t

4.28.4.10 kmdw_camera_init()

```
kmdw_status_t kmdw_camera_init (
            void  )
```

Initializes camera setting.

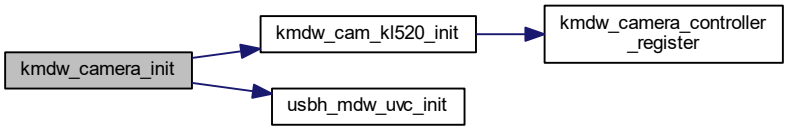
Parameters

in	*N/A*	

Returns

kmdw_status_t see kmdw_status_t

Here is the call graph for this function:



4.28.4.11 kmdw_camera_ioctl()

```
kmdw_status_t kmdw_camera_ioctl (
            uint32_t cam_idx,
```

Kneron Confidential

```
uint32_t cid,
void * data,
uint16_t len )
```

register specific cam ops with cam_idx

Kneron Confidential

Parameters

in	*cam_idx*	camera id
in	*cam_ops↩ _p*	incidence for each cam_idx

Returns

kmdw_status_t see kmdw_status_t

camera ioctl function

Parameters

in	*cam_idx*	camera id
in	*cid*	control command id
	[in|out]	∗data poniter to the parameter structure, a control command specific
in	*len*	structure length

Returns

kmdw_status_t see kmdw_status_t

4.28.4.12 kmdw_camera_led_switch()

```
kmdw_status_t kmdw_camera_led_switch (
            uint32_t cam_idx,
            uint32_t on )
```

camera set nir led on/off function

Parameters

in	*cam_idx*	camera id
in	*on*	LED on/off cmd, 1: on, 0:off

Returns

kmdw_status_t see kmdw_status_t

4.28.4.13 kmdw_camera_open()

```
kmdw_status_t kmdw_camera_open (
            uint32_t cam_idx )
```

camera open function

Kneron Confidential

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.14 kmdw_camera_set_aec()

```
kmdw_status_t kmdw_camera_set_aec (
            uint32_t cam_idx,
            struct cam_sensor_aec * aec_p )
```

camera set ae controller ROI area function

Parameters

in	*cam_idx*	camera id
in	*aec_p*	point of cam_sensor_aec

Returns

kmdw_status_t see kmdw_status_t

4.28.4.15 kmdw_camera_set_exp_time()

```
kmdw_status_t kmdw_camera_set_exp_time (
            uint32_t cam_idx,
            uint32_t gain1,
            uint32_t gain2 )
```

camera set exposure time function

Parameters

in	*cam_idx*	camera id
in	*gain1*	exposure time parameter 1
in	*gain2*	exposure time parameter 2

Returns

kmdw_status_t see kmdw_status_t

Kneron Confidential

4.28.4.16 kmdw_camera_set_frame_format()

```
kmdw_status_t kmdw_camera_set_frame_format (
            uint32_t cam_idx,
            struct cam_format * format )
```

camera set frame format function

Parameters

in	*cam_idx*	camera id
in	*cap*	point of format information.

Returns

kmdw_status_t see kmdw_status_t

4.28.4.17 kmdw_camera_set_gain()

```
kmdw_status_t kmdw_camera_set_gain (
            uint32_t cam_idx,
            uint32_t gain1,
            uint32_t gain2 )
```

camera set gain function

Parameters

in	*cam_idx*	camera id
in	*gain1*	gain parameter 1
in	*gain2*	gain parameter 2

Returns

kmdw_status_t see kmdw_status_t

4.28.4.18 kmdw_camera_start()

```
kmdw_status_t kmdw_camera_start (
            uint32_t cam_idx )
```

camera start function

Kneron Confidential

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.19 kmdw_camera_stop()

```
kmdw_status_t kmdw_camera_stop (
            uint32_t cam_idx )
```

camera stop function

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.20 kmdw_camera_stream_off()

```
kmdw_status_t kmdw_camera_stream_off (
            uint32_t cam_idx )
```

camera streaming off function

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.4.21 kmdw_camera_stream_on()

```
kmdw_status_t kmdw_camera_stream_on (
            uint32_t cam_idx )
```

Kneron Confidential

camera streaming on function

Kneron Confidential

Parameters

in	*cam_idx*	camera id

Returns

kmdw_status_t see kmdw_status_t

4.28.5 Variable Documentation

4.28.5.1 buffer_capture

```
kmdw_status_t(* cam_ops::buffer_capture) (uint32_t cam_idx, uint32_t *addr, uint32_t *size)
```

4.28.5.2 buffer_init

```
kmdw_status_t(* cam_ops::buffer_init) (uint32_t cam_idx)
```

4.28.5.3 buffer_prepare

```
kmdw_status_t(* cam_ops::buffer_prepare) (uint32_t cam_idx)
```

4.28.5.4 bytesperline

```
uint32_t cam_format::bytesperline
```

4.28.5.5 capabilities

```
uint32_t cam_capability::capabilities
```

Kneron Confidential

4.28.5.6 center_x1

```
uint8_t cam_sensor_aec::center_x1
```

4.28.5.7 center_x2

```
uint8_t cam_sensor_aec::center_x2
```

4.28.5.8 center_y1

```
uint8_t cam_sensor_aec::center_y1
```

4.28.5.9 center_y2

```
uint8_t cam_sensor_aec::center_y2
```

4.28.5.10 close

```
kmdw_status_t(* cam_ops::close) (uint32_t cam_idx)
```

4.28.5.11 colorspace

```
uint32_t cam_format::colorspace
```

4.28.5.12 desc

```
char cam_capability::desc[16]
```

4.28.5.13 driver

```
char cam_capability::driver[16]
```

Kneron Confidential

4.28.5.14 field

```
uint32_t cam_format::field
```

4.28.5.15 get_device_id

```
uint32_t(* cam_ops::get_device_id) (uint32_t cam_idx)
```

4.28.5.16 get_format

```
kmdw_status_t(* cam_ops::get_format) (uint32_t cam_idx, struct cam_format *format)
```

4.28.5.17 get_lux

```
kmdw_status_t(* cam_ops::get_lux) (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8↩
_t *post_gain, uint8_t *global_gain, uint8_t *y_average)
```

4.28.5.18 height

```
uint32_t cam_format::height
```

4.28.5.19 ioctl

```
kmdw_status_t(* cam_ops::ioctl) (uint32_t cam_idx, uint32_t cid, void *data, uint16_t len)
```

4.28.5.20 led_switch

```
kmdw_status_t(* cam_ops::led_switch) (uint32_t cam_idx, uint32_t on)
```

Kneron Confidential

4.28.5.21 open

```
kmdw_status_t(* cam_ops::open) (uint32_t cam_idx)
```

4.28.5.22 pixelformat

```
uint32_t cam_format::pixelformat
```

4.28.5.23 query_capability

```
kmdw_status_t(* cam_ops::query_capability) (uint32_t cam_idx, struct cam_capability *cap)
```

4.28.5.24 set_aec

```
kmdw_status_t(* cam_ops::set_aec) (uint32_t cam_idx, struct cam_sensor_aec *aec_p)
```

4.28.5.25 set_exp_time

```
kmdw_status_t(* cam_ops::set_exp_time) (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
```

4.28.5.26 set_flip

```
kmdw_status_t(* cam_ops::set_flip) (uint32_t cam_idx, uint32_t enable)
```

4.28.5.27 set_format

```
kmdw_status_t(* cam_ops::set_format) (uint32_t cam_idx, struct cam_format *format)
```

4.28.5.28 set_gain

```
kmdw_status_t(* cam_ops::set_gain) (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
```

Kneron Confidential

4.28.5.29 set_mirror

```
kmdw_status_t(* cam_ops::set_mirror) (uint32_t cam_idx, uint32_t enable)
```

4.28.5.30 sizeimage

```
uint32_t cam_format::sizeimage
```

4.28.5.31 start_capture

```
kmdw_status_t(* cam_ops::start_capture) (uint32_t cam_idx)
```

4.28.5.32 stop_capture

```
kmdw_status_t(* cam_ops::stop_capture) (uint32_t cam_idx)
```

4.28.5.33 stream_off

```
kmdw_status_t(* cam_ops::stream_off) (uint32_t cam_idx)
```

4.28.5.34 stream_on

```
kmdw_status_t(* cam_ops::stream_on) (uint32_t cam_idx)
```

4.28.5.35 version

```
uint32_t cam_capability::version
```

4.28.5.36 width

```
uint32_t cam_format::width
```

Kneron Confidential

4.28.5.37 x1

```
uint8_t cam_sensor_aec::x1
```

4.28.5.38 x2

```
uint8_t cam_sensor_aec::x2
```

4.28.5.39 y1

```
uint8_t cam_sensor_aec::y1
```

4.28.5.40 y2

```
uint8_t cam_sensor_aec::y2
```

Kneron Confidential

4.29 KMDW_DISPLAY

Kneron display middleware.

Data Structures

- struct kmdw_display_panel_drv

  Structure of representing display and panel driver compatibility.

Functions

- int kmdw_display_initialize (void)

  Initialize display and panel driver.
- int kmdw_video_renderer_open (struct video_input_params *params)

  Open a video renderer to display frame buffer.
- int kmdw_video_renderer_set_camera (unsigned int cam_idx)

  Set camera source index which be displayed on LCD.
- int kmdw_video_renderer_buffer_initialize (struct video_input_params *params)

  Initilize display frame buffer.
- int kmdw_video_renderer_start (void)

  Turn on display preview.
- int kmdw_video_renderer_stop (void)

  Turn off display preview.
- uint32_t kmdw_video_renderer_get_buffer_addr (void)

  Get display snapshot frame buffer.
- int kmdw_display_get_params (struct video_input_params *dp_params)

  Get the input parameters of display.
- int kmdw_display_update_draw_fb (uint32_t addr, unsigned int cam_idx)

  Update frame buffer which be used to draw something on display.
- int kmdw_display_set_pen_rgb565 (uint16_t color, uint16_t pen_width)

  Set pen width and color.
- int kmdw_display_draw_rect (uint32_t x, uint32_t y, uint32_t width, uint32_t height, uint8_t draw_mode)

  Draw rectangle without filling color on display.
- int kmdw_display_draw_line (uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- int kmdw_display_fill_rect (uint32_t x, uint32_t y, uint32_t width, uint32_t height)

  Draw rectangle with filling color on display.
- int kmdw_display_draw_bitmap (uint32_t x, uint32_t y, uint32_t width, uint32_t height, void *buf)

  Draw bitmap on display.
- int kmdw_display_test_pattern_gen (bool pat_gen)

  Generate display test image display.
- int kmdw_display_refresh (void)
- int kmdw_display_set_backlight (int duty)

  Set display backlight.

Kneron Confidential

4.29.1 Detailed Description

Kneron display middleware.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.29.2 Function Documentation

4.29.2.1 kmdw_display_draw_bitmap()

```
int kmdw_display_draw_bitmap (
            uint32_t x,
            uint32_t y,
            uint32_t width,
            uint32_t height,
            void * buf )
```

Draw bitmap on display.

Parameters

in	*x*	Start x-axis of bitmap on display to draw
in	*y*	Start y-axis of bitmap on display to draw
in	*width*	Width of bitmap
in	*height*	Height of bitmap
in	*buf*	Bitmap of target

Returns

0, Draw bitmap on display successfully
-1, Empty or wrong lcd driver or other errors

Kneron Confidential

Note

Exmpale:
uint32_t buf_idx, buf_addr;
buf_addr = kdp_fb_mgr_next_read(CAMERA_RGB_IDX, &buf_idx);
kmdw_display_update_draw_fb(buf_addr, CAMERA_RGB_IDX);
kmdw_display_draw_bitmap(0, 450, 100, 50, (void ∗)USER_IMG_ICON_ADDR);

Here is the call graph for this function:

4.29.2.2 kmdw_display_draw_line()

```
int kmdw_display_draw_line (
            uint32_t xs,
            uint32_t ys,
            uint32_t xe,
            uint32_t ye )
```

Draw line on display.

Parameters

in	*xs*	Start x-axis of line on display to draw
in	*xe*	End x-axis of line on display to draw
in	*ys*	Start y-axis of line on display to draw
in	*ye*	End y-axis of line on display to draw

Returns

0, Draw line on display successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
uint32_t buf_idx, buf_addr;
buf_addr = kdp_fb_mgr_next_read(CAMERA_RGB_IDX, &buf_idx);
kmdw_display_update_draw_fb(buf_addr, CAMERA_RGB_IDX);
kmdw_display_set_pen_rgb565(BLACK, 4);
kmdw_display_draw_line(50, 400, 50, 300);

Kneron Confidential

Here is the call graph for this function:

4.29.2.3 kmdw_display_draw_rect()

```
int kmdw_display_draw_rect (
            uint32_t x,
            uint32_t y,
            uint32_t width,
            uint32_t height,
            uint8_t draw_mode )
```

Draw rectangle without filling color on display.

Parameters

in	*x*	Start x-axis of rectangle on display to draw
in	*y*	Start y-axis of rectangle on display to draw
in	*width*	Width of rectangle
in	*height*	Height of rectangle
in	*draw_mode*	see KDP_BOUNDINGBOX_MODE

Returns

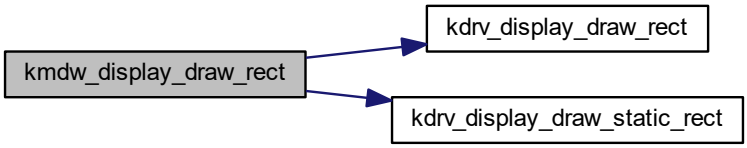
0, Draw rectangle on display successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
uint32_t buf_idx, buf_addr;
buf_addr = kdp_fb_mgr_next_read(CAMERA_RGB_IDX, &buf_idx);
kmdw_display_update_draw_fb(buf_addr, CAMERA_RGB_IDX);
kmdw_display_set_pen_rgb565(BLUE, 8);
kmdw_display_draw_rect(50, 55, 120, 100, DRAW_FDR_RESULT_BOUNDINGBOX);

Kneron Confidential

Here is the call graph for this function:



4.29.2.4 kmdw_display_fill_rect()

```
int kmdw_display_fill_rect (
            uint32_t x,
            uint32_t y,
            uint32_t width,
            uint32_t height )
```

Draw rectangle with filling color on display.

Parameters

in	*x*	Start x-axis of rectangle on display to draw
in	*y*	Start y-axis of rectangle on display to draw
in	*width*	Width of rectangle
in	*height*	Height of rectangle

Returns

0, Draw rectangle on display successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
uint32_t buf_idx, buf_addr;
buf_addr = kdp_fb_mgr_next_read(CAMERA_RGB_IDX, &buf_idx);
kmdw_display_update_draw_fb(buf_addr, CAMERA_RGB_IDX);
kmdw_display_set_pen_rgb565(BLACK, 8);
kmdw_display_fill_rect(0, 0, 640, 480);

Kneron Confidential

Here is the call graph for this function:

4.29.2.5 kmdw_display_get_params()

```
int kmdw_display_get_params (
            struct video_input_params * dp_params )
```

Get the input parameters of display.

Parameters

in	*dp_params*	see video_input_params

Returns

0, Update frame buffer successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
uint32_t buf_idx, buf_addr;
buf_addr = kdp_fb_mgr_next_read(CAMERA_RGB_IDX, &buf_idx);
kmdw_display_update_draw_fb(buf_addr, CAMERA_RGB_IDX)
kmdw_display_draw_xxx();

Here is the call graph for this function:

Kneron Confidential

4.29.2.6 kmdw_display_initialize()

```
int kmdw_display_initialize (
            void  )
```

Initialize display and panel driver.

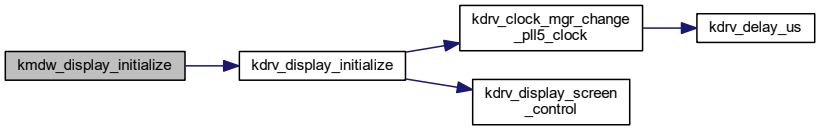
Parameters

in	*N/A*	

Returns

N/A

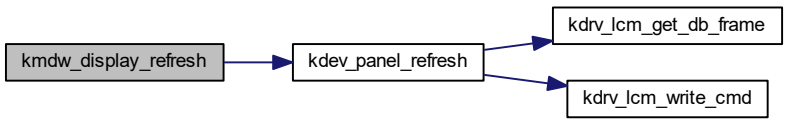
Here is the call graph for this function:



4.29.2.7 kmdw_display_refresh()

```
int kmdw_display_refresh (
            void  )
```

Here is the call graph for this function:



4.29.2.8 kmdw_display_set_backlight()

```
int kmdw_display_set_backlight (
            int duty )
```

Set display backlight.

Kneron Confidential

Parameters

in	*duty*	see kdrv_display_backlight_t

Returns

0, Set display backlight successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
kkmdw_display_set_backlight(KDRV_DISPLAY_BACKLIGHT_ON);

Here is the call graph for this function:

4.29.2.9 kmdw_display_set_pen_rgb565()

```
int kmdw_display_set_pen_rgb565 (
            uint16_t color,
            uint16_t pen_width )
```

Set pen width and color.

Parameters

in	*color*	Color of pen
in	*pen_width*	Width of pen

Returns

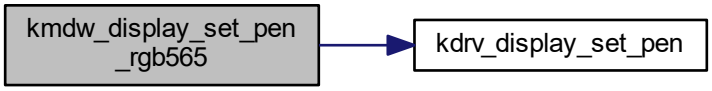
0, Set pen width and color successfully
-1, Empty or wrong lcd driver or other errors

Kneron Confidential

Note

Exmpale:
kmdw_display_set_pen_rgb565(BLUE, 8);

Here is the call graph for this function:



4.29.2.10 kmdw_display_test_pattern_gen()

```
int kmdw_display_test_pattern_gen (
            bool pat_gen )
```

Generate display test image display.

Parameters

in	*pat_gen*	TRUE or FALSE to generate test pattern

Returns

0, Draw rectangle on display successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
kmdw_display_test_pattern_gen(TRUE);
kmdw_display_initialize(); kmdw_video_renderer_open(&params);

Here is the call graph for this function:

Kneron Confidential

4.29.2.11 kmdw_display_update_draw_fb()

```
int kmdw_display_update_draw_fb (
            uint32_t addr,
            unsigned int cam_idx )
```

Update frame buffer which be used to draw something on display.

Parameters

in	*addr*	Frame buffer address
in	*cam_idx*	Camera source index

Returns

0, Update frame buffer successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
uint32_t buf_idx, buf_addr;
buf_addr = kdp_fb_mgr_next_read(CAMERA_RGB_IDX, &buf_idx);
kmdw_display_update_draw_fb(buf_addr, CAMERA_RGB_IDX)
kmdw_display_draw_xxx();

4.29.2.12 kmdw_video_renderer_buffer_initialize()

```
int kmdw_video_renderer_buffer_initialize (
            struct video_input_params * params )
```

Initilize display frame buffer.

Parameters

in	*params*	see video_input_params

Returns

0, Initilize display frame buffer successfully
-1, Empty or wrong lcd driver or other errors

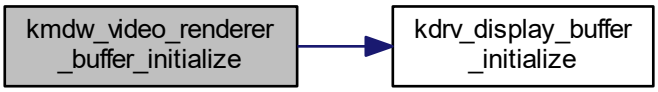
Note

Exmpale:
struct video_input_params params;
params.input_fmt = V2K_PIX_FMT_RGB565;
params.input_xres = LCD_WIDTH;

Kneron Confidential

```
params.input_yres = LCD_HEIGHT;
kmdw_video_renderer_open(&params);
kmdw_video_renderer_set_camera(CAMERA_RGB_IDX);
kmdw_video_renderer_buffer_initialize(&params);
```

Here is the call graph for this function:



4.29.2.13 kmdw_video_renderer_get_buffer_addr()

```
uint32_t kmdw_video_renderer_get_buffer_addr (
            void  )
```

Get display snapshot frame buffer.

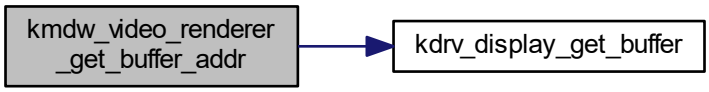
Returns

>0, Return snapshot frame buffer successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
uint32_t buf_addr;
buf_addr = kmdw_video_renderer_get_buffer_addr();

Here is the call graph for this function:



4.29.2.14 kmdw_video_renderer_open()

```
int kmdw_video_renderer_open (
            struct video_input_params * params )
```

Open a video renderer to display frame buffer.

Kneron Confidential

Parameters

in	*params*	see video_input_params

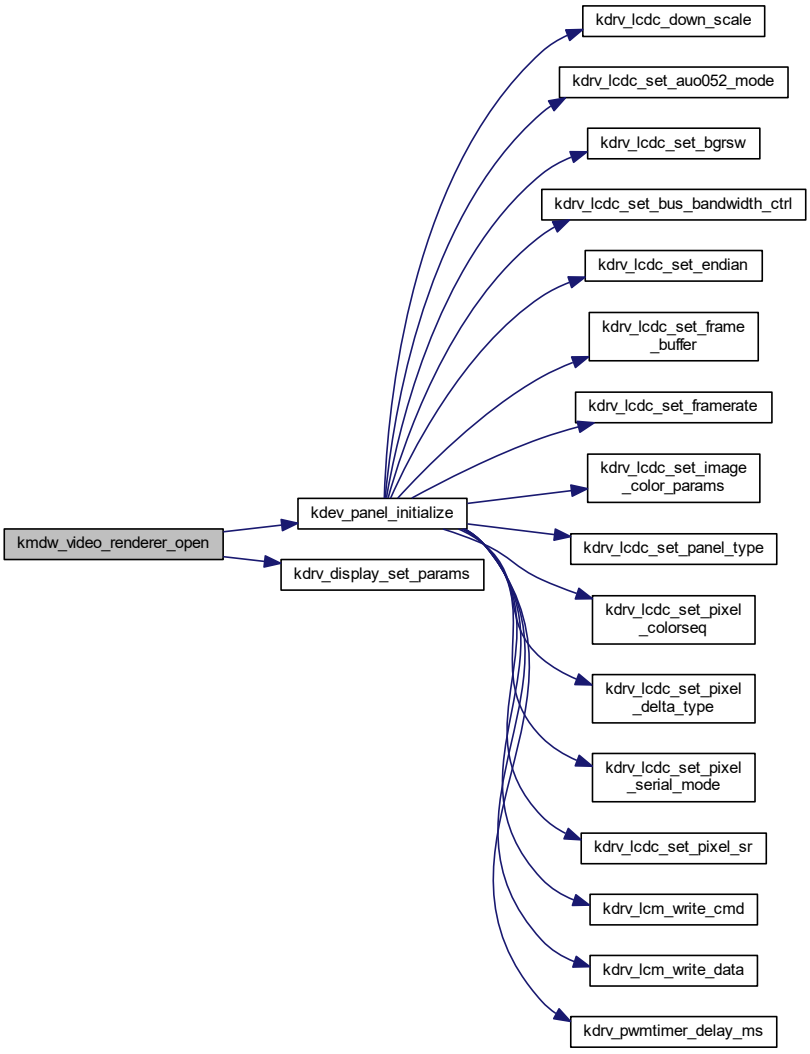
Returns

0, Open a video renderer to display frame buffer successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
struct video_input_params params;
params.input_fmt = V2K_PIX_FMT_RGB565;
params.input_xres = LCD_WIDTH;
params.input_yres = LCD_HEIGHT;
kmdw_video_renderer_open(&params);

Here is the call graph for this function:



Kneron Confidential

4.29.2.15 kmdw_video_renderer_set_camera()

```
int kmdw_video_renderer_set_camera (
            unsigned int cam_idx )
```

Set camera source index which be displayed on LCD.

Parameters

in	*cam_idx*	Camera source index

Returns

0, Set camera source index which be displayed on LCD successfully
-1, Empty or wrong lcd driver or other errors

Note

Exmpale:
struct video_input_params params;
params.input_fmt = V2K_PIX_FMT_RGB565;
params.input_xres = LCD_WIDTH;
params.input_yres = LCD_HEIGHT;
kmdw_video_renderer_open(&params);
kmdw_video_renderer_set_camera(CAMERA_RGB_IDX);

4.29.2.16 kmdw_video_renderer_start()

```
int kmdw_video_renderer_start (
            void  )
```

Turn on display preview.

Returns

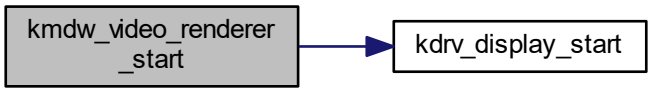
0, Turn on display preview successfully
-1, Empty or wrong lcd driver or other errors

Kneron Confidential

Note

Exmpale:
struct video_input_params params;
params.input_fmt = V2K_PIX_FMT_RGB565;
params.input_xres = LCD_WIDTH;
params.input_yres = LCD_HEIGHT;
kmdw_video_renderer_open(&params);
kmdw_video_renderer_set_camera(CAMERA_RGB_IDX);
kmdw_video_renderer_buffer_initialize(&params);
kmdw_video_renderer_start();

Here is the call graph for this function:



4.29.2.17 kmdw_video_renderer_stop()

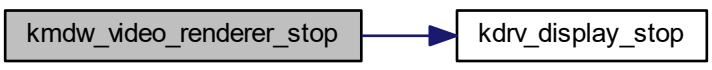
```
int kmdw_video_renderer_stop (
            void  )
```

Turn off display preview.

Returns

0, Turn off display preview successfully
-1, Empty or wrong lcd driver or other errors

Here is the call graph for this function:



Kneron Confidential

4.30 KMDW_SENSOR

Kneron sensor middleware for sensor device.

Data Structures

- struct sensor_device
- struct sensor_init_seq
- struct sensor_datafmt_info
- struct sensor_win_size

Macros

- #define fourcc(a, b, c, d) ((uint32_t)(a) | ((uint32_t)(b) << 8) | ((uint32_t)(c) << 16) | ((uint32_t)(d) << 24))
- #define PIX_FMT_YCBCR fourcc('Y', 'B', 'Y', 'R')
- #define PIX_FMT_RGB565 fourcc('R', 'G', 'B', 'P')
- #define PIX_FMT_RAW10 fourcc('R', 'A', '1', '0')
- #define PIX_FMT_RAW8 fourcc('R', 'A', 'W', '8')

Enumerations

- enum colorspace { COLORSPACE_RGB = 0, COLORSPACE_YUV = 1, COLORSPACE_RAW = 2 }

Functions

- struct sensor_init_seq __attribute__ ((packed))
- kmdw_status_t kmdw_sensor_s_power (uint32_t cam_idx, uint32_t on)

  set sensor power function
- kmdw_status_t kmdw_sensor_reset (uint32_t cam_idx)

  sensor reset function
- kmdw_status_t kmdw_sensor_s_stream (uint32_t cam_idx, uint32_t enable)

  set sensor stream function
- kmdw_status_t kmdw_sensor_enum_fmt (uint32_t cam_idx, uint32_t index, uint32_t *fourcc)

  set sensor enum function
- kmdw_status_t kmdw_sensor_set_fmt (uint32_t cam_idx, struct cam_format *format)

  set sensor format function
- kmdw_status_t kmdw_sensor_get_fmt (uint32_t cam_idx, struct cam_format *format)

  get sensor format function
- kmdw_status_t kmdw_sensor_set_gain (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  get sensor gain function
- kmdw_status_t kmdw_sensor_set_aec (uint32_t cam_idx, struct cam_sensor_aec *aec_p)

  sensor set ae controller ROI area function
- kmdw_status_t kmdw_sensor_set_exp_time (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  sensor set exposure time function
- kmdw_status_t kmdw_sensor_get_lux (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post↩ _gain, uint8_t *global_gain, uint8_t *y_average)

  sensor get lum and other parameter function
- kmdw_status_t kmdw_sensor_led_switch (uint32_t cam_idx, uint32_t on)

Kneron Confidential

sensor set nir led on/off function

- kmdw_status_t kmdw_sensor_set_mirror (uint32_t cam_idx, uint32_t enable)

  sensor set image mirror on/off function
- kmdw_status_t kmdw_sensor_set_flip (uint32_t cam_idx, uint32_t enable)

  sensor set image flip on/off function
- uint32_t kmdw_sensor_get_dev_id (uint32_t cam_idx)

  sensor get device ID function
- kmdw_status_t kmdw_sensor_register (uint32_t cam_idx, uint32_t sensor_idx)

  register sensor

Variables

- uint16_t sensor_device::addr
- uint16_t sensor_init_seq::addr
- uint8_t sensor_init_seq::value
- uint16_t addr
- uint8_t value
- uint32_t sensor_datafmt_info::fourcc
- enum colorspace sensor_datafmt_info::colorspace
- struct sensor_datafmt_info __attribute__

  Endpoint descriptor struct.
- uint32_t sensor_win_size::width
- uint32_t sensor_win_size::height

4.30.1 Detailed Description

Kneron sensor middleware for sensor device.

Copyright

Copyright (C) 2020 Kneron, Inc. All rights reserved.

4.30.2 Macro Definition Documentation

4.30.2.1 fourcc

```
#define fourcc(
              a,
              b,
              c,
              d ) ((uint32_t)(a) | ((uint32_t)(b) << 8) | ((uint32_t)(c) << 16) | ((uint32_↩
t)(d) << 24))
```

Kneron Confidential

4.30.2.2 PIX_FMT_RAW10

```
#define PIX_FMT_RAW10 fourcc('R', 'A', '1', '0')
```

4.30.2.3 PIX_FMT_RAW8

```
#define PIX_FMT_RAW8 fourcc('R', 'A', 'W', '8')
```

4.30.2.4 PIX_FMT_RGB565

```
#define PIX_FMT_RGB565 fourcc('R', 'G', 'B', 'P')
```

4.30.2.5 PIX_FMT_YCBCR

```
#define PIX_FMT_YCBCR fourcc('Y', 'B', 'Y', 'R')
```

4.30.3 Enumeration Type Documentation

4.30.3.1 colorspace

```
enum colorspace
```

Enumerator

COLORSPACE_RGB	
COLORSPACE_YUV	
COLORSPACE_RAW	

4.30.4 Function Documentation

4.30.4.1 __attribute__()

```
struct sensor_init_seq __attribute__ (
            (packed)  )
```

Kneron Confidential

4.30.4.2 kmdw_sensor_enum_fmt()

```
kmdw_status_t kmdw_sensor_enum_fmt (
            uint32_t cam_idx,
            uint32_t index,
            uint32_t * fourcc )
```

set sensor enum function

Parameters

in	*cam_idx*	camera id
in	*index*	index
out	*fourcc*	point of fourcc

Returns

kmdw_status_t see kmdw_status_t

4.30.4.3 kmdw_sensor_get_dev_id()

```
uint32_t kmdw_sensor_get_dev_id (
            uint32_t cam_idx )
```

sensor get device ID function

Parameters

in	*cam_idx*	camera id

Returns

sensor id

4.30.4.4 kmdw_sensor_get_fmt()

```
kmdw_status_t kmdw_sensor_get_fmt (
            uint32_t cam_idx,
            struct cam_format * format )
```

get sensor format function

Kneron Confidential

Parameters

in	*cam_idx*	camera id
out	*format*	point of cam_format

Returns

kmdw_status_t see kmdw_status_t

4.30.4.5 kmdw_sensor_get_lux()

```
kmdw_status_t kmdw_sensor_get_lux (
            uint32_t cam_idx,
            uint16_t * expo,
            uint8_t * pre_gain,
            uint8_t * post_gain,
            uint8_t * global_gain,
            uint8_t * y_average )
```

sensor get lum and other parameter function

Parameters

in	*cam_idx*	camera id
out	*expo*	exposure time parameter
out	*pre_gain*	exposure time parameter
out	*post_gain*	exposure time parameter
out	*global_gain*	exposure time parameter
in	*y_average*	exposure time parameter 2

Returns

kmdw_status_t see kmdw_status_t

4.30.4.6 kmdw_sensor_led_switch()

```
kmdw_status_t kmdw_sensor_led_switch (
            uint32_t cam_idx,
            uint32_t on )
```

sensor set nir led on/off function

Parameters

in	*cam_idx*	camera id
in	*on*	LED on/off cmd, 1: on, 0:off

Kneron Confidential

Returns

kmdw_status_t see kmdw_status_t

4.30.4.7 kmdw_sensor_register()

```
kmdw_status_t kmdw_sensor_register (
            uint32_t cam_idx,
            uint32_t sensor_idx )
```

register sensor

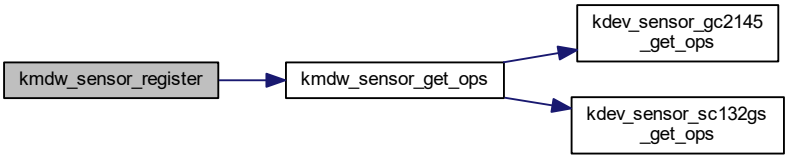
Parameters

in	*cam_idx*	camera id
in	*sensor_idx*	sensor id

Returns

kmdw_status_t see kmdw_status_t

Here is the call graph for this function:



4.30.4.8 kmdw_sensor_reset()

```
kmdw_status_t kmdw_sensor_reset (
            uint32_t cam_idx )
```

sensor reset function

Parameters

in	*cam_idx*	camera id

Kneron confidential

Returns

kmdw_status_t see kmdw_status_t

4.30.4.9 kmdw_sensor_s_power()

```
kmdw_status_t kmdw_sensor_s_power (
            uint32_t cam_idx,
            uint32_t on )
```

set sensor power function

Parameters

in	*cam_idx*	camera id
in	*on*	power on/off, 1:on, 0:off

Returns

kmdw_status_t see kmdw_status_t

4.30.4.10 kmdw_sensor_s_stream()

```
kmdw_status_t kmdw_sensor_s_stream (
            uint32_t cam_idx,
            uint32_t enable )
```

set sensor stream function

Parameters

in	*cam_idx*	camera id
in	*enable*	stream enable/disable, 1:enable, 0:disable

Returns

kmdw_status_t see kmdw_status_t

4.30.4.11 kmdw_sensor_set_aec()

```
kmdw_status_t kmdw_sensor_set_aec (
            uint32_t cam_idx,
            struct cam_sensor_aec * aec_p )
```

sensor set ae controller ROI area function

Kneron Confidential

Parameters

in	*cam_idx*	camera id
in	*aec_p*	point of cam_sensor_aec

Returns

kmdw_status_t see kmdw_status_t

4.30.4.12 kmdw_sensor_set_exp_time()

```
kmdw_status_t kmdw_sensor_set_exp_time (
            uint32_t cam_idx,
            uint32_t gain1,
            uint32_t gain2 )
```

sensor set exposure time function

Parameters

in	*cam_idx*	camera id
in	*gain1*	exposure time parameter 1
in	*gain2*	exposure time parameter 2

Returns

kmdw_status_t see kmdw_status_t

4.30.4.13 kmdw_sensor_set_flip()

```
kmdw_status_t kmdw_sensor_set_flip (
            uint32_t cam_idx,
            uint32_t enable )
```

sensor set image flip on/off function

Parameters

in	*cam_idx*	camera id
in	*enable*	enable/disable cmd, 1: enable, 0:disable

Returns

kmdw_status_t see kmdw_status_t

Kneron Confidential

4.30.4.14 kmdw_sensor_set_fmt()

```
kmdw_status_t kmdw_sensor_set_fmt (
            uint32_t cam_idx,
            struct cam_format * format )
```

set sensor format function

Parameters

in	*cam_idx*	camera id
in	*format*	point of cam_format

Returns

kmdw_status_t see kmdw_status_t

4.30.4.15 kmdw_sensor_set_gain()

```
kmdw_status_t kmdw_sensor_set_gain (
            uint32_t cam_idx,
            uint32_t gain1,
            uint32_t gain2 )
```

get sensor gain function

Parameters

in	*cam_idx*	camera id
in	*gain1*	aec gain parameter 1
in	*gain2*	aec gain parameter 2

Returns

kmdw_status_t see kmdw_status_t

4.30.4.16 kmdw_sensor_set_mirror()

```
kmdw_status_t kmdw_sensor_set_mirror (
            uint32_t cam_idx,
            uint32_t enable )
```

sensor set image mirror on/off function

Kneron Confidential

Parameters

in	*cam_idx*	camera id
in	*enable*	enable/disable cmd, 1: enable, 0:disable

Returns

kmdw_status_t see kmdw_status_t

4.30.5 Variable Documentation

4.30.5.1 __attribute__

```
struct uvc_format_frame_based __attribute__
```

Endpoint descriptor struct.

USB Interface Association Descriptor.

USB Common Descriptor.

USB String Descriptor.

USB Standard Endpoint Descriptor.

USB Standard Interface Descriptor.

USB Standard Configuration Descriptor.

USB 2.0 Device Qualifier Descriptor.

Device qualifier descriptor struct.

Device descriptor struct.

Configuration descriptor struct.

Interface descriptor struct.

4.30.5.2 addr [1/3]

```
uint16_t sensor_device::addr
```

4.30.5.3 addr [2/3]

```
uint16_t sensor_init_seq::addr
```

Kneron Confidential

4.30.5.4 addr [3/3]

```
uint16_t addr
```

4.30.5.5 colorspace

```
enum colorspace sensor_datafmt_info::colorspace
```

4.30.5.6 fourcc

```
uint32_t sensor_datafmt_info::fourcc
```

4.30.5.7 height

```
uint32_t sensor_win_size::height
```

4.30.5.8 value [1/2]

```
uint8_t sensor_init_seq::value
```

4.30.5.9 value [2/2]

```
uint8_t value
```

4.30.5.10 width

```
uint32_t sensor_win_size::width
```

Kneron Confidential

Chapter 5

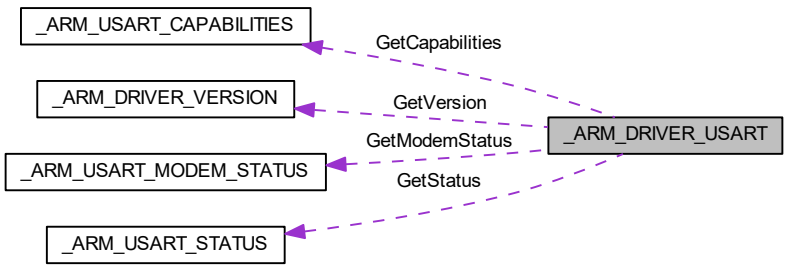
Data Structure Documentation

5.1 _ARM_DRIVER_USART Struct Reference

Access structure of the USART Driver.

`#include <Driver_USART.h>`

Collaboration diagram for _ARM_DRIVER_USART:



Data Fields

- ARM_DRIVER_VERSION(∗ GetVersion)(void)
  Pointer to ARM_USART_GetVersion : Get driver version.
- ARM_USART_CAPABILITIES(∗ GetCapabilities)(void)
  Pointer to ARM_USART_GetCapabilities : Get driver capabilities.
- int32_t(∗ Initialize)(ARM_USART_SignalEvent_t cb_event)
  Pointer to ARM_USART_Initialize : Initialize USART Interface.
- int32_t(∗ Uninitialize)(void)
  Pointer to ARM_USART_Uninitialize : De-initialize USART Interface.
- int32_t(∗ PowerControl)(ARM_POWER_STATE state)
  Pointer to ARM_USART_PowerControl : Control USART Interface Power.
- int32_t(∗ Send)(const void ∗data, uint32_t num)

Kneron Confidential

Pointer to ARM_USART_Send : Start sending data to USART transmitter.

- int32_t(∗ Receive)(void ∗data, uint32_t num)

  Pointer to ARM_USART_Receive : Start receiving data from USART receiver.
- int32_t(∗ Transfer)(const void ∗data_out, void ∗data_in, uint32_t num)

  Pointer to ARM_USART_Transfer : Start sending/receiving data to/from USART.
- uint32_t(∗ GetTxCount)(void)

  Pointer to ARM_USART_GetTxCount : Get transmitted data count.
- uint32_t(∗ GetRxCount)(void)

  Pointer to ARM_USART_GetRxCount : Get received data count.
- int32_t(∗ Control)(uint32_t control, uint32_t arg)

  Pointer to ARM_USART_Control : Control USART Interface.
- ARM_USART_STATUS(∗ GetStatus)(void)

  Pointer to ARM_USART_GetStatus : Get USART status.
- int32_t(∗ SetModemControl)(ARM_USART_MODEM_CONTROL control)

  Pointer to ARM_USART_SetModemControl : Set USART Modem Control line state.
- ARM_USART_MODEM_STATUS(∗ GetModemStatus)(void)

  Pointer to ARM_USART_GetModemStatus : Get USART Modem Status lines state.

5.1.1 Detailed Description

Access structure of the USART Driver.

5.1.2 Field Documentation

5.1.2.1 Control

```
int32_t(* _ARM_DRIVER_USART::Control) (uint32_t control, uint32_t arg)
```

Pointer to ARM_USART_Control : Control USART Interface.

5.1.2.2 GetCapabilities

```
ARM_USART_CAPABILITIES(* _ARM_DRIVER_USART::GetCapabilities) (void)
```

Pointer to ARM_USART_GetCapabilities : Get driver capabilities.

5.1.2.3 GetModemStatus

```
ARM_USART_MODEM_STATUS(* _ARM_DRIVER_USART::GetModemStatus) (void)
```

Pointer to ARM_USART_GetModemStatus : Get USART Modem Status lines state.

Kneron Confidential

5.1.2.4 GetRxCount

```
uint32_t(* _ARM_DRIVER_USART::GetRxCount) (void)
```

Pointer to ARM_USART_GetRxCount : Get received data count.

5.1.2.5 GetStatus

```
ARM_USART_STATUS(* _ARM_DRIVER_USART::GetStatus) (void)
```

Pointer to ARM_USART_GetStatus : Get USART status.

5.1.2.6 GetTxCount

```
uint32_t(* _ARM_DRIVER_USART::GetTxCount) (void)
```

Pointer to ARM_USART_GetTxCount : Get transmitted data count.

5.1.2.7 GetVersion

```
ARM_DRIVER_VERSION(* _ARM_DRIVER_USART::GetVersion) (void)
```

Pointer to ARM_USART_GetVersion : Get driver version.

5.1.2.8 Initialize

```
int32_t(* _ARM_DRIVER_USART::Initialize) (ARM_USART_SignalEvent_t cb_event)
```

Pointer to ARM_USART_Initialize : Initialize USART Interface.

5.1.2.9 PowerControl

```
int32_t(* _ARM_DRIVER_USART::PowerControl) (ARM_POWER_STATE state)
```

Pointer to ARM_USART_PowerControl : Control USART Interface Power.

Kneron Confidential

5.1.2.10 Receive

```
int32_t(* _ARM_DRIVER_USART::Receive) (void *data, uint32_t num)
```

Pointer to ARM_USART_Receive : Start receiving data from USART receiver.

5.1.2.11 Send

```
int32_t(* _ARM_DRIVER_USART::Send) (const void *data, uint32_t num)
```

Pointer to ARM_USART_Send : Start sending data to USART transmitter.

5.1.2.12 SetModemControl

```
int32_t(* _ARM_DRIVER_USART::SetModemControl) (ARM_USART_MODEM_CONTROL control)
```

Pointer to ARM_USART_SetModemControl : Set USART Modem Control line state.

5.1.2.13 Transfer

```
int32_t(* _ARM_DRIVER_USART::Transfer) (const void *data_out, void *data_in, uint32_t num)
```

Pointer to ARM_USART_Transfer : Start sending/receiving data to/from USART.

5.1.2.14 Uninitialize

```
int32_t(* _ARM_DRIVER_USART::Uninitialize) (void)
```

Pointer to ARM_USART_Uninitialize : De-initialize USART Interface.

The documentation for this struct was generated from the following file:

- Driver_USART.h

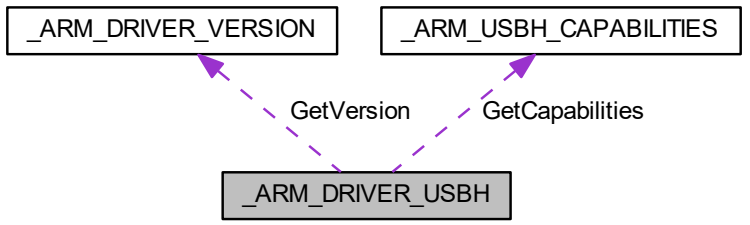
Kneron Confidential

5.2 _ARM_DRIVER_USBH Struct Reference

Access structure of USB Host Driver.

`#include <Driver_USBH.h>`

Collaboration diagram for _ARM_DRIVER_USBH:



Data Fields

- ARM_DRIVER_VERSION(∗ GetVersion)(void)

  Pointer to ARM_USBH_GetVersion : Get driver version.
- ARM_USBH_CAPABILITIES(∗ GetCapabilities)(void)

  Pointer to ARM_USBH_GetCapabilities : Get driver capabilities.
- int32_t(∗ Initialize)(ARM_USBH_SignalPortEvent_t cb_port_event, ARM_USBH_SignalPipeEvent_t cb_↩ pipe_event)

  Pointer to ARM_USBH_Initialize : Initialize USB Host Interface.
- int32_t(∗ Uninitialize)(void)

  Pointer to ARM_USBH_Uninitialize : De-initialize USB Host Interface.
- int32_t(∗ PowerControl)(ARM_POWER_STATE state)

  Pointer to ARM_USBH_PowerControl : Control USB Host Interface Power.
- int32_t(∗ PortVbusOnOff)(uint8_t port, bool vbus)

  Pointer to ARM_USBH_PortVbusOnOff : Root HUB Port VBUS on/off.
- int32_t(∗ PortReset)(uint8_t port)

  Pointer to ARM_USBH_PortReset : Do Root HUB Port Reset.
- int32_t(∗ PortSuspend)(uint8_t port)

  Pointer to ARM_USBH_PortSuspend : Suspend Root HUB Port (stop generating SOFs).
- int32_t(∗ PortResume)(uint8_t port)

  Pointer to ARM_USBH_PortResume : Resume Root HUB Port (start generating SOFs).
- ARM_USBH_PORT_STATE(∗ PortGetState)(uint8_t port)

  Pointer to ARM_USBH_PortGetState : Get current Root HUB Port State.
- ARM_USBH_PIPE_HANDLE(∗ PipeCreate)(uint8_t dev_addr, uint8_t dev_speed, uint8_t hub_addr, uint8↩ _t hub_port, uint8_t ep_addr, uint8_t ep_type, uint16_t ep_max_packet_size, uint8_t ep_interval)

  Pointer to ARM_USBH_PipeCreate : Create Pipe in System.
- int32_t(∗ PipeModify)(ARM_USBH_PIPE_HANDLE pipe_hndl, uint8_t dev_addr, uint8_t dev_speed, uint8↩ _t hub_addr, uint8_t hub_port, uint16_t ep_max_packet_size)

  Pointer to ARM_USBH_PipeModify : Modify Pipe in System.
- int32_t(∗ PipeDelete)(ARM_USBH_PIPE_HANDLE pipe_hndl)

Kneron Confidential

Pointer to ARM_USBH_PipeDelete : Delete Pipe from System.

- int32_t(∗ PipeReset)(ARM_USBH_PIPE_HANDLE pipe_hndl)

  Pointer to ARM_USBH_PipeReset : Reset Pipe.

- int32_t(∗ PipeTransfer)(ARM_USBH_PIPE_HANDLE pipe_hndl, uint32_t packet, uint8_t ∗data, uint32_t num)

  Pointer to ARM_USBH_PipeTransfer : Transfer packets through USB Pipe.

- uint32_t(∗ PipeTransferGetResult)(ARM_USBH_PIPE_HANDLE pipe_hndl)

  Pointer to ARM_USBH_PipeTransferGetResult : Get result of USB Pipe transfer.

- int32_t(∗ PipeTransferAbort)(ARM_USBH_PIPE_HANDLE pipe_hndl)

  Pointer to ARM_USBH_PipeTransferAbort : Abort current USB Pipe transfer.

- uint16_t(∗ GetFrameNumber)(void)

  Pointer to ARM_USBH_GetFrameNumber : Get current USB Frame Number.

5.2.1 Detailed Description

Access structure of USB Host Driver.

5.2.2 Field Documentation

5.2.2.1 GetCapabilities

```
ARM_USBH_CAPABILITIES(* _ARM_DRIVER_USBH::GetCapabilities) (void)
```

Pointer to ARM_USBH_GetCapabilities : Get driver capabilities.

5.2.2.2 GetFrameNumber

```
uint16_t(* _ARM_DRIVER_USBH::GetFrameNumber) (void)
```

Pointer to ARM_USBH_GetFrameNumber : Get current USB Frame Number.

5.2.2.3 GetVersion

```
ARM_DRIVER_VERSION(* _ARM_DRIVER_USBH::GetVersion) (void)
```

Pointer to ARM_USBH_GetVersion : Get driver version.

Kneron Confidential

5.2.2.4 Initialize

```
int32_t(* _ARM_DRIVER_USBH::Initialize) (ARM_USBH_SignalPortEvent_t cb_port_event, ARM_USBH_SignalPipeEvent_t
cb_pipe_event)
```

Pointer to ARM_USBH_Initialize : Initialize USB Host Interface.

5.2.2.5 PipeCreate

```
ARM_USBH_PIPE_HANDLE(* _ARM_DRIVER_USBH::PipeCreate) (uint8_t dev_addr, uint8_t dev_speed,
uint8_t hub_addr, uint8_t hub_port, uint8_t ep_addr, uint8_t ep_type, uint16_t ep_max_packet↩
_size, uint8_t ep_interval)
```

Pointer to ARM_USBH_PipeCreate : Create Pipe in System.

5.2.2.6 PipeDelete

```
int32_t(* _ARM_DRIVER_USBH::PipeDelete) (ARM_USBH_PIPE_HANDLE pipe_hndl)
```

Pointer to ARM_USBH_PipeDelete : Delete Pipe from System.

5.2.2.7 PipeModify

```
int32_t(* _ARM_DRIVER_USBH::PipeModify) (ARM_USBH_PIPE_HANDLE pipe_hndl, uint8_t dev_addr,
uint8_t dev_speed, uint8_t hub_addr, uint8_t hub_port, uint16_t ep_max_packet_size)
```

Pointer to ARM_USBH_PipeModify : Modify Pipe in System.

5.2.2.8 PipeReset

```
int32_t(* _ARM_DRIVER_USBH::PipeReset) (ARM_USBH_PIPE_HANDLE pipe_hndl)
```

Pointer to ARM_USBH_PipeReset : Reset Pipe.

5.2.2.9 PipeTransfer

```
int32_t(* _ARM_DRIVER_USBH::PipeTransfer) (ARM_USBH_PIPE_HANDLE pipe_hndl, uint32_t packet,
uint8_t *data, uint32_t num)
```

Pointer to ARM_USBH_PipeTransfer : Transfer packets through USB Pipe.

Kneron Confidential

5.2.2.10 PipeTransferAbort

```
int32_t(* _ARM_DRIVER_USBH::PipeTransferAbort) (ARM_USBH_PIPE_HANDLE pipe_hndl)
```

Pointer to ARM_USBH_PipeTransferAbort : Abort current USB Pipe transfer.

5.2.2.11 PipeTransferGetResult

```
uint32_t(* _ARM_DRIVER_USBH::PipeTransferGetResult) (ARM_USBH_PIPE_HANDLE pipe_hndl)
```

Pointer to ARM_USBH_PipeTransferGetResult : Get result of USB Pipe transfer.

5.2.2.12 PortGetState

```
ARM_USBH_PORT_STATE(* _ARM_DRIVER_USBH::PortGetState) (uint8_t port)
```

Pointer to ARM_USBH_PortGetState : Get current Root HUB Port State.

5.2.2.13 PortReset

```
int32_t(* _ARM_DRIVER_USBH::PortReset) (uint8_t port)
```

Pointer to ARM_USBH_PortReset : Do Root HUB Port Reset.

5.2.2.14 PortResume

```
int32_t(* _ARM_DRIVER_USBH::PortResume) (uint8_t port)
```

Pointer to ARM_USBH_PortResume : Resume Root HUB Port (start generating SOFs).

5.2.2.15 PortSuspend

```
int32_t(* _ARM_DRIVER_USBH::PortSuspend) (uint8_t port)
```

Pointer to ARM_USBH_PortSuspend : Suspend Root HUB Port (stop generating SOFs).

Kneron Confidential

5.2.2.16 PortVbusOnOff

```
int32_t(* _ARM_DRIVER_USBH::PortVbusOnOff) (uint8_t port, bool vbus)
```

Pointer to ARM_USBH_PortVbusOnOff : Root HUB Port VBUS on/off.

5.2.2.17 PowerControl

```
int32_t(* _ARM_DRIVER_USBH::PowerControl) (ARM_POWER_STATE state)
```

Pointer to ARM_USBH_PowerControl : Control USB Host Interface Power.

5.2.2.18 Uninitialize

```
int32_t(* _ARM_DRIVER_USBH::Uninitialize) (void)
```

Pointer to ARM_USBH_Uninitialize : De-initialize USB Host Interface.

The documentation for this struct was generated from the following file:

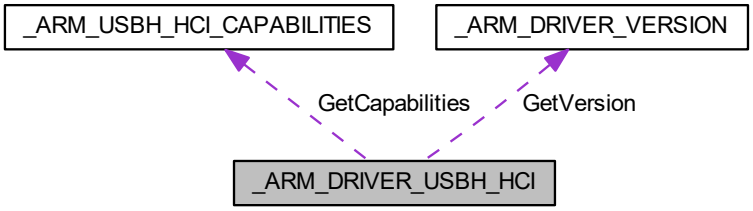
- Driver_USBH.h

5.3 _ARM_DRIVER_USBH_HCI Struct Reference

Access structure of USB Host HCI (OHCI/EHCI) Driver.

```
#include <Driver_USBH.h>
```

Collaboration diagram for _ARM_DRIVER_USBH_HCI:



Kneron Confidential

Data Fields

- ARM_DRIVER_VERSION(∗ GetVersion)(void)

  Pointer to ARM_USBH_HCI_GetVersion : Get USB Host HCI (OHCI/EHCI) driver version.
- ARM_USBH_HCI_CAPABILITIES(∗ GetCapabilities)(void)

  Pointer to ARM_USBH_HCI_GetCapabilities : Get driver capabilities.
- int32_t(∗ Initialize)(ARM_USBH_HCI_Interrupt_t cb_interrupt)

  Pointer to ARM_USBH_HCI_Initialize : Initialize USB Host HCI (OHCI/EHCI) Interface.
- int32_t(∗ Uninitialize)(void)

  Pointer to ARM_USBH_HCI_Uninitialize : De-initialize USB Host HCI (OHCI/EHCI) Interface.
- int32_t(∗ PowerControl)(ARM_POWER_STATE state)

  Pointer to ARM_USBH_HCI_PowerControl : Control USB Host HCI (OHCI/EHCI) Interface Power.
- int32_t(∗ PortVbusOnOff)(uint8_t port, bool vbus)

  Pointer to ARM_USBH_HCI_PortVbusOnOff : USB Host HCI (OHCI/EHCI) Root HUB Port VBUS on/off.

5.3.1 Detailed Description

Access structure of USB Host HCI (OHCI/EHCI) Driver.

5.3.2 Field Documentation

5.3.2.1 GetCapabilities

```
ARM_USBH_HCI_CAPABILITIES(* _ARM_DRIVER_USBH_HCI::GetCapabilities) (void)
```

Pointer to ARM_USBH_HCI_GetCapabilities : Get driver capabilities.

5.3.2.2 GetVersion

```
ARM_DRIVER_VERSION(* _ARM_DRIVER_USBH_HCI::GetVersion) (void)
```

Pointer to ARM_USBH_HCI_GetVersion : Get USB Host HCI (OHCI/EHCI) driver version.

5.3.2.3 Initialize

```
int32_t(* _ARM_DRIVER_USBH_HCI::Initialize) (ARM_USBH_HCI_Interrupt_t cb_interrupt)
```

Pointer to ARM_USBH_HCI_Initialize : Initialize USB Host HCI (OHCI/EHCI) Interface.

Kneron Confidential

5.3.2.4 PortVbusOnOff

```
int32_t(* _ARM_DRIVER_USBH_HCI::PortVbusOnOff) (uint8_t port, bool vbus)
```

Pointer to ARM_USBH_HCI_PortVbusOnOff : USB Host HCI (OHCI/EHCI) Root HUB Port VBUS on/off.

5.3.2.5 PowerControl

```
int32_t(* _ARM_DRIVER_USBH_HCI::PowerControl) (ARM_POWER_STATE state)
```

Pointer to ARM_USBH_HCI_PowerControl : Control USB Host HCI (OHCI/EHCI) Interface Power.

5.3.2.6 Uninitialize

```
int32_t(* _ARM_DRIVER_USBH_HCI::Uninitialize) (void)
```

Pointer to ARM_USBH_HCI_Uninitialize : De-initialize USB Host HCI (OHCI/EHCI) Interface.

The documentation for this struct was generated from the following file:

- Driver_USBH.h

5.4 _ARM_DRIVER_VERSION Struct Reference

Driver Version.

```
#include <Driver_Common.h>
```

Data Fields

- uint16_t api
  API version.
- uint16_t drv
  Driver version.

5.4.1 Detailed Description

Driver Version.

5.4.2 Field Documentation

Kneron Confidential

5.4.2.1 api

```
uint16_t _ARM_DRIVER_VERSION::api
```

API version.

5.4.2.2 drv

```
uint16_t _ARM_DRIVER_VERSION::drv
```

Driver version.

The documentation for this struct was generated from the following file:

- Driver_Common.h

5.5 _ARM_USART_CAPABILITIES Struct Reference

USART Device Driver Capabilities.

```
#include <Driver_USART.h>
```

Data Fields

- uint32_t asynchronous: 1
  supports UART (Asynchronous) mode
- uint32_t synchronous_master: 1
  supports Synchronous Master mode
- uint32_t synchronous_slave: 1
  supports Synchronous Slave mode
- uint32_t single_wire: 1
  supports UART Single-wire mode
- uint32_t irda: 1
  supports UART IrDA mode
- uint32_t smart_card: 1
  supports UART Smart Card mode
- uint32_t smart_card_clock: 1
  Smart Card Clock generator available.
- uint32_t flow_control_rts: 1
  RTS Flow Control available.
- uint32_t flow_control_cts: 1
  CTS Flow Control available.
- uint32_t event_tx_complete: 1
  Transmit completed event: ARM_USART_EVENT_TX_COMPLETE.
- uint32_t event_rx_timeout: 1
  Signal receive character timeout event: ARM_USART_EVENT_RX_TIMEOUT.

Kneron Confidential

- uint32_t rts: 1

  RTS Line: 0=not available, 1=available.
- uint32_t cts: 1

  CTS Line: 0=not available, 1=available.
- uint32_t dtr: 1

  DTR Line: 0=not available, 1=available.
- uint32_t dsr: 1

  DSR Line: 0=not available, 1=available.
- uint32_t dcd: 1

  DCD Line: 0=not available, 1=available.
- uint32_t ri: 1

  RI Line: 0=not available, 1=available.
- uint32_t event_cts: 1

  Signal CTS change event: ARM_USART_EVENT_CTS.
- uint32_t event_dsr: 1

  Signal DSR change event: ARM_USART_EVENT_DSR.
- uint32_t event_dcd: 1

  Signal DCD change event: ARM_USART_EVENT_DCD.
- uint32_t event_ri: 1

  Signal RI change event: ARM_USART_EVENT_RI.

5.5.1 Detailed Description

USART Device Driver Capabilities.

5.5.2 Field Documentation

5.5.2.1 asynchronous

```
uint32_t _ARM_USART_CAPABILITIES::asynchronous
```

supports UART (Asynchronous) mode

5.5.2.2 cts

```
uint32_t _ARM_USART_CAPABILITIES::cts
```

CTS Line: 0=not available, 1=available.

Kneron Confidential

5.5.2.3 dcd

```
uint32_t _ARM_USART_CAPABILITIES::dcd
```

DCD Line: 0=not available, 1=available.

5.5.2.4 dsr

```
uint32_t _ARM_USART_CAPABILITIES::dsr
```

DSR Line: 0=not available, 1=available.

5.5.2.5 dtr

```
uint32_t _ARM_USART_CAPABILITIES::dtr
```

DTR Line: 0=not available, 1=available.

5.5.2.6 event_cts

```
uint32_t _ARM_USART_CAPABILITIES::event_cts
```

Signal CTS change event: ARM_USART_EVENT_CTS.

5.5.2.7 event_dcd

```
uint32_t _ARM_USART_CAPABILITIES::event_dcd
```

Signal DCD change event: ARM_USART_EVENT_DCD.

5.5.2.8 event_dsr

```
uint32_t _ARM_USART_CAPABILITIES::event_dsr
```

Signal DSR change event: ARM_USART_EVENT_DSR.

Kneron Confidential

5.5.2.9 event_ri

```
uint32_t _ARM_USART_CAPABILITIES::event_ri
```

Signal RI change event: ARM_USART_EVENT_RI.

5.5.2.10 event_rx_timeout

```
uint32_t _ARM_USART_CAPABILITIES::event_rx_timeout
```

Signal receive character timeout event: ARM_USART_EVENT_RX_TIMEOUT.

5.5.2.11 event_tx_complete

```
uint32_t _ARM_USART_CAPABILITIES::event_tx_complete
```

Transmit completed event: ARM_USART_EVENT_TX_COMPLETE.

5.5.2.12 flow_control_cts

```
uint32_t _ARM_USART_CAPABILITIES::flow_control_cts
```

CTS Flow Control available.

5.5.2.13 flow_control_rts

```
uint32_t _ARM_USART_CAPABILITIES::flow_control_rts
```

RTS Flow Control available.

5.5.2.14 irda

```
uint32_t _ARM_USART_CAPABILITIES::irda
```

supports UART IrDA mode

Kneron Confidential

5.5.2.15 ri

```
uint32_t _ARM_USART_CAPABILITIES::ri
```

RI Line: 0=not available, 1=available.

5.5.2.16 rts

```
uint32_t _ARM_USART_CAPABILITIES::rts
```

RTS Line: 0=not available, 1=available.

5.5.2.17 single_wire

```
uint32_t _ARM_USART_CAPABILITIES::single_wire
```

supports UART Single-wire mode

5.5.2.18 smart_card

```
uint32_t _ARM_USART_CAPABILITIES::smart_card
```

supports UART Smart Card mode

5.5.2.19 smart_card_clock

```
uint32_t _ARM_USART_CAPABILITIES::smart_card_clock
```

Smart Card Clock generator available.

5.5.2.20 synchronous_master

```
uint32_t _ARM_USART_CAPABILITIES::synchronous_master
```

supports Synchronous Master mode

Kneron Confidential

5.5.2.21 synchronous_slave

```
uint32_t _ARM_USART_CAPABILITIES::synchronous_slave
```

supports Synchronous Slave mode

The documentation for this struct was generated from the following file:

- Driver_USART.h

5.6 _ARM_USART_MODEM_STATUS Struct Reference

USART Modem Status.

```
#include <Driver_USART.h>
```

Data Fields

- uint32_t cts: 1
  CTS state: 1=Active, 0=Inactive.
- uint32_t dsr: 1
  DSR state: 1=Active, 0=Inactive.
- uint32_t dcd: 1
  DCD state: 1=Active, 0=Inactive.
- uint32_t ri: 1
  RI state: 1=Active, 0=Inactive.

5.6.1 Detailed Description

USART Modem Status.

5.6.2 Field Documentation

5.6.2.1 cts

```
uint32_t _ARM_USART_MODEM_STATUS::cts
```

CTS state: 1=Active, 0=Inactive.

Kneron Confidential

5.6.2.2 dcd

```
uint32_t _ARM_USART_MODEM_STATUS::dcd
```

DCD state: 1=Active, 0=Inactive.

5.6.2.3 dsr

```
uint32_t _ARM_USART_MODEM_STATUS::dsr
```

DSR state: 1=Active, 0=Inactive.

5.6.2.4 ri

```
uint32_t _ARM_USART_MODEM_STATUS::ri
```

RI state: 1=Active, 0=Inactive.

The documentation for this struct was generated from the following file:

- Driver_USART.h

5.7 _ARM_USART_STATUS Struct Reference

USART Status.

```
#include <Driver_USART.h>
```

Data Fields

- uint32_t tx_busy: 1
  Transmitter busy flag.
- uint32_t rx_busy: 1
  Receiver busy flag.
- uint32_t tx_underflow: 1
  Transmit data underflow detected (cleared on start of next send operation)
- uint32_t rx_overflow: 1
  Receive data overflow detected (cleared on start of next receive operation)
- uint32_t rx_break: 1
  Break detected on receive (cleared on start of next receive operation)
- uint32_t rx_framing_error: 1
  Framing error detected on receive (cleared on start of next receive operation)
- uint32_t rx_parity_error: 1
  Parity error detected on receive (cleared on start of next receive operation)

Kneron Confidential

5.7.1 Detailed Description

USART Status.

5.7.2 Field Documentation

5.7.2.1 rx_break

```
uint32_t _ARM_USART_STATUS::rx_break
```

Break detected on receive (cleared on start of next receive operation)

5.7.2.2 rx_busy

```
uint32_t _ARM_USART_STATUS::rx_busy
```

Receiver busy flag.

5.7.2.3 rx_framing_error

```
uint32_t _ARM_USART_STATUS::rx_framing_error
```

Framing error detected on receive (cleared on start of next receive operation)

5.7.2.4 rx_overflow

```
uint32_t _ARM_USART_STATUS::rx_overflow
```

Receive data overflow detected (cleared on start of next receive operation)

5.7.2.5 rx_parity_error

```
uint32_t _ARM_USART_STATUS::rx_parity_error
```

Parity error detected on receive (cleared on start of next receive operation)

Kneron Confidential

5.7.2.6 tx_busy

```
uint32_t _ARM_USART_STATUS::tx_busy
```

Transmitter busy flag.

5.7.2.7 tx_underflow

```
uint32_t _ARM_USART_STATUS::tx_underflow
```

Transmit data underflow detected (cleared on start of next send operation)

The documentation for this struct was generated from the following file:

- Driver_USART.h

5.8 _ARM_USBH_CAPABILITIES Struct Reference

USB Host Driver Capabilities.

```
#include <Driver_USBH.h>
```

Data Fields

- uint32_t port_mask: 15
  Root HUB available Ports Mask.
- uint32_t auto_split: 1
  Automatic SPLIT packet handling.
- uint32_t event_connect: 1
  Signal Connect event.
- uint32_t event_disconnect: 1
  Signal Disconnect event.
- uint32_t event_overcurrent: 1
  Signal Overcurrent event.
- uint32_t reserved: 13
  Reserved (must be zero)

5.8.1 Detailed Description

USB Host Driver Capabilities.

5.8.2 Field Documentation

Kneron Confidential

5.8.2.1 auto_split

`uint32_t _ARM_USBH_CAPABILITIES::auto_split`

Automatic SPLIT packet handling.

5.8.2.2 event_connect

`uint32_t _ARM_USBH_CAPABILITIES::event_connect`

Signal Connect event.

5.8.2.3 event_disconnect

`uint32_t _ARM_USBH_CAPABILITIES::event_disconnect`

Signal Disconnect event.

5.8.2.4 event_overcurrent

`uint32_t _ARM_USBH_CAPABILITIES::event_overcurrent`

Signal Overcurrent event.

5.8.2.5 port_mask

`uint32_t _ARM_USBH_CAPABILITIES::port_mask`

Root HUB available Ports Mask.

5.8.2.6 reserved

`uint32_t _ARM_USBH_CAPABILITIES::reserved`

Reserved (must be zero)

The documentation for this struct was generated from the following file:

- Driver_USBH.h

Kneron Confidential

5.9 _ARM_USBH_HCI_CAPABILITIES Struct Reference

USB Host HCI (OHCI/EHCI) Driver Capabilities.

```
#include <Driver_USBH.h>
```

Data Fields

- uint32_t port_mask: 15
  Root HUB available Ports Mask.
- uint32_t reserved: 17
  Reserved (must be zero)

5.9.1 Detailed Description

USB Host HCI (OHCI/EHCI) Driver Capabilities.

5.9.2 Field Documentation

5.9.2.1 port_mask

```
uint32_t _ARM_USBH_HCI_CAPABILITIES::port_mask
```

Root HUB available Ports Mask.

5.9.2.2 reserved

```
uint32_t _ARM_USBH_HCI_CAPABILITIES::reserved
```

Reserved (must be zero)

The documentation for this struct was generated from the following file:

- Driver_USBH.h

5.10 _ARM_USBH_PORT_STATE Struct Reference

USB Host Port State.

```
#include <Driver_USBH.h>
```

Kneron Confidential

Data Fields

- uint32_t connected: 1

  USB Host Port connected flag.
- uint32_t overcurrent: 1

  USB Host Port overcurrent flag.
- uint32_t speed: 2

  USB Host Port speed setting (ARM_USB_SPEED_xxx)
- uint32_t reserved: 28

5.10.1 Detailed Description

USB Host Port State.

5.10.2 Field Documentation

5.10.2.1 connected

```
uint32_t _ARM_USBH_PORT_STATE::connected
```

USB Host Port connected flag.

5.10.2.2 overcurrent

```
uint32_t _ARM_USBH_PORT_STATE::overcurrent
```

USB Host Port overcurrent flag.

5.10.2.3 reserved

```
uint32_t _ARM_USBH_PORT_STATE::reserved
```

5.10.2.4 speed

```
uint32_t _ARM_USBH_PORT_STATE::speed
```

USB Host Port speed setting (ARM_USB_SPEED_xxx)

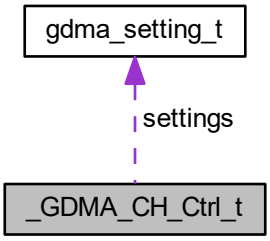
The documentation for this struct was generated from the following file:

- Driver_USBH.h

Kneron Confidential

5.11 _GDMA_CH_Ctrl_t Struct Reference

Collaboration diagram for _GDMA_CH_Ctrl_t:



Data Fields

- gdma_setting_t settings
- uint8_t in_use
- uint8_t running
- osThreadId_t user_tid
- gdma_xfer_callback_t user_cb
- void * user_arg
- uint32_t src_end
- uint32_t dst_end
- kdrv_status_t ret_sts

5.11.1 Field Documentation

5.11.1.1 dst_end

```
uint32_t _GDMA_CH_Ctrl_t::dst_end
```

5.11.1.2 in_use

```
uint8_t _GDMA_CH_Ctrl_t::in_use
```

Kneron Confidential

5.11.1.3 ret_sts

kdrv_status_t _GDMA_CH_Ctrl_t::ret_sts

5.11.1.4 running

uint8_t _GDMA_CH_Ctrl_t::running

5.11.1.5 settings

gdma_setting_t _GDMA_CH_Ctrl_t::settings

5.11.1.6 src_end

uint32_t _GDMA_CH_Ctrl_t::src_end

5.11.1.7 user_arg

void* _GDMA_CH_Ctrl_t::user_arg

5.11.1.8 user_cb

gdma_xfer_callback_t _GDMA_CH_Ctrl_t::user_cb

5.11.1.9 user_tid

osThreadId_t _GDMA_CH_Ctrl_t::user_tid

The documentation for this struct was generated from the following file:

- kdrv_gdma.c

Kneron Confidential

5.12 _UVC_FRAME_LINK Struct Reference

Collaboration diagram for _UVC_FRAME_LINK:

Data Fields

- uint32_t * user_frame
- uint32_t size
- uint8_t in_use
- struct _UVC_FRAME_LINK * next

5.12.1 Field Documentation

5.12.1.1 in_use

```
uint8_t _UVC_FRAME_LINK::in_use
```

5.12.1.2 next

```
struct _UVC_FRAME_LINK* _UVC_FRAME_LINK::next
```

5.12.1.3 size

```
uint32_t _UVC_FRAME_LINK::size
```

Kneron Confidential

5.12.1.4 user_frame

```
uint32_t* _UVC_FRAME_LINK::user_frame
```

The documentation for this struct was generated from the following file:

- kmdw_uvc.c

5.13 Buffer_Page_0_DWord Struct Reference

Data Fields

- uint32_t dev_addr: 7
- uint32_t reserved: 1
- uint32_t endpoint: 4
- uint32_t ptr_address: 20

5.13.1 Field Documentation

5.13.1.1 dev_addr

```
uint32_t Buffer_Page_0_DWord::dev_addr
```

5.13.1.2 endpoint

```
uint32_t Buffer_Page_0_DWord::endpoint
```

5.13.1.3 ptr_address

```
uint32_t Buffer_Page_0_DWord::ptr_address
```

5.13.1.4 reserved

```
uint32_t Buffer_Page_0_DWord::reserved
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

Kneron Confidential

5.14 Buffer_Page_1_DWord Struct Reference

Data Fields

- uint32_t max_packet_size: 11
- uint32_t direction: 1
- uint32_t ptr_address: 20

5.14.1 Field Documentation

5.14.1.1 direction

```
uint32_t Buffer_Page_1_DWord::direction
```

5.14.1.2 max_packet_size

```
uint32_t Buffer_Page_1_DWord::max_packet_size
```

5.14.1.3 ptr_address

```
uint32_t Buffer_Page_1_DWord::ptr_address
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

5.15 Buffer_Page_2_DWord Struct Reference

Data Fields

- uint32_t mult: 2
- uint32_t reserved: 10
- uint32_t ptr_address: 20

5.15.1 Field Documentation

Kneron Confidential

5.15.1.1 mult

```
uint32_t Buffer_Page_2_DWord::mult
```

5.15.1.2 ptr_address

```
uint32_t Buffer_Page_2_DWord::ptr_address
```

5.15.1.3 reserved

```
uint32_t Buffer_Page_2_DWord::reserved
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

5.16 Buffer_Page_DWord Struct Reference

Data Fields

- uint32_t misc: 12
- uint32_t ptr_address: 20

5.16.1 Field Documentation

5.16.1.1 misc

```
uint32_t Buffer_Page_DWord::misc
```

5.16.1.2 ptr_address

```
uint32_t Buffer_Page_DWord::ptr_address
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

Kneron Confidential

5.17 Buffer_Pointer_DWord Struct Reference

Data Fields

- uint32_t cur_offset: 12
- uint32_t ptr_address: 20

5.17.1 Field Documentation

5.17.1.1 cur_offset

```
uint32_t Buffer_Pointer_DWord::cur_offset
```

5.17.1.2 ptr_address

```
uint32_t Buffer_Pointer_DWord::ptr_address
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

5.18 cam_capability Struct Reference

```
#include <kmdw_camera.h>
```

Data Fields

- char driver [16]
- char desc [16]
- uint32_t version
- uint32_t capabilities

The documentation for this struct was generated from the following file:

- kmdw_camera.h

5.19 cam_format Struct Reference

```
#include <kmdw_camera.h>
```

Kneron Confidential

Data Fields

- uint32_t width
- uint32_t height
- uint32_t pixelformat
- uint32_t field
- uint32_t bytesperline
- uint32_t sizeimage
- uint32_t colorspace

The documentation for this struct was generated from the following file:

- kmdw_camera.h

5.20 cam_ops Struct Reference

```
#include <kmdw_camera.h>
```

Data Fields

- kmdw_status_t(* open)(uint32_t cam_idx)
- kmdw_status_t(* close)(uint32_t cam_idx)
- kmdw_status_t(* set_format)(uint32_t cam_idx, struct cam_format *format)
- kmdw_status_t(* get_format)(uint32_t cam_idx, struct cam_format *format)
- kmdw_status_t(* buffer_init)(uint32_t cam_idx)
- kmdw_status_t(* start_capture)(uint32_t cam_idx)
- kmdw_status_t(* stop_capture)(uint32_t cam_idx)
- kmdw_status_t(* buffer_prepare)(uint32_t cam_idx)
- kmdw_status_t(* buffer_capture)(uint32_t cam_idx, uint32_t *addr, uint32_t *size)
- kmdw_status_t(* stream_on)(uint32_t cam_idx)
- kmdw_status_t(* stream_off)(uint32_t cam_idx)
- kmdw_status_t(* query_capability)(uint32_t cam_idx, struct cam_capability *cap)
- kmdw_status_t(* set_gain)(uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
- kmdw_status_t(* set_aec)(uint32_t cam_idx, struct cam_sensor_aec *aec_p)
- kmdw_status_t(* set_exp_time)(uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
- kmdw_status_t(* get_lux)(uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain, uint8_t *global_gain, uint8_t *y_average)
- kmdw_status_t(* led_switch)(uint32_t cam_idx, uint32_t on)
- kmdw_status_t(* set_mirror)(uint32_t cam_idx, uint32_t enable)
- kmdw_status_t(* set_flip)(uint32_t cam_idx, uint32_t enable)
- uint32_t(* get_device_id)(uint32_t cam_idx)
- kmdw_status_t(* ioctl)(uint32_t cam_idx, uint32_t cid, void *data, uint16_t len)

The documentation for this struct was generated from the following file:

- kmdw_camera.h

Kneron Confidential

5.21 cam_sensor_aec Struct Reference

`#include <kmdw_camera.h>`

Data Fields

- uint8_t x1
- uint8_t x2
- uint8_t y1
- uint8_t y2
- uint8_t center_x1
- uint8_t center_x2
- uint8_t center_y1
- uint8_t center_y2

The documentation for this struct was generated from the following file:

- kmdw_camera.h

5.22 cont_frame_intervals Struct Reference

`#include <uvc_dev.h>`

Data Fields

- uint32_t dwMinFrameInterval
- uint32_t dwMaxFrameInterval
- uint32_t dwFrameIntervalStep

5.22.1 Field Documentation

5.22.1.1 dwFrameIntervalStep

`uint32_t cont_frame_intervals::dwFrameIntervalStep`

5.22.1.2 dwMaxFrameInterval

`uint32_t cont_frame_intervals::dwMaxFrameInterval`

Kneron Confidential

5.22.1.3 dwMinFrameInterval

```
uint32_t cont_frame_intervals::dwMinFrameInterval
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.23 ct_aem Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum aem_req req
- uint8_t caps
- uint8_t bAutoExposureMode

5.23.1 Field Documentation

5.23.1.1 bAutoExposureMode

```
uint8_t ct_aem::bAutoExposureMode
```

5.23.1.2 caps

```
uint8_t ct_aem::caps
```

5.23.1.3 req

```
enum aem_req ct_aem::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

Kneron Confidential

5.24 ct_aep Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum aep_req req
- uint8_t caps
- uint8_t bAutoExposurePriority

5.24.1 Field Documentation

5.24.1.1 bAutoExposurePriority

`uint8_t ct_aep::bAutoExposurePriority`

5.24.1.2 caps

`uint8_t ct_aep::caps`

5.24.1.3 req

`enum aep_req ct_aep::req`

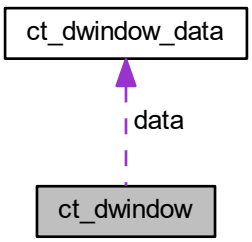
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.25 ct_dwindow Struct Reference

`#include <uvc_camera.h>`

Collaboration diagram for ct_dwindow:



Kneron Confidential

Data Fields

- enum dwindow_req req
- struct ct_dwindow_data data

5.25.1 Field Documentation

5.25.1.1 data

```
struct ct_dwindow_data ct_dwindow::data
```

5.25.1.2 req

```
enum dwindow_req ct_dwindow::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.26 ct_dwindow_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- uint16_t wWindow_Top
- uint16_t wWindow_Left
- uint16_t wWindow_Bottom
- uint16_t wWindow_Right
- uint16_t wNumSteps
- uint16_t bmNumStepsUnits

5.26.1 Field Documentation

5.26.1.1 bmNumStepsUnits

```
uint16_t ct_dwindow_data::bmNumStepsUnits
```

Kneron Confidential

5.26.1.2 wNumSteps

```
uint16_t ct_dwindow_data::wNumSteps
```

5.26.1.3 wWindow_Bottom

```
uint16_t ct_dwindow_data::wWindow_Bottom
```

5.26.1.4 wWindow_Left

```
uint16_t ct_dwindow_data::wWindow_Left
```

5.26.1.5 wWindow_Right

```
uint16_t ct_dwindow_data::wWindow_Right
```

5.26.1.6 wWindow_Top

```
uint16_t ct_dwindow_data::wWindow_Top
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.27 ct_eta Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum eta_req req
- uint8_t caps
- uint32_t bExposureTimeAbsolute

5.27.1 Field Documentation

Kneron Confidential

5.27.1.1 bExposureTimeAbsolute

```
uint32_t ct_eta::bExposureTimeAbsolute
```

5.27.1.2 caps

```
uint8_t ct_eta::caps
```

5.27.1.3 req

```
enum eta_req ct_eta::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.28 ct_etr Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum etr_req req
- uint8_t caps
- int8_t bExposureTimeRelative

5.28.1 Field Documentation

5.28.1.1 bExposureTimeRelative

```
int8_t ct_etr::bExposureTimeRelative
```

5.28.1.2 caps

```
uint8_t ct_etr::caps
```

Kneron Confidential

5.28.1.3 req

```
enum etr_req ct_etr::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.29 ct_fauto Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum fauto_req req
- uint8_t caps
- uint8_t bFocusAuto

5.29.1 Field Documentation

5.29.1.1 bFocusAuto

```
uint8_t ct_fauto::bFocusAuto
```

5.29.1.2 caps

```
uint8_t ct_fauto::caps
```

5.29.1.3 req

```
enum fauto_req ct_fauto::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

Kneron Confidential

5.30 ct_focus_a Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum focus_a_req req
- uint8_t caps
- uint16_t wFocusAbsolute

5.30.1 Field Documentation

5.30.1.1 caps

`uint8_t ct_focus_a::caps`

5.30.1.2 req

`enum focus_a_req ct_focus_a::req`

5.30.1.3 wFocusAbsolute

`uint16_t ct_focus_a::wFocusAbsolute`

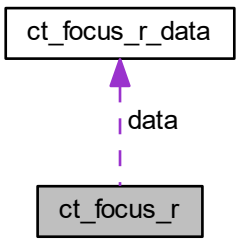
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.31 ct_focus_r Struct Reference

`#include <uvc_camera.h>`

Collaboration diagram for ct_focus_r:



Kneron Confidential

Data Fields

- enum fr_req req
- uint8_t caps
- struct ct_focus_r_data data

5.31.1 Field Documentation

5.31.1.1 caps

```
uint8_t ct_focus_r::caps
```

5.31.1.2 data

```
struct ct_focus_r_data ct_focus_r::data
```

5.31.1.3 req

```
enum fr_req ct_focus_r::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.32 ct_focus_r_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- int8_t bFocusRelative
- uint8_t bSpeed

5.32.1 Field Documentation

Kneron Confidential

5.32.1.1 bFocusRelative

```
int8_t ct_focus_r_data::bFocusRelative
```

5.32.1.2 bSpeed

```
uint8_t ct_focus_r_data::bSpeed
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.33 ct_focus_sr Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum fsr_req req
- uint8_t caps
- uint8_t bFocus

5.33.1 Field Documentation

5.33.1.1 bFocus

```
uint8_t ct_focus_sr::bFocus
```

5.33.1.2 caps

```
uint8_t ct_focus_sr::caps
```

Kneron Confidential

5.33.1.3 req

```
enum fsr_req ct_focus_sr::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.34 ct_iris_a Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum irisa_req req
- uint8_t caps
- uint16_t wIrisAbsolute

5.34.1 Field Documentation

5.34.1.1 caps

```
uint8_t ct_iris_a::caps
```

5.34.1.2 req

```
enum irisa_req ct_iris_a::req
```

5.34.1.3 wIrisAbsolute

```
uint16_t ct_iris_a::wIrisAbsolute
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

Kneron Confidential

5.35 ct_iris_r Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum irisr_req req
- uint8_t caps
- uint8_t bIrisRelative

5.35.1 Field Documentation

5.35.1.1 bIrisRelative

`uint8_t ct_iris_r::bIrisRelative`

5.35.1.2 caps

`uint8_t ct_iris_r::caps`

5.35.1.3 req

`enum irisr_req ct_iris_r::req`

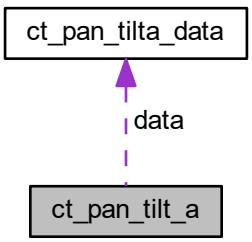
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.36 ct_pan_tilt_a Struct Reference

`#include <uvc_camera.h>`

Collaboration diagram for ct_pan_tilt_a:



Kneron Confidential

Data Fields

- enum pantilta_req req
- uint8_t caps
- struct ct_pan_tilta_data data

5.36.1 Field Documentation

5.36.1.1 caps

```
uint8_t ct_pan_tilt_a::caps
```

5.36.1.2 data

```
struct ct_pan_tilta_data ct_pan_tilt_a::data
```

5.36.1.3 req

```
enum pantilta_req ct_pan_tilt_a::req
```

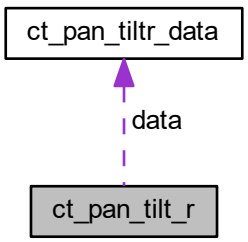
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.37 ct_pan_tilt_r Struct Reference

```
#include <uvc_camera.h>
```

Collaboration diagram for ct_pan_tilt_r:



Kneron Confidential

Data Fields

- enum pantiltr_req req
- uint8_t caps
- struct ct_pan_tiltr_data data

5.37.1 Field Documentation

5.37.1.1 caps

```
uint8_t ct_pan_tilt_r::caps
```

5.37.1.2 data

```
struct ct_pan_tiltr_data ct_pan_tilt_r::data
```

5.37.1.3 req

```
enum pantiltr_req ct_pan_tilt_r::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.38 ct_pan_tilta_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- int32_t dwPanAbsolute
- int32_t dwTiltAbsolute

5.38.1 Field Documentation

Kneron Confidential

5.38.1.1 dwPanAbsolute

```
int32_t ct_pan_tilta_data::dwPanAbsolute
```

5.38.1.2 dwTiltAbsolute

```
int32_t ct_pan_tilta_data::dwTiltAbsolute
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.39 ct_pan_tiltr_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- int8_t bPanRelative
- uint8_t bPanSpeed
- int8_t bTiltRelative
- uint8_t bTiltSpeed

5.39.1 Field Documentation

5.39.1.1 bPanRelative

```
int8_t ct_pan_tiltr_data::bPanRelative
```

5.39.1.2 bPanSpeed

```
uint8_t ct_pan_tiltr_data::bPanSpeed
```

Kneron Confidential

5.39.1.3 bTiltRelative

```
int8_t ct_pan_tiltr_data::bTiltRelative
```

5.39.1.4 bTiltSpeed

```
uint8_t ct_pan_tiltr_data::bTiltSpeed
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.40 ct_privacy_shutter Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum ps_req req
- uint8_t caps
- bool bPrivacy

5.40.1 Field Documentation

5.40.1.1 bPrivacy

```
bool ct_privacy_shutter::bPrivacy
```

5.40.1.2 caps

```
uint8_t ct_privacy_shutter::caps
```

Kneron Confidential

5.40.1.3 req

```
enum ps_req ct_privacy_shutter::req
```

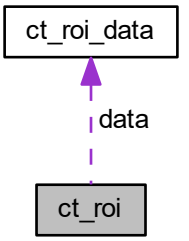
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.41 ct_roi Struct Reference

```
#include <uvc_camera.h>
```

Collaboration diagram for ct_roi:



Data Fields

- enum roi_req req
- struct ct_roi_data data

5.41.1 Field Documentation

5.41.1.1 data

```
struct ct_roi_data ct_roi::data
```

Kneron Confidential

5.41.1.2 req

```
enum roi_req ct_roi::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.42 ct_roi_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- uint16_t wROI_Top
- uint16_t wROI_Left
- uint16_t wROI_Bottom
- uint16_t wROI_Right
- uint16_t bmAutoControls

5.42.1 Field Documentation

5.42.1.1 bmAutoControls

```
uint16_t ct_roi_data::bmAutoControls
```

5.42.1.2 wROI_Bottom

```
uint16_t ct_roi_data::wROI_Bottom
```

5.42.1.3 wROI_Left

```
uint16_t ct_roi_data::wROI_Left
```

Kneron Confidential

5.42.1.4 wROI_Right

```
uint16_t ct_roi_data::wROI_Right
```

5.42.1.5 wROI_Top

```
uint16_t ct_roi_data::wROI_Top
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.43 ct_roll_a Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum rolla_req req
- uint8_t caps
- int16_t wAbsolute

5.43.1 Field Documentation

5.43.1.1 caps

```
uint8_t ct_roll_a::caps
```

5.43.1.2 req

```
enum rolla_req ct_roll_a::req
```

Kneron Confidential

5.43.1.3 wAbsolute

```
int16_t ct_roll_a::wAbsolute
```

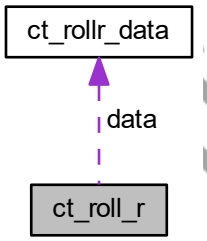
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.44 ct_roll_r Struct Reference

```
#include <uvc_camera.h>
```

Collaboration diagram for ct_roll_r:



Data Fields

- enum rollr_req req
- uint8_t caps
- struct ct_rollr_data data

5.44.1 Field Documentation

5.44.1.1 caps

```
uint8_t ct_roll_r::caps
```

Kneron Confidential

5.44.1.2 data

```
struct ct_rollr_data ct_roll_r::data
```

5.44.1.3 req

```
enum rollr_req ct_roll_r::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.45 ct_rollr_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- int8_t bRollRelative
- uint8_t bSpeed

5.45.1 Field Documentation

5.45.1.1 bRollRelative

```
int8_t ct_rollr_data::bRollRelative
```

5.45.1.2 bSpeed

```
uint8_t ct_rollr_data::bSpeed
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.46 ct_scm Struct Reference

```
#include <uvc_camera.h>
```

Kneron Confidential

Data Fields

- enum scm_req req
- uint8_t caps
- bool bScanningMode

5.46.1 Field Documentation

5.46.1.1 bScanningMode

```
bool ct_scm::bScanningMode
```

5.46.1.2 caps

```
uint8_t ct_scm::caps
```

5.46.1.3 req

```
enum scm_req ct_scm::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.47 ct_zoom_a Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum zooma_req req
- uint8_t caps
- uint16_t wObjectiveFocalLength

5.47.1 Field Documentation

Kneron Confidential

5.47.1.1 caps

```
uint8_t ct_zoom_a::caps
```

5.47.1.2 req

```
enum zooma_req ct_zoom_a::req
```

5.47.1.3 wObjectiveFocalLength

```
uint16_t ct_zoom_a::wObjectiveFocalLength
```

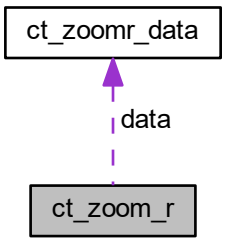
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.48 ct_zoom_r Struct Reference

```
#include <uvc_camera.h>
```

Collaboration diagram for ct_zoom_r:



Data Fields

- enum zoomr_req req
- uint8_t caps
- struct ct_zoomr_data data

Kneron Confidential

5.48.1 Field Documentation

5.48.1.1 caps

```
uint8_t ct_zoom_r::caps
```

5.48.1.2 data

```
struct ct_zoomr_data ct_zoom_r::data
```

5.48.1.3 req

```
enum zoomr_req ct_zoom_r::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.49 ct_zoomr_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- int8_t bZoom
- bool bDigitalZoom
- uint16_t bSpeed

5.49.1 Field Documentation

5.49.1.1 bDigitalZoom

```
bool ct_zoomr_data::bDigitalZoom
```

Kneron Confidential

5.49.1.2 bSpeed

```
uint16_t ct_zoomr_data::bSpeed
```

5.49.1.3 bZoom

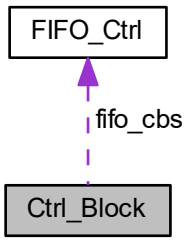
```
int8_t ct_zoomr_data::bZoom
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.50 Ctrl_Block Struct Reference

Collaboration diagram for Ctrl_Block:



Data Fields

- osThreadId_t notifyTid
- osEventFlagsId_t evt_id
- uint32_t notifyFlag
- bool ep0_halted
- kdrv_usbd_device_descriptor_t * dev_desc
- kdrv_usbd_device_qualifier_descriptor_t * dev_qual_desc
- uint32_t config_state
- FIFO_Ctrl fifo_cbs [FIFO_NUM]

5.50.1 Field Documentation

Kneron Confidential

5.50.1.1 config_state

```
uint32_t Ctrl_Block::config_state
```

5.50.1.2 dev_desc

```
kdrv_usbd_device_descriptor_t* Ctrl_Block::dev_desc
```

5.50.1.3 dev_qual_desc

```
kdrv_usbd_device_qualifier_descriptor_t* Ctrl_Block::dev_qual_desc
```

5.50.1.4 ep0_halted

```
bool Ctrl_Block::ep0_halted
```

5.50.1.5 evt_id

```
osEventFlagsId_t Ctrl_Block::evt_id
```

5.50.1.6 fifo_cbs

```
FIFO_Ctrl Ctrl_Block::fifo_cbs[FIFO_NUM]
```

5.50.1.7 notifyFlag

```
uint32_t Ctrl_Block::notifyFlag
```

Kneron Confidential

5.50.1.8 notifyTid

```
osThreadId_t Ctrl_Block::notifyTid
```

The documentation for this struct was generated from the following file:

- kdrv_usbd.c

5.51 ctrl_info Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint32_t cid
- uint8_t eid
- uint8_t cs
- bool supported
- bool cached
- uint8_t para_size
- uint8_t * para
- uint8_t caps
- uint8_t len
- uint8_t ctl_flag

5.51.1 Field Documentation

5.51.1.1 cached

```
bool ctrl_info::cached
```

5.51.1.2 caps

```
uint8_t ctrl_info::caps
```

5.51.1.3 cid

```
uint32_t ctrl_info::cid
```

Kneron Confidential

5.51.1.4 cs

```
uint8_t ctrl_info::cs
```

5.51.1.5 ctl_flag

```
uint8_t ctrl_info::ctl_flag
```

5.51.1.6 eid

```
uint8_t ctrl_info::eid
```

5.51.1.7 len

```
uint8_t ctrl_info::len
```

5.51.1.8 para

```
uint8_t* ctrl_info::para
```

5.51.1.9 para_size

```
uint8_t ctrl_info::para_size
```

5.51.1.10 supported

```
bool ctrl_info::supported
```

The documentation for this struct was generated from the following file:

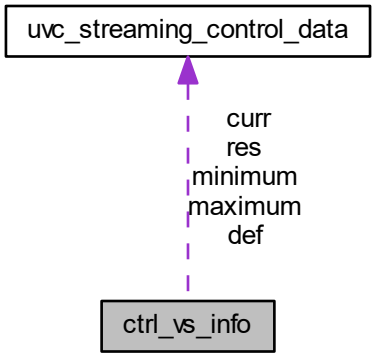
- uvc_dev.h

Kneron Confidential

5.52 ctrl_vs_info Struct Reference

`#include <uvc_dev.h>`

Collaboration diagram for ctrl_vs_info:



Data Fields

- bool cached
- struct uvc_streaming_control_data * def
- struct uvc_streaming_control_data * curr
- struct uvc_streaming_control_data * minimum
- struct uvc_streaming_control_data * maximum
- struct uvc_streaming_control_data * res
- uint8_t caps
- uint16_t len

5.52.1 Field Documentation

5.52.1.1 cached

`bool ctrl_vs_info::cached`

5.52.1.2 caps

`uint8_t ctrl_vs_info::caps`

Kneron Confidential

5.52.1.3 curr

```
struct uvc_streaming_control_data* ctrl_vs_info::curr
```

5.52.1.4 def

```
struct uvc_streaming_control_data* ctrl_vs_info::def
```

5.52.1.5 len

```
uint16_t ctrl_vs_info::len
```

5.52.1.6 maximum

```
struct uvc_streaming_control_data* ctrl_vs_info::maximum
```

5.52.1.7 minimum

```
struct uvc_streaming_control_data* ctrl_vs_info::minimum
```

5.52.1.8 res

```
struct uvc_streaming_control_data* ctrl_vs_info::res
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.53 Device_Address_DWord Struct Reference

Data Fields

- uint32_t dev_addr: 7
- uint32_t inactivate: 1
- uint32_t endpoint: 4
- uint32_t endpt_speed: 2
- uint32_t dtc: 1
- uint32_t H_reclamation: 1
- uint32_t max_packet_length: 11
- uint32_t C_endpt_flag: 1
- uint32_t RL_Nak_count: 4

Kneron Confidential

5.53.1 Field Documentation

5.53.1.1 C_endpt_flag

```
uint32_t Device_Address_DWord::C_endpt_flag
```

5.53.1.2 dev_addr

```
uint32_t Device_Address_DWord::dev_addr
```

5.53.1.3 dtc

```
uint32_t Device_Address_DWord::dtc
```

5.53.1.4 endpoint

```
uint32_t Device_Address_DWord::endpoint
```

5.53.1.5 endpt_speed

```
uint32_t Device_Address_DWord::endpt_speed
```

5.53.1.6 H_reclamation

```
uint32_t Device_Address_DWord::H_reclamation
```

5.53.1.7 inactivate

```
uint32_t Device_Address_DWord::inactivate
```

Kneron Confidential

5.53.1.8 max_packet_length

```
uint32_t Device_Address_DWord::max_packet_length
```

5.53.1.9 RL_Nak_count

```
uint32_t Device_Address_DWord::RL_Nak_count
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

5.54 ErrorResiliencyFeatures Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint16_t bmErrorResiliencyFeatures

5.54.1 Field Documentation

5.54.1.1 bmErrorResiliencyFeatures

```
uint16_t ErrorResiliencyFeatures::bmErrorResiliencyFeatures
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.55 eu_levelidc Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bLevelIDC

Kneron Confidential

5.55.1 Field Documentation

5.55.1.1 bLevelIDC

```
uint8_t eu_levelidc::bLevelIDC
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.56 eu_ltrbuffers Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bNumHostControlLTRBuffers
- uint8_t bTrustMode

5.56.1 Field Documentation

5.56.1.1 bNumHostControlLTRBuffers

```
uint8_t eu_ltrbuffers::bNumHostControlLTRBuffers
```

5.56.1.2 bTrustMode

```
uint8_t eu_ltrbuffers::bTrustMode
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.57 eu_ltrpicture Struct Reference

```
#include <uvc_ctrl.h>
```

Kneron Confidential

Data Fields

- uint8_t bPutAtPositionInLTRBuffer
- uint8_t bLTRMode

5.57.1 Field Documentation

5.57.1.1 bLTRMode

```
uint8_t eu_ltrpicture::bLTRMode
```

5.57.1.2 bPutAtPositionInLTRBuffer

```
uint8_t eu_ltrpicture::bPutAtPositionInLTRBuffer
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.58 eu_ltrvalidation Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint16_t bmValidLTRs

5.58.1 Field Documentation

5.58.1.1 bmValidLTRs

```
uint16_t eu_ltrvalidation::bmValidLTRs
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

5.59 eu_min_frame_interval Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint32_t dwFrameInterval

5.59.1 Field Documentation

5.59.1.1 dwFrameInterval

`uint32_t eu_min_frame_interval::dwFrameInterval`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.60 eu_priority Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint8_t bPriority

5.60.1 Field Documentation

5.60.1.1 bPriority

`uint8_t eu_priority::bPriority`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

5.61 eu_profile Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint16_t wProfile
- uint16_t wConstrainedToolset
- uint8_t bmSettings

5.61.1 Field Documentation

5.61.1.1 bmSettings

`uint8_t eu_profile::bmSettings`

5.61.1.2 wConstrainedToolset

`uint16_t eu_profile::wConstrainedToolset`

5.61.1.3 wProfile

`uint16_t eu_profile::wProfile`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.62 eu_qpprime Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint16_t wQpPrime_I
- uint16_t wQpPrime_P
- uint16_t wQpPrime_B

Kneron Confidential

5.62.1 Field Documentation

5.62.1.1 wQpPrime_B

```
uint16_t eu_qpprime::wQpPrime_B
```

5.62.1.2 wQpPrime_I

```
uint16_t eu_qpprime::wQpPrime_I
```

5.62.1.3 wQpPrime_P

```
uint16_t eu_qpprime::wQpPrime_P
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.63 eu_range Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bMinQp
- uint8_t bMaxQp

5.63.1 Field Documentation

5.63.1.1 bMaxQp

```
uint8_t eu_range::bMaxQp
```

Kneron Confidential

5.63.1.2 bMinQp

```
uint8_t eu_range::bMinQp
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.64 eu_ratecontrolmode Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bRateControlMode
- uint32_t dwAverageBitRate
- uint32_t dwCPBsize
- uint32_t dwPeakBitRate

5.64.1 Field Documentation

5.64.1.1 bRateControlMode

```
uint8_t eu_ratecontrolmode::bRateControlMode
```

5.64.1.2 dwAverageBitRate

```
uint32_t eu_ratecontrolmode::dwAverageBitRate
```

5.64.1.3 dwCPBsize

```
uint32_t eu_ratecontrolmode::dwCPBsize
```

Kneron Confidential

5.64.1.4 dwPeakBitRate

```
uint32_t eu_ratecontrolmode::dwPeakBitRate
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.65 eu_seimessages Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bmSEIMessages [8]

5.65.1 Field Documentation

5.65.1.1 bmSEIMessages

```
uint8_t eu_seimessages::bmSEIMessages[8]
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.66 eu_select_layer Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint16_t wLayerOrViewID

5.66.1 Field Documentation

Kneron Confidential

5.66.1.1 wLayerOrViewID

```
uint16_t eu_select_layer::wLayerOrViewID
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.67 eu_slicemode Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint16_t wSliceMode
- uint16_t wSliceConfigSetting

5.67.1 Field Documentation

5.67.1.1 wSliceConfigSetting

```
uint16_t eu_slicemode::wSliceConfigSetting
```

5.67.1.2 wSliceMode

```
uint16_t eu_slicemode::wSliceMode
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.68 eu_syncframe Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bSyncFrameType
- uint16_t wSyncFrameInterval
- uint8_t bGradualDecoderRefresh

Kneron Confidential

5.68.1 Field Documentation

5.68.1.1 bGradualDecoderRefresh

```
uint8_t eu_syncframe::bGradualDecoderRefresh
```

5.68.1.2 bSyncFrameType

```
uint8_t eu_syncframe::bSyncFrameType
```

5.68.1.3 wSyncFrameInterval

```
uint16_t eu_syncframe::wSyncFrameInterval
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.69 eu_videoresolution Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint16_t wWidth
- uint16_t wHeight

5.69.1 Field Documentation

5.69.1.1 wHeight

```
uint16_t eu_videoresolution::wHeight
```

Kneron Confidential

5.69.1.2 wWidth

```
uint16_t eu_videoresolution::wWidth
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.70 FIFO_Ctrl Struct Reference

Data Fields

- uint8_t enabled
- uint8_t enpNo
- uint32_t endpointAddress
- uint32_t maxPacketSize
- uint8_t transferType
- uint32_t byteCntReg
- uint8_t isTransferring
- uint32_t user_buf_addr
- uint32_t user_buf_len
- uint32_t received_length
- uint8_t short_or_zl_packet
- uint8_t isBlockingCall
- kdrv_usbd_event_name_t cur_event

5.70.1 Field Documentation

5.70.1.1 byteCntReg

```
uint32_t FIFO_Ctrl::byteCntReg
```

5.70.1.2 cur_event

```
kdrv_usbd_event_name_t FIFO_Ctrl::cur_event
```

5.70.1.3 enabled

```
uint8_t FIFO_Ctrl::enabled
```

Kneron Confidential

5.70.1.4 endpointAddress

```
uint32_t FIFO_Ctrl::endpointAddress
```

5.70.1.5 enpNo

```
uint8_t FIFO_Ctrl::enpNo
```

5.70.1.6 isBlockingCall

```
uint8_t FIFO_Ctrl::isBlockingCall
```

5.70.1.7 isTransferring

```
uint8_t FIFO_Ctrl::isTransferring
```

5.70.1.8 maxPacketSize

```
uint32_t FIFO_Ctrl::maxPacketSize
```

5.70.1.9 received_length

```
uint32_t FIFO_Ctrl::received_length
```

5.70.1.10 short_or_zl_packet

```
uint8_t FIFO_Ctrl::short_or_zl_packet
```

5.70.1.11 transferType

```
uint8_t FIFO_Ctrl::transferType
```

Kneron Confidential

5.70.1.12 user_buf_addr

```
uint32_t FIFO_Ctrl::user_buf_addr
```

5.70.1.13 user_buf_len

```
uint32_t FIFO_Ctrl::user_buf_len
```

The documentation for this struct was generated from the following file:

- kdrv_usbd.c

5.71 gdma_setting_t Struct Reference

Structure of GDMA advanced settings for a specified DMA handle (channel)

```
#include <kdrv_gdma.h>
```

Data Fields

- gdma_transfer_width_t dst_width
- gdma_transfer_width_t src_width
- gdma_burst_size_t burst_size
- gdma_address_control_t dst_addr_ctrl
- gdma_address_control_t src_addr_ctrl
- gdma_work_mode_t dma_mode
- uint32_t dma_dst_req
- uint32_t dma_src_req

5.71.1 Detailed Description

Structure of GDMA advanced settings for a specified DMA handle (channel)

The documentation for this struct was generated from the following file:

- kdrv_gdma.h

5.72 hmx2056_context Struct Reference

Data Fields

- struct v2k_subdev subdev

Kneron Confidential

5.72.1 Field Documentation

5.72.1.1 subdev

```
struct v2k_subdev hmx2056_context::subdev
```

The documentation for this struct was generated from the following file:

- kdev_sensor_hmx2056.c

5.73 hmxrica_context Struct Reference

Data Fields

- struct v2k_subdev subdev
- const struct hmxrica_datafmt * fmt

5.73.1 Field Documentation

5.73.1.1 fmt

```
const struct hmxrica_datafmt* hmxrica_context::fmt
```

5.73.1.2 subdev

```
struct v2k_subdev hmxrica_context::subdev
```

The documentation for this struct was generated from the following file:

- kdev_sensor_hmxrica.c

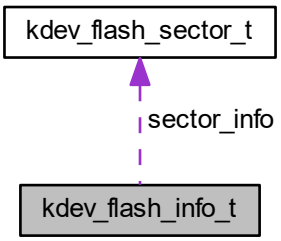
Kneron Confidential

5.74 kdev_flash_info_t Struct Reference

Flash information struct.

`#include <kdev_flash.h>`

Collaboration diagram for kdev_flash_info_t:



Data Fields

- kdev_flash_sector_t * sector_info
- uint32_t sector_count
- uint32_t sector_size
- uint32_t page_size
- uint32_t program_unit
- uint8_t erased_value
- uint32_t flash_size

5.74.1 Detailed Description

Flash information struct.

5.74.2 Field Documentation

5.74.2.1 erased_value

`uint8_t kdev_flash_info_t::erased_value`

Contents of erased memory (usually 0xFF)

Kneron Confidential

5.74.2.2 flash_size

```
uint32_t kdev_flash_info_t::flash_size
```

5.74.2.3 page_size

```
uint32_t kdev_flash_info_t::page_size
```

Optimal programming page size in bytes

5.74.2.4 program_unit

```
uint32_t kdev_flash_info_t::program_unit
```

Smallest programmable unit in bytes

5.74.2.5 sector_count

```
uint32_t kdev_flash_info_t::sector_count
```

Number of sectors

5.74.2.6 sector_info

```
kdev_flash_sector_t* kdev_flash_info_t::sector_info
```

Sector layout information (NULL=Uniform sectors)

5.74.2.7 sector_size

```
uint32_t kdev_flash_info_t::sector_size
```

Uniform sector size in bytes (0=sector_info used)

The documentation for this struct was generated from the following file:

- kdev_flash.h

5.75 kdev_flash_sector_t Struct Reference

Flash Sector index struct.

```
#include <kdev_flash.h>
```

Kneron Confidential

Data Fields

- uint32_t start
- uint32_t end

5.75.1 Detailed Description

Flash Sector index struct.

5.75.2 Field Documentation

5.75.2.1 end

```
uint32_t kdev_flash_sector_t::end
```

Sector End address (start+size-1)

5.75.2.2 start

```
uint32_t kdev_flash_sector_t::start
```

Sector Start address

The documentation for this struct was generated from the following file:

- kdev_flash.h

5.76 kdev_flash_status_t Struct Reference

Flash Status struct.

```
#include <kdev_flash.h>
```

Data Fields

- uint32_t busy: 1
- uint32_t error: 1

5.76.1 Detailed Description

Flash Status struct.

Kneron Confidential

5.76.2 Field Documentation

5.76.2.1 busy

```
uint32_t kdev_flash_status_t::busy
```

Flash busy flag

5.76.2.2 error

```
uint32_t kdev_flash_status_t::error
```

Read/Program/Erase error flag (cleared on start of next operation)

The documentation for this struct was generated from the following file:

- kdev_flash.h

5.77 kdp520_dpi2ahb_context Struct Reference

Data Fields

- uint32_t irq
- uint32_t page_done_num
- uint32_t tile_avg_en
- uint32_t tile_n_w
- uint32_t tile_n_h

5.77.1 Field Documentation

5.77.1.1 irq

```
uint32_t kdp520_dpi2ahb_context::irq
```

5.77.1.2 page_done_num

```
uint32_t kdp520_dpi2ahb_context::page_done_num
```

Kneron Confidential

5.77.1.3 tile_avg_en

```
uint32_t kdp520_dpi2ahb_context::tile_avg_en
```

5.77.1.4 tile_n_h

```
uint32_t kdp520_dpi2ahb_context::tile_n_h
```

5.77.1.5 tile_n_w

```
uint32_t kdp520_dpi2ahb_context::tile_n_w
```

The documentation for this struct was generated from the following file:

- kdrv_dpi2ahb.c

5.78 kdp520_scu_extreg Struct Reference

```
#include <kdrv_scu_ext.h>
```

Data Fields

- volatile unsigned int pll0_setting
- volatile unsigned int pll1_setting
- volatile unsigned int pll2_setting
- volatile unsigned int pll3_setting
- volatile unsigned int pll4_setting
- volatile unsigned int clk_en0
- volatile unsigned int clk_en1
- volatile unsigned int clk_en2
- volatile unsigned int clk_muxsel
- volatile unsigned int clk_div0
- volatile unsigned int clk_div1
- volatile unsigned int clk_div2
- volatile unsigned int clk_div3
- volatile unsigned int clk_div4
- volatile unsigned int clk_div5
- volatile unsigned int reserved01
- volatile unsigned int pll5_setting
- volatile unsigned int clk_div6
- volatile unsigned int clk_div7
- volatile unsigned int reserved02 [10]
- volatile unsigned int spi_cs_n_io
- volatile unsigned int spi_clk_io
- volatile unsigned int spi_do_io
- volatile unsigned int spi_di_io
- volatile unsigned int spi_wp_n_io
- volatile unsigned int spi_hold_n_io
- volatile unsigned int swj_trst_io
- volatile unsigned int swj_tdi_io
- volatile unsigned int swj_swditms_io
- volatile unsigned int swj_swclktck_io
- volatile unsigned int swj_swj_tdo_io
- volatile unsigned int lc_pclk_io

Kneron Confidential

5.78.1 Field Documentation

5.78.1.1 clk_div0

```
volatile unsigned int kdp520_scu_extreg::clk_div0
```

5.78.1.2 clk_div1

```
volatile unsigned int kdp520_scu_extreg::clk_div1
```

5.78.1.3 clk_div2

```
volatile unsigned int kdp520_scu_extreg::clk_div2
```

5.78.1.4 clk_div3

```
volatile unsigned int kdp520_scu_extreg::clk_div3
```

5.78.1.5 clk_div4

```
volatile unsigned int kdp520_scu_extreg::clk_div4
```

5.78.1.6 clk_div5

```
volatile unsigned int kdp520_scu_extreg::clk_div5
```

5.78.1.7 clk_div6

```
volatile unsigned int kdp520_scu_extreg::clk_div6
```

Kneron Confidential

5.78.1.8 clk_div7

```
volatile unsigned int kdp520_scu_extreg::clk_div7
```

5.78.1.9 clk_en0

```
volatile unsigned int kdp520_scu_extreg::clk_en0
```

5.78.1.10 clk_en1

```
volatile unsigned int kdp520_scu_extreg::clk_en1
```

5.78.1.11 clk_en2

```
volatile unsigned int kdp520_scu_extreg::clk_en2
```

5.78.1.12 clk_muxsel

```
volatile unsigned int kdp520_scu_extreg::clk_muxsel
```

5.78.1.13 lc_pclk_io

```
volatile unsigned int kdp520_scu_extreg::lc_pclk_io
```

5.78.1.14 pll0_setting

```
volatile unsigned int kdp520_scu_extreg::pll0_setting
```

5.78.1.15 pll1_setting

```
volatile unsigned int kdp520_scu_extreg::pll1_setting
```

Kneron Confidential

5.78.1.16 pll2_setting

```
volatile unsigned int kdp520_scu_extreg::pll2_setting
```

5.78.1.17 pll3_setting

```
volatile unsigned int kdp520_scu_extreg::pll3_setting
```

5.78.1.18 pll4_setting

```
volatile unsigned int kdp520_scu_extreg::pll4_setting
```

5.78.1.19 pll5_setting

```
volatile unsigned int kdp520_scu_extreg::pll5_setting
```

5.78.1.20 reserved01

```
volatile unsigned int kdp520_scu_extreg::reserved01
```

5.78.1.21 reserved02

```
volatile unsigned int kdp520_scu_extreg::reserved02[10]
```

5.78.1.22 spi_clk_io

```
volatile unsigned int kdp520_scu_extreg::spi_clk_io
```

5.78.1.23 spi_cs_n_io

```
volatile unsigned int kdp520_scu_extreg::spi_cs_n_io
```

Kneron Confidential

5.78.1.24 spi_di_io

```
volatile unsigned int kdp520_scu_extreg::spi_di_io
```

5.78.1.25 spi_do_io

```
volatile unsigned int kdp520_scu_extreg::spi_do_io
```

5.78.1.26 spi_hold_n_io

```
volatile unsigned int kdp520_scu_extreg::spi_hold_n_io
```

5.78.1.27 spi_wp_n_io

```
volatile unsigned int kdp520_scu_extreg::spi_wp_n_io
```

5.78.1.28 swj_swclktck_io

```
volatile unsigned int kdp520_scu_extreg::swj_swclktck_io
```

5.78.1.29 swj_swditms_io

```
volatile unsigned int kdp520_scu_extreg::swj_swditms_io
```

5.78.1.30 swj_swj_tdo_io

```
volatile unsigned int kdp520_scu_extreg::swj_swj_tdo_io
```

5.78.1.31 swj_tdi_io

```
volatile unsigned int kdp520_scu_extreg::swj_tdi_io
```

Kneron Confidential

5.78.1.32 swj_trst_io

```
volatile unsigned int kdp520_scu_extreg::swj_trst_io
```

The documentation for this struct was generated from the following file:

- kdrv_scu_ext.h

5.79 kdp_timer_t Struct Reference

Data Fields

- uint32_t in_use
- timer_cb_fr_isr_t cb
- void * user_arg
- osThreadId_t tid
- uint32_t * ptmid
- uint32_t perf_in_use
- uint32_t perf_cnt
- uint32_t perf_last_instant

5.79.1 Field Documentation

5.79.1.1 cb

```
timer_cb_fr_isr_t kdp_timer_t::cb
```

5.79.1.2 in_use

```
uint32_t kdp_timer_t::in_use
```

5.79.1.3 perf_cnt

```
uint32_t kdp_timer_t::perf_cnt
```

Kneron Confidential

5.79.1.4 perf_in_use

```
uint32_t kdp_timer_t::perf_in_use
```

5.79.1.5 perf_last_instant

```
uint32_t kdp_timer_t::perf_last_instant
```

5.79.1.6 ptmid

```
uint32_t* kdp_timer_t::ptmid
```

5.79.1.7 tid

```
osThreadId_t kdp_timer_t::tid
```

5.79.1.8 user_arg

```
void* kdp_timer_t::user_arg
```

The documentation for this struct was generated from the following file:

- kdrv_timer.c

5.80 kdp_uvc_id Struct Reference

```
#include <kdp_usb_api.h>
```

Data Fields

- uint16_t idVendor
- uint16_t idProduct

5.80.1 Field Documentation

Kneron Confidential

5.80.1.1 idProduct

```
uint16_t kdp_uvc_id::idProduct
```

5.80.1.2 idVendor

```
uint16_t kdp_uvc_id::idVendor
```

The documentation for this struct was generated from the following file:

- kdp_usb_api.h

5.81 kdrv_adc_regs_t Struct Reference

```
#include <kdrv_adc.h>
```

Data Fields

- uint32_t data [8]
- uint32_t reserve [24]
- uint32_t thrhold [8]
- uint32_t reserve1 [24]
- uint32_t ctrl
- uint32_t trim
- uint32_t inten
- uint32_t intst
- uint32_t tparam
- uint32_t smpr
- uint32_t reserve2
- uint32_t prescal
- uint32_t sqr

5.81.1 Detailed Description

ADC Register Configuration

The documentation for this struct was generated from the following file:

- kdrv_adc.h

5.82 kdrv_adc_resource_t Struct Reference

```
#include <kdrv_adc.h>
```

Kneron Confidential

Data Fields

- int io_base
- int irq

5.82.1 Detailed Description

ADC Resource Configuration

The documentation for this struct was generated from the following file:

- kdrv_adc.h

5.83 kdrv_clock_list Struct Reference

`#include <kdrv_clock.h>`

Collaboration diagram for kdrv_clock_list:

Data Fields

- struct kdrv_clock_list * next

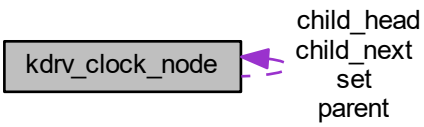
The documentation for this struct was generated from the following file:

- kdrv_clock.h

5.84 kdrv_clock_node Struct Reference

`#include <kdrv_clock.h>`

Collaboration diagram for kdrv_clock_node:



Kneron Confidential

Data Fields

- struct kdrv_clock_node * parent
- struct kdrv_clock_node * child_head
- struct kdrv_clock_node * child_next
- fn_set set
- uint8_t is_enabled
- char name [15]

The documentation for this struct was generated from the following file:

- kdrv_clock.h

5.85 kdrv_clock_value Struct Reference

```
#include <kdrv_clock.h>
```

Data Fields

- uint16_t ms
- uint16_t ns
- uint16_t ps
- uint8_t div
- uint8_t enable

The documentation for this struct was generated from the following file:

- kdrv_clock.h

5.86 kdrv_csirx_context Struct Reference

Data Fields

- uint32_t csi_rx_base
- uint32_t mipi_rx_phy_csr
- uint32_t mipi_lane_num

5.86.1 Field Documentation

5.86.1.1 csi_rx_base

```
uint32_t kdrv_csirx_context::csi_rx_base
```

Kneron Confidential

5.86.1.2 mipi_lane_num

```
uint32_t kdrv_csirx_context::mipi_lane_num
```

5.86.1.3 mipi_rx_phy_csr

```
uint32_t kdrv_csirx_context::mipi_rx_phy_csr
```

The documentation for this struct was generated from the following file:

- kdrv_mipicsirx.c

5.87 kdrv_display_pen_info_t Struct Reference

Enumerations of display pen setting.

```
#include <kdrv_display.h>
```

Data Fields

- unsigned int width
- uint16_t color

5.87.1 Detailed Description

Enumerations of display pen setting.

5.87.2 Field Documentation

5.87.2.1 color

```
uint16_t kdrv_display_pen_info_t::color
```

Color of pen

Kneron Confidential

5.87.2.2 width

```
unsigned int kdrv_display_pen_info_t::width
```

Width of pen

The documentation for this struct was generated from the following file:

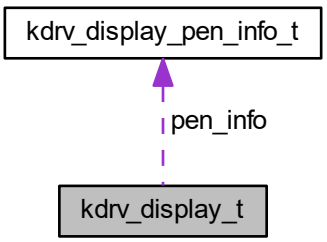
- kdrv_display.h

5.88 kdrv_display_t Struct Reference

Enumerations of display driver setting.

```
#include <kdrv_display.h>
```

Collaboration diagram for kdrv_display_t:



Data Fields

- struct video_input_params vi_params
- uint8_t cam_src_idx
- uint32_t fb_size
- uint32_t fb_addr
- uint32_t base
- uint16_t display_id
- kdrv_display_pen_info_t pen_info

5.88.1 Detailed Description

Enumerations of display driver setting.

Kneron Confidential

5.88.2 Field Documentation

5.88.2.1 base

```
uint32_t kdrv_display_t::base
```

Display driver base address

5.88.2.2 cam_src_idx

```
uint8_t kdrv_display_t::cam_src_idx
```

Camera source index

5.88.2.3 display_id

```
uint16_t kdrv_display_t::display_id
```

Display driver id

5.88.2.4 fb_addr

```
uint32_t kdrv_display_t::fb_addr
```

Frame buffer address

5.88.2.5 fb_size

```
uint32_t kdrv_display_t::fb_size
```

Frame buffer size

5.88.2.6 pen_info

```
kdrv_display_pen_info_t kdrv_display_t::pen_info
```

see kdrv_display_pen_info_t

5.88.2.7 vi_params

```
struct video_input_params kdrv_display_t::vi_params
```

see video_input_params

The documentation for this struct was generated from the following file:

- kdrv_display.h

Kneron Confidential

5.89 kdrv_lcdc_fb_t Struct Reference

Data Fields

- int buf_size
- uint32_t buf_addr
- bool init_done

5.89.1 Field Documentation

5.89.1.1 buf_addr

```
uint32_t kdrv_lcdc_fb_t::buf_addr
```

5.89.1.2 buf_size

```
int kdrv_lcdc_fb_t::buf_size
```

5.89.1.3 init_done

```
bool kdrv_lcdc_fb_t::init_done
```

The documentation for this struct was generated from the following file:

- kdrv_lcdc.c

5.90 kdrv_lcm_context_t Struct Reference

Data Fields

- uint32_t dp_buffer_addr

5.90.1 Field Documentation

Kneron Confidential

5.90.1.1 dp_buffer_addr

```
uint32_t kdrv_lcm_context_t::dp_buffer_addr
```

The documentation for this struct was generated from the following file:

- kdrv_lcm.c

5.91 kdrv_lcm_fb_t Struct Reference

Data Fields

- int buf_size
- uint32_t buf_addr
- bool init_done

5.91.1 Field Documentation

5.91.1.1 buf_addr

```
uint32_t kdrv_lcm_fb_t::buf_addr
```

5.91.1.2 buf_size

```
int kdrv_lcm_fb_t::buf_size
```

5.91.1.3 init_done

```
bool kdrv_lcm_fb_t::init_done
```

The documentation for this struct was generated from the following file:

- kdrv_lcm.c

Kneron Confidential

5.92 kdrv_pwmtimer_control Struct Reference

Data Fields

- uint32_t TmSrc: 1
- uint32_t TmStart: 1
- uint32_t TmUpdate: 1
- uint32_t TmOutInv: 1
- uint32_t TmAutoLoad:1
- uint32_t TmIntEn: 1
- uint32_t TmIntMode: 1
- uint32_t TmDmaEn: 1
- uint32_t TmPwmEn: 1
- uint32_t Reserved: 15
- uint32_t TmDeadZone:8

5.92.1 Field Documentation

5.92.1.1 Reserved

```
uint32_t kdrv_pwmtimer_control::Reserved
```

5.92.1.2 TmAutoLoad

```
uint32_t kdrv_pwmtimer_control::TmAutoLoad
```

5.92.1.3 TmDeadZone

```
uint32_t kdrv_pwmtimer_control::TmDeadZone
```

5.92.1.4 TmDmaEn

```
uint32_t kdrv_pwmtimer_control::TmDmaEn
```

Kneron Confidential

5.92.1.5 TmIntEn

```
uint32_t kdrv_pwmtimer_control::TmIntEn
```

5.92.1.6 TmIntMode

```
uint32_t kdrv_pwmtimer_control::TmIntMode
```

5.92.1.7 TmOutInv

```
uint32_t kdrv_pwmtimer_control::TmOutInv
```

5.92.1.8 TmPwmEn

```
uint32_t kdrv_pwmtimer_control::TmPwmEn
```

5.92.1.9 TmSrc

```
uint32_t kdrv_pwmtimer_control::TmSrc
```

5.92.1.10 TmStart

```
uint32_t kdrv_pwmtimer_control::TmStart
```

5.92.1.11 TmUpdate

```
uint32_t kdrv_pwmtimer_control::TmUpdate
```

The documentation for this struct was generated from the following file:

- kdrv_pwm.c

Kneron Confidential

5.93 kdrv_pwmtimer_struct Struct Reference

Data Fields

- uint32_t IntNum
- uint32_t Tick
- uint32_t Running

5.93.1 Field Documentation

5.93.1.1 IntNum

```
uint32_t kdrv_pwmtimer_struct::IntNum
```

5.93.1.2 Running

```
uint32_t kdrv_pwmtimer_struct::Running
```

5.93.1.3 Tick

```
uint32_t kdrv_pwmtimer_struct::Tick
```

The documentation for this struct was generated from the following file:

- kdrv_pwm.c

5.94 kdrv_sd_status_t Struct Reference

```
#include <kdrv_sdc.h>
```

Kneron Confidential

Data Fields

- uint32_t reserved1:5
- uint32_t secured_mode:1
- uint32_t dat_bus_width:2
- uint32_t sd_card_type_hi:8
- uint32_t reserved2:8
- uint32_t sd_card_type_lo:8
- uint32_t size_of_protected_area
- uint8_t speed_class
- uint8_t performance_move
- uint32_t reserved3:4
- uint32_t au_size:4
- uint8_t erase_size [2]
- uint32_t erase_offset:2
- uint32_t erase_timeout:6
- uint8_t reserved4 [11]
- uint8_t reserved5 [39]

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.95 kdrv_sdc_adma2desc_table_t Struct Reference

```
#include <kdrv_sdc.h>
```

Data Fields

- uint16_t attr
- uint16_t lgth
- uint32_t addr

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.96 kdrv_sdc_csd_v1_t Struct Reference

```
#include <kdrv_sdc.h>
```

Kneron Confidential

Data Fields

- uint32_t csd_structure:2
- uint32_t reserved1:6
- uint8_t taac
- uint8_t nsac
- uint8_t tran_speed
- uint32_t ccc:12
- uint32_t read_bl_len:4
- uint32_t read_bl_partial:1
- uint32_t write_blk_misalign:1
- uint32_t read_blk_misalign:1
- uint32_t dsr_imp:1
- uint32_t reserved2:2
- uint32_t c_size:12
- uint32_t vdd_r_curr_min:3
- uint32_t vdd_r_curr_max:3
- uint32_t vdd_w_curr_min:3
- uint32_t vdd_w_curr_max:3
- uint32_t c_size_mult:3
- uint32_t erase_blk_en:1
- uint32_t sector_size:7
- uint32_t wp_grp_size:7
- uint32_t wp_grp_enable:1
- uint32_t reserved3:2
- uint32_t r2w_factor:3
- uint32_t write_bl_len:4
- uint32_t write_bl_partial:1
- uint32_t reserved4:5
- uint32_t file_format_grp:1
- uint32_t copy:1
- uint32_t perm_write_protect:1
- uint32_t tmp_write_protect:1
- uint32_t file_format:2
- uint32_t Reserver5:2

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.97 kdrv_sdc_csd_v2_t Struct Reference

```
#include <kdrv_sdc.h>
```

Kneron Confidential

Data Fields

- uint32_t csd_structure:2
- uint32_t reserved1:6
- uint8_t taac
- uint8_t nsac
- uint8_t tran_speed
- uint32_t ccc:12
- uint32_t read_bl_len:4
- uint32_t read_bl_partial:1
- uint32_t write_blk_misalign:1
- uint32_t read_blk_misalign:1
- uint32_t dsr_imp:1
- uint32_t reserved2:6
- uint32_t c_size:22
- uint32_t reserved3:1
- uint32_t erase_blk_en:1
- uint32_t sector_size:7
- uint32_t wp_grp_size:7
- uint32_t wp_grp_enable:1
- uint32_t reserved4:2
- uint32_t r2w_factor:3
- uint32_t write_bl_len:4
- uint32_t write_bl_partial:1
- uint32_t reserved5:5
- uint32_t file_format_grp:1
- uint32_t copy:1
- uint32_t perm_write_protect:1
- uint32_t tmp_write_protect:1
- uint32_t file_format:2
- uint32_t reserver6:2

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.98 kdrv_sdc_flow_info_t Struct Reference

```
#include <kdrv_sdc.h>
```

Data Fields

- kdrv_sdc_transfer_type_e use_dma
- uint16_t line_bound
- uint16_t adma2rand
- kdrv_sdc_abort_type_e sync_abort
- uint8_t erasing
- uint8_t auto_cmd
- uint8_t reserved

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

Kneron Confidential

5.99 kdrv_sdc_mmc_csd_t Struct Reference

`#include <kdrv_sdc_mmc.h>`

Data Fields

- uint32_t csd_structure:2
- uint32_t spec_vers:4
- uint32_t reserved1:2
- uint8_t taac
- uint8_t nsac
- uint8_t tran_speed
- uint32_t ccc:12
- uint32_t read_bl_len:4
- uint32_t read_bl_partial:1
- uint32_t write_blk_misalign:1
- uint32_t read_blk_misalign:1
- uint32_t dsr_imp:1
- uint32_t reserved2:2
- uint32_t c_size:12
- uint32_t vdd_r_curr_min:3
- uint32_t vdd_r_curr_max:3
- uint32_t vdd_w_curr_min:3
- uint32_t vdd_w_curr_max:3
- uint32_t c_size_mult:3
- uint32_t erase_grp_size:5
- uint32_t erase_grp_mult:5
- uint32_t wp_grp_size:5
- uint32_t wp_grp_enable:1
- uint32_t default_ecc:2
- uint32_t r2w_factor:3
- uint32_t write_bl_len:4
- uint32_t write_bl_partial:1
- uint32_t reserved3:4
- uint32_t content_prot_app:1
- uint32_t file_format_grp:1
- uint32_t COPY:1
- uint32_t perm_write_protect:1
- uint32_t tmp_write_protect:1
- uint32_t file_format:2
- uint32_t ecc:2

The documentation for this struct was generated from the following file:

- kdrv_sdc_mmc.h

5.100 kdrv_sdc_mmc_ext_csd_t Struct Reference

`#include <kdrv_sdc_mmc.h>`

Kneron Confidential

Data Fields

- uint8_t reserved27 [134]
- uint8_t sec_bad_blk_mgmnt
- uint8_t reserved26
- uint8_t enh_start_addr [4]
- uint8_t enh_size_mult [3]
- uint8_t gp_size_mult [12]
- uint8_t partition_setting_completed
- uint8_t partitioning_attribute
- uint8_t max_enh_size_mult [3]
- uint8_t partitioning_support
- uint8_t reserved25
- uint8_t rst_n_function
- uint8_t reserved24 [5]
- uint8_t rpmb_size_mult
- uint8_t fw_config
- uint8_t reserved23
- uint8_t user_wp
- uint8_t reserved22
- uint8_t boot_wp
- uint8_t reserved21
- uint8_t erase_group_def
- uint8_t reserved20
- uint8_t boot_bus_width
- uint8_t boot_config_prot
- uint8_t partition_conf
- uint8_t reserved19
- uint8_t erased_mem_cont
- uint8_t reserved18
- uint8_t bus_width
- uint8_t reserved17
- uint8_t hs_timing
- uint8_t reserved16
- uint8_t power_class
- uint8_t reserved15
- uint8_t cmd_set_rev
- uint8_t reserved14
- uint8_t cmd_set
- uint8_t ext_csd_rev
- uint8_t reserved13
- uint8_t csd_structure
- uint8_t reserved12
- uint8_t cardtype
- uint8_t reserved11 [3]
- uint8_t pwr_cl_52_195
- uint8_t pwr_cl_26_195
- uint8_t pwr_cl_52_360
- uint8_t pwr_cl_26_360
- uint8_t reserved10
- uint8_t min_perf_r_4_26
- uint8_t min_perf_w_4_26
- uint8_t min_perf_r_8_26_4_52
- uint8_t min_perf_w_8_26_4_52
- uint8_t min_perf_r_8_52

Kneron Confidential

- uint8_t min_perf_w_8_52
- uint8_t reserved9
- uint32_t sec_count
- uint8_t reserved8
- uint8_t s_a_timeout
- uint8_t reserved7
- uint8_t s_c_vccq
- uint8_t s_c_vcc
- uint8_t hc_wp_grp_size
- uint8_t ref_wr_sec_c
- uint8_t erase_timeout_mult
- uint8_t hc_erase_grp_size
- uint8_t acc_size
- uint8_t boot_size_mult
- uint8_t reserved6
- uint8_t boot_info
- uint8_t sec_trim_mult
- uint8_t sec_erase_mult
- uint8_t sec_feature_support
- uint8_t trim_mult
- uint8_t reserved5
- uint8_t min_perf_ddr_r_8_52
- uint8_t min_perf_ddr_w_8_52_8_52
- uint8_t reserved4 [2]
- uint8_t pwr_cl_ddr_52_195
- uint8_t pwr_cl_ddr_52_360
- uint8_t reserved3
- uint8_t ini_timeout_ap
- uint8_t reserved2 [262]
- uint8_t s_cmd_set
- uint8_t reserved1 [7]

The documentation for this struct was generated from the following file:

- kdrv_sdc_mmc.h

5.101 kdrv_sdc_reg_t Struct Reference

```
#include <kdrv_sdc.h>
```

Data Fields

- uint32_t sdma_addr
- uint16_t blk_size
- uint16_t blk_cnt
- uint32_t cmd_argu
- uint16_t txmode
- uint16_t cmd_reg
- uint64_t cmd_resplo
- uint64_t cmd_resphi

Kneron Confidential

- uint32_t buf_data
- uint32_t present_state
- uint8_t hcreg
- uint8_t pwr_ctl
- uint8_t blk_gap_ctl
- uint8_t wakeup_ctl
- uint16_t clk_ctl
- uint8_t timeout_ctl
- uint8_t softrst
- uint16_t intr_sts
- uint16_t err_sts
- uint16_t intr_en
- uint16_t err_en
- uint16_t intr_sig_en
- uint16_t err_sig_en
- uint16_t auto_cmd_err
- uint16_t host_ctl2
- uint32_t cap_reg
- uint32_t cap_reg2
- uint64_t max_curr
- uint16_t cmd12_force_evt
- uint16_t force_evt
- uint32_t adma_err_sts
- uint64_t adma_addr
- uint16_t preset_val_init
- uint16_t preset_val_ds
- uint16_t preset_val_hs
- uint16_t preset_val_sdr12
- uint16_t preset_val_sdr25
- uint16_t preset_val_sdr50
- uint16_t preset_val_sdr104
- uint16_t preset_val_ddr50
- uint32_t reserved [28]
- uint32_t share_bus_ctl
- uint32_t reserved2 [6]
- uint16_t slt_intr_sts
- uint16_t hcver
- uint32_t vendor_reg0
- uint32_t vendor_reg1
- uint32_t vendor_reg2
- uint32_t vendor_reg3
- uint32_t vendor_reg4
- uint32_t vendor_reg5
- uint32_t vendor_reg6
- uint32_t ahb_err_sts
- uint32_t ahb_err_en
- uint32_t ahb_err_sig_en
- uint32_t dma_hndshk
- uint32_t reserved4 [19]
- uint32_t hw_attr
- uint32_t ip_ver
- uint32_t ciph_m_ctl
- uint32_t ciph_m_sts
- uint16_t ciph_m_sts_en
- uint16_t ciph_m_sig_en

Kneron Confidential

- uint32_t in_data_lo
- uint32_t in_data_hi
- uint32_t in_key_lo
- uint32_t in_key_hi
- uint32_t out_data_lo
- uint32_t out_data_hi
- uint32_t secr_table_port

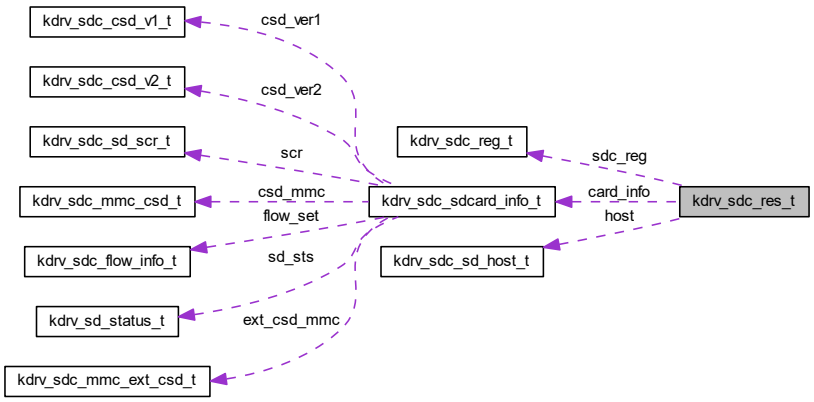
The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.102 kdrv_sdc_res_t Struct Reference

`#include <kdrv_sdc.h>`

Collaboration diagram for kdrv_sdc_res_t:



Data Fields

- volatile kdrv_sdc_reg_t * sdc_reg
- volatile kdrv_sdc_sdcard_info_t * card_info
- kdrv_sdc_sd_host_t * host
- kdrv_sdc_infinite_test_e infinite_mode
- uint16_t fifo_depth
- uint32_t timeout_ms
- uint32_t data_present
- uint32_t adma2_use_interrupt
- uint32_t adma2_insert_nop
- uint32_t response_type
- uint32_t inhibit_datchk

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.103 kdrv_sdc_sd_host_t Struct Reference

`#include <kdrv_sdc.h>`

Data Fields

- uint32_t max_clk
- uint32_t min_clk
- uint32_t clock
- uint8_t power
- uint32_t ocr_avail

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.104 kdrv_sdc_sd_scr_t Struct Reference

`#include <kdrv_sdc.h>`

Data Fields

- uint32_t sd_spec:4
- uint32_t scr_structure:4
- uint32_t sd_bus_widths:4
- uint32_t sd_security:3
- uint32_t data_stat_after_erase:1
- uint32_t reserved1:7
- uint32_t sd_spec3:1
- uint32_t cmd20_support:1
- uint32_t cmd23_support:1
- uint32_t reserverd2:6
- uint32_t reserverd3

The documentation for this struct was generated from the following file:

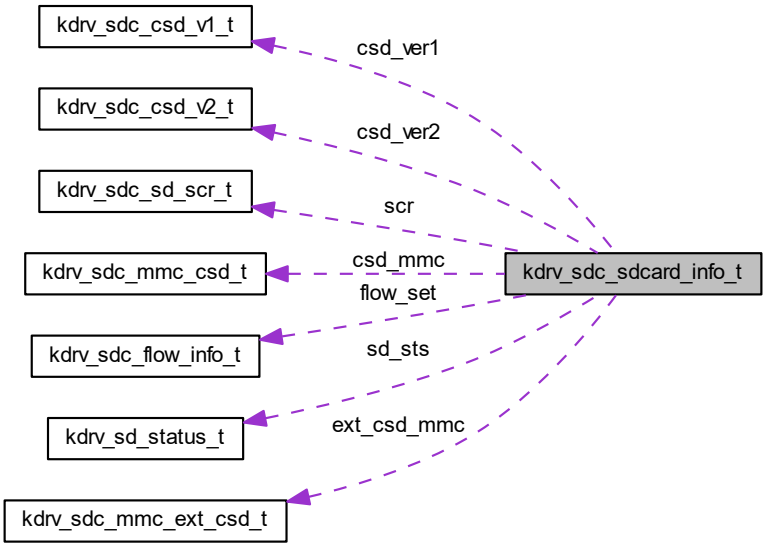
- kdrv_sdc.h

Kneron Confidential

5.105 kdrv_sdc_sdcard_info_t Struct Reference

`#include <kdrv_sdc.h>`

Collaboration diagram for kdrv_sdc_sdcard_info_t:



Data Fields

- uint32_t card_insert
- kdrv_sdc_flow_info_t flow_set
- uint16_t rca
- uint16_t dsr
- kdrv_sdc_sd_scr_t scr
- uint32_t ocr
- uint64_t csd_lo
- uint64_t csd_hi
- kdrv_sdc_csd_v1_t csd_ver1
- kdrv_sdc_csd_v2_t csd_ver2
- uint64_t cid_lo
- uint64_t cid_hi
- uint64_t resp_lo
- uint64_t resp_hi
- uint32_t num_of_blks
- uint8_t switch_sts [64]
- volatile uint16_t err_sts
- volatile uint16_t auto_err
- volatile uint8_t cmpl_mask
- kdrv_sd_status_t sd_sts
- uint16_t bs_mode
- uint8_t bus_width

Kneron Confidential

- uint8_t already_init
- uint8_t blk_addr
- uint32_t max_dtr
- kdrv_sdc_bus_speed_e speed
- uint32_t fifo_depth
- kdrv_sdc_mmc_csd_t csd_mmc
- kdrv_sdc_mmc_ext_csd_t ext_csd_mmc
- uint32_t num_of_boot_blks
- uint8_t u8_sdma_lock
- uint8_t sdma_intr
- uint32_t card_type
- uint8_t num_io_func
- uint8_t mem_present
- uint32_t drive
- uint32_t sys_freq
- uint32_t protected_drive
- uint32_t cprm_init
- uint32_t kmu_lo
- uint32_t kmu_hi
- uint32_t auto_cbc

The documentation for this struct was generated from the following file:

- kdrv_sdc.h

5.106 kdrv_uart_config_t Struct Reference

The structure of UART configuration parameters.

```
#include <kdrv_uart.h>
```

Data Fields

- uint32_t baudrate
- uint8_t data_bits
- uint8_t frame_length
- uint8_t stop_bits
- uint8_t parity_mode
- bool fifo_en

5.106.1 Detailed Description

The structure of UART configuration parameters.

The documentation for this struct was generated from the following file:

- kdrv_uart.h

Kneron Confidential

5.107 kdrv_uart_fifo_config_t Struct Reference

The structure of UART FIFO configuration parameters.

`#include <kdrv_uart.h>`

Data Fields

- bool bEnFifo
- uint8_t fifo_trig_level

5.107.1 Detailed Description

The structure of UART FIFO configuration parameters.

The documentation for this struct was generated from the following file:

- kdrv_uart.h

5.108 kdrv_usbd_event_t Struct Reference

USB event, it includes kdrv_usbd_event_name_t and related data.

`#include <kdrv_usbd.h>`

Data Fields

- kdrv_usbd_event_name_t ename
- union {
  kdrv_usbd_setup_packet_t kdrv_usbd_event_t::setup
  struct {
  uint32_t kdrv_usbd_event_t::data1
  uint32_t kdrv_usbd_event_t::data2
  }
  };

5.108.1 Detailed Description

USB event, it includes kdrv_usbd_event_name_t and related data.

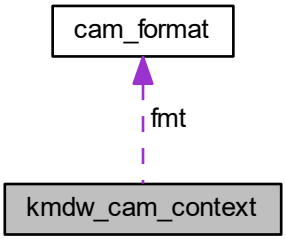
The documentation for this struct was generated from the following file:

- kdrv_usbd.h

Kneron Confidential

5.109 kmdw_cam_context Struct Reference

Collaboration diagram for kmdw_cam_context:



Data Fields

- uint32_t id
- uint32_t sensor_id
- uint32_t cam_input_type
- uint32_t capabilities
- struct cam_format fmt

5.109.1 Field Documentation

5.109.1.1 cam_input_type

```
uint32_t kmdw_cam_context::cam_input_type
```

5.109.1.2 capabilities

```
uint32_t kmdw_cam_context::capabilities
```

5.109.1.3 fmt

```
struct cam_format kmdw_cam_context::fmt
```

Kneron Confidential

5.109.1.4 id

```
uint32_t kmdw_cam_context::id
```

5.109.1.5 sensor_id

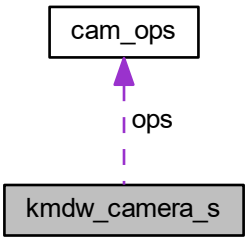
```
uint32_t kmdw_cam_context::sensor_id
```

The documentation for this struct was generated from the following file:

- kmdw_camera_kl520.c

5.110 kmdw_camera_s Struct Reference

Collaboration diagram for kmdw_camera_s:



Data Fields

- uint32_t inuse
- struct cam_ops * ops

5.110.1 Field Documentation

5.110.1.1 inuse

```
uint32_t kmdw_camera_s::inuse
```

Kneron Confidential

5.110.1.2 ops

```
struct cam_ops* kmdw_camera_s::ops
```

The documentation for this struct was generated from the following file:

- kmdw_camera.c

5.111 kmdw_display_panel_drv Struct Reference

Structure of representing display and panel driver compatibility.

```
#include <kmdw_display.h>
```

Data Fields

- struct video_input_params vi_params
- uint32_t fb_size
- uint32_t type
- uint32_t base
- uint16_t display_id

5.111.1 Detailed Description

Structure of representing display and panel driver compatibility.

5.111.2 Field Documentation

5.111.2.1 base

```
uint32_t kmdw_display_panel_drv::base
```

5.111.2.2 display_id

```
uint16_t kmdw_display_panel_drv::display_id
```

Kneron Confidential

5.111.2.3 fb_size

```
uint32_t kmdw_display_panel_drv::fb_size
```

5.111.2.4 type

```
uint32_t kmdw_display_panel_drv::type
```

5.111.2.5 vi_params

```
struct video_input_params kmdw_display_panel_drv::vi_params
```

The documentation for this struct was generated from the following file:

- kmdw_display.h

5.112 kmdw_img_data_t Struct Reference

Data Fields

- int32_t raw_img_idx
- osEventFlagsId_t evt_caller
- uint32_t caller_e
- osEventFlagsId_t evt_result
- uint32_t result_e

5.112.1 Field Documentation

5.112.1.1 caller_e

```
uint32_t kmdw_img_data_t::caller_e
```

5.112.1.2 evt_caller

```
osEventFlagsId_t kmdw_img_data_t::evt_caller
```

Kneron Confidential

5.112.1.3 evt_result

```
osEventFlagsId_t kmdw_img_data_t::evt_result
```

5.112.1.4 raw_img_idx

```
int32_t kmdw_img_data_t::raw_img_idx
```

5.112.1.5 result_e

```
uint32_t kmdw_img_data_t::result_e
```

The documentation for this struct was generated from the following file:

- kmdw_model.c

5.113 kmdw_model_data_t Struct Reference

Data Fields

- uint32_t n_model_count
- struct kdp_model_s * p_model_info
- uint8_t * pn_is_model_loaded_table
- uint32_t n_ddr_addr_model_end
- int32_t n_model_slot_index

5.113.1 Field Documentation

5.113.1.1 n_ddr_addr_model_end

```
uint32_t kmdw_model_data_t::n_ddr_addr_model_end
```

5.113.1.2 n_model_count

```
uint32_t kmdw_model_data_t::n_model_count
```

Kneron Confidential

5.113.1.3 n_model_slot_index

```
int32_t kmdw_model_data_t::n_model_slot_index
```

5.113.1.4 p_model_info

```
struct kdp_model_s* kmdw_model_data_t::p_model_info
```

5.113.1.5 pn_is_model_loaded_table

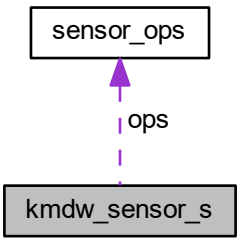
```
uint8_t* kmdw_model_data_t::pn_is_model_loaded_table
```

The documentation for this struct was generated from the following file:

- kmdw_model.c

5.114 kmdw_sensor_s Struct Reference

Collaboration diagram for kmdw_sensor_s:



Data Fields

- uint32_t inuse
- struct sensor_ops * ops

5.114.1 Field Documentation

Kneron Confidential

5.114.1.1 inuse

```
uint32_t kmdw_sensor_s::inuse
```

5.114.1.2 ops

```
struct sensor_ops* kmdw_sensor_s::ops
```

The documentation for this struct was generated from the following file:

- kmdw_sensor.c

5.115 lcdc_img_pixfmt_pxp Struct Reference

Data Fields

- kdrv_lcdc_img_pixfmt_t pixfmt_img0
- kdrv_lcdc_img_pixfmt_t pixfmt_img1
- kdrv_lcdc_img_pixfmt_t pixfmt_img2
- kdrv_lcdc_img_pixfmt_t pixfmt_img3

5.115.1 Field Documentation

5.115.1.1 pixfmt_img0

```
kdrv_lcdc_img_pixfmt_t lcdc_img_pixfmt_pxp::pixfmt_img0
```

5.115.1.2 pixfmt_img1

```
kdrv_lcdc_img_pixfmt_t lcdc_img_pixfmt_pxp::pixfmt_img1
```

5.115.1.3 pixfmt_img2

```
kdrv_lcdc_img_pixfmt_t lcdc_img_pixfmt_pxp::pixfmt_img2
```

Kneron Confidential

5.115.1.4 pixfmt_img3

```
kdrv_lcdc_img_pixfmt_t lcdc_img_pixfmt_pxp::pixfmt_img3
```

The documentation for this struct was generated from the following file:

- kdrv_lcdc.c

5.116 Link_Pointer_DWord Struct Reference

Data Fields

- uint32_t terminate: 1
- uint32_t type: 2
- uint32_t reserved: 2
- uint32_t ptr_address: 27

5.116.1 Field Documentation

5.116.1.1 ptr_address

```
uint32_t Link_Pointer_DWord::ptr_address
```

5.116.1.2 reserved

```
uint32_t Link_Pointer_DWord::reserved
```

5.116.1.3 terminate

```
uint32_t Link_Pointer_DWord::terminate
```

5.116.1.4 type

```
uint32_t Link_Pointer_DWord::type
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

Kneron Confidential

5.117 Mask_DWord Struct Reference

Data Fields

- uint32_t S_mask: 8
- uint32_t C_mask: 8
- uint32_t hub_addr: 7
- uint32_t port_number: 7
- uint32_t mult: 2

5.117.1 Field Documentation

5.117.1.1 C_mask

```
uint32_t Mask_DWord::C_mask
```

5.117.1.2 hub_addr

```
uint32_t Mask_DWord::hub_addr
```

5.117.1.3 mult

```
uint32_t Mask_DWord::mult
```

5.117.1.4 port_number

```
uint32_t Mask_DWord::port_number
```

5.117.1.5 S_mask

```
uint32_t Mask_DWord::S_mask
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

Kneron Confidential

5.118 ota_boot_cfg_item_t Struct Reference

Data Fields

- u32 partition_id
- u32 seq
- u32 flag

5.118.1 Field Documentation

5.118.1.1 flag

```
u32 ota_boot_cfg_item_t::flag
```

5.118.1.2 partition_id

```
u32 ota_boot_cfg_item_t::partition_id
```

5.118.1.3 seq

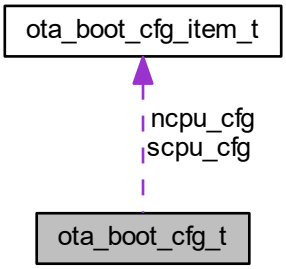
```
u32 ota_boot_cfg_item_t::seq
```

The documentation for this struct was generated from the following file:

- kmdw_ota.c

5.119 ota_boot_cfg_t Struct Reference

Collaboration diagram for ota_boot_cfg_t:



Kneron Confidential

Data Fields

- ota_boot_cfg_item_t scpu_cfg
- ota_boot_cfg_item_t ncpu_cfg

5.119.1 Field Documentation

5.119.1.1 ncpu_cfg

```
ota_boot_cfg_item_t ota_boot_cfg_t::ncpu_cfg
```

5.119.1.2 scpu_cfg

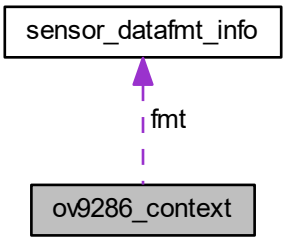
```
ota_boot_cfg_item_t ota_boot_cfg_t::scpu_cfg
```

The documentation for this struct was generated from the following file:

- kmdw_ota.c

5.120 ov9286_context Struct Reference

Collaboration diagram for ov9286_context:



Data Fields

- struct v2k_subdev subdev
- const struct sensor_datafmt_info * fmt

Kneron Confidential

5.120.1 Field Documentation

5.120.1.1 fmt

```
const struct sensor_datafmt_info* ov9286_context::fmt
```

5.120.1.2 subdev

```
struct v2k_subdev ov9286_context::subdev
```

The documentation for this struct was generated from the following file:

- kdev_sensor_ov9286.c

5.121 pu_backlight Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum backlight_req req
- uint8_t caps
- uint16_t wBacklightCompensation

5.121.1 Field Documentation

5.121.1.1 caps

```
uint8_t pu_backlight::caps
```

5.121.1.2 req

```
enum backlight_req pu_backlight::req
```

Kneron Confidential

5.121.1.3 wBacklightCompensation

```
uint16_t pu_backlight::wBacklightCompensation
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.122 pu_brightness Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum brightness_req req
- uint8_t caps
- int16_t wBrightness

5.122.1 Field Documentation

5.122.1.1 caps

```
uint8_t pu_brightness::caps
```

5.122.1.2 req

```
enum brightness_req pu_brightness::req
```

5.122.1.3 wBrightness

```
int16_t pu_brightness::wBrightness
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

Kneron Confidential

5.123 pu_contrast Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum contrast_req req
- uint8_t caps
- uint16_t wContrast

5.123.1 Field Documentation

5.123.1.1 caps

`uint8_t pu_contrast::caps`

5.123.1.2 req

`enum contrast_req pu_contrast::req`

5.123.1.3 wContrast

`uint16_t pu_contrast::wContrast`

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.124 pu_contrast_auto Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum contrast_auto_req req
- uint8_t caps
- uint8_t bContrastAuto

Kneron Confidential

5.124.1 Field Documentation

5.124.1.1 bContrastAuto

```
uint8_t pu_contrast_auto::bContrastAuto
```

5.124.1.2 caps

```
uint8_t pu_contrast_auto::caps
```

5.124.1.3 req

```
enum contrast_auto_req pu_contrast_auto::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.125 pu_dmultiplier Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum dmpl_req req
- uint8_t caps
- uint16_t wMultiplierStep

5.125.1 Field Documentation

5.125.1.1 caps

```
uint8_t pu_dmultiplier::caps
```

Kneron Confidential

5.125.1.2 req

```
enum dmpl_req pu_dmultiplier::req
```

5.125.1.3 wMultiplierStep

```
uint16_t pu_dmultiplier::wMultiplierStep
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.126 pu_dmultiplierlimit Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum dmpl_limit_req req
- uint8_t caps
- uint16_t wMultiplierLimit

5.126.1 Field Documentation

5.126.1.1 caps

```
uint8_t pu_dmultiplierlimit::caps
```

5.126.1.2 req

```
enum dmpl_limit_req pu_dmultiplierlimit::req
```

Kneron Confidential

5.126.1.3 wMultiplierLimit

```
uint16_t pu_dmultiplierlimit::wMultiplierLimit
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.127 pu_gain Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum gain_req req
- uint8_t caps
- uint16_t wGain

5.127.1 Field Documentation

5.127.1.1 caps

```
uint8_t pu_gain::caps
```

5.127.1.2 req

```
enum gain_req pu_gain::req
```

5.127.1.3 wGain

```
uint16_t pu_gain::wGain
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

Kneron Confidential

5.128 pu_gamma Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum gamma_req req
- uint8_t caps
- uint16_t wGamma

5.128.1 Field Documentation

5.128.1.1 caps

`uint8_t pu_gamma::caps`

5.128.1.2 req

`enum gamma_req pu_gamma::req`

5.128.1.3 wGamma

`uint16_t pu_gamma::wGamma`

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.129 pu_hue Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum hue_req req
- uint8_t caps
- uint16_t wHue

Kneron Confidential

5.129.1 Field Documentation

5.129.1.1 caps

```
uint8_t pu_hue::caps
```

5.129.1.2 req

```
enum hue_req pu_hue::req
```

5.129.1.3 wHue

```
uint16_t pu_hue::wHue
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.130 pu_hue_auto Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum hue_auto_req req
- uint8_t caps
- uint8_t bHueAuto

5.130.1 Field Documentation

5.130.1.1 bHueAuto

```
uint8_t pu_hue_auto::bHueAuto
```

Kneron Confidential

5.130.1.2 caps

```
uint8_t pu_hue_auto::caps
```

5.130.1.3 req

```
enum hue_auto_req pu_hue_auto::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.131 pu_power_line_frequency Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum power_line_freq_req req
- uint8_t caps
- uint8_t bPowerLineFrequency

5.131.1 Field Documentation

5.131.1.1 bPowerLineFrequency

```
uint8_t pu_power_line_frequency::bPowerLineFrequency
```

5.131.1.2 caps

```
uint8_t pu_power_line_frequency::caps
```

Kneron Confidential

5.131.1.3 req

```
enum power_line_freq_req pu_power_line_frequency::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.132 pu_saturation Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum saturation_req req
- uint8_t caps
- uint16_t wSaturation

5.132.1 Field Documentation

5.132.1.1 caps

```
uint8_t pu_saturation::caps
```

5.132.1.2 req

```
enum saturation_req pu_saturation::req
```

5.132.1.3 wSaturation

```
uint16_t pu_saturation::wSaturation
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

Kneron Confidential

5.133 pu_sharpness Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum sharpness_req req
- uint8_t caps
- uint16_t wSharpness

5.133.1 Field Documentation

5.133.1.1 caps

`uint8_t pu_sharpness::caps`

5.133.1.2 req

`enum sharpness_req pu_sharpness::req`

5.133.1.3 wSharpness

`uint16_t pu_sharpness::wSharpness`

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.134 pu_wbc_auto Struct Reference

`#include <uvc_camera.h>`

Data Fields

- enum wbc_auto_req req
- uint8_t caps
- uint8_t bWhiteBalanceComponentAuto

Kneron Confidential

5.134.1 Field Documentation

5.134.1.1 bWhiteBalanceComponentAuto

```
uint8_t pu_wbc_auto::bWhiteBalanceComponentAuto
```

5.134.1.2 caps

```
uint8_t pu_wbc_auto::caps
```

5.134.1.3 req

```
enum wbc_auto_req pu_wbc_auto::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.135 pu_white_balance_temp Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum wbt_req req
- uint8_t caps
- uint16_t wWhiteBalanceTemperature

5.135.1 Field Documentation

5.135.1.1 caps

```
uint8_t pu_white_balance_temp::caps
```

Kneron Confidential

5.135.1.2 req

```
enum wbt_req pu_white_balance_temp::req
```

5.135.1.3 wWhiteBalanceTemperature

```
uint16_t pu_white_balance_temp::wWhiteBalanceTemperature
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.136 pu_white_balance_temp_auto Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- enum wbt_auto_req req
- uint8_t caps
- uint8_t bWhiteBalanceTemperatureAuto

5.136.1 Field Documentation

5.136.1.1 bWhiteBalanceTemperatureAuto

```
uint8_t pu_white_balance_temp_auto::bWhiteBalanceTemperatureAuto
```

5.136.1.2 caps

```
uint8_t pu_white_balance_temp_auto::caps
```

Kneron Confidential

5.136.1.3 req

```
enum wbt_auto_req pu_white_balance_temp_auto::req
```

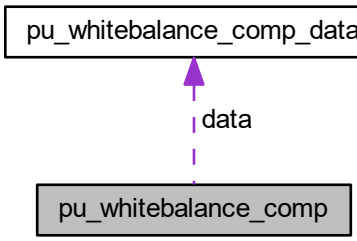
The documentation for this struct was generated from the following file:

- uvc_camera.h

5.137 pu_whitebalance_comp Struct Reference

```
#include <uvc_camera.h>
```

Collaboration diagram for pu_whitebalance_comp:



Data Fields

- enum whitebalance_comp_req req
- uint8_t caps
- struct pu_whitebalance_comp_data data

5.137.1 Field Documentation

5.137.1.1 caps

```
uint8_t pu_whitebalance_comp::caps
```

Kneron Confidential

5.137.1.2 data

```
struct pu_whitebalance_comp_data pu_whitebalance_comp::data
```

5.137.1.3 req

```
enum whitebalance_comp_req pu_whitebalance_comp::req
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.138 pu_whitebalance_comp_data Struct Reference

```
#include <uvc_camera.h>
```

Data Fields

- int16_t wWhiteBalanceBlue
- int16_t wWhiteBalanceRed

5.138.1 Field Documentation

5.138.1.1 wWhiteBalanceBlue

```
int16_t pu_whitebalance_comp_data::wWhiteBalanceBlue
```

5.138.1.2 wWhiteBalanceRed

```
int16_t pu_whitebalance_comp_data::wWhiteBalanceRed
```

The documentation for this struct was generated from the following file:

- uvc_camera.h

5.139 s_kdp_memxfer Struct Reference

```
#include <kmdw_memxfer.h>
```

Kneron Confidential

Data Fields

- int(∗ init)(uint8_t flash_mode, uint8_t mem_mode)
- int(∗ flash_to_ddr)(uint32_t dst, uint32_t src, size_t bytes)
- int(∗ ddr_to_flash)(uint32_t dst, uint32_t src, size_t bytes)
- int(∗ flash_sector_erase64k)(uint32_t addr)
- int(∗ flash_to_niram)(int part_idx)
- uint8_t(∗ flash_get_device_id)(void)

5.139.1 Field Documentation

5.139.1.1 ddr_to_flash

```
int(* s_kdp_memxfer::ddr_to_flash) (uint32_t dst, uint32_t src, size_t bytes)
```

5.139.1.2 flash_get_device_id

```
uint8_t(* s_kdp_memxfer::flash_get_device_id) (void)
```

5.139.1.3 flash_sector_erase64k

```
int(* s_kdp_memxfer::flash_sector_erase64k) (uint32_t addr)
```

5.139.1.4 flash_to_ddr

```
int(* s_kdp_memxfer::flash_to_ddr) (uint32_t dst, uint32_t src, size_t bytes)
```

5.139.1.5 flash_to_niram

```
int(* s_kdp_memxfer::flash_to_niram) (int part_idx)
```

Kneron Confidential

5.139.1.6 init

```
int(* s_kdp_memxfer::init) (uint8_t flash_mode, uint8_t mem_mode)
```

The documentation for this struct was generated from the following file:

- kmdw_memxfer.h

5.140 sensor_datafmt_info Struct Reference

```
#include <kmdw_sensor.h>
```

Data Fields

- uint32_t fourcc
- enum colorspace colorspace

The documentation for this struct was generated from the following file:

- kmdw_sensor.h

5.141 sensor_device Struct Reference

```
#include <kmdw_sensor.h>
```

Data Fields

- uint16_t addr

The documentation for this struct was generated from the following file:

- kmdw_sensor.h

5.142 sensor_init_seq Struct Reference

```
#include <kmdw_sensor.h>
```

Data Fields

- uint16_t addr
- uint8_t value

The documentation for this struct was generated from the following file:

- kmdw_sensor.h

Kneron Confidential

5.143 sensor_ops Struct Reference

`#include <kdev_sensor.h>`

Data Fields

- kdev_status_t(* s_power)(uint32_t on)
- kdev_status_t(* reset)(void)
- kdev_status_t(* s_stream)(uint32_t enable)
- kdev_status_t(* enum_fmt)(uint32_t index, uint32_t *fourcc)
- kdev_status_t(* get_fmt)(struct cam_format *format)
- kdev_status_t(* set_fmt)(struct cam_format *format)
- kdev_status_t(* set_gain)(uint32_t gain1, uint32_t gain2)
- kdev_status_t(* set_aec)(struct cam_sensor_aec *aec_p)
- kdev_status_t(* set_exp_time)(uint32_t gain1, uint32_t gain2)
- kdev_status_t(* get_lux)(uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain, uint8_t *global_gain, uint8_t *y_average)
- kdev_status_t(* led_switch)(uint32_t on)
- kdev_status_t(* set_mirror)(uint32_t enable)
- kdev_status_t(* set_flip)(uint32_t enable)
- uint32_t(* get_dev_id)(void)

5.143.1 Field Documentation

5.143.1.1 enum_fmt

```
kdev_status_t(* sensor_ops::enum_fmt) (uint32_t index, uint32_t *fourcc)
```

5.143.1.2 get_dev_id

```
uint32_t(* sensor_ops::get_dev_id) (void)
```

5.143.1.3 get_fmt

```
kdev_status_t(* sensor_ops::get_fmt) (struct cam_format *format)
```

Kneron Confidential

5.143.1.4 get_lux

```
kdev_status_t(* sensor_ops::get_lux) (uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain,
uint8_t *global_gain, uint8_t *y_average)
```

5.143.1.5 led_switch

```
kdev_status_t(* sensor_ops::led_switch) (uint32_t on)
```

5.143.1.6 reset

```
kdev_status_t(* sensor_ops::reset) (void)
```

5.143.1.7 s_power

```
kdev_status_t(* sensor_ops::s_power) (uint32_t on)
```

5.143.1.8 s_stream

```
kdev_status_t(* sensor_ops::s_stream) (uint32_t enable)
```

5.143.1.9 set_aec

```
kdev_status_t(* sensor_ops::set_aec) (struct cam_sensor_aec *aec_p)
```

5.143.1.10 set_exp_time

```
kdev_status_t(* sensor_ops::set_exp_time) (uint32_t gain1, uint32_t gain2)
```

Kneron Confidential

5.143.1.11 set_flip

```
kdev_status_t(* sensor_ops::set_flip) (uint32_t enable)
```

5.143.1.12 set_fmt

```
kdev_status_t(* sensor_ops::set_fmt) (struct cam_format *format)
```

5.143.1.13 set_gain

```
kdev_status_t(* sensor_ops::set_gain) (uint32_t gain1, uint32_t gain2)
```

5.143.1.14 set_mirror

```
kdev_status_t(* sensor_ops::set_mirror) (uint32_t enable)
```

The documentation for this struct was generated from the following file:

- kdev_sensor.h

5.144 sensor_win_size Struct Reference

```
#include <kmdw_sensor.h>
```

Data Fields

- uint32_t width
- uint32_t height

The documentation for this struct was generated from the following file:

- kmdw_sensor.h

5.145 spi_flash_t Struct Reference

```
#include <kdrv_SPI020.h>
```

Kneron Confidential

Data Fields

- uint8_t reserved
- uint8_t manufacturer
- uint16_t flash_id
- uint32_t flash_size
- uint8_t support_dual
- uint8_t sys_version
- uint8_t dev_mode
- uint8_t vender_specific

5.145.1 Field Documentation

5.145.1.1 dev_mode

```
uint8_t spi_flash_t::dev_mode
```

5.145.1.2 flash_id

```
uint16_t spi_flash_t::flash_id
```

5.145.1.3 flash_size

```
uint32_t spi_flash_t::flash_size
```

5.145.1.4 manufacturer

```
uint8_t spi_flash_t::manufacturer
```

5.145.1.5 reserved

```
uint8_t spi_flash_t::reserved
```

Kneron Confidential

5.145.1.6 support_dual

```
uint8_t spi_flash_t::support_dual
```

5.145.1.7 sys_version

```
uint8_t spi_flash_t::sys_version
```

5.145.1.8 vender_specific

```
uint8_t spi_flash_t::vender_specific
```

The documentation for this struct was generated from the following file:

- kdrv_SPI020.h

5.146 start_stop_layer Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bUpdate

5.146.1 Field Documentation

5.146.1.1 bUpdate

```
uint8_t start_stop_layer::bUpdate
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

5.147 Status_DWord Struct Reference

Data Fields

- uint32_t status: 8
- uint32_t PID: 2
- uint32_t CERR: 2
- uint32_t c_page: 3
- uint32_t ioc: 1
- uint32_t total_bytes_txfer: 15
- uint32_t dt: 1

5.147.1 Field Documentation

5.147.1.1 c_page

```
uint32_t Status_DWord::c_page
```

5.147.1.2 CERR

```
uint32_t Status_DWord::CERR
```

5.147.1.3 dt

```
uint32_t Status_DWord::dt
```

5.147.1.4 ioc

```
uint32_t Status_DWord::ioc
```

5.147.1.5 PID

```
uint32_t Status_DWord::PID
```

Kneron Confidential

5.147.1.6 status

```
uint32_t Status_DWord::status
```

5.147.1.7 total_bytes_txfer

```
uint32_t Status_DWord::total_bytes_txfer
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

5.148 Transaction_DWord Struct Reference

Data Fields

- uint32_t offset: 12
- uint32_t pg: 3
- uint32_t ioc: 1
- uint32_t length: 12
- uint32_t status: 4

5.148.1 Field Documentation

5.148.1.1 ioc

```
uint32_t Transaction_DWord::ioc
```

5.148.1.2 length

```
uint32_t Transaction_DWord::length
```

5.148.1.3 offset

```
uint32_t Transaction_DWord::offset
```

Kneron Confidential

5.148.1.4 pg

```
uint32_t Transaction_DWord::pg
```

5.148.1.5 status

```
uint32_t Transaction_DWord::status
```

The documentation for this struct was generated from the following file:

- kdrv_usbh.c

5.149 U_regDPI2AHB Union Reference

Data Fields

- struct {
  uint32_t D2ACtrl
  uint32_t D2AFrameNumCtrl
  uint32_t D2APage0Addr
  uint32_t D2APage1Addr
  uint32_t D2AIntrCtrl
  uint32_t D2AIntrStat
  uint32_t D2AStatus
  uint32_t D2APacketType
  uint32_t FTCInternal1
  uint32_t FTCInternal2
  uint32_t TAVGRn
  } dw

- struct {
  uint32_t D2a_sr:1
  uint32_t D2a_hp:1
  uint32_t D2a_vp:1
  uint32_t D2a_da:1
  uint32_t D2a_fr_df:7
  uint32_t D2a_ft:4
  uint32_t Gamma_en:1
  uint32_t Tile_avg_size:2
  uint32_t Reserve0:14
  uint32_t D2a_fn:1
  uint32_t Tile_avg_en:1
  uint32_t Reserve1:30
  uint32_t D2a_pg0_addr:32
  uint32_t D2a_pg1_addr:32
  uint32_t D2a_wrd_ce:1
  uint32_t D2a_fn_ol_ce:1
  uint32_t D2a_fifo_of_ce:1
  uint32_t D2a_fifo_uf_ce:1
  uint32_t Ahb_tx_err_ce:1
  uint32_t Tile_avg_d_ce:1

Kneron Confidential

```
        uint32_t Reserve2:26
        uint32_t D2a_wrd:1
        uint32_t D2a_fn_ol:1
        uint32_t D2a_fifo_of:1
        uint32_t D2a_fifo_uf:1
        uint32_t Ahb_tx_err:1
        uint32_t tile_avg_d:1
        uint32_t Reserve3:26
        uint32_t D2a_pg1_b:1
        uint32_t D2a_pg0_b:1
        uint32_t Reserve4:30
        uint32_t D2a_pt:6
        uint32_t Reserve5:26
        uint32_t D2a_fn_tg:1
        uint32_t Reserve6:31
        uint32_t cs:5
        uint32_t Reserve7:27
        uint32_t TAVGR1_0:8
        uint32_t TAVGR1_1:8
        uint32_t TAVGR1_2:8
        uint32_t TAVGR1_3:8
    } bf
```

5.149.1 Field Documentation

5.149.1.1 Ahb_tx_err

```
uint32_t U_regDPI2AHB::Ahb_tx_err
```

5.149.1.2 Ahb_tx_err_ce

```
uint32_t U_regDPI2AHB::Ahb_tx_err_ce
```

5.149.1.3 bf

```
struct { ... } U_regDPI2AHB::bf
```

5.149.1.4 cs

```
uint32_t U_regDPI2AHB::cs
```

Kneron Confidential

5.149.1.5 D2a_da

```
uint32_t U_regDPI2AHB::D2a_da
```

5.149.1.6 D2a_fifo_of

```
uint32_t U_regDPI2AHB::D2a_fifo_of
```

5.149.1.7 D2a_fifo_of_ce

```
uint32_t U_regDPI2AHB::D2a_fifo_of_ce
```

5.149.1.8 D2a_fifo_uf

```
uint32_t U_regDPI2AHB::D2a_fifo_uf
```

5.149.1.9 D2a_fifo_uf_ce

```
uint32_t U_regDPI2AHB::D2a_fifo_uf_ce
```

5.149.1.10 D2a_fn

```
uint32_t U_regDPI2AHB::D2a_fn
```

5.149.1.11 D2a_fn_ol

```
uint32_t U_regDPI2AHB::D2a_fn_ol
```

5.149.1.12 D2a_fn_ol_ce

```
uint32_t U_regDPI2AHB::D2a_fn_ol_ce
```

Kneron Confidential

5.149.1.13 D2a_fn_tg

```
uint32_t U_regDPI2AHB::D2a_fn_tg
```

5.149.1.14 D2a_fr_df

```
uint32_t U_regDPI2AHB::D2a_fr_df
```

5.149.1.15 D2a_ft

```
uint32_t U_regDPI2AHB::D2a_ft
```

5.149.1.16 D2a_hp

```
uint32_t U_regDPI2AHB::D2a_hp
```

5.149.1.17 D2a_pg0_addr

```
uint32_t U_regDPI2AHB::D2a_pg0_addr
```

5.149.1.18 D2a_pg0_b

```
uint32_t U_regDPI2AHB::D2a_pg0_b
```

5.149.1.19 D2a_pg1_addr

```
uint32_t U_regDPI2AHB::D2a_pg1_addr
```

5.149.1.20 D2a_pg1_b

```
uint32_t U_regDPI2AHB::D2a_pg1_b
```

Kneron Confidential

5.149.1.21 D2a_pt

```
uint32_t U_regDPI2AHB::D2a_pt
```

5.149.1.22 D2a_sr

```
uint32_t U_regDPI2AHB::D2a_sr
```

5.149.1.23 D2a_vp

```
uint32_t U_regDPI2AHB::D2a_vp
```

5.149.1.24 D2a_wrd

```
uint32_t U_regDPI2AHB::D2a_wrd
```

5.149.1.25 D2a_wrd_ce

```
uint32_t U_regDPI2AHB::D2a_wrd_ce
```

5.149.1.26 D2ACtrl

```
uint32_t U_regDPI2AHB::D2ACtrl
```

5.149.1.27 D2AFrameNumCtrl

```
uint32_t U_regDPI2AHB::D2AFrameNumCtrl
```

5.149.1.28 D2AIntrCtrl

```
uint32_t U_regDPI2AHB::D2AIntrCtrl
```

Kneron Confidential

5.149.1.29 D2AIntrStat

```
uint32_t U_regDPI2AHB::D2AIntrStat
```

5.149.1.30 D2APacketType

```
uint32_t U_regDPI2AHB::D2APacketType
```

5.149.1.31 D2APage0Addr

```
uint32_t U_regDPI2AHB::D2APage0Addr
```

5.149.1.32 D2APage1Addr

```
uint32_t U_regDPI2AHB::D2APage1Addr
```

5.149.1.33 D2AStatus

```
uint32_t U_regDPI2AHB::D2AStatus
```

5.149.1.34 dw

```
struct { ... } U_regDPI2AHB::dw
```

5.149.1.35 FTCInternal1

```
uint32_t U_regDPI2AHB::FTCInternal1
```

5.149.1.36 FTCInternal2

```
uint32_t U_regDPI2AHB::FTCInternal2
```

Kneron Confidential

5.149.1.37 Gamma_en

```
uint32_t U_regDPI2AHB::Gamma_en
```

5.149.1.38 Reserve0

```
uint32_t U_regDPI2AHB::Reserve0
```

5.149.1.39 Reserve1

```
uint32_t U_regDPI2AHB::Reserve1
```

5.149.1.40 Reserve2

```
uint32_t U_regDPI2AHB::Reserve2
```

5.149.1.41 Reserve3

```
uint32_t U_regDPI2AHB::Reserve3
```

5.149.1.42 Reserve4

```
uint32_t U_regDPI2AHB::Reserve4
```

5.149.1.43 Reserve5

```
uint32_t U_regDPI2AHB::Reserve5
```

5.149.1.44 Reserve6

```
uint32_t U_regDPI2AHB::Reserve6
```

Kneron Confidential

5.149.1.45 Reserve7

```
uint32_t U_regDPI2AHB::Reserve7
```

5.149.1.46 TAVGR1_0

```
uint32_t U_regDPI2AHB::TAVGR1_0
```

5.149.1.47 TAVGR1_1

```
uint32_t U_regDPI2AHB::TAVGR1_1
```

5.149.1.48 TAVGR1_2

```
uint32_t U_regDPI2AHB::TAVGR1_2
```

5.149.1.49 TAVGR1_3

```
uint32_t U_regDPI2AHB::TAVGR1_3
```

5.149.1.50 TAVGRn

```
uint32_t U_regDPI2AHB::TAVGRn
```

5.149.1.51 tile_avg_d

```
uint32_t U_regDPI2AHB::tile_avg_d
```

5.149.1.52 Tile_avg_d_ce

```
uint32_t U_regDPI2AHB::Tile_avg_d_ce
```

Kneron Confidential

5.149.1.53 Tile_avg_en

```
uint32_t U_regDPI2AHB::Tile_avg_en
```

5.149.1.54 Tile_avg_size

```
uint32_t U_regDPI2AHB::Tile_avg_size
```

The documentation for this union was generated from the following file:

- kdrv_dpi2ahb.c

5.150 U_regDPI2AHBCtrl Union Reference

Data Fields

- struct {
    uint32_t D2A_CtrlReg
  } dw

- struct {
    uint32_t pwr_rst_n:1
    uint32_t sys_rst_n:1
    uint32_t ahb_rst_n:1
    uint32_t rst_n:1
    uint32_t pwr_rst_n_1:1
    uint32_t sys_rst_n_1:1
    uint32_t ahb_rst_n_1:1
    uint32_t rst_n_1:1
    uint32_t Reserve0:24
  } bf

5.150.1 Field Documentation

5.150.1.1 ahb_rst_n

```
uint32_t U_regDPI2AHBCtrl::ahb_rst_n
```

Kneron Confidential

5.150.1.2 ahb_rst_n_1

```
uint32_t U_regDPI2AHBCtrl::ahb_rst_n_1
```

5.150.1.3 bf

```
struct { ... } U_regDPI2AHBCtrl::bf
```

5.150.1.4 D2A_CtrlReg

```
uint32_t U_regDPI2AHBCtrl::D2A_CtrlReg
```

5.150.1.5 dw

```
struct { ... } U_regDPI2AHBCtrl::dw
```

5.150.1.6 pwr_rst_n

```
uint32_t U_regDPI2AHBCtrl::pwr_rst_n
```

5.150.1.7 pwr_rst_n_1

```
uint32_t U_regDPI2AHBCtrl::pwr_rst_n_1
```

5.150.1.8 Reserve0

```
uint32_t U_regDPI2AHBCtrl::Reserve0
```

5.150.1.9 rst_n

```
uint32_t U_regDPI2AHBCtrl::rst_n
```

Kneron Confidential

5.150.1.10 rst_n_1

```
uint32_t U_regDPI2AHBCtrl::rst_n_1
```

5.150.1.11 sys_rst_n

```
uint32_t U_regDPI2AHBCtrl::sys_rst_n
```

5.150.1.12 sys_rst_n_1

```
uint32_t U_regDPI2AHBCtrl::sys_rst_n_1
```

The documentation for this union was generated from the following file:

- kdrv_dpi2ahb.c

5.151 U_regGDMA Union Reference

Data Fields

- struct {
  uint32_t INT
  uint32_t INT_TC
  uint32_t INT_TC_CLR
  uint32_t INT_ERR_ABT
  uint32_t INT_ERR_ABT_CLR
  uint32_t TC
  uint32_t ERR_ABT
  uint32_t CH_EN
  uint32_t CH_BUSY
  uint32_t CSR
  uint32_t SYNC
  uint32_t DMAC_REVISION
  uint32_t DMAC_FEATURE
  } dw

5.151.1 Field Documentation

Kneron Confidential

5.151.1.1 CH_BUSY

```
uint32_t U_regGDMA::CH_BUSY
```

5.151.1.2 CH_EN

```
uint32_t U_regGDMA::CH_EN
```

5.151.1.3 CSR

```
uint32_t U_regGDMA::CSR
```

5.151.1.4 DMAC_FEATURE

```
uint32_t U_regGDMA::DMAC_FEATURE
```

5.151.1.5 DMAC_REVISION

```
uint32_t U_regGDMA::DMAC_REVISION
```

5.151.1.6 dw

```
struct { ... } U_regGDMA::dw
```

5.151.1.7 ERR_ABT

```
uint32_t U_regGDMA::ERR_ABT
```

5.151.1.8 INT

```
uint32_t U_regGDMA::INT
```

Kneron Confidential

5.151.1.9 INT_ERR_ABT

```
uint32_t U_regGDMA::INT_ERR_ABT
```

5.151.1.10 INT_ERR_ABT_CLR

```
uint32_t U_regGDMA::INT_ERR_ABT_CLR
```

5.151.1.11 INT_TC

```
uint32_t U_regGDMA::INT_TC
```

5.151.1.12 INT_TC_CLR

```
uint32_t U_regGDMA::INT_TC_CLR
```

5.151.1.13 SYNC

```
uint32_t U_regGDMA::SYNC
```

5.151.1.14 TC

```
uint32_t U_regGDMA::TC
```

The documentation for this union was generated from the following file:

- kdrv_gdma.c

Kneron Confidential

5.152 U_regGDMA_CH Union Reference

Data Fields

- struct {
  uint32_t CSR
  uint32_t CFG
  uint32_t SrcAddr
  uint32_t DstAddr
  uint32_t LLP
  uint32_t SIZE
  uint32_t reserved [2]
  } dw

- struct {
  uint32_t CSR_CH_EN: 1
  uint32_t CSR_DST_SEL: 1
  uint32_t CSR_SRC_SEL: 1
  uint32_t CSR_DSTAD_CTL: 2
  uint32_t CSR_SRCAD_CTL: 2
  uint32_t CSR_MODE: 1
  uint32_t CSR_DST_WIDTH: 3
  uint32_t CSR_SRC_WIDTH: 3
  uint32_t CSR_reserved0: 1
  uint32_t CSR_ABT: 1
  uint32_t CSR_SRC_SIZE: 3
  uint32_t CSR_PROT1: 1
  uint32_t CSR_PROT2: 1
  uint32_t CSR_PROT3: 1
  uint32_t CSR_PRIORITY: 2
  uint32_t CSR_DMA_FF_TH: 3
  uint32_t CSR_reserved2: 4
  uint32_t CSR_TC_MSK: 1
  uint32_t CFG_INT_TC_MSK: 1
  uint32_t CFG_INT_ERR_MSK: 1
  uint32_t CFG_INT_ABT_MSK: 1
  uint32_t CFG_SRC_RS: 4
  uint32_t CFG_SRC_HE: 1
  uint32_t CFG_BUSY: 1
  uint32_t CFG_DST_RS: 4
  uint32_t CFG_DST_HE: 1
  uint32_t CFG_reserved1: 2
  uint32_t CFG_LLP_CNT: 4
  uint32_t CFG_reserved2: 12
  } bf

5.152.1 Field Documentation

5.152.1.1 bf

```
struct { ... } U_regGDMA_CH::bf
```

Kneron Confidential

5.152.1.2 CFG

uint32_t U_regGDMA_CH::CFG

5.152.1.3 CFG_BUSY

uint32_t U_regGDMA_CH::CFG_BUSY

5.152.1.4 CFG_DST_HE

uint32_t U_regGDMA_CH::CFG_DST_HE

5.152.1.5 CFG_DST_RS

uint32_t U_regGDMA_CH::CFG_DST_RS

5.152.1.6 CFG_INT_ABT_MSK

uint32_t U_regGDMA_CH::CFG_INT_ABT_MSK

5.152.1.7 CFG_INT_ERR_MSK

uint32_t U_regGDMA_CH::CFG_INT_ERR_MSK

5.152.1.8 CFG_INT_TC_MSK

uint32_t U_regGDMA_CH::CFG_INT_TC_MSK

5.152.1.9 CFG_LLP_CNT

uint32_t U_regGDMA_CH::CFG_LLP_CNT

Kneron Confidential

5.152.1.10 CFG_reserved1

uint32_t U_regGDMA_CH::CFG_reserved1

5.152.1.11 CFG_reserved2

uint32_t U_regGDMA_CH::CFG_reserved2

5.152.1.12 CFG_SRC_HE

uint32_t U_regGDMA_CH::CFG_SRC_HE

5.152.1.13 CFG_SRC_RS

uint32_t U_regGDMA_CH::CFG_SRC_RS

5.152.1.14 CSR

uint32_t U_regGDMA_CH::CSR

5.152.1.15 CSR_ABT

uint32_t U_regGDMA_CH::CSR_ABT

5.152.1.16 CSR_CH_EN

uint32_t U_regGDMA_CH::CSR_CH_EN

5.152.1.17 CSR_DMA_FF_TH

uint32_t U_regGDMA_CH::CSR_DMA_FF_TH

Kneron Confidential

5.152.1.18 CSR_DST_SEL

uint32_t U_regGDMA_CH::CSR_DST_SEL

5.152.1.19 CSR_DST_WIDTH

uint32_t U_regGDMA_CH::CSR_DST_WIDTH

5.152.1.20 CSR_DSTAD_CTL

uint32_t U_regGDMA_CH::CSR_DSTAD_CTL

5.152.1.21 CSR_MODE

uint32_t U_regGDMA_CH::CSR_MODE

5.152.1.22 CSR_PRIORITY

uint32_t U_regGDMA_CH::CSR_PRIORITY

5.152.1.23 CSR_PROT1

uint32_t U_regGDMA_CH::CSR_PROT1

5.152.1.24 CSR_PROT2

uint32_t U_regGDMA_CH::CSR_PROT2

5.152.1.25 CSR_PROT3

uint32_t U_regGDMA_CH::CSR_PROT3

Kneron Confidential

5.152.1.26 CSR_reserved0

uint32_t U_regGDMA_CH::CSR_reserved0

5.152.1.27 CSR_reserved2

uint32_t U_regGDMA_CH::CSR_reserved2

5.152.1.28 CSR_SRC_SEL

uint32_t U_regGDMA_CH::CSR_SRC_SEL

5.152.1.29 CSR_SRC_SIZE

uint32_t U_regGDMA_CH::CSR_SRC_SIZE

5.152.1.30 CSR_SRC_WIDTH

uint32_t U_regGDMA_CH::CSR_SRC_WIDTH

5.152.1.31 CSR_SRCAD_CTL

uint32_t U_regGDMA_CH::CSR_SRCAD_CTL

5.152.1.32 CSR_TC_MSK

uint32_t U_regGDMA_CH::CSR_TC_MSK

5.152.1.33 DstAddr

uint32_t U_regGDMA_CH::DstAddr

Kneron Confidential

5.152.1.34 dw

```
struct { ... } U_regGDMA_CH::dw
```

5.152.1.35 LLP

```
uint32_t U_regGDMA_CH::LLP
```

5.152.1.36 reserved

```
uint32_t U_regGDMA_CH::reserved[2]
```

5.152.1.37 SIZE

```
uint32_t U_regGDMA_CH::SIZE
```

5.152.1.38 SrcAddr

```
uint32_t U_regGDMA_CH::SrcAddr
```

The documentation for this union was generated from the following file:

- kdrv_gdma.c

5.153 U_regGPIO Union Reference

Data Fields

- struct {
  uint32_t DOUT_OFFSET
  uint32_t DIN_OFFSET
  uint32_t PINOUT_OFFSET
  uint32_t PIN_BYPASS
  uint32_t DATASET
  uint32_t DATACLR
  uint32_t PULLENABLE
  uint32_t PULLTYPE
  uint32_t INT_ENABLE
  uint32_t INT_RAWSTATE
  uint32_t INT_MASKSTATE
  uint32_t INT_MASK
  uint32_t INT_CLEAR
  uint32_t INT_TRIGGER
  uint32_t INT_BOTH
  uint32_t INT_RISENEG
  uint32_t INT_BOUNCEENABLE
  uint32_t INT_PRESCALE
  } dw

Kneron Confidential

5.153.1 Field Documentation

5.153.1.1 DATACLR

```
uint32_t U_regGPIO::DATACLR
```

5.153.1.2 DATASET

```
uint32_t U_regGPIO::DATASET
```

5.153.1.3 DIN_OFFSET

```
uint32_t U_regGPIO::DIN_OFFSET
```

5.153.1.4 DOUT_OFFSET

```
uint32_t U_regGPIO::DOUT_OFFSET
```

5.153.1.5 dw

```
struct { ... } U_regGPIO::dw
```

5.153.1.6 INT_BOTH

```
uint32_t U_regGPIO::INT_BOTH
```

5.153.1.7 INT_BOUNCEENABLE

```
uint32_t U_regGPIO::INT_BOUNCEENABLE
```

Kneron Confidential

5.153.1.8 INT_CLEAR

uint32_t U_regGPIO::INT_CLEAR

5.153.1.9 INT_ENABLE

uint32_t U_regGPIO::INT_ENABLE

5.153.1.10 INT_MASK

uint32_t U_regGPIO::INT_MASK

5.153.1.11 INT_MASKSTATE

uint32_t U_regGPIO::INT_MASKSTATE

5.153.1.12 INT_PRESCALE

uint32_t U_regGPIO::INT_PRESCALE

5.153.1.13 INT_RAWSTATE

uint32_t U_regGPIO::INT_RAWSTATE

5.153.1.14 INT_RISENEG

uint32_t U_regGPIO::INT_RISENEG

5.153.1.15 INT_TRIGGER

uint32_t U_regGPIO::INT_TRIGGER

Kneron Confidential

5.153.1.16 PIN_BYPASS

```
uint32_t U_regGPIO::PIN_BYPASS
```

5.153.1.17 PINOUT_OFFSET

```
uint32_t U_regGPIO::PINOUT_OFFSET
```

5.153.1.18 PULLENABLE

```
uint32_t U_regGPIO::PULLENABLE
```

5.153.1.19 PULLTYPE

```
uint32_t U_regGPIO::PULLTYPE
```

The documentation for this union was generated from the following file:

- kdrv_gpio.c

5.154 U_regTIMER Union Reference

Data Fields

- struct {
  uint32_t Tm1Counter
  uint32_t Tm1Load
  uint32_t Tm1Match1
  uint32_t Tm1Match2
  uint32_t Tm2Counter
  uint32_t Tm2Load
  uint32_t Tm2Match1
  uint32_t Tm2Match2
  uint32_t Tm3Counter
  uint32_t Tm3Load
  uint32_t Tm3Match1
  uint32_t Tm3Match2
  uint32_t TmCR
  uint32_t IntrState
  uint32_t IntrMask
  uint32_t TmRevision
  } dw

Kneron Confidential

- struct {
  uint32_t Tm1Counter:32
  uint32_t Tm1Load:32
  uint32_t Tm1Match1:32
  uint32_t Tm1Match2:32
  uint32_t Tm2Counter:32
  uint32_t Tm2Load:32
  uint32_t Tm2Match1:32
  uint32_t Tm2Match2:32
  uint32_t Tm3Counter:32
  uint32_t Tm3Load:32
  uint32_t Tm3Match1:32
  uint32_t Tm3Match2:32
  uint32_t Tm1En:1
  uint32_t Tm1Clock:1
  uint32_t Tm1OfEn:1
  uint32_t Tm2En:1
  uint32_t Tm2Clock:1
  uint32_t Tm2OfEn:1
  uint32_t Tm3En:1
  uint32_t Tm3Clock:1
  uint32_t Tm3OfEn:1
  uint32_t Tm1UpDown:1
  uint32_t Tm2UpDown:1
  uint32_t Tm3UpDown:1
  uint32_t Reserve0:20
  uint32_t STm1Match1:1
  uint32_t STm1Match2:1
  uint32_t STm1Overflow:1
  uint32_t STm2Match1:1
  uint32_t STm2Match2:1
  uint32_t STm2Overflow:1
  uint32_t STm3Match1:1
  uint32_t STm3Match2:1
  uint32_t STm3Overflow:1
  uint32_t Reserve1:23
  uint32_t MTm1Match1:1
  uint32_t MTm1Match2:1
  uint32_t MTm1Overflow:1
  uint32_t MTm2Match1:1
  uint32_t MTm2Match2:1
  uint32_t MTm2Overflow:1
  uint32_t MTm3Match1:1
  uint32_t MTm3Match2:1
  uint32_t MTm3Overflow:1
  uint32_t Reserve2:23
  } bf

5.154.1 Field Documentation

5.154.1.1 bf

```
struct { ... } U_regTIMER::bf
```

Kneron Confidential

5.154.1.2 dw

```
struct { ...  } U_regTIMER::dw
```

5.154.1.3 IntrMask

```
uint32_t U_regTIMER::IntrMask
```

5.154.1.4 IntrState

```
uint32_t U_regTIMER::IntrState
```

5.154.1.5 MTm1Match1

```
uint32_t U_regTIMER::MTm1Match1
```

5.154.1.6 MTm1Match2

```
uint32_t U_regTIMER::MTm1Match2
```

5.154.1.7 MTm1Overflow

```
uint32_t U_regTIMER::MTm1Overflow
```

5.154.1.8 MTm2Match1

```
uint32_t U_regTIMER::MTm2Match1
```

5.154.1.9 MTm2Match2

```
uint32_t U_regTIMER::MTm2Match2
```

Kneron Confidential

5.154.1.10 MTm2Overflow

```
uint32_t U_regTIMER::MTm2Overflow
```

5.154.1.11 MTm3Match1

```
uint32_t U_regTIMER::MTm3Match1
```

5.154.1.12 MTm3Match2

```
uint32_t U_regTIMER::MTm3Match2
```

5.154.1.13 MTm3Overflow

```
uint32_t U_regTIMER::MTm3Overflow
```

5.154.1.14 Reserve0

```
uint32_t U_regTIMER::Reserve0
```

5.154.1.15 Reserve1

```
uint32_t U_regTIMER::Reserve1
```

5.154.1.16 Reserve2

```
uint32_t U_regTIMER::Reserve2
```

5.154.1.17 STm1Match1

```
uint32_t U_regTIMER::STm1Match1
```

Kneron Confidential

5.154.1.18 STm1Match2

```
uint32_t U_regTIMER::STm1Match2
```

5.154.1.19 STm1Overflow

```
uint32_t U_regTIMER::STm1Overflow
```

5.154.1.20 STm2Match1

```
uint32_t U_regTIMER::STm2Match1
```

5.154.1.21 STm2Match2

```
uint32_t U_regTIMER::STm2Match2
```

5.154.1.22 STm2Overflow

```
uint32_t U_regTIMER::STm2Overflow
```

5.154.1.23 STm3Match1

```
uint32_t U_regTIMER::STm3Match1
```

5.154.1.24 STm3Match2

```
uint32_t U_regTIMER::STm3Match2
```

5.154.1.25 STm3Overflow

```
uint32_t U_regTIMER::STm3Overflow
```

Kneron Confidential

5.154.1.26 Tm1Clock

```
uint32_t U_regTIMER::Tm1Clock
```

5.154.1.27 Tm1Counter

```
uint32_t U_regTIMER::Tm1Counter
```

5.154.1.28 Tm1En

```
uint32_t U_regTIMER::Tm1En
```

5.154.1.29 Tm1Load

```
uint32_t U_regTIMER::Tm1Load
```

5.154.1.30 Tm1Match1

```
uint32_t U_regTIMER::Tm1Match1
```

5.154.1.31 Tm1Match2

```
uint32_t U_regTIMER::Tm1Match2
```

5.154.1.32 Tm1OfEn

```
uint32_t U_regTIMER::Tm1OfEn
```

5.154.1.33 Tm1UpDown

```
uint32_t U_regTIMER::Tm1UpDown
```

Kneron Confidential

5.154.1.34 Tm2Clock

```
uint32_t U_regTIMER::Tm2Clock
```

5.154.1.35 Tm2Counter

```
uint32_t U_regTIMER::Tm2Counter
```

5.154.1.36 Tm2En

```
uint32_t U_regTIMER::Tm2En
```

5.154.1.37 Tm2Load

```
uint32_t U_regTIMER::Tm2Load
```

5.154.1.38 Tm2Match1

```
uint32_t U_regTIMER::Tm2Match1
```

5.154.1.39 Tm2Match2

```
uint32_t U_regTIMER::Tm2Match2
```

5.154.1.40 Tm2OfEn

```
uint32_t U_regTIMER::Tm2OfEn
```

5.154.1.41 Tm2UpDown

```
uint32_t U_regTIMER::Tm2UpDown
```

Kneron Confidential

5.154.1.42 Tm3Clock

```
uint32_t U_regTIMER::Tm3Clock
```

5.154.1.43 Tm3Counter

```
uint32_t U_regTIMER::Tm3Counter
```

5.154.1.44 Tm3En

```
uint32_t U_regTIMER::Tm3En
```

5.154.1.45 Tm3Load

```
uint32_t U_regTIMER::Tm3Load
```

5.154.1.46 Tm3Match1

```
uint32_t U_regTIMER::Tm3Match1
```

5.154.1.47 Tm3Match2

```
uint32_t U_regTIMER::Tm3Match2
```

5.154.1.48 Tm3OfEn

```
uint32_t U_regTIMER::Tm3OfEn
```

5.154.1.49 Tm3UpDown

```
uint32_t U_regTIMER::Tm3UpDown
```

Kneron Confidential

5.154.1.50 TmCR

```
uint32_t U_regTIMER::TmCR
```

5.154.1.51 TmRevision

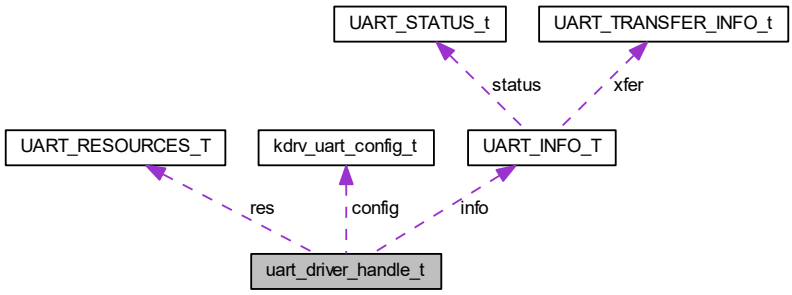
```
uint32_t U_regTIMER::TmRevision
```

The documentation for this union was generated from the following file:

- kdrv_timer.c

5.155 uart_driver_handle_t Struct Reference

Collaboration diagram for uart_driver_handle_t:



Data Fields

- uint32_t uart_port
- uart_state_t state
- kdrv_uart_config_t config
- UART_INFO_T info
- UART_RESOURCES_T res
- int32_t nTimeOutTx
- int32_t nTimeOutRx
- uint32_t iir

5.155.1 Field Documentation

Kneron Confidential

5.155.1.1 config

kdrv_uart_config_t uart_driver_handle_t::config

5.155.1.2 iir

uint32_t uart_driver_handle_t::iir

5.155.1.3 info

UART_INFO_T uart_driver_handle_t::info

5.155.1.4 nTimeOutRx

int32_t uart_driver_handle_t::nTimeOutRx

5.155.1.5 nTimeOutTx

int32_t uart_driver_handle_t::nTimeOutTx

5.155.1.6 res

UART_RESOURCES_T uart_driver_handle_t::res

5.155.1.7 state

uart_state_t uart_driver_handle_t::state

Kneron Confidential

5.155.1.8 uart_port

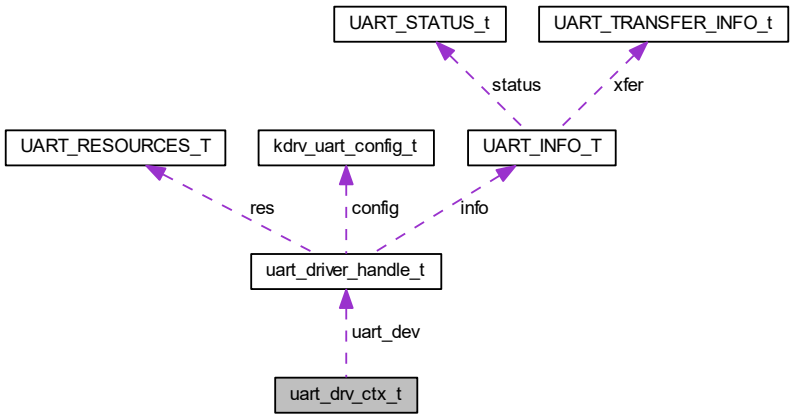
```
uint32_t uart_driver_handle_t::uart_port
```

The documentation for this struct was generated from the following file:

- kdrv_uart.c

5.156 uart_drv_ctx_t Struct Reference

Collaboration diagram for uart_drv_ctx_t:



Data Fields

- int8_t total_open_uarts
- bool active_dev [TOTAL_UART_DEV]
- uart_driver_handle_t * uart_dev [MAX_UART_INST]

5.156.1 Field Documentation

5.156.1.1 active_dev

```
bool uart_drv_ctx_t::active_dev[TOTAL_UART_DEV]
```

5.156.1.2 total_open_uarts

```
int8_t uart_drv_ctx_t::total_open_uarts
```

5.156.1.3 uart_dev

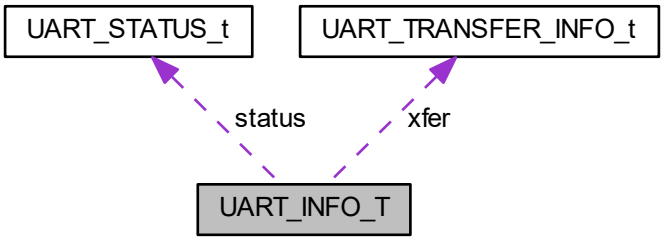
```
uart_driver_handle_t* uart_drv_ctx_t::uart_dev[MAX_UART_INST]
```

The documentation for this struct was generated from the following file:

- kdrv_uart.c

5.157 UART_INFO_T Struct Reference

Collaboration diagram for UART_INFO_T:



Data Fields

- ARM_USART_SignalEvent_t cb_event
- UART_STATUS_t status
- UART_TRANSFER_INFO_t xfer
- uint32_t flags
- uint32_t mode

5.157.1 Field Documentation

Kneron Confidential

5.157.1.1 cb_event

```
ARM_USART_SignalEvent_t UART_INFO_T::cb_event
```

5.157.1.2 flags

```
uint32_t UART_INFO_T::flags
```

5.157.1.3 mode

```
uint32_t UART_INFO_T::mode
```

5.157.1.4 status

```
UART_STATUS_t UART_INFO_T::status
```

5.157.1.5 xfer

```
UART_TRANSFER_INFO_t UART_INFO_T::xfer
```

The documentation for this struct was generated from the following file:

- kdrv_uart.c

5.158 UART_RESOURCES_T Struct Reference

Data Fields

- uint32_t irq_num
- uart_isr_t isr
- uint32_t fifo_depth
- uint32_t tx_fifo_threshold
- uint32_t rx_fifo_threshold
- uint32_t fifo_len
- uint32_t clock
- uint32_t hw_base

Kneron Confidential

5.158.1 Field Documentation

5.158.1.1 clock

```
uint32_t UART_RESOURCES_T::clock
```

5.158.1.2 fifo_depth

```
uint32_t UART_RESOURCES_T::fifo_depth
```

5.158.1.3 fifo_len

```
uint32_t UART_RESOURCES_T::fifo_len
```

5.158.1.4 hw_base

```
uint32_t UART_RESOURCES_T::hw_base
```

5.158.1.5 irq_num

```
uint32_t UART_RESOURCES_T::irq_num
```

5.158.1.6 isr

```
uart_isr_t UART_RESOURCES_T::isr
```

5.158.1.7 rx_fifo_threshold

```
uint32_t UART_RESOURCES_T::rx_fifo_threshold
```

Kneron Confidential

5.158.1.8 tx_fifo_threshold

```
uint32_t UART_RESOURCES_T::tx_fifo_threshold
```

The documentation for this struct was generated from the following file:

- kdrv_uart.c

5.159 UART_STATUS_t Struct Reference

Data Fields

- volatile uint8_t tx_busy
- volatile uint8_t rx_busy
- uint8_t tx_underflow
- uint8_t rx_overflow
- uint8_t rx_break
- uint8_t rx_framing_error
- uint8_t rx_parity_error

5.159.1 Field Documentation

5.159.1.1 rx_break

```
uint8_t UART_STATUS_t::rx_break
```

5.159.1.2 rx_busy

```
volatile uint8_t UART_STATUS_t::rx_busy
```

5.159.1.3 rx_framing_error

```
uint8_t UART_STATUS_t::rx_framing_error
```

5.159.1.4 rx_overflow

```
uint8_t UART_STATUS_t::rx_overflow
```

Kneron Confidential

5.159.1.5 rx_parity_error

```
uint8_t UART_STATUS_t::rx_parity_error
```

5.159.1.6 tx_busy

```
volatile uint8_t UART_STATUS_t::tx_busy
```

5.159.1.7 tx_underflow

```
uint8_t UART_STATUS_t::tx_underflow
```

The documentation for this struct was generated from the following file:

- kdrv_uart.c

5.160 UART_TRANSFER_INFO_t Struct Reference

Data Fields

- volatile uint32_t rx_num
- volatile uint32_t tx_num
- volatile uint8_t * rx_buf
- volatile uint8_t * tx_buf
- volatile uint32_t rx_cnt
- volatile uint32_t tx_cnt

5.160.1 Field Documentation

5.160.1.1 rx_buf

```
volatile uint8_t* UART_TRANSFER_INFO_t::rx_buf
```

5.160.1.2 rx_cnt

```
volatile uint32_t UART_TRANSFER_INFO_t::rx_cnt
```

Kneron Confidential

5.160.1.3 rx_num

```
volatile uint32_t UART_TRANSFER_INFO_t::rx_num
```

5.160.1.4 tx_buf

```
volatile uint8_t* UART_TRANSFER_INFO_t::tx_buf
```

5.160.1.5 tx_cnt

```
volatile uint32_t UART_TRANSFER_INFO_t::tx_cnt
```

5.160.1.6 tx_num

```
volatile uint32_t UART_TRANSFER_INFO_t::tx_num
```

The documentation for this struct was generated from the following file:

- kdrv_uart.c

5.161 usb_bos_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t wTotalLength
- uint8_t bNumDeviceCaps

5.161.1 Field Documentation

Kneron Confidential

5.161.1.1 bDescriptorType

```
uint8_t usb_bos_descriptor::bDescriptorType
```

5.161.1.2 bLength

```
uint8_t usb_bos_descriptor::bLength
```

5.161.1.3 bNumDeviceCaps

```
uint8_t usb_bos_descriptor::bNumDeviceCaps
```

5.161.1.4 wTotalLength

```
uint16_t usb_bos_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

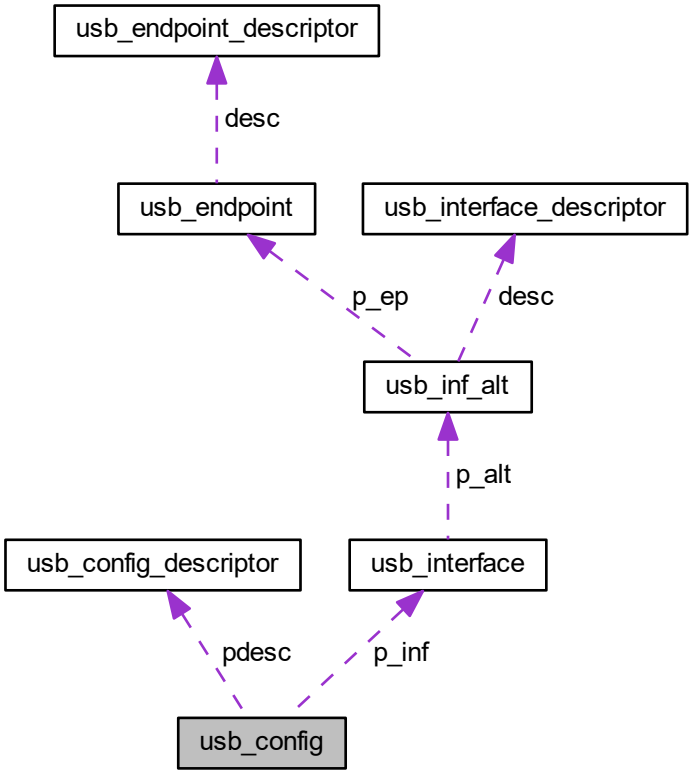
- kdp_usb_ch9.h

5.162 usb_config Struct Reference

```
#include <kdp_usb_api.h>
```

Kneron Confidential

Collaboration diagram for usb_config:



Data Fields

- struct usb_config_descriptor * pdesc
- uint8_t num_inf
- uint8_t cur_inf
- struct usb_interface * p_inf

5.162.1 Field Documentation

5.162.1.1 cur_inf

```
uint8_t usb_config::cur_inf
```

5.162.1.2 num_inf

```
uint8_t usb_config::num_inf
```

5.162.1.3 p_inf

```
struct usb_interface* usb_config::p_inf
```

5.162.1.4 pdesc

```
struct usb_config_descriptor* usb_config::pdesc
```

The documentation for this struct was generated from the following file:

- kdp_usb_api.h

5.163 usb_config_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t wTotalLength
- uint8_t bNumInterfaces
- uint8_t bConfigurationValue
- uint8_t iConfiguration
- uint8_t bmAttributes
- uint8_t bMaxPower

5.163.1 Field Documentation

5.163.1.1 bConfigurationValue

```
uint8_t usb_config_descriptor::bConfigurationValue
```

Kneron Confidential

5.163.1.2 bDescriptorType

```
uint8_t usb_config_descriptor::bDescriptorType
```

5.163.1.3 bLength

```
uint8_t usb_config_descriptor::bLength
```

5.163.1.4 bmAttributes

```
uint8_t usb_config_descriptor::bmAttributes
```

5.163.1.5 bMaxPower

```
uint8_t usb_config_descriptor::bMaxPower
```

5.163.1.6 bNumInterfaces

```
uint8_t usb_config_descriptor::bNumInterfaces
```

5.163.1.7 iConfiguration

```
uint8_t usb_config_descriptor::iConfiguration
```

5.163.1.8 wTotalLength

```
uint16_t usb_config_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

Kneron Confidential

5.164 usb_ctrlrequest Struct Reference

`#include <kdp_usb_ch9.h>`

Data Fields

- uint8_t bRequestType
- uint8_t bRequest
- uint16_t wValue
- uint16_t wIndex
- uint16_t wLength

5.164.1 Detailed Description

struct usb_ctrlrequest - SETUP data for a USB device control request @bRequestType: matches the USB bm↩ RequestType field @bRequest: matches the USB bRequest field @wValue: matches the USB wValue field (le16 byte order) @wIndex: matches the USB wIndex field (le16 byte order) @wLength: matches the USB wLength field (le16 byte order)

This structure is used to send control requests to a USB device. It matches the different fields of the USB 2.0 Spec section 9.3, table 9-2. See the USB spec for a fuller description of the different fields, and what they are used for.

Note that the driver for any interface can issue control requests. For most devices, interfaces don't coordinate with each other, so such requests may be made at any time.

5.164.2 Field Documentation

5.164.2.1 bRequest

`uint8_t usb_ctrlrequest::bRequest`

5.164.2.2 bRequestType

`uint8_t usb_ctrlrequest::bRequestType`

5.164.2.3 wIndex

`uint16_t usb_ctrlrequest::wIndex`

Kneron Confidential

5.164.2.4 wLength

```
uint16_t usb_ctrlrequest::wLength
```

5.164.2.5 wValue

```
uint16_t usb_ctrlrequest::wValue
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.165 usb_debug_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDebugInEndpoint
- uint8_t bDebugOutEndpoint

5.165.1 Field Documentation

5.165.1.1 bDebugInEndpoint

```
uint8_t usb_debug_descriptor::bDebugInEndpoint
```

5.165.1.2 bDebugOutEndpoint

```
uint8_t usb_debug_descriptor::bDebugOutEndpoint
```

Kneron Confidential

5.165.1.3 bDescriptorType

```
uint8_t usb_debug_descriptor::bDescriptorType
```

5.165.1.4 bLength

```
uint8_t usb_debug_descriptor::bLength
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.166 usb_descriptor_header Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType

5.166.1 Field Documentation

5.166.1.1 bDescriptorType

```
uint8_t usb_descriptor_header::bDescriptorType
```

5.166.1.2 bLength

```
uint8_t usb_descriptor_header::bLength
```

The documentation for this struct was generated from the following file:

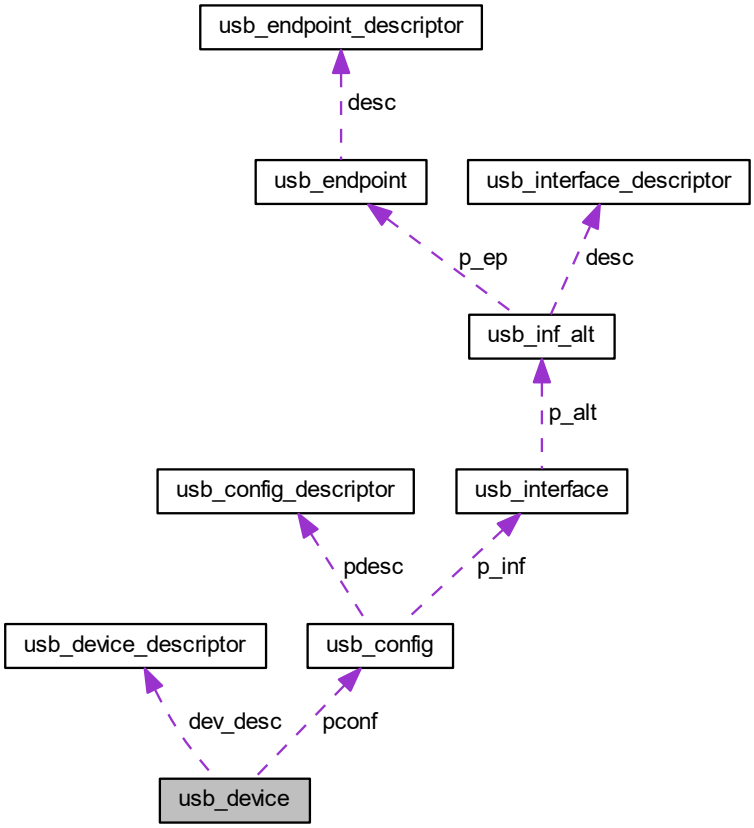
- kdp_usb_ch9.h

Kneron Confidential

5.167 usb_device Struct Reference

`#include <kdp_usb_api.h>`

Collaboration diagram for usb_device:



Data Fields

- struct usb_device_descriptor * dev_desc
- uint8_t num_conf
- uint8_t cur_conf
- struct usb_config * pconf

5.167.1 Field Documentation

5.167.1.1 cur_conf

`uint8_t usb_device::cur_conf`

5.167.1.2 dev_desc

```
struct usb_device_descriptor* usb_device::dev_desc
```

5.167.1.3 num_conf

```
uint8_t usb_device::num_conf
```

5.167.1.4 pconf

```
struct usb_config* usb_device::pconf
```

The documentation for this struct was generated from the following file:

- kdp_usb_api.h

5.168 usb_device_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t bcdUSB
- uint8_t bDeviceClass
- uint8_t bDeviceSubClass
- uint8_t bDeviceProtocol
- uint8_t bMaxPacketSize0
- uint16_t idVendor
- uint16_t idProduct
- uint16_t bcdDevice
- uint8_t iManufacturer
- uint8_t iProduct
- uint8_t iSerialNumber
- uint8_t bNumConfigurations

5.168.1 Field Documentation

Kneron Confidential

5.168.1.1 bcdDevice

```
uint16_t usb_device_descriptor::bcdDevice
```

5.168.1.2 bcdUSB

```
uint16_t usb_device_descriptor::bcdUSB
```

5.168.1.3 bDescriptorType

```
uint8_t usb_device_descriptor::bDescriptorType
```

5.168.1.4 bDeviceClass

```
uint8_t usb_device_descriptor::bDeviceClass
```

5.168.1.5 bDeviceProtocol

```
uint8_t usb_device_descriptor::bDeviceProtocol
```

5.168.1.6 bDeviceSubClass

```
uint8_t usb_device_descriptor::bDeviceSubClass
```

5.168.1.7 bLength

```
uint8_t usb_device_descriptor::bLength
```

5.168.1.8 bMaxPacketSize0

```
uint8_t usb_device_descriptor::bMaxPacketSize0
```

Kneron Confidential

5.168.1.9 bNumConfigurations

```
uint8_t usb_device_descriptor::bNumConfigurations
```

5.168.1.10 idProduct

```
uint16_t usb_device_descriptor::idProduct
```

5.168.1.11 idVendor

```
uint16_t usb_device_descriptor::idVendor
```

5.168.1.12 iManufacturer

```
uint8_t usb_device_descriptor::iManufacturer
```

5.168.1.13 iProduct

```
uint8_t usb_device_descriptor::iProduct
```

5.168.1.14 iSerialNumber

```
uint8_t usb_device_descriptor::iSerialNumber
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.169 usb_device_id Struct Reference

```
#include <kdp_usb.h>
```

Kneron Confidential

Data Fields

- uint16_t idVendor
- uint16_t idProduct

5.169.1 Field Documentation

5.169.1.1 idProduct

```
uint16_t usb_device_id::idProduct
```

5.169.1.2 idVendor

```
uint16_t usb_device_id::idVendor
```

The documentation for this struct was generated from the following file:

- kdp_usb.h

5.170 usb_device_qualifier_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t bcdUSB
- uint8_t bDeviceClass
- uint8_t bDeviceSubClass
- uint8_t bDeviceProtocol
- uint8_t bMaxPacketSize0
- uint8_t bNumConfigurations
- uint8_t bReserved

5.170.1 Field Documentation

Kneron Confidential

5.170.1.1 bcdUSB

```
uint16_t usb_device_qualifier_descriptor::bcdUSB
```

5.170.1.2 bDescriptorType

```
uint8_t usb_device_qualifier_descriptor::bDescriptorType
```

5.170.1.3 bDeviceClass

```
uint8_t usb_device_qualifier_descriptor::bDeviceClass
```

5.170.1.4 bDeviceProtocol

```
uint8_t usb_device_qualifier_descriptor::bDeviceProtocol
```

5.170.1.5 bDeviceSubClass

```
uint8_t usb_device_qualifier_descriptor::bDeviceSubClass
```

5.170.1.6 bLength

```
uint8_t usb_device_qualifier_descriptor::bLength
```

5.170.1.7 bMaxPacketSize0

```
uint8_t usb_device_qualifier_descriptor::bMaxPacketSize0
```

5.170.1.8 bNumConfigurations

```
uint8_t usb_device_qualifier_descriptor::bNumConfigurations
```

Kneron Confidential

5.170.1.9 bReserved

```
uint8_t usb_device_qualifier_descriptor::bReserved
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.171 usb_encryption_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bEncryptionType
- uint8_t bEncryptionValue
- uint8_t bAuthKeyIndex

5.171.1 Field Documentation

5.171.1.1 bAuthKeyIndex

```
uint8_t usb_encryption_descriptor::bAuthKeyIndex
```

5.171.1.2 bDescriptorType

```
uint8_t usb_encryption_descriptor::bDescriptorType
```

5.171.1.3 bEncryptionType

```
uint8_t usb_encryption_descriptor::bEncryptionType
```

Kneron Confidential

5.171.1.4 bEncryptionValue

```
uint8_t usb_encryption_descriptor::bEncryptionValue
```

5.171.1.5 bLength

```
uint8_t usb_encryption_descriptor::bLength
```

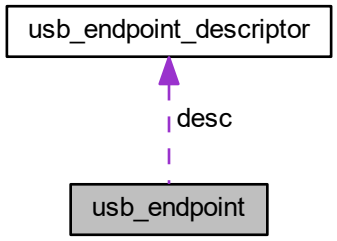
The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.172 usb_endpoint Struct Reference

```
#include <kdp_usb_api.h>
```

Collaboration diagram for usb_endpoint:



Data Fields

- struct usb_endpoint_descriptor desc
- void * hcpriv
- unsigned char * extra
- int extralen
- int enabled
- int streams

5.172.1 Field Documentation

Kneron Confidential

5.172.1.1 desc

```
struct usb_endpoint_descriptor usb_endpoint::desc
```

5.172.1.2 enabled

```
int usb_endpoint::enabled
```

5.172.1.3 extra

```
unsigned char* usb_endpoint::extra
```

5.172.1.4 extralen

```
int usb_endpoint::extralen
```

5.172.1.5 hcpriv

```
void* usb_endpoint::hcpriv
```

5.172.1.6 streams

```
int usb_endpoint::streams
```

The documentation for this struct was generated from the following file:

- kdp_usb_api.h

5.173 usb_endpoint_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bEndpointAddress
- uint8_t bmAttributes
- uint16_t wMaxPacketSize
- uint8_t bInterval
- uint8_t bRefresh
- uint8_t bSynchAddress

5.173.1 Field Documentation

5.173.1.1 bDescriptorType

```
uint8_t usb_endpoint_descriptor::bDescriptorType
```

5.173.1.2 bEndpointAddress

```
uint8_t usb_endpoint_descriptor::bEndpointAddress
```

5.173.1.3 bInterval

```
uint8_t usb_endpoint_descriptor::bInterval
```

5.173.1.4 bLength

```
uint8_t usb_endpoint_descriptor::bLength
```

5.173.1.5 bmAttributes

```
uint8_t usb_endpoint_descriptor::bmAttributes
```

Kneron Confidential

5.173.1.6 bRefresh

```
uint8_t usb_endpoint_descriptor::bRefresh
```

5.173.1.7 bSynchAddress

```
uint8_t usb_endpoint_descriptor::bSynchAddress
```

5.173.1.8 wMaxPacketSize

```
uint16_t usb_endpoint_descriptor::wMaxPacketSize
```

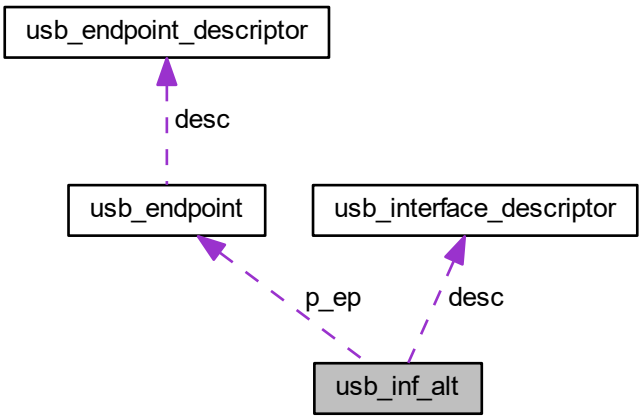
The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.174 usb_inf_alt Struct Reference

```
#include <kdp_usb_api.h>
```

Collaboration diagram for usb_inf_alt:



Kneron Confidential

Data Fields

- struct usb_interface_descriptor desc
- int extralen
- unsigned char * extra
- uint8_t num_ep
- struct usb_endpoint * p_ep

5.174.1 Field Documentation

5.174.1.1 desc

```
struct usb_interface_descriptor usb_inf_alt::desc
```

5.174.1.2 extra

```
unsigned char* usb_inf_alt::extra
```

5.174.1.3 extralen

```
int usb_inf_alt::extralen
```

5.174.1.4 num_ep

```
uint8_t usb_inf_alt::num_ep
```

5.174.1.5 p_ep

```
struct usb_endpoint* usb_inf_alt::p_ep
```

The documentation for this struct was generated from the following file:

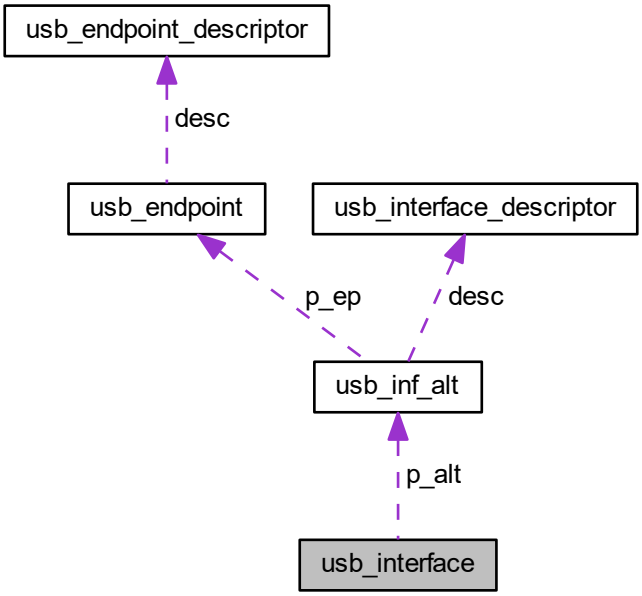
- kdp_usb_api.h

Kneron Confidential

5.175 usb_interface Struct Reference

```
#include <kdp_usb_api.h>
```

Collaboration diagram for usb_interface:



Data Fields

- struct usb_inf_alt * p_alt
- uint8_t num_alt
- uint8_t cur_num

5.175.1 Field Documentation

5.175.1.1 cur_num

```
uint8_t usb_interface::cur_num
```

Kneron Confidential

5.175.1.2 num_alt

```
uint8_t usb_interface::num_alt
```

5.175.1.3 p_alt

```
struct usb_inf_alt* usb_interface::p_alt
```

The documentation for this struct was generated from the following file:

- kdp_usb_api.h

5.176 usb_interface_assoc_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bFirstInterface
- uint8_t bInterfaceCount
- uint8_t bFunctionClass
- uint8_t bFunctionSubClass
- uint8_t bFunctionProtocol
- uint8_t iFunction

5.176.1 Field Documentation

5.176.1.1 bDescriptorType

```
uint8_t usb_interface_assoc_descriptor::bDescriptorType
```

5.176.1.2 bFirstInterface

```
uint8_t usb_interface_assoc_descriptor::bFirstInterface
```

Kneron Confidential

5.176.1.3 bFunctionClass

```
uint8_t usb_interface_assoc_descriptor::bFunctionClass
```

5.176.1.4 bFunctionProtocol

```
uint8_t usb_interface_assoc_descriptor::bFunctionProtocol
```

5.176.1.5 bFunctionSubClass

```
uint8_t usb_interface_assoc_descriptor::bFunctionSubClass
```

5.176.1.6 bInterfaceCount

```
uint8_t usb_interface_assoc_descriptor::bInterfaceCount
```

5.176.1.7 bLength

```
uint8_t usb_interface_assoc_descriptor::bLength
```

5.176.1.8 iFunction

```
uint8_t usb_interface_assoc_descriptor::iFunction
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.177 usb_interface_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bInterfaceNumber
- uint8_t bAlternateSetting
- uint8_t bNumEndpoints
- uint8_t bInterfaceClass
- uint8_t bInterfaceSubClass
- uint8_t bInterfaceProtocol
- uint8_t iInterface

5.177.1 Field Documentation

5.177.1.1 bAlternateSetting

```
uint8_t usb_interface_descriptor::bAlternateSetting
```

5.177.1.2 bDescriptorType

```
uint8_t usb_interface_descriptor::bDescriptorType
```

5.177.1.3 bInterfaceClass

```
uint8_t usb_interface_descriptor::bInterfaceClass
```

5.177.1.4 bInterfaceNumber

```
uint8_t usb_interface_descriptor::bInterfaceNumber
```

5.177.1.5 bInterfaceProtocol

```
uint8_t usb_interface_descriptor::bInterfaceProtocol
```

Kneron Confidential

5.177.1.6 bInterfaceSubClass

```
uint8_t usb_interface_descriptor::bInterfaceSubClass
```

5.177.1.7 bLength

```
uint8_t usb_interface_descriptor::bLength
```

5.177.1.8 bNumEndpoints

```
uint8_t usb_interface_descriptor::bNumEndpoints
```

5.177.1.9 iInterface

```
uint8_t usb_interface_descriptor::iInterface
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.178 usb_key_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t tTKID [3]
- uint8_t bReserved
- uint8_t bKeyData [0]

5.178.1 Field Documentation

Kneron Confidential

5.178.1.1 bDescriptorType

```
uint8_t usb_key_descriptor::bDescriptorType
```

5.178.1.2 bKeyData

```
uint8_t usb_key_descriptor::bKeyData[0]
```

5.178.1.3 bLength

```
uint8_t usb_key_descriptor::bLength
```

5.178.1.4 bReserved

```
uint8_t usb_key_descriptor::bReserved
```

5.178.1.5 tTKID

```
uint8_t usb_key_descriptor::tTKID[3]
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.179 usb_otg20_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bmAttributes
- uint16_t bcdOTG

Kneron Confidential

5.179.1 Field Documentation

5.179.1.1 bcdOTG

```
uint16_t usb_otg20_descriptor::bcdOTG
```

5.179.1.2 bDescriptorType

```
uint8_t usb_otg20_descriptor::bDescriptorType
```

5.179.1.3 bLength

```
uint8_t usb_otg20_descriptor::bLength
```

5.179.1.4 bmAttributes

```
uint8_t usb_otg20_descriptor::bmAttributes
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.180 usb_otg_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bmAttributes

5.180.1 Field Documentation

Kneron Confidential

5.180.1.1 bDescriptorType

```
uint8_t usb_otg_descriptor::bDescriptorType
```

5.180.1.2 bLength

```
uint8_t usb_otg_descriptor::bLength
```

5.180.1.3 bmAttributes

```
uint8_t usb_otg_descriptor::bmAttributes
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.181 usb_other_speed_configuration_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t wTotalLength
- uint8_t bNumInterfaces
- uint8_t bConfigurationValue
- uint8_t iConfiguration
- uint8_t bmAttributes
- uint8_t bMaxPower

5.181.1 Field Documentation

5.181.1.1 bConfigurationValue

```
uint8_t usb_other_speed_configuration_descriptor::bConfigurationValue
```

Kneron Confidential

5.181.1.2 bDescriptorType

```
uint8_t usb_other_speed_configuration_descriptor::bDescriptorType
```

5.181.1.3 bLength

```
uint8_t usb_other_speed_configuration_descriptor::bLength
```

5.181.1.4 bmAttributes

```
uint8_t usb_other_speed_configuration_descriptor::bmAttributes
```

5.181.1.5 bMaxPower

```
uint8_t usb_other_speed_configuration_descriptor::bMaxPower
```

5.181.1.6 bNumInterfaces

```
uint8_t usb_other_speed_configuration_descriptor::bNumInterfaces
```

5.181.1.7 iConfiguration

```
uint8_t usb_other_speed_configuration_descriptor::iConfiguration
```

5.181.1.8 wTotalLength

```
uint16_t usb_other_speed_configuration_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

Kneron Confidential

5.182 usb_qualifier_descriptor Struct Reference

`#include <kdp_usb_ch9.h>`

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t bcdUSB
- uint8_t bDeviceClass
- uint8_t bDeviceSubClass
- uint8_t bDeviceProtocol
- uint8_t bMaxPacketSize0
- uint8_t bNumConfigurations
- uint8_t bRESERVED

5.182.1 Field Documentation

5.182.1.1 bcdUSB

`uint16_t usb_qualifier_descriptor::bcdUSB`

5.182.1.2 bDescriptorType

`uint8_t usb_qualifier_descriptor::bDescriptorType`

5.182.1.3 bDeviceClass

`uint8_t usb_qualifier_descriptor::bDeviceClass`

5.182.1.4 bDeviceProtocol

`uint8_t usb_qualifier_descriptor::bDeviceProtocol`

Kneron Confidential

5.182.1.5 bDeviceSubClass

```
uint8_t usb_qualifier_descriptor::bDeviceSubClass
```

5.182.1.6 bLength

```
uint8_t usb_qualifier_descriptor::bLength
```

5.182.1.7 bMaxPacketSize0

```
uint8_t usb_qualifier_descriptor::bMaxPacketSize0
```

5.182.1.8 bNumConfigurations

```
uint8_t usb_qualifier_descriptor::bNumConfigurations
```

5.182.1.9 bRESERVED

```
uint8_t usb_qualifier_descriptor::bRESERVED
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.183 USB_REQUEST_TYPE Struct Reference

bmRequestType Definition

```
#include <usb_def.h>
```

Data Fields

- uint8_t Recipient: 5
  D4..0: Recipient.
- uint8_t Type: 2
  D6..5: Type.
- uint8_t Dir: 1
  D7: Data Transfer Direction.

Kneron Confidential

5.183.1 Detailed Description

bmRequestType Definition

5.183.2 Field Documentation

5.183.2.1 Dir

`uint8_t USB_REQUEST_TYPE::Dir`

D7: Data Transfer Direction.

5.183.2.2 Recipient

`uint8_t USB_REQUEST_TYPE::Recipient`

D4..0: Recipient.

5.183.2.3 Type

`uint8_t USB_REQUEST_TYPE::Type`

D6..5: Type.

The documentation for this struct was generated from the following file:

- usb_def.h

5.184 usb_security_descriptor Struct Reference

`#include <kdp_usb_ch9.h>`

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t wTotalLength
- uint8_t bNumEncryptionTypes

Kneron Confidential

5.184.1 Field Documentation

5.184.1.1 bDescriptorType

```
uint8_t usb_security_descriptor::bDescriptorType
```

5.184.1.2 bLength

```
uint8_t usb_security_descriptor::bLength
```

5.184.1.3 bNumEncryptionTypes

```
uint8_t usb_security_descriptor::bNumEncryptionTypes
```

5.184.1.4 wTotalLength

```
uint16_t usb_security_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.185 usb_set_sel_req Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t u1_sel
- uint8_t u1_pel
- uint16_t u2_sel
- uint16_t u2_pel

5.185.1 Field Documentation

Kneron Confidential

5.185.1.1 u1_pel

```
uint8_t usb_set_sel_req::u1_pel
```

5.185.1.2 u1_sel

```
uint8_t usb_set_sel_req::u1_sel
```

5.185.1.3 u2_pel

```
uint16_t usb_set_sel_req::u2_pel
```

5.185.1.4 u2_sel

```
uint16_t usb_set_sel_req::u2_sel
```

The documentation for this struct was generated from the following file:

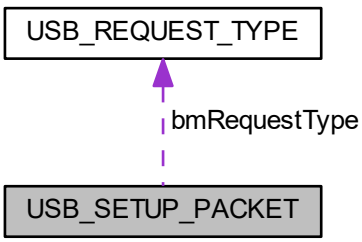
- kdp_usb_ch9.h

5.186 USB_SETUP_PACKET Struct Reference

USB Default Control Pipe Setup Packet.

```
#include <usb_def.h>
```

Collaboration diagram for USB_SETUP_PACKET:



Kneron Confidential

Data Fields

- USB_REQUEST_TYPE bmRequestType

  Characteristics of request.
- uint8_t bRequest

  Specific request.
- uint16_t wValue

  Value according to request.
- uint16_t wIndex

  Index or Offset according to request.
- uint16_t wLength

  Number of bytes to transfer if there is a Data stage.

5.186.1 Detailed Description

USB Default Control Pipe Setup Packet.

5.186.2 Field Documentation

5.186.2.1 bmRequestType

```
USB_REQUEST_TYPE USB_SETUP_PACKET::bmRequestType
```

Characteristics of request.

5.186.2.2 bRequest

```
uint8_t USB_SETUP_PACKET::bRequest
```

Specific request.

5.186.2.3 wIndex

```
uint16_t USB_SETUP_PACKET::wIndex
```

Index or Offset according to request.

Kneron Confidential

5.186.2.4 wLength

```
uint16_t USB_SETUP_PACKET::wLength
```

Number of bytes to transfer if there is a Data stage.

5.186.2.5 wValue

```
uint16_t USB_SETUP_PACKET::wValue
```

Value according to request.

The documentation for this struct was generated from the following file:

- usb_def.h

5.187 usb_ss_ep_comp_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bMaxBurst
- uint8_t bmAttributes
- uint16_t wBytesPerInterval

5.187.1 Field Documentation

5.187.1.1 bDescriptorType

```
uint8_t usb_ss_ep_comp_descriptor::bDescriptorType
```

5.187.1.2 bLength

```
uint8_t usb_ss_ep_comp_descriptor::bLength
```

Kneron Confidential

5.187.1.3 bmAttributes

```
uint8_t usb_ss_ep_comp_descriptor::bmAttributes
```

5.187.1.4 bMaxBurst

```
uint8_t usb_ss_ep_comp_descriptor::bMaxBurst
```

5.187.1.5 wBytesPerInterval

```
uint16_t usb_ss_ep_comp_descriptor::wBytesPerInterval
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.188 usb_ssp_isoc_ep_comp_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t wReseved
- uint32_t dwBytesPerInterval

5.188.1 Field Documentation

5.188.1.1 bDescriptorType

```
uint8_t usb_ssp_isoc_ep_comp_descriptor::bDescriptorType
```

Kneron Confidential

5.188.1.2 bLength

```
uint8_t usb_ssp_isoc_ep_comp_descriptor::bLength
```

5.188.1.3 dwBytesPerInterval

```
uint32_t usb_ssp_isoc_ep_comp_descriptor::dwBytesPerInterval
```

5.188.1.4 wReseved

```
uint16_t usb_ssp_isoc_ep_comp_descriptor::wReseved
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.189 usb_string_descriptor Struct Reference

```
#include <kdp_usb_ch9.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t wData [1]

5.189.1 Field Documentation

5.189.1.1 bDescriptorType

```
uint8_t usb_string_descriptor::bDescriptorType
```

5.189.1.2 bLength

```
uint8_t usb_string_descriptor::bLength
```

Kneron Confidential

5.189.1.3 wData

```
uint16_t usb_string_descriptor::wData[1]
```

The documentation for this struct was generated from the following file:

- kdp_usb_ch9.h

5.190 USBH_Event_t Struct Reference

Data Fields

- uint8_t type
- uint8_t port
- uint32_t event

5.190.1 Field Documentation

5.190.1.1 event

```
uint32_t USBH_Event_t::event
```

5.190.1.2 port

```
uint8_t USBH_Event_t::port
```

5.190.1.3 type

```
uint8_t USBH_Event_t::type
```

The documentation for this struct was generated from the following file:

- kmdw_usbh.c

5.191 USBH_PIPE_TID_t Struct Reference

Data Fields

- USBH_PIPE_HANDLE pipe
- osThreadId_t thread

Kneron Confidential

5.191.1 Field Documentation

5.191.1.1 pipe

```
USBH_PIPE_HANDLE USBH_PIPE_TID_t::pipe
```

5.191.1.2 thread

```
osThreadId_t USBH_PIPE_TID_t::thread
```

The documentation for this struct was generated from the following file:

- kmdw_usbh.c

5.192 uvc_camera_terminal_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bTerminalID
- uint16_t wTerminalType
- uint8_t bAssocTerminal
- uint8_t iTerminal
- uint16_t wObjectiveFocalLengthMin
- uint16_t wObjectiveFocalLengthMax
- uint16_t wOcularFocalLength
- uint8_t bControlSize
- uint8_t bmControls [0]

5.192.1 Field Documentation

5.192.1.1 bAssocTerminal

```
uint8_t uvc_camera_terminal_descriptor::bAssocTerminal
```

Kneron Confidential

5.192.1.2 bControlSize

```
uint8_t uvc_camera_terminal_descriptor::bControlSize
```

5.192.1.3 bDescriptorSubType

```
uint8_t uvc_camera_terminal_descriptor::bDescriptorSubType
```

5.192.1.4 bDescriptorType

```
uint8_t uvc_camera_terminal_descriptor::bDescriptorType
```

5.192.1.5 bLength

```
uint8_t uvc_camera_terminal_descriptor::bLength
```

5.192.1.6 bmControls

```
uint8_t uvc_camera_terminal_descriptor::bmControls[0]
```

5.192.1.7 bTerminalID

```
uint8_t uvc_camera_terminal_descriptor::bTerminalID
```

5.192.1.8 iTerminal

```
uint8_t uvc_camera_terminal_descriptor::iTerminal
```

5.192.1.9 wObjectiveFocalLengthMax

```
uint16_t uvc_camera_terminal_descriptor::wObjectiveFocalLengthMax
```

Kneron Confidential

5.192.1.10 wObjectiveFocalLengthMin

```
uint16_t uvc_camera_terminal_descriptor::wObjectiveFocalLengthMin
```

5.192.1.11 wOcularFocalLength

```
uint16_t uvc_camera_terminal_descriptor::wOcularFocalLength
```

5.192.1.12 wTerminalType

```
uint16_t uvc_camera_terminal_descriptor::wTerminalType
```

The documentation for this struct was generated from the following file:

- uvc.h

5.193 uvc_color_mat_desc Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint8_t bColorPrimaries
- uint8_t bTransferCharacteristics
- uint8_t bMatrixCoefficients

5.193.1 Field Documentation

5.193.1.1 bColorPrimaries

```
uint8_t uvc_color_mat_desc::bColorPrimaries
```

5.193.1.2 bMatrixCoefficients

```
uint8_t uvc_color_mat_desc::bMatrixCoefficients
```

Kneron Confidential

5.193.1.3 bTransferCharacteristics

```
uint8_t uvc_color_mat_desc::bTransferCharacteristics
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.194 uvc_color_matching_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bColorPrimaries
- uint8_t bTransferCharacteristics
- uint8_t bMatrixCoefficients

5.194.1 Field Documentation

5.194.1.1 bColorPrimaries

```
uint8_t uvc_color_matching_descriptor::bColorPrimaries
```

5.194.1.2 bDescriptorSubType

```
uint8_t uvc_color_matching_descriptor::bDescriptorSubType
```

5.194.1.3 bDescriptorType

```
uint8_t uvc_color_matching_descriptor::bDescriptorType
```

Kneron Confidential

5.194.1.4 bLength

```
uint8_t uvc_color_matching_descriptor::bLength
```

5.194.1.5 bMatrixCoefficients

```
uint8_t uvc_color_matching_descriptor::bMatrixCoefficients
```

5.194.1.6 bTransferCharacteristics

```
uint8_t uvc_color_matching_descriptor::bTransferCharacteristics
```

The documentation for this struct was generated from the following file:

- uvc.h

5.195 uvc_control Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint8_t entity_id
- uint8_t index
- uint8_t size
- uint8_t flags

5.195.1 Field Documentation

5.195.1.1 entity_id

```
uint8_t uvc_control::entity_id
```

Kneron Confidential

5.195.1.2 flags

```
uint8_t uvc_control::flags
```

5.195.1.3 index

```
uint8_t uvc_control::index
```

5.195.1.4 size

```
uint8_t uvc_control::size
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.196 uvc_control_endpoint_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint16_t wMaxTransferSize

5.196.1 Field Documentation

5.196.1.1 bDescriptorSubType

```
uint8_t uvc_control_endpoint_descriptor::bDescriptorSubType
```

Kneron Confidential

5.196.1.2 bDescriptorType

```
uint8_t uvc_control_endpoint_descriptor::bDescriptorType
```

5.196.1.3 bLength

```
uint8_t uvc_control_endpoint_descriptor::bLength
```

5.196.1.4 wMaxTransferSize

```
uint16_t uvc_control_endpoint_descriptor::wMaxTransferSize
```

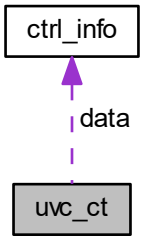
The documentation for this struct was generated from the following file:

- uvc.h

5.197 uvc_ct Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_ct:



Kneron Confidential

Data Fields

- uint16_t wObjectiveFocalLengthMin
- uint16_t wObjectiveFocalLengthMax
- uint16_t wOcularFocalLength
- uint32_t bmControls
- struct ctrl_info * data

5.197.1 Field Documentation

5.197.1.1 bmControls

```
uint32_t uvc_ct::bmControls
```

5.197.1.2 data

```
struct ctrl_info* uvc_ct::data
```

5.197.1.3 wObjectiveFocalLengthMax

```
uint16_t uvc_ct::wObjectiveFocalLengthMax
```

5.197.1.4 wObjectiveFocalLengthMin

```
uint16_t uvc_ct::wObjectiveFocalLengthMin
```

5.197.1.5 wOcularFocalLength

```
uint16_t uvc_ct::wOcularFocalLength
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.198 uvc_decode_op Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- struct uvc_buffer * buf
- void * dst
- const uint8_t * src
- int len

Kneron Confidential

5.198.1 Field Documentation

5.198.1.1 buf

```
struct uvc_buffer* uvc_decode_op::buf
```

5.198.1.2 dst

```
void* uvc_decode_op::dst
```

5.198.1.3 len

```
int uvc_decode_op::len
```

5.198.1.4 src

```
const uint8_t* uvc_decode_op::src
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.199 uvc_descriptor_header Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType

5.199.1 Field Documentation

Kneron Confidential

5.199.1.1 bDescriptorSubType

```
uint8_t uvc_descriptor_header::bDescriptorSubType
```

5.199.1.2 bDescriptorType

```
uint8_t uvc_descriptor_header::bDescriptorType
```

5.199.1.3 bLength

```
uint8_t uvc_descriptor_header::bLength
```

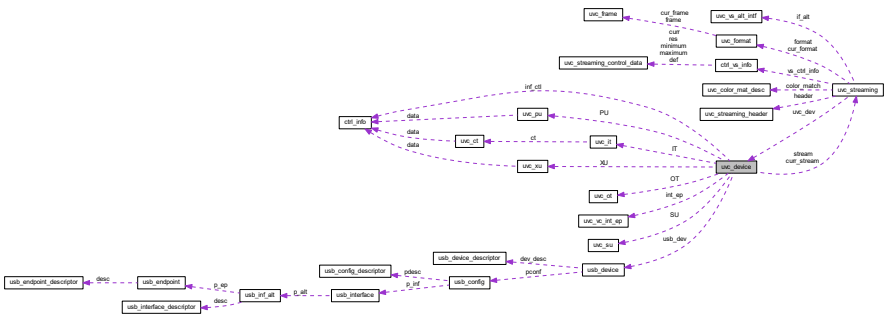
The documentation for this struct was generated from the following file:

- uvc.h

5.200 uvc_device Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_device:



Kneron Confidential

Data Fields

- char name [32]
- uint32_t device_caps
- struct usb_device * usb_dev
- uint8_t num_inf
- uint8_t vc_inf
- struct ctrl_info * inf_ctl
- uint16_t uvc_version
- uint32_t clock_frequency
- uint8_t nITs
- struct uvc_it * IT
- uint8_t nOTs
- struct uvc_ot * OT
- uint8_t nSUs
- struct uvc_su * SU
- uint8_t nPUs
- struct uvc_pu * PU
- uint8_t nXUs
- struct uvc_xu * XU
- uint8_t num_vs_inf
- struct uvc_streaming * curr_stream
- struct uvc_streaming * stream
- struct uvc_vc_int_ep * int_ep
- uint8_t * status
- bool inited
- bool opened

5.200.1 Field Documentation

5.200.1.1 clock_frequency

```
uint32_t uvc_device::clock_frequency
```

5.200.1.2 curr_stream

```
struct uvc_streaming* uvc_device::curr_stream
```

5.200.1.3 device_caps

```
uint32_t uvc_device::device_caps
```

Kneron Confidential

5.200.1.4 inf_ctl

```
struct ctrl_info* uvc_device::inf_ctl
```

5.200.1.5 inited

```
bool uvc_device::inited
```

5.200.1.6 int_ep

```
struct uvc_vc_int_ep* uvc_device::int_ep
```

5.200.1.7 IT

```
struct uvc_it* uvc_device::IT
```

5.200.1.8 name

```
char uvc_device::name[32]
```

5.200.1.9 nITs

```
uint8_t uvc_device::nITs
```

5.200.1.10 nOTs

```
uint8_t uvc_device::nOTs
```

5.200.1.11 nPUs

```
uint8_t uvc_device::nPUs
```

Kneron Confidential

5.200.1.12 nSUs

```
uint8_t uvc_device::nSUs
```

5.200.1.13 num_inf

```
uint8_t uvc_device::num_inf
```

5.200.1.14 num_vs_inf

```
uint8_t uvc_device::num_vs_inf
```

5.200.1.15 nXUs

```
uint8_t uvc_device::nXUs
```

5.200.1.16 opened

```
bool uvc_device::opened
```

5.200.1.17 OT

```
struct uvc_ot* uvc_device::OT
```

5.200.1.18 PU

```
struct uvc_pu* uvc_device::PU
```

5.200.1.19 status

```
uint8_t* uvc_device::status
```

Kneron Confidential

5.200.1.20 stream

```
struct uvc_streaming* uvc_device::stream
```

5.200.1.21 SU

```
struct uvc_su* uvc_device::SU
```

5.200.1.22 usb_dev

```
struct usb_device* uvc_device::usb_dev
```

5.200.1.23 uvc_version

```
uint16_t uvc_device::uvc_version
```

5.200.1.24 vc_inf

```
uint8_t uvc_device::vc_inf
```

5.200.1.25 XU

```
struct uvc_xu* uvc_device::XU
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.201 uvc_encoding_unit_descriptor Struct Reference

```
#include <uvc.h>
```

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bUnitID
- uint8_t bSourceID
- uint8_t iEncoding
- uint8_t bControlSize
- uint8_t bmControls [0]
- uint8_t bmControlsRuntime [0]

5.201.1 Field Documentation

5.201.1.1 bControlSize

```
uint8_t uvc_encoding_unit_descriptor::bControlSize
```

5.201.1.2 bDescriptorSubType

```
uint8_t uvc_encoding_unit_descriptor::bDescriptorSubType
```

5.201.1.3 bDescriptorType

```
uint8_t uvc_encoding_unit_descriptor::bDescriptorType
```

5.201.1.4 bLength

```
uint8_t uvc_encoding_unit_descriptor::bLength
```

5.201.1.5 bmControls

```
uint8_t uvc_encoding_unit_descriptor::bmControls[0]
```

Kneron Confidential

5.201.1.6 bmControlsRuntime

```
uint8_t uvc_encoding_unit_descriptor::bmControlsRuntime[0]
```

5.201.1.7 bSourceID

```
uint8_t uvc_encoding_unit_descriptor::bSourceID
```

5.201.1.8 bUnitID

```
uint8_t uvc_encoding_unit_descriptor::bUnitID
```

5.201.1.9 iEncoding

```
uint8_t uvc_encoding_unit_descriptor::iEncoding
```

The documentation for this struct was generated from the following file:

- uvc.h

5.202 uvc_ET_Head_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType

5.202.1 Field Documentation

5.202.1.1 bDescriptorSubType

```
uint8_t uvc_ET_Head_descriptor::bDescriptorSubType
```

Kneron Confidential

5.202.1.2 bDescriptorType

```
uint8_t uvc_ET_Head_descriptor::bDescriptorType
```

5.202.1.3 bLength

```
uint8_t uvc_ET_Head_descriptor::bLength
```

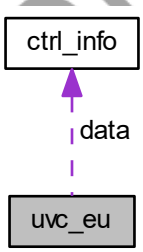
The documentation for this struct was generated from the following file:

- uvc.h

5.203 uvc_eu Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_eu:



Data Fields

- uint8_t id
- uint8_t s_id
- uint8_t bSourceID
- uint8_t bNumControls
- uint32_t bmControls
- uint32_t bmControlsRuntime
- struct ctrl_info * data

5.203.1 Field Documentation

Kneron Confidential

5.203.1.1 bmControls

```
uint32_t uvc_eu::bmControls
```

5.203.1.2 bmControlsRuntime

```
uint32_t uvc_eu::bmControlsRuntime
```

5.203.1.3 bNumControls

```
uint8_t uvc_eu::bNumControls
```

5.203.1.4 bSourceID

```
uint8_t uvc_eu::bSourceID
```

5.203.1.5 data

```
struct ctrl_info* uvc_eu::data
```

5.203.1.6 id

```
uint8_t uvc_eu::id
```

5.203.1.7 s_id

```
uint8_t uvc_eu::s_id
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

Kneron Confidential

5.204 uvc_extension_unit_descriptor Struct Reference

`#include <uvc.h>`

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bUnitID
- uint8_t guidExtensionCode [16]
- uint8_t bNumControls
- uint8_t bNrInPins
- uint8_t baSourceID [0]
- uint8_t bControlSize
- uint8_t bmControls [0]
- uint8_t iExtension

5.204.1 Field Documentation

5.204.1.1 baSourceID

`uint8_t uvc_extension_unit_descriptor::baSourceID[0]`

5.204.1.2 bControlSize

`uint8_t uvc_extension_unit_descriptor::bControlSize`

5.204.1.3 bDescriptorSubType

`uint8_t uvc_extension_unit_descriptor::bDescriptorSubType`

5.204.1.4 bDescriptorType

`uint8_t uvc_extension_unit_descriptor::bDescriptorType`

Kneron Confidential

5.204.1.5 bLength

```
uint8_t uvc_extension_unit_descriptor::bLength
```

5.204.1.6 bmControls

```
uint8_t uvc_extension_unit_descriptor::bmControls[0]
```

5.204.1.7 bNrInPins

```
uint8_t uvc_extension_unit_descriptor::bNrInPins
```

5.204.1.8 bNumControls

```
uint8_t uvc_extension_unit_descriptor::bNumControls
```

5.204.1.9 bUnitID

```
uint8_t uvc_extension_unit_descriptor::bUnitID
```

5.204.1.10 guidExtensionCode

```
uint8_t uvc_extension_unit_descriptor::guidExtensionCode[16]
```

5.204.1.11 iExtension

```
uint8_t uvc_extension_unit_descriptor::iExtension
```

The documentation for this struct was generated from the following file:

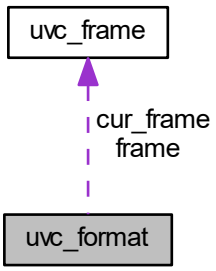
- uvc.h

Kneron Confidential

5.205 uvc_format Struct Reference

`#include <uvc_dev.h>`

Collaboration diagram for uvc_format:



Data Fields

- uint8_t type
- uint8_t index
- uint8_t bpp
- uint8_t colorspace
- uint32_t fcc
- char name [32]
- unsigned int nframes
- uint8_t cur_frame_num
- struct uvc_frame * frame
- struct uvc_frame * cur_frame

5.205.1 Field Documentation

5.205.1.1 bpp

`uint8_t uvc_format::bpp`

5.205.1.2 colorspace

`uint8_t uvc_format::colorspace`

Kneron Confidential

5.205.1.3 cur_frame

```
struct uvc_frame* uvc_format::cur_frame
```

5.205.1.4 cur_frame_num

```
uint8_t uvc_format::cur_frame_num
```

5.205.1.5 fcc

```
uint32_t uvc_format::fcc
```

5.205.1.6 frame

```
struct uvc_frame* uvc_format::frame
```

5.205.1.7 index

```
uint8_t uvc_format::index
```

5.205.1.8 name

```
char uvc_format::name[32]
```

5.205.1.9 nframes

```
unsigned int uvc_format::nframes
```

Kneron Confidential

5.205.1.10 type

```
uint8_t uvc_format::type
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.206 uvc_format_desc Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- char * name
- uint32_t fcc

5.206.1 Field Documentation

5.206.1.1 fcc

```
uint32_t uvc_format_desc::fcc
```

5.206.1.2 name

```
char* uvc_format_desc::name
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.207 uvc_format_desc_head Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubtype
- uint8_t bFormatIndex

Kneron Confidential

5.207.1 Field Documentation

5.207.1.1 bDescriptorSubtype

```
uint8_t uvc_format_desc_head::bDescriptorSubtype
```

5.207.1.2 bDescriptorType

```
uint8_t uvc_format_desc_head::bDescriptorType
```

5.207.1.3 bFormatIndex

```
uint8_t uvc_format_desc_head::bFormatIndex
```

5.207.1.4 bLength

```
uint8_t uvc_format_desc_head::bLength
```

The documentation for this struct was generated from the following file:

- uvc.h

5.208 uvc_format_frame_based Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bFrameIndex
- uint8_t bNumFrameDescriptors
- uint8_t guidFormat [16]
- uint8_t bBitsPerPixel
- uint8_t bDefaultFrameIndex
- uint8_t bAspectRatioX
- uint8_t bAspectRatioY
- uint8_t bmInterfaceFlags
- uint8_t bCopyProtect
- uint8_t bVariableSize

Kneron Confidential

5.208.1 Field Documentation

5.208.1.1 bAspectRatioX

```
uint8_t uvc_format_frame_based::bAspectRatioX
```

5.208.1.2 bAspectRatioY

```
uint8_t uvc_format_frame_based::bAspectRatioY
```

5.208.1.3 bBitsPerPixel

```
uint8_t uvc_format_frame_based::bBitsPerPixel
```

5.208.1.4 bCopyProtect

```
uint8_t uvc_format_frame_based::bCopyProtect
```

5.208.1.5 bDefaultFrameIndex

```
uint8_t uvc_format_frame_based::bDefaultFrameIndex
```

5.208.1.6 bDescriptorSubType

```
uint8_t uvc_format_frame_based::bDescriptorSubType
```

5.208.1.7 bDescriptorType

```
uint8_t uvc_format_frame_based::bDescriptorType
```

Kneron Confidential

5.208.1.8 bFrameIndex

```
uint8_t uvc_format_frame_based::bFrameIndex
```

5.208.1.9 bLength

```
uint8_t uvc_format_frame_based::bLength
```

5.208.1.10 bmInterfaceFlags

```
uint8_t uvc_format_frame_based::bmInterfaceFlags
```

5.208.1.11 bNumFrameDescriptors

```
uint8_t uvc_format_frame_based::bNumFrameDescriptors
```

5.208.1.12 bVariableSize

```
uint8_t uvc_format_frame_based::bVariableSize
```

5.208.1.13 guidFormat

```
uint8_t uvc_format_frame_based::guidFormat[16]
```

The documentation for this struct was generated from the following file:

- uvc.h

5.209 uvc_format_mjpeg Struct Reference

```
#include <uvc.h>
```

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bFormatIndex
- uint8_t bNumFrameDescriptors
- uint8_t bmFlags
- uint8_t bDefaultFrameIndex
- uint8_t bAspectRatioX
- uint8_t bAspectRatioY
- uint8_t bmInterfaceFlags
- uint8_t bCopyProtect

5.209.1 Field Documentation

5.209.1.1 bAspectRatioX

```
uint8_t uvc_format_mjpeg::bAspectRatioX
```

5.209.1.2 bAspectRatioY

```
uint8_t uvc_format_mjpeg::bAspectRatioY
```

5.209.1.3 bCopyProtect

```
uint8_t uvc_format_mjpeg::bCopyProtect
```

5.209.1.4 bDefaultFrameIndex

```
uint8_t uvc_format_mjpeg::bDefaultFrameIndex
```

5.209.1.5 bDescriptorSubType

```
uint8_t uvc_format_mjpeg::bDescriptorSubType
```

Kneron Confidential

5.209.1.6 bDescriptorType

```
uint8_t uvc_format_mjpeg::bDescriptorType
```

5.209.1.7 bFormatIndex

```
uint8_t uvc_format_mjpeg::bFormatIndex
```

5.209.1.8 bLength

```
uint8_t uvc_format_mjpeg::bLength
```

5.209.1.9 bmFlags

```
uint8_t uvc_format_mjpeg::bmFlags
```

5.209.1.10 bmInterfaceFlags

```
uint8_t uvc_format_mjpeg::bmInterfaceFlags
```

5.209.1.11 bNumFrameDescriptors

```
uint8_t uvc_format_mjpeg::bNumFrameDescriptors
```

The documentation for this struct was generated from the following file:

- uvc.h

5.210 uvc_format_uncompressed Struct Reference

```
#include <uvc.h>
```

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bFormatIndex
- uint8_t bNumFrameDescriptors
- uint8_t guidFormat [16]
- uint8_t bBitsPerPixel
- uint8_t bDefaultFrameIndex
- uint8_t bAspectRatioX
- uint8_t bAspectRatioY
- uint8_t bmInterfaceFlags
- uint8_t bCopyProtect

5.210.1 Field Documentation

5.210.1.1 bAspectRatioX

```
uint8_t uvc_format_uncompressed::bAspectRatioX
```

5.210.1.2 bAspectRatioY

```
uint8_t uvc_format_uncompressed::bAspectRatioY
```

5.210.1.3 bBitsPerPixel

```
uint8_t uvc_format_uncompressed::bBitsPerPixel
```

5.210.1.4 bCopyProtect

```
uint8_t uvc_format_uncompressed::bCopyProtect
```

5.210.1.5 bDefaultFrameIndex

```
uint8_t uvc_format_uncompressed::bDefaultFrameIndex
```

Kneron Confidential

5.210.1.6 bDescriptorSubType

```
uint8_t uvc_format_uncompressed::bDescriptorSubType
```

5.210.1.7 bDescriptorType

```
uint8_t uvc_format_uncompressed::bDescriptorType
```

5.210.1.8 bFormatIndex

```
uint8_t uvc_format_uncompressed::bFormatIndex
```

5.210.1.9 bLength

```
uint8_t uvc_format_uncompressed::bLength
```

5.210.1.10 bmInterfaceFlags

```
uint8_t uvc_format_uncompressed::bmInterfaceFlags
```

5.210.1.11 bNumFrameDescriptors

```
uint8_t uvc_format_uncompressed::bNumFrameDescriptors
```

5.210.1.12 guidFormat

```
uint8_t uvc_format_uncompressed::guidFormat[16]
```

The documentation for this struct was generated from the following file:

- uvc.h

Kneron Confidential

5.211 uvc_frame Struct Reference

`#include <uvc_dev.h>`

Data Fields

- uint8_t bDescriptorType
- uint8_t bFrameIndex
- uint8_t bmCapabilities
- uint16_t wWidth
- uint16_t wHeight
- uint32_t dwMinBitRate
- uint32_t dwMaxBitRate
- uint32_t dwMaxVideoFrameBufferSize
- uint32_t dwDefaultFrameInterval
- uint8_t bFrameIntervalType
- uint32_t * dwFrameInterval

5.211.1 Field Documentation

5.211.1.1 bDescriptorType

`uint8_t uvc_frame::bDescriptorType`

5.211.1.2 bFrameIndex

`uint8_t uvc_frame::bFrameIndex`

5.211.1.3 bFrameIntervalType

`uint8_t uvc_frame::bFrameIntervalType`

5.211.1.4 bmCapabilities

`uint8_t uvc_frame::bmCapabilities`

Kneron Confidential

5.211.1.5 dwDefaultFrameInterval

```
uint32_t uvc_frame::dwDefaultFrameInterval
```

5.211.1.6 dwFrameInterval

```
uint32_t* uvc_frame::dwFrameInterval
```

5.211.1.7 dwMaxBitRate

```
uint32_t uvc_frame::dwMaxBitRate
```

5.211.1.8 dwMaxVideoFrameBufferSize

```
uint32_t uvc_frame::dwMaxVideoFrameBufferSize
```

5.211.1.9 dwMinBitRate

```
uint32_t uvc_frame::dwMinBitRate
```

5.211.1.10 wHeight

```
uint16_t uvc_frame::wHeight
```

5.211.1.11 wWidth

```
uint16_t uvc_frame::wWidth
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

Kneron Confidential

5.212 uvc_frame_desc_head Struct Reference

`#include <uvc.h>`

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubtype
- uint8_t bFrameIndex

5.212.1 Field Documentation

5.212.1.1 bDescriptorSubtype

`uint8_t uvc_frame_desc_head::bDescriptorSubtype`

5.212.1.2 bDescriptorType

`uint8_t uvc_frame_desc_head::bDescriptorType`

5.212.1.3 bFrameIndex

`uint8_t uvc_frame_desc_head::bFrameIndex`

5.212.1.4 bLength

`uint8_t uvc_frame_desc_head::bLength`

The documentation for this struct was generated from the following file:

- uvc.h

5.213 uvc_frame_mjpeg Struct Reference

`#include <uvc.h>`

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bFrameIndex
- uint8_t bmCapabilities
- uint16_t wWidth
- uint16_t wHeight
- uint32_t dwMinBitRate
- uint32_t dwMaxBitRate
- uint32_t dwMaxVideoFrameBufferSize
- uint32_t dwDefaultFrameInterval
- uint8_t bFrameIntervalType
- uint32_t dwFrameInterval [ ]

5.213.1 Field Documentation

5.213.1.1 bDescriptorSubType

```
uint8_t uvc_frame_mjpeg::bDescriptorSubType
```

5.213.1.2 bDescriptorType

```
uint8_t uvc_frame_mjpeg::bDescriptorType
```

5.213.1.3 bFrameIndex

```
uint8_t uvc_frame_mjpeg::bFrameIndex
```

5.213.1.4 bFrameIntervalType

```
uint8_t uvc_frame_mjpeg::bFrameIntervalType
```

5.213.1.5 bLength

```
uint8_t uvc_frame_mjpeg::bLength
```

Kneron Confidential

5.213.1.6 bmCapabilities

```
uint8_t uvc_frame_mjpeg::bmCapabilities
```

5.213.1.7 dwDefaultFrameInterval

```
uint32_t uvc_frame_mjpeg::dwDefaultFrameInterval
```

5.213.1.8 dwFrameInterval

```
uint32_t uvc_frame_mjpeg::dwFrameInterval[]
```

5.213.1.9 dwMaxBitRate

```
uint32_t uvc_frame_mjpeg::dwMaxBitRate
```

5.213.1.10 dwMaxVideoFrameBufferSize

```
uint32_t uvc_frame_mjpeg::dwMaxVideoFrameBufferSize
```

5.213.1.11 dwMinBitRate

```
uint32_t uvc_frame_mjpeg::dwMinBitRate
```

5.213.1.12 wHeight

```
uint16_t uvc_frame_mjpeg::wHeight
```

Kneron Confidential

5.213.1.13 wWidth

```
uint16_t uvc_frame_mjpeg::wWidth
```

The documentation for this struct was generated from the following file:

- uvc.h

5.214 uvc_frame_uncompressed Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bFrameIndex
- uint8_t bmCapabilities
- uint16_t wWidth
- uint16_t wHeight
- uint32_t dwMinBitRate
- uint32_t dwMaxBitRate
- uint32_t dwMaxVideoFrameBufferSize
- uint32_t dwDefaultFrameInterval
- uint8_t bFrameIntervalType
- uint32_t dwFrameInterval []

5.214.1 Field Documentation

5.214.1.1 bDescriptorSubType

```
uint8_t uvc_frame_uncompressed::bDescriptorSubType
```

5.214.1.2 bDescriptorType

```
uint8_t uvc_frame_uncompressed::bDescriptorType
```

Kneron Confidential

5.214.1.3 bFrameIndex

```
uint8_t uvc_frame_uncompressed::bFrameIndex
```

5.214.1.4 bFrameIntervalType

```
uint8_t uvc_frame_uncompressed::bFrameIntervalType
```

5.214.1.5 bLength

```
uint8_t uvc_frame_uncompressed::bLength
```

5.214.1.6 bmCapabilities

```
uint8_t uvc_frame_uncompressed::bmCapabilities
```

5.214.1.7 dwDefaultFrameInterval

```
uint32_t uvc_frame_uncompressed::dwDefaultFrameInterval
```

5.214.1.8 dwFrameInterval

```
uint32_t uvc_frame_uncompressed::dwFrameInterval[]
```

5.214.1.9 dwMaxBitRate

```
uint32_t uvc_frame_uncompressed::dwMaxBitRate
```

5.214.1.10 dwMaxVideoFrameBufferSize

```
uint32_t uvc_frame_uncompressed::dwMaxVideoFrameBufferSize
```

Kneron Confidential

5.214.1.11 dwMinBitRate

```
uint32_t uvc_frame_uncompressed::dwMinBitRate
```

5.214.1.12 wHeight

```
uint16_t uvc_frame_uncompressed::wHeight
```

5.214.1.13 wWidth

```
uint16_t uvc_frame_uncompressed::wWidth
```

The documentation for this struct was generated from the following file:

- uvc.h

5.215 uvc_inf_assoc_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bFirstInterface
- uint8_t bInterfaceCount
- uint8_t bFunctionClass
- uint8_t bFunctionSubClass
- uint8_t bFunctionProtocol
- uint8_t iFunction

5.215.1 Field Documentation

5.215.1.1 bDescriptorType

```
uint8_t uvc_inf_assoc_descriptor::bDescriptorType
```

Kneron Confidential

5.215.1.2 bFirstInterface

```
uint8_t uvc_inf_assoc_descriptor::bFirstInterface
```

5.215.1.3 bFunctionClass

```
uint8_t uvc_inf_assoc_descriptor::bFunctionClass
```

5.215.1.4 bFunctionProtocol

```
uint8_t uvc_inf_assoc_descriptor::bFunctionProtocol
```

5.215.1.5 bFunctionSubClass

```
uint8_t uvc_inf_assoc_descriptor::bFunctionSubClass
```

5.215.1.6 bInterfaceCount

```
uint8_t uvc_inf_assoc_descriptor::bInterfaceCount
```

5.215.1.7 bLength

```
uint8_t uvc_inf_assoc_descriptor::bLength
```

5.215.1.8 iFunction

```
uint8_t uvc_inf_assoc_descriptor::iFunction
```

The documentation for this struct was generated from the following file:

- uvc.h

Kneron Confidential

5.216 uvc_input_header_descriptor Struct Reference

`#include <uvc.h>`

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bNumFormats
- uint16_t wTotalLength
- uint8_t bEndpointAddress
- uint8_t bmInfo
- uint8_t bTerminalLink
- uint8_t bStillCaptureMethod
- uint8_t bTriggerSupport
- uint8_t bTriggerUsage
- uint8_t bControlSize
- uint8_t bmaControls []

5.216.1 Field Documentation

5.216.1.1 bControlSize

`uint8_t uvc_input_header_descriptor::bControlSize`

5.216.1.2 bDescriptorSubType

`uint8_t uvc_input_header_descriptor::bDescriptorSubType`

5.216.1.3 bDescriptorType

`uint8_t uvc_input_header_descriptor::bDescriptorType`

5.216.1.4 bEndpointAddress

`uint8_t uvc_input_header_descriptor::bEndpointAddress`

Kneron Confidential

5.216.1.5 bLength

```
uint8_t uvc_input_header_descriptor::bLength
```

5.216.1.6 bmaControls

```
uint8_t uvc_input_header_descriptor::bmaControls[]
```

5.216.1.7 bmInfo

```
uint8_t uvc_input_header_descriptor::bmInfo
```

5.216.1.8 bNumFormats

```
uint8_t uvc_input_header_descriptor::bNumFormats
```

5.216.1.9 bStillCaptureMethod

```
uint8_t uvc_input_header_descriptor::bStillCaptureMethod
```

5.216.1.10 bTerminalLink

```
uint8_t uvc_input_header_descriptor::bTerminalLink
```

5.216.1.11 bTriggerSupport

```
uint8_t uvc_input_header_descriptor::bTriggerSupport
```

5.216.1.12 bTriggerUsage

```
uint8_t uvc_input_header_descriptor::bTriggerUsage
```

Kneron Confidential

5.216.1.13 wTotalLength

```
uint16_t uvc_input_header_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

- uvc.h

5.217 uvc_input_terminal_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bTerminalID
- uint16_t wTerminalType
- uint8_t bAssocTerminal
- uint8_t iTerminal

5.217.1 Field Documentation

5.217.1.1 bAssocTerminal

```
uint8_t uvc_input_terminal_descriptor::bAssocTerminal
```

5.217.1.2 bDescriptorSubType

```
uint8_t uvc_input_terminal_descriptor::bDescriptorSubType
```

5.217.1.3 bDescriptorType

```
uint8_t uvc_input_terminal_descriptor::bDescriptorType
```

Kneron Confidential

5.217.1.4 bLength

```
uint8_t uvc_input_terminal_descriptor::bLength
```

5.217.1.5 bTerminalID

```
uint8_t uvc_input_terminal_descriptor::bTerminalID
```

5.217.1.6 iTerminal

```
uint8_t uvc_input_terminal_descriptor::iTerminal
```

5.217.1.7 wTerminalType

```
uint16_t uvc_input_terminal_descriptor::wTerminalType
```

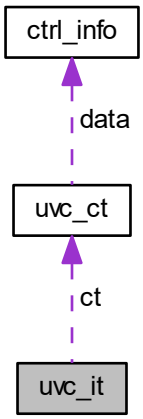
The documentation for this struct was generated from the following file:

- uvc.h

5.218 uvc_it Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_it:



Kneron Confidential

Data Fields

- uint8_t id
- uint16_t wTerminalType
- uint8_t bAssocTerminal
- struct uvc_ct * ct

5.218.1 Field Documentation

5.218.1.1 bAssocTerminal

```
uint8_t uvc_it::bAssocTerminal
```

5.218.1.2 ct

```
struct uvc_ct* uvc_it::ct
```

5.218.1.3 id

```
uint8_t uvc_it::id
```

5.218.1.4 wTerminalType

```
uint16_t uvc_it::wTerminalType
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.219 uvc_ot Struct Reference

```
#include <uvc_dev.h>
```

Kneron Confidential

Data Fields

- uint8_t id
- uint16_t wTerminalType
- uint8_t bAssocTerminal
- uint8_t baSourceID

5.219.1 Field Documentation

5.219.1.1 baSourceID

```
uint8_t uvc_ot::baSourceID
```

5.219.1.2 bAssocTerminal

```
uint8_t uvc_ot::bAssocTerminal
```

5.219.1.3 id

```
uint8_t uvc_ot::id
```

5.219.1.4 wTerminalType

```
uint16_t uvc_ot::wTerminalType
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.220 uvc_output_header_descriptor Struct Reference

```
#include <uvc.h>
```

Kneron Confidential

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bNumFormats
- uint16_t wTotalLength
- uint8_t bEndpointAddress
- uint8_t bTerminalLink
- uint8_t bControlSize
- uint8_t bmaControls [ ]

5.220.1 Field Documentation

5.220.1.1 bControlSize

```
uint8_t uvc_output_header_descriptor::bControlSize
```

5.220.1.2 bDescriptorSubType

```
uint8_t uvc_output_header_descriptor::bDescriptorSubType
```

5.220.1.3 bDescriptorType

```
uint8_t uvc_output_header_descriptor::bDescriptorType
```

5.220.1.4 bEndpointAddress

```
uint8_t uvc_output_header_descriptor::bEndpointAddress
```

5.220.1.5 bLength

```
uint8_t uvc_output_header_descriptor::bLength
```

Kneron Confidential

5.220.1.6 bmaControls

```
uint8_t uvc_output_header_descriptor::bmaControls[]
```

5.220.1.7 bNumFormats

```
uint8_t uvc_output_header_descriptor::bNumFormats
```

5.220.1.8 bTerminalLink

```
uint8_t uvc_output_header_descriptor::bTerminalLink
```

5.220.1.9 wTotalLength

```
uint16_t uvc_output_header_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

- uvc.h

5.221 uvc_output_terminal_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bTerminalID
- uint16_t wTerminalType
- uint8_t bAssocTerminal
- uint8_t bSourceID
- uint8_t iTerminal

5.221.1 Field Documentation

Kneron Confidential

5.221.1.1 bAssocTerminal

```
uint8_t uvc_output_terminal_descriptor::bAssocTerminal
```

5.221.1.2 bDescriptorSubType

```
uint8_t uvc_output_terminal_descriptor::bDescriptorSubType
```

5.221.1.3 bDescriptorType

```
uint8_t uvc_output_terminal_descriptor::bDescriptorType
```

5.221.1.4 bLength

```
uint8_t uvc_output_terminal_descriptor::bLength
```

5.221.1.5 bSourceID

```
uint8_t uvc_output_terminal_descriptor::bSourceID
```

5.221.1.6 bTerminalID

```
uint8_t uvc_output_terminal_descriptor::bTerminalID
```

5.221.1.7 iTerminal

```
uint8_t uvc_output_terminal_descriptor::iTerminal
```

Kneron Confidential

5.221.1.8 wTerminalType

```
uint16_t uvc_output_terminal_descriptor::wTerminalType
```

The documentation for this struct was generated from the following file:

- uvc.h

5.222 uvc_processing_unit_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bUnitID
- uint8_t bSourceID
- uint16_t wMaxMultiplier
- uint8_t bControlSize
- uint8_t bmControls [0]
- uint8_t iProcessing
- uint8_t bmVideoStandards

5.222.1 Field Documentation

5.222.1.1 bControlSize

```
uint8_t uvc_processing_unit_descriptor::bControlSize
```

5.222.1.2 bDescriptorSubType

```
uint8_t uvc_processing_unit_descriptor::bDescriptorSubType
```

5.222.1.3 bDescriptorType

```
uint8_t uvc_processing_unit_descriptor::bDescriptorType
```

Kneron Confidential

5.222.1.4 bLength

```
uint8_t uvc_processing_unit_descriptor::bLength
```

5.222.1.5 bmControls

```
uint8_t uvc_processing_unit_descriptor::bmControls[0]
```

5.222.1.6 bmVideoStandards

```
uint8_t uvc_processing_unit_descriptor::bmVideoStandards
```

5.222.1.7 bSourceID

```
uint8_t uvc_processing_unit_descriptor::bSourceID
```

5.222.1.8 bUnitID

```
uint8_t uvc_processing_unit_descriptor::bUnitID
```

5.222.1.9 iProcessing

```
uint8_t uvc_processing_unit_descriptor::iProcessing
```

5.222.1.10 wMaxMultiplier

```
uint16_t uvc_processing_unit_descriptor::wMaxMultiplier
```

The documentation for this struct was generated from the following file:

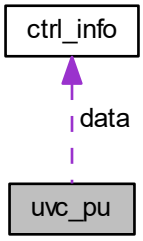
- uvc.h

Kneron Confidential

5.223 uvc_pu Struct Reference

`#include <uvc_dev.h>`

Collaboration diagram for uvc_pu:



Data Fields

- uint8_t id
- uint8_t baSourceID
- uint16_t wMaxMultiplier
- uint32_t bmControls
- uint8_t bmVideoStandards
- struct ctrl_info * data

5.223.1 Field Documentation

5.223.1.1 baSourceID

`uint8_t uvc_pu::baSourceID`

5.223.1.2 bmControls

`uint32_t uvc_pu::bmControls`

Kneron Confidential

5.223.1.3 bmVideoStandards

```
uint8_t uvc_pu::bmVideoStandards
```

5.223.1.4 data

```
struct ctrl_info* uvc_pu::data
```

5.223.1.5 id

```
uint8_t uvc_pu::id
```

5.223.1.6 wMaxMultiplier

```
uint16_t uvc_pu::wMaxMultiplier
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.224 uvc_selector_unit_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bUnitID
- uint8_t bNrInPins
- uint8_t baSourceID [0]
- uint8_t iSelector

5.224.1 Field Documentation

Kneron Confidential

5.224.1.1 baSourceID

```
uint8_t uvc_selector_unit_descriptor::baSourceID[0]
```

5.224.1.2 bDescriptorSubType

```
uint8_t uvc_selector_unit_descriptor::bDescriptorSubType
```

5.224.1.3 bDescriptorType

```
uint8_t uvc_selector_unit_descriptor::bDescriptorType
```

5.224.1.4 bLength

```
uint8_t uvc_selector_unit_descriptor::bLength
```

5.224.1.5 bNrInPins

```
uint8_t uvc_selector_unit_descriptor::bNrInPins
```

5.224.1.6 bUnitID

```
uint8_t uvc_selector_unit_descriptor::bUnitID
```

5.224.1.7 iSelector

```
uint8_t uvc_selector_unit_descriptor::iSelector
```

The documentation for this struct was generated from the following file:

- uvc.h

Kneron Confidential

5.225 uvc_still_image_frame_descriptor Struct Reference

`#include <uvc.h>`

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint8_t bEndpointAddress
- uint8_t bNumImageSizePatterns
- uint16_t wWidth [0]
- uint16_t wHeight [0]
- uint8_t bNumCompressionPattern
- uint8_t bCompression [ ]

5.225.1 Field Documentation

5.225.1.1 bCompression

`uint8_t uvc_still_image_frame_descriptor::bCompression[ ]`

5.225.1.2 bDescriptorSubType

`uint8_t uvc_still_image_frame_descriptor::bDescriptorSubType`

5.225.1.3 bDescriptorType

`uint8_t uvc_still_image_frame_descriptor::bDescriptorType`

5.225.1.4 bEndpointAddress

`uint8_t uvc_still_image_frame_descriptor::bEndpointAddress`

Kneron Confidential

5.225.1.5 bLength

```
uint8_t uvc_still_image_frame_descriptor::bLength
```

5.225.1.6 bNumCompressionPattern

```
uint8_t uvc_still_image_frame_descriptor::bNumCompressionPattern
```

5.225.1.7 bNumImageSizePatterns

```
uint8_t uvc_still_image_frame_descriptor::bNumImageSizePatterns
```

5.225.1.8 wHeight

```
uint16_t uvc_still_image_frame_descriptor::wHeight[0]
```

5.225.1.9 wWidth

```
uint16_t uvc_still_image_frame_descriptor::wWidth[0]
```

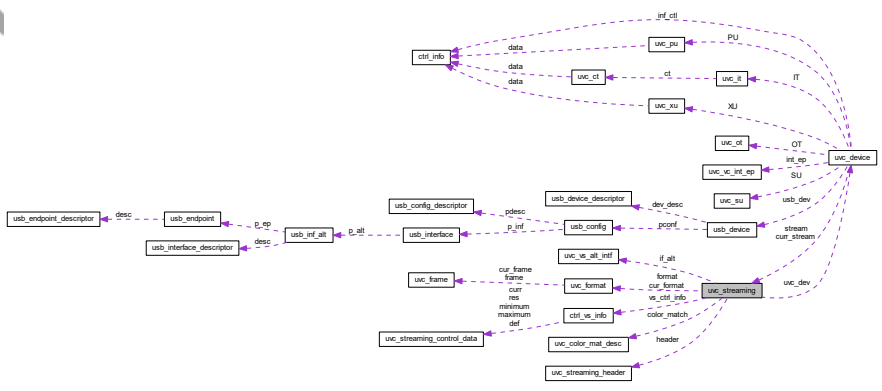
The documentation for this struct was generated from the following file:

- uvc.h

5.226 uvc_streaming Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_streaming:



Kneron Confidential

Data Fields

- struct uvc_device * uvc_dev
- int ifnum
- uint8_t num_alt
- uint8_t curr_altnum
- struct uvc_vs_alt_intf * if_alt
- uint8_t num_ep
- uint16_t ep_addr
- struct uvc_streaming_header header
- uint8_t ep_type
- uint8_t TerminalId
- uint8_t ControlSize
- unsigned int nformats
- struct uvc_format * format
- uint8_t cur_format_num
- struct uvc_format * cur_format
- struct ctrl_vs_info * vs_ctrl_info
- struct uvc_color_mat_desc * color_match
- USBH_PIPE_HANDLE isoch_pipe
- uint32_t imagesize
- uint8_t * all_frame_buf
- int write_idx
- bool running

5.226.1 Field Documentation

5.226.1.1 all_frame_buf

```
uint8_t* uvc_streaming::all_frame_buf
```

5.226.1.2 color_match

```
struct uvc_color_mat_desc* uvc_streaming::color_match
```

5.226.1.3 ControlSize

```
uint8_t uvc_streaming::ControlSize
```

Kneron Confidential

5.226.1.4 cur_format

```
struct uvc_format* uvc_streaming::cur_format
```

5.226.1.5 cur_format_num

```
uint8_t uvc_streaming::cur_format_num
```

5.226.1.6 curr_altnum

```
uint8_t uvc_streaming::curr_altnum
```

5.226.1.7 ep_addr

```
uint16_t uvc_streaming::ep_addr
```

5.226.1.8 ep_type

```
uint8_t uvc_streaming::ep_type
```

5.226.1.9 format

```
struct uvc_format* uvc_streaming::format
```

5.226.1.10 header

```
struct uvc_streaming_header uvc_streaming::header
```

5.226.1.11 if_alt

```
struct uvc_vs_alt_intf* uvc_streaming::if_alt
```

Kneron Confidential

5.226.1.12 ifnum

```
int uvc_streaming::ifnum
```

5.226.1.13 imagesize

```
uint32_t uvc_streaming::imagesize
```

5.226.1.14 isoch_pipe

```
USBH_PIPE_HANDLE uvc_streaming::isoch_pipe
```

5.226.1.15 nformats

```
unsigned int uvc_streaming::nformats
```

5.226.1.16 num_alt

```
uint8_t uvc_streaming::num_alt
```

5.226.1.17 num_ep

```
uint8_t uvc_streaming::num_ep
```

5.226.1.18 running

```
bool uvc_streaming::running
```

5.226.1.19 TerminalId

```
uint8_t uvc_streaming::TerminalId
```

Kneron Confidential

5.226.1.20 uvc_dev

```
struct uvc_device* uvc_streaming::uvc_dev
```

5.226.1.21 vs_ctrl_info

```
struct ctrl_vs_info* uvc_streaming::vs_ctrl_info
```

5.226.1.22 write_idx

```
int uvc_streaming::write_idx
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.227 uvc_streaming_control_data Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint16_t wmHint
- uint8_t bFormatIndex
- uint8_t bFrameIndex
- uint32_t dwFrameInterval
- uint16_t wKeyFrameRate
- uint16_t wPFrameRate
- uint16_t wCompQuality
- uint16_t wCompWindowSize
- uint16_t wDelay
- uint32_t dwMaxVideoFrameSize
- uint32_t dwMaxPayloadTransferSize
- uint32_t dwClockFrequency
- uint8_t bmFramingInfo
- uint8_t bPreferedVersion
- uint8_t bMinVersion
- uint8_t bMaxVersion
- uint8_t bUsage
- uint8_t bBitDrpthLuma
- uint8_t bmSetting
- uint8_t bMaxNumberOfRefFeamesPlus1
- uint16_t wRateControlModes
- uint8_t bmLayoutPerStream [8]

Kneron Confidential

5.227.1 Field Documentation

5.227.1.1 bBitDrpthLuma

```
uint8_t uvc_streaming_control_data::bBitDrpthLuma
```

5.227.1.2 bFormatIndex

```
uint8_t uvc_streaming_control_data::bFormatIndex
```

5.227.1.3 bFrameIndex

```
uint8_t uvc_streaming_control_data::bFrameIndex
```

5.227.1.4 bMaxNumberOfRefFeamesPlus1

```
uint8_t uvc_streaming_control_data::bMaxNumberOfRefFeamesPlus1
```

5.227.1.5 bMaxVersion

```
uint8_t uvc_streaming_control_data::bMaxVersion
```

5.227.1.6 bmFramingInfo

```
uint8_t uvc_streaming_control_data::bmFramingInfo
```

5.227.1.7 bMinVersion

```
uint8_t uvc_streaming_control_data::bMinVersion
```

Kneron Confidential

5.227.1.8 bmLayoutPerStream

```
uint8_t uvc_streaming_control_data::bmLayoutPerStream[8]
```

5.227.1.9 bmSetting

```
uint8_t uvc_streaming_control_data::bmSetting
```

5.227.1.10 bPreferedVersion

```
uint8_t uvc_streaming_control_data::bPreferedVersion
```

5.227.1.11 bUsage

```
uint8_t uvc_streaming_control_data::bUsage
```

5.227.1.12 dwClockFrequency

```
uint32_t uvc_streaming_control_data::dwClockFrequency
```

5.227.1.13 dwFrameInterval

```
uint32_t uvc_streaming_control_data::dwFrameInterval
```

5.227.1.14 dwMaxPayloadTransferSize

```
uint32_t uvc_streaming_control_data::dwMaxPayloadTransferSize
```

5.227.1.15 dwMaxVideoFrameSize

```
uint32_t uvc_streaming_control_data::dwMaxVideoFrameSize
```

Kneron Confidential

5.227.1.16 wCompQuality

```
uint16_t uvc_streaming_control_data::wCompQuality
```

5.227.1.17 wCompWindowSize

```
uint16_t uvc_streaming_control_data::wCompWindowSize
```

5.227.1.18 wDelay

```
uint16_t uvc_streaming_control_data::wDelay
```

5.227.1.19 wKeyFrameRate

```
uint16_t uvc_streaming_control_data::wKeyFrameRate
```

5.227.1.20 wmHint

```
uint16_t uvc_streaming_control_data::wmHint
```

5.227.1.21 wPFrameRate

```
uint16_t uvc_streaming_control_data::wPFrameRate
```

5.227.1.22 wRateControlModes

```
uint16_t uvc_streaming_control_data::wRateControlModes
```

The documentation for this struct was generated from the following file:

- uvc.h

Kneron Confidential

5.228 uvc_streaming_header Struct Reference

`#include <uvc_dev.h>`

Data Fields

- uint8_t bNumFormats
- uint8_t bEndpointAddress
- uint8_t bTerminalLink
- uint8_t bControlSize
- uint8_t * bmaControls
- uint8_t bmInfo
- uint8_t bStillCaptureMethod
- uint8_t bTriggerSupport
- uint8_t bTriggerUsage

5.228.1 Field Documentation

5.228.1.1 bControlSize

`uint8_t uvc_streaming_header::bControlSize`

5.228.1.2 bEndpointAddress

`uint8_t uvc_streaming_header::bEndpointAddress`

5.228.1.3 bmaControls

`uint8_t* uvc_streaming_header::bmaControls`

5.228.1.4 bmInfo

`uint8_t uvc_streaming_header::bmInfo`

Kneron Confidential

5.228.1.5 bNumFormats

```
uint8_t uvc_streaming_header::bNumFormats
```

5.228.1.6 bStillCaptureMethod

```
uint8_t uvc_streaming_header::bStillCaptureMethod
```

5.228.1.7 bTerminalLink

```
uint8_t uvc_streaming_header::bTerminalLink
```

5.228.1.8 bTriggerSupport

```
uint8_t uvc_streaming_header::bTriggerSupport
```

5.228.1.9 bTriggerUsage

```
uint8_t uvc_streaming_header::bTriggerUsage
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.229 uvc_su Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint8_t id
- uint8_t bNrInPins
- uint8_t * baSourceID

5.229.1 Field Documentation

Kneron Confidential

5.229.1.1 baSourceID

```
uint8_t* uvc_su::baSourceID
```

5.229.1.2 bNrInPins

```
uint8_t uvc_su::bNrInPins
```

5.229.1.3 id

```
uint8_t uvc_su::id
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.230 uvc_usb_ctrlreq Struct Reference

```
#include <uvc_ctrl.h>
```

Data Fields

- uint8_t bRequestType
- uint8_t bRequest
- uint16_t wValue
- uint16_t wIndex
- uint16_t wLength
- uint8_t * data

5.230.1 Field Documentation

5.230.1.1 bRequest

```
uint8_t uvc_usb_ctrlreq::bRequest
```

Kneron Confidential

5.230.1.2 bRequestType

```
uint8_t uvc_usb_ctrlreq::bRequestType
```

5.230.1.3 data

```
uint8_t* uvc_usb_ctrlreq::data
```

5.230.1.4 wIndex

```
uint16_t uvc_usb_ctrlreq::wIndex
```

5.230.1.5 wLength

```
uint16_t uvc_usb_ctrlreq::wLength
```

5.230.1.6 wValue

```
uint16_t uvc_usb_ctrlreq::wValue
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.231 uvc_vc_if_header_descriptor Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bDescriptorSubType
- uint16_t bcdUVC
- uint16_t wTotalLength
- uint32_t dwClockFrequency
- uint8_t bInCollection
- uint8_t baInterfaceNr [ ]

Kneron Confidential

5.231.1 Field Documentation

5.231.1.1 baInterfaceNr

```
uint8_t uvc_vc_if_header_descriptor::baInterfaceNr[]
```

5.231.1.2 bcdUVC

```
uint16_t uvc_vc_if_header_descriptor::bcdUVC
```

5.231.1.3 bDescriptorSubType

```
uint8_t uvc_vc_if_header_descriptor::bDescriptorSubType
```

5.231.1.4 bDescriptorType

```
uint8_t uvc_vc_if_header_descriptor::bDescriptorType
```

5.231.1.5 bInCollection

```
uint8_t uvc_vc_if_header_descriptor::bInCollection
```

5.231.1.6 bLength

```
uint8_t uvc_vc_if_header_descriptor::bLength
```

5.231.1.7 dwClockFrequency

```
uint32_t uvc_vc_if_header_descriptor::dwClockFrequency
```

Kneron Confidential

5.231.1.8 wTotalLength

```
uint16_t uvc_vc_if_header_descriptor::wTotalLength
```

The documentation for this struct was generated from the following file:

- uvc.h

5.232 uvc_vc_int_ep Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint16_t addr
- uint16_t maxpacketsize
- uint16_t wMaxTransferSize
- uint8_t interval
- struct urb * int_urb

5.232.1 Field Documentation

5.232.1.1 addr

```
uint16_t uvc_vc_int_ep::addr
```

5.232.1.2 int_urb

```
struct urb* uvc_vc_int_ep::int_urb
```

5.232.1.3 interval

```
uint8_t uvc_vc_int_ep::interval
```

Kneron Confidential

5.232.1.4 maxpacketsize

```
uint16_t uvc_vc_int_ep::maxpacketsize
```

5.232.1.5 wMaxTransferSize

```
uint16_t uvc_vc_int_ep::wMaxTransferSize
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.233 uvc_vs_alt_intf Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint8_t ep_type
- uint16_t addr
- uint16_t maxpacketsize
- uint8_t interval
- uint8_t alt_num

5.233.1 Field Documentation

5.233.1.1 addr

```
uint16_t uvc_vs_alt_intf::addr
```

5.233.1.2 alt_num

```
uint8_t uvc_vs_alt_intf::alt_num
```

Kneron Confidential

5.233.1.3 ep_type

```
uint8_t uvc_vs_alt_intf::ep_type
```

5.233.1.4 interval

```
uint8_t uvc_vs_alt_intf::interval
```

5.233.1.5 maxpacketsize

```
uint16_t uvc_vs_alt_intf::maxpacketsize
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.234 uvc_vs_ctl_data Struct Reference

```
#include <uvc_dev.h>
```

Data Fields

- uint16_t wmHint
- uint8_t bFormatIndex
- uint8_t bFrameIndex
- uint32_t dwFrameInterval
- uint16_t wDelay
- uint32_t dwMaxVideoFrameSize
- uint32_t dwMaxPayloadTransferSize
- uint32_t dwClockFrequency
- uint8_t bPreferedVersion
- uint8_t bMinVersion
- uint8_t bMaxVersion

5.234.1 Field Documentation

5.234.1.1 bFormatIndex

```
uint8_t uvc_vs_ctl_data::bFormatIndex
```

Kneron Confidential

5.234.1.2 bFrameIndex

```
uint8_t uvc_vs_ctl_data::bFrameIndex
```

5.234.1.3 bMaxVersion

```
uint8_t uvc_vs_ctl_data::bMaxVersion
```

5.234.1.4 bMinVersion

```
uint8_t uvc_vs_ctl_data::bMinVersion
```

5.234.1.5 bPreferedVersion

```
uint8_t uvc_vs_ctl_data::bPreferedVersion
```

5.234.1.6 dwClockFrequency

```
uint32_t uvc_vs_ctl_data::dwClockFrequency
```

5.234.1.7 dwFrameInterval

```
uint32_t uvc_vs_ctl_data::dwFrameInterval
```

5.234.1.8 dwMaxPayloadTransferSize

```
uint32_t uvc_vs_ctl_data::dwMaxPayloadTransferSize
```

5.234.1.9 dwMaxVideoFrameSize

```
uint32_t uvc_vs_ctl_data::dwMaxVideoFrameSize
```

Kneron Confidential

5.234.1.10 wDelay

```
uint16_t uvc_vs_ctl_data::wDelay
```

5.234.1.11 wmHint

```
uint16_t uvc_vs_ctl_data::wmHint
```

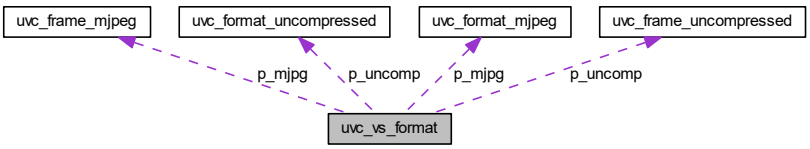
The documentation for this struct was generated from the following file:

- uvc_dev.h

5.235 uvc_vs_format Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_vs_format:



Data Fields

- uint8_t len
- union {
  struct uvc_format_uncompressed * p_uncomp
  struct uvc_format_mjpeg * p_mjpg
  } format

- uint8_t num_frame
- uint8_t curr_framenum
- uint8_t frame_len
- union {
  struct uvc_frame_uncompressed * p_uncomp
  struct uvc_frame_mjpeg * p_mjpg
  } frame

5.235.1 Field Documentation

Kneron Confidential

5.235.1.1 curr_framenum

```
uint8_t uvc_vs_format::curr_framenum
```

5.235.1.2 format

```
union { ... } uvc_vs_format::format
```

5.235.1.3 frame

```
union { ... } uvc_vs_format::frame
```

5.235.1.4 frame_len

```
uint8_t uvc_vs_format::frame_len
```

5.235.1.5 len

```
uint8_t uvc_vs_format::len
```

5.235.1.6 num_frame

```
uint8_t uvc_vs_format::num_frame
```

5.235.1.7 p_mjpg [1/2]

```
struct uvc_format_mjpeg* uvc_vs_format::p_mjpg
```

5.235.1.8 p_mjpg [2/2]

```
struct uvc_frame_mjpeg* uvc_vs_format::p_mjpg
```

Kneron Confidential

5.235.1.9 p_uncomp [1/2]

```
struct uvc_format_uncompressed* uvc_vs_format::p_uncomp
```

5.235.1.10 p_uncomp [2/2]

```
struct uvc_frame_uncompressed* uvc_vs_format::p_uncomp
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.236 uvc_vs_still_control_data Struct Reference

```
#include <uvc.h>
```

Data Fields

- uint8_t bFormatIndex
- uint8_t bFrameIndex
- uint8_t bCompressionIndex
- uint32_t dwMaxVideoFrameSize
- uint32_t dwMaxPayloadTransferSize

5.236.1 Field Documentation

5.236.1.1 bCompressionIndex

```
uint8_t uvc_vs_still_control_data::bCompressionIndex
```

5.236.1.2 bFormatIndex

```
uint8_t uvc_vs_still_control_data::bFormatIndex
```

Kneron Confidential

5.236.1.3 bFrameIndex

```
uint8_t uvc_vs_still_control_data::bFrameIndex
```

5.236.1.4 dwMaxPayloadTransferSize

```
uint32_t uvc_vs_still_control_data::dwMaxPayloadTransferSize
```

5.236.1.5 dwMaxVideoFrameSize

```
uint32_t uvc_vs_still_control_data::dwMaxVideoFrameSize
```

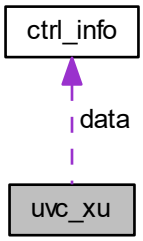
The documentation for this struct was generated from the following file:

- uvc.h

5.237 uvc_xu Struct Reference

```
#include <uvc_dev.h>
```

Collaboration diagram for uvc_xu:



Kneron Confidential

Data Fields

- uint8_t id
- uint8_t bNrInPins
- uint8_t * baSourceID
- uint8_t bNumControls
- uint32_t bmControls
- struct ctrl_info * data

5.237.1 Field Documentation

5.237.1.1 baSourceID

```
uint8_t* uvc_xu::baSourceID
```

5.237.1.2 bmControls

```
uint32_t uvc_xu::bmControls
```

5.237.1.3 bNrInPins

```
uint8_t uvc_xu::bNrInPins
```

5.237.1.4 bNumControls

```
uint8_t uvc_xu::bNumControls
```

5.237.1.5 data

```
struct ctrl_info* uvc_xu::data
```

5.237.1.6 id

```
uint8_t uvc_xu::id
```

The documentation for this struct was generated from the following file:

- uvc_dev.h

5.238 uvc_xu_control_query Struct Reference

```
#include <uvc_ctrl.h>
```

Kneron Confidential

Data Fields

- uint8_t unit
- uint8_t selector
- uint8_t query
- uint16_t size
- uint8_t * data

5.238.1 Field Documentation

5.238.1.1 data

```
uint8_t* uvc_xu_control_query::data
```

5.238.1.2 query

```
uint8_t uvc_xu_control_query::query
```

5.238.1.3 selector

```
uint8_t uvc_xu_control_query::selector
```

5.238.1.4 size

```
uint16_t uvc_xu_control_query::size
```

5.238.1.5 unit

```
uint8_t uvc_xu_control_query::unit
```

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

5.239 vs_generatekeyframe Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint8_t bGenerateKeyFrame

5.239.1 Field Documentation

5.239.1.1 bGenerateKeyFrame

`uint8_t vs_generatekeyframe::bGenerateKeyFrame`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.240 vs_still_image_trigger Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint8_t bTrigger

5.240.1 Field Documentation

5.240.1.1 bTrigger

`uint8_t vs_still_image_trigger::bTrigger`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

5.241 vs_streamerrorcode Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint8_t bStreamErrorCode

5.241.1 Field Documentation

5.241.1.1 bStreamErrorCode

`uint8_t vs_streamerrorcode::bStreamErrorCode`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

5.242 vs_synch_delay Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint16_t wDelay

5.242.1 Field Documentation

5.242.1.1 wDelay

`uint16_t vs_synch_delay::wDelay`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

5.243 vs_updateframe Struct Reference

`#include <uvc_ctrl.h>`

Data Fields

- uint8_t bStartFrameSegment
- uint8_t bEndFrameSegment

5.243.1 Field Documentation

5.243.1.1 bEndFrameSegment

`uint8_t vs_updateframe::bEndFrameSegment`

5.243.1.2 bStartFrameSegment

`uint8_t vs_updateframe::bStartFrameSegment`

The documentation for this struct was generated from the following file:

- uvc_ctrl.h

Kneron Confidential

Kneron Confidential

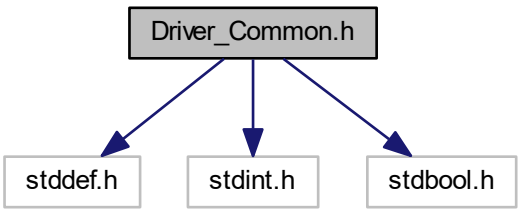
Chapter 6

File Documentation

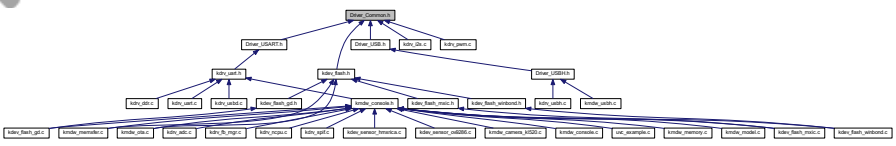
6.1 Driver_Common.h File Reference

```
#include <stddef.h>
#include <stdint.h>
#include <stdbool.h>
```

Include dependency graph for Driver_Common.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct _ARM_DRIVER_VERSION

  Driver Version.

Knero Confidential

Macros

- #define ARM_DRIVER_VERSION_MAJOR_MINOR(major, minor) (((major) << 8) | (minor))
- #define ARM_DRIVER_OK 0

  Operation succeeded.
- #define ARM_DRIVER_ERROR -1

  Unspecified error.
- #define ARM_DRIVER_ERROR_BUSY -2

  Driver is busy.
- #define ARM_DRIVER_ERROR_TIMEOUT -3

  Timeout occurred.
- #define ARM_DRIVER_ERROR_UNSUPPORTED -4

  Operation not supported.
- #define ARM_DRIVER_ERROR_PARAMETER -5

  Parameter error.
- #define ARM_DRIVER_ERROR_SPECIFIC -6

  Start of driver specific errors.

Typedefs

- typedef struct _ARM_DRIVER_VERSION ARM_DRIVER_VERSION

  Driver Version.
- typedef enum _ARM_POWER_STATE ARM_POWER_STATE

  General power states.

Enumerations

- enum _ARM_POWER_STATE { ARM_POWER_OFF, ARM_POWER_LOW, ARM_POWER_FULL }

  General power states.

6.1.1 Macro Definition Documentation

6.1.1.1 ARM_DRIVER_ERROR

```
#define ARM_DRIVER_ERROR -1
```

Unspecified error.

6.1.1.2 ARM_DRIVER_ERROR_BUSY

```
#define ARM_DRIVER_ERROR_BUSY -2
```

Driver is busy.

Kneron Confidential

6.1.1.3 ARM_DRIVER_ERROR_PARAMETER

```
#define ARM_DRIVER_ERROR_PARAMETER -5
```

Parameter error.

6.1.1.4 ARM_DRIVER_ERROR_SPECIFIC

```
#define ARM_DRIVER_ERROR_SPECIFIC -6
```

Start of driver specific errors.

6.1.1.5 ARM_DRIVER_ERROR_TIMEOUT

```
#define ARM_DRIVER_ERROR_TIMEOUT -3
```

Timeout occurred.

6.1.1.6 ARM_DRIVER_ERROR_UNSUPPORTED

```
#define ARM_DRIVER_ERROR_UNSUPPORTED -4
```

Operation not supported.

6.1.1.7 ARM_DRIVER_OK

```
#define ARM_DRIVER_OK 0
```

Operation succeeded.

6.1.1.8 ARM_DRIVER_VERSION_MAJOR_MINOR

```
#define ARM_DRIVER_VERSION_MAJOR_MINOR(
            major,
            minor ) (((major) << 8) | (minor))
```

Kneron Confidential

6.1.2 Typedef Documentation

6.1.2.1 ARM_DRIVER_VERSION

```
typedef struct _ARM_DRIVER_VERSION ARM_DRIVER_VERSION
```

Driver Version.

6.1.2.2 ARM_POWER_STATE

```
typedef enum _ARM_POWER_STATE ARM_POWER_STATE
```

General power states.

6.1.3 Enumeration Type Documentation

6.1.3.1 _ARM_POWER_STATE

```
enum _ARM_POWER_STATE
```

General power states.

Enumerator

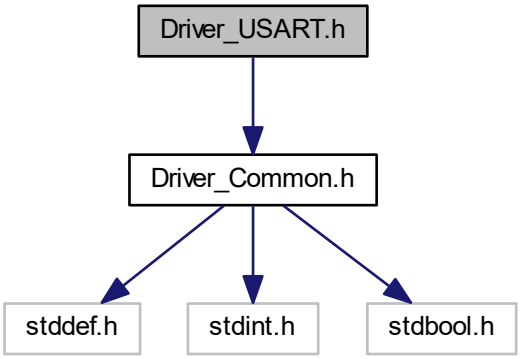
ARM_POWER_OFF	Power off: no operation possible.
ARM_POWER_LOW	Low Power mode: retain state, detect and signal wake-up events.
ARM_POWER_FULL	Power on: full operation at maximum performance.

Kneron Confidential

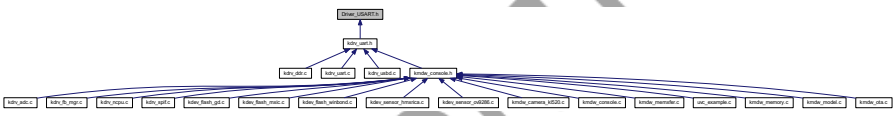
6.2 Driver_USART.h File Reference

```
#include "Driver_Common.h"
```

Include dependency graph for Driver_USART.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct _ARM_USART_STATUS
  USART Status.
- struct _ARM_USART_MODEM_STATUS
  USART Modem Status.
- struct _ARM_USART_CAPABILITIES
  USART Device Driver Capabilities.
- struct _ARM_DRIVER_USART
  Access structure of the USART Driver.

Macros

- #define ARM_USART_API_VERSION ARM_DRIVER_VERSION_MAJOR_MINOR(2,01) /* API version */
- #define ARM_USART_CONTROL_Pos 0
- #define ARM_USART_CONTROL_Msk (0xFFUL << ARM_USART_CONTROL_Pos)
- #define ARM_USART_MODE_ASYNCHRONOUS (0x01UL << ARM_USART_CONTROL_Pos)
  UART (Asynchronous); arg = Baudrate.
- #define ARM_USART_MODE_SYNCHRONOUS_MASTER (0x02UL << ARM_USART_CONTROL_Pos)
  Synchronous Master (generates clock signal); arg = Baudrate.
- #define ARM_USART_MODE_SYNCHRONOUS_SLAVE (0x03UL << ARM_USART_CONTROL_Pos)
  Synchronous Slave (external clock signal)

Kneron Confidential

- #define ARM_USART_MODE_SINGLE_WIRE (0x04UL << ARM_USART_CONTROL_Pos)

  UART Single-wire (half-duplex); arg = Baudrate.
- #define ARM_USART_MODE_IRDA (0x05UL << ARM_USART_CONTROL_Pos)

  UART IrDA; arg = Baudrate.
- #define ARM_USART_MODE_SMART_CARD (0x06UL << ARM_USART_CONTROL_Pos)

  UART Smart Card; arg = Baudrate.
- #define ARM_USART_DATA_BITS_Pos 8
- #define ARM_USART_DATA_BITS_Msk (7UL << ARM_USART_DATA_BITS_Pos)
- #define ARM_USART_DATA_BITS_5 (5UL << ARM_USART_DATA_BITS_Pos)

  5 Data bits
- #define ARM_USART_DATA_BITS_6 (6UL << ARM_USART_DATA_BITS_Pos)

  6 Data bit
- #define ARM_USART_DATA_BITS_7 (7UL << ARM_USART_DATA_BITS_Pos)

  7 Data bits
- #define ARM_USART_DATA_BITS_8 (0UL << ARM_USART_DATA_BITS_Pos)

  8 Data bits (default)
- #define ARM_USART_DATA_BITS_9 (1UL << ARM_USART_DATA_BITS_Pos)

  9 Data bits
- #define ARM_USART_PARITY_Pos 12
- #define ARM_USART_PARITY_Msk (3UL << ARM_USART_PARITY_Pos)
- #define ARM_USART_PARITY_NONE (0UL << ARM_USART_PARITY_Pos)

  No Parity (default)
- #define ARM_USART_PARITY_EVEN (1UL << ARM_USART_PARITY_Pos)

  Even Parity.
- #define ARM_USART_PARITY_ODD (2UL << ARM_USART_PARITY_Pos)

  Odd Parity.
- #define ARM_USART_STOP_BITS_Pos 14
- #define ARM_USART_STOP_BITS_Msk (3UL << ARM_USART_STOP_BITS_Pos)
- #define ARM_USART_STOP_BITS_1 (0UL << ARM_USART_STOP_BITS_Pos)

  1 Stop bit (default)
- #define ARM_USART_STOP_BITS_2 (1UL << ARM_USART_STOP_BITS_Pos)

  2 Stop bits
- #define ARM_USART_STOP_BITS_1_5 (2UL << ARM_USART_STOP_BITS_Pos)

  1.5 Stop bits
- #define ARM_USART_STOP_BITS_0_5 (3UL << ARM_USART_STOP_BITS_Pos)

  0.5 Stop bits
- #define ARM_USART_FLOW_CONTROL_Pos 16
- #define ARM_USART_FLOW_CONTROL_Msk (3UL << ARM_USART_FLOW_CONTROL_Pos)
- #define ARM_USART_FLOW_CONTROL_NONE (0UL << ARM_USART_FLOW_CONTROL_Pos)

  No Flow Control (default)
- #define ARM_USART_FLOW_CONTROL_RTS (1UL << ARM_USART_FLOW_CONTROL_Pos)

  RTS Flow Control.
- #define ARM_USART_FLOW_CONTROL_CTS (2UL << ARM_USART_FLOW_CONTROL_Pos)

  CTS Flow Control.
- #define ARM_USART_FLOW_CONTROL_RTS_CTS (3UL << ARM_USART_FLOW_CONTROL_Pos)

  RTS/CTS Flow Control.
- #define ARM_USART_CPOL_Pos 17
- #define ARM_USART_CPOL_Msk (1UL << ARM_USART_CPOL_Pos)
- #define ARM_USART_CPOL0 (0UL << ARM_USART_CPOL_Pos)

  CPOL = 0 (default)
- #define ARM_USART_CPOL1 (1UL << ARM_USART_CPOL_Pos)

Kneron Confidential

CPOL = 1.

- #define ARM_USART_CPHA_Pos 18
- #define ARM_USART_CPHA_Msk (1UL << ARM_USART_CPHA_Pos)
- #define ARM_USART_CPHA0 (0UL << ARM_USART_CPHA_Pos)

  CPHA = 0 (default)
- #define ARM_USART_CPHA1 (1UL << ARM_USART_CPHA_Pos)

  CPHA = 1.
- #define ARM_USART_SET_DEFAULT_TX_VALUE (0x10UL << ARM_USART_CONTROL_Pos)

  Set default Transmit value (Synchronous Receive only); arg = value.
- #define ARM_USART_SET_IRDA_PULSE (0x11UL << ARM_USART_CONTROL_Pos)

  Set IrDA Pulse in ns; arg: 0=3/16 of bit period
- #define ARM_USART_SET_SMART_CARD_GUARD_TIME (0x12UL << ARM_USART_CONTROL_Pos)

  Set Smart Card Guard Time; arg = number of bit periods.
- #define ARM_USART_SET_SMART_CARD_CLOCK (0x13UL << ARM_USART_CONTROL_Pos)

  Set Smart Card Clock in Hz; arg: 0=Clock not generated.
- #define ARM_USART_CONTROL_SMART_CARD_NACK (0x14UL << ARM_USART_CONTROL_Pos)

  Smart Card NACK generation; arg: 0=disabled, 1=enabled.
- #define ARM_USART_CONTROL_TX (0x15UL << ARM_USART_CONTROL_Pos)

  Transmitter; arg: 0=disabled, 1=enabled.
- #define ARM_USART_CONTROL_RX (0x16UL << ARM_USART_CONTROL_Pos)

  Receiver; arg: 0=disabled, 1=enabled.
- #define ARM_USART_CONTROL_BREAK (0x17UL << ARM_USART_CONTROL_Pos)

  Continuous Break transmission; arg: 0=disabled, 1=enabled.
- #define ARM_USART_ABORT_SEND (0x18UL << ARM_USART_CONTROL_Pos)

  Abort ARM_USART_Send.
- #define ARM_USART_ABORT_RECEIVE (0x19UL << ARM_USART_CONTROL_Pos)

  Abort ARM_USART_Receive.
- #define ARM_USART_ABORT_TRANSFER (0x1AUL << ARM_USART_CONTROL_Pos)

  Abort ARM_USART_Transfer.
- #define ARM_USART_ERROR_MODE (ARM_DRIVER_ERROR_SPECIFIC - 1)

  Specified Mode not supported.
- #define ARM_USART_ERROR_BAUDRATE (ARM_DRIVER_ERROR_SPECIFIC - 2)

  Specified baudrate not supported.
- #define ARM_USART_ERROR_DATA_BITS (ARM_DRIVER_ERROR_SPECIFIC - 3)

  Specified number of Data bits not supported.
- #define ARM_USART_ERROR_PARITY (ARM_DRIVER_ERROR_SPECIFIC - 4)

  Specified Parity not supported.
- #define ARM_USART_ERROR_STOP_BITS (ARM_DRIVER_ERROR_SPECIFIC - 5)

  Specified number of Stop bits not supported.
- #define ARM_USART_ERROR_FLOW_CONTROL (ARM_DRIVER_ERROR_SPECIFIC - 6)

  Specified Flow Control not supported.
- #define ARM_USART_ERROR_CPOL (ARM_DRIVER_ERROR_SPECIFIC - 7)

  Specified Clock Polarity not supported.
- #define ARM_USART_ERROR_CPHA (ARM_DRIVER_ERROR_SPECIFIC - 8)

  Specified Clock Phase not supported.
- #define ARM_USART_EVENT_SEND_COMPLETE (1UL << 0)

  Send completed; however USART may still transmit data.
- #define ARM_USART_EVENT_RECEIVE_COMPLETE (1UL << 1)

  Receive completed.
- #define ARM_USART_EVENT_TRANSFER_COMPLETE (1UL << 2)

Kneron Confidential

Transfer completed.

- #define ARM_USART_EVENT_TX_COMPLETE (1UL << 3)

  Transmit completed (optional)
- #define ARM_USART_EVENT_TX_UNDERFLOW (1UL << 4)

  Transmit data not available (Synchronous Slave)
- #define ARM_USART_EVENT_RX_OVERFLOW (1UL << 5)

  Receive data overflow.
- #define ARM_USART_EVENT_RX_TIMEOUT (1UL << 6)

  Receive character timeout (optional)
- #define ARM_USART_EVENT_RX_BREAK (1UL << 7)

  Break detected on receive.
- #define ARM_USART_EVENT_RX_FRAMING_ERROR (1UL << 8)

  Framing error detected on receive.
- #define ARM_USART_EVENT_RX_PARITY_ERROR (1UL << 9)

  Parity error detected on receive.
- #define ARM_USART_EVENT_CTS (1UL << 10)

  CTS state changed (optional)
- #define ARM_USART_EVENT_DSR (1UL << 11)

  DSR state changed (optional)
- #define ARM_USART_EVENT_DCD (1UL << 12)

  DCD state changed (optional)
- #define ARM_USART_EVENT_RI (1UL << 13)

  RI state changed (optional)

Typedefs

- typedef struct _ARM_USART_STATUS ARM_USART_STATUS

  USART Status.
- typedef enum _ARM_USART_MODEM_CONTROL ARM_USART_MODEM_CONTROL

  USART Modem Control.
- typedef struct _ARM_USART_MODEM_STATUS ARM_USART_MODEM_STATUS

  USART Modem Status.
- typedef void(* ARM_USART_SignalEvent_t) (uint32_t event)

  Pointer to ARM_USART_SignalEvent : Signal USART Event.
- typedef struct _ARM_USART_CAPABILITIES ARM_USART_CAPABILITIES

  USART Device Driver Capabilities.
- typedef struct _ARM_DRIVER_USART ARM_DRIVER_USART

  Access structure of the USART Driver.

Enumerations

- enum _ARM_USART_MODEM_CONTROL { ARM_USART_RTS_CLEAR, ARM_USART_RTS_SET, ARM_USART_DTR_CLEAR, ARM_USART_DTR_SET }

  USART Modem Control.

6.2.1 Macro Definition Documentation

Kneron Confidential

6.2.1.1 ARM_USART_ABORT_RECEIVE

```
#define ARM_USART_ABORT_RECEIVE (0x19UL << ARM_USART_CONTROL_Pos)
```

Abort ARM_USART_Receive.

6.2.1.2 ARM_USART_ABORT_SEND

```
#define ARM_USART_ABORT_SEND (0x18UL << ARM_USART_CONTROL_Pos)
```

Abort ARM_USART_Send.

6.2.1.3 ARM_USART_ABORT_TRANSFER

```
#define ARM_USART_ABORT_TRANSFER (0x1AUL << ARM_USART_CONTROL_Pos)
```

Abort ARM_USART_Transfer.

6.2.1.4 ARM_USART_API_VERSION

```
#define ARM_USART_API_VERSION ARM_DRIVER_VERSION_MAJOR_MINOR(2,01) /* API version */
```

6.2.1.5 ARM_USART_CONTROL_BREAK

```
#define ARM_USART_CONTROL_BREAK (0x17UL << ARM_USART_CONTROL_Pos)
```

Continuous Break transmission; arg: 0=disabled, 1=enabled.

6.2.1.6 ARM_USART_CONTROL_Msk

```
#define ARM_USART_CONTROL_Msk (0xFFUL << ARM_USART_CONTROL_Pos)
```

6.2.1.7 ARM_USART_CONTROL_Pos

```
#define ARM_USART_CONTROL_Pos 0
```

Kneron Confidential

6.2.1.8 ARM_USART_CONTROL_RX

```
#define ARM_USART_CONTROL_RX (0x16UL << ARM_USART_CONTROL_Pos)
```

Receiver; arg: 0=disabled, 1=enabled.

6.2.1.9 ARM_USART_CONTROL_SMART_CARD_NACK

```
#define ARM_USART_CONTROL_SMART_CARD_NACK (0x14UL << ARM_USART_CONTROL_Pos)
```

Smart Card NACK generation; arg: 0=disabled, 1=enabled.

6.2.1.10 ARM_USART_CONTROL_TX

```
#define ARM_USART_CONTROL_TX (0x15UL << ARM_USART_CONTROL_Pos)
```

Transmitter; arg: 0=disabled, 1=enabled.

6.2.1.11 ARM_USART_CPHA0

```
#define ARM_USART_CPHA0 (0UL << ARM_USART_CPHA_Pos)
```

CPHA = 0 (default)

6.2.1.12 ARM_USART_CPHA1

```
#define ARM_USART_CPHA1 (1UL << ARM_USART_CPHA_Pos)
```

CPHA = 1.

6.2.1.13 ARM_USART_CPHA_Msk

```
#define ARM_USART_CPHA_Msk (1UL << ARM_USART_CPHA_Pos)
```

Kneron Confidential

6.2.1.14 ARM_USART_CPHA_Pos

```
#define ARM_USART_CPHA_Pos 18
```

6.2.1.15 ARM_USART_CPOL0

```
#define ARM_USART_CPOL0 (0UL << ARM_USART_CPOL_Pos)
```

CPOL = 0 (default)

6.2.1.16 ARM_USART_CPOL1

```
#define ARM_USART_CPOL1 (1UL << ARM_USART_CPOL_Pos)
```

CPOL = 1.

6.2.1.17 ARM_USART_CPOL_Msk

```
#define ARM_USART_CPOL_Msk (1UL << ARM_USART_CPOL_Pos)
```

6.2.1.18 ARM_USART_CPOL_Pos

```
#define ARM_USART_CPOL_Pos 17
```

6.2.1.19 ARM_USART_DATA_BITS_5

```
#define ARM_USART_DATA_BITS_5 (5UL << ARM_USART_DATA_BITS_Pos)
```

5 Data bits

6.2.1.20 ARM_USART_DATA_BITS_6

```
#define ARM_USART_DATA_BITS_6 (6UL << ARM_USART_DATA_BITS_Pos)
```

6 Data bit

Kneron Confidential

6.2.1.21 ARM_USART_DATA_BITS_7

```
#define ARM_USART_DATA_BITS_7 (7UL << ARM_USART_DATA_BITS_Pos)
```

7 Data bits

6.2.1.22 ARM_USART_DATA_BITS_8

```
#define ARM_USART_DATA_BITS_8 (0UL << ARM_USART_DATA_BITS_Pos)
```

8 Data bits (default)

6.2.1.23 ARM_USART_DATA_BITS_9

```
#define ARM_USART_DATA_BITS_9 (1UL << ARM_USART_DATA_BITS_Pos)
```

9 Data bits

6.2.1.24 ARM_USART_DATA_BITS_Msk

```
#define ARM_USART_DATA_BITS_Msk (7UL << ARM_USART_DATA_BITS_Pos)
```

6.2.1.25 ARM_USART_DATA_BITS_Pos

```
#define ARM_USART_DATA_BITS_Pos 8
```

6.2.1.26 ARM_USART_ERROR_BAUDRATE

```
#define ARM_USART_ERROR_BAUDRATE (ARM_DRIVER_ERROR_SPECIFIC - 2)
```

Specified baudrate not supported.

Kneron Confidential

6.2.1.27 ARM_USART_ERROR_CPHA

```
#define ARM_USART_ERROR_CPHA (ARM_DRIVER_ERROR_SPECIFIC - 8)
```

Specified Clock Phase not supported.

6.2.1.28 ARM_USART_ERROR_CPOL

```
#define ARM_USART_ERROR_CPOL (ARM_DRIVER_ERROR_SPECIFIC - 7)
```

Specified Clock Polarity not supported.

6.2.1.29 ARM_USART_ERROR_DATA_BITS

```
#define ARM_USART_ERROR_DATA_BITS (ARM_DRIVER_ERROR_SPECIFIC - 3)
```

Specified number of Data bits not supported.

6.2.1.30 ARM_USART_ERROR_FLOW_CONTROL

```
#define ARM_USART_ERROR_FLOW_CONTROL (ARM_DRIVER_ERROR_SPECIFIC - 6)
```

Specified Flow Control not supported.

6.2.1.31 ARM_USART_ERROR_MODE

```
#define ARM_USART_ERROR_MODE (ARM_DRIVER_ERROR_SPECIFIC - 1)
```

Specified Mode not supported.

6.2.1.32 ARM_USART_ERROR_PARITY

```
#define ARM_USART_ERROR_PARITY (ARM_DRIVER_ERROR_SPECIFIC - 4)
```

Specified Parity not supported.

Kneron Confidential

6.2.1.33 ARM_USART_ERROR_STOP_BITS

```
#define ARM_USART_ERROR_STOP_BITS (ARM_DRIVER_ERROR_SPECIFIC - 5)
```

Specified number of Stop bits not supported.

6.2.1.34 ARM_USART_EVENT_CTS

```
#define ARM_USART_EVENT_CTS (1UL << 10)
```

CTS state changed (optional)

6.2.1.35 ARM_USART_EVENT_DCD

```
#define ARM_USART_EVENT_DCD (1UL << 12)
```

DCD state changed (optional)

6.2.1.36 ARM_USART_EVENT_DSR

```
#define ARM_USART_EVENT_DSR (1UL << 11)
```

DSR state changed (optional)

6.2.1.37 ARM_USART_EVENT_RECEIVE_COMPLETE

```
#define ARM_USART_EVENT_RECEIVE_COMPLETE (1UL << 1)
```

Receive completed.

6.2.1.38 ARM_USART_EVENT_RI

```
#define ARM_USART_EVENT_RI (1UL << 13)
```

RI state changed (optional)

Kneron Confidential

6.2.1.39 ARM_USART_EVENT_RX_BREAK

```
#define ARM_USART_EVENT_RX_BREAK (1UL << 7)
```

Break detected on receive.

6.2.1.40 ARM_USART_EVENT_RX_FRAMING_ERROR

```
#define ARM_USART_EVENT_RX_FRAMING_ERROR (1UL << 8)
```

Framing error detected on receive.

6.2.1.41 ARM_USART_EVENT_RX_OVERFLOW

```
#define ARM_USART_EVENT_RX_OVERFLOW (1UL << 5)
```

Receive data overflow.

6.2.1.42 ARM_USART_EVENT_RX_PARITY_ERROR

```
#define ARM_USART_EVENT_RX_PARITY_ERROR (1UL << 9)
```

Parity error detected on receive.

6.2.1.43 ARM_USART_EVENT_RX_TIMEOUT

```
#define ARM_USART_EVENT_RX_TIMEOUT (1UL << 6)
```

Receive character timeout (optional)

6.2.1.44 ARM_USART_EVENT_SEND_COMPLETE

```
#define ARM_USART_EVENT_SEND_COMPLETE (1UL << 0)
```

Send completed; however USART may still transmit data.

Kneron Confidential

6.2.1.45 ARM_USART_EVENT_TRANSFER_COMPLETE

```
#define ARM_USART_EVENT_TRANSFER_COMPLETE (1UL << 2)
```

Transfer completed.

6.2.1.46 ARM_USART_EVENT_TX_COMPLETE

```
#define ARM_USART_EVENT_TX_COMPLETE (1UL << 3)
```

Transmit completed (optional)

6.2.1.47 ARM_USART_EVENT_TX_UNDERFLOW

```
#define ARM_USART_EVENT_TX_UNDERFLOW (1UL << 4)
```

Transmit data not available (Synchronous Slave)

6.2.1.48 ARM_USART_FLOW_CONTROL_CTS

```
#define ARM_USART_FLOW_CONTROL_CTS (2UL << ARM_USART_FLOW_CONTROL_Pos)
```

CTS Flow Control.

6.2.1.49 ARM_USART_FLOW_CONTROL_Msk

```
#define ARM_USART_FLOW_CONTROL_Msk (3UL << ARM_USART_FLOW_CONTROL_Pos)
```

6.2.1.50 ARM_USART_FLOW_CONTROL_NONE

```
#define ARM_USART_FLOW_CONTROL_NONE (0UL << ARM_USART_FLOW_CONTROL_Pos)
```

No Flow Control (default)

Kneron Confidential

6.2.1.51 ARM_USART_FLOW_CONTROL_Pos

```
#define ARM_USART_FLOW_CONTROL_Pos 16
```

6.2.1.52 ARM_USART_FLOW_CONTROL_RTS

```
#define ARM_USART_FLOW_CONTROL_RTS (1UL << ARM_USART_FLOW_CONTROL_Pos)
```

RTS Flow Control.

6.2.1.53 ARM_USART_FLOW_CONTROL_RTS_CTS

```
#define ARM_USART_FLOW_CONTROL_RTS_CTS (3UL << ARM_USART_FLOW_CONTROL_Pos)
```

RTS/CTS Flow Control.

6.2.1.54 ARM_USART_MODE_ASYNCHRONOUS

```
#define ARM_USART_MODE_ASYNCHRONOUS (0x01UL << ARM_USART_CONTROL_Pos)
```

UART (Asynchronous); arg = Baudrate.

6.2.1.55 ARM_USART_MODE_IRDA

```
#define ARM_USART_MODE_IRDA (0x05UL << ARM_USART_CONTROL_Pos)
```

UART IrDA; arg = Baudrate.

6.2.1.56 ARM_USART_MODE_SINGLE_WIRE

```
#define ARM_USART_MODE_SINGLE_WIRE (0x04UL << ARM_USART_CONTROL_Pos)
```

UART Single-wire (half-duplex); arg = Baudrate.

Kneron Confidential

6.2.1.57 ARM_USART_MODE_SMART_CARD

```
#define ARM_USART_MODE_SMART_CARD (0x06UL << ARM_USART_CONTROL_Pos)
```

UART Smart Card; arg = Baudrate.

6.2.1.58 ARM_USART_MODE_SYNCHRONOUS_MASTER

```
#define ARM_USART_MODE_SYNCHRONOUS_MASTER (0x02UL << ARM_USART_CONTROL_Pos)
```

Synchronous Master (generates clock signal); arg = Baudrate.

6.2.1.59 ARM_USART_MODE_SYNCHRONOUS_SLAVE

```
#define ARM_USART_MODE_SYNCHRONOUS_SLAVE (0x03UL << ARM_USART_CONTROL_Pos)
```

Synchronous Slave (external clock signal)

6.2.1.60 ARM_USART_PARITY_EVEN

```
#define ARM_USART_PARITY_EVEN (1UL << ARM_USART_PARITY_Pos)
```

Even Parity.

6.2.1.61 ARM_USART_PARITY_Msk

```
#define ARM_USART_PARITY_Msk (3UL << ARM_USART_PARITY_Pos)
```

6.2.1.62 ARM_USART_PARITY_NONE

```
#define ARM_USART_PARITY_NONE (0UL << ARM_USART_PARITY_Pos)
```

No Parity (default)

Kneron Confidential

6.2.1.63 ARM_USART_PARITY_ODD

```
#define ARM_USART_PARITY_ODD (2UL << ARM_USART_PARITY_Pos)
```

Odd Parity.

6.2.1.64 ARM_USART_PARITY_Pos

```
#define ARM_USART_PARITY_Pos 12
```

6.2.1.65 ARM_USART_SET_DEFAULT_TX_VALUE

```
#define ARM_USART_SET_DEFAULT_TX_VALUE (0x10UL << ARM_USART_CONTROL_Pos)
```

Set default Transmit value (Synchronous Receive only); arg = value.

6.2.1.66 ARM_USART_SET_IRDA_PULSE

```
#define ARM_USART_SET_IRDA_PULSE (0x11UL << ARM_USART_CONTROL_Pos)
```

Set IrDA Pulse in ns; arg: 0=3/16 of bit period

6.2.1.67 ARM_USART_SET_SMART_CARD_CLOCK

```
#define ARM_USART_SET_SMART_CARD_CLOCK (0x13UL << ARM_USART_CONTROL_Pos)
```

Set Smart Card Clock in Hz; arg: 0=Clock not generated.

6.2.1.68 ARM_USART_SET_SMART_CARD_GUARD_TIME

```
#define ARM_USART_SET_SMART_CARD_GUARD_TIME (0x12UL << ARM_USART_CONTROL_Pos)
```

Set Smart Card Guard Time; arg = number of bit periods.

Kneron Confidential

6.2.1.69 ARM_USART_STOP_BITS_0_5

```
#define ARM_USART_STOP_BITS_0_5 (3UL << ARM_USART_STOP_BITS_Pos)
```

0.5 Stop bits

6.2.1.70 ARM_USART_STOP_BITS_1

```
#define ARM_USART_STOP_BITS_1 (0UL << ARM_USART_STOP_BITS_Pos)
```

1 Stop bit (default)

6.2.1.71 ARM_USART_STOP_BITS_1_5

```
#define ARM_USART_STOP_BITS_1_5 (2UL << ARM_USART_STOP_BITS_Pos)
```

1.5 Stop bits

6.2.1.72 ARM_USART_STOP_BITS_2

```
#define ARM_USART_STOP_BITS_2 (1UL << ARM_USART_STOP_BITS_Pos)
```

2 Stop bits

6.2.1.73 ARM_USART_STOP_BITS_Msk

```
#define ARM_USART_STOP_BITS_Msk (3UL << ARM_USART_STOP_BITS_Pos)
```

6.2.1.74 ARM_USART_STOP_BITS_Pos

```
#define ARM_USART_STOP_BITS_Pos 14
```

6.2.2 Typedef Documentation

Kneron Confidential

6.2.2.1 ARM_DRIVER_USART

```
typedef struct _ARM_DRIVER_USART ARM_DRIVER_USART
```

Access structure of the USART Driver.

6.2.2.2 ARM_USART_CAPABILITIES

```
typedef struct _ARM_USART_CAPABILITIES ARM_USART_CAPABILITIES
```

USART Device Driver Capabilities.

6.2.2.3 ARM_USART_MODEM_CONTROL

```
typedef enum _ARM_USART_MODEM_CONTROL ARM_USART_MODEM_CONTROL
```

USART Modem Control.

6.2.2.4 ARM_USART_MODEM_STATUS

```
typedef struct _ARM_USART_MODEM_STATUS ARM_USART_MODEM_STATUS
```

USART Modem Status.

6.2.2.5 ARM_USART_SignalEvent_t

```
typedef void(* ARM_USART_SignalEvent_t) (uint32_t event)
```

Pointer to ARM_USART_SignalEvent : Signal USART Event.

6.2.2.6 ARM_USART_STATUS

```
typedef struct _ARM_USART_STATUS ARM_USART_STATUS
```

USART Status.

6.2.3 Enumeration Type Documentation

6.2.3.1 _ARM_USART_MODEM_CONTROL

```
enum _ARM_USART_MODEM_CONTROL
```

USART Modem Control.

Kneron Confidential

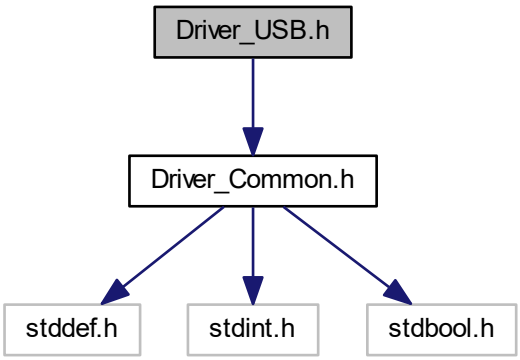
Enumerator

ARM_USART_RTS_CLEAR	Deactivate RTS.
ARM_USART_RTS_SET	Activate RTS.
ARM_USART_DTR_CLEAR	Deactivate DTR.
ARM_USART_DTR_SET	Activate DTR.

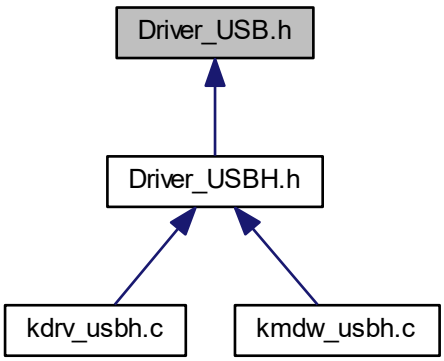
6.3 Driver_USB.h File Reference

```
#include "Driver_Common.h"
```

Include dependency graph for Driver_USB.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define ARM_USB_ROLE_NONE (0)
- #define ARM_USB_ROLE_HOST (1)
- #define ARM_USB_ROLE_DEVICE (2)
- #define ARM_USB_PIN_DP (1 << 0)

  USB D+ pin.
- #define ARM_USB_PIN_DM (1 << 1)

  USB D- pin.
- #define ARM_USB_PIN_VBUS (1 << 2)

  USB VBUS pin.
- #define ARM_USB_PIN_OC (1 << 3)

  USB OverCurrent pin.
- #define ARM_USB_PIN_ID (1 << 4)

  USB ID pin.
- #define ARM_USB_SPEED_LOW (0)

  Low-speed USB.
- #define ARM_USB_SPEED_FULL (1)

  Full-speed USB.
- #define ARM_USB_SPEED_HIGH (2)

  High-speed USB.
- #define ARM_USB_PID_OUT (1)
- #define ARM_USB_PID_IN (9)
- #define ARM_USB_PID_SOF (5)
- #define ARM_USB_PID_SETUP (13)
- #define ARM_USB_PID_DATA0 (3)
- #define ARM_USB_PID_DATA1 (11)
- #define ARM_USB_PID_DATA2 (7)
- #define ARM_USB_PID_MDATA (15)
- #define ARM_USB_PID_ACK (2)
- #define ARM_USB_PID_NAK (10)
- #define ARM_USB_PID_STALL (14)
- #define ARM_USB_PID_NYET (6)
- #define ARM_USB_PID_PRE (12)
- #define ARM_USB_PID_ERR (12)
- #define ARM_USB_PID_SPLIT (8)
- #define ARM_USB_PID_PING (4)
- #define ARM_USB_PID_RESERVED (0)
- #define ARM_USB_ENDPOINT_NUMBER_MASK (0x0F)
- #define ARM_USB_ENDPOINT_DIRECTION_MASK (0x80)
- #define ARM_USB_ENDPOINT_CONTROL (0)

  Control Endpoint.
- #define ARM_USB_ENDPOINT_ISOCHRONOUS (1)

  Isochronous Endpoint.
- #define ARM_USB_ENDPOINT_BULK (2)

  Bulk Endpoint.
- #define ARM_USB_ENDPOINT_INTERRUPT (3)

  Interrupt Endpoint.
- #define ARM_USB_ENDPOINT_MAX_PACKET_SIZE_MASK (0x07FF)
- #define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_MASK (0x1800)
- #define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_1 (0x0000)
- #define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_2 (0x0800)
- #define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_3 (0x1000)

Kneron Confidential

6.3.1 Macro Definition Documentation

6.3.1.1 ARM_USB_ENDPOINT_BULK

```
#define ARM_USB_ENDPOINT_BULK (2)
```

Bulk Endpoint.

6.3.1.2 ARM_USB_ENDPOINT_CONTROL

```
#define ARM_USB_ENDPOINT_CONTROL (0)
```

Control Endpoint.

6.3.1.3 ARM_USB_ENDPOINT_DIRECTION_MASK

```
#define ARM_USB_ENDPOINT_DIRECTION_MASK (0x80)
```

6.3.1.4 ARM_USB_ENDPOINT_INTERRUPT

```
#define ARM_USB_ENDPOINT_INTERRUPT (3)
```

Interrupt Endpoint.

6.3.1.5 ARM_USB_ENDPOINT_ISOCHRONOUS

```
#define ARM_USB_ENDPOINT_ISOCHRONOUS (1)
```

Isochronous Endpoint.

6.3.1.6 ARM_USB_ENDPOINT_MAX_PACKET_SIZE_MASK

```
#define ARM_USB_ENDPOINT_MAX_PACKET_SIZE_MASK (0x07FF)
```

Kneron Confidential

6.3.1.7 ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_1

```
#define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_1 (0x0000)
```

6.3.1.8 ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_2

```
#define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_2 (0x0800)
```

6.3.1.9 ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_3

```
#define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_3 (0x1000)
```

6.3.1.10 ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_MASK

```
#define ARM_USB_ENDPOINT_MICROFRAME_TRANSACTIONS_MASK (0x1800)
```

6.3.1.11 ARM_USB_ENDPOINT_NUMBER_MASK

```
#define ARM_USB_ENDPOINT_NUMBER_MASK (0x0F)
```

6.3.1.12 ARM_USB_PID_ACK

```
#define ARM_USB_PID_ACK (2)
```

6.3.1.13 ARM_USB_PID_DATA0

```
#define ARM_USB_PID_DATA0 (3)
```

6.3.1.14 ARM_USB_PID_DATA1

```
#define ARM_USB_PID_DATA1 (11)
```

Kneron Confidential

6.3.1.15 ARM_USB_PID_DATA2

```
#define ARM_USB_PID_DATA2 (7)
```

6.3.1.16 ARM_USB_PID_ERR

```
#define ARM_USB_PID_ERR (12)
```

6.3.1.17 ARM_USB_PID_IN

```
#define ARM_USB_PID_IN (9)
```

6.3.1.18 ARM_USB_PID_MDATA

```
#define ARM_USB_PID_MDATA (15)
```

6.3.1.19 ARM_USB_PID_NAK

```
#define ARM_USB_PID_NAK (10)
```

6.3.1.20 ARM_USB_PID_NYET

```
#define ARM_USB_PID_NYET (6)
```

6.3.1.21 ARM_USB_PID_OUT

```
#define ARM_USB_PID_OUT (1)
```

6.3.1.22 ARM_USB_PID_PING

```
#define ARM_USB_PID_PING (4)
```

Kneron Confidential

6.3.1.23 ARM_USB_PID_PRE

```
#define ARM_USB_PID_PRE (12)
```

6.3.1.24 ARM_USB_PID_RESERVED

```
#define ARM_USB_PID_RESERVED (0)
```

6.3.1.25 ARM_USB_PID_SETUP

```
#define ARM_USB_PID_SETUP (13)
```

6.3.1.26 ARM_USB_PID_SOF

```
#define ARM_USB_PID_SOF (5)
```

6.3.1.27 ARM_USB_PID_SPLIT

```
#define ARM_USB_PID_SPLIT (8)
```

6.3.1.28 ARM_USB_PID_STALL

```
#define ARM_USB_PID_STALL (14)
```

6.3.1.29 ARM_USB_PIN_DM

```
#define ARM_USB_PIN_DM (1 << 1)
```

USB D- pin.

Kneron Confidential

6.3.1.30 ARM_USB_PIN_DP

```
#define ARM_USB_PIN_DP (1 << 0)
```

USB D+ pin.

6.3.1.31 ARM_USB_PIN_ID

```
#define ARM_USB_PIN_ID (1 << 4)
```

USB ID pin.

6.3.1.32 ARM_USB_PIN_OC

```
#define ARM_USB_PIN_OC (1 << 3)
```

USB OverCurrent pin.

6.3.1.33 ARM_USB_PIN_VBUS

```
#define ARM_USB_PIN_VBUS (1 << 2)
```

USB VBUS pin.

6.3.1.34 ARM_USB_ROLE_DEVICE

```
#define ARM_USB_ROLE_DEVICE (2)
```

6.3.1.35 ARM_USB_ROLE_HOST

```
#define ARM_USB_ROLE_HOST (1)
```

6.3.1.36 ARM_USB_ROLE_NONE

```
#define ARM_USB_ROLE_NONE (0)
```

Kneron Confidential

6.3.1.37 ARM_USB_SPEED_FULL

```
#define ARM_USB_SPEED_FULL (1)
```

Full-speed USB.

6.3.1.38 ARM_USB_SPEED_HIGH

```
#define ARM_USB_SPEED_HIGH (2)
```

High-speed USB.

6.3.1.39 ARM_USB_SPEED_LOW

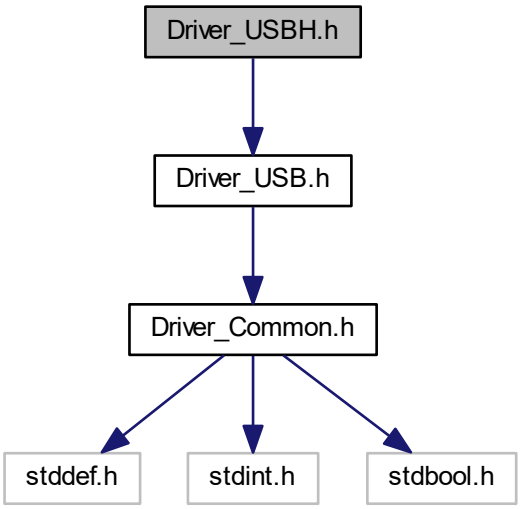
```
#define ARM_USB_SPEED_LOW (0)
```

Low-speed USB.

6.4 Driver_USBH.h File Reference

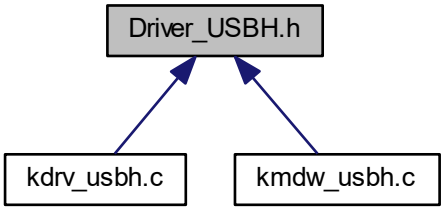
```
#include "Driver_USB.h"
```

Include dependency graph for Driver_USBH.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct _ARM_USBH_PORT_STATE

  USB Host Port State.
- struct _ARM_USBH_CAPABILITIES

  USB Host Driver Capabilities.
- struct _ARM_DRIVER_USBH

  Access structure of USB Host Driver.
- struct _ARM_USBH_HCI_CAPABILITIES

  USB Host HCI (OHCI/EHCI) Driver Capabilities.
- struct _ARM_DRIVER_USBH_HCI

  Access structure of USB Host HCI (OHCI/EHCI) Driver.

Macros

- #define ARM_USBH_API_VERSION ARM_DRIVER_VERSION_MAJOR_MINOR(2,2) /* API version */
- #define ARM_USBH_EP_HANDLE ARM_USBH_PIPE_HANDLE /* Legacy name */
- #define ARM_USBH_PACKET_TOKEN_Pos 0
- #define ARM_USBH_PACKET_TOKEN_Msk (0x0FUL << ARM_USBH_PACKET_TOKEN_Pos)
- #define ARM_USBH_PACKET_SETUP (0x01UL << ARM_USBH_PACKET_TOKEN_Pos)

  SETUP Packet.
- #define ARM_USBH_PACKET_OUT (0x02UL << ARM_USBH_PACKET_TOKEN_Pos)

  OUT Packet.
- #define ARM_USBH_PACKET_IN (0x03UL << ARM_USBH_PACKET_TOKEN_Pos)

  IN Packet.
- #define ARM_USBH_PACKET_PING (0x04UL << ARM_USBH_PACKET_TOKEN_Pos)

  PING Packet.
- #define ARM_USBH_PACKET_DATA_Pos 4
- #define ARM_USBH_PACKET_DATA_Msk (0x0FUL << ARM_USBH_PACKET_DATA_Pos)
- #define ARM_USBH_PACKET_DATA0 (0x01UL << ARM_USBH_PACKET_DATA_Pos)

  DATA0 PID.
- #define ARM_USBH_PACKET_DATA1 (0x02UL << ARM_USBH_PACKET_DATA_Pos)

  DATA1 PID.
- #define ARM_USBH_PACKET_SPLIT_Pos 8
- #define ARM_USBH_PACKET_SPLIT_Msk (0x0FUL << ARM_USBH_PACKET_SPLIT_Pos)

Kneron Confidential

- #define ARM_USBH_PACKET_SSPLIT (0x08UL << ARM_USBH_PACKET_SPLIT_Pos)
  SSPLIT Packet.
- #define ARM_USBH_PACKET_SSPLIT_S (0x09UL << ARM_USBH_PACKET_SPLIT_Pos)
  SSPLIT Packet: Data Start.
- #define ARM_USBH_PACKET_SSPLIT_E (0x0AUL << ARM_USBH_PACKET_SPLIT_Pos)
  SSPLIT Packet: Data End.
- #define ARM_USBH_PACKET_SSPLIT_S_E (0x0BUL << ARM_USBH_PACKET_SPLIT_Pos)
  SSPLIT Packet: Data All.
- #define ARM_USBH_PACKET_CSPLIT (0x0CUL << ARM_USBH_PACKET_SPLIT_Pos)
  CSPLIT Packet.
- #define ARM_USBH_PACKET_PRE (1UL << 12)
  PRE Token.
- #define ARM_USBH_EVENT_CONNECT (1UL << 0)
  USB Device Connected to Port.
- #define ARM_USBH_EVENT_DISCONNECT (1UL << 1)
  USB Device Disconnected from Port.
- #define ARM_USBH_EVENT_OVERCURRENT (1UL << 2)
  USB Device caused Overcurrent.
- #define ARM_USBH_EVENT_RESET (1UL << 3)
  USB Reset completed.
- #define ARM_USBH_EVENT_SUSPEND (1UL << 4)
  USB Suspend occurred.
- #define ARM_USBH_EVENT_RESUME (1UL << 5)
  USB Resume occurred.
- #define ARM_USBH_EVENT_REMOTE_WAKEUP (1UL << 6)
  USB Device activated Remote Wakeup.
- #define ARM_USBH_EVENT_TRANSFER_COMPLETE (1UL << 0)
  Transfer completed.
- #define ARM_USBH_EVENT_HANDSHAKE_NAK (1UL << 1)
  NAK Handshake received.
- #define ARM_USBH_EVENT_HANDSHAKE_NYET (1UL << 2)
  NYET Handshake received.
- #define ARM_USBH_EVENT_HANDSHAKE_MDATA (1UL << 3)
  MDATA Handshake received.
- #define ARM_USBH_EVENT_HANDSHAKE_STALL (1UL << 4)
  STALL Handshake received.
- #define ARM_USBH_EVENT_HANDSHAKE_ERR (1UL << 5)
  ERR Handshake received.
- #define ARM_USBH_EVENT_BUS_ERROR (1UL << 6)
  Bus Error detected.
- #define ARM_USBH_SignalEndpointEvent_t ARM_USBH_SignalPipeEvent_t /* Legacy name */

Typedefs

- typedef volatile struct _ARM_USBH_PORT_STATE ARM_USBH_PORT_STATE
  USB Host Port State.
- typedef uint32_t ARM_USBH_PIPE_HANDLE
  USB Host Pipe Handle.
- typedef void(* ARM_USBH_SignalPortEvent_t) (uint8_t port, uint32_t event)
  Pointer to ARM_USBH_SignalPortEvent : Signal Root HUB Port Event.

Kneron Confidential

- typedef void(∗ ARM_USBH_SignalPipeEvent_t) (ARM_USBH_PIPE_HANDLE pipe_hndl, uint32_t event)

  Pointer to ARM_USBH_SignalPipeEvent : Signal Pipe Event.
- typedef struct _ARM_USBH_CAPABILITIES ARM_USBH_CAPABILITIES

  USB Host Driver Capabilities.
- typedef struct _ARM_DRIVER_USBH ARM_DRIVER_USBH

  Access structure of USB Host Driver.
- typedef void(∗ ARM_USBH_HCI_Interrupt_t) (void)

  Pointer to Interrupt Handler Routine.
- typedef struct _ARM_USBH_HCI_CAPABILITIES ARM_USBH_HCI_CAPABILITIES

  USB Host HCI (OHCI/EHCI) Driver Capabilities.
- typedef struct _ARM_DRIVER_USBH_HCI ARM_DRIVER_USBH_HCI

  Access structure of USB Host HCI (OHCI/EHCI) Driver.

Variables

- ARM_DRIVER_USBH Driver_USBH0

6.4.1 Macro Definition Documentation

6.4.1.1 ARM_USBH_API_VERSION

```
#define ARM_USBH_API_VERSION ARM_DRIVER_VERSION_MAJOR_MINOR(2,2) /* API version */
```

6.4.1.2 ARM_USBH_EP_HANDLE

```
#define ARM_USBH_EP_HANDLE ARM_USBH_PIPE_HANDLE /* Legacy name */
```

6.4.1.3 ARM_USBH_EVENT_BUS_ERROR

```
#define ARM_USBH_EVENT_BUS_ERROR (1UL << 6)
```

Bus Error detected.

6.4.1.4 ARM_USBH_EVENT_CONNECT

```
#define ARM_USBH_EVENT_CONNECT (1UL << 0)
```

USB Device Connected to Port.

Kneron Confidential

6.4.1.5 ARM_USBH_EVENT_DISCONNECT

```
#define ARM_USBH_EVENT_DISCONNECT (1UL << 1)
```

USB Device Disconnected from Port.

6.4.1.6 ARM_USBH_EVENT_HANDSHAKE_ERR

```
#define ARM_USBH_EVENT_HANDSHAKE_ERR (1UL << 5)
```

ERR Handshake received.

6.4.1.7 ARM_USBH_EVENT_HANDSHAKE_MDATA

```
#define ARM_USBH_EVENT_HANDSHAKE_MDATA (1UL << 3)
```

MDATA Handshake received.

6.4.1.8 ARM_USBH_EVENT_HANDSHAKE_NAK

```
#define ARM_USBH_EVENT_HANDSHAKE_NAK (1UL << 1)
```

NAK Handshake received.

6.4.1.9 ARM_USBH_EVENT_HANDSHAKE_NYET

```
#define ARM_USBH_EVENT_HANDSHAKE_NYET (1UL << 2)
```

NYET Handshake received.

6.4.1.10 ARM_USBH_EVENT_HANDSHAKE_STALL

```
#define ARM_USBH_EVENT_HANDSHAKE_STALL (1UL << 4)
```

STALL Handshake received.

Kneron Confidential

6.4.1.11 ARM_USBH_EVENT_OVERCURRENT

```
#define ARM_USBH_EVENT_OVERCURRENT (1UL << 2)
```

USB Device caused Overcurrent.

6.4.1.12 ARM_USBH_EVENT_REMOTE_WAKEUP

```
#define ARM_USBH_EVENT_REMOTE_WAKEUP (1UL << 6)
```

USB Device activated Remote Wakeup.

6.4.1.13 ARM_USBH_EVENT_RESET

```
#define ARM_USBH_EVENT_RESET (1UL << 3)
```

USB Reset completed.

6.4.1.14 ARM_USBH_EVENT_RESUME

```
#define ARM_USBH_EVENT_RESUME (1UL << 5)
```

USB Resume occurred.

6.4.1.15 ARM_USBH_EVENT_SUSPEND

```
#define ARM_USBH_EVENT_SUSPEND (1UL << 4)
```

USB Suspend occurred.

6.4.1.16 ARM_USBH_EVENT_TRANSFER_COMPLETE

```
#define ARM_USBH_EVENT_TRANSFER_COMPLETE (1UL << 0)
```

Transfer completed.

Kneron Confidential

6.4.1.17 ARM_USBH_PACKET_CSPLIT

```
#define ARM_USBH_PACKET_CSPLIT (0x0CUL << ARM_USBH_PACKET_SPLIT_Pos)
```

CSPLIT Packet.

6.4.1.18 ARM_USBH_PACKET_DATA0

```
#define ARM_USBH_PACKET_DATA0 (0x01UL << ARM_USBH_PACKET_DATA_Pos)
```

DATA0 PID.

6.4.1.19 ARM_USBH_PACKET_DATA1

```
#define ARM_USBH_PACKET_DATA1 (0x02UL << ARM_USBH_PACKET_DATA_Pos)
```

DATA1 PID.

6.4.1.20 ARM_USBH_PACKET_DATA_Msk

```
#define ARM_USBH_PACKET_DATA_Msk (0x0FUL << ARM_USBH_PACKET_DATA_Pos)
```

6.4.1.21 ARM_USBH_PACKET_DATA_Pos

```
#define ARM_USBH_PACKET_DATA_Pos 4
```

6.4.1.22 ARM_USBH_PACKET_IN

```
#define ARM_USBH_PACKET_IN (0x03UL << ARM_USBH_PACKET_TOKEN_Pos)
```

IN Packet.

Kneron Confidential

6.4.1.23 ARM_USBH_PACKET_OUT

```
#define ARM_USBH_PACKET_OUT (0x02UL << ARM_USBH_PACKET_TOKEN_Pos)
```

OUT Packet.

6.4.1.24 ARM_USBH_PACKET_PING

```
#define ARM_USBH_PACKET_PING (0x04UL << ARM_USBH_PACKET_TOKEN_Pos)
```

PING Packet.

6.4.1.25 ARM_USBH_PACKET_PRE

```
#define ARM_USBH_PACKET_PRE (1UL << 12)
```

PRE Token.

6.4.1.26 ARM_USBH_PACKET_SETUP

```
#define ARM_USBH_PACKET_SETUP (0x01UL << ARM_USBH_PACKET_TOKEN_Pos)
```

SETUP Packet.

6.4.1.27 ARM_USBH_PACKET_SPLIT_Msk

```
#define ARM_USBH_PACKET_SPLIT_Msk (0x0FUL << ARM_USBH_PACKET_SPLIT_Pos)
```

6.4.1.28 ARM_USBH_PACKET_SPLIT_Pos

```
#define ARM_USBH_PACKET_SPLIT_Pos 8
```

Kneron Confidential

6.4.1.29 ARM_USBH_PACKET_SSPLIT

```
#define ARM_USBH_PACKET_SSPLIT (0x08UL << ARM_USBH_PACKET_SPLIT_Pos)
```

SSPLIT Packet.

6.4.1.30 ARM_USBH_PACKET_SSPLIT_E

```
#define ARM_USBH_PACKET_SSPLIT_E (0x0AUL << ARM_USBH_PACKET_SPLIT_Pos)
```

SSPLIT Packet: Data End.

6.4.1.31 ARM_USBH_PACKET_SSPLIT_S

```
#define ARM_USBH_PACKET_SSPLIT_S (0x09UL << ARM_USBH_PACKET_SPLIT_Pos)
```

SSPLIT Packet: Data Start.

6.4.1.32 ARM_USBH_PACKET_SSPLIT_S_E

```
#define ARM_USBH_PACKET_SSPLIT_S_E (0x0BUL << ARM_USBH_PACKET_SPLIT_Pos)
```

SSPLIT Packet: Data All.

6.4.1.33 ARM_USBH_PACKET_TOKEN_Msk

```
#define ARM_USBH_PACKET_TOKEN_Msk (0x0FUL << ARM_USBH_PACKET_TOKEN_Pos)
```

6.4.1.34 ARM_USBH_PACKET_TOKEN_Pos

```
#define ARM_USBH_PACKET_TOKEN_Pos 0
```

6.4.1.35 ARM_USBH_SignalEndpointEvent_t

```
#define ARM_USBH_SignalEndpointEvent_t ARM_USBH_SignalPipeEvent_t /* Legacy name */
```

Kneron Confidential

6.4.2 Typedef Documentation

6.4.2.1 ARM_DRIVER_USBH

```
typedef struct _ARM_DRIVER_USBH ARM_DRIVER_USBH
```

Access structure of USB Host Driver.

6.4.2.2 ARM_DRIVER_USBH_HCI

```
typedef struct _ARM_DRIVER_USBH_HCI ARM_DRIVER_USBH_HCI
```

Access structure of USB Host HCI (OHCI/EHCI) Driver.

6.4.2.3 ARM_USBH_CAPABILITIES

```
typedef struct _ARM_USBH_CAPABILITIES ARM_USBH_CAPABILITIES
```

USB Host Driver Capabilities.

6.4.2.4 ARM_USBH_HCI_CAPABILITIES

```
typedef struct _ARM_USBH_HCI_CAPABILITIES ARM_USBH_HCI_CAPABILITIES
```

USB Host HCI (OHCI/EHCI) Driver Capabilities.

6.4.2.5 ARM_USBH_HCI_Interrupt_t

```
typedef void(* ARM_USBH_HCI_Interrupt_t) (void)
```

Pointer to Interrupt Handler Routine.

Kneron Confidential

6.4.2.6 ARM_USBH_PIPE_HANDLE

```
typedef uint32_t ARM_USBH_PIPE_HANDLE
```

USB Host Pipe Handle.

6.4.2.7 ARM_USBH_PORT_STATE

```
typedef volatile struct _ARM_USBH_PORT_STATE ARM_USBH_PORT_STATE
```

USB Host Port State.

6.4.2.8 ARM_USBH_SignalPipeEvent_t

```
typedef void(* ARM_USBH_SignalPipeEvent_t) (ARM_USBH_PIPE_HANDLE pipe_hndl, uint32_t event)
```

Pointer to ARM_USBH_SignalPipeEvent : Signal Pipe Event.

6.4.2.9 ARM_USBH_SignalPortEvent_t

```
typedef void(* ARM_USBH_SignalPortEvent_t) (uint8_t port, uint32_t event)
```

Pointer to ARM_USBH_SignalPortEvent : Signal Root HUB Port Event.

6.4.3 Variable Documentation

6.4.3.1 Driver_USBH0

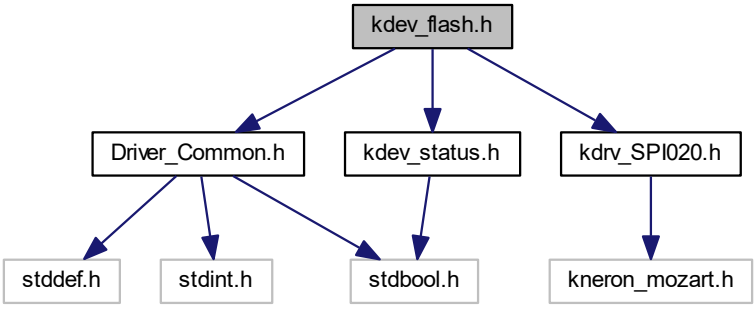
```
ARM_DRIVER_USBH Driver_USBH0
```

Kneron Confidential

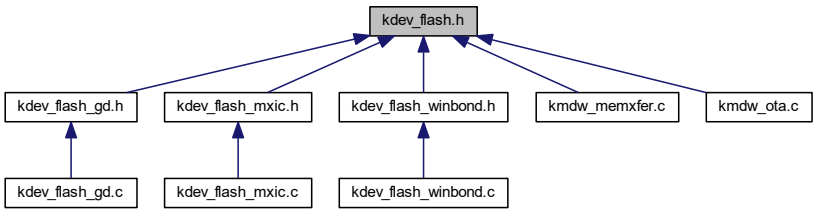
6.5 kdev_flash.h File Reference

```
#include "Driver_Common.h"
#include "kdrv_SPI020.h"
#include "kdev_status.h"
```

Include dependency graph for kdev_flash.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct kdev_flash_sector_t
  Flash Sector index struct.
- struct kdev_flash_info_t
  Flash information struct.
- struct kdev_flash_status_t
  Flash Status struct.

Macros

- #define SPI020_SECTOR_SIZE 4096
- #define SPI020_BLOCK_64SIZE 65536

Kneron Confidential

Functions

- uint32_t kdev_flash_probe (spi_flash_t *flash)
- kdev_status_t kdev_flash_initialize (void)

  Initialize spi flash interface include hardware setting, get flash information and set to 4byte address if flash size is bigger than 16Mbytes.
- kdev_status_t kdev_flash_uninitialize (void)

  Uinitialize the spi flash interface.
- kdev_status_t kdev_flash_power_control (ARM_POWER_STATE state)

  Power handling for spi flasg.
- kdev_status_t kdev_flash_readdata (uint32_t addr, void *data, uint32_t cnt)

  Read data from specific index of spi flash.
- kdev_status_t kdev_flash_programdata (uint32_t addr, const void *data, uint32_t cnt)

  Program data to specific index in spi flash.
- kdev_status_t kdev_flash_erase_sector (uint32_t addr)

  Erase Flash by Sector(4k bytes).
- kdev_status_t kdev_flash_erase_multi_sector (uint16_t start_addr, uint16_t end_addr)

  Erase multiple Flash Sectors(continuously).
- kdev_status_t kdev_flash_erase_chip (void)

  Erase whole Flash at once. Optional function for faster full chip erase.
- kdev_flash_status_t kdev_flash_get_status (void)

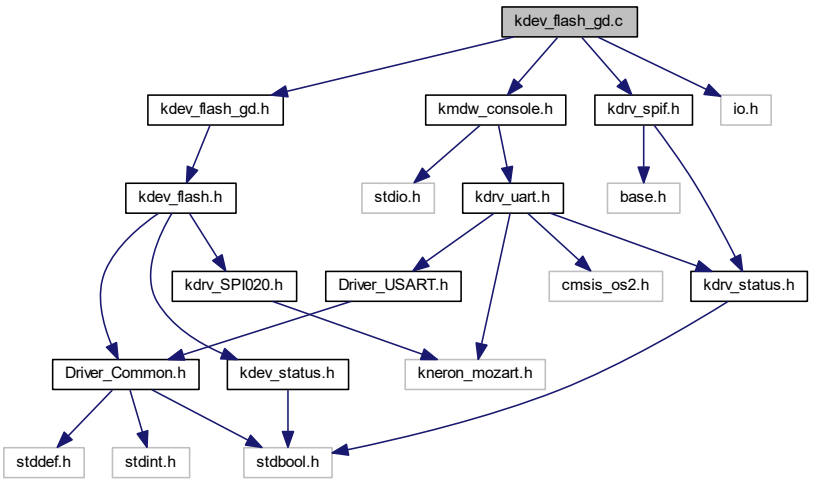
  Get Flash status.
- kdev_flash_info_t * kdev_flash_get_info (void)

  Get Flash information.

6.6 kdev_flash_gd.c File Reference

```
#include "kdev_flash_gd.h"
#include "kdrv_spif.h"
#include "kmdw_console.h"
#include "io.h"
```

Include dependency graph for kdev_flash_gd.c:



Kneron Confidential

Functions

- void kdev_flash_4Bytes_ctrl (uint8_t enable)
- void kdev_flash_write_control (uint8_t enable)
- void kdev_flash_64kErase (uint32_t offset)
- void kdev_flash_32kErase (uint32_t offset)
- void kdev_flash_4kErase (uint32_t offset)
- uint32_t kdev_flash_probe (spi_flash_t *flash)
- void kdev_flash_read_flash_id (void)
- void kdev_flash_read_status (void)
- kdev_status_t kdev_flash_initialize (void)

  Initialize spi flash interface include hardware setting, get flash information and set to 4byte address if flash size is bigger than 16Mbytes.
- kdev_status_t kdev_flash_uninitialize (void)

  Uinitialize the spi flash interface.
- kdev_status_t kdev_flash_power_control (ARM_POWER_STATE state)

  Power handling for spi flasg.
- void kdev_flash_read (uint8_t type, uint32_t offset, uint32_t len, void *buf)
- void kdev_flash_dma_read_stop (void)
- void kdev_flash_dma_write_stop (void)
- uint8_t kdev_flash_r_state_OpCode_35 (void)
- void kdev_flash_write (uint8_t type, uint32_t offset, uint32_t len, void *buf, uint32_t buf_offset)
- kdev_status_t kdev_flash_readdata (uint32_t addr, void *data, uint32_t cnt)

  Read data from specific index of spi flash.
- kdev_status_t kdev_flash_programdata (uint32_t addr, const void *data, uint32_t cnt)

  Program data to specific index in spi flash.
- kdev_status_t kdev_flash_erase_sector (uint32_t addr)

  Erase Flash by Sector(4k bytes).
- kdev_status_t kdev_flash_erase_multi_sector (uint16_t start_addr, uint16_t end_addr)

  Erase multiple Flash Sectors(continuously).
- kdev_status_t kdev_flash_erase_chip (void)

  Erase whole Flash at once. Optional function for faster full chip erase.
- kdev_flash_status_t kdev_flash_get_status (void)

  Get Flash status.
- kdev_flash_info_t * kdev_flash_get_info (void)

  Get Flash information.

Variables

- spi_flash_t flash_info

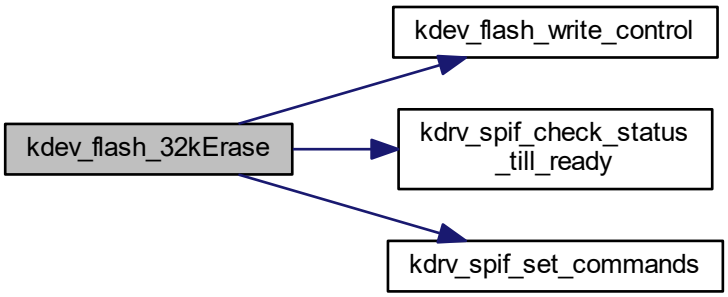
6.6.1 Function Documentation

Kneron Confidential

6.6.1.1 kdev_flash_32kErase()

```
void kdev_flash_32kErase (
            uint32_t offset )
```

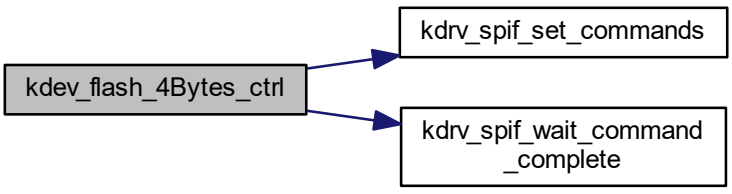
Here is the call graph for this function:



6.6.1.2 kdev_flash_4Bytes_ctrl()

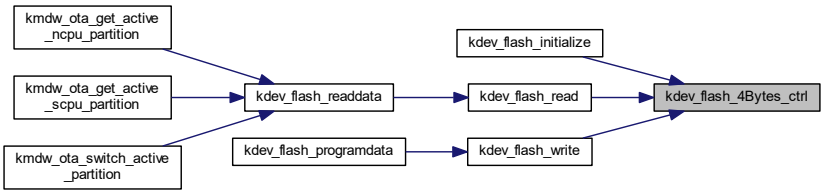
```
void kdev_flash_4Bytes_ctrl (
            uint8_t enable )
```

Here is the call graph for this function:



Kneron Confidential

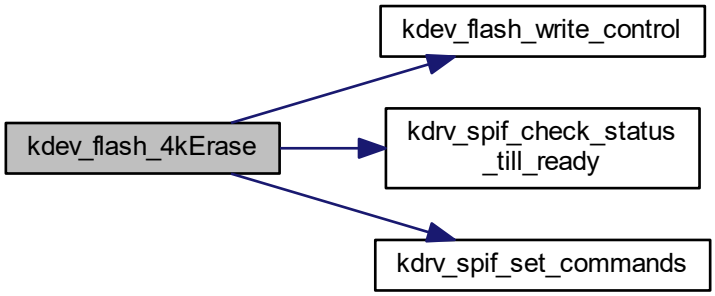
Here is the caller graph for this function:



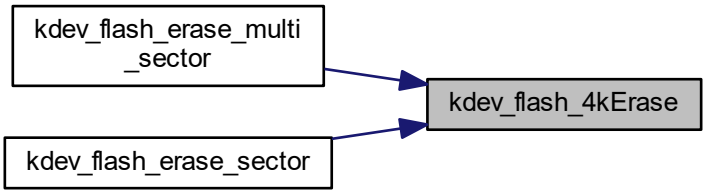
6.6.1.3 kdev_flash_4kErase()

```
void kdev_flash_4kErase (
            uint32_t offset )
```

Here is the call graph for this function:



Here is the caller graph for this function:



Kneron Confidential

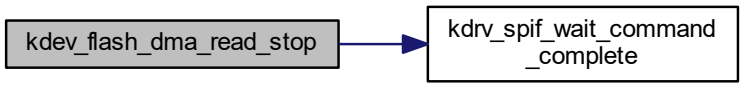
6.6.1.4 kdev_flash_64kErase()

```
void kdev_flash_64kErase (
            uint32_t offset )
```

6.6.1.5 kdev_flash_dma_read_stop()

```
void kdev_flash_dma_read_stop (
            void  )
```

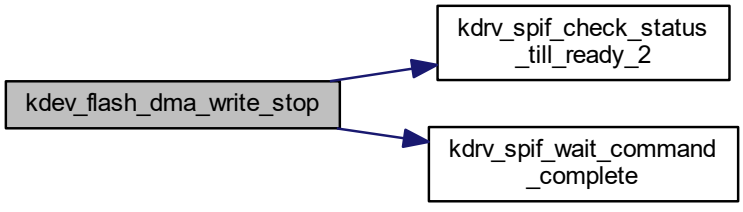
Here is the call graph for this function:



6.6.1.6 kdev_flash_dma_write_stop()

```
void kdev_flash_dma_write_stop (
            void  )
```

Here is the call graph for this function:

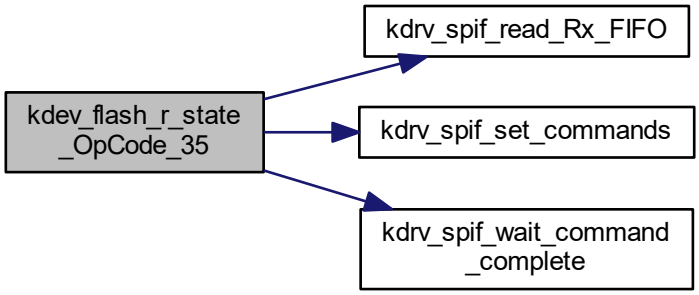


Kneron Confidential

6.6.1.7 kdev_flash_r_state_OpCode_35()

```
uint8_t kdev_flash_r_state_OpCode_35 (
            void  )
```

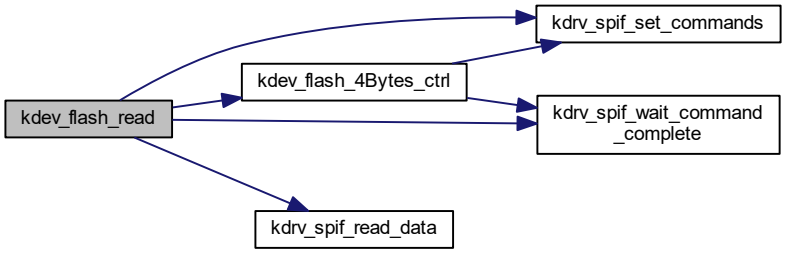
Here is the call graph for this function:



6.6.1.8 kdev_flash_read()

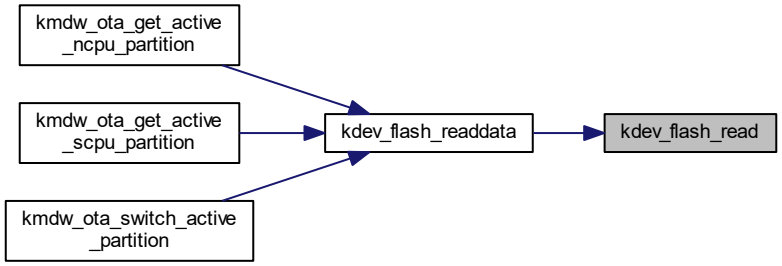
```
void kdev_flash_read (
            uint8_t type,
            uint32_t offset,
            uint32_t len,
            void * buf )
```

Here is the call graph for this function:



Kneron Confidential

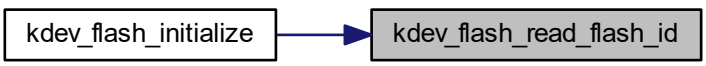
Here is the caller graph for this function:



6.6.1.9 kdev_flash_read_flash_id()

```
void kdev_flash_read_flash_id (
            void  )
```

Here is the caller graph for this function:

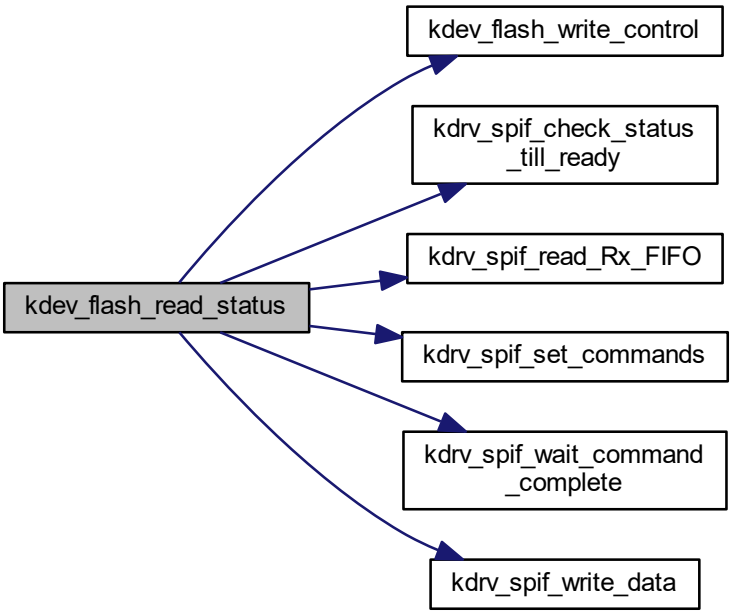


6.6.1.10 kdev_flash_read_status()

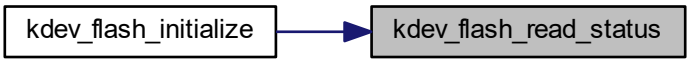
```
void kdev_flash_read_status (
            void  )
```

Kneron Confidential

Here is the call graph for this function:



Here is the caller graph for this function:

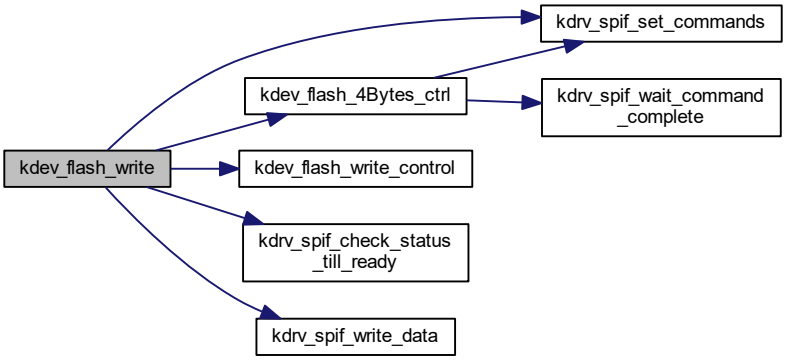


Kneron Confidential

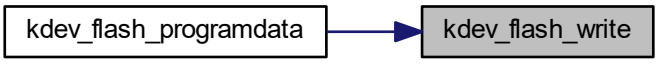
6.6.1.11 kdev_flash_write()

```
void kdev_flash_write (
            uint8_t type,
            uint32_t offset,
            uint32_t len,
            void * buf,
            uint32_t buf_offset )
```

Here is the call graph for this function:



Here is the caller graph for this function:

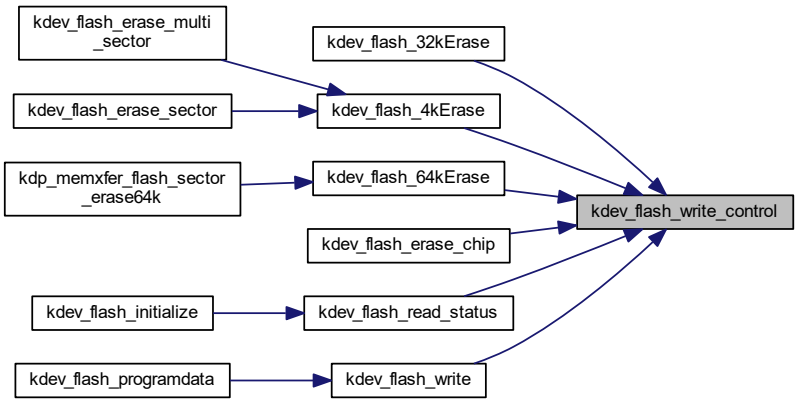


Kneron Confidential

6.6.1.12 kdev_flash_write_control()

```
void kdev_flash_write_control (
            uint8_t enable )
```

Here is the caller graph for this function:



6.6.2 Variable Documentation

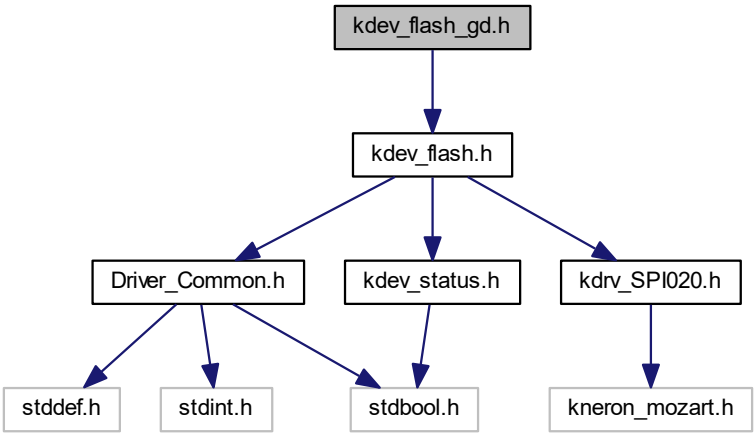
6.6.2.1 flash_info

```
spi_flash_t flash_info
```

6.7 kdev_flash_gd.h File Reference

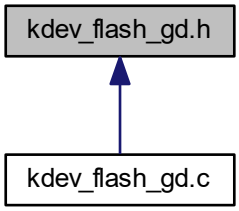
```
#include "kdev_flash.h"
```

Include dependency graph for kdev_flash_gd.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Macros

- #define FLASH_WB_DEV 0xEF
- #define FLASH_GD_DEV 0xC8
- #define FLASH_MXIC_DEV 0xC2
- #define FLASH_Micron_DEV 0x20
- #define FLASH_ZBIT_DEV 0x5E
- #define FLASH_IS_WB_DEV(x) ((x>>16)&0xFF == FLASH_WB_DEV)
- #define FLASH_IS_MXIC_DEV(x) ((x>>16)&0xFF == FLASH_MXIC_DEV)
- #define FLASH_IS_GD_DEV(x) ((x>>16)&0xFF == FLASH_GD_DEV)
- #define WB_W25Q01JV_ID 0x2140
- #define WB_W25Q512JV_ID 0x2040
- #define WB_W25L256JV_ID 0x1940
- #define WB_W25Q128BVFG_ID 0x1840
- #define WB_W25Q64CV_ID_9F 0x1740
- #define WB_W25Q32CV_ID_9F 0x1640
- #define WB_W25P80_ID_9F 0x1420
- #define WB_W25P16_ID_9F 0x1520
- #define WB_W25P32_ID_9F 0x1620
- #define WB_W25P16_ID_90 0x14
- #define MX_MX25L51245G 0x1A20
- #define MX_MX25L25645G 0x1920
- #define MX_MX25L12845EM1 0x1820
- #define MX_MX25L6405D 0x1720
- #define GD_GD25Q64CSIG_ID 0x1740
- #define GD_GD25Q127CSIG_ID 0x1840
- #define GD_GD25Q256_ID 0x1940
- #define GD_GD25Q256_ID_9F 0x1940
- #define GD25Q512MC_ID_9F 0x2040
- #define INVALID_CHIP_ID 0xFFFF
- #define INVALID_MANU_ID 0xFF
- #define FLASH_SIZE_32MB_ID 0x19
- #define FLASH_SIZE_16MB_ID 0x18
- #define FLASH_SIZE_8MB_ID 0x17
- #define FLASH_SIZE_4MB_ID 0x16
- #define FLASH_SIZE_2MB_ID 0x15
- #define FLASH_SIZE_1MB_ID 0x14
- #define FLASH_SIZE_512MB_ID 0x20
- #define FLASH_SIZE_SHIFT (FLASH_SIZE_512MB_ID - 0x1A)
- #define FLASH_3BYTE_ADDR_MAX 0x4000

Kneron Confidential

6.7.1 Macro Definition Documentation

6.7.1.1 FLASH_3BYTE_ADDR_MAX

```
#define FLASH_3BYTE_ADDR_MAX 0x4000
```

6.7.1.2 FLASH_GD_DEV

```
#define FLASH_GD_DEV 0xC8
```

6.7.1.3 FLASH_IS_GD_DEV

```
#define FLASH_IS_GD_DEV(
            x ) ((x>>16)&0xFF == FLASH_GD_DEV)
```

6.7.1.4 FLASH_IS_MXIC_DEV

```
#define FLASH_IS_MXIC_DEV(
            x ) ((x>>16)&0xFF == FLASH_MXIC_DEV)
```

6.7.1.5 FLASH_IS_WB_DEV

```
#define FLASH_IS_WB_DEV(
            x ) ((x>>16)&0xFF == FLASH_WB_DEV)
```

6.7.1.6 FLASH_Micron_DEV

```
#define FLASH_Micron_DEV 0x20
```

Kneron Confidential

6.7.1.7 FLASH_MXIC_DEV

```
#define FLASH_MXIC_DEV 0xC2
```

6.7.1.8 FLASH_SIZE_16MB_ID

```
#define FLASH_SIZE_16MB_ID 0x18
```

6.7.1.9 FLASH_SIZE_1MB_ID

```
#define FLASH_SIZE_1MB_ID 0x14
```

6.7.1.10 FLASH_SIZE_2MB_ID

```
#define FLASH_SIZE_2MB_ID 0x15
```

6.7.1.11 FLASH_SIZE_32MB_ID

```
#define FLASH_SIZE_32MB_ID 0x19
```

6.7.1.12 FLASH_SIZE_4MB_ID

```
#define FLASH_SIZE_4MB_ID 0x16
```

6.7.1.13 FLASH_SIZE_512MB_ID

```
#define FLASH_SIZE_512MB_ID 0x20
```

6.7.1.14 FLASH_SIZE_8MB_ID

```
#define FLASH_SIZE_8MB_ID 0x17
```

Kneron Confidential

6.7.1.15 FLASH_SIZE_SHIFT

```
#define FLASH_SIZE_SHIFT (FLASH_SIZE_512MB_ID - 0x1A)
```

6.7.1.16 FLASH_WB_DEV

```
#define FLASH_WB_DEV 0xEF
```

6.7.1.17 FLASH_ZBIT_DEV

```
#define FLASH_ZBIT_DEV 0x5E
```

6.7.1.18 GD25Q512MC_ID_9F

```
#define GD25Q512MC_ID_9F 0x2040
```

6.7.1.19 GD_GD25Q127CSIG_ID

```
#define GD_GD25Q127CSIG_ID 0x1840
```

6.7.1.20 GD_GD25Q256_ID

```
#define GD_GD25Q256_ID 0x1940
```

6.7.1.21 GD_GD25Q256_ID_9F

```
#define GD_GD25Q256_ID_9F 0x1940
```

6.7.1.22 GD_GD25Q64CSIG_ID

```
#define GD_GD25Q64CSIG_ID 0x1740
```

Kneron Confidential

6.7.1.23 INVALID_CHIP_ID

```
#define INVALID_CHIP_ID 0xFFFF
```

6.7.1.24 INVALID_MANU_ID

```
#define INVALID_MANU_ID 0xFF
```

6.7.1.25 MX_MX25L12845EM1

```
#define MX_MX25L12845EM1 0x1820
```

6.7.1.26 MX_MX25L25645G

```
#define MX_MX25L25645G 0x1920
```

6.7.1.27 MX_MX25L51245G

```
#define MX_MX25L51245G 0x1A20
```

6.7.1.28 MX_MX25L6405D

```
#define MX_MX25L6405D 0x1720
```

6.7.1.29 WB_W25L256JV_ID

```
#define WB_W25L256JV_ID 0x1940
```

6.7.1.30 WB_W25P16_ID_90

```
#define WB_W25P16_ID_90 0x14
```

Kneron Confidential

6.7.1.31 WB_W25P16_ID_9F

```
#define WB_W25P16_ID_9F 0x1520
```

6.7.1.32 WB_W25P32_ID_9F

```
#define WB_W25P32_ID_9F 0x1620
```

6.7.1.33 WB_W25P80_ID_9F

```
#define WB_W25P80_ID_9F 0x1420
```

6.7.1.34 WB_W25Q01JV_ID

```
#define WB_W25Q01JV_ID 0x2140
```

6.7.1.35 WB_W25Q128BVFG_ID

```
#define WB_W25Q128BVFG_ID 0x1840
```

6.7.1.36 WB_W25Q32CV_ID_9F

```
#define WB_W25Q32CV_ID_9F 0x1640
```

6.7.1.37 WB_W25Q512JV_ID

```
#define WB_W25Q512JV_ID 0x2040
```

6.7.1.38 WB_W25Q64CV_ID_9F

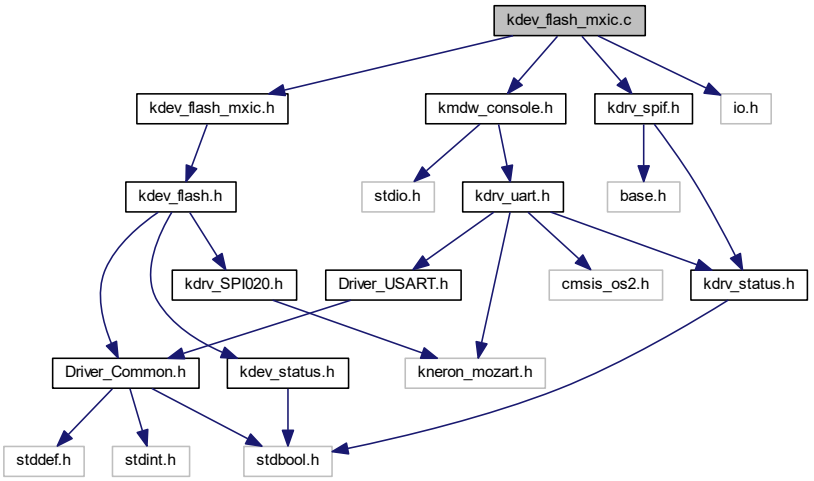
```
#define WB_W25Q64CV_ID_9F 0x1740
```

Kneron Confidential

6.8 kdev_flash_mxic.c File Reference

```
#include "kdev_flash_mxic.h"
#include "kdrv_spif.h"
#include "kmdw_console.h"
#include "io.h"
```

Include dependency graph for kdev_flash_mxic.c:



Macros

- #define mxic_msg(fmt, ...)

Functions

- void kdev_flash_4Bytes_ctrl (uint8_t enable)
- void kdev_flash_write_control (uint8_t enable)
- void kdev_flash_64kErase (uint32_t offset)
- void kdev_flash_32kErase (uint32_t offset)
- void kdev_flash_4kErase (uint32_t offset)
- uint32_t kdev_flash_probe (spi_flash_t *flash)
- void kdev_flash_read_flash_id (void)
- void kdev_flash_read_status (void)
- kdev_status_t kdev_flash_initialize (void)

  Initialize spi flash interface include hardware setting, get flash information and set to 4byte address if flash size is bigger than 16Mbytes.
- kdev_status_t kdev_flash_uninitialize (void)

  Uinitialize the spi flash interface.
- kdev_status_t kdev_flash_power_control (ARM_POWER_STATE state)

  Power handling for spi flasg.
- void kdev_flash_read (uint8_t type, uint32_t offset, uint32_t len, void *buf)
- void kdev_flash_dma_read_stop (void)
- void kdev_flash_dma_write_stop (void)

Kneron Confidential

- uint8_t kdev_flash_r_state_OpCode_35 (void)
- void kdev_flash_write (uint8_t type, uint32_t offset, uint32_t len, void *buf, uint32_t buf_offset)
- kdev_status_t kdev_flash_readdata (uint32_t addr, void *data, uint32_t cnt)

  Read data from specific index of spi flash.
- kdev_status_t kdev_flash_programdata (uint32_t addr, const void *data, uint32_t cnt)

  Program data to specific index in spi flash.
- kdev_status_t kdev_flash_erase_sector (uint32_t addr)

  Erase Flash by Sector(4k bytes).
- kdev_status_t kdev_flash_erase_multi_sector (uint16_t start_addr, uint16_t end_addr)

  Erase multiple Flash Sectors(continuously).
- kdev_status_t kdev_flash_erase_chip (void)

  Erase whole Flash at once. Optional function for faster full chip erase.
- kdev_flash_status_t kdev_flash_get_status (void)

  Get Flash status.
- kdev_flash_info_t * kdev_flash_get_info (void)

  Get Flash information.

Variables

- spi_flash_t flash_info

6.8.1 Macro Definition Documentation

6.8.1.1 mxic_msg

```
#define mxic_msg(
                fmt,
                ...  )
```

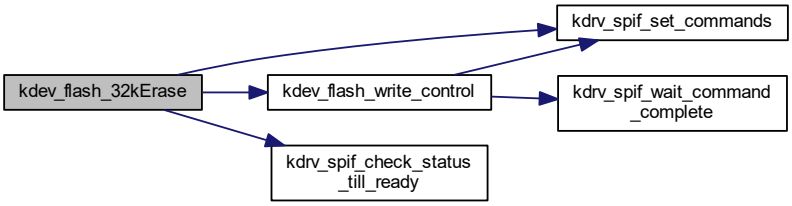
6.8.2 Function Documentation

Kneron Confidential

6.8.2.1 kdev_flash_32kErase()

```
void kdev_flash_32kErase (
            uint32_t offset )
```

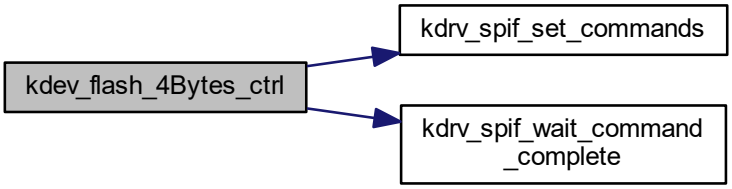
Here is the call graph for this function:



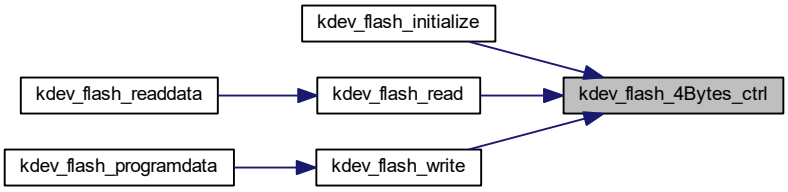
6.8.2.2 kdev_flash_4Bytes_ctrl()

```
void kdev_flash_4Bytes_ctrl (
            uint8_t enable )
```

Here is the call graph for this function:



Here is the caller graph for this function:

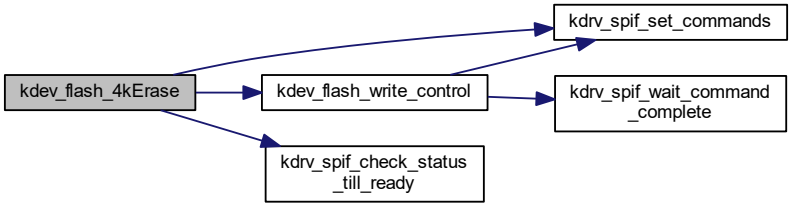


Kneron Confidential

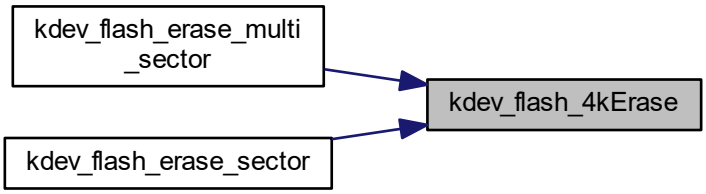
6.8.2.3 kdev_flash_4kErase()

```
void kdev_flash_4kErase (
            uint32_t offset )
```

Here is the call graph for this function:



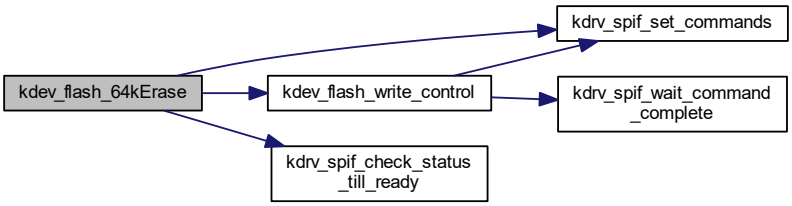
Here is the caller graph for this function:



6.8.2.4 kdev_flash_64kErase()

```
void kdev_flash_64kErase (
            uint32_t offset )
```

Here is the call graph for this function:

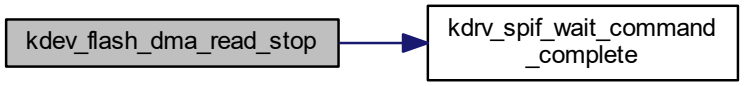


Kneron Confidential

6.8.2.5 kdev_flash_dma_read_stop()

```
void kdev_flash_dma_read_stop (
            void  )
```

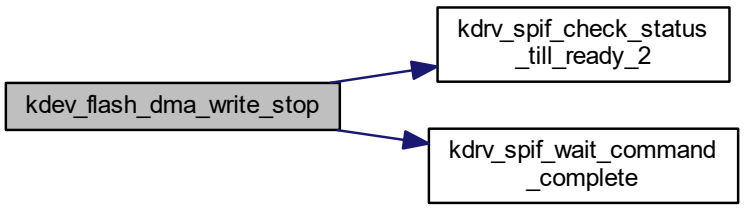
Here is the call graph for this function:



6.8.2.6 kdev_flash_dma_write_stop()

```
void kdev_flash_dma_write_stop (
            void  )
```

Here is the call graph for this function:

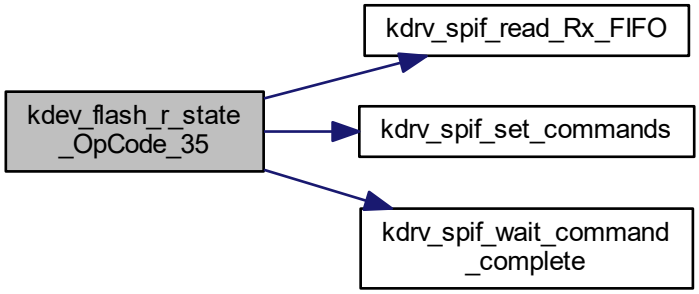


6.8.2.7 kdev_flash_r_state_OpCode_35()

```
uint8_t kdev_flash_r_state_OpCode_35 (
            void  )
```

Kneron Confidential

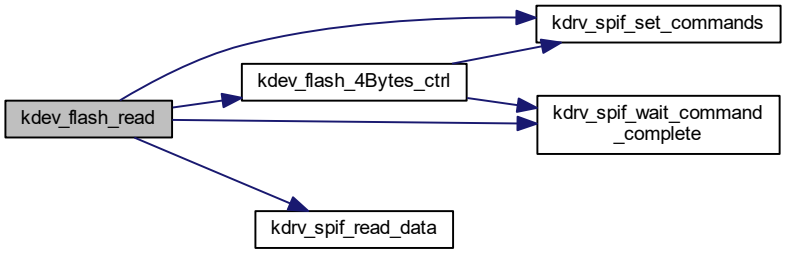
Here is the call graph for this function:



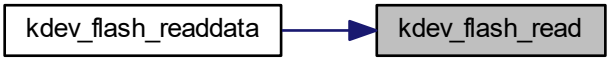
6.8.2.8 kdev_flash_read()

```
void kdev_flash_read (
            uint8_t type,
            uint32_t offset,
            uint32_t len,
            void * buf )
```

Here is the call graph for this function:



Here is the caller graph for this function:

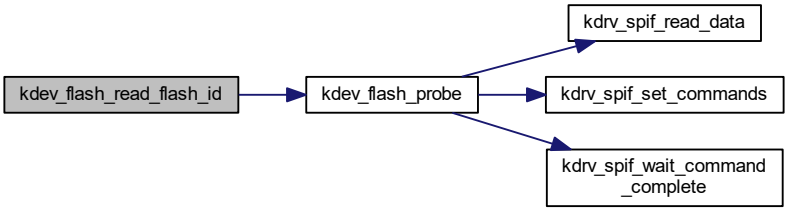


Kneron Confidential

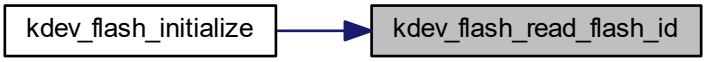
6.8.2.9 kdev_flash_read_flash_id()

```
void kdev_flash_read_flash_id (
            void  )
```

Here is the call graph for this function:



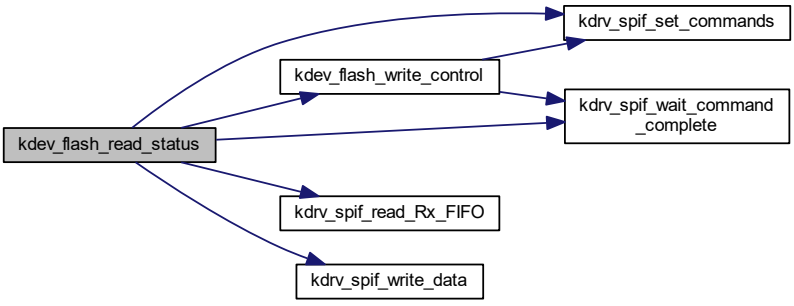
Here is the caller graph for this function:



6.8.2.10 kdev_flash_read_status()

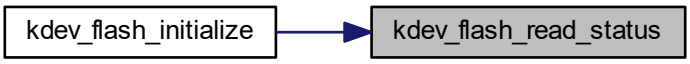
```
void kdev_flash_read_status (
            void  )
```

Here is the call graph for this function:



Kneron Confidential

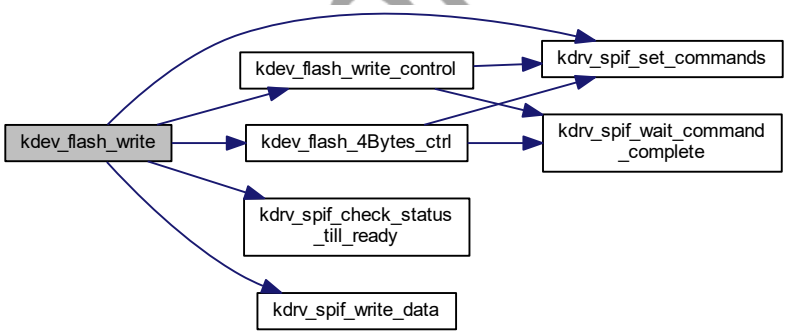
Here is the caller graph for this function:



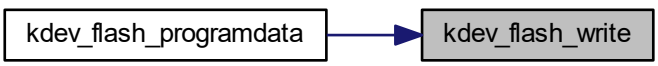
6.8.2.11 kdev_flash_write()

```
void kdev_flash_write (
            uint8_t type,
            uint32_t offset,
            uint32_t len,
            void * buf,
            uint32_t buf_offset )
```

Here is the call graph for this function:



Here is the caller graph for this function:

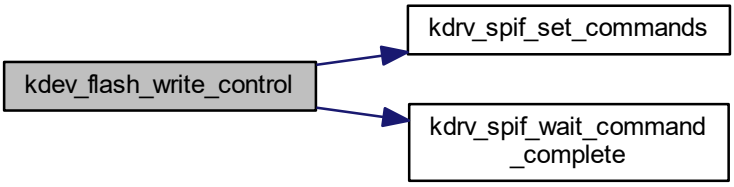


Kneron Confidential

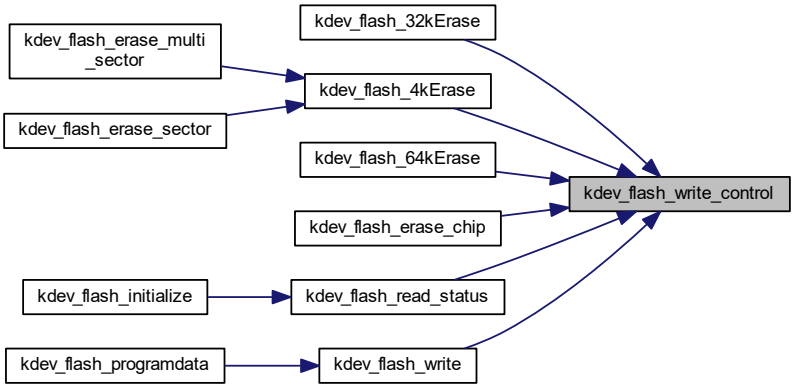
6.8.2.12 kdev_flash_write_control()

```
void kdev_flash_write_control (
            uint8_t enable )
```

Here is the call graph for this function:



Here is the caller graph for this function:



Kneron Confidential

6.8.3 Variable Documentation

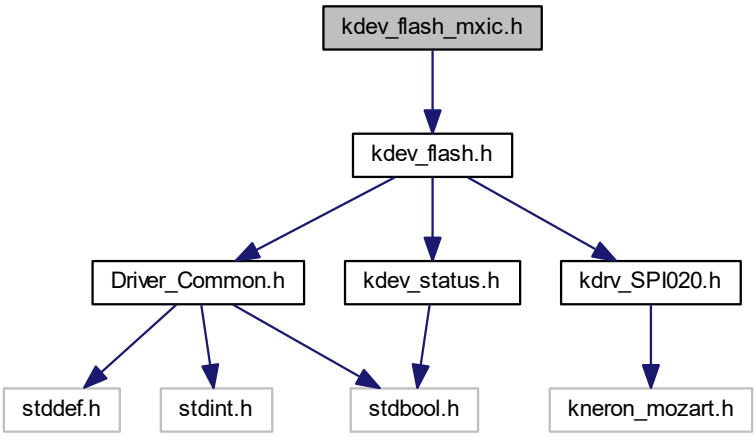
6.8.3.1 flash_info

```
spi_flash_t flash_info
```

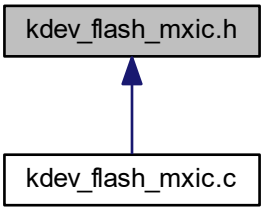
6.9 kdev_flash_mxic.h File Reference

```
#include "kdev_flash.h"
```

Include dependency graph for kdev_flash_mxic.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define FLASH_WB_DEV 0xEF
- #define FLASH_GD_DEV 0xC8
- #define FLASH_MXIC_DEV 0xC2
- #define FLASH_Micron_DEV 0x20
- #define FLASH_ZBIT_DEV 0x5E
- #define FLASH_IS_WB_DEV(x) ((x>>16)&0xFF == FLASH_WB_DEV)
- #define FLASH_IS_MXIC_DEV(x) ((x>>16)&0xFF == FLASH_MXIC_DEV)
- #define FLASH_IS_GD_DEV(x) ((x>>16)&0xFF == FLASH_GD_DEV)
- #define WB_W25Q01JV_ID 0x2140

- #define WB_W25Q512JV_ID 0x2040
- #define WB_W25L256JV_ID 0x1940
- #define WB_W25Q128BVFG_ID 0x1840
- #define WB_W25Q64CV_ID_9F 0x1740
- #define WB_W25Q32CV_ID_9F 0x1640
- #define WB_W25P80_ID_9F 0x1420
- #define WB_W25P16_ID_9F 0x1520
- #define WB_W25P32_ID_9F 0x1620
- #define WB_W25P16_ID_90 0x14
- #define MX_MX25L51245G 0x1A20
- #define MX_MX25L25645G 0x1920
- #define MX_MX25L12845EM1 0x1820
- #define MX_MX25L6405D 0x1720
- #define GD_GD25Q64CSIG_ID 0x1740
- #define GD_GD25Q127CSIG_ID 0x1840
- #define GD_GD25Q256_ID 0x1940
- #define GD_GD25Q256_ID_9F 0x1940
- #define GD25Q512MC_ID_9F 0x2040
- #define INVALID_CHIP_ID 0xFFFF
- #define INVALID_MANU_ID 0xFF
- #define FLASH_SIZE_32MB_ID 0x19
- #define FLASH_SIZE_16MB_ID 0x18
- #define FLASH_SIZE_8MB_ID 0x17
- #define FLASH_SIZE_4MB_ID 0x16
- #define FLASH_SIZE_2MB_ID 0x15
- #define FLASH_SIZE_1MB_ID 0x14
- #define FLASH_SIZE_512MB_ID 0x20
- #define FLASH_SIZE_SHIFT (FLASH_SIZE_512MB_ID - 0x1A)
- #define FLASH_3BYTE_ADDR_MAX 0x4000

6.9.1 Macro Definition Documentation

6.9.1.1 FLASH_3BYTE_ADDR_MAX

```
#define FLASH_3BYTE_ADDR_MAX 0x4000
```

6.9.1.2 FLASH_GD_DEV

```
#define FLASH_GD_DEV 0xC8
```

6.9.1.3 FLASH_IS_GD_DEV

```
#define FLASH_IS_GD_DEV(
             x ) ((x>>16)&0xFF == FLASH_GD_DEV)
```

Kneron Confidential

6.9.1.4 FLASH_IS_MXIC_DEV

```
#define FLASH_IS_MXIC_DEV(
              x ) ((x>>16)&0xFF == FLASH_MXIC_DEV)
```

6.9.1.5 FLASH_IS_WB_DEV

```
#define FLASH_IS_WB_DEV(
              x ) ((x>>16)&0xFF == FLASH_WB_DEV)
```

6.9.1.6 FLASH_Micron_DEV

```
#define FLASH_Micron_DEV 0x20
```

6.9.1.7 FLASH_MXIC_DEV

```
#define FLASH_MXIC_DEV 0xC2
```

6.9.1.8 FLASH_SIZE_16MB_ID

```
#define FLASH_SIZE_16MB_ID 0x18
```

6.9.1.9 FLASH_SIZE_1MB_ID

```
#define FLASH_SIZE_1MB_ID 0x14
```

6.9.1.10 FLASH_SIZE_2MB_ID

```
#define FLASH_SIZE_2MB_ID 0x15
```

Kneron Confidential

6.9.1.11 FLASH_SIZE_32MB_ID

```
#define FLASH_SIZE_32MB_ID 0x19
```

6.9.1.12 FLASH_SIZE_4MB_ID

```
#define FLASH_SIZE_4MB_ID 0x16
```

6.9.1.13 FLASH_SIZE_512MB_ID

```
#define FLASH_SIZE_512MB_ID 0x20
```

6.9.1.14 FLASH_SIZE_8MB_ID

```
#define FLASH_SIZE_8MB_ID 0x17
```

6.9.1.15 FLASH_SIZE_SHIFT

```
#define FLASH_SIZE_SHIFT (FLASH_SIZE_512MB_ID - 0x1A)
```

6.9.1.16 FLASH_WB_DEV

```
#define FLASH_WB_DEV 0xEF
```

6.9.1.17 FLASH_ZBIT_DEV

```
#define FLASH_ZBIT_DEV 0x5E
```

6.9.1.18 GD25Q512MC_ID_9F

```
#define GD25Q512MC_ID_9F 0x2040
```

Kneron Confidential

6.9.1.19 GD_GD25Q127CSIG_ID

```
#define GD_GD25Q127CSIG_ID 0x1840
```

6.9.1.20 GD_GD25Q256_ID

```
#define GD_GD25Q256_ID 0x1940
```

6.9.1.21 GD_GD25Q256_ID_9F

```
#define GD_GD25Q256_ID_9F 0x1940
```

6.9.1.22 GD_GD25Q64CSIG_ID

```
#define GD_GD25Q64CSIG_ID 0x1740
```

6.9.1.23 INVALID_CHIP_ID

```
#define INVALID_CHIP_ID 0xFFFF
```

6.9.1.24 INVALID_MANU_ID

```
#define INVALID_MANU_ID 0xFF
```

6.9.1.25 MX_MX25L12845EM1

```
#define MX_MX25L12845EM1 0x1820
```

6.9.1.26 MX_MX25L25645G

```
#define MX_MX25L25645G 0x1920
```

Kneron Confidential

6.9.1.27 MX_MX25L51245G

```
#define MX_MX25L51245G 0x1A20
```

6.9.1.28 MX_MX25L6405D

```
#define MX_MX25L6405D 0x1720
```

6.9.1.29 WB_W25L256JV_ID

```
#define WB_W25L256JV_ID 0x1940
```

6.9.1.30 WB_W25P16_ID_90

```
#define WB_W25P16_ID_90 0x14
```

6.9.1.31 WB_W25P16_ID_9F

```
#define WB_W25P16_ID_9F 0x1520
```

6.9.1.32 WB_W25P32_ID_9F

```
#define WB_W25P32_ID_9F 0x1620
```

6.9.1.33 WB_W25P80_ID_9F

```
#define WB_W25P80_ID_9F 0x1420
```

6.9.1.34 WB_W25Q01JV_ID

```
#define WB_W25Q01JV_ID 0x2140
```

Kneron Confidential

6.9.1.35 WB_W25Q128BVFG_ID

```
#define WB_W25Q128BVFG_ID 0x1840
```

6.9.1.36 WB_W25Q32CV_ID_9F

```
#define WB_W25Q32CV_ID_9F 0x1640
```

6.9.1.37 WB_W25Q512JV_ID

```
#define WB_W25Q512JV_ID 0x2040
```

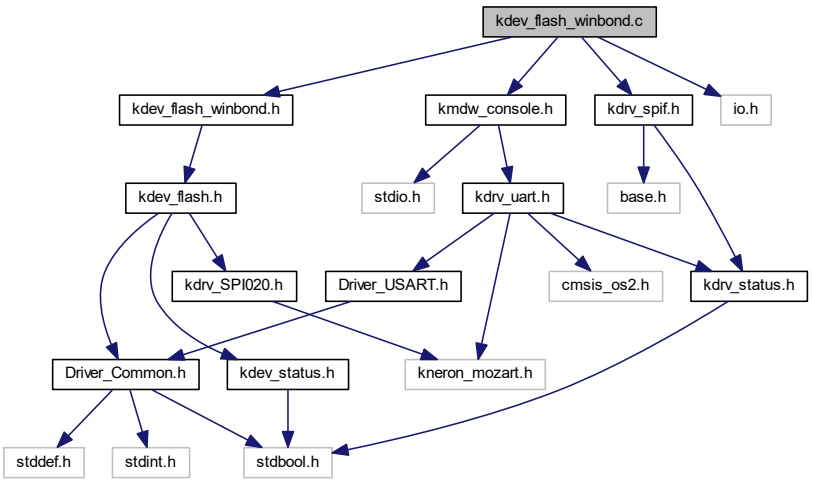
6.9.1.38 WB_W25Q64CV_ID_9F

```
#define WB_W25Q64CV_ID_9F 0x1740
```

6.10 kdev_flash_winbond.c File Reference

```
#include "kdev_flash_winbond.h"
#include "kdrv_spif.h"
#include "kmdw_console.h"
#include "io.h"
```

Include dependency graph for kdev_flash_winbond.c:



Kneron Confidential

Functions

- void kdev_flash_4Bytes_ctrl (uint8_t enable)
- void kdev_flash_write_control (uint8_t enable)
- void kdev_flash_64kErase (uint32_t offset)
- void kdev_flash_32kErase (uint32_t offset)
- void kdev_flash_4kErase (uint32_t offset)
- uint32_t kdev_flash_probe (spi_flash_t *flash)
- void kdev_flash_read_flash_id (void)
- void kdev_flash_read_status (void)
- kdev_status_t kdev_flash_initialize (void)

  Initialize spi flash interface include hardware setting, get flash information and set to 4byte address if flash size is bigger than 16Mbytes.
- kdev_status_t kdev_flash_uninitialize (void)

  Uinitialize the spi flash interface.
- kdev_status_t kdev_flash_power_control (ARM_POWER_STATE state)

  Power handling for spi flasg.
- void kdev_flash_read (uint8_t type, uint32_t offset, uint32_t len, void *buf)
- void kdev_flash_dma_read_stop (void)
- void kdev_flash_dma_write_stop (void)
- uint8_t kdev_flash_r_state_OpCode_35 (void)
- void kdev_flash_write (uint8_t type, uint32_t offset, uint32_t len, void *buf, uint32_t buf_offset)
- kdev_status_t kdev_flash_readdata (uint32_t addr, void *data, uint32_t cnt)

  Read data from specific index of spi flash.
- kdev_status_t kdev_flash_programdata (uint32_t addr, const void *data, uint32_t cnt)

  Program data to specific index in spi flash.
- kdev_status_t kdev_flash_erase_sector (uint32_t addr)

  Erase Flash by Sector(4k bytes).
- kdev_status_t kdev_flash_erase_multi_sector (uint16_t start_addr, uint16_t end_addr)

  Erase multiple Flash Sectors(continuously).
- kdev_status_t kdev_flash_erase_chip (void)

  Erase whole Flash at once. Optional function for faster full chip erase.
- kdev_flash_status_t kdev_flash_get_status (void)

  Get Flash status.
- kdev_flash_info_t * kdev_flash_get_info (void)

  Get Flash information.

Variables

- spi_flash_t flash_info

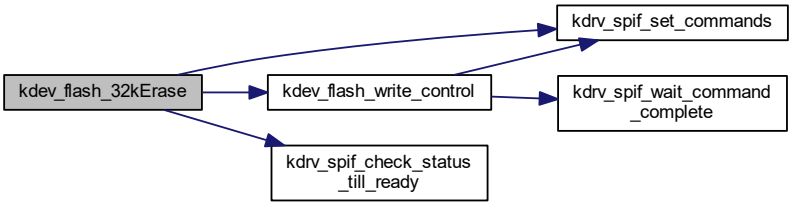
6.10.1 Function Documentation

Kneron Confidential

6.10.1.1 kdev_flash_32kErase()

```
void kdev_flash_32kErase (
            uint32_t offset )
```

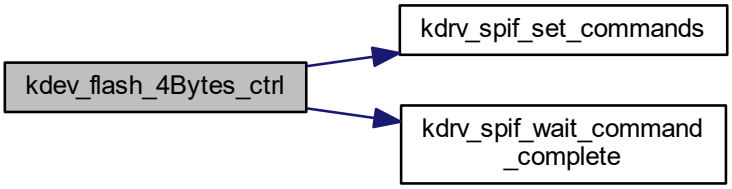
Here is the call graph for this function:



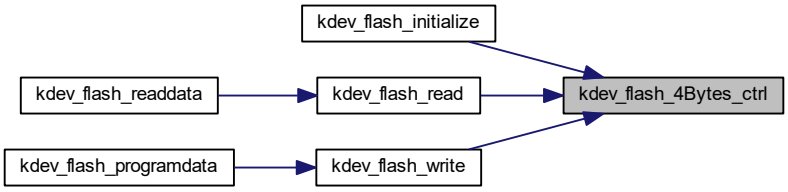
6.10.1.2 kdev_flash_4Bytes_ctrl()

```
void kdev_flash_4Bytes_ctrl (
            uint8_t enable )
```

Here is the call graph for this function:



Here is the caller graph for this function:

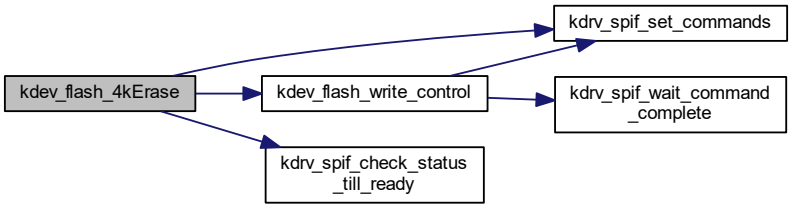


Kneron Confidential

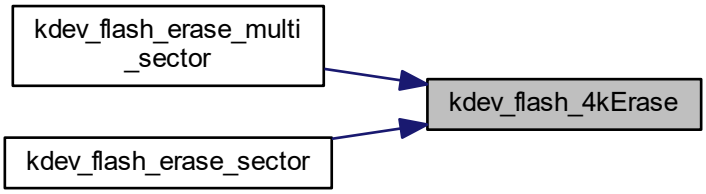
6.10.1.3 kdev_flash_4kErase()

```
void kdev_flash_4kErase (
            uint32_t offset )
```

Here is the call graph for this function:



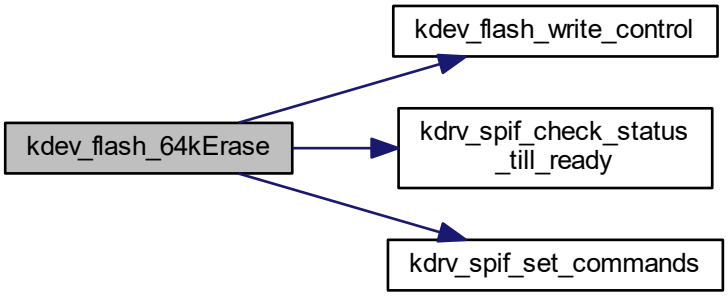
Here is the caller graph for this function:



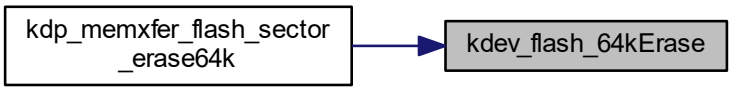
6.10.1.4 kdev_flash_64kErase()

```
void kdev_flash_64kErase (
            uint32_t offset )
```

Here is the call graph for this function:



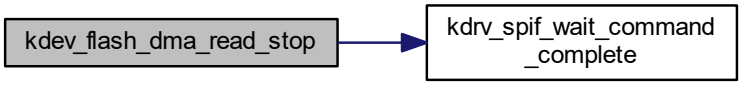
Here is the caller graph for this function:



6.10.1.5 kdev_flash_dma_read_stop()

```
void kdev_flash_dma_read_stop (
            void  )
```

Here is the call graph for this function:

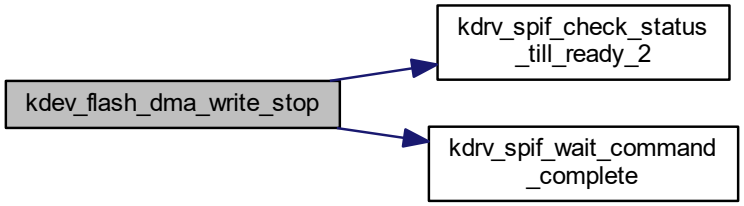


Kneron Confidential

6.10.1.6 kdev_flash_dma_write_stop()

```
void kdev_flash_dma_write_stop (
            void  )
```

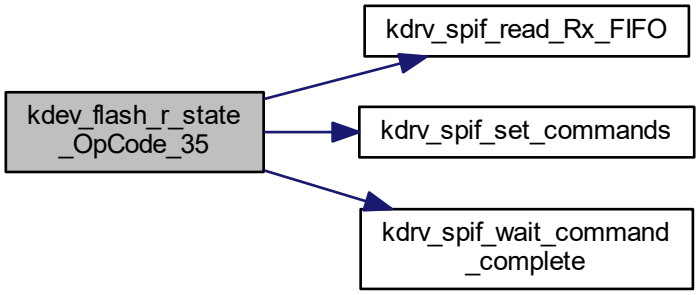
Here is the call graph for this function:



6.10.1.7 kdev_flash_r_state_OpCode_35()

```
uint8_t kdev_flash_r_state_OpCode_35 (
            void  )
```

Here is the call graph for this function:

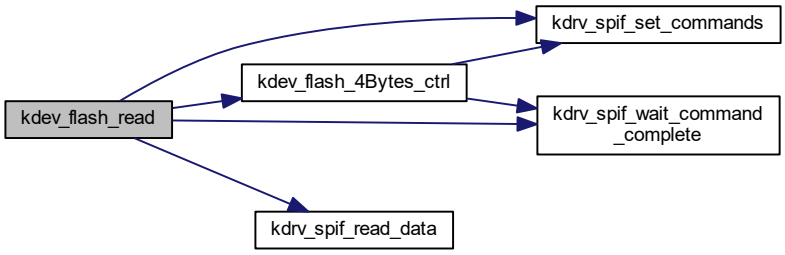


Kneron Confidential

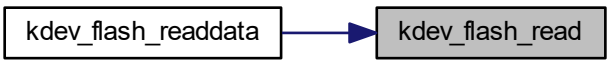
6.10.1.8 kdev_flash_read()

```
void kdev_flash_read (
            uint8_t type,
            uint32_t offset,
            uint32_t len,
            void * buf )
```

Here is the call graph for this function:



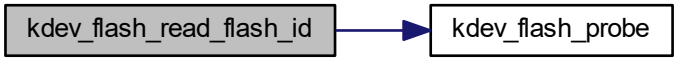
Here is the caller graph for this function:



6.10.1.9 kdev_flash_read_flash_id()

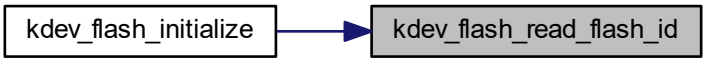
```
void kdev_flash_read_flash_id (
            void  )
```

Here is the call graph for this function:



Kneron Confidential

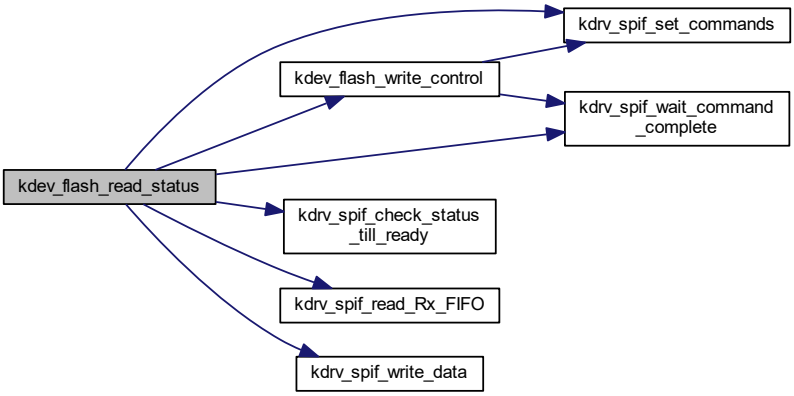
Here is the caller graph for this function:



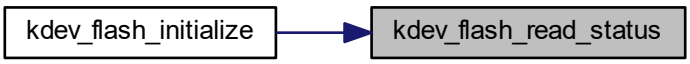
6.10.1.10 kdev_flash_read_status()

```
void kdev_flash_read_status (
            void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:

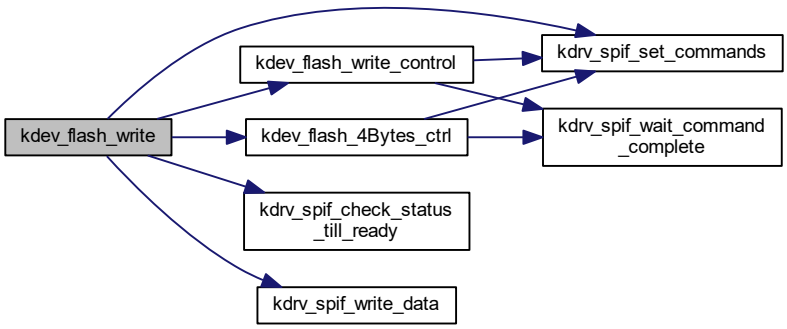


Kneron Confidential

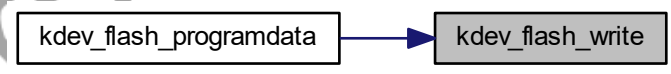
6.10.1.11 kdev_flash_write()

```
void kdev_flash_write (
            uint8_t type,
            uint32_t offset,
            uint32_t len,
            void * buf,
            uint32_t buf_offset )
```

Here is the call graph for this function:



Here is the caller graph for this function:

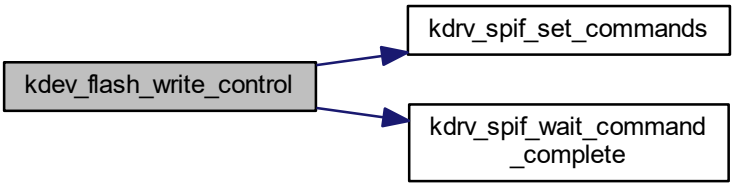


6.10.1.12 kdev_flash_write_control()

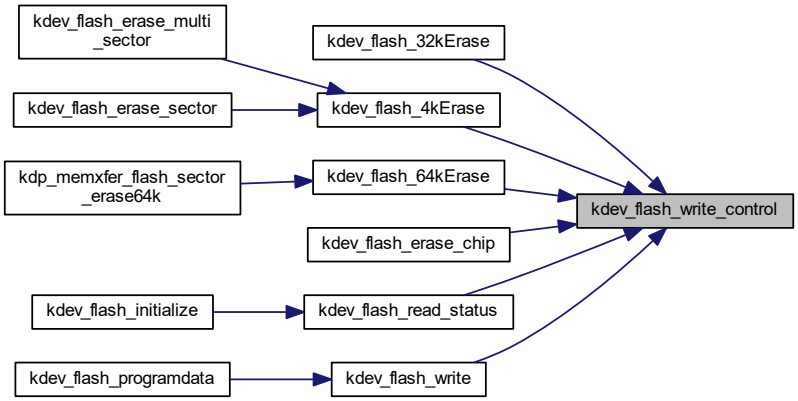
```
void kdev_flash_write_control (
            uint8_t enable )
```

Kneron Confidential

Here is the call graph for this function:



Here is the caller graph for this function:



6.10.2 Variable Documentation

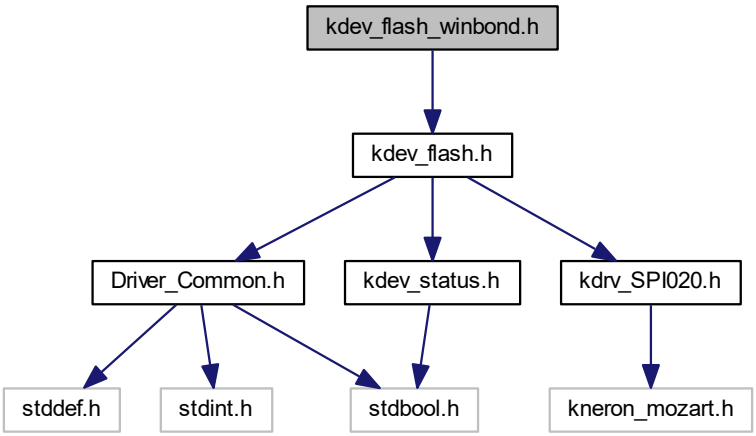
6.10.2.1 flash_info

```
spi_flash_t flash_info
```

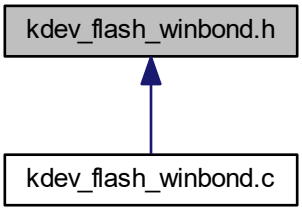
6.11 kdev_flash_winbond.h File Reference

```
#include "kdev_flash.h"
```

Include dependency graph for kdev_flash_winbond.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define FLASH_WB_DEV 0xEF
- #define FLASH_GD_DEV 0xC8
- #define FLASH_MXIC_DEV 0xC2
- #define FLASH_Micron_DEV 0x20
- #define FLASH_ZBIT_DEV 0x5E
- #define FLASH_IS_WB_DEV(x) ((x>>16)&0xFF == FLASH_WB_DEV)
- #define FLASH_IS_MXIC_DEV(x) ((x>>16)&0xFF == FLASH_MXIC_DEV)
- #define FLASH_IS_GD_DEV(x) ((x>>16)&0xFF == FLASH_GD_DEV)
- #define WB_W25Q01JV_ID 0x2140

- #define WB_W25Q512JV_ID 0x2040
- #define WB_W25L256JV_ID 0x1940
- #define WB_W25Q128BVFG_ID 0x1840
- #define WB_W25Q64CV_ID_9F 0x1740
- #define WB_W25Q32CV_ID_9F 0x1640
- #define WB_W25P80_ID_9F 0x1420
- #define WB_W25P16_ID_9F 0x1520
- #define WB_W25P32_ID_9F 0x1620
- #define WB_W25P16_ID_90 0x14
- #define MX_MX25L51245G 0x1A20
- #define MX_MX25L25645G 0x1920
- #define MX_MX25L12845EM1 0x1820
- #define MX_MX25L6405D 0x1720
- #define GD_GD25Q64CSIG_ID 0x1740
- #define GD_GD25Q127CSIG_ID 0x1840
- #define GD_GD25Q256_ID 0x1940
- #define GD_GD25Q256_ID_9F 0x1940
- #define GD25Q512MC_ID_9F 0x2040
- #define INVALID_CHIP_ID 0xFFFF
- #define INVALID_MANU_ID 0xFF
- #define FLASH_SIZE_32MB_ID 0x19
- #define FLASH_SIZE_16MB_ID 0x18
- #define FLASH_SIZE_8MB_ID 0x17
- #define FLASH_SIZE_4MB_ID 0x16
- #define FLASH_SIZE_2MB_ID 0x15
- #define FLASH_SIZE_1MB_ID 0x14
- #define FLASH_SIZE_512MB_ID 0x20
- #define FLASH_SIZE_SHIFT (FLASH_SIZE_512MB_ID - 0x1A)
- #define FLASH_3BYTE_ADDR_MAX 0x4000

6.11.1 Macro Definition Documentation

6.11.1.1 FLASH_3BYTE_ADDR_MAX

```
#define FLASH_3BYTE_ADDR_MAX 0x4000
```

6.11.1.2 FLASH_GD_DEV

```
#define FLASH_GD_DEV 0xC8
```

6.11.1.3 FLASH_IS_GD_DEV

```
#define FLASH_IS_GD_DEV(
              x ) ((x>>16)&0xFF == FLASH_GD_DEV)
```

Kneron Confidential

6.11.1.4 FLASH_IS_MXIC_DEV

```
#define FLASH_IS_MXIC_DEV(
              x ) ((x>>16)&0xFF == FLASH_MXIC_DEV)
```

6.11.1.5 FLASH_IS_WB_DEV

```
#define FLASH_IS_WB_DEV(
              x ) ((x>>16)&0xFF == FLASH_WB_DEV)
```

6.11.1.6 FLASH_Micron_DEV

```
#define FLASH_Micron_DEV 0x20
```

6.11.1.7 FLASH_MXIC_DEV

```
#define FLASH_MXIC_DEV 0xC2
```

6.11.1.8 FLASH_SIZE_16MB_ID

```
#define FLASH_SIZE_16MB_ID 0x18
```

6.11.1.9 FLASH_SIZE_1MB_ID

```
#define FLASH_SIZE_1MB_ID 0x14
```

6.11.1.10 FLASH_SIZE_2MB_ID

```
#define FLASH_SIZE_2MB_ID 0x15
```

Kneron Confidential

6.11.1.11 FLASH_SIZE_32MB_ID

```
#define FLASH_SIZE_32MB_ID 0x19
```

6.11.1.12 FLASH_SIZE_4MB_ID

```
#define FLASH_SIZE_4MB_ID 0x16
```

6.11.1.13 FLASH_SIZE_512MB_ID

```
#define FLASH_SIZE_512MB_ID 0x20
```

6.11.1.14 FLASH_SIZE_8MB_ID

```
#define FLASH_SIZE_8MB_ID 0x17
```

6.11.1.15 FLASH_SIZE_SHIFT

```
#define FLASH_SIZE_SHIFT (FLASH_SIZE_512MB_ID - 0x1A)
```

6.11.1.16 FLASH_WB_DEV

```
#define FLASH_WB_DEV 0xEF
```

6.11.1.17 FLASH_ZBIT_DEV

```
#define FLASH_ZBIT_DEV 0x5E
```

6.11.1.18 GD25Q512MC_ID_9F

```
#define GD25Q512MC_ID_9F 0x2040
```

Kneron Confidential

6.11.1.19 GD_GD25Q127CSIG_ID

```
#define GD_GD25Q127CSIG_ID 0x1840
```

6.11.1.20 GD_GD25Q256_ID

```
#define GD_GD25Q256_ID 0x1940
```

6.11.1.21 GD_GD25Q256_ID_9F

```
#define GD_GD25Q256_ID_9F 0x1940
```

6.11.1.22 GD_GD25Q64CSIG_ID

```
#define GD_GD25Q64CSIG_ID 0x1740
```

6.11.1.23 INVALID_CHIP_ID

```
#define INVALID_CHIP_ID 0xFFFF
```

6.11.1.24 INVALID_MANU_ID

```
#define INVALID_MANU_ID 0xFF
```

6.11.1.25 MX_MX25L12845EM1

```
#define MX_MX25L12845EM1 0x1820
```

6.11.1.26 MX_MX25L25645G

```
#define MX_MX25L25645G 0x1920
```

Kneron Confidential

6.11.1.27 MX_MX25L51245G

```
#define MX_MX25L51245G 0x1A20
```

6.11.1.28 MX_MX25L6405D

```
#define MX_MX25L6405D 0x1720
```

6.11.1.29 WB_W25L256JV_ID

```
#define WB_W25L256JV_ID 0x1940
```

6.11.1.30 WB_W25P16_ID_90

```
#define WB_W25P16_ID_90 0x14
```

6.11.1.31 WB_W25P16_ID_9F

```
#define WB_W25P16_ID_9F 0x1520
```

6.11.1.32 WB_W25P32_ID_9F

```
#define WB_W25P32_ID_9F 0x1620
```

6.11.1.33 WB_W25P80_ID_9F

```
#define WB_W25P80_ID_9F 0x1420
```

6.11.1.34 WB_W25Q01JV_ID

```
#define WB_W25Q01JV_ID 0x2140
```

Kneron Confidential

6.11.1.35 WB_W25Q128BVFG_ID

```
#define WB_W25Q128BVFG_ID 0x1840
```

6.11.1.36 WB_W25Q32CV_ID_9F

```
#define WB_W25Q32CV_ID_9F 0x1640
```

6.11.1.37 WB_W25Q512JV_ID

```
#define WB_W25Q512JV_ID 0x2040
```

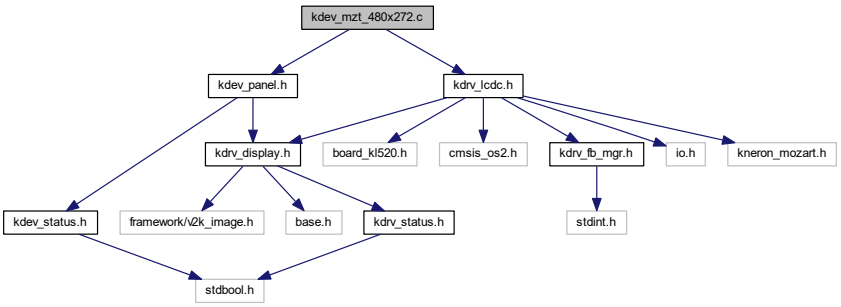
6.11.1.38 WB_W25Q64CV_ID_9F

```
#define WB_W25Q64CV_ID_9F 0x1740
```

6.12 kdev_mzt_480x272.c File Reference

```
#include "kdev_panel.h"
#include "kdrv_lcdc.h"
```

Include dependency graph for kdev_mzt_480x272.c:



Macros

- #define MZT_PANEL_WIDTH TFT43_WIDTH
- #define MZT_PANEL_HEIGHT TFT43_HEIGHT

Kneron Confidential

Functions

- kdev_status_t kdev_panel_initialize (kdrv_display_t *display_drv)

  Initializes kdev panel driver.
- kdev_status_t kdev_panel_clear (kdrv_display_t *display_drv, u32 color)
- uint16_t kdev_panel_read_display_id (kdrv_display_t *display_drv)
- kdev_status_t kdev_panel_refresh (kdrv_display_t *display_drv)

6.12.1 Macro Definition Documentation

6.12.1.1 MZT_PANEL_HEIGHT

```
#define MZT_PANEL_HEIGHT TFT43_HEIGHT
```

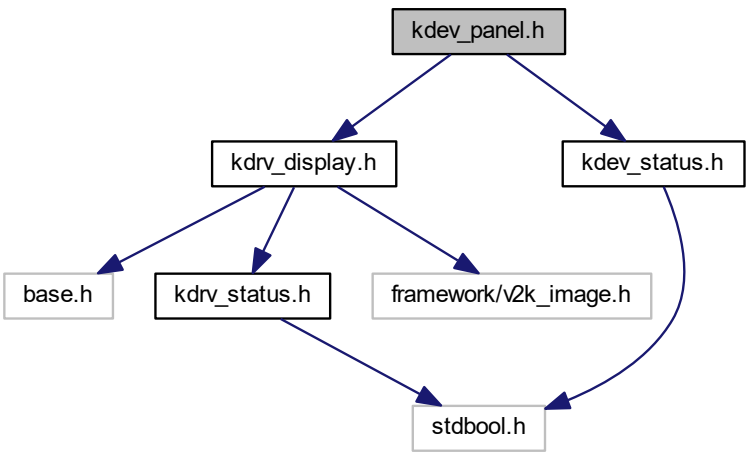
6.12.1.2 MZT_PANEL_WIDTH

```
#define MZT_PANEL_WIDTH TFT43_WIDTH
```

6.13 kdev_panel.h File Reference

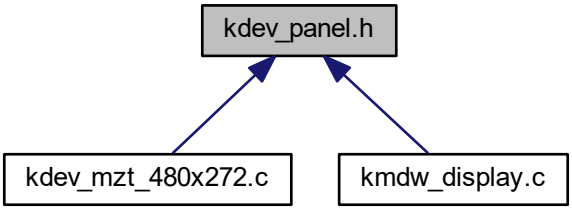
```
#include "kdrv_display.h"
#include "kdev_status.h"
```

Include dependency graph for kdev_panel.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Functions

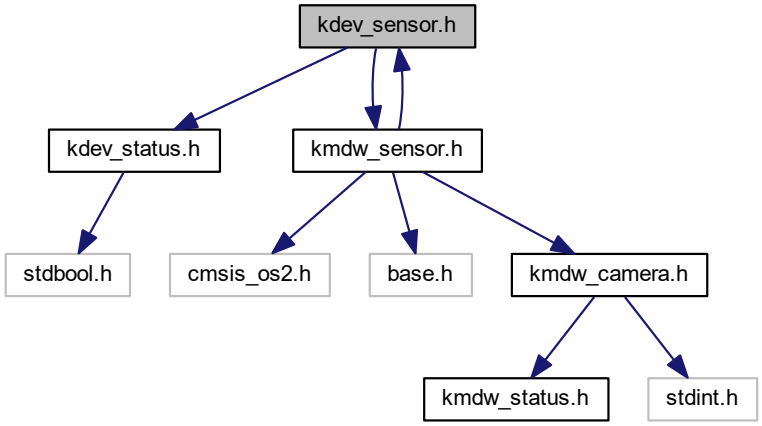
- kdev_status_t kdev_panel_initialize (kdrv_display_t *display_drv)

  Initializes kdev panel driver.
- kdev_status_t kdev_panel_clear (kdrv_display_t *display_drv, u32 color)
- uint16_t kdev_panel_read_display_id (kdrv_display_t *display_drv)
- kdev_status_t kdev_panel_refresh (kdrv_display_t *display_drv)

6.14 kdev_sensor.h File Reference

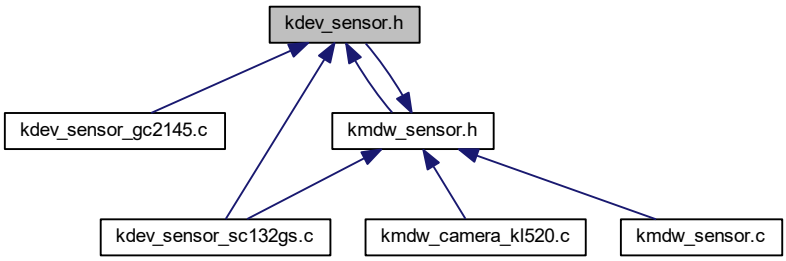
```
#include "kdev_status.h"
#include "kmdw_sensor.h"
```

Include dependency graph for kdev_sensor.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



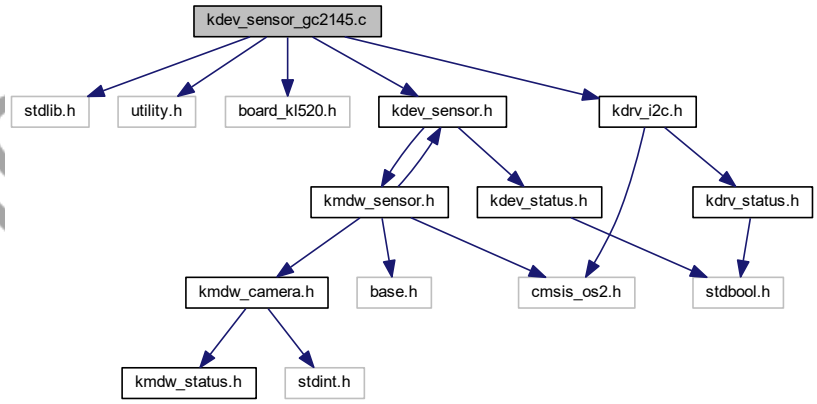
Data Structures

- struct sensor_ops

6.15 kdev_sensor_gc2145.c File Reference

```
#include <stdlib.h>
#include "utility.h"
#include "board_kl520.h"
#include "kdev_sensor.h"
#include "kdrv_i2c.h"
```

Include dependency graph for kdev_sensor_gc2145.c:



Macros

- #define sensor_msg(fmt, ...)

Kneron Confidential

Functions

- uint32_t gc2145_init (struct sensor_device *sensor_dev, struct sensor_init_seq *seq)
- struct sensor_ops * kdev_sensor_gc2145_get_ops (void)

Variables

- struct sensor_init_seq gc2145_init_regs [ ]

6.15.1 Macro Definition Documentation

6.15.1.1 sensor_msg

```
#define sensor_msg(
              fmt,
              ...  )
```

6.15.2 Function Documentation

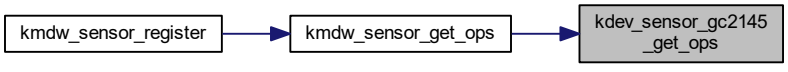
6.15.2.1 gc2145_init()

```
uint32_t gc2145_init (
            struct sensor_device * sensor_dev,
            struct sensor_init_seq * seq )
```

6.15.2.2 kdev_sensor_gc2145_get_ops()

```
struct sensor_ops* kdev_sensor_gc2145_get_ops (
            void  )
```

Here is the caller graph for this function:



Kneron Confidential

6.15.3 Variable Documentation

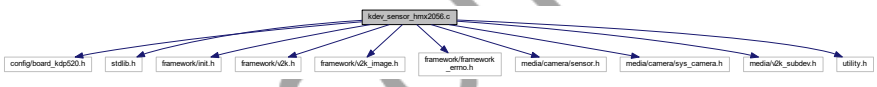
6.15.3.1 gc2145_init_regs

```
struct sensor_init_seq gc2145_init_regs[]
```

6.16 kdev_sensor_hmx2056.c File Reference

```
#include "config/board_kdp520.h"
#include <stdlib.h>
#include "framework/init.h"
#include "framework/v2k.h"
#include "framework/v2k_image.h"
#include "framework/framework_errno.h"
#include "media/camera/sensor.h"
#include "media/camera/sys_camera.h"
#include "media/v2k_subdev.h"
#include "utility.h"
```

Include dependency graph for kdev_sensor_hmx2056.c:



Data Structures

- struct hmx2056_context

Macros

- #define RES_640x480 1
- #define RES_480x272 2
- #define HMX2056_RES RES_640x480

Functions

- void hmx2056_init (struct sensor_device *sensor_dev, struct sensor_init_seq *seq)
- KDP_SENSOR_DRIVER_SETUP (hmx2056_i2c_driver)

Variables

- struct sensor_init_seq INITDATA hmx2056_init_regs [ ]
- struct core_device hmx2056_link_device

Kneron Confidential

6.16.1 Macro Definition Documentation

6.16.1.1 HMX2056_RES

```
#define HMX2056_RES RES_640x480
```

6.16.1.2 RES_480x272

```
#define RES_480x272 2
```

6.16.1.3 RES_640x480

```
#define RES_640x480 1
```

6.16.2 Function Documentation

6.16.2.1 hmx2056_init()

```
void hmx2056_init (
            struct sensor_device * sensor_dev,
            struct sensor_init_seq * seq )
```

6.16.2.2 KDP_SENSOR_DRIVER_SETUP()

```
KDP_SENSOR_DRIVER_SETUP (
            hmx2056_i2c_driver  )
```

6.16.3 Variable Documentation

Kneron Confidential

6.16.3.1 hmx2056_init_regs

```
struct sensor_init_seq INITDATA hmx2056_init_regs[]
```

6.16.3.2 hmx2056_link_device

```
struct core_device hmx2056_link_device
```

6.17 kdev_sensor_hmxrica.c File Reference

```
#include "config/board_kdp520.h"
#include <stdlib.h>
#include "framework/init.h"
#include "framework/framework_driver.h"
#include "framework/v2k.h"
#include "framework/v2k_image.h"
#include "framework/framework_errno.h"
#include "media/camera/sensor.h"
#include "media/camera/sys_camera.h"
#include "media/v2k_subdev.h"
#include "utility.h"
#include "kmdw_console.h"
```

Include dependency graph for kdev_sensor_hmxrica.c:

Data Structures

- struct hmxrica_context

Functions

- void hmxrica_init (struct sensor_device *sensor_dev, struct sensor_init_seq *seq)
- KDP_SENSOR_DRIVER_SETUP (hmxrica_i2c_driver)

Variables

- struct core_device hmxrica_link_device
- struct sensor_driver hmxrica_i2c_driver

Kneron Confidential

6.17.1 Function Documentation

6.17.1.1 hmxrica_init()

```
void hmxrica_init (
            struct sensor_device * sensor_dev,
            struct sensor_init_seq * seq )
```

6.17.1.2 KDP_SENSOR_DRIVER_SETUP()

```
KDP_SENSOR_DRIVER_SETUP (
            hmxrica_i2c_driver  )
```

6.17.2 Variable Documentation

6.17.2.1 hmxrica_i2c_driver

```
struct sensor_driver hmxrica_i2c_driver
```

Initial value:

```
= {
    .driver = {
        .name = "sensor-hmxrica",
    },
    .probe      = hmxrica_probe,
    .remove     = hmxrica_remove,
    .core_dev   = &hmxrica_link_device,
}
```

6.17.2.2 hmxrica_link_device

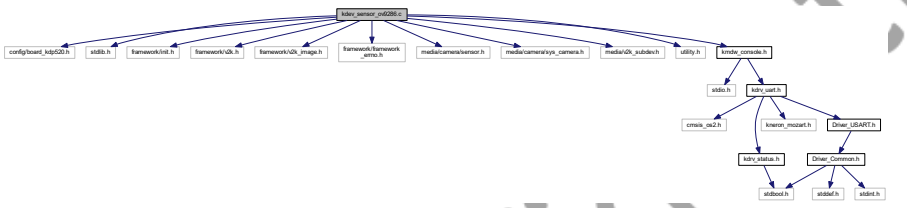
```
struct core_device hmxrica_link_device
```

Kneron Confidential

6.18 kdev_sensor_ov9286.c File Reference

```
#include "config/board_kdp520.h"
#include <stdlib.h>
#include "framework/init.h"
#include "framework/v2k.h"
#include "framework/v2k_image.h"
#include "framework/framework_errno.h"
#include "media/camera/sensor.h"
#include "media/camera/sys_camera.h"
#include "media/v2k_subdev.h"
#include "utility.h"
#include "kmdw_console.h"
```

Include dependency graph for kdev_sensor_ov9286.c:



Data Structures

- struct ov9286_context

Macros

- #define FPS_30 0
- #define FPS_16 1
- #define OV8286_FPS FPS_30

Functions

- void ov9286_init (struct sensor_device *sensor_device, struct sensor_init_seq *seq)
- KDP_SENSOR_DRIVER_SETUP (ov9286_i2c_driver)

Variables

- struct sensor_init_seq INITDATA ov9286_init_regs [ ]
- struct core_device ov9286_link_device

6.18.1 Macro Definition Documentation

Kneron Confidential

6.18.1.1 FPS_16

```
#define FPS_16 1
```

6.18.1.2 FPS_30

```
#define FPS_30 0
```

6.18.1.3 OV8286_FPS

```
#define OV8286_FPS FPS_30
```

6.18.2 Function Documentation

6.18.2.1 KDP_SENSOR_DRIVER_SETUP()

```
KDP_SENSOR_DRIVER_SETUP (
            ov9286_i2c_driver  )
```

6.18.2.2 ov9286_init()

```
void ov9286_init (
            struct sensor_device * sensor_device,
            struct sensor_init_seq * seq )
```

6.18.3 Variable Documentation

6.18.3.1 ov9286_init_regs

```
struct sensor_init_seq INITDATA ov9286_init_regs[]
```

Kneron Confidential

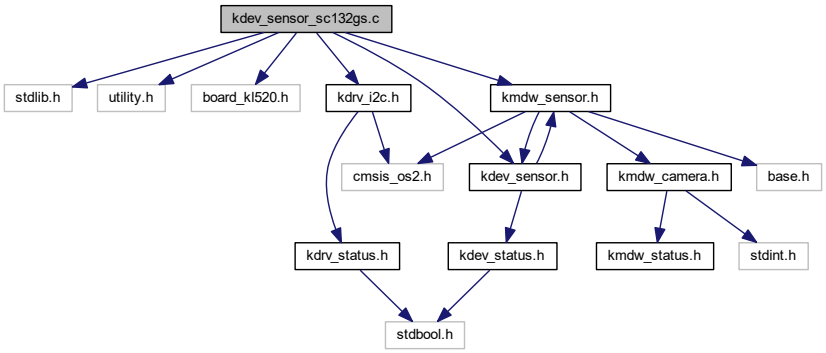
6.18.3.2 ov9286_link_device

```
struct core_device ov9286_link_device
```

6.19 kdev_sensor_sc132gs.c File Reference

```
#include <stdlib.h>
#include "utility.h"
#include "board_kl520.h"
#include "kmdw_sensor.h"
#include "kdev_sensor.h"
#include "kdrv_i2c.h"
```

Include dependency graph for kdev_sensor_sc132gs.c:



Macros

- #define sensor_msg(fmt, ...)

Functions

- void sc132gs_init (struct sensor_device *sensor_dev, struct sensor_init_seq *seq)
- struct sensor_ops * kdev_sensor_sc132gs_get_ops (void)

Variables

- struct sensor_init_seq sc132gs_init_regs [ ]

6.19.1 Macro Definition Documentation

Kneron Confidential

6.19.1.1 sensor_msg

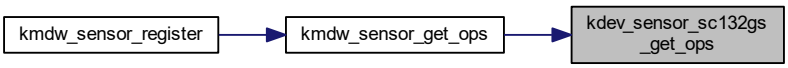
```
#define sensor_msg(
              fmt,
              ...  )
```

6.19.2 Function Documentation

6.19.2.1 kdev_sensor_sc132gs_get_ops()

```
struct sensor_ops* kdev_sensor_sc132gs_get_ops (
             void  )
```

Here is the caller graph for this function:



6.19.2.2 sc132gs_init()

```
void sc132gs_init (
             struct sensor_device * sensor_dev,
             struct sensor_init_seq * seq )
```

6.19.3 Variable Documentation

6.19.3.1 sc132gs_init_regs

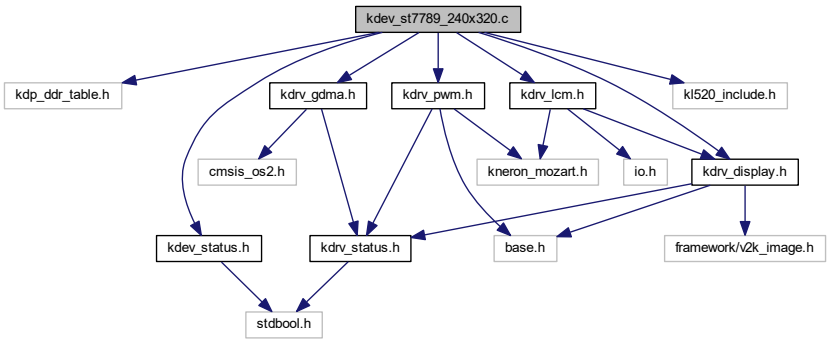
```
struct sensor_init_seq sc132gs_init_regs[]
```

Kneron Confidential

6.20 kdev_st7789_240x320.c File Reference

```
#include "kdp_ddr_table.h"
#include "kdrv_gdma.h"
#include "kdrv_display.h"
#include "kdrv_lcm.h"
#include "kl520_include.h"
#include "kdev_status.h"
#include "kdrv_pwm.h"
```

Include dependency graph for kdev_st7789_240x320.c:



Macros

- #define TE_ENABLE
- #define PORTRAIT_DMA_ENABLE
- #define FRAME_RATE_60HZ 0
- #define FRAME_RATE_39HZ 1
- #define FRAME_RATE_CTRL FRAME_RATE_60HZ
- #define RAM_ENDIAN_MSB 0
- #define RAM_ENDIAN_LSB 1
- #define RAM_ENDIAN_TYPE RAM_ENDIAN_MSB
- #define PANEL_REG_RAMWR 0x2C
- #define PANEL_REG_WRMEMC 0x3C
- #define ST7789_PORTRAIT
- #define ST7789_PANEL_WIDTH QVGA_PORTRAIT_WIDTH
- #define ST7789_PANEL_HEIGHT QVGA_PORTRAIT_HEIGHT
- #define ST7789_PANEL_TOTAL_PIXEL ST7789_PANEL_WIDTH ∗ ST7789_PANEL_HEIGHT
- #define OPEN_DOWN_SCALING
- #define BYTES_PER_PIXEL 2
- #define FRAME_RATE_CTRL FRAME_RATE_39HZ
- #define RAM_ENDIAN_TYPE RAM_ENDIAN_LSB
- #define TE_ENABLE

Functions

- kdev_status_t kdev_panel_initialize (kdrv_display_t ∗display_drv)

  Initializes kdev panel driver.
- kdev_status_t kdev_panel_clear (kdrv_display_t ∗display_drv, u32 color)
- uint16_t kdev_panel_read_display_id (kdrv_display_t ∗display_drv)
- kdev_status_t kdev_panel_refresh (kdrv_display_t ∗display_drv)

Kneron Confidential

Variables

- osMutexId_t mutex_st7789 = NULL
- osMutexId_t mutex_snapshot = NULL

6.20.1 Macro Definition Documentation

6.20.1.1 BYTES_PER_PIXEL

```
#define BYTES_PER_PIXEL 2
```

6.20.1.2 FRAME_RATE_39HZ

```
#define FRAME_RATE_39HZ 1
```

6.20.1.3 FRAME_RATE_60HZ

```
#define FRAME_RATE_60HZ 0
```

6.20.1.4 FRAME_RATE_CTRL [1/2]

```
#define FRAME_RATE_CTRL FRAME_RATE_60HZ
```

6.20.1.5 FRAME_RATE_CTRL [2/2]

```
#define FRAME_RATE_CTRL FRAME_RATE_39HZ
```

6.20.1.6 OPEN_DOWN_SCALING

```
#define OPEN_DOWN_SCALING
```

Kneron Confidential

6.20.1.7 PANEL_REG_RAMWR

```
#define PANEL_REG_RAMWR 0x2C
```

6.20.1.8 PANEL_REG_WRMEMC

```
#define PANEL_REG_WRMEMC 0x3C
```

6.20.1.9 PORTRAIT_DMA_ENABLE

```
#define PORTRAIT_DMA_ENABLE
```

6.20.1.10 RAM_ENDIAN_LSB

```
#define RAM_ENDIAN_LSB 1
```

6.20.1.11 RAM_ENDIAN_MSB

```
#define RAM_ENDIAN_MSB 0
```

6.20.1.12 RAM_ENDIAN_TYPE [1/2]

```
#define RAM_ENDIAN_TYPE RAM_ENDIAN_MSB
```

6.20.1.13 RAM_ENDIAN_TYPE [2/2]

```
#define RAM_ENDIAN_TYPE RAM_ENDIAN_LSB
```

6.20.1.14 ST7789_PANEL_HEIGHT

```
#define ST7789_PANEL_HEIGHT QVGA_PORTRAIT_HEIGHT
```

Kneron Confidential

6.20.1.15 ST7789_PANEL_TOTAL_PIXEL

```
#define ST7789_PANEL_TOTAL_PIXEL ST7789_PANEL_WIDTH * ST7789_PANEL_HEIGHT
```

6.20.1.16 ST7789_PANEL_WIDTH

```
#define ST7789_PANEL_WIDTH QVGA_PORTRAIT_WIDTH
```

6.20.1.17 ST7789_PORTRAIT

```
#define ST7789_PORTRAIT
```

6.20.1.18 TE_ENABLE [1/2]

```
#define TE_ENABLE
```

6.20.1.19 TE_ENABLE [2/2]

```
#define TE_ENABLE
```

6.20.2 Variable Documentation

6.20.2.1 mutex_snapshot

```
osMutexId_t mutex_snapshot = NULL
```

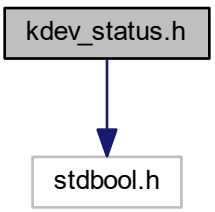
6.20.2.2 mutex_st7789

```
osMutexId_t mutex_st7789 = NULL
```

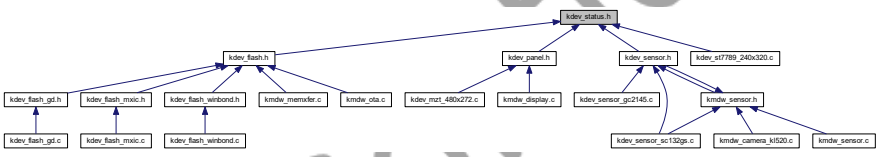
Kneron Confidential

6.21 kdev_status.h File Reference

`#include <stdbool.h>`
Include dependency graph for kdev_status.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum kdev_status_t { KDEV_STATUS_OK = 0, KDEV_STATUS_ERROR }

6.21.1 Enumeration Type Documentation

6.21.1.1 kdev_status_t

`enum kdev_status_t`

Enumerator

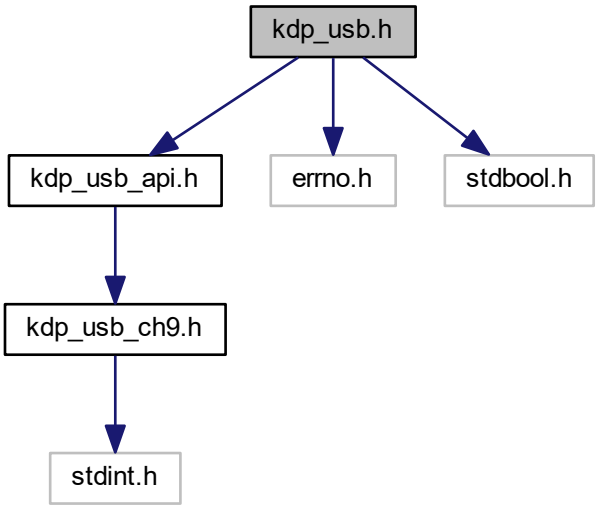
KDEV_STATUS_OK	driver status OK
KDEV_STATUS_ERROR	driver status error

Kneron Confidential

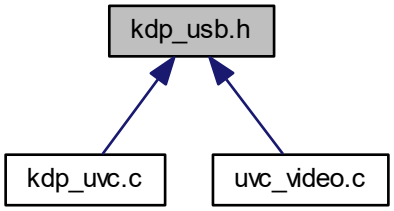
6.22 kdp_usb.h File Reference

```
#include "kdp_usb_api.h"
#include <errno.h>
#include <stdbool.h>
```

Include dependency graph for kdp_usb.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct usb_device_id

Macros

- #define USB_DEVICE_ID_MATCH_DEVICE (USB_DEVICE_ID_MATCH_VENDOR | USB_DEVICE_ID_MATCH_PRODUCT)
- #define USB_DEVICE(vend, prod)

6.22.1 Macro Definition Documentation

6.22.1.1 USB_DEVICE

```
#define USB_DEVICE(
                vend,
                prod )
```

Value:

```
.match_flags = USB_DEVICE_ID_MATCH_DEVICE, \
        .idVendor = (vend), \
        .idProduct = (prod)
```

USB_DEVICE - macro used to describe a specific usb device @vend: the 16 bit USB Vendor ID @prod: the 16 bit USB Product ID

This macro is used to create a struct usb_device_id that matches a specific device.

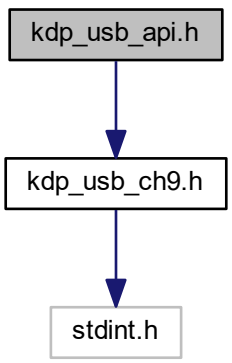
6.22.1.2 USB_DEVICE_ID_MATCH_DEVICE

```
#define USB_DEVICE_ID_MATCH_DEVICE (USB_DEVICE_ID_MATCH_VENDOR | USB_DEVICE_ID_MATCH_PRODUCT)
```

6.23 kdp_usb_api.h File Reference

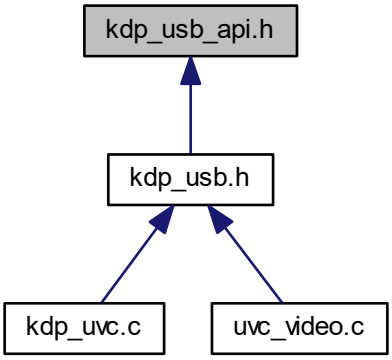
```
#include "kdp_usb_ch9.h"
```

Include dependency graph for kdp_usb_api.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct kdp_uvc_id
- struct usb_endpoint
- struct usb_inf_alt
- struct usb_interface
- struct usb_config
- struct usb_device

Macros

- #define USB_DEVICE_ID_MATCH_VENDOR 0x0001
- #define USB_DEVICE_ID_MATCH_PRODUCT 0x0002
- #define USB_DEVICE_ID_MATCH_INT_CLASS 0x0080
- #define USB_DEVICE_ID_MATCH_INT_SUBCLASS 0x0100
- #define USB_DEVICE_ID_MATCH_INT_PROTOCOL 0x0200
- #define USB_DEVICE_ID_MATCH_DEV_HI 0x0008
- #define USB_DEVICE_ID_MATCH_DEVICE (USB_DEVICE_ID_MATCH_VENDOR | USB_DEVICE_ID_MATCH_PRODUCT)
- #define USB_DEVICE_ID_MATCH_INT_INFO

Typedefs

- typedef unsigned int size_t

Enumerations

- enum usb_init_type { USB_INIT_HOST = 0, USB_INIT_DEVICE }

Kneron Confidential

Functions

- struct usb_endpoint __attribute__ ((packed))

Variables

- struct usb_endpoint_descriptor desc
- void * hcpriv
- unsigned char * extra
- int extralen
- int enabled
- int streams
- uint8_t num_ep
- struct usb_endpoint * p_ep
- struct usb_inf_alt * p_alt
- uint8_t num_alt
- uint8_t cur_num
- struct usb_config_descriptor * pdesc
- uint8_t num_inf
- uint8_t cur_inf
- struct usb_interface * p_inf
- struct usb_device __attribute__

6.23.1 Macro Definition Documentation

6.23.1.1 USB_DEVICE_ID_MATCH_DEV_HI

```
#define USB_DEVICE_ID_MATCH_DEV_HI 0x0008
```

6.23.1.2 USB_DEVICE_ID_MATCH_DEVICE

```
#define USB_DEVICE_ID_MATCH_DEVICE (USB_DEVICE_ID_MATCH_VENDOR | USB_DEVICE_ID_MATCH_PRODUCT)
```

6.23.1.3 USB_DEVICE_ID_MATCH_INT_CLASS

```
#define USB_DEVICE_ID_MATCH_INT_CLASS 0x0080
```

Kneron Confidential

6.23.1.4 USB_DEVICE_ID_MATCH_INT_INFO

```
#define USB_DEVICE_ID_MATCH_INT_INFO
```

Value:

```
(USB_DEVICE_ID_MATCH_INT_CLASS | \
                USB_DEVICE_ID_MATCH_INT_SUBCLASS | \
                USB_DEVICE_ID_MATCH_INT_PROTOCOL)
```

6.23.1.5 USB_DEVICE_ID_MATCH_INT_PROTOCOL

```
#define USB_DEVICE_ID_MATCH_INT_PROTOCOL 0x0200
```

6.23.1.6 USB_DEVICE_ID_MATCH_INT_SUBCLASS

```
#define USB_DEVICE_ID_MATCH_INT_SUBCLASS 0x0100
```

6.23.1.7 USB_DEVICE_ID_MATCH_PRODUCT

```
#define USB_DEVICE_ID_MATCH_PRODUCT 0x0002
```

6.23.1.8 USB_DEVICE_ID_MATCH_VENDOR

```
#define USB_DEVICE_ID_MATCH_VENDOR 0x0001
```

6.23.2 Typedef Documentation

6.23.2.1 size_t

```
typedef unsigned int size_t
```

6.23.3 Enumeration Type Documentation

6.23.3.1 usb_init_type

```
enum usb_init_type
```

Kneron Confidential

Enumerator

USB_INIT_HOST	
USB_INIT_DEVICE	

6.23.4 Function Documentation

6.23.4.1 __attribute__()

```
struct usb_endpoint __attribute__ (
            (packed)  )
```

6.23.5 Variable Documentation

6.23.5.1 __attribute__

```
struct usb_device __attribute__
```

6.23.5.2 cur_inf

```
uint8_t cur_inf
```

6.23.5.3 cur_num

```
uint8_t cur_num
```

6.23.5.4 desc

```
struct usb_interface_descriptor desc
```

Kneron Confidential

6.23.5.5 enabled

```
int enabled
```

6.23.5.6 extra

```
unsigned char * extra
```

6.23.5.7 extralen

```
int extralen
```

6.23.5.8 hcpriv

```
void* hcpriv
```

6.23.5.9 num_alt

```
uint8_t num_alt
```

6.23.5.10 num_ep

```
uint8_t num_ep
```

6.23.5.11 num_inf

```
uint8_t num_inf
```

6.23.5.12 p_alt

```
struct usb_inf_alt* p_alt
```

Kneron Confidential

6.23.5.13 p_ep

```
struct usb_endpoint* p_ep
```

6.23.5.14 p_inf

```
struct usb_interface* p_inf
```

6.23.5.15 pdesc

```
struct usb_config_descriptor* pdesc
```

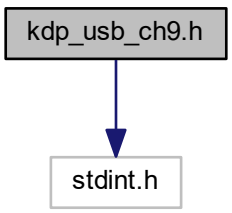
6.23.5.16 streams

```
int streams
```

6.24 kdp_usb_ch9.h File Reference

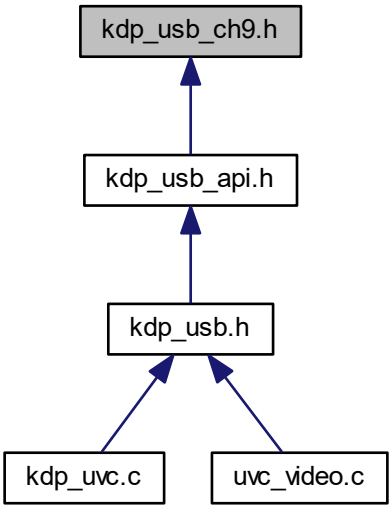
```
#include <stdint.h>
```

Include dependency graph for kdp_usb_ch9.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct usb_ctrlrequest
- struct usb_descriptor_header
- struct usb_device_descriptor
- struct usb_device_qualifier_descriptor
- struct usb_other_speed_configuration_descriptor
- struct usb_otg_descriptor
- struct usb_otg20_descriptor
- struct usb_config_descriptor
- struct usb_string_descriptor
- struct usb_interface_descriptor
- struct usb_endpoint_descriptor
- struct usb_ssp_isoc_ep_comp_descriptor
- struct usb_ss_ep_comp_descriptor
- struct usb_qualifier_descriptor
- struct usb_debug_descriptor
- struct usb_interface_assoc_descriptor
- struct usb_security_descriptor
- struct usb_key_descriptor
- struct usb_encryption_descriptor
- struct usb_bos_descriptor
- struct usb_set_sel_req

Kneron Confidential

Macros

- #define USB_DIR_OUT 0x0 /* to device */
- #define USB_DIR_IN 0x80 /* to host */
- #define USB_TYPE_MASK (0x03 << 5)
- #define USB_TYPE_STANDARD (0x00 << 5)
- #define USB_TYPE_CLASS (0x01 << 5)
- #define USB_TYPE_VENDOR (0x02 << 5)
- #define USB_TYPE_RESERVED (0x03 << 5)
- #define USB_RECIP_MASK 0x1f
- #define USB_RECIP_DEVICE 0x00
- #define USB_RECIP_INTERFACE 0x01
- #define USB_RECIP_ENDPOINT 0x02
- #define USB_RECIP_OTHER 0x03
- #define USB_RECIP_PORT 0x04
- #define USB_RECIP_RPIPE 0x05
- #define USB_REQ_GET_STATUS 0x00
- #define USB_REQ_CLEAR_FEATURE 0x01
- #define USB_REQ_SET_FEATURE 0x03
- #define USB_REQ_SET_ADDRESS 0x05
- #define USB_REQ_GET_DESCRIPTOR 0x06
- #define USB_REQ_SET_DESCRIPTOR 0x07
- #define USB_REQ_GET_CONFIGURATION 0x08
- #define USB_REQ_SET_CONFIGURATION 0x09
- #define USB_REQ_GET_INTERFACE 0x0A
- #define USB_REQ_SET_INTERFACE 0x0B
- #define USB_REQ_SYNCH_FRAME 0x0C
- #define USB_REQ_SET_SEL 0x30
- #define USB_REQ_SET_ISOCH_DELAY 0x31
- #define USB_REQ_SET_ENCRYPTION 0x0D /* Wireless USB */
- #define USB_REQ_GET_ENCRYPTION 0x0E
- #define USB_REQ_RPIPE_ABORT 0x0E
- #define USB_REQ_SET_HANDSHAKE 0x0F
- #define USB_REQ_RPIPE_RESET 0x0F
- #define USB_REQ_GET_HANDSHAKE 0x10
- #define USB_REQ_SET_CONNECTION 0x11
- #define USB_REQ_SET_SECURITY_DATA 0x12
- #define USB_REQ_GET_SECURITY_DATA 0x13
- #define USB_REQ_SET_WUSB_DATA 0x14
- #define USB_REQ_LOOPBACK_DATA_WRITE 0x15
- #define USB_REQ_LOOPBACK_DATA_READ 0x16
- #define USB_REQ_SET_INTERFACE_DS 0x17
- #define USB_REQ_GET_PARTNER_PDO 20
- #define USB_REQ_GET_BATTERY_STATUS 21
- #define USB_REQ_SET_PDO 22
- #define USB_REQ_GET_VDM 23
- #define USB_REQ_SEND_VDM 24
- #define USB_REQ_GET_REPORT 0x01
- #define USB_REQ_GET_IDLE 0x02
- #define USB_REQ_GET_PROTOCOL 0x03
- #define USB_REQ_SET_REPORT 0x09
- #define USB_REQ_SET_IDLE 0x0A
- #define USB_REQ_SET_PROTOCOL 0x0B
- #define USB_DEVICE_SELF_POWERED 0 /* (read only) */
- #define USB_DEVICE_REMOTE_WAKEUP 1 /* dev may initiate wakeup */

Kneron Confidential

- #define USB_DEVICE_TEST_MODE 2 /* (wired high speed only) */
- #define USB_DEVICE_BATTERY 2 /* (wireless) */
- #define USB_DEVICE_B_HNP_ENABLE 3 /* (otg) dev may initiate HNP */
- #define USB_DEVICE_WUSB_DEVICE 3 /* (wireless)*/
- #define USB_DEVICE_A_HNP_SUPPORT 4 /* (otg) RH port supports HNP */
- #define USB_DEVICE_A_ALT_HNP_SUPPORT 5 /* (otg) other RH port does */
- #define USB_DEVICE_DEBUG_MODE 6 /* (special devices only) */
- #define TEST_J 1
- #define TEST_K 2
- #define TEST_SE0_NAK 3
- #define TEST_PACKET 4
- #define TEST_FORCE_EN 5
- #define USB_DEVICE_U1_ENABLE 48 /* dev may initiate U1 transition */
- #define USB_DEVICE_U2_ENABLE 49 /* dev may initiate U2 transition */
- #define USB_DEVICE_LTM_ENABLE 50 /* dev may send LTM */
- #define USB_INTRF_FUNC_SUSPEND 0 /* function suspend */
- #define USB_INTR_FUNC_SUSPEND_OPT_MASK 0xFF00
- #define USB_INTRF_FUNC_SUSPEND_LP (1 << (8 + 0))
- #define USB_INTRF_FUNC_SUSPEND_RW (1 << (8 + 1))
- #define USB_INTRF_STAT_FUNC_RW_CAP 1
- #define USB_INTRF_STAT_FUNC_RW 2
- #define USB_ENDPOINT_HALT 0 /* IN/OUT will STALL */
- #define USB_DEV_STAT_U1_ENABLED 2 /* transition into U1 state */
- #define USB_DEV_STAT_U2_ENABLED 3 /* transition into U2 state */
- #define USB_DEV_STAT_LTM_ENABLED 4 /* Latency tolerance messages */
- #define USB_DEVICE_BATTERY_WAKE_MASK 40
- #define USB_DEVICE_OS_IS_PD_AWARE 41
- #define USB_DEVICE_POLICY_MODE 42
- #define USB_PORT_PR_SWAP 43
- #define USB_PORT_GOTO_MIN 44
- #define USB_PORT_RETURN_POWER 45
- #define USB_PORT_ACCEPT_PD_REQUEST 46
- #define USB_PORT_REJECT_PD_REQUEST 47
- #define USB_PORT_PORT_PD_RESET 48
- #define USB_PORT_C_PORT_PD_CHANGE 49
- #define USB_PORT_CABLE_PD_RESET 50
- #define USB_DEVICE_CHARGING_POLICY 54
- #define USB_DT_DEVICE 0x01
- #define USB_DT_CONFIG 0x02
- #define USB_DT_STRING 0x03
- #define USB_DT_INTERFACE 0x04
- #define USB_DT_ENDPOINT 0x05
- #define USB_DT_DEVICE_QUALIFIER 0x06
- #define USB_DT_OTHER_SPEED_CONFIG 0x07
- #define USB_DT_INTERFACE_POWER 0x08
- #define USB_DT_OTG 0x09
- #define USB_DT_DEBUG 0x0a
- #define USB_DT_INTERFACE_ASSOCIATION 0x0b
- #define USB_DT_SECURITY 0x0c
- #define USB_DT_KEY 0x0d
- #define USB_DT_ENCRYPTION_TYPE 0x0e
- #define USB_DT_BOS 0x0f
- #define USB_DT_DEVICE_CAPABILITY 0x10
- #define USB_DT_WIRELESS_ENDPOINT_COMP 0x11
- #define USB_DT_WIRE_ADAPTER 0x21

Kneron Confidential

- #define USB_DT_RPIPE 0x22
- #define USB_DT_CS_RADIO_CONTROL 0x23
- #define USB_DT_PIPE_USAGE 0x24
- #define USB_DT_SS_ENDPOINT_COMP 0x30
- #define USB_DT_SSP_ISOC_ENDPOINT_COMP 0x31
- #define USB_DT_CS_DEVICE (USB_TYPE_CLASS | USB_DT_DEVICE)
- #define USB_DT_CS_CONFIG (USB_TYPE_CLASS | USB_DT_CONFIG)
- #define USB_DT_CS_STRING (USB_TYPE_CLASS | USB_DT_STRING)
- #define USB_DT_CS_INTERFACE (USB_TYPE_CLASS | USB_DT_INTERFACE)
- #define USB_DT_CS_ENDPOINT (USB_TYPE_CLASS | USB_DT_ENDPOINT)
- #define USB_DT_DEVICE_SIZE 18
- #define USB_CLASS_PER_INTERFACE 0 /* for DeviceClass */
- #define USB_CLASS_AUDIO 1
- #define USB_CLASS_COMM 2
- #define USB_CLASS_HID 3
- #define USB_CLASS_PHYSICAL 5
- #define USB_CLASS_STILL_IMAGE 6
- #define USB_CLASS_PRINTER 7
- #define USB_CLASS_MASS_STORAGE 8
- #define USB_CLASS_HUB 9
- #define USB_CLASS_CDC_DATA 0x0a
- #define USB_CLASS_CSCID 0x0b /* chip+ smart card */
- #define USB_CLASS_CONTENT_SEC 0x0d /* content security */
- #define USB_CLASS_VIDEO 0x0e
- #define USB_CLASS_WIRELESS_CONTROLLER 0xe0
- #define USB_CLASS_MISC 0xef
- #define USB_CLASS_APP_SPEC 0xfe
- #define USB_CLASS_VENDOR_SPEC 0xff
- #define USB_SUBCLASS_VENDOR_SPEC 0xff
- #define USB_DT_CONFIG_SIZE 9
- #define USB_CONFIG_ATT_ONE (1 << 7) /* must be set */
- #define USB_CONFIG_ATT_SELFPOWER (1 << 6) /* self powered */
- #define USB_CONFIG_ATT_WAKEUP (1 << 5) /* can wakeup */
- #define USB_CONFIG_ATT_BATTERY (1 << 4) /* battery powered */
- #define USB_DT_INTERFACE_SIZE 9
- #define USB_DT_ENDPOINT_SIZE 7
- #define USB_DT_ENDPOINT_AUDIO_SIZE 9 /* Audio extension */
- #define USB_ENDPOINT_NUMBER_MASK 0x0f /* in bEndpointAddress */
- #define USB_ENDPOINT_DIR_MASK 0x80
- #define USB_ENDPOINT_XFERTYPE_MASK 0x03 /* in bmAttributes */
- #define USB_ENDPOINT_XFER_CONTROL 0
- #define USB_ENDPOINT_XFER_ISOC 1
- #define USB_ENDPOINT_XFER_BULK 2
- #define USB_ENDPOINT_XFER_INT 3
- #define USB_ENDPOINT_MAX_ADJUSTABLE 0x80
- #define USB_ENDPOINT_INTRTYPE 0x30
- #define USB_ENDPOINT_INTR_PERIODIC (0 << 4)
- #define USB_ENDPOINT_INTR_NOTIFICATION (1 << 4)
- #define USB_DT_SSP_ISOC_EP_COMP_SIZE 8
- #define USB_DT_SS_EP_COMP_SIZE 6
- #define USB_SS_MULT(p) (1 + ((p) & 0x3))
- #define USB_SS_SSP_ISOC_COMP(p) ((p) & (1 << 7))
- #define USB_OTG_SRP (1 << 0)
- #define USB_OTG_HNP (1 << 1) /* swap host/device roles */
- #define USB_OTG_ADP (1 << 2) /* support ADP */

Kneron Confidential

- #define OTG_STS_SELECTOR 0xF000 /* OTG status selector */
- #define USB_ENC_TYPE_UNSECURE 0
- #define USB_ENC_TYPE_WIRED 1 /* non-wireless mode */
- #define USB_ENC_TYPE_CCM_1 2 /* aes128/cbc session */
- #define USB_ENC_TYPE_RSA_1 3 /* rsa3072/sha1 auth */
- #define USB_DT_BOS_SIZE 5
- #define USB3_LPM_DISABLED 0x0
- #define USB3_LPM_U1_MAX_TIMEOUT 0x7F
- #define USB3_LPM_U2_MAX_TIMEOUT 0xFE
- #define USB3_LPM_DEVICE_INITIATED 0xFF
- #define USB3_LPM_MAX_U1_SEL_PEL 0xFF
- #define USB3_LPM_MAX_U2_SEL_PEL 0xFFFF
- #define USB_SELF_POWER_VBUS_MAX_DRAW 100

Enumerations

- enum usb_device_speed { USB_SPEED_UNKNOWN = 0, USB_SPEED_LOW1, USB_SPEED_FULL1, USB_SPEED_HIGH1, USB_SPEED_WIRELESS, USB_SPEED_SUPER, USB_SPEED_SUPER_PLUS }
- enum usb_device_state { USB_STATE_NOTATTACHED = 0, USB_STATE_ATTACHED, USB_STATE_POWERED, USB_STATE_RECONNECTING, USB_STATE_UNAUTHENTICATED, USB_STATE_DEFAULT, USB_STATE_ADDRESS, USB_STATE_CONFIGURED, USB_STATE_SUSPENDED }
- enum usb3_link_state { USB3_LPM_U0 = 0, USB3_LPM_U1, USB3_LPM_U2, USB3_LPM_U3 }

Functions

- struct usb_ctrlrequest __attribute__ ((packed))

Variables

- uint8_t bRequestType
- uint8_t bRequest
- uint16_t wValue
- uint16_t wIndex
- uint16_t wLength
- uint8_t bLength
- uint8_t bDescriptorType
- uint16_t bcdUSB
- uint8_t bDeviceClass
- uint8_t bDeviceSubClass
- uint8_t bDeviceProtocol
- uint8_t bMaxPacketSize0
- uint16_t idVendor
- uint16_t idProduct
- uint16_t bcdDevice
- uint8_t iManufacturer
- uint8_t iProduct
- uint8_t iSerialNumber
- uint8_t bNumConfigurations
- uint8_t bReserved
- uint16_t wTotalLength

Kneron Confidential

- uint8_t bNumInterfaces
- uint8_t bConfigurationValue
- uint8_t iConfiguration
- uint8_t bmAttributes
- uint8_t bMaxPower
- uint16_t bcdOTG
- uint16_t wData [1]
- uint8_t bInterfaceNumber
- uint8_t bAlternateSetting
- uint8_t bNumEndpoints
- uint8_t bInterfaceClass
- uint8_t bInterfaceSubClass
- uint8_t bInterfaceProtocol
- uint8_t iInterface
- uint8_t bEndpointAddress
- uint16_t wMaxPacketSize
- uint8_t bInterval
- uint8_t bRefresh
- uint8_t bSynchAddress
- uint16_t wReseved
- uint32_t dwBytesPerInterval
- uint8_t bMaxBurst
- uint16_t wBytesPerInterval
- uint8_t bRESERVED
- uint8_t bDebugInEndpoint
- uint8_t bDebugOutEndpoint
- uint8_t bFirstInterface
- uint8_t bInterfaceCount
- uint8_t bFunctionClass
- uint8_t bFunctionSubClass
- uint8_t bFunctionProtocol
- uint8_t iFunction
- uint8_t bNumEncryptionTypes
- uint8_t tTKID [3]
- uint8_t bKeyData [0]
- uint8_t bEncryptionType
- uint8_t bEncryptionValue
- uint8_t bAuthKeyIndex
- uint8_t bNumDeviceCaps
- enum usb_device_speed __attribute__
- uint8_t u1_sel
- uint8_t u1_pel
- uint16_t u2_sel
- uint16_t u2_pel

Kneron Confidential

6.24.1 Macro Definition Documentation

6.24.1.1 OTG_STS_SELECTOR

```
#define OTG_STS_SELECTOR 0xF000 /* OTG status selector */
```

6.24.1.2 TEST_FORCE_EN

```
#define TEST_FORCE_EN 5
```

6.24.1.3 TEST_J

```
#define TEST_J 1
```

6.24.1.4 TEST_K

```
#define TEST_K 2
```

6.24.1.5 TEST_PACKET

```
#define TEST_PACKET 4
```

6.24.1.6 TEST_SE0_NAK

```
#define TEST_SE0_NAK 3
```

6.24.1.7 USB3_LPM_DEVICE_INITIATED

```
#define USB3_LPM_DEVICE_INITIATED 0xFF
```

6.24.1.8 USB3_LPM_DISABLED

```
#define USB3_LPM_DISABLED 0x0
```

6.24.1.9 USB3_LPM_MAX_U1_SEL_PEL

```
#define USB3_LPM_MAX_U1_SEL_PEL 0xFF
```

Kneron Confidential

6.24.1.10 USB3_LPM_MAX_U2_SEL_PEL

```
#define USB3_LPM_MAX_U2_SEL_PEL 0xFFFF
```

6.24.1.11 USB3_LPM_U1_MAX_TIMEOUT

```
#define USB3_LPM_U1_MAX_TIMEOUT 0x7F
```

6.24.1.12 USB3_LPM_U2_MAX_TIMEOUT

```
#define USB3_LPM_U2_MAX_TIMEOUT 0xFE
```

6.24.1.13 USB_CLASS_APP_SPEC

```
#define USB_CLASS_APP_SPEC 0xfe
```

6.24.1.14 USB_CLASS_AUDIO

```
#define USB_CLASS_AUDIO 1
```

6.24.1.15 USB_CLASS_CDC_DATA

```
#define USB_CLASS_CDC_DATA 0x0a
```

6.24.1.16 USB_CLASS_COMM

```
#define USB_CLASS_COMM 2
```

6.24.1.17 USB_CLASS_CONTENT_SEC

```
#define USB_CLASS_CONTENT_SEC 0x0d /* content security */
```

Kneron Confidential

6.24.1.18 USB_CLASS_CSCID

```
#define USB_CLASS_CSCID 0x0b /* chip+ smart card */
```

6.24.1.19 USB_CLASS_HID

```
#define USB_CLASS_HID 3
```

6.24.1.20 USB_CLASS_HUB

```
#define USB_CLASS_HUB 9
```

6.24.1.21 USB_CLASS_MASS_STORAGE

```
#define USB_CLASS_MASS_STORAGE 8
```

6.24.1.22 USB_CLASS_MISC

```
#define USB_CLASS_MISC 0xef
```

6.24.1.23 USB_CLASS_PER_INTERFACE

```
#define USB_CLASS_PER_INTERFACE 0 /* for DeviceClass */
```

6.24.1.24 USB_CLASS_PHYSICAL

```
#define USB_CLASS_PHYSICAL 5
```

6.24.1.25 USB_CLASS_PRINTER

```
#define USB_CLASS_PRINTER 7
```

Kneron Confidential

6.24.1.26 USB_CLASS_STILL_IMAGE

```
#define USB_CLASS_STILL_IMAGE 6
```

6.24.1.27 USB_CLASS_VENDOR_SPEC

```
#define USB_CLASS_VENDOR_SPEC 0xff
```

6.24.1.28 USB_CLASS_VIDEO

```
#define USB_CLASS_VIDEO 0x0e
```

6.24.1.29 USB_CLASS_WIRELESS_CONTROLLER

```
#define USB_CLASS_WIRELESS_CONTROLLER 0xe0
```

6.24.1.30 USB_CONFIG_ATT_BATTERY

```
#define USB_CONFIG_ATT_BATTERY (1 << 4) /* battery powered */
```

6.24.1.31 USB_CONFIG_ATT_ONE

```
#define USB_CONFIG_ATT_ONE (1 << 7) /* must be set */
```

6.24.1.32 USB_CONFIG_ATT_SELFPOWER

```
#define USB_CONFIG_ATT_SELFPOWER (1 << 6) /* self powered */
```

6.24.1.33 USB_CONFIG_ATT_WAKEUP

```
#define USB_CONFIG_ATT_WAKEUP (1 << 5) /* can wakeup */
```

Kneron Confidential

6.24.1.34 USB_DEV_STAT_LTM_ENABLED

```
#define USB_DEV_STAT_LTM_ENABLED 4 /* Latency tolerance messages */
```

6.24.1.35 USB_DEV_STAT_U1_ENABLED

```
#define USB_DEV_STAT_U1_ENABLED 2 /* transition into U1 state */
```

6.24.1.36 USB_DEV_STAT_U2_ENABLED

```
#define USB_DEV_STAT_U2_ENABLED 3 /* transition into U2 state */
```

6.24.1.37 USB_DEVICE_A_ALT_HNP_SUPPORT

```
#define USB_DEVICE_A_ALT_HNP_SUPPORT 5 /* (otg) other RH port does */
```

6.24.1.38 USB_DEVICE_A_HNP_SUPPORT

```
#define USB_DEVICE_A_HNP_SUPPORT 4 /* (otg) RH port supports HNP */
```

6.24.1.39 USB_DEVICE_B_HNP_ENABLE

```
#define USB_DEVICE_B_HNP_ENABLE 3 /* (otg) dev may initiate HNP */
```

6.24.1.40 USB_DEVICE_BATTERY

```
#define USB_DEVICE_BATTERY 2 /* (wireless) */
```

6.24.1.41 USB_DEVICE_BATTERY_WAKE_MASK

```
#define USB_DEVICE_BATTERY_WAKE_MASK 40
```

Kneron Confidential

6.24.1.42 USB_DEVICE_CHARGING_POLICY

```
#define USB_DEVICE_CHARGING_POLICY 54
```

6.24.1.43 USB_DEVICE_DEBUG_MODE

```
#define USB_DEVICE_DEBUG_MODE 6 /* (special devices only) */
```

6.24.1.44 USB_DEVICE_LTM_ENABLE

```
#define USB_DEVICE_LTM_ENABLE 50 /* dev may send LTM */
```

6.24.1.45 USB_DEVICE_OS_IS_PD_AWARE

```
#define USB_DEVICE_OS_IS_PD_AWARE 41
```

6.24.1.46 USB_DEVICE_POLICY_MODE

```
#define USB_DEVICE_POLICY_MODE 42
```

6.24.1.47 USB_DEVICE_REMOTE_WAKEUP

```
#define USB_DEVICE_REMOTE_WAKEUP 1 /* dev may initiate wakeup */
```

6.24.1.48 USB_DEVICE_SELF_POWERED

```
#define USB_DEVICE_SELF_POWERED 0 /* (read only) */
```

6.24.1.49 USB_DEVICE_TEST_MODE

```
#define USB_DEVICE_TEST_MODE 2 /* (wired high speed only) */
```

Kneron Confidential

6.24.1.50 USB_DEVICE_U1_ENABLE

```
#define USB_DEVICE_U1_ENABLE 48 /* dev may initiate U1 transition */
```

6.24.1.51 USB_DEVICE_U2_ENABLE

```
#define USB_DEVICE_U2_ENABLE 49 /* dev may initiate U2 transition */
```

6.24.1.52 USB_DEVICE_WUSB_DEVICE

```
#define USB_DEVICE_WUSB_DEVICE 3 /* (wireless)*/
```

6.24.1.53 USB_DIR_IN

```
#define USB_DIR_IN 0x80 /* to host */
```

6.24.1.54 USB_DIR_OUT

```
#define USB_DIR_OUT 0x0 /* to device */
```

6.24.1.55 USB_DT_BOS

```
#define USB_DT_BOS 0x0f
```

6.24.1.56 USB_DT_BOS_SIZE

```
#define USB_DT_BOS_SIZE 5
```

6.24.1.57 USB_DT_CONFIG

```
#define USB_DT_CONFIG 0x02
```

Kneron Confidential

6.24.1.58 USB_DT_CONFIG_SIZE

```
#define USB_DT_CONFIG_SIZE 9
```

6.24.1.59 USB_DT_CS_CONFIG

```
#define USB_DT_CS_CONFIG (USB_TYPE_CLASS | USB_DT_CONFIG)
```

6.24.1.60 USB_DT_CS_DEVICE

```
#define USB_DT_CS_DEVICE (USB_TYPE_CLASS | USB_DT_DEVICE)
```

6.24.1.61 USB_DT_CS_ENDPOINT

```
#define USB_DT_CS_ENDPOINT (USB_TYPE_CLASS | USB_DT_ENDPOINT)
```

6.24.1.62 USB_DT_CS_INTERFACE

```
#define USB_DT_CS_INTERFACE (USB_TYPE_CLASS | USB_DT_INTERFACE)
```

6.24.1.63 USB_DT_CS_RADIO_CONTROL

```
#define USB_DT_CS_RADIO_CONTROL 0x23
```

6.24.1.64 USB_DT_CS_STRING

```
#define USB_DT_CS_STRING (USB_TYPE_CLASS | USB_DT_STRING)
```

6.24.1.65 USB_DT_DEBUG

```
#define USB_DT_DEBUG 0x0a
```

Kneron Confidential

6.24.1.66 USB_DT_DEVICE

```
#define USB_DT_DEVICE 0x01
```

6.24.1.67 USB_DT_DEVICE_CAPABILITY

```
#define USB_DT_DEVICE_CAPABILITY 0x10
```

6.24.1.68 USB_DT_DEVICE_QUALIFIER

```
#define USB_DT_DEVICE_QUALIFIER 0x06
```

6.24.1.69 USB_DT_DEVICE_SIZE

```
#define USB_DT_DEVICE_SIZE 18
```

6.24.1.70 USB_DT_ENCRYPTION_TYPE

```
#define USB_DT_ENCRYPTION_TYPE 0x0e
```

6.24.1.71 USB_DT_ENDPOINT

```
#define USB_DT_ENDPOINT 0x05
```

6.24.1.72 USB_DT_ENDPOINT_AUDIO_SIZE

```
#define USB_DT_ENDPOINT_AUDIO_SIZE 9 /* Audio extension */
```

6.24.1.73 USB_DT_ENDPOINT_SIZE

```
#define USB_DT_ENDPOINT_SIZE 7
```

Kneron Confidential

6.24.1.74 USB_DT_INTERFACE

```
#define USB_DT_INTERFACE 0x04
```

6.24.1.75 USB_DT_INTERFACE_ASSOCIATION

```
#define USB_DT_INTERFACE_ASSOCIATION 0x0b
```

6.24.1.76 USB_DT_INTERFACE_POWER

```
#define USB_DT_INTERFACE_POWER 0x08
```

6.24.1.77 USB_DT_INTERFACE_SIZE

```
#define USB_DT_INTERFACE_SIZE 9
```

6.24.1.78 USB_DT_KEY

```
#define USB_DT_KEY 0x0d
```

6.24.1.79 USB_DT_OTG

```
#define USB_DT_OTG 0x09
```

6.24.1.80 USB_DT_OTHER_SPEED_CONFIG

```
#define USB_DT_OTHER_SPEED_CONFIG 0x07
```

6.24.1.81 USB_DT_PIPE_USAGE

```
#define USB_DT_PIPE_USAGE 0x24
```

Kneron Confidential

6.24.1.82 USB_DT_RPIPE

```
#define USB_DT_RPIPE 0x22
```

6.24.1.83 USB_DT_SECURITY

```
#define USB_DT_SECURITY 0x0c
```

6.24.1.84 USB_DT_SS_ENDPOINT_COMP

```
#define USB_DT_SS_ENDPOINT_COMP 0x30
```

6.24.1.85 USB_DT_SS_EP_COMP_SIZE

```
#define USB_DT_SS_EP_COMP_SIZE 6
```

6.24.1.86 USB_DT_SSP_ISOC_ENDPOINT_COMP

```
#define USB_DT_SSP_ISOC_ENDPOINT_COMP 0x31
```

6.24.1.87 USB_DT_SSP_ISOC_EP_COMP_SIZE

```
#define USB_DT_SSP_ISOC_EP_COMP_SIZE 8
```

6.24.1.88 USB_DT_STRING

```
#define USB_DT_STRING 0x03
```

6.24.1.89 USB_DT_WIRE_ADAPTER

```
#define USB_DT_WIRE_ADAPTER 0x21
```

Kneron Confidential

6.24.1.90 USB_DT_WIRELESS_ENDPOINT_COMP

```
#define USB_DT_WIRELESS_ENDPOINT_COMP 0x11
```

6.24.1.91 USB_ENC_TYPE_CCM_1

```
#define USB_ENC_TYPE_CCM_1 2 /* aes128/cbc session */
```

6.24.1.92 USB_ENC_TYPE_RSA_1

```
#define USB_ENC_TYPE_RSA_1 3 /* rsa3072/sha1 auth */
```

6.24.1.93 USB_ENC_TYPE_UNSECURE

```
#define USB_ENC_TYPE_UNSECURE 0
```

6.24.1.94 USB_ENC_TYPE_WIRED

```
#define USB_ENC_TYPE_WIRED 1 /* non-wireless mode */
```

6.24.1.95 USB_ENDPOINT_DIR_MASK

```
#define USB_ENDPOINT_DIR_MASK 0x80
```

6.24.1.96 USB_ENDPOINT_HALT

```
#define USB_ENDPOINT_HALT 0 /* IN/OUT will STALL */
```

6.24.1.97 USB_ENDPOINT_INTR_NOTIFICATION

```
#define USB_ENDPOINT_INTR_NOTIFICATION (1 << 4)
```

Kneron Confidential

6.24.1.98 USB_ENDPOINT_INTR_PERIODIC

```
#define USB_ENDPOINT_INTR_PERIODIC (0 << 4)
```

6.24.1.99 USB_ENDPOINT_INTRTYPE

```
#define USB_ENDPOINT_INTRTYPE 0x30
```

6.24.1.100 USB_ENDPOINT_MAX_ADJUSTABLE

```
#define USB_ENDPOINT_MAX_ADJUSTABLE 0x80
```

6.24.1.101 USB_ENDPOINT_NUMBER_MASK

```
#define USB_ENDPOINT_NUMBER_MASK 0x0f /* in bEndpointAddress */
```

6.24.1.102 USB_ENDPOINT_XFER_BULK

```
#define USB_ENDPOINT_XFER_BULK 2
```

6.24.1.103 USB_ENDPOINT_XFER_CONTROL

```
#define USB_ENDPOINT_XFER_CONTROL 0
```

6.24.1.104 USB_ENDPOINT_XFER_INT

```
#define USB_ENDPOINT_XFER_INT 3
```

6.24.1.105 USB_ENDPOINT_XFER_ISOC

```
#define USB_ENDPOINT_XFER_ISOC 1
```

Kneron Confidential

6.24.1.106 USB_ENDPOINT_XFERTYPE_MASK

```
#define USB_ENDPOINT_XFERTYPE_MASK 0x03 /* in bmAttributes */
```

6.24.1.107 USB_INTR_FUNC_SUSPEND_OPT_MASK

```
#define USB_INTR_FUNC_SUSPEND_OPT_MASK 0xFF00
```

6.24.1.108 USB_INTRF_FUNC_SUSPEND

```
#define USB_INTRF_FUNC_SUSPEND 0 /* function suspend */
```

6.24.1.109 USB_INTRF_FUNC_SUSPEND_LP

```
#define USB_INTRF_FUNC_SUSPEND_LP (1 << (8 + 0))
```

6.24.1.110 USB_INTRF_FUNC_SUSPEND_RW

```
#define USB_INTRF_FUNC_SUSPEND_RW (1 << (8 + 1))
```

6.24.1.111 USB_INTRF_STAT_FUNC_RW

```
#define USB_INTRF_STAT_FUNC_RW 2
```

6.24.1.112 USB_INTRF_STAT_FUNC_RW_CAP

```
#define USB_INTRF_STAT_FUNC_RW_CAP 1
```

6.24.1.113 USB_OTG_ADP

```
#define USB_OTG_ADP (1 << 2) /* support ADP */
```

Kneron Confidential

6.24.1.114 USB_OTG_HNP

```
#define USB_OTG_HNP (1 << 1) /* swap host/device roles */
```

6.24.1.115 USB_OTG_SRP

```
#define USB_OTG_SRP (1 << 0)
```

6.24.1.116 USB_PORT_ACCEPT_PD_REQUEST

```
#define USB_PORT_ACCEPT_PD_REQUEST 46
```

6.24.1.117 USB_PORT_C_PORT_PD_CHANGE

```
#define USB_PORT_C_PORT_PD_CHANGE 49
```

6.24.1.118 USB_PORT_CABLE_PD_RESET

```
#define USB_PORT_CABLE_PD_RESET 50
```

6.24.1.119 USB_PORT_GOTO_MIN

```
#define USB_PORT_GOTO_MIN 44
```

6.24.1.120 USB_PORT_PORT_PD_RESET

```
#define USB_PORT_PORT_PD_RESET 48
```

6.24.1.121 USB_PORT_PR_SWAP

```
#define USB_PORT_PR_SWAP 43
```

Kneron Confidential

6.24.1.122 USB_PORT_REJECT_PD_REQUEST

```
#define USB_PORT_REJECT_PD_REQUEST 47
```

6.24.1.123 USB_PORT_RETURN_POWER

```
#define USB_PORT_RETURN_POWER 45
```

6.24.1.124 USB_RECIP_DEVICE

```
#define USB_RECIP_DEVICE 0x00
```

6.24.1.125 USB_RECIP_ENDPOINT

```
#define USB_RECIP_ENDPOINT 0x02
```

6.24.1.126 USB_RECIP_INTERFACE

```
#define USB_RECIP_INTERFACE 0x01
```

6.24.1.127 USB_RECIP_MASK

```
#define USB_RECIP_MASK 0x1f
```

6.24.1.128 USB_RECIP_OTHER

```
#define USB_RECIP_OTHER 0x03
```

6.24.1.129 USB_RECIP_PORT

```
#define USB_RECIP_PORT 0x04
```

Kneron Confidential

6.24.1.130 USB_RECIP_RPIPE

```
#define USB_RECIP_RPIPE 0x05
```

6.24.1.131 USB_REQ_CLEAR_FEATURE

```
#define USB_REQ_CLEAR_FEATURE 0x01
```

6.24.1.132 USB_REQ_GET_BATTERY_STATUS

```
#define USB_REQ_GET_BATTERY_STATUS 21
```

6.24.1.133 USB_REQ_GET_CONFIGURATION

```
#define USB_REQ_GET_CONFIGURATION 0x08
```

6.24.1.134 USB_REQ_GET_DESCRIPTOR

```
#define USB_REQ_GET_DESCRIPTOR 0x06
```

6.24.1.135 USB_REQ_GET_ENCRYPTION

```
#define USB_REQ_GET_ENCRYPTION 0x0E
```

6.24.1.136 USB_REQ_GET_HANDSHAKE

```
#define USB_REQ_GET_HANDSHAKE 0x10
```

6.24.1.137 USB_REQ_GET_IDLE

```
#define USB_REQ_GET_IDLE 0x02
```

Kneron Confidential

6.24.1.138 USB_REQ_GET_INTERFACE

```
#define USB_REQ_GET_INTERFACE 0x0A
```

6.24.1.139 USB_REQ_GET_PARTNER_PDO

```
#define USB_REQ_GET_PARTNER_PDO 20
```

6.24.1.140 USB_REQ_GET_PROTOCOL

```
#define USB_REQ_GET_PROTOCOL 0x03
```

6.24.1.141 USB_REQ_GET_REPORT

```
#define USB_REQ_GET_REPORT 0x01
```

6.24.1.142 USB_REQ_GET_SECURITY_DATA

```
#define USB_REQ_GET_SECURITY_DATA 0x13
```

6.24.1.143 USB_REQ_GET_STATUS

```
#define USB_REQ_GET_STATUS 0x00
```

6.24.1.144 USB_REQ_GET_VDM

```
#define USB_REQ_GET_VDM 23
```

6.24.1.145 USB_REQ_LOOPBACK_DATA_READ

```
#define USB_REQ_LOOPBACK_DATA_READ 0x16
```

Kneron Confidential

6.24.1.146 USB_REQ_LOOPBACK_DATA_WRITE

```
#define USB_REQ_LOOPBACK_DATA_WRITE 0x15
```

6.24.1.147 USB_REQ_RPIPE_ABORT

```
#define USB_REQ_RPIPE_ABORT 0x0E
```

6.24.1.148 USB_REQ_RPIPE_RESET

```
#define USB_REQ_RPIPE_RESET 0x0F
```

6.24.1.149 USB_REQ_SEND_VDM

```
#define USB_REQ_SEND_VDM 24
```

6.24.1.150 USB_REQ_SET_ADDRESS

```
#define USB_REQ_SET_ADDRESS 0x05
```

6.24.1.151 USB_REQ_SET_CONFIGURATION

```
#define USB_REQ_SET_CONFIGURATION 0x09
```

6.24.1.152 USB_REQ_SET_CONNECTION

```
#define USB_REQ_SET_CONNECTION 0x11
```

6.24.1.153 USB_REQ_SET_DESCRIPTOR

```
#define USB_REQ_SET_DESCRIPTOR 0x07
```

Kneron Confidential

6.24.1.154 USB_REQ_SET_ENCRYPTION

```
#define USB_REQ_SET_ENCRYPTION 0x0D /* Wireless USB */
```

6.24.1.155 USB_REQ_SET_FEATURE

```
#define USB_REQ_SET_FEATURE 0x03
```

6.24.1.156 USB_REQ_SET_HANDSHAKE

```
#define USB_REQ_SET_HANDSHAKE 0x0F
```

6.24.1.157 USB_REQ_SET_IDLE

```
#define USB_REQ_SET_IDLE 0x0A
```

6.24.1.158 USB_REQ_SET_INTERFACE

```
#define USB_REQ_SET_INTERFACE 0x0B
```

6.24.1.159 USB_REQ_SET_INTERFACE_DS

```
#define USB_REQ_SET_INTERFACE_DS 0x17
```

6.24.1.160 USB_REQ_SET_ISOCH_DELAY

```
#define USB_REQ_SET_ISOCH_DELAY 0x31
```

6.24.1.161 USB_REQ_SET_PDO

```
#define USB_REQ_SET_PDO 22
```

Kneron Confidential

6.24.1.162 USB_REQ_SET_PROTOCOL

```
#define USB_REQ_SET_PROTOCOL 0x0B
```

6.24.1.163 USB_REQ_SET_REPORT

```
#define USB_REQ_SET_REPORT 0x09
```

6.24.1.164 USB_REQ_SET_SECURITY_DATA

```
#define USB_REQ_SET_SECURITY_DATA 0x12
```

6.24.1.165 USB_REQ_SET_SEL

```
#define USB_REQ_SET_SEL 0x30
```

6.24.1.166 USB_REQ_SET_WUSB_DATA

```
#define USB_REQ_SET_WUSB_DATA 0x14
```

6.24.1.167 USB_REQ_SYNCH_FRAME

```
#define USB_REQ_SYNCH_FRAME 0x0C
```

6.24.1.168 USB_SELF_POWER_VBUS_MAX_DRAW

```
#define USB_SELF_POWER_VBUS_MAX_DRAW 100
```

6.24.1.169 USB_SS_MULT

```
#define USB_SS_MULT(
             p ) (1 + ((p) & 0x3))
```

Kneron Confidential

6.24.1.170 USB_SS_SSP_ISOC_COMP

```
#define USB_SS_SSP_ISOC_COMP(
              p ) ((p) & (1 << 7))
```

6.24.1.171 USB_SUBCLASS_VENDOR_SPEC

```
#define USB_SUBCLASS_VENDOR_SPEC 0xff
```

6.24.1.172 USB_TYPE_CLASS

```
#define USB_TYPE_CLASS (0x01 << 5)
```

6.24.1.173 USB_TYPE_MASK

```
#define USB_TYPE_MASK (0x03 << 5)
```

6.24.1.174 USB_TYPE_RESERVED

```
#define USB_TYPE_RESERVED (0x03 << 5)
```

6.24.1.175 USB_TYPE_STANDARD

```
#define USB_TYPE_STANDARD (0x00 << 5)
```

6.24.1.176 USB_TYPE_VENDOR

```
#define USB_TYPE_VENDOR (0x02 << 5)
```

6.24.2 Enumeration Type Documentation

6.24.2.1 usb3_link_state

```
enum usb3_link_state
```

Kneron Confidential

Enumerator

USB3_LPM_U0	
USB3_LPM_U1	
USB3_LPM_U2	
USB3_LPM_U3	

6.24.2.2 usb_device_speed

`enum usb_device_speed`

Enumerator

USB_SPEED_UNKNOWN	
USB_SPEED_LOW1	
USB_SPEED_FULL1	
USB_SPEED_HIGH1	
USB_SPEED_WIRELESS	
USB_SPEED_SUPER	
USB_SPEED_SUPER_PLUS	

6.24.2.3 usb_device_state

`enum usb_device_state`

Enumerator

USB_STATE_NOTATTACHED	
USB_STATE_ATTACHED	
USB_STATE_POWERED	
USB_STATE_RECONNECTING	
USB_STATE_UNAUTHENTICATED	
USB_STATE_DEFAULT	
USB_STATE_ADDRESS	
USB_STATE_CONFIGURED	
USB_STATE_SUSPENDED	

6.24.3 Function Documentation

Kneron Confidential

6.24.3.1 __attribute__()

```
struct usb_ctrlrequest __attribute__ (
            (packed)  )
```

6.24.4 Variable Documentation

6.24.4.1 __attribute__

```
enum usb_device_speed __attribute__
```

6.24.4.2 bAlternateSetting

```
uint8_t bAlternateSetting
```

6.24.4.3 bAuthKeyIndex

```
uint8_t bAuthKeyIndex
```

6.24.4.4 bcdDevice

```
uint16_t bcdDevice
```

6.24.4.5 bcdOTG

```
uint16_t bcdOTG
```

6.24.4.6 bcdUSB

```
uint16_t bcdUSB
```

Kneron Confidential

6.24.4.7 bConfigurationValue

```
uint8_t bConfigurationValue
```

6.24.4.8 bDebugInEndpoint

```
uint8_t bDebugInEndpoint
```

6.24.4.9 bDebugOutEndpoint

```
uint8_t bDebugOutEndpoint
```

6.24.4.10 bDescriptorType

```
uint8_t bDescriptorType
```

6.24.4.11 bDeviceClass

```
uint8_t bDeviceClass
```

6.24.4.12 bDeviceProtocol

```
uint8_t bDeviceProtocol
```

6.24.4.13 bDeviceSubClass

```
uint8_t bDeviceSubClass
```

6.24.4.14 bEncryptionType

```
uint8_t bEncryptionType
```

Kneron Confidential

6.24.4.15 bEncryptionValue

```
uint8_t bEncryptionValue
```

6.24.4.16 bEndpointAddress

```
uint8_t bEndpointAddress
```

6.24.4.17 bFirstInterface

```
uint8_t bFirstInterface
```

6.24.4.18 bFunctionClass

```
uint8_t bFunctionClass
```

6.24.4.19 bFunctionProtocol

```
uint8_t bFunctionProtocol
```

6.24.4.20 bFunctionSubClass

```
uint8_t bFunctionSubClass
```

6.24.4.21 bInterfaceClass

```
uint8_t bInterfaceClass
```

6.24.4.22 bInterfaceCount

```
uint8_t bInterfaceCount
```

Kneron Confidential

6.24.4.23 bInterfaceNumber

```
uint8_t bInterfaceNumber
```

6.24.4.24 bInterfaceProtocol

```
uint8_t bInterfaceProtocol
```

6.24.4.25 bInterfaceSubClass

```
uint8_t bInterfaceSubClass
```

6.24.4.26 bInterval

```
uint8_t bInterval
```

6.24.4.27 bKeyData

```
uint8_t bKeyData[0]
```

6.24.4.28 bLength

```
uint8_t bLength
```

6.24.4.29 bmAttributes

```
uint8_t bmAttributes
```

6.24.4.30 bMaxBurst

```
uint8_t bMaxBurst
```

Kneron Confidential

6.24.4.31 bMaxPacketSize0

```
uint8_t bMaxPacketSize0
```

6.24.4.32 bMaxPower

```
uint8_t bMaxPower
```

6.24.4.33 bNumConfigurations

```
uint8_t bNumConfigurations
```

6.24.4.34 bNumDeviceCaps

```
uint8_t bNumDeviceCaps
```

6.24.4.35 bNumEncryptionTypes

```
uint8_t bNumEncryptionTypes
```

6.24.4.36 bNumEndpoints

```
uint8_t bNumEndpoints
```

6.24.4.37 bNumInterfaces

```
uint8_t bNumInterfaces
```

6.24.4.38 bRefresh

```
uint8_t bRefresh
```

Kneron Confidential

6.24.4.39 bRequest

```
uint8_t bRequest
```

6.24.4.40 bRequestType

```
uint8_t bRequestType
```

6.24.4.41 bReserved

```
uint8_t bReserved
```

6.24.4.42 bRESERVED

```
uint8_t bRESERVED
```

6.24.4.43 bSynchAddress

```
uint8_t bSynchAddress
```

6.24.4.44 dwBytesPerInterval

```
uint32_t dwBytesPerInterval
```

6.24.4.45 iConfiguration

```
uint8_t iConfiguration
```

6.24.4.46 idProduct

```
uint16_t idProduct
```

Kneron Confidential

6.24.4.47 idVendor

```
uint16_t idVendor
```

6.24.4.48 iFunction

```
uint8_t iFunction
```

6.24.4.49 iInterface

```
uint8_t iInterface
```

6.24.4.50 iManufacturer

```
uint8_t iManufacturer
```

6.24.4.51 iProduct

```
uint8_t iProduct
```

6.24.4.52 iSerialNumber

```
uint8_t iSerialNumber
```

6.24.4.53 tTKID

```
uint8_t tTKID[3]
```

6.24.4.54 u1_pel

```
uint8_t u1_pel
```

Kneron Confidential

6.24.4.55 u1_sel

```
uint8_t u1_sel
```

6.24.4.56 u2_pel

```
uint16_t u2_pel
```

6.24.4.57 u2_sel

```
uint16_t u2_sel
```

6.24.4.58 wBytesPerInterval

```
uint16_t wBytesPerInterval
```

6.24.4.59 wData

```
uint16_t wData[1]
```

6.24.4.60 wIndex

```
uint16_t wIndex
```

6.24.4.61 wLength

```
uint16_t wLength
```

6.24.4.62 wMaxPacketSize

```
uint16_t wMaxPacketSize
```

Kneron Confidential

6.24.4.63 wReseved

```
uint16_t wReseved
```

6.24.4.64 wTotalLength

```
uint16_t wTotalLength
```

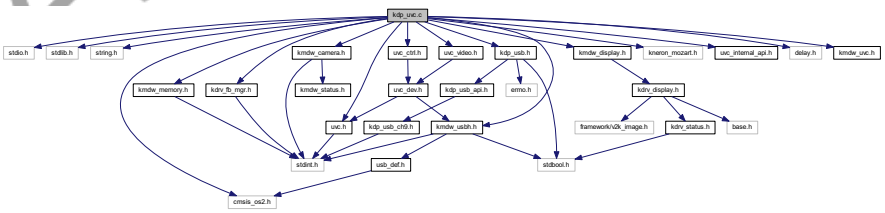
6.24.4.65 wValue

```
uint16_t wValue
```

6.25 kdp_uvc.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <cmsis_os2.h>
#include "kneron_mozart.h"
#include "kdp_usb.h"
#include "uvc.h"
#include <uvc_video.h>
#include <uvc_ctrl.h>
#include "kmdw_memory.h"
#include "uvc_internal_api.h"
#include "kdrv_fb_mgr.h"
#include "kmdw_camera.h"
#include "kmdw_display.h"
#include "delay.h"
#include "kmdw_usbh.h"
#include "kmdw_uvc.h"
```

Include dependency graph for kdp_uvc.c:



Macros

- #define KDP_UVC_DRIVER_DESC "KDP UVC driver"
- #define KDP_UVC_DRIVER 1
- #define MIN(a, b) (((a)<(b))?(a):(b))
- #define MAX(a, b) (((a)>(b))?(a):(b))
- #define ROUND_UP(x, y) ((((x) + (y - 1)) / y) * y)

Kneron Confidential

6.25.1 Macro Definition Documentation

6.25.1.1 KDP_UVC_DRIVER

```
#define KDP_UVC_DRIVER 1
```

6.25.1.2 KDP_UVC_DRIVER_DESC

```
#define KDP_UVC_DRIVER_DESC "KDP UVC driver"
```

6.25.1.3 MAX

```
#define MAX(
            a,
            b ) (((a)>(b))?(a):(b))
```

6.25.1.4 MIN

```
#define MIN(
            a,
            b ) (((a)<(b))?(a):(b))
```

6.25.1.5 ROUND_UP

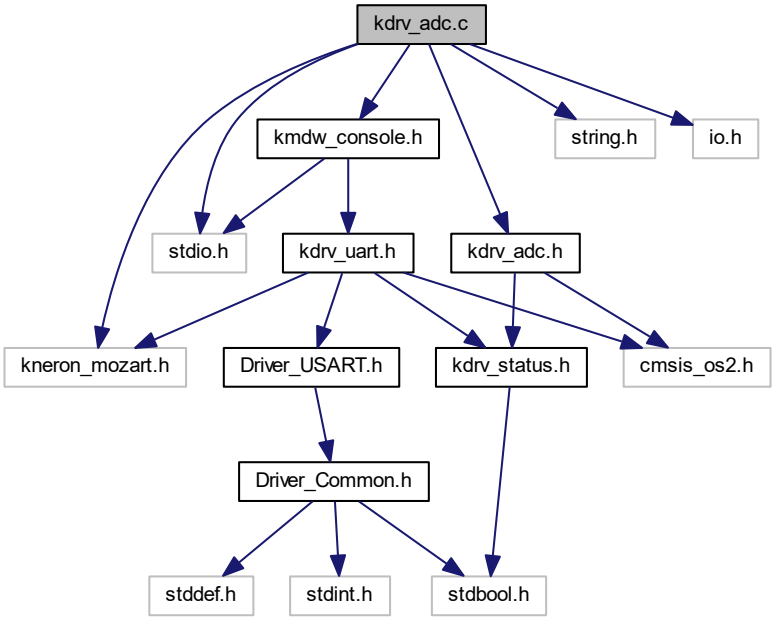
```
#define ROUND_UP(
            x,
            y ) ((((x) + (y - 1)) / y) * y)
```

Kneron Confidential

6.26 kdrv_adc.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "io.h"
#include "kneron_mozart.h"
#include "kdrv_adc.h"
#include "kmdw_console.h"
```

Include dependency graph for kdrv_adc.c:



Macros

- #define NUM_TDC 4
- #define SCANMODE SCANMODE_CONT

Functions

- void _kdrv_adc_config (void)
- kdrv_status_t kdrv_adc_initialize (void)

  ADC driver initialization.
- kdrv_status_t kdrv_adc_uninitialize (kdrv_adc_resource_t *res)

  ADC driver uninitialization.
- kdrv_status_t kdrv_adc_rest (kdrv_adc_resource_t *res)

  ADC reset control.
- kdrv_status_t kdrv_adc_enable (kdrv_adc_resource_t *res, int mode)

  ADC enable control.
- int kdrv_adc_read (int id)

  ADC data read control.
- void _print_adc_register (kdrv_adc_resource_t *res)

Kneron Confidential

Variables

- kdrv_adc_resource_t kdrv_adc_resource

6.26.1 Macro Definition Documentation

6.26.1.1 NUM_TDC

```
#define NUM_TDC 4
```

6.26.1.2 SCANMODE

```
#define SCANMODE SCANMODE_CONT
```

6.26.2 Function Documentation

6.26.2.1 _kdrv_adc_config()

```
void _kdrv_adc_config (
             void  )
```

6.26.2.2 _print_adc_register()

```
void _print_adc_register (
            kdrv_adc_resource_t * res )
```

6.26.3 Variable Documentation

6.26.3.1 kdrv_adc_resource

```
kdrv_adc_resource_t kdrv_adc_resource
```

Initial value:

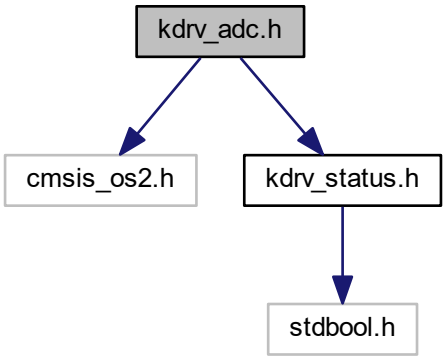
```
= {
    ADC_FTTSC010_0_PA_BASE,
    ADC_FTADCC010_IRQ,
}
```

Kneron Confidential

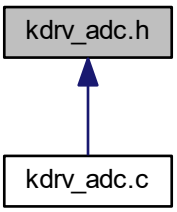
6.27 kdrv_adc.h File Reference

```
#include "cmsis_os2.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_adc.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_adc_resource_t
- struct kdrv_adc_regs_t

Macros

- #define HTHR_EN (1<<31)
- #define HTHR(x) (((x)&0xFFF)<<16)
- #define LTHR_EN (1<<15)
- #define LTHR(x) (((x)&0xFFF)<<0)

- #define SCAN_NUM(x) (x<<16)
- #define SCANMODE_CONT (1<<9)
- #define SCANMODE_SGL (1<<8)
- #define SWSTART (1<<4)
- #define ADC_EN (1<<0)
- #define CHDONE_INTEN(x) (1<<((x)+8))
- #define TS_OVR_INTREN (1<<3)
- #define TS_UDR_INTREN (1<<2)
- #define STOP_INTEN (1<<1)
- #define DONE_INTEN (1<<0)
- #define CH_INTRSTS(x) (1<<((x)+8))
- #define TS_THDOD_INTRSTS (1<<3)
- #define TS_THDUD_INTRSTS (1<<2)
- #define ADC_STOP_INTRSTS (1<<1)
- #define ADC_DONE_INTSTS (1<<0)

Functions

- kdrv_status_t kdrv_adc_initialize (void)

  ADC driver initialization.
- kdrv_status_t kdrv_adc_uninitialize (kdrv_adc_resource_t *res)

  ADC driver uninitialization.
- kdrv_status_t kdrv_adc_rest (kdrv_adc_resource_t *res)

  ADC reset control.
- kdrv_status_t kdrv_adc_enable (kdrv_adc_resource_t *res, int mode)

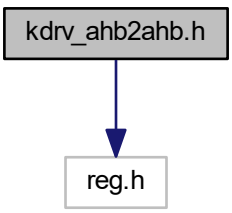
  ADC enable control.
- int kdrv_adc_read (int id)

  ADC data read control.

6.28 kdrv_ahb2ahb.h File Reference

```
#include "reg.h"
```

Include dependency graph for kdrv_ahb2ahb.h:

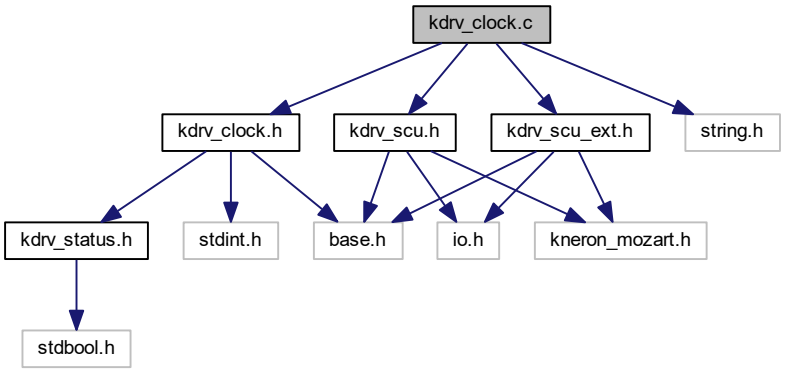


Kneron Confidential

6.29 kdrv_clock.c File Reference

```
#include "kdrv_clock.h"
#include <string.h>
#include "kdrv_scu.h"
#include "kdrv_scu_ext.h"
```

Include dependency graph for kdrv_clock.c:



Macros

- #define CYCLES_PER_LOOP 2
- #define DEBUG_CLOCK(__node, __clock_val)
- #define _clock_ref_add(__node) { ++__node->ref_cnt; }
- #define _clock_ref_sub(__node) { if (__node->ref_cnt > 0) --__node->ref_cnt; }

Functions

- void kdrv_delay_us (uint32_t usec)
- struct kdrv_clock_node * _find_child_head (struct kdrv_clock_node *node)
- struct kdrv_clock_node * _find_sibling_is_enabled (struct kdrv_clock_node *node)
- void kdrv_clock_mgr_init ()
- void kdrv_clock_mgr_open (struct kdrv_clock_node *node, struct kdrv_clock_value *clock_val)
- void kdrv_clock_mgr_close (struct kdrv_clock_node *node)
- void kdrv_clock_mgr_set_scuclkin (enum scuclkin_type type, bool enable)
- void kdrv_clock_mgr_set_muxsel (uint32_t flags)
- uint32_t kdrv_clock_mgr_calculate_clockout (enum pll_id id, uint16_t ms, uint16_t ns, uint16_t F_ps)
- void kdrv_clock_mgr_open_pll1 (void)
- void kdrv_clock_mgr_open_pll2 (void)
- void kdrv_clock_mgr_open_pll3 (void)
- void kdrv_clock_mgr_open_pll4 (void)
- void kdrv_clock_mgr_open_pll5 (void)
- void kdrv_clock_mgr_close_pll1 (void)
- void kdrv_clock_mgr_close_pll2 (void)
- void clock_mgr_close_pll3 (void)
- void kdrv_clock_mgr_close_pll4 (void)

Kneron Confidential

- void clock_mgr_close_pll5 (void)
- void kdrv_clock_mgr_change_pll3_clock (uint32_t ms, uint32_t ns, uint32_t ps, uint32_t csi0_txesc, uint32_t csi0_csi, uint32_t csi0_vc0, uint32_t csi1_txesc, uint32_t csi1_csi, uint32_t csi1_vc0)
- void kdrv_clock_mgr_change_pll5_clock (uint32_t ms, uint32_t ns, uint32_t ps)
- void kdrv_clock_enable (enum clk clk)
- void kdrv_clock_disable (enum clk clk)
- void kdrv_clock_set_csiclk (uint32_t idx, uint32_t enable)
- void kdrv_debug_pll_clock (void)

Variables

- uint32_t SystemCoreClock
- struct kdrv_clock_node clock_node_pll1 = { .name = "pll1" }
- struct kdrv_clock_node clock_node_pll1_out = { .name = "pll1o" }
- struct kdrv_clock_node clock_node_pll2 = { .name = "pll2" }
- struct kdrv_clock_node clock_node_pll2_out = { .name = "pll2o" }
- struct kdrv_clock_node clock_node_pll3 = { .name = "pll3" }
- struct kdrv_clock_node clock_node_pll3_out1 = { .name = "pll3o1" }
- struct kdrv_clock_node clock_node_pll3_out2 = {.name = "pll3o2" }
- struct kdrv_clock_node clock_node_csirx0_hs_csi = { .name = "pll3csirx0csi" }
- struct kdrv_clock_node clock_node_csirx0_hs_vc0 = { .name = "pll3csirx0vc0" }
- struct kdrv_clock_node clock_node_csirx0_lp = { .name = "pll3csirx0lp" }
- struct kdrv_clock_node clock_node_pll4 = { .name = "pll4" }
- struct kdrv_clock_node clock_node_pll4_out1 = { .name = "pll4o1" }
- struct kdrv_clock_node clock_node_pll4_fref_pll0 = { .name = "pll4frefpll0" }
- struct kdrv_clock_node clock_node_pll5 = { .name = "pll5" }
- struct kdrv_clock_node clock_node_pll5_out1 = { .name = "pll5o1" }
- struct kdrv_clock_node clock_node_pll5_out2 = { .name = "pll5o2" }
- struct kdrv_clock_node clock_node_lcdc = { .name = "pll5lcclk" }

6.29.1 Macro Definition Documentation

6.29.1.1 _clock_ref_add

```
#define _clock_ref_add(
        __node ) { ++__node->ref_cnt; }
```

6.29.1.2 _clock_ref_sub

```
#define _clock_ref_sub(
        __node ) { if (__node->ref_cnt > 0) --__node->ref_cnt; }
```

Kneron Confidential

6.29.1.3 CYCLES_PER_LOOP

```
#define CYCLES_PER_LOOP 2
```

6.29.1.4 DEBUG_CLOCK

```
#define DEBUG_CLOCK(
              __node,
              __clock_val )
```

6.29.2 Function Documentation

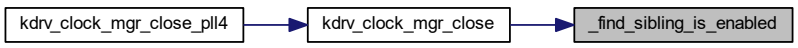
6.29.2.1 _find_child_head()

```
struct kdrv_clock_node* _find_child_head (
             struct kdrv_clock_node * node )  [inline]
```

6.29.2.2 _find_sibling_is_enabled()

```
struct kdrv_clock_node* _find_sibling_is_enabled (
             struct kdrv_clock_node * node )
```

Here is the caller graph for this function:



6.29.2.3 clock_mgr_close_pll3()

```
void clock_mgr_close_pll3 (
             void  )
```

Kneron Confidential

6.29.2.4 clock_mgr_close_pll5()

```
void clock_mgr_close_pll5 (
            void  )
```

6.29.3 Variable Documentation

6.29.3.1 clock_node_csirx0_hs_csi

```
struct kdrv_clock_node clock_node_csirx0_hs_csi = { .name = "pll3csirx0csi" }
```

6.29.3.2 clock_node_csirx0_hs_vc0

```
struct kdrv_clock_node clock_node_csirx0_hs_vc0 = { .name = "pll3csirx0vc0" }
```

6.29.3.3 clock_node_csirx0_lp

```
struct kdrv_clock_node clock_node_csirx0_lp = { .name = "pll3csirx0lp" }
```

6.29.3.4 clock_node_lcdc

```
struct kdrv_clock_node clock_node_lcdc = { .name = "pll5lcclk" }
```

6.29.3.5 clock_node_pll1

```
struct kdrv_clock_node clock_node_pll1 = { .name = "pll1" }
```

6.29.3.6 clock_node_pll2

```
struct kdrv_clock_node clock_node_pll2 = { .name = "pll2" }
```

Kneron Confidential

6.29.3.7 clock_node_pll2_out

```
struct kdrv_clock_node clock_node_pll2_out = { .name = "pll2o" }
```

6.29.3.8 clock_node_pll3

```
struct kdrv_clock_node clock_node_pll3 = { .name = "pll3" }
```

6.29.3.9 clock_node_pll3_out1

```
struct kdrv_clock_node clock_node_pll3_out1 = { .name = "pll3o1" }
```

6.29.3.10 clock_node_pll3_out2

```
struct kdrv_clock_node clock_node_pll3_out2 = {.name = "pll3o2" }
```

6.29.3.11 clock_node_pll4

```
struct kdrv_clock_node clock_node_pll4 = { .name = "pll4" }
```

6.29.3.12 clock_node_pll4_fref_pll0

```
struct kdrv_clock_node clock_node_pll4_fref_pll0 = { .name = "pll4frefpll0" }
```

6.29.3.13 clock_node_pll4_out1

```
struct kdrv_clock_node clock_node_pll4_out1 = { .name = "pll4o1" }
```

6.29.3.14 clock_node_pll5

```
struct kdrv_clock_node clock_node_pll5 = { .name = "pll5" }
```

Kneron Confidential

6.29.3.15 clock_node_pll5_out1

```
struct kdrv_clock_node clock_node_pll5_out1 = { .name = "pll5o1" }
```

6.29.3.16 clock_node_pll5_out2

```
struct kdrv_clock_node clock_node_pll5_out2 = { .name = "pll5o2" }
```

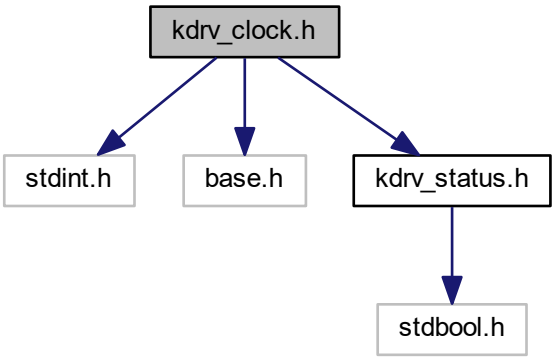
6.29.3.17 SystemCoreClock

```
uint32_t SystemCoreClock
```

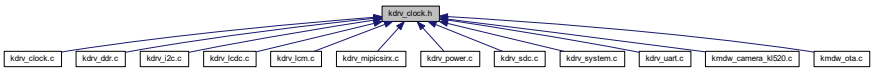
6.30 kdrv_clock.h File Reference

```
#include <stdint.h>
#include <base.h>
#include "kdrv_status.h"
```

Include dependency graph for kdrv_clock.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_clock_value
- struct kdrv_clock_list
- struct kdrv_clock_node

Macros

- #define CLOCK_MUXSEL_NCPU_TRACECLK_DEFAULT 0x10000000
- #define CLOCK_MUXSEL_NCPU_TRACECLK_FROM_SCPU_TRACECLK 0x20000000
- #define CLOCK_MUXSEL_NCPU_TRACECLK_MASK 0x30000000
- #define CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV3 0x01000000
- #define CLOCK_MUXSEL_SCPU_TRACECLK_SRC_PLL0DIV2 0x02000000
- #define CLOCK_MUXSEL_SCPU_TRACECLK_MASK 0x03000000
- #define CLOCK_MUXSEL_CSIRX1_CLK_PLL5 0x00100000
- #define CLOCK_MUXSEL_CSIRX1_CLK_PLL3 0x00200000
- #define CLOCK_MUXSEL_CSIRX1_CLK_MASK 0x00300000
- #define CLOCK_MUXSEL_NPU_CLK_PLL4 0x00010000
- #define CLOCK_MUXSEL_NPU_CLK_PLL5 0x00020000
- #define CLOCK_MUXSEL_NPU_CLK_PLL0 0x00040000
- #define CLOCK_MUXSEL_NPU_CLK_MASK 0x00070000
- #define CLOCK_MUXSEL_PLL4_FREF_PLL0DIV 0x00001000
- #define CLOCK_MUXSEL_PLL4_FREF_OSC 0x00002000
- #define CLOCK_MUXSEL_PLL4_MASK 0x00003000
- #define CLOCK_MUXSEL_UART_0_IRDA_UCLK_UART 0x00000100
- #define CLOCK_MUXSEL_UART_0_IRDA_UCLK_IRDA 0x00000200
- #define CLOCK_MUXSEL_UART_0_IRDA_UCLK_MASK 0x00000300

Typedefs

- typedef int(∗ fn_set) (struct kdrv_clock_node ∗, struct kdrv_clock_value ∗)

Enumerations

- enum clk {
  CLK_PLL1 = 1, CLK_PLL1_OUT, CLK_PLL2, CLK_PLL2_OUT,
  CLK_PLL3, CLK_PLL3_OUT1, CLK_PLL3_OUT2, CLK_PLL4,
  CLK_PLL4_OUT, CLK_PLL5, CLK_PLL5_OUT1, CLK_PLL5_OUT2,
  CLK_FCS_PLL2 = 20, CLK_FCS_DLL, CLK_PLL4_FREF_PLL0, CLK_BUS_SAHB = 30,
  CLK_BUS_NAHB, CLK_BUS_PAHB1, CLK_BUS_PAHB2, CLK_BUS_APB0,
  CLK_BUS_APB1, CLK_SCPU = 50, CLK_SCPU_TRACE, CLK_NCPU = 60,
  CLK_NCPU_TRACE, CLK_NPU, CLK_SPI_CLK = 100, CLK_ADC_CLK,
  CLK_WDT_EXT_CLK, CLK_SD_CLK, CLK_MIPI_TXHSPLLREF_CLK, CLK_MIPI_TX_ESC_CLK,
  CLK_MIPI_CSITX_DSI_CLK, CLK_MIPI_CSITX_CSI_CLK, CLK_MIPI_CSIRX1_TXESC_CLK, CLK_MIPI_CSIRX1_CSI_CLK
  CLK_MIPI_CSIRX1_VC0_CLK, CLK_MIPI_CSIRX0_TXESC_CLK, CLK_MIPI_CSIRX0_CSI_CLK, CLK_MIPI_CSIRX0_VC0_
  CLK_LC_SCALER, CLK_LC_CLK, CLK_TMR1_EXTCLK3, CLK_TMR1_EXTCLK2,
  CLK_TMR1_EXTCLK1, CLK_TMR0_EXTCLK3, CLK_TMR0_EXTCLK2, CLK_TMR0_EXTCLK1,
  CLK_PWM_EXTCLK6, CLK_PWM_EXTCLK5, CLK_PWM_EXTCLK4, CLK_PWM_EXTCLK3,
  CLK_PWM_EXTCLK2, CLK_PWM_EXTCLK1, CLK_UART1_3_FREF, CLK_UART1_2_FREF,
  CLK_UART1_1_FREF, CLK_UART1_0_FREF, CLK_UART0_FREF, CLK_SSP1_1_SSPCLK,
  CLK_SSP1_0_SSPCLK, CLK_SSP0_1_SSPCLK, CLK_SSP0_0_SSPCLK }
- enum pll_id {
  pll_1 = 0, pll_2, pll_3, pll_4,
  pll_5 }
- enum scuclkin_type { scuclkin_osc = 0, scuclkin_rtcosc, scuclkin_pll0div3, scuclkin_pll0div4 }

Kneron Confidential

Functions

- void kdrv_clock_mgr_init (void)
- void kdrv_clock_mgr_open (struct kdrv_clock_node *node, struct kdrv_clock_value *clock_val)
- void kdrv_clock_mgr_close (struct kdrv_clock_node *node)
- void kdrv_clock_mgr_set_scuclkin (enum scuclkin_type type, bool enable)
- void kdrv_clock_mgr_set_muxsel (uint32_t flags)
- uint32_t kdrv_clock_mgr_calculate_clockout (enum pll_id id, uint16_t ms, uint16_t ns, uint16_t F_ps)
- void kdrv_clock_mgr_open_pll1 (void)
- void kdrv_clock_mgr_open_pll2 (void)
- void kdrv_clock_mgr_open_pll3 (void)
- void kdrv_clock_mgr_open_pll4 (void)
- void kdrv_clock_mgr_open_pll5 (void)
- void kdrv_clock_mgr_close_pll1 (void)
- void kdrv_clock_mgr_close_pll2 (void)
- void kdrv_clock_mgr_close_pll4 (void)
- void kdrv_clock_mgr_change_pll3_clock (uint32_t ms, uint32_t ns, uint32_t ps, uint32_t csi0_txesc, uint32_t csi0_csi, uint32_t csi0_vc0, uint32_t csi1_txesc, uint32_t csi1_csi, uint32_t csi1_vc0)
- void kdrv_clock_mgr_change_pll5_clock (uint32_t ms, uint32_t ns, uint32_t ps)
- void kdrv_debug_pll_clock (void)
- void kdrv_clock_set_csiclk (uint32_t cam_idx, uint32_t enable)
- void kdrv_clock_enable (enum clk clk)
- void kdrv_clock_disable (enum clk clk)
- void kdrv_delay_us (uint32_t usec)

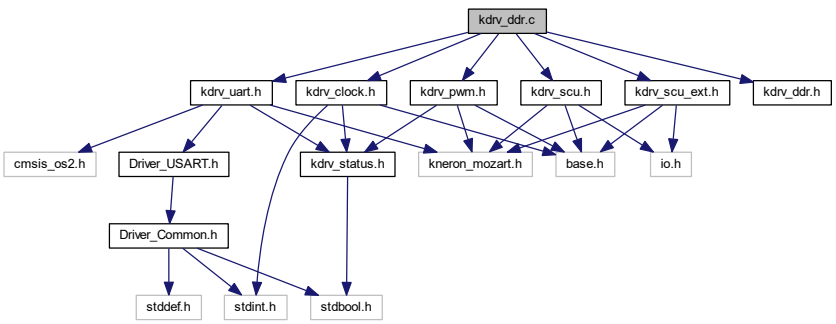
Variables

- struct kdrv_clock_node clock_node_pll1_out

6.31 kdrv_ddr.c File Reference

```
#include "kdrv_uart.h"
#include "kdrv_ddr.h"
#include "kdrv_scu_ext.h"
#include "kdrv_clock.h"
#include "kdrv_pwm.h"
#include "kdrv_scu.h"
```

Include dependency graph for kdrv_ddr.c:



Kneron Confidential

Macros

- #define DDR_BASE DDRC_FTDDR3030_PA_BASE
- #define DDR_REG_MCSR (DDR_BASE + 0x0004)
- #define DDR_REG_MCSR_EXIT_SELF_REFRESH BIT3
- #define DDR_REG_MCSR_SELF_REFRESH BIT2
- #define DDR_REG_MCSR_INIT_CMD BIT0
- #define DDR_DORMANT_BOOTUP (SCU_REG_BTUP_STS_RTC_BTUPTS | SCU_REG_BTUP_STS_SMR)

Functions

- void kdrv_ddr_self_refresh_enter (void)

  DDR enter self-refresh mode to save power.
- void kdrv_ddr_self_refresh_exit (void)

  DDR exit self-refresh mode to normal mode.
- void kdrv_ddr_wakeup (void)

  DDR wakeup and de-assert reset of DDR controller.
- void kdrv_ddr_system_init (enum kdrv_ddr_init_mode mode)

  DDR initialize.

Variables

- uint32_t bootup_status

6.31.1 Macro Definition Documentation

6.31.1.1 DDR_BASE

```
#define DDR_BASE DDRC_FTDDR3030_PA_BASE
```

6.31.1.2 DDR_DORMANT_BOOTUP

```
#define DDR_DORMANT_BOOTUP (SCU_REG_BTUP_STS_RTC_BTUPTS | SCU_REG_BTUP_STS_SMR)
```

6.31.1.3 DDR_REG_MCSR

```
#define DDR_REG_MCSR (DDR_BASE + 0x0004)
```

Kneron Confidential

6.31.1.4 DDR_REG_MCSR_EXIT_SELF_REFRESH

```
#define DDR_REG_MCSR_EXIT_SELF_REFRESH BIT3
```

6.31.1.5 DDR_REG_MCSR_INIT_CMD

```
#define DDR_REG_MCSR_INIT_CMD BIT0
```

6.31.1.6 DDR_REG_MCSR_SELF_REFRESH

```
#define DDR_REG_MCSR_SELF_REFRESH BIT2
```

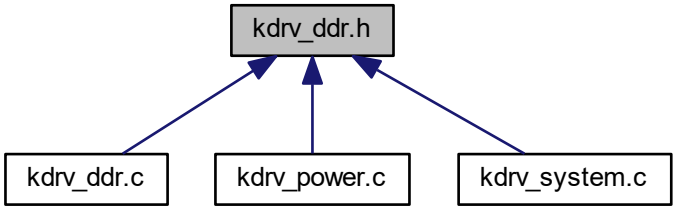
6.31.2 Variable Documentation

6.31.2.1 bootup_status

```
uint32_t bootup_status
```

6.32 kdrv_ddr.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum kdrv_ddr_init_mode { DDR_INIT_WAKEUP_ONLY = 0, DDR_INIT_ALL, DDR_INIT_ALL_EXIT_SELF_REFRESH }

  Enumeration of ddr init mode.

Kneron Confidential

Functions

- void kdrv_ddr_wakeup (void)

  DDR wakeup and de-assert reset of DDR controller.
- void kdrv_ddr_system_init (enum kdrv_ddr_init_mode mode)

  DDR initialize.
- void kdrv_ddr_self_refresh_enter (void)

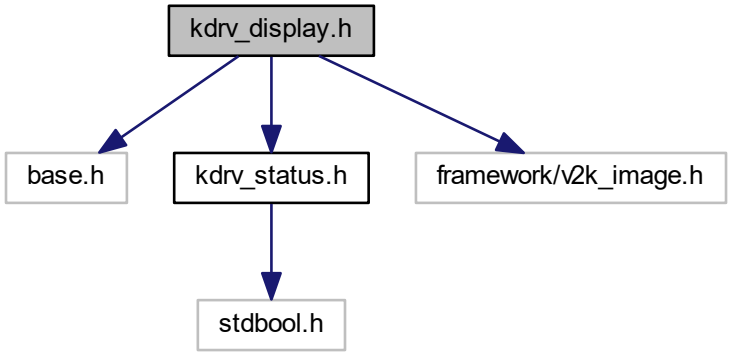
  DDR enter self-refresh mode to save power.
- void kdrv_ddr_self_refresh_exit (void)

  DDR exit self-refresh mode to normal mode.

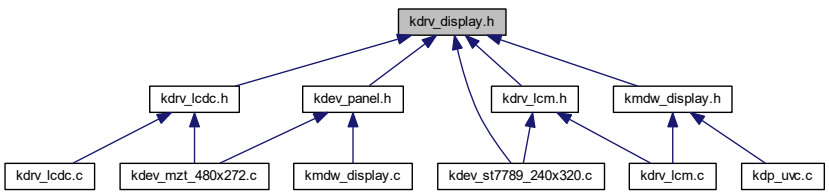
6.33 kdrv_display.h File Reference

```
#include "base.h"
#include "kdrv_status.h"
#include "framework/v2k_image.h"
```

Include dependency graph for kdrv_display.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_display_pen_info_t

  Enumerations of display pen setting.
- struct kdrv_display_t

  Enumerations of display driver setting.

Macros

- #define FRAME_SIZE_RGB(xres, yres, mbpp) ((xres) * (yres) * (mbpp) / 8)

Functions

- int kdrv_display_cal_framesize (unsigned short width, unsigned short height, unsigned int input_fmt)

  inline function to calculate frame size for display
- kdrv_display_t * kdrv_display_initialize (void)

  Initialize display driver.
- kdrv_status_t kdrv_display_buffer_initialize (struct video_input_params *params)

  Initialize display frame buffer.
- uint32_t kdrv_display_get_buffer (void)
- kdrv_status_t kdrv_display_set_params (kdrv_display_t *display_drv, struct video_input_params *params)

  Set display parameters.
- kdrv_status_t kdrv_display_get_params (kdrv_display_t *display_drv, struct video_input_params *params)

  Get display parameters.
- kdrv_status_t kdrv_display_set_camera (kdrv_display_t *display_drv, uint8_t cam_idx)

  Set camera source which will be preview on display.
- kdrv_status_t kdrv_display_start (kdrv_display_t *display_drv)

  Start display image preview.
- kdrv_status_t kdrv_display_stop (kdrv_display_t *display_drv)

  Stop display image preview.
- kdrv_status_t kdrv_display_set_pen (kdrv_display_t *display_drv, uint16_t color, uint32_t width)

  Set pen setting.
- kdrv_status_t kdrv_display_update_draw_fb (kdrv_display_t *display_drv, uint32_t addr, uint8_t cam_idx)

  Update frame buffer which be used to draw something on display.
- kdrv_status_t kdrv_display_draw_static_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_line (kdrv_display_t *display_drv, uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- kdrv_status_t kdrv_display_fill_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle with filling color on display.
- kdrv_status_t kdrv_display_draw_bitmap (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height, void *pBuf)

  Draw bitmap on display.
- kdrv_status_t kdrv_display_test_pattern_gen (kdrv_display_t *display_drv, bool pat_gen)

Kneron Confidential

Set display backlight.

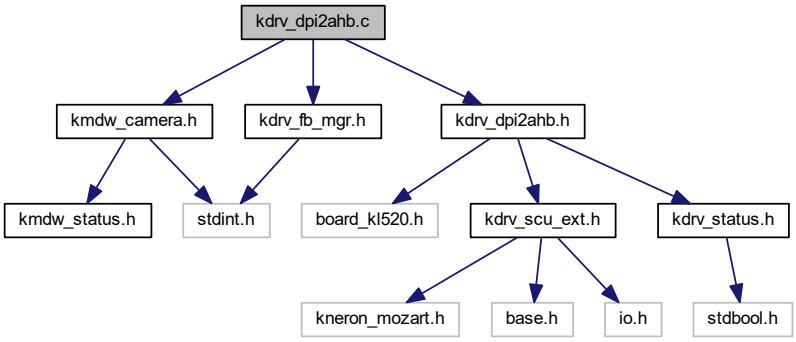
- kdrv_status_t kdrv_display_set_backlight (kdrv_display_t *display_drv, int light)

  Set display backlight.
- kdrv_status_t kdrv_display_set_frame_margin_len (uint16_t margin_len)
- uint16_t kdrv_display_get_frame_margin_len (void)
- kdrv_status_t kdrv_display_set_review_snapshot_en (bool enable)
- bool kdrv_display_get_review_snapshot_en (void)
- kdrv_status_t kdrv_display_set_snapshot_preview_en (bool enable)
- bool kdrv_display_get_snapshot_preview_en (void)

6.34 kdrv_dpi2ahb.c File Reference

```
#include "kmdw_camera.h"
#include "kdrv_dpi2ahb.h"
#include "kdrv_fb_mgr.h"
```

Include dependency graph for kdrv_dpi2ahb.c:



Data Structures

- struct kdp520_dpi2ahb_context
- union U_regDPI2AHB
- union U_regDPI2AHBCtrl

Macros

- #define DPI2AHB_PAGE_NUM 2
- #define D2A_REG_CTRL 0x00
- #define D2A_REG_FNC 0x04
- #define D2A_REG_P0ADDR 0x08
- #define D2A_REG_P1ADDR 0x0C
- #define D2A_REG_ICT 0x10
- #define D2A_REG_IS 0x14
- #define D2A_REG_ST 0x18
- #define D2A_REG_PT 0x1C
- #define D2A_REG_FIU0 0x20

Kneron Confidential

- #define D2A_REG_FIU1 0x24
- #define D2A_REG_TAVR 0x28
- #define D2A_INT_TILE_AVG_D BIT5
- #define D2A_INT_AHB_TX_ERR BIT4
- #define D2A_INT_FIFO_UF BIT3
- #define D2A_INT_FIFO_OF BIT2
- #define D2A_INT_FN_OL BIT1
- #define D2A_INT_WRD BIT0
- #define D2A_INT_ALL 0x3F
- #define D2A_INT_ERRORS (D2A_INT_AHB_TX_ERR | D2A_INT_FIFO_UF | D2A_INT_FIFO_OF | D2A_INT_FN_OL)
- #define D2A_ST_PG 0x3
- #define D2A_PT_YUV422 0x1E
- #define D2A_PT_RGB565 0x22
- #define D2A_PT_RGB888 0x24
- #define D2A_PT_RAW8 0x2A
- #define D2A_PT_RAW10 0x2B
- #define D2A_PT_RAW12 0x2C
- #define D2A_PT_RAW14 0x2D
- #define D2A_PT_RAW16 0x2E
- #define TILE_AVG_SIZE_128 0x00000000
- #define TILE_AVG_SIZE_64 BIT16
- #define TILE_AVG_SIZE_32 BIT17
- #define TILE_AVG_SIZE_VAL TILE_AVG_SIZE_128
- #define TILE_AVG_SIZE_PIXELS 128
- #define D2A_REG_CTRL_0 0x3000
- #define D2A_REG_CTRL_1 0x3000
- #define regdpi2ahb_0 ((union U_regDPI2AHB *) DPI2AHB_CSR_PA_BASE)
- #define regdpi2ahb_1 ((union U_regDPI2AHB *) DPI2AHB_CSR_1_PA_BASE)
- #define regdpi2ahb_ctrl ((union U_regDPI2AHBCtrl *) SCU_EXTREG_DPI2AHB_CTRL)

Typedefs

- typedef volatile union U_regDPI2AHB U_regDPI2AHB
- typedef volatile union U_regDPI2AHBCtrl U_regDPI2AHBCtrl

Functions

- void kdrv_dpi2ahb_irqhandler (uint32_t cam_idx)
- void kdrv_dpi2ahb_isr_0 (void)
- void kdrv_dpi2ahb_isr_1 (void)
- kdrv_status_t kdrv_dpi2ahb_enable (uint32_t cam_idx, struct cam_format *fmt)

  kdrv_dpi2ahb_enable Enable dpi2ahb IP,
- kdrv_status_t kdrv_dpi2ahb_start (uint32_t cam_idx)

  kdrv_dpi2ahb_start Start dpi2ahb interrup, enable IRQ.
- kdrv_status_t kdrv_dpi2ahb_stop (uint32_t cam_idx)

  kdrv_dpi2ahb_stop Stop dpi2ahb interrup, disable IRQ.
- kdrv_status_t kdrv_dpi2ahb_buf_init (uint32_t cam_idx)

  kdrv_dpi2ahb_buf_init Set dpi2ahb page buffer default address.
- kdrv_status_t kdrv_dpi2ahb_initialize (uint32_t cam_idx)

  kdrv_dpi2ahb_initialize Init dpi2ahb IRQ and reset IP.

Kneron Confidential

Variables

- const uint32_t PixelFmtMapping [4] = {D2A_PT_RGB565, D2A_PT_RAW10, D2A_PT_RAW8, D2A_PT_YUV422}
- struct kdp520_dpi2ahb_context dpi2ahb_ctx [D2A_CAM_NUM]

6.34.1 Macro Definition Documentation

6.34.1.1 D2A_INT_AHB_TX_ERR

```
#define D2A_INT_AHB_TX_ERR BIT4
```

6.34.1.2 D2A_INT_ALL

```
#define D2A_INT_ALL 0x3F
```

6.34.1.3 D2A_INT_ERRORS

```
#define D2A_INT_ERRORS (D2A_INT_AHB_TX_ERR | D2A_INT_FIFO_UF | D2A_INT_FIFO_OF | D2A_INT_FN_OL)
```

6.34.1.4 D2A_INT_FIFO_OF

```
#define D2A_INT_FIFO_OF BIT2
```

6.34.1.5 D2A_INT_FIFO_UF

```
#define D2A_INT_FIFO_UF BIT3
```

6.34.1.6 D2A_INT_FN_OL

```
#define D2A_INT_FN_OL BIT1
```

Kneron Confidential

6.34.1.7 D2A_INT_TILE_AVG_D

```
#define D2A_INT_TILE_AVG_D BIT5
```

6.34.1.8 D2A_INT_WRD

```
#define D2A_INT_WRD BIT0
```

6.34.1.9 D2A_PT_RAW10

```
#define D2A_PT_RAW10 0x2B
```

6.34.1.10 D2A_PT_RAW12

```
#define D2A_PT_RAW12 0x2C
```

6.34.1.11 D2A_PT_RAW14

```
#define D2A_PT_RAW14 0x2D
```

6.34.1.12 D2A_PT_RAW16

```
#define D2A_PT_RAW16 0x2E
```

6.34.1.13 D2A_PT_RAW8

```
#define D2A_PT_RAW8 0x2A
```

6.34.1.14 D2A_PT_RGB565

```
#define D2A_PT_RGB565 0x22
```

Kneron Confidential

6.34.1.15 D2A_PT_RGB888

```
#define D2A_PT_RGB888 0x24
```

6.34.1.16 D2A_PT_YUV422

```
#define D2A_PT_YUV422 0x1E
```

6.34.1.17 D2A_REG_CTRL

```
#define D2A_REG_CTRL 0x00
```

6.34.1.18 D2A_REG_CTRL_0

```
#define D2A_REG_CTRL_0 0x3000
```

6.34.1.19 D2A_REG_CTRL_1

```
#define D2A_REG_CTRL_1 0x3000
```

6.34.1.20 D2A_REG_FIU0

```
#define D2A_REG_FIU0 0x20
```

6.34.1.21 D2A_REG_FIU1

```
#define D2A_REG_FIU1 0x24
```

6.34.1.22 D2A_REG_FNC

```
#define D2A_REG_FNC 0x04
```

Kneron Confidential

6.34.1.23 D2A_REG_ICT

```
#define D2A_REG_ICT 0x10
```

6.34.1.24 D2A_REG_IS

```
#define D2A_REG_IS 0x14
```

6.34.1.25 D2A_REG_P0ADDR

```
#define D2A_REG_P0ADDR 0x08
```

6.34.1.26 D2A_REG_P1ADDR

```
#define D2A_REG_P1ADDR 0x0C
```

6.34.1.27 D2A_REG_PT

```
#define D2A_REG_PT 0x1C
```

6.34.1.28 D2A_REG_ST

```
#define D2A_REG_ST 0x18
```

6.34.1.29 D2A_REG_TAVR

```
#define D2A_REG_TAVR 0x28
```

6.34.1.30 D2A_ST_PG

```
#define D2A_ST_PG 0x3
```

Kneron Confidential

6.34.1.31 DPI2AHB_PAGE_NUM

```
#define DPI2AHB_PAGE_NUM 2
```

6.34.1.32 regdpi2ahb_0

```
#define regdpi2ahb_0 ((union U_regDPI2AHB *) DPI2AHB_CSR_PA_BASE)
```

6.34.1.33 regdpi2ahb_1

```
#define regdpi2ahb_1 ((union U_regDPI2AHB *) DPI2AHB_CSR_1_PA_BASE)
```

6.34.1.34 regdpi2ahb_ctrl

```
#define regdpi2ahb_ctrl ((union U_regDPI2AHBCtrl *) SCU_EXTREG_DPI2AHB_CTRL)
```

6.34.1.35 TILE_AVG_SIZE_128

```
#define TILE_AVG_SIZE_128 0x00000000
```

6.34.1.36 TILE_AVG_SIZE_32

```
#define TILE_AVG_SIZE_32 BIT17
```

6.34.1.37 TILE_AVG_SIZE_64

```
#define TILE_AVG_SIZE_64 BIT16
```

6.34.1.38 TILE_AVG_SIZE_PIXELS

```
#define TILE_AVG_SIZE_PIXELS 128
```

Kneron Confidential

6.34.1.39 TILE_AVG_SIZE_VAL

```
#define TILE_AVG_SIZE_VAL TILE_AVG_SIZE_128
```

6.34.2 Typedef Documentation

6.34.2.1 U_regDPI2AHB

```
typedef volatile union U_regDPI2AHB U_regDPI2AHB
```

6.34.2.2 U_regDPI2AHBCtrl

```
typedef volatile union U_regDPI2AHBCtrl U_regDPI2AHBCtrl
```

6.34.3 Function Documentation

6.34.3.1 kdrv_dpi2ahb_irqhandler()

```
void kdrv_dpi2ahb_irqhandler (
            uint32_t cam_idx )
```

6.34.3.2 kdrv_dpi2ahb_isr_0()

```
void kdrv_dpi2ahb_isr_0 (
            void  )
```

6.34.3.3 kdrv_dpi2ahb_isr_1()

```
void kdrv_dpi2ahb_isr_1 (
            void  )
```

6.34.4 Variable Documentation

Kneron Confidential

6.34.4.1 dpi2ahb_ctx

```
struct kdp520_dpi2ahb_context dpi2ahb_ctx[D2A_CAM_NUM]
```

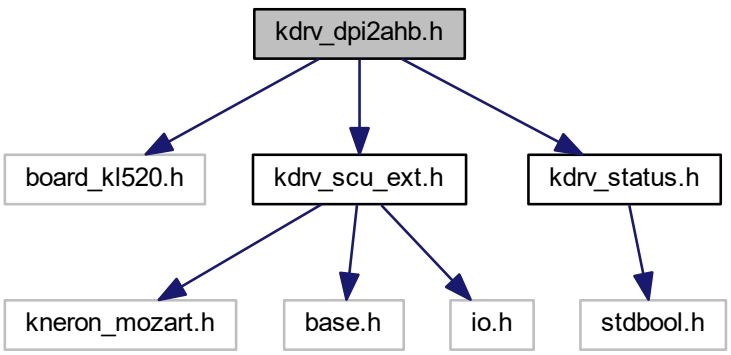
6.34.4.2 PixelFmtMapping

```
const uint32_t PixelFmtMapping[4] = {D2A_PT_RGB565, D2A_PT_RAW10, D2A_PT_RAW8, D2A_PT_YUV422}
```

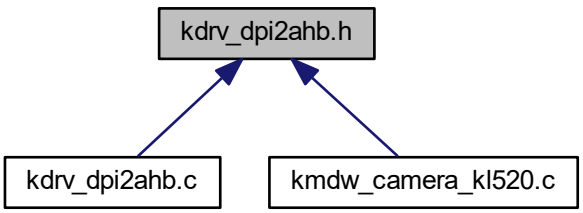
6.35 kdrv_dpi2ahb.h File Reference

```
#include "board_kl520.h"
#include "kdrv_scu_ext.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_dpi2ahb.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define TILE_BLOCK_MAX_W 10
- #define TILE_BLOCK_MAX_H 6
- #define TILE_BLOCKS_MAX (TILE_BLOCK_MAX_W * TILE_BLOCK_MAX_H)
- #define TILE_REGS_MAX (TILE_BLOCKS_MAX / 4)

Enumerations

- enum { D2A_CAM_0, D2A_CAM_1, D2A_CAM_NUM }

Functions

- kdrv_status_t kdrv_dpi2ahb_enable (uint32_t cam_idx, struct cam_format *fmt)

  kdrv_dpi2ahb_enable Enable dpi2ahb IP,
- kdrv_status_t kdrv_dpi2ahb_stop (uint32_t cam_idx)

  kdrv_dpi2ahb_stop Stop dpi2ahb interrup, disable IRQ.
- kdrv_status_t kdrv_dpi2ahb_start (uint32_t cam_idx)

  kdrv_dpi2ahb_start Start dpi2ahb interrup, enable IRQ.
- kdrv_status_t kdrv_dpi2ahb_buf_init (uint32_t cam_idx)

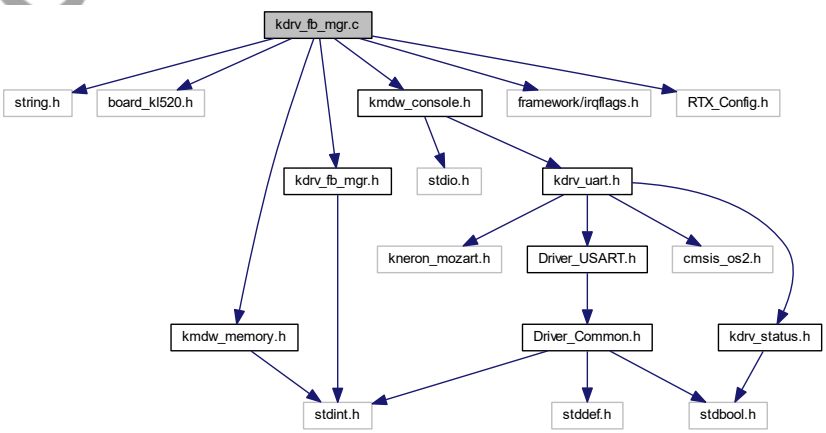
  kdrv_dpi2ahb_buf_init Set dpi2ahb page buffer default address.
- kdrv_status_t kdrv_dpi2ahb_initialize (uint32_t cam_idx)

  kdrv_dpi2ahb_initialize Init dpi2ahb IRQ and reset IP.

6.36 kdrv_fb_mgr.c File Reference

```
#include <string.h>
#include "board_kl520.h"
#include "kmdw_memory.h"
#include "kdrv_fb_mgr.h"
#include "kmdw_console.h"
#include "framework/irqflags.h"
#include "RTX_Config.h"
```

Include dependency graph for kdrv_fb_mgr.c:



Kneron Confidential

Data Structures

- struct **kdrv_fb_mgr_s**

Macros

- #define USE_GUARDBAND 4
- #define INF_WAIT_MAX_TIME_MS 500
- #define INF_WAIT_OS_DELAY_MS 10
- #define INF_WAIT_OS_DELAY_TICKS (OS_TICK_FREQ * INF_WAIT_OS_DELAY_MS / 1000)
- #define fb_msg(fmt, ...)

Functions

- int kdrv_fb_mgr_init (int cam_idx, uint32_t buf_size, int buf_num, int frame_info_size)

  To initialize frame buffer pool for camera.
- uint32_t kdrv_fb_mgr_next_write (int cam_idx, int *write_idx)

  To get next available buffer for camera to write.
- int kdrv_fb_mgr_write_done (int cam_idx, int write_idx)

  To notify the finish of a buffer written by camera.
- void kdrv_fb_mgr_free_write_buf (int cam_idx)

  To free the use of frame buffers for camera.
- uint32_t kdrv_fb_mgr_get_buf_seq_num (int cam_idx, int idx)

  To get the (camera write) sequence number of a buffer.
- uint32_t kdrv_fb_mgr_get_current_buf_seq_num (int cam_idx)

  To get current buffer sequence number.
- uint32_t kdrv_fb_mgr_get_buf (int cam_idx, int idx)

  To get a buffer of an index.
- uint32_t kdrv_fb_mgr_get_frame_info_buf (int cam_idx, int idx)

  To get a buffer of an index.
- uint32_t kdrv_fb_mgr_next_read (int cam_idx, int *read_idx)

  To get the next buffer for drawing.
- int kdrv_fb_mgr_read_done (int cam_idx, int read_idx)

  To notify the finish of a buffer done for drawing.
- void kdrv_fb_mgr_free_read_buf (int cam_idx)

  To free the use of frame buffers for drawing.
- uint32_t kdrv_fb_mgr_next_inf (int cam_idx, int *inf_idx)

  To get the next buffer for inference.
- int kdrv_fb_mgr_inf_done (int cam_idx, int inf_idx)

  To notify the finish of a buffer done for inference.
- void kdrv_fb_mgr_free_inf_buf (int cam_idx)

  To free the use of frame buffers for inference.
- uint32_t kdrv_fb_mgr_next_display (int cam_idx, int *disp_idx)

  To get the buffer for display.
- int kdrv_fb_mgr_display_set_src (int cam_idx, int disp_src)

  To set display buffer source.
- int kdrv_fb_mgr_notifier_register (int cam_idx, fb_write_done_notify callback)

  To register a callback for a finished frame by camera.

Kneron Confidential

6.36.1 Macro Definition Documentation

6.36.1.1 fb_msg

```
#define fb_msg(
              fmt,
              ...  )
```

6.36.1.2 INF_WAIT_MAX_TIME_MS

```
#define INF_WAIT_MAX_TIME_MS 500
```

6.36.1.3 INF_WAIT_OS_DELAY_MS

```
#define INF_WAIT_OS_DELAY_MS 10
```

6.36.1.4 INF_WAIT_OS_DELAY_TICKS

```
#define INF_WAIT_OS_DELAY_TICKS (OS_TICK_FREQ * INF_WAIT_OS_DELAY_MS / 1000)
```

6.36.1.5 USE_GUARDBAND

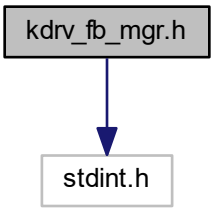
```
#define USE_GUARDBAND 4
```

Kneron Confidential

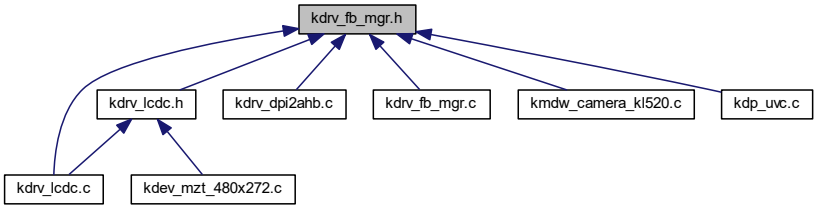
6.37 kdrv_fb_mgr.h File Reference

```
#include <stdint.h>
```

Include dependency graph for kdrv_fb_mgr.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define MAX_FRAME_BUFFER 7
- #define DISPLAY_SRC_READ 0
- #define DISPLAY_SRC_INF 1
- #define DISPLAY_SRC_WRITE 2

Typedefs

- typedef int(∗ fb_write_done_notify) (int cam_idx, int write_idx)

Functions

- int kdrv_fb_mgr_init (int cam_idx, uint32_t buf_size, int buf_num, int frame_info_size)

  To initialize frame buffer pool for camera.
- uint32_t kdrv_fb_mgr_next_write (int cam_idx, int ∗write_idx)

  To get next available buffer for camera to write.

Kneron Confidential

- int kdrv_fb_mgr_write_done (int cam_idx, int write_idx)

  To notify the finish of a buffer written by camera.

- void kdrv_fb_mgr_free_write_buf (int cam_idx)

  To free the use of frame buffers for camera.

- uint32_t kdrv_fb_mgr_get_buf (int cam_idx, int idx)

  To get a buffer of an index.

- uint32_t kdrv_fb_mgr_get_frame_info_buf (int cam_idx, int idx)

  To get a buffer of an index.

- uint32_t kdrv_fb_mgr_get_buf_seq_num (int cam_idx, int idx)

  To get the (camera write) sequence number of a buffer.

- uint32_t kdrv_fb_mgr_get_current_buf_seq_num (int cam_idx)

  To get current buffer sequence number.

- uint32_t kdrv_fb_mgr_next_read (int cam_idx, int *read_idx)

  To get the next buffer for drawing.

- int kdrv_fb_mgr_read_done (int cam_idx, int read_idx)

  To notify the finish of a buffer done for drawing.

- void kdrv_fb_mgr_free_read_buf (int cam_idx)

  To free the use of frame buffers for drawing.

- uint32_t kdrv_fb_mgr_next_inf (int cam_idx, int *inf_idx)

  To get the next buffer for inference.

- int kdrv_fb_mgr_inf_done (int cam_idx, int inf_idx)

  To notify the finish of a buffer done for inference.

- void kdrv_fb_mgr_free_inf_buf (int cam_idx)

  To free the use of frame buffers for inference.

- uint32_t kdrv_fb_mgr_next_display (int cam_idx, int *disp_idx)

  To get the buffer for display.

- int kdrv_fb_mgr_display_set_src (int cam_idx, int disp_src)

  To set display buffer source.

- int kdrv_fb_mgr_notifier_register (int cam_idx, fb_write_done_notify callback)

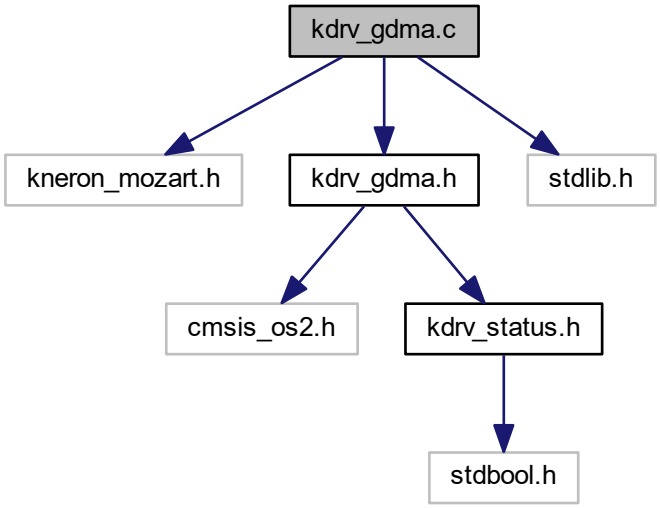
  To register a callback for a finished frame by camera.

Kneron Confidential

6.38 kdrv_gdma.c File Reference

```
#include "kneron_mozart.h"
#include "kdrv_gdma.h"
#include <stdlib.h>
```

Include dependency graph for kdrv_gdma.c:



Data Structures

- union U_regGDMA
- union U_regGDMA_CH
- struct _GDMA_CH_Ctrl_t

Macros

- #define DMA_FIFO_SIZE 4
- #define NUM_DMA_CHANNELS 8
- #define NUM_CH_FOR_MEMCPY 3
- #define DMA_CHANNEL_BIT_MASK ((0x1 << NUM_DMA_CHANNELS) - 1)
- #define DMACEN 0x1
- #define GDMA_FLAG_XFER_DONE 0x100
- #define regGDMA ((U_regGDMA *)DMAC_FTDMAC020_PA_BASE)
- #define regGDMA_ch ((U_regGDMA_CH *)(DMAC_FTDMAC020_PA_BASE + 0x100))

Functions

- kdrv_status_t kdrv_gdma_initialize (void)

  GDMA driver initialization.
- kdrv_status_t kdrv_gdma_uninitialize (void)

  GDMA driver uninitialization.
- kdrv_status_t kdrv_gdma_acquire_handle (kdrv_gdma_handle_t *handle)

  Acquire a GDMA handle.
- kdrv_status_t kdrv_gdma_configure_setting (kdrv_gdma_handle_t handle, gdma_setting_t *dma_setting)

Kneron Confidential

Configure the DMA working behavior on specified DMA handle with specified dma settings.

- kdrv_status_t kdrv_gdma_release_handle (kdrv_gdma_handle_t handle)

  Release the DMA handle.

- kdrv_status_t kdrv_gdma_transfer_async (kdrv_gdma_handle_t handle, uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes, gdma_xfer_callback_t xfer_isr_cb, void *usr_arg)

  Start DMA transfer with specified DMA handle running in asynchronous (non-blocking) mode.

- kdrv_status_t kdrv_gdma_transfer (kdrv_gdma_handle_t handle, uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes)

  Start DMA transfer with specified DMA handle running in synchronous (blocking) mode.

- kdrv_status_t kdrv_gdma_memcpy_async (uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes, gdma_xfer_callback_t xfer_isr_cb, void *usr_arg)

  Start DMA transfer with automatic DMA handle running in asynchronous (non-blocking) mode.

- kdrv_status_t kdrv_gdma_memcpy (uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes)

  Start DMA transfer with automatic DMA handle running in synchronous (blocking) mode.

6.38.1 Macro Definition Documentation

6.38.1.1 DMA_CHANNEL_BIT_MASK

```
#define DMA_CHANNEL_BIT_MASK ((0x1 << NUM_DMA_CHANNELS) - 1)
```

6.38.1.2 DMA_FIFO_SIZE

```
#define DMA_FIFO_SIZE 4
```

6.38.1.3 DMACEN

```
#define DMACEN 0x1
```

6.38.1.4 GDMA_FLAG_XFER_DONE

```
#define GDMA_FLAG_XFER_DONE 0x100
```

6.38.1.5 NUM_CH_FOR_MEMCPY

```
#define NUM_CH_FOR_MEMCPY 3
```

Kneron Confidential

6.38.1.6 NUM_DMA_CHANNELS

```
#define NUM_DMA_CHANNELS 8
```

6.38.1.7 regGDMA

```
#define regGDMA ((U_regGDMA *)DMAC_FTDMAC020_PA_BASE)
```

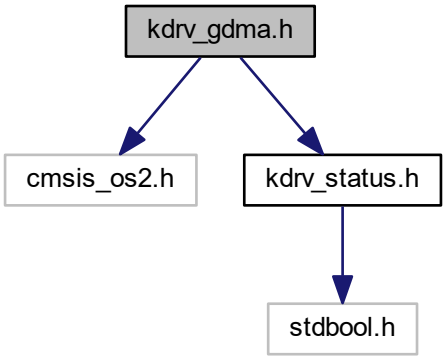
6.38.1.8 regGDMA_ch

```
#define regGDMA_ch ((U_regGDMA_CH *)(DMAC_FTDMAC020_PA_BASE + 0x100))
```

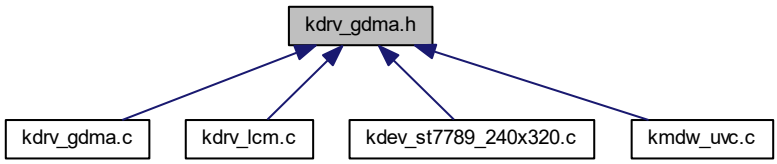
6.39 kdrv_gdma.h File Reference

```
#include "cmsis_os2.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_gdma.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct gdma_setting_t

  Structure of GDMA advanced settings for a specified DMA handle (channel)

Typedefs

- typedef int32_t kdrv_gdma_handle_t

  GDMA handle type which represents for a DMA channel and related DMA operations.
- typedef void(∗ gdma_xfer_callback_t) (kdrv_status_t status, void ∗arg)

  GDMA user callback function with transfer status notification. Note that this is callback form ISR context.

Enumerations

- enum gdma_transfer_width_t { GDMA_TXFER_WIDTH_8_BITS = 0x0, GDMA_TXFER_WIDTH_16_BITS, GDMA_TXFER_WIDTH_32_BITS }

  Enumeration of GDMA transfer size: 8/16/32 bits, this is about byte-alignment.
- enum gdma_burst_size_t { GDMA_BURST_SIZE_1 = 0x0, GDMA_BURST_SIZE_4, GDMA_BURST_SIZE_8, GDMA_BURST_SIZE_16, GDMA_BURST_SIZE_32, GDMA_BURST_SIZE_64, GDMA_BURST_SIZE_128, GDMA_BURST_SIZE_256 }

  Enumeration of GDMA transfer burst : 1/4/8/16/32/64/128/256, this is about performance.
- enum gdma_address_control_t { GDMA_INCREMENT_ADDRESS = 0x0, GDMA_DECREMENT_ADDRESS, GDMA_FIXED_ADDRESS }

  Enumeration of DMA address control, auto-increasing/descreading or fixed.
- enum gdma_work_mode_t { GDMA_NORMAL_MODE = 0x0, GDMA_HW_HANDSHAKE_MODE }

  Enumeration of DMA working mode, can be normal or hardware handshake mode.

Functions

- kdrv_status_t kdrv_gdma_initialize (void)

  GDMA driver initialization.
- kdrv_status_t kdrv_gdma_uninitialize (void)

  GDMA driver uninitialization.
- kdrv_status_t kdrv_gdma_acquire_handle (kdrv_gdma_handle_t ∗handle)

  Acquire a GDMA handle.
- kdrv_status_t kdrv_gdma_configure_setting (kdrv_gdma_handle_t handle, gdma_setting_t ∗dma_setting)

  Configure the DMA working behavior on specified DMA handle with specified dma settings.
- kdrv_status_t kdrv_gdma_release_handle (kdrv_gdma_handle_t handle)

  Release the DMA handle.
- kdrv_status_t kdrv_gdma_transfer_async (kdrv_gdma_handle_t handle, uint32_t dst_addr, uint32_t src_↩ addr, uint32_t num_bytes, gdma_xfer_callback_t xfer_isr_cb, void ∗usr_arg)

  Start DMA transfer with specified DMA handle running in asynchronous (non-blocking) mode.
- kdrv_status_t kdrv_gdma_transfer (kdrv_gdma_handle_t handle, uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes)

  Start DMA transfer with specified DMA handle running in synchronous (blocking) mode.
- kdrv_status_t kdrv_gdma_memcpy_async (uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes, gdma_xfer_callback_t xfer_isr_cb, void ∗usr_arg)

  Start DMA transfer with automatic DMA handle running in asynchronous (non-blocking) mode.
- kdrv_status_t kdrv_gdma_memcpy (uint32_t dst_addr, uint32_t src_addr, uint32_t num_bytes)

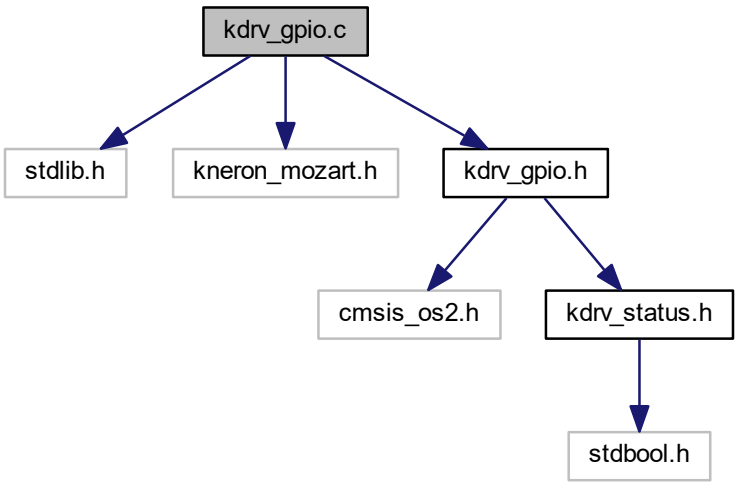
  Start DMA transfer with automatic DMA handle running in synchronous (blocking) mode.

Kneron Confidential

6.40 kdrv_gpio.c File Reference

```
#include <stdlib.h>
#include "kneron_mozart.h"
#include "kdrv_gpio.h"
```

Include dependency graph for kdrv_gpio.c:



Data Structures

- union U_regGPIO

Macros

- #define regGPIO ((U_regGPIO *)GPIO_FTGPIO010_PA_BASE)

Functions

- kdrv_status_t kdrv_gpio_initialize ()
  GPIO driver initialization, this must be invoked once before any GPIO manipulations.
- kdrv_status_t kdrv_gpio_uninitialize ()
  GPIO driver uninitialization.
- kdrv_status_t kdrv_gpio_set_attribute (kdrv_gpio_pin_t pin, uint32_t attributes)
  set pin attributes for a specified GPIO pin
- kdrv_status_t kdrv_gpio_register_callback (gpio_interrupt_callback_t gpio_isr_cb, void *usr_arg)
  register user callback with user argument for GPIO interrupt in this callback can get interrupts for all GPIO pins
- kdrv_status_t kdrv_gpio_set_interrupt (kdrv_gpio_pin_t pin, bool isEnable)
  set interrupt enable/disable for a specified GPIO pin
- kdrv_status_t kdrv_gpio_set_debounce (kdrv_gpio_pin_t pin, bool isEnable, uint32_t debounce_clock)
  set debounce enable/disable with clock setting in Hz
- kdrv_status_t kdrv_gpio_write_pin (kdrv_gpio_pin_t pin, bool value)
  write GPIO digitial pin value
- kdrv_status_t kdrv_gpio_read_pin (kdrv_gpio_pin_t pin, bool *pValue)
  read GPIO digitial pin value

Kneron Confidential

6.40.1 Macro Definition Documentation

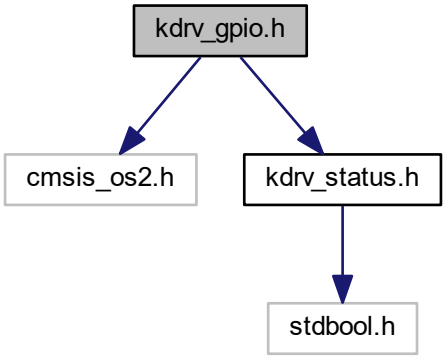
6.40.1.1 regGPIO

```
#define regGPIO ((U_regGPIO *)GPIO_FTGPIO010_PA_BASE)
```

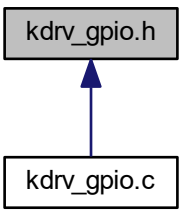
6.41 kdrv_gpio.h File Reference

```
#include <cmsis_os2.h>
#include "kdrv_status.h"
```

Include dependency graph for kdrv_gpio.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Typedefs

- typedef void(∗ gpio_interrupt_callback_t) (kdrv_gpio_pin_t pin, void ∗arg)

Enumerations

- enum kdrv_gpio_attribute_t {
  GPIO_DIR_INPUT = 0x1, GPIO_DIR_OUTPUT = 0x2, GPIO_INT_EDGE_RISING = 0x4, GPIO_INT_EDGE_FALLING = 0x8,
  GPIO_INT_EDGE_BOTH = 0x10, GPIO_INT_LEVEL_HIGH = 0x20, GPIO_INT_LEVEL_LOW = 0x40 }

  Enumerations of GPIO pin attributes, input or output, interrupt trigger settings.
- enum kdrv_gpio_pin_t {
  GPIO_PIN_0 = 0, GPIO_PIN_1, GPIO_PIN_2, GPIO_PIN_3,
  GPIO_PIN_4, GPIO_PIN_5, GPIO_PIN_6, GPIO_PIN_7,
  GPIO_PIN_8, GPIO_PIN_9, GPIO_PIN_10, GPIO_PIN_11,
  GPIO_PIN_12, GPIO_PIN_13, GPIO_PIN_14, GPIO_PIN_15,
  GPIO_PIN_16, GPIO_PIN_17, GPIO_PIN_18, GPIO_PIN_19,
  GPIO_PIN_20, GPIO_PIN_21, GPIO_PIN_22, GPIO_PIN_23,
  GPIO_PIN_24, GPIO_PIN_25, GPIO_PIN_26, GPIO_PIN_27,
  GPIO_PIN_28, GPIO_PIN_29, GPIO_PIN_30, GPIO_PIN_31 }

  Enumerations of GPIO pin ID, there 32 GPIO pins.

Functions

- kdrv_status_t kdrv_gpio_initialize (void)

  GPIO driver initialization, this must be invoked once before any GPIO manipulations.
- kdrv_status_t kdrv_gpio_uninitialize (void)

  GPIO driver uninitialization.
- kdrv_status_t kdrv_gpio_set_attribute (kdrv_gpio_pin_t pin, uint32_t attributes)

  set pin attributes for a specified GPIO pin
- kdrv_status_t kdrv_gpio_register_callback (gpio_interrupt_callback_t gpio_isr_cb, void ∗usr_arg)

  register user callback with user argument for GPIO interrupt in this callback can get interrupts for all GPIO pins
- kdrv_status_t kdrv_gpio_set_interrupt (kdrv_gpio_pin_t pin, bool isEnable)

  set interrupt enable/disable for a specified GPIO pin
- kdrv_status_t kdrv_gpio_set_debounce (kdrv_gpio_pin_t pin, bool isEnable, uint32_t debounce_clock)

  set debounce enable/disable with clock setting in Hz
- kdrv_status_t kdrv_gpio_write_pin (kdrv_gpio_pin_t pin, bool value)

  write GPIO digitial pin value
- kdrv_status_t kdrv_gpio_read_pin (kdrv_gpio_pin_t pin, bool ∗pValue)

  read GPIO digitial pin value

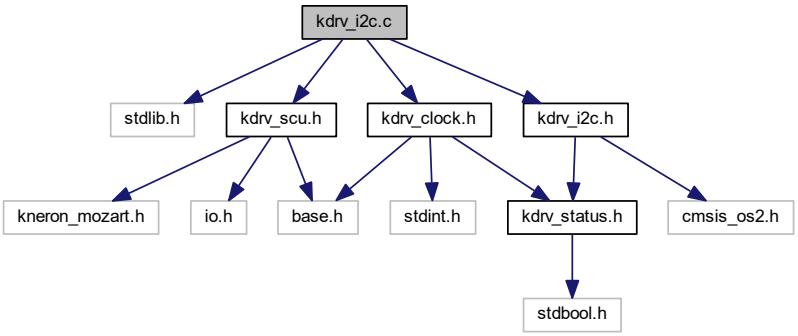
6.42 kdrv_i2c.c File Reference

```
#include <stdlib.h>
#include "kdrv_scu.h"
#include "kdrv_clock.h"
```

Kneron Confidential

```
#include "kdrv_i2c.h"
```

Include dependency graph for kdrv_i2c.c:



Macros

- #define I2C_ENABLE_THREAD_SYNC
- #define I2CRegRead(id, reg_offset) inw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset)
- #define I2CRegWrite(id, reg_offset, val) outw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset, val)
- #define I2CRegMaskedSet(id, reg_offset, val) masked_outw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset, val, val)
- #define I2CRegMaskedClr(id, reg_offset, val) masked_outw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset, 0, val)
- #define REG_I2C_CR 0x00
- #define CR_ALIRQ BIT13 /* arbitration lost interrupt (master) */
- #define CR_SAMIRQ BIT12 /* slave address match interrupt (slave) */
- #define CR_STOPIRQ BIT11 /* stop condition interrupt (slave) */
- #define CR_NAKRIRQ BIT10 /* NACK response interrupt (master) */
- #define CR_DRIRQ BIT9 /* rx interrupt (both) */
- #define CR_DTIRQ BIT8 /* tx interrupt (both) */
- #define CR_TBEN BIT7 /* tx enable (both) */
- #define CR_NAK BIT6 /* NACK (both) */
- #define CR_STOP BIT5 /* stop (master) */
- #define CR_START BIT4 /* start (master) */
- #define CR_GCEN BIT3 /* general call support (slave) */
- #define CR_MST_EN BIT2 /* enable clock out (master) */
- #define CR_I2C_EN BIT1 /* enable I2C (both) */
- #define CR_I2C_RST BIT0 /* reset I2C (both) */
- #define CR_ENABLE (CR_ALIRQ | CR_NAKRIRQ | CR_DRIRQ | CR_DTIRQ | CR_MST_EN | CR_I2C_EN)
- #define REG_I2C_SR 0x04
- #define SR_SBS BIT23 /* start byte */
- #define SR_HSS BIT22 /* high speed mode */
- #define SR_START BIT11 /* start */
- #define SR_AL BIT10 /* arbitration lost */
- #define SR_GC BIT9 /* general call */
- #define SR_SAM BIT8 /* slave address match */
- #define SR_STOP BIT7 /* stop received */
- #define SR_NACK BIT6 /* NACK received */
- #define SR_TD BIT5 /* transfer done */

Kneron Confidential

- #define SR_BB BIT3 /* bus busy */
- #define SR_I2CB BIT2 /* chip busy */
- #define SR_RW BIT0 /* set when master-rx or slave-tx mode */
- #define REG_I2C_CDR 0x08
- #define CDR_COUNT_100KHZ ((uint32_t)((APB_CLOCK) / (2 * 100000) - (GSR_VALUE / 2) - 2))
- #define CDR_COUNT_400KHZ ((uint32_t)((APB_CLOCK) / (2 * 400000) - (GSR_VALUE / 2) - 2))
- #define CDR_COUNT_1MHZ ((uint32_t)((APB_CLOCK) / (2 * 1000000) - (GSR_VALUE / 2) - 2))
- #define REG_I2C_DR 0x0C
- #define REG_I2C_AR 0x10
- #define REG_I2C_TGSR 0x14
- #define GSR_VALUE 0x5
- #define TSR_VALUE 0x5
- #define REG_I2C_BMR 0x18
- #define REG_I2C_BSTMR 0x1C
- #define REG_I2C_REVISION 0x30

Functions

- kdrv_status_t kdrv_i2c_initialize (kdrv_i2c_ctrl_t ctrl_id, kdrv_i2c_bus_speed_t bus_speed)

  Initializes Kdrv I2C driver (as master) and configures it for the specified speed.
- kdrv_status_t kdrv_i2c_uninitialize (kdrv_i2c_ctrl_t ctrl_id)

  Uninitializes Kdrv I2C driver.
- kdrv_status_t kdp_i2c_transmit (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint8_t *data, uint32_t num, bool with_STOP)

  transmit data to a specified slave address, the STOP condition can be optionally not generated.
- kdrv_status_t kdp_i2c_receive (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint8_t *data, uint32_t num, bool with_STOP)

  receive data from a specified slave address, the STOP condition can be optionally not generated.
- kdrv_status_t kdrv_i2c_write_register (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint16_t reg, uint16_t reg_size, uint16_t len, uint8_t *data)

  specialized function to write to the register of slave device, register address can be 1 or 2 bytes.
- kdrv_status_t kdrv_i2c_read_register (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint16_t reg, uint16_t reg_size, uint16_t len, uint8_t *data)

  specialized function to read from the register of slave device, register address can be 1 or 2 bytes.

6.42.1 Macro Definition Documentation

6.42.1.1 CDR_COUNT_100KHZ

```
#define CDR_COUNT_100KHZ ((uint32_t)((APB_CLOCK) / (2 * 100000) - (GSR_VALUE / 2) - 2))
```

6.42.1.2 CDR_COUNT_1MHZ

```
#define CDR_COUNT_1MHZ ((uint32_t)((APB_CLOCK) / (2 * 1000000) - (GSR_VALUE / 2) - 2))
```

Kneron Confidential

6.42.1.3 CDR_COUNT_400KHZ

```
#define CDR_COUNT_400KHZ ((uint32_t)((APB_CLOCK) / (2 * 400000) - (GSR_VALUE / 2) - 2))
```

6.42.1.4 CR_ALIRQ

```
#define CR_ALIRQ BIT13 /* arbitration lost interrupt (master) */
```

6.42.1.5 CR_DRIRQ

```
#define CR_DRIRQ BIT9 /* rx interrupt (both) */
```

6.42.1.6 CR_DTIRQ

```
#define CR_DTIRQ BIT8 /* tx interrupt (both) */
```

6.42.1.7 CR_ENABLE

```
#define CR_ENABLE (CR_ALIRQ | CR_NAKRIRQ | CR_DRIRQ | CR_DTIRQ | CR_MST_EN | CR_I2C_EN)
```

6.42.1.8 CR_GCEN

```
#define CR_GCEN BIT3 /* general call support (slave) */
```

6.42.1.9 CR_I2C_EN

```
#define CR_I2C_EN BIT1 /* enable I2C (both) */
```

6.42.1.10 CR_I2C_RST

```
#define CR_I2C_RST BIT0 /* reset I2C (both) */
```

Kneron Confidential

6.42.1.11 CR_MST_EN

```
#define CR_MST_EN BIT2 /* enable clock out (master) */
```

6.42.1.12 CR_NAK

```
#define CR_NAK BIT6 /* NACK (both) */
```

6.42.1.13 CR_NAKRIRQ

```
#define CR_NAKRIRQ BIT10 /* NACK response interrupt (master) */
```

6.42.1.14 CR_SAMIRQ

```
#define CR_SAMIRQ BIT12 /* slave address match interrupt (slave) */
```

6.42.1.15 CR_START

```
#define CR_START BIT4 /* start (master) */
```

6.42.1.16 CR_STOP

```
#define CR_STOP BIT5 /* stop (master) */
```

6.42.1.17 CR_STOPIRQ

```
#define CR_STOPIRQ BIT11 /* stop condition interrupt (slave) */
```

6.42.1.18 CR_TBEN

```
#define CR_TBEN BIT7 /* tx enable (both) */
```

Kneron Confidential

6.42.1.19 GSR_VALUE

```
#define GSR_VALUE 0x5
```

6.42.1.20 I2C_ENABLE_THREAD_SYNC

```
#define I2C_ENABLE_THREAD_SYNC
```

6.42.1.21 I2CRegMaskedClr

```
#define I2CRegMaskedClr(
              id,
              reg_offset,
              val ) masked_outw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset, 0, val)
```

6.42.1.22 I2CRegMaskedSet

```
#define I2CRegMaskedSet(
              id,
              reg_offset,
              val ) masked_outw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset, val, val)
```

6.42.1.23 I2CRegRead

```
#define I2CRegRead(
              id,
              reg_offset ) inw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset)
```

6.42.1.24 I2CRegWrite

```
#define I2CRegWrite(
              id,
              reg_offset,
              val ) outw(IIC_FTIIC010_0_PA_BASE + id * 0x100000 + reg_offset, val)
```

Kneron Confidential

6.42.1.25 REG_I2C_AR

```
#define REG_I2C_AR 0x10
```

6.42.1.26 REG_I2C_BMR

```
#define REG_I2C_BMR 0x18
```

6.42.1.27 REG_I2C_BSTMR

```
#define REG_I2C_BSTMR 0x1C
```

6.42.1.28 REG_I2C_CDR

```
#define REG_I2C_CDR 0x08
```

6.42.1.29 REG_I2C_CR

```
#define REG_I2C_CR 0x00
```

6.42.1.30 REG_I2C_DR

```
#define REG_I2C_DR 0x0C
```

6.42.1.31 REG_I2C_REVISION

```
#define REG_I2C_REVISION 0x30
```

6.42.1.32 REG_I2C_SR

```
#define REG_I2C_SR 0x04
```

Kneron Confidential

6.42.1.33 REG_I2C_TGSR

```
#define REG_I2C_TGSR 0x14
```

6.42.1.34 SR_AL

```
#define SR_AL BIT10 /* arbitration lost */
```

6.42.1.35 SR_BB

```
#define SR_BB BIT3 /* bus busy */
```

6.42.1.36 SR_GC

```
#define SR_GC BIT9 /* general call */
```

6.42.1.37 SR_HSS

```
#define SR_HSS BIT22 /* high speed mode */
```

6.42.1.38 SR_I2CB

```
#define SR_I2CB BIT2 /* chip busy */
```

6.42.1.39 SR_NACK

```
#define SR_NACK BIT6 /* NACK received */
```

6.42.1.40 SR_RW

```
#define SR_RW BIT0 /* set when master-rx or slave-tx mode */
```

Kneron Confidential

6.42.1.41 SR_SAM

```
#define SR_SAM BIT8 /* slave address match */
```

6.42.1.42 SR_SBS

```
#define SR_SBS BIT23 /* start byte */
```

6.42.1.43 SR_START

```
#define SR_START BIT11 /* start */
```

6.42.1.44 SR_STOP

```
#define SR_STOP BIT7 /* stop received */
```

6.42.1.45 SR_TD

```
#define SR_TD BIT5 /* transfer done */
```

6.42.1.46 TSR_VALUE

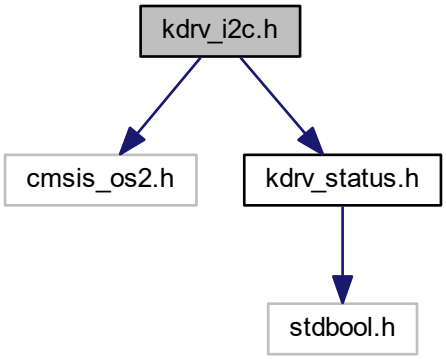
```
#define TSR_VALUE 0x5
```

Kneron Confidential

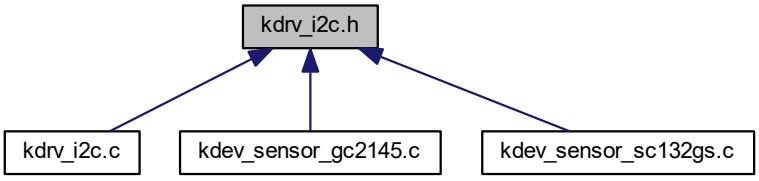
6.43 kdrv_i2c.h File Reference

```
#include "cmsis_os2.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_i2c.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Enumerations

- enum kdrv_i2c_bus_speed_t { KDRV_I2C_SPEED_100K = 0, KDRV_I2C_SPEED_200K, KDRV_I2C_SPEED_400K, KDRV_I2C_SPEED_1M }

  Enumerations of I2C bus speed.

- enum kdrv_i2c_ctrl_t { KDRV_I2C_CTRL_0 = 0, KDRV_I2C_CTRL_1, KDRV_I2C_CTRL_2, KDRV_I2C_CTRL_3, TOTAL_KDRV_I2C_CTRL }

  Enumerations of I2C controller instances.

Functions

- kdrv_status_t kdrv_i2c_initialize (kdrv_i2c_ctrl_t ctrl_id, kdrv_i2c_bus_speed_t bus_speed)

  Initializes Kdrv I2C driver (as master) and configures it for the specified speed.
- kdrv_status_t kdrv_i2c_uninitialize (kdrv_i2c_ctrl_t ctrl_id)

  Uninitializes Kdrv I2C driver.
- kdrv_status_t kdp_i2c_transmit (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint8_t *data, uint32_t num, bool with_STOP)

  transmit data to a specified slave address, the STOP condition can be optionally not generated.
- kdrv_status_t kdp_i2c_receive (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint8_t *data, uint32_t num, bool with_STOP)

  receive data from a specified slave address, the STOP condition can be optionally not generated.
- kdrv_status_t kdrv_i2c_write_register (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint16_t reg, uint16_t reg_size, uint16_t len, uint8_t *data)

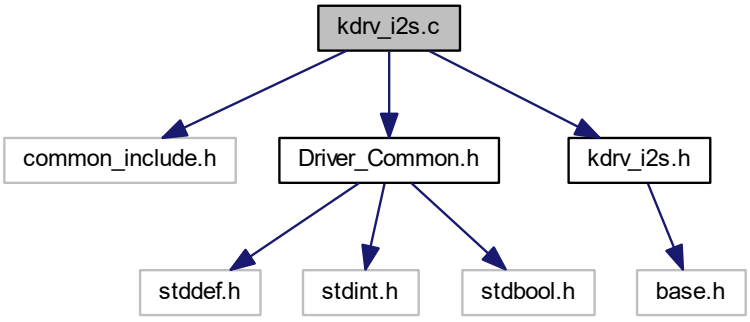
  specialized function to write to the register of slave device, register address can be 1 or 2 bytes.
- kdrv_status_t kdrv_i2c_read_register (kdrv_i2c_ctrl_t ctrl_id, uint16_t slave_addr, uint16_t reg, uint16_t reg_size, uint16_t len, uint8_t *data)

  specialized function to read from the register of slave device, register address can be 1 or 2 bytes.

6.44 kdrv_i2s.c File Reference

```
#include "common_include.h"
#include "Driver_Common.h"
#include "kdrv_i2s.h"
```

Include dependency graph for kdrv_i2s.c:



Functions

- void fLib_I2S_Init (u32 base_address, u32 codec, SSP_MODE_T mode)
- void fLib_SetSSPFrame (u32 base_address, u32 frame)
- void fLib_SetSSPFramePolar (u32 base_address, u32 polar)
- void fLib_SetSSPOPMode (u32 base_address, u32 mode)
- void fLib_SetSSPDataLen (u32 base_address, u32 len)
- void fLib_SetSSPpaddingDataLen (u32 base_address, u32 len)

Kneron Confidential

- void fLib_SetSSPClkdiv (u32 base_address, u32 divider)
- void fLib_SSPClearTxFIFO (u32 base_address)
- void fLib_SSPClearRxFIFO (u32 base_address)
- void fLib_WriteSSP (u32 base_address, u32 data)
- u32 fLib_ReadSSPStatus (u32 base_address)
- u32 fLib_ReadSSP (u32 base_address)
- void fLib_SetSSPSCLKPO (u32 base_address, u32 spolarity)
- void fLib_SetSSPSCLKPH (u32 base_address, u32 sphase)
- u32 fLib_ReadSSPIntStatus (u32 base_address)
- void fLib_SetSSP_TXFIFO (u32 base_address, u32 threshold, u32 underrun)
- void fLib_SetSSP_RXFIFO (u32 base_address, u32 threshold, u32 underrun)
- void fLib_SetSSP_DMA (u32 base_address, u32 trans, u32 rec)
- void fLib_SetSSP_FIFO_Threshold (u32 base_address, u32 trans_len, u32 rec_len)
- void fLib_SetSSP_WarmReset (u32 base_address)
- void fLib_SetSSP_ColdReset (u32 base_address)
- void fLib_SetSSP_Enable (u32 base_address, int enable)
- void fLib_SetSSP_IntMask (u32 base_address, int Mask)
- u32 fLib_SSP_GetTxFIFOLen (u32 base_address)
- u32 fLib_SSP_GetRxFIFOLen (u32 base_address)
- u32 fLib_SSP_GetRxFIFOValidEntries (u32 base_addr)
- void fLib_AC97_SetSlotValidReg (u32 base_address, u32 SlotValid)
- void fLib_InitAC97 (u32 base_address)
- void fLib_AC97_WriteData (u32 base_address, u32 *data, u32 Len)
- void fLib_AC97_ReadData (u32 base_address, u32 *data, u32 Len)
- u32 fLib_ReadSSP32Bit (u32 base_address)
- u32 fLib_AC97_ReadOneWordData (u32 base_address)
- int fLib_AC97_ReadRegister (u32 base_address, u32 Reg_Index, u16 *data)
- int fLib_ReturnTxFIFO_Count (u32 base_address)
- int fLib_ReturnRxFIFO_Count (u32 base_address)
- int fLib_AC97_ReadRegisterEx (u32 base_address, u32 Reg_Index, u16 *data)
- void fLib_AC97_WriteRegister (u32 base_address, u32 Reg_Index, u32 data)
- void fLib_AC97_WriteRegisterEx (u32 base_address, u32 Reg_Index, u32 data)
- u32 fLib_SSP_busy (u32 base_addr)
- void fLib_SetSSP_SDL_Write (u32 base_address, u32 len, u32 w_data)

Variables

- int TxFIFOLen = 0
- int RxFIFOLen = 0

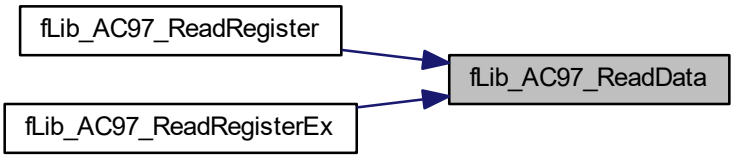
6.44.1 Function Documentation

Kneron Confidential

6.44.1.1 fLib_AC97_ReadData()

```
void fLib_AC97_ReadData (
            u32 base_address,
            u32 * data,
            u32 Len )
```

Here is the caller graph for this function:



6.44.1.2 fLib_AC97_ReadOneWordData()

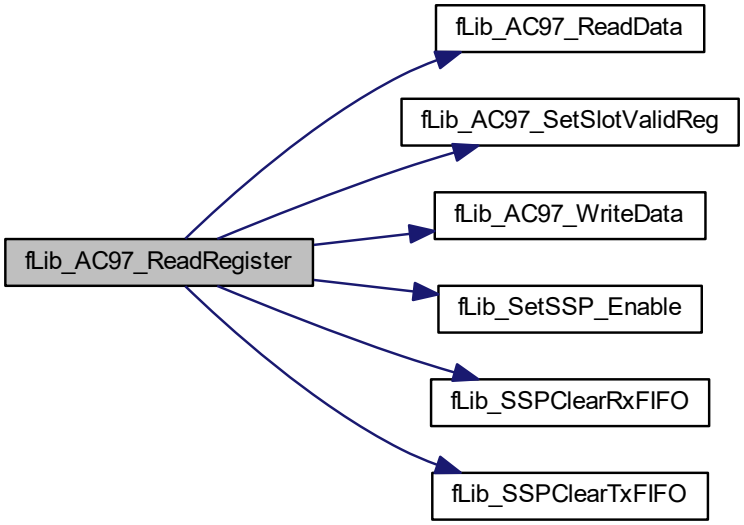
```
u32 fLib_AC97_ReadOneWordData (
            u32 base_address )
```

6.44.1.3 fLib_AC97_ReadRegister()

```
int fLib_AC97_ReadRegister (
            u32 base_address,
            u32 Reg_Index,
            u16 * data )
```

Kneron Confidential

Here is the call graph for this function:

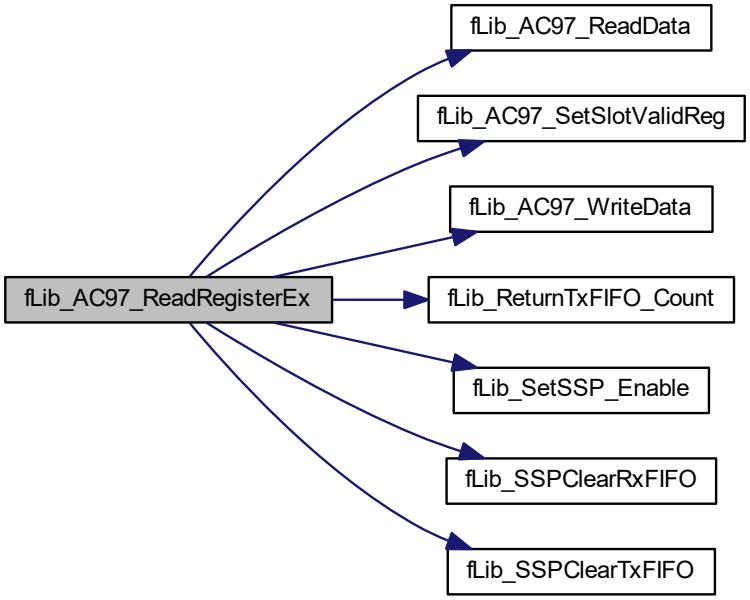


Kneron Confidential

6.44.1.4 fLib_AC97_ReadRegisterEx()

```
int fLib_AC97_ReadRegisterEx (
            u32 base_address,
            u32 Reg_Index,
            u16 * data )
```

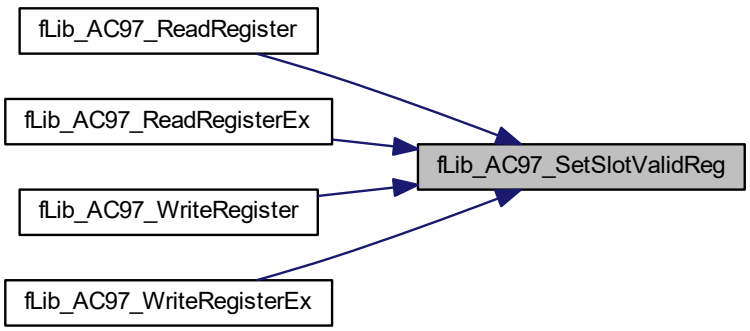
Here is the call graph for this function:



6.44.1.5 fLib_AC97_SetSlotValidReg()

```
void fLib_AC97_SetSlotValidReg (
            u32 base_address,
            u32 SlotValid )
```

Here is the caller graph for this function:

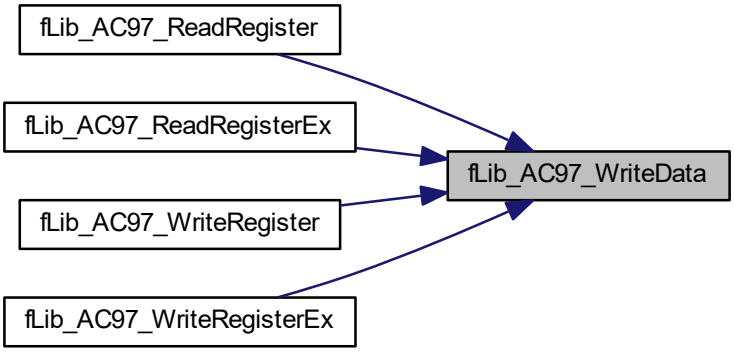


Kneron Confidential

6.44.1.6 fLib_AC97_WriteData()

```
void fLib_AC97_WriteData (
            u32 base_address,
            u32 * data,
            u32 Len )
```

Here is the caller graph for this function:

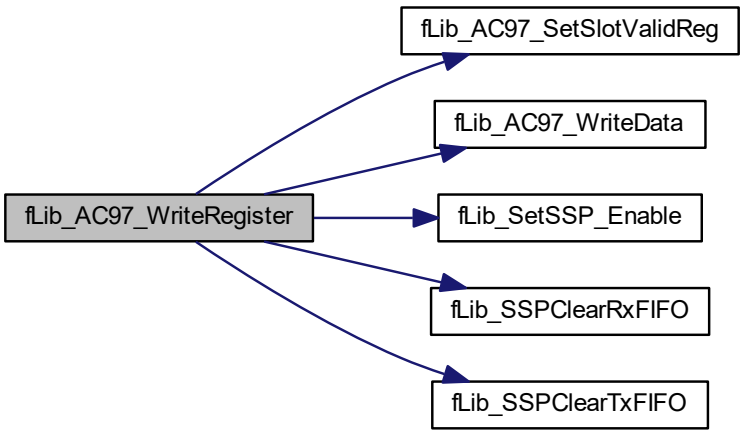


Kneron Confidential

6.44.1.7 fLib_AC97_WriteRegister()

```
void fLib_AC97_WriteRegister (
            u32 base_address,
            u32 Reg_Index,
            u32 data )
```

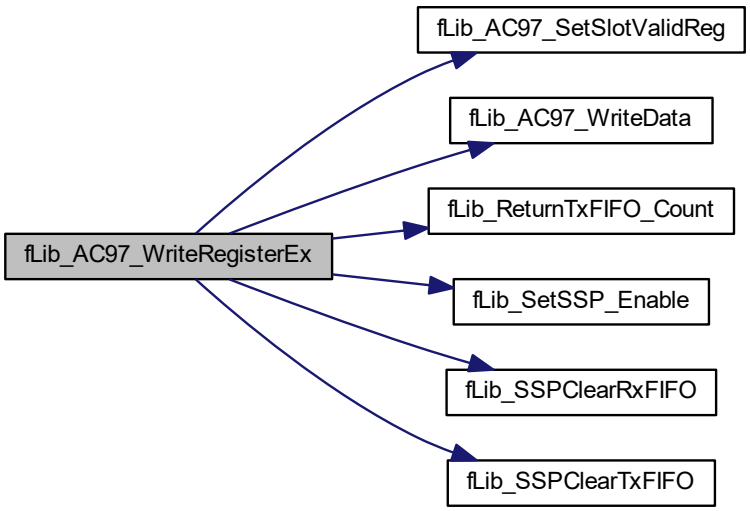
Here is the call graph for this function:



6.44.1.8 fLib_AC97_WriteRegisterEx()

```
void fLib_AC97_WriteRegisterEx (
            u32 base_address,
            u32 Reg_Index,
            u32 data )
```

Here is the call graph for this function:

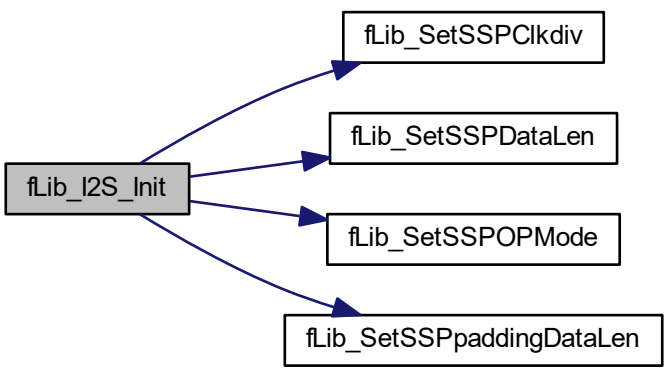


Kneron Confidential

6.44.1.9 fLib_I2S_Init()

```
void fLib_I2S_Init (
            u32 base_address,
            u32 codec,
            SSP_MODE_T mode )
```

Here is the call graph for this function:

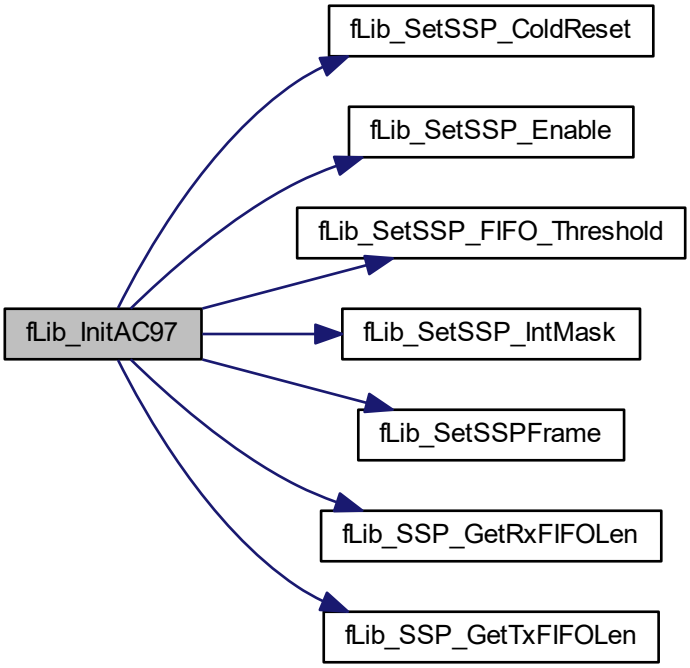


Kneron Confidential

6.44.1.10 fLib_InitAC97()

```
void fLib_InitAC97 (
            u32 base_address )
```

Here is the call graph for this function:



6.44.1.11 fLib_ReadSSP()

```
u32 fLib_ReadSSP (
            u32 base_address )
```

6.44.1.12 fLib_ReadSSP32Bit()

```
u32 fLib_ReadSSP32Bit (
            u32 base_address )
```

6.44.1.13 fLib_ReadSSPIntStatus()

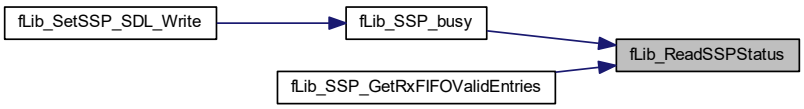
```
u32 fLib_ReadSSPIntStatus (
            u32 base_address )
```

Kneron Confidential

6.44.1.14 fLib_ReadSSPStatus()

```
u32 fLib_ReadSSPStatus (
            u32 base_address )
```

Here is the caller graph for this function:



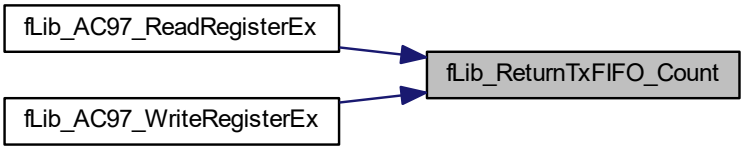
6.44.1.15 fLib_ReturnRxFIFO_Count()

```
int fLib_ReturnRxFIFO_Count (
            u32 base_address )
```

6.44.1.16 fLib_ReturnTxFIFO_Count()

```
int fLib_ReturnTxFIFO_Count (
            u32 base_address )
```

Here is the caller graph for this function:

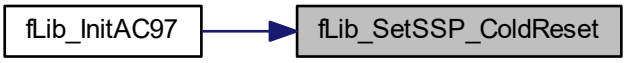


Kneron Confidential

6.44.1.17 fLib_SetSSP_ColdReset()

```
void fLib_SetSSP_ColdReset (
            u32 base_address )
```

Here is the caller graph for this function:



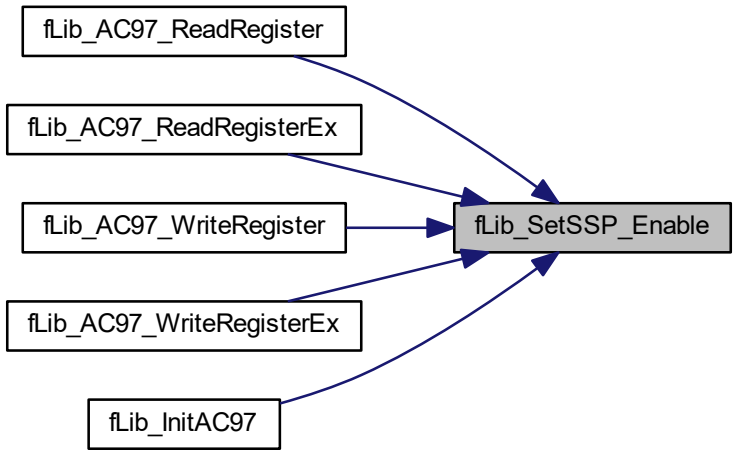
6.44.1.18 fLib_SetSSP_DMA()

```
void fLib_SetSSP_DMA (
            u32 base_address,
            u32 trans,
            u32 rec )
```

6.44.1.19 fLib_SetSSP_Enable()

```
void fLib_SetSSP_Enable (
            u32 base_address,
            int enable )
```

Here is the caller graph for this function:

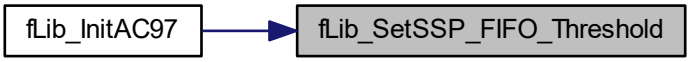


Kneron Confidential

6.44.1.20 fLib_SetSSP_FIFO_Threshold()

```
void fLib_SetSSP_FIFO_Threshold (
            u32 base_address,
            u32 trans_len,
            u32 rec_len )
```

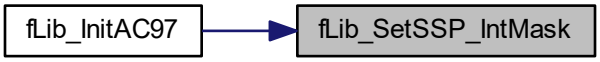
Here is the caller graph for this function:



6.44.1.21 fLib_SetSSP_IntMask()

```
void fLib_SetSSP_IntMask (
            u32 base_address,
            int Mask )
```

Here is the caller graph for this function:



Kneron Confidential

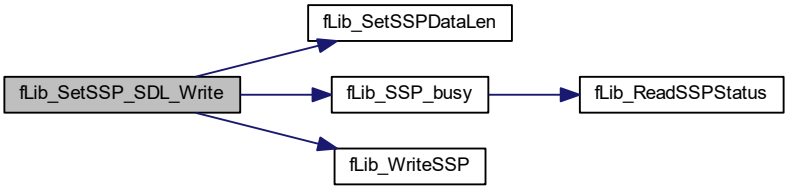
6.44.1.22 fLib_SetSSP_RXFIFO()

```
void fLib_SetSSP_RXFIFO (
            u32 base_address,
            u32 threshold,
            u32 underrun )
```

6.44.1.23 fLib_SetSSP_SDL_Write()

```
void fLib_SetSSP_SDL_Write (
            u32 base_address,
            u32 len,
            u32 w_data )
```

Here is the call graph for this function:



6.44.1.24 fLib_SetSSP_TXFIFO()

```
void fLib_SetSSP_TXFIFO (
            u32 base_address,
            u32 threshold,
            u32 underrun )
```

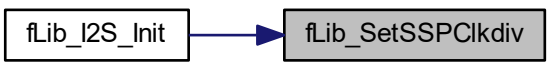
6.44.1.25 fLib_SetSSP_WarmReset()

```
void fLib_SetSSP_WarmReset (
            u32 base_address )
```

6.44.1.26 fLib_SetSSPClkdiv()

```
void fLib_SetSSPClkdiv (
            u32 base_address,
            u32 divider )
```

Here is the caller graph for this function:

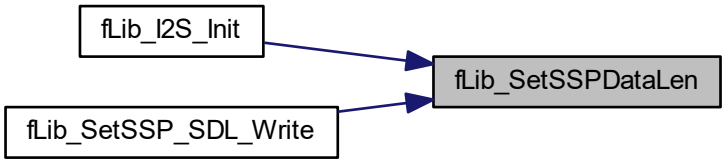


Kneron Confidential

6.44.1.27 fLib_SetSSPDataLen()

```
void fLib_SetSSPDataLen (
            u32 base_address,
            u32 len )
```

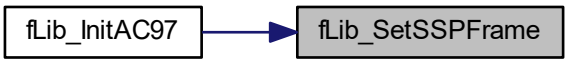
Here is the caller graph for this function:



6.44.1.28 fLib_SetSSPFrame()

```
void fLib_SetSSPFrame (
            u32 base_address,
            u32 frame )
```

Here is the caller graph for this function:



6.44.1.29 fLib_SetSSPFramePolar()

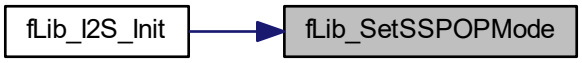
```
void fLib_SetSSPFramePolar (
            u32 base_address,
            u32 polar )
```

Kneron Confidential

6.44.1.30 fLib_SetSSPOPMode()

```
void fLib_SetSSPOPMode (
            u32 base_address,
            u32 mode )
```

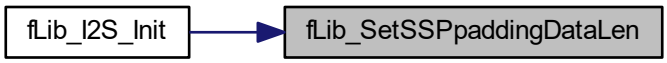
Here is the caller graph for this function:



6.44.1.31 fLib_SetSSPpaddingDataLen()

```
void fLib_SetSSPpaddingDataLen (
            u32 base_address,
            u32 len )
```

Here is the caller graph for this function:



6.44.1.32 fLib_SetSSPSCLKPH()

```
void fLib_SetSSPSCLKPH (
            u32 base_address,
            u32 sphase )
```

Kneron Confidential

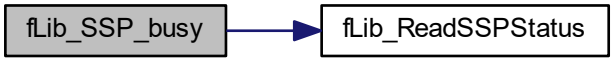
6.44.1.33 fLib_SetSSPSCLKPO()

```
void fLib_SetSSPSCLKPO (
            u32 base_address,
            u32 spolarity )
```

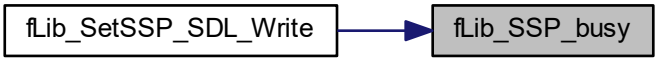
6.44.1.34 fLib_SSP_busy()

```
u32 fLib_SSP_busy (
            u32 base_addr )
```

Here is the call graph for this function:



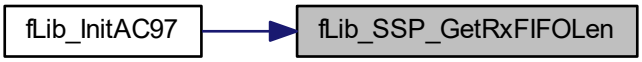
Here is the caller graph for this function:



6.44.1.35 fLib_SSP_GetRxFIFOLen()

```
u32 fLib_SSP_GetRxFIFOLen (
            u32 base_address )
```

Here is the caller graph for this function:



Kneron Confidential

6.44.1.36 fLib_SSP_GetRxFIFOValidEntries()

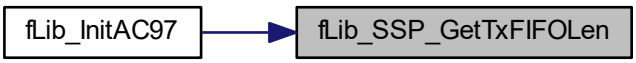
```
u32 fLib_SSP_GetRxFIFOValidEntries (
            u32 base_addr )
```

Here is the call graph for this function:

6.44.1.37 fLib_SSP_GetTxFIFOLen()

```
u32 fLib_SSP_GetTxFIFOLen (
            u32 base_address )
```

Here is the caller graph for this function:

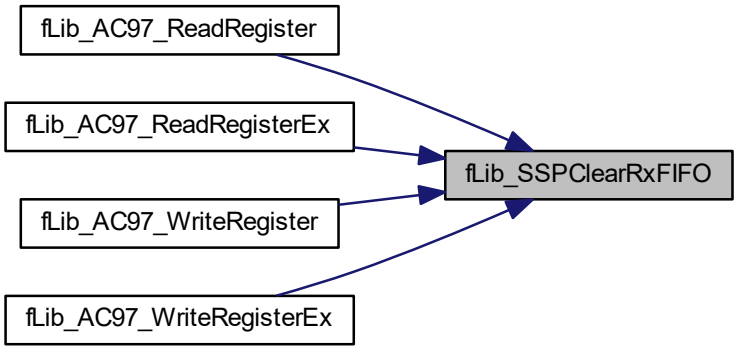


6.44.1.38 fLib_SSPClearRxFIFO()

```
void fLib_SSPClearRxFIFO (
            u32 base_address )
```

Kneron Confidential

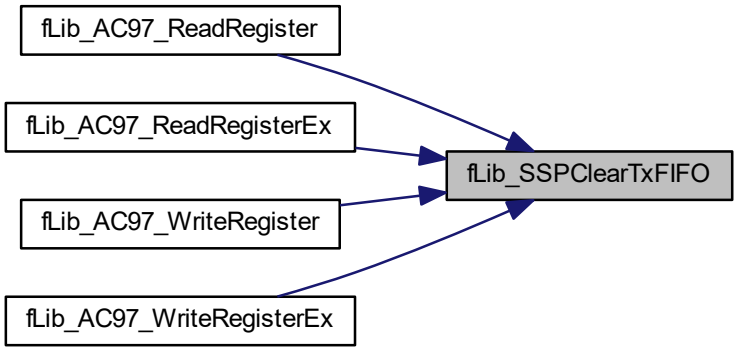
Here is the caller graph for this function:



6.44.1.39 fLib_SSPClearTxFIFO()

```
void fLib_SSPClearTxFIFO (
             u32 base_address )
```

Here is the caller graph for this function:

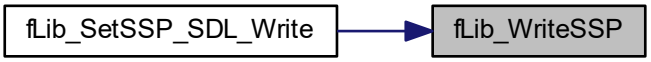


Kneron Confidential

6.44.1.40 fLib_WriteSSP()

```
void fLib_WriteSSP (
            u32 base_address,
            u32 data )
```

Here is the caller graph for this function:



6.44.2 Variable Documentation

6.44.2.1 RxFIFOLen

```
int RxFIFOLen = 0
```

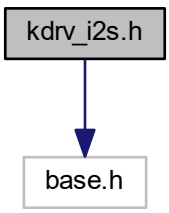
6.44.2.2 TxFIFOLen

```
int TxFIFOLen = 0
```

6.45 kdrv_i2s.h File Reference

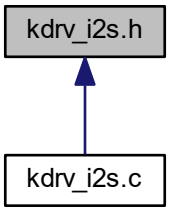
```
#include "base.h"
```

Include dependency graph for kdrv_i2s.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Macros

- #define SSP_CONTROL0 0x0
- #define SSP_CONTROL1 0x4
- #define SSP_CONTROL2 0x8
- #define SSP_STATUS 0xC
- #define SSP_INT_CONTROL 0X10
- #define SSP_INT_STATUS 0x14
- #define SSP_DATA 0x18
- #define SSP_INFO 0x1C
- #define SSP_ACLINK_SLOT_VALID 0x20
- #define SSP_Format 0x0000
- #define SPI_Format 0x1000
- #define Microwire_Format 0x2000
- #define I2S_Format 0x3000
- #define AC97_Format 0x4000
- #define SSP_FSDIST 0x300
- #define SSP_LBM 0x80 /∗ loopback mode ∗/
- #define SSP_LSB 0x40 /∗ LSB first ∗/
- #define SSP_FSPO_LOW 0x20 /∗ Frame sync atcive low ∗/
- #define SSP_FSPO_HIGH 0x0 /∗ Frame sync atcive high ∗/
- #define SSP_DATAJUSTIFY 0x10 /∗ data padding in front of serial data ∗/
- #define SSP_OPM_MS 0x03
- #define SSP_OPM_SL 0x01
- #define SSP_OPM_MSST 0xC /∗ Master stereo mode ∗/
- #define SSP_OPM_MSMO 0x8 /∗ Master mono mode ∗/
- #define SSP_OPM_SLST 0x4 /∗ Slave stereo mode ∗/
- #define SSP_OPM_SLMO 0x0 /∗ Slave mono mode ∗/
- #define SSP_SCLKPO_HIGH 0x2 /∗ SCLK Remain HIGH ∗/
- #define SSP_SCLKPO_LOW 0x0 /∗ SCLK Remain LOW ∗/
- #define SSP_SCLKPH_HALFCLK 0x1 /∗ Half CLK cycle ∗/
- #define SSP_SCLKPH_ONECLK 0x0 /∗ One CLK cycle ∗/
- #define SSP_PDL 0xFF000000 /∗ paddinf data length ∗/
- #define SSP_SDL 0x7F0000 /∗ Serial data length(actual data length-1) ∗/
- #define SSP_CLKDIV 0xFFFF /∗ clk divider ∗/
- #define SSP_TXEN (1<<8) /∗ Transmit function enable ∗/
- #define SSP_RXEN (1<<7) /∗ Receive function enable ∗/

Kneron Confidential

- #define SSP_SSPRST (1<<6)
- #define SSP_ACCRST 0x20 /* AC-Link Cold Reset Enable */
- #define SSP_ACWRST 0x10 /* AC-Link Warm Reset Enable */
- #define SSP_TXFCLR 0x8 /* TX FIFO Clear */
- #define SSP_RXFCLR 0x4 /* RX FIFO Clear */
- #define SSP_TXDOE 0x2 /* TX Data Output Enable */
- #define SSP_SSPEN 0x1 /* SSP Enable */
- #define SSP_TFVE 0x3f000 /* Tx FIFO Valid Entries */
- #define SSP_RFVE 0x3f0 /* Rx FIFO Valid Entries */
- #define SSP_BUSY 0x4 /* Busy for recv or tx */
- #define SSP_TFNF 0x2 /* TX FIFO Not Full */
- #define SSP_RFF 0x1 /* RX FIFO Full */
- #define SSP_TXDMAEN 0x20 /* TX DMA Enable */
- #define SSP_RXDMAEN 0x10 /* RX DMA Enable */
- #define SSP_TFIEN 0x8 /* TX FIFO Int Enable */
- #define SSP_RFIEN 0x4 /* RX FIFO Int Enable */
- #define SSP_TFURIEN 0x2 /* TX FIFO Underrun int enable */
- #define SSP_RFORIEN 0x1 /* RX FIFO Overrun int enable */
- #define SSP_AC97FCI 0x10
- #define SSP_TFTHI 0x8 /* TX FIFO Threshold Interrupt */
- #define SSP_RFTHI 0x4 /* RX FIFO Threshold Interrupt */
- #define TFURI 0x2 /* TX FIFO Underrun interrupt */
- #define RFORI 0x1 /* RX FIFO Overrun interrupt */
- #define SSP_TXFIFO_DEPTH 0xFF0000
- #define SSP_RXFIFO_DEPTH 0xFF00
- #define SSP_FIFO_WIDTH 0xFF
- #define MAX_SSP 0x4 /* ssp device number(include AC97 and I2S) */
- #define AC97_SLOT_NUM 13
- #define AC97_TX_BUF_SIZE 16
- #define AC97_RX_BUF_SIZE 16
- #define SSPCK_BASE CPE_PWM_BASE+0x30
- #define SSP1CKDIV 0xF
- #define I2SCKDIV 0xF0
- #define AC97CKDIV 0xF00
- #define SSP2CKDIV 0xF000
- #define UDA1345TS_I2S 0
- #define W6631TS_I2S 1
- #define NORMAL_I2S 2
- #define FSA0AC108_I2S 3
- #define WM8510_I2S 4
- #define PCM3793_I2S 5
- #define WM8731S_I2S 6
- #define WM9081_I2S 7
- #define I2S_FIFO_WIDTH 16

Enumerations

- enum SSP_MODE_T { SSP_AS_MASTER = 0, SSP_AS_SLAVE = 1 }

Kneron Confidential

Functions

- void fLib_I2S_Init (u32, u32, SSP_MODE_T)
- void fLib_WriteSSP (u32, u32)
- u32 fLib_ReadSSP (u32)
- void fLib_SetSSPFrame (u32, u32)
- void fLib_SetSSPFramePolar (u32, u32)
- void fLib_SetSSPOPMode (u32, u32)
- void fLib_SetSSPDataLen (u32, u32)
- void fLib_SetSSPpaddingDataLen (u32, u32)
- void fLib_SetSSPClkdiv (u32, u32)
- void fLib_SSPClearTxFIFO (u32)
- void fLib_SSPClearRxFIFO (u32)
- u32 fLib_ReadSSPStatus (u32)
- void fLib_SetSSPSCLKPO (u32, u32)
- void fLib_SetSSPSCLKPH (u32, u32)
- u32 fLib_ReadSSPIntStatus (u32)
- void fLib_SetSSP_TXFIFO (u32, u32, u32)
- void fLib_SetSSP_RXFIFO (u32, u32, u32)
- void fLib_SetSSP_DMA (u32, u32, u32)
- void fLib_SetSSP_FIFO_Threshold (u32, u32, u32)
- void fLib_SetSSP_WarmReset (u32)
- void fLib_SetSSP_ColdReset (u32)
- u32 fLib_ReadSSP32Bit (u32)
- void fLib_SetSSP_Enable (u32, int)
- void fLib_SetSSP_IntMask (u32, int)
- u32 fLib_SSP_GetRxFIFOValidEntries (u32 base_addr)
- void fLib_InitAC97 (u32)
- void fLib_AC97_SetSlotValidReg (u32, u32)
- int fLib_AC97_ReadRegister (u32, u32, u16 *)
- void fLib_AC97_WriteRegister (u32, u32, u32)
- void fLib_AC97_WriteData (u32, u32 *, u32)
- void fLib_AC97_ReadData (u32, u32 *, u32)
- u32 fLib_AC97_ReadOneWordData (u32)
- int fLib_AC97_ReadRegisterEx (u32, u32, u16 *)
- void fLib_AC97_WriteRegisterEx (u32, u32, u32)
- int fLib_ReturnTxFIFO_Count (u32)
- int fLib_ReturnRxFIFO_Count (u32)
- u32 fLib_SSP_GetTxFIFOLen (u32)
- void fLib_SetSSP_SDL_Write (u32, u32, u32)
- u32 fLib_SSP_busy (u32 base_addr)
- void delay (int ticks)

6.45.1 Macro Definition Documentation

6.45.1.1 AC97_Format

```
#define AC97_Format 0x4000
```

Kneron Confidential

6.45.1.2 AC97_RX_BUF_SIZE

```
#define AC97_RX_BUF_SIZE 16
```

6.45.1.3 AC97_SLOT_NUM

```
#define AC97_SLOT_NUM 13
```

6.45.1.4 AC97_TX_BUF_SIZE

```
#define AC97_TX_BUF_SIZE 16
```

6.45.1.5 AC97CKDIV

```
#define AC97CKDIV 0xF00
```

6.45.1.6 FSA0AC108_I2S

```
#define FSA0AC108_I2S 3
```

6.45.1.7 I2S_FIFO_WIDTH

```
#define I2S_FIFO_WIDTH 16
```

6.45.1.8 I2S_Format

```
#define I2S_Format 0x3000
```

6.45.1.9 I2SCKDIV

```
#define I2SCKDIV 0xF0
```

Kneron Confidential

6.45.1.10 MAX_SSP

```
#define MAX_SSP 0x4 /* ssp device number(include AC97 and I2S) */
```

6.45.1.11 Microwire_Format

```
#define Microwire_Format 0x2000
```

6.45.1.12 NORMAL_I2S

```
#define NORMAL_I2S 2
```

6.45.1.13 PCM3793_I2S

```
#define PCM3793_I2S 5
```

6.45.1.14 RFORI

```
#define RFORI 0x1 /* RX FIFO Overrun interrupt */
```

6.45.1.15 SPI_Format

```
#define SPI_Format 0x1000
```

6.45.1.16 SSP1CKDIV

```
#define SSP1CKDIV 0xF
```

6.45.1.17 SSP2CKDIV

```
#define SSP2CKDIV 0xF000
```

Kneron Confidential

6.45.1.18 SSP_AC97FCI

```
#define SSP_AC97FCI 0x10
```

6.45.1.19 SSP_ACCRST

```
#define SSP_ACCRST 0x20 /* AC-Link Cold Reset Enable */
```

6.45.1.20 SSP_ACLINK_SLOT_VALID

```
#define SSP_ACLINK_SLOT_VALID 0x20
```

6.45.1.21 SSP_ACWRST

```
#define SSP_ACWRST 0x10 /* AC-Link Warm Reset Enable */
```

6.45.1.22 SSP_BUSY

```
#define SSP_BUSY 0x4 /* Busy for recv or tx */
```

6.45.1.23 SSP_CLKDIV

```
#define SSP_CLKDIV 0xFFFF /* clk divider */
```

6.45.1.24 SSP_CONTROL0

```
#define SSP_CONTROL0 0x0
```

6.45.1.25 SSP_CONTROL1

```
#define SSP_CONTROL1 0x4
```

Kneron Confidential

6.45.1.26 SSP_CONTROL2

```
#define SSP_CONTROL2 0x8
```

6.45.1.27 SSP_DATA

```
#define SSP_DATA 0x18
```

6.45.1.28 SSP_DATAJUSTIFY

```
#define SSP_DATAJUSTIFY 0x10 /* data padding in front of serial data */
```

6.45.1.29 SSP_FIFO_WIDTH

```
#define SSP_FIFO_WIDTH 0xFF
```

6.45.1.30 SSP_Format

```
#define SSP_Format 0x0000
```

6.45.1.31 SSP_FSDIST

```
#define SSP_FSDIST 0x300
```

6.45.1.32 SSP_FSPO_HIGH

```
#define SSP_FSPO_HIGH 0x0 /* Frame sync atcive high */
```

6.45.1.33 SSP_FSPO_LOW

```
#define SSP_FSPO_LOW 0x20 /* Frame sync atcive low */
```

Kneron Confidential

6.45.1.34 SSP_INFO

```
#define SSP_INFO 0x1C
```

6.45.1.35 SSP_INT_CONTROL

```
#define SSP_INT_CONTROL 0X10
```

6.45.1.36 SSP_INT_STATUS

```
#define SSP_INT_STATUS 0x14
```

6.45.1.37 SSP_LBM

```
#define SSP_LBM 0x80 /* loopback mode */
```

6.45.1.38 SSP_LSB

```
#define SSP_LSB 0x40 /* LSB first */
```

6.45.1.39 SSP_OPM_MS

```
#define SSP_OPM_MS 0x03
```

6.45.1.40 SSP_OPM_MSMO

```
#define SSP_OPM_MSMO 0x8 /* Master mono mode */
```

6.45.1.41 SSP_OPM_MSST

```
#define SSP_OPM_MSST 0xC /* Master stereo mode */
```

Kneron Confidential

6.45.1.42 SSP_OPM_SL

```
#define SSP_OPM_SL 0x01
```

6.45.1.43 SSP_OPM_SLMO

```
#define SSP_OPM_SLMO 0x0 /* Slave mono mode */
```

6.45.1.44 SSP_OPM_SLST

```
#define SSP_OPM_SLST 0x4 /* Slave stereo mode */
```

6.45.1.45 SSP_PDL

```
#define SSP_PDL 0xFF000000 /* paddinf data length */
```

6.45.1.46 SSP_RFF

```
#define SSP_RFF 0x1 /* RX FIFO Full */
```

6.45.1.47 SSP_RFIEN

```
#define SSP_RFIEN 0x4 /* RX FIFO Int Enable */
```

6.45.1.48 SSP_RFORIEN

```
#define SSP_RFORIEN 0x1 /* RX FIFO Overrun int enable */
```

6.45.1.49 SSP_RFTHI

```
#define SSP_RFTHI 0x4 /* RX FIFO Threshold Interrupt */
```

Kneron Confidential

6.45.1.50 SSP_RFVE

```
#define SSP_RFVE 0x3f0 /* Rx FIFO Valid Entries */
```

6.45.1.51 SSP_RXDMAEN

```
#define SSP_RXDMAEN 0x10 /* RX DMA Enable */
```

6.45.1.52 SSP_RXEN

```
#define SSP_RXEN (1<<7) /* Receive function enable */
```

6.45.1.53 SSP_RXFCLR

```
#define SSP_RXFCLR 0x4 /* RX FIFO Clear */
```

6.45.1.54 SSP_RXFIFO_DEPTH

```
#define SSP_RXFIFO_DEPTH 0xFF00
```

6.45.1.55 SSP_SCLKPH_HALFCLK

```
#define SSP_SCLKPH_HALFCLK 0x1 /* Half CLK cycle */
```

6.45.1.56 SSP_SCLKPH_ONECLK

```
#define SSP_SCLKPH_ONECLK 0x0 /* One CLK cycle */
```

6.45.1.57 SSP_SCLKPO_HIGH

```
#define SSP_SCLKPO_HIGH 0x2 /* SCLK Remain HIGH */
```

Kneron Confidential

6.45.1.58 SSP_SCLKPO_LOW

```
#define SSP_SCLKPO_LOW 0x0 /* SCLK Remain LOW */
```

6.45.1.59 SSP_SDL

```
#define SSP_SDL 0x7F0000 /* Serial data length(actual data length-1) */
```

6.45.1.60 SSP_SSPEN

```
#define SSP_SSPEN 0x1 /* SSP Enable */
```

6.45.1.61 SSP_SSPRST

```
#define SSP_SSPRST (1<<6)
```

6.45.1.62 SSP_STATUS

```
#define SSP_STATUS 0xC
```

6.45.1.63 SSP_TFIEN

```
#define SSP_TFIEN 0x8 /* TX FIFO Int Enable */
```

6.45.1.64 SSP_TFNF

```
#define SSP_TFNF 0x2 /* TX FIFO Not Full */
```

6.45.1.65 SSP_TFTHI

```
#define SSP_TFTHI 0x8 /* TX FIFO Threshold Interrupt */
```

Kneron Confidential

6.45.1.66 SSP_TFURIEN

```
#define SSP_TFURIEN 0x2 /* TX FIFO Underrun int enable */
```

6.45.1.67 SSP_TFVE

```
#define SSP_TFVE 0x3f000 /* Tx FIFO Valid Entries */
```

6.45.1.68 SSP_TXDMAEN

```
#define SSP_TXDMAEN 0x20 /* TX DMA Enable */
```

6.45.1.69 SSP_TXDOE

```
#define SSP_TXDOE 0x2 /* TX Data Output Enable */
```

6.45.1.70 SSP_TXEN

```
#define SSP_TXEN (1<<8) /* Transmit function enable */
```

6.45.1.71 SSP_TXFCLR

```
#define SSP_TXFCLR 0x8 /* TX FIFO Clear */
```

6.45.1.72 SSP_TXFIFO_DEPTH

```
#define SSP_TXFIFO_DEPTH 0xFF0000
```

6.45.1.73 SSPCK_BASE

```
#define SSPCK_BASE CPE_PWM_BASE+0x30
```

Kneron Confidential

6.45.1.74 TFURI

```
#define TFURI 0x2 /* TX FIFO Underrun interrupt */
```

6.45.1.75 UDA1345TS_I2S

```
#define UDA1345TS_I2S 0
```

6.45.1.76 W6631TS_I2S

```
#define W6631TS_I2S 1
```

6.45.1.77 WM8510_I2S

```
#define WM8510_I2S 4
```

6.45.1.78 WM8731S_I2S

```
#define WM8731S_I2S 6
```

6.45.1.79 WM9081_I2S

```
#define WM9081_I2S 7
```

6.45.2 Enumeration Type Documentation

6.45.2.1 SSP_MODE_T

```
enum SSP_MODE_T
```

Kneron Confidential

Enumerator

SSP_AS_MASTER	
SSP_AS_SLAVE	

6.45.3 Function Documentation

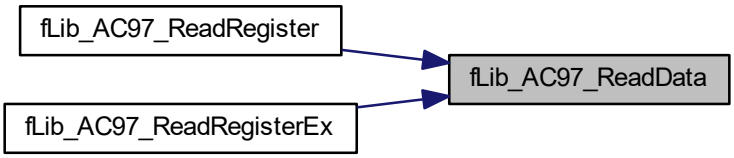
6.45.3.1 delay()

```
void delay (
            int ticks )
```

6.45.3.2 fLib_AC97_ReadData()

```
void fLib_AC97_ReadData (
            u32 ,
            u32 * ,
            u32  )
```

Here is the caller graph for this function:



Kneron Confidential

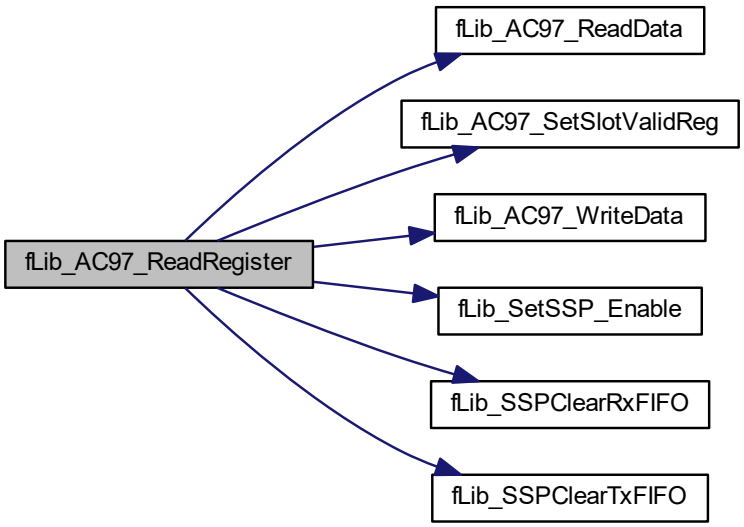
6.45.3.3 fLib_AC97_ReadOneWordData()

```
u32 fLib_AC97_ReadOneWordData (
            u32  )
```

6.45.3.4 fLib_AC97_ReadRegister()

```
int fLib_AC97_ReadRegister (
            u32 ,
            u32 ,
            u16 *  )
```

Here is the call graph for this function:

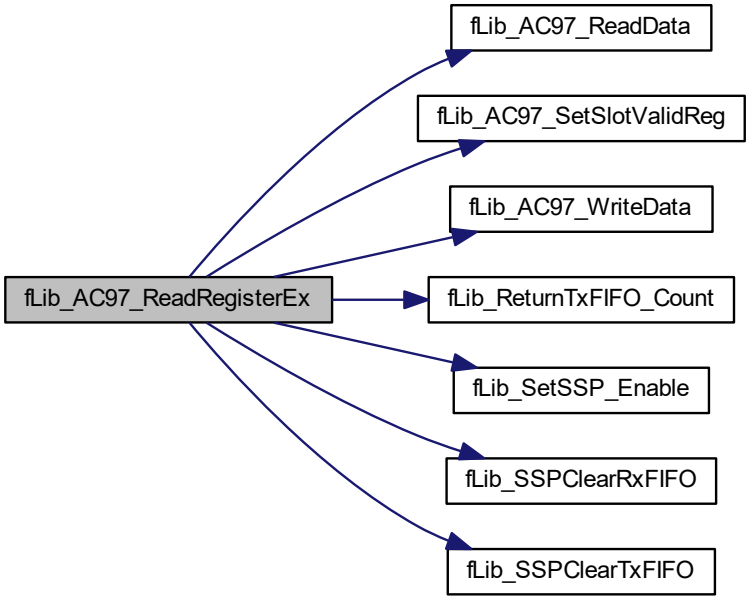


Kneron Confidential

6.45.3.5 fLib_AC97_ReadRegisterEx()

```
int fLib_AC97_ReadRegisterEx (
            u32 ,
            u32 ,
            u16 *  )
```

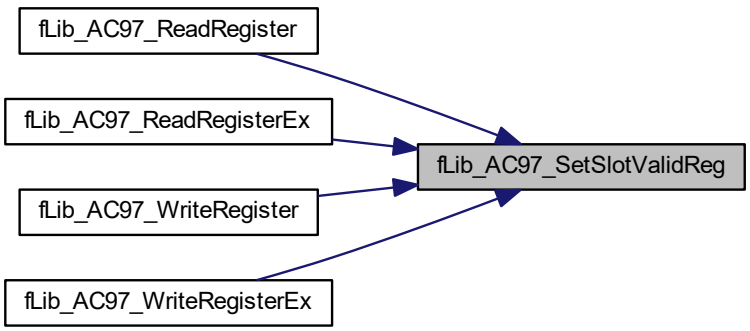
Here is the call graph for this function:



6.45.3.6 fLib_AC97_SetSlotValidReg()

```
void fLib_AC97_SetSlotValidReg (
             u32 ,
             u32  )
```

Here is the caller graph for this function:

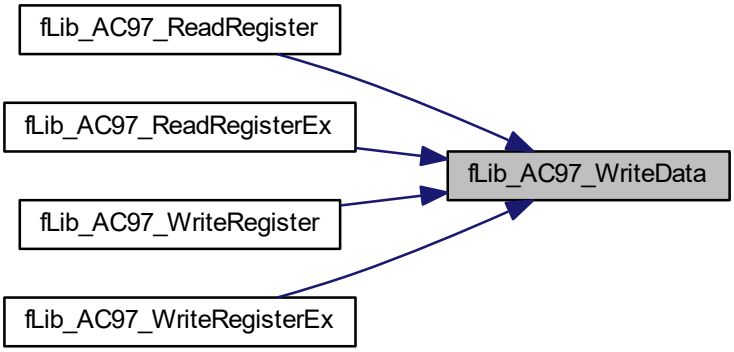


Kneron Confidential

6.45.3.7 fLib_AC97_WriteData()

```
void fLib_AC97_WriteData (
            u32 ,
            u32 * ,
            u32  )
```

Here is the caller graph for this function:

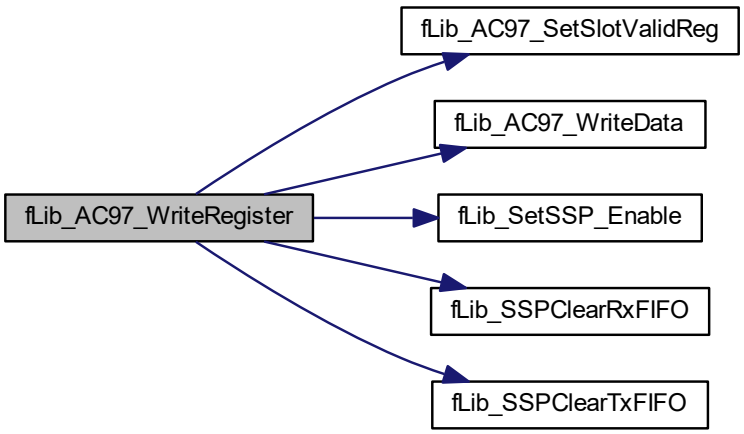


Kneron Confidential

6.45.3.8 fLib_AC97_WriteRegister()

```
void fLib_AC97_WriteRegister (
            u32 ,
            u32 ,
            u32  )
```

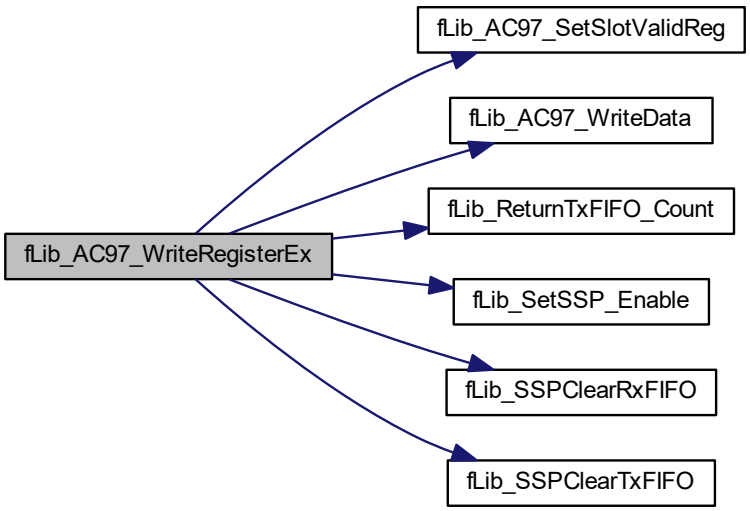
Here is the call graph for this function:



6.45.3.9 fLib_AC97_WriteRegisterEx()

```
void fLib_AC97_WriteRegisterEx (
            u32 ,
            u32 ,
            u32  )
```

Here is the call graph for this function:

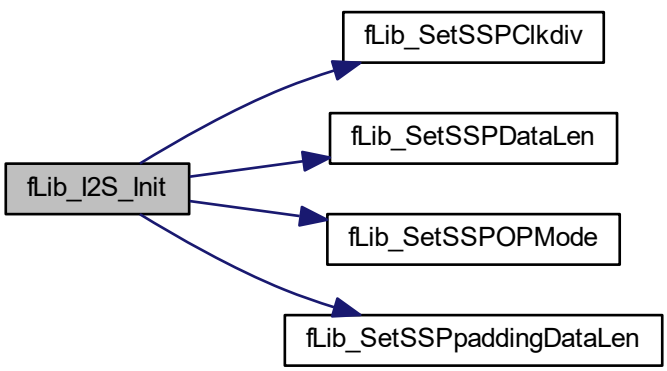


Kneron Confidential

6.45.3.10 fLib_I2S_Init()

```
void fLib_I2S_Init (
            u32 ,
            u32 ,
            SSP_MODE_T  )
```

Here is the call graph for this function:

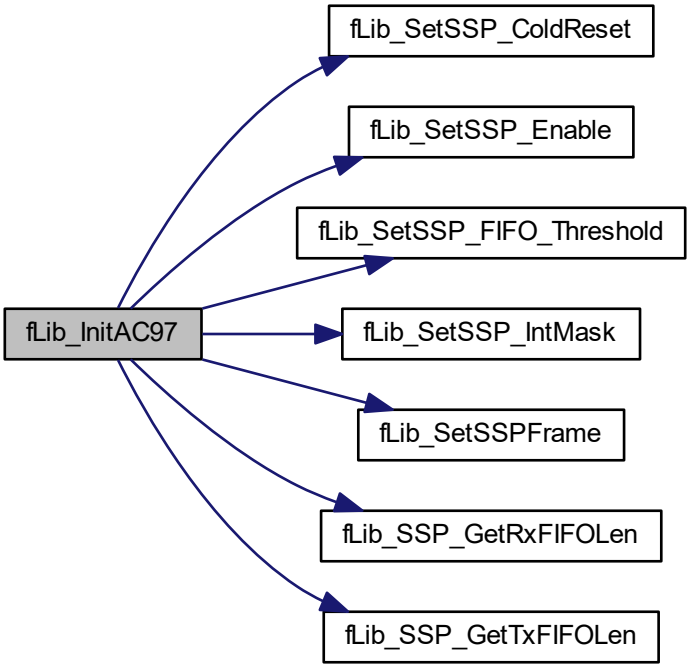


Kneron Confidential

6.45.3.11 fLib_InitAC97()

```
void fLib_InitAC97 (
            u32  )
```

Here is the call graph for this function:



6.45.3.12 fLib_ReadSSP()

```
u32 fLib_ReadSSP (
              u32  )
```

6.45.3.13 fLib_ReadSSP32Bit()

```
u32 fLib_ReadSSP32Bit (
              u32  )
```

6.45.3.14 fLib_ReadSSPIntStatus()

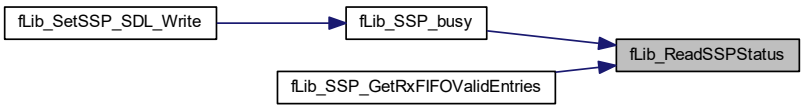
```
u32 fLib_ReadSSPIntStatus (
              u32  )
```

Kneron Confidential

6.45.3.15 fLib_ReadSSPStatus()

```
u32 fLib_ReadSSPStatus (
            u32  )
```

Here is the caller graph for this function:



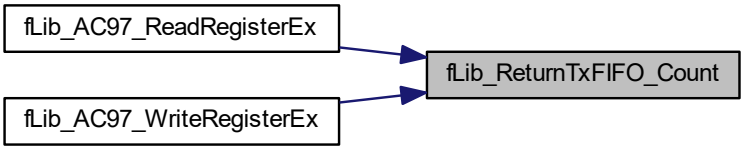
6.45.3.16 fLib_ReturnRxFIFO_Count()

```
int fLib_ReturnRxFIFO_Count (
            u32  )
```

6.45.3.17 fLib_ReturnTxFIFO_Count()

```
int fLib_ReturnTxFIFO_Count (
            u32  )
```

Here is the caller graph for this function:

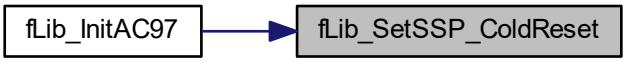


Kneron Confidential

6.45.3.18 fLib_SetSSP_ColdReset()

```
void fLib_SetSSP_ColdReset (
            u32  )
```

Here is the caller graph for this function:



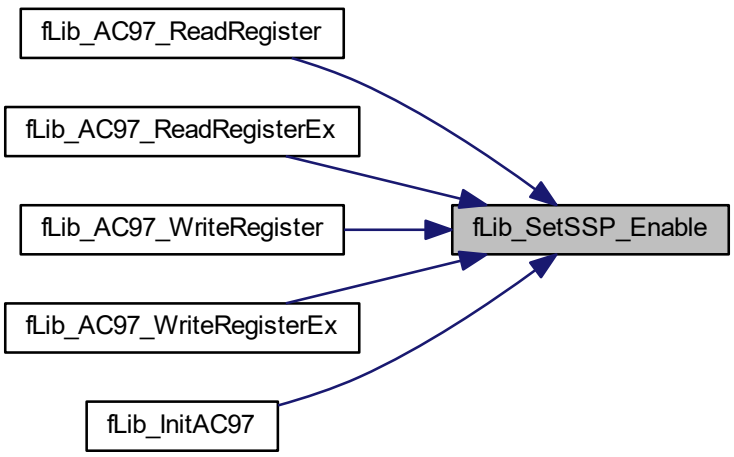
6.45.3.19 fLib_SetSSP_DMA()

```
void fLib_SetSSP_DMA (
            u32 ,
            u32 ,
            u32  )
```

6.45.3.20 fLib_SetSSP_Enable()

```
void fLib_SetSSP_Enable (
            u32 ,
            int  )
```

Here is the caller graph for this function:

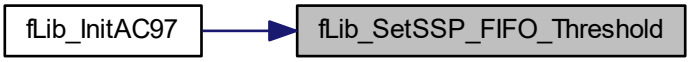


Kneron Confidential

6.45.3.21 fLib_SetSSP_FIFO_Threshold()

```
void fLib_SetSSP_FIFO_Threshold (
            u32 ,
            u32 ,
            u32  )
```

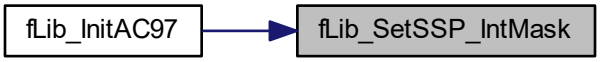
Here is the caller graph for this function:



6.45.3.22 fLib_SetSSP_IntMask()

```
void fLib_SetSSP_IntMask (
            u32 ,
            int  )
```

Here is the caller graph for this function:



6.45.3.23 fLib_SetSSP_RXFIFO()

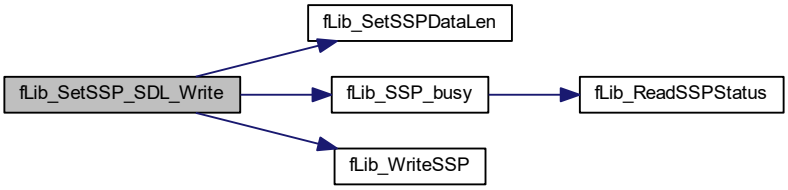
```
void fLib_SetSSP_RXFIFO (
            u32 ,
            u32 ,
            u32  )
```

Kneron Confidential

6.45.3.24 fLib_SetSSP_SDL_Write()

```
void fLib_SetSSP_SDL_Write (
            u32 ,
            u32 ,
            u32  )
```

Here is the call graph for this function:



6.45.3.25 fLib_SetSSP_TXFIFO()

```
void fLib_SetSSP_TXFIFO (
            u32 ,
            u32 ,
            u32  )
```

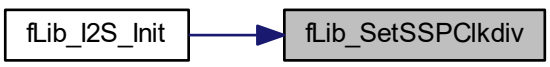
6.45.3.26 fLib_SetSSP_WarmReset()

```
void fLib_SetSSP_WarmReset (
            u32  )
```

6.45.3.27 fLib_SetSSPClkdiv()

```
void fLib_SetSSPClkdiv (
            u32 ,
            u32  )
```

Here is the caller graph for this function:

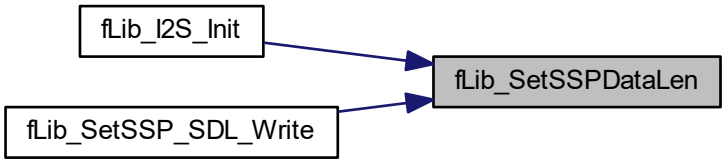


Kneron Confidential

6.45.3.28 fLib_SetSSPDataLen()

```
void fLib_SetSSPDataLen (
            u32 ,
            u32  )
```

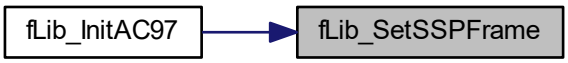
Here is the caller graph for this function:



6.45.3.29 fLib_SetSSPFrame()

```
void fLib_SetSSPFrame (
            u32 ,
            u32  )
```

Here is the caller graph for this function:



6.45.3.30 fLib_SetSSPFramePolar()

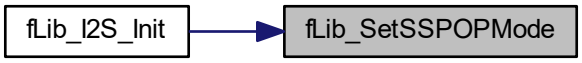
```
void fLib_SetSSPFramePolar (
            u32 ,
            u32  )
```

Kneron Confidential

6.45.3.31 fLib_SetSSPOPMode()

```
void fLib_SetSSPOPMode (
            u32 ,
            u32  )
```

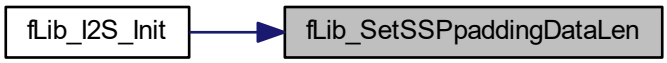
Here is the caller graph for this function:



6.45.3.32 fLib_SetSSPpaddingDataLen()

```
void fLib_SetSSPpaddingDataLen (
            u32 ,
            u32  )
```

Here is the caller graph for this function:



6.45.3.33 fLib_SetSSPSCLKPH()

```
void fLib_SetSSPSCLKPH (
            u32 ,
            u32  )
```

Kneron Confidential

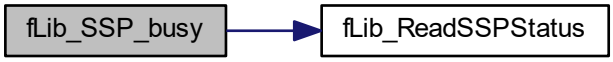
6.45.3.34 fLib_SetSSPSCLKPO()

```
void fLib_SetSSPSCLKPO (
            u32 ,
            u32  )
```

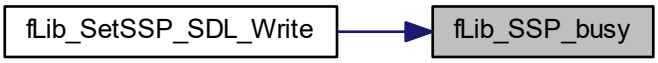
6.45.3.35 fLib_SSP_busy()

```
u32 fLib_SSP_busy (
            u32 base_addr )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.45.3.36 fLib_SSP_GetRxFIFOValidEntries()

```
u32 fLib_SSP_GetRxFIFOValidEntries (
            u32 base_addr )
```

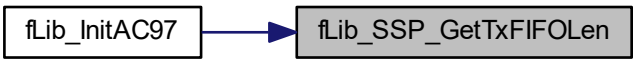
Here is the call graph for this function:

Kneron Confidential

6.45.3.37 fLib_SSP_GetTxFIFOLen()

```
u32 fLib_SSP_GetTxFIFOLen (
              u32  )
```

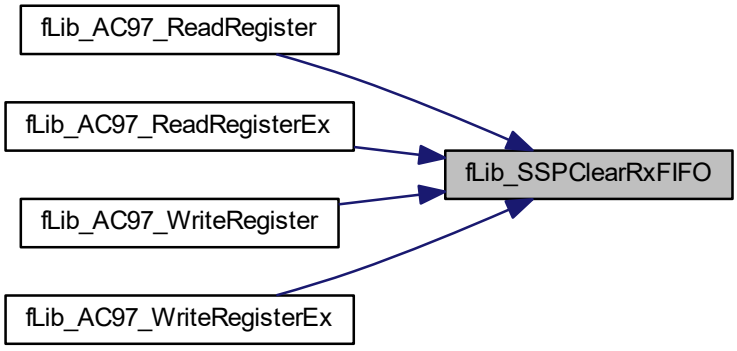
Here is the caller graph for this function:



6.45.3.38 fLib_SSPClearRxFIFO()

```
void fLib_SSPClearRxFIFO (
              u32  )
```

Here is the caller graph for this function:

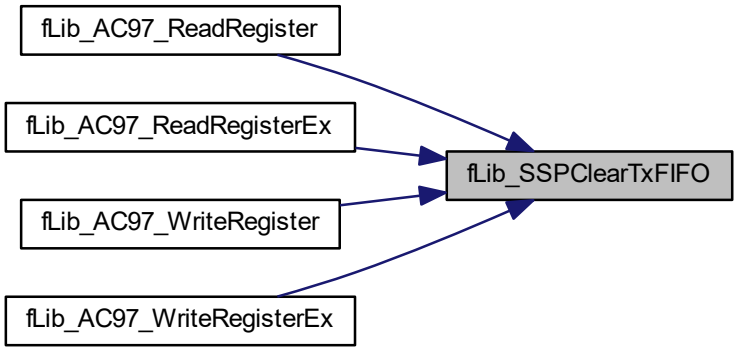


Kneron Confidential

6.45.3.39 fLib_SSPClearTxFIFO()

```
void fLib_SSPClearTxFIFO (
            u32  )
```

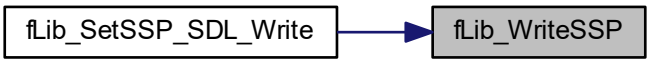
Here is the caller graph for this function:



6.45.3.40 fLib_WriteSSP()

```
void fLib_WriteSSP (
            u32 ,
            u32  )
```

Here is the caller graph for this function:



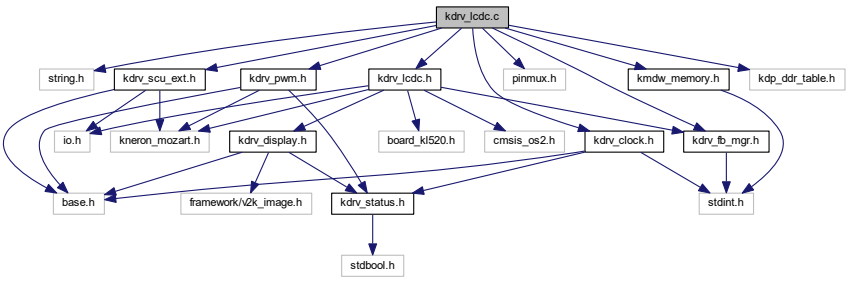
Kneron Confidential

6.46 kdrv_lcdc.c File Reference

```
#include <string.h>
#include "kdrv_scu_ext.h"
#include "kdrv_clock.h"
#include "kdrv_lcdc.h"
```

```
#include "pinmux.h"
#include "kdrv_fb_mgr.h"
#include "kmdw_memory.h"
#include "kdrv_pwm.h"
#include "kdp_ddr_table.h"
```

Include dependency graph for kdrv_lcdc.c:



Data Structures

- struct kdrv_lcdc_fb_t
- struct lcdc_img_pixfmt_pxp

Macros

- #define lcdc_msg(fmt, ...)
- #define PWM0_FREQ_CNT (2000000)
- #define LCDC_REG_FUNC_ENABLE (LCD_FTLCDC210_PA_BASE + 0x0000)
- #define LCDC_REG_PANEL_PIXEL (LCD_FTLCDC210_PA_BASE + 0x0004)
- #define LCDC_REG_INTR_ENABLE_MASK (LCD_FTLCDC210_PA_BASE + 0x0008)
- #define LCDC_REG_INTR_CLEAR (LCD_FTLCDC210_PA_BASE + 0x000C)
- #define LCDC_REG_INTR_STATUS (LCD_FTLCDC210_PA_BASE + 0x0010)
- #define LCDC_REG_FRAME_BUFFER (LCD_FTLCDC210_PA_BASE + 0x0014)
- #define LCDC_REG_PANEL_IMAGE0_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x0018)
- #define LCDC_REG_PANEL_IMAGE1_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x0024)
- #define LCDC_REG_PANEL_IMAGE2_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x0030)
- #define LCDC_REG_PANEL_IMAGE3_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x003C)
- #define LCDC_REG_PATGEN_PATTERN_BAR_DISTANCE (LCD_FTLCDC210_PA_BASE + 0x0048)
- #define LCDC_REG_FIFO_THRESHOLD (LCD_FTLCDC210_PA_BASE + 0x004C)
- #define LCDC_REG_BANDWIDTH_CTRL (LCD_FTLCDC210_PA_BASE + 0x0050)
- #define LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM (LCD_FTLCDC210_PA_BASE + 0x0060)
- #define LCDC_REG_HORIZONTAL_TIMING_CTRL (LCD_FTLCDC210_PA_BASE + 0x0100)
- #define LCDC_REG_VERTICAL_TIMING_CTRL (LCD_FTLCDC210_PA_BASE + 0x0104)
- #define LCDC_REG_VERTICAL_BACK_PORCH (LCD_FTLCDC210_PA_BASE + 0x0108)
- #define LCDC_REG_POLARITY_CTRL (LCD_FTLCDC210_PA_BASE + 0x010C)
- #define LCDC_REG_SERIAL_PANEL_PIXEL (LCD_FTLCDC210_PA_BASE + 0x0200)
- #define LCDC_REG_PIPPOP_FMT_1 (LCD_FTLCDC210_PA_BASE + 0x0318)
- #define LCDC_REG_COLOR_MGR_0 (LCD_FTLCDC210_PA_BASE + 0x0400)
- #define LCDC_REG_COLOR_MGR_1 (LCD_FTLCDC210_PA_BASE + 0x0404)
- #define LCDC_REG_COLOR_MGR_2 (LCD_FTLCDC210_PA_BASE + 0x0408)
- #define LCDC_REG_COLOR_MGR_3 (LCD_FTLCDC210_PA_BASE + 0x040C)

Kneron Confidential

- #define LCDC_REG_LT_OF_GAMMA_RED (LCD_FTLCDC210_PA_BASE + 0x0600)
- #define LCDC_REG_LT_OF_GAMMA_GREEN (LCD_FTLCDC210_PA_BASE + 0x0700)
- #define LCDC_REG_LT_OF_GAMMA_BLUE (LCD_FTLCDC210_PA_BASE + 0x0800)
- #define LCDC_REG_PALETTE (LCD_FTLCDC210_PA_BASE + 0x0A00)
- #define LCDC_REG_SCALER_HOR_RES_IN (LCD_FTLCDC210_PA_BASE + 0x1100)
- #define LCDC_REG_SCALER_VER_RES_IN (LCD_FTLCDC210_PA_BASE + 0x1104)
- #define LCDC_REG_SCALER_HOR_RES_OUT (LCD_FTLCDC210_PA_BASE + 0x1108)
- #define LCDC_REG_SCALER_VER_RES_OUT (LCD_FTLCDC210_PA_BASE + 0x110C)
- #define LCDC_REG_SCALER_MISC (LCD_FTLCDC210_PA_BASE + 0x1110)
- #define LCDC_REG_SCALER_RES (LCD_FTLCDC210_PA_BASE + 0x112C)
- #define LCDC_REG_FUNC_ENABLE_SET_PenGen(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 14)
- #define LCDC_REG_FUNC_ENABLE_SET_PiPEn(val) SET_MASKED_BITS(LCDC_REG_FUNC_ENABLE, val, 10, 11)
- #define LCDC_REG_FUNC_ENABLE_SET_BlendEn(val) SET_MASKED_BITS(LCDC_REG_FUNC_ENABLE, val, 8, 9)
- #define LCDC_REG_FUNC_ENABLE_SET_ScalerEn(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 5)
- #define LCDC_REG_FUNC_ENABLE_SET_OSDEn(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 4)
- #define LCDC_REG_FUNC_ENABLE_SET_EnYCbCr(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 3)
- #define LCDC_REG_FUNC_ENABLE_SET_EnYCbCr420(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 2)
- #define LCDC_REG_FUNC_ENABLE_SET_LCDon(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 1)
- #define LCDC_REG_FUNC_ENABLE_SET_LCDen(val) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 0)
- #define LCDC_REG_FUNC_ENABLE_GET_Values() GET_BITS(LCDC_REG_FUNC_ENABLE, 0, 19)
- #define LCDC_REG_PANEL_PIXEL_SET_AddrUpdate(val) SET_MASKED_BIT(LCDC_REG_PANEL_PIXEL, val, 16)
- #define LCDC_REG_PANEL_PIXEL_SET_UpdateSrc(val) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 14, 15)
- #define LCDC_REG_PANEL_PIXEL_SET_DitherType(val) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 12, 13)
- #define LCDC_REG_PANEL_PIXEL_SET_PanelType(val) SET_MASKED_BIT(LCDC_REG_PANEL_PIXEL, val, 11)
- #define LCDC_REG_PANEL_PIXEL_SET_Vcomp(val) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 9, 10)
- #define LCDC_REG_PANEL_PIXEL_SET_RGBTYPE(val) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 7, 8)
- #define LCDC_REG_PANEL_PIXEL_SET_Endian(val) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 5, 6)
- #define LCDC_REG_PANEL_PIXEL_SET_BGRSW(val) SET_MASKED_BIT(LCDC_REG_PANEL_PIXEL, val, 4)
- #define LCDC_REG_PANEL_PIXEL_SET_BppFifo(val) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 0, 2)
- #define LCDC_REG_PANEL_PIXEL_GET_Values() GET_BITS(LCDC_REG_PANEL_PIXEL, 0, 20)
- #define LCDC_REG_PANEL_PIXEL_RGBTYPE_565 (0x0 << 7)
- #define LCDC_REG_PANEL_PIXEL_RGBTYPE_555 (0x1 << 7)
- #define LCDC_REG_PANEL_PIXEL_RGBTYPE_444 (0x2 << 7)
- #define LCDC_REG_PANEL_PIXEL_RGBTYPE_MASK (BIT7 | BIT8)
- #define LCDC_REG_PANEL_PIXEL_BppFifo_1bpp (0x0)
- #define LCDC_REG_PANEL_PIXEL_BppFifo_2bpp (0x1)
- #define LCDC_REG_PANEL_PIXEL_BppFifo_4bpp (0x2)
- #define LCDC_REG_PANEL_PIXEL_BppFifo_8bpp (0x3)

Kneron Confidential

- #define LCDC_REG_PANEL_PIXEL_BppFifo_16bpp (0x4)
- #define LCDC_REG_PANEL_PIXEL_BppFifo_24bpp (0x5)
- #define LCDC_REG_PANEL_PIXEL_BppFifo_MASK (BIT0 | BIT1 | BIT2)
- #define LCDC_REG_PANEL_PIXEL_VSync (0x00)
- #define LCDC_REG_PANEL_PIXEL_VBackPorch (0x01)
- #define LCDC_REG_PANEL_PIXEL_VActiveImg (0x02)
- #define LCDC_REG_PANEL_PIXEL_VFrontPorch (0x03)
- #define LCDC_REG_INTR_ENABLE_MASK_SET(val) SET_MASKED_BITS(LCDC_REG_INTR_ENABLE_MASK, val, 0, 3)
- #define LCDC_REG_INTR_ENABLE_MASK_GET_Values() GET_BITS(LCDC_REG_INTR_ENABLE_MASK, 0, 3)
- #define LCDC_REG_INTR_CLEAR_BusErr() SET_BIT(LCDC_REG_INTR_CLEAR, 3)
- #define LCDC_REG_INTR_CLEAR_Vstatus() SET_BIT(LCDC_REG_INTR_CLEAR, 2)
- #define LCDC_REG_INTR_CLEAR_NxtBase() SET_BIT(LCDC_REG_INTR_CLEAR, 1)
- #define LCDC_REG_INTR_CLEAR_FIFOUdn() SET_BIT(LCDC_REG_INTR_CLEAR, 0)
- #define LCDC_REG_INTR_CLEAR_AllStatus(val) SET_MASKED_BITS(LCDC_REG_INTR_CLEAR, val, 0, 3)
- #define LCDC_REG_INTR_STATUS_IntBusErr BIT3
- #define LCDC_REG_INTR_STATUS_IntVstatus BIT2
- #define LCDC_REG_INTR_STATUS_IntNxtBase BIT1
- #define LCDC_REG_INTR_STATUS_IntFIFOUdn BIT0
- #define LCDC_REG_INTR_GET_IntStatus() GET_BITS(LCDC_REG_INTR_STATUS, 0, 3)
- #define LCDC_REG_FRAME_BUFFER_GET_LmScalDownValues() GET_BITS(LCDC_REG_FRAME_BUFFER, 8, 15)
- #define LCDC_REG_FIFO_THRESHOLD_SET_BufThreshold(val) SET_MASKED_BITS(LCDC_REG_FIFO_THRESHOLD, val, 0, 31)
- #define LCDC_REG_FIFO_THRESHOLD_GET_BufThresholdValues() GET_BITS(LCDC_REG_FIFO_THRESHOLD, 0, 31)
- #define LCDC_REG_BANDWIDTH_CTRL_GET_Values() GET_BITS(LCDC_REG_BANDWIDTH_CTRL, 0, 9)
- #define LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM_GET_Buf4Thrshold() GET_BITS(LCDC_REG_FIFO_THRESHOLD 0, 7)
- #define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HBP(val) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMIN val, 24, 31)
- #define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HFP(val) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMIN val, 16, 23)
- #define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HW(val) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMIN val, 8, 15)
- #define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_PL(val) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMING val, 0, 7)
- #define LCDC_REG_HORIZONTAL_TIMING_CTRL_GET_Values() GET_BITS(LCDC_REG_HORIZONTAL_TIMING_CTRL, 0, 31)
- #define LCDC_REG_VERTICAL_TIMING_CTRL_SET_VFP(val) SET_MASKED_BITS(LCDC_REG_VERTICAL_TIMING_CTR val, 24, 31)
- #define LCDC_REG_VERTICAL_TIMING_CTRL_SET_VW(val) SET_MASKED_BITS(LCDC_REG_VERTICAL_TIMING_CTR val, 16, 21)
- #define LCDC_REG_VERTICAL_TIMING_CTRL_SET_LF(val) SET_MASKED_BITS(LCDC_REG_VERTICAL_TIMING_CTRL val, 0, 11)
- #define LCDC_REG_VERTICAL_TIMING_CTRL_GET_Values() GET_BITS(LCDC_REG_VERTICAL_TIMING_CTRL, 0, 31)
- #define LCDC_REG_VERTICAL_BACK_PORCH_SET_VBP(val) SET_MASKED_BITS(LCDC_REG_VERTICAL_BACK_POR val, 0, 7)
- #define LCDC_REG_VERTICAL_BACK_PORCH_GET_VBP() GET_BITS(LCDC_REG_VERTICAL_BACK_PORCH, 0, 7)
- #define LCDC_REG_POLARITY_CTRL_SET_DivNo(val) SET_MASKED_BITS(LCDC_REG_POLARITY_CTRL, val, 8, 14)

Kneron Confidential

- #define LCDC_REG_POLARITY_CTRL_SET_IPWR(val) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 4)
- #define LCDC_REG_POLARITY_CTRL_SET_IDE(val) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 3)
- #define LCDC_REG_POLARITY_CTRL_SET_ICK(val) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 2)
- #define LCDC_REG_POLARITY_CTRL_SET_IHS(val) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 1)
- #define LCDC_REG_POLARITY_CTRL_SET_IVS(val) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 0)
- #define LCDC_REG_POLARITY_CTRL_GET_Values() GET_BITS(LCDC_REG_POLARITY_CTRL, 0, 14)
- #define LCDC_REG_SERIAL_PANEL_PIXEL_SET_AUO052(val) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL, val, 5)
- #define LCDC_REG_SERIAL_PANEL_PIXEL_SET_LSR(val) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL, val, 4)
- #define LCDC_REG_SERIAL_PANEL_PIXEL_SET_ColorSeq(val) SET_MASKED_BITS(LCDC_REG_SERIAL_PANEL_PIXE val, 2, 3)
- #define LCDC_REG_SERIAL_PANEL_PIXEL_SET_DeltaType(val) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL val, 1)
- #define LCDC_REG_SERIAL_PANEL_PIXEL_SET_SerialMode(val) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXE val, 0)
- #define LCDC_REG_SERIAL_PANEL_PIXEL_GET_Values() GET_BITS(LCDC_REG_SERIAL_PANEL_PIXEL, 0, 5)
- #define LCDC_REG_PIPPOP_FMT_1_SET_BppFifo0(val) SET_MASKED_BITS(LCDC_REG_PIPPOP_FMT_1, val, 0, 2)
- #define LCDC_REG_PIPPOP_FMT_1_GET_BppFifo0() GET_BITS(LCDC_REG_PIPPOP_FMT_1, 0, 2)
- #define LCDC_REG_SCALER_HOR_RES_IN_SET_hor_no_in(val) SET_MASKED_BITS(LCDC_REG_SCALER_HOR_RES_ val, 0, 11)
- #define LCDC_REG_SCALER_HOR_RES_IN_GET_hor_no_in() GET_BITS(LCDC_REG_SCALER_HOR_RES_IN, 0, 11)
- #define LCDC_REG_SCALER_VER_RES_IN_SET_ver_no_in(val) SET_MASKED_BITS(LCDC_REG_SCALER_VER_RES_I val, 0, 11)
- #define LCDC_REG_SCALER_VER_RES_IN_GET_ver_no_in() GET_BITS(LCDC_REG_SCALER_VER_RES_IN, 0, 11)
- #define LCDC_REG_SCALER_HOR_RES_OUT_SET_hor_no_out(val) SET_MASKED_BITS(LCDC_REG_SCALER_HOR_R val, 0, 13)
- #define LCDC_REG_SCALER_HOR_RES_OUT_GET_hor_no_out() GET_BITS(LCDC_REG_SCALER_HOR_RES_OUT, 0, 13)
- #define LCDC_REG_SCALER_VER_RES_OUT_SET_ver_no_out(val) SET_MASKED_BITS(LCDC_REG_SCALER_VER_RE val, 0, 13)
- #define LCDC_REG_SCALER_VER_RES_OUT_GET_ver_no_out() GET_BITS(LCDC_REG_SCALER_VER_RES_OUT, 0, 13)
- #define LCDC_REG_SCALER_MISC_SET_fir_sel(val) SET_MASKED_BITS(LCDC_REG_SCALER_MISC, val, 6, 8)
- #define LCDC_REG_SCALER_MISC_SET_hor_inter_mode(val) SET_MASKED_BITS(LCDC_REG_SCALER_MISC, val, 3, 4)
- #define LCDC_REG_SCALER_MISC_SET_ver_inter_mode(val) SET_MASKED_BITS(LCDC_REG_SCALER_MISC, val, 1, 2)
- #define LCDC_REG_SCALER_MISC_SET_bypass_mode(val) SET_MASKED_BIT(LCDC_REG_SCALER_MISC, val, 0)
- #define LCDC_REG_SCALER_MISC_GET_Values() GET_BITS(LCDC_REG_SCALER_MISC, 0, 8)
- #define LCDC_REG_SCALER_RES_SET_scal_hor_num(val) SET_MASKED_BITS(LCDC_REG_SCALER_RES, val, 8, 15)
- #define LCDC_REG_SCALER_RES_SET_scal_ver_num(val) SET_MASKED_BITS(LCDC_REG_SCALER_RES, val, 0, 7)
- #define LCDC_REG_SCALER_RES_GET_Values() GET_BITS(LCDC_REG_SCALER_RES, 0, 15)
- #define LCDC_DOWN_SCALE_WITH_CROP (NO)
- #define SWAP(a, b)

Kneron Confidential

Functions

- void lcd_update_snapshot_data (uint32_t addr)
- void lcdc_vstatus_isr (void)
- kdrv_display_t * kdrv_display_initialize (void)

  Initialize display driver.
- kdrv_status_t kdrv_display_buffer_initialize (struct video_input_params *params)

  Initialize display frame buffer.
- uint32_t kdrv_display_get_buffer (void)
- kdrv_status_t kdrv_display_set_params (kdrv_display_t *display_drv, struct video_input_params *params)

  Set display parameters.
- kdrv_status_t kdrv_display_get_params (kdrv_display_t *display_drv, struct video_input_params *params)

  Get display parameters.
- kdrv_status_t kdrv_display_set_camera (kdrv_display_t *display_drv, uint8_t cam_idx)

  Set camera source which will be preview on display.
- kdrv_status_t kdrv_display_start (kdrv_display_t *display_drv)

  Start display image preview.
- kdrv_status_t kdrv_display_stop (kdrv_display_t *display_drv)

  Stop display image preview.
- kdrv_status_t kdrv_display_set_pen (kdrv_display_t *display_drv, uint16_t color, uint32_t width)

  Set pen setting.
- kdrv_status_t kdrv_display_update_draw_fb (kdrv_display_t *display_drv, uint32_t addr, uint8_t cam_idx)

  Update frame buffer which be used to draw something on display.
- kdrv_status_t kdrv_display_draw_static_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_line (kdrv_display_t *display_drv, uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- kdrv_status_t kdrv_display_fill_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle with filling color on display.
- kdrv_status_t kdrv_display_draw_bitmap (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height, void *pBuf)

  Draw bitmap on display.
- kdrv_status_t kdrv_display_test_pattern_gen (kdrv_display_t *display_drv, bool pat_gen)

  Set display backlight.
- kdrv_status_t kdrv_display_set_backlight (kdrv_display_t *display_drv, int light)

  Set display backlight.
- kdrv_status_t kdrv_display_screen_control (kdrv_lcdc_screen_ctrl_t ctrl)

  Control display screen ON/OFF.
- kdrv_status_t kdrv_lcdc_set_panel_type (kdrv_lcdc_panel_type_t type)

  Set TFT panel color depth selection of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_bgrsw (kdrv_lcdc_output_fmt_t format)

  Set output format selection of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_sr (kdrv_lcdc_serial_pix_sr_t rotate)

  Set odd line shift rotate of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_colorseq (kdrv_lcdc_serial_pix_colorseq_t color)

  Set color sequence of odd line of LCD serial panel pixel parameter.

Kneron Confidential

- kdrv_status_t kdrv_lcdc_set_pixel_delta_type (kdrv_lcdc_serial_pix_delta_type_t type)

  Set delta type arrangement of color filter of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_serial_mode (kdrv_lcdc_serial_pix_output_mode_t mode)

  Set RGB serial output mode of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_endian (kdrv_lcdc_fb_data_endianness_t endian_type)

  Set data endian.
- kdrv_status_t kdrv_lcdc_set_auo052_mode (kdrv_lcdc_auo052_mode_t mode)

  Set data endian.
- kdrv_status_t kdrv_lcdc_set_bus_bandwidth_ctrl (uint32_t ctrl)

  Set frame rate of lcdc vsync.
- kdrv_status_t kdrv_lcdc_down_scale (uint16_t hor_no_in, uint16_t hor_no_out, uint16_t ver_no_in, uint16_t ver_no_out)

  Set image down scale.
- kdrv_status_t kdrv_lcdc_set_framerate (int framerate, int width, int height)

  Set frame rate of lcdc vsync.
- kdrv_status_t kdrv_lcdc_set_image_color_params (uint32_t color0, uint32_t color1, uint32_t color2, uint32_t color3)

  Set LCD color management parameter.
- kdrv_status_t kdrv_lcdc_set_frame_buffer (uint32_t img_scal_down)

  Set frame buffer parameter.
- kdrv_status_t kdrv_display_set_frame_margin_len (uint16_t margin_len)
- uint16_t kdrv_display_get_frame_margin_len (void)
- kdrv_status_t kdrv_display_set_review_snapshot_en (bool enable)
- bool kdrv_display_get_review_snapshot_en (void)
- kdrv_status_t kdrv_display_set_snapshot_preview_en (bool enable)
- bool kdrv_display_get_snapshot_preview_en (void)

Variables

- kdrv_lcdc_fb_t kdrv_lcdc_frame_buffer

6.46.1 Macro Definition Documentation

6.46.1.1 LCDC_DOWN_SCALE_WITH_CROP

```
#define LCDC_DOWN_SCALE_WITH_CROP (NO)
```

6.46.1.2 lcdc_msg

```
#define lcdc_msg(
              fmt,
              ... )
```

Kneron Confidential

6.46.1.3 LCDC_REG_BANDWIDTH_CTRL

```
#define LCDC_REG_BANDWIDTH_CTRL (LCD_FTLCDC210_PA_BASE + 0x0050)
```

6.46.1.4 LCDC_REG_BANDWIDTH_CTRL_GET_Values

```
#define LCDC_REG_BANDWIDTH_CTRL_GET_Values( ) GET_BITS(LCDC_REG_BANDWIDTH_CTRL, 0, 9)
```

6.46.1.5 LCDC_REG_COLOR_MGR_0

```
#define LCDC_REG_COLOR_MGR_0 (LCD_FTLCDC210_PA_BASE + 0x0400)
```

6.46.1.6 LCDC_REG_COLOR_MGR_1

```
#define LCDC_REG_COLOR_MGR_1 (LCD_FTLCDC210_PA_BASE + 0x0404)
```

6.46.1.7 LCDC_REG_COLOR_MGR_2

```
#define LCDC_REG_COLOR_MGR_2 (LCD_FTLCDC210_PA_BASE + 0x0408)
```

6.46.1.8 LCDC_REG_COLOR_MGR_3

```
#define LCDC_REG_COLOR_MGR_3 (LCD_FTLCDC210_PA_BASE + 0x040C)
```

6.46.1.9 LCDC_REG_FIFO_THRESHOLD

```
#define LCDC_REG_FIFO_THRESHOLD (LCD_FTLCDC210_PA_BASE + 0x004C)
```

6.46.1.10 LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM

```
#define LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM (LCD_FTLCDC210_PA_BASE + 0x0060)
```

Kneron Confidential

6.46.1.11 LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM_GET_Buf4Thrshold

```
#define LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM_GET_Buf4Thrshold( ) GET_BITS(LCDC_REG_FIFO_THRESHOLD_CTRL_PARAM,
0, 7)
```

6.46.1.12 LCDC_REG_FIFO_THRESHOLD_GET_BufThresholdValues

```
#define LCDC_REG_FIFO_THRESHOLD_GET_BufThresholdValues( ) GET_BITS(LCDC_REG_FIFO_THRESHOLD, 0,
31)
```

6.46.1.13 LCDC_REG_FIFO_THRESHOLD_SET_BufThreshold

```
#define LCDC_REG_FIFO_THRESHOLD_SET_BufThreshold(
             val ) SET_MASKED_BITS(LCDC_REG_FIFO_THRESHOLD, val, 0, 31)
```

6.46.1.14 LCDC_REG_FRAME_BUFFER

```
#define LCDC_REG_FRAME_BUFFER (LCD_FTLCDC210_PA_BASE + 0x0014)
```

6.46.1.15 LCDC_REG_FRAME_BUFFER_GET_LmScalDownValues

```
#define LCDC_REG_FRAME_BUFFER_GET_LmScalDownValues( ) GET_BITS(LCDC_REG_FRAME_BUFFER, 8, 15)
```

6.46.1.16 LCDC_REG_FUNC_ENABLE

```
#define LCDC_REG_FUNC_ENABLE (LCD_FTLCDC210_PA_BASE + 0x0000)
```

6.46.1.17 LCDC_REG_FUNC_ENABLE_GET_Values

```
#define LCDC_REG_FUNC_ENABLE_GET_Values( ) GET_BITS(LCDC_REG_FUNC_ENABLE, 0, 19)
```

Kneron Confidential

6.46.1.18 LCDC_REG_FUNC_ENABLE_SET_BlendEn

```
#define LCDC_REG_FUNC_ENABLE_SET_BlendEn(
              val ) SET_MASKED_BITS(LCDC_REG_FUNC_ENABLE, val, 8, 9)
```

6.46.1.19 LCDC_REG_FUNC_ENABLE_SET_EnYCbCr

```
#define LCDC_REG_FUNC_ENABLE_SET_EnYCbCr(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 3)
```

6.46.1.20 LCDC_REG_FUNC_ENABLE_SET_EnYCbCr420

```
#define LCDC_REG_FUNC_ENABLE_SET_EnYCbCr420(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 2)
```

6.46.1.21 LCDC_REG_FUNC_ENABLE_SET_LCDen

```
#define LCDC_REG_FUNC_ENABLE_SET_LCDen(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 0)
```

6.46.1.22 LCDC_REG_FUNC_ENABLE_SET_LCDon

```
#define LCDC_REG_FUNC_ENABLE_SET_LCDon(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 1)
```

6.46.1.23 LCDC_REG_FUNC_ENABLE_SET_OSDEn

```
#define LCDC_REG_FUNC_ENABLE_SET_OSDEn(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 4)
```

6.46.1.24 LCDC_REG_FUNC_ENABLE_SET_PenGen

```
#define LCDC_REG_FUNC_ENABLE_SET_PenGen(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 14)
```

Kneron Confidential

6.46.1.25 LCDC_REG_FUNC_ENABLE_SET_PiPEn

```
#define LCDC_REG_FUNC_ENABLE_SET_PiPEn(
              val ) SET_MASKED_BITS(LCDC_REG_FUNC_ENABLE, val, 10, 11)
```

6.46.1.26 LCDC_REG_FUNC_ENABLE_SET_ScalerEn

```
#define LCDC_REG_FUNC_ENABLE_SET_ScalerEn(
              val ) SET_MASKED_BIT(LCDC_REG_FUNC_ENABLE, val, 5)
```

6.46.1.27 LCDC_REG_HORIZONTAL_TIMING_CTRL

```
#define LCDC_REG_HORIZONTAL_TIMING_CTRL (LCD_FTLCDC210_PA_BASE + 0x0100)
```

6.46.1.28 LCDC_REG_HORIZONTAL_TIMING_CTRL_GET_Values

```
#define LCDC_REG_HORIZONTAL_TIMING_CTRL_GET_Values( ) GET_BITS(LCDC_REG_HORIZONTAL_TIMING_CTRL,
0, 31)
```

6.46.1.29 LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HBP

```
#define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HBP(
              val ) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMING_CTRL, val, 24, 31)
```

6.46.1.30 LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HFP

```
#define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HFP(
              val ) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMING_CTRL, val, 16, 23)
```

6.46.1.31 LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HW

```
#define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_HW(
              val ) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMING_CTRL, val, 8, 15)
```

Kneron Confidential

6.46.1.32 LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_PL

```
#define LCDC_REG_HORIZONTAL_TIMING_CTRL_SET_PL(
             val ) SET_MASKED_BITS(LCDC_REG_HORIZONTAL_TIMING_CTRL, val, 0, 7)
```

6.46.1.33 LCDC_REG_INTR_CLEAR

```
#define LCDC_REG_INTR_CLEAR (LCD_FTLCDC210_PA_BASE + 0x000C)
```

6.46.1.34 LCDC_REG_INTR_CLEAR_AllStatus

```
#define LCDC_REG_INTR_CLEAR_AllStatus(
             val ) SET_MASKED_BITS(LCDC_REG_INTR_CLEAR, val, 0, 3)
```

6.46.1.35 LCDC_REG_INTR_CLEAR_BusErr

```
#define LCDC_REG_INTR_CLEAR_BusErr( ) SET_BIT(LCDC_REG_INTR_CLEAR, 3)
```

6.46.1.36 LCDC_REG_INTR_CLEAR_FIFOUdn

```
#define LCDC_REG_INTR_CLEAR_FIFOUdn( ) SET_BIT(LCDC_REG_INTR_CLEAR, 0)
```

6.46.1.37 LCDC_REG_INTR_CLEAR_NxtBase

```
#define LCDC_REG_INTR_CLEAR_NxtBase( ) SET_BIT(LCDC_REG_INTR_CLEAR, 1)
```

6.46.1.38 LCDC_REG_INTR_CLEAR_Vstatus

```
#define LCDC_REG_INTR_CLEAR_Vstatus( ) SET_BIT(LCDC_REG_INTR_CLEAR, 2)
```

Kneron Confidential

6.46.1.39 LCDC_REG_INTR_ENABLE_MASK

```
#define LCDC_REG_INTR_ENABLE_MASK (LCD_FTLCDC210_PA_BASE + 0x0008)
```

6.46.1.40 LCDC_REG_INTR_ENABLE_MASK_GET_Values

```
#define LCDC_REG_INTR_ENABLE_MASK_GET_Values( ) GET_BITS(LCDC_REG_INTR_ENABLE_MASK, 0, 3)
```

6.46.1.41 LCDC_REG_INTR_ENABLE_MASK_SET

```
#define LCDC_REG_INTR_ENABLE_MASK_SET(
              val ) SET_MASKED_BITS(LCDC_REG_INTR_ENABLE_MASK, val, 0, 3)
```

6.46.1.42 LCDC_REG_INTR_GET_IntStatus

```
#define LCDC_REG_INTR_GET_IntStatus( ) GET_BITS(LCDC_REG_INTR_STATUS, 0, 3)
```

6.46.1.43 LCDC_REG_INTR_STATUS

```
#define LCDC_REG_INTR_STATUS (LCD_FTLCDC210_PA_BASE + 0x0010)
```

6.46.1.44 LCDC_REG_INTR_STATUS_IntBusErr

```
#define LCDC_REG_INTR_STATUS_IntBusErr BIT3
```

6.46.1.45 LCDC_REG_INTR_STATUS_IntFIFOUdn

```
#define LCDC_REG_INTR_STATUS_IntFIFOUdn BIT0
```

Kneron Confidential

6.46.1.46 LCDC_REG_INTR_STATUS_IntNxtBase

```
#define LCDC_REG_INTR_STATUS_IntNxtBase BIT1
```

6.46.1.47 LCDC_REG_INTR_STATUS_IntVstatus

```
#define LCDC_REG_INTR_STATUS_IntVstatus BIT2
```

6.46.1.48 LCDC_REG_LT_OF_GAMMA_BLUE

```
#define LCDC_REG_LT_OF_GAMMA_BLUE (LCD_FTLCDC210_PA_BASE + 0x0800)
```

6.46.1.49 LCDC_REG_LT_OF_GAMMA_GREEN

```
#define LCDC_REG_LT_OF_GAMMA_GREEN (LCD_FTLCDC210_PA_BASE + 0x0700)
```

6.46.1.50 LCDC_REG_LT_OF_GAMMA_RED

```
#define LCDC_REG_LT_OF_GAMMA_RED (LCD_FTLCDC210_PA_BASE + 0x0600)
```

6.46.1.51 LCDC_REG_PALETTE

```
#define LCDC_REG_PALETTE (LCD_FTLCDC210_PA_BASE + 0x0A00)
```

6.46.1.52 LCDC_REG_PANEL_IMAGE0_FRAME0

```
#define LCDC_REG_PANEL_IMAGE0_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x0018)
```

6.46.1.53 LCDC_REG_PANEL_IMAGE1_FRAME0

```
#define LCDC_REG_PANEL_IMAGE1_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x0024)
```

Kneron Confidential

6.46.1.54 LCDC_REG_PANEL_IMAGE2_FRAME0

```
#define LCDC_REG_PANEL_IMAGE2_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x0030)
```

6.46.1.55 LCDC_REG_PANEL_IMAGE3_FRAME0

```
#define LCDC_REG_PANEL_IMAGE3_FRAME0 (LCD_FTLCDC210_PA_BASE + 0x003C)
```

6.46.1.56 LCDC_REG_PANEL_PIXEL

```
#define LCDC_REG_PANEL_PIXEL (LCD_FTLCDC210_PA_BASE + 0x0004)
```

6.46.1.57 LCDC_REG_PANEL_PIXEL_BppFifo_16bpp

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_16bpp (0x4)
```

6.46.1.58 LCDC_REG_PANEL_PIXEL_BppFifo_1bpp

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_1bpp (0x0)
```

6.46.1.59 LCDC_REG_PANEL_PIXEL_BppFifo_24bpp

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_24bpp (0x5)
```

6.46.1.60 LCDC_REG_PANEL_PIXEL_BppFifo_2bpp

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_2bpp (0x1)
```

6.46.1.61 LCDC_REG_PANEL_PIXEL_BppFifo_4bpp

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_4bpp (0x2)
```

Kneron Confidential

6.46.1.62 LCDC_REG_PANEL_PIXEL_BppFifo_8bpp

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_8bpp (0x3)
```

6.46.1.63 LCDC_REG_PANEL_PIXEL_BppFifo_MASK

```
#define LCDC_REG_PANEL_PIXEL_BppFifo_MASK (BIT0 | BIT1 | BIT2)
```

6.46.1.64 LCDC_REG_PANEL_PIXEL_GET_Values

```
#define LCDC_REG_PANEL_PIXEL_GET_Values( ) GET_BITS(LCDC_REG_PANEL_PIXEL, 0, 20)
```

6.46.1.65 LCDC_REG_PANEL_PIXEL_RGBTYPE_444

```
#define LCDC_REG_PANEL_PIXEL_RGBTYPE_444 (0x2 << 7)
```

6.46.1.66 LCDC_REG_PANEL_PIXEL_RGBTYPE_555

```
#define LCDC_REG_PANEL_PIXEL_RGBTYPE_555 (0x1 << 7)
```

6.46.1.67 LCDC_REG_PANEL_PIXEL_RGBTYPE_565

```
#define LCDC_REG_PANEL_PIXEL_RGBTYPE_565 (0x0 << 7)
```

6.46.1.68 LCDC_REG_PANEL_PIXEL_RGBTYPE_MASK

```
#define LCDC_REG_PANEL_PIXEL_RGBTYPE_MASK (BIT7 | BIT8)
```

6.46.1.69 LCDC_REG_PANEL_PIXEL_SET_AddrUpdate

```
#define LCDC_REG_PANEL_PIXEL_SET_AddrUpdate(
             val ) SET_MASKED_BIT(LCDC_REG_PANEL_PIXEL, val, 16)
```

Kneron Confidential

6.46.1.70 LCDC_REG_PANEL_PIXEL_SET_BGRSW

```
#define LCDC_REG_PANEL_PIXEL_SET_BGRSW(
              val ) SET_MASKED_BIT(LCDC_REG_PANEL_PIXEL, val, 4)
```

6.46.1.71 LCDC_REG_PANEL_PIXEL_SET_BppFifo

```
#define LCDC_REG_PANEL_PIXEL_SET_BppFifo(
              val ) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 0, 2)
```

6.46.1.72 LCDC_REG_PANEL_PIXEL_SET_DitherType

```
#define LCDC_REG_PANEL_PIXEL_SET_DitherType(
              val ) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 12, 13)
```

6.46.1.73 LCDC_REG_PANEL_PIXEL_SET_Endian

```
#define LCDC_REG_PANEL_PIXEL_SET_Endian(
              val ) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 5, 6)
```

6.46.1.74 LCDC_REG_PANEL_PIXEL_SET_PanelType

```
#define LCDC_REG_PANEL_PIXEL_SET_PanelType(
              val ) SET_MASKED_BIT(LCDC_REG_PANEL_PIXEL, val, 11)
```

6.46.1.75 LCDC_REG_PANEL_PIXEL_SET_RGBTYPE

```
#define LCDC_REG_PANEL_PIXEL_SET_RGBTYPE(
              val ) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 7, 8)
```

6.46.1.76 LCDC_REG_PANEL_PIXEL_SET_UpdateSrc

```
#define LCDC_REG_PANEL_PIXEL_SET_UpdateSrc(
              val ) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 14, 15)
```

Kneron Confidential

6.46.1.77 LCDC_REG_PANEL_PIXEL_SET_Vcomp

```
#define LCDC_REG_PANEL_PIXEL_SET_Vcomp(
             val ) SET_MASKED_BITS(LCDC_REG_PANEL_PIXEL, val, 9, 10)
```

6.46.1.78 LCDC_REG_PANEL_PIXEL_VActiveImg

```
#define LCDC_REG_PANEL_PIXEL_VActiveImg (0x02)
```

6.46.1.79 LCDC_REG_PANEL_PIXEL_VBackPorch

```
#define LCDC_REG_PANEL_PIXEL_VBackPorch (0x01)
```

6.46.1.80 LCDC_REG_PANEL_PIXEL_VFrontPorch

```
#define LCDC_REG_PANEL_PIXEL_VFrontPorch (0x03)
```

6.46.1.81 LCDC_REG_PANEL_PIXEL_VSync

```
#define LCDC_REG_PANEL_PIXEL_VSync (0x00)
```

6.46.1.82 LCDC_REG_PATGEN_PATTERN_BAR_DISTANCE

```
#define LCDC_REG_PATGEN_PATTERN_BAR_DISTANCE (LCD_FTLCDC210_PA_BASE + 0x0048)
```

6.46.1.83 LCDC_REG_PIPPOP_FMT_1

```
#define LCDC_REG_PIPPOP_FMT_1 (LCD_FTLCDC210_PA_BASE + 0x0318)
```

Kneron Confidential

6.46.1.84 LCDC_REG_PIPPOP_FMT_1_GET_BppFifo0

```
#define LCDC_REG_PIPPOP_FMT_1_GET_BppFifo0( ) GET_BITS(LCDC_REG_PIPPOP_FMT_1, 0, 2)
```

6.46.1.85 LCDC_REG_PIPPOP_FMT_1_SET_BppFifo0

```
#define LCDC_REG_PIPPOP_FMT_1_SET_BppFifo0(
             val ) SET_MASKED_BITS(LCDC_REG_PIPPOP_FMT_1, val, 0, 2)
```

6.46.1.86 LCDC_REG_POLARITY_CTRL

```
#define LCDC_REG_POLARITY_CTRL (LCD_FTLCDC210_PA_BASE + 0x010C)
```

6.46.1.87 LCDC_REG_POLARITY_CTRL_GET_Values

```
#define LCDC_REG_POLARITY_CTRL_GET_Values( ) GET_BITS(LCDC_REG_POLARITY_CTRL, 0, 14)
```

6.46.1.88 LCDC_REG_POLARITY_CTRL_SET_DivNo

```
#define LCDC_REG_POLARITY_CTRL_SET_DivNo(
             val ) SET_MASKED_BITS(LCDC_REG_POLARITY_CTRL, val, 8, 14)
```

6.46.1.89 LCDC_REG_POLARITY_CTRL_SET_ICK

```
#define LCDC_REG_POLARITY_CTRL_SET_ICK(
             val ) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 2)
```

6.46.1.90 LCDC_REG_POLARITY_CTRL_SET_IDE

```
#define LCDC_REG_POLARITY_CTRL_SET_IDE(
             val ) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 3)
```

Kneron Confidential

6.46.1.91 LCDC_REG_POLARITY_CTRL_SET_IHS

```
#define LCDC_REG_POLARITY_CTRL_SET_IHS(
              val ) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 1)
```

6.46.1.92 LCDC_REG_POLARITY_CTRL_SET_IPWR

```
#define LCDC_REG_POLARITY_CTRL_SET_IPWR(
              val ) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 4)
```

6.46.1.93 LCDC_REG_POLARITY_CTRL_SET_IVS

```
#define LCDC_REG_POLARITY_CTRL_SET_IVS(
              val ) SET_MASKED_BIT(LCDC_REG_POLARITY_CTRL, val, 0)
```

6.46.1.94 LCDC_REG_SCALER_HOR_RES_IN

```
#define LCDC_REG_SCALER_HOR_RES_IN (LCD_FTLCDC210_PA_BASE + 0x1100)
```

6.46.1.95 LCDC_REG_SCALER_HOR_RES_IN_GET_hor_no_in

```
#define LCDC_REG_SCALER_HOR_RES_IN_GET_hor_no_in( ) GET_BITS(LCDC_REG_SCALER_HOR_RES_IN, 0,
11)
```

6.46.1.96 LCDC_REG_SCALER_HOR_RES_IN_SET_hor_no_in

```
#define LCDC_REG_SCALER_HOR_RES_IN_SET_hor_no_in(
              val ) SET_MASKED_BITS(LCDC_REG_SCALER_HOR_RES_IN, val, 0, 11)
```

6.46.1.97 LCDC_REG_SCALER_HOR_RES_OUT

```
#define LCDC_REG_SCALER_HOR_RES_OUT (LCD_FTLCDC210_PA_BASE + 0x1108)
```

Kneron Confidential

6.46.1.98 LCDC_REG_SCALER_HOR_RES_OUT_GET_hor_no_out

```
#define LCDC_REG_SCALER_HOR_RES_OUT_GET_hor_no_out( ) GET_BITS(LCDC_REG_SCALER_HOR_RES_OUT, 0,
13)
```

6.46.1.99 LCDC_REG_SCALER_HOR_RES_OUT_SET_hor_no_out

```
#define LCDC_REG_SCALER_HOR_RES_OUT_SET_hor_no_out(
              val ) SET_MASKED_BITS(LCDC_REG_SCALER_HOR_RES_OUT, val, 0, 13)
```

6.46.1.100 LCDC_REG_SCALER_MISC

```
#define LCDC_REG_SCALER_MISC (LCD_FTLCDC210_PA_BASE + 0x1110)
```

6.46.1.101 LCDC_REG_SCALER_MISC_GET_Values

```
#define LCDC_REG_SCALER_MISC_GET_Values( ) GET_BITS(LCDC_REG_SCALER_MISC, 0, 8)
```

6.46.1.102 LCDC_REG_SCALER_MISC_SET_bypass_mode

```
#define LCDC_REG_SCALER_MISC_SET_bypass_mode(
              val ) SET_MASKED_BIT(LCDC_REG_SCALER_MISC, val, 0)
```

6.46.1.103 LCDC_REG_SCALER_MISC_SET_fir_sel

```
#define LCDC_REG_SCALER_MISC_SET_fir_sel(
              val ) SET_MASKED_BITS(LCDC_REG_SCALER_MISC, val, 6, 8)
```

6.46.1.104 LCDC_REG_SCALER_MISC_SET_hor_inter_mode

```
#define LCDC_REG_SCALER_MISC_SET_hor_inter_mode(
              val ) SET_MASKED_BITS(LCDC_REG_SCALER_MISC, val, 3, 4)
```

Kneron Confidential

6.46.1.105 LCDC_REG_SCALER_MISC_SET_ver_inter_mode

```
#define LCDC_REG_SCALER_MISC_SET_ver_inter_mode(
             val ) SET_MASKED_BITS(LCDC_REG_SCALER_MISC, val, 1, 2)
```

6.46.1.106 LCDC_REG_SCALER_RES

```
#define LCDC_REG_SCALER_RES (LCD_FTLCDC210_PA_BASE + 0x112C)
```

6.46.1.107 LCDC_REG_SCALER_RES_GET_Values

```
#define LCDC_REG_SCALER_RES_GET_Values( ) GET_BITS(LCDC_REG_SCALER_RES, 0, 15)
```

6.46.1.108 LCDC_REG_SCALER_RES_SET_scal_hor_num

```
#define LCDC_REG_SCALER_RES_SET_scal_hor_num(
             val ) SET_MASKED_BITS(LCDC_REG_SCALER_RES, val, 8, 15)
```

6.46.1.109 LCDC_REG_SCALER_RES_SET_scal_ver_num

```
#define LCDC_REG_SCALER_RES_SET_scal_ver_num(
             val ) SET_MASKED_BITS(LCDC_REG_SCALER_RES, val, 0, 7)
```

6.46.1.110 LCDC_REG_SCALER_VER_RES_IN

```
#define LCDC_REG_SCALER_VER_RES_IN (LCD_FTLCDC210_PA_BASE + 0x1104)
```

6.46.1.111 LCDC_REG_SCALER_VER_RES_IN_GET_ver_no_in

```
#define LCDC_REG_SCALER_VER_RES_IN_GET_ver_no_in( ) GET_BITS(LCDC_REG_SCALER_VER_RES_IN, 0,
11)
```

Kneron Confidential

6.46.1.112 LCDC_REG_SCALER_VER_RES_IN_SET_ver_no_in

```
#define LCDC_REG_SCALER_VER_RES_IN_SET_ver_no_in(
              val ) SET_MASKED_BITS(LCDC_REG_SCALER_VER_RES_IN, val, 0, 11)
```

6.46.1.113 LCDC_REG_SCALER_VER_RES_OUT

```
#define LCDC_REG_SCALER_VER_RES_OUT (LCD_FTLCDC210_PA_BASE + 0x110C)
```

6.46.1.114 LCDC_REG_SCALER_VER_RES_OUT_GET_ver_no_out

```
#define LCDC_REG_SCALER_VER_RES_OUT_GET_ver_no_out( ) GET_BITS(LCDC_REG_SCALER_VER_RES_OUT, 0,
13)
```

6.46.1.115 LCDC_REG_SCALER_VER_RES_OUT_SET_ver_no_out

```
#define LCDC_REG_SCALER_VER_RES_OUT_SET_ver_no_out(
              val ) SET_MASKED_BITS(LCDC_REG_SCALER_VER_RES_OUT, val, 0, 13)
```

6.46.1.116 LCDC_REG_SERIAL_PANEL_PIXEL

```
#define LCDC_REG_SERIAL_PANEL_PIXEL (LCD_FTLCDC210_PA_BASE + 0x0200)
```

6.46.1.117 LCDC_REG_SERIAL_PANEL_PIXEL_GET_Values

```
#define LCDC_REG_SERIAL_PANEL_PIXEL_GET_Values( ) GET_BITS(LCDC_REG_SERIAL_PANEL_PIXEL, 0, 5)
```

6.46.1.118 LCDC_REG_SERIAL_PANEL_PIXEL_SET_AUO052

```
#define LCDC_REG_SERIAL_PANEL_PIXEL_SET_AUO052(
              val ) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL, val, 5)
```

Kneron Confidential

6.46.1.119 LCDC_REG_SERIAL_PANEL_PIXEL_SET_ColorSeq

```
#define LCDC_REG_SERIAL_PANEL_PIXEL_SET_ColorSeq(
              val ) SET_MASKED_BITS(LCDC_REG_SERIAL_PANEL_PIXEL, val, 2, 3)
```

6.46.1.120 LCDC_REG_SERIAL_PANEL_PIXEL_SET_DeltaType

```
#define LCDC_REG_SERIAL_PANEL_PIXEL_SET_DeltaType(
              val ) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL, val, 1)
```

6.46.1.121 LCDC_REG_SERIAL_PANEL_PIXEL_SET_LSR

```
#define LCDC_REG_SERIAL_PANEL_PIXEL_SET_LSR(
              val ) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL, val, 4)
```

6.46.1.122 LCDC_REG_SERIAL_PANEL_PIXEL_SET_SerialMode

```
#define LCDC_REG_SERIAL_PANEL_PIXEL_SET_SerialMode(
              val ) SET_MASKED_BIT(LCDC_REG_SERIAL_PANEL_PIXEL, val, 0)
```

6.46.1.123 LCDC_REG_VERTICAL_BACK_PORCH

```
#define LCDC_REG_VERTICAL_BACK_PORCH (LCD_FTLCDC210_PA_BASE + 0x0108)
```

6.46.1.124 LCDC_REG_VERTICAL_BACK_PORCH_GET_VBP

```
#define LCDC_REG_VERTICAL_BACK_PORCH_GET_VBP( ) GET_BITS(LCDC_REG_VERTICAL_BACK_PORCH, 0, 7)
```

6.46.1.125 LCDC_REG_VERTICAL_BACK_PORCH_SET_VBP

```
#define LCDC_REG_VERTICAL_BACK_PORCH_SET_VBP(
              val ) SET_MASKED_BITS(LCDC_REG_VERTICAL_BACK_PORCH, val, 0, 7)
```

Kneron Confidential

6.46.1.126 LCDC_REG_VERTICAL_TIMING_CTRL

```
#define LCDC_REG_VERTICAL_TIMING_CTRL (LCD_FTLCDC210_PA_BASE + 0x0104)
```

6.46.1.127 LCDC_REG_VERTICAL_TIMING_CTRL_GET_Values

```
#define LCDC_REG_VERTICAL_TIMING_CTRL_GET_Values( ) GET_BITS(LCDC_REG_VERTICAL_TIMING_CTRL, 0,
31)
```

6.46.1.128 LCDC_REG_VERTICAL_TIMING_CTRL_SET_LF

```
#define LCDC_REG_VERTICAL_TIMING_CTRL_SET_LF(
              val ) SET_MASKED_BITS(LCDC_REG_VERTICAL_TIMING_CTRL, val, 0, 11)
```

6.46.1.129 LCDC_REG_VERTICAL_TIMING_CTRL_SET_VFP

```
#define LCDC_REG_VERTICAL_TIMING_CTRL_SET_VFP(
              val ) SET_MASKED_BITS(LCDC_REG_VERTICAL_TIMING_CTRL, val, 24, 31)
```

6.46.1.130 LCDC_REG_VERTICAL_TIMING_CTRL_SET_VW

```
#define LCDC_REG_VERTICAL_TIMING_CTRL_SET_VW(
              val ) SET_MASKED_BITS(LCDC_REG_VERTICAL_TIMING_CTRL, val, 16, 21)
```

6.46.1.131 PWM0_FREQ_CNT

```
#define PWM0_FREQ_CNT (2000000)
```

6.46.1.132 SWAP

```
#define SWAP(
              a,
              b )
```

Value:

```
do \
                                    { a^=b; \
                                      b^=a; \
                                      a^=b; }while(0)
```

Kneron Confidential

6.46.2 Function Documentation

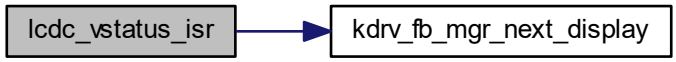
6.46.2.1 lcd_update_snapshot_data()

```
void lcd_update_snapshot_data (
            uint32_t addr )
```

6.46.2.2 lcdc_vstatus_isr()

```
void lcdc_vstatus_isr (
            void  )
```

Here is the call graph for this function:



6.46.3 Variable Documentation

6.46.3.1 kdrv_lcdc_frame_buffer

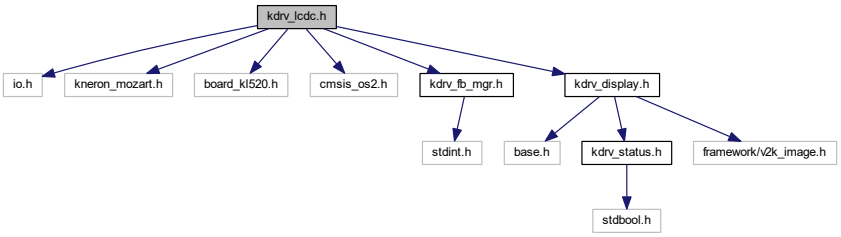
```
kdrv_lcdc_fb_t kdrv_lcdc_frame_buffer
```

Kneron Confidential

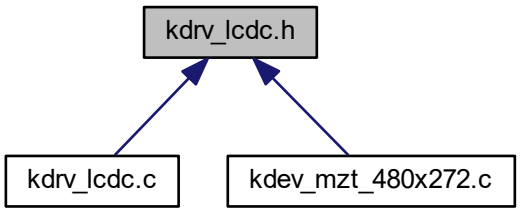
6.47 kdrv_lcdc.h File Reference

```
#include "io.h"
#include "kneron_mozart.h"
#include "board_kl520.h"
#include <cmsis_os2.h>
#include "kdrv_fb_mgr.h"
#include "kdrv_display.h"
```

Include dependency graph for kdrv_lcdc.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define FLAGS_KDP520_LCDC_START_DRAW_RECT_EVT BIT0
- #define FLAGS_KDP520_LCDC_STOP_DRAW_RECT_EVT BIT1
- #define MAX_FRAME_NUM (1)
- #define LCDC_HINT_BOUNDINGBOX_MARGIN_LEN (30)

Enumerations

- enum kdrv_lcdc_screen_ctrl_t { KDRV_LCDC_SCREEN_OFF = 0, KDRV_LCDC_SCREEN_ON }
- enum kdrv_lcdc_img_pixfmt_t { KDRV_LCDC_IMG_PIXFMT_1BPP = 0, KDRV_LCDC_IMG_PIXFMT_2BPP, KDRV_LCDC_IMG_PIXFMT_4BPP, KDRV_LCDC_IMG_PIXFMT_8BPP, KDRV_LCDC_IMG_PIXFMT_16BPP, KDRV_LCDC_IMG_PIXFMT_24BPP, KDRV_LCDC_IMG_PIXFMT_ARGB8888, KDRV_LCDC_IMG_PIXFMT_ARGB1555 }

Kneron Confidential

- enum kdrv_lcdc_panel_type_t { KDRV_LCDC_6BIT_PER_CHANNEL = 0, KDRV_LCDC_8BIT_PER_CHANNEL }
- enum kdrv_lcdc_output_fmt_t { KDRV_LCDC_OUTPUT_FMT_RGB = 0, KDRV_LCDC_OUTPUT_FMT_BGR }
- enum kdrv_lcdc_serial_pix_sr_t { KDRV_LCDC_SERIAL_PIX_RSR = 0, KDRV_LCDC_SERIAL_PIX_LSR }
- enum kdrv_lcdc_serial_pix_colorseq_t { KDRV_LCDC_SERIAL_PIX_COLORSEQ_RGB = 0, KDRV_LCDC_SERIAL_PIX_COL KDRV_LCDC_SERIAL_PIX_COLORSEQ_GBR }
- enum kdrv_lcdc_serial_pix_delta_type_t { KDRV_LCDC_SERIAL_PIX_DELTA_TYPE_SAME_SEQ = 0, KDRV_LCDC_SERIAL_PIX_DELTA_TYPE_DIFF_SEQ }
- enum kdrv_lcdc_serial_pix_output_mode_t { KDRV_LCDC_SERIAL_PIX_RGB_PARALLEL_OUTPUT = 0, KDRV_LCDC_SERIAL_PIX_RGB_SERIAL_OUTPUT }
- enum kdrv_lcdc_fb_data_endianness_t { KDRV_LCDC_FB_DATA_ENDIAN_LBLP = 0, KDRV_LCDC_FB_DATA_ENDIAN_BE KDRV_LCDC_FB_DATA_ENDIAN_LBBP }
- enum kdrv_lcdc_pat_gen_t { KDRV_LCDC_PAT_GEN_DISABLE = 0, KDRV_LCDC_PAT_GEN_ENABLE }
- enum kdrv_lcdc_auo052_mode_t { KDRV_LCDC_AUO052_OFF = 0, KDRV_LCDC_AUO052_ON }

Functions

- kdrv_status_t kdrv_display_screen_control (kdrv_lcdc_screen_ctrl_t ctrl)

  Control display screen ON/OFF.
- kdrv_status_t kdrv_lcdc_set_panel_type (kdrv_lcdc_panel_type_t type)

  Set TFT panel color depth selection of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_bgrsw (kdrv_lcdc_output_fmt_t format)

  Set output format selection of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_sr (kdrv_lcdc_serial_pix_sr_t rotate)

  Set odd line shift rotate of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_colorseq (kdrv_lcdc_serial_pix_colorseq_t color)

  Set color sequence of odd line of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_delta_type (kdrv_lcdc_serial_pix_delta_type_t type)

  Set delta type arrangement of color filter of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_pixel_serial_mode (kdrv_lcdc_serial_pix_output_mode_t mode)

  Set RGB serial output mode of LCD serial panel pixel parameter.
- kdrv_status_t kdrv_lcdc_set_endian (kdrv_lcdc_fb_data_endianness_t endian_type)

  Set data endian.
- kdrv_status_t kdrv_lcdc_set_auo052_mode (kdrv_lcdc_auo052_mode_t mode)

  Set data endian.
- kdrv_status_t kdrv_lcdc_down_scale (uint16_t hor_no_in, uint16_t hor_no_out, uint16_t ver_no_in, uint16_t ver_no_out)

  Set image down scale.
- kdrv_status_t kdrv_lcdc_set_framerate (int framerate, int width, int height)

  Set frame rate of lcdc vsync.
- kdrv_status_t kdrv_lcdc_set_image_color_params (uint32_t color0, uint32_t color1, uint32_t color2, uint32↩ _t color3)

  Set LCD color management parameter.
- kdrv_status_t kdrv_lcdc_set_bus_bandwidth_ctrl (uint32_t ctrl)

  Set frame rate of lcdc vsync.
- kdrv_status_t kdrv_lcdc_set_frame_buffer (uint32_t img_scal_down)

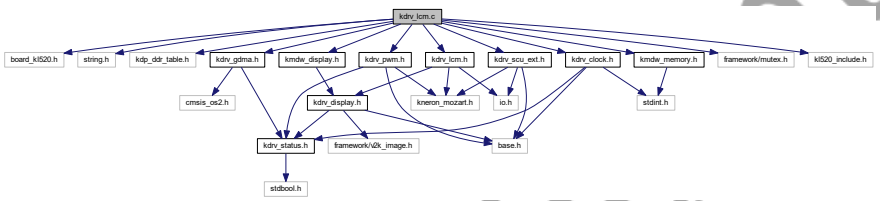
  Set frame buffer parameter.

Kneron Confidential

6.48 kdrv_lcm.c File Reference

```
#include "board_kl520.h"
#include <string.h>
#include "kdp_ddr_table.h"
#include "kdrv_gdma.h"
#include "kdrv_scu_ext.h"
#include "kdrv_lcm.h"
#include "kdrv_pwm.h"
#include "kdrv_clock.h"
#include "kmdw_display.h"
#include <framework/mutex.h>
#include "kl520_include.h"
#include "kmdw_memory.h"
```

Include dependency graph for kdrv_lcm.c:



Data Structures

- struct kdrv_lcm_context_t
- struct kdrv_lcm_fb_t

Macros

- #define lcm_msg(fmt, ...)
- #define LCM_BASE SLCD_FTLCDC210_PA_BASE
- #define LCM_REG_TIMING (LCM_BASE + 0x200)
- #define LCM_REG_RDY (LCM_BASE + 0x204)
- #define LCM_REG_RS (LCM_BASE + 0x208)
- #define LCM_REG_DATA (LCM_BASE + 0x20C)
- #define LCM_REG_CMD (LCM_BASE + 0x210)
- #define LCM_REG_OP_MODE (LCM_BASE + 0x214)
- #define LCM_REG_ENABLE (LCM_BASE + 0x228)
- #define LCM_REG_TIMING_P(base) (base + 0x200)
- #define LCM_REG_RDY_P(base) (base + 0x204)
- #define LCM_REG_RS_P(base) (base + 0x208)
- #define LCM_REG_DATA_P(base) (base + 0x20C)
- #define LCM_REG_CMD_P(base) (base + 0x210)
- #define LCM_REG_OP_MODE_P(base) (base + 0x214)
- #define LCM_REG_ENABLE_P(base) (base + 0x228)
- #define LCM_REG_TPWH_WIDTH_4_READ_CYCLE() GET_BITS(LCM_REG_TIMING, 16, 19)
- #define LCM_REG_TAS_WIDTH() GET_BITS(LCM_REG_TIMING, 12, 15)
- #define LCM_REG_TAH_WIDTH() GET_BITS(LCM_REG_TIMING, 8, 11)
- #define LCM_REG_TPWL_WIDTH() GET_BITS(LCM_REG_TIMING, 4, 7)

Kneron Confidential

- #define LCM_REG_TPWH_WIDTH_4_WRITE_CYCLE() GET_BITS(LCM_REG_TIMING, 0, 3)
- #define LCM_REG_LCM_RDY() GET_BIT(LCM_REG_RDY, 0)
- #define LCM_REG_LCM_DATA() GET_BITS(LCM_REG_DATA, 0, 31)
- #define LCM_REG_LCM_CMD() GET_BITS(LCM_REG_CMD, 0, 31)
- #define LCM_REG_LCM_OP_MODE_16_BIT_MODE_SELECTION() GET_BITS(LCM_REG_OP_MODE, 6, 7)
- #define LCM_REG_LCM_OP_MODE_BIT_SELECTION_BUS_INTERFACE() GET_BITS(LCM_REG_OP_MODE, 4, 5)
- #define LCM_REG_LCM_OP_MODE_BIT_SELECTION_PANEL_INTERFACE() GET_BITS(LCM_REG_OP_MODE, 2, 3)
- #define LCM_REG_LCM_OP_MODE_MCU_SELECTION() GET_BIT(LCM_REG_OP_MODE, 0)
- #define LCM_REG_LCM_ENABLE_BACKLIGHT_CTRL() GET_BIT(LCM_REG_ENABLE, 12)
- #define LCM_REG_LCM_ENABLE_SIGNAL_INVERSE() GET_BIT(LCM_REG_ENABLE, 11)
- #define LCM_REG_LCM_ENABLE_RESET_ASSERT() GET_BIT(LCM_REG_ENABLE, 8)
- #define LCM_REG_LCM_ENABLE() GET_BIT(LCM_REG_ENABLE, 0)
- #define LCM_REG_TIMING_SET_Tpwh_r(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 16, 19)
- #define LCM_REG_TIMING_SET_Tas(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 12, 15)
- #define LCM_REG_TIMING_SET_Tah(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 8, 11)
- #define LCM_REG_TIMING_SET_Tpwl(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 4, 7)
- #define LCM_REG_TIMING_SET_Tpwh_w(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 0, 3)
- #define LCM_REG_RDY_READY_FOR_ACCESS BIT0
- #define LCM_REG_RDY_GET_Rdy_C_D GET_BIT(LCM_REG_RDY, 0)
- #define LCM_REG_RS_DMYRD_RS BIT0
- #define LCM_REG_RS_GET_DMYRD_RS() GET_BIT(LCM_REG_RS, 0)
- #define LCM_REG_RS_SET_DMYRD_RS(val) SET_MASKED_BIT(LCM_REG_RS, val, 0)
- #define LCM_REG_OP_MODE_SET_16bit_mode(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 6, 7)
- #define LCM_REG_OP_MODE_SET_Bus_IF(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 4, 5)
- #define LCM_REG_OP_MODE_SET_Panel_IF(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 2, 3)
- #define LCM_REG_OP_MODE_SET_C68(val) SET_MASKED_BITS(LCM_REG_TIMING, val, 0, 1)
- #define LCM_REG_ENABLE_GET_BLCTRL_SET() GET_BIT(LCM_REG_ENABLE, 12)
- #define LCM_REG_ENABLE_GET_RST_INV() GET_BIT(LCM_REG_ENABLE, 11)
- #define LCM_REG_ENABLE_GET_RST_CLR() GET_BIT(LCM_REG_ENABLE, 9)
- #define LCM_REG_ENABLE_GET_RST_SET() GET_BIT(LCM_REG_ENABLE, 8)
- #define LCM_REG_ENABLE_GET_LCM_En() GET_BIT(LCM_REG_ENABLE, 0)
- #define LCM_REG_ENABLE_SET_BLCTRL_CLR(val) SET_MASKED_BIT(LCM_REG_ENABLE, val, 13)
- #define LCM_REG_ENABLE_SET_BLCTRL_SET(val) SET_MASKED_BIT(LCM_REG_ENABLE, val, 12)
- #define LCM_REG_ENABLE_SET_RST_INV(val) SET_MASKED_BIT(LCM_REG_ENABLE, val, 11)
- #define LCM_REG_ENABLE_SET_RST_CLR() SET_BIT(LCM_REG_ENABLE, 9)
- #define LCM_REG_ENABLE_SET_RST_SET(val)
- #define LCM_REG_ENABLE_SET_LCM_En(val) SET_MASKED_BIT(LCM_REG_ENABLE, val, 0)
- #define LCM_PROFILE_ENABLE
- #define LCM_PROFILE_START() _prof_start = osKernelGetTickCount();
- #define LCM_PROFILE_STOP()
- #define LCM_OPS_WAIT_TIMEOUT_CNT (1000)
- #define PWM6_FREQ_CNT (2000000)

Enumerations

- enum lcm_signal_cycle {
  LCM_SIGNAL_WIDTH_LC_PCLK_X1 = 0x00, LCM_SIGNAL_WIDTH_LC_PCLK_X2 = 0x01, LCM_SIGNAL_WIDTH_LC_PCL
  = 0x02, LCM_SIGNAL_WIDTH_LC_PCLK_X4 = 0x03,
  LCM_SIGNAL_WIDTH_LC_PCLK_X5 = 0x04, LCM_SIGNAL_WIDTH_LC_PCLK_X6 = 0x05, LCM_SIGNAL_WIDTH_LC_PCL
  = 0x06, LCM_SIGNAL_WIDTH_LC_PCLK_X8 = 0x07,
  LCM_SIGNAL_WIDTH_LC_PCLK_X9 = 0x08, LCM_SIGNAL_WIDTH_LC_PCLK_X10 = 0x09, LCM_SIGNAL_WIDTH_LC_PC
  = 0x0A, LCM_SIGNAL_WIDTH_LC_PCLK_X12 = 0x0B,
  LCM_SIGNAL_WIDTH_LC_PCLK_X13 = 0x0C, LCM_SIGNAL_WIDTH_LC_PCLK_X14 = 0x0D, LCM_SIGNAL_WIDTH_LC_P
  = 0x0E, LCM_SIGNAL_WIDTH_LC_PCLK_X16 = 0x0F }

Kneron Confidential

- enum lcm_16bitmode_selection { LCM_16BITMODE_ONESHOT = 0x00, LCM_16BITMODE_16_AT_MSB = 0x01, LCM_16BITMODE_16_AT_LSB = 0x01 }
- enum lcm_data_bus_type { LCM_DATA_BUS_8_BITS = 0x00, LCM_DATA_BUS_9_BITS = 0x01, LCM_DATA_BUS_16_BITS = 0x02, LCM_DATA_BUS_18_BITS = 0x03 }
- enum lcm_panel_interface_type { LCM_PANEL_INTEFACE_MONO = 0x00, LCM_PANEL_INTEFACE_16_BITS = 0x00, LCM_PANEL_INTEFACE_18_BITS = 0x00 }
- enum lcm_mcu_interface_mode { LCM_MCU_INTERFACE_8080 = 0x00, LCM_MCU_INTERFACE_6800 = 0x01 }
- enum lcm_state {
  LCM_STATE_IDLE = 0, LCM_STATE_INITED, LCM_STATE_STOPPED, LCM_STATE_PROBED, LCM_STATE_STARTED }

Functions

- kdrv_display_t * kdrv_display_initialize (void)

  Initialize display driver.
- kdrv_status_t kdrv_display_set_pen (kdrv_display_t *display_drv, uint16_t color, unsigned int width)
- kdrv_status_t kdrv_display_set_params (kdrv_display_t *display_drv, struct video_input_params *params)

  Set display parameters.
- kdrv_status_t kdrv_display_get_params (kdrv_display_t *display_drv, struct video_input_params *params)

  Get display parameters.
- kdrv_status_t kdrv_display_set_camera (kdrv_display_t *display_drv, uint8_t cam_idx)

  Set camera source which will be preview on display.
- kdrv_status_t kdrv_display_buffer_initialize (struct video_input_params *params)

  Initialize display frame buffer.
- uint32_t kdrv_display_get_buffer (void)
- kdrv_status_t kdrv_display_start (kdrv_display_t *display_drv)

  Start display image preview.
- kdrv_status_t kdrv_display_stop (kdrv_display_t *display_drv)

  Stop display image preview.
- kdrv_status_t kdrv_display_draw_line (kdrv_display_t *display_drv, uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- kdrv_status_t kdrv_display_draw_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_draw_static_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_↩ y, uint32_t width, uint32_t height)

  Draw rectangle without filling color on display.
- kdrv_status_t kdrv_display_test_pattern_gen (kdrv_display_t *display_drv, bool pat_gen)

  Set display backlight.
- kdrv_status_t kdrv_display_fill_rect (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_y, uint32_↩ t width, uint32_t height)

  Draw rectangle with filling color on display.
- kdrv_status_t kdrv_display_draw_bitmap (kdrv_display_t *display_drv, uint32_t org_x, uint32_t org_↩ y, uint32_t width, uint32_t height, void *pBuf)

  Draw bitmap on display.
- kdrv_status_t kdrv_display_set_backlight (kdrv_display_t *display_drv, int light)

  Set display backlight.
- kdrv_status_t kdrv_display_set_frame_margin_len (uint16_t margin_len)
- uint16_t kdrv_display_get_frame_margin_len (void)
- kdrv_status_t kdrv_display_set_review_snapshot_en (bool enable)

Kneron Confidential

- bool kdrv_display_get_review_snapshot_en (void)
- kdrv_status_t kdrv_display_set_snapshot_preview_en (bool enable)
- bool kdrv_display_get_snapshot_preview_en (void)
- kdrv_status_t kdrv_lcm_write_cmd (uint32_t base, unsigned char data)
- kdrv_status_t kdrv_lcm_write_data (uint32_t base, unsigned char data)
- unsigned int kdrv_lcm_read_data (uint32_t base)
- kdrv_status_t kdrv_lcm_pressingnir (kdrv_display_t *display_drv, u32 addr)
- kdrv_status_t kdrv_lcm_pressing (kdrv_display_t *display_drv, u32 addr)
- kdrv_status_t kdrv_display_update_draw_fb (kdrv_display_t *display_drv, uint32_t addr, uint8_t cam_idx)

  Update frame buffer which be used to draw something on display.
- uint8_t kdrv_lcm_get_backlight (void)
- uint32_t kdrv_lcm_get_db_frame (void)

Variables

- kdrv_lcm_context_t kdrv_lcm_ctx
- uint8_t _glight = 50
- bool m_show_txt = true
- struct mutex display_reg_lock
- enum lcm_state m_lcm_state
- u32 sys_fdr_state
- kdrv_lcm_fb_t kdrv_lcm_frame_buffer
- kdrv_display_t kdrv_display
- osMutexId_t mutex_snapshot

6.48.1 Macro Definition Documentation

6.48.1.1 LCM_BASE

```
#define LCM_BASE SLCD_FTLCDC210_PA_BASE
```

6.48.1.2 lcm_msg

```
#define lcm_msg(
                fmt,
                ...  )
```

6.48.1.3 LCM_OPS_WAIT_TIMEOUT_CNT

```
#define LCM_OPS_WAIT_TIMEOUT_CNT (1000)
```

Kneron Confidential

6.48.1.4 LCM_PROFILE_ENABLE

```
#define LCM_PROFILE_ENABLE
```

6.48.1.5 LCM_PROFILE_START

```
#define LCM_PROFILE_START( ) _prof_start = osKernelGetTickCount();
```

6.48.1.6 LCM_PROFILE_STOP

```
#define LCM_PROFILE_STOP( )
```

Value:

```
{ \
                _prof_end = osKernelGetTickCount(); \
                _prof_offset = _prof_end - _prof_start; \
                _prof_total += _prof_offset; \
                ++_prof_frame_cnt; \
                /*DSG("[%s] offset=%u average=%u", __func__, _prof_offset, _prof_total / _prof_frame_cnt);*/
        \
                }
```

6.48.1.7 LCM_REG_CMD

```
#define LCM_REG_CMD (LCM_BASE + 0x210)
```

6.48.1.8 LCM_REG_CMD_P

```
#define LCM_REG_CMD_P(
                base ) (base + 0x210)
```

6.48.1.9 LCM_REG_DATA

```
#define LCM_REG_DATA (LCM_BASE + 0x20C)
```

Kneron Confidential

6.48.1.10 LCM_REG_DATA_P

```
#define LCM_REG_DATA_P(
             base ) (base + 0x20C)
```

6.48.1.11 LCM_REG_ENABLE

```
#define LCM_REG_ENABLE (LCM_BASE + 0x228)
```

6.48.1.12 LCM_REG_ENABLE_GET_BLCTRL_SET

```
#define LCM_REG_ENABLE_GET_BLCTRL_SET( ) GET_BIT(LCM_REG_ENABLE, 12)
```

6.48.1.13 LCM_REG_ENABLE_GET_LCM_En

```
#define LCM_REG_ENABLE_GET_LCM_En( ) GET_BIT(LCM_REG_ENABLE, 0)
```

6.48.1.14 LCM_REG_ENABLE_GET_RST_CLR

```
#define LCM_REG_ENABLE_GET_RST_CLR( ) GET_BIT(LCM_REG_ENABLE, 9)
```

6.48.1.15 LCM_REG_ENABLE_GET_RST_INV

```
#define LCM_REG_ENABLE_GET_RST_INV( ) GET_BIT(LCM_REG_ENABLE, 11)
```

6.48.1.16 LCM_REG_ENABLE_GET_RST_SET

```
#define LCM_REG_ENABLE_GET_RST_SET( ) GET_BIT(LCM_REG_ENABLE, 8)
```

Kneron Confidential

6.48.1.17 LCM_REG_ENABLE_P

```
#define LCM_REG_ENABLE_P(
              base ) (base + 0x228)
```

6.48.1.18 LCM_REG_ENABLE_SET_BLCTRL_CLR

```
#define LCM_REG_ENABLE_SET_BLCTRL_CLR(
              val ) SET_MASKED_BIT(LCM_REG_ENABLE, val, 13)
```

6.48.1.19 LCM_REG_ENABLE_SET_BLCTRL_SET

```
#define LCM_REG_ENABLE_SET_BLCTRL_SET(
              val ) SET_MASKED_BIT(LCM_REG_ENABLE, val, 12)
```

6.48.1.20 LCM_REG_ENABLE_SET_LCM_En

```
#define LCM_REG_ENABLE_SET_LCM_En(
              val ) SET_MASKED_BIT(LCM_REG_ENABLE, val, 0)
```

6.48.1.21 LCM_REG_ENABLE_SET_RST_CLR

```
#define LCM_REG_ENABLE_SET_RST_CLR( ) SET_BIT(LCM_REG_ENABLE, 9)
```

6.48.1.22 LCM_REG_ENABLE_SET_RST_INV

```
#define LCM_REG_ENABLE_SET_RST_INV(
              val ) SET_MASKED_BIT(LCM_REG_ENABLE, val, 11)
```

6.48.1.23 LCM_REG_ENABLE_SET_RST_SET

```
#define LCM_REG_ENABLE_SET_RST_SET(
              val )
```

Kneron Confidential

6.48.1.24 LCM_REG_LCM_CMD

```
#define LCM_REG_LCM_CMD( ) GET_BITS(LCM_REG_CMD, 0, 31)
```

6.48.1.25 LCM_REG_LCM_DATA

```
#define LCM_REG_LCM_DATA( ) GET_BITS(LCM_REG_DATA, 0, 31)
```

6.48.1.26 LCM_REG_LCM_ENABLE

```
#define LCM_REG_LCM_ENABLE( ) GET_BIT(LCM_REG_ENABLE, 0)
```

6.48.1.27 LCM_REG_LCM_ENABLE_BACKLIGHT_CTRL

```
#define LCM_REG_LCM_ENABLE_BACKLIGHT_CTRL( ) GET_BIT(LCM_REG_ENABLE, 12)
```

6.48.1.28 LCM_REG_LCM_ENABLE_RESET_ASSERT

```
#define LCM_REG_LCM_ENABLE_RESET_ASSERT( ) GET_BIT(LCM_REG_ENABLE, 8)
```

6.48.1.29 LCM_REG_LCM_ENABLE_SIGNAL_INVERSE

```
#define LCM_REG_LCM_ENABLE_SIGNAL_INVERSE( ) GET_BIT(LCM_REG_ENABLE, 11)
```

6.48.1.30 LCM_REG_LCM_OP_MODE_16_BIT_MODE_SELECTION

```
#define LCM_REG_LCM_OP_MODE_16_BIT_MODE_SELECTION( ) GET_BITS(LCM_REG_OP_MODE, 6, 7)
```

6.48.1.31 LCM_REG_LCM_OP_MODE_BIT_SELECTION_BUS_INTERFACE

```
#define LCM_REG_LCM_OP_MODE_BIT_SELECTION_BUS_INTERFACE( ) GET_BITS(LCM_REG_OP_MODE, 4, 5)
```

Kneron Confidential

6.48.1.32 LCM_REG_LCM_OP_MODE_BIT_SELECTION_PANEL_INTERFACE

```
#define LCM_REG_LCM_OP_MODE_BIT_SELECTION_PANEL_INTERFACE( ) GET_BITS(LCM_REG_OP_MODE, 2, 3)
```

6.48.1.33 LCM_REG_LCM_OP_MODE_MCU_SELECTION

```
#define LCM_REG_LCM_OP_MODE_MCU_SELECTION( ) GET_BIT(LCM_REG_OP_MODE, 0)
```

6.48.1.34 LCM_REG_LCM_RDY

```
#define LCM_REG_LCM_RDY( ) GET_BIT(LCM_REG_RDY, 0)
```

6.48.1.35 LCM_REG_OP_MODE

```
#define LCM_REG_OP_MODE (LCM_BASE + 0x214)
```

6.48.1.36 LCM_REG_OP_MODE_P

```
#define LCM_REG_OP_MODE_P(
              base ) (base + 0x214)
```

6.48.1.37 LCM_REG_OP_MODE_SET_16bit_mode

```
#define LCM_REG_OP_MODE_SET_16bit_mode(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 6, 7)
```

6.48.1.38 LCM_REG_OP_MODE_SET_Bus_IF

```
#define LCM_REG_OP_MODE_SET_Bus_IF(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 4, 5)
```

Kneron Confidential

6.48.1.39 LCM_REG_OP_MODE_SET_C68

```
#define LCM_REG_OP_MODE_SET_C68(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 0, 1)
```

6.48.1.40 LCM_REG_OP_MODE_SET_Panel_IF

```
#define LCM_REG_OP_MODE_SET_Panel_IF(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 2, 3)
```

6.48.1.41 LCM_REG_RDY

```
#define LCM_REG_RDY (LCM_BASE + 0x204)
```

6.48.1.42 LCM_REG_RDY_GET_Rdy_C_D

```
#define LCM_REG_RDY_GET_Rdy_C_D GET_BIT(LCM_REG_RDY, 0)
```

6.48.1.43 LCM_REG_RDY_P

```
#define LCM_REG_RDY_P(
              base ) (base + 0x204)
```

6.48.1.44 LCM_REG_RDY_READY_FOR_ACCESS

```
#define LCM_REG_RDY_READY_FOR_ACCESS BIT0
```

6.48.1.45 LCM_REG_RS

```
#define LCM_REG_RS (LCM_BASE + 0x208)
```

Kneron Confidential

6.48.1.46 LCM_REG_RS_DMYRD_RS

```
#define LCM_REG_RS_DMYRD_RS BIT0
```

6.48.1.47 LCM_REG_RS_GET_DMYRD_RS

```
#define LCM_REG_RS_GET_DMYRD_RS( ) GET_BIT(LCM_REG_RS, 0)
```

6.48.1.48 LCM_REG_RS_P

```
#define LCM_REG_RS_P(
             base ) (base + 0x208)
```

6.48.1.49 LCM_REG_RS_SET_DMYRD_RS

```
#define LCM_REG_RS_SET_DMYRD_RS(
             val ) SET_MASKED_BIT(LCM_REG_RS, val, 0)
```

6.48.1.50 LCM_REG_TAH_WIDTH

```
#define LCM_REG_TAH_WIDTH( ) GET_BITS(LCM_REG_TIMING, 8, 11)
```

6.48.1.51 LCM_REG_TAS_WIDTH

```
#define LCM_REG_TAS_WIDTH( ) GET_BITS(LCM_REG_TIMING, 12, 15)
```

6.48.1.52 LCM_REG_TIMING

```
#define LCM_REG_TIMING (LCM_BASE + 0x200)
```

Kneron Confidential

6.48.1.53 LCM_REG_TIMING_P

```
#define LCM_REG_TIMING_P(
              base ) (base + 0x200)
```

6.48.1.54 LCM_REG_TIMING_SET_Tah

```
#define LCM_REG_TIMING_SET_Tah(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 8, 11)
```

6.48.1.55 LCM_REG_TIMING_SET_Tas

```
#define LCM_REG_TIMING_SET_Tas(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 12, 15)
```

6.48.1.56 LCM_REG_TIMING_SET_Tpwh_r

```
#define LCM_REG_TIMING_SET_Tpwh_r(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 16, 19)
```

6.48.1.57 LCM_REG_TIMING_SET_Tpwh_w

```
#define LCM_REG_TIMING_SET_Tpwh_w(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 0, 3)
```

6.48.1.58 LCM_REG_TIMING_SET_Tpwl

```
#define LCM_REG_TIMING_SET_Tpwl(
              val ) SET_MASKED_BITS(LCM_REG_TIMING, val, 4, 7)
```

6.48.1.59 LCM_REG_TPWH_WIDTH_4_READ_CYCLE

```
#define LCM_REG_TPWH_WIDTH_4_READ_CYCLE( ) GET_BITS(LCM_REG_TIMING, 16, 19)
```

Kneron Confidential

6.48.1.60 LCM_REG_TPWH_WIDTH_4_WRITE_CYCLE

```
#define LCM_REG_TPWH_WIDTH_4_WRITE_CYCLE( ) GET_BITS(LCM_REG_TIMING, 0, 3)
```

6.48.1.61 LCM_REG_TPWL_WIDTH

```
#define LCM_REG_TPWL_WIDTH( ) GET_BITS(LCM_REG_TIMING, 4, 7)
```

6.48.1.62 PWM6_FREQ_CNT

```
#define PWM6_FREQ_CNT (2000000)
```

6.48.2 Enumeration Type Documentation

6.48.2.1 lcm_16bitmode_selection

```
enum lcm_16bitmode_selection
```

Enumerator

LCM_16BITMODE_ONESHOT	
LCM_16BITMODE_16_AT_MSB	
LCM_16BITMODE_16_AT_LSB	

6.48.2.2 lcm_data_bus_type

```
enum lcm_data_bus_type
```

Enumerator

LCM_DATA_BUS_8_BITS	
LCM_DATA_BUS_9_BITS	
LCM_DATA_BUS_16_BITS	
LCM_DATA_BUS_18_BITS	

Kneron Confidential

6.48.2.3 lcm_mcu_interface_mode

enum lcm_mcu_interface_mode

Enumerator

LCM_MCU_INTERFACE_8080	
LCM_MCU_INTERFACE_6800	

6.48.2.4 lcm_panel_interface_type

enum lcm_panel_interface_type

Enumerator

LCM_PANEL_INTEFACE_MONO	
LCM_PANEL_INTEFACE_16_BITS	
LCM_PANEL_INTEFACE_18_BITS	

6.48.2.5 lcm_signal_cycle

enum lcm_signal_cycle

Enumerator

LCM_SIGNAL_WIDTH_LC_PCLK_X1	
LCM_SIGNAL_WIDTH_LC_PCLK_X2	
LCM_SIGNAL_WIDTH_LC_PCLK_X3	
LCM_SIGNAL_WIDTH_LC_PCLK_X4	
LCM_SIGNAL_WIDTH_LC_PCLK_X5	
LCM_SIGNAL_WIDTH_LC_PCLK_X6	
LCM_SIGNAL_WIDTH_LC_PCLK_X7	
LCM_SIGNAL_WIDTH_LC_PCLK_X8	
LCM_SIGNAL_WIDTH_LC_PCLK_X9	
LCM_SIGNAL_WIDTH_LC_PCLK_X10	
LCM_SIGNAL_WIDTH_LC_PCLK_X11	
LCM_SIGNAL_WIDTH_LC_PCLK_X12	
LCM_SIGNAL_WIDTH_LC_PCLK_X13	
LCM_SIGNAL_WIDTH_LC_PCLK_X14	
LCM_SIGNAL_WIDTH_LC_PCLK_X15	
LCM_SIGNAL_WIDTH_LC_PCLK_X16	

Kneron Confidential

6.48.2.6 lcm_state

```
enum lcm_state
```

Enumerator

LCM_STATE_IDLE	
LCM_STATE_INITED	
LCM_STATE_STOPPED	
LCM_STATE_PROBED	
LCM_STATE_STARTED	

6.48.3 Function Documentation

6.48.3.1 kdrv_display_set_pen()

```
kdrv_status_t kdrv_display_set_pen (
            kdrv_display_t * display_drv,
            uint16_t color,
            unsigned int width )
```

6.48.4 Variable Documentation

6.48.4.1 _glight

```
uint8_t _glight = 50
```

6.48.4.2 display_reg_lock

```
struct mutex display_reg_lock
```

6.48.4.3 kdrv_display

```
kdrv_display_t kdrv_display
```

Kneron Confidential

6.48.4.4 kdrv_lcm_ctx

```
kdrv_lcm_context_t kdrv_lcm_ctx
```

6.48.4.5 kdrv_lcm_frame_buffer

```
kdrv_lcm_fb_t kdrv_lcm_frame_buffer
```

6.48.4.6 m_lcm_state

```
enum lcm_state m_lcm_state
```

6.48.4.7 m_show_txt

```
bool m_show_txt = true
```

6.48.4.8 mutex_snapshot

```
osMutexId_t mutex_snapshot
```

6.48.4.9 sys_fdr_state

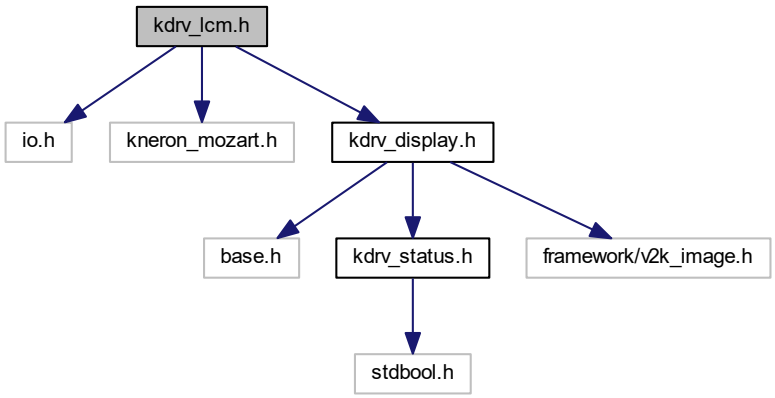
```
u32 sys_fdr_state
```

Kneron Confidential

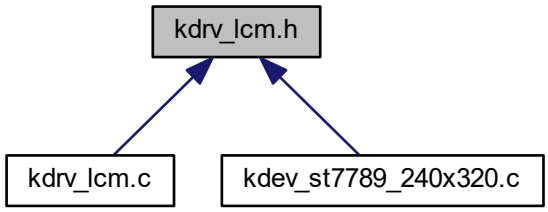
6.49 kdrv_lcm.h File Reference

```
#include "io.h"
#include "kneron_mozart.h"
#include "kdrv_display.h"
```

Include dependency graph for kdrv_lcm.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

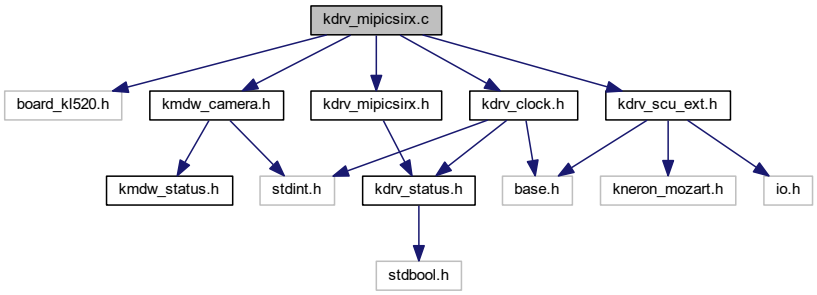
Functions

- u8 kdrv_lcm_get_backlight (void)
- kdrv_status_t kdrv_lcm_pressing (kdrv_display_t *display_drv, u32 addr)
- kdrv_status_t kdrv_lcm_pressingnir (kdrv_display_t *display_drv, u32 addr)
- kdrv_status_t kdrv_lcm_write_cmd (uint32_t base, unsigned char data)
- kdrv_status_t kdrv_lcm_write_data (uint32_t base, unsigned char data)
- unsigned int kdrv_lcm_read_data (uint32_t base)
- uint32_t kdrv_lcm_get_db_frame (void)

6.50 kdrv_mipicsirx.c File Reference

```
#include "board_kl520.h"
#include "kmdw_camera.h"
#include "kdrv_mipicsirx.h"
#include "kdrv_clock.h"
#include "kdrv_scu_ext.h"
```

Include dependency graph for kdrv_mipicsirx.c:



Data Structures

- struct kdrv_csirx_context

Macros

- #define ROUND_UP(x, y) ((((x) + (y - 1)) / y) * y)
- #define CSI2RX_REG_VIDR 0x00
- #define CSI2RX_REG_DIDR 0x01
- #define CSI2RX_REG_CR 0x04
- #define CSI2RX_REG_VSCR 0x05
- #define CSI2RX_REG_ECR 0x06
- #define CSI2RX_REG_TCNR 0x08
- #define CSI2RX_REG_HRTVR 0x0A
- #define CSI2RX_REG_FIUR 0x0B
- #define CSI2RX_REG_ITR 0x12
- #define CSI2RX_REG_VSTR0 0x14
- #define CSI2RX_REG_HSTR0 0x15
- #define CSI2RX_REG_VSTR1 0x16
- #define CSI2RX_REG_HSTR1 0x17
- #define CSI2RX_REG_VSTR2 0x18
- #define CSI2RX_REG_HSTR2 0x19
- #define CSI2RX_REG_VSTR3 0x1A
- #define CSI2RX_REG_HSTR3 0x1B
- #define CSI2RX_REG_MCR 0x1C
- #define CSI2RX_REG_VSTER 0x1E
- #define CSI2RX_REG_HPNR 0x20
- #define CSI2RX_REG_PECR 0x28
- #define CSI2RX_REG_DLMR 0x2A

Kneron Confidential

- #define CSI2RX_REG_CSIERR 0x30
- #define CSI2RX_REG_INTSTS 0x33
- #define CSI2RX_REG_ESR 0x34
- #define CSI2RX_REG_DPISR 0x38
- #define CSI2RX_REG_INTER 0x3C
- #define CSI2RX_REG_FFR 0x3D
- #define CSI2RX_REG_DPCMR 0x48
- #define CSI2RX_REG_FRR 0x4C
- #define CSI2RX_REG_PFTR 0x50
- #define CSI2RX_REG_PUDTR 0x58
- #define CSI2RX_REG_FRCR 0x80
- #define CSI2RX_REG_FNR 0x88
- #define CSI2RX_REG_BPGLR 0x90
- #define CSI2RX_REG_BPGHR 0x91
- #define MAX_VIDEO_MEM 8 << 20
- #define CSIRX_0_BASE CSIRX_FTCSIRX100_PA_BASE
- #define CSIRX_1_BASE CSIRX_FTCSIRX100_1_PA_BASE
- #define CSIRX_REG_VSCR_SET_VSTU0(addr, val) SET_MASKED_BIT(addr, val, 0)
- #define CSIRX_REG_PECR_GET_PEC(addr) GET_BITS(addr, 0, 7)
- #define CSIRX_REG_PECR_SET_PEC(addr, val) SET_MASKED_BITS(addr, val, 0, 7)
- #define CSIRX_REG_0_DLMR0 (CSIRX_0_BASE + 0x2A)
- #define CSIRX_REG_0_DLMR0_GET_L1() GET_BITS(CSIRX_REG_0_DLMR0, 4, 6)
- #define CSIRX_REG_0_DLMR0_GET_L0() GET_BITS(CSIRX_REG_0_DLMR0, 0, 2)
- #define CSIRX_REG_1_DLMR0 (CSIRX_1_BASE + 0x2A)
- #define CSIRX_REG_1_DLMR0_GET_L1() GET_BITS(CSIRX_REG_1_DLMR0, 4, 6)
- #define CSIRX_REG_1_DLMR0_GET_L0() GET_BITS(CSIRX_REG_1_DLMR0, 0, 2)
- #define CSIRX_REG_0_FEATURE0 (CSIRX_0_BASE + 0x40)
- #define CSIRX_REG_1_FEATURE0 (CSIRX_1_BASE + 0x40)
- #define CSIRX_REG_0_FEATURE0_GET_DFAD() GET_BITS(CSIRX_REG_0_FEATURE0, 5, 7)
- #define CSIRX_REG_1_FEATURE0_GET_DFAD() GET_BITS(CSIRX_REG_1_FEATURE0, 5, 7)
- #define CSIRX_REG_0_FEATURE0_GET_LN() GET_BITS(CSIRX_REG_0_FEATURE0, 2, 4)
- #define CSIRX_REG_1_FEATURE0_GET_LN() GET_BITS(CSIRX_REG_1_FEATURE0, 2, 4)
- #define CSIRX_REG_0_FEATURE0_GET_VCN() GET_BITS(CSIRX_REG_0_FEATURE0, 0, 1)
- #define CSIRX_REG_1_FEATURE0_GET_VCN() GET_BITS(CSIRX_REG_1_FEATURE0, 0, 1)
- #define CSIRX_REG_0_FEATURE6 (CSIRX_0_BASE + 0x46)
- #define CSIRX_REG_1_FEATURE6 (CSIRX_1_BASE + 0x46)
- #define CSIRX_REG_0_FEATURE6_GET_ASC() GET_BIT(CSIRX_REG_0_FEATURE6, 7)
- #define CSIRX_REG_1_FEATURE6_GET_ASC() GET_BIT(CSIRX_REG_1_FEATURE6, 7)
- #define CSIRX_REG_0_FEATURE6_GET_APB() GET_BIT(CSIRX_REG_0_FEATURE6, 6)
- #define CSIRX_REG_1_FEATURE6_GET_APB() GET_BIT(CSIRX_REG_1_FEATURE6, 6)
- #define CSIRX_REG_0_FEATURE6_GET_I2C() GET_BIT(CSIRX_REG_0_FEATURE6, 5)
- #define CSIRX_REG_1_FEATURE6_GET_I2C() GET_BIT(CSIRX_REG_1_FEATURE6, 5)
- #define CSIRX_REG_0_FEATURE6_GET_DPCM() GET_BIT(CSIRX_REG_0_FEATURE6, 4)
- #define CSIRX_REG_1_FEATURE6_GET_DPCM() GET_BIT(CSIRX_REG_1_FEATURE6, 4)
- #define CSIRX_REG_0_FEATURE6_GET_CD() GET_BIT(CSIRX_REG_0_FEATURE6, 3)
- #define CSIRX_REG_1_FEATURE6_GET_CD() GET_BIT(CSIRX_REG_1_FEATURE6, 3)
- #define CSIRX_REG_0_FEATURE6_GET_DP() GET_BIT(CSIRX_REG_0_FEATURE6, 2)
- #define CSIRX_REG_1_FEATURE6_GET_DP() GET_BIT(CSIRX_REG_1_FEATURE6, 2)
- #define CSIRX_REG_0_FEATURE6_GET_FRC() GET_BIT(CSIRX_REG_0_FEATURE6, 1)
- #define CSIRX_REG_1_FEATURE6_GET_FRC() GET_BIT(CSIRX_REG_1_FEATURE6, 1)
- #define CSIRX_REG_0_FEATURE6_GET_PG() GET_BIT(CSIRX_REG_0_FEATURE6, 0)
- #define CSIRX_REG_1_FEATURE6_GET_PG() GET_BIT(CSIRX_REG_1_FEATURE6, 0)
- #define CSIRX_REG_0_FEATURE7 (CSIRX_0_BASE + 0x47)
- #define CSIRX_REG_1_FEATURE7 (CSIRX_1_BASE + 0x47)
- #define CSIRX_REG_0_FEATURE7_GET_OP() GET_BIT(CSIRX_REG_0_FEATURE7, 4)

Kneron Confidential

- #define CSIRX_REG_1_FEATURE7_GET_OP() GET_BIT(CSIRX_REG_1_FEATURE7, 4)
- #define CSIRX_REG_0_FEATURE7_GET_MR() GET_BIT(CSIRX_REG_0_FEATURE7, 3)
- #define CSIRX_REG_1_FEATURE7_GET_MR() GET_BIT(CSIRX_REG_1_FEATURE7, 3)
- #define CSIRX_REG_0_FEATURE7_GET_CRC() GET_BIT(CSIRX_REG_0_FEATURE7, 2)
- #define CSIRX_REG_1_FEATURE7_GET_CRC() GET_BIT(CSIRX_REG_1_FEATURE7, 2)
- #define CSIRX_REG_0_FEATURE7_GET_SWAP() GET_BIT(CSIRX_REG_0_FEATURE7, 1)
- #define CSIRX_REG_1_FEATURE7_GET_SWAP() GET_BIT(CSIRX_REG_1_FEATURE7, 1)
- #define CSIRX_REG_0_FEATURE7_GET_QP() GET_BIT(CSIRX_REG_0_FEATURE7, 0)
- #define CSIRX_REG_1_FEATURE7_GET_QP() GET_BIT(CSIRX_REG_1_FEATURE7, 0)
- #define CSIRX_REG_0_BPGLR0 (CSIRX_0_BASE + 0x90)
- #define CSIRX_REG_1_BPGLR0 (CSIRX_1_BASE + 0x90)
- #define CSIRX_REG_0_BPGLR0_SET_VLN(val) SET_MASKED_BITS(CSIRX_REG_0_BPGLR0, val, 0, 7)
- #define CSIRX_REG_1_BPGLR0_SET_VLN(val) SET_MASKED_BITS(CSIRX_REG_1_BPGLR0, val, 0, 7)
- #define CSIRX_REG_0_BPGHR0 (CSIRX_0_BASE + 0x91)
- #define CSIRX_REG_1_BPGHR0 (CSIRX_1_BASE + 0x91)
- #define CSIRX_REG_0_BPGHR0_SET_PT(val) SET_MASKED_BITS(CSIRX_REG_0_BPGHR0, val, 6, 7)
- #define CSIRX_REG_1_BPGHR0_SET_PT(val) SET_MASKED_BITS(CSIRX_REG_1_BPGHR0, val, 6, 7)
- #define CSIRX_REG_0_BPGHR0_SET_PS(val) SET_MASKED_BIT(CSIRX_REG_0_BPGHR0, val, 5)
- #define CSIRX_REG_1_BPGHR0_SET_PS(val) SET_MASKED_BIT(CSIRX_REG_1_BPGHR0, val, 5)
- #define CSIRX_REG_0_BPGHR0_SET_GE(val) SET_MASKED_BIT(CSIRX_REG_0_BPGHR0, val, 4)
- #define CSIRX_REG_1_BPGHR0_SET_GE(val) SET_MASKED_BIT(CSIRX_REG_1_BPGHR0, val, 4)
- #define CSIRX_REG_0_BPGHR0_SET_VLN(val) SET_MASKED_BITS(CSIRX_REG_0_BPGHR0, val, 0, 3)
- #define CSIRX_REG_1_BPGHR0_SET_VLN(val) SET_MASKED_BITS(CSIRX_REG_1_BPGHR0, val, 0, 3)

Functions

- kdrv_status_t kdrv_csi2rx_initialize (uint32_t cam_idx)

  kdrv_csi2rx_initialize Initiclize mipicsirx related variable.
- kdrv_status_t kdrv_csi2rx_enable (uint32_t input_type, uint32_t cam_idx, uint32_t sensor_idx, struct cam_format *fmt)

  kdrv_csi2rx_enable Set mipicsirx related register for IP enable.
- kdrv_status_t kdrv_csi2rx_start (uint32_t input_type, uint32_t cam_idx)

  kdrv_csi2rx_start Set mipicsirx related register for IP start.
- kdrv_status_t kdrv_csi2rx_set_power (uint32_t cam_idx, uint32_t on)

  kdrv_csi2rx_set_power Set mipicsirx power related register.
- kdrv_status_t kdrv_csi2rx_reset (uint32_t cam_idx, uint32_t sensor_idx)

  kdrv_csi2rx_reset Reset mipicsirx.

Variables

- struct kdrv_csirx_context csirx_ctx [CSI2RX_CAM_NUM]

6.50.1 Macro Definition Documentation

6.50.1.1 CSI2RX_REG_BPGHR

```
#define CSI2RX_REG_BPGHR 0x91
```

Kneron Confidential

6.50.1.2 CSI2RX_REG_BPGLR

```
#define CSI2RX_REG_BPGLR 0x90
```

6.50.1.3 CSI2RX_REG_CR

```
#define CSI2RX_REG_CR 0x04
```

6.50.1.4 CSI2RX_REG_CSIERR

```
#define CSI2RX_REG_CSIERR 0x30
```

6.50.1.5 CSI2RX_REG_DIDR

```
#define CSI2RX_REG_DIDR 0x01
```

6.50.1.6 CSI2RX_REG_DLMR

```
#define CSI2RX_REG_DLMR 0x2A
```

6.50.1.7 CSI2RX_REG_DPCMR

```
#define CSI2RX_REG_DPCMR 0x48
```

6.50.1.8 CSI2RX_REG_DPISR

```
#define CSI2RX_REG_DPISR 0x38
```

6.50.1.9 CSI2RX_REG_ECR

```
#define CSI2RX_REG_ECR 0x06
```

Kneron Confidential

6.50.1.10 CSI2RX_REG_ESR

```
#define CSI2RX_REG_ESR 0x34
```

6.50.1.11 CSI2RX_REG_FFR

```
#define CSI2RX_REG_FFR 0x3D
```

6.50.1.12 CSI2RX_REG_FIUR

```
#define CSI2RX_REG_FIUR 0x0B
```

6.50.1.13 CSI2RX_REG_FNR

```
#define CSI2RX_REG_FNR 0x88
```

6.50.1.14 CSI2RX_REG_FRCR

```
#define CSI2RX_REG_FRCR 0x80
```

6.50.1.15 CSI2RX_REG_FRR

```
#define CSI2RX_REG_FRR 0x4C
```

6.50.1.16 CSI2RX_REG_HPNR

```
#define CSI2RX_REG_HPNR 0x20
```

6.50.1.17 CSI2RX_REG_HRTVR

```
#define CSI2RX_REG_HRTVR 0x0A
```

Kneron Confidential

6.50.1.18 CSI2RX_REG_HSTR0

```
#define CSI2RX_REG_HSTR0 0x15
```

6.50.1.19 CSI2RX_REG_HSTR1

```
#define CSI2RX_REG_HSTR1 0x17
```

6.50.1.20 CSI2RX_REG_HSTR2

```
#define CSI2RX_REG_HSTR2 0x19
```

6.50.1.21 CSI2RX_REG_HSTR3

```
#define CSI2RX_REG_HSTR3 0x1B
```

6.50.1.22 CSI2RX_REG_INTER

```
#define CSI2RX_REG_INTER 0x3C
```

6.50.1.23 CSI2RX_REG_INTSTS

```
#define CSI2RX_REG_INTSTS 0x33
```

6.50.1.24 CSI2RX_REG_ITR

```
#define CSI2RX_REG_ITR 0x12
```

6.50.1.25 CSI2RX_REG_MCR

```
#define CSI2RX_REG_MCR 0x1C
```

Kneron Confidential

6.50.1.26 CSI2RX_REG_PECR

```
#define CSI2RX_REG_PECR 0x28
```

6.50.1.27 CSI2RX_REG_PFTR

```
#define CSI2RX_REG_PFTR 0x50
```

6.50.1.28 CSI2RX_REG_PUDTR

```
#define CSI2RX_REG_PUDTR 0x58
```

6.50.1.29 CSI2RX_REG_TCNR

```
#define CSI2RX_REG_TCNR 0x08
```

6.50.1.30 CSI2RX_REG_VIDR

```
#define CSI2RX_REG_VIDR 0x00
```

6.50.1.31 CSI2RX_REG_VSCR

```
#define CSI2RX_REG_VSCR 0x05
```

6.50.1.32 CSI2RX_REG_VSTER

```
#define CSI2RX_REG_VSTER 0x1E
```

6.50.1.33 CSI2RX_REG_VSTR0

```
#define CSI2RX_REG_VSTR0 0x14
```

Kneron Confidential

6.50.1.34 CSI2RX_REG_VSTR1

```
#define CSI2RX_REG_VSTR1 0x16
```

6.50.1.35 CSI2RX_REG_VSTR2

```
#define CSI2RX_REG_VSTR2 0x18
```

6.50.1.36 CSI2RX_REG_VSTR3

```
#define CSI2RX_REG_VSTR3 0x1A
```

6.50.1.37 CSIRX_0_BASE

```
#define CSIRX_0_BASE CSIRX_FTCSIRX100_PA_BASE
```

6.50.1.38 CSIRX_1_BASE

```
#define CSIRX_1_BASE CSIRX_FTCSIRX100_1_PA_BASE
```

6.50.1.39 CSIRX_REG_0_BPGHR0

```
#define CSIRX_REG_0_BPGHR0 (CSIRX_0_BASE + 0x91)
```

6.50.1.40 CSIRX_REG_0_BPGHR0_SET_GE

```
#define CSIRX_REG_0_BPGHR0_SET_GE(
              val ) SET_MASKED_BIT(CSIRX_REG_0_BPGHR0, val, 4)
```

Kneron Confidential

6.50.1.41 CSIRX_REG_0_BPGHR0_SET_PS

```
#define CSIRX_REG_0_BPGHR0_SET_PS(
              val ) SET_MASKED_BIT(CSIRX_REG_0_BPGHR0, val, 5)
```

6.50.1.42 CSIRX_REG_0_BPGHR0_SET_PT

```
#define CSIRX_REG_0_BPGHR0_SET_PT(
              val ) SET_MASKED_BITS(CSIRX_REG_0_BPGHR0, val, 6, 7)
```

6.50.1.43 CSIRX_REG_0_BPGHR0_SET_VLN

```
#define CSIRX_REG_0_BPGHR0_SET_VLN(
              val ) SET_MASKED_BITS(CSIRX_REG_0_BPGHR0, val, 0, 3)
```

6.50.1.44 CSIRX_REG_0_BPGLR0

```
#define CSIRX_REG_0_BPGLR0 (CSIRX_0_BASE + 0x90)
```

6.50.1.45 CSIRX_REG_0_BPGLR0_SET_VLN

```
#define CSIRX_REG_0_BPGLR0_SET_VLN(
              val ) SET_MASKED_BITS(CSIRX_REG_0_BPGLR0, val, 0, 7)
```

6.50.1.46 CSIRX_REG_0_DLMR0

```
#define CSIRX_REG_0_DLMR0 (CSIRX_0_BASE + 0x2A)
```

6.50.1.47 CSIRX_REG_0_DLMR0_GET_L0

```
#define CSIRX_REG_0_DLMR0_GET_L0( ) GET_BITS(CSIRX_REG_0_DLMR0, 0, 2)
```

Kneron Confidential

6.50.1.48 CSIRX_REG_0_DLMR0_GET_L1

```
#define CSIRX_REG_0_DLMR0_GET_L1( ) GET_BITS(CSIRX_REG_0_DLMR0, 4, 6)
```

6.50.1.49 CSIRX_REG_0_FEATURE0

```
#define CSIRX_REG_0_FEATURE0 (CSIRX_0_BASE + 0x40)
```

6.50.1.50 CSIRX_REG_0_FEATURE0_GET_DFAD

```
#define CSIRX_REG_0_FEATURE0_GET_DFAD( ) GET_BITS(CSIRX_REG_0_FEATURE0, 5, 7)
```

6.50.1.51 CSIRX_REG_0_FEATURE0_GET_LN

```
#define CSIRX_REG_0_FEATURE0_GET_LN( ) GET_BITS(CSIRX_REG_0_FEATURE0, 2, 4)
```

6.50.1.52 CSIRX_REG_0_FEATURE0_GET_VCN

```
#define CSIRX_REG_0_FEATURE0_GET_VCN( ) GET_BITS(CSIRX_REG_0_FEATURE0, 0, 1)
```

6.50.1.53 CSIRX_REG_0_FEATURE6

```
#define CSIRX_REG_0_FEATURE6 (CSIRX_0_BASE + 0x46)
```

6.50.1.54 CSIRX_REG_0_FEATURE6_GET_APB

```
#define CSIRX_REG_0_FEATURE6_GET_APB( ) GET_BIT(CSIRX_REG_0_FEATURE6, 6)
```

6.50.1.55 CSIRX_REG_0_FEATURE6_GET_ASC

```
#define CSIRX_REG_0_FEATURE6_GET_ASC( ) GET_BIT(CSIRX_REG_0_FEATURE6, 7)
```

Kneron Confidential

6.50.1.56 CSIRX_REG_0_FEATURE6_GET_CD

```
#define CSIRX_REG_0_FEATURE6_GET_CD( ) GET_BIT(CSIRX_REG_0_FEATURE6, 3)
```

6.50.1.57 CSIRX_REG_0_FEATURE6_GET_DP

```
#define CSIRX_REG_0_FEATURE6_GET_DP( ) GET_BIT(CSIRX_REG_0_FEATURE6, 2)
```

6.50.1.58 CSIRX_REG_0_FEATURE6_GET_DPCM

```
#define CSIRX_REG_0_FEATURE6_GET_DPCM( ) GET_BIT(CSIRX_REG_0_FEATURE6, 4)
```

6.50.1.59 CSIRX_REG_0_FEATURE6_GET_FRC

```
#define CSIRX_REG_0_FEATURE6_GET_FRC( ) GET_BIT(CSIRX_REG_0_FEATURE6, 1)
```

6.50.1.60 CSIRX_REG_0_FEATURE6_GET_I2C

```
#define CSIRX_REG_0_FEATURE6_GET_I2C( ) GET_BIT(CSIRX_REG_0_FEATURE6, 5)
```

6.50.1.61 CSIRX_REG_0_FEATURE6_GET_PG

```
#define CSIRX_REG_0_FEATURE6_GET_PG( ) GET_BIT(CSIRX_REG_0_FEATURE6, 0)
```

6.50.1.62 CSIRX_REG_0_FEATURE7

```
#define CSIRX_REG_0_FEATURE7 (CSIRX_0_BASE + 0x47)
```

6.50.1.63 CSIRX_REG_0_FEATURE7_GET_CRC

```
#define CSIRX_REG_0_FEATURE7_GET_CRC( ) GET_BIT(CSIRX_REG_0_FEATURE7, 2)
```

Kneron Confidential

6.50.1.64 CSIRX_REG_0_FEATURE7_GET_MR

```
#define CSIRX_REG_0_FEATURE7_GET_MR( ) GET_BIT(CSIRX_REG_0_FEATURE7, 3)
```

6.50.1.65 CSIRX_REG_0_FEATURE7_GET_OP

```
#define CSIRX_REG_0_FEATURE7_GET_OP( ) GET_BIT(CSIRX_REG_0_FEATURE7, 4)
```

6.50.1.66 CSIRX_REG_0_FEATURE7_GET_QP

```
#define CSIRX_REG_0_FEATURE7_GET_QP( ) GET_BIT(CSIRX_REG_0_FEATURE7, 0)
```

6.50.1.67 CSIRX_REG_0_FEATURE7_GET_SWAP

```
#define CSIRX_REG_0_FEATURE7_GET_SWAP( ) GET_BIT(CSIRX_REG_0_FEATURE7, 1)
```

6.50.1.68 CSIRX_REG_1_BPGHR0

```
#define CSIRX_REG_1_BPGHR0 (CSIRX_1_BASE + 0x91)
```

6.50.1.69 CSIRX_REG_1_BPGHR0_SET_GE

```
#define CSIRX_REG_1_BPGHR0_SET_GE(
            val ) SET_MASKED_BIT(CSIRX_REG_1_BPGHR0, val, 4)
```

6.50.1.70 CSIRX_REG_1_BPGHR0_SET_PS

```
#define CSIRX_REG_1_BPGHR0_SET_PS(
            val ) SET_MASKED_BIT(CSIRX_REG_1_BPGHR0, val, 5)
```

Kneron Confidential

6.50.1.71 CSIRX_REG_1_BPGHR0_SET_PT

```
#define CSIRX_REG_1_BPGHR0_SET_PT(
              val ) SET_MASKED_BITS(CSIRX_REG_1_BPGHR0, val, 6, 7)
```

6.50.1.72 CSIRX_REG_1_BPGHR0_SET_VLN

```
#define CSIRX_REG_1_BPGHR0_SET_VLN(
              val ) SET_MASKED_BITS(CSIRX_REG_1_BPGHR0, val, 0, 3)
```

6.50.1.73 CSIRX_REG_1_BPGLR0

```
#define CSIRX_REG_1_BPGLR0 (CSIRX_1_BASE + 0x90)
```

6.50.1.74 CSIRX_REG_1_BPGLR0_SET_VLN

```
#define CSIRX_REG_1_BPGLR0_SET_VLN(
              val ) SET_MASKED_BITS(CSIRX_REG_1_BPGLR0, val, 0, 7)
```

6.50.1.75 CSIRX_REG_1_DLMR0

```
#define CSIRX_REG_1_DLMR0 (CSIRX_1_BASE + 0x2A)
```

6.50.1.76 CSIRX_REG_1_DLMR0_GET_L0

```
#define CSIRX_REG_1_DLMR0_GET_L0( ) GET_BITS(CSIRX_REG_1_DLMR0, 0, 2)
```

6.50.1.77 CSIRX_REG_1_DLMR0_GET_L1

```
#define CSIRX_REG_1_DLMR0_GET_L1( ) GET_BITS(CSIRX_REG_1_DLMR0, 4, 6)
```

Kneron Confidential

6.50.1.78 CSIRX_REG_1_FEATURE0

```
#define CSIRX_REG_1_FEATURE0 (CSIRX_1_BASE + 0x40)
```

6.50.1.79 CSIRX_REG_1_FEATURE0_GET_DFAD

```
#define CSIRX_REG_1_FEATURE0_GET_DFAD( ) GET_BITS(CSIRX_REG_1_FEATURE0, 5, 7)
```

6.50.1.80 CSIRX_REG_1_FEATURE0_GET_LN

```
#define CSIRX_REG_1_FEATURE0_GET_LN( ) GET_BITS(CSIRX_REG_1_FEATURE0, 2, 4)
```

6.50.1.81 CSIRX_REG_1_FEATURE0_GET_VCN

```
#define CSIRX_REG_1_FEATURE0_GET_VCN( ) GET_BITS(CSIRX_REG_1_FEATURE0, 0, 1)
```

6.50.1.82 CSIRX_REG_1_FEATURE6

```
#define CSIRX_REG_1_FEATURE6 (CSIRX_1_BASE + 0x46)
```

6.50.1.83 CSIRX_REG_1_FEATURE6_GET_APB

```
#define CSIRX_REG_1_FEATURE6_GET_APB( ) GET_BIT(CSIRX_REG_1_FEATURE6, 6)
```

6.50.1.84 CSIRX_REG_1_FEATURE6_GET_ASC

```
#define CSIRX_REG_1_FEATURE6_GET_ASC( ) GET_BIT(CSIRX_REG_1_FEATURE6, 7)
```

6.50.1.85 CSIRX_REG_1_FEATURE6_GET_CD

```
#define CSIRX_REG_1_FEATURE6_GET_CD( ) GET_BIT(CSIRX_REG_1_FEATURE6, 3)
```

Kneron Confidential

6.50.1.86 CSIRX_REG_1_FEATURE6_GET_DP

```
#define CSIRX_REG_1_FEATURE6_GET_DP( ) GET_BIT(CSIRX_REG_1_FEATURE6, 2)
```

6.50.1.87 CSIRX_REG_1_FEATURE6_GET_DPCM

```
#define CSIRX_REG_1_FEATURE6_GET_DPCM( ) GET_BIT(CSIRX_REG_1_FEATURE6, 4)
```

6.50.1.88 CSIRX_REG_1_FEATURE6_GET_FRC

```
#define CSIRX_REG_1_FEATURE6_GET_FRC( ) GET_BIT(CSIRX_REG_1_FEATURE6, 1)
```

6.50.1.89 CSIRX_REG_1_FEATURE6_GET_I2C

```
#define CSIRX_REG_1_FEATURE6_GET_I2C( ) GET_BIT(CSIRX_REG_1_FEATURE6, 5)
```

6.50.1.90 CSIRX_REG_1_FEATURE6_GET_PG

```
#define CSIRX_REG_1_FEATURE6_GET_PG( ) GET_BIT(CSIRX_REG_1_FEATURE6, 0)
```

6.50.1.91 CSIRX_REG_1_FEATURE7

```
#define CSIRX_REG_1_FEATURE7 (CSIRX_1_BASE + 0x47)
```

6.50.1.92 CSIRX_REG_1_FEATURE7_GET_CRC

```
#define CSIRX_REG_1_FEATURE7_GET_CRC( ) GET_BIT(CSIRX_REG_1_FEATURE7, 2)
```

6.50.1.93 CSIRX_REG_1_FEATURE7_GET_MR

```
#define CSIRX_REG_1_FEATURE7_GET_MR( ) GET_BIT(CSIRX_REG_1_FEATURE7, 3)
```

Kneron Confidential

6.50.1.94 CSIRX_REG_1_FEATURE7_GET_OP

```
#define CSIRX_REG_1_FEATURE7_GET_OP( ) GET_BIT(CSIRX_REG_1_FEATURE7, 4)
```

6.50.1.95 CSIRX_REG_1_FEATURE7_GET_QP

```
#define CSIRX_REG_1_FEATURE7_GET_QP( ) GET_BIT(CSIRX_REG_1_FEATURE7, 0)
```

6.50.1.96 CSIRX_REG_1_FEATURE7_GET_SWAP

```
#define CSIRX_REG_1_FEATURE7_GET_SWAP( ) GET_BIT(CSIRX_REG_1_FEATURE7, 1)
```

6.50.1.97 CSIRX_REG_PECR_GET_PEC

```
#define CSIRX_REG_PECR_GET_PEC(
              addr ) GET_BITS(addr, 0, 7)
```

6.50.1.98 CSIRX_REG_PECR_SET_PEC

```
#define CSIRX_REG_PECR_SET_PEC(
              addr,
              val ) SET_MASKED_BITS(addr, val, 0, 7)
```

6.50.1.99 CSIRX_REG_VSCR_SET_VSTU0

```
#define CSIRX_REG_VSCR_SET_VSTU0(
              addr,
              val ) SET_MASKED_BIT(addr, val, 0)
```

6.50.1.100 MAX_VIDEO_MEM

```
#define MAX_VIDEO_MEM 8 << 20
```

Kneron Confidential

6.50.1.101 ROUND_UP

```
#define ROUND_UP(
                x,
                y ) ((((x) + (y - 1)) / y) * y)
```

6.50.2 Variable Documentation

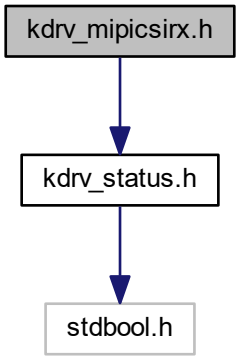
6.50.2.1 csirx_ctx

```
struct kdrv_csirx_context csirx_ctx[CSI2RX_CAM_NUM]
```

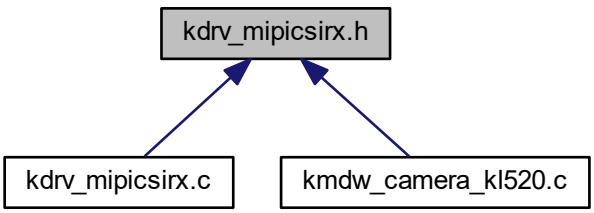
6.51 kdrv_mipicsirx.h File Reference

```
#include "kdrv_status.h"
```

Include dependency graph for kdrv_mipicsirx.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Enumerations

- enum { CSI2RX_CAM_0, CSI2RX_CAM_1, CSI2RX_CAM_NUM }

Functions

- kdrv_status_t kdrv_csi2rx_initialize (uint32_t cam_idx)

  kdrv_csi2rx_initialize Initiclize mipicsirx related variable.
- kdrv_status_t kdrv_csi2rx_enable (uint32_t input_type, uint32_t cam_idx, uint32_t sensor_idx, struct cam_format *fmt)

  kdrv_csi2rx_enable Set mipicsirx related register for IP enable.
- kdrv_status_t kdrv_csi2rx_start (uint32_t input_type, uint32_t cam_idx)

  kdrv_csi2rx_start Set mipicsirx related register for IP start.
- kdrv_status_t kdrv_csi2rx_set_power (uint32_t cam_idx, uint32_t on)

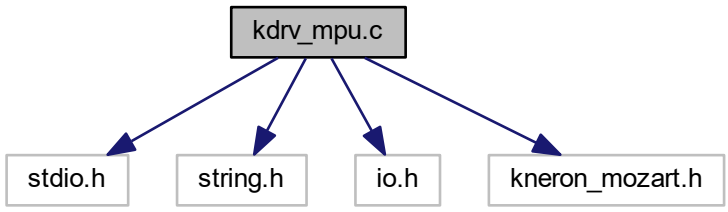
  kdrv_csi2rx_set_power Set mipicsirx power related register.
- kdrv_status_t kdrv_csi2rx_reset (uint32_t cam_idx, uint32_t sensor_idx)

  kdrv_csi2rx_reset Reset mipicsirx.

6.52 kdrv_mpu.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "io.h"
#include "kneron_mozart.h"
```

Include dependency graph for kdrv_mpu.c:



Kneron Confidential

Macros

- #define ROM_MEM_BASE 0x10000000
- #define SiRAM_MEM_VECTOR_OFFSET 0x2000
- #define MPU_DEFAULT_MAP 0x04
- #define MPU_XN 1U << 28
- #define MPU_1KB (ARM_MPU_REGION_SIZE_1KB << 1)
- #define MPU_32KB (ARM_MPU_REGION_SIZE_32KB << 1)
- #define MPU_64KB (ARM_MPU_REGION_SIZE_64KB << 1)
- #define MPU_128KB (ARM_MPU_REGION_SIZE_128KB << 1)

- #define MPU_96KB_SRD (0xC0 << 8)
- #define MPU_64MB (ARM_MPU_REGION_SIZE_64MB << 1)
- #define MPU_4GB (ARM_MPU_REGION_SIZE_4GB << 1)
- #define MPU_2pGB_SRD (0xE0 << 8)
- #define MPU_RW (ARM_MPU_AP_FULL << 24)
- #define MPU_RO (ARM_MPU_AP_RO << 24)
- #define MPU_NA (ARM_MPU_AP_NONE << 24)
- #define MPU_CACHE (0x0F << 16)
- #define MPU_DEV_S (0x05 << 16)
- #define MPU_EN 1

Functions

- void kdrv_mpu_config (void)

  config memorty protect space, siram + niram
- void kdrv_mpu_niram_enable (void)

  mpu protect enable for niram memory space
- void kdrv_mpu_niram_disable (void)

  mpu protect disable for niram memory space

6.52.1 Macro Definition Documentation

6.52.1.1 MPU_128KB

```
#define MPU_128KB (ARM_MPU_REGION_SIZE_128KB << 1)
```

6.52.1.2 MPU_1KB

```
#define MPU_1KB (ARM_MPU_REGION_SIZE_1KB << 1)
```

6.52.1.3 MPU_2pGB_SRD

```
#define MPU_2pGB_SRD (0xE0 << 8)
```

6.52.1.4 MPU_32KB

```
#define MPU_32KB (ARM_MPU_REGION_SIZE_32KB << 1)
```

Kneron Confidential

6.52.1.5 MPU_4GB

```
#define MPU_4GB (ARM_MPU_REGION_SIZE_4GB << 1)
```

6.52.1.6 MPU_64KB

```
#define MPU_64KB (ARM_MPU_REGION_SIZE_64KB << 1)
```

6.52.1.7 MPU_64MB

```
#define MPU_64MB (ARM_MPU_REGION_SIZE_64MB << 1)
```

6.52.1.8 MPU_96KB_SRD

```
#define MPU_96KB_SRD (0xC0 << 8)
```

6.52.1.9 MPU_CACHE

```
#define MPU_CACHE (0x0F << 16)
```

6.52.1.10 MPU_DEFAULT_MAP

```
#define MPU_DEFAULT_MAP 0x04
```

6.52.1.11 MPU_DEV_S

```
#define MPU_DEV_S (0x05 << 16)
```

6.52.1.12 MPU_EN

```
#define MPU_EN 1
```

Kneron Confidential

6.52.1.13 MPU_NA

```
#define MPU_NA (ARM_MPU_AP_NONE << 24)
```

6.52.1.14 MPU_RO

```
#define MPU_RO (ARM_MPU_AP_RO << 24)
```

6.52.1.15 MPU_RW

```
#define MPU_RW (ARM_MPU_AP_FULL << 24)
```

6.52.1.16 MPU_XN

```
#define MPU_XN 1U << 28
```

6.52.1.17 ROM_MEM_BASE

```
#define ROM_MEM_BASE 0x10000000
```

6.52.1.18 SiRAM_MEM_VECTOR_OFFSET

```
#define SiRAM_MEM_VECTOR_OFFSET 0x2000
```

6.53 kdrv_mpu.h File Reference

Functions

- void kdrv_mpu_config (void)

  config memorty protect space, siram + niram
- void kdrv_mpu_niram_enable (void)

  mpu protect enable for niram memory space
- void kdrv_mpu_niram_disable (void)

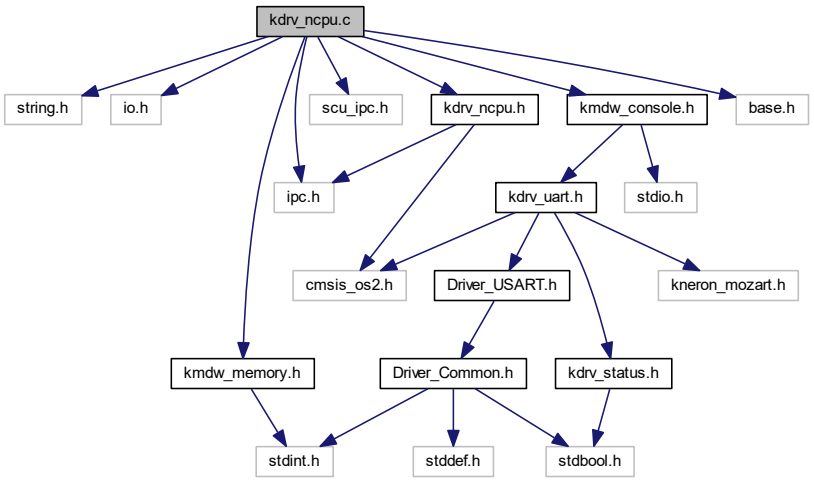
  mpu protect disable for niram memory space

Kneron Confidential

6.54 kdrv_ncpu.c File Reference

```
#include <string.h>
#include "io.h"
#include "kmdw_memory.h"
#include "ipc.h"
#include "scu_ipc.h"
#include "kdrv_ncpu.h"
#include "kmdw_console.h"
#include "base.h"
```

Include dependency graph for kdrv_ncpu.c:



Macros

- #define NCPU_IRQ 51
- #define INPROC_ARRAY_MAX_SIZE 1000
- #define CMD_FLAGS(i) (s_out_comm->cmd_flags[i])
- #define CMD_FLAGS_IS_ACTIVE(i) (s_out_comm->cmd_flags[i] == IMAGE_STATE_ACTIVE)
- #define CMD_FLAGS_IS_INACTIVE(i) (s_out_comm->cmd_flags[i] == IMAGE_STATE_INACTIVE)
- #define CMD_STATUS(i) (s_in_comm->cmd_status[i])
- #define CMD_STATUS_IS_NPU_BUSY(i) (s_in_comm->cmd_status[i] == IMAGE_STATE_NPU_BUSY)
- #define CMD_STATUS_IS_NPU_DONE(i) (s_in_comm->cmd_status[i] == IMAGE_STATE_NPU_DONE)
- #define CMD_STATUS_IS_PROC_DONE(i) (s_in_comm->cmd_status[i] == IMAGE_STATE_POST_PR↩ OCESSING_DONE)

Functions

- void kdrv_ncpu_trigger_int (void)

  Trigger NCPU interrupt.
- int kdrv_ncpu_get_avail_com (void)

  Get available COM.
- int kdrv_ncpu_set_image_active (uint32_t n_index)

  Set active image index.

Kneron Confidential

- int kdrv_ncpu_get_image_active ()

  Get active image index.
- int kdrv_ncpu_set_model_active (uint32_t n_index)

  Set active model index.
- int kdrv_ncpu_get_model_active ()

  Get active model index.
- void kdrv_ncpu_set_model (struct kdp_model_s *model_info_addr, uint32_t info_idx, int32_t slot_idx)

  Set model information.
- void kdrv_ncpu_initialize (ipc_handler_t ipc_handler)

  Initialize NPU functionality.
- struct ncpu_to_scpu_s * kdrv_ncpu_get_input (void)

  Get ncpu_to_scpu_s.
- struct scpu_to_ncpu_s * kdrv_ncpu_get_output (void)

  Get scpu_to_ncpu_s.
- void kdrv_ncpu_set_scpu_debug_lvl (uint32_t lvl)

  Set SCPU debug level.
- void kdrv_ncpu_set_ncpu_debug_lvl (uint32_t lvl)

  Set NCPU debug level.

Variables

- ipc_handler_t ipc_handler_cb
- struct scpu_to_ncpu_s * s_out_comm
- struct ncpu_to_scpu_s * s_in_comm

6.54.1 Macro Definition Documentation

6.54.1.1 CMD_FLAGS

```
#define CMD_FLAGS(
              i ) (s_out_comm->cmd_flags[i])
```

6.54.1.2 CMD_FLAGS_IS_ACTIVE

```
#define CMD_FLAGS_IS_ACTIVE(
              i ) (s_out_comm->cmd_flags[i] == IMAGE_STATE_ACTIVE)
```

6.54.1.3 CMD_FLAGS_IS_INACTIVE

```
#define CMD_FLAGS_IS_INACTIVE(
              i ) (s_out_comm->cmd_flags[i] == IMAGE_STATE_INACTIVE)
```

Kneron Confidential

6.54.1.4 CMD_STATUS

```
#define CMD_STATUS(
              i ) (s_in_comm->cmd_status[i])
```

6.54.1.5 CMD_STATUS_IS_NPU_BUSY

```
#define CMD_STATUS_IS_NPU_BUSY(
              i ) (s_in_comm->cmd_status[i] == IMAGE_STATE_NPU_BUSY)
```

6.54.1.6 CMD_STATUS_IS_NPU_DONE

```
#define CMD_STATUS_IS_NPU_DONE(
              i ) (s_in_comm->cmd_status[i] == IMAGE_STATE_NPU_DONE)
```

6.54.1.7 CMD_STATUS_IS_PROC_DONE

```
#define CMD_STATUS_IS_PROC_DONE(
              i ) (s_in_comm->cmd_status[i] == IMAGE_STATE_POST_PROCESSING_DONE)
```

6.54.1.8 INPROC_ARRAY_MAX_SIZE

```
#define INPROC_ARRAY_MAX_SIZE 1000
```

6.54.1.9 NCPU_IRQ

```
#define NCPU_IRQ 51
```

6.54.2 Variable Documentation

6.54.2.1 ipc_handler_cb

```
ipc_handler_t ipc_handler_cb
```

Kneron Confidential

6.54.2.2 s_in_comm

```
struct ncpu_to_scpu_s* s_in_comm
```

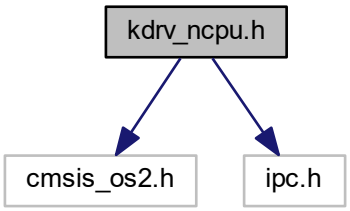
6.54.2.3 s_out_comm

```
struct scpu_to_ncpu_s* s_out_comm
```

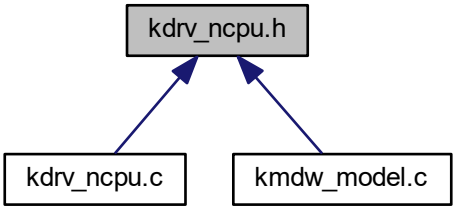
6.55 kdrv_ncpu.h File Reference

```
#include "cmsis_os2.h"
#include "ipc.h"
```

Include dependency graph for kdrv_ncpu.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef void(* ipc_handler_t) (int ipc_idx, int state)

Kneron Confidential

Functions

- void kdrv_ncpu_initialize (ipc_handler_t ipc_handler)

  Initialize NPU functionality.
- void kdrv_ncpu_set_model (struct kdp_model_s *model_info_addr, uint32_t info_idx, int32_t slot_idx)

  Set model information.
- int kdrv_ncpu_get_avail_com (void)

  Get available COM.
- int kdrv_ncpu_set_model_active (uint32_t index)

  Set active model index.
- int kdrv_ncpu_get_model_active (void)

  Get active model index.
- int kdrv_ncpu_set_image_active (uint32_t index)

  Set active image index.
- int kdrv_ncpu_get_image_active (void)

  Get active image index.
- void kdrv_ncpu_set_scpu_debug_lvl (uint32_t lvl)

  Set SCPU debug level.
- void kdrv_ncpu_set_ncpu_debug_lvl (uint32_t lvl)

  Set NCPU debug level.
- void kdrv_ncpu_trigger_int (void)

  Trigger NCPU interrupt.
- struct ncpu_to_scpu_s * kdrv_ncpu_get_input (void)

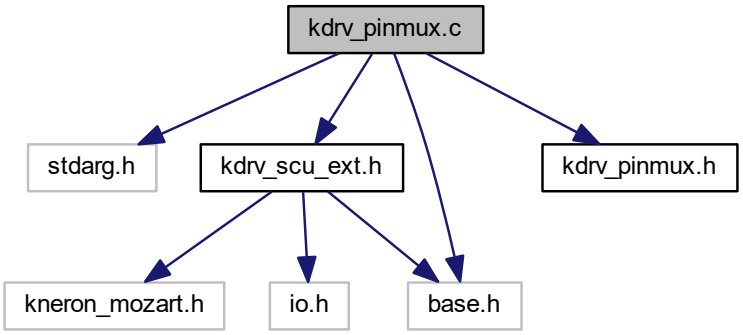
  Get ncpu_to_scpu_s.
- struct scpu_to_ncpu_s * kdrv_ncpu_get_output (void)

  Get scpu_to_ncpu_s.

6.56 kdrv_pinmux.c File Reference

```
#include <stdarg.h>
#include "kdrv_scu_ext.h"
#include "base.h"
#include "kdrv_pinmux.h"
```

Include dependency graph for kdrv_pinmux.c:



Kneron Confidential

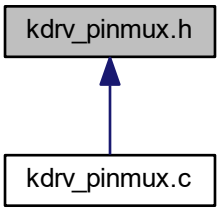
Functions

- void kdrv_pinmux_config (kdrv_pin_name pin, kdrv_pinmux_mode mode, kdrv_pin_pull pull_type, kdrv_pin_driving driving)

  Pinmux configure.

6.57 kdrv_pinmux.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum kdrv_pin_name {
  KDRV_PIN_SPI_WP_N = 0, KDRV_PIN_SPI_HOLD_N, KDRV_PIN_JTAG_TRST_N, KDRV_PIN_JTAG_TDI, KDRV_PIN_JTAG_SWDITMS, KDRV_PIN_JTAG_SWCLKTCK, KDRV_PIN_JTAG_TDO, KDRV_PIN_LC_PCLK, KDRV_PIN_LC_VS, KDRV_PIN_LC_HS, KDRV_PIN_LC_DE, KDRV_PIN_LC_DATA_0, KDRV_PIN_LC_DATA_1, KDRV_PIN_LC_DATA_2, KDRV_PIN_LC_DATA_3, KDRV_PIN_LC_DATA_4, KDRV_PIN_LC_DATA_5, KDRV_PIN_LC_DATA_6, KDRV_PIN_LC_DATA_7, KDRV_PIN_LC_DATA_8, KDRV_PIN_LC_DATA_9, KDRV_PIN_LC_DATA_10, KDRV_PIN_LC_DATA_11, KDRV_PIN_LC_DATA_12, KDRV_PIN_LC_DATA_13, KDRV_PIN_LC_DATA_14, KDRV_PIN_LC_DATA_15, KDRV_PIN_SD_CLK, KDRV_PIN_SD_CMD, KDRV_PIN_SD_DAT_0, KDRV_PIN_SD_DAT_1, KDRV_PIN_SD_DAT_2, KDRV_PIN_SD_DAT_3, KDRV_PIN_UART0_RX, KDRV_PIN_UART0_TX, KDRV_PIN_I2C0_SCL, KDRV_PIN_I2C0_SDA, KDRV_PIN_PWM0 }

  Enumerations of KDP520 all configurable pins.
- enum kdrv_pinmux_mode {
  PIN_MODE_0 = 0, PIN_MODE_1, PIN_MODE_2, PIN_MODE_3, PIN_MODE_4, PIN_MODE_5, PIN_MODE_6, PIN_MODE_7 }

  Enumerations of KDP520 pinmux modes.
- enum kdrv_pin_pull { PIN_PULL_NONE, PIN_PULL_UP, PIN_PULL_DOWN }

  Enumerations of KDP520 pull status.
- enum kdrv_pin_driving {
  PIN_DRIVING_NONE, PIN_DRIVING_4MA, PIN_DRIVING_8MA, PIN_DRIVING_12MA, PIN_DRIVING_16MA }

  Enumerations of KDP520 output driving capability.

Kneron Confidential

Functions

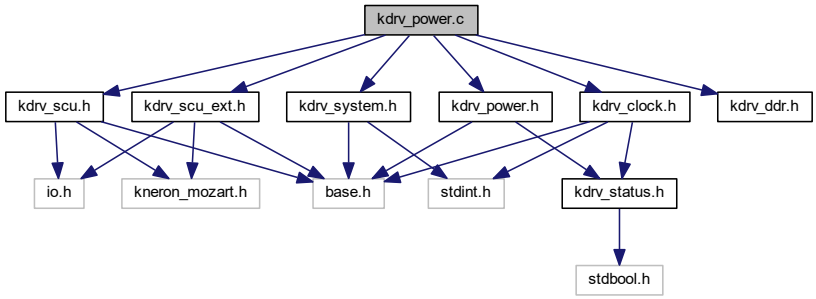
- void kdrv_pinmux_config (kdrv_pin_name pin, kdrv_pinmux_mode mode, kdrv_pin_pull pull_type, kdrv_pin_driving driving)

  Pinmux configure.

6.58 kdrv_power.c File Reference

```
#include "kdrv_power.h"
#include "kdrv_scu.h"
#include "kdrv_scu_ext.h"
#include "kdrv_system.h"
#include "kdrv_clock.h"
#include "kdrv_ddr.h"
```

Include dependency graph for kdrv_power.c:



Macros

- #define POWER_DOMAIN_WORKING_NONE 0x00000000
- #define POWER_DOMAIN_WORKING_DEFAULT 0x00000001
- #define POWER_DOMAIN_WORKING_NPU 0x00000002
- #define POWER_DOMAIN_WORKING_DDR 0x00000004
- #define POWER_DOMAIN_WORKING_ALL (POWER_DOMAIN_WORKING_DEFAULT | POWER_DOMAIN_WORKING_NPU | POWER_DOMAIN_WORKING_DDR)
- #define POWER_DOMAIN_SOFTOFF_DEFAULT 0x00000010
- #define POWER_DOMAIN_SOFTOFF_NPU 0x00000020
- #define POWER_DOMAIN_SOFTOFF_DDR 0x00000040
- #define POWER_DOMAIN_SOFTOFF_MASK
- #define PWR_CTRL_SOFTOFF_MASK
- #define PWR_CTRL_SOFTOFF_DEEP_RETENTION
- #define PWR_CTRL_SOFTOFF_RTC_MODE

Kneron Confidential

Functions

- void kdrv_power_sw_reset (void)

  Watchdog reset.
- kdrv_status_t kdrv_power_set_domain (kdrv_power_domain_t domain, int enable)

  Set power domain.
- kdrv_status_t kdrv_power_softoff (kdrv_power_mode_t mode)

  Shutdown the power supply to all blocks, except the logic in the RTC domain and DDR memory is in self-refresh state.
- kdrv_status_t kdrv_power_ops (kdrv_power_ops_t ops)

  Power operation.

Variables

- kdrv_power_mode_t __power_mgr_mode = POWER_MODE_RTC

6.58.1 Macro Definition Documentation

6.58.1.1 POWER_DOMAIN_SOFTOFF_DDR

```
#define POWER_DOMAIN_SOFTOFF_DDR 0x00000040
```

6.58.1.2 POWER_DOMAIN_SOFTOFF_DEFAULT

```
#define POWER_DOMAIN_SOFTOFF_DEFAULT 0x00000010
```

6.58.1.3 POWER_DOMAIN_SOFTOFF_MASK

```
#define POWER_DOMAIN_SOFTOFF_MASK
```

Value:

```
(POWER_DOMAIN_SOFTOFF_DEFAULT | \
                                        POWER_DOMAIN_SOFTOFF_NPU | \
                                        POWER_DOMAIN_SOFTOFF_DDR)
```

6.58.1.4 POWER_DOMAIN_SOFTOFF_NPU

```
#define POWER_DOMAIN_SOFTOFF_NPU 0x00000020
```

Kneron Confidential

6.58.1.5 POWER_DOMAIN_WORKING_ALL

```
#define POWER_DOMAIN_WORKING_ALL (POWER_DOMAIN_WORKING_DEFAULT | POWER_DOMAIN_WORKING_NPU |
POWER_DOMAIN_WORKING_DDR)
```

6.58.1.6 POWER_DOMAIN_WORKING_DDR

```
#define POWER_DOMAIN_WORKING_DDR 0x00000004
```

6.58.1.7 POWER_DOMAIN_WORKING_DEFAULT

```
#define POWER_DOMAIN_WORKING_DEFAULT 0x00000001
```

6.58.1.8 POWER_DOMAIN_WORKING_NONE

```
#define POWER_DOMAIN_WORKING_NONE 0x00000000
```

6.58.1.9 POWER_DOMAIN_WORKING_NPU

```
#define POWER_DOMAIN_WORKING_NPU 0x00000002
```

6.58.1.10 PWR_CTRL_SOFTOFF_DEEP_RETENTION

```
#define PWR_CTRL_SOFTOFF_DEEP_RETENTION
```

Value:

```
(SCU_REG_PWR_CTRL_PWRUP_UPDATE | \
                                    SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DDRCK | \
                                    SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DEFAULT | \
                                    SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DDRCK)
```

6.58.1.11 PWR_CTRL_SOFTOFF_MASK

```
#define PWR_CTRL_SOFTOFF_MASK
```

Value:

```
(SCU_REG_PWR_CTRL_PWRUP_UPDATE | \
                                SCU_REG_PWR_CTRL_PWRUP_CTRL_MASK | \
                                SCU_REG_PWR_CTRL_PWRDN_CTRL_MASK)
```

Kneron Confidential

6.58.1.12 PWR_CTRL_SOFTOFF_RTC_MODE

```
#define PWR_CTRL_SOFTOFF_RTC_MODE
```

Value:

```
(SCU_REG_PWR_CTRL_PWRUP_UPDATE | \
                                        SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DEFAULT)
```

6.58.2 Variable Documentation

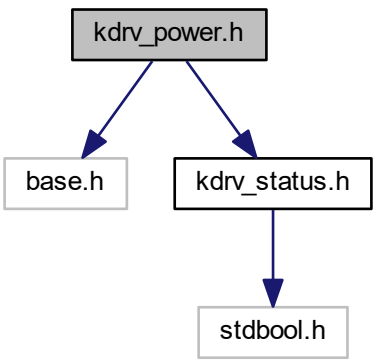
6.58.2.1 __power_mgr_mode

```
kdrv_power_mode_t __power_mgr_mode = POWER_MODE_RTC
```

6.59 kdrv_power.h File Reference

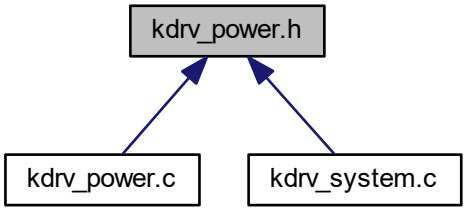
```
#include <base.h>
#include "kdrv_status.h"
```

Include dependency graph for kdrv_power.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Enumerations

- enum kdrv_power_domain_t { POWER_DOMAIN_DEFAULT = 1, POWER_DOMAIN_NPU, POWER_DOMAIN_DDRCK }

  Enumerations of kl520 power domains.
- enum kdrv_power_ops_t { POWER_OPS_FCS = 0, POWER_OPS_CHANGE_BUS_SPEED, POWER_OPS_PLL_UPDATE, POWER_OPS_SLEEPING }

  Enumerations of kl520 power operations.
- enum kdrv_power_mode_t { POWER_MODE_RTC = 0, POWER_MODE_ALWAYSON, POWER_MODE_FULL, POWER_MODE_RETENTION, POWER_MODE_DEEP_RETENTION }

  Enumerations of kl520 power modes.

Functions

- void kdrv_power_sw_reset (void)

  Watchdog reset.
- kdrv_status_t kdrv_power_ops (kdrv_power_ops_t ops)

  Power operation.
- kdrv_status_t kdrv_power_set_domain (kdrv_power_domain_t domain, int enable)

  Set power domain.
- kdrv_status_t kdrv_power_softoff (kdrv_power_mode_t mode)

  Shutdown the power supply to all blocks, except the logic in the RTC domain and DDR memory is in self-refresh state.

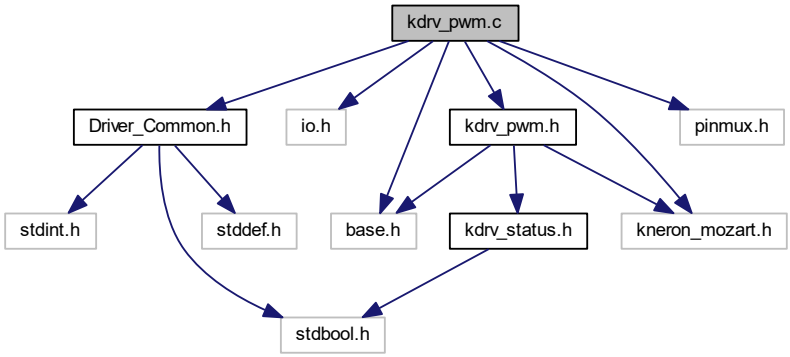
6.60 kdrv_pwm.c File Reference

```
#include "base.h"
#include "io.h"
#include "Driver_Common.h"
#include "kneron_mozart.h"
#include "kdrv_pwm.h"
```

Kneron Confidential

```
#include "pinmux.h"
```

Include dependency graph for kdrv_pwm.c:



Data Structures

- struct kdrv_pwmtimer_control
- struct kdrv_pwmtimer_struct

Macros

- #define pwm_msg(fmt, ...)
- #define MAX_PWM_TIMER 6
- #define HZ 100
- #define MAX_TIMER 7
- #define TIMER_INTSTAT 0x0
- #define TIMER_CR 0x0
- #define TIMER_LOAD 0x4
- #define TIMER_COMPARE 0x8
- #define TIMER_CNTO 0xc

Typedefs

- typedef void(∗ timer_isr) (void)

Functions

- uint32_t kdrv_current_t1_tick (void)

  Get t1 tick.
- uint32_t kdrv_current_t2_tick (void)

  Get t2 tick.
- uint32_t kdrv_current_t3_tick (void)

  Get t3 tick.
- uint32_t kdrv_current_t4_tick (void)

Kneron Confidential

Get t4 tick.

- uint32_t kdrv_current_t5_tick (void)

  Get t5 tick.

- uint32_t kdrv_current_t6_tick (void)

  Get t6 tick.

- kdrv_status_t kdrv_pwmtimer_initialize (pwmtimer timer, uint32_t tick)

  Initialize specific timer id and give tick.

- kdrv_status_t kdrv_pwmtimer_close (pwmtimer timer)

  Close specifiec pwm timer id.

- kdrv_status_t kdrv_pwmtimer_tick_reset (pwmtimer timer)

  Reset pwm timer tick.

- kdrv_status_t kdrv_pwmtimer_delay_ms (uint32_t msec)

  Use pwm timer to delay for certain time interval.

- kdrv_status_t kdrv_pwm_config (pwmtimer timer, pwmpolarity polarity, uint32_t duty, uint32_t period, bool ns2clkcnt)

  kdrv_pwm_config

- kdrv_status_t kdrv_pwm_enable (pwmtimer timer)

  kdrv_pwm_enable

- kdrv_status_t kdrv_pwm_disable (pwmtimer timer)

  kdrv_pwm_disable

Variables

- kdrv_pwmtimer_control * timer_control [MAX_TIMER+1]
- uint32_t * CNTBBase_tmp
- uint32_t * CMPBBase_tmp
- uint32_t t1_tick = 0
- uint32_t t2_tick = 0
- uint32_t t3_tick = 0
- uint32_t t4_tick = 0
- uint32_t t5_tick = 0
- uint32_t t6_tick = 0

6.60.1 Macro Definition Documentation

6.60.1.1 HZ

```
#define HZ 100
```

6.60.1.2 MAX_PWM_TIMER

```
#define MAX_PWM_TIMER 6
```

Kneron Confidential

6.60.1.3 MAX_TIMER

```
#define MAX_TIMER 7
```

6.60.1.4 pwm_msg

```
#define pwm_msg(
              fmt,
              ...  )
```

6.60.1.5 TIMER_CNTO

```
#define TIMER_CNTO 0xc
```

6.60.1.6 TIMER_COMPARE

```
#define TIMER_COMPARE 0x8
```

6.60.1.7 TIMER_CR

```
#define TIMER_CR 0x0
```

6.60.1.8 TIMER_INTSTAT

```
#define TIMER_INTSTAT 0x0
```

6.60.1.9 TIMER_LOAD

```
#define TIMER_LOAD 0x4
```

6.60.2 Typedef Documentation

Kneron Confidential

6.60.2.1 timer_isr

```
typedef void(* timer_isr) (void)
```

6.60.3 Variable Documentation

6.60.3.1 CMPBBase_tmp

```
uint32_t* CMPBBase_tmp
```

6.60.3.2 CNTBBase_tmp

```
uint32_t* CNTBBase_tmp
```

6.60.3.3 t1_tick

```
uint32_t t1_tick = 0
```

6.60.3.4 t2_tick

```
uint32_t t2_tick = 0
```

6.60.3.5 t3_tick

```
uint32_t t3_tick = 0
```

6.60.3.6 t4_tick

```
uint32_t t4_tick = 0
```

Kneron Confidential

6.60.3.7 t5_tick

```
uint32_t t5_tick = 0
```

6.60.3.8 t6_tick

```
uint32_t t6_tick = 0
```

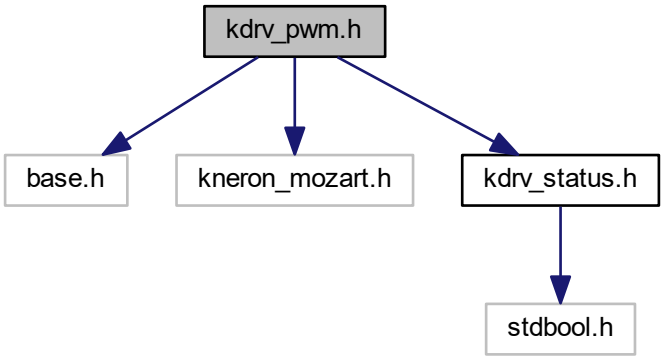
6.60.3.9 timer_control

```
kdrv_pwmtimer_control* timer_control[MAX_TIMER+1]
```

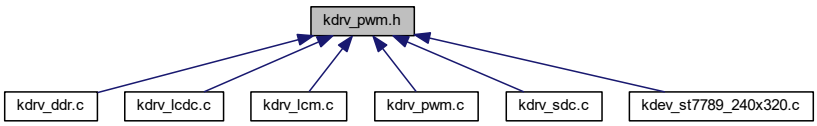
6.61 kdrv_pwm.h File Reference

```
#include "base.h"
#include "kneron_mozart.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_pwm.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define APB_CLK APB_CLOCK
- #define PWMTMR_1000MSEC_PERIOD (uint32_t)(APB_CLK)
- #define PWMTMR_5000MSEC_PERIOD (uint32_t)(APB_CLK∗5)
- #define PWMTMR_1000MSEC_PERIOD (uint32_t)(APB_CLK)
- #define PWMTMR_100MSEC_PERIOD (uint32_t)(APB_CLK/10)
- #define PWMTMR_20MSEC_PERIOD (uint32_t)(APB_CLK/50)
- #define PWMTMR_15MSEC_PERIOD (uint32_t)(((APB_CLK/100)∗3)/2)
- #define PWMTMR_10MSEC_PERIOD (uint32_t)(APB_CLK/100)
- #define PWMTMR_1MSEC_PERIOD (uint32_t)(APB_CLK/1000)
- #define PWMTMR_01MSEC_PERIOD (uint32_t)(APB_CLK/10000)

Typedefs

- typedef enum Timer_IoType timeriotype

  Enumerations of kl520 power domains.

Enumerations

- enum pwmtimer {
  PWMTIMER1 =1, PWMTIMER2 =2, PWMTIMER3 =3, PWMTIMER4 =4,
  PWMTIMER5 =5, PWMTIMER6 =6 }

  Enumerations of all timer callback event return status.
- enum Timer_IoType { IO_TIMER_RESETALL, IO_TIMER_GETTICK, IO_TIMER_SETTICK, IO_TIMER_SETCLKSRC }

  Enumerations of kl520 power domains.
- enum pwmpolarity { PWM_POLARITY_NORMAL = 0, PWM_POLARITY_INVERSED }

  Enumerations of polarity of a PWM signal.

Functions

- uint32_t kdrv_current_t1_tick (void)

  Get t1 tick.
- uint32_t kdrv_current_t2_tick (void)

  Get t2 tick.
- uint32_t kdrv_current_t3_tick (void)

  Get t3 tick.
- uint32_t kdrv_current_t4_tick (void)

  Get t4 tick.
- uint32_t kdrv_current_t5_tick (void)

  Get t5 tick.
- uint32_t kdrv_current_t6_tick (void)

  Get t6 tick.
- kdrv_status_t kdrv_pwmtimer_initialize (pwmtimer timer, uint32_t tick)

  Initialize specific timer id and give tick.
- kdrv_status_t kdrv_pwmtimer_close (pwmtimer timer)

  Close specifiec pwm timer id.
- kdrv_status_t kdrv_pwmtimer_tick_reset (pwmtimer timer)

  Reset pwm timer tick.

Kneron Confidential

- kdrv_status_t kdrv_pwmtimer_delay_ms (uint32_t msec)

  Use pwm timer to delay for certain time interval.

- kdrv_status_t kdrv_pwm_config (pwmtimer timer, pwmpolarity polarity, uint32_t duty, uint32_t period, bool ns2clkcnt)

  kdrv_pwm_config

- kdrv_status_t kdrv_pwm_enable (pwmtimer timer)

  kdrv_pwm_enable

- kdrv_status_t kdrv_pwm_disable (pwmtimer timer)

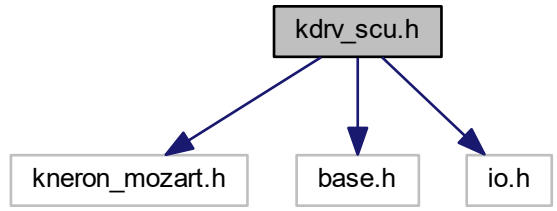
  kdrv_pwm_disable

6.62 kdrv_reg.h File Reference

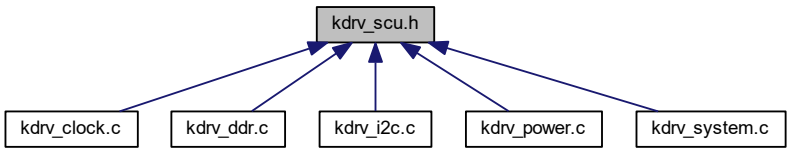
6.63 kdrv_scu.h File Reference

```
#include "kneron_mozart.h"
#include "base.h"
#include "io.h"
```

Include dependency graph for kdrv_scu.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define SCU_REG_BTUP_STS (SCU_FTSCU100_PA_BASE + 0x0000)
- #define SCU_REG_BTUP_CTRL (SCU_FTSCU100_PA_BASE + 0x0004)
- #define SCU_REG_PWR_CTRL (SCU_FTSCU100_PA_BASE + 0x0008)
- #define SCU_REG_PWR_MOD (SCU_FTSCU100_PA_BASE + 0x0020)
- #define SCU_REG_INT_STS (SCU_FTSCU100_PA_BASE + 0x0024)
- #define SCU_REG_INT_EN (SCU_FTSCU100_PA_BASE + 0x0028)
- #define SCU_REG_PLL_CTRL (SCU_FTSCU100_PA_BASE + 0x0030)
- #define SCU_REG_PLL2_CTRL (SCU_FTSCU100_PA_BASE + 0x0040)
- #define SCU_REG_DLL_CTRL (SCU_FTSCU100_PA_BASE + 0x0044)
- #define SCU_REG_PWR_VCCSTS (SCU_FTSCU100_PA_BASE + 0x0048)
- #define SCU_REG_AHBCLKG (SCU_FTSCU100_PA_BASE + 0x0050)
- #define SCU_REG_SLP_AHBCLKG (SCU_FTSCU100_PA_BASE + 0x0058)
- #define SCU_REG_APBCLKG (SCU_FTSCU100_PA_BASE + 0x0060)
- #define SCU_REG_APBCLKG2 (SCU_FTSCU100_PA_BASE + 0x0064)
- #define SCU_REG_SLP_APBCLKG (SCU_FTSCU100_PA_BASE + 0x0068)
- #define SCU_REG_SLP_WAKUP_ST (SCU_FTSCU100_PA_BASE + 0x00C0)
- #define SCU_REG_SLP_WAKUP_EN (SCU_FTSCU100_PA_BASE + 0x00C4)
- #define SCU_REG_RTC_TIME1 (SCU_FTSCU100_PA_BASE + 0x0200)
- #define SCU_REG_RTC_TIME2 (SCU_FTSCU100_PA_BASE + 0x0204)
- #define SCU_REG_RTC_ALM1 (SCU_FTSCU100_PA_BASE + 0x0208)
- #define SCU_REG_RTC_ALM2 (SCU_FTSCU100_PA_BASE + 0x020C)
- #define SCU_REG_RTC_CTRL (SCU_FTSCU100_PA_BASE + 0x0210)
- #define SCU_REG_BTUP_STS_RTC_BTUPTS BIT17
- #define SCU_REG_BTUP_STS_PWRBTN_STS BIT16
- #define SCU_REG_BTUP_STS_PMR BIT11
- #define SCU_REG_BTUP_STS_SMR BIT10
- #define SCU_REG_BTUP_STS_WDR BIT9
- #define SCU_REG_BTUP_STS_HWR BIT8
- #define SCU_REG_BTUP_STS_PMR1 BIT7
- #define SCU_REG_BTUP_STS_PMR2 BIT6
- #define SCU_REG_BTUP_CTRL_RTC_BU_EN BIT17
- #define SCU_REG_BTUP_CTRL_PWRBTN_EN BIT16
- #define SCU_REG_BTUP_CTRL_GPO_OUT BIT0
- #define SCU_REG_PWR_CTRL_PWRUP_UPDATE BIT24
- #define SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DDRCK BIT10
- #define SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_NPU BIT9
- #define SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DEFAULT BIT8
- #define SCU_REG_PWR_CTRL_PWRUP_CTRL_MASK
- #define SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DDRCK BIT2
- #define SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_NPU BIT1
- #define SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DEFAULT BIT0
- #define SCU_REG_PWR_CTRL_PWRDN_CTRL_MASK
- #define SCU_REG_PWR_MOD_SELFR_CMD_OFF BIT31
- #define SCU_REG_PWR_MOD_FCS_PLL2_RSTn BIT30
- #define SCU_REG_PWR_MOD_FCS_DLL_RSTn BIT29
- #define SCU_REG_PWR_MOD_FCS_PLL_RSTn BIT28
- #define SCU_REG_PWR_MOD_SW_RST BIT10
- #define SCU_REG_PWR_MOD_FCS BIT6
- #define SCU_REG_PWR_MOD_BUS_SPEED BIT5
- #define SCU_REG_PWR_MOD_SOFTOFF BIT1
- #define SCU_REG_PWR_MOD_SET_SOFTOFF(val) SET_MASKED_BIT(SCU_REG_PWR_MOD, val, 1)
- #define SCU_REG_INT_STS_PWRSTATE_CHG BIT28
- #define SCU_REG_INT_STS_RTC_SEC BIT18

Kneron Confidential

- #define SCU_REG_INT_STS_RTC_PER BIT17
- #define SCU_REG_INT_STS_RTC_ALARM BIT16
- #define SCU_REG_INT_STS_PLL_UPDATE BIT8
- #define SCU_REG_INT_STS_FCS BIT6
- #define SCU_REG_INT_STS_BUSSPEED BIT5
- #define SCU_REG_INT_STS_WAKEUP BIT3
- #define SCU_REG_INT_STS_PWRBTN_RISE BIT1
- #define SCU_REG_INT_STS_PWRBTN_FALL BIT0
- #define SCU_REG_PLL_CTRL_GET_CLKIN_MUX() GET_BITS(SCU_EXTREG_USB_OTG_CTRL, 4, 5)
- #define SCU_REG_PLL_CTRL_SET_CLKIN_MUX(val) SET_MASKED_BITS(SCU_EXTREG_USB_OTG_CTRL, val, 4, 5)
- #define SCU_REG_PLL_CTRL_CLKIN_MUX_BIT_START 4
- #define SCU_REG_PLL_CTRL_CLKIN_MUX_MASK BIT4 | BIT5 | BIT6 | BIT7
- #define SCU_REG_PLL_CTRL_PLLEN BIT0
- #define SCU_REG_PLL_CTRL_SET_PLLEN(val) SET_MASKED_BIT(SCU_REG_DLL_CTRL, val, 0)
- #define SCU_REG_PLL2_CTRL_PLL2EN BIT0
- #define SCU_REG_PLL2_CTRL_SET_PLL2EN(val) SET_MASKED_BIT(SCU_REG_PLL2_CTRL, val, 0)
- #define SCU_REG_DLL_CTRL_DLLEN BIT0
- #define SCU_REG_DLL_CTRL_SET_DLLEN(val) SET_MASKED_BIT(SCU_REG_DLL_CTRL, val, 0)
- #define SCU_REG_PWR_VCCSTS_GET_PWR_READY() GET_BITS(SCU_REG_PWR_VCCSTS, 0, 2)
- #define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DDRCK BIT2
- #define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_NPU BIT1
- #define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DEFAULT BIT0
- #define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_MASK
- #define SCU_REG_AHBCLKG_HCLK_EN_DPI1_HCLK BIT21
- #define SCU_REG_AHBCLKG_HCLK_EN_DPI0_HCLK BIT20
- #define SCU_REG_APBCLKG2_PCLK_EN_MIPI_RX1_PHY_PCLK BIT1
- #define SCU_REG_APBCLKG2_PCLK_EN_MIPI_RX0_PHY_PCLK BIT0
- #define SCU_REG_APBCLKG_PCLK_EN_CSIRX1_PCLK BIT29
- #define SCU_REG_APBCLKG_PCLK_EN_CSIRX0_PCLK BIT28
- #define SCU_REG_APBCLKG_PCLK_EN_GPIO_PCLK BIT20
- #define SCU_REG_APBCLKG_PCLK_EN_PWM_PCLK BIT17
- #define SCU_REG_APBCLKG_PCLK_EN_I2C3_PCLK BIT5
- #define SCU_REG_APBCLKG_PCLK_EN_I2C2_PCLK BIT4
- #define SCU_REG_APBCLKG_PCLK_EN_I2C1_PCLK BIT3
- #define SCU_REG_APBCLKG_PCLK_EN_I2C0_PCLK BIT2
- #define SCU_REG_APBCLKG2 (SCU_FTSCU100_PA_BASE + 0x0064)
- #define SCU_REG_SLP_WAUP_EN_GET_SLP_WAKUP_ST0() GET_BIT(SCU_REG_SLP_WAKUP_ST, 0)
- #define SCU_REG_SLP_WAUP_EN_SET_SLP_WAKUP_ST0(val) SET_MASKED_BIT(SCU_REG_SLP_WAKUP_ST, val, 0)
- #define SCU_REG_SLP_WAUP_EN_GET_SLP_WAKUP_EN0() GET_BIT(SCU_REG_SLP_WAKUP_EN, 0)
- #define SCU_REG_SLP_WAUP_EN_SET_SLP_WAKUP_EN0(val) SET_MASKED_BIT(SCU_REG_SLP_WAKUP_EN, val, 0)

6.63.1 Macro Definition Documentation

Kneron Confidential

6.63.1.1 SCU_REG_AHBCLKG

```
#define SCU_REG_AHBCLKG (SCU_FTSCU100_PA_BASE + 0x0050)
```

6.63.1.2 SCU_REG_AHBCLKG_HCLK_EN_DPI0_HCLK

```
#define SCU_REG_AHBCLKG_HCLK_EN_DPI0_HCLK BIT20
```

6.63.1.3 SCU_REG_AHBCLKG_HCLK_EN_DPI1_HCLK

```
#define SCU_REG_AHBCLKG_HCLK_EN_DPI1_HCLK BIT21
```

6.63.1.4 SCU_REG_APBCLKG

```
#define SCU_REG_APBCLKG (SCU_FTSCU100_PA_BASE + 0x0060)
```

6.63.1.5 SCU_REG_APBCLKG2 [1/2]

```
#define SCU_REG_APBCLKG2 (SCU_FTSCU100_PA_BASE + 0x0064)
```

6.63.1.6 SCU_REG_APBCLKG2 [2/2]

```
#define SCU_REG_APBCLKG2 (SCU_FTSCU100_PA_BASE + 0x0064)
```

6.63.1.7 SCU_REG_APBCLKG2_PCLK_EN_MIPI_RX0_PHY_PCLK

```
#define SCU_REG_APBCLKG2_PCLK_EN_MIPI_RX0_PHY_PCLK BIT0
```

6.63.1.8 SCU_REG_APBCLKG2_PCLK_EN_MIPI_RX1_PHY_PCLK

```
#define SCU_REG_APBCLKG2_PCLK_EN_MIPI_RX1_PHY_PCLK BIT1
```

Kneron Confidential

6.63.1.9 SCU_REG_APBCLKG_PCLK_EN_CSIRX0_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_CSIRX0_PCLK BIT28
```

6.63.1.10 SCU_REG_APBCLKG_PCLK_EN_CSIRX1_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_CSIRX1_PCLK BIT29
```

6.63.1.11 SCU_REG_APBCLKG_PCLK_EN_GPIO_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_GPIO_PCLK BIT20
```

6.63.1.12 SCU_REG_APBCLKG_PCLK_EN_I2C0_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_I2C0_PCLK BIT2
```

6.63.1.13 SCU_REG_APBCLKG_PCLK_EN_I2C1_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_I2C1_PCLK BIT3
```

6.63.1.14 SCU_REG_APBCLKG_PCLK_EN_I2C2_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_I2C2_PCLK BIT4
```

6.63.1.15 SCU_REG_APBCLKG_PCLK_EN_I2C3_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_I2C3_PCLK BIT5
```

6.63.1.16 SCU_REG_APBCLKG_PCLK_EN_PWM_PCLK

```
#define SCU_REG_APBCLKG_PCLK_EN_PWM_PCLK BIT17
```

Kneron Confidential

6.63.1.17 SCU_REG_BTUP_CTRL

```
#define SCU_REG_BTUP_CTRL (SCU_FTSCU100_PA_BASE + 0x0004)
```

6.63.1.18 SCU_REG_BTUP_CTRL_GPO_OUT

```
#define SCU_REG_BTUP_CTRL_GPO_OUT BIT0
```

6.63.1.19 SCU_REG_BTUP_CTRL_PWRBTN_EN

```
#define SCU_REG_BTUP_CTRL_PWRBTN_EN BIT16
```

6.63.1.20 SCU_REG_BTUP_CTRL_RTC_BU_EN

```
#define SCU_REG_BTUP_CTRL_RTC_BU_EN BIT17
```

6.63.1.21 SCU_REG_BTUP_STS

```
#define SCU_REG_BTUP_STS (SCU_FTSCU100_PA_BASE + 0x0000)
```

6.63.1.22 SCU_REG_BTUP_STS_HWR

```
#define SCU_REG_BTUP_STS_HWR BIT8
```

6.63.1.23 SCU_REG_BTUP_STS_PMR

```
#define SCU_REG_BTUP_STS_PMR BIT11
```

6.63.1.24 SCU_REG_BTUP_STS_PMR1

```
#define SCU_REG_BTUP_STS_PMR1 BIT7
```

Kneron Confidential

6.63.1.25 SCU_REG_BTUP_STS_PMR2

```
#define SCU_REG_BTUP_STS_PMR2 BIT6
```

6.63.1.26 SCU_REG_BTUP_STS_PWRBTN_STS

```
#define SCU_REG_BTUP_STS_PWRBTN_STS BIT16
```

6.63.1.27 SCU_REG_BTUP_STS_RTC_BTUPTS

```
#define SCU_REG_BTUP_STS_RTC_BTUPTS BIT17
```

6.63.1.28 SCU_REG_BTUP_STS_SMR

```
#define SCU_REG_BTUP_STS_SMR BIT10
```

6.63.1.29 SCU_REG_BTUP_STS_WDR

```
#define SCU_REG_BTUP_STS_WDR BIT9
```

6.63.1.30 SCU_REG_DLL_CTRL

```
#define SCU_REG_DLL_CTRL (SCU_FTSCU100_PA_BASE + 0x0044)
```

6.63.1.31 SCU_REG_DLL_CTRL_DLLEN

```
#define SCU_REG_DLL_CTRL_DLLEN BIT0
```

6.63.1.32 SCU_REG_DLL_CTRL_SET_DLLEN

```
#define SCU_REG_DLL_CTRL_SET_DLLEN(
             val ) SET_MASKED_BIT(SCU_REG_DLL_CTRL, val, 0)
```

Kneron Confidential

6.63.1.33 SCU_REG_INT_EN

```
#define SCU_REG_INT_EN (SCU_FTSCU100_PA_BASE + 0x0028)
```

6.63.1.34 SCU_REG_INT_STS

```
#define SCU_REG_INT_STS (SCU_FTSCU100_PA_BASE + 0x0024)
```

6.63.1.35 SCU_REG_INT_STS_BUSSPEED

```
#define SCU_REG_INT_STS_BUSSPEED BIT5
```

6.63.1.36 SCU_REG_INT_STS_FCS

```
#define SCU_REG_INT_STS_FCS BIT6
```

6.63.1.37 SCU_REG_INT_STS_PLL_UPDATE

```
#define SCU_REG_INT_STS_PLL_UPDATE BIT8
```

6.63.1.38 SCU_REG_INT_STS_PWRBTN_FALL

```
#define SCU_REG_INT_STS_PWRBTN_FALL BIT0
```

6.63.1.39 SCU_REG_INT_STS_PWRBTN_RISE

```
#define SCU_REG_INT_STS_PWRBTN_RISE BIT1
```

6.63.1.40 SCU_REG_INT_STS_PWRSTATE_CHG

```
#define SCU_REG_INT_STS_PWRSTATE_CHG BIT28
```

Kneron Confidential

6.63.1.41 SCU_REG_INT_STS_RTC_ALARM

```
#define SCU_REG_INT_STS_RTC_ALARM BIT16
```

6.63.1.42 SCU_REG_INT_STS_RTC_PER

```
#define SCU_REG_INT_STS_RTC_PER BIT17
```

6.63.1.43 SCU_REG_INT_STS_RTC_SEC

```
#define SCU_REG_INT_STS_RTC_SEC BIT18
```

6.63.1.44 SCU_REG_INT_STS_WAKEUP

```
#define SCU_REG_INT_STS_WAKEUP BIT3
```

6.63.1.45 SCU_REG_PLL2_CTRL

```
#define SCU_REG_PLL2_CTRL (SCU_FTSCU100_PA_BASE + 0x0040)
```

6.63.1.46 SCU_REG_PLL2_CTRL_PLL2EN

```
#define SCU_REG_PLL2_CTRL_PLL2EN BIT0
```

6.63.1.47 SCU_REG_PLL2_CTRL_SET_PLL2EN

```
#define SCU_REG_PLL2_CTRL_SET_PLL2EN(
              val ) SET_MASKED_BIT(SCU_REG_PLL2_CTRL, val, 0)
```

Kneron Confidential

6.63.1.48 SCU_REG_PLL_CTRL

```
#define SCU_REG_PLL_CTRL (SCU_FTSCU100_PA_BASE + 0x0030)
```

6.63.1.49 SCU_REG_PLL_CTRL_CLKIN_MUX_BIT_START

```
#define SCU_REG_PLL_CTRL_CLKIN_MUX_BIT_START 4
```

6.63.1.50 SCU_REG_PLL_CTRL_CLKIN_MUX_MASK

```
#define SCU_REG_PLL_CTRL_CLKIN_MUX_MASK BIT4 | BIT5 | BIT6 | BIT7
```

6.63.1.51 SCU_REG_PLL_CTRL_GET_CLKIN_MUX

```
#define SCU_REG_PLL_CTRL_GET_CLKIN_MUX( ) GET_BITS(SCU_EXTREG_USB_OTG_CTRL, 4, 5)
```

6.63.1.52 SCU_REG_PLL_CTRL_PLLEN

```
#define SCU_REG_PLL_CTRL_PLLEN BIT0
```

6.63.1.53 SCU_REG_PLL_CTRL_SET_CLKIN_MUX

```
#define SCU_REG_PLL_CTRL_SET_CLKIN_MUX(
            val ) SET_MASKED_BITS(SCU_EXTREG_USB_OTG_CTRL, val, 4, 5)
```

6.63.1.54 SCU_REG_PLL_CTRL_SET_PLLEN

```
#define SCU_REG_PLL_CTRL_SET_PLLEN(
             val ) SET_MASKED_BIT(SCU_REG_DLL_CTRL, val, 0)
```

Kneron Confidential

6.63.1.55 SCU_REG_PWR_CTRL

```
#define SCU_REG_PWR_CTRL (SCU_FTSCU100_PA_BASE + 0x0008)
```

6.63.1.56 SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DDRCK

```
#define SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DDRCK BIT2
```

6.63.1.57 SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DEFAULT

```
#define SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DEFAULT BIT0
```

6.63.1.58 SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_NPU

```
#define SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_NPU BIT1
```

6.63.1.59 SCU_REG_PWR_CTRL_PWRDN_CTRL_MASK

```
#define SCU_REG_PWR_CTRL_PWRDN_CTRL_MASK
```

Value:

```
(SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DDRCK | \
                                                SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_NPU | \
                                                SCU_REG_PWR_CTRL_PWRDN_CTRL_DOMAIN_DEFAULT)
```

6.63.1.60 SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DDRCK

```
#define SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DDRCK BIT10
```

6.63.1.61 SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DEFAULT

```
#define SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DEFAULT BIT8
```

Kneron Confidential

6.63.1.62 SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_NPU

```
#define SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_NPU BIT9
```

6.63.1.63 SCU_REG_PWR_CTRL_PWRUP_CTRL_MASK

```
#define SCU_REG_PWR_CTRL_PWRUP_CTRL_MASK
```

Value:

```
(SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DDRCK | \
                                                SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_NPU | \
                                                SCU_REG_PWR_CTRL_PWRUP_CTRL_DOMAIN_DEFAULT)
```

6.63.1.64 SCU_REG_PWR_CTRL_PWRUP_UPDATE

```
#define SCU_REG_PWR_CTRL_PWRUP_UPDATE BIT24
```

6.63.1.65 SCU_REG_PWR_MOD

```
#define SCU_REG_PWR_MOD (SCU_FTSCU100_PA_BASE + 0x0020)
```

6.63.1.66 SCU_REG_PWR_MOD_BUS_SPEED

```
#define SCU_REG_PWR_MOD_BUS_SPEED BIT5
```

6.63.1.67 SCU_REG_PWR_MOD_FCS

```
#define SCU_REG_PWR_MOD_FCS BIT6
```

6.63.1.68 SCU_REG_PWR_MOD_FCS_DLL_RSTn

```
#define SCU_REG_PWR_MOD_FCS_DLL_RSTn BIT29
```

Kneron Confidential

6.63.1.69 SCU_REG_PWR_MOD_FCS_PLL2_RSTn

```
#define SCU_REG_PWR_MOD_FCS_PLL2_RSTn BIT30
```

6.63.1.70 SCU_REG_PWR_MOD_FCS_PLL_RSTn

```
#define SCU_REG_PWR_MOD_FCS_PLL_RSTn BIT28
```

6.63.1.71 SCU_REG_PWR_MOD_SELFR_CMD_OFF

```
#define SCU_REG_PWR_MOD_SELFR_CMD_OFF BIT31
```

6.63.1.72 SCU_REG_PWR_MOD_SET_SOFTOFF

```
#define SCU_REG_PWR_MOD_SET_SOFTOFF(
             val ) SET_MASKED_BIT(SCU_REG_PWR_MOD, val, 1)
```

6.63.1.73 SCU_REG_PWR_MOD_SOFTOFF

```
#define SCU_REG_PWR_MOD_SOFTOFF BIT1
```

6.63.1.74 SCU_REG_PWR_MOD_SW_RST

```
#define SCU_REG_PWR_MOD_SW_RST BIT10
```

6.63.1.75 SCU_REG_PWR_VCCSTS

```
#define SCU_REG_PWR_VCCSTS (SCU_FTSCU100_PA_BASE + 0x0048)
```

Kneron Confidential

6.63.1.76 SCU_REG_PWR_VCCSTS_GET_PWR_READY

```
#define SCU_REG_PWR_VCCSTS_GET_PWR_READY( ) GET_BITS(SCU_REG_PWR_VCCSTS, 0, 2)
```

6.63.1.77 SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DDRCK

```
#define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DDRCK BIT2
```

6.63.1.78 SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DEFAULT

```
#define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DEFAULT BIT0
```

6.63.1.79 SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_MASK

```
#define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_MASK
```

Value:

```
(SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DDRCK | \
                                                SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_NPU | \
                                                SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_DEFAULT)
```

6.63.1.80 SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_NPU

```
#define SCU_REG_PWR_VCCSTS_PWR_READY_DOMAIN_NPU BIT1
```

6.63.1.81 SCU_REG_RTC_ALM1

```
#define SCU_REG_RTC_ALM1 (SCU_FTSCU100_PA_BASE + 0x0208)
```

6.63.1.82 SCU_REG_RTC_ALM2

```
#define SCU_REG_RTC_ALM2 (SCU_FTSCU100_PA_BASE + 0x020C)
```

Kneron Confidential

6.63.1.83 SCU_REG_RTC_CTRL

```
#define SCU_REG_RTC_CTRL (SCU_FTSCU100_PA_BASE + 0x0210)
```

6.63.1.84 SCU_REG_RTC_TIME1

```
#define SCU_REG_RTC_TIME1 (SCU_FTSCU100_PA_BASE + 0x0200)
```

6.63.1.85 SCU_REG_RTC_TIME2

```
#define SCU_REG_RTC_TIME2 (SCU_FTSCU100_PA_BASE + 0x0204)
```

6.63.1.86 SCU_REG_SLP_AHBCLKG

```
#define SCU_REG_SLP_AHBCLKG (SCU_FTSCU100_PA_BASE + 0x0058)
```

6.63.1.87 SCU_REG_SLP_APBCLKG

```
#define SCU_REG_SLP_APBCLKG (SCU_FTSCU100_PA_BASE + 0x0068)
```

6.63.1.88 SCU_REG_SLP_WAKUP_EN

```
#define SCU_REG_SLP_WAKUP_EN (SCU_FTSCU100_PA_BASE + 0x00C4)
```

6.63.1.89 SCU_REG_SLP_WAKUP_ST

```
#define SCU_REG_SLP_WAKUP_ST (SCU_FTSCU100_PA_BASE + 0x00C0)
```

6.63.1.90 SCU_REG_SLP_WAUP_EN_GET_SLP_WAKUP_EN0

```
#define SCU_REG_SLP_WAUP_EN_GET_SLP_WAKUP_EN0( ) GET_BIT(SCU_REG_SLP_WAKUP_EN, 0)
```

Kneron Confidential

6.63.1.91 SCU_REG_SLP_WAUP_EN_GET_SLP_WAKUP_ST0

```
#define SCU_REG_SLP_WAUP_EN_GET_SLP_WAKUP_ST0( ) GET_BIT(SCU_REG_SLP_WAKUP_ST, 0)
```

6.63.1.92 SCU_REG_SLP_WAUP_EN_SET_SLP_WAKUP_EN0

```
#define SCU_REG_SLP_WAUP_EN_SET_SLP_WAKUP_EN0(
            val ) SET_MASKED_BIT(SCU_REG_SLP_WAKUP_EN, val, 0)
```

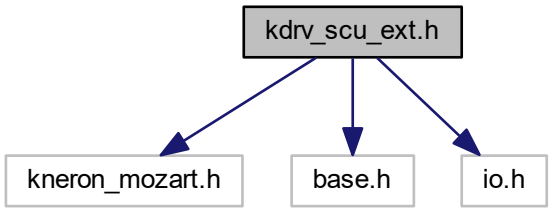
6.63.1.93 SCU_REG_SLP_WAUP_EN_SET_SLP_WAKUP_ST0

```
#define SCU_REG_SLP_WAUP_EN_SET_SLP_WAKUP_ST0(
            val ) SET_MASKED_BIT(SCU_REG_SLP_WAKUP_ST, val, 0)
```

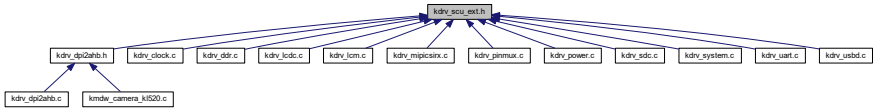
6.64 kdrv_scu_ext.h File Reference

```
#include "kneron_mozart.h"
#include "base.h"
#include "io.h"
```

Include dependency graph for kdrv_scu_ext.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdp520_scu_extreg

Macros

- #define PINMUX_SET(reg, mode) outw(reg, ((inw(reg) & ~(BIT(2) | BIT(1) | BIT(0))) | mode))
- #define PINMUX_CLR(reg) outw(reg, (inw(reg) & ~(BIT(2) | BIT(1) | BIT(0))))
- #define SCU_EXTREG_PLL0_SETTING (SCU_EXTREG_PA_BASE + 0x0000)
- #define SCU_EXTREG_PLL1_SETTING (SCU_EXTREG_PA_BASE + 0x0004)
- #define SCU_EXTREG_PLL2_SETTING (SCU_EXTREG_PA_BASE + 0x0008)
- #define SCU_EXTREG_PLL3_SETTING (SCU_EXTREG_PA_BASE + 0x000C)
- #define SCU_EXTREG_PLL4_SETTING (SCU_EXTREG_PA_BASE + 0x0010)
- #define SCU_EXTREG_PLL5_SETTING (SCU_EXTREG_PA_BASE + 0x003C)
- #define SCU_EXTREG_CLK_EN0 (SCU_EXTREG_PA_BASE + 0x0014)
- #define SCU_EXTREG_CLK_EN1 (SCU_EXTREG_PA_BASE + 0x0018)
- #define SCU_EXTREG_CLK_EN2 (SCU_EXTREG_PA_BASE + 0x001C)
- #define SCU_EXTREG_CLK_MUX_SEL (SCU_EXTREG_PA_BASE + 0x0020)
- #define SCU_EXTREG_CLK_DIV0 (SCU_EXTREG_PA_BASE + 0x0024)
- #define SCU_EXTREG_CLK_DIV1 (SCU_EXTREG_PA_BASE + 0x0028)
- #define SCU_EXTREG_CLK_DIV2 (SCU_EXTREG_PA_BASE + 0x002C)
- #define SCU_EXTREG_CLK_DIV3 (SCU_EXTREG_PA_BASE + 0x0030)
- #define SCU_EXTREG_CLK_DIV4 (SCU_EXTREG_PA_BASE + 0x0034)
- #define SCU_EXTREG_CLK_DIV5 (SCU_EXTREG_PA_BASE + 0x0038)
- #define SCU_EXTREG_SWRST_MASK0 (SCU_EXTREG_PA_BASE + 0x0040)
- #define SCU_EXTREG_SWRST_MASK1 (SCU_EXTREG_PA_BASE + 0x0044)
- #define SCU_EXTREG_SWRST_MASK2 (SCU_EXTREG_PA_BASE + 0x0048)
- #define SCU_EXTREG_SWRST (SCU_EXTREG_PA_BASE + 0x004C)
- #define SCU_EXTREG_CM4_NCPU_CTRL (SCU_EXTREG_PA_BASE + 0x0068)
- #define SCU_EXTREG_DDR_CTRL (SCU_EXTREG_PA_BASE + 0x0080)
- #define SCU_EXTREG_USB_OTG_CTRL (SCU_EXTREG_PA_BASE + 0x008C)
- #define SCU_EXTREG_CSIRX_CTRL0 (SCU_EXTREG_PA_BASE + 0x0090)
- #define SCU_EXTREG_CSIRX_CTRL1 (SCU_EXTREG_PA_BASE + 0x0094)
- #define SCU_EXTREG_DPI2AHB_CTRL (SCU_EXTREG_PA_BASE + 0x009C)
- #define SCU_EXTREG_BONDING_OPTION (SCU_EXTREG_PA_BASE + 0x00A0)
- #define SCU_EXTREG_SD_CTRL (SCU_EXTREG_PA_BASE + 0x00A4)
- #define SCU_EXTREG_ADC_CTRL (SCU_EXTREG_PA_BASE + 0x00A8)
- #define SCU_EXTREG_MISC (SCU_EXTREG_PA_BASE + 0x00B0)
- #define SCU_EXTREG_CLK_DIV6 (SCU_EXTREG_PA_BASE + 0x00D0)
- #define SCU_EXTREG_CLK_DIV7 (SCU_EXTREG_PA_BASE + 0x00D4)
- #define SCU_EXTREG_SPI_CS_N (SCU_EXTREG_PA_BASE + 0x0100)
- #define SCU_EXTREG_SPI_CLK (SCU_EXTREG_PA_BASE + 0x0104)
- #define SCU_EXTREG_SPI_DO (SCU_EXTREG_PA_BASE + 0x0108)
- #define SCU_EXTREG_SPI_DI (SCU_EXTREG_PA_BASE + 0x010C)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL (SCU_EXTREG_PA_BASE + 0x0110)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL (SCU_EXTREG_PA_BASE + 0x0114)
- #define SCU_EXTREG_SWJ_TRST_IOCTRL (SCU_EXTREG_PA_BASE + 0x0118)
- #define SCU_EXTREG_SWJ_TDI_IOCTRL (SCU_EXTREG_PA_BASE + 0x011C)
- #define SCU_EXTREG_SWJ_SWDITMS_IOCTRL (SCU_EXTREG_PA_BASE + 0x0120)
- #define SCU_EXTREG_SWJ_SWCLKTCK_IOCTRL (SCU_EXTREG_PA_BASE + 0x0124)
- #define SCU_EXTREG_SWJ_TDO_IOCTRL (SCU_EXTREG_PA_BASE + 0x0128)
- #define SCU_EXTREG_LC_PCLK_IOCTRL (SCU_EXTREG_PA_BASE + 0x012C)
- #define SCU_EXTREG_LC_VS_IOCTRL (SCU_EXTREG_PA_BASE + 0x0130)
- #define SCU_EXTREG_LC_HS_IOCTRL (SCU_EXTREG_PA_BASE + 0x0134)

Kneron Confidential

- #define SCU_EXTREG_LC_DE_IOCTRL (SCU_EXTREG_PA_BASE + 0x0138)
- #define SCU_EXTREG_LC_DATA0_IOCTRL (SCU_EXTREG_PA_BASE + 0x013C)
- #define SCU_EXTREG_LC_DATA1_IOCTRL (SCU_EXTREG_PA_BASE + 0x0140)
- #define SCU_EXTREG_LC_DATA2_IOCTRL (SCU_EXTREG_PA_BASE + 0x0144)
- #define SCU_EXTREG_LC_DATA3_IOCTRL (SCU_EXTREG_PA_BASE + 0x0148)
- #define SCU_EXTREG_LC_DATA4_IOCTRL (SCU_EXTREG_PA_BASE + 0x014C)
- #define SCU_EXTREG_LC_DATA5_IOCTRL (SCU_EXTREG_PA_BASE + 0x0150)
- #define SCU_EXTREG_LC_DATA6_IOCTRL (SCU_EXTREG_PA_BASE + 0x0154)
- #define SCU_EXTREG_LC_DATA7_IOCTRL (SCU_EXTREG_PA_BASE + 0x0158)
- #define SCU_EXTREG_LC_DATA8_IOCTRL (SCU_EXTREG_PA_BASE + 0x015C)
- #define SCU_EXTREG_LC_DATA9_IOCTRL (SCU_EXTREG_PA_BASE + 0x0160)
- #define SCU_EXTREG_LC_DATA10_IOCTRL (SCU_EXTREG_PA_BASE + 0x0164)
- #define SCU_EXTREG_LC_DATA11_IOCTRL (SCU_EXTREG_PA_BASE + 0x0168)
- #define SCU_EXTREG_LC_DATA12_IOCTRL (SCU_EXTREG_PA_BASE + 0x016C)
- #define SCU_EXTREG_LC_DATA13_IOCTRL (SCU_EXTREG_PA_BASE + 0x0170)
- #define SCU_EXTREG_LC_DATA14_IOCTRL (SCU_EXTREG_PA_BASE + 0x0174)
- #define SCU_EXTREG_LC_DATA15_IOCTRL (SCU_EXTREG_PA_BASE + 0x0178)
- #define SCU_EXTREG_SD_CLK_IOCTRL (SCU_EXTREG_PA_BASE + 0x017C)
- #define SCU_EXTREG_SD_CMD_IOCTRL (SCU_EXTREG_PA_BASE + 0x0180)
- #define SCU_EXTREG_SD_DATA0_IOCTRL (SCU_EXTREG_PA_BASE + 0x0184)
- #define SCU_EXTREG_SD_DATA1_IOCTRL (SCU_EXTREG_PA_BASE + 0x0188)
- #define SCU_EXTREG_SD_DATA2_IOCTRL (SCU_EXTREG_PA_BASE + 0x018C)
- #define SCU_EXTREG_SD_DATA3_IOCTRL (SCU_EXTREG_PA_BASE + 0x0190)
- #define SCU_EXTREG_UART0_RX_IOCTRL (SCU_EXTREG_PA_BASE + 0x0194)
- #define SCU_EXTREG_UART0_TX_IOCTRL (SCU_EXTREG_PA_BASE + 0x0198)
- #define SCU_EXTREG_I2C0_CLK_IOCTRL (SCU_EXTREG_PA_BASE + 0x019C)
- #define SCU_EXTREG_I2C0_DATA_IOCTRL (SCU_EXTREG_PA_BASE + 0x01A0)
- #define SCU_EXTREG_PWM0_IOCTRL (SCU_EXTREG_PA_BASE + 0x01A4)
- #define SCU_EXTREG_OTG_DRV_VBUS_IOCTRL (SCU_EXTREG_PA_BASE + 0x01A8)
- #define SCU_EXTREG_SPARE0_IOCTRL (SCU_EXTREG_PA_BASE + 0x01B0)
- #define SCU_EXTREG_SPARE1_IOCTRL (SCU_EXTREG_PA_BASE + 0x01B4)
- #define SCU_EXTREG_PLL0_SETTING_GET_ns() GET_BITS(SCU_EXTREG_PLL0_SETTING, 24, 30)
- #define SCU_EXTREG_PLL0_SETTING_GET_ms() GET_BITS(SCU_EXTREG_PLL0_SETTING, 16, 20)
- #define SCU_EXTREG_PLL0_SETTING_GET_cc() GET_BITS(SCU_EXTREG_PLL0_SETTING, 12, 13)
- #define SCU_EXTREG_PLL0_SETTING_GET_f() GET_BITS(SCU_EXTREG_PLL0_SETTING, 8, 9)
- #define SCU_EXTREG_PLL0_SETTING_GET_en() GET_BIT(SCU_EXTREG_PLL0_SETTING, 0)
- #define SCU_EXTREG_PLL0_SETTING_SET_ns() SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 24, 30)
- #define SCU_EXTREG_PLL0_SETTING_SET_ms() SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 16, 20)
- #define SCU_EXTREG_PLL0_SETTING_SET_cc() SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 12, 13)
- #define SCU_EXTREG_PLL0_SETTING_SET_f() SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 8, 9)
- #define SCU_EXTREG_PLL0_SETTING_SET_en() SET_MASKED_BIT(SCU_EXTREG_PLL0_SETTING, 0)
- #define SCU_EXTREG_PLL1_SETTING_GET_ns() GET_BITS(SCU_EXTREG_PLL1_SETTING, 24, 30)
- #define SCU_EXTREG_PLL1_SETTING_GET_ms() GET_BITS(SCU_EXTREG_PLL1_SETTING, 16, 20)
- #define SCU_EXTREG_PLL1_SETTING_GET_cc() GET_BITS(SCU_EXTREG_PLL1_SETTING, 12, 13)
- #define SCU_EXTREG_PLL1_SETTING_GET_f() GET_BITS(SCU_EXTREG_PLL1_SETTING, 8, 9)
- #define SCU_EXTREG_PLL1_SETTING_GET_en() GET_BIT(SCU_EXTREG_PLL1_SETTING, 0)
- #define SCU_EXTREG_PLL1_SETTING_SET_ns(val) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 24, 30)
- #define SCU_EXTREG_PLL1_SETTING_SET_ms(val) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 16, 20)

Kneron Confidential

- #define SCU_EXTREG_PLL1_SETTING_SET_cc(val) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 12, 13)
- #define SCU_EXTREG_PLL1_SETTING_SET_f(val) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 8, 9)
- #define SCU_EXTREG_PLL1_SETTING_SET_en(val) SET_MASKED_BIT(SCU_EXTREG_PLL1_SETTING, val, 0)
- #define SCU_EXTREG_PLL2_SETTING_GET_ns() GET_BITS(SCU_EXTREG_PLL2_SETTING, 24, 30)
- #define SCU_EXTREG_PLL2_SETTING_GET_ms() GET_BITS(SCU_EXTREG_PLL2_SETTING, 16, 20)
- #define SCU_EXTREG_PLL2_SETTING_GET_cc() GET_BITS(SCU_EXTREG_PLL2_SETTING, 12, 13)
- #define SCU_EXTREG_PLL2_SETTING_GET_f() GET_BITS(SCU_EXTREG_PLL2_SETTING, 8, 9)
- #define SCU_EXTREG_PLL2_SETTING_GET_en() GET_BIT(SCU_EXTREG_PLL2_SETTING, 0)
- #define SCU_EXTREG_PLL2_SETTING_SET_ns(val) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 24, 30)
- #define SCU_EXTREG_PLL2_SETTING_SET_ms(val) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 16, 20)
- #define SCU_EXTREG_PLL2_SETTING_SET_cc(val) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 12, 13)
- #define SCU_EXTREG_PLL2_SETTING_SET_f(val) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 8, 9)
- #define SCU_EXTREG_PLL2_SETTING_SET_en(val) SET_MASKED_BIT(SCU_EXTREG_PLL2_SETTING, val, 0)
- #define SCU_EXTREG_PLL3_SETTING_ps_MASK (BIT24|BIT25|BIT26|BIT27|BIT28)
- #define SCU_EXTREG_PLL3_SETTING_ns_MASK (BIT12|BIT13|BIT14|BIT15|BIT16|BIT17|BIT18|BI↩ T19|BIT20)
- #define SCU_EXTREG_PLL3_SETTING_ms_MASK (BIT8|BIT9|BIT10)
- #define SCU_EXTREG_PLL3_SETTING_is_MASK (BIT4|BIT5|BIT6)
- #define SCU_EXTREG_PLL3_SETTING_rs_MASK (BIT2|BIT3)
- #define SCU_EXTREG_PLL3_SETTING_ps_START 24
- #define SCU_EXTREG_PLL3_SETTING_ns_START 12
- #define SCU_EXTREG_PLL3_SETTING_ms_START 8
- #define SCU_EXTREG_PLL3_SETTING_is_START 4
- #define SCU_EXTREG_PLL3_SETTING_rs_START 2
- #define SCU_EXTREG_PLL3_SETTING_GET_ps() GET_BITS(SCU_EXTREG_PLL3_SETTING, 24, 28)
- #define SCU_EXTREG_PLL3_SETTING_GET_ns() GET_BITS(SCU_EXTREG_PLL3_SETTING, 12, 20)
- #define SCU_EXTREG_PLL3_SETTING_GET_ms() GET_BITS(SCU_EXTREG_PLL3_SETTING, 8, 10)
- #define SCU_EXTREG_PLL3_SETTING_GET_is() GET_BITS(SCU_EXTREG_PLL3_SETTING, 4, 6)
- #define SCU_EXTREG_PLL3_SETTING_GET_rs() GET_BITS(SCU_EXTREG_PLL3_SETTING, 2, 3)
- #define SCU_EXTREG_PLL3_SETTING_GET_en() GET_BIT(SCU_EXTREG_PLL3_SETTING, 0)
- #define SCU_EXTREG_PLL3_SETTING_SET_ps(val) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 24, 28)
- #define SCU_EXTREG_PLL3_SETTING_SET_ns(val) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 12, 20)
- #define SCU_EXTREG_PLL3_SETTING_SET_ms(val) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 8, 10)
- #define SCU_EXTREG_PLL3_SETTING_SET_is(val) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 4, 6)
- #define SCU_EXTREG_PLL3_SETTING_SET_rs(val) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 2, 3)
- #define SCU_EXTREG_PLL3_SETTING_SET_en(val) SET_MASKED_BIT(SCU_EXTREG_PLL3_SETTING, val, 0)
- #define SCU_EXTREG_PLL4_SETTING_GET_ps() GET_BITS(SCU_EXTREG_PLL4_SETTING, 24, 28)
- #define SCU_EXTREG_PLL4_SETTING_GET_ns() GET_BITS(SCU_EXTREG_PLL4_SETTING, 12, 20)
- #define SCU_EXTREG_PLL4_SETTING_GET_ms() GET_BITS(SCU_EXTREG_PLL4_SETTING, 8, 10)
- #define SCU_EXTREG_PLL4_SETTING_GET_is() GET_BITS(SCU_EXTREG_PLL4_SETTING, 4, 6)
- #define SCU_EXTREG_PLL4_SETTING_GET_rs() GET_BITS(SCU_EXTREG_PLL4_SETTING, 2, 3)

Kneron Confidential

- #define SCU_EXTREG_PLL4_SETTING_GET_en() GET_BIT(SCU_EXTREG_PLL4_SETTING, 0)
- #define SCU_EXTREG_PLL4_SETTING_SET_ps(val) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 24, 28)
- #define SCU_EXTREG_PLL4_SETTING_SET_ns(val) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 12, 20)
- #define SCU_EXTREG_PLL4_SETTING_SET_ms(val) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 8, 10)
- #define SCU_EXTREG_PLL4_SETTING_SET_is(val) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 4, 6)
- #define SCU_EXTREG_PLL4_SETTING_SET_rs(val) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 2, 3)
- #define SCU_EXTREG_PLL4_SETTING_SET_en(val) SET_MASKED_BIT(SCU_EXTREG_PLL4_SETTING, val, 0)
- #define SCU_EXTREG_CLK_EN0_GET_ncpu_traceclk() GET_BIT(SCU_EXTREG_CLK_EN0, 24)
- #define SCU_EXTREG_CLK_EN0_GET_scpu_traceclk() GET_BIT(SCU_EXTREG_CLK_EN0, 23)
- #define SCU_EXTREG_CLK_EN0_GET_ncpu_fclk_src() GET_BIT(SCU_EXTREG_CLK_EN0, 22)
- #define SCU_EXTREG_CLK_EN0_GET_pll4_fref_pll0() GET_BIT(SCU_EXTREG_CLK_EN0, 12)
- #define SCU_EXTREG_CLK_EN0_GET_pll5_out2() GET_BIT(SCU_EXTREG_CLK_EN0, 9)
- #define SCU_EXTREG_CLK_EN0_GET_pll5_out1() GET_BIT(SCU_EXTREG_CLK_EN0, 8)
- #define SCU_EXTREG_CLK_EN0_GET_pll4_out1() GET_BIT(SCU_EXTREG_CLK_EN0, 6)
- #define SCU_EXTREG_CLK_EN0_GET_pll3_out2() GET_BIT(SCU_EXTREG_CLK_EN0, 5)
- #define SCU_EXTREG_CLK_EN0_GET_pll3_out1() GET_BIT(SCU_EXTREG_CLK_EN0, 4)
- #define SCU_EXTREG_CLK_EN0_GET_pll2_out() GET_BIT(SCU_EXTREG_CLK_EN0, 2)
- #define SCU_EXTREG_CLK_EN0_GET_pll1_out() GET_BIT(SCU_EXTREG_CLK_EN0, 1)
- #define SCU_EXTREG_CLK_EN0_SET_ncpu_traceclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 24)
- #define SCU_EXTREG_CLK_EN0_SET_scpu_traceclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 23)
- #define SCU_EXTREG_CLK_EN0_SET_ncpu_fclk_src(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 22)
- #define SCU_EXTREG_CLK_EN0_SET_pll4_fref_pll0(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 12)
- #define SCU_EXTREG_CLK_EN0_SET_pll5_out2(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 9)
- #define SCU_EXTREG_CLK_EN0_SET_pll5_out1(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 8)
- #define SCU_EXTREG_CLK_EN0_SET_pll4_out1(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 6)
- #define SCU_EXTREG_CLK_EN0_SET_pll3_out2(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 5)
- #define SCU_EXTREG_CLK_EN0_SET_pll3_out1(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 4)
- #define SCU_EXTREG_CLK_EN0_SET_pll2_out(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 2)
- #define SCU_EXTREG_CLK_EN0_SET_pll1_out(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 1)
- #define SCU_EXTREG_CLK_EN1_GET_spi_clk() GET_BIT(SCU_EXTREG_CLK_EN1, 24)
- #define SCU_EXTREG_CLK_EN1_GET_npu() GET_BIT(SCU_EXTREG_CLK_EN1, 23)
- #define SCU_EXTREG_CLK_EN1_GET_adcclk() GET_BIT(SCU_EXTREG_CLK_EN1, 22)
- #define SCU_EXTREG_CLK_EN1_GET_wdt_extclk() GET_BIT(SCU_EXTREG_CLK_EN1, 21)
- #define SCU_EXTREG_CLK_EN1_GET_sdclk() GET_BIT(SCU_EXTREG_CLK_EN1, 20)
- #define SCU_EXTREG_CLK_EN1_GET_TxHsPllRefClk() GET_BIT(SCU_EXTREG_CLK_EN1, 16)
- #define SCU_EXTREG_CLK_EN1_GET_tx_EscClk() GET_BIT(SCU_EXTREG_CLK_EN1, 14)
- #define SCU_EXTREG_CLK_EN1_GET_csitx_dsi() GET_BIT(SCU_EXTREG_CLK_EN1, 13)
- #define SCU_EXTREG_CLK_EN1_GET_csitx_csi() GET_BIT(SCU_EXTREG_CLK_EN1, 12)
- #define SCU_EXTREG_CLK_EN1_GET_csirx1_TxEscClk() GET_BIT(SCU_EXTREG_CLK_EN1, 10)

Kneron Confidential

- #define SCU_EXTREG_CLK_EN1_GET_csirx1_csi() GET_BIT(SCU_EXTREG_CLK_EN1, 9)
- #define SCU_EXTREG_CLK_EN1_GET_csirx1_vc0() GET_BIT(SCU_EXTREG_CLK_EN1, 8)
- #define SCU_EXTREG_CLK_EN1_GET_csirx0_TxEscClk() GET_BIT(SCU_EXTREG_CLK_EN1, 6)
- #define SCU_EXTREG_CLK_EN1_GET_csirx0_csi() GET_BIT(SCU_EXTREG_CLK_EN1, 5)
- #define SCU_EXTREG_CLK_EN1_GET_csirx0_vc0() GET_BIT(SCU_EXTREG_CLK_EN1, 4)
- #define SCU_EXTREG_CLK_EN1_GET_LC_SCALER() GET_BIT(SCU_EXTREG_CLK_EN1, 1)
- #define SCU_EXTREG_CLK_EN1_GET_LC_CLK() GET_BIT(SCU_EXTREG_CLK_EN1, 0)
- #define SCU_EXTREG_CLK_EN1_SET_spi_clk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 24)
- #define SCU_EXTREG_CLK_EN1_SET_npu(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 23)
- #define SCU_EXTREG_CLK_EN1_SET_adcclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 22)
- #define SCU_EXTREG_CLK_EN1_SET_wdt_extclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 21)
- #define SCU_EXTREG_CLK_EN1_SET_sdclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 20)
- #define SCU_EXTREG_CLK_EN1_SET_TxHsPllRefClk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 16)
- #define SCU_EXTREG_CLK_EN1_SET_tx_EscClk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 14)
- #define SCU_EXTREG_CLK_EN1_SET_csitx_dsi(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 13)
- #define SCU_EXTREG_CLK_EN1_SET_csitx_csi(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 12)
- #define SCU_EXTREG_CLK_EN1_SET_csirx1_TxEscClk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 10)
- #define SCU_EXTREG_CLK_EN1_SET_csirx1_csi(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 9)
- #define SCU_EXTREG_CLK_EN1_SET_csirx1_vc0(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 8)
- #define SCU_EXTREG_CLK_EN1_SET_csirx0_TxEscClk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 6)
- #define SCU_EXTREG_CLK_EN1_SET_csirx0_csi(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 5)
- #define SCU_EXTREG_CLK_EN1_SET_csirx0_vc0(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 4)
- #define SCU_EXTREG_CLK_EN1_SET_LC_SCALER(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 1)
- #define SCU_EXTREG_CLK_EN1_SET_LC_CLK(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 0)
- #define SCU_EXTREG_CLK_EN2_GET_tmr1_extclk3() GET_BIT(SCU_EXTREG_PLL2_SETTING, 29)
- #define SCU_EXTREG_CLK_EN2_GET_tmr1_extclk2() GET_BIT(SCU_EXTREG_PLL2_SETTING, 28)
- #define SCU_EXTREG_CLK_EN2_GET_tmr1_extclk1() GET_BIT(SCU_EXTREG_PLL2_SETTING, 27)
- #define SCU_EXTREG_CLK_EN2_GET_tmr0_extclk3() GET_BIT(SCU_EXTREG_PLL2_SETTING, 26)
- #define SCU_EXTREG_CLK_EN2_GET_tmr0_extclk2() GET_BIT(SCU_EXTREG_PLL2_SETTING, 25)
- #define SCU_EXTREG_CLK_EN2_GET_tmr0_extclk1() GET_BIT(SCU_EXTREG_PLL2_SETTING, 24)
- #define SCU_EXTREG_CLK_EN2_GET_pwm_extclk6() GET_BIT(SCU_EXTREG_PLL2_SETTING, 21)
- #define SCU_EXTREG_CLK_EN2_GET_pwm_extclk5() GET_BIT(SCU_EXTREG_PLL2_SETTING, 20)
- #define SCU_EXTREG_CLK_EN2_GET_pwm_extclk4() GET_BIT(SCU_EXTREG_PLL2_SETTING, 19)
- #define SCU_EXTREG_CLK_EN2_GET_pwm_extclk3() GET_BIT(SCU_EXTREG_PLL2_SETTING, 18)
- #define SCU_EXTREG_CLK_EN2_GET_pwm_extclk2() GET_BIT(SCU_EXTREG_PLL2_SETTING, 17)
- #define SCU_EXTREG_CLK_EN2_GET_pwm_extclk1() GET_BIT(SCU_EXTREG_PLL2_SETTING, 16)
- #define SCU_EXTREG_CLK_EN2_GET_uart1_3_fref() GET_BIT(SCU_EXTREG_PLL2_SETTING, 15)
- #define SCU_EXTREG_CLK_EN2_GET_uart1_2_fref() GET_BIT(SCU_EXTREG_PLL2_SETTING, 14)
- #define SCU_EXTREG_CLK_EN2_GET_uart1_1_fref() GET_BIT(SCU_EXTREG_PLL2_SETTING, 13)
- #define SCU_EXTREG_CLK_EN2_GET_uart1_0_fref() GET_BIT(SCU_EXTREG_PLL2_SETTING, 12)

Kneron Confidential

- #define SCU_EXTREG_CLK_EN2_GET_uart0_fref() GET_BIT(SCU_EXTREG_PLL2_SETTING, 8)
- #define SCU_EXTREG_CLK_EN2_GET_ssp1_1_sspclk() GET_BIT(SCU_EXTREG_PLL2_SETTING, 6)
- #define SCU_EXTREG_CLK_EN2_GET_ssp1_0_sspclk() GET_BIT(SCU_EXTREG_PLL2_SETTING, 4)
- #define SCU_EXTREG_CLK_EN2_GET_ssp0_1_sspclk() GET_BIT(SCU_EXTREG_PLL2_SETTING, 2)
- #define SCU_EXTREG_CLK_EN2_GET_ssp0_0_sspclk() GET_BIT(SCU_EXTREG_PLL2_SETTING, 0)
- #define SCU_EXTREG_CLK_EN2_SET_tmr1_extclk3(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 29)
- #define SCU_EXTREG_CLK_EN2_SET_tmr1_extclk2(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 28)
- #define SCU_EXTREG_CLK_EN2_SET_tmr1_extclk1(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 27)
- #define SCU_EXTREG_CLK_EN2_SET_tmr0_extclk3(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 26)
- #define SCU_EXTREG_CLK_EN2_SET_tmr0_extclk2(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 25)
- #define SCU_EXTREG_CLK_EN2_SET_tmr0_extclk1(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 24)
- #define SCU_EXTREG_CLK_EN2_SET_pwm_extclk6(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 21)
- #define SCU_EXTREG_CLK_EN2_SET_pwm_extclk5(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 20)
- #define SCU_EXTREG_CLK_EN2_SET_pwm_extclk4(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 19)
- #define SCU_EXTREG_CLK_EN2_SET_pwm_extclk3(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 18)
- #define SCU_EXTREG_CLK_EN2_SET_pwm_extclk2(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 17)
- #define SCU_EXTREG_CLK_EN2_SET_pwm_extclk1(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 16)
- #define SCU_EXTREG_CLK_EN2_SET_uart1_3_fref(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 15)
- #define SCU_EXTREG_CLK_EN2_SET_uart1_2_fref(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 14)
- #define SCU_EXTREG_CLK_EN2_SET_uart1_1_fref(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 13)
- #define SCU_EXTREG_CLK_EN2_SET_uart1_0_fref(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 12)
- #define SCU_EXTREG_CLK_EN2_SET_uart0_fref(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 8)
- #define SCU_EXTREG_CLK_EN2_SET_ssp1_1_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 6)
- #define SCU_EXTREG_CLK_EN2_SET_ssp1_0_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 4)
- #define SCU_EXTREG_CLK_EN2_SET_ssp0_1_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 2)
- #define SCU_EXTREG_CLK_EN2_SET_ssp0_0_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 0)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_ncpu_traceclk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 14)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_scpu_traceclk_src() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 13)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_csirx1_clk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 12)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_npu_clk() GET_BITS(SCU_EXTREG_CLK_MUX_SEL, 8, 9)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_pll4_fref() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 6)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_uart_0_irda_uclk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 4)

Kneron Confidential

- #define SCU_EXTREG_CLK_MUX_SEL_GET_ssp1_1_sspclk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 3)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_ssp1_0_sspclk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 2)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_ssp0_1_sspclk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 1)
- #define SCU_EXTREG_CLK_MUX_SEL_GET_ssp0_0_sspclk() GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 0)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_ncpu_traceclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 14)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_scpu_traceclk_src(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 13)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_csirx1_clk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 12)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_npu_clk(val) SET_MASKED_BITS(SCU_EXTREG_CLK_MUX_SEL, val, 8, 9)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_pll4_fref(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 6)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_uart_0_irda_uclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 4)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_ssp1_1_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 3)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_ssp1_0_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 2)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_ssp0_1_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 1)
- #define SCU_EXTREG_CLK_MUX_SEL_SET_ssp0_0_sspclk(val) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 0)
- #define SCU_EXTREG_CLK_DIV0_GET_ncpu_fclk() GET_BITS(SCU_EXTREG_CLK_DIV0, 28, 31)
- #define SCU_EXTREG_CLK_DIV0_GET_sdclk2x() GET_BITS(SCU_EXTREG_CLK_DIV0, 24, 27)
- #define SCU_EXTREG_CLK_DIV0_GET_spi_clk() GET_BITS(SCU_EXTREG_CLK_DIV0, 20, 23)
- #define SCU_EXTREG_CLK_DIV0_GET_pll4_fref_pll0() GET_BITS(SCU_EXTREG_CLK_DIV0, 8, 12)
- #define SCU_EXTREG_CLK_DIV0_SET_ncpu_fclk(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 28, 31)
- #define SCU_EXTREG_CLK_DIV0_SET_sdclk2x(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 24, 27)
- #define SCU_EXTREG_CLK_DIV0_SET_spi_clk(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 20, 23)
- #define SCU_EXTREG_CLK_DIV0_SET_pll4_fref_pll0(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 8, 12)
- #define SCU_EXTREG_CLK_DIV1_GET_csirx0_TxEscClk() GET_BITS(SCU_EXTREG_CLK_DIV1, 28, 31)
- #define SCU_EXTREG_CLK_DIV1_GET_csirx0_csi() GET_BITS(SCU_EXTREG_CLK_DIV1, 20, 24)
- #define SCU_EXTREG_CLK_DIV1_GET_csirx0_vc0() GET_BITS(SCU_EXTREG_CLK_DIV1, 12, 16)
- #define SCU_EXTREG_CLK_DIV1_GET_LC_CLK() GET_BITS(SCU_EXTREG_CLK_DIV1, 4, 8)
- #define SCU_EXTREG_CLK_DIV1_GET_LC_SCALER_CLK() GET_BITS(SCU_EXTREG_CLK_DIV1, 0, 3)
- #define SCU_EXTREG_CLK_DIV1_SET_csirx0_TxEscClk(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 28, 31)
- #define SCU_EXTREG_CLK_DIV1_SET_csirx0_csi(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 20, 24)
- #define SCU_EXTREG_CLK_DIV1_SET_csirx0_vc0(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 12, 16)
- #define SCU_EXTREG_CLK_DIV1_SET_LC_CLK(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 4, 8)
- #define SCU_EXTREG_CLK_DIV1_SET_LC_SCALER_CLK(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 0, 3)
- #define SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll5() GET_BITS(SCU_EXTREG_CLK_DIV2, 16, 18)

Kneron Confidential

- #define SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll4() GET_BITS(SCU_EXTREG_CLK_DIV2, 12, 14)
- #define SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll0() GET_BITS(SCU_EXTREG_CLK_DIV2, 8, 10)
- #define SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll5(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV2, val, 16, 18)
- #define SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll4(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV2, val, 12, 14)
- #define SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll0(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV2, val, 8, 10)
- #define SCU_EXTREG_CLK_DIV3_GET_ssp0_1_sspclk_slv() GET_BITS(SCU_EXTREG_CLK_DIV3, 24, 26)
- #define SCU_EXTREG_CLK_DIV3_GET_ssp0_1_sspclk_mst() GET_BITS(SCU_EXTREG_CLK_DIV3, 16, 21)
- #define SCU_EXTREG_CLK_DIV3_GET_ssp0_0_sspclk_slv() GET_BITS(SCU_EXTREG_CLK_DIV3, 8, 10)
- #define SCU_EXTREG_CLK_DIV3_GET_ssp0_0_sspclk_mst() GET_BITS(SCU_EXTREG_CLK_DIV3, 0, 5)
- #define SCU_EXTREG_CLK_DIV3_SET_ssp0_1_sspclk_slv(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 24, 26)
- #define SCU_EXTREG_CLK_DIV3_SET_ssp0_1_sspclk_mst(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 16, 21)
- #define SCU_EXTREG_CLK_DIV3_SET_ssp0_0_sspclk_slv(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 8, 10)
- #define SCU_EXTREG_CLK_DIV3_SET_ssp0_0_sspclk_mst(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 0, 5)
- #define SCU_EXTREG_CLK_DIV4_GET_ssp1_1_sspclk_slv() GET_BITS(SCU_EXTREG_CLK_DIV4, 24, 26)
- #define SCU_EXTREG_CLK_DIV4_GET_ssp1_1_sspclk_mst() GET_BITS(SCU_EXTREG_CLK_DIV4, 16, 21)
- #define SCU_EXTREG_CLK_DIV4_GET_ssp1_0_sspclk_slv() GET_BITS(SCU_EXTREG_CLK_DIV4, 8, 10)
- #define SCU_EXTREG_CLK_DIV4_GET_ssp1_0_sspclk_mst() GET_BITS(SCU_EXTREG_CLK_DIV4, 0, 5)
- #define SCU_EXTREG_CLK_DIV4_SET_ssp1_1_sspclk_slv(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 24, 26)
- #define SCU_EXTREG_CLK_DIV4_SET_ssp1_1_sspclk_mst(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 16, 21)
- #define SCU_EXTREG_CLK_DIV4_SET_ssp1_0_sspclk_slv(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 8, 10)
- #define SCU_EXTREG_CLK_DIV4_SET_ssp1_0_sspclk_mst(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 0, 5)
- #define SCU_EXTREG_PLL5_SETTING_ps_MASK (BIT24|BIT25|BIT26|BIT27|BIT28)
- #define SCU_EXTREG_PLL5_SETTING_ns_MASK (BIT12|BIT13|BIT14|BIT15|BIT16|BIT17|BIT18|BIT19|BIT20)
- #define SCU_EXTREG_PLL5_SETTING_ms_MASK (BIT8|BIT9|BIT10)
- #define SCU_EXTREG_PLL5_SETTING_is_MASK (BIT4|BIT5|BIT6)
- #define SCU_EXTREG_PLL5_SETTING_rs_MASK (BIT2|BIT3)
- #define SCU_EXTREG_PLL5_SETTING_ps_START 24
- #define SCU_EXTREG_PLL5_SETTING_ns_START 12
- #define SCU_EXTREG_PLL5_SETTING_ms_START 8
- #define SCU_EXTREG_PLL5_SETTING_is_START 4
- #define SCU_EXTREG_PLL5_SETTING_rs_START 2
- #define SCU_EXTREG_PLL5_SETTING_GET_ps() GET_BITS(SCU_EXTREG_PLL5_SETTING, 24, 28)
- #define SCU_EXTREG_PLL5_SETTING_GET_ns() GET_BITS(SCU_EXTREG_PLL5_SETTING, 12, 20)
- #define SCU_EXTREG_PLL5_SETTING_GET_ms() GET_BITS(SCU_EXTREG_PLL5_SETTING, 8, 10)
- #define SCU_EXTREG_PLL5_SETTING_GET_is() GET_BITS(SCU_EXTREG_PLL5_SETTING, 4, 6)
- #define SCU_EXTREG_PLL5_SETTING_GET_rs() GET_BITS(SCU_EXTREG_PLL5_SETTING, 2, 3)

Kneron Confidential

- #define SCU_EXTREG_PLL5_SETTING_GET_en() GET_BIT(SCU_EXTREG_PLL5_SETTING, 0)
- #define SCU_EXTREG_PLL5_SETTING_SET_ps(val) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 24, 28)
- #define SCU_EXTREG_PLL5_SETTING_SET_ns(val) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 12, 20)
- #define SCU_EXTREG_PLL5_SETTING_SET_ms(val) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 8, 10)
- #define SCU_EXTREG_PLL5_SETTING_SET_is(val) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 4, 6)
- #define SCU_EXTREG_PLL5_SETTING_SET_rs(val) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 2, 3)
- #define SCU_EXTREG_PLL5_SETTING_SET_en(val) SET_MASKED_BIT(SCU_EXTREG_PLL5_SETTING, val, 0)
- #define SCU_EXTREG_SWRST_MASK0_GET_cpu_resetreq_n() GET_BIT(SCU_EXTREG_SWRST_MASK0, 0)
- #define SCU_EXTREG_SWRST_MASK0_SET_cpu_resetreq_n(val) SET_MASKED_BIT(SCU_EXTREG_SWRST_MASK0, val, 0)
- #define SCU_EXTREG_SWRST_MASK1_lcm_reset_n BIT23
- #define SCU_EXTREG_SWRST_MASK1_AResetn_u_FTLCDC210 BIT20
- #define SCU_EXTREG_SWRST_MASK1_PRESETn_u_FTLCDC210 BIT19
- #define SCU_EXTREG_SWRST_MASK1_TV_RSTn_FTLCDC210 BIT18
- #define SCU_EXTREG_SWRST_MASK1_LC_SCALER_RSTn_FTLCDC210 BIT17
- #define SCU_EXTREG_SWRST_MASK1_LC_RSTn_FTLCDC210 BIT16
- #define SCU_EXTREG_SWRST_MASK1_GET_lcm_reset_n() GET_BIT(SCU_EXTREG_PLL5_SETTING, 23)
- #define SCU_EXTREG_SWRST_MASK1_SET_lcm_reset_n(val) SET_MASKED_BIT(SCU_EXTREG_PLL5_SETTING, val, 23)
- #define SCU_EXTREG_SWRST_GET_NPU_resetn() GET_BIT(SCU_EXTREG_SWRST, 2)
- #define SCU_EXTREG_SWRST_GET_PD_NPU_resetn() GET_BIT(SCU_EXTREG_SWRST, 1)
- #define SCU_EXTREG_SWRST_GET_LCDC_resetn() GET_BIT(SCU_EXTREG_SWRST, 0)
- #define SCU_EXTREG_SWRST_SET_NPU_resetn(val) SET_MASKED_BIT(SCU_EXTREG_SWRST, val, 2)
- #define SCU_EXTREG_SWRST_SET_PD_NPU_resetn(val) SET_MASKED_BIT(SCU_EXTREG_SWRST, val, 1)
- #define SCU_EXTREG_SWRST_SET_LCDC_resetn(val) SET_MASKED_BIT(SCU_EXTREG_SWRST, val, 0)
- #define SCU_EXTREG_CM4_NCPU_CTRL_GET_wakeup() GET_BIT(SCU_EXTREG_CM4_NCPU_CTRL, 12)
- #define SCU_EXTREG_CM4_NCPU_CTRL_SET_wakeup(val) SET_MASKED_BIT(SCU_EXTREG_CM4_NCPU_CTRL, val, 12)
- #define SCU_EXTREG_DDR_CTRL_Dphy_resetn BIT29
- #define SCU_EXTREG_DDR_CTRL_wakeup BIT28
- #define SCU_EXTREG_DDR_CTRL_SELFBIAS BIT15
- #define SCU_EXTREG_DDR_CTRL_SET_Dphy_resetn(val) SET_MASKED_BIT(SCU_EXTREG_DDR_CTRL, val, 29)
- #define SCU_EXTREG_DDR_CTRL_SET_wakeup(val) SET_MASKED_BIT(SCU_EXTREG_DDR_CTRL, val, 28)
- #define SCU_EXTREG_DDR_CTRL_SET_SELFBIAS(val) SET_MASKED_BIT(SCU_EXTREG_DDR_CTRL, val, 15)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_EXTCTRL_SUSPENDM() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 8)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_u_iddig() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 7)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_wakeup() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 6)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_l1_wakeup() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 5)

Kneron Confidential

- #define SCU_EXTREG_USB_OTG_CTRL_GET_OSCOUTEN() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 4)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_PLLALIV() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 3)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_XTLSEL() GET_BITS(SCU_EXTREG_USB_OTG_CTRL, 1, 2)
- #define SCU_EXTREG_USB_OTG_CTRL_GET_OUTCLKSEL() GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 0)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_EXTCTRL_SUSPENDM(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG
val, 8)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_u_iddig(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 7)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_wakeup(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 6)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_l1_wakeup(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 5)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_OSCOUTEN(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 4)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_PLLALIV(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 3)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_XTLSEL(val) SET_MASKED_BITS(SCU_EXTREG_USB_OTG_CTRL, val, 1, 2)
- #define SCU_EXTREG_USB_OTG_CTRL_SET_OUTCLKSEL(val) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 0)
- #define SCU_EXTREG_CSIRX_CTRL0_apb_rst_n BIT28
- #define SCU_EXTREG_CSIRX_CTRL0_pwr_rst_n BIT25
- #define SCU_EXTREG_CSIRX_CTRL0_sys_rst_n BIT24
- #define SCU_EXTREG_CSIRX_CTRL0_ClkLnEn BIT1
- #define SCU_EXTREG_CSIRX_CTRL0_Enable BIT0
- #define SCU_EXTREG_CSIRX_CTRL0_GET_apb_rst_n() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 28)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_pwr_rst_n() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 25)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_sys_rst_n() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 24)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP1() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 17)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN1() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 16)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP0() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 15)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN0() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 14)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKP() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 13)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKN() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 12)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_ClkLnEn() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 1)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_Enable() GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 0)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_apb_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 28)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_pwr_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 25)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_sys_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 24)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DP1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 17)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DN1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 16)

Kneron Confidential

- #define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DP0(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 15)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DN0(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 14)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_CKP(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 13)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_CKN(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 12)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_ClkLnEn(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 1)
- #define SCU_EXTREG_CSIRX_CTRL0_SET_Enable(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 0)
- #define SCU_EXTREG_CSIRX_CTRL1_apb_rst_n BIT28
- #define SCU_EXTREG_CSIRX_CTRL1_pwr_rst_n BIT25
- #define SCU_EXTREG_CSIRX_CTRL1_sys_rst_n BIT24
- #define SCU_EXTREG_CSIRX_CTRL1_ClkLnEn BIT1
- #define SCU_EXTREG_CSIRX_CTRL1_Enable BIT0
- #define SCU_EXTREG_CSIRX_CTRL1_GET_apb_rst_n() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 28)
- #define SCU_EXTREG_CSIRX_CTRL1_GET_pwr_rst_n() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 25)
- #define SCU_EXTREG_CSIRX_CTRL1_GET_sys_rst_n() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 24)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP1_1() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 17)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN1_1() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 16)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP0_1() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 15)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN0_1() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 14)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKP_1() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 13)
- #define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKN_1() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 12)
- #define SCU_EXTREG_CSIRX_CTRL1_GET_ClkLnEn() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 1)
- #define SCU_EXTREG_CSIRX_CTRL1_GET_Enable() GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 0)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_apb_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 28)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_pwr_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 25)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_sys_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 24)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DP1_1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 17)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DN1_1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 16)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DP0_1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 15)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DN0_1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 14)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_CKP_1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 13)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_CKN_1(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 12)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_ClkLnEn(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 1)
- #define SCU_EXTREG_CSIRX_CTRL1_SET_Enable(val) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 0)

Kneron Confidential

- #define SCU_EXTREG_DPI2AHB_CTRL_rst_n_1 BIT7
- #define SCU_EXTREG_DPI2AHB_CTRL_apb_rst_n_1 BIT6
- #define SCU_EXTREG_DPI2AHB_CTRL_sys_rst_n_1 BIT5
- #define SCU_EXTREG_DPI2AHB_CTRL_pwr_rst_n_1 BIT4
- #define SCU_EXTREG_DPI2AHB_CTRL_rst_n BIT3
- #define SCU_EXTREG_DPI2AHB_CTRL_apb_rst_n BIT2
- #define SCU_EXTREG_DPI2AHB_CTRL_sys_rst_n BIT1
- #define SCU_EXTREG_DPI2AHB_CTRL_pwr_rst_n BIT0
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_rst_n_1() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 7)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_apb_rst_n_1() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 6)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_sys_rst_n_1() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 5)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_pwr_rst_n_1() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 4)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_rst_n() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 3)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_apb_rst_n() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 2)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_sys_rst_n() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 1)
- #define SCU_EXTREG_DPI2AHB_CTRL_GET_pwr_rst_n() GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 0)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_rst_n_1(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 7)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_apb_rst_n_1(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 6)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_sys_rst_n_1(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 5)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_pwr_rst_n_1(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 4)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 3)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_apb_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 2)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_sys_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 1)
- #define SCU_EXTREG_DPI2AHB_CTRL_SET_pwr_rst_n(val) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 0)
- #define SCU_EXTREG_BONDING_OPTION_GET_pkg_214() GET_BIT(SCU_EXTREG_BONDING_OPTION, 0)
- #define SCU_EXTREG_SD_CTRL_GET_io_sd_cd(val) GET_BIT(SCU_EXTREG_SD_CTRL, 0)
- #define SCU_EXTREG_SD_CTRL_GET_io_sd_wp(val) GET_BIT(SCU_EXTREG_SD_CTRL, 1)
- #define SCU_EXTREG_SD_CTRL_SET_io_sd_cd(val) SET_MASKED_BIT(SCU_EXTREG_SD_CTRL, 0)
- #define SCU_EXTREG_SD_CTRL_SET_io_sd_wp(val) SET_MASKED_BIT(SCU_EXTREG_SD_CTRL, 1)
- #define SCU_EXTREG_MISC_GET_smr_por_n() GET_BITS(SCU_EXTREG_MISC, 4, 6)
- #define SCU_EXTREG_MISC_GET_lcm_cken() GET_BIT(SCU_EXTREG_MISC, 12)
- #define SCU_EXTREG_MISC_SET_lcm_cken(val) SET_MASKED_BIT(SCU_EXTREG_MISC, val, 12)
- #define SCU_EXTREG_MISC_SET_DPI_MUX_0_sel(val) SET_MASKED_BIT(SCU_EXTREG_MISC, val, 8)
- #define SCU_EXTREG_MISC_SET_DPI_MUX_0_get() GET_BIT(SCU_EXTREG_MISC, 8)
- #define SCU_EXTREG_MISC_SET_DPI_MUX_1_sel(val) SET_MASKED_BIT(SCU_EXTREG_MISC, val, 9)
- #define SCU_EXTREG_MISC_SET_DPI_MUX_1_get() GET_BIT(SCU_EXTREG_MISC, 9)
- #define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DDRCK BIT6
- #define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_NPU BIT5
- #define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DEFAULT BIT4
- #define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_MASK
- #define SCU_EXTREG_CLK_DIV6_GET_uart1_3_fref() GET_BITS(SCU_EXTREG_CLK_DIV6, 24, 26)

Kneron Confidential

- #define SCU_EXTREG_CLK_DIV6_GET_uart1_2_fref() GET_BITS(SCU_EXTREG_CLK_DIV6, 20, 22)
- #define SCU_EXTREG_CLK_DIV6_GET_uart1_1_fref() GET_BITS(SCU_EXTREG_CLK_DIV6, 16, 18)
- #define SCU_EXTREG_CLK_DIV6_GET_uart1_0_fref() GET_BITS(SCU_EXTREG_CLK_DIV6, 12, 14)
- #define SCU_EXTREG_CLK_DIV6_GET_uart0_fref() GET_BITS(SCU_EXTREG_CLK_DIV6, 8, 10)
- #define SCU_EXTREG_CLK_DIV6_GET_uart0_fir_fref() GET_BITS(SCU_EXTREG_CLK_DIV6, 0, 4)
- #define SCU_EXTREG_CLK_DIV6_SET_uart1_3_fref(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 24, 26)
- #define SCU_EXTREG_CLK_DIV6_SET_uart1_2_fref(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 20, 22)
- #define SCU_EXTREG_CLK_DIV6_SET_uart1_1_fref(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 16, 18)
- #define SCU_EXTREG_CLK_DIV6_SET_uart1_0_fref(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 12, 14)
- #define SCU_EXTREG_CLK_DIV6_SET_uart0_fref(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 8, 10)
- #define SCU_EXTREG_CLK_DIV6_SET_uart0_fir_fref(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 0, 4)
- #define SCU_EXTREG_CLK_DIV7_csirx1_TxEscClk_pll3_MASK (BIT16|BIT17|BIT18|BIT19)
- #define SCU_EXTREG_CLK_DIV7_csi1_csi_pll3_MASK (BIT8|BIT9|BIT10|BIT11|BIT12)
- #define SCU_EXTREG_CLK_DIV7_csi1_vc0_pll3_MASK (BIT0|BIT1|BIT2|BIT3|BIT4)
- #define SCU_EXTREG_CLK_DIV7_csirx1_TxEscClk_pll3_START 16
- #define SCU_EXTREG_CLK_DIV7_csi1_csi_pll3_MASK_START 8
- #define SCU_EXTREG_CLK_DIV7_csi1_vc0_pll3_MASK_START 0
- #define SCU_EXTREG_CLK_DIV7_GET_ncpu_traceclk_div() GET_BITS(SCU_EXTREG_CLK_DIV7, 23, 25)
- #define SCU_EXTREG_CLK_DIV7_GET_scpu_traceclk_div() GET_BITS(SCU_EXTREG_CLK_DIV7, 20, 22)
- #define SCU_EXTREG_CLK_DIV7_GET_csirx1_TxEscClk_pll3() GET_BITS(SCU_EXTREG_CLK_DIV7, 16, 19)
- #define SCU_EXTREG_CLK_DIV7_GET_csi1_csi_pll3() GET_BITS(SCU_EXTREG_CLK_DIV7, 8, 12)
- #define SCU_EXTREG_CLK_DIV7_GET_csi1_vc0_pll3() GET_BITS(SCU_EXTREG_CLK_DIV7, 0, 4)
- #define SCU_EXTREG_CLK_DIV7_SET_ncpu_traceclk_div(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 23, 25)
- #define SCU_EXTREG_CLK_DIV7_SET_scpu_traceclk_div(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 20, 22)
- #define SCU_EXTREG_CLK_DIV7_SET_csirx1_TxEscClk_pll3(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 16, 19)
- #define SCU_EXTREG_CLK_DIV7_SET_csi1_csi_pll3(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 8, 12)
- #define SCU_EXTREG_CLK_DIV7_SET_csi1_vc0_pll3(val) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 0, 4)
- #define SCU_EXTREG_SPI_CS_N_GET_dcsr() GET_BITS(SCU_EXTREG_SPI_CS_N, 5, 8)
- #define SCU_EXTREG_SPI_CS_N_GET_pd() GET_BIT(SCU_EXTREG_SPI_CS_N, 4)
- #define SCU_EXTREG_SPI_CS_N_GET_pu() GET_BIT(SCU_EXTREG_SPI_CS_N, 3)
- #define SCU_EXTREG_SPI_CS_N_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_SPI_CS_N, val, 5, 8)
- #define SCU_EXTREG_SPI_CS_N_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_SPI_CS_N, 4)
- #define SCU_EXTREG_SPI_CS_N_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_SPI_CS_N, 3)
- #define SCU_EXTREG_SPI_CLK_GET_dcsr() GET_BITS(SCU_EXTREG_SPI_CLK, 5, 8)
- #define SCU_EXTREG_SPI_CLK_GET_pd() GET_BIT(SCU_EXTREG_SPI_CLK, 4)
- #define SCU_EXTREG_SPI_CLK_GET_pu() GET_BIT(SCU_EXTREG_SPI_CLK, 3)
- #define SCU_EXTREG_SPI_CLK_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_SPI_CLK, val, 5, 8)
- #define SCU_EXTREG_SPI_CLK_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_SPI_CLK, 4)
- #define SCU_EXTREG_SPI_CLK_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_SPI_CLK, 3)
- #define SCU_EXTREG_SPI_DO_GET_dcsr() GET_BITS(SCU_EXTREG_SPI_DO, 5, 8)

Kneron Confidential

- #define SCU_EXTREG_SPI_DO_GET_pd() GET_BIT(SCU_EXTREG_SPI_DO, 4)
- #define SCU_EXTREG_SPI_DO_GET_pu() GET_BIT(SCU_EXTREG_SPI_DO, 3)
- #define SCU_EXTREG_SPI_DO_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_SPI_DO, val, 5, 8)
- #define SCU_EXTREG_SPI_DO_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_SPI_DO, 4)
- #define SCU_EXTREG_SPI_DO_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_SPI_DO, 3)
- #define SCU_EXTREG_SPI_DI_GET_dcsr() GET_BITS(SCU_EXTREG_SPI_DI, 5, 8)
- #define SCU_EXTREG_SPI_DI_GET_pd() GET_BIT(SCU_EXTREG_SPI_DI, 4)
- #define SCU_EXTREG_SPI_DI_GET_pu() GET_BIT(SCU_EXTREG_SPI_DI, 3)
- #define SCU_EXTREG_SPI_DI_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_SPI_DI, val, 5, 8)
- #define SCU_EXTREG_SPI_DI_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_SPI_DI, 4)
- #define SCU_EXTREG_SPI_DI_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_SPI_DI, 3)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL_GET_dcsr() GET_BITS(SCU_EXTREG_SPI_WP_N_IOCTRL, 5, 8)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL_GET_pd() GET_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 4)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL_GET_pu() GET_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 3)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_SPI_WP_N_IOCTRL, val, 5, 8)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 4)
- #define SCU_EXTREG_SPI_WP_N_IOCTRL_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 3)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_pd() GET_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 4)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_dcsr() GET_BITS(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 5, 8)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_pu() GET_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 3)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_SPI_HOLD_N_IOCTRL, val, 5, 8)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 4)
- #define SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 3)
- #define SCU_EXTREG_LC_DATA6_GET_dcsr() GET_BITS(SCU_EXTREG_LC_DATA6_IOCTRL, 5, 8)
- #define SCU_EXTREG_LC_DATA6_GET_pd() GET_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 4)
- #define SCU_EXTREG_LC_DATA6_GET_pu() GET_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 3)
- #define SCU_EXTREG_LC_DATA6_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_LC_DATA6_IOCTRL, val, 5, 8)
- #define SCU_EXTREG_LC_DATA6_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 4)
- #define SCU_EXTREG_LC_DATA6_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 3)
- #define SCU_EXTREG_LC_DATA7_GET_dcsr() GET_BITS(SCU_EXTREG_LC_DATA7_IOCTRL, 5, 8)
- #define SCU_EXTREG_LC_DATA7_GET_pd() GET_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 4)
- #define SCU_EXTREG_LC_DATA7_GET_pu() GET_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 3)
- #define SCU_EXTREG_LC_DATA7_SET_dcsr(val) SET_MASKED_BITS(SCU_EXTREG_LC_DATA7_IOCTRL, val, 5, 8)
- #define SCU_EXTREG_LC_DATA7_SET_pd(val) SET_MASKED_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 4)
- #define SCU_EXTREG_LC_DATA7_SET_pu(val) SET_MASKED_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 3)
- #define SCU_EXTREG_PRINT(__desc, __regtype, __symbol) DSG("%s %s %x", __desc, #__symbol, SC↩ U_EXTREG_##__regtype##_GET##_##__symbol())
- #define SCU_EXTREG_PRINT2(__label) {DSG("%s=%x", #__label, inw(__label))};
- #define SCU_EXTREG_PRINT_ALL()

Kneron Confidential

Enumerations

- enum {
  PINMUX_MODE0 = 0, PINMUX_MODE1, PINMUX_MODE2, PINMUX_MODE3,
  PINMUX_MODE4, PINMUX_MODE5, PINMUX_MODE6, PINMUX_MODE7 }

6.64.1 Macro Definition Documentation

6.64.1.1 PINMUX_CLR

```
#define PINMUX_CLR(
              reg ) outw(reg, (inw(reg) & ~(BIT(2) | BIT(1) | BIT(0))))
```

6.64.1.2 PINMUX_SET

```
#define PINMUX_SET(
              reg,
              mode ) outw(reg, ((inw(reg) & ~(BIT(2) | BIT(1) | BIT(0))) | mode))
```

6.64.1.3 SCU_EXTREG_ADC_CTRL

```
#define SCU_EXTREG_ADC_CTRL (SCU_EXTREG_PA_BASE + 0x00A8)
```

6.64.1.4 SCU_EXTREG_BONDING_OPTION

```
#define SCU_EXTREG_BONDING_OPTION (SCU_EXTREG_PA_BASE + 0x00A0)
```

6.64.1.5 SCU_EXTREG_BONDING_OPTION_GET_pkg_214

```
#define SCU_EXTREG_BONDING_OPTION_GET_pkg_214( ) GET_BIT(SCU_EXTREG_BONDING_OPTION, 0)
```

6.64.1.6 SCU_EXTREG_CLK_DIV0

```
#define SCU_EXTREG_CLK_DIV0 (SCU_EXTREG_PA_BASE + 0x0024)
```

Kneron Confidential

6.64.1.7 SCU_EXTREG_CLK_DIV0_GET_ncpu_fclk

```
#define SCU_EXTREG_CLK_DIV0_GET_ncpu_fclk( ) GET_BITS(SCU_EXTREG_CLK_DIV0, 28, 31)
```

6.64.1.8 SCU_EXTREG_CLK_DIV0_GET_pll4_fref_pll0

```
#define SCU_EXTREG_CLK_DIV0_GET_pll4_fref_pll0( ) GET_BITS(SCU_EXTREG_CLK_DIV0, 8, 12)
```

6.64.1.9 SCU_EXTREG_CLK_DIV0_GET_sdclk2x

```
#define SCU_EXTREG_CLK_DIV0_GET_sdclk2x( ) GET_BITS(SCU_EXTREG_CLK_DIV0, 24, 27)
```

6.64.1.10 SCU_EXTREG_CLK_DIV0_GET_spi_clk

```
#define SCU_EXTREG_CLK_DIV0_GET_spi_clk( ) GET_BITS(SCU_EXTREG_CLK_DIV0, 20, 23)
```

6.64.1.11 SCU_EXTREG_CLK_DIV0_SET_ncpu_fclk

```
#define SCU_EXTREG_CLK_DIV0_SET_ncpu_fclk(
            val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 28, 31)
```

6.64.1.12 SCU_EXTREG_CLK_DIV0_SET_pll4_fref_pll0

```
#define SCU_EXTREG_CLK_DIV0_SET_pll4_fref_pll0(
            val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 8, 12)
```

6.64.1.13 SCU_EXTREG_CLK_DIV0_SET_sdclk2x

```
#define SCU_EXTREG_CLK_DIV0_SET_sdclk2x(
            val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 24, 27)
```

Kneron Confidential

6.64.1.14 SCU_EXTREG_CLK_DIV0_SET_spi_clk

```
#define SCU_EXTREG_CLK_DIV0_SET_spi_clk(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV0, val, 20, 23)
```

6.64.1.15 SCU_EXTREG_CLK_DIV1

```
#define SCU_EXTREG_CLK_DIV1 (SCU_EXTREG_PA_BASE + 0x0028)
```

6.64.1.16 SCU_EXTREG_CLK_DIV1_GET_csirx0_csi

```
#define SCU_EXTREG_CLK_DIV1_GET_csirx0_csi( ) GET_BITS(SCU_EXTREG_CLK_DIV1, 20, 24)
```

6.64.1.17 SCU_EXTREG_CLK_DIV1_GET_csirx0_TxEscClk

```
#define SCU_EXTREG_CLK_DIV1_GET_csirx0_TxEscClk( ) GET_BITS(SCU_EXTREG_CLK_DIV1, 28, 31)
```

6.64.1.18 SCU_EXTREG_CLK_DIV1_GET_csirx0_vc0

```
#define SCU_EXTREG_CLK_DIV1_GET_csirx0_vc0( ) GET_BITS(SCU_EXTREG_CLK_DIV1, 12, 16)
```

6.64.1.19 SCU_EXTREG_CLK_DIV1_GET_LC_CLK

```
#define SCU_EXTREG_CLK_DIV1_GET_LC_CLK( ) GET_BITS(SCU_EXTREG_CLK_DIV1, 4, 8)
```

6.64.1.20 SCU_EXTREG_CLK_DIV1_GET_LC_SCALER_CLK

```
#define SCU_EXTREG_CLK_DIV1_GET_LC_SCALER_CLK( ) GET_BITS(SCU_EXTREG_CLK_DIV1, 0, 3)
```

Kneron Confidential

6.64.1.21 SCU_EXTREG_CLK_DIV1_SET_csirx0_csi

```
#define SCU_EXTREG_CLK_DIV1_SET_csirx0_csi(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 20, 24)
```

6.64.1.22 SCU_EXTREG_CLK_DIV1_SET_csirx0_TxEscClk

```
#define SCU_EXTREG_CLK_DIV1_SET_csirx0_TxEscClk(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 28, 31)
```

6.64.1.23 SCU_EXTREG_CLK_DIV1_SET_csirx0_vc0

```
#define SCU_EXTREG_CLK_DIV1_SET_csirx0_vc0(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 12, 16)
```

6.64.1.24 SCU_EXTREG_CLK_DIV1_SET_LC_CLK

```
#define SCU_EXTREG_CLK_DIV1_SET_LC_CLK(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 4, 8)
```

6.64.1.25 SCU_EXTREG_CLK_DIV1_SET_LC_SCALER_CLK

```
#define SCU_EXTREG_CLK_DIV1_SET_LC_SCALER_CLK(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV1, val, 0, 3)
```

6.64.1.26 SCU_EXTREG_CLK_DIV2

```
#define SCU_EXTREG_CLK_DIV2 (SCU_EXTREG_PA_BASE + 0x002C)
```

6.64.1.27 SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll0

```
#define SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll0( ) GET_BITS(SCU_EXTREG_CLK_DIV2, 8, 10)
```

Kneron Confidential

6.64.1.28 SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll4

```
#define SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll4( ) GET_BITS(SCU_EXTREG_CLK_DIV2, 12, 14)
```

6.64.1.29 SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll5

```
#define SCU_EXTREG_CLK_DIV2_GET_npu_clk_pll5( ) GET_BITS(SCU_EXTREG_CLK_DIV2, 16, 18)
```

6.64.1.30 SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll0

```
#define SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll0(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV2, val, 8, 10)
```

6.64.1.31 SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll4

```
#define SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll4(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV2, val, 12, 14)
```

6.64.1.32 SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll5

```
#define SCU_EXTREG_CLK_DIV2_SET_npu_clk_pll5(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV2, val, 16, 18)
```

6.64.1.33 SCU_EXTREG_CLK_DIV3

```
#define SCU_EXTREG_CLK_DIV3 (SCU_EXTREG_PA_BASE + 0x0030)
```

6.64.1.34 SCU_EXTREG_CLK_DIV3_GET_ssp0_0_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV3_GET_ssp0_0_sspclk_mst( ) GET_BITS(SCU_EXTREG_CLK_DIV3, 0, 5)
```

Kneron Confidential

6.64.1.35 SCU_EXTREG_CLK_DIV3_GET_ssp0_0_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV3_GET_ssp0_0_sspclk_slv( ) GET_BITS(SCU_EXTREG_CLK_DIV3, 8, 10)
```

6.64.1.36 SCU_EXTREG_CLK_DIV3_GET_ssp0_1_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV3_GET_ssp0_1_sspclk_mst( ) GET_BITS(SCU_EXTREG_CLK_DIV3, 16, 21)
```

6.64.1.37 SCU_EXTREG_CLK_DIV3_GET_ssp0_1_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV3_GET_ssp0_1_sspclk_slv( ) GET_BITS(SCU_EXTREG_CLK_DIV3, 24, 26)
```

6.64.1.38 SCU_EXTREG_CLK_DIV3_SET_ssp0_0_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV3_SET_ssp0_0_sspclk_mst(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 0, 5)
```

6.64.1.39 SCU_EXTREG_CLK_DIV3_SET_ssp0_0_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV3_SET_ssp0_0_sspclk_slv(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 8, 10)
```

6.64.1.40 SCU_EXTREG_CLK_DIV3_SET_ssp0_1_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV3_SET_ssp0_1_sspclk_mst(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 16, 21)
```

6.64.1.41 SCU_EXTREG_CLK_DIV3_SET_ssp0_1_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV3_SET_ssp0_1_sspclk_slv(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV3, val, 24, 26)
```

Kneron Confidential

6.64.1.42 SCU_EXTREG_CLK_DIV4

```
#define SCU_EXTREG_CLK_DIV4 (SCU_EXTREG_PA_BASE + 0x0034)
```

6.64.1.43 SCU_EXTREG_CLK_DIV4_GET_ssp1_0_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV4_GET_ssp1_0_sspclk_mst( ) GET_BITS(SCU_EXTREG_CLK_DIV4, 0, 5)
```

6.64.1.44 SCU_EXTREG_CLK_DIV4_GET_ssp1_0_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV4_GET_ssp1_0_sspclk_slv( ) GET_BITS(SCU_EXTREG_CLK_DIV4, 8, 10)
```

6.64.1.45 SCU_EXTREG_CLK_DIV4_GET_ssp1_1_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV4_GET_ssp1_1_sspclk_mst( ) GET_BITS(SCU_EXTREG_CLK_DIV4, 16, 21)
```

6.64.1.46 SCU_EXTREG_CLK_DIV4_GET_ssp1_1_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV4_GET_ssp1_1_sspclk_slv( ) GET_BITS(SCU_EXTREG_CLK_DIV4, 24, 26)
```

6.64.1.47 SCU_EXTREG_CLK_DIV4_SET_ssp1_0_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV4_SET_ssp1_0_sspclk_mst(
            val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 0, 5)
```

6.64.1.48 SCU_EXTREG_CLK_DIV4_SET_ssp1_0_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV4_SET_ssp1_0_sspclk_slv(
             val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 8, 10)
```

Kneron Confidential

6.64.1.49 SCU_EXTREG_CLK_DIV4_SET_ssp1_1_sspclk_mst

```
#define SCU_EXTREG_CLK_DIV4_SET_ssp1_1_sspclk_mst(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 16, 21)
```

6.64.1.50 SCU_EXTREG_CLK_DIV4_SET_ssp1_1_sspclk_slv

```
#define SCU_EXTREG_CLK_DIV4_SET_ssp1_1_sspclk_slv(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV4, val, 24, 26)
```

6.64.1.51 SCU_EXTREG_CLK_DIV5

```
#define SCU_EXTREG_CLK_DIV5 (SCU_EXTREG_PA_BASE + 0x0038)
```

6.64.1.52 SCU_EXTREG_CLK_DIV6

```
#define SCU_EXTREG_CLK_DIV6 (SCU_EXTREG_PA_BASE + 0x00D0)
```

6.64.1.53 SCU_EXTREG_CLK_DIV6_GET_uart0_fir_fref

```
#define SCU_EXTREG_CLK_DIV6_GET_uart0_fir_fref( ) GET_BITS(SCU_EXTREG_CLK_DIV6, 0, 4)
```

6.64.1.54 SCU_EXTREG_CLK_DIV6_GET_uart0_fref

```
#define SCU_EXTREG_CLK_DIV6_GET_uart0_fref( ) GET_BITS(SCU_EXTREG_CLK_DIV6, 8, 10)
```

6.64.1.55 SCU_EXTREG_CLK_DIV6_GET_uart1_0_fref

```
#define SCU_EXTREG_CLK_DIV6_GET_uart1_0_fref( ) GET_BITS(SCU_EXTREG_CLK_DIV6, 12, 14)
```

Kneron Confidential

6.64.1.56 SCU_EXTREG_CLK_DIV6_GET_uart1_1_fref

```
#define SCU_EXTREG_CLK_DIV6_GET_uart1_1_fref( ) GET_BITS(SCU_EXTREG_CLK_DIV6, 16, 18)
```

6.64.1.57 SCU_EXTREG_CLK_DIV6_GET_uart1_2_fref

```
#define SCU_EXTREG_CLK_DIV6_GET_uart1_2_fref( ) GET_BITS(SCU_EXTREG_CLK_DIV6, 20, 22)
```

6.64.1.58 SCU_EXTREG_CLK_DIV6_GET_uart1_3_fref

```
#define SCU_EXTREG_CLK_DIV6_GET_uart1_3_fref( ) GET_BITS(SCU_EXTREG_CLK_DIV6, 24, 26)
```

6.64.1.59 SCU_EXTREG_CLK_DIV6_SET_uart0_fir_fref

```
#define SCU_EXTREG_CLK_DIV6_SET_uart0_fir_fref(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 0, 4)
```

6.64.1.60 SCU_EXTREG_CLK_DIV6_SET_uart0_fref

```
#define SCU_EXTREG_CLK_DIV6_SET_uart0_fref(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 8, 10)
```

6.64.1.61 SCU_EXTREG_CLK_DIV6_SET_uart1_0_fref

```
#define SCU_EXTREG_CLK_DIV6_SET_uart1_0_fref(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 12, 14)
```

6.64.1.62 SCU_EXTREG_CLK_DIV6_SET_uart1_1_fref

```
#define SCU_EXTREG_CLK_DIV6_SET_uart1_1_fref(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 16, 18)
```

Kneron Confidential

6.64.1.63 SCU_EXTREG_CLK_DIV6_SET_uart1_2_fref

```
#define SCU_EXTREG_CLK_DIV6_SET_uart1_2_fref(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 20, 22)
```

6.64.1.64 SCU_EXTREG_CLK_DIV6_SET_uart1_3_fref

```
#define SCU_EXTREG_CLK_DIV6_SET_uart1_3_fref(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV6, val, 24, 26)
```

6.64.1.65 SCU_EXTREG_CLK_DIV7

```
#define SCU_EXTREG_CLK_DIV7 (SCU_EXTREG_PA_BASE + 0x00D4)
```

6.64.1.66 SCU_EXTREG_CLK_DIV7_csi1_csi_pll3_MASK

```
#define SCU_EXTREG_CLK_DIV7_csi1_csi_pll3_MASK (BIT8|BIT9|BIT10|BIT11|BIT12)
```

6.64.1.67 SCU_EXTREG_CLK_DIV7_csi1_csi_pll3_MASK_START

```
#define SCU_EXTREG_CLK_DIV7_csi1_csi_pll3_MASK_START 8
```

6.64.1.68 SCU_EXTREG_CLK_DIV7_csi1_vc0_pll3_MASK

```
#define SCU_EXTREG_CLK_DIV7_csi1_vc0_pll3_MASK (BIT0|BIT1|BIT2|BIT3|BIT4)
```

6.64.1.69 SCU_EXTREG_CLK_DIV7_csi1_vc0_pll3_MASK_START

```
#define SCU_EXTREG_CLK_DIV7_csi1_vc0_pll3_MASK_START 0
```

Kneron Confidential

6.64.1.70 SCU_EXTREG_CLK_DIV7_csirx1_TxEscClk_pll3_MASK

```
#define SCU_EXTREG_CLK_DIV7_csirx1_TxEscClk_pll3_MASK (BIT16|BIT17|BIT18|BIT19)
```

6.64.1.71 SCU_EXTREG_CLK_DIV7_csirx1_TxEscClk_pll3_START

```
#define SCU_EXTREG_CLK_DIV7_csirx1_TxEscClk_pll3_START 16
```

6.64.1.72 SCU_EXTREG_CLK_DIV7_GET_csi1_csi_pll3

```
#define SCU_EXTREG_CLK_DIV7_GET_csi1_csi_pll3( ) GET_BITS(SCU_EXTREG_CLK_DIV7, 8, 12)
```

6.64.1.73 SCU_EXTREG_CLK_DIV7_GET_csi1_vc0_pll3

```
#define SCU_EXTREG_CLK_DIV7_GET_csi1_vc0_pll3( ) GET_BITS(SCU_EXTREG_CLK_DIV7, 0, 4)
```

6.64.1.74 SCU_EXTREG_CLK_DIV7_GET_csirx1_TxEscClk_pll3

```
#define SCU_EXTREG_CLK_DIV7_GET_csirx1_TxEscClk_pll3( ) GET_BITS(SCU_EXTREG_CLK_DIV7, 16, 19)
```

6.64.1.75 SCU_EXTREG_CLK_DIV7_GET_ncpu_traceclk_div

```
#define SCU_EXTREG_CLK_DIV7_GET_ncpu_traceclk_div( ) GET_BITS(SCU_EXTREG_CLK_DIV7, 23, 25)
```

6.64.1.76 SCU_EXTREG_CLK_DIV7_GET_scpu_traceclk_div

```
#define SCU_EXTREG_CLK_DIV7_GET_scpu_traceclk_div( ) GET_BITS(SCU_EXTREG_CLK_DIV7, 20, 22)
```

6.64.1.77 SCU_EXTREG_CLK_DIV7_SET_csi1_csi_pll3

```
#define SCU_EXTREG_CLK_DIV7_SET_csi1_csi_pll3(
            val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 8, 12)
```

Kneron Confidential

6.64.1.78 SCU_EXTREG_CLK_DIV7_SET_csi1_vc0_pll3

```
#define SCU_EXTREG_CLK_DIV7_SET_csi1_vc0_pll3(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 0, 4)
```

6.64.1.79 SCU_EXTREG_CLK_DIV7_SET_csirx1_TxEscClk_pll3

```
#define SCU_EXTREG_CLK_DIV7_SET_csirx1_TxEscClk_pll3(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 16, 19)
```

6.64.1.80 SCU_EXTREG_CLK_DIV7_SET_ncpu_traceclk_div

```
#define SCU_EXTREG_CLK_DIV7_SET_ncpu_traceclk_div(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 23, 25)
```

6.64.1.81 SCU_EXTREG_CLK_DIV7_SET_scpu_traceclk_div

```
#define SCU_EXTREG_CLK_DIV7_SET_scpu_traceclk_div(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_DIV7, val, 20, 22)
```

6.64.1.82 SCU_EXTREG_CLK_EN0

```
#define SCU_EXTREG_CLK_EN0 (SCU_EXTREG_PA_BASE + 0x0014)
```

6.64.1.83 SCU_EXTREG_CLK_EN0_GET_ncpu_fclk_src

```
#define SCU_EXTREG_CLK_EN0_GET_ncpu_fclk_src( ) GET_BIT(SCU_EXTREG_CLK_EN0, 22)
```

6.64.1.84 SCU_EXTREG_CLK_EN0_GET_ncpu_traceclk

```
#define SCU_EXTREG_CLK_EN0_GET_ncpu_traceclk( ) GET_BIT(SCU_EXTREG_CLK_EN0, 24)
```

Kneron Confidential

6.64.1.85 SCU_EXTREG_CLK_EN0_GET_pll1_out

```
#define SCU_EXTREG_CLK_EN0_GET_pll1_out( ) GET_BIT(SCU_EXTREG_CLK_EN0, 1)
```

6.64.1.86 SCU_EXTREG_CLK_EN0_GET_pll2_out

```
#define SCU_EXTREG_CLK_EN0_GET_pll2_out( ) GET_BIT(SCU_EXTREG_CLK_EN0, 2)
```

6.64.1.87 SCU_EXTREG_CLK_EN0_GET_pll3_out1

```
#define SCU_EXTREG_CLK_EN0_GET_pll3_out1( ) GET_BIT(SCU_EXTREG_CLK_EN0, 4)
```

6.64.1.88 SCU_EXTREG_CLK_EN0_GET_pll3_out2

```
#define SCU_EXTREG_CLK_EN0_GET_pll3_out2( ) GET_BIT(SCU_EXTREG_CLK_EN0, 5)
```

6.64.1.89 SCU_EXTREG_CLK_EN0_GET_pll4_fref_pll0

```
#define SCU_EXTREG_CLK_EN0_GET_pll4_fref_pll0( ) GET_BIT(SCU_EXTREG_CLK_EN0, 12)
```

6.64.1.90 SCU_EXTREG_CLK_EN0_GET_pll4_out1

```
#define SCU_EXTREG_CLK_EN0_GET_pll4_out1( ) GET_BIT(SCU_EXTREG_CLK_EN0, 6)
```

6.64.1.91 SCU_EXTREG_CLK_EN0_GET_pll5_out1

```
#define SCU_EXTREG_CLK_EN0_GET_pll5_out1( ) GET_BIT(SCU_EXTREG_CLK_EN0, 8)
```

6.64.1.92 SCU_EXTREG_CLK_EN0_GET_pll5_out2

```
#define SCU_EXTREG_CLK_EN0_GET_pll5_out2( ) GET_BIT(SCU_EXTREG_CLK_EN0, 9)
```

Kneron Confidential

6.64.1.93 SCU_EXTREG_CLK_EN0_GET_scpu_traceclk

```
#define SCU_EXTREG_CLK_EN0_GET_scpu_traceclk( ) GET_BIT(SCU_EXTREG_CLK_EN0, 23)
```

6.64.1.94 SCU_EXTREG_CLK_EN0_SET_ncpu_fclk_src

```
#define SCU_EXTREG_CLK_EN0_SET_ncpu_fclk_src(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 22)
```

6.64.1.95 SCU_EXTREG_CLK_EN0_SET_ncpu_traceclk

```
#define SCU_EXTREG_CLK_EN0_SET_ncpu_traceclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 24)
```

6.64.1.96 SCU_EXTREG_CLK_EN0_SET_pll1_out

```
#define SCU_EXTREG_CLK_EN0_SET_pll1_out(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 1)
```

6.64.1.97 SCU_EXTREG_CLK_EN0_SET_pll2_out

```
#define SCU_EXTREG_CLK_EN0_SET_pll2_out(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 2)
```

6.64.1.98 SCU_EXTREG_CLK_EN0_SET_pll3_out1

```
#define SCU_EXTREG_CLK_EN0_SET_pll3_out1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 4)
```

6.64.1.99 SCU_EXTREG_CLK_EN0_SET_pll3_out2

```
#define SCU_EXTREG_CLK_EN0_SET_pll3_out2(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 5)
```

Kneron Confidential

6.64.1.100 SCU_EXTREG_CLK_EN0_SET_pll4_fref_pll0

```
#define SCU_EXTREG_CLK_EN0_SET_pll4_fref_pll0(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 12)
```

6.64.1.101 SCU_EXTREG_CLK_EN0_SET_pll4_out1

```
#define SCU_EXTREG_CLK_EN0_SET_pll4_out1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 6)
```

6.64.1.102 SCU_EXTREG_CLK_EN0_SET_pll5_out1

```
#define SCU_EXTREG_CLK_EN0_SET_pll5_out1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 8)
```

6.64.1.103 SCU_EXTREG_CLK_EN0_SET_pll5_out2

```
#define SCU_EXTREG_CLK_EN0_SET_pll5_out2(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 9)
```

6.64.1.104 SCU_EXTREG_CLK_EN0_SET_scpu_traceclk

```
#define SCU_EXTREG_CLK_EN0_SET_scpu_traceclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN0, val, 23)
```

6.64.1.105 SCU_EXTREG_CLK_EN1

```
#define SCU_EXTREG_CLK_EN1 (SCU_EXTREG_PA_BASE + 0x0018)
```

6.64.1.106 SCU_EXTREG_CLK_EN1_GET_adcclk

```
#define SCU_EXTREG_CLK_EN1_GET_adcclk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 22)
```

Kneron Confidential

6.64.1.107 SCU_EXTREG_CLK_EN1_GET_csirx0_csi

```
#define SCU_EXTREG_CLK_EN1_GET_csirx0_csi( ) GET_BIT(SCU_EXTREG_CLK_EN1, 5)
```

6.64.1.108 SCU_EXTREG_CLK_EN1_GET_csirx0_TxEscClk

```
#define SCU_EXTREG_CLK_EN1_GET_csirx0_TxEscClk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 6)
```

6.64.1.109 SCU_EXTREG_CLK_EN1_GET_csirx0_vc0

```
#define SCU_EXTREG_CLK_EN1_GET_csirx0_vc0( ) GET_BIT(SCU_EXTREG_CLK_EN1, 4)
```

6.64.1.110 SCU_EXTREG_CLK_EN1_GET_csirx1_csi

```
#define SCU_EXTREG_CLK_EN1_GET_csirx1_csi( ) GET_BIT(SCU_EXTREG_CLK_EN1, 9)
```

6.64.1.111 SCU_EXTREG_CLK_EN1_GET_csirx1_TxEscClk

```
#define SCU_EXTREG_CLK_EN1_GET_csirx1_TxEscClk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 10)
```

6.64.1.112 SCU_EXTREG_CLK_EN1_GET_csirx1_vc0

```
#define SCU_EXTREG_CLK_EN1_GET_csirx1_vc0( ) GET_BIT(SCU_EXTREG_CLK_EN1, 8)
```

6.64.1.113 SCU_EXTREG_CLK_EN1_GET_csitx_csi

```
#define SCU_EXTREG_CLK_EN1_GET_csitx_csi( ) GET_BIT(SCU_EXTREG_CLK_EN1, 12)
```

6.64.1.114 SCU_EXTREG_CLK_EN1_GET_csitx_dsi

```
#define SCU_EXTREG_CLK_EN1_GET_csitx_dsi( ) GET_BIT(SCU_EXTREG_CLK_EN1, 13)
```

Kneron Confidential

6.64.1.115 SCU_EXTREG_CLK_EN1_GET_LC_CLK

```
#define SCU_EXTREG_CLK_EN1_GET_LC_CLK( ) GET_BIT(SCU_EXTREG_CLK_EN1, 0)
```

6.64.1.116 SCU_EXTREG_CLK_EN1_GET_LC_SCALER

```
#define SCU_EXTREG_CLK_EN1_GET_LC_SCALER( ) GET_BIT(SCU_EXTREG_CLK_EN1, 1)
```

6.64.1.117 SCU_EXTREG_CLK_EN1_GET_npu

```
#define SCU_EXTREG_CLK_EN1_GET_npu( ) GET_BIT(SCU_EXTREG_CLK_EN1, 23)
```

6.64.1.118 SCU_EXTREG_CLK_EN1_GET_sdclk

```
#define SCU_EXTREG_CLK_EN1_GET_sdclk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 20)
```

6.64.1.119 SCU_EXTREG_CLK_EN1_GET_spi_clk

```
#define SCU_EXTREG_CLK_EN1_GET_spi_clk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 24)
```

6.64.1.120 SCU_EXTREG_CLK_EN1_GET_tx_EscClk

```
#define SCU_EXTREG_CLK_EN1_GET_tx_EscClk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 14)
```

6.64.1.121 SCU_EXTREG_CLK_EN1_GET_TxHsPllRefClk

```
#define SCU_EXTREG_CLK_EN1_GET_TxHsPllRefClk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 16)
```

6.64.1.122 SCU_EXTREG_CLK_EN1_GET_wdt_extclk

```
#define SCU_EXTREG_CLK_EN1_GET_wdt_extclk( ) GET_BIT(SCU_EXTREG_CLK_EN1, 21)
```

Kneron Confidential

6.64.1.123 SCU_EXTREG_CLK_EN1_SET_adcclk

```
#define SCU_EXTREG_CLK_EN1_SET_adcclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 22)
```

6.64.1.124 SCU_EXTREG_CLK_EN1_SET_csirx0_csi

```
#define SCU_EXTREG_CLK_EN1_SET_csirx0_csi(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 5)
```

6.64.1.125 SCU_EXTREG_CLK_EN1_SET_csirx0_TxEscClk

```
#define SCU_EXTREG_CLK_EN1_SET_csirx0_TxEscClk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 6)
```

6.64.1.126 SCU_EXTREG_CLK_EN1_SET_csirx0_vc0

```
#define SCU_EXTREG_CLK_EN1_SET_csirx0_vc0(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 4)
```

6.64.1.127 SCU_EXTREG_CLK_EN1_SET_csirx1_csi

```
#define SCU_EXTREG_CLK_EN1_SET_csirx1_csi(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 9)
```

6.64.1.128 SCU_EXTREG_CLK_EN1_SET_csirx1_TxEscClk

```
#define SCU_EXTREG_CLK_EN1_SET_csirx1_TxEscClk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 10)
```

6.64.1.129 SCU_EXTREG_CLK_EN1_SET_csirx1_vc0

```
#define SCU_EXTREG_CLK_EN1_SET_csirx1_vc0(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 8)
```

Kneron Confidential

6.64.1.130 SCU_EXTREG_CLK_EN1_SET_csitx_csi

```
#define SCU_EXTREG_CLK_EN1_SET_csitx_csi(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 12)
```

6.64.1.131 SCU_EXTREG_CLK_EN1_SET_csitx_dsi

```
#define SCU_EXTREG_CLK_EN1_SET_csitx_dsi(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 13)
```

6.64.1.132 SCU_EXTREG_CLK_EN1_SET_LC_CLK

```
#define SCU_EXTREG_CLK_EN1_SET_LC_CLK(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 0)
```

6.64.1.133 SCU_EXTREG_CLK_EN1_SET_LC_SCALER

```
#define SCU_EXTREG_CLK_EN1_SET_LC_SCALER(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 1)
```

6.64.1.134 SCU_EXTREG_CLK_EN1_SET_npu

```
#define SCU_EXTREG_CLK_EN1_SET_npu(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 23)
```

6.64.1.135 SCU_EXTREG_CLK_EN1_SET_sdclk

```
#define SCU_EXTREG_CLK_EN1_SET_sdclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 20)
```

6.64.1.136 SCU_EXTREG_CLK_EN1_SET_spi_clk

```
#define SCU_EXTREG_CLK_EN1_SET_spi_clk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 24)
```

Kneron Confidential

6.64.1.137 SCU_EXTREG_CLK_EN1_SET_tx_EscClk

```
#define SCU_EXTREG_CLK_EN1_SET_tx_EscClk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 14)
```

6.64.1.138 SCU_EXTREG_CLK_EN1_SET_TxHsPllRefClk

```
#define SCU_EXTREG_CLK_EN1_SET_TxHsPllRefClk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 16)
```

6.64.1.139 SCU_EXTREG_CLK_EN1_SET_wdt_extclk

```
#define SCU_EXTREG_CLK_EN1_SET_wdt_extclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN1, val, 21)
```

6.64.1.140 SCU_EXTREG_CLK_EN2

```
#define SCU_EXTREG_CLK_EN2 (SCU_EXTREG_PA_BASE + 0x001C)
```

6.64.1.141 SCU_EXTREG_CLK_EN2_GET_pwm_extclk1

```
#define SCU_EXTREG_CLK_EN2_GET_pwm_extclk1( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 16)
```

6.64.1.142 SCU_EXTREG_CLK_EN2_GET_pwm_extclk2

```
#define SCU_EXTREG_CLK_EN2_GET_pwm_extclk2( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 17)
```

6.64.1.143 SCU_EXTREG_CLK_EN2_GET_pwm_extclk3

```
#define SCU_EXTREG_CLK_EN2_GET_pwm_extclk3( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 18)
```

Kneron Confidential

6.64.1.144 SCU_EXTREG_CLK_EN2_GET_pwm_extclk4

```
#define SCU_EXTREG_CLK_EN2_GET_pwm_extclk4( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 19)
```

6.64.1.145 SCU_EXTREG_CLK_EN2_GET_pwm_extclk5

```
#define SCU_EXTREG_CLK_EN2_GET_pwm_extclk5( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 20)
```

6.64.1.146 SCU_EXTREG_CLK_EN2_GET_pwm_extclk6

```
#define SCU_EXTREG_CLK_EN2_GET_pwm_extclk6( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 21)
```

6.64.1.147 SCU_EXTREG_CLK_EN2_GET_ssp0_0_sspclk

```
#define SCU_EXTREG_CLK_EN2_GET_ssp0_0_sspclk( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 0)
```

6.64.1.148 SCU_EXTREG_CLK_EN2_GET_ssp0_1_sspclk

```
#define SCU_EXTREG_CLK_EN2_GET_ssp0_1_sspclk( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 2)
```

6.64.1.149 SCU_EXTREG_CLK_EN2_GET_ssp1_0_sspclk

```
#define SCU_EXTREG_CLK_EN2_GET_ssp1_0_sspclk( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 4)
```

6.64.1.150 SCU_EXTREG_CLK_EN2_GET_ssp1_1_sspclk

```
#define SCU_EXTREG_CLK_EN2_GET_ssp1_1_sspclk( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 6)
```

6.64.1.151 SCU_EXTREG_CLK_EN2_GET_tmr0_extclk1

```
#define SCU_EXTREG_CLK_EN2_GET_tmr0_extclk1( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 24)
```

Kneron Confidential

6.64.1.152 SCU_EXTREG_CLK_EN2_GET_tmr0_extclk2

```
#define SCU_EXTREG_CLK_EN2_GET_tmr0_extclk2( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 25)
```

6.64.1.153 SCU_EXTREG_CLK_EN2_GET_tmr0_extclk3

```
#define SCU_EXTREG_CLK_EN2_GET_tmr0_extclk3( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 26)
```

6.64.1.154 SCU_EXTREG_CLK_EN2_GET_tmr1_extclk1

```
#define SCU_EXTREG_CLK_EN2_GET_tmr1_extclk1( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 27)
```

6.64.1.155 SCU_EXTREG_CLK_EN2_GET_tmr1_extclk2

```
#define SCU_EXTREG_CLK_EN2_GET_tmr1_extclk2( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 28)
```

6.64.1.156 SCU_EXTREG_CLK_EN2_GET_tmr1_extclk3

```
#define SCU_EXTREG_CLK_EN2_GET_tmr1_extclk3( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 29)
```

6.64.1.157 SCU_EXTREG_CLK_EN2_GET_uart0_fref

```
#define SCU_EXTREG_CLK_EN2_GET_uart0_fref( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 8)
```

6.64.1.158 SCU_EXTREG_CLK_EN2_GET_uart1_0_fref

```
#define SCU_EXTREG_CLK_EN2_GET_uart1_0_fref( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 12)
```

6.64.1.159 SCU_EXTREG_CLK_EN2_GET_uart1_1_fref

```
#define SCU_EXTREG_CLK_EN2_GET_uart1_1_fref( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 13)
```

Kneron Confidential

6.64.1.160 SCU_EXTREG_CLK_EN2_GET_uart1_2_fref

```
#define SCU_EXTREG_CLK_EN2_GET_uart1_2_fref( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 14)
```

6.64.1.161 SCU_EXTREG_CLK_EN2_GET_uart1_3_fref

```
#define SCU_EXTREG_CLK_EN2_GET_uart1_3_fref( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 15)
```

6.64.1.162 SCU_EXTREG_CLK_EN2_SET_pwm_extclk1

```
#define SCU_EXTREG_CLK_EN2_SET_pwm_extclk1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 16)
```

6.64.1.163 SCU_EXTREG_CLK_EN2_SET_pwm_extclk2

```
#define SCU_EXTREG_CLK_EN2_SET_pwm_extclk2(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 17)
```

6.64.1.164 SCU_EXTREG_CLK_EN2_SET_pwm_extclk3

```
#define SCU_EXTREG_CLK_EN2_SET_pwm_extclk3(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 18)
```

6.64.1.165 SCU_EXTREG_CLK_EN2_SET_pwm_extclk4

```
#define SCU_EXTREG_CLK_EN2_SET_pwm_extclk4(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 19)
```

6.64.1.166 SCU_EXTREG_CLK_EN2_SET_pwm_extclk5

```
#define SCU_EXTREG_CLK_EN2_SET_pwm_extclk5(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 20)
```

Kneron Confidential

6.64.1.167 SCU_EXTREG_CLK_EN2_SET_pwm_extclk6

```
#define SCU_EXTREG_CLK_EN2_SET_pwm_extclk6(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 21)
```

6.64.1.168 SCU_EXTREG_CLK_EN2_SET_ssp0_0_sspclk

```
#define SCU_EXTREG_CLK_EN2_SET_ssp0_0_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 0)
```

6.64.1.169 SCU_EXTREG_CLK_EN2_SET_ssp0_1_sspclk

```
#define SCU_EXTREG_CLK_EN2_SET_ssp0_1_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 2)
```

6.64.1.170 SCU_EXTREG_CLK_EN2_SET_ssp1_0_sspclk

```
#define SCU_EXTREG_CLK_EN2_SET_ssp1_0_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 4)
```

6.64.1.171 SCU_EXTREG_CLK_EN2_SET_ssp1_1_sspclk

```
#define SCU_EXTREG_CLK_EN2_SET_ssp1_1_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 6)
```

6.64.1.172 SCU_EXTREG_CLK_EN2_SET_tmr0_extclk1

```
#define SCU_EXTREG_CLK_EN2_SET_tmr0_extclk1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 24)
```

6.64.1.173 SCU_EXTREG_CLK_EN2_SET_tmr0_extclk2

```
#define SCU_EXTREG_CLK_EN2_SET_tmr0_extclk2(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 25)
```

Kneron Confidential

6.64.1.174 SCU_EXTREG_CLK_EN2_SET_tmr0_extclk3

```
#define SCU_EXTREG_CLK_EN2_SET_tmr0_extclk3(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 26)
```

6.64.1.175 SCU_EXTREG_CLK_EN2_SET_tmr1_extclk1

```
#define SCU_EXTREG_CLK_EN2_SET_tmr1_extclk1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 27)
```

6.64.1.176 SCU_EXTREG_CLK_EN2_SET_tmr1_extclk2

```
#define SCU_EXTREG_CLK_EN2_SET_tmr1_extclk2(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 28)
```

6.64.1.177 SCU_EXTREG_CLK_EN2_SET_tmr1_extclk3

```
#define SCU_EXTREG_CLK_EN2_SET_tmr1_extclk3(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 29)
```

6.64.1.178 SCU_EXTREG_CLK_EN2_SET_uart0_fref

```
#define SCU_EXTREG_CLK_EN2_SET_uart0_fref(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 8)
```

6.64.1.179 SCU_EXTREG_CLK_EN2_SET_uart1_0_fref

```
#define SCU_EXTREG_CLK_EN2_SET_uart1_0_fref(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 12)
```

6.64.1.180 SCU_EXTREG_CLK_EN2_SET_uart1_1_fref

```
#define SCU_EXTREG_CLK_EN2_SET_uart1_1_fref(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 13)
```

Kneron Confidential

6.64.1.181 SCU_EXTREG_CLK_EN2_SET_uart1_2_fref

```
#define SCU_EXTREG_CLK_EN2_SET_uart1_2_fref(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 14)
```

6.64.1.182 SCU_EXTREG_CLK_EN2_SET_uart1_3_fref

```
#define SCU_EXTREG_CLK_EN2_SET_uart1_3_fref(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_EN2, val, 15)
```

6.64.1.183 SCU_EXTREG_CLK_MUX_SEL

```
#define SCU_EXTREG_CLK_MUX_SEL (SCU_EXTREG_PA_BASE + 0x0020)
```

6.64.1.184 SCU_EXTREG_CLK_MUX_SEL_GET_csirx1_clk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_csirx1_clk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 12)
```

6.64.1.185 SCU_EXTREG_CLK_MUX_SEL_GET_ncpu_traceclk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_ncpu_traceclk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 14)
```

6.64.1.186 SCU_EXTREG_CLK_MUX_SEL_GET_npu_clk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_npu_clk( ) GET_BITS(SCU_EXTREG_CLK_MUX_SEL, 8, 9)
```

6.64.1.187 SCU_EXTREG_CLK_MUX_SEL_GET_pll4_fref

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_pll4_fref( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 6)
```

Kneron Confidential

6.64.1.188 SCU_EXTREG_CLK_MUX_SEL_GET_scpu_traceclk_src

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_scpu_traceclk_src( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 13)
```

6.64.1.189 SCU_EXTREG_CLK_MUX_SEL_GET_ssp0_0_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_ssp0_0_sspclk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 0)
```

6.64.1.190 SCU_EXTREG_CLK_MUX_SEL_GET_ssp0_1_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_ssp0_1_sspclk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 1)
```

6.64.1.191 SCU_EXTREG_CLK_MUX_SEL_GET_ssp1_0_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_ssp1_0_sspclk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 2)
```

6.64.1.192 SCU_EXTREG_CLK_MUX_SEL_GET_ssp1_1_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_ssp1_1_sspclk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 3)
```

6.64.1.193 SCU_EXTREG_CLK_MUX_SEL_GET_uart_0_irda_uclk

```
#define SCU_EXTREG_CLK_MUX_SEL_GET_uart_0_irda_uclk( ) GET_BIT(SCU_EXTREG_CLK_MUX_SEL, 4)
```

6.64.1.194 SCU_EXTREG_CLK_MUX_SEL_SET_csirx1_clk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_csirx1_clk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 12)
```

Kneron Confidential

6.64.1.195 SCU_EXTREG_CLK_MUX_SEL_SET_ncpu_traceclk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_ncpu_traceclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 14)
```

6.64.1.196 SCU_EXTREG_CLK_MUX_SEL_SET_npu_clk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_npu_clk(
              val ) SET_MASKED_BITS(SCU_EXTREG_CLK_MUX_SEL, val, 8, 9)
```

6.64.1.197 SCU_EXTREG_CLK_MUX_SEL_SET_pll4_fref

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_pll4_fref(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 6)
```

6.64.1.198 SCU_EXTREG_CLK_MUX_SEL_SET_scpu_traceclk_src

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_scpu_traceclk_src(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 13)
```

6.64.1.199 SCU_EXTREG_CLK_MUX_SEL_SET_ssp0_0_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_ssp0_0_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 0)
```

6.64.1.200 SCU_EXTREG_CLK_MUX_SEL_SET_ssp0_1_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_ssp0_1_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 1)
```

6.64.1.201 SCU_EXTREG_CLK_MUX_SEL_SET_ssp1_0_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_ssp1_0_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 2)
```

Kneron Confidential

6.64.1.202 SCU_EXTREG_CLK_MUX_SEL_SET_ssp1_1_sspclk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_ssp1_1_sspclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 3)
```

6.64.1.203 SCU_EXTREG_CLK_MUX_SEL_SET_uart_0_irda_uclk

```
#define SCU_EXTREG_CLK_MUX_SEL_SET_uart_0_irda_uclk(
              val ) SET_MASKED_BIT(SCU_EXTREG_CLK_MUX_SEL, val, 4)
```

6.64.1.204 SCU_EXTREG_CM4_NCPU_CTRL

```
#define SCU_EXTREG_CM4_NCPU_CTRL (SCU_EXTREG_PA_BASE + 0x0068)
```

6.64.1.205 SCU_EXTREG_CM4_NCPU_CTRL_GET_wakeup

```
#define SCU_EXTREG_CM4_NCPU_CTRL_GET_wakeup( ) GET_BIT(SCU_EXTREG_CM4_NCPU_CTRL, 12)
```

6.64.1.206 SCU_EXTREG_CM4_NCPU_CTRL_SET_wakeup

```
#define SCU_EXTREG_CM4_NCPU_CTRL_SET_wakeup(
              val ) SET_MASKED_BIT(SCU_EXTREG_CM4_NCPU_CTRL, val, 12)
```

6.64.1.207 SCU_EXTREG_CSIRX_CTRL0

```
#define SCU_EXTREG_CSIRX_CTRL0 (SCU_EXTREG_PA_BASE + 0x0090)
```

6.64.1.208 SCU_EXTREG_CSIRX_CTRL0_apb_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_apb_rst_n BIT28
```

Kneron Confidential

6.64.1.209 SCU_EXTREG_CSIRX_CTRL0_ClkLnEn

```
#define SCU_EXTREG_CSIRX_CTRL0_ClkLnEn BIT1
```

6.64.1.210 SCU_EXTREG_CSIRX_CTRL0_Enable

```
#define SCU_EXTREG_CSIRX_CTRL0_Enable BIT0
```

6.64.1.211 SCU_EXTREG_CSIRX_CTRL0_GET_apb_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_apb_rst_n( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 28)
```

6.64.1.212 SCU_EXTREG_CSIRX_CTRL0_GET_ClkLnEn

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_ClkLnEn( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 1)
```

6.64.1.213 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKN

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKN( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 12)
```

6.64.1.214 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKN_1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKN_1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 12)
```

6.64.1.215 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKP

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKP( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 13)
```

6.64.1.216 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKP_1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_CKP_1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 13)
```

Kneron Confidential

6.64.1.217 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN0

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN0( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 14)
```

6.64.1.218 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN0_1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN0_1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 14)
```

6.64.1.219 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 16)
```

6.64.1.220 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN1_1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DN1_1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 16)
```

6.64.1.221 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP0

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP0( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 15)
```

6.64.1.222 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP0_1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP0_1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 15)
```

6.64.1.223 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 17)
```

6.64.1.224 SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP1_1

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_CMOS_IE_DP1_1( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 17)
```

Kneron Confidential

6.64.1.225 SCU_EXTREG_CSIRX_CTRL0_GET_Enable

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_Enable( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 0)
```

6.64.1.226 SCU_EXTREG_CSIRX_CTRL0_GET_pwr_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_pwr_rst_n( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 25)
```

6.64.1.227 SCU_EXTREG_CSIRX_CTRL0_GET_sys_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_GET_sys_rst_n( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL0, 24)
```

6.64.1.228 SCU_EXTREG_CSIRX_CTRL0_pwr_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_pwr_rst_n BIT25
```

6.64.1.229 SCU_EXTREG_CSIRX_CTRL0_SET_apb_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_apb_rst_n(
            val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 28)
```

6.64.1.230 SCU_EXTREG_CSIRX_CTRL0_SET_ClkLnEn

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_ClkLnEn(
            val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 1)
```

6.64.1.231 SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_CKN

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_CKN(
            val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 12)
```

Kneron Confidential

6.64.1.232 SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_CKP

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_CKP(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 13)
```

6.64.1.233 SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DN0

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DN0(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 14)
```

6.64.1.234 SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DN1

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DN1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 16)
```

6.64.1.235 SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DP0

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DP0(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 15)
```

6.64.1.236 SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DP1

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_CMOS_IE_DP1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 17)
```

6.64.1.237 SCU_EXTREG_CSIRX_CTRL0_SET_Enable

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_Enable(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 0)
```

6.64.1.238 SCU_EXTREG_CSIRX_CTRL0_SET_pwr_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_pwr_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 25)
```

Kneron Confidential

6.64.1.239 SCU_EXTREG_CSIRX_CTRL0_SET_sys_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_SET_sys_rst_n(
             val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL0, val, 24)
```

6.64.1.240 SCU_EXTREG_CSIRX_CTRL0_sys_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL0_sys_rst_n BIT24
```

6.64.1.241 SCU_EXTREG_CSIRX_CTRL1

```
#define SCU_EXTREG_CSIRX_CTRL1 (SCU_EXTREG_PA_BASE + 0x0094)
```

6.64.1.242 SCU_EXTREG_CSIRX_CTRL1_apb_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_apb_rst_n BIT28
```

6.64.1.243 SCU_EXTREG_CSIRX_CTRL1_ClkLnEn

```
#define SCU_EXTREG_CSIRX_CTRL1_ClkLnEn BIT1
```

6.64.1.244 SCU_EXTREG_CSIRX_CTRL1_Enable

```
#define SCU_EXTREG_CSIRX_CTRL1_Enable BIT0
```

6.64.1.245 SCU_EXTREG_CSIRX_CTRL1_GET_apb_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_GET_apb_rst_n( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 28)
```

Kneron Confidential

6.64.1.246 SCU_EXTREG_CSIRX_CTRL1_GET_ClkLnEn

```
#define SCU_EXTREG_CSIRX_CTRL1_GET_ClkLnEn( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 1)
```

6.64.1.247 SCU_EXTREG_CSIRX_CTRL1_GET_Enable

```
#define SCU_EXTREG_CSIRX_CTRL1_GET_Enable( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 0)
```

6.64.1.248 SCU_EXTREG_CSIRX_CTRL1_GET_pwr_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_GET_pwr_rst_n( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 25)
```

6.64.1.249 SCU_EXTREG_CSIRX_CTRL1_GET_sys_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_GET_sys_rst_n( ) GET_BIT(SCU_EXTREG_CSIRX_CTRL1, 24)
```

6.64.1.250 SCU_EXTREG_CSIRX_CTRL1_pwr_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_pwr_rst_n BIT25
```

6.64.1.251 SCU_EXTREG_CSIRX_CTRL1_SET_apb_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_apb_rst_n(
            val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 28)
```

6.64.1.252 SCU_EXTREG_CSIRX_CTRL1_SET_ClkLnEn

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_ClkLnEn(
             val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 1)
```

Kneron Confidential

6.64.1.253 SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_CKN_1

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_CKN_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 12)
```

6.64.1.254 SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_CKP_1

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_CKP_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 13)
```

6.64.1.255 SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DN0_1

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DN0_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 14)
```

6.64.1.256 SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DN1_1

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DN1_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 16)
```

6.64.1.257 SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DP0_1

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DP0_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 15)
```

6.64.1.258 SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DP1_1

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_CMOS_IE_DP1_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 17)
```

6.64.1.259 SCU_EXTREG_CSIRX_CTRL1_SET_Enable

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_Enable(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 0)
```

Kneron Confidential

6.64.1.260 SCU_EXTREG_CSIRX_CTRL1_SET_pwr_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_pwr_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 25)
```

6.64.1.261 SCU_EXTREG_CSIRX_CTRL1_SET_sys_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_SET_sys_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_CSIRX_CTRL1, val, 24)
```

6.64.1.262 SCU_EXTREG_CSIRX_CTRL1_sys_rst_n

```
#define SCU_EXTREG_CSIRX_CTRL1_sys_rst_n BIT24
```

6.64.1.263 SCU_EXTREG_DDR_CTRL

```
#define SCU_EXTREG_DDR_CTRL (SCU_EXTREG_PA_BASE + 0x0080)
```

6.64.1.264 SCU_EXTREG_DDR_CTRL_Dphy_resetn

```
#define SCU_EXTREG_DDR_CTRL_Dphy_resetn BIT29
```

6.64.1.265 SCU_EXTREG_DDR_CTRL_SELFBIAS

```
#define SCU_EXTREG_DDR_CTRL_SELFBIAS BIT15
```

6.64.1.266 SCU_EXTREG_DDR_CTRL_SET_Dphy_resetn

```
#define SCU_EXTREG_DDR_CTRL_SET_Dphy_resetn(
              val ) SET_MASKED_BIT(SCU_EXTREG_DDR_CTRL, val, 29)
```

Kneron Confidential

6.64.1.267 SCU_EXTREG_DDR_CTRL_SET_SELFBIAS

```
#define SCU_EXTREG_DDR_CTRL_SET_SELFBIAS(
              val ) SET_MASKED_BIT(SCU_EXTREG_DDR_CTRL, val, 15)
```

6.64.1.268 SCU_EXTREG_DDR_CTRL_SET_wakeup

```
#define SCU_EXTREG_DDR_CTRL_SET_wakeup(
              val ) SET_MASKED_BIT(SCU_EXTREG_DDR_CTRL, val, 28)
```

6.64.1.269 SCU_EXTREG_DDR_CTRL_wakeup

```
#define SCU_EXTREG_DDR_CTRL_wakeup BIT28
```

6.64.1.270 SCU_EXTREG_DPI2AHB_CTRL

```
#define SCU_EXTREG_DPI2AHB_CTRL (SCU_EXTREG_PA_BASE + 0x009C)
```

6.64.1.271 SCU_EXTREG_DPI2AHB_CTRL_apb_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_apb_rst_n BIT2
```

6.64.1.272 SCU_EXTREG_DPI2AHB_CTRL_apb_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_apb_rst_n_1 BIT6
```

6.64.1.273 SCU_EXTREG_DPI2AHB_CTRL_GET_apb_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_apb_rst_n( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 2)
```

Kneron Confidential

6.64.1.274 SCU_EXTREG_DPI2AHB_CTRL_GET_apb_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_apb_rst_n_1( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 6)
```

6.64.1.275 SCU_EXTREG_DPI2AHB_CTRL_GET_pwr_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_pwr_rst_n( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 0)
```

6.64.1.276 SCU_EXTREG_DPI2AHB_CTRL_GET_pwr_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_pwr_rst_n_1( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 4)
```

6.64.1.277 SCU_EXTREG_DPI2AHB_CTRL_GET_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_rst_n( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 3)
```

6.64.1.278 SCU_EXTREG_DPI2AHB_CTRL_GET_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_rst_n_1( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 7)
```

6.64.1.279 SCU_EXTREG_DPI2AHB_CTRL_GET_sys_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_sys_rst_n( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 1)
```

6.64.1.280 SCU_EXTREG_DPI2AHB_CTRL_GET_sys_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_GET_sys_rst_n_1( ) GET_BIT(SCU_EXTREG_DPI2AHB_CTRL, 5)
```

6.64.1.281 SCU_EXTREG_DPI2AHB_CTRL_pwr_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_pwr_rst_n BIT0
```

Kneron Confidential

6.64.1.282 SCU_EXTREG_DPI2AHB_CTRL_pwr_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_pwr_rst_n_1 BIT4
```

6.64.1.283 SCU_EXTREG_DPI2AHB_CTRL_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_rst_n BIT3
```

6.64.1.284 SCU_EXTREG_DPI2AHB_CTRL_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_rst_n_1 BIT7
```

6.64.1.285 SCU_EXTREG_DPI2AHB_CTRL_SET_apb_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_apb_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 2)
```

6.64.1.286 SCU_EXTREG_DPI2AHB_CTRL_SET_apb_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_apb_rst_n_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 6)
```

6.64.1.287 SCU_EXTREG_DPI2AHB_CTRL_SET_pwr_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_pwr_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 0)
```

6.64.1.288 SCU_EXTREG_DPI2AHB_CTRL_SET_pwr_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_pwr_rst_n_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 4)
```

Kneron Confidential

6.64.1.289 SCU_EXTREG_DPI2AHB_CTRL_SET_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 3)
```

6.64.1.290 SCU_EXTREG_DPI2AHB_CTRL_SET_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_rst_n_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 7)
```

6.64.1.291 SCU_EXTREG_DPI2AHB_CTRL_SET_sys_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_sys_rst_n(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 1)
```

6.64.1.292 SCU_EXTREG_DPI2AHB_CTRL_SET_sys_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_SET_sys_rst_n_1(
              val ) SET_MASKED_BIT(SCU_EXTREG_DPI2AHB_CTRL, val, 5)
```

6.64.1.293 SCU_EXTREG_DPI2AHB_CTRL_sys_rst_n

```
#define SCU_EXTREG_DPI2AHB_CTRL_sys_rst_n BIT1
```

6.64.1.294 SCU_EXTREG_DPI2AHB_CTRL_sys_rst_n_1

```
#define SCU_EXTREG_DPI2AHB_CTRL_sys_rst_n_1 BIT5
```

6.64.1.295 SCU_EXTREG_I2C0_CLK_IOCTRL

```
#define SCU_EXTREG_I2C0_CLK_IOCTRL (SCU_EXTREG_PA_BASE + 0x019C)
```

Kneron Confidential

6.64.1.296 SCU_EXTREG_I2C0_DATA_IOCTRL

```
#define SCU_EXTREG_I2C0_DATA_IOCTRL (SCU_EXTREG_PA_BASE + 0x01A0)
```

6.64.1.297 SCU_EXTREG_LC_DATA0_IOCTRL

```
#define SCU_EXTREG_LC_DATA0_IOCTRL (SCU_EXTREG_PA_BASE + 0x013C)
```

6.64.1.298 SCU_EXTREG_LC_DATA10_IOCTRL

```
#define SCU_EXTREG_LC_DATA10_IOCTRL (SCU_EXTREG_PA_BASE + 0x0164)
```

6.64.1.299 SCU_EXTREG_LC_DATA11_IOCTRL

```
#define SCU_EXTREG_LC_DATA11_IOCTRL (SCU_EXTREG_PA_BASE + 0x0168)
```

6.64.1.300 SCU_EXTREG_LC_DATA12_IOCTRL

```
#define SCU_EXTREG_LC_DATA12_IOCTRL (SCU_EXTREG_PA_BASE + 0x016C)
```

6.64.1.301 SCU_EXTREG_LC_DATA13_IOCTRL

```
#define SCU_EXTREG_LC_DATA13_IOCTRL (SCU_EXTREG_PA_BASE + 0x0170)
```

6.64.1.302 SCU_EXTREG_LC_DATA14_IOCTRL

```
#define SCU_EXTREG_LC_DATA14_IOCTRL (SCU_EXTREG_PA_BASE + 0x0174)
```

6.64.1.303 SCU_EXTREG_LC_DATA15_IOCTRL

```
#define SCU_EXTREG_LC_DATA15_IOCTRL (SCU_EXTREG_PA_BASE + 0x0178)
```

Kneron Confidential

6.64.1.304 SCU_EXTREG_LC_DATA1_IOCTRL

```
#define SCU_EXTREG_LC_DATA1_IOCTRL (SCU_EXTREG_PA_BASE + 0x0140)
```

6.64.1.305 SCU_EXTREG_LC_DATA2_IOCTRL

```
#define SCU_EXTREG_LC_DATA2_IOCTRL (SCU_EXTREG_PA_BASE + 0x0144)
```

6.64.1.306 SCU_EXTREG_LC_DATA3_IOCTRL

```
#define SCU_EXTREG_LC_DATA3_IOCTRL (SCU_EXTREG_PA_BASE + 0x0148)
```

6.64.1.307 SCU_EXTREG_LC_DATA4_IOCTRL

```
#define SCU_EXTREG_LC_DATA4_IOCTRL (SCU_EXTREG_PA_BASE + 0x014C)
```

6.64.1.308 SCU_EXTREG_LC_DATA5_IOCTRL

```
#define SCU_EXTREG_LC_DATA5_IOCTRL (SCU_EXTREG_PA_BASE + 0x0150)
```

6.64.1.309 SCU_EXTREG_LC_DATA6_GET_dcsr

```
#define SCU_EXTREG_LC_DATA6_GET_dcsr( ) GET_BITS(SCU_EXTREG_LC_DATA6_IOCTRL, 5, 8)
```

6.64.1.310 SCU_EXTREG_LC_DATA6_GET_pd

```
#define SCU_EXTREG_LC_DATA6_GET_pd( ) GET_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 4)
```

6.64.1.311 SCU_EXTREG_LC_DATA6_GET_pu

```
#define SCU_EXTREG_LC_DATA6_GET_pu( ) GET_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 3)
```

Kneron Confidential

6.64.1.312 SCU_EXTREG_LC_DATA6_IOCTRL

```
#define SCU_EXTREG_LC_DATA6_IOCTRL (SCU_EXTREG_PA_BASE + 0x0154)
```

6.64.1.313 SCU_EXTREG_LC_DATA6_SET_dcsr

```
#define SCU_EXTREG_LC_DATA6_SET_dcsr(
            val ) SET_MASKED_BITS(SCU_EXTREG_LC_DATA6_IOCTRL, val, 5, 8)
```

6.64.1.314 SCU_EXTREG_LC_DATA6_SET_pd

```
#define SCU_EXTREG_LC_DATA6_SET_pd(
            val ) SET_MASKED_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 4)
```

6.64.1.315 SCU_EXTREG_LC_DATA6_SET_pu

```
#define SCU_EXTREG_LC_DATA6_SET_pu(
            val ) SET_MASKED_BIT(SCU_EXTREG_LC_DATA6_IOCTRL, 3)
```

6.64.1.316 SCU_EXTREG_LC_DATA7_GET_dcsr

```
#define SCU_EXTREG_LC_DATA7_GET_dcsr( ) GET_BITS(SCU_EXTREG_LC_DATA7_IOCTRL, 5, 8)
```

6.64.1.317 SCU_EXTREG_LC_DATA7_GET_pd

```
#define SCU_EXTREG_LC_DATA7_GET_pd( ) GET_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 4)
```

6.64.1.318 SCU_EXTREG_LC_DATA7_GET_pu

```
#define SCU_EXTREG_LC_DATA7_GET_pu( ) GET_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 3)
```

Kneron Confidential

6.64.1.319 SCU_EXTREG_LC_DATA7_IOCTRL

```
#define SCU_EXTREG_LC_DATA7_IOCTRL (SCU_EXTREG_PA_BASE + 0x0158)
```

6.64.1.320 SCU_EXTREG_LC_DATA7_SET_dcsr

```
#define SCU_EXTREG_LC_DATA7_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_LC_DATA7_IOCTRL, val, 5, 8)
```

6.64.1.321 SCU_EXTREG_LC_DATA7_SET_pd

```
#define SCU_EXTREG_LC_DATA7_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 4)
```

6.64.1.322 SCU_EXTREG_LC_DATA7_SET_pu

```
#define SCU_EXTREG_LC_DATA7_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_LC_DATA7_IOCTRL, 3)
```

6.64.1.323 SCU_EXTREG_LC_DATA8_IOCTRL

```
#define SCU_EXTREG_LC_DATA8_IOCTRL (SCU_EXTREG_PA_BASE + 0x015C)
```

6.64.1.324 SCU_EXTREG_LC_DATA9_IOCTRL

```
#define SCU_EXTREG_LC_DATA9_IOCTRL (SCU_EXTREG_PA_BASE + 0x0160)
```

6.64.1.325 SCU_EXTREG_LC_DE_IOCTRL

```
#define SCU_EXTREG_LC_DE_IOCTRL (SCU_EXTREG_PA_BASE + 0x0138)
```

Kneron Confidential

6.64.1.326 SCU_EXTREG_LC_HS_IOCTRL

```
#define SCU_EXTREG_LC_HS_IOCTRL (SCU_EXTREG_PA_BASE + 0x0134)
```

6.64.1.327 SCU_EXTREG_LC_PCLK_IOCTRL

```
#define SCU_EXTREG_LC_PCLK_IOCTRL (SCU_EXTREG_PA_BASE + 0x012C)
```

6.64.1.328 SCU_EXTREG_LC_VS_IOCTRL

```
#define SCU_EXTREG_LC_VS_IOCTRL (SCU_EXTREG_PA_BASE + 0x0130)
```

6.64.1.329 SCU_EXTREG_MISC

```
#define SCU_EXTREG_MISC (SCU_EXTREG_PA_BASE + 0x00B0)
```

6.64.1.330 SCU_EXTREG_MISC_GET_lcm_cken

```
#define SCU_EXTREG_MISC_GET_lcm_cken( ) GET_BIT(SCU_EXTREG_MISC, 12)
```

6.64.1.331 SCU_EXTREG_MISC_GET_smr_por_n

```
#define SCU_EXTREG_MISC_GET_smr_por_n( ) GET_BITS(SCU_EXTREG_MISC, 4, 6)
```

6.64.1.332 SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DDRCK

```
#define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DDRCK BIT6
```

6.64.1.333 SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DEFAULT

```
#define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DEFAULT BIT4
```

Kneron Confidential

6.64.1.334 SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_MASK

```
#define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_MASK
```

Value:

```
(SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DDRCK | \
                                                  SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_NPU |
      \
      SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_DEFAULT)
```

6.64.1.335 SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_NPU

```
#define SCU_EXTREG_MISC_PWR_RESET_RELEASE_DOMAIN_NPU BIT5
```

6.64.1.336 SCU_EXTREG_MISC_SET_DPI_MUX_0_get

```
#define SCU_EXTREG_MISC_SET_DPI_MUX_0_get( ) GET_BIT(SCU_EXTREG_MISC, 8)
```

6.64.1.337 SCU_EXTREG_MISC_SET_DPI_MUX_0_sel

```
#define SCU_EXTREG_MISC_SET_DPI_MUX_0_sel(
             val ) SET_MASKED_BIT(SCU_EXTREG_MISC, val, 8)
```

6.64.1.338 SCU_EXTREG_MISC_SET_DPI_MUX_1_get

```
#define SCU_EXTREG_MISC_SET_DPI_MUX_1_get( ) GET_BIT(SCU_EXTREG_MISC, 9)
```

6.64.1.339 SCU_EXTREG_MISC_SET_DPI_MUX_1_sel

```
#define SCU_EXTREG_MISC_SET_DPI_MUX_1_sel(
             val ) SET_MASKED_BIT(SCU_EXTREG_MISC, val, 9)
```

6.64.1.340 SCU_EXTREG_MISC_SET_lcm_cken

```
#define SCU_EXTREG_MISC_SET_lcm_cken(
             val ) SET_MASKED_BIT(SCU_EXTREG_MISC, val, 12)
```

Kneron Confidential

6.64.1.341 SCU_EXTREG_OTG_DRV_VBUS_IOCTRL

```
#define SCU_EXTREG_OTG_DRV_VBUS_IOCTRL (SCU_EXTREG_PA_BASE + 0x01A8)
```

6.64.1.342 SCU_EXTREG_PLL0_SETTING

```
#define SCU_EXTREG_PLL0_SETTING (SCU_EXTREG_PA_BASE + 0x0000)
```

6.64.1.343 SCU_EXTREG_PLL0_SETTING_GET_cc

```
#define SCU_EXTREG_PLL0_SETTING_GET_cc( ) GET_BITS(SCU_EXTREG_PLL0_SETTING, 12, 13)
```

6.64.1.344 SCU_EXTREG_PLL0_SETTING_GET_en

```
#define SCU_EXTREG_PLL0_SETTING_GET_en( ) GET_BIT(SCU_EXTREG_PLL0_SETTING, 0)
```

6.64.1.345 SCU_EXTREG_PLL0_SETTING_GET_f

```
#define SCU_EXTREG_PLL0_SETTING_GET_f( ) GET_BITS(SCU_EXTREG_PLL0_SETTING, 8, 9)
```

6.64.1.346 SCU_EXTREG_PLL0_SETTING_GET_ms

```
#define SCU_EXTREG_PLL0_SETTING_GET_ms( ) GET_BITS(SCU_EXTREG_PLL0_SETTING, 16, 20)
```

6.64.1.347 SCU_EXTREG_PLL0_SETTING_GET_ns

```
#define SCU_EXTREG_PLL0_SETTING_GET_ns( ) GET_BITS(SCU_EXTREG_PLL0_SETTING, 24, 30)
```

6.64.1.348 SCU_EXTREG_PLL0_SETTING_SET_cc

```
#define SCU_EXTREG_PLL0_SETTING_SET_cc( ) SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 12, 13)
```

Kneron Confidential

6.64.1.349 SCU_EXTREG_PLL0_SETTING_SET_en

```
#define SCU_EXTREG_PLL0_SETTING_SET_en( ) SET_MASKED_BIT(SCU_EXTREG_PLL0_SETTING, 0)
```

6.64.1.350 SCU_EXTREG_PLL0_SETTING_SET_f

```
#define SCU_EXTREG_PLL0_SETTING_SET_f( ) SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 8, 9)
```

6.64.1.351 SCU_EXTREG_PLL0_SETTING_SET_ms

```
#define SCU_EXTREG_PLL0_SETTING_SET_ms( ) SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 16, 20)
```

6.64.1.352 SCU_EXTREG_PLL0_SETTING_SET_ns

```
#define SCU_EXTREG_PLL0_SETTING_SET_ns( ) SET_MASKED_BITS(SCU_EXTREG_PLL0_SETTING, 24, 30)
```

6.64.1.353 SCU_EXTREG_PLL1_SETTING

```
#define SCU_EXTREG_PLL1_SETTING (SCU_EXTREG_PA_BASE + 0x0004)
```

6.64.1.354 SCU_EXTREG_PLL1_SETTING_GET_cc

```
#define SCU_EXTREG_PLL1_SETTING_GET_cc( ) GET_BITS(SCU_EXTREG_PLL1_SETTING, 12, 13)
```

6.64.1.355 SCU_EXTREG_PLL1_SETTING_GET_en

```
#define SCU_EXTREG_PLL1_SETTING_GET_en( ) GET_BIT(SCU_EXTREG_PLL1_SETTING, 0)
```

6.64.1.356 SCU_EXTREG_PLL1_SETTING_GET_f

```
#define SCU_EXTREG_PLL1_SETTING_GET_f( ) GET_BITS(SCU_EXTREG_PLL1_SETTING, 8, 9)
```

Kneron Confidential

6.64.1.357 SCU_EXTREG_PLL1_SETTING_GET_ms

```
#define SCU_EXTREG_PLL1_SETTING_GET_ms( ) GET_BITS(SCU_EXTREG_PLL1_SETTING, 16, 20)
```

6.64.1.358 SCU_EXTREG_PLL1_SETTING_GET_ns

```
#define SCU_EXTREG_PLL1_SETTING_GET_ns( ) GET_BITS(SCU_EXTREG_PLL1_SETTING, 24, 30)
```

6.64.1.359 SCU_EXTREG_PLL1_SETTING_SET_cc

```
#define SCU_EXTREG_PLL1_SETTING_SET_cc(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 12, 13)
```

6.64.1.360 SCU_EXTREG_PLL1_SETTING_SET_en

```
#define SCU_EXTREG_PLL1_SETTING_SET_en(
              val ) SET_MASKED_BIT(SCU_EXTREG_PLL1_SETTING, val, 0)
```

6.64.1.361 SCU_EXTREG_PLL1_SETTING_SET_f

```
#define SCU_EXTREG_PLL1_SETTING_SET_f(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 8, 9)
```

6.64.1.362 SCU_EXTREG_PLL1_SETTING_SET_ms

```
#define SCU_EXTREG_PLL1_SETTING_SET_ms(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 16, 20)
```

6.64.1.363 SCU_EXTREG_PLL1_SETTING_SET_ns

```
#define SCU_EXTREG_PLL1_SETTING_SET_ns(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL1_SETTING, val, 24, 30)
```

Kneron Confidential

6.64.1.364 SCU_EXTREG_PLL2_SETTING

```
#define SCU_EXTREG_PLL2_SETTING (SCU_EXTREG_PA_BASE + 0x0008)
```

6.64.1.365 SCU_EXTREG_PLL2_SETTING_GET_cc

```
#define SCU_EXTREG_PLL2_SETTING_GET_cc( ) GET_BITS(SCU_EXTREG_PLL2_SETTING, 12, 13)
```

6.64.1.366 SCU_EXTREG_PLL2_SETTING_GET_en

```
#define SCU_EXTREG_PLL2_SETTING_GET_en( ) GET_BIT(SCU_EXTREG_PLL2_SETTING, 0)
```

6.64.1.367 SCU_EXTREG_PLL2_SETTING_GET_f

```
#define SCU_EXTREG_PLL2_SETTING_GET_f( ) GET_BITS(SCU_EXTREG_PLL2_SETTING, 8, 9)
```

6.64.1.368 SCU_EXTREG_PLL2_SETTING_GET_ms

```
#define SCU_EXTREG_PLL2_SETTING_GET_ms( ) GET_BITS(SCU_EXTREG_PLL2_SETTING, 16, 20)
```

6.64.1.369 SCU_EXTREG_PLL2_SETTING_GET_ns

```
#define SCU_EXTREG_PLL2_SETTING_GET_ns( ) GET_BITS(SCU_EXTREG_PLL2_SETTING, 24, 30)
```

6.64.1.370 SCU_EXTREG_PLL2_SETTING_SET_cc

```
#define SCU_EXTREG_PLL2_SETTING_SET_cc(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 12, 13)
```

Kneron Confidential

6.64.1.371 SCU_EXTREG_PLL2_SETTING_SET_en

```
#define SCU_EXTREG_PLL2_SETTING_SET_en(
              val ) SET_MASKED_BIT(SCU_EXTREG_PLL2_SETTING, val, 0)
```

6.64.1.372 SCU_EXTREG_PLL2_SETTING_SET_f

```
#define SCU_EXTREG_PLL2_SETTING_SET_f(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 8, 9)
```

6.64.1.373 SCU_EXTREG_PLL2_SETTING_SET_ms

```
#define SCU_EXTREG_PLL2_SETTING_SET_ms(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 16, 20)
```

6.64.1.374 SCU_EXTREG_PLL2_SETTING_SET_ns

```
#define SCU_EXTREG_PLL2_SETTING_SET_ns(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL2_SETTING, val, 24, 30)
```

6.64.1.375 SCU_EXTREG_PLL3_SETTING

```
#define SCU_EXTREG_PLL3_SETTING (SCU_EXTREG_PA_BASE + 0x000C)
```

6.64.1.376 SCU_EXTREG_PLL3_SETTING_GET_en

```
#define SCU_EXTREG_PLL3_SETTING_GET_en( ) GET_BIT(SCU_EXTREG_PLL3_SETTING, 0)
```

6.64.1.377 SCU_EXTREG_PLL3_SETTING_GET_is

```
#define SCU_EXTREG_PLL3_SETTING_GET_is( ) GET_BITS(SCU_EXTREG_PLL3_SETTING, 4, 6)
```

Kneron Confidential

6.64.1.378 SCU_EXTREG_PLL3_SETTING_GET_ms

```
#define SCU_EXTREG_PLL3_SETTING_GET_ms( ) GET_BITS(SCU_EXTREG_PLL3_SETTING, 8, 10)
```

6.64.1.379 SCU_EXTREG_PLL3_SETTING_GET_ns

```
#define SCU_EXTREG_PLL3_SETTING_GET_ns( ) GET_BITS(SCU_EXTREG_PLL3_SETTING, 12, 20)
```

6.64.1.380 SCU_EXTREG_PLL3_SETTING_GET_ps

```
#define SCU_EXTREG_PLL3_SETTING_GET_ps( ) GET_BITS(SCU_EXTREG_PLL3_SETTING, 24, 28)
```

6.64.1.381 SCU_EXTREG_PLL3_SETTING_GET_rs

```
#define SCU_EXTREG_PLL3_SETTING_GET_rs( ) GET_BITS(SCU_EXTREG_PLL3_SETTING, 2, 3)
```

6.64.1.382 SCU_EXTREG_PLL3_SETTING_is_MASK

```
#define SCU_EXTREG_PLL3_SETTING_is_MASK (BIT4|BIT5|BIT6)
```

6.64.1.383 SCU_EXTREG_PLL3_SETTING_is_START

```
#define SCU_EXTREG_PLL3_SETTING_is_START 4
```

6.64.1.384 SCU_EXTREG_PLL3_SETTING_ms_MASK

```
#define SCU_EXTREG_PLL3_SETTING_ms_MASK (BIT8|BIT9|BIT10)
```

6.64.1.385 SCU_EXTREG_PLL3_SETTING_ms_START

```
#define SCU_EXTREG_PLL3_SETTING_ms_START 8
```

Kneron Confidential

6.64.1.386 SCU_EXTREG_PLL3_SETTING_ns_MASK

```
#define SCU_EXTREG_PLL3_SETTING_ns_MASK (BIT12|BIT13|BIT14|BIT15|BIT16|BIT17|BIT18|BIT19|BIT20)
```

6.64.1.387 SCU_EXTREG_PLL3_SETTING_ns_START

```
#define SCU_EXTREG_PLL3_SETTING_ns_START 12
```

6.64.1.388 SCU_EXTREG_PLL3_SETTING_ps_MASK

```
#define SCU_EXTREG_PLL3_SETTING_ps_MASK (BIT24|BIT25|BIT26|BIT27|BIT28)
```

6.64.1.389 SCU_EXTREG_PLL3_SETTING_ps_START

```
#define SCU_EXTREG_PLL3_SETTING_ps_START 24
```

6.64.1.390 SCU_EXTREG_PLL3_SETTING_rs_MASK

```
#define SCU_EXTREG_PLL3_SETTING_rs_MASK (BIT2|BIT3)
```

6.64.1.391 SCU_EXTREG_PLL3_SETTING_rs_START

```
#define SCU_EXTREG_PLL3_SETTING_rs_START 2
```

6.64.1.392 SCU_EXTREG_PLL3_SETTING_SET_en

```
#define SCU_EXTREG_PLL3_SETTING_SET_en(
              val ) SET_MASKED_BIT(SCU_EXTREG_PLL3_SETTING, val, 0)
```

Kneron Confidential

6.64.1.393 SCU_EXTREG_PLL3_SETTING_SET_is

```
#define SCU_EXTREG_PLL3_SETTING_SET_is(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 4, 6)
```

6.64.1.394 SCU_EXTREG_PLL3_SETTING_SET_ms

```
#define SCU_EXTREG_PLL3_SETTING_SET_ms(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 8, 10)
```

6.64.1.395 SCU_EXTREG_PLL3_SETTING_SET_ns

```
#define SCU_EXTREG_PLL3_SETTING_SET_ns(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 12, 20)
```

6.64.1.396 SCU_EXTREG_PLL3_SETTING_SET_ps

```
#define SCU_EXTREG_PLL3_SETTING_SET_ps(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 24, 28)
```

6.64.1.397 SCU_EXTREG_PLL3_SETTING_SET_rs

```
#define SCU_EXTREG_PLL3_SETTING_SET_rs(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL3_SETTING, val, 2, 3)
```

6.64.1.398 SCU_EXTREG_PLL4_SETTING

```
#define SCU_EXTREG_PLL4_SETTING (SCU_EXTREG_PA_BASE + 0x0010)
```

6.64.1.399 SCU_EXTREG_PLL4_SETTING_GET_en

```
#define SCU_EXTREG_PLL4_SETTING_GET_en( ) GET_BIT(SCU_EXTREG_PLL4_SETTING, 0)
```

Kneron Confidential

6.64.1.400 SCU_EXTREG_PLL4_SETTING_GET_is

```
#define SCU_EXTREG_PLL4_SETTING_GET_is( ) GET_BITS(SCU_EXTREG_PLL4_SETTING, 4, 6)
```

6.64.1.401 SCU_EXTREG_PLL4_SETTING_GET_ms

```
#define SCU_EXTREG_PLL4_SETTING_GET_ms( ) GET_BITS(SCU_EXTREG_PLL4_SETTING, 8, 10)
```

6.64.1.402 SCU_EXTREG_PLL4_SETTING_GET_ns

```
#define SCU_EXTREG_PLL4_SETTING_GET_ns( ) GET_BITS(SCU_EXTREG_PLL4_SETTING, 12, 20)
```

6.64.1.403 SCU_EXTREG_PLL4_SETTING_GET_ps

```
#define SCU_EXTREG_PLL4_SETTING_GET_ps( ) GET_BITS(SCU_EXTREG_PLL4_SETTING, 24, 28)
```

6.64.1.404 SCU_EXTREG_PLL4_SETTING_GET_rs

```
#define SCU_EXTREG_PLL4_SETTING_GET_rs( ) GET_BITS(SCU_EXTREG_PLL4_SETTING, 2, 3)
```

6.64.1.405 SCU_EXTREG_PLL4_SETTING_SET_en

```
#define SCU_EXTREG_PLL4_SETTING_SET_en(
            val ) SET_MASKED_BIT(SCU_EXTREG_PLL4_SETTING, val, 0)
```

6.64.1.406 SCU_EXTREG_PLL4_SETTING_SET_is

```
#define SCU_EXTREG_PLL4_SETTING_SET_is(
             val ) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 4, 6)
```

Kneron Confidential

6.64.1.407 SCU_EXTREG_PLL4_SETTING_SET_ms

```
#define SCU_EXTREG_PLL4_SETTING_SET_ms(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 8, 10)
```

6.64.1.408 SCU_EXTREG_PLL4_SETTING_SET_ns

```
#define SCU_EXTREG_PLL4_SETTING_SET_ns(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 12, 20)
```

6.64.1.409 SCU_EXTREG_PLL4_SETTING_SET_ps

```
#define SCU_EXTREG_PLL4_SETTING_SET_ps(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 24, 28)
```

6.64.1.410 SCU_EXTREG_PLL4_SETTING_SET_rs

```
#define SCU_EXTREG_PLL4_SETTING_SET_rs(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL4_SETTING, val, 2, 3)
```

6.64.1.411 SCU_EXTREG_PLL5_SETTING

```
#define SCU_EXTREG_PLL5_SETTING (SCU_EXTREG_PA_BASE + 0x003C)
```

6.64.1.412 SCU_EXTREG_PLL5_SETTING_GET_en

```
#define SCU_EXTREG_PLL5_SETTING_GET_en( ) GET_BIT(SCU_EXTREG_PLL5_SETTING, 0)
```

6.64.1.413 SCU_EXTREG_PLL5_SETTING_GET_is

```
#define SCU_EXTREG_PLL5_SETTING_GET_is( ) GET_BITS(SCU_EXTREG_PLL5_SETTING, 4, 6)
```

Kneron Confidential

6.64.1.414 SCU_EXTREG_PLL5_SETTING_GET_ms

```
#define SCU_EXTREG_PLL5_SETTING_GET_ms( ) GET_BITS(SCU_EXTREG_PLL5_SETTING, 8, 10)
```

6.64.1.415 SCU_EXTREG_PLL5_SETTING_GET_ns

```
#define SCU_EXTREG_PLL5_SETTING_GET_ns( ) GET_BITS(SCU_EXTREG_PLL5_SETTING, 12, 20)
```

6.64.1.416 SCU_EXTREG_PLL5_SETTING_GET_ps

```
#define SCU_EXTREG_PLL5_SETTING_GET_ps( ) GET_BITS(SCU_EXTREG_PLL5_SETTING, 24, 28)
```

6.64.1.417 SCU_EXTREG_PLL5_SETTING_GET_rs

```
#define SCU_EXTREG_PLL5_SETTING_GET_rs( ) GET_BITS(SCU_EXTREG_PLL5_SETTING, 2, 3)
```

6.64.1.418 SCU_EXTREG_PLL5_SETTING_is_MASK

```
#define SCU_EXTREG_PLL5_SETTING_is_MASK (BIT4|BIT5|BIT6)
```

6.64.1.419 SCU_EXTREG_PLL5_SETTING_is_START

```
#define SCU_EXTREG_PLL5_SETTING_is_START 4
```

6.64.1.420 SCU_EXTREG_PLL5_SETTING_ms_MASK

```
#define SCU_EXTREG_PLL5_SETTING_ms_MASK (BIT8|BIT9|BIT10)
```

6.64.1.421 SCU_EXTREG_PLL5_SETTING_ms_START

```
#define SCU_EXTREG_PLL5_SETTING_ms_START 8
```

Kneron Confidential

6.64.1.422 SCU_EXTREG_PLL5_SETTING_ns_MASK

```
#define SCU_EXTREG_PLL5_SETTING_ns_MASK (BIT12|BIT13|BIT14|BIT15|BIT16|BIT17|BIT18|BIT19|BIT20)
```

6.64.1.423 SCU_EXTREG_PLL5_SETTING_ns_START

```
#define SCU_EXTREG_PLL5_SETTING_ns_START 12
```

6.64.1.424 SCU_EXTREG_PLL5_SETTING_ps_MASK

```
#define SCU_EXTREG_PLL5_SETTING_ps_MASK (BIT24|BIT25|BIT26|BIT27|BIT28)
```

6.64.1.425 SCU_EXTREG_PLL5_SETTING_ps_START

```
#define SCU_EXTREG_PLL5_SETTING_ps_START 24
```

6.64.1.426 SCU_EXTREG_PLL5_SETTING_rs_MASK

```
#define SCU_EXTREG_PLL5_SETTING_rs_MASK (BIT2|BIT3)
```

6.64.1.427 SCU_EXTREG_PLL5_SETTING_rs_START

```
#define SCU_EXTREG_PLL5_SETTING_rs_START 2
```

6.64.1.428 SCU_EXTREG_PLL5_SETTING_SET_en

```
#define SCU_EXTREG_PLL5_SETTING_SET_en(
              val ) SET_MASKED_BIT(SCU_EXTREG_PLL5_SETTING, val, 0)
```

Kneron Confidential

6.64.1.429 SCU_EXTREG_PLL5_SETTING_SET_is

```
#define SCU_EXTREG_PLL5_SETTING_SET_is(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 4, 6)
```

6.64.1.430 SCU_EXTREG_PLL5_SETTING_SET_ms

```
#define SCU_EXTREG_PLL5_SETTING_SET_ms(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 8, 10)
```

6.64.1.431 SCU_EXTREG_PLL5_SETTING_SET_ns

```
#define SCU_EXTREG_PLL5_SETTING_SET_ns(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 12, 20)
```

6.64.1.432 SCU_EXTREG_PLL5_SETTING_SET_ps

```
#define SCU_EXTREG_PLL5_SETTING_SET_ps(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 24, 28)
```

6.64.1.433 SCU_EXTREG_PLL5_SETTING_SET_rs

```
#define SCU_EXTREG_PLL5_SETTING_SET_rs(
              val ) SET_MASKED_BITS(SCU_EXTREG_PLL5_SETTING, val, 2, 3)
```

6.64.1.434 SCU_EXTREG_PRINT

```
#define SCU_EXTREG_PRINT(
              __desc,
              __regtype,
              __symbol ) DSG("%s %s %x", __desc, #__symbol, SCU_EXTREG_##__regtype##_GET##_##↩
__symbol())
```

Kneron Confidential

6.64.1.435 SCU_EXTREG_PRINT2

```
#define SCU_EXTREG_PRINT2(
              __label ) {DSG("%s=%x", #__label, inw(__label))};
```

6.64.1.436 SCU_EXTREG_PRINT_ALL

```
#define SCU_EXTREG_PRINT_ALL( )
```

6.64.1.437 SCU_EXTREG_PWM0_IOCTRL

```
#define SCU_EXTREG_PWM0_IOCTRL (SCU_EXTREG_PA_BASE + 0x01A4)
```

6.64.1.438 SCU_EXTREG_SD_CLK_IOCTRL

```
#define SCU_EXTREG_SD_CLK_IOCTRL (SCU_EXTREG_PA_BASE + 0x017C)
```

6.64.1.439 SCU_EXTREG_SD_CMD_IOCTRL

```
#define SCU_EXTREG_SD_CMD_IOCTRL (SCU_EXTREG_PA_BASE + 0x0180)
```

6.64.1.440 SCU_EXTREG_SD_CTRL

```
#define SCU_EXTREG_SD_CTRL (SCU_EXTREG_PA_BASE + 0x00A4)
```

6.64.1.441 SCU_EXTREG_SD_CTRL_GET_io_sd_cd

```
#define SCU_EXTREG_SD_CTRL_GET_io_sd_cd(
              val ) GET_BIT(SCU_EXTREG_SD_CTRL, 0)
```

Kneron Confidential

6.64.1.442 SCU_EXTREG_SD_CTRL_GET_io_sd_wp

```
#define SCU_EXTREG_SD_CTRL_GET_io_sd_wp(
              val ) GET_BIT(SCU_EXTREG_SD_CTRL, 1)
```

6.64.1.443 SCU_EXTREG_SD_CTRL_SET_io_sd_cd

```
#define SCU_EXTREG_SD_CTRL_SET_io_sd_cd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SD_CTRL, 0)
```

6.64.1.444 SCU_EXTREG_SD_CTRL_SET_io_sd_wp

```
#define SCU_EXTREG_SD_CTRL_SET_io_sd_wp(
              val ) SET_MASKED_BIT(SCU_EXTREG_SD_CTRL, 1)
```

6.64.1.445 SCU_EXTREG_SD_DATA0_IOCTRL

```
#define SCU_EXTREG_SD_DATA0_IOCTRL (SCU_EXTREG_PA_BASE + 0x0184)
```

6.64.1.446 SCU_EXTREG_SD_DATA1_IOCTRL

```
#define SCU_EXTREG_SD_DATA1_IOCTRL (SCU_EXTREG_PA_BASE + 0x0188)
```

6.64.1.447 SCU_EXTREG_SD_DATA2_IOCTRL

```
#define SCU_EXTREG_SD_DATA2_IOCTRL (SCU_EXTREG_PA_BASE + 0x018C)
```

6.64.1.448 SCU_EXTREG_SD_DATA3_IOCTRL

```
#define SCU_EXTREG_SD_DATA3_IOCTRL (SCU_EXTREG_PA_BASE + 0x0190)
```

Kneron Confidential

6.64.1.449 SCU_EXTREG_SPARE0_IOCTRL

```
#define SCU_EXTREG_SPARE0_IOCTRL (SCU_EXTREG_PA_BASE + 0x01B0)
```

6.64.1.450 SCU_EXTREG_SPARE1_IOCTRL

```
#define SCU_EXTREG_SPARE1_IOCTRL (SCU_EXTREG_PA_BASE + 0x01B4)
```

6.64.1.451 SCU_EXTREG_SPI_CLK

```
#define SCU_EXTREG_SPI_CLK (SCU_EXTREG_PA_BASE + 0x0104)
```

6.64.1.452 SCU_EXTREG_SPI_CLK_GET_dcsr

```
#define SCU_EXTREG_SPI_CLK_GET_dcsr( ) GET_BITS(SCU_EXTREG_SPI_CLK, 5, 8)
```

6.64.1.453 SCU_EXTREG_SPI_CLK_GET_pd

```
#define SCU_EXTREG_SPI_CLK_GET_pd( ) GET_BIT(SCU_EXTREG_SPI_CLK, 4)
```

6.64.1.454 SCU_EXTREG_SPI_CLK_GET_pu

```
#define SCU_EXTREG_SPI_CLK_GET_pu( ) GET_BIT(SCU_EXTREG_SPI_CLK, 3)
```

6.64.1.455 SCU_EXTREG_SPI_CLK_SET_dcsr

```
#define SCU_EXTREG_SPI_CLK_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_SPI_CLK, val, 5, 8)
```

Kneron Confidential

6.64.1.456 SCU_EXTREG_SPI_CLK_SET_pd

```
#define SCU_EXTREG_SPI_CLK_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_CLK, 4)
```

6.64.1.457 SCU_EXTREG_SPI_CLK_SET_pu

```
#define SCU_EXTREG_SPI_CLK_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_CLK, 3)
```

6.64.1.458 SCU_EXTREG_SPI_CS_N

```
#define SCU_EXTREG_SPI_CS_N (SCU_EXTREG_PA_BASE + 0x0100)
```

6.64.1.459 SCU_EXTREG_SPI_CS_N_GET_dcsr

```
#define SCU_EXTREG_SPI_CS_N_GET_dcsr( ) GET_BITS(SCU_EXTREG_SPI_CS_N, 5, 8)
```

6.64.1.460 SCU_EXTREG_SPI_CS_N_GET_pd

```
#define SCU_EXTREG_SPI_CS_N_GET_pd( ) GET_BIT(SCU_EXTREG_SPI_CS_N, 4)
```

6.64.1.461 SCU_EXTREG_SPI_CS_N_GET_pu

```
#define SCU_EXTREG_SPI_CS_N_GET_pu( ) GET_BIT(SCU_EXTREG_SPI_CS_N, 3)
```

6.64.1.462 SCU_EXTREG_SPI_CS_N_SET_dcsr

```
#define SCU_EXTREG_SPI_CS_N_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_SPI_CS_N, val, 5, 8)
```

Kneron Confidential

6.64.1.463 SCU_EXTREG_SPI_CS_N_SET_pd

```
#define SCU_EXTREG_SPI_CS_N_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_CS_N, 4)
```

6.64.1.464 SCU_EXTREG_SPI_CS_N_SET_pu

```
#define SCU_EXTREG_SPI_CS_N_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_CS_N, 3)
```

6.64.1.465 SCU_EXTREG_SPI_DI

```
#define SCU_EXTREG_SPI_DI (SCU_EXTREG_PA_BASE + 0x010C)
```

6.64.1.466 SCU_EXTREG_SPI_DI_GET_dcsr

```
#define SCU_EXTREG_SPI_DI_GET_dcsr( ) GET_BITS(SCU_EXTREG_SPI_DI, 5, 8)
```

6.64.1.467 SCU_EXTREG_SPI_DI_GET_pd

```
#define SCU_EXTREG_SPI_DI_GET_pd( ) GET_BIT(SCU_EXTREG_SPI_DI, 4)
```

6.64.1.468 SCU_EXTREG_SPI_DI_GET_pu

```
#define SCU_EXTREG_SPI_DI_GET_pu( ) GET_BIT(SCU_EXTREG_SPI_DI, 3)
```

6.64.1.469 SCU_EXTREG_SPI_DI_SET_dcsr

```
#define SCU_EXTREG_SPI_DI_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_SPI_DI, val, 5, 8)
```

Kneron Confidential

6.64.1.470 SCU_EXTREG_SPI_DI_SET_pd

```
#define SCU_EXTREG_SPI_DI_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_DI, 4)
```

6.64.1.471 SCU_EXTREG_SPI_DI_SET_pu

```
#define SCU_EXTREG_SPI_DI_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_DI, 3)
```

6.64.1.472 SCU_EXTREG_SPI_DO

```
#define SCU_EXTREG_SPI_DO (SCU_EXTREG_PA_BASE + 0x0108)
```

6.64.1.473 SCU_EXTREG_SPI_DO_GET_dcsr

```
#define SCU_EXTREG_SPI_DO_GET_dcsr( ) GET_BITS(SCU_EXTREG_SPI_DO, 5, 8)
```

6.64.1.474 SCU_EXTREG_SPI_DO_GET_pd

```
#define SCU_EXTREG_SPI_DO_GET_pd( ) GET_BIT(SCU_EXTREG_SPI_DO, 4)
```

6.64.1.475 SCU_EXTREG_SPI_DO_GET_pu

```
#define SCU_EXTREG_SPI_DO_GET_pu( ) GET_BIT(SCU_EXTREG_SPI_DO, 3)
```

6.64.1.476 SCU_EXTREG_SPI_DO_SET_dcsr

```
#define SCU_EXTREG_SPI_DO_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_SPI_DO, val, 5, 8)
```

Kneron Confidential

6.64.1.477 SCU_EXTREG_SPI_DO_SET_pd

```
#define SCU_EXTREG_SPI_DO_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_DO, 4)
```

6.64.1.478 SCU_EXTREG_SPI_DO_SET_pu

```
#define SCU_EXTREG_SPI_DO_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_DO, 3)
```

6.64.1.479 SCU_EXTREG_SPI_HOLD_N_IOCTRL

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL (SCU_EXTREG_PA_BASE + 0x0114)
```

6.64.1.480 SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_dcsr

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_dcsr( ) GET_BITS(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 5, 8)
```

6.64.1.481 SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_pd

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_pd( ) GET_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 4)
```

6.64.1.482 SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_pu

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL_GET_pu( ) GET_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 3)
```

6.64.1.483 SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_dcsr

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_SPI_HOLD_N_IOCTRL, val, 5, 8)
```

Kneron Confidential

6.64.1.484 SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_pd

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 4)
```

6.64.1.485 SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_pu

```
#define SCU_EXTREG_SPI_HOLD_N_IOCTRL_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_HOLD_N_IOCTRL, 3)
```

6.64.1.486 SCU_EXTREG_SPI_WP_N_IOCTRL

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL (SCU_EXTREG_PA_BASE + 0x0110)
```

6.64.1.487 SCU_EXTREG_SPI_WP_N_IOCTRL_GET_dcsr

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL_GET_dcsr( ) GET_BITS(SCU_EXTREG_SPI_WP_N_IOCTRL, 5, 8)
```

6.64.1.488 SCU_EXTREG_SPI_WP_N_IOCTRL_GET_pd

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL_GET_pd( ) GET_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 4)
```

6.64.1.489 SCU_EXTREG_SPI_WP_N_IOCTRL_GET_pu

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL_GET_pu( ) GET_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 3)
```

6.64.1.490 SCU_EXTREG_SPI_WP_N_IOCTRL_SET_dcsr

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL_SET_dcsr(
              val ) SET_MASKED_BITS(SCU_EXTREG_SPI_WP_N_IOCTRL, val, 5, 8)
```

Kneron Confidential

6.64.1.491 SCU_EXTREG_SPI_WP_N_IOCTRL_SET_pd

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL_SET_pd(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 4)
```

6.64.1.492 SCU_EXTREG_SPI_WP_N_IOCTRL_SET_pu

```
#define SCU_EXTREG_SPI_WP_N_IOCTRL_SET_pu(
              val ) SET_MASKED_BIT(SCU_EXTREG_SPI_WP_N_IOCTRL, 3)
```

6.64.1.493 SCU_EXTREG_SWJ_SWCLKTCK_IOCTRL

```
#define SCU_EXTREG_SWJ_SWCLKTCK_IOCTRL (SCU_EXTREG_PA_BASE + 0x0124)
```

6.64.1.494 SCU_EXTREG_SWJ_SWDITMS_IOCTRL

```
#define SCU_EXTREG_SWJ_SWDITMS_IOCTRL (SCU_EXTREG_PA_BASE + 0x0120)
```

6.64.1.495 SCU_EXTREG_SWJ_TDI_IOCTRL

```
#define SCU_EXTREG_SWJ_TDI_IOCTRL (SCU_EXTREG_PA_BASE + 0x011C)
```

6.64.1.496 SCU_EXTREG_SWJ_TDO_IOCTRL

```
#define SCU_EXTREG_SWJ_TDO_IOCTRL (SCU_EXTREG_PA_BASE + 0x0128)
```

6.64.1.497 SCU_EXTREG_SWJ_TRST_IOCTRL

```
#define SCU_EXTREG_SWJ_TRST_IOCTRL (SCU_EXTREG_PA_BASE + 0x0118)
```

Kneron Confidential

6.64.1.498 SCU_EXTREG_SWRST

```
#define SCU_EXTREG_SWRST (SCU_EXTREG_PA_BASE + 0x004C)
```

6.64.1.499 SCU_EXTREG_SWRST_GET_LCDC_resetn

```
#define SCU_EXTREG_SWRST_GET_LCDC_resetn( ) GET_BIT(SCU_EXTREG_SWRST, 0)
```

6.64.1.500 SCU_EXTREG_SWRST_GET_NPU_resetn

```
#define SCU_EXTREG_SWRST_GET_NPU_resetn( ) GET_BIT(SCU_EXTREG_SWRST, 2)
```

6.64.1.501 SCU_EXTREG_SWRST_GET_PD_NPU_resetn

```
#define SCU_EXTREG_SWRST_GET_PD_NPU_resetn( ) GET_BIT(SCU_EXTREG_SWRST, 1)
```

6.64.1.502 SCU_EXTREG_SWRST_MASK0

```
#define SCU_EXTREG_SWRST_MASK0 (SCU_EXTREG_PA_BASE + 0x0040)
```

6.64.1.503 SCU_EXTREG_SWRST_MASK0_GET_cpu_resetreq_n

```
#define SCU_EXTREG_SWRST_MASK0_GET_cpu_resetreq_n( ) GET_BIT(SCU_EXTREG_SWRST_MASK0, 0)
```

6.64.1.504 SCU_EXTREG_SWRST_MASK0_SET_cpu_resetreq_n

```
#define SCU_EXTREG_SWRST_MASK0_SET_cpu_resetreq_n(
            val ) SET_MASKED_BIT(SCU_EXTREG_SWRST_MASK0, val, 0)
```

Kneron Confidential

6.64.1.505 SCU_EXTREG_SWRST_MASK1

```
#define SCU_EXTREG_SWRST_MASK1 (SCU_EXTREG_PA_BASE + 0x0044)
```

6.64.1.506 SCU_EXTREG_SWRST_MASK1_AResetn_u_FTLCDC210

```
#define SCU_EXTREG_SWRST_MASK1_AResetn_u_FTLCDC210 BIT20
```

6.64.1.507 SCU_EXTREG_SWRST_MASK1_GET_lcm_reset_n

```
#define SCU_EXTREG_SWRST_MASK1_GET_lcm_reset_n( ) GET_BIT(SCU_EXTREG_PLL5_SETTING, 23)
```

6.64.1.508 SCU_EXTREG_SWRST_MASK1_LC_RSTn_FTLCDC210

```
#define SCU_EXTREG_SWRST_MASK1_LC_RSTn_FTLCDC210 BIT16
```

6.64.1.509 SCU_EXTREG_SWRST_MASK1_LC_SCALER_RSTn_FTLCDC210

```
#define SCU_EXTREG_SWRST_MASK1_LC_SCALER_RSTn_FTLCDC210 BIT17
```

6.64.1.510 SCU_EXTREG_SWRST_MASK1_lcm_reset_n

```
#define SCU_EXTREG_SWRST_MASK1_lcm_reset_n BIT23
```

6.64.1.511 SCU_EXTREG_SWRST_MASK1_PRESETn_u_FTLCDC210

```
#define SCU_EXTREG_SWRST_MASK1_PRESETn_u_FTLCDC210 BIT19
```

6.64.1.512 SCU_EXTREG_SWRST_MASK1_SET_lcm_reset_n

```
#define SCU_EXTREG_SWRST_MASK1_SET_lcm_reset_n(
             val ) SET_MASKED_BIT(SCU_EXTREG_PLL5_SETTING, val, 23)
```

Kneron Confidential

6.64.1.513 SCU_EXTREG_SWRST_MASK1_TV_RSTn_FTLCDC210

```
#define SCU_EXTREG_SWRST_MASK1_TV_RSTn_FTLCDC210 BIT18
```

6.64.1.514 SCU_EXTREG_SWRST_MASK2

```
#define SCU_EXTREG_SWRST_MASK2 (SCU_EXTREG_PA_BASE + 0x0048)
```

6.64.1.515 SCU_EXTREG_SWRST_SET_LCDC_resetn

```
#define SCU_EXTREG_SWRST_SET_LCDC_resetn(
              val ) SET_MASKED_BIT(SCU_EXTREG_SWRST, val, 0)
```

6.64.1.516 SCU_EXTREG_SWRST_SET_NPU_resetn

```
#define SCU_EXTREG_SWRST_SET_NPU_resetn(
              val ) SET_MASKED_BIT(SCU_EXTREG_SWRST, val, 2)
```

6.64.1.517 SCU_EXTREG_SWRST_SET_PD_NPU_resetn

```
#define SCU_EXTREG_SWRST_SET_PD_NPU_resetn(
              val ) SET_MASKED_BIT(SCU_EXTREG_SWRST, val, 1)
```

6.64.1.518 SCU_EXTREG_UART0_RX_IOCTRL

```
#define SCU_EXTREG_UART0_RX_IOCTRL (SCU_EXTREG_PA_BASE + 0x0194)
```

6.64.1.519 SCU_EXTREG_UART0_TX_IOCTRL

```
#define SCU_EXTREG_UART0_TX_IOCTRL (SCU_EXTREG_PA_BASE + 0x0198)
```

Kneron Confidential

6.64.1.520 SCU_EXTREG_USB_OTG_CTRL

```
#define SCU_EXTREG_USB_OTG_CTRL (SCU_EXTREG_PA_BASE + 0x008C)
```

6.64.1.521 SCU_EXTREG_USB_OTG_CTRL_GET_EXTCTRL_SUSPENDM

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_EXTCTRL_SUSPENDM( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 8)
```

6.64.1.522 SCU_EXTREG_USB_OTG_CTRL_GET_l1_wakeup

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_l1_wakeup( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 5)
```

6.64.1.523 SCU_EXTREG_USB_OTG_CTRL_GET_OSCOUTEN

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_OSCOUTEN( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 4)
```

6.64.1.524 SCU_EXTREG_USB_OTG_CTRL_GET_OUTCLKSEL

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_OUTCLKSEL( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 0)
```

6.64.1.525 SCU_EXTREG_USB_OTG_CTRL_GET_PLLALIV

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_PLLALIV( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 3)
```

6.64.1.526 SCU_EXTREG_USB_OTG_CTRL_GET_u_iddig

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_u_iddig( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 7)
```

6.64.1.527 SCU_EXTREG_USB_OTG_CTRL_GET_wakeup

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_wakeup( ) GET_BIT(SCU_EXTREG_USB_OTG_CTRL, 6)
```

Kneron Confidential

6.64.1.528 SCU_EXTREG_USB_OTG_CTRL_GET_XTLSEL

```
#define SCU_EXTREG_USB_OTG_CTRL_GET_XTLSEL( ) GET_BITS(SCU_EXTREG_USB_OTG_CTRL, 1, 2)
```

6.64.1.529 SCU_EXTREG_USB_OTG_CTRL_SET_EXTCTRL_SUSPENDM

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_EXTCTRL_SUSPENDM(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 8)
```

6.64.1.530 SCU_EXTREG_USB_OTG_CTRL_SET_l1_wakeup

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_l1_wakeup(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 5)
```

6.64.1.531 SCU_EXTREG_USB_OTG_CTRL_SET_OSCOUTEN

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_OSCOUTEN(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 4)
```

6.64.1.532 SCU_EXTREG_USB_OTG_CTRL_SET_OUTCLKSEL

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_OUTCLKSEL(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 0)
```

6.64.1.533 SCU_EXTREG_USB_OTG_CTRL_SET_PLLALIV

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_PLLALIV(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 3)
```

6.64.1.534 SCU_EXTREG_USB_OTG_CTRL_SET_u_iddig

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_u_iddig(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 7)
```

Kneron Confidential

6.64.1.535 SCU_EXTREG_USB_OTG_CTRL_SET_wakeup

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_wakeup(
              val ) SET_MASKED_BIT(SCU_EXTREG_USB_OTG_CTRL, val, 6)
```

6.64.1.536 SCU_EXTREG_USB_OTG_CTRL_SET_XTLSEL

```
#define SCU_EXTREG_USB_OTG_CTRL_SET_XTLSEL(
              val ) SET_MASKED_BITS(SCU_EXTREG_USB_OTG_CTRL, val, 1, 2)
```

6.64.2 Enumeration Type Documentation

6.64.2.1 anonymous enum

```
anonymous enum
```

Enumerator

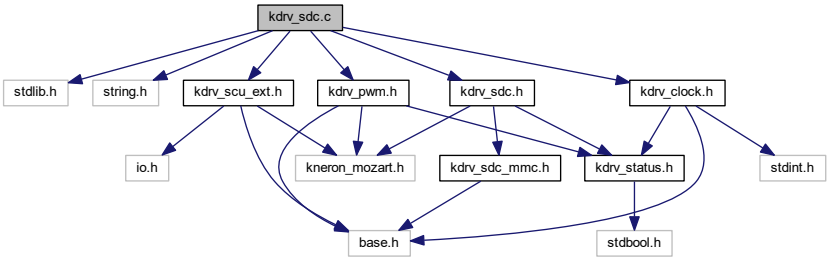
PINMUX_MODE0	
PINMUX_MODE1	
PINMUX_MODE2	
PINMUX_MODE3	
PINMUX_MODE4	
PINMUX_MODE5	
PINMUX_MODE6	
PINMUX_MODE7	

Kneron Confidential

6.65 kdrv_sdc.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include "kdrv_sdc.h"
#include "kdrv_clock.h"
#include "kdrv_scu_ext.h"
#include "kdrv_pwm.h"
```

Include dependency graph for kdrv_sdc.c:



Macros

- #define kmdw_dbg_printf(fmt, ...)
- #define FTSDC021_BASE_CLOCK 100 /∗ 100MHz ∗/
- #define SDC_AHB_CHANNEL 0
- #define USE_PIO

Typedefs

- typedef signed long clock_t

Functions

- kdrv_status_t kdrv_sdc_initialize (void)

  kdrv_sdc_initialize, initail sd/emmc card interface
- kdrv_status_t kdrv_sdc_uninitialize (void)

  kdrv_sdc_uninitialize, uninitail sd card interface and resource
- kdrv_status_t kdrv_sdc_dev_scan (void)

  kdrv_sdc_dev_scan() scan sd/mmc memory card
- kdrv_sdc_res_t ∗ kdrv_sdc_get_dev (void)

  kdrv_sdc_get_dev() get device infromation
- kdrv_status_t kdrv_sdc_read (uint8_t ∗buf, uint32_t sd_offset, uint32_t size)

  kdrv_sdc_read read data from sd/mmc card
- kdrv_status_t kdrv_sdc_write (uint8_t ∗buf, uint32_t sd_offset, uint32_t size)

  kdrv_sdc_write write data from sd/mmc card

Variables

- kdrv_sdc_res_t sdc0_res
- uint32_t sdma_bound_mask
- kdrv_sdc_adma2desc_table_t sdc_adma2_desc_table [ADMA2_NUM_OF_LINES]
- volatile uint8_t err_recover = 0
- uint32_t rd_bl_len
- uint32_t wr_bl_len

Kneron Confidential

6.65.1 Macro Definition Documentation

6.65.1.1 FTSDC021_BASE_CLOCK

```
#define FTSDC021_BASE_CLOCK 100 /* 100MHz */
```

6.65.1.2 kmdw_dbg_printf

```
#define kmdw_dbg_printf(
              fmt,
              ...  )
```

Kneron Peripheral API - SDC

Copyright (C) 2019 Kneron, Inc. All rights reserved.

6.65.1.3 SDC_AHB_CHANNEL

```
#define SDC_AHB_CHANNEL 0
```

6.65.1.4 USE_PIO

```
#define USE_PIO
```

6.65.2 Typedef Documentation

6.65.2.1 clock_t

```
typedef signed long clock_t
```

6.65.3 Variable Documentation

Kneron Confidential

6.65.3.1 err_recover

```
volatile uint8_t err_recover = 0
```

6.65.3.2 rd_bl_len

```
uint32_t rd_bl_len
```

6.65.3.3 sdc0_res

```
kdrv_sdc_res_t sdc0_res
```

Initial value:

```
= {
    (kdrv_sdc_reg_t *)SDC_FTSDC021_PA_BASE,
    NULL,
    NULL,
    INFINITE_NO,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
}
```

6.65.3.4 sdc_adma2_desc_table

```
kdrv_sdc_adma2desc_table_t sdc_adma2_desc_table[ADMA2_NUM_OF_LINES]
```

6.65.3.5 sdma_bound_mask

```
uint32_t sdma_bound_mask
```

6.65.3.6 wr_bl_len

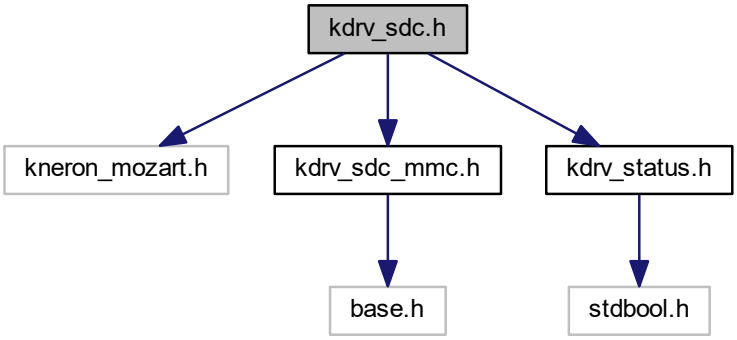
```
uint32_t wr_bl_len
```

Kneron Confidential

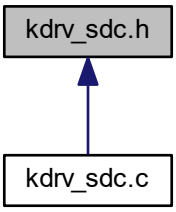
6.66 kdrv_sdc.h File Reference

```
#include "kneron_mozart.h"
#include "kdrv_sdc_mmc.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_sdc.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_sdc_reg_t
- struct kdrv_sdc_flow_info_t
- struct kdrv_sd_status_t
- struct kdrv_sdc_csd_v1_t
- struct kdrv_sdc_csd_v2_t
- struct kdrv_sdc_sd_scr_t
- struct kdrv_sdc_sdcard_info_t
- struct kdrv_sdc_sd_host_t
- struct kdrv_sdc_adma2desc_table_t
- struct kdrv_sdc_res_t

Macros

- #define ADMA2_NUM_OF_LINES 64
- #define CARD_TYPE_UNKNOWN 0
- #define MEMORY_CARD_TYPE_SD 1
- #define MEMORY_CARD_TYPE_MMC 2
- #define SDIO_TYPE_CARD 3
- #define MEMORY_SDIO_COMBO 4
- #define SCR_LENGTH 8
- #define SD_STATUS_LENGTH 64
- #define EXT_CSD_LENGTH 512
- #define SDHCI_SCR_SUPPORT_4BIT_BUS 0x4
- #define SDHCI_SCR_SUPPORT_1BIT_BUS 0x1
- #define WAIT_CMD_COMPLETE BIT(0)
- #define WAIT_TRANS_COMPLETE BIT(1)
- #define WAIT_DMA_INTR BIT(2)
- #define WAIT_BLOCK_GAP BIT(3)
- #define KDRV_SDC_BASE SDC_FTSDC021_PA_BASE
- #define SDHCI_TXMODE_DMA_EN BIT(0)
- #define SDHCI_TXMODE_BLKCNT_EN BIT(1)
- #define SDHCI_TXMODE_AUTOCMD12_EN BIT(2)
- #define SDHCI_TXMODE_AUTOCMD23_EN (2 << 2)
- #define SDHCI_TXMODE_READ_DIRECTION BIT(4)
- #define SDHCI_TXMODE_WRITE_DIRECTION (0 << 4)
- #define SDHCI_TXMODE_MULTI_SEL BIT(5)
- #define SDHCI_CMD_IDX_SHIFT 0x08
- #define SDHCI_CMD_TYPE_SHIFT 0x06
- #define SDHCI_CMD_DATA_PRESEL_SHIFT 0x05
- #define SDHCI_CMD_NO_RESPONSE 0x00
- #define SDHCI_CMD_RTYPE_R2 0x09
- #define SDHCI_CMD_RTYPE_R3R4 0x02
- #define SDHCI_CMD_RTYPE_R1R5R6R7 0x1A
- #define SDHCI_CMD_RTYPE_R1BR5B 0x1B
- #define SDHCI_CMD_TYPE_NORMAL 0x00
- #define SDHCI_CMD_TYPE_SUSPEND 0x01
- #define SDHCI_CMD_TYPE_RESUME 0x02
- #define SDHCI_CMD_TYPE_ABORT 0x03
- #define SDHCI_CMD_DATA_PRESENT 0x01
- #define SDHCI_REG_DATA_PORT 0x20
- #define SDHCI_REG_PRE_STATE 0x24
- #define SDHCI_STS_CMD_INHIBIT BIT(0)
- #define SDHCI_STS_CMD_DAT_INHIBIT BIT(1)
- #define SDHCI_STS_DAT_LINE_ACT BIT(2)
- #define SDHCI_STS_WRITE_TRAN_ACT BIT(8)
- #define SDHCI_STS_READ_TRAN_ACT BIT(9)
- #define SDHCI_STS_BUFF_WRITE BIT(10)
- #define SDHCI_STS_BUFF_READ BIT(11)
- #define SDHCI_STS_CARD_INSERT BIT(16)
- #define SDHCI_STS_CARD_STABLE BIT(17)
- #define SDHCI_STS_CARD_WP BIT(19)
- #define SDHCI_STS_DAT_LINE_LEVEL (0xF << 20)
- #define SDHCI_STS_CMD_LINE_LEVEL BIT(24)
- #define SDHCI_REG_HC 0x28
- #define SDHCI_HC_LED_ON BIT(0)
- #define SDHCI_HC_BUS_WIDTH_4BIT BIT(1)

Kneron Confidential

- #define SDHCI_HC_HI_SPEED BIT(2)
- #define SDHCI_HC_USE_ADMA2 BIT(3)
- #define SDHCI_HC_BUS_WIDTH_8BIT BIT(5)
- #define SDHCI_HC_CARD_DETECT_TEST BIT(6)
- #define SDHCI_HC_CARD_DETECT_SIGNAL BIT(7)
- #define SDHCI_POWER_ON BIT(0)
- #define SDHCI_POWER_180 (5 << 1)
- #define SDHCI_POWER_300 (6 << 1)
- #define SDHCI_POWER_330 (7 << 1)
- #define SDHCI_STOP_AT_BLOCK_GAP_REQ BIT(0)
- #define SDHCI_CONTINUE_REQ BIT(1)
- #define SDHCI_READ_WAIT_CTL BIT(2)
- #define SDHCI_INT_AT_BLOCK_GAP BIT(3)
- #define SDHCI_REG_CLK_CTRL 0x2C
- #define SDHCI_CLK_CTRL_LOW_CLK_SEL_SHIFT 8
- #define SDHCI_CLK_CTRL_UP_CLK_SEL_SHIFT 6
- #define SDHCI_CLK_CTRL_INTERNALCLK_EN BIT(0)
- #define SDHCI_CLK_CTRL_INTERNALCLK_STABLE BIT(1)
- #define SDHCI_CLK_CTRL_SDCLK_EN BIT(2)
- #define SDHCI_CLK_CTRL_CLK_GEN_SEL_PRO BIT(5)
- #define SDHCI_SOFTRST_ALL BIT(0)
- #define SDHCI_SOFTRST_CMD BIT(1)
- #define SDHCI_SOFTRST_DAT BIT(2)
- #define SDHCI_REG_INTR_STATE 0x30
- #define SDHCI_INTR_STS_ERR BIT(15)
- #define SDHCI_INTR_STS_CARD_INTR BIT(8)
- #define SDHCI_INTR_STS_CARD_REMOVE BIT(7)
- #define SDHCI_INTR_STS_CARD_INSERT BIT(6)
- #define SDHCI_INTR_STS_BUFF_READ_READY BIT(5)
- #define SDHCI_INTR_STS_BUFF_WRITE_READY BIT(4)
- #define SDHCI_INTR_STS_DMA BIT(3)
- #define SDHCI_INTR_STS_BLKGAP BIT(2)
- #define SDHCI_INTR_STS_TXR_COMPLETE BIT(1)
- #define SDHCI_INTR_STS_CMD_COMPLETE BIT(0) /∗ CMD completed, CMD12/CMD23 will not generate this command ∗/
- #define SDHCI_INTR_ERR_TUNING BIT(10)
- #define SDHCI_INTR_ERR_ADMA BIT(9)
- #define SDHCI_INTR_ERR_AUTOCMD BIT(8)
- #define SDHCI_INTR_ERR_CURR_LIMIT BIT(7)
- #define SDHCI_INTR_ERR_DATA_ENDBIT BIT(6)
- #define SDHCI_INTR_ERR_DATA_CRC BIT(5)
- #define SDHCI_INTR_ERR_DATA_TIMEOUT BIT(4)
- #define SDHCI_INTR_ERR_CMD_INDEX BIT(3)
- #define SDHCI_INTR_ERR_CMD_ENDBIT BIT(2)
- #define SDHCI_INTR_ERR_CMD_CRC BIT(1)
- #define SDHCI_INTR_ERR_CMD_TIMEOUT BIT(0)
- #define SDHCI_INTR_ERR_CMD_LINE (SDHCI_INTR_ERR_CMD_INDEX | SDHCI_INTR_ERR_CMD_ENDBIT | SDHCI_INTR_ERR_CMD_CRC | SDHCI_INTR_ERR_CMD_TIMEOUT)
- #define SDHCI_INTR_ERR_DAT_LINE (SDHCI_INTR_ERR_DATA_ENDBIT | SDHCI_INTR_ERR_DATA_CRC | SDHCI_INTR_ERR_DATA_TIMEOUT)
- #define SDHCI_INTR_EN_ALL (0x10FF)
- #define SDHCI_ERR_EN_ALL (0xF7FF)
- #define SDHCI_INTR_SIG_EN (SDHCI_INTR_STS_CARD_REMOVE | SDHCI_INTR_STS_CARD_INSERT | SDHCI_INTR_STS_CMD_COMPLETE | SDHCI_INTR_STS_TXR_COMPLETE)
- #define SDHCI_INTR_SIGN_EN_SDMA (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_DMA | SDHCI_INTR_STS_BLKGAP)

Kneron Confidential

- #define SDHCI_INTR_SIGN_EN_ADMA (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_DMA)
- #define SDHCI_INTR_SIGN_EN_PIO (SDHCI_INTR_SIG_EN | SDHCI_INTR_STS_BLKGAP)
- #define SDHCI_ERR_SIG_EN_ALL (0xF3FF)
- #define SDHCI_AUTOCMD12_ERR_NOT_EXECUTED BIT(0)
- #define SDHCI_AUTOCMD12_ERR_TIMEOUT BIT(1)
- #define SDHCI_AUTOCMD12_ERR_CRC BIT(2)
- #define SDHCI_AUTOCMD12_ERR_END_BIT BIT(3)
- #define SDHCI_AUTOCMD12_ERR_INDEX BIT(4)
- #define SDHCI_AUTOCMD12_ERR_CMD_NOT_ISSUE BIT(7)
- #define SDHCI_REG_HOST_CTRL2 0x3E
- #define SDHCI_PRESET_VAL_EN BIT(15)
- #define SDHCI_ASYNC_INT_EN BIT(14)
- #define SDHCI_SMPL_CLCK_SELECT BIT(7)
- #define SDHCI_EXECUTE_TUNING BIT(6) /∗ Write 1 Auto clear ∗/
- #define SDHCI_DRV_TYPE_MASK BIT(4)
- #define SDHCI_DRV_TYPE_SHIFT 4
- #define SDHCI_DRV_TYPEB 0
- #define SDHCI_DRV_TYPEA 1
- #define SDHCI_DRV_TYPEC 2
- #define SDHCI_DRV_TYPED 3
- #define SDHCI_18V_SIGNAL BIT(3)
- #define SDHCI_UHS_MODE_MASK (7 << 0)
- #define SDHCI_SDR12 0
- #define SDHCI_SDR25 1
- #define SDHCI_SDR50 2
- #define SDHCI_SDR104 3
- #define SDHCI_DDR50 4
- #define SDHCI_CAP_VOLTAGE_33V BIT(24)
- #define SDHCI_CAP_VOLTAGE_30V BIT(25)
- #define SDHCI_CAP_VOLTAGE_18V BIT(26)
- #define SDHCI_CAP_FIFO_DEPTH_16BYTE (0 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_32BYTE (1 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_64BYTE (2 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_512BYTE (3 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_1024BYTE (4 << 29)
- #define SDHCI_CAP_FIFO_DEPTH_2048BYTE (5 << 29)
- #define SDHCI_SUPPORT_SDR50 BIT(0)
- #define SDHCI_SUPPORT_SDR104 BIT(1)
- #define SDHCI_SUPPORT_DDR50 BIT(2)
- #define SDHCI_SUPPORT_DRV_TYPEA BIT(4)
- #define SDHCI_SUPPORT_DRV_TYPEC BIT(5)
- #define SDHCI_SUPPORT_DRV_TYPED BIT(6)
- #define SDHCI_RETUNING_TIME_MAS 0xF
- #define SDHCI_RETUNING_TIME_SHIFT 8
- #define SDHCI_SDR50_TUNING BIT(13)
- #define SDCHI_RETUNING_MODE_MASK 0x3
- #define SDHCI_RETUNING_MODE_SHIFT 14
- #define MMC_BOOT_ACK BIT(2)
- #define MMC_BUS_TEST_MODE 0x3
- #define MMC_ALTERNATIVE_BOOT_MODE 0x2
- #define MMC_BOOT_MODE 0x1
- #define NORMAL_MODE 0x0
- #define SDHCI_CMD0_GO_IDLE_STATE 0
- #define SDHCI_CMD1_MMC_SEND_OP_COND 1
- #define SDHCI_CMD2_SEND_ALL_CID 2

Kneron Confidential

- #define SDHCI_CMD3_SEND_RELATIVE_ADDR 3
- #define SDHCI_CMD5_IO_SEND_OP_COND 5
- #define SDHCI_CMD6_SWITCH_FUNC 6
- #define SDHCI_CMD6_SET_BUS_WIDTH 6
- #define SDHCI_CMD7_SELECT_CARD 7
- #define SDHCI_CMD8_SEND_IF_COND 8
- #define SDHCI_CMD8_SEND_EXT_CSD 8
- #define SDHCI_CMD9_SEND_CSD 9
- #define SDHCI_CMD10_SEND_CID 10
- #define SDHCI_CMD11_VOLTAGE_SWITCH 11
- #define SDHCI_CMD12_STOP_TRANS 12
- #define SDHCI_CMD13_SEND_STATUS 13
- #define SDHCI_CMD13_SD_STATUS 13
- #define SDHCI_CMD16_SET_BLOCKLEN 16
- #define SDHCI_CMD17_READ_SINGLE_BLOCK 17
- #define SDHCI_CMD18_READ_MULTI_BLOCK 18
- #define SDHCI_CMD19_SEND_TUNE_BLOCK 19
- #define SDHCI_CMD23_SET_WR_BLOCK_CNT 23
- #define SDHCI_CMD24_WRITE_BLOCK 24
- #define SDHCI_CMD25_WRITE_MULTI_BLOCK 25
- #define SDHCI_CMD32_ERASE_WR_BLK_START 32
- #define SDHCI_CMD33_ERASE_WR_BLK_END 33
- #define SDHCI_CMD35_ERASE_GROUP_START 35
- #define SDHCI_CMD36_ERASE_GROUP_END 36
- #define SDHCI_CMD38_ERASE 38
- #define SDHCI_CMD41_SD_SEND_OP_COND 41
- #define SDHCI_CMD43_GET_MKB 43
- #define SDHCI_CMD44_GET_MID 44
- #define SDHCI_CMD45_CER_RN1 45
- #define SDHCI_CMD46_CER_RN2 46
- #define SDHCI_CMD47_CER_RES2 47
- #define SDHCI_CMD48_CER_RES1 48
- #define SDHCI_CMD51_SEND_SCR 51
- #define SDHCI_CMD52_IO_RW_DIRECT 52
- #define SDHCI_CMD53_IO_RW_EXTENDED 53
- #define SDHCI_CMD55_APP 55
- #define SDHCI_CMD56_GEN 56
- #define SDHCI_CMD8_SEND_IF_COND_ARGU 0x1AA
- #define SDHCI_CMD41_SD_SEND_OP_COND_HCS_ARGU 0xC0FF8000
- #define SDHCI_CMD41_SD_SEND_OP_COND_ARGU 0x00FF8000
- #define SDHCI_CMD1_MMC_SEND_OP_COND_BYTE_MODE 0x80FF8000
- #define SDHCI_CMD1_MMC_SEND_OP_COND_SECTOR_MODE 0xC0FF8000
- #define CMD_RETRY_CNT 5
- #define SDHCI_TIMEOUT 0xFFF
- #define SD_CMD52_RW_in_W 0x80000000
- #define SD_CMD52_RW_in_R 0x00000000
- #define SD_CMD52_RAW 0x08000000
- #define SD_CMD52_no_RAW 0x00000000
- #define SD_CMD52_FUNC(Num) (Num << 28)
- #define SD_CMD52_Reg_Addr(Addr) (Addr << 9)
- #define SD_CMD53_RW_in_W 0x80000000
- #define SD_CMD53_RW_in_R 0x00000000
- #define SD_CMD53_FUNC(Num) (Num << 28)
- #define SD_CMD53_Block_Mode 0x08000000
- #define SD_CMD53_Byte_Mode 0x00000000

Kneron Confidential

- #define SD_CMD53_OP_inc 0x04000000
- #define SD_CMD53_OP_fix 0x00000000
- #define SD_CMD53_Reg_Addr(Addr) (Addr << 9)
- #define SD_STATUS_OUT_OF_RANGE 0x80000000
- #define SD_STATUS_ADDRESS_ERROR BIT(30)
- #define SD_STATUS_BLOCK_LEN_ERROR BIT(29)
- #define SD_STATUS_ERASE_SEQ_ERROR BIT(28)
- #define SD_STATUS_ERASE_PARAM BIT(27)
- #define SD_STATUS_WP_VIOLATION BIT(26)
- #define SD_STATUS_CARD_IS_LOCK BIT(25)
- #define SD_STATUS_LOCK_UNLOCK_FAILED BIT(24)
- #define SD_STATUS_COM_CRC_ERROR BIT(23)
- #define SD_STATUS_ILLEGAL_COMMAND BIT(22)
- #define SD_STATUS_CARD_ECC_FAILED BIT(21)
- #define SD_STATUS_CC_ERROR BIT(20)
- #define SD_STATUS_ERROR BIT(19)
- #define SD_STATUS_UNDERRUN BIT(18)
- #define SD_STATUS_OVERRUN BIT(17)
- #define SD_STATUS_CSD_OVERWRITE BIT(16)
- #define SD_STATUS_WP_ERASE_SKIP BIT(15)
- #define SD_STATUS_CARD_ECC_DISABLE BIT(14)
- #define SD_STATUS_ERASE_RESET BIT(13)
- #define SD_STATUS_CURRENT_STATE (0xF << 9)
- #define SD_STATUS_READY_FOR_DATA BIT(8)
- #define MMC_STATUS_SWITCH_ERROR BIT(7)
- #define SD_STATUS_APP_CMD BIT(5)
- #define SD_STATUS_AKE_SEQ_ERROR BIT(3)
- #define SD_STATUS_ERROR_BITS
- #define SDHCI_1BIT_BUS_WIDTH 0x0
- #define SDHCI_4BIT_BUS_WIDTH 0x2
- #define ADMA2_ENTRY_VALID BIT(0)
- #define ADMA2_ENTRY_END BIT(1)
- #define ADMA2_ENTRY_INT BIT(2)
- #define ADMA2_NOP (0 << 4)
- #define ADMA2_SET (1 << 4)
- #define ADMA2_TRAN (2 << 4)
- #define ADMA2_LINK (3 << 4)

Enumerations

- enum kdrv_sdc_infinite_test_e { INFINITE_NO = 0, INFINITE_MODE_1, INFINITE_MODE_2 }
- enum kdrv_sdc_transfer_act_e { WRITE = 0, READ }
- enum kdrv_sdc_transfer_type_e {
  ADMA = 0, SDMA, PIO, EDMA,
  TRANS_UNKNOWN }
- enum kdrv_sdc_abort_type_e { ABORT_ASYNCHRONOUS = 0, ABORT_SYNCHRONOUS, ABORT_UNDEFINED }
- enum kdrv_sdc_cprm_test_e { CPRM_PROTECT_RW, CPRM_FILESYS, CPRM_UNKNOWN }
- enum kdrv_sdc_bus_speed_e {
  SPEED_DEFAULT = 0, SPEED_SDR25, SPEED_SDR50, SPEED_SDR104,
  SPEED_DDR50, SPEED_RSRV }
- enum kdrv_sdc_card_state_e {
  CUR_STATE_IDLE = 0, CUR_STATE_READY, CUR_STATE_IDENT, CUR_STATE_STBY,
  CUR_STATE_TRAN, CUR_STATE_DATA, CUR_STATE_RCV, CUR_STATE_PRG,
  CUR_STATE_DIS, CUR_STATE_RSV }

Kneron Confidential

Functions

- kdrv_status_t kdrv_sdc_initialize (void)

  kdrv_sdc_initialize, initail sd/emmc card interface
- kdrv_status_t kdrv_sdc_uninitialize (void)

  kdrv_sdc_uninitialize, uninitail sd/emmc card interface and resource
- kdrv_status_t kdrv_sdc_dev_scan (void)

  kdrv_sdc_dev_scan() scan sd/mmc memory card
- kdrv_sdc_res_t * kdrv_sdc_get_dev (void)

  kdrv_sdc_get_dev() get device structure
- kdrv_status_t kdrv_sdc_read (uint8_t *buf, uint32_t sd_offset, uint32_t size)

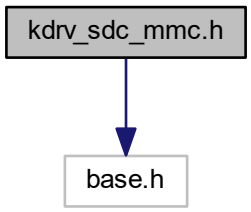
  kdrv_sdc_read read data from sd/mmc card
- kdrv_status_t kdrv_sdc_write (uint8_t *buf, uint32_t sd_offset, uint32_t size)

  kdrv_sdc_write write data from sd/mmc card

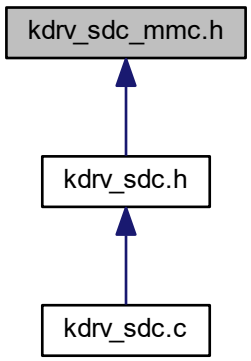
6.67 kdrv_sdc_mmc.h File Reference

```
#include "base.h"
```

Include dependency graph for kdrv_sdc_mmc.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_sdc_mmc_csd_t
- struct kdrv_sdc_mmc_ext_csd_t

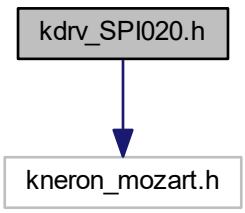
Macros

- #define SDHCI_MMC_SWITCH 6
- #define SDHCI_MMC_VENDOR_CMD 62
- #define EXT_CSD_CMD_SET_NORMAL (1<<0)
- #define EXT_CSD_CMD_SET_SECURE (1<<1)
- #define EXT_CSD_CMD_SET_CPSECURE (1<<2)
- #define EXT_CSD_PARTITION_SETTING_COMPLETED 156
- #define EXT_CSD_PARTITION_CONF 179
- #define EXT_CSD_BUS_WIDTH 183 /* R/W */
- #define EXT_CSD_HS_TIMING 185 /* R/W */
- #define EXT_CSD_CARD_TYPE 196 /* RO */
- #define EXT_CSD_SEC_CNT 212 /* RO, 4 bytes */
- #define EXT_CSD_BOOT_SIZE_MULT 226
- #define EXT_CSD_CMD_SET 0x0
- #define EXT_CSD_SET_BIT 0x1
- #define EXT_CSD_CLR_BYTE 0x2
- #define EXT_CSD_WRITE_BYTE 0x3
- #define EXT_CSD_BUS_8BIT 0x2
- #define EXT_CSD_BUS_4BIT 0x1
- #define EXT_CSD_BUS_1BIT 0x0
- #define MMC_CMD6_ACCESS_MODE(x) (uint32_t)(x << 24)
- #define MMC_CMD6_INDEX(x) (uint32_t)(x << 16)
- #define MMC_CMD6_VALUE(x) (uint32_t)(x << 8)
- #define MMC_CMD6_CMD_SET(x) (uint32_t)(x)
- #define MMC_CARD_BUSY 0x80000000 /* Card Power up status bit */

6.68 kdrv_SPI020.h File Reference

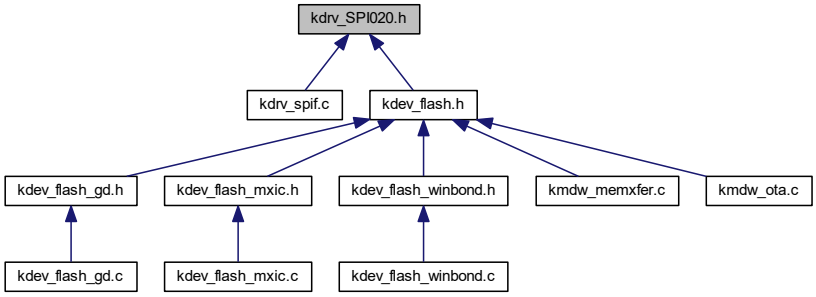
```
#include "kneron_mozart.h"
```

Include dependency graph for kdrv_SPI020.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct spi_flash_t

Macros

- #define FLASH_64K 0x10000
- #define FLASH_PAGE_SIZE 256
- #define FLASH_NORMAL 0x00
- #define FLASH_DTR_RW 0x01
- #define FLASH_DUAL_READ 0x02
- #define FLASH_DMA_READ 0x04
- #define FLASH_DMA_WRITE 0x08
- #define FLASH_IO_RW 0x10
- #define FLASH_BYTE_MODE 0x20
- #define FLASH_QUAD_RW 0x40
- #define FLASH_FAST_READ 0x80
- #define FLASH_OP_MODE FLASH_NORMAL
- #define FLASH_CTL_SPI010 0x01
- #define FLASH_CTL_SPI020 0x02
- #define FLASH_4BYTES_CMD_EN 0x01
- #define SPI_CLK_DIVIDER_2 0x00
- #define SPI_CLK_DIVIDER_4 0x01
- #define SPI_CLK_DIVIDER_6 0x02
- #define SPI_CLK_DIVIDER_8 0x03
- #define SPI020_CE_0 0x0000
- #define SPI020_CE_1 0x0100
- #define SPI020_CE_2 0x0200
- #define SPI020_CE_3 0x0300
- #define SPI020_CE_VALUE SPI020_CE_0
- #define SPI020_INTR_CFG 0x00000000
- #define SPI020REG_CMD0 (SPI_FTSPI020_PA_BASE+0x00)
- #define SPI020REG_CMD1 (SPI_FTSPI020_PA_BASE+0x04)
- #define SPI020REG_CMD2 (SPI_FTSPI020_PA_BASE+0x08)
- #define SPI020REG_CMD3 (SPI_FTSPI020_PA_BASE+0x0C)
- #define SPI020REG_CONTROL (SPI_FTSPI020_PA_BASE+0x10)
- #define SPI020REG_ACTIMER (SPI_FTSPI020_PA_BASE+0x14)

Kneron Confidential

- #define SPI020REG_STATUS (SPI_FTSPI020_PA_BASE+0x18)
- #define SPI020REG_INTERRUPT (SPI_FTSPI020_PA_BASE+0x20)
- #define SPI020REG_INTR_ST (SPI_FTSPI020_PA_BASE+0x24)
- #define SPI020REG_READ_ST (SPI_FTSPI020_PA_BASE+0x28)
- #define SPI020REG_VERSION (SPI_FTSPI020_PA_BASE+0x50)
- #define SPI020REG_FEATURE (SPI_FTSPI020_PA_BASE+0x54)
- #define SPI020REG_SCKINDLY (SPI_FTSPI020_PA_BASE+0x58)
- #define SPI020REG_DATAPORT (SPI_FTSPI020_PA_BASE+0x100)
- #define SPI020_ABORT BIT8
- #define SPI020_CLK_MODE BIT4
- #define SPI020_CLK_DIVIDER (BIT0|BIT1)
- #define SPI020_RXFIFO_READY BIT1
- #define SPI020_TXFIFO_READY BIT0
- #define SPI020_cmd_cmplt_intr_en BIT1
- #define SPI020_DMA_EN BIT0
- #define SPI020_CMD_CMPLT BIT0
- #define SPI020_RX_DEPTH 0xFF00
- #define SPI020_TX_DEPTH 0x00FF
- #define SPI020_B7_CMD0 0x0
- #define SPI020_B7_CMD1 0x01000000
- #define SPI020_B7_CMD2 0x0
- #define SPI020_B7_CMD3 (0xB7000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_13_CMD0 0x0
- #define SPI020_13_CMD1 0x01000004
- #define SPI020_13_CMD2 0x0
- #define SPI020_13_CMD3 (0x13000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_0C_CMD0 0x0
- #define SPI020_0C_CMD1 0x01080004
- #define SPI020_0C_CMD2 0x0
- #define SPI020_0C_CMD3 (0x0C000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_12_CMD0 0x0
- #define SPI020_12_CMD1 0x01000004
- #define SPI020_12_CMD2 0x0
- #define SPI020_12_CMD3 (0x12000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_34_CMD0 0x0
- #define SPI020_34_CMD1 0x01000004
- #define SPI020_34_CMD2 0x0
- #define SPI020_34_CMD3 (0x34000042|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_21_CMD0 0x0
- #define SPI020_21_CMD1 0x01000004
- #define SPI020_21_CMD2 0x0
- #define SPI020_21_CMD3 (0x21000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_DC_CMD0 0x0
- #define SPI020_DC_CMD1 0x01000004
- #define SPI020_DC_CMD2 0x0
- #define SPI020_DC_CMD3 (0xDC000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_6C_CMD0 0x0
- #define SPI020_6C_CMD1 0x01080004
- #define SPI020_6C_CMD2 0x0
- #define SPI020_6C_CMD3 (0x6C000040|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_BC_CMD0 0x0
- #define SPI020_BC_CMD1 0x01040004
- #define SPI020_BC_CMD2 0x0
- #define SPI020_BC_CMD3 (0xBC000060|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_EC_CMD0 0x0

Kneron Confidential

- #define SPI020_EC_CMD1 0x01060004
- #define SPI020_EC_CMD2 0x0
- #define SPI020_EC_CMD3 (0xEC000080|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_3C_CMD0 0x0
- #define SPI020_3C_CMD1 0x01080004
- #define SPI020_3C_CMD2 0x0
- #define SPI020_3C_CMD3 (0x3C000020|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_34_CMD0 0x0
- #define SPI020_34_CMD1 0x01000004
- #define SPI020_34_CMD2 0x0
- #define SPI020_34_CMD3 (0x34000042|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_E9_CMD0 0x0
- #define SPI020_E9_CMD1 0x01000000
- #define SPI020_E9_CMD2 0x0
- #define SPI020_E9_CMD3 (0xE9000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_9F_CMD0 0x0
- #define SPI020_9F_CMD1 0x01020000
- #define SPI020_9F_CMD2 0x3
- #define SPI020_9F_CMD3 (0x9F000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_AB_CMD0 0x0
- #define SPI020_AB_CMD1 0x01000000
- #define SPI020_AB_CMD2 0x0
- #define SPI020_AB_CMD3 (0xAB000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_B9_CMD0 0x0
- #define SPI020_B9_CMD1 0x01000000
- #define SPI020_B9_CMD2 0x0
- #define SPI020_B9_CMD3 (0xB9000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_05_CMD0 0x0
- #define SPI020_05_CMD1 0x01000000
- #define SPI020_05_CMD2 0x0
- #define SPI020_05_CMD3 (0x05000004|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_05_CMD0_w 0x0
- #define SPI020_05_CMD1_w 0x01000000
- #define SPI020_05_CMD2_w 0x1
- #define SPI020_05_CMD3_w (0x05000008|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_35_CMD0 0x0
- #define SPI020_35_CMD1 0x01000000
- #define SPI020_35_CMD2 0x1
- #define SPI020_35_CMD3 (0x35000008|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_15_CMD0 0x0
- #define SPI020_15_CMD1 0x01000000
- #define SPI020_15_CMD2 0x1
- #define SPI020_15_CMD3 (0x15000008|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_11_CMD0 0x0
- #define SPI020_11_CMD1 0x01000000
- #define SPI020_11_CMD2 0x1
- #define SPI020_11_CMD3 (0x11000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_31_CMD0 0x0
- #define SPI020_31_CMD1 0x01000000
- #define SPI020_31_CMD2 0x1
- #define SPI020_31_CMD3 (0x31000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_03_CMD0 0x0
- #define SPI020_03_CMD1 0x01000003
- #define SPI020_03_CMD2 0x0
- #define SPI020_03_CMD3 (0x03000000|SPI020_CE_VALUE|SPI020_INTR_CFG)

Kneron Confidential

- #define SPI020_0B_CMD0 0x0
- #define SPI020_0B_CMD1 0x01080003
- #define SPI020_0B_CMD2 0x0
- #define SPI020_0B_CMD3 (0x0B000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_0D_CMD0 0x0
- #define SPI020_0D_CMD1 0x01060003
- #define SPI020_0D_CMD2 0x0
- #define SPI020_0D_CMD3 (0x0D000010|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_BD_CMD0 0x0
- #define SPI020_BD_CMD1 0x01080003
- #define SPI020_BD_CMD2 0x0
- #define SPI020_BD_CMD3 (0xBD000070|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_ED_CMD0 0x0
- #define SPI020_ED_CMD1 0x01080003
- #define SPI020_ED_CMD2 0x0
- #define SPI020_ED_CMD3 (0xED000090|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_30_CMD0 0x0
- #define SPI020_30_CMD1 0x01000000
- #define SPI020_30_CMD2 0x0
- #define SPI020_30_CMD3 (0x30000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_06_CMD0 0x0
- #define SPI020_06_CMD1 0x01000000
- #define SPI020_06_CMD2 0x0
- #define SPI020_06_CMD3 (0x06000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_04_CMD0 0x0
- #define SPI020_04_CMD1 0x01000000
- #define SPI020_04_CMD2 0x0
- #define SPI020_04_CMD3 (0x04000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_02_CMD0 0x0
- #define SPI020_02_CMD1 0x01000003
- #define SPI020_02_CMD2 0x0
- #define SPI020_02_CMD3 (0x02000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_32_CMD0 0x0
- #define SPI020_32_CMD1 0x01000003
- #define SPI020_32_CMD2 0x0
- #define SPI020_32_CMD3 (0x32000042|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_C7_CMD0 0x0
- #define SPI020_C7_CMD1 0x01000000
- #define SPI020_C7_CMD2 0x0
- #define SPI020_C7_CMD3 (0xC7000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_60_CMD0 0x0
- #define SPI020_60_CMD1 0x01000000
- #define SPI020_60_CMD2 0x0
- #define SPI020_60_CMD3 (0x60000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_D8_CMD0 0x0
- #define SPI020_D8_CMD1 0x01000003
- #define SPI020_D8_CMD2 0x0
- #define SPI020_D8_CMD3 (0xD8000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_52_CMD0 0x0
- #define SPI020_52_CMD1 0x01000003
- #define SPI020_52_CMD2 0x0
- #define SPI020_52_CMD3 (0x52000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_20_CMD0 0x0
- #define SPI020_20_CMD1 0x01000003
- #define SPI020_20_CMD2 0x0

Kneron Confidential

- #define SPI020_20_CMD3 (0x20000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_3B_CMD0 0x0
- #define SPI020_3B_CMD1 0x01080003
- #define SPI020_3B_CMD2 0x0
- #define SPI020_3B_CMD3 (0x3B000020|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_6B_CMD0 0x0
- #define SPI020_6B_CMD1 0x01080003
- #define SPI020_6B_CMD2 0x0
- #define SPI020_6B_CMD3 (0x6B000040|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_BB_CMD0 0x0
- #define SPI020_BB_CMD1 0x01040003
- #define SPI020_BB_CMD2 0x0
- #define SPI020_BB_CMD3 (0xBB000060|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_EB_CMD0 0x0
- #define SPI020_EB_CMD1 0x01060003
- #define SPI020_EB_CMD2 0x0
- #define SPI020_EB_CMD3 (0xEB000080|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_50_CMD0 0x0
- #define SPI020_50_CMD1 0x01000000
- #define SPI020_50_CMD2 0x0
- #define SPI020_50_CMD3 (0x50000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_01_CMD0 0x0
- #define SPI020_01_CMD1 0x01000000
- #define SPI020_01_CMD2 0x2
- #define SPI020_01_CMD3 (0x01000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_90_CMD0 0x0
- #define SPI020_90_CMD1 0x01000003
- #define SPI020_90_CMD2 0x02
- #define SPI020_90_CMD3 (0x90000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_66_CMD0 0x0
- #define SPI020_66_CMD1 0x01000000
- #define SPI020_66_CMD2 0x00
- #define SPI020_66_CMD3 (0x66000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_66_CMD0_ORG 0x0
- #define SPI020_66_CMD1_ORG 0x01000000
- #define SPI020_66_CMD2_ORG 0x00
- #define SPI020_66_CMD3_ORG (0x66000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
- #define SPI020_99_CMD0 0x0
- #define SPI020_99_CMD1 0x01000000
- #define SPI020_99_CMD2 0x00
- #define SPI020_99_CMD3 (0x99000002|SPI020_CE_VALUE|SPI020_INTR_CFG)

Enumerations

- enum spi_clk_mode_t { SPI_CLK_MODE0 =0, SPI_CLK_MODE3 =0x10 }

6.68.1 Macro Definition Documentation

Kneron Confidential

6.68.1.1 FLASH_4BYTES_CMD_EN

```
#define FLASH_4BYTES_CMD_EN 0x01
```

6.68.1.2 FLASH_64K

```
#define FLASH_64K 0x10000
```

6.68.1.3 FLASH_BYTE_MODE

```
#define FLASH_BYTE_MODE 0x20
```

6.68.1.4 FLASH_CTL_SPI010

```
#define FLASH_CTL_SPI010 0x01
```

6.68.1.5 FLASH_CTL_SPI020

```
#define FLASH_CTL_SPI020 0x02
```

6.68.1.6 FLASH_DMA_READ

```
#define FLASH_DMA_READ 0x04
```

6.68.1.7 FLASH_DMA_WRITE

```
#define FLASH_DMA_WRITE 0x08
```

6.68.1.8 FLASH_DTR_RW

```
#define FLASH_DTR_RW 0x01
```

Kneron Confidential

6.68.1.9 FLASH_DUAL_READ

```
#define FLASH_DUAL_READ 0x02
```

6.68.1.10 FLASH_FAST_READ

```
#define FLASH_FAST_READ 0x80
```

6.68.1.11 FLASH_IO_RW

```
#define FLASH_IO_RW 0x10
```

6.68.1.12 FLASH_NORMAL

```
#define FLASH_NORMAL 0x00
```

6.68.1.13 FLASH_OP_MODE

```
#define FLASH_OP_MODE FLASH_NORMAL
```

6.68.1.14 FLASH_PAGE_SIZE

```
#define FLASH_PAGE_SIZE 256
```

6.68.1.15 FLASH_QUAD_RW

```
#define FLASH_QUAD_RW 0x40
```

6.68.1.16 SPI020_01_CMD0

```
#define SPI020_01_CMD0 0x0
```

Kneron Confidential

6.68.1.17 SPI020_01_CMD1

```
#define SPI020_01_CMD1 0x01000000
```

6.68.1.18 SPI020_01_CMD2

```
#define SPI020_01_CMD2 0x2
```

6.68.1.19 SPI020_01_CMD3

```
#define SPI020_01_CMD3 (0x01000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.20 SPI020_02_CMD0

```
#define SPI020_02_CMD0 0x0
```

6.68.1.21 SPI020_02_CMD1

```
#define SPI020_02_CMD1 0x01000003
```

6.68.1.22 SPI020_02_CMD2

```
#define SPI020_02_CMD2 0x0
```

6.68.1.23 SPI020_02_CMD3

```
#define SPI020_02_CMD3 (0x02000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.24 SPI020_03_CMD0

```
#define SPI020_03_CMD0 0x0
```

Kneron Confidential

6.68.1.25 SPI020_03_CMD1

```
#define SPI020_03_CMD1 0x01000003
```

6.68.1.26 SPI020_03_CMD2

```
#define SPI020_03_CMD2 0x0
```

6.68.1.27 SPI020_03_CMD3

```
#define SPI020_03_CMD3 (0x03000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.28 SPI020_04_CMD0

```
#define SPI020_04_CMD0 0x0
```

6.68.1.29 SPI020_04_CMD1

```
#define SPI020_04_CMD1 0x01000000
```

6.68.1.30 SPI020_04_CMD2

```
#define SPI020_04_CMD2 0x0
```

6.68.1.31 SPI020_04_CMD3

```
#define SPI020_04_CMD3 (0x04000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.32 SPI020_05_CMD0

```
#define SPI020_05_CMD0 0x0
```

Kneron Confidential

6.68.1.33 SPI020_05_CMD0_w

```
#define SPI020_05_CMD0_w 0x0
```

6.68.1.34 SPI020_05_CMD1

```
#define SPI020_05_CMD1 0x01000000
```

6.68.1.35 SPI020_05_CMD1_w

```
#define SPI020_05_CMD1_w 0x01000000
```

6.68.1.36 SPI020_05_CMD2

```
#define SPI020_05_CMD2 0x0
```

6.68.1.37 SPI020_05_CMD2_w

```
#define SPI020_05_CMD2_w 0x1
```

6.68.1.38 SPI020_05_CMD3

```
#define SPI020_05_CMD3 (0x05000004|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.39 SPI020_05_CMD3_w

```
#define SPI020_05_CMD3_w (0x05000008|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.40 SPI020_06_CMD0

```
#define SPI020_06_CMD0 0x0
```

Kneron Confidential

6.68.1.41 SPI020_06_CMD1

```
#define SPI020_06_CMD1 0x01000000
```

6.68.1.42 SPI020_06_CMD2

```
#define SPI020_06_CMD2 0x0
```

6.68.1.43 SPI020_06_CMD3

```
#define SPI020_06_CMD3 (0x06000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.44 SPI020_0B_CMD0

```
#define SPI020_0B_CMD0 0x0
```

6.68.1.45 SPI020_0B_CMD1

```
#define SPI020_0B_CMD1 0x01080003
```

6.68.1.46 SPI020_0B_CMD2

```
#define SPI020_0B_CMD2 0x0
```

6.68.1.47 SPI020_0B_CMD3

```
#define SPI020_0B_CMD3 (0x0B000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.48 SPI020_0C_CMD0

```
#define SPI020_0C_CMD0 0x0
```

Kneron Confidential

6.68.1.49 SPI020_0C_CMD1

```
#define SPI020_0C_CMD1 0x01080004
```

6.68.1.50 SPI020_0C_CMD2

```
#define SPI020_0C_CMD2 0x0
```

6.68.1.51 SPI020_0C_CMD3

```
#define SPI020_0C_CMD3 (0x0C000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.52 SPI020_0D_CMD0

```
#define SPI020_0D_CMD0 0x0
```

6.68.1.53 SPI020_0D_CMD1

```
#define SPI020_0D_CMD1 0x01060003
```

6.68.1.54 SPI020_0D_CMD2

```
#define SPI020_0D_CMD2 0x0
```

6.68.1.55 SPI020_0D_CMD3

```
#define SPI020_0D_CMD3 (0x0D000010|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.56 SPI020_11_CMD0

```
#define SPI020_11_CMD0 0x0
```

Kneron Confidential

6.68.1.57 SPI020_11_CMD1

```
#define SPI020_11_CMD1 0x01000000
```

6.68.1.58 SPI020_11_CMD2

```
#define SPI020_11_CMD2 0x1
```

6.68.1.59 SPI020_11_CMD3

```
#define SPI020_11_CMD3 (0x11000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.60 SPI020_12_CMD0

```
#define SPI020_12_CMD0 0x0
```

6.68.1.61 SPI020_12_CMD1

```
#define SPI020_12_CMD1 0x01000004
```

6.68.1.62 SPI020_12_CMD2

```
#define SPI020_12_CMD2 0x0
```

6.68.1.63 SPI020_12_CMD3

```
#define SPI020_12_CMD3 (0x12000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.64 SPI020_13_CMD0

```
#define SPI020_13_CMD0 0x0
```

Kneron Confidential

6.68.1.65 SPI020_13_CMD1

```
#define SPI020_13_CMD1 0x01000004
```

6.68.1.66 SPI020_13_CMD2

```
#define SPI020_13_CMD2 0x0
```

6.68.1.67 SPI020_13_CMD3

```
#define SPI020_13_CMD3 (0x13000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.68 SPI020_15_CMD0

```
#define SPI020_15_CMD0 0x0
```

6.68.1.69 SPI020_15_CMD1

```
#define SPI020_15_CMD1 0x01000000
```

6.68.1.70 SPI020_15_CMD2

```
#define SPI020_15_CMD2 0x1
```

6.68.1.71 SPI020_15_CMD3

```
#define SPI020_15_CMD3 (0x15000008|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.72 SPI020_20_CMD0

```
#define SPI020_20_CMD0 0x0
```

Kneron Confidential

6.68.1.73 SPI020_20_CMD1

```
#define SPI020_20_CMD1 0x01000003
```

6.68.1.74 SPI020_20_CMD2

```
#define SPI020_20_CMD2 0x0
```

6.68.1.75 SPI020_20_CMD3

```
#define SPI020_20_CMD3 (0x20000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.76 SPI020_21_CMD0

```
#define SPI020_21_CMD0 0x0
```

6.68.1.77 SPI020_21_CMD1

```
#define SPI020_21_CMD1 0x01000004
```

6.68.1.78 SPI020_21_CMD2

```
#define SPI020_21_CMD2 0x0
```

6.68.1.79 SPI020_21_CMD3

```
#define SPI020_21_CMD3 (0x21000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.80 SPI020_30_CMD0

```
#define SPI020_30_CMD0 0x0
```

Kneron Confidential

6.68.1.81 SPI020_30_CMD1

```
#define SPI020_30_CMD1 0x01000000
```

6.68.1.82 SPI020_30_CMD2

```
#define SPI020_30_CMD2 0x0
```

6.68.1.83 SPI020_30_CMD3

```
#define SPI020_30_CMD3 (0x30000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.84 SPI020_31_CMD0

```
#define SPI020_31_CMD0 0x0
```

6.68.1.85 SPI020_31_CMD1

```
#define SPI020_31_CMD1 0x01000000
```

6.68.1.86 SPI020_31_CMD2

```
#define SPI020_31_CMD2 0x1
```

6.68.1.87 SPI020_31_CMD3

```
#define SPI020_31_CMD3 (0x31000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.88 SPI020_32_CMD0

```
#define SPI020_32_CMD0 0x0
```

Kneron Confidential

6.68.1.89 SPI020_32_CMD1

```
#define SPI020_32_CMD1 0x01000003
```

6.68.1.90 SPI020_32_CMD2

```
#define SPI020_32_CMD2 0x0
```

6.68.1.91 SPI020_32_CMD3

```
#define SPI020_32_CMD3 (0x32000042|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.92 SPI020_34_CMD0 [1/2]

```
#define SPI020_34_CMD0 0x0
```

6.68.1.93 SPI020_34_CMD0 [2/2]

```
#define SPI020_34_CMD0 0x0
```

6.68.1.94 SPI020_34_CMD1 [1/2]

```
#define SPI020_34_CMD1 0x01000004
```

6.68.1.95 SPI020_34_CMD1 [2/2]

```
#define SPI020_34_CMD1 0x01000004
```

6.68.1.96 SPI020_34_CMD2 [1/2]

```
#define SPI020_34_CMD2 0x0
```

Kneron Confidential

6.68.1.97 SPI020_34_CMD2 [2/2]

```
#define SPI020_34_CMD2 0x0
```

6.68.1.98 SPI020_34_CMD3 [1/2]

```
#define SPI020_34_CMD3 (0x34000042|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.99 SPI020_34_CMD3 [2/2]

```
#define SPI020_34_CMD3 (0x34000042|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.100 SPI020_35_CMD0

```
#define SPI020_35_CMD0 0x0
```

6.68.1.101 SPI020_35_CMD1

```
#define SPI020_35_CMD1 0x01000000
```

6.68.1.102 SPI020_35_CMD2

```
#define SPI020_35_CMD2 0x1
```

6.68.1.103 SPI020_35_CMD3

```
#define SPI020_35_CMD3 (0x35000008|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.104 SPI020_3B_CMD0

```
#define SPI020_3B_CMD0 0x0
```

Kneron Confidential

6.68.1.105 SPI020_3B_CMD1

```
#define SPI020_3B_CMD1 0x01080003
```

6.68.1.106 SPI020_3B_CMD2

```
#define SPI020_3B_CMD2 0x0
```

6.68.1.107 SPI020_3B_CMD3

```
#define SPI020_3B_CMD3 (0x3B000020|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.108 SPI020_3C_CMD0

```
#define SPI020_3C_CMD0 0x0
```

6.68.1.109 SPI020_3C_CMD1

```
#define SPI020_3C_CMD1 0x01080004
```

6.68.1.110 SPI020_3C_CMD2

```
#define SPI020_3C_CMD2 0x0
```

6.68.1.111 SPI020_3C_CMD3

```
#define SPI020_3C_CMD3 (0x3C000020|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.112 SPI020_50_CMD0

```
#define SPI020_50_CMD0 0x0
```

Kneron Confidential

6.68.1.113 SPI020_50_CMD1

```
#define SPI020_50_CMD1 0x01000000
```

6.68.1.114 SPI020_50_CMD2

```
#define SPI020_50_CMD2 0x0
```

6.68.1.115 SPI020_50_CMD3

```
#define SPI020_50_CMD3 (0x50000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.116 SPI020_52_CMD0

```
#define SPI020_52_CMD0 0x0
```

6.68.1.117 SPI020_52_CMD1

```
#define SPI020_52_CMD1 0x01000003
```

6.68.1.118 SPI020_52_CMD2

```
#define SPI020_52_CMD2 0x0
```

6.68.1.119 SPI020_52_CMD3

```
#define SPI020_52_CMD3 (0x52000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.120 SPI020_60_CMD0

```
#define SPI020_60_CMD0 0x0
```

Kneron Confidential

6.68.1.121 SPI020_60_CMD1

```
#define SPI020_60_CMD1 0x01000000
```

6.68.1.122 SPI020_60_CMD2

```
#define SPI020_60_CMD2 0x0
```

6.68.1.123 SPI020_60_CMD3

```
#define SPI020_60_CMD3 (0x60000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.124 SPI020_66_CMD0

```
#define SPI020_66_CMD0 0x0
```

6.68.1.125 SPI020_66_CMD0_ORG

```
#define SPI020_66_CMD0_ORG 0x0
```

6.68.1.126 SPI020_66_CMD1

```
#define SPI020_66_CMD1 0x01000000
```

6.68.1.127 SPI020_66_CMD1_ORG

```
#define SPI020_66_CMD1_ORG 0x01000000
```

6.68.1.128 SPI020_66_CMD2

```
#define SPI020_66_CMD2 0x00
```

Kneron Confidential

6.68.1.129 SPI020_66_CMD2_ORG

```
#define SPI020_66_CMD2_ORG 0x00
```

6.68.1.130 SPI020_66_CMD3

```
#define SPI020_66_CMD3 (0x66000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.131 SPI020_66_CMD3_ORG

```
#define SPI020_66_CMD3_ORG (0x66000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.132 SPI020_6B_CMD0

```
#define SPI020_6B_CMD0 0x0
```

6.68.1.133 SPI020_6B_CMD1

```
#define SPI020_6B_CMD1 0x01080003
```

6.68.1.134 SPI020_6B_CMD2

```
#define SPI020_6B_CMD2 0x0
```

6.68.1.135 SPI020_6B_CMD3

```
#define SPI020_6B_CMD3 (0x6B000040|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.136 SPI020_6C_CMD0

```
#define SPI020_6C_CMD0 0x0
```

Kneron Confidential

6.68.1.137 SPI020_6C_CMD1

```
#define SPI020_6C_CMD1 0x01080004
```

6.68.1.138 SPI020_6C_CMD2

```
#define SPI020_6C_CMD2 0x0
```

6.68.1.139 SPI020_6C_CMD3

```
#define SPI020_6C_CMD3 (0x6C000040|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.140 SPI020_90_CMD0

```
#define SPI020_90_CMD0 0x0
```

6.68.1.141 SPI020_90_CMD1

```
#define SPI020_90_CMD1 0x01000003
```

6.68.1.142 SPI020_90_CMD2

```
#define SPI020_90_CMD2 0x02
```

6.68.1.143 SPI020_90_CMD3

```
#define SPI020_90_CMD3 (0x90000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.144 SPI020_99_CMD0

```
#define SPI020_99_CMD0 0x0
```

Kneron Confidential

6.68.1.145 SPI020_99_CMD1

```
#define SPI020_99_CMD1 0x01000000
```

6.68.1.146 SPI020_99_CMD2

```
#define SPI020_99_CMD2 0x00
```

6.68.1.147 SPI020_99_CMD3

```
#define SPI020_99_CMD3 (0x99000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.148 SPI020_9F_CMD0

```
#define SPI020_9F_CMD0 0x0
```

6.68.1.149 SPI020_9F_CMD1

```
#define SPI020_9F_CMD1 0x01020000
```

6.68.1.150 SPI020_9F_CMD2

```
#define SPI020_9F_CMD2 0x3
```

6.68.1.151 SPI020_9F_CMD3

```
#define SPI020_9F_CMD3 (0x9F000000|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.152 SPI020_AB_CMD0

```
#define SPI020_AB_CMD0 0x0
```

Kneron Confidential

6.68.1.153 SPI020_AB_CMD1

```
#define SPI020_AB_CMD1 0x01000000
```

6.68.1.154 SPI020_AB_CMD2

```
#define SPI020_AB_CMD2 0x0
```

6.68.1.155 SPI020_AB_CMD3

```
#define SPI020_AB_CMD3 (0xAB000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.156 SPI020_ABORT

```
#define SPI020_ABORT BIT8
```

6.68.1.157 SPI020_B7_CMD0

```
#define SPI020_B7_CMD0 0x0
```

6.68.1.158 SPI020_B7_CMD1

```
#define SPI020_B7_CMD1 0x01000000
```

6.68.1.159 SPI020_B7_CMD2

```
#define SPI020_B7_CMD2 0x0
```

6.68.1.160 SPI020_B7_CMD3

```
#define SPI020_B7_CMD3 (0xB7000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

Kneron Confidential

6.68.1.161 SPI020_B9_CMD0

```
#define SPI020_B9_CMD0 0x0
```

6.68.1.162 SPI020_B9_CMD1

```
#define SPI020_B9_CMD1 0x01000000
```

6.68.1.163 SPI020_B9_CMD2

```
#define SPI020_B9_CMD2 0x0
```

6.68.1.164 SPI020_B9_CMD3

```
#define SPI020_B9_CMD3 (0xB9000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.165 SPI020_BB_CMD0

```
#define SPI020_BB_CMD0 0x0
```

6.68.1.166 SPI020_BB_CMD1

```
#define SPI020_BB_CMD1 0x01040003
```

6.68.1.167 SPI020_BB_CMD2

```
#define SPI020_BB_CMD2 0x0
```

6.68.1.168 SPI020_BB_CMD3

```
#define SPI020_BB_CMD3 (0xBB000060|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

Kneron Confidential

6.68.1.169 SPI020_BC_CMD0

```
#define SPI020_BC_CMD0 0x0
```

6.68.1.170 SPI020_BC_CMD1

```
#define SPI020_BC_CMD1 0x01040004
```

6.68.1.171 SPI020_BC_CMD2

```
#define SPI020_BC_CMD2 0x0
```

6.68.1.172 SPI020_BC_CMD3

```
#define SPI020_BC_CMD3 (0xBC000060|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.173 SPI020_BD_CMD0

```
#define SPI020_BD_CMD0 0x0
```

6.68.1.174 SPI020_BD_CMD1

```
#define SPI020_BD_CMD1 0x01080003
```

6.68.1.175 SPI020_BD_CMD2

```
#define SPI020_BD_CMD2 0x0
```

6.68.1.176 SPI020_BD_CMD3

```
#define SPI020_BD_CMD3 (0xBD000070|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

Kneron Confidential

6.68.1.177 SPI020_C7_CMD0

```
#define SPI020_C7_CMD0 0x0
```

6.68.1.178 SPI020_C7_CMD1

```
#define SPI020_C7_CMD1 0x01000000
```

6.68.1.179 SPI020_C7_CMD2

```
#define SPI020_C7_CMD2 0x0
```

6.68.1.180 SPI020_C7_CMD3

```
#define SPI020_C7_CMD3 (0xC7000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.181 SPI020_CE_0

```
#define SPI020_CE_0 0x0000
```

6.68.1.182 SPI020_CE_1

```
#define SPI020_CE_1 0x0100
```

6.68.1.183 SPI020_CE_2

```
#define SPI020_CE_2 0x0200
```

6.68.1.184 SPI020_CE_3

```
#define SPI020_CE_3 0x0300
```

Kneron Confidential

6.68.1.185 SPI020_CE_VALUE

```
#define SPI020_CE_VALUE SPI020_CE_0
```

6.68.1.186 SPI020_CLK_DIVIDER

```
#define SPI020_CLK_DIVIDER (BIT0|BIT1)
```

6.68.1.187 SPI020_CLK_MODE

```
#define SPI020_CLK_MODE BIT4
```

6.68.1.188 SPI020_CMD_CMPLT

```
#define SPI020_CMD_CMPLT BIT0
```

6.68.1.189 SPI020_cmd_cmplt_intr_en

```
#define SPI020_cmd_cmplt_intr_en BIT1
```

6.68.1.190 SPI020_D8_CMD0

```
#define SPI020_D8_CMD0 0x0
```

6.68.1.191 SPI020_D8_CMD1

```
#define SPI020_D8_CMD1 0x01000003
```

6.68.1.192 SPI020_D8_CMD2

```
#define SPI020_D8_CMD2 0x0
```

Kneron Confidential

6.68.1.193 SPI020_D8_CMD3

```
#define SPI020_D8_CMD3 (0xD8000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.194 SPI020_DC_CMD0

```
#define SPI020_DC_CMD0 0x0
```

6.68.1.195 SPI020_DC_CMD1

```
#define SPI020_DC_CMD1 0x01000004
```

6.68.1.196 SPI020_DC_CMD2

```
#define SPI020_DC_CMD2 0x0
```

6.68.1.197 SPI020_DC_CMD3

```
#define SPI020_DC_CMD3 (0xDC000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.198 SPI020_DMA_EN

```
#define SPI020_DMA_EN BIT0
```

6.68.1.199 SPI020_E9_CMD0

```
#define SPI020_E9_CMD0 0x0
```

6.68.1.200 SPI020_E9_CMD1

```
#define SPI020_E9_CMD1 0x01000000
```

Kneron Confidential

6.68.1.201 SPI020_E9_CMD2

```
#define SPI020_E9_CMD2 0x0
```

6.68.1.202 SPI020_E9_CMD3

```
#define SPI020_E9_CMD3 (0xE9000002|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.203 SPI020_EB_CMD0

```
#define SPI020_EB_CMD0 0x0
```

6.68.1.204 SPI020_EB_CMD1

```
#define SPI020_EB_CMD1 0x01060003
```

6.68.1.205 SPI020_EB_CMD2

```
#define SPI020_EB_CMD2 0x0
```

6.68.1.206 SPI020_EB_CMD3

```
#define SPI020_EB_CMD3 (0xEB000080|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.207 SPI020_EC_CMD0

```
#define SPI020_EC_CMD0 0x0
```

6.68.1.208 SPI020_EC_CMD1

```
#define SPI020_EC_CMD1 0x01060004
```

Kneron Confidential

6.68.1.209 SPI020_EC_CMD2

```
#define SPI020_EC_CMD2 0x0
```

6.68.1.210 SPI020_EC_CMD3

```
#define SPI020_EC_CMD3 (0xEC000080|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.211 SPI020_ED_CMD0

```
#define SPI020_ED_CMD0 0x0
```

6.68.1.212 SPI020_ED_CMD1

```
#define SPI020_ED_CMD1 0x01080003
```

6.68.1.213 SPI020_ED_CMD2

```
#define SPI020_ED_CMD2 0x0
```

6.68.1.214 SPI020_ED_CMD3

```
#define SPI020_ED_CMD3 (0xED000090|SPI020_CE_VALUE|SPI020_INTR_CFG)
```

6.68.1.215 SPI020_INTR_CFG

```
#define SPI020_INTR_CFG 0x00000000
```

6.68.1.216 SPI020_RX_DEPTH

```
#define SPI020_RX_DEPTH 0xFF00
```

Kneron Confidential

6.68.1.217 SPI020_RXFIFO_READY

```
#define SPI020_RXFIFO_READY BIT1
```

6.68.1.218 SPI020_TX_DEPTH

```
#define SPI020_TX_DEPTH 0x00FF
```

6.68.1.219 SPI020_TXFIFO_READY

```
#define SPI020_TXFIFO_READY BIT0
```

6.68.1.220 SPI020REG_ACTIMER

```
#define SPI020REG_ACTIMER (SPI_FTSPI020_PA_BASE+0x14)
```

6.68.1.221 SPI020REG_CMD0

```
#define SPI020REG_CMD0 (SPI_FTSPI020_PA_BASE+0x00)
```

6.68.1.222 SPI020REG_CMD1

```
#define SPI020REG_CMD1 (SPI_FTSPI020_PA_BASE+0x04)
```

6.68.1.223 SPI020REG_CMD2

```
#define SPI020REG_CMD2 (SPI_FTSPI020_PA_BASE+0x08)
```

6.68.1.224 SPI020REG_CMD3

```
#define SPI020REG_CMD3 (SPI_FTSPI020_PA_BASE+0x0C)
```

Kneron Confidential

6.68.1.225 SPI020REG_CONTROL

```
#define SPI020REG_CONTROL (SPI_FTSPI020_PA_BASE+0x10)
```

6.68.1.226 SPI020REG_DATAPORT

```
#define SPI020REG_DATAPORT (SPI_FTSPI020_PA_BASE+0x100)
```

6.68.1.227 SPI020REG_FEATURE

```
#define SPI020REG_FEATURE (SPI_FTSPI020_PA_BASE+0x54)
```

6.68.1.228 SPI020REG_INTERRUPT

```
#define SPI020REG_INTERRUPT (SPI_FTSPI020_PA_BASE+0x20)
```

6.68.1.229 SPI020REG_INTR_ST

```
#define SPI020REG_INTR_ST (SPI_FTSPI020_PA_BASE+0x24)
```

6.68.1.230 SPI020REG_READ_ST

```
#define SPI020REG_READ_ST (SPI_FTSPI020_PA_BASE+0x28)
```

6.68.1.231 SPI020REG_SCKINDLY

```
#define SPI020REG_SCKINDLY (SPI_FTSPI020_PA_BASE+0x58)
```

6.68.1.232 SPI020REG_STATUS

```
#define SPI020REG_STATUS (SPI_FTSPI020_PA_BASE+0x18)
```

Kneron Confidential

6.68.1.233 SPI020REG_VERSION

```
#define SPI020REG_VERSION (SPI_FTSPI020_PA_BASE+0x50)
```

6.68.1.234 SPI_CLK_DIVIDER_2

```
#define SPI_CLK_DIVIDER_2 0x00
```

6.68.1.235 SPI_CLK_DIVIDER_4

```
#define SPI_CLK_DIVIDER_4 0x01
```

6.68.1.236 SPI_CLK_DIVIDER_6

```
#define SPI_CLK_DIVIDER_6 0x02
```

6.68.1.237 SPI_CLK_DIVIDER_8

```
#define SPI_CLK_DIVIDER_8 0x03
```

6.68.2 Enumeration Type Documentation

6.68.2.1 spi_clk_mode_t

```
enum spi_clk_mode_t
```

Enumerator

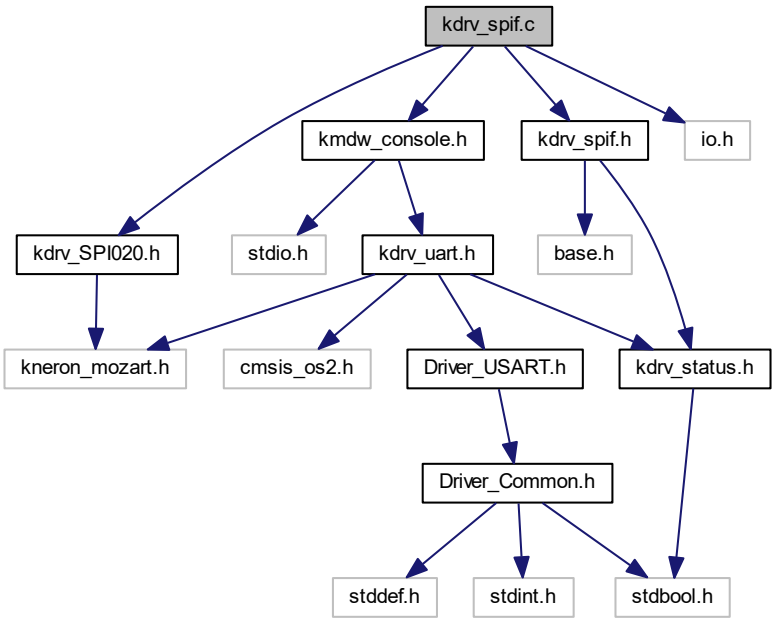
SPI_CLK_MODE0	
SPI_CLK_MODE3	

Kneron Confidential

6.69 kdrv_spif.c File Reference

```
#include "kdrv_SPI020.h"
#include "kdrv_spif.h"
#include "kmdw_console.h"
#include "io.h"
```

Include dependency graph for kdrv_spif.c:



Macros

- #define spif_msg(fmt, ...)
- #define min_t(x, y) (x < y ? x: y)
- #define MEMXFER_OPS_NONE 0x00
- #define MEMXFER_OPS_CPU 0x01
- #define MEMXFER_OPS_DMA 0x02
- #define MEMXFER_INITED 0x10
- #define MEMXFER_OPS_MASK MEMXFER_OPS_CPU | MEMXFER_OPS_DMA

Functions

- void kdrv_spif_initialize (void)

  Initialize spi flash include hardware setting, operation frequency, and flash status check.
- void kdrv_spif_memxfer_initialize (uint8_t flash_mode, uint8_t mem_mode)

  Initialize spi flash for memxfer include hardware setting, operation frequency, and flash status check.
- void kdrv_spif_set_commands (uint32_t cmd0, uint32_t cmd1, uint32_t cmd2, uint32_t cmd3)

  set spi communication commands including read/write by 3/4bytes address, dummy byte size, operation mode, etc
- void kdrv_spif_wait_command_complete (void)

Kneron Confidential

Check status bit to wait until command completed.

- uint8_t kdrv_spif_rx_FIFO_empty_check (void)
- void kdrv_spif_wait_rx_full (void)

  Wait until the RX FIFO is full so ready to read.
- void kdrv_spif_wait_tx_empty (void)

  Wait until the TX FIFO is empty so ready to write.
- uint32_t kdrv_spif_rxfifo_depth (void)

  Check the RX FIFO size, unit in byte.
- uint32_t kdrv_spif_txfifo_depth (void)

  Check the TX FIFO size, unit in byte.
- void kdrv_spif_read_Rx_FIFO (uint32_t *buf_word, uint16_t *buf_word_index, uint32_t target_byte)

  read Rx FIFO data
- void kdrv_spif_read_data (uint32_t *buf, uint32_t length)

  read data from specific index in spi flash
- void kdrv_spif_check_status_till_ready_2 (void)

  check status till the progress is done and ready for next step
- void kdrv_spif_check_status_till_ready (void)

  wait command completed and check status till it's ready
- void kdrv_spif_check_quad_status_till_ready (void)

  wait quad read command completed and check status till ready
- void kdrv_spif_write_data (uint8_t *buf, uint32_t length)

  write data to specific index in spi flash

6.69.1 Macro Definition Documentation

6.69.1.1 MEMXFER_INITED

```
#define MEMXFER_INITED 0x10
```

6.69.1.2 MEMXFER_OPS_CPU

```
#define MEMXFER_OPS_CPU 0x01
```

6.69.1.3 MEMXFER_OPS_DMA

```
#define MEMXFER_OPS_DMA 0x02
```

Kneron Confidential

6.69.1.4 MEMXFER_OPS_MASK

```
#define MEMXFER_OPS_MASK MEMXFER_OPS_CPU | MEMXFER_OPS_DMA
```

6.69.1.5 MEMXFER_OPS_NONE

```
#define MEMXFER_OPS_NONE 0x00
```

6.69.1.6 min_t

```
#define min_t(
              x,
              y ) ( x < y ?  x:  y )
```

6.69.1.7 spif_msg

```
#define spif_msg(
              fmt,
              ... )
```

6.69.2 Function Documentation

6.69.2.1 kdrv_spif_rx_FIFO_empty_check()

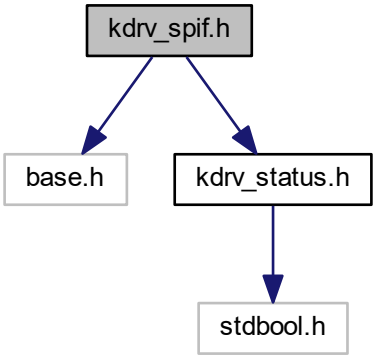
```
uint8_t kdrv_spif_rx_FIFO_empty_check (
            void  )
```

Kneron Confidential

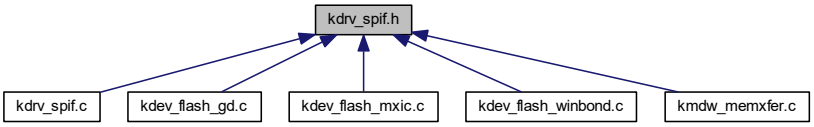
6.70 kdrv_spif.h File Reference

```
#include "base.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_spif.h:



This graph shows which files directly or indirectly include this file:



Functions

- void kdrv_spif_initialize (void)

  Initialize spi flash include hardware setting, operation frequency, and flash status check.
- void kdrv_spif_memxfer_initialize (uint8_t flash_mode, uint8_t mem_mode)

  Initialize spi flash for memxfer include hardware setting, operation frequency, and flash status check.
- kdrv_status_t kdrv_spif_uninitialize (void)

  Uninitialize spi flash and clear related variables.
- void kdrv_spif_set_commands (uint32_t cmd0, uint32_t cmd1, uint32_t cmd2, uint32_t cmd3)

  set spi communication commands including read/write by 3/4bytes address, dummy byte size, operation mode, etc
- void kdrv_spif_wait_command_complete (void)

  Check status bit to wait until command completed.
- void kdrv_spif_wait_rx_full (void)

  Wait until the RX FIFO is full so ready to read.
- void kdrv_spif_wait_tx_empty (void)

  Wait until the TX FIFO is empty so ready to write.

Kneron Confidential

- uint32_t kdrv_spif_rxfifo_depth (void)

  Check the RX FIFO size, unit in byte.
- uint32_t kdrv_spif_txfifo_depth (void)

  Check the TX FIFO size, unit in byte.
- void kdrv_spif_read_data (uint32_t *buf, uint32_t length)

  read data from specific index in spi flash
- void kdrv_spif_write_data (uint8_t *buf, uint32_t length)

  write data to specific index in spi flash
- void kdrv_spif_read_Rx_FIFO (uint32_t *buf_word, uint16_t *buf_word_index, uint32_t target_byte)

  read Rx FIFO data
- void kdrv_spif_check_status_till_ready_2 (void)

  check status till the progress is done and ready for next step
- void kdrv_spif_check_status_till_ready (void)

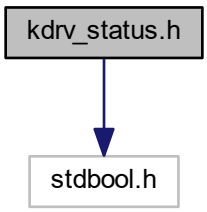
  wait command completed and check status till it's ready
- void kdrv_spif_check_quad_status_till_ready (void)

  wait quad read command completed and check status till ready

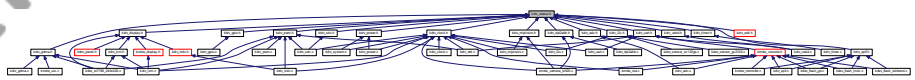
6.71 kdrv_status.h File Reference

```
#include <stdbool.h>
```

Include dependency graph for kdrv_status.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum kdrv_status_t {
  KDRV_STATUS_OK = 0, KDRV_STATUS_ERROR, KDRV_STATUS_INVALID_PARAM, KDRV_STATUS_I2C_BUS_BUSY,
  KDRV_STATUS_I2C_DEVICE_NACK, KDRV_STATUS_I2C_TIMEOUT, KDRV_STATUS_USBD_INVALID_ENDPOINT,
  KDRV_STATUS_USBD_TRANSFER_TIMEOUT,
  KDRV_STATUS_USBD_INVALID_TRANSFER, KDRV_STATUS_USBD_TRANSFER_IN_PROGRESS,

Kneron Confidential

KDRV_STATUS_GDMA_ERROR_NO_RESOURCE, KDRV_STATUS_TIMER_ID_NOT_IN_USED,
KDRV_STATUS_TIMER_ID_IN_USED, KDRV_STATUS_TIMER_ID_NOT_AVAILABLE, KDRV_STATUS_TIMER_INVALID_T
KDRV_STATUS_UART_TX_RX_BUSY,
KDRV_STATUS_UART_TIMEOUT, KDRV_STATUS_SDC_CMD_ERR, KDRV_STATUS_SDC_INIT_ERR,
KDRV_STATUS_SDC_CARD_NO_EXISTED,
KDRV_STATUS_SDC_CARD_TYPE_ERR, KDRV_STATUS_SDC_CSD_EXT_READ_ERR, KDRV_STATUS_SDC_CID_REA
KDRV_STATUS_SDC_MEM_ALLOC_ERR,
KDRV_STATUS_SDC_READ_FAIL, KDRV_STATUS_SDC_WRITE_FAIL, KDRV_STATUS_SDC_TRANSFER_FAIL,
KDRV_STATUS_SDC_TIMEOUT,
KDRV_STATUS_SDC_CMD_NOT_SUPPORT, KDRV_STATUS_SDC_BUS_WIDTH_NOT_SUPPORT,
KDRV_STATUS_SDC_BUS_WIDTH_ERR, KDRV_STATUS_SDC_SPEED_MOD_ERR,
KDRV_STATUS_SDC_VOL_ERR, KDRV_STATUS_SDC_INHIBIT_ERR, KDRV_STATUS_SDC_RECOVERABLE_ERR,
KDRV_STATUS_SDC_ABORT_ERR,
KDRV_STATUS_SDC_SWITCH_ERR, KDRV_STATUS_SDC_PWR_SET_ERR }

6.71.1 Enumeration Type Documentation

6.71.1.1 kdrv_status_t

enum kdrv_status_t

Enumerator

KDRV_STATUS_OK	driver status OK
KDRV_STATUS_ERROR	driver status error
KDRV_STATUS_INVALID_PARAM	driver invalid parameters
KDRV_STATUS_I2C_BUS_BUSY	I2C : bus is busy
KDRV_STATUS_I2C_DEVICE_NACK	I2C : slave device does not send ACK
KDRV_STATUS_I2C_TIMEOUT	I2C : transfer timeout
KDRV_STATUS_USBD_INVALID_ENDPOINT	USBD : invalid endpoint
KDRV_STATUS_USBD_TRANSFER_TIMEOUT	USBD : transfer timeout
KDRV_STATUS_USBD_INVALID_TRANSFER	USBD : invalid transfer operation
KDRV_STATUS_USBD_TRANSFER_IN_PROGRESS	USBD : transfer is in progress
KDRV_STATUS_GDMA_ERROR_NO_RESOURCE	GDMA : DMA channel is not available
KDRV_STATUS_TIMER_ID_NOT_IN_USED	Timer: TIMER Id NOT in used
KDRV_STATUS_TIMER_ID_IN_USED	Timer: TIMER Id in used
KDRV_STATUS_TIMER_ID_NOT_AVAILABLE	Timer: TIMER Id is NOT available
KDRV_STATUS_TIMER_INVALID_TIMER_ID	Timer: TIMER Id is invalid
KDRV_STATUS_UART_TX_RX_BUSY	UART: TX or RX is busy
KDRV_STATUS_UART_TIMEOUT	UART: timeout
KDRV_STATUS_SDC_CMD_ERR	SDC: command erro
KDRV_STATUS_SDC_INIT_ERR	SDC: init erro
KDRV_STATUS_SDC_CARD_NO_EXISTED	SDC: card not existed
KDRV_STATUS_SDC_CARD_TYPE_ERR	SDC: card type error
KDRV_STATUS_SDC_CSD_EXT_READ_ERR	SDC: csd info read error
KDRV_STATUS_SDC_CID_READ_ERR	SDC: cid info read error
KDRV_STATUS_SDC_MEM_ALLOC_ERR	SDC: mem alloc fail
KDRV_STATUS_SDC_READ_FAIL	SDC: sdc read fail
KDRV_STATUS_SDC_WRITE_FAIL	SDC: sdc write fail

Kneron Confidential

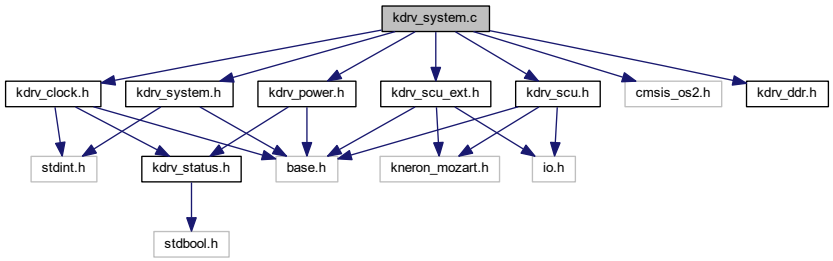
Enumerator

KDRV_STATUS_SDC_TRANSFER_FAIL	SDC: sdc transfer fail
KDRV_STATUS_SDC_TIMEOUT	SDC: sdc timeout
KDRV_STATUS_SDC_CMD_NOT_SUPPORT	SDC: sdc command not support
KDRV_STATUS_SDC_BUS_WIDTH_NOT_SUPPORT	SDC: sdc bus width not support
KDRV_STATUS_SDC_BUS_WIDTH_ERR	SDC: sdc bus width fail
KDRV_STATUS_SDC_SPEED_MOD_ERR	SDC: sdc set speed mode error
KDRV_STATUS_SDC_VOL_ERR	SDC: sdc voltage error
KDRV_STATUS_SDC_INHIBIT_ERR	SDC: sdc inhibit error
KDRV_STATUS_SDC_RECOVERABLE_ERR	SDC: sdc none recoverable
KDRV_STATUS_SDC_ABORT_ERR	SDC: sdc abort error
KDRV_STATUS_SDC_SWITCH_ERR	SDC: sdc switch fail
KDRV_STATUS_SDC_PWR_SET_ERR	SDC: sdc pwr set fail

6.72 kdrv_system.c File Reference

```
#include "kdrv_system.h"
#include <cmsis_os2.h>
#include "kdrv_scu.h"
#include "kdrv_scu_ext.h"
#include "kdrv_power.h"
#include "kdrv_clock.h"
#include "kdrv_ddr.h"
```

Include dependency graph for kdrv_system.c:



Macros

- #define BOOTUP_STATUS_WARM (SCU_REG_BTUP_STS_SMR | SCU_REG_BTUP_STS_PMR2)

Functions

- void kdrv_system_reset (int32_t subsystem)

  System reset.
- void system_isr (void)
- void kdrv_system_init (void)

  System initialize.
- void kdrv_system_init_ncpu (void)

  NCPU system initialize.

Kneron Confidential

Variables

- uint32_t bootup_status
- uint32_t warm_boot
- uint32_t __sys_int_flag

6.72.1 Macro Definition Documentation

6.72.1.1 BOOTUP_STATUS_WARM

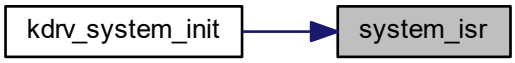
```
#define BOOTUP_STATUS_WARM (SCU_REG_BTUP_STS_SMR | SCU_REG_BTUP_STS_PMR2)
```

6.72.2 Function Documentation

6.72.2.1 system_isr()

```
void system_isr (
            void  )
```

Here is the caller graph for this function:



6.72.3 Variable Documentation

6.72.3.1 __sys_int_flag

```
uint32_t __sys_int_flag
```

Kneron Confidential

6.72.3.2 bootup_status

```
uint32_t bootup_status
```

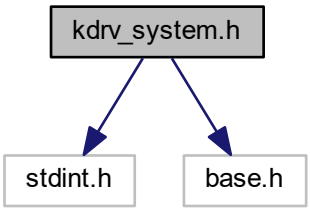
6.72.3.3 warm_boot

```
uint32_t warm_boot
```

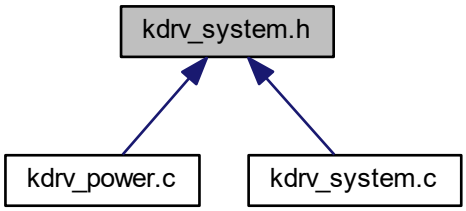
6.73 kdrv_system.h File Reference

```
#include <stdint.h>
#include "base.h"
```

Include dependency graph for kdrv_system.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define FLAGS_SOURCE_READY_EVT 0x91ad
- #define SCPU_FW 1
- #define NCPU_FW 2

Enumerations

- enum { SUBSYS_NPU = 1, SUBSYS_PD_NPU, SUBSYS_LCDC, SUBSYS_NCPU }

  Enumeration of system reset.

Functions

- void kdrv_system_init (void)

  System initialize.
- void kdrv_system_init_ncpu (void)

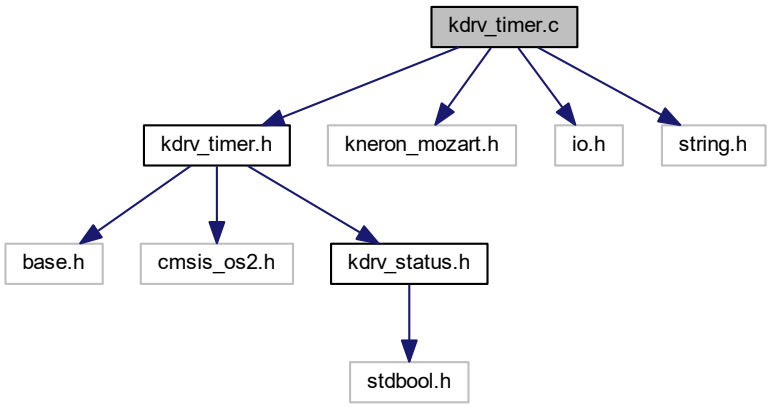
  NCPU system initialize.
- void kdrv_system_reset (int32_t subsystem)

  System reset.

6.74 kdrv_timer.c File Reference

```
#include "kdrv_timer.h"
#include "kneron_mozart.h"
#include "io.h"
#include <string.h>
```

Include dependency graph for kdrv_timer.c:



Kneron Confidential

Data Structures

- struct kdp_timer_t
- union U_regTIMER

Macros

- #define TIMER_DGB
- #define PERF_ISR_CNT 99
- #define FLAGS_TIMER_M1_DONE_EVENT BIT0
- #define FLAGS_TIMER_M2_DONE_EVENT BIT1
- #define FLAGS_TIMER_OF_DONE_EVENT BIT2
- #define FLAGS_TIMER_ALL_EVENTS (FLAGS_TIMER_M1_DONE_EVENT | FLAGS_TIMER_M2_DONE_EVENT | FLAGS_TIMER_OF_DONE_EVENT)
- #define TM_PER_IP 3
- #define TM_REG_OFS 3
- #define TM_ENABLE 0x01
- #define TM_CLK 0x02
- #define TM_OFENABLE 0x04
- #define TM_INT_MASK_M1 0x01
- #define TM_INT_MASK_M2 0x02
- #define TM_INT_MASK_OF 0x04
- #define TM_TMCR 0x30
- #define TM_TMINTSTAT 0x34
- #define TM_TMINTMASK 0x38
- #define TM_ID_1 0
- #define TM_ID_2 1
- #define TM_ID_3 2
- #define TM_DOWN 0
- #define TM_UP 1
- #define REGTIMER_0 ((U_regTIMER∗) TMR_FTPWMTMR010_0_PA_BASE)
- #define REGTIMER_1 ((U_regTIMER∗) TMR_FTPWMTMR010_1_PA_BASE)

Typedefs

- typedef volatile union U_regTIMER U_regTIMER

Enumerations

- enum TIMER_ID {
  TIMER_ID_0, TIMER_ID_1, TIMER_ID_2, TIMER_ID_3,
  TIMER_ID_4, TIMER_ID_5, TIMER_ID_MAX, TIMER_ID_INIT =0xFF }

Functions

- uint32_t kdrv_timer_get_bit (uint32_t mask)
- kdrv_status_t kdrv_timer_intclear (uint32_t TimerId, uint32_t IntId)
- void kdrv_timer_irqhandler0 (void)
- void kdrv_timer_irqhandler1 (void)
- kdrv_status_t kdrv_timer_register (uint32_t ∗TimerId, timer_cb_fr_isr_t cb_event, void ∗arg)
- kdrv_status_t kdrv_timer_unregister (uint32_t ∗TimerId)
- kdrv_status_t kdrv_timer_open (uint32_t ∗TimerId, timer_cb_fr_isr_t cb_event, void ∗arg)

  Request one timer id for further usage.
- kdrv_status_t kdrv_timer_close (uint32_t ∗TimerId)

  Close specific timer id.
- kdrv_status_t kdrv_timer_set (uint32_t ∗TimerId, uint32_t Intval, timer_stat_t State)

Kneron Confidential

Set specific timer with interval and status.

- kdrv_status_t kdrv_timer_perf_open (uint32_t *TimerId)

  Open a timer with specific timer id for performance measurement.

- kdrv_status_t kdrv_timer_perf_set (uint32_t *TimerId)

  Set specific timer for performance measurment usage.

- kdrv_status_t kdrv_timer_perf_get_instant (uint32_t *TimerId, uint32_t *instant, uint32_t *time)

  Get time consumption.

- kdrv_status_t kdrv_timer_perf_reset (uint32_t *TimerId)

  Reset performance timer.

- kdrv_status_t kdrv_timer_initialize (void)

  Enable clock, init timer ip, register IRQ/ISR function.

- kdrv_status_t kdrv_timer_uninitialize (void)

  Disable clock, and timer IRQ.

- kdrv_status_t kdrv_timer_delay_ms (uint32_t msec)

  Let system delay ms.

- kdrv_status_t kdrv_timer_delay_us (uint32_t usec)

  Let system delay us.

- kdrv_status_t kdrv_timer_perf_measure_start (void)

  Start to use performance measurement function.

- kdrv_status_t kdrv_timer_perf_measure_get (uint32_t *instant, uint32_t *time)

  Get time interval.

Variables

- volatile uint32_t timer_used_mask
- volatile uint32_t timer_wakeup
- volatile kdp_timer_t timer [TIMER_ID_MAX]
- uint32_t ErrCnt [2]
- uint32_t TCnt [2]
- uint32_t perftimerid
- uint32_t timer_inited = 0

6.74.1 Macro Definition Documentation

6.74.1.1 FLAGS_TIMER_ALL_EVENTS

```
#define FLAGS_TIMER_ALL_EVENTS ( FLAGS_TIMER_M1_DONE_EVENT | FLAGS_TIMER_M2_DONE_EVENT | FLAGS_TIMER_OF_DONE_EV
```

6.74.1.2 FLAGS_TIMER_M1_DONE_EVENT

```
#define FLAGS_TIMER_M1_DONE_EVENT BIT0
```

Kneron Confidential

6.74.1.3 FLAGS_TIMER_M2_DONE_EVENT

```
#define FLAGS_TIMER_M2_DONE_EVENT BIT1
```

6.74.1.4 FLAGS_TIMER_OF_DONE_EVENT

```
#define FLAGS_TIMER_OF_DONE_EVENT BIT2
```

6.74.1.5 PERF_ISR_CNT

```
#define PERF_ISR_CNT 99
```

6.74.1.6 REGTIMER_0

```
#define REGTIMER_0 ((U_regTIMER*) TMR_FTPWMTMR010_0_PA_BASE)
```

6.74.1.7 REGTIMER_1

```
#define REGTIMER_1 ((U_regTIMER*) TMR_FTPWMTMR010_1_PA_BASE)
```

6.74.1.8 TIMER_DGB

```
#define TIMER_DGB
```

6.74.1.9 TM_CLK

```
#define TM_CLK 0x02
```

6.74.1.10 TM_DOWN

```
#define TM_DOWN 0
```

Kneron Confidential

6.74.1.11 TM_ENABLE

```
#define TM_ENABLE 0x01
```

6.74.1.12 TM_ID_1

```
#define TM_ID_1 0
```

6.74.1.13 TM_ID_2

```
#define TM_ID_2 1
```

6.74.1.14 TM_ID_3

```
#define TM_ID_3 2
```

6.74.1.15 TM_INT_MASK_M1

```
#define TM_INT_MASK_M1 0x01
```

6.74.1.16 TM_INT_MASK_M2

```
#define TM_INT_MASK_M2 0x02
```

6.74.1.17 TM_INT_MASK_OF

```
#define TM_INT_MASK_OF 0x04
```

6.74.1.18 TM_OFENABLE

```
#define TM_OFENABLE 0x04
```

Kneron Confidential

6.74.1.19 TM_PER_IP

```
#define TM_PER_IP 3
```

6.74.1.20 TM_REG_OFS

```
#define TM_REG_OFS 3
```

6.74.1.21 TM_TMCR

```
#define TM_TMCR 0x30
```

6.74.1.22 TM_TMINTMASK

```
#define TM_TMINTMASK 0x38
```

6.74.1.23 TM_TMINTSTAT

```
#define TM_TMINTSTAT 0x34
```

6.74.1.24 TM_UP

```
#define TM_UP 1
```

6.74.2 Typedef Documentation

6.74.2.1 U_regTIMER

```
typedef volatile union U_regTIMER U_regTIMER
```

6.74.3 Enumeration Type Documentation

6.74.3.1 TIMER_ID

```
enum TIMER_ID
```

Kneron Confidential

Enumerator

TIMER_ID_0	
TIMER_ID_1	
TIMER_ID_2	
TIMER_ID_3	
TIMER_ID_4	
TIMER_ID_5	
TIMER_ID_MAX	
TIMER_ID_INIT	

6.74.4 Function Documentation

6.74.4.1 kdrv_timer_get_bit()

```
uint32_t kdrv_timer_get_bit (
            uint32_t mask )
```

6.74.4.2 kdrv_timer_intclear()

```
kdrv_status_t kdrv_timer_intclear (
            uint32_t TimerId,
            uint32_t IntId )
```

6.74.4.3 kdrv_timer_irqhandler0()

```
void kdrv_timer_irqhandler0 (
            void  )
```

6.74.4.4 kdrv_timer_irqhandler1()

```
void kdrv_timer_irqhandler1 (
            void  )
```

Kneron Confidential

6.74.4.5 kdrv_timer_register()

```
kdrv_status_t kdrv_timer_register (
            uint32_t * TimerId,
            timer_cb_fr_isr_t cb_event,
            void * arg )
```

6.74.4.6 kdrv_timer_unregister()

```
kdrv_status_t kdrv_timer_unregister (
            uint32_t * TimerId )
```

6.74.5 Variable Documentation

6.74.5.1 ErrCnt

```
uint32_t ErrCnt[2]
```

6.74.5.2 perftimerid

```
uint32_t perftimerid
```

6.74.5.3 TCnt

```
uint32_t TCnt[2]
```

6.74.5.4 timer

```
volatile kdp_timer_t timer[TIMER_ID_MAX]
```

6.74.5.5 timer_inited

```
uint32_t timer_inited = 0
```

Kneron Confidential

6.74.5.6 timer_used_mask

```
volatile uint32_t timer_used_mask
```

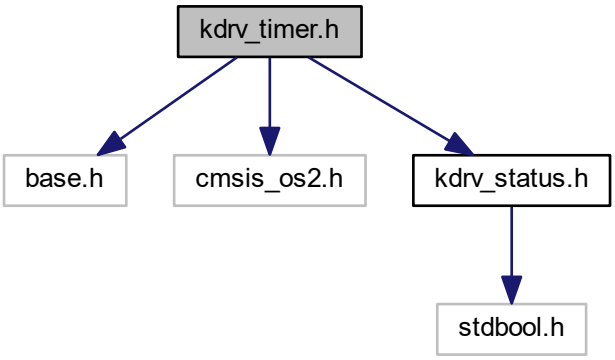
6.74.5.7 timer_wakeup

```
volatile uint32_t timer_wakeup
```

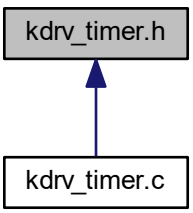
6.75 kdrv_timer.h File Reference

```
#include "base.h"
#include <cmsis_os2.h>
#include "kdrv_status.h"
```

Include dependency graph for kdrv_timer.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Typedefs

- typedef void(∗ timer_cb_fr_isr_t) (cb_event_t argument, void ∗arg)

  Enumerations of all timer status for kdrv_timer_set.

Enumerations

- enum cb_event_t { TIMER_M1_TIMEOUT, TIMER_M2_TIMEOUT, TIMER_OF_TIMEOUT }

  Enumerations of all timer call back even return status.
- enum timer_stat_t { TIMER_PAUSE, TIMER_START, TIMER_STAT_DEFAULT }

  Enumerations of all timer status for kdrv_timer_set.

Functions

- kdrv_status_t kdrv_timer_initialize (void)

  Enable clock, init timer ip, register IRQ/ISR function.
- kdrv_status_t kdrv_timer_uninitialize (void)

  Disable clock, and timer IRQ.
- kdrv_status_t kdrv_timer_open (uint32_t ∗TimerId, timer_cb_fr_isr_t event_cb, void ∗arg)

  Request one timer id for further usage.
- kdrv_status_t kdrv_timer_close (uint32_t ∗TimerId)

  Close specific timer id.
- kdrv_status_t kdrv_timer_set (uint32_t ∗TimerId, uint32_t Intval, timer_stat_t State)

  Set specific timer with interval and status.
- kdrv_status_t kdrv_timer_perf_open (uint32_t ∗TimerId)

  Open a timer with specific timer id for performance measurement.
- kdrv_status_t kdrv_timer_perf_set (uint32_t ∗TimerId)

  Set specific timer for performance measurment usage.
- kdrv_status_t kdrv_timer_perf_get_instant (uint32_t ∗TimerId, uint32_t ∗instant, uint32_t ∗time)

  Get time consumption.
- kdrv_status_t kdrv_timer_perf_reset (uint32_t ∗TimerId)

  Reset performance timer.
- kdrv_status_t kdrv_timer_delay_ms (uint32_t msec)

  Let system delay ms.
- kdrv_status_t kdrv_timer_delay_us (uint32_t usec)

  Let system delay us.
- kdrv_status_t kdrv_timer_perf_measure_start (void)

  Start to use performance measurement function.
- kdrv_status_t kdrv_timer_perf_measure_get (uint32_t ∗instant, uint32_t ∗time)

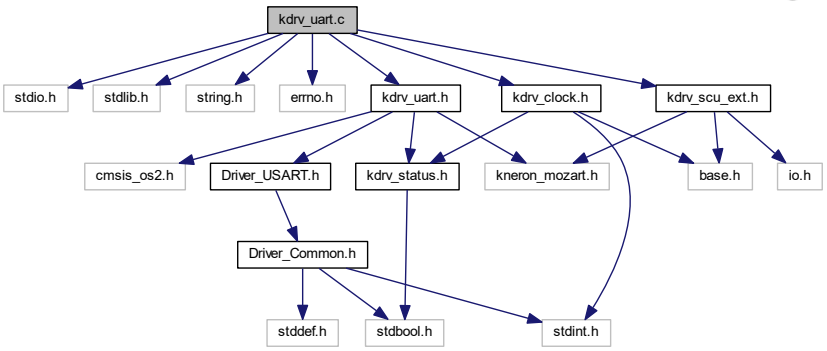
  Get time interval.

Kneron Confidential

6.76 kdrv_uart.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include "kdrv_uart.h"
#include "kdrv_scu_ext.h"
#include "kdrv_clock.h"
```

Include dependency graph for kdrv_uart.c:



Data Structures

- struct UART_STATUS_t
- struct UART_TRANSFER_INFO_t
- struct UART_INFO_T
- struct UART_RESOURCES_T
- struct uart_driver_handle_t
- struct uart_drv_ctx_t

Macros

- #define SERIAL_THR 0x00 /∗ Transmitter Holding Register(Write).∗/
- #define SERIAL_RBR 0x00 /∗ Receive Buffer register (Read).∗/
- #define SERIAL_IER 0x04 /∗ Interrupt Enable register.∗/
- #define SERIAL_IIR 0x08 /∗ Interrupt Identification register(Read).∗/
- #define SERIAL_FCR 0x08 /∗ FIFO control register(Write).∗/
- #define SERIAL_LCR 0x0C /∗ Line Control register.∗/
- #define SERIAL_MCR 0x10 /∗ Modem Control Register.∗/
- #define SERIAL_LSR 0x14 /∗ Line status register(Read) .∗/
- #define SERIAL_MSR 0x18 /∗ Modem Status register (Read).∗/
- #define SERIAL_SPR 0x1C /∗ Scratch pad register ∗/
- #define SERIAL_DLL 0x0 /∗ Divisor Register LSB ∗/
- #define SERIAL_DLM 0x4 /∗ Divisor Register MSB ∗/
- #define SERIAL_PSR 0x8 /∗ Prescale Divison Factor ∗/
- #define SERIAL_MDR 0x20
- #define SERIAL_ACR 0x24

Kneron Confidential

- #define SERIAL_TXLENL 0x28
- #define SERIAL_TXLENH 0x2C
- #define SERIAL_MRXLENL 0x30
- #define SERIAL_MRXLENH 0x34
- #define SERIAL_PLR 0x38
- #define SERIAL_FMIIR_PIO 0x3C
- #define SERIAL_FEATURE 0x68
- #define SERIAL_IER_DR 0x1 /* Data ready Enable */
- #define SERIAL_IER_TE 0x2 /* THR Empty Enable */
- #define SERIAL_IER_RLS 0x4 /* Receive Line Status Enable */
- #define SERIAL_IER_MS 0x8 /* Modem Staus Enable */
- #define SERIAL_IIR_NONE 0x1 /* No interrupt pending */
- #define SERIAL_IIR_RLS 0x6 /* Receive Line Status */
- #define SERIAL_IIR_DR 0x4 /* Receive Data Ready */
- #define SERIAL_IIR_TIMEOUT 0xc /* Receive Time Out */
- #define SERIAL_IIR_TE 0x2 /* THR Empty */
- #define SERIAL_IIR_MODEM 0x0 /* Modem Status */
- #define SERIAL_FCR_FE 0x1 /* FIFO Enable */
- #define SERIAL_FCR_RXFR 0x2 /* Rx FIFO Reset */
- #define SERIAL_FCR_TXFR 0x4 /* Tx FIFO Reset */
- #define SERIAL_LCR_LEN5 0x0
- #define SERIAL_LCR_LEN6 0x1
- #define SERIAL_LCR_LEN7 0x2
- #define SERIAL_LCR_LEN8 0x3
- #define SERIAL_LCR_STOP 0x4
- #define SERIAL_LCR_EVEN 0x18 /* Even Parity */
- #define SERIAL_LCR_ODD 0x8 /* Odd Parity */
- #define SERIAL_LCR_PE 0x8 /* Parity Enable */
- #define SERIAL_LCR_SETBREAK 0x40 /* Set Break condition */
- #define SERIAL_LCR_STICKPARITY 0x20 /* Stick Parity Enable */
- #define SERIAL_LCR_DLAB 0x80 /* Divisor Latch Access Bit */
- #define SERIAL_LSR_DR 0x1 /* Data Ready */
- #define SERIAL_LSR_OE 0x2 /* Overrun Error */
- #define SERIAL_LSR_PE 0x4 /* Parity Error */
- #define SERIAL_LSR_FE 0x8 /* Framing Error */
- #define SERIAL_LSR_BI 0x10 /* Break Interrupt */
- #define SERIAL_LSR_THRE 0x20 /* THR Empty */
- #define SERIAL_LSR_TE 0x40 /* Transmitte Empty */
- #define SERIAL_LSR_DE 0x80 /* FIFO Data Error */
- #define SERIAL_MCR_DTR 0x1 /* Data Terminal Ready */
- #define SERIAL_MCR_RTS 0x2 /* Request to Send */
- #define SERIAL_MCR_OUT1 0x4 /* output 1 */
- #define SERIAL_MCR_OUT2 0x8 /* output2 or global interrupt enable */
- #define SERIAL_MCR_LPBK 0x10 /* loopback mode */
- #define SERIAL_MSR_DELTACTS 0x1 /* Delta CTS */
- #define SERIAL_MSR_DELTADSR 0x2 /* Delta DSR */
- #define SERIAL_MSR_TERI 0x4 /* Trailing Edge RI */
- #define SERIAL_MSR_DELTACD 0x8 /* Delta CD */
- #define SERIAL_MSR_CTS 0x10 /* Clear To Send */
- #define SERIAL_MSR_DSR 0x20 /* Data Set Ready */
- #define SERIAL_MSR_RI 0x40 /* Ring Indicator */
- #define SERIAL_MSR_DCD 0x80 /* Data Carrier Detect */
- #define SERIAL_MDR_MODE_SEL 0x03
- #define SERIAL_MDR_UART 0x0
- #define SERIAL_MDR_SIR 0x1

Kneron Confidential

- #define SERIAL_MDR_FIR 0x2
- #define SERIAL_ACR_TXENABLE 0x1
- #define SERIAL_ACR_RXENABLE 0x2
- #define SERIAL_ACR_SET_EOT 0x4
- #define IIR_CODE_MASK 0xf
- #define SERIAL_IIR_TX_FIFO_FULL 0x10
- #define UART_INITIALIZED (1 << 0)
- #define UART_BASIC_CONFIGURED (1 << 2)
- #define UART_FIFO_RX_CONFIGURED (1 << 3)
- #define UART_FIFO_TX_CONFIGURED (1 << 4)
- #define UART_TX_ENABLED (1 << 5)
- #define UART_RX_ENABLED (1 << 6)
- #define UART_LOOPBACK_ENABLED (1 << 7)
- #define SERIAL_FIFO_DEPTH_REG 0x68
- #define SERIAL_FIFO_DEPTH_16B 0x1
- #define SERIAL_FIFO_DEPTH_32B 0x2
- #define SERIAL_FIFO_DEPTH_64B 0x4
- #define SERIAL_FIFO_DEPTH_128B 0x8
- #define SERIAL_FIFO_TRIG_LVEL_1 0x0
- #define SERIAL_FIFO_TRIG_LVEL_4 0x1
- #define SERIAL_FIFO_TRIG_LVEL_8 0x2
- #define SERIAL_FIFO_TRIG_LVEL_14 0x3
- #define DEFAULT_SYNC_TIMEOUT_CHARS_TIME 0
- #define MAX_UART_INST 5

Typedefs

- typedef void(* uart_isr_t) (void)

Enumerations

- enum uart_state_t { UART_UNINIT, UART_INIT_DONE, UART_WORKING, UART_CLOSED }

Functions

- void UART0_ISR (void)
- void UART1_ISR (void)
- void UART2_ISR (void)
- void UART3_ISR (void)
- void UART4_ISR (void)
- void UART_ISR (uint8_t port_no)
- kdrv_status_t kdrv_uart_initialize (void)

  UART driver initialization, it shall be called once in lifecycle.
- kdrv_status_t kdrv_uart_uninitialize (void)

  UART driver uninitialization.
- kdrv_status_t kdrv_uart_open (kdrv_uart_handle_t *handle, uint8_t com_port, uint32_t mode, kdrv_uart_callback_t cb)

  Open one UART port and acquire a uart port handle.
- kdrv_status_t kdrv_uart_close (kdrv_uart_handle_t handle)

  close the UART port
- kdrv_status_t kdrv_uart_configure (kdrv_uart_handle_t handle, kdrv_uart_control_t prop, uint8_t *pVal)

Kneron Confidential

set control for the opened UART port

- kdrv_status_t kdrv_uart_write (kdrv_uart_handle_t handle, uint8_t *data, uint32_t len)

  write data to uart port, such as command, parameters, but not suitable for chunk data
- kdrv_status_t kdrv_uart_get_char (kdrv_uart_handle_t handle, char *ch)

  read character data from UART port
- kdrv_status_t kdrv_uart_read (kdrv_uart_handle_t handle, uint8_t *data, uint32_t len)

  read data from the UART port
- uint32_t kdrv_uart_get_rx_count (kdrv_uart_handle_t handle)

  get char number in RX buffer
- uint32_t kdrv_uart_get_tx_count (kdrv_uart_handle_t handle)

  get char number in TX buffer

Variables

- uint32_t UART_PORT [5]
- IRQn_Type gUartIRQTbl [5]
- uart_isr_t gUartISRs [5]
- uint32_t gUartClk [5]

6.76.1 Macro Definition Documentation

6.76.1.1 DEFAULT_SYNC_TIMEOUT_CHARS_TIME

```
#define DEFAULT_SYNC_TIMEOUT_CHARS_TIME 0
```

6.76.1.2 IIR_CODE_MASK

```
#define IIR_CODE_MASK 0xf
```

6.76.1.3 MAX_UART_INST

```
#define MAX_UART_INST 5
```

6.76.1.4 SERIAL_ACR

```
#define SERIAL_ACR 0x24
```

Kneron Confidential

6.76.1.5 SERIAL_ACR_RXENABLE

```
#define SERIAL_ACR_RXENABLE 0x2
```

6.76.1.6 SERIAL_ACR_SET_EOT

```
#define SERIAL_ACR_SET_EOT 0x4
```

6.76.1.7 SERIAL_ACR_TXENABLE

```
#define SERIAL_ACR_TXENABLE 0x1
```

6.76.1.8 SERIAL_DLL

```
#define SERIAL_DLL 0x0 /* Divisor Register LSB */
```

6.76.1.9 SERIAL_DLM

```
#define SERIAL_DLM 0x4 /* Divisor Register MSB */
```

6.76.1.10 SERIAL_FCR

```
#define SERIAL_FCR 0x08 /* FIFO control register(Write).*/
```

6.76.1.11 SERIAL_FCR_FE

```
#define SERIAL_FCR_FE 0x1 /* FIFO Enable */
```

6.76.1.12 SERIAL_FCR_RXFR

```
#define SERIAL_FCR_RXFR 0x2 /* Rx FIFO Reset */
```

Kneron Confidential

6.76.1.13 SERIAL_FCR_TXFR

```
#define SERIAL_FCR_TXFR 0x4 /* Tx FIFO Reset */
```

6.76.1.14 SERIAL_FEATURE

```
#define SERIAL_FEATURE 0x68
```

6.76.1.15 SERIAL_FIFO_DEPTH_128B

```
#define SERIAL_FIFO_DEPTH_128B 0x8
```

6.76.1.16 SERIAL_FIFO_DEPTH_16B

```
#define SERIAL_FIFO_DEPTH_16B 0x1
```

6.76.1.17 SERIAL_FIFO_DEPTH_32B

```
#define SERIAL_FIFO_DEPTH_32B 0x2
```

6.76.1.18 SERIAL_FIFO_DEPTH_64B

```
#define SERIAL_FIFO_DEPTH_64B 0x4
```

6.76.1.19 SERIAL_FIFO_DEPTH_REG

```
#define SERIAL_FIFO_DEPTH_REG 0x68
```

6.76.1.20 SERIAL_FIFO_TRIG_LVEL_1

```
#define SERIAL_FIFO_TRIG_LVEL_1 0x0
```

Kneron Confidential

6.76.1.21 SERIAL_FIFO_TRIG_LVEL_14

```
#define SERIAL_FIFO_TRIG_LVEL_14 0x3
```

6.76.1.22 SERIAL_FIFO_TRIG_LVEL_4

```
#define SERIAL_FIFO_TRIG_LVEL_4 0x1
```

6.76.1.23 SERIAL_FIFO_TRIG_LVEL_8

```
#define SERIAL_FIFO_TRIG_LVEL_8 0x2
```

6.76.1.24 SERIAL_FMIIR_PIO

```
#define SERIAL_FMIIR_PIO 0x3C
```

6.76.1.25 SERIAL_IER

```
#define SERIAL_IER 0x04 /* Interrupt Enable register.*/
```

6.76.1.26 SERIAL_IER_DR

```
#define SERIAL_IER_DR 0x1 /* Data ready Enable */
```

6.76.1.27 SERIAL_IER_MS

```
#define SERIAL_IER_MS 0x8 /* Modem Staus Enable */
```

6.76.1.28 SERIAL_IER_RLS

```
#define SERIAL_IER_RLS 0x4 /* Receive Line Status Enable */
```

Kneron Confidential

6.76.1.29 SERIAL_IER_TE

```
#define SERIAL_IER_TE 0x2 /* THR Empty Enable */
```

6.76.1.30 SERIAL_IIR

```
#define SERIAL_IIR 0x08 /* Interrupt Identification register(Read).*/
```

6.76.1.31 SERIAL_IIR_DR

```
#define SERIAL_IIR_DR 0x4 /* Receive Data Ready */
```

6.76.1.32 SERIAL_IIR_MODEM

```
#define SERIAL_IIR_MODEM 0x0 /* Modem Status */
```

6.76.1.33 SERIAL_IIR_NONE

```
#define SERIAL_IIR_NONE 0x1 /* No interrupt pending */
```

6.76.1.34 SERIAL_IIR_RLS

```
#define SERIAL_IIR_RLS 0x6 /* Receive Line Status */
```

6.76.1.35 SERIAL_IIR_TE

```
#define SERIAL_IIR_TE 0x2 /* THR Empty */
```

6.76.1.36 SERIAL_IIR_TIMEOUT

```
#define SERIAL_IIR_TIMEOUT 0xc /* Receive Time Out */
```

Kneron Confidential

6.76.1.37 SERIAL_IIR_TX_FIFO_FULL

```
#define SERIAL_IIR_TX_FIFO_FULL 0x10
```

6.76.1.38 SERIAL_LCR

```
#define SERIAL_LCR 0x0C /* Line Control register.*/
```

6.76.1.39 SERIAL_LCR_DLAB

```
#define SERIAL_LCR_DLAB 0x80 /* Divisor Latch Access Bit */
```

6.76.1.40 SERIAL_LCR_EVEN

```
#define SERIAL_LCR_EVEN 0x18 /* Even Parity */
```

6.76.1.41 SERIAL_LCR_LEN5

```
#define SERIAL_LCR_LEN5 0x0
```

6.76.1.42 SERIAL_LCR_LEN6

```
#define SERIAL_LCR_LEN6 0x1
```

6.76.1.43 SERIAL_LCR_LEN7

```
#define SERIAL_LCR_LEN7 0x2
```

6.76.1.44 SERIAL_LCR_LEN8

```
#define SERIAL_LCR_LEN8 0x3
```

Kneron Confidential

6.76.1.45 SERIAL_LCR_ODD

```
#define SERIAL_LCR_ODD 0x8 /* Odd Parity */
```

6.76.1.46 SERIAL_LCR_PE

```
#define SERIAL_LCR_PE 0x8 /* Parity Enable */
```

6.76.1.47 SERIAL_LCR_SETBREAK

```
#define SERIAL_LCR_SETBREAK 0x40 /* Set Break condition */
```

6.76.1.48 SERIAL_LCR_STICKPARITY

```
#define SERIAL_LCR_STICKPARITY 0x20 /* Stick Parity Enable */
```

6.76.1.49 SERIAL_LCR_STOP

```
#define SERIAL_LCR_STOP 0x4
```

6.76.1.50 SERIAL_LSR

```
#define SERIAL_LSR 0x14 /* Line status register(Read) .*/
```

6.76.1.51 SERIAL_LSR_BI

```
#define SERIAL_LSR_BI 0x10 /* Break Interrupt */
```

6.76.1.52 SERIAL_LSR_DE

```
#define SERIAL_LSR_DE 0x80 /* FIFO Data Error */
```

Kneron Confidential

6.76.1.53 SERIAL_LSR_DR

```
#define SERIAL_LSR_DR 0x1 /* Data Ready */
```

6.76.1.54 SERIAL_LSR_FE

```
#define SERIAL_LSR_FE 0x8 /* Framing Error */
```

6.76.1.55 SERIAL_LSR_OE

```
#define SERIAL_LSR_OE 0x2 /* Overrun Error */
```

6.76.1.56 SERIAL_LSR_PE

```
#define SERIAL_LSR_PE 0x4 /* Parity Error */
```

6.76.1.57 SERIAL_LSR_TE

```
#define SERIAL_LSR_TE 0x40 /* Transmitte Empty */
```

6.76.1.58 SERIAL_LSR_THRE

```
#define SERIAL_LSR_THRE 0x20 /* THR Empty */
```

6.76.1.59 SERIAL_MCR

```
#define SERIAL_MCR 0x10 /* Modem Control Register.*/
```

6.76.1.60 SERIAL_MCR_DTR

```
#define SERIAL_MCR_DTR 0x1 /* Data Terminal Ready */
```

Kneron Confidential

6.76.1.61 SERIAL_MCR_LPBK

```
#define SERIAL_MCR_LPBK 0x10 /* loopback mode */
```

6.76.1.62 SERIAL_MCR_OUT1

```
#define SERIAL_MCR_OUT1 0x4 /* output 1 */
```

6.76.1.63 SERIAL_MCR_OUT2

```
#define SERIAL_MCR_OUT2 0x8 /* output2 or global interrupt enable */
```

6.76.1.64 SERIAL_MCR_RTS

```
#define SERIAL_MCR_RTS 0x2 /* Request to Send */
```

6.76.1.65 SERIAL_MDR

```
#define SERIAL_MDR 0x20
```

6.76.1.66 SERIAL_MDR_FIR

```
#define SERIAL_MDR_FIR 0x2
```

6.76.1.67 SERIAL_MDR_MODE_SEL

```
#define SERIAL_MDR_MODE_SEL 0x03
```

6.76.1.68 SERIAL_MDR_SIR

```
#define SERIAL_MDR_SIR 0x1
```

Kneron Confidential

6.76.1.69 SERIAL_MDR_UART

```
#define SERIAL_MDR_UART 0x0
```

6.76.1.70 SERIAL_MRXLENH

```
#define SERIAL_MRXLENH 0x34
```

6.76.1.71 SERIAL_MRXLENL

```
#define SERIAL_MRXLENL 0x30
```

6.76.1.72 SERIAL_MSR

```
#define SERIAL_MSR 0x18 /* Modem Status register (Read).*/
```

6.76.1.73 SERIAL_MSR_CTS

```
#define SERIAL_MSR_CTS 0x10 /* Clear To Send */
```

6.76.1.74 SERIAL_MSR_DCD

```
#define SERIAL_MSR_DCD 0x80 /* Data Carrier Detect */
```

6.76.1.75 SERIAL_MSR_DELTACD

```
#define SERIAL_MSR_DELTACD 0x8 /* Delta CD */
```

6.76.1.76 SERIAL_MSR_DELTACTS

```
#define SERIAL_MSR_DELTACTS 0x1 /* Delta CTS */
```

Kneron Confidential

6.76.1.77 SERIAL_MSR_DELTADSR

```
#define SERIAL_MSR_DELTADSR 0x2 /* Delta DSR */
```

6.76.1.78 SERIAL_MSR_DSR

```
#define SERIAL_MSR_DSR 0x20 /* Data Set Ready */
```

6.76.1.79 SERIAL_MSR_RI

```
#define SERIAL_MSR_RI 0x40 /* Ring Indicator */
```

6.76.1.80 SERIAL_MSR_TERI

```
#define SERIAL_MSR_TERI 0x4 /* Trailing Edge RI */
```

6.76.1.81 SERIAL_PLR

```
#define SERIAL_PLR 0x38
```

6.76.1.82 SERIAL_PSR

```
#define SERIAL_PSR 0x8 /* Prescale Divison Factor */
```

6.76.1.83 SERIAL_RBR

```
#define SERIAL_RBR 0x00 /* Receive Buffer register (Read).*/
```

6.76.1.84 SERIAL_SPR

```
#define SERIAL_SPR 0x1C /* Scratch pad register */
```

Kneron Confidential

6.76.1.85 SERIAL_THR

```
#define SERIAL_THR 0x00 /* Transmitter Holding Register(Write).*/
```

Kneron Peripheral API - UART

Copyright (C) 2019 Kneron, Inc. All rights reserved.

6.76.1.86 SERIAL_TXLENH

```
#define SERIAL_TXLENH 0x2C
```

6.76.1.87 SERIAL_TXLENL

```
#define SERIAL_TXLENL 0x28
```

6.76.1.88 UART_BASIC_CONFIGURED

```
#define UART_BASIC_CONFIGURED (1 << 2)
```

6.76.1.89 UART_FIFO_RX_CONFIGURED

```
#define UART_FIFO_RX_CONFIGURED (1 << 3)
```

6.76.1.90 UART_FIFO_TX_CONFIGURED

```
#define UART_FIFO_TX_CONFIGURED (1 << 4)
```

6.76.1.91 UART_INITIALIZED

```
#define UART_INITIALIZED (1 << 0)
```

Kneron Confidential

6.76.1.92 UART_LOOPBACK_ENABLED

```
#define UART_LOOPBACK_ENABLED (1 << 7)
```

6.76.1.93 UART_RX_ENABLED

```
#define UART_RX_ENABLED (1 << 6)
```

6.76.1.94 UART_TX_ENABLED

```
#define UART_TX_ENABLED (1 << 5)
```

6.76.2 Typedef Documentation

6.76.2.1 uart_isr_t

```
typedef void(* uart_isr_t) (void)
```

6.76.3 Enumeration Type Documentation

6.76.3.1 uart_state_t

```
enum uart_state_t
```

Enumerator

UART_UNINIT	
UART_INIT_DONE	
UART_WORKING	
UART_CLOSED	

6.76.4 Function Documentation

Kneron Confidential

6.76.4.1 UART0_ISR()

```
void UART0_ISR (
             void  )
```

6.76.4.2 UART1_ISR()

```
void UART1_ISR (
             void  )
```

Here is the call graph for this function:

6.76.4.3 UART2_ISR()

```
void UART2_ISR (
             void  )
```

Here is the call graph for this function:

Kneron Confidential

6.76.4.4 UART3_ISR()

```
void UART3_ISR (
            void  )
```

Here is the call graph for this function:

6.76.4.5 UART4_ISR()

```
void UART4_ISR (
            void  )
```

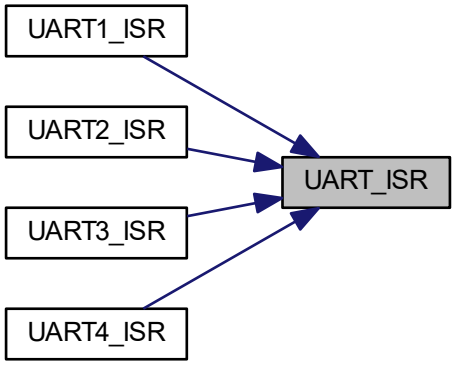
Here is the call graph for this function:

6.76.4.6 UART_ISR()

```
void UART_ISR (
            uint8_t port_no )
```

Kneron Confidential

Here is the caller graph for this function:



6.76.5 Variable Documentation

6.76.5.1 gUartClk

```
uint32_t gUartClk[5]
```

Initial value:

```
= {
    UART_CLOCK,
    UART_CLOCK_2,
    UART_CLOCK_2,
    UART_CLOCK,
    UART_CLOCK}
```

6.76.5.2 gUartIRQTbl

```
IRQn_Type gUartIRQTbl[5]
```

Initial value:

```
= {
    UART_FTUART010_0_IRQ,
    UART_FTUART010_1_IRQ,
    UART_FTUART010_1_1_IRQ,
    UART_FTUART010_1_2_IRQ,
    UART_FTUART010_1_3_IRQ
}
```

Kneron Confidential

6.76.5.3 gUartISRs

```
uart_isr_t gUartISRs[5]
```

Initial value:
```
= {
    UART0_ISR,
    UART1_ISR,
    UART2_ISR,
    UART3_ISR,
    UART4_ISR}
```

6.76.5.4 UART_PORT

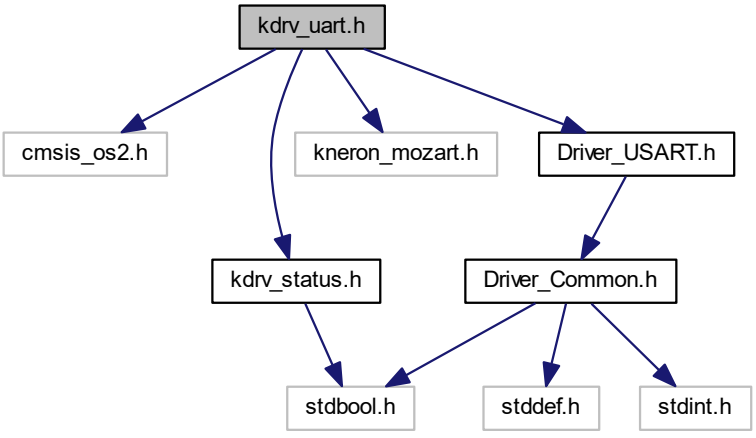
```
static uint32_t UART_PORT
```

Initial value:
```
= {UART_FTUART010_0_PA_BASE, UART_FTUART010_1_PA_BASE,
                            UART_FTUART010_1_1_PA_BASE, UART_FTUART010_1_2_PA_BASE,
      UART_FTUART010_1_3_PA_BASE}
```

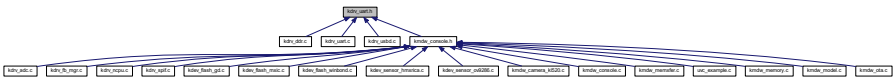
6.77 kdrv_uart.h File Reference

```
#include "cmsis_os2.h"
#include "kdrv_status.h"
#include "kneron_mozart.h"
#include "Driver_USART.h"
```

Include dependency graph for kdrv_uart.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_uart_config_t

  The structure of UART configuration parameters.
- struct kdrv_uart_fifo_config_t

  The structure of UART FIFO configuration parameters.

Macros

- #define BAUD_921600 (UART_CLOCK / 14745600)

  Enumerations of UART baud rate.
- #define BAUD_460800 (UART_CLOCK / 7372800)
- #define BAUD_115200 (UART_CLOCK / 1843200)
- #define BAUD_57600 (UART_CLOCK / 921600)
- #define BAUD_38400 (UART_CLOCK / 614400)
- #define BAUD_19200 (UART_CLOCK / 307200)
- #define BAUD_14400 (UART_CLOCK / 230400)
- #define BAUD_9600 (UART_CLOCK / 153600)
- #define BAUD_4800 (UART_CLOCK / 76800)
- #define BAUD_2400 (UART_CLOCK / 38400)
- #define BAUD_1200 (UART_CLOCK / 19200)
- #define PARITY_NONE 0

  The definition of UART parity.
- #define PARITY_ODD 1
- #define PARITY_EVEN 2
- #define PARITY_MARK 3
- #define PARITY_SPACE 4

Typedefs

- typedef int32_t kdrv_uart_handle_t
- typedef void(∗ kdrv_uart_callback_t) (uint32_t event)

Enumerations

- enum kdrv_uart_mode_t { UART_MODE_ASYN_RX = 0x1, UART_MODE_ASYN_TX = 0x2, UART_MODE_SYNC_RX = 0x4, UART_MODE_SYNC_TX = 0x8 }

  Enumerations of UART mode parameters.
- enum kdrv_uart_dev_id_t {
  UART0_DEV = 0, UART1_DEV, UART2_DEV, UART3_DEV,
  UART4_DEV, TOTAL_UART_DEV }

  Enumerations of UART device instance parameters.
- enum DRVUART_PORT {
  DRVUART_PORT0 = 0, DRVUART_PORT1 = 1, DRVUART_PORT2 = 2, DRVUART_PORT3 = 3,
  DRVUART_PORT4 = 4 }

  Enumerations of UART port parameters.
- enum kdrv_uart_control_t {
  UART_CTRL_CONFIG, UART_CTRL_FIFO_RX, UART_CTRL_FIFO_TX, UART_CTRL_LOOPBACK,
  UART_CTRL_TX_EN, UART_CTRL_RX_EN, UART_CTRL_ABORT_TX, UART_CTRL_ABORT_RX,
  UART_CTRL_TIMEOUT_RX, UART_CTRL_TIMEOUT_TX }

  Enumerations of UART control hardware signals.

Kneron Confidential

Functions

- kdrv_status_t kdrv_uart_initialize (void)

  UART driver initialization, it shall be called once in lifecycle.
- kdrv_status_t kdrv_uart_uninitialize (void)

  UART driver uninitialization.
- kdrv_status_t kdrv_uart_open (kdrv_uart_handle_t *handle, uint8_t com_port, uint32_t mode, kdrv_uart_callback_t callback)

  Open one UART port and acquire a uart port handle.
- kdrv_status_t kdrv_uart_configure (kdrv_uart_handle_t handle, kdrv_uart_control_t prop, uint8_t *val)

  set control for the opened UART port
- kdrv_status_t kdrv_uart_write (kdrv_uart_handle_t hdl, uint8_t *buf, uint32_t len)

  write data to uart port, such as command, parameters, but not suitable for chunk data
- kdrv_status_t kdrv_uart_get_char (kdrv_uart_handle_t handle, char *ch)

  read character data from UART port
- kdrv_status_t kdrv_uart_read (kdrv_uart_handle_t handle, uint8_t *buf, uint32_t len)

  read data from the UART port
- kdrv_status_t kdrv_uart_close (kdrv_uart_handle_t handle)

  close the UART port
- uint32_t kdrv_uart_get_rx_count (kdrv_uart_handle_t handle)

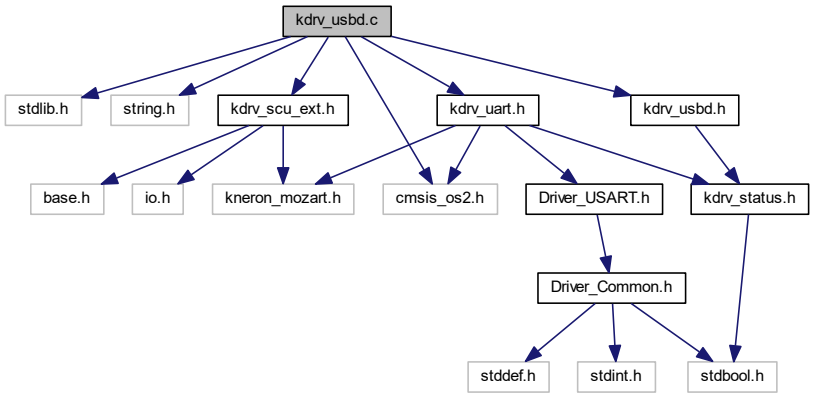
  get char number in RX buffer
- uint32_t kdrv_uart_get_tx_count (kdrv_uart_handle_t handle)

  get char number in TX buffer

6.78 kdrv_usbd.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include "cmsis_os2.h"
#include "kdrv_scu_ext.h"
#include "kdrv_uart.h"
#include "kdrv_usbd.h"
```

Include dependency graph for kdrv_usbd.c:



Kneron Confidential

Data Structures

- struct FIFO_Ctrl
- struct Ctrl_Block

Macros

- #define REG_OTG_CSR 0x80
- #define VBUS_VLD_RO BIT19
- #define B_SESS_END_RO BIT16
- #define REG_OTG_ISR 0x84
- #define REG_OTG_IER 0x88
- #define OTG_APLGRMV_RW1C BIT12
- #define OTG_A_WAIT_CON_RW1C BIT11
- #define OTG_OVC_RW1C BIT10
- #define OTG_IDCHG_RW1C BIT9
- #define OTG_RLCHG_RW1C BIT8
- #define OTG_B_SESS_END_RW1C BIT6
- #define OTG_A_VBUS_ERR_RW1C BIT5
- #define OTG_A_SRP_DET_RW1C BIT4
- #define OTG_B_SRP_DN_RW1C BIT0
- #define REG_GLB_ISR 0xC0
- #define REG_GLB_INT 0xC4
- #define INT_POLARITY BIT3
- #define OTG_INT BIT1
- #define DEV_INT BIT0
- #define REG_DEV_CTL 0x100
- #define REG_DEV_ADR 0x104
- #define AFT_CONF BIT7
- #define REG_DEV_TST 0x108
- #define TST_CLRFF BIT0
- #define REG_DEV_SMT 0x110
- #define REG_PHY_TST 0x114
- #define TST_JSTA BIT0
- #define REG_CXCFE 0x120
- #define F_EMP_0 BIT8
- #define CX_CLR BIT3
- #define CX_STL BIT2
- #define CX_DONE BIT0
- #define REG_DEV_ICR 0x124
- #define REG_DEV_MIGR 0x130
- #define REG_DEV_IGR 0x140
- #define GX_INT_G3_RO BIT3
- #define GX_INT_G2_RO BIT2
- #define GX_INT_G1_RO BIT1
- #define GX_INT_G0_RO BIT0
- #define REG_DEV_MISG0 0x134
- #define REG_DEV_ISG0 0x144
- #define G0_CX_COMABT_INT_RW1C BIT5
- #define G0_CX_COMFAIL_INT_RO BIT4
- #define G0_CX_COMEND_INT_RO BIT3
- #define G0_CX_OUT_INT_RO BIT2
- #define G0_CX_IN_INT_RO BIT1
- #define G0_CX_SETUP_INT_RO BIT0

Kneron Confidential

- #define REG_DEV_MISG1 0x138
- #define REG_DEV_ISG1 0x148
- #define MF0_IN_INT BIT16
- #define MF0_SPK_INT BIT1
- #define MF0_OUT_INT BIT0
- #define G1_F3_IN_INT_RO BIT19
- #define G1_F2_IN_INT_RO BIT18
- #define G1_F1_IN_INT_RO BIT17
- #define G1_F0_IN_INT_RO BIT16
- #define G1_F3_SPK_INT_RO BIT7
- #define G1_F3_OUT_INT_RO BIT6
- #define G1_F2_SPK_INT_RO BIT5
- #define G1_F2_OUT_INT_RO BIT4
- #define G1_F1_SPK_INT_RO BIT3
- #define G1_F1_OUT_INT_RO BIT2
- #define G1_F0_SPK_INT_RO BIT1
- #define G1_F0_OUT_INT_RO BIT0
- #define REG_DEV_MISG2 0x13C
- #define REG_DEV_ISG2 0x14C
- #define G2_Dev_Wakeup_byVbus_RO BIT10
- #define G2_Dev_Idle_RO BIT9
- #define G2_DMA_ERROR_RW1C BIT8
- #define G2_DMA_CMPLT_RW1C BIT7
- #define G2_RX0BYTE_INT_RW1C BIT6
- #define G2_TX0BYTE_INT_RW1C BIT5
- #define G2_ISO_SEQ_ABORT_INT_RW1C BIT4
- #define G2_ISO_SEQ_ERR_INT_RW1C BIT3
- #define G2_RESM_INT_RW1C BIT2
- #define G2_SUSP_INT_RW1C BIT1
- #define G2_USBRST_INT_RW1C BIT0
- #define REG_DEV_RXZ 0x150
- #define RX0BYTE_EP1 BIT0
- #define REG_DEV_INMPS_1 0x160
- #define TX0BYTE_IEPn BIT15
- #define RSTG_IEPn BIT12
- #define STL_IEPn BIT11
- #define REG_DEV_OUTMPS_1 0x180
- #define RSTG_OEPn BIT12
- #define STL_IEPn BIT11
- #define REG_DEV_EPMAP0 0x1A0
- #define REG_DEV_EPMAP1 0x1A4
- #define REG_DEV_FMAP 0x1A8
- #define REG_DEV_FCFG 0x1AC
- #define FFRST BIT12
- #define BC_Fn 0x7ff
- #define REG_DMA_TFN 0x1C0
- #define DMA_TARGET_ACC_CXF BIT4
- #define DMA_TARGET_ACC_F3 BIT3
- #define DMA_TARGET_ACC_F2 BIT2
- #define DMA_TARGET_ACC_F1 BIT1
- #define DMA_TARGET_ACC_F0 BIT0
- #define DMA_TARGET_ACC_NONE 0x0
- #define REG_DMA_CPS1 0x1C8
- #define DMA_TYPE BIT1
- #define DMA_START BIT0

Kneron Confidential

- #define REG_DMA_CPS2 0x1CC
- #define REG_DMA_CPS3 0x1D0
- #define FIFO_NUM 4
- #define UsbRegRead(reg_offset) inw(USB_FOTG210_PA_BASE + reg_offset)
- #define UsbRegWrite(reg_offset, val) outw(USB_FOTG210_PA_BASE + reg_offset, val)
- #define UsbRegMaskedSet(reg_offset, val) masked_outw(USB_FOTG210_PA_BASE + reg_offset, val, val)
- #define UsbRegMaskedClr(reg_offset, val) masked_outw(USB_FOTG210_PA_BASE + reg_offset, 0, val)
- #define MIN(x, y) (((x) < (y)) ? (x) : (y))
- #define QLEN 30
- #define FIFO_SHORT_PKT_INTERRUPTS (G1_F0_SPK_INT_RO | G1_F1_SPK_INT_RO | G1_F2_SPK_INT_RO | G1_F3_SPK_INT_RO)
- #define FIFO_OUT_INTERRUPTS (G1_F0_OUT_INT_RO | G1_F1_OUT_INT_RO | G1_F2_OUT_INT_RO | G1_F3_OUT_INT_RO)
- #define FIFO_IN_INTERRUPTS (G1_F0_IN_INT_RO | G1_F1_IN_INT_RO | G1_F2_IN_INT_RO | G1_F3_IN_INT_RO)

Enumerations

- enum { CONFIG_DEFAULT_STATE = 0, CONFIG_ADDRESS_STATE, CONFIG_CONFIGURED_STATE }
- enum { TXFER_CONTROL = 0, TXFER_ISO, TXFER_BULK, TXFER_INT }
- enum { RESP_NACK = 1, RESP_ACK, RESP_STALL }
- enum { READ_FIFO = 0, WRITE_FIFO = 1 }

Functions

- kdrv_status_t kdrv_usbd_initialize (void)

  USB device mode driver initialization.
- kdrv_status_t kdrv_usbd_uninitialize (void)

  USB device mode driver uninitialization.
- kdrv_status_t kdrv_usbd_reset_device ()

  reset device and then it can be re-enumerated by host
- kdrv_status_t kdrv_usbd_set_device_descriptor (kdrv_usbd_speed_t speed, kdrv_usbd_device_descriptor_t *dev_desc)

  configure device descriptor including configuration, interface and all endpoints descriptors
- kdrv_status_t kdrv_usbd_set_device_qualifier_descriptor (kdrv_usbd_speed_t speed, kdrv_usbd_device_qualifier_descriptor_t *dev_qual_desc)

  configure device qualifier descriptor, this is optional
- kdrv_status_t kdrv_usbd_set_enable (bool enable)

  set enable/disabale of USB device mode, host can enumerate this device only if device is enabled
- bool kdrv_usbd_is_dev_configured (void)

  check if device is enumerated and configured by a host
- kdrv_status_t kdrv_usbd_get_event (kdrv_usbd_event_t *uevent)

  get a usbd event, this is a blocking function for sync mode usage of USBD APIs
- kdrv_status_t kdrv_usbd_register_thread_notification (osThreadId_t tid, uint32_t tflag)

  register user thread ID and thread flag for notifications including events or transfer completion/errors
- kdrv_status_t kdrv_usbd_control_send (uint8_t *buf, uint32_t size, uint32_t timeout_ms)

  Control-IN transfer, send data to host through the control endpont.
- kdrv_status_t kdrv_usbd_control_receive (uint8_t *buf, uint32_t *size, uint32_t timeout_ms)

  Control-OUT transfer, receive data from host through the control endpont.
- kdrv_status_t kdrv_usbd_control_respond (kdrv_usbd_status_respond_t status)

  respond to host through control transfer in the status stage

Kneron Confidential

- kdrv_status_t kdrv_usbd_bulk_send (uint32_t endpoint, uint32_t *buf, uint32_t txLen, uint32_t timeout_ms)

  Bulk-IN transfser, send data to host through a bulk-in endpont in blocking mode.
- kdrv_status_t kdrv_usbd_reset_endpoint (uint32_t endpoint)

  reset specified endpoint
- kdrv_status_t kdrv_usbd_bulk_send_async (uint32_t endpoint, uint32_t *buf, uint32_t txLen)

  Bulk-IN transfser, send data to host through a bulk-in endpont in non-blocking mode.
- kdrv_status_t kdrv_usbd_bulk_receive (uint32_t endpoint, uint32_t *buf, uint32_t *blen, uint32_t timeout_ms)

  Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in blocking mode.
- kdrv_status_t kdrv_usbd_bulk_receive_async (uint32_t endpoint, uint32_t *buf, uint32_t blen)

  Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in non-blocking mode.
- kdrv_status_t kdrv_usbd_interrupt_send (uint32_t endpoint, uint32_t *buf, uint32_t txLen, uint32_t timeout_ms)

  Interrupt-IN transfer in blocking mode.
- kdrv_status_t kdrv_usbd_interrupt_receive (uint32_t endpoint, uint32_t *buf, uint32_t *rxLen, uint32_t timeout_ms)

  Interrupt-OUT transfer in blocking mode.

Variables

- Ctrl_Block cb

6.78.1 Macro Definition Documentation

6.78.1.1 AFT_CONF

```
#define AFT_CONF BIT7
```

6.78.1.2 B_SESS_END_RO

```
#define B_SESS_END_RO BIT16
```

6.78.1.3 BC_Fn

```
#define BC_Fn 0x7ff
```

6.78.1.4 CX_CLR

```
#define CX_CLR BIT3
```

Kneron Confidential

6.78.1.5 CX_DONE

```
#define CX_DONE BIT0
```

6.78.1.6 CX_STL

```
#define CX_STL BIT2
```

6.78.1.7 DEV_INT

```
#define DEV_INT BIT0
```

6.78.1.8 DMA_START

```
#define DMA_START BIT0
```

6.78.1.9 DMA_TARGET_ACC_CXF

```
#define DMA_TARGET_ACC_CXF BIT4
```

6.78.1.10 DMA_TARGET_ACC_F0

```
#define DMA_TARGET_ACC_F0 BIT0
```

6.78.1.11 DMA_TARGET_ACC_F1

```
#define DMA_TARGET_ACC_F1 BIT1
```

6.78.1.12 DMA_TARGET_ACC_F2

```
#define DMA_TARGET_ACC_F2 BIT2
```

Kneron Confidential

6.78.1.13 DMA_TARGET_ACC_F3

```
#define DMA_TARGET_ACC_F3 BIT3
```

6.78.1.14 DMA_TARGET_ACC_NONE

```
#define DMA_TARGET_ACC_NONE 0x0
```

6.78.1.15 DMA_TYPE

```
#define DMA_TYPE BIT1
```

6.78.1.16 F_EMP_0

```
#define F_EMP_0 BIT8
```

6.78.1.17 FFRST

```
#define FFRST BIT12
```

6.78.1.18 FIFO_IN_INTERRUPTS

```
#define FIFO_IN_INTERRUPTS (G1_F0_IN_INT_RO | G1_F1_IN_INT_RO | G1_F2_IN_INT_RO | G1_F3_IN_INT_RO)
```

6.78.1.19 FIFO_NUM

```
#define FIFO_NUM 4
```

6.78.1.20 FIFO_OUT_INTERRUPTS

```
#define FIFO_OUT_INTERRUPTS (G1_F0_OUT_INT_RO | G1_F1_OUT_INT_RO | G1_F2_OUT_INT_RO | G1_F3_OUT_INT_RO)
```

Kneron Confidential

6.78.1.21 FIFO_SHORT_PKT_INTERRUPTS

```
#define FIFO_SHORT_PKT_INTERRUPTS (G1_F0_SPK_INT_RO | G1_F1_SPK_INT_RO | G1_F2_SPK_INT_RO |
G1_F3_SPK_INT_RO)
```

6.78.1.22 G0_CX_COMABT_INT_RW1C

```
#define G0_CX_COMABT_INT_RW1C BIT5
```

6.78.1.23 G0_CX_COMEND_INT_RO

```
#define G0_CX_COMEND_INT_RO BIT3
```

6.78.1.24 G0_CX_COMFAIL_INT_RO

```
#define G0_CX_COMFAIL_INT_RO BIT4
```

6.78.1.25 G0_CX_IN_INT_RO

```
#define G0_CX_IN_INT_RO BIT1
```

6.78.1.26 G0_CX_OUT_INT_RO

```
#define G0_CX_OUT_INT_RO BIT2
```

6.78.1.27 G0_CX_SETUP_INT_RO

```
#define G0_CX_SETUP_INT_RO BIT0
```

Kneron Confidential

6.78.1.28 G1_F0_IN_INT_RO

```
#define G1_F0_IN_INT_RO BIT16
```

6.78.1.29 G1_F0_OUT_INT_RO

```
#define G1_F0_OUT_INT_RO BIT0
```

6.78.1.30 G1_F0_SPK_INT_RO

```
#define G1_F0_SPK_INT_RO BIT1
```

6.78.1.31 G1_F1_IN_INT_RO

```
#define G1_F1_IN_INT_RO BIT17
```

6.78.1.32 G1_F1_OUT_INT_RO

```
#define G1_F1_OUT_INT_RO BIT2
```

6.78.1.33 G1_F1_SPK_INT_RO

```
#define G1_F1_SPK_INT_RO BIT3
```

6.78.1.34 G1_F2_IN_INT_RO

```
#define G1_F2_IN_INT_RO BIT18
```

6.78.1.35 G1_F2_OUT_INT_RO

```
#define G1_F2_OUT_INT_RO BIT4
```

Kneron Confidential

6.78.1.36 G1_F2_SPK_INT_RO

```
#define G1_F2_SPK_INT_RO BIT5
```

6.78.1.37 G1_F3_IN_INT_RO

```
#define G1_F3_IN_INT_RO BIT19
```

6.78.1.38 G1_F3_OUT_INT_RO

```
#define G1_F3_OUT_INT_RO BIT6
```

6.78.1.39 G1_F3_SPK_INT_RO

```
#define G1_F3_SPK_INT_RO BIT7
```

6.78.1.40 G2_Dev_Idle_RO

```
#define G2_Dev_Idle_RO BIT9
```

6.78.1.41 G2_Dev_Wakeup_byVbus_RO

```
#define G2_Dev_Wakeup_byVbus_RO BIT10
```

6.78.1.42 G2_DMA_CMPLT_RW1C

```
#define G2_DMA_CMPLT_RW1C BIT7
```

6.78.1.43 G2_DMA_ERROR_RW1C

```
#define G2_DMA_ERROR_RW1C BIT8
```

Kneron Confidential

6.78.1.44 G2_ISO_SEQ_ABORT_INT_RW1C

```
#define G2_ISO_SEQ_ABORT_INT_RW1C BIT4
```

6.78.1.45 G2_ISO_SEQ_ERR_INT_RW1C

```
#define G2_ISO_SEQ_ERR_INT_RW1C BIT3
```

6.78.1.46 G2_RESM_INT_RW1C

```
#define G2_RESM_INT_RW1C BIT2
```

6.78.1.47 G2_RX0BYTE_INT_RW1C

```
#define G2_RX0BYTE_INT_RW1C BIT6
```

6.78.1.48 G2_SUSP_INT_RW1C

```
#define G2_SUSP_INT_RW1C BIT1
```

6.78.1.49 G2_TX0BYTE_INT_RW1C

```
#define G2_TX0BYTE_INT_RW1C BIT5
```

6.78.1.50 G2_USBRST_INT_RW1C

```
#define G2_USBRST_INT_RW1C BIT0
```

6.78.1.51 GX_INT_G0_RO

```
#define GX_INT_G0_RO BIT0
```

Kneron Confidential

6.78.1.52 GX_INT_G1_RO

```
#define GX_INT_G1_RO BIT1
```

6.78.1.53 GX_INT_G2_RO

```
#define GX_INT_G2_RO BIT2
```

6.78.1.54 GX_INT_G3_RO

```
#define GX_INT_G3_RO BIT3
```

6.78.1.55 INT_POLARITY

```
#define INT_POLARITY BIT3
```

6.78.1.56 MF0_IN_INT

```
#define MF0_IN_INT BIT16
```

6.78.1.57 MF0_OUT_INT

```
#define MF0_OUT_INT BIT0
```

6.78.1.58 MF0_SPK_INT

```
#define MF0_SPK_INT BIT1
```

Kneron Confidential

6.78.1.59 MIN

```
#define MIN(
             x,
             y ) (((x) < (y)) ?  (x) :  (y))
```

6.78.1.60 OTG_A_SRP_DET_RW1C

```
#define OTG_A_SRP_DET_RW1C BIT4
```

6.78.1.61 OTG_A_VBUS_ERR_RW1C

```
#define OTG_A_VBUS_ERR_RW1C BIT5
```

6.78.1.62 OTG_A_WAIT_CON_RW1C

```
#define OTG_A_WAIT_CON_RW1C BIT11
```

6.78.1.63 OTG_APLGRMV_RW1C

```
#define OTG_APLGRMV_RW1C BIT12
```

6.78.1.64 OTG_B_SESS_END_RW1C

```
#define OTG_B_SESS_END_RW1C BIT6
```

6.78.1.65 OTG_B_SRP_DN_RW1C

```
#define OTG_B_SRP_DN_RW1C BIT0
```

Kneron Confidential

6.78.1.66 OTG_IDCHG_RW1C

```
#define OTG_IDCHG_RW1C BIT9
```

6.78.1.67 OTG_INT

```
#define OTG_INT BIT1
```

6.78.1.68 OTG_OVC_RW1C

```
#define OTG_OVC_RW1C BIT10
```

6.78.1.69 OTG_RLCHG_RW1C

```
#define OTG_RLCHG_RW1C BIT8
```

6.78.1.70 QLEN

```
#define QLEN 30
```

6.78.1.71 REG_CXCFE

```
#define REG_CXCFE 0x120
```

6.78.1.72 REG_DEV_ADR

```
#define REG_DEV_ADR 0x104
```

6.78.1.73 REG_DEV_CTL

```
#define REG_DEV_CTL 0x100
```

Kneron Confidential

6.78.1.74 REG_DEV_EPMAP0

```
#define REG_DEV_EPMAP0 0x1A0
```

6.78.1.75 REG_DEV_EPMAP1

```
#define REG_DEV_EPMAP1 0x1A4
```

6.78.1.76 REG_DEV_FCFG

```
#define REG_DEV_FCFG 0x1AC
```

6.78.1.77 REG_DEV_FMAP

```
#define REG_DEV_FMAP 0x1A8
```

6.78.1.78 REG_DEV_ICR

```
#define REG_DEV_ICR 0x124
```

6.78.1.79 REG_DEV_IGR

```
#define REG_DEV_IGR 0x140
```

6.78.1.80 REG_DEV_INMPS_1

```
#define REG_DEV_INMPS_1 0x160
```

6.78.1.81 REG_DEV_ISG0

```
#define REG_DEV_ISG0 0x144
```

Kneron Confidential

6.78.1.82 REG_DEV_ISG1

```
#define REG_DEV_ISG1 0x148
```

6.78.1.83 REG_DEV_ISG2

```
#define REG_DEV_ISG2 0x14C
```

6.78.1.84 REG_DEV_MIGR

```
#define REG_DEV_MIGR 0x130
```

6.78.1.85 REG_DEV_MISG0

```
#define REG_DEV_MISG0 0x134
```

6.78.1.86 REG_DEV_MISG1

```
#define REG_DEV_MISG1 0x138
```

6.78.1.87 REG_DEV_MISG2

```
#define REG_DEV_MISG2 0x13C
```

6.78.1.88 REG_DEV_OUTMPS_1

```
#define REG_DEV_OUTMPS_1 0x180
```

6.78.1.89 REG_DEV_RXZ

```
#define REG_DEV_RXZ 0x150
```

Kneron Confidential

6.78.1.90 REG_DEV_SMT

```
#define REG_DEV_SMT 0x110
```

6.78.1.91 REG_DEV_TST

```
#define REG_DEV_TST 0x108
```

6.78.1.92 REG_DMA_CPS1

```
#define REG_DMA_CPS1 0x1C8
```

6.78.1.93 REG_DMA_CPS2

```
#define REG_DMA_CPS2 0x1CC
```

6.78.1.94 REG_DMA_CPS3

```
#define REG_DMA_CPS3 0x1D0
```

6.78.1.95 REG_DMA_TFN

```
#define REG_DMA_TFN 0x1C0
```

6.78.1.96 REG_GLB_INT

```
#define REG_GLB_INT 0xC4
```

6.78.1.97 REG_GLB_ISR

```
#define REG_GLB_ISR 0xC0
```

Kneron Confidential

6.78.1.98 REG_OTG_CSR

```
#define REG_OTG_CSR 0x80
```

6.78.1.99 REG_OTG_IER

```
#define REG_OTG_IER 0x88
```

6.78.1.100 REG_OTG_ISR

```
#define REG_OTG_ISR 0x84
```

6.78.1.101 REG_PHY_TST

```
#define REG_PHY_TST 0x114
```

6.78.1.102 RSTG_IEPn

```
#define RSTG_IEPn BIT12
```

6.78.1.103 RSTG_OEPn

```
#define RSTG_OEPn BIT12
```

6.78.1.104 RX0BYTE_EP1

```
#define RX0BYTE_EP1 BIT0
```

6.78.1.105 STL_IEPn [1/2]

```
#define STL_IEPn BIT11
```

Kneron Confidential

6.78.1.106 STL_IEPn [2/2]

```
#define STL_IEPn BIT11
```

6.78.1.107 TST_CLRFF

```
#define TST_CLRFF BIT0
```

6.78.1.108 TST_JSTA

```
#define TST_JSTA BIT0
```

6.78.1.109 TX0BYTE_IEPn

```
#define TX0BYTE_IEPn BIT15
```

6.78.1.110 UsbRegMaskedClr

```
#define UsbRegMaskedClr(
            reg_offset,
            val ) masked_outw(USB_FOTG210_PA_BASE + reg_offset, 0, val)
```

6.78.1.111 UsbRegMaskedSet

```
#define UsbRegMaskedSet(
            reg_offset,
            val ) masked_outw(USB_FOTG210_PA_BASE + reg_offset, val, val)
```

6.78.1.112 UsbRegRead

```
#define UsbRegRead(
            reg_offset ) inw(USB_FOTG210_PA_BASE + reg_offset)
```

Kneron Confidential

6.78.1.113 UsbRegWrite

```
#define UsbRegWrite(
              reg_offset,
              val ) outw(USB_FOTG210_PA_BASE + reg_offset, val)
```

6.78.1.114 VBUS_VLD_RO

```
#define VBUS_VLD_RO BIT19
```

6.78.2 Enumeration Type Documentation

6.78.2.1 anonymous enum

```
anonymous enum
```

Enumerator

CONFIG_DEFAULT_STATE	
CONFIG_ADDRESS_STATE	
CONFIG_CONFIGURED_STATE	

6.78.2.2 anonymous enum

```
anonymous enum
```

Enumerator

TXFER_CONTROL	
TXFER_ISO	
TXFER_BULK	
TXFER_INT	

6.78.2.3 anonymous enum

```
anonymous enum
```

Kneron Confidential

Enumerator

RESP_NACK	
RESP_ACK	
RESP_STALL	

6.78.2.4 anonymous enum

```
anonymous enum
```

Enumerator

READ_FIFO	
WRITE_FIFO	

6.78.3 Variable Documentation

6.78.3.1 cb

```
Ctrl_Block cb
```

Initial value:

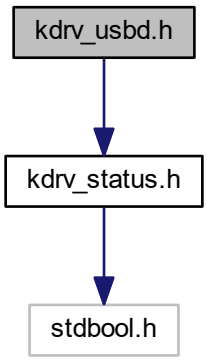
```
=
    {
        .notifyTid = 0,
        .notifyFlag = 0,
        .ep0_halted = false,
        .dev_desc = NULL,
        .config_state = CONFIG_DEFAULT_STATE,
}
```

Kneron Confidential

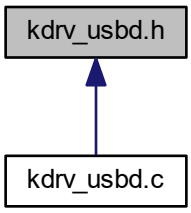
6.79 kdrv_usbd.h File Reference

```
#include "kdrv_status.h"
```

Include dependency graph for kdrv_usbd.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct kdrv_usbd_event_t

  USB event, it includes kdrv_usbd_event_name_t and related data.

Macros

- #define MAX_USBD_CONFIG 1
- #define MAX_USBD_INTERFACE 1
- #define MAX_USBD_ENDPOINT 4

Enumerations

- enum kdrv_usbd_speed_t { KDRV_USBD_HIGH_SPEED, KDRV_USBD_FULL_SPEED }

  Enumeration of connection speed.
- enum kdrv_usbd_event_name_t {
  KDRV_USBD_EVENT_BUS_RESET = 1, KDRV_USBD_EVENT_BUS_SUSPEND, KDRV_USBD_EVENT_BUS_RESUME,
  KDRV_USBD_EVENT_SETUP_PACKET,
  KDRV_USBD_EVENT_DEV_CONFIGURED, KDRV_USBD_EVENT_TRANSFER_BUF_FULL, KDRV_USBD_EVENT_TRAN
  KDRV_USBD_EVENT_TRANSFER_OUT,
  KDRV_USBD_EVENT_TRANSFER_TERMINATED, KDRV_USBD_EVENT_DMA_ERROR }

  Enumeration of USB event name type.
- enum kdrv_usbd_status_respond_t { KDRV_USBD_RESPOND_OK, KDRV_USBD_RESPOND_ERROR }

  Enumeration of code for response to host in control transfer.

Functions

- struct __attribute__ ((__packed__))

  8-byte setup packet struct
- kdrv_status_t kdrv_usbd_initialize (void)

  USB device mode driver initialization.
- kdrv_status_t kdrv_usbd_uninitialize (void)

  USB device mode driver uninitialization.
- kdrv_status_t kdrv_usbd_reset_device (void)

  reset device and then it can be re-enumerated by host
- kdrv_status_t kdrv_usbd_set_device_descriptor (kdrv_usbd_speed_t speed, kdrv_usbd_device_descriptor_t *dev_desc)

  configure device descriptor including configuration, interface and all endpoints descriptors
- kdrv_status_t kdrv_usbd_set_device_qualifier_descriptor (kdrv_usbd_speed_t speed, kdrv_usbd_device_qualifier_descriptor_t *dev_qual_desc)

  configure device qualifier descriptor, this is optional
- kdrv_status_t kdrv_usbd_register_thread_notification (osThreadId_t tid, uint32_t tflag)

  register user thread ID and thread flag for notifications including events or transfer completion/errors
- kdrv_status_t kdrv_usbd_set_enable (bool enable)

  set enable/disabale of USB device mode, host can enumerate this device only if device is enabled
- bool kdrv_usbd_is_dev_configured (void)

  check if device is enumerated and configured by a host
- kdrv_status_t kdrv_usbd_get_event (kdrv_usbd_event_t *uevent)

  get a usbd event, this is a blocking function for sync mode usage of USBD APIs
- kdrv_status_t kdrv_usbd_control_send (uint8_t *buf, uint32_t size, uint32_t timeout_ms)

  Control-IN transfer, send data to host through the control endpont.
- kdrv_status_t kdrv_usbd_control_receive (uint8_t *buf, uint32_t *size, uint32_t timeout_ms)

  Control-OUT transfer, receive data from host through the control endpont.
- kdrv_status_t kdrv_usbd_control_respond (kdrv_usbd_status_respond_t status)

  respond to host through control transfer in the status stage
- kdrv_status_t kdrv_usbd_reset_endpoint (uint32_t endpoint)

  reset specified endpoint
- kdrv_status_t kdrv_usbd_bulk_send (uint32_t endpoint, uint32_t *buf, uint32_t txLen, uint32_t timeout_ms)

  Bulk-IN transfser, send data to host through a bulk-in endpont in blocking mode.
- kdrv_status_t kdrv_usbd_bulk_send_async (uint32_t endpoint, uint32_t *buf, uint32_t txLen)

  Bulk-IN transfser, send data to host through a bulk-in endpont in non-blocking mode.
- kdrv_status_t kdrv_usbd_bulk_receive (uint32_t endpoint, uint32_t *buf, uint32_t *blen, uint32_t timeout_ms)

Kneron Confidential

Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in blocking mode.

- kdrv_status_t kdrv_usbd_bulk_receive_async (uint32_t endpoint, uint32_t *buf, uint32_t blen)

  Bulk-OUT transfser, receive data from the host through a bulk-out endpoint in non-blocking mode.

- kdrv_status_t kdrv_usbd_interrupt_send (uint32_t endpoint, uint32_t *buf, uint32_t txLen, uint32_t timeout_ms)

  Interrupt-IN transfer in blocking mode.

- kdrv_status_t kdrv_usbd_interrupt_receive (uint32_t endpoint, uint32_t *buf, uint32_t *rxLen, uint32_t timeout_ms)

  Interrupt-OUT transfer in blocking mode.

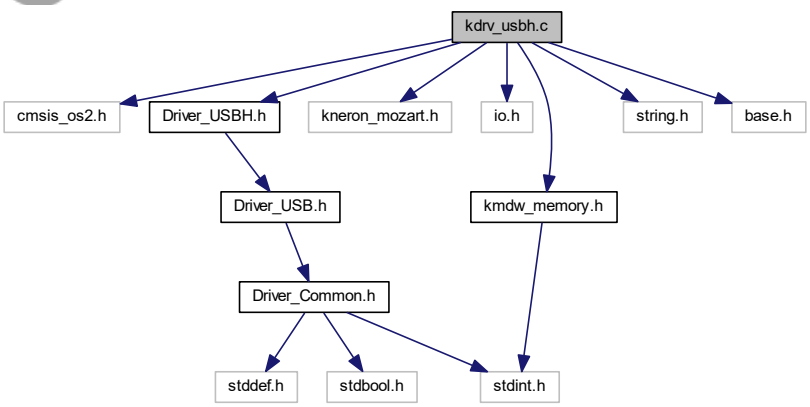
Variables

- kdrv_usbd_setup_packet_t
- kdrv_usbd_endpoint_descriptor_t
- kdrv_usbd_interface_descriptor_t
- kdrv_usbd_config_descriptor_t
- kdrv_usbd_device_descriptor_t
- kdrv_usbd_device_qualifier_descriptor_t

6.80 kdrv_usbh.c File Reference

```
#include "cmsis_os2.h"
#include "Driver_USBH.h"
#include "kneron_mozart.h"
#include "io.h"
#include "kmdw_memory.h"
#include <string.h>
#include "base.h"
```

Include dependency graph for kdrv_usbh.c:



Kneron Confidential

Data Structures

- struct Link_Pointer_DWord
- struct Status_DWord
- struct Buffer_Pointer_DWord
- struct Transaction_DWord
- struct Buffer_Page_DWord
- struct Buffer_Page_0_DWord
- struct Buffer_Page_1_DWord
- struct Buffer_Page_2_DWord
- struct Device_Address_DWord
- struct Mask_DWord

Macros

- #define USBH_DRV_FULL
- #define EHCI_USBCMD 0x10
- #define ASCH_EN BIT5
- #define PSCH_EN BIT4
- #define HC_RESET BIT1
- #define RS BIT0
- #define EHCI_USBSTS 0x14
- #define ASCH_STS BIT15
- #define PSCH_STS BIT14
- #define HCHalted BIT12
- #define INT_OAA BIT5
- #define H_SYSERR BIT4
- #define FRL_ROL BIT3
- #define PO_CHG_DET BIT2
- #define USBERR_INT BIT1
- #define USB_INT BIT0
- #define EHCI_USBINTR 0x18
- #define H_SYSERR_EN BIT4
- #define PO_CHG_INT_EN BIT2
- #define USBERR_INT_EN BIT1
- #define USB_INT_EN BIT0
- #define EHCI_FRINDEX 0x1C
- #define EHCI_PERIODIC_LIST_ADDR 0x24
- #define EHCI_ASYNC_LIST_ADDR 0x28
- #define EHCI_PORTSC 0x30
- #define PO_RESET BIT8
- #define PO_SUSP BIT7
- #define F_PO_RESM BIT6
- #define PO_EN BIT2
- #define CONN_CHG BIT1
- #define CONN_STS BIT0
- #define REG_OTG_CSR 0x80
- #define A_BUS_DROP BIT5
- #define A_BUS_REQ BIT4
- #define REG_OTG_ISR 0x84
- #define OTG_OVC_RW1C BIT10
- #define REG_GLB_INT 0xC4
- #define REG_DEV_CTL 0x100
- #define REG_PHY_TST 0x114

Kneron Confidential

- #define TST_JSTA BIT0
- #define UsbRegRead(reg_offset) inw(USB_FOTG210_PA_BASE + reg_offset)
- #define UsbRegWrite(reg_offset, val) outw(USB_FOTG210_PA_BASE + reg_offset, val)
- #define UsbRegMaskedSet(reg_offset, val) masked_outw(USB_FOTG210_PA_BASE + reg_offset, val, val)
- #define UsbRegMaskedClr(reg_offset, val) masked_outw(USB_FOTG210_PA_BASE + reg_offset, 0, val)
- #define ARM_USBH_DRV_VERSION ARM_DRIVER_VERSION_MAJOR_MINOR(2, 0) /* driver version */
- #define QH_NUM 10
- #define QH_REAL_SIZE 48
- #define QH_ALLOC_IZE 64
- #define qTD_NUM 50
- #define qTD_REAL_SIZE 32
- #define qTD_ALLOC_SIZE 64
- #define MAX_PIPES QH_NUM
- #define FRAME_LIST_SIZE 1024
- #define FRAME_LIST_ROLL_MASK (FRAME_LIST_SIZE - 1)
- #define iTD_MAX_NUM 1024
- #define iTD_SIZE 64
- #define HORI_LINK_TPYE_iTD 0x0
- #define HORI_LINK_TPYE_QH 0x1

Enumerations

- enum EHCI_SCHEDULE_TYPE { ASYNC_SCHEDULE = ASCH_EN, PERIODIC_SCHEDULE = PSCH_EN }

Functions

- struct __attribute__ ((__packed__))

  8-byte setup packet struct
- uint32_t scpu_remap_addr (uint32_t addr)

Variables

- Q_TD
- I_TD
- Queue_Head
- ARM_DRIVER_USBH Driver_USBH0

6.80.1 Macro Definition Documentation

6.80.1.1 A_BUS_DROP

```
#define A_BUS_DROP BIT5
```

Kneron Confidential

6.80.1.2 A_BUS_REQ

```
#define A_BUS_REQ BIT4
```

6.80.1.3 ARM_USBH_DRV_VERSION

```
#define ARM_USBH_DRV_VERSION ARM_DRIVER_VERSION_MAJOR_MINOR(2, 0) /* driver version */
```

6.80.1.4 ASCH_EN

```
#define ASCH_EN BIT5
```

6.80.1.5 ASCH_STS

```
#define ASCH_STS BIT15
```

6.80.1.6 CONN_CHG

```
#define CONN_CHG BIT1
```

6.80.1.7 CONN_STS

```
#define CONN_STS BIT0
```

6.80.1.8 EHCI_ASYNC_LIST_ADDR

```
#define EHCI_ASYNC_LIST_ADDR 0x28
```

6.80.1.9 EHCI_FRINDEX

```
#define EHCI_FRINDEX 0x1C
```

Kneron Confidential

6.80.1.10 EHCI_PERIODIC_LIST_ADDR

```
#define EHCI_PERIODIC_LIST_ADDR 0x24
```

6.80.1.11 EHCI_PORTSC

```
#define EHCI_PORTSC 0x30
```

6.80.1.12 EHCI_USBCMD

```
#define EHCI_USBCMD 0x10
```

6.80.1.13 EHCI_USBINTR

```
#define EHCI_USBINTR 0x18
```

6.80.1.14 EHCI_USBSTS

```
#define EHCI_USBSTS 0x14
```

6.80.1.15 F_PO_RESM

```
#define F_PO_RESM BIT6
```

6.80.1.16 FRAME_LIST_ROLL_MASK

```
#define FRAME_LIST_ROLL_MASK (FRAME_LIST_SIZE - 1)
```

6.80.1.17 FRAME_LIST_SIZE

```
#define FRAME_LIST_SIZE 1024
```

Kneron Confidential

6.80.1.18 FRL_ROL

```
#define FRL_ROL BIT3
```

6.80.1.19 H_SYSERR

```
#define H_SYSERR BIT4
```

6.80.1.20 H_SYSERR_EN

```
#define H_SYSERR_EN BIT4
```

6.80.1.21 HC_RESET

```
#define HC_RESET BIT1
```

6.80.1.22 HCHalted

```
#define HCHalted BIT12
```

6.80.1.23 HORI_LINK_TPYE_iTD

```
#define HORI_LINK_TPYE_iTD 0x0
```

6.80.1.24 HORI_LINK_TPYE_QH

```
#define HORI_LINK_TPYE_QH 0x1
```

6.80.1.25 INT_OAA

```
#define INT_OAA BIT5
```

Kneron Confidential

6.80.1.26 iTD_MAX_NUM

```
#define iTD_MAX_NUM 1024
```

6.80.1.27 iTD_SIZE

```
#define iTD_SIZE 64
```

6.80.1.28 MAX_PIPES

```
#define MAX_PIPES QH_NUM
```

6.80.1.29 OTG_OVC_RW1C

```
#define OTG_OVC_RW1C BIT10
```

6.80.1.30 PO_CHG_DET

```
#define PO_CHG_DET BIT2
```

6.80.1.31 PO_CHG_INT_EN

```
#define PO_CHG_INT_EN BIT2
```

6.80.1.32 PO_EN

```
#define PO_EN BIT2
```

6.80.1.33 PO_RESET

```
#define PO_RESET BIT8
```

Kneron Confidential

6.80.1.34 PO_SUSP

```
#define PO_SUSP BIT7
```

6.80.1.35 PSCH_EN

```
#define PSCH_EN BIT4
```

6.80.1.36 PSCH_STS

```
#define PSCH_STS BIT14
```

6.80.1.37 QH_ALLOC_IZE

```
#define QH_ALLOC_IZE 64
```

6.80.1.38 QH_NUM

```
#define QH_NUM 10
```

6.80.1.39 QH_REAL_SIZE

```
#define QH_REAL_SIZE 48
```

6.80.1.40 qTD_ALLOC_SIZE

```
#define qTD_ALLOC_SIZE 64
```

6.80.1.41 qTD_NUM

```
#define qTD_NUM 50
```

Kneron Confidential

6.80.1.42 qTD_REAL_SIZE

```
#define qTD_REAL_SIZE 32
```

6.80.1.43 REG_DEV_CTL

```
#define REG_DEV_CTL 0x100
```

6.80.1.44 REG_GLB_INT

```
#define REG_GLB_INT 0xC4
```

6.80.1.45 REG_OTG_CSR

```
#define REG_OTG_CSR 0x80
```

6.80.1.46 REG_OTG_ISR

```
#define REG_OTG_ISR 0x84
```

6.80.1.47 REG_PHY_TST

```
#define REG_PHY_TST 0x114
```

6.80.1.48 RS

```
#define RS BIT0
```

6.80.1.49 TST_JSTA

```
#define TST_JSTA BIT0
```

Kneron Confidential

6.80.1.50 USB_INT

```
#define USB_INT BIT0
```

6.80.1.51 USB_INT_EN

```
#define USB_INT_EN BIT0
```

6.80.1.52 USBERR_INT

```
#define USBERR_INT BIT1
```

6.80.1.53 USBERR_INT_EN

```
#define USBERR_INT_EN BIT1
```

6.80.1.54 USBH_DRV_FULL

```
#define USBH_DRV_FULL
```

6.80.1.55 UsbRegMaskedClr

```
#define UsbRegMaskedClr(
            reg_offset,
            val ) masked_outw(USB_FOTG210_PA_BASE + reg_offset, 0, val)
```

6.80.1.56 UsbRegMaskedSet

```
#define UsbRegMaskedSet(
             reg_offset,
             val ) masked_outw(USB_FOTG210_PA_BASE + reg_offset, val, val)
```

Kneron Confidential

6.80.1.57 UsbRegRead

```
#define UsbRegRead(
              reg_offset ) inw(USB_FOTG210_PA_BASE + reg_offset)
```

6.80.1.58 UsbRegWrite

```
#define UsbRegWrite(
              reg_offset,
              val ) outw(USB_FOTG210_PA_BASE + reg_offset, val)
```

6.80.2 Enumeration Type Documentation

6.80.2.1 EHCI_SCHEDULE_TYPE

```
enum EHCI_SCHEDULE_TYPE
```

Enumerator

ASYNC_SCHEDULE	
PERIODIC_SCHEDULE	

6.80.3 Function Documentation

6.80.3.1 scpu_remap_addr()

```
uint32_t scpu_remap_addr (
             uint32_t addr )
```

6.80.4 Variable Documentation

Kneron Confidential

6.80.4.1 Driver_USBH0

```
ARM_DRIVER_USBH Driver_USBH0
```

Initial value:

```
= {
    usbh_driver_get_version,
    usbh_driver_get_capabilities,
    usbh_driver_initialize,
    usbh_driver_uninitialize,
    usbh_driver_power_control,
    usbh_driver_vbus_on_off,
    usbh_driver_port_reset,
    usbh_driver_port_suspend,
    usbh_driver_port_resume,
    usbh_driver_port_get_state,
    usbh_driver_pipe_create,
    usbh_driver_pipe_modify,
    usbh_driver_pipe_delete,
    usbh_driver_pipe_reset,
    usbh_driver_pipe_transfer,
    usbh_driver_pipe_transfer_get_result,
    usbh_driver_pipe_transfer_abort,
    usbh_driver_get_frame_number,
}
```

6.80.4.2 I_TD

```
I_TD
```

6.80.4.3 Q_TD

```
Q_TD
```

6.80.4.4 Queue_Head

```
Queue_Head
```

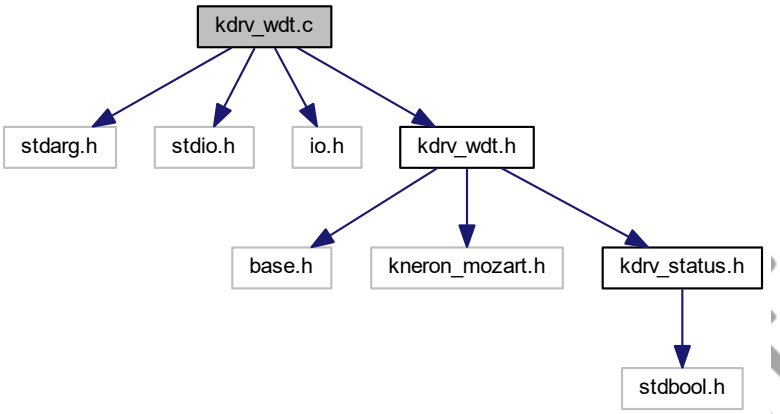
6.81 kdrv_wdt.c File Reference

```
#include <stdarg.h>
#include <stdio.h>
#include "io.h"
```

Kneron Confidential

```
#include "kdrv_wdt.h"
```

Include dependency graph for kdrv_wdt.c:



Functions

- void kdrv_wdt_enable (void)

  watchdog enable
- void kdrv_wdt_disable (void)

  watchdog disable
- void kdrv_wdt_reset (void)

  watchdog reset
- void kdrv_wdt_set_auto_reload (uint32_t value)

  watchdog reload
- void kdrv_wdt_set_clear_status (void)

  watchdog status clear
- void kdrv_wdt_set_int_counter (uint8_t counter)

  watchdog set interrupt counter
- void kdrv_wdt_sys_int_enable (void)

  watchdog interrupt enable
- void kdrv_wdt_sys_int_disable (void)

  watchdog interrupt disable
- void kdrv_wdt_sys_reset_enable (void)

  watchdog reset enable
- void kdrv_wdt_sys_reset_disable (void)

  watchdog reset disable
- bool kdrv_wdt_is_counter_zero ()

  watchdog, is counter zero
- uint32_t kdrv_wdt_read_counter ()

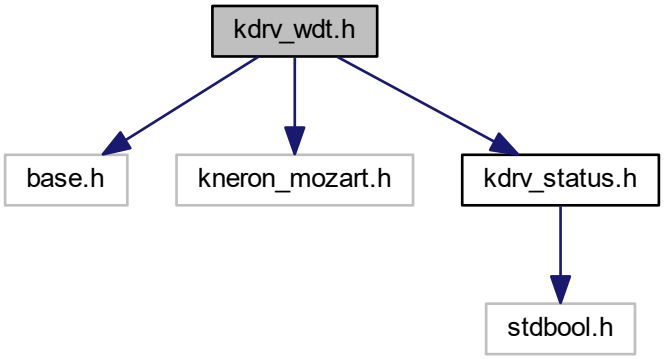
  watchdog read counter

Kneron Confidential

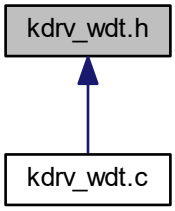
6.82 kdrv_wdt.h File Reference

```
#include "base.h"
#include "kneron_mozart.h"
#include "kdrv_status.h"
```

Include dependency graph for kdrv_wdt.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define KDRV_WDT_BASE WDT_FTWDT010_PA_BASE
- #define REG_WDT_CNT (KDRV_WDT_BASE + 0x00) /* wdt timer counter */
- #define REG_WDT_LOAD (KDRV_WDT_BASE + 0x04) /* auto reload register */
- #define REG_WDT_RST (KDRV_WDT_BASE + 0x08) /* restart register */
- #define REG_WDT_CR (KDRV_WDT_BASE + 0x0C) /* control register */
- #define REG_WDT_STS (KDRV_WDT_BASE + 0x10) /* wdt status register */
- #define REG_WDT_CLR (KDRV_WDT_BASE + 0x14) /* wdt time cleared register */
- #define REG_WDT_INTR_LEN (KDRV_WDT_BASE + 0x18) /* wdt intr length register */
- #define REG_WDT_REV (KDRV_WDT_BASE + 0x1C) /* wdt revision */

- #define WDT_CR_EN BIT(0) /* WDT enable bit, 0: disable, 1: enable */
- #define WDT_CR_RST_EN BIT(1) /* WDT reset bit, 0: disable, 1: enable */
- #define WDT_CR_INT_EN BIT(2) /* WDT int enable bit, 0: disable, 1: enable */
- #define WDT_CR_EXT_EN BIT(3) /* WDT extclk enable bit, 0:disable, 1:enable */
- #define WDT_CR_EXTCLK BIT(4) /* WDT clock source bit, 0:PCLK, 1:EXTCLK */
- #define WDT_RST_AUTO_RELOAD_KEY 0x5AB9

Functions

- void kdrv_wdt_enable (void)

  watchdog enable
- void kdrv_wdt_disable (void)

  watchdog disable
- void kdrv_wdt_reset (void)

  watchdog reset
- void kdrv_wdt_set_auto_reload (uint32_t value)

  watchdog reload
- void kdrv_wdt_sys_int_enable (void)

  watchdog interrupt enable
- void kdrv_wdt_sys_int_disable (void)

  watchdog interrupt disable
- void kdrv_wdt_sys_reset_enable (void)

  watchdog reset enable
- void kdrv_wdt_sys_reset_disable (void)

  watchdog reset disable
- uint32_t kdrv_wdt_read_counter (void)

  watchdog read counter
- void kdrv_wdt_set_clear_status (void)

  watchdog status clear
- void kdrv_wdt_set_int_counter (uint8_t counter)

  watchdog set interrupt counter
- bool kdrv_wdt_is_counter_zero (void)

  watchdog, is counter zero

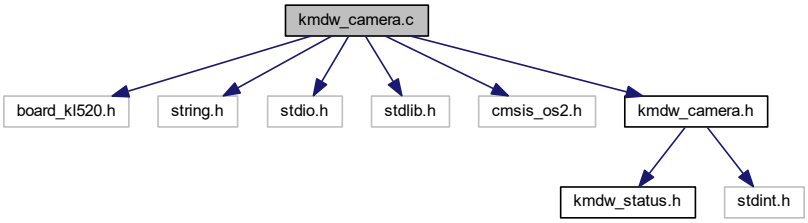
Kneron Confidential

6.83 kmdw_camera.c File Reference

```
#include "board_kl520.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <cmsis_os2.h>
```

```
#include "kmdw_camera.h"
```

Include dependency graph for kmdw_camera.c:



Data Structures

- struct kmdw_camera_s

Functions

- kmdw_status_t kmdw_camera_init (void)
  Initializes camera setting.
- kmdw_status_t kmdw_camera_open (uint32_t cam_idx)
  camera open function
- kmdw_status_t kmdw_camera_close (uint32_t cam_idx)
  camera close function
- kmdw_status_t kmdw_camera_get_device_info (uint32_t cam_idx, struct cam_capability *cap)
  camera get device information function
- kmdw_status_t kmdw_camera_set_frame_format (uint32_t cam_idx, struct cam_format *format)
  camera set frame format function
- kmdw_status_t kmdw_camera_get_frame_format (uint32_t cam_idx, struct cam_format *format)
  camera get frame format function
- kmdw_status_t kmdw_camera_buffer_init (uint32_t cam_idx)
  camera buffer init function
- kmdw_status_t kmdw_camera_start (uint32_t cam_idx)
  camera start function
- kmdw_status_t kmdw_camera_stop (uint32_t cam_idx)
  camera stop function
- kmdw_status_t kmdw_camera_buffer_prepare (uint32_t cam_idx)
  camera buffer prepare function
- kmdw_status_t kmdw_camera_buffer_capture (uint32_t cam_idx, uint32_t *addr, uint32_t *size)
  camera buffer capture function
- kmdw_status_t kmdw_camera_stream_on (uint32_t cam_idx)
  camera streaming on function
- kmdw_status_t kmdw_camera_stream_off (uint32_t cam_idx)
  camera streaming off function
- kmdw_status_t kmdw_camera_set_gain (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
  camera set gain function
- kmdw_status_t kmdw_camera_set_aec (uint32_t cam_idx, struct cam_sensor_aec *aec_p)

Kneron Confidential

camera set ae controller ROI area function

- kmdw_status_t kmdw_camera_set_exp_time (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  camera set exposure time function
- kmdw_status_t kmdw_camera_get_lux (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain, uint8_t *global_gain, uint8_t *y_average)

  camera get lum and other parameter function
- kmdw_status_t kmdw_camera_led_switch (uint32_t cam_idx, uint32_t on)

  camera set nir led on/off function
- kmdw_status_t kmdw_camera_set_mirror (uint32_t cam_idx, uint32_t enable)
- kmdw_status_t kmdw_camera_set_flip (uint32_t cam_idx, uint32_t enable)
- uint32_t kmdw_camera_get_device_id (uint32_t cam_idx)
- kmdw_status_t kmdw_camera_ioctl (uint32_t cam_idx, uint32_t cid, void *data, uint16_t len)

  register specific cam ops with cam_idx
- kmdw_status_t kmdw_camera_controller_register (uint32_t cam_idx, struct cam_ops *cam_ops_p)
- kmdw_status_t kmdw_camera_controller_unregister (uint32_t cam_idx, struct cam_ops *cam_ops_p)

  unregister specific cam ops with cam_idx

Variables

- struct kmdw_camera_s camera_s [IMGSRC_NUM]

6.83.1 Function Documentation

6.83.1.1 kmdw_camera_get_device_id()

```
uint32_t kmdw_camera_get_device_id (
            uint32_t cam_idx )
```

6.83.1.2 kmdw_camera_set_flip()

```
kmdw_status_t kmdw_camera_set_flip (
            uint32_t cam_idx,
            uint32_t enable )
```

6.83.1.3 kmdw_camera_set_mirror()

```
kmdw_status_t kmdw_camera_set_mirror (
            uint32_t cam_idx,
            uint32_t enable )
```

Kneron Confidential

6.83.2 Variable Documentation

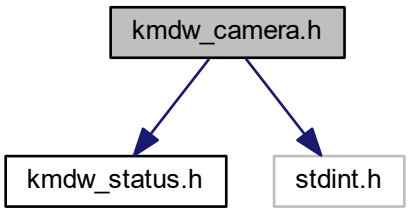
6.83.2.1 camera_s

```
struct kmdw_camera_s camera_s[IMGSRC_NUM]
```

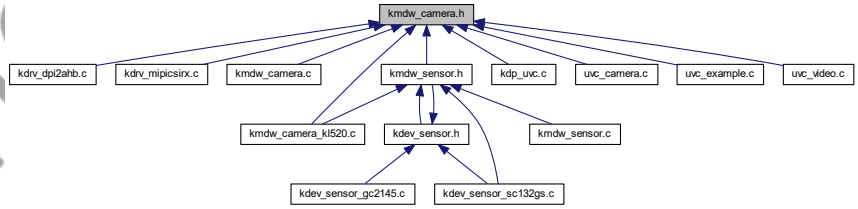
6.84 kmdw_camera.h File Reference

```
#include "kmdw_status.h"
#include <stdint.h>
```

Include dependency graph for kmdw_camera.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct cam_format
- struct cam_capability
- struct cam_sensor_aec
- struct cam_ops

Kneron Confidential

Macros

- #define CAP_VIDEO_CAPTURE 0x00000001 /∗ Is a video capture device ∗/
- #define CAP_STREAMING 0x00000002 /∗ can stream on/off ∗/
- #define CAP_DEVICE_CAPS 0x00000004 /∗ can query capabilities ∗/

Enumerations

- enum {
  CID_SCANNING_MODE = 0x1, CID_AUTO_EXPOSURE_MODE, CID_AUTO_EXPOSURE_PRIORITY,
  CID_EXPOSURE_TIME_ABSOLUTE,
  CID_EXPOSURE_TIME_RELATIVE, CID_FOCUS_ABSOLUTE, CID_FOCUS_RELATIVE, CID_IRIS_ABSOLUTE,
  CID_IRIS_RELATIVE, CID_ZOOM_ABSOLUTE, CID_ZOOM_RELATIVE, CID_PANTILT_ABSOLUTE,
  CID_PANTILT_RELATIVE, CID_ROLL_ABSOLUTE, CID_ROLL_RELATIVE, CID_FOCUS_AUTO,
  CID_PRIVACY, CID_FOCUS_SIMPLE, CID_DIGITAL_WINDOW, CID_REGION_OF_INTEREST,
  CID_BRIGHTNESS, CID_CONTRAST, CID_HUE, CID_SATURATION,
  CID_SHARPNESS, CID_GAMMA, CID_WHITE_BALANCE_TEMPERATURE, CID_WHITE_BALANCE_COMPONENT,
  CID_BACKLIGHT_COMPENSATION, CID_GAIN, CID_POWER_LINE_FREQUENCY, CID_HUE_AUTO,
  CID_WHITE_BALANCE_TEMPERATURE_AUTO, CID_WHITE_BALANCE_COMPONENT_AUTO, CID_DIGITAL_MULTIPLIE
  CID_DIGITAL_MULTIPLIER_LIMIT,
  CID_CONTRAST_AUTO, CID_LIST_ALL = 0xFF }
- enum { KDP_CAM_0, KDP_CAM_1, KDP_CAM_NUM }
- enum camera_state {
  CAMERA_STATE_IDLE = 0, CAMERA_STATE_INITED, CAMERA_STATE_RUNNING, CAMERA_STATE_IN_FDR_INFERE
  CAMERA_STATE_IN_FDR_REGISTRATION, CAMERA_STATE_IN_FDR_AUTO_REGISTRATION,
  CAMERA_STATE_IN_FDR_REGISTRATION_CONFIRM, CAMERA_STATE_IN_FDR_BOTH_REGISTRATION,
  CAMERA_STATE_IN_FDR_BOTH_REGISTRATION_CONFIRM, CAMERA_STATE_IN_FDR_BOTH_INFERENCE,
  CAMERA_STATE_IN_FDR_REGISTRATION_POSE_JUSTIFY }

Functions

- kmdw_status_t kmdw_camera_init (void)

  Initializes camera setting.
- kmdw_status_t kmdw_camera_open (uint32_t cam_idx)

  camera open function
- kmdw_status_t kmdw_camera_close (uint32_t cam_idx)

  camera close function
- kmdw_status_t kmdw_camera_get_device_info (uint32_t cam_idx, struct cam_capability ∗cap)

  camera get device information function
- kmdw_status_t kmdw_camera_set_frame_format (uint32_t cam_idx, struct cam_format ∗format)

  camera set frame format function
- kmdw_status_t kmdw_camera_get_frame_format (uint32_t cam_idx, struct cam_format ∗format)

  camera get frame format function
- kmdw_status_t kmdw_camera_buffer_init (uint32_t cam_idx)

  camera buffer init function
- kmdw_status_t kmdw_camera_start (uint32_t cam_idx)

  camera start function
- kmdw_status_t kmdw_camera_stop (uint32_t cam_idx)

  camera stop function
- kmdw_status_t kmdw_camera_buffer_prepare (uint32_t cam_idx)

  camera buffer prepare function
- kmdw_status_t kmdw_camera_buffer_capture (uint32_t cam_idx, uint32_t ∗addr, uint32_t ∗size)

Kneron Confidential

camera buffer capture function

- kmdw_status_t kmdw_camera_stream_on (uint32_t cam_idx)

  camera streaming on function
- kmdw_status_t kmdw_camera_stream_off (uint32_t cam_idx)

  camera streaming off function
- kmdw_status_t kmdw_camera_set_gain (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  camera set gain function
- kmdw_status_t kmdw_camera_set_aec (uint32_t cam_idx, struct cam_sensor_aec *aec_p)

  camera set ae controller ROI area function
- kmdw_status_t kmdw_camera_set_exp_time (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  camera set exposure time function
- kmdw_status_t kmdw_camera_get_lux (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain, uint8_t *global_gain, uint8_t *y_average)

  camera get lum and other parameter function
- kmdw_status_t kmdw_camera_led_switch (uint32_t cam_idx, uint32_t on)

  camera set nir led on/off function
- kmdw_status_t kmdw_camera_ioctl (uint32_t cam_idx, uint32_t cid, void *data, uint16_t len)

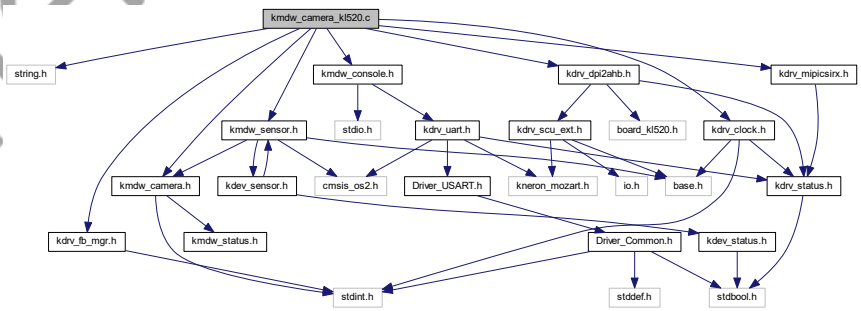
  register specific cam ops with cam_idx
- kmdw_status_t kmdw_camera_controller_register (uint32_t cam_idx, struct cam_ops *cam_ops_p)
- kmdw_status_t kmdw_camera_controller_unregister (uint32_t cam_idx, struct cam_ops *cam_ops_p)

  unregister specific cam ops with cam_idx

6.85 kmdw_camera_kl520.c File Reference

```
#include <string.h>
#include "kmdw_camera.h"
#include "kmdw_sensor.h"
#include "kmdw_console.h"
#include "kdrv_fb_mgr.h"
#include "kdrv_clock.h"
#include "kdrv_dpi2ahb.h"
#include "kdrv_mipicsirx.h"
```

Include dependency graph for kmdw_camera_kl520.c:



Data Structures

- struct kmdw_cam_context

Kneron Confidential

Macros

- #define cam_msg(fmt, ...)

Functions

- void kmdw_cam_mipi_init (uint32_t cam_idx)
- void kmdw_cam_dpi_init (uint32_t cam_idx)
- void kmdw_cam_uvc_init (void)
- void kmdw_cam_port_init (uint32_t cam_idx)
- kmdw_status_t kmdw_cam_kl520_init (void)

Variables

- struct kmdw_cam_context cam_ctx [KDP_CAM_NUM]

6.85.1 Macro Definition Documentation

6.85.1.1 cam_msg

```
#define cam_msg(
                fmt,
                ...  )
```

6.85.2 Function Documentation

6.85.2.1 kmdw_cam_dpi_init()

```
void kmdw_cam_dpi_init (
            uint32_t cam_idx )
```

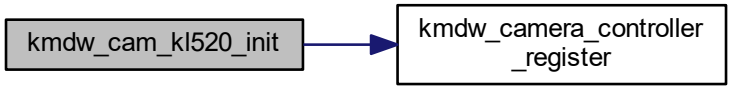
Here is the caller graph for this function:

Kneron Confidential

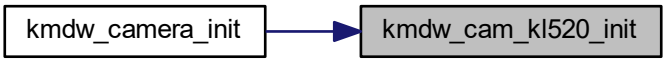
6.85.2.2 kmdw_cam_kl520_init()

```
kmdw_status_t kmdw_cam_kl520_init (
            void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



6.85.2.3 kmdw_cam_mipi_init()

```
void kmdw_cam_mipi_init (
            uint32_t cam_idx )
```

Here is the call graph for this function:

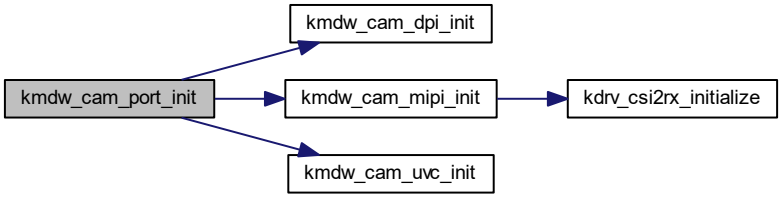
Kneron Confidential

Here is the caller graph for this function:

6.85.2.4 kmdw_cam_port_init()

```
void kmdw_cam_port_init (
            uint32_t cam_idx )
```

Here is the call graph for this function:



6.85.2.5 kmdw_cam_uvc_init()

```
void kmdw_cam_uvc_init (
            void  )
```

Here is the caller graph for this function:

Kneron Confidential

6.85.3 Variable Documentation

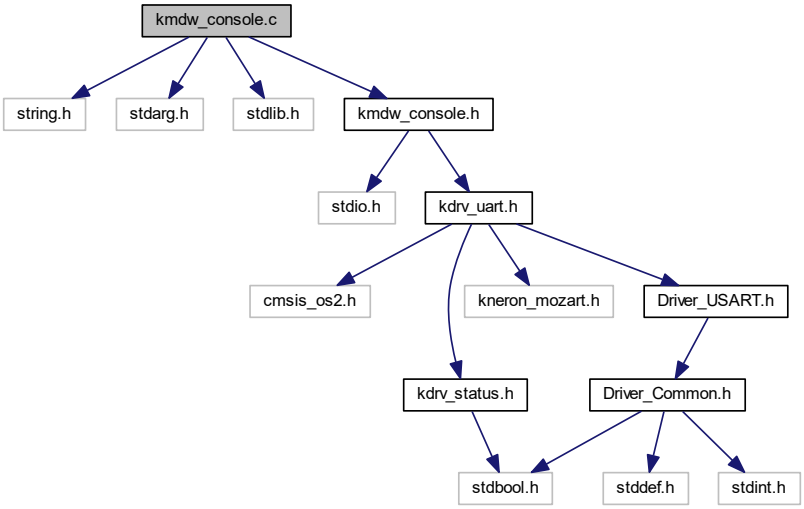
6.85.3.1 cam_ctx

```
struct kmdw_cam_context cam_ctx[KDP_CAM_NUM]
```

6.86 kmdw_console.c File Reference

```
#include <string.h>
#include <stdarg.h>
#include <stdlib.h>
#include "kmdw_console.h"
```

Include dependency graph for kmdw_console.c:



Kneron Confidential

Macros

- #define BACKSP_KEY 0x08
- #define RETURN_KEY 0x0D
- #define DELETE_KEY 0x7F
- #define BELL 0x07

Functions

- void kmdw_console_init (kdrv_uart_dev_id_t uart_dev)
- void kmdw_printf (const char *fmt,...)
- void kmdw_level_printf (int level, const char *fmt,...)
- char kmdw_console_getc (void)
- void kmdw_console_putc (char Ch)
- void kmdw_console_puts (char *str)
- int kmdw_console_echo_gets (char *buf, int len)
- __weak void kdrv_ncpu_set_scpu_debug_lvl (uint32_t lvl)

  Set SCPU debug level.
- __weak void kdrv_ncpu_set_ncpu_debug_lvl (uint32_t lvl)

  Set NCPU debug level.
- void kmdw_console_set_log_level_scpu (uint32_t level)
- uint32_t kmdw_console_get_log_level_scpu (void)
- void kmdw_console_set_log_level_ncpu (uint32_t level)

Variables

- bool UART0_Rx = false
- bool UART0_Tx = false
- const char uart_error_msg [30] = "ERROR: UART Buffer Overrun.\r\n\0"

6.86.1 Macro Definition Documentation

6.86.1.1 BACKSP_KEY

```
#define BACKSP_KEY 0x08
```

6.86.1.2 BELL

```
#define BELL 0x07
```

6.86.1.3 DELETE_KEY

```
#define DELETE_KEY 0x7F
```

6.86.1.4 RETURN_KEY

```
#define RETURN_KEY 0x0D
```

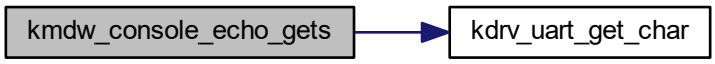
Kneron Confidential

6.86.2 Function Documentation

6.86.2.1 kmdw_console_echo_gets()

```
int kmdw_console_echo_gets (
            char * buf,
            int len )
```

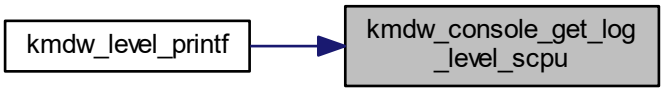
Here is the call graph for this function:



6.86.2.2 kmdw_console_get_log_level_scpu()

```
uint32_t kmdw_console_get_log_level_scpu (
            void  )
```

Here is the caller graph for this function:

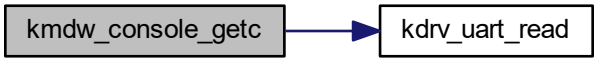


Kneron Confidential

6.86.2.3 kmdw_console_getc()

```
char kmdw_console_getc (
            void  )
```

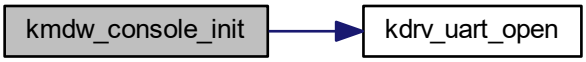
Here is the call graph for this function:



6.86.2.4 kmdw_console_init()

```
void kmdw_console_init (
            kdrv_uart_dev_id_t uart_dev )
```

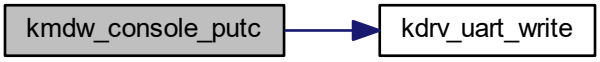
Here is the call graph for this function:



6.86.2.5 kmdw_console_putc()

```
void kmdw_console_putc (
            char Ch )
```

Here is the call graph for this function:



Kneron Confidential

Here is the caller graph for this function:

6.86.2.6 kmdw_console_puts()

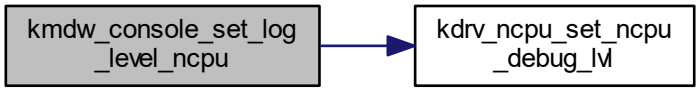
```
void kmdw_console_puts (
            char * str )
```

Here is the call graph for this function:

6.86.2.7 kmdw_console_set_log_level_ncpu()

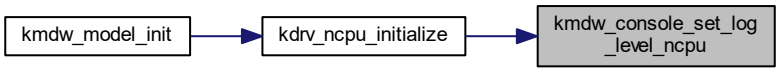
```
void kmdw_console_set_log_level_ncpu (
            uint32_t level )
```

Here is the call graph for this function:



Kneron Confidential

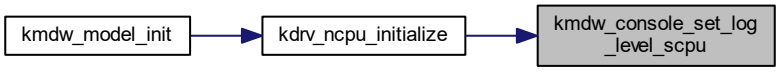
Here is the caller graph for this function:



6.86.2.8 kmdw_console_set_log_level_scpu()

```
void kmdw_console_set_log_level_scpu (
            uint32_t level )
```

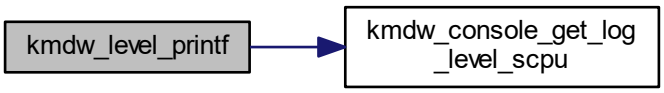
Here is the caller graph for this function:



6.86.2.9 kmdw_level_printf()

```
void kmdw_level_printf (
            int level,
            const char * fmt,
            ... )
```

Here is the call graph for this function:

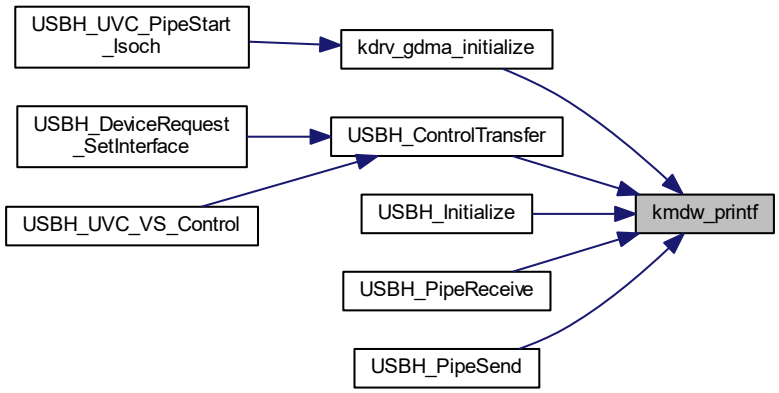


Kneron Confidential

6.86.2.10 kmdw_printf()

```
void kmdw_printf (
            const char * fmt,
             ...  )
```

Here is the caller graph for this function:



6.86.3 Variable Documentation

6.86.3.1 UART0_Rx

```
bool UART0_Rx = false
```

6.86.3.2 UART0_Tx

```
bool UART0_Tx = false
```

6.86.3.3 uart_error_msg

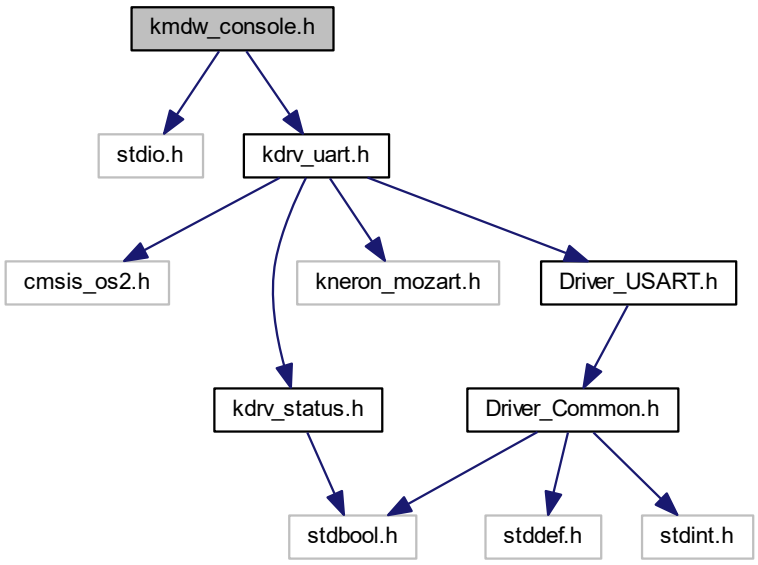
```
const char uart_error_msg[30] = "ERROR: UART Buffer Overrun.\r\n\0"
```

Kneron Confidential

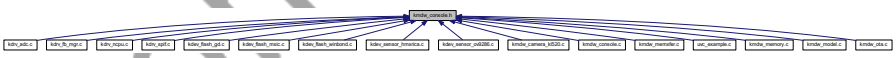
6.87 kmdw_console.h File Reference

```
#include <stdio.h>
#include "kdrv_uart.h"
```

Include dependency graph for kmdw_console.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define LOG_NONE 0
- #define LOG_CRITICAL 1
- #define LOG_ERROR 2
- #define LOG_USER 3
- #define LOG_INFO 4
- #define LOG_TRACE 5
- #define LOG_DBG 6
- #define LOG_PROFILE 9
- #define DSG_NOLF(__format__, ...)
- #define DSG(__format__, ...)
- #define dbg_msg(fmt, ...)
- #define trace_msg(fmt, ...)
- #define info_msg(fmt, ...)
- #define err_msg(fmt, ...)

Kneron Confidential

- #define critical_msg(fmt, ...)
- #define dlog(fmt, ...)
- #define dbg_msg_api(fmt, ...) dbg_msg(fmt, ##__VA_ARGS__)
- #define dbg_msg_app(fmt, ...) dbg_msg(fmt, ##__VA_ARGS__)
- #define dbg_msg_algo(fmt, ...) critical_msg(fmt, ##__VA_ARGS__)
- #define dbg_msg_console(fmt, ...) critical_msg(fmt "\n", ##__VA_ARGS__)
- #define dbg_msg_user(fmt, ...) kmdw_level_printf(LOG_USER, fmt, ##__VA_ARGS__)

Functions

- void kmdw_console_init (kdrv_uart_dev_id_t uart_dev)
- void kmdw_console_set_log_level_scpu (uint32_t level)
- void kmdw_console_set_log_level_ncpu (uint32_t level)
- char kmdw_console_getc (void)
- void kmdw_console_putc (char Ch)
- void kmdw_console_puts (char *str)
- int kmdw_console_echo_gets (char *buf, int len)
- uint32_t kmdw_console_get_log_level_scpu (void)
- void kmdw_printf (const char *fmt,...)
- void kmdw_level_printf (int level, const char *fmt,...)

6.87.1 Macro Definition Documentation

6.87.1.1 critical_msg

```
#define critical_msg(
                fmt,
                ...  )
```

6.87.1.2 dbg_msg

```
#define dbg_msg(
                fmt,
                ...  )
```

6.87.1.3 dbg_msg_algo

```
#define dbg_msg_algo(
                fmt,
                ...  ) critical_msg(fmt, ##__VA_ARGS__)
```

Kneron Confidential

6.87.1.4 dbg_msg_api

```
#define dbg_msg_api(
              fmt,
              ...  ) dbg_msg(fmt, ##__VA_ARGS__)
```

6.87.1.5 dbg_msg_app

```
#define dbg_msg_app(
              fmt,
              ...  ) dbg_msg(fmt, ##__VA_ARGS__)
```

6.87.1.6 dbg_msg_console

```
#define dbg_msg_console(
              fmt,
              ...  ) critical_msg(fmt "\n", ##__VA_ARGS__)
```

6.87.1.7 dbg_msg_user

```
#define dbg_msg_user(
              fmt,
              ...  ) kmdw_level_printf(LOG_USER, fmt, ##__VA_ARGS__)
```

6.87.1.8 dlog

```
#define dlog(
              fmt,
              ...  )
```

6.87.1.9 DSG

```
#define DSG(
             __format__,
              ...  )
```

Kneron Confidential

6.87.1.10 DSG_NOLF

```
#define DSG_NOLF(
              __format__,
              ...  )
```

6.87.1.11 err_msg

```
#define err_msg(
              fmt,
              ...  )
```

6.87.1.12 info_msg

```
#define info_msg(
              fmt,
              ...  )
```

6.87.1.13 LOG_CRITICAL

```
#define LOG_CRITICAL 1
```

6.87.1.14 LOG_DBG

```
#define LOG_DBG 6
```

6.87.1.15 LOG_ERROR

```
#define LOG_ERROR 2
```

6.87.1.16 LOG_INFO

```
#define LOG_INFO 4
```

Kneron Confidential

6.87.1.17 LOG_NONE

```
#define LOG_NONE 0
```

6.87.1.18 LOG_PROFILE

```
#define LOG_PROFILE 9
```

6.87.1.19 LOG_TRACE

```
#define LOG_TRACE 5
```

6.87.1.20 LOG_USER

```
#define LOG_USER 3
```

6.87.1.21 trace_msg

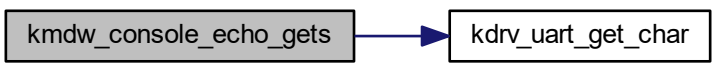
```
#define trace_msg(
              fmt,
              ...  )
```

6.87.2 Function Documentation

6.87.2.1 kmdw_console_echo_gets()

```
int kmdw_console_echo_gets (
          char * buf,
          int len )
```

Here is the call graph for this function:

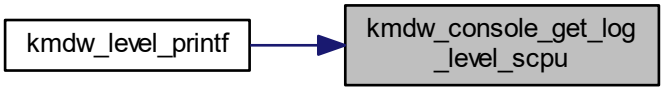


Kneron Confidential

6.87.2.2 kmdw_console_get_log_level_scpu()

```
uint32_t kmdw_console_get_log_level_scpu (
            void  )
```

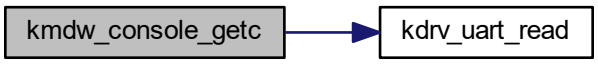
Here is the caller graph for this function:



6.87.2.3 kmdw_console_getc()

```
char kmdw_console_getc (
            void  )
```

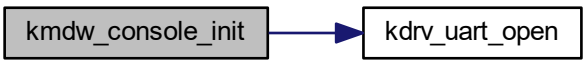
Here is the call graph for this function:



6.87.2.4 kmdw_console_init()

```
void kmdw_console_init (
            kdrv_uart_dev_id_t uart_dev )
```

Here is the call graph for this function:

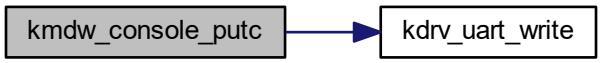


Kneron Confidential

6.87.2.5 kmdw_console_putc()

```
void kmdw_console_putc (
            char Ch )
```

Here is the call graph for this function:

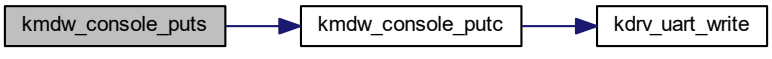


Here is the caller graph for this function:

6.87.2.6 kmdw_console_puts()

```
void kmdw_console_puts (
            char * str )
```

Here is the call graph for this function:

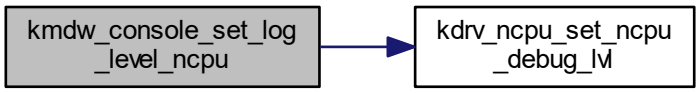


Kneron Confidential

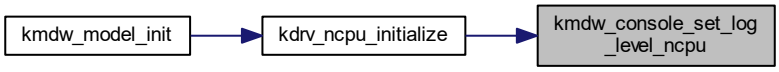
6.87.2.7 kmdw_console_set_log_level_ncpu()

```
void kmdw_console_set_log_level_ncpu (
            uint32_t level )
```

Here is the call graph for this function:



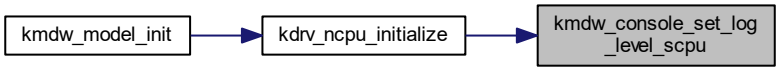
Here is the caller graph for this function:



6.87.2.8 kmdw_console_set_log_level_scpu()

```
void kmdw_console_set_log_level_scpu (
            uint32_t level )
```

Here is the caller graph for this function:

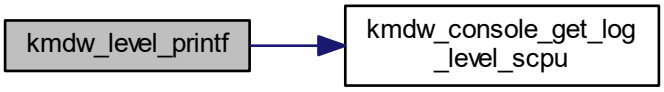


Kneron Confidential

6.87.2.9 kmdw_level_printf()

```
void kmdw_level_printf (
             int level,
             const char * fmt,
              ...  )
```

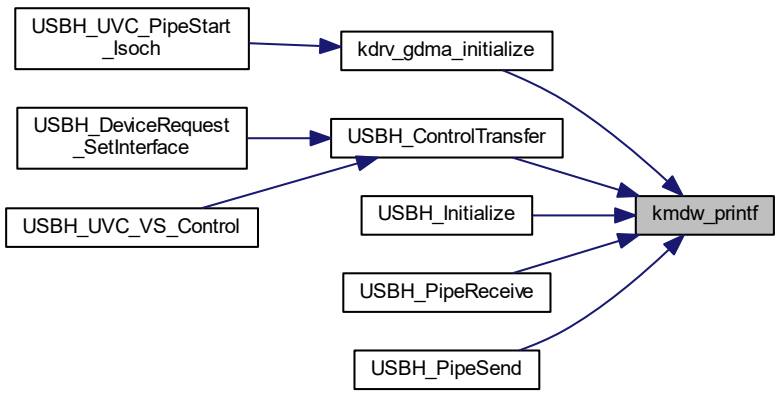
Here is the call graph for this function:



6.87.2.10 kmdw_printf()

```
void kmdw_printf (
             const char * fmt,
              ...  )
```

Here is the caller graph for this function:

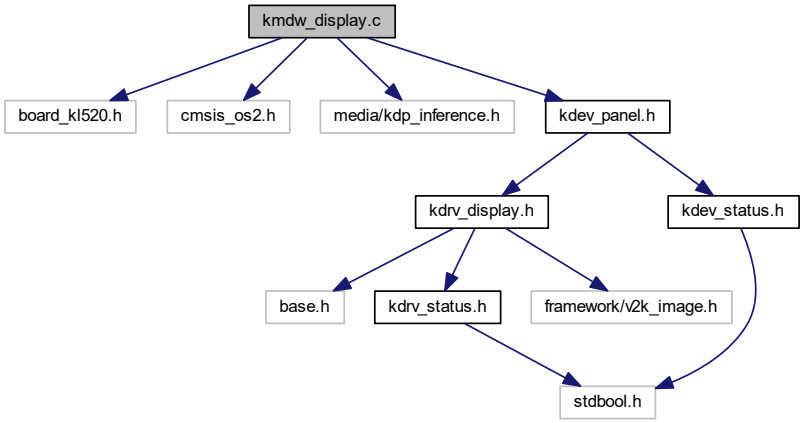


Kneron Confidential

6.88 kmdw_display.c File Reference

```
#include "board_kl520.h"
#include <cmsis_os2.h>
#include "media/kdp_inference.h"
#include "kdev_panel.h"
```

Include dependency graph for kmdw_display.c:



Functions

- int kmdw_video_renderer_buffer_initialize (struct video_input_params *dp_params)

  Initilize display frame buffer.
- uint32_t kmdw_video_renderer_get_buffer_addr (void)

  Get display snapshot frame buffer.
- int kmdw_display_get_params (struct video_input_params *dp_params)

  Get the input parameters of display.
- int kmdw_display_get_device_id (void)
- int kmdw_video_renderer_open (struct video_input_params *params)

  Open a video renderer to display frame buffer.
- int kmdw_video_renderer_set_camera (uint8_t cam_idx)
- int kmdw_video_renderer_start (void)

  Turn on display preview.
- int kmdw_video_renderer_stop (void)

  Turn off display preview.
- int kmdw_display_set_pen_rgb565 (uint16_t color, uint16_t pen_width)

  Set pen width and color.
- int kmdw_display_draw_rect (uint32_t x, uint32_t y, uint32_t width, uint32_t height, uint8_t draw_mode)

  Draw rectangle without filling color on display.
- int kmdw_display_draw_line (uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- int kmdw_display_fill_rect (uint32_t x, uint32_t y, uint32_t width, uint32_t height)

  Draw rectangle with filling color on display.
- int kmdw_display_draw_bitmap (uint32_t x, uint32_t y, uint32_t width, uint32_t height, void *buf)

Kneron Confidential

Draw bitmap on display.

- int kmdw_display_update_draw_fb (uint32_t addr, uint8_t cam_idx)
- int kmdw_display_refresh (void)
- int kmdw_display_test_pattern_gen (bool pat_gen)

  Generate display test image display.
- int kmdw_display_set_backlight (int duty)

  Set display backlight.
- int kmdw_display_initialize (void)

  Initialize display and panel driver.

Variables

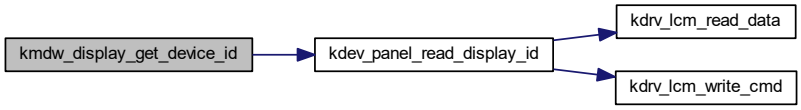
- kdrv_display_t * p_display_drv

6.88.1 Function Documentation

6.88.1.1 kmdw_display_get_device_id()

```
int kmdw_display_get_device_id (
            void  )
```

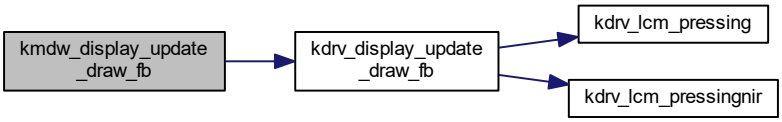
Here is the call graph for this function:



6.88.1.2 kmdw_display_update_draw_fb()

```
int kmdw_display_update_draw_fb (
            uint32_t addr,
            uint8_t cam_idx )
```

Here is the call graph for this function:

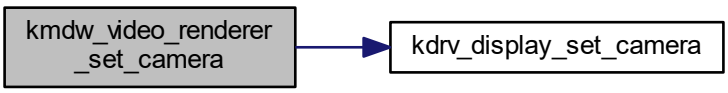


Kneron Confidential

6.88.1.3 kmdw_video_renderer_set_camera()

```
int kmdw_video_renderer_set_camera (
            uint8_t cam_idx )
```

Here is the call graph for this function:



6.88.2 Variable Documentation

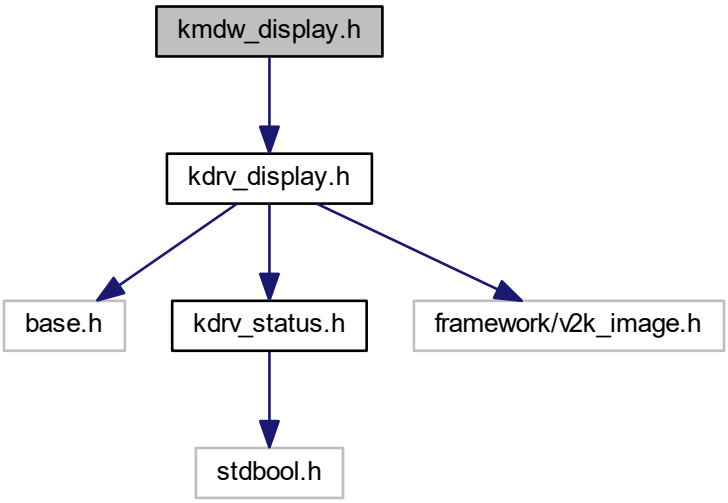
6.88.2.1 p_display_drv

```
kdrv_display_t* p_display_drv
```

6.89 kmdw_display.h File Reference

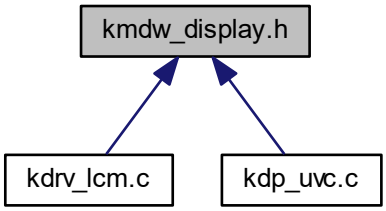
```
#include "kdrv_display.h"
```

Include dependency graph for kmdw_display.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct kmdw_display_panel_drv

  Structure of representing display and panel driver compatibility.

Functions

- int kmdw_display_initialize (void)

  Initialize display and panel driver.
- int kmdw_video_renderer_open (struct video_input_params ∗params)

  Open a video renderer to display frame buffer.
- int kmdw_video_renderer_set_camera (unsigned int cam_idx)

  Set camera source index which be displayed on LCD.
- int kmdw_video_renderer_buffer_initialize (struct video_input_params ∗params)

  Initilize display frame buffer.
- int kmdw_video_renderer_start (void)

  Turn on display preview.
- int kmdw_video_renderer_stop (void)

  Turn off display preview.
- uint32_t kmdw_video_renderer_get_buffer_addr (void)

  Get display snapshot frame buffer.
- int kmdw_display_get_params (struct video_input_params ∗dp_params)

  Get the input parameters of display.
- int kmdw_display_update_draw_fb (uint32_t addr, unsigned int cam_idx)

  Update frame buffer which be used to draw something on display.
- int kmdw_display_set_pen_rgb565 (uint16_t color, uint16_t pen_width)

  Set pen width and color.
- int kmdw_display_draw_rect (uint32_t x, uint32_t y, uint32_t width, uint32_t height, uint8_t draw_mode)

  Draw rectangle without filling color on display.
- int kmdw_display_draw_line (uint32_t xs, uint32_t ys, uint32_t xe, uint32_t ye)

  Draw line on display.
- int kmdw_display_fill_rect (uint32_t x, uint32_t y, uint32_t width, uint32_t height)

  Draw rectangle with filling color on display.
- int kmdw_display_draw_bitmap (uint32_t x, uint32_t y, uint32_t width, uint32_t height, void ∗buf)

Kneron Confidential

Draw bitmap on display.

- int kmdw_display_test_pattern_gen (bool pat_gen)

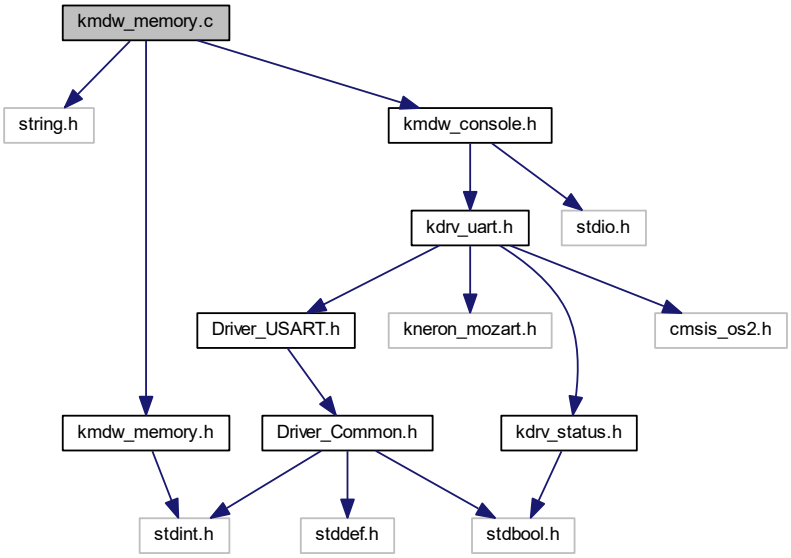
  Generate display test image display.
- int kmdw_display_refresh (void)
- int kmdw_display_set_backlight (int duty)

  Set display backlight.

6.90 kmdw_memory.c File Reference

```
#include <string.h>
#include "kmdw_memory.h"
#include "kmdw_console.h"
```

Include dependency graph for kmdw_memory.c:



Macros

- #define ALIGN_BYTE 16

Functions

- void kmdw_ddr_init (uint32_t start_addr, uint32_t end_addr)

  To initialize available DDR block.
- uint32_t kmdw_ddr_reserve (uint32_t numbyte)

  to allocate DDR memory

6.90.1 Macro Definition Documentation

6.90.1.1 ALIGN_BYTE

```
#define ALIGN_BYTE 16
```

6.90.2 Function Documentation

6.90.2.1 kmdw_ddr_init()

```
void kmdw_ddr_init (
            uint32_t start_addr,
            uint32_t end_addr )
```

To initialize available DDR block.

Parameters

start_addr	the start address of DDR block
eed_addr	the end address of DDR block

6.90.2.2 kmdw_ddr_reserve()

```
uint32_t kmdw_ddr_reserve (
            uint32_t numbyte )
```

to allocate DDR memory

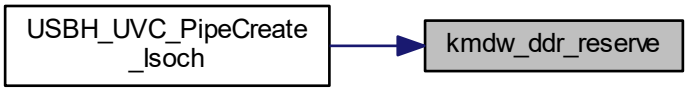
Parameters

numbtye	size in byte

Returns

the address of allocated block

Here is the caller graph for this function:

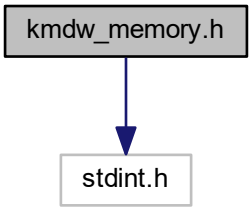


Kneron Confidential

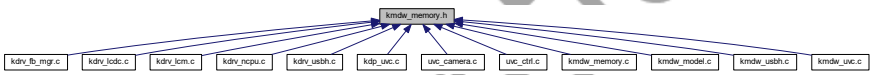
6.91 kmdw_memory.h File Reference

```
#include <stdint.h>
```

Include dependency graph for kmdw_memory.h:



This graph shows which files directly or indirectly include this file:



Functions

- void kmdw_ddr_init (uint32_t start_addr, uint32_t end_addr)

  To initialize available DDR block.
- uint32_t kmdw_ddr_reserve (uint32_t numbyte)

  to allocate DDR memory

6.91.1 Function Documentation

6.91.1.1 kmdw_ddr_init()

```
void kmdw_ddr_init (
            uint32_t start_addr,
            uint32_t end_addr )
```

To initialize available DDR block.

Parameters

start_addr	the start address of DDR block
eed_addr	the end address of DDR block

Kneron Confidential

6.91.1.2 kmdw_ddr_reserve()

```
uint32_t kmdw_ddr_reserve (
            uint32_t numbyte )
```

to allocate DDR memory

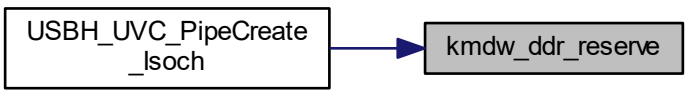
Parameters

numbtye	size in byte

Returns

the address of allocated block

Here is the caller graph for this function:

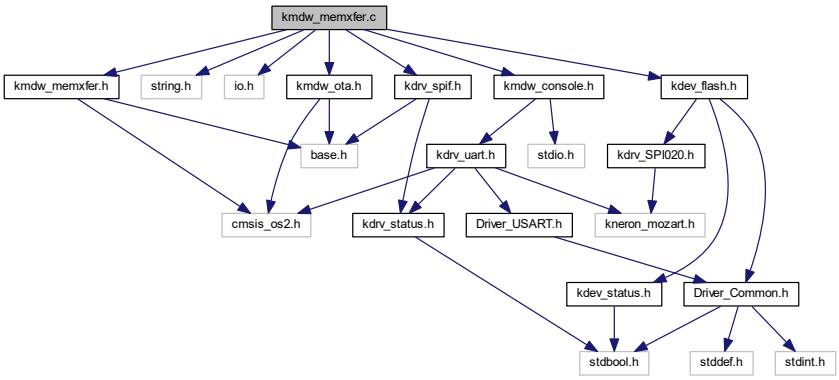


Kneron Confidential

6.92 kmdw_memxfer.c File Reference

```
#include "kmdw_memxfer.h"
#include <string.h>
#include <io.h>
#include "kmdw_ota.h"
#include "kdrv_spif.h"
#include "kdev_flash.h"
#include "kmdw_console.h"
```

Include dependency graph for kmdw_memxfer.c:



Macros

- #define FLASH_WB_DEV 0xEF
- #define FLASH_MXIC_DEV 0xC2
- #define FLASH_Micron_DEV 0x20
- #define FLASH_SIZE_1MB_ID 0x14
- #define MEMXFER_INITED 0x10
- #define MEMXFER_OPS_MASK MEMXFER_OPS_CPU | MEMXFER_OPS_DMA
- #define SPI_QUAD_MODE
- #define SPI_BUS_SPEED_100MHZ 0x01
- #define SPI_BUS_SPEED_50MHZ 0x02
- #define SPI_BUS_SPEED_25MHZ 0x04
- #define SPI_BUS_SPEED SPI_BUS_SPEED_100MHZ
- #define _get_min(x, y) (x < y ? x: y)
- #define BLOCK_SIZE 256
- #define SECTOR_ERASE_SIZE 4096
- #define SECTOR64_ERASE_SIZE 0x10000

Functions

- void kdrv_spif_set_commands (uint32_t cmd0, uint32_t cmd1, uint32_t cmd2, uint32_t cmd3)

  set spi communication commands including read/write by 3/4bytes address, dummy byte size, operation mode, etc
- void kdrv_spif_wait_command_complete (void)

  Check status bit to wait until command completed.
- void kdev_flash_write_control (uint8_t enable)
- void kdrv_spif_check_quad_status_till_ready (void)

  wait quad read command completed and check status till ready
- void kdev_flash_64kErase (uint32_t offset)
- void kdev_flash_read_flash_id (void)
- int kdp_memxfer_init (uint8_t flash_mode, uint8_t mem_mode)
- int kdp_memxfer_flash_to_ddr (uint32_t dst, uint32_t src, size_t bytes)
- int kdp_memxfer_ddr_to_flash (uint32_t dst, uint32_t src, size_t bytes)
- int kdp_memxfer_flash_sector_erase64k (uint32_t addr)

  flash 64k sector erase
- int kdp_memxfer_flash_to_niram (int part_idx)

  load ncpu firmware code from flash to niram
- uint8_t kdp_memxfer_get_flash_device_id (void)

Kneron Confidential

Variables

- const struct s_kdp_memxfer kdp_memxfer_module

6.92.1 Macro Definition Documentation

6.92.1.1 _get_min

```
#define _get_min(
              x,
              y ) ( x < y ?  x:  y )
```

6.92.1.2 BLOCK_SIZE

```
#define BLOCK_SIZE 256
```

6.92.1.3 FLASH_Micron_DEV

```
#define FLASH_Micron_DEV 0x20
```

6.92.1.4 FLASH_MXIC_DEV

```
#define FLASH_MXIC_DEV 0xC2
```

6.92.1.5 FLASH_SIZE_1MB_ID

```
#define FLASH_SIZE_1MB_ID 0x14
```

6.92.1.6 FLASH_WB_DEV

```
#define FLASH_WB_DEV 0xEF
```

Kneron Confidential

6.92.1.7 MEMXFER_INITED

```
#define MEMXFER_INITED 0x10
```

6.92.1.8 MEMXFER_OPS_MASK

```
#define MEMXFER_OPS_MASK MEMXFER_OPS_CPU | MEMXFER_OPS_DMA
```

6.92.1.9 SECTOR64_ERASE_SIZE

```
#define SECTOR64_ERASE_SIZE 0x10000
```

6.92.1.10 SECTOR_ERASE_SIZE

```
#define SECTOR_ERASE_SIZE 4096
```

6.92.1.11 SPI_BUS_SPEED

```
#define SPI_BUS_SPEED SPI_BUS_SPEED_100MHZ
```

6.92.1.12 SPI_BUS_SPEED_100MHZ

```
#define SPI_BUS_SPEED_100MHZ 0x01
```

6.92.1.13 SPI_BUS_SPEED_25MHZ

```
#define SPI_BUS_SPEED_25MHZ 0x04
```

6.92.1.14 SPI_BUS_SPEED_50MHZ

```
#define SPI_BUS_SPEED_50MHZ 0x02
```

Kneron Confidential

6.92.1.15 SPI_QUAD_MODE

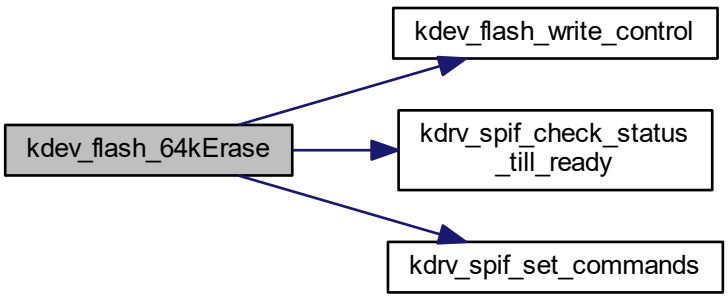
```
#define SPI_QUAD_MODE
```

6.92.2 Function Documentation

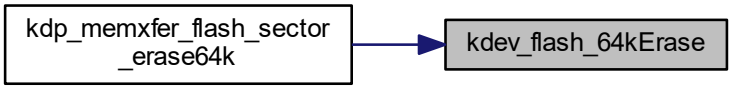
6.92.2.1 kdev_flash_64kErase()

```
void kdev_flash_64kErase (
            uint32_t offset )
```

Here is the call graph for this function:



Here is the caller graph for this function:

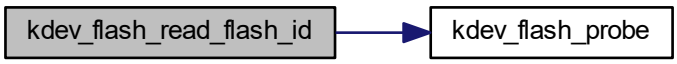


Kneron Confidential

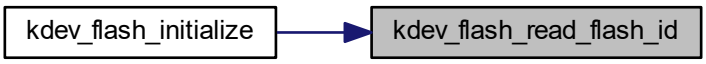
6.92.2.2 kdev_flash_read_flash_id()

```
void kdev_flash_read_flash_id (
            void  )
```

Here is the call graph for this function:



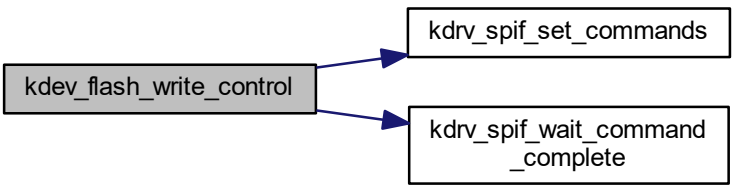
Here is the caller graph for this function:



6.92.2.3 kdev_flash_write_control()

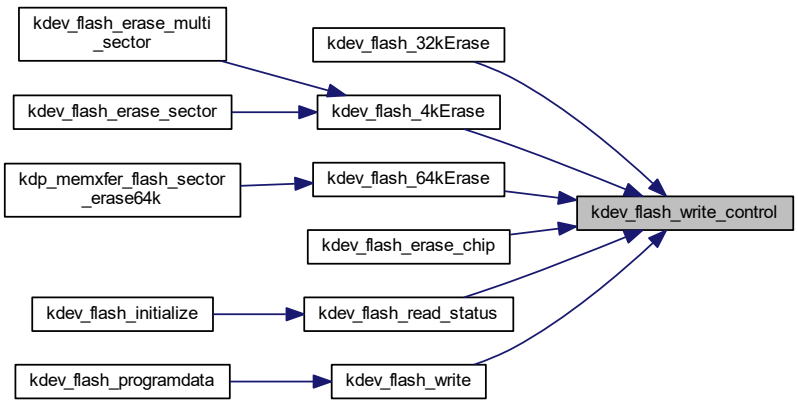
```
void kdev_flash_write_control (
            uint8_t enable )
```

Here is the call graph for this function:



Kneron Confidential

Here is the caller graph for this function:



6.92.2.4 kdp_memxfer_ddr_to_flash()

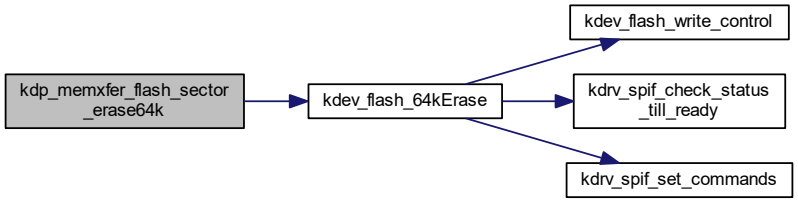
```
int kdp_memxfer_ddr_to_flash (
            uint32_t dst,
            uint32_t src,
            size_t bytes )
```

6.92.2.5 kdp_memxfer_flash_sector_erase64k()

```
int kdp_memxfer_flash_sector_erase64k (
            uint32_t addr )
```

flash 64k sector erase

Here is the call graph for this function:

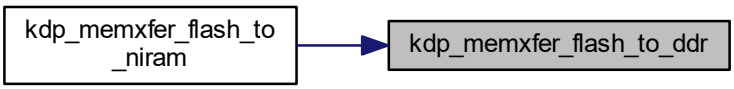


Kneron Confidential

6.92.2.6 kdp_memxfer_flash_to_ddr()

```
int kdp_memxfer_flash_to_ddr (
            uint32_t dst,
            uint32_t src,
            size_t bytes )
```

Here is the caller graph for this function:

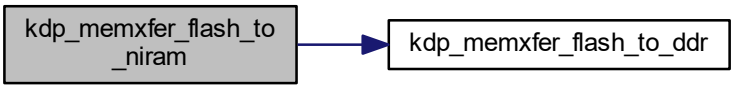


6.92.2.7 kdp_memxfer_flash_to_niram()

```
int kdp_memxfer_flash_to_niram (
            int part_idx )
```

load ncpu firmware code from flash to niram

Here is the call graph for this function:



6.92.2.8 kdp_memxfer_get_flash_device_id()

```
uint8_t kdp_memxfer_get_flash_device_id (
            void  )
```

Kneron Confidential

6.92.2.9 kdp_memxfer_init()

```
int kdp_memxfer_init (
            uint8_t flash_mode,
            uint8_t mem_mode )
```

6.92.3 Variable Documentation

6.92.3.1 kdp_memxfer_module

```
const struct s_kdp_memxfer kdp_memxfer_module
```

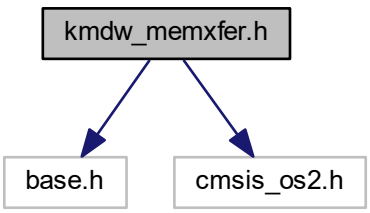
Initial value:

```
= {
    kdp_memxfer_init,
    kdp_memxfer_flash_to_ddr,
    kdp_memxfer_ddr_to_flash,
    kdp_memxfer_flash_sector_erase64k,
    kdp_memxfer_flash_to_niram,
    kdp_memxfer_get_flash_device_id,
}
```

6.93 kmdw_memxfer.h File Reference

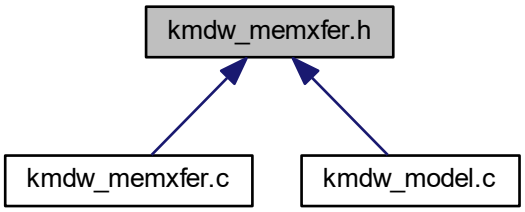
```
#include "base.h"
#include "cmsis_os2.h"
```

Include dependency graph for kmdw_memxfer.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct s_kdp_memxfer

Macros

- #define MEMXFER_OPS_NONE 0x00
- #define MEMXFER_OPS_CPU 0x01
- #define MEMXFER_OPS_DMA 0x02

Variables

- const struct s_kdp_memxfer kdp_memxfer_module

6.93.1 Macro Definition Documentation

6.93.1.1 MEMXFER_OPS_CPU

```
#define MEMXFER_OPS_CPU 0x01
```

6.93.1.2 MEMXFER_OPS_DMA

```
#define MEMXFER_OPS_DMA 0x02
```

Kneron Confidential

6.93.1.3 MEMXFER_OPS_NONE

```
#define MEMXFER_OPS_NONE 0x00
```

6.93.2 Variable Documentation

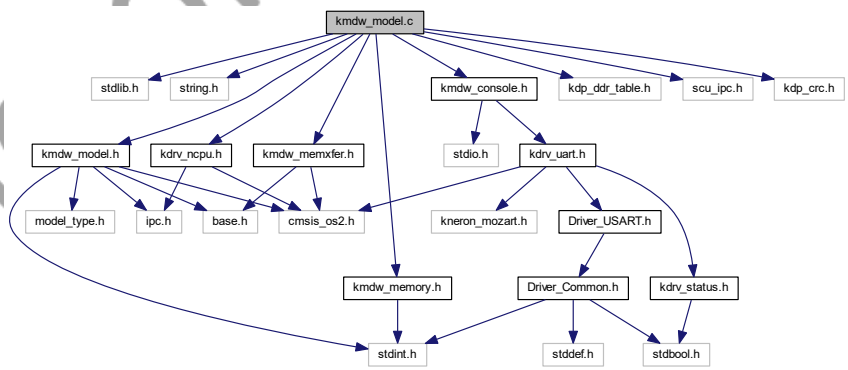
6.93.2.1 kdp_memxfer_module

```
const struct s_kdp_memxfer kdp_memxfer_module
```

6.94 kmdw_model.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include "kdrv_ncpu.h"
#include "kmdw_model.h"
#include "kmdw_console.h"
#include "kmdw_memxfer.h"
#include "kmdw_memory.h"
#include "kdp_ddr_table.h"
#include "scu_ipc.h"
#include "kdp_crc.h"
```

Include dependency graph for kmdw_model.c:



Data Structures

- struct kmdw_model_data_t
- struct kmdw_img_data_t

Kneron Confidential

Macros

- #define DEBUG 0
- #define ENABLE_CHKSUM_ON_MODEL 0
- #define FLAG_MODELMGR_ABORT BIT(29)
- #define FLAG_MODELMGR_FROM_NCPU BIT(30)
- #define FLAG_MODELMGR_FROM_NPU BIT(28)

Functions

- void kmdw_model_init (void)

  Init model functionality.
- int32_t kmdw_model_load_model (int8_t model_info_index_p)

  A wrapper of load_model.
- void kmdw_model_reload_model_info (bool from_ddr)

  A wrapper of load_model_info.
- void kmdw_model_refresh_models (void)

  Refresh all models.
- int32_t kmdw_model_config_result (osEventFlagsId_t result_evt, uint32_t result_evt_flag)

  Specify output address for model run in ncpu/npu !!! must be called after kmdw_model_config_model()
- void kmdw_model_config_img (struct kdp_img_cfg *img_cfg, struct kdp_crop_box_s *crop_box, struct kdp_pad_value_s *pad_values, void *ext_param)

  Config model image.
- struct kdp_img_raw_s * kmdw_model_get_raw_img (int idx)

  Get raw image config.
- int kmdw_model_run (const char *tag, void *output, uint32_t model_type, bool dme)

  Run model.
- void kmdw_model_abort (void)

  abort NCPU/NPU operations (stop sending task to ncpu/npu)
- struct kdp_model_s * kmdw_model_get_model_info (int model_idx_p)

  Output model_info of specified index.

Variables

- const struct s_kdp_memxfer kdp_memxfer_module

6.94.1 Macro Definition Documentation

6.94.1.1 DEBUG

```
#define DEBUG 0
```

Kneron Confidential

6.94.1.2 ENABLE_CHKSUM_ON_MODEL

```
#define ENABLE_CHKSUM_ON_MODEL 0
```

6.94.1.3 FLAG_MODELMGR_ABORT

```
#define FLAG_MODELMGR_ABORT BIT(29)
```

6.94.1.4 FLAG_MODELMGR_FROM_NCPU

```
#define FLAG_MODELMGR_FROM_NCPU BIT(30)
```

6.94.1.5 FLAG_MODELMGR_FROM_NPU

```
#define FLAG_MODELMGR_FROM_NPU BIT(28)
```

6.94.2 Function Documentation

6.94.2.1 kmdw_model_abort()

```
void kmdw_model_abort (
            void  )
```

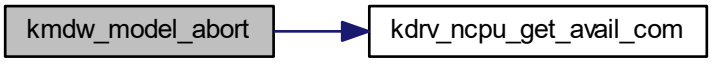
abort NCPU/NPU operations (stop sending task to ncpu/npu)

Abort model execution.

Returns

n/a

Here is the call graph for this function:



Kneron Confidential

6.94.2.2 kmdw_model_config_img()

```
void kmdw_model_config_img (
            struct kdp_img_cfg * img_cfg,
            struct kdp_crop_box_s * crop_box,
            struct kdp_pad_value_s * pad_values,
            void * ext_param )
```

Config model image.

Parameters

img_cfg	image config
crop_box	image crop config
pad_values	image padding config
ext_param	extra param

Here is the call graph for this function:

6.94.2.3 kmdw_model_config_result()

```
int32_t kmdw_model_config_result (
            osEventFlagsId_t result_evt,
            uint32_t result_evt_flag )
```

Specify output address for model run in ncpu/npu !!! must be called after kmdw_model_config_model()

Returns

always 0

Here is the call graph for this function:

Kneron Confidential

6.94.2.4 kmdw_model_get_model_info()

```
struct kdp_model_s* kmdw_model_get_model_info (
            int idx_p )
```

Output model_info of specified index.

Parameters

in	*idx_p*	the index of programmed models

Returns

model_info defined in ipc.h

6.94.2.5 kmdw_model_get_raw_img()

```
struct kdp_img_raw_s* kmdw_model_get_raw_img (
            int idx )
```

Get raw image config.

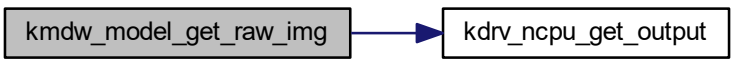
Parameters

idx	image index

Returns

raw image config

Here is the call graph for this function:



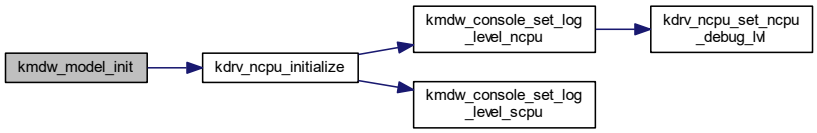
Kneron Confidential

6.94.2.6 kmdw_model_init()

```
void kmdw_model_init (
            void  )
```

Init model functionality.

Here is the call graph for this function:



6.94.2.7 kmdw_model_load_model()

```
int32_t kmdw_model_load_model (
            int8_t model_info_index_p )
```

A wrapper of load_model.

Parameters

model_index_p	model info index, 0-n: info_index of model to load -1 means to load all models

Returns

always 0

6.94.2.8 kmdw_model_refresh_models()

```
void kmdw_model_refresh_models (
            void  )
```

Refresh all models.

6.94.2.9 kmdw_model_reload_model_info()

```
void kmdw_model_reload_model_info (
            bool from_ddr )
```

A wrapper of load_model_info.

Kneron Confidential

Parameters

in	*is_model_from_ddr*	if model is from ddr/host command

6.94.2.10 kmdw_model_run()

```
int kmdw_model_run (
            const char * tag,
            void * output,
            uint32_t model_type,
            bool dme )
```

Run model.

Parameters

tag	model tag
output	model output
model_type	model type
dme	DME mode

Returns

kmdw_model_rc

6.94.3 Variable Documentation

6.94.3.1 kdp_memxfer_module

```
const struct s_kdp_memxfer kdp_memxfer_module
```

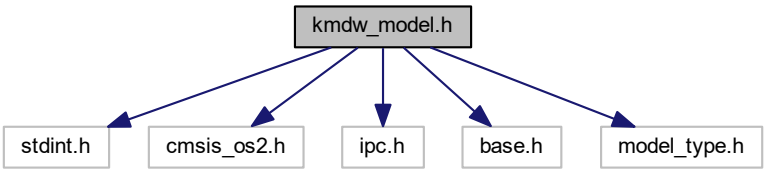
6.95 kmdw_model.h File Reference

```
#include <stdint.h>
#include "cmsis_os2.h"
#include "ipc.h"
#include "base.h"
```

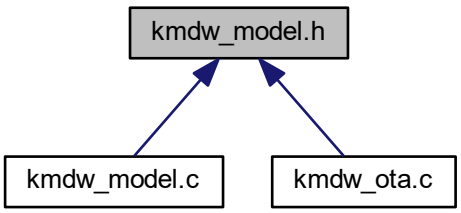
Kneron Confidential

```
#include "model_type.h"
```

Include dependency graph for kmdw_model.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define KMDW_MODEL_ALL_MODELS -1
- #define MODEL_MGR_SIZE_MODEL_INFO 224
- #define MODEL_MGR_SIZE_MODEL_POOL 3572576
- #define MODEL_MGR_FLASH_ADDR_HEAD 0x00300000
- #define MODEL_MGR_FLASH_ADDR_MODEL_COUNT (MODEL_MGR_FLASH_ADDR_HEAD)
- #define MODEL_MGR_FLASH_ADDR_MODEL_INFO (MODEL_MGR_FLASH_ADDR_MODEL_COUNT + 4)
- #define MODEL_MGR_FLASH_ADDR_MODEL_DDR_END_ADDR (MODEL_MGR_FLASH_ADDR_MODEL_INFO + MODEL_MGR_SIZE_MODEL_INFO)
- #define MODEL_MGR_FLASH_ADDR_MODEL_POOL (MODEL_MGR_FLASH_ADDR_HEAD + 0x1000)
- #define MODEL_INFO_CRC16_VALUE 0x5341
- #define MODEL_INFO_SUM32_VALUE 0x00eee074
- #define MODEL0_FD_CRC16_VALUE 0x939f /∗ original: 0x4318 ∗/
- #define MODEL1_LM_CRC16_VALUE 0x1cd0 /∗ original: 0xfce0 ∗/
- #define MODEL3_FR_CRC16_VALUE 0x5df0 /∗ original: 0xb359 ∗/
- #define MODEL0_FD_SUM32_VALUE 0xf9e571d8
- #define MODEL1_LM_SUM32_VALUE 0x3f96c4fc
- #define MODEL3_FR_SUM32_VALUE 0xbc998f45

Kneron Confidential

Enumerations

- enum kmdw_model_rc { KMDW_MODEL_RUN_RC_ABORT = 10, KMDW_MODEL_RUN_RC_ERROR = 11, KMDW_MODEL_RUN_RC_END }

Functions

- void kmdw_model_init (void)

  Init model functionality.
- int32_t kmdw_model_load_model (int8_t model_info_index_p)

  A wrapper of load_model.
- void kmdw_model_reload_model_info (bool from_ddr)

  A wrapper of load_model_info.
- void kmdw_model_refresh_models (void)

  Refresh all models.
- struct kdp_model_s * kmdw_model_get_model_info (int idx_p)

  Output model_info of specified index.
- int32_t kmdw_model_config_result (osEventFlagsId_t result_evt, uint32_t result_evt_flag)

  Specify output address for model run in ncpu/npu !!! must be called after kmdw_model_config_model()
- void kmdw_model_config_img (struct kdp_img_cfg *img_cfg, struct kdp_crop_box_s *crop_box, struct kdp_pad_value_s *pad_values, void *ext_param)

  Config model image.
- struct kdp_img_raw_s * kmdw_model_get_raw_img (int idx)

  Get raw image config.
- int kmdw_model_run (const char *tag, void *output, uint32_t model_type, bool dme)

  Run model.
- void kmdw_model_abort (void)

  Abort model execution.

6.95.1 Macro Definition Documentation

6.95.1.1 KMDW_MODEL_ALL_MODELS

```
#define KMDW_MODEL_ALL_MODELS -1
```

6.95.1.2 MODEL0_FD_CRC16_VALUE

```
#define MODEL0_FD_CRC16_VALUE 0x939f /* original:  0x4318 */
```

Kneron Confidential

6.95.1.3 MODEL0_FD_SUM32_VALUE

```
#define MODEL0_FD_SUM32_VALUE 0xf9e571d8
```

6.95.1.4 MODEL1_LM_CRC16_VALUE

```
#define MODEL1_LM_CRC16_VALUE 0x1cd0 /* original:  0xfce0 */
```

6.95.1.5 MODEL1_LM_SUM32_VALUE

```
#define MODEL1_LM_SUM32_VALUE 0x3f96c4fc
```

6.95.1.6 MODEL3_FR_CRC16_VALUE

```
#define MODEL3_FR_CRC16_VALUE 0x5df0 /* original:  0xb359 */
```

6.95.1.7 MODEL3_FR_SUM32_VALUE

```
#define MODEL3_FR_SUM32_VALUE 0xbc998f45
```

6.95.1.8 MODEL_INFO_CRC16_VALUE

```
#define MODEL_INFO_CRC16_VALUE 0x5341
```

6.95.1.9 MODEL_INFO_SUM32_VALUE

```
#define MODEL_INFO_SUM32_VALUE 0x00eee074
```

6.95.1.10 MODEL_MGR_FLASH_ADDR_HEAD

```
#define MODEL_MGR_FLASH_ADDR_HEAD 0x00300000
```

Kneron Confidential

6.95.1.11 MODEL_MGR_FLASH_ADDR_MODEL_COUNT

```
#define MODEL_MGR_FLASH_ADDR_MODEL_COUNT (MODEL_MGR_FLASH_ADDR_HEAD)
```

6.95.1.12 MODEL_MGR_FLASH_ADDR_MODEL_DDR_END_ADDR

```
#define MODEL_MGR_FLASH_ADDR_MODEL_DDR_END_ADDR (MODEL_MGR_FLASH_ADDR_MODEL_INFO + MODEL_MGR_SIZE_MODEL_INFO)
```

6.95.1.13 MODEL_MGR_FLASH_ADDR_MODEL_INFO

```
#define MODEL_MGR_FLASH_ADDR_MODEL_INFO (MODEL_MGR_FLASH_ADDR_MODEL_COUNT + 4)
```

6.95.1.14 MODEL_MGR_FLASH_ADDR_MODEL_POOL

```
#define MODEL_MGR_FLASH_ADDR_MODEL_POOL (MODEL_MGR_FLASH_ADDR_HEAD + 0x1000)
```

6.95.1.15 MODEL_MGR_SIZE_MODEL_INFO

```
#define MODEL_MGR_SIZE_MODEL_INFO 224
```

6.95.1.16 MODEL_MGR_SIZE_MODEL_POOL

```
#define MODEL_MGR_SIZE_MODEL_POOL 3572576
```

6.95.2 Enumeration Type Documentation

6.95.2.1 kmdw_model_rc

```
enum kmdw_model_rc
```

Kneron Confidential

Enumerator

KMDW_MODEL_RUN_RC_ABORT	
KMDW_MODEL_RUN_RC_ERROR	
KMDW_MODEL_RUN_RC_END	

6.95.3 Function Documentation

6.95.3.1 kmdw_model_abort()

```
void kmdw_model_abort (
            void  )
```

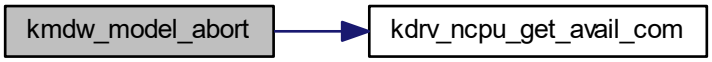
Abort model execution.

Abort model execution.

Returns

n/a

Here is the call graph for this function:



6.95.3.2 kmdw_model_config_img()

```
void kmdw_model_config_img (
            struct kdp_img_cfg * img_cfg,
            struct kdp_crop_box_s * crop_box,
            struct kdp_pad_value_s * pad_values,
            void * ext_param )
```

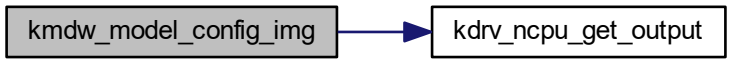
Config model image.

Kneron Confidential

Parameters

img_cfg	image config
crop_box	image crop config
pad_values	image padding config
ext_param	extra param

Here is the call graph for this function:



6.95.3.3 kmdw_model_config_result()

```
int32_t kmdw_model_config_result (
            osEventFlagsId_t result_evt,
            uint32_t result_evt_flag )
```

Specify output address for model run in ncpu/npu !!! must be called after kmdw_model_config_model()

Returns

always 0

Here is the call graph for this function:

6.95.3.4 kmdw_model_get_model_info()

```
struct kdp_model_s* kmdw_model_get_model_info (
            int idx_p )
```

Output model_info of specified index.

Kneron Confidential

Parameters

in	*idx_p*	the index of programmed models

Returns

model_info defined in ipc.h

6.95.3.5 kmdw_model_get_raw_img()

```
struct kdp_img_raw_s* kmdw_model_get_raw_img (
            int idx )
```

Get raw image config.

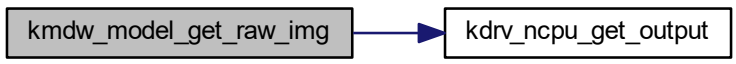
Parameters

idx	image index

Returns

raw image config

Here is the call graph for this function:



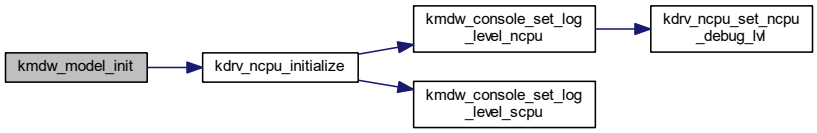
6.95.3.6 kmdw_model_init()

```
void kmdw_model_init (
            void  )
```

Init model functionality.

Kneron Confidential

Here is the call graph for this function:



6.95.3.7 kmdw_model_load_model()

```
int32_t kmdw_model_load_model (
            int8_t model_info_index_p )
```

A wrapper of load_model.

Parameters

model_index_p	model info index, 0-n: info_index of model to load -1 means to load all models

Returns

always 0

6.95.3.8 kmdw_model_refresh_models()

```
void kmdw_model_refresh_models (
            void  )
```

Refresh all models.

6.95.3.9 kmdw_model_reload_model_info()

```
void kmdw_model_reload_model_info (
            bool from_ddr )
```

A wrapper of load_model_info.

Kneron Confidential

Parameters

in	*is_model_from_ddr*	if model is from ddr/host command

6.95.3.10 kmdw_model_run()

```
int kmdw_model_run (
            const char * tag,
            void * output,
            uint32_t model_type,
            bool dme )
```

Run model.

Parameters

tag	model tag
output	model output
model_type	model type
dme	DME mode

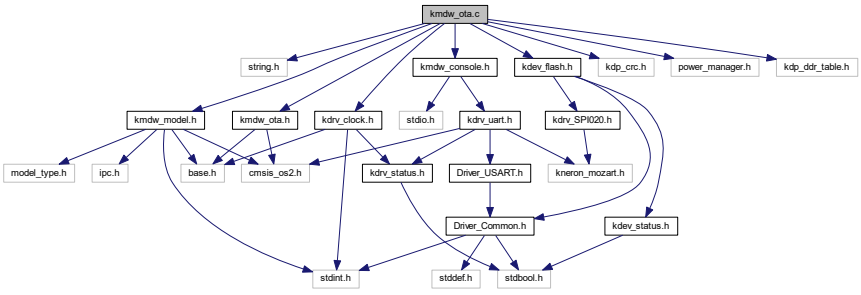
Returns

kmdw_model_rc

6.96 kmdw_ota.c File Reference

```
#include <string.h>
#include "kmdw_ota.h"
#include "kdp_crc.h"
#include "kdev_flash.h"
#include "kmdw_model.h"
#include "kmdw_console.h"
#include "kdrv_clock.h"
#include "power_manager.h"
#include "kdp_ddr_table.h"
```

Include dependency graph for kmdw_ota.c:



Kneron Confidential

Data Structures

- struct ota_boot_cfg_item_t
- struct ota_boot_cfg_t

Macros

- #define VERIFY_BLK_SZ 0x1000
- #define MODEL_INFO_FLASH_ADDR 0x00300000
- #define MODEL_ALL_BIN_FLASH_ADDR 0x00301000
- #define POLY 0x8408
- #define BOOT_STATE_CONFIRMED 0x1
- #define BOOT_STATE_FIRST_BOOT 0x2
- #define BOOT_STATE_POST_FIRST_BOOT 0x4
- #define BOOT_STATE_NOT_CONFIRMED 0x8
- #define MAX(a, b) (((a) < (b)) ? (b) : (a))
- #define MAX_BOOT_SEQ 0x7ffffff0

Functions

- int kmdw_ota_init (u8 *tmp_buf, FnReadData fn_read)
- int kmdw_ota_get_active_scpu_partition ()

  get active SCPU partition ID
- int kmdw_ota_get_active_ncpu_partition ()

  get active NCPU partition ID
- int kmdw_ota_update_scpu ()

  Update SCPU firmware.
- int kmdw_ota_update_ncpu ()

  Update NCPU firmware.
- int kmdw_ota_update_model (u32 size)
- int kmdw_ota_switch_active_partition (u32 partition)

Variables

- int flashing

6.96.1 Macro Definition Documentation

6.96.1.1 BOOT_STATE_CONFIRMED

```
#define BOOT_STATE_CONFIRMED 0x1
```

Kneron Confidential

6.96.1.2 BOOT_STATE_FIRST_BOOT

```
#define BOOT_STATE_FIRST_BOOT 0x2
```

6.96.1.3 BOOT_STATE_NOT_CONFIRMED

```
#define BOOT_STATE_NOT_CONFIRMED 0x8
```

6.96.1.4 BOOT_STATE_POST_FIRST_BOOT

```
#define BOOT_STATE_POST_FIRST_BOOT 0x4
```

6.96.1.5 MAX

```
#define MAX(
              a,
              b ) (((a) < (b)) ?  (b) :  (a))
```

6.96.1.6 MAX_BOOT_SEQ

```
#define MAX_BOOT_SEQ 0x7ffffff0
```

6.96.1.7 MODEL_ALL_BIN_FLASH_ADDR

```
#define MODEL_ALL_BIN_FLASH_ADDR 0x00301000
```

6.96.1.8 MODEL_INFO_FLASH_ADDR

```
#define MODEL_INFO_FLASH_ADDR 0x00300000
```

Kneron Confidential

6.96.1.9 POLY

```
#define POLY 0x8408
```

6.96.1.10 VERIFY_BLK_SZ

```
#define VERIFY_BLK_SZ 0x1000
```

6.96.2 Function Documentation

6.96.2.1 kmdw_ota_get_active_ncpu_partition()

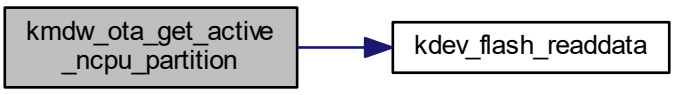
```
int kmdw_ota_get_active_ncpu_partition (
            void  )
```

get active NCPU partition ID

Returns

0 - partition 0 1 - partition 1 -1 - error condition (2 active partitions)

Here is the call graph for this function:



Kneron Confidential

6.96.2.2 kmdw_ota_get_active_scpu_partition()

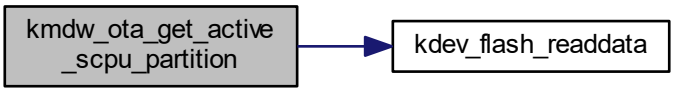
```
int kmdw_ota_get_active_scpu_partition (
            void  )
```

get active SCPU partition ID

Returns

0 - partition 0 1 - partition 1 -1 - error condition (2 active partitions)

Here is the call graph for this function:



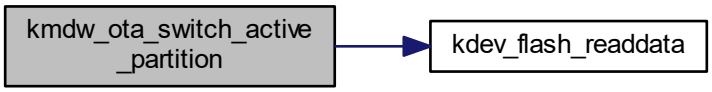
6.96.2.3 kmdw_ota_init()

```
int kmdw_ota_init (
            u8 * tmp_buf,
            FnReadData fn_read )
```

6.96.2.4 kmdw_ota_switch_active_partition()

```
int kmdw_ota_switch_active_partition (
            u32 partition )
```

Here is the call graph for this function:



Kneron Confidential

6.96.2.5 kmdw_ota_update_model()

```
int kmdw_ota_update_model (
            u32 size )
```

6.96.2.6 kmdw_ota_update_ncpu()

```
int kmdw_ota_update_ncpu (
            void  )
```

Update NCPU firmware.

Returns

0 on success

6.96.2.7 kmdw_ota_update_scpu()

```
int kmdw_ota_update_scpu (
            void  )
```

Update SCPU firmware.

Returns

0 on success

6.96.3 Variable Documentation

6.96.3.1 flashing

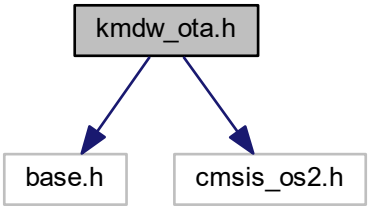
```
int flashing
```

Kneron Confidential

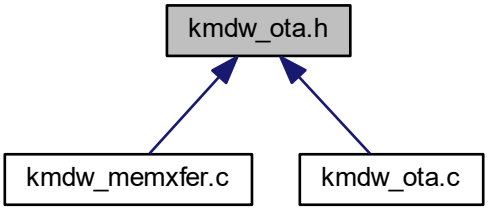
6.97 kmdw_ota.h File Reference

```
#include "base.h"
#include "cmsis_os2.h"
```

Include dependency graph for kmdw_ota.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define SCPU_PARTITION0_START_IN_FLASH 0x2000
- #define NCPU_PARTITION0_START_IN_FLASH 0x18000
- #define PARTITION_0_CFG_START_IN_FLASH 0X28000
- #define SCPU_PARTITION1_START_IN_FLASH 0x41000
- #define NCPU_PARTITION1_START_IN_FLASH 0x57000
- #define PARTITION_1_CFG_START_IN_FLASH 0X67000
- #define SCPU_START_ADDRESS 0x10102000
- #define NCPU_START_ADDRESS 0x28000000
- #define SCPU_IMAGE_SIZE 0x16000
- #define NCPU_IMAGE_SIZE 0x10000
- #define PARTITION_CFG_SIZE 32
- #define SUCCESS 0
- #define MSG_AUTH_FAIL 251
- #define MSG_FLASH_FAIL 252
- #define MSG_DATA_ERROR 253

Kneron Confidential

Typedefs

- typedef uint32_t(∗ FnReadData) (uint32_t addr, uint32_t img_size)

Functions

- int kmdw_ota_init (uint8_t ∗tmp_buf, FnReadData fn_read_data)
- int kmdw_ota_update_scpu (void)

  Update SCPU firmware.
- int kmdw_ota_update_ncpu (void)

  Update NCPU firmware.
- int kmdw_ota_update_model (uint32_t size)

  Update model.
- int kmdw_ota_switch_active_partition (uint32_t partition)

  Switch active partition.
- int kmdw_ota_get_active_scpu_partition (void)

  get active SCPU partition ID
- int kmdw_ota_get_active_ncpu_partition (void)

  get active NCPU partition ID

6.97.1 Macro Definition Documentation

6.97.1.1 MSG_AUTH_FAIL

```
#define MSG_AUTH_FAIL 251
```

6.97.1.2 MSG_DATA_ERROR

```
#define MSG_DATA_ERROR 253
```

6.97.1.3 MSG_FLASH_FAIL

```
#define MSG_FLASH_FAIL 252
```

6.97.1.4 NCPU_IMAGE_SIZE

```
#define NCPU_IMAGE_SIZE 0x10000
```

Kneron Confidential

6.97.1.5 NCPU_PARTITION0_START_IN_FLASH

```
#define NCPU_PARTITION0_START_IN_FLASH 0x18000
```

6.97.1.6 NCPU_PARTITION1_START_IN_FLASH

```
#define NCPU_PARTITION1_START_IN_FLASH 0x57000
```

6.97.1.7 NCPU_START_ADDRESS

```
#define NCPU_START_ADDRESS 0x28000000
```

6.97.1.8 PARTITION_0_CFG_START_IN_FLASH

```
#define PARTITION_0_CFG_START_IN_FLASH 0X28000
```

6.97.1.9 PARTITION_1_CFG_START_IN_FLASH

```
#define PARTITION_1_CFG_START_IN_FLASH 0X67000
```

6.97.1.10 PARTITION_CFG_SIZE

```
#define PARTITION_CFG_SIZE 32
```

6.97.1.11 SCPU_IMAGE_SIZE

```
#define SCPU_IMAGE_SIZE 0x16000
```

6.97.1.12 SCPU_PARTITION0_START_IN_FLASH

```
#define SCPU_PARTITION0_START_IN_FLASH 0x2000
```

Kneron Confidential

6.97.1.13 SCPU_PARTITION1_START_IN_FLASH

```
#define SCPU_PARTITION1_START_IN_FLASH 0x41000
```

6.97.1.14 SCPU_START_ADDRESS

```
#define SCPU_START_ADDRESS 0x10102000
```

6.97.1.15 SUCCESS

```
#define SUCCESS 0
```

6.97.2 Typedef Documentation

6.97.2.1 FnReadData

```
typedef uint32_t(* FnReadData) (uint32_t addr, uint32_t img_size)
```

6.97.3 Function Documentation

6.97.3.1 kmdw_ota_get_active_ncpu_partition()

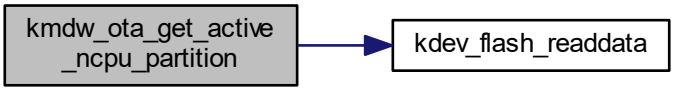
```
int kmdw_ota_get_active_ncpu_partition (
            void  )
```

get active NCPU partition ID

Returns

0 - partition 0 1 - partition 1 -1 - error condition (2 active partitions)

Here is the call graph for this function:



Kneron Confidential

6.97.3.2 kmdw_ota_get_active_scpu_partition()

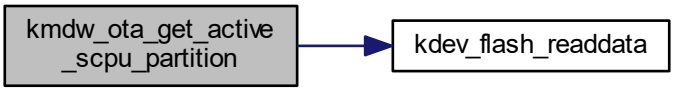
```
int kmdw_ota_get_active_scpu_partition (
            void  )
```

get active SCPU partition ID

Returns

0 - partition 0 1 - partition 1 -1 - error condition (2 active partitions)

Here is the call graph for this function:



6.97.3.3 kmdw_ota_init()

```
int kmdw_ota_init (
            uint8_t * tmp_buf,
            FnReadData fn_read_data )
```

6.97.3.4 kmdw_ota_switch_active_partition()

```
int kmdw_ota_switch_active_partition (
            uint32_t partition )
```

Switch active partition.

Parameters

partition	

Returns

0 on success

Kneron Confidential

6.97.3.5 kmdw_ota_update_model()

```
int kmdw_ota_update_model (
            uint32_t size )
```

Update model.

Parameters

size	model size

Returns

0 on success

6.97.3.6 kmdw_ota_update_ncpu()

```
int kmdw_ota_update_ncpu (
            void  )
```

Update NCPU firmware.

Returns

0 on success

6.97.3.7 kmdw_ota_update_scpu()

```
int kmdw_ota_update_scpu (
            void  )
```

Update SCPU firmware.

Returns

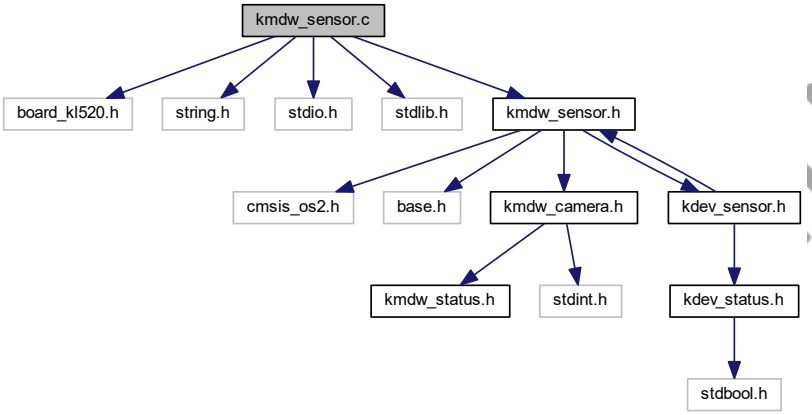
0 on success

Kneron Confidential

6.98 kmdw_sensor.c File Reference

```
#include "board_kl520.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "kmdw_sensor.h"
```

Include dependency graph for kmdw_sensor.c:



Data Structures

- struct kmdw_sensor_s

Functions

- kmdw_status_t kmdw_sensor_s_power (uint32_t cam_idx, uint32_t on)
  set sensor power function
- kmdw_status_t kmdw_sensor_reset (uint32_t cam_idx)
  sensor reset function
- kmdw_status_t kmdw_sensor_s_stream (uint32_t cam_idx, uint32_t enable)
  set sensor stream function
- kmdw_status_t kmdw_sensor_enum_fmt (uint32_t cam_idx, uint32_t index, uint32_t *fourcc)
  set sensor enum function
- kmdw_status_t kmdw_sensor_set_fmt (uint32_t cam_idx, struct cam_format *format)
  set sensor format function
- kmdw_status_t kmdw_sensor_get_fmt (uint32_t cam_idx, struct cam_format *format)
  get sensor format function
- kmdw_status_t kmdw_sensor_set_gain (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
  get sensor gain function
- kmdw_status_t kmdw_sensor_set_aec (uint32_t cam_idx, struct cam_sensor_aec *aec_p)
  sensor set ae controller ROI area function
- kmdw_status_t kmdw_sensor_set_exp_time (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)
  sensor set exposure time function

Kneron Confidential

- kmdw_status_t kmdw_sensor_get_lux (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post_gain, uint8_t *global_gain, uint8_t *y_average)

  sensor get lum and other parameter function
- kmdw_status_t kmdw_sensor_led_switch (uint32_t cam_idx, uint32_t on)

  sensor set nir led on/off function
- kmdw_status_t kmdw_sensor_set_mirror (uint32_t cam_idx, uint32_t enable)

  sensor set image mirror on/off function
- kmdw_status_t kmdw_sensor_set_flip (uint32_t cam_idx, uint32_t enable)

  sensor set image flip on/off function
- uint32_t kmdw_sensor_get_dev_id (uint32_t cam_idx)

  sensor get device ID function
- struct sensor_ops * kmdw_sensor_get_ops (uint32_t sensor_idx)
- kmdw_status_t kmdw_sensor_register (uint32_t cam_idx, uint32_t sensor_idx)

  register sensor
- kmdw_status_t kmdw_sensor_unregister (uint32_t cam_idx, struct sensor_ops *sensor_ops_p)

Variables

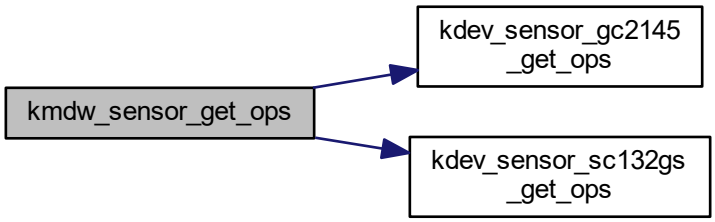
- struct kmdw_sensor_s sensor_s [CAM_ID_MAX]

6.98.1 Function Documentation

6.98.1.1 kmdw_sensor_get_ops()

```
struct sensor_ops* kmdw_sensor_get_ops (
            uint32_t sensor_idx )
```

Here is the call graph for this function:



Kneron Confidential

Here is the caller graph for this function:

6.98.1.2 kmdw_sensor_unregister()

```
kmdw_status_t kmdw_sensor_unregister (
            uint32_t cam_idx,
            struct sensor_ops * sensor_ops_p )
```

6.98.2 Variable Documentation

6.98.2.1 sensor_s

```
struct kmdw_sensor_s sensor_s[CAM_ID_MAX]
```

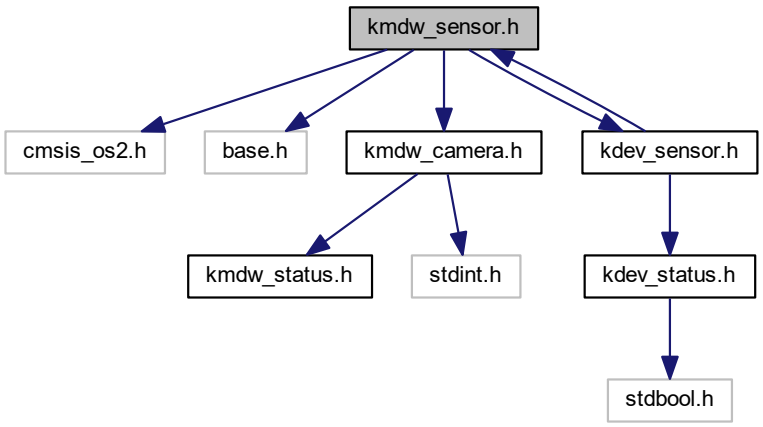
6.99 kmdw_sensor.h File Reference

```
#include <cmsis_os2.h>
#include "base.h"
#include "kmdw_camera.h"
```

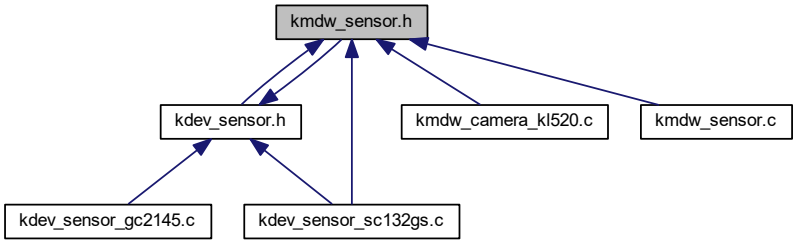
Kneron Confidential

```
#include "kdev_sensor.h"
```

Include dependency graph for kmdw_sensor.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sensor_device
- struct sensor_init_seq
- struct sensor_datafmt_info
- struct sensor_win_size

Macros

- #define fourcc(a, b, c, d) ((uint32_t)(a) | ((uint32_t)(b) << 8) | ((uint32_t)(c) << 16) | ((uint32_t)(d) << 24))
- #define PIX_FMT_YCBCR fourcc('Y', 'B', 'Y', 'R')
- #define PIX_FMT_RGB565 fourcc('R', 'G', 'B', 'P')
- #define PIX_FMT_RAW10 fourcc('R', 'A', '1', '0')
- #define PIX_FMT_RAW8 fourcc('R', 'A', 'W', '8')

Kneron Confidential

Enumerations

- enum colorspace { COLORSPACE_RGB = 0, COLORSPACE_YUV = 1, COLORSPACE_RAW = 2 }

Functions

- struct sensor_init_seq __attribute__ ((packed))
- kmdw_status_t kmdw_sensor_s_power (uint32_t cam_idx, uint32_t on)

  set sensor power function
- kmdw_status_t kmdw_sensor_reset (uint32_t cam_idx)

  sensor reset function
- kmdw_status_t kmdw_sensor_s_stream (uint32_t cam_idx, uint32_t enable)

  set sensor stream function
- kmdw_status_t kmdw_sensor_enum_fmt (uint32_t cam_idx, uint32_t index, uint32_t *fourcc)

  set sensor enum function
- kmdw_status_t kmdw_sensor_set_fmt (uint32_t cam_idx, struct cam_format *format)

  set sensor format function
- kmdw_status_t kmdw_sensor_get_fmt (uint32_t cam_idx, struct cam_format *format)

  get sensor format function
- kmdw_status_t kmdw_sensor_set_gain (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  get sensor gain function
- kmdw_status_t kmdw_sensor_set_aec (uint32_t cam_idx, struct cam_sensor_aec *aec_p)

  sensor set ae controller ROI area function
- kmdw_status_t kmdw_sensor_set_exp_time (uint32_t cam_idx, uint32_t gain1, uint32_t gain2)

  sensor set exposure time function
- kmdw_status_t kmdw_sensor_get_lux (uint32_t cam_idx, uint16_t *expo, uint8_t *pre_gain, uint8_t *post↩ _gain, uint8_t *global_gain, uint8_t *y_average)

  sensor get lum and other parameter function
- kmdw_status_t kmdw_sensor_led_switch (uint32_t cam_idx, uint32_t on)

  sensor set nir led on/off function
- kmdw_status_t kmdw_sensor_set_mirror (uint32_t cam_idx, uint32_t enable)

  sensor set image mirror on/off function
- kmdw_status_t kmdw_sensor_set_flip (uint32_t cam_idx, uint32_t enable)

  sensor set image flip on/off function
- uint32_t kmdw_sensor_get_dev_id (uint32_t cam_idx)

  sensor get device ID function
- kmdw_status_t kmdw_sensor_register (uint32_t cam_idx, uint32_t sensor_idx)

  register sensor

Variables

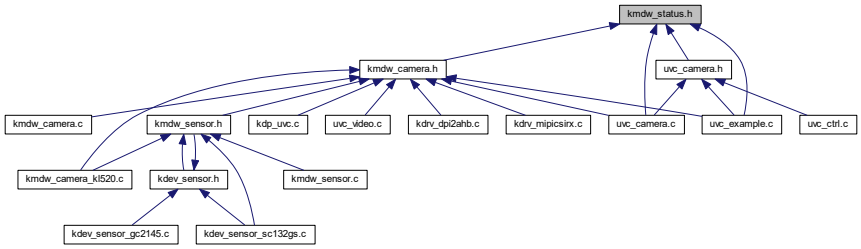
- uint16_t addr
- uint8_t value
- struct sensor_datafmt_info __attribute__

  Endpoint descriptor struct.

Kneron Confidential

6.100 kmdw_status.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum kmdw_status_t { KMDW_STATUS_OK = 0, KMDW_STATUS_ERROR }

6.100.1 Enumeration Type Documentation

6.100.1.1 kmdw_status_t

```
enum kmdw_status_t
```

Enumerator

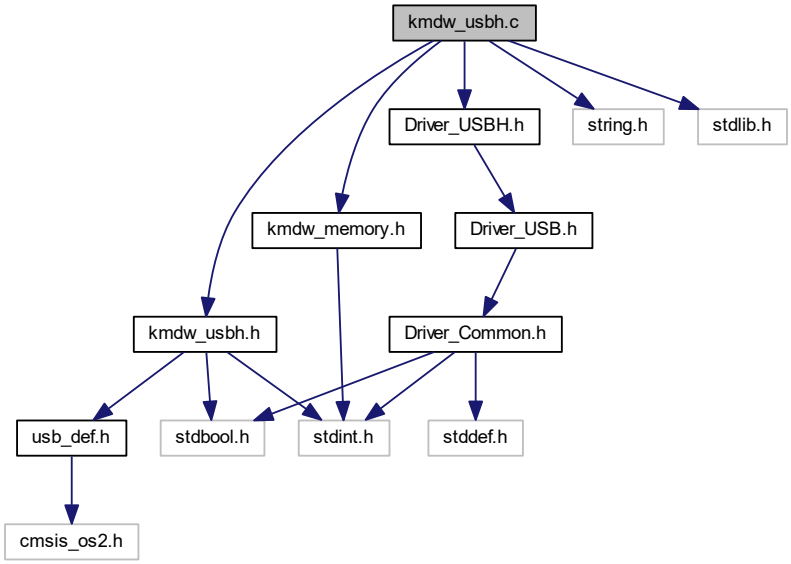
KMDW_STATUS_OK	mdw status OK
KMDW_STATUS_ERROR	mdw status error

Kneron Confidential

6.101 kmdw_usbh.c File Reference

```
#include "kmdw_usbh.h"
#include "Driver_USBH.h"
#include "kmdw_memory.h"
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for kmdw_usbh.c:



Data Structures

- struct USBH_Event_t
- struct USBH_PIPE_TID_t

Macros

- #define __USBH_MDW_H__
- #define USBH_EVENT_QUEUE_LEN 16
- #define MAX_PIPE_NUM 6
- #define USER_FLAG_XFER_COMPLETE 0x100U
- #define MDW_FLAG_MESSAGE 0x1000U
- #define MDW_FLAG_ITD_WORK 0x2000U
- #define DEV_ADDR 0x3

Enumerations

- enum USBH_STM_t {
  USBH_STM_INITED = 0, USBH_STM_CONNECTED_FS, USBH_STM_RESET_HS_DONE, USBH_STM_RESET_HS_DONE
  USBH_STM_CUSTOM_CONFIGURE, USBH_STM_CUSTOM_INITIALIZE, USBH_STM_CUSTOM_INITIALIZE_DONE,
  USBH_STM_FAILED = 0x100 }

Kneron Confidential

Functions

- __weak uint8_t USBH_CustomClass_Configure (uint8_t device, const USB_DEVICE_DESCRIPTOR *ptr↩ _dev_desc, const USB_CONFIGURATION_DESCRIPTOR *ptr_cfg_desc)

  Callback function called when custom class device is connected and needs to configure resources used by custom class device instance.
- __weak usbStatus USBH_CustomClass_Initialize (uint8_t instance)

  Callback function called when custom class device is connected and needs to initialize custom class device instance.
- usbStatus USBH_Initialize (uint8_t ctrl)

  Initialize USB Host stack and controller.
- USBH_PIPE_HANDLE USBH_PipeCreate (uint8_t device, uint8_t ep_addr, uint8_t ep_type, uint16_t ep_↩ max_packet_size, uint8_t ep_interval)

  Create Pipe.
- usbStatus USBH_PipeSend (USBH_PIPE_HANDLE pipe_hndl, const uint8_t *buf, uint32_t len)

  Send data on Pipe.
- usbStatus USBH_PipeReceive (USBH_PIPE_HANDLE pipe_hndl, uint8_t *buf, uint32_t len)

  Receive data on Pipe.
- uint32_t USBH_PipeSendGetResult (USBH_PIPE_HANDLE pipe_hndl)

  Get result of send data operation on Pipe.
- uint32_t USBH_PipeReceiveGetResult (USBH_PIPE_HANDLE pipe_hndl)

  Get result of receive data operation on Pipe.
- usbStatus USBH_ControlTransfer (uint8_t device, const USB_SETUP_PACKET *setup_packet, uint8_↩ t *data, uint32_t len)

  Do a Control Transfer on Default Pipe.
- USBH_PIPE_HANDLE USBH_Pipe_ISOCH_PipeCreate (uint8_t device, uint8_t ep_addr, uint32_↩ t wMaxPacketSize, uint8_t bInterval, uint8_t *buf, uint32_t buf_size)
- usbStatus USBH_Pipe_ISOCH_Start (USBH_PIPE_HANDLE pipe_hndl, USBH_CB_ISR_Isoch_transfer user_isoch_cb)
- usbStatus USBH_Pipe_ISOCH_Stop (USBH_PIPE_HANDLE pipe_hndl)
- usbStatus USBH_DeviceRequest_SetInterface (uint8_t device, uint8_t index, uint8_t alternate)

  Standard Device Request on Default Pipe - SET_INTERFACE.

Variables

- ARM_USBH_ISOCH_ITD_WORK_FUNC itd_work_func = 0

6.101.1 Macro Definition Documentation

6.101.1.1 __USBH_MDW_H__

```
#define __USBH_MDW_H__
```

6.101.1.2 DEV_ADDR

```
#define DEV_ADDR 0x3
```

Kneron Confidential

6.101.1.3 MAX_PIPE_NUM

```
#define MAX_PIPE_NUM 6
```

6.101.1.4 MDW_FLAG_ITD_WORK

```
#define MDW_FLAG_ITD_WORK 0x2000U
```

6.101.1.5 MDW_FLAG_MESSAGE

```
#define MDW_FLAG_MESSAGE 0x1000U
```

6.101.1.6 USBH_EVENT_QUEUE_LEN

```
#define USBH_EVENT_QUEUE_LEN 16
```

6.101.1.7 USER_FLAG_XFER_COMPLETE

```
#define USER_FLAG_XFER_COMPLETE 0x100U
```

6.101.2 Enumeration Type Documentation

6.101.2.1 USBH_STM_t

```
enum USBH_STM_t
```

Enumerator

USBH_STM_INITED	
USBH_STM_CONNECTED_FS	
USBH_STM_RESET_HS_DONE	
USBH_STM_RESET_HS_DONE_2	
USBH_STM_CUSTOM_CONFIGURE	
USBH_STM_CUSTOM_INITIALIZE	
USBH_STM_CUSTOM_INITIALIZE_DONE	
USBH_STM_FAILED	

Kneron Confidential

6.101.3 Function Documentation

6.101.3.1 USBH_ControlTransfer()

```
usbStatus USBH_ControlTransfer (
            uint8_t device,
            const USB_SETUP_PACKET * setup_packet,
            uint8_t * data,
            uint32_t len )
```

Do a Control Transfer on Default Pipe.

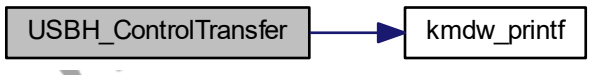
Parameters

in	*device*	index of USB Device.
in	*setup_packet*	pointer to setup packet.
in,out	*data*	buffer containing data bytes to send or where data should be received in data stage of Control Transfer.
in	*len*	number of bytes to send or receive in data stage of Control Transfer.

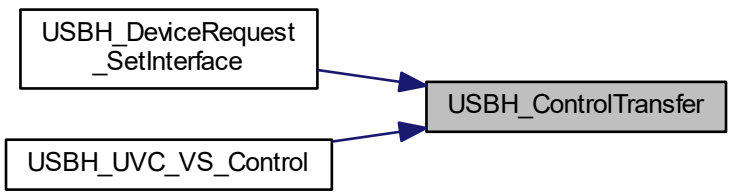
Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



Here is the caller graph for this function:



Kneron Confidential

6.101.3.2 USBH_CustomClass_Configure()

```
__weak uint8_t USBH_CustomClass_Configure (
            uint8_t device,
            const USB_DEVICE_DESCRIPTOR * ptr_dev_desc,
            const USB_CONFIGURATION_DESCRIPTOR * ptr_cfg_desc )
```

Callback function called when custom class device is connected and needs to configure resources used by custom class device instance.

Parameters

in	*device*	index of USB Device.
in	*ptr_dev_desc*	pointer to device descriptor.
in	*ptr_cfg_desc*	pointer to configuration descriptor.

Returns

index of configured custom class device instance or configuration failed :

- value <= 127 : index of configured custom class device instance
- value 255 : configuration failed

6.101.3.3 USBH_CustomClass_Initialize()

```
__weak usbStatus USBH_CustomClass_Initialize (
            uint8_t instance )
```

Callback function called when custom class device is connected and needs to initialize custom class device instance.

Parameters

in	*instance*	index of custom class device instance.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.101.3.4 USBH_DeviceRequest_SetInterface()

```
usbStatus USBH_DeviceRequest_SetInterface (
            uint8_t device,
```

Kneron Confidential

```
uint8_t index,
uint8_t alternate )
```

Standard Device Request on Default Pipe - SET_INTERFACE.

Kneron Confidential

Parameters

in	*device*	index of USB Device.
in	*index*	interface index.
in	*alternate*	alternate setting.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:

6.101.3.5 USBH_Initialize()

```
usbStatus USBH_Initialize (
              uint8_t ctrl )
```

Initialize USB Host stack and controller.

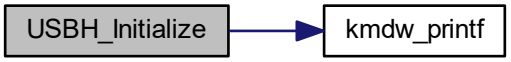
Parameters

in	*ctrl*	index of USB Host controller.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:

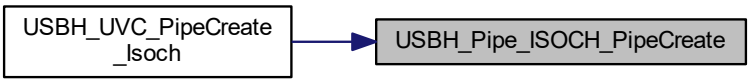


Kneron Confidential

6.101.3.6 USBH_Pipe_ISOCH_PipeCreate()

```
USBH_PIPE_HANDLE USBH_Pipe_ISOCH_PipeCreate (
            uint8_t device,
            uint8_t ep_addr,
            uint32_t wMaxPacketSize,
            uint8_t bInterval,
            uint8_t * buf,
            uint32_t buf_size )
```

Here is the caller graph for this function:



6.101.3.7 USBH_Pipe_ISOCH_Start()

```
usbStatus USBH_Pipe_ISOCH_Start (
            USBH_PIPE_HANDLE pipe_hndl,
            USBH_CB_ISR_Isoch_transfer user_isoch_cb )
```

6.101.3.8 USBH_Pipe_ISOCH_Stop()

```
usbStatus USBH_Pipe_ISOCH_Stop (
            USBH_PIPE_HANDLE pipe_hndl )
```

6.101.3.9 USBH_PipeCreate()

```
USBH_PIPE_HANDLE USBH_PipeCreate (
            uint8_t device,
            uint8_t ep_addr,
            uint8_t ep_type,
            uint16_t ep_max_packet_size,
            uint8_t ep_interval )
```

Create Pipe.

Kneron Confidential

Parameters

in	*device*	index of USB Device.
in	*ep_addr*	endpoint address : • ep_addr.0..3 : address • ep_addr.7 : direction
in	*ep_type*	endpoint type.
in	*ep_max_packet_size*	endpoint maximum packet size.
in	*ep_interval*	endpoint polling interval.

Returns

pipe handle or pipe creation failed :

- value $>$ 0 : pipe handle
- value 0 : pipe creation failed

6.101.3.10 USBH_PipeReceive()

```
usbStatus USBH_PipeReceive (
            USBH_PIPE_HANDLE pipe_hndl,
            uint8_t * buf,
            uint32_t len )
```

Receive data on Pipe.

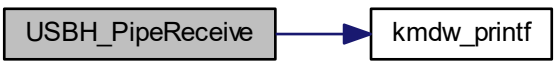
Parameters

in	*pipe_hndl*	pipe handle.
out	*buf*	buffer that receives data.
in	*len*	maximum number of bytes to receive.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



Kneron Confidential

6.101.3.11 USBH_PipeReceiveGetResult()

```
uint32_t USBH_PipeReceiveGetResult (
            USBH_PIPE_HANDLE pipe_hndl )
```

Get result of receive data operation on Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

number of successfully received data bytes.

6.101.3.12 USBH_PipeSend()

```
usbStatus USBH_PipeSend (
            USBH_PIPE_HANDLE pipe_hndl,
            const uint8_t * buf,
            uint32_t len )
```

Send data on Pipe.

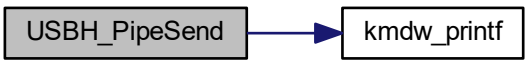
Parameters

in	*pipe_hndl*	pipe handle.
in	*buf*	buffer containing data bytes to send.
in	*len*	number of bytes to send.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



Kneron Confidential

6.101.3.13 USBH_PipeSendGetResult()

```
uint32_t USBH_PipeSendGetResult (
            USBH_PIPE_HANDLE pipe_hndl )
```

Get result of send data operation on Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

number of successfully sent data bytes.

6.101.4 Variable Documentation

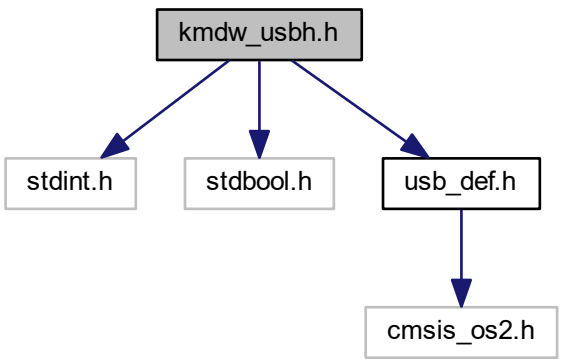
6.101.4.1 itd_work_func

```
ARM_USBH_ISOCH_ITD_WORK_FUNC itd_work_func = 0
```

6.102 kmdw_usbh.h File Reference

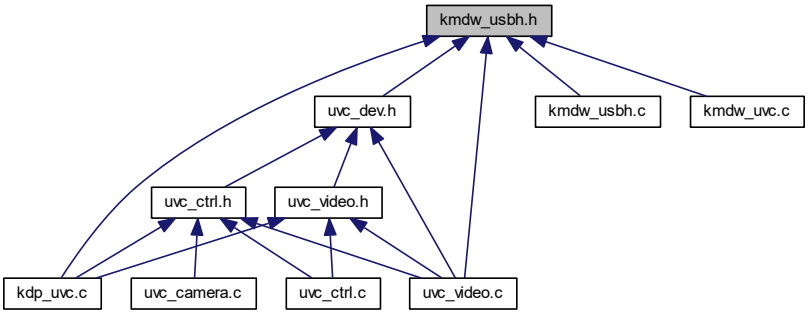
```
#include <stdint.h>
#include <stdbool.h>
#include "usb_def.h"
```

Include dependency graph for kmdw_usbh.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Typedefs

- typedef uint32_t USBH_PIPE_HANDLE
- typedef void(∗ USBH_CB_ISR_Isoch_transfer) (uint32_t ∗payload, uint32_t length)

Enumerations

- enum usbStatus {
  usbOK = 0U, usbTimeout, usbInvalidParameter, usbThreadError = 0x10U,
  usbTimerError, usbSemaphoreError, usbMutexError, usbControllerError = 0x20U,
  usbDeviceError, usbDriverError, usbDriverBusy, usbMemoryError,
  usbNotConfigured, usbClassErrorADC, usbClassErrorCDC, usbClassErrorHID,
  usbClassErrorMSC, usbClassErrorCustom, usbUnsupportedClass, usbTransferStall = 0x40U,
  usbTransferError, usbUnknownError = 0xFFU }

  ==== USB Constants and Defines ====
- enum USBH_NOTIFY {
  USBH_NOTIFY_CONNECT = 0U, USBH_NOTIFY_DISCONNECT, USBH_NOTIFY_OVERCURRENT,
  USBH_NOTIFY_REMOTE_WAKEUP,
  USBH_NOTIFY_READY, USBH_NOTIFY_UNKNOWN_DEVICE, USBH_NOTIFY_INSUFFICIENT_POWER,
  USBH_NOTIFY_CONFIGURATION_FAILED,
  USBH_NOTIFY_INITIALIZATION_FAILED }

  USB Host Notification enumerated constants.

Functions

- uint32_t USBH_GetVersion (void)

  Get version of USB Host stack.
- usbStatus USBH_Initialize (uint8_t ctrl)

  Initialize USB Host stack and controller.
- usbStatus USBH_Uninitialize (uint8_t ctrl)

  De-initialize USB Host stack and controller.
- usbStatus USBH_Port_Suspend (uint8_t ctrl, uint8_t port)

  Suspend a root HUB port on specified controller.
- usbStatus USBH_Port_Resume (uint8_t ctrl, uint8_t port)

Kneron Confidential

Resume a root HUB port on specified controller.

- uint8_t USBH_Device_GetController (uint8_t device)

  Get index of USB Host controller to which USB Device is connected.
- uint8_t USBH_Device_GetPort (uint8_t device)

  Get index of USB Host Root HUB port to which USB Device is connected.
- usbStatus USBH_Device_GetStatus (uint8_t device)

  Get status of USB Device.
- int32_t USBH_Device_GetSpeed (uint8_t device)

  Get communication speed of USB Device.
- uint8_t USBH_Device_GetAddress (uint8_t device)

  Get communication address of USB Device.
- uint16_t USBH_Device_GetVID (uint8_t device)

  Get Vendor ID (VID) of USB Device.
- uint16_t USBH_Device_GetPID (uint8_t device)

  Get Product ID (PID) of USB Device.
- usbStatus USBH_Device_GetStringDescriptor (uint8_t device, uint8_t index, uint16_t language_id, uint8_t *descriptor_data, uint16_t descriptor_length)

  Get String Descriptor of USB Device.
- void USBH_Notify (uint8_t ctrl, uint8_t port, uint8_t device, USBH_NOTIFY notify)

  Callback function called when some event has happened on corresponding controller and port.
- uint8_t USBH_CustomClass_GetDevice (uint8_t instance)

  Get Device instance of Custom Class Device.
- usbStatus USBH_CustomClass_GetStatus (uint8_t instance)

  Get status of Custom Class Device.
- uint8_t USBH_CustomClass_Configure (uint8_t device, const USB_DEVICE_DESCRIPTOR *ptr_dev_desc, const USB_CONFIGURATION_DESCRIPTOR *ptr_cfg_desc)

  Callback function called when custom class device is connected and needs to configure resources used by custom class device instance.
- usbStatus USBH_CustomClass_Unconfigure (uint8_t instance)

  Callback function called when custom class device is disconnected and needs to de-configure resources used by custom class device instance.
- usbStatus USBH_CustomClass_Initialize (uint8_t instance)

  Callback function called when custom class device is connected and needs to initialize custom class device instance.
- usbStatus USBH_CustomClass_Uninitialize (uint8_t instance)

  Callback function called when custom class device is disconnected and needs to de-initialize custom class device instance.
- USBH_PIPE_HANDLE USBH_PipeCreate (uint8_t device, uint8_t ep_addr, uint8_t ep_type, uint16_t ep_↩ max_packet_size, uint8_t ep_interval)

  Create Pipe.
- usbStatus USBH_PipeUpdate (USBH_PIPE_HANDLE pipe_hndl)

  Update Pipe.
- usbStatus USBH_PipeDelete (USBH_PIPE_HANDLE pipe_hndl)

  Delete Pipe.
- usbStatus USBH_PipeReset (USBH_PIPE_HANDLE pipe_hndl)

  Reset Pipe (reset data toggle)
- usbStatus USBH_PipeReceive (USBH_PIPE_HANDLE pipe_hndl, uint8_t *buf, uint32_t len)

  Receive data on Pipe.
- uint32_t USBH_PipeReceiveGetResult (USBH_PIPE_HANDLE pipe_hndl)

  Get result of receive data operation on Pipe.
- usbStatus USBH_PipeSend (USBH_PIPE_HANDLE pipe_hndl, const uint8_t *buf, uint32_t len)

  Send data on Pipe.
- uint32_t USBH_PipeSendGetResult (USBH_PIPE_HANDLE pipe_hndl)

Kneron Confidential

Get result of send data operation on Pipe.

- usbStatus USBH_PipeAbort (USBH_PIPE_HANDLE pipe_hndl)

  Abort send/receive operation on Pipe.

- usbStatus USBH_ControlTransfer (uint8_t device, const USB_SETUP_PACKET *setup_packet, uint8_t *data, uint32_t len)

  Do a Control Transfer on Default Pipe.

- usbStatus USBH_DeviceRequest_GetStatus (uint8_t device, uint8_t recipient, uint8_t index, uint8_t *ptr_stat_dat)

  Standard Device Request on Default Pipe - GET_STATUS.

- usbStatus USBH_DeviceRequest_ClearFeature (uint8_t device, uint8_t recipient, uint8_t index, uint8_t feature_selector)

  Standard Device Request on Default Pipe - CLEAR_FEATURE.

- usbStatus USBH_DeviceRequest_SetFeature (uint8_t device, uint8_t recipient, uint8_t index, uint8_t feature_selector)

  Standard Device Request on Default Pipe - SET_FEATURE.

- usbStatus USBH_DeviceRequest_SetAddress (uint8_t device, uint8_t device_address)

  Standard Device Request on Default Pipe - SET_ADDRESS.

- usbStatus USBH_DeviceRequest_GetDescriptor (uint8_t device, uint8_t recipient, uint8_t descriptor_type, uint8_t descriptor_index, uint16_t language_id, uint8_t *descriptor_data, uint16_t descriptor_length)

  Standard Device Request on Default Pipe - GET_DESCRIPTOR.

- usbStatus USBH_DeviceRequest_SetDescriptor (uint8_t device, uint8_t recipient, uint8_t descriptor_type, uint8_t descriptor_index, uint16_t language_id, const uint8_t *descriptor_data, uint16_t descriptor_length)

  Standard Device Request on Default Pipe - SET_DESCRIPTOR.

- usbStatus USBH_DeviceRequest_GetConfiguration (uint8_t device, uint8_t *ptr_configuration)

  Standard Device Request on Default Pipe - GET_CONFIGURATION.

- usbStatus USBH_DeviceRequest_SetConfiguration (uint8_t device, uint8_t configuration)

  Standard Device Request on Default Pipe - SET_CONFIGURATION.

- usbStatus USBH_DeviceRequest_GetInterface (uint8_t device, uint8_t index, uint8_t *ptr_alternate)

  Standard Device Request on Default Pipe - GET_INTERFACE.

- usbStatus USBH_DeviceRequest_SetInterface (uint8_t device, uint8_t index, uint8_t alternate)

  Standard Device Request on Default Pipe - SET_INTERFACE.

- usbStatus USBH_DeviceRequest_SynchFrame (uint8_t device, uint8_t index, uint8_t *ptr_frame_number)

  Standard Device Request on Default Pipe - SYNCH_FRAME.

- USBH_PIPE_HANDLE USBH_Pipe_ISOCH_PipeCreate (uint8_t device, uint8_t ep_addr, uint32_t wMaxPacketSize, uint8_t bInterval, uint8_t *buf, uint32_t buf_size)
- usbStatus USBH_Pipe_ISOCH_Start (USBH_PIPE_HANDLE pipe_hndl, USBH_CB_ISR_Isoch_transfer isoch_cb)
- usbStatus USBH_Pipe_ISOCH_Stop (USBH_PIPE_HANDLE pipe_hndl)

6.102.1 Typedef Documentation

6.102.1.1 USBH_CB_ISR_Isoch_transfer

```
typedef void(* USBH_CB_ISR_Isoch_transfer) (uint32_t *payload, uint32_t length)
```

Kneron Confidential

6.102.1.2 USBH_PIPE_HANDLE

```
typedef uint32_t USBH_PIPE_HANDLE
```

6.102.2 Enumeration Type Documentation

6.102.2.1 USBH_NOTIFY

```
enum USBH_NOTIFY
```

USB Host Notification enumerated constants.

Enumerator

USBH_NOTIFY_CONNECT	Port connection happened.
USBH_NOTIFY_DISCONNECT	Port disconnection happened.
USBH_NOTIFY_OVERCURRENT	Port overcurrent happened.
USBH_NOTIFY_REMOTE_WAKEUP	Port remote wakeup signaling happened.
USBH_NOTIFY_READY	Device was successfully enumerated, initialized and is ready for communication.
USBH_NOTIFY_UNKNOWN_DEVICE	Device was successfully enumerated but there is no driver for it.
USBH_NOTIFY_INSUFFICIENT_POWER	Device requires more power consumption than available.
USBH_NOTIFY_CONFIGURATION_FAILED	Device was not successfully configured (not enough resources)
USBH_NOTIFY_INITIALIZATION_FAILED	Device was not successfully initialized.

6.102.2.2 usbStatus

```
enum usbStatus
```

==== USB Constants and Defines ====

Status code values returned by USB library functions.

Enumerator

usbOK	Function completed with no error.
usbTimeout	Function completed; time-out occurred.
usbInvalidParameter	Invalid Parameter error: a mandatory parameter was missing or specified an incorrect object.
usbThreadError	CMSIS-RTOS Thread creation/termination failed.
usbTimerError	CMSIS-RTOS Timer creation/deletion failed.
usbSemaphoreError	CMSIS-RTOS Semaphore creation failed.

Kneron Confidential

Enumerator

usbMutexError	CMSIS-RTOS Mutex creation failed.
usbControllerError	Controller does not exist.
usbDeviceError	Device does not exist.
usbDriverError	Driver function produced error.
usbDriverBusy	Driver function is busy.
usbMemoryError	Memory management function produced error.
usbNotConfigured	Device is not configured (is connected)
usbClassErrorADC	Audio Device Class (ADC) error (no device or device produced error)
usbClassErrorCDC	Communication Device Class (CDC) error (no device or device produced error)
usbClassErrorHID	Human Interface Device (HID) error (no device or device produced error)
usbClassErrorMSC	Mass Storage Device (MSC) error (no device or device produced error)
usbClassErrorCustom	Custom device Class (Class) error (no device or device produced error)
usbUnsupportedClass	Unsupported Class.
usbTransferStall	Transfer handshake was stall.
usbTransferError	Transfer error.
usbUnknownError	Unspecified USB error.

6.102.3 Function Documentation

6.102.3.1 USBH_ControlTransfer()

```
usbStatus USBH_ControlTransfer (
             uint8_t device,
             const USB_SETUP_PACKET * setup_packet,
             uint8_t * data,
             uint32_t len )
```

Do a Control Transfer on Default Pipe.

Parameters

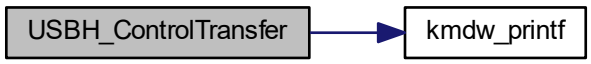
in	*device*	index of USB Device.
in	*setup_packet*	pointer to setup packet.
in,out	*data*	buffer containing data bytes to send or where data should be received in data stage of Control Transfer.
in	*len*	number of bytes to send or receive in data stage of Control Transfer.

Kneron Confidential

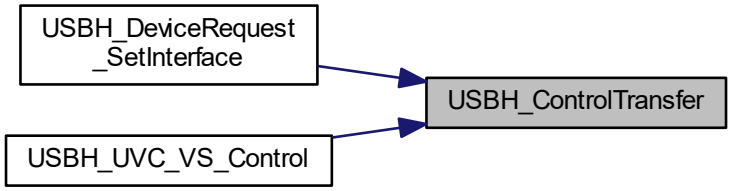
Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



Here is the caller graph for this function:



6.102.3.2 USBH_CustomClass_Configure()

```
uint8_t USBH_CustomClass_Configure (
            uint8_t device,
            const USB_DEVICE_DESCRIPTOR * ptr_dev_desc,
            const USB_CONFIGURATION_DESCRIPTOR * ptr_cfg_desc )
```

Callback function called when custom class device is connected and needs to configure resources used by custom class device instance.

Parameters

in	*device*	index of USB Device.
in	*ptr_dev_desc*	pointer to device descriptor.
in	*ptr_cfg_desc*	pointer to configuration descriptor.

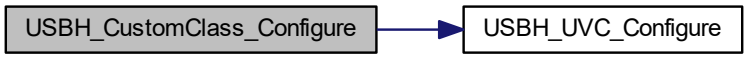
Kneron Confidential

Returns

index of configured custom class device instance or configuration failed :

- value <= 127 : index of configured custom class device instance
- value 255 : configuration failed

Here is the call graph for this function:



6.102.3.3 USBH_CustomClass_GetDevice()

```
uint8_t USBH_CustomClass_GetDevice (
            uint8_t instance )
```

Get Device instance of Custom Class Device.

Parameters

in	*instance*	instance of Custom Class Device.

Returns

instance of Device or non-existing Device instance :

- value <= 127 : instance of Device
- value 255 : non-existing Device instance

6.102.3.4 USBH_CustomClass_GetStatus()

```
usbStatus USBH_CustomClass_GetStatus (
            uint8_t instance )
```

Get status of Custom Class Device.

Parameters

in	*instance*	instance of Custom Class Device.

Kneron Confidential

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.5 USBH_CustomClass_Initialize()

```
usbStatus USBH_CustomClass_Initialize (
            uint8_t instance )
```

Callback function called when custom class device is connected and needs to initialize custom class device instance.

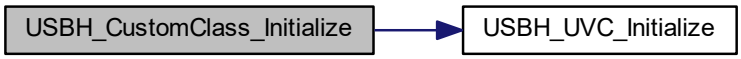
Parameters

in	*instance*	index of custom class device instance.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



6.102.3.6 USBH_CustomClass_Unconfigure()

```
usbStatus USBH_CustomClass_Unconfigure (
            uint8_t instance )
```

Callback function called when custom class device is disconnected and needs to de-configure resources used by custom class device instance.

Parameters

in	*instance*	index of custom class device instance.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Kneron Confidential

6.102.3.7 USBH_CustomClass_Uninitialize()

```
usbStatus USBH_CustomClass_Uninitialize (
            uint8_t instance )
```

Callback function called when custom class device is disconnected and needs to de-initialize custom class device instance.

Parameters

in	*instance*	index of custom class device instance.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.8 USBH_Device_GetAddress()

```
uint8_t USBH_Device_GetAddress (
            uint8_t device )
```

Get communication address of USB Device.

Parameters

in	*device*	index of USB Device.

Returns

enumerated address or invalid address :

- value <= 127 : enumerated address
- value 255 : invalid address

6.102.3.9 USBH_Device_GetController()

```
uint8_t USBH_Device_GetController (
            uint8_t device )
```

Get index of USB Host controller to which USB Device is connected.

Kneron Confidential

Parameters

in	*device*	index of USB Device.

Returns

index of USB Host controller or non-existing USB Host controller :

- value != 255 : index of USB Host controller
- value 255 : non-existing USB Host controller

6.102.3.10 USBH_Device_GetPID()

```
uint16_t USBH_Device_GetPID (
            uint8_t device )
```

Get Product ID (PID) of USB Device.

Parameters

in	*device*	index of USB Device.

Returns

Product ID.

6.102.3.11 USBH_Device_GetPort()

```
uint8_t USBH_Device_GetPort (
            uint8_t device )
```

Get index of USB Host Root HUB port to which USB Device is connected.

Parameters

in	*device*	index of USB Device.

Returns

index of USB Host Root HUB port or non-existing USB Host Root HUB port :

- value <= 15 : index of USB Host Root HUB port
- value 255 : non-existing USB Host Root HUB port

Kneron Confidential

6.102.3.12 USBH_Device_GetSpeed()

```
int32_t USBH_Device_GetSpeed (
            uint8_t device )
```

Get communication speed of USB Device.

Parameters

in	*device*	index of USB Device.

Returns

communication speed :

- USB_SPEED_LOW = low speed
- USB_SPEED_FULL = full speed
- USB_SPEED_HIGH = high speed

6.102.3.13 USBH_Device_GetStatus()

```
usbStatus USBH_Device_GetStatus (
            uint8_t device )
```

Get status of USB Device.

Parameters

in	*device*	index of USB Device.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.14 USBH_Device_GetStringDescriptor()

```
usbStatus USBH_Device_GetStringDescriptor (
            uint8_t device,
            uint8_t index,
            uint16_t language_id,
            uint8_t * descriptor_data,
            uint16_t descriptor_length )
```

Get String Descriptor of USB Device.

Kneron Confidential

Parameters

in	*device*	index of USB Device.
in	*index*	index of string descriptor.
in	*language_id*	language ID.
out	*descriptor_data*	pointer to where descriptor data will be read.
in	*descriptor_length*	maximum descriptor length.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.15 USBH_Device_GetVID()

```
uint16_t USBH_Device_GetVID (
            uint8_t device )
```

Get Vendor ID (VID) of USB Device.

Parameters

in	*device*	index of USB Device.

Returns

Vendor ID.

6.102.3.16 USBH_DeviceRequest_ClearFeature()

```
usbStatus USBH_DeviceRequest_ClearFeature (
            uint8_t device,
            uint8_t recipient,
            uint8_t index,
            uint8_t feature_selector )
```

Standard Device Request on Default Pipe - CLEAR_FEATURE.

Parameters

in	*device*	index of USB Device.
in	*recipient*	recipient.
in	*index*	interface or endpoint index.
in	*feature_selector*	feature selector.

Kneron Confidential

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.17 USBH_DeviceRequest_GetConfiguration()

```
usbStatus USBH_DeviceRequest_GetConfiguration (
            uint8_t device,
            uint8_t * ptr_configuration )
```

Standard Device Request on Default Pipe - GET_CONFIGURATION.

Parameters

in	*device*	index of USB Device.
out	*ptr_configuration*	pointer to where configuration will be read.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.18 USBH_DeviceRequest_GetDescriptor()

```
usbStatus USBH_DeviceRequest_GetDescriptor (
            uint8_t device,
            uint8_t recipient,
            uint8_t descriptor_type,
            uint8_t descriptor_index,
            uint16_t language_id,
            uint8_t * descriptor_data,
            uint16_t descriptor_length )
```

Standard Device Request on Default Pipe - GET_DESCRIPTOR.

Parameters

in	*device*	index of USB Device.
in	*recipient*	recipient.
in	*descriptor_type*	descriptor type.
in	*descriptor_index*	descriptor index.
in	*language_id*	language ID.
out	*descriptor_data*	pointer to where descriptor data will be read.
in	*descriptor_length*	maximum descriptor length.

Kneron Confidential

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.19 USBH_DeviceRequest_GetInterface()

```
usbStatus USBH_DeviceRequest_GetInterface (
            uint8_t device,
            uint8_t index,
            uint8_t * ptr_alternate )
```

Standard Device Request on Default Pipe - GET_INTERFACE.

Parameters

in	*device*	index of USB Device.
in	*index*	interface index.
out	*ptr_alternate*	pointer to where alternate setting data will be read.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.20 USBH_DeviceRequest_GetStatus()

```
usbStatus USBH_DeviceRequest_GetStatus (
            uint8_t device,
            uint8_t recipient,
            uint8_t index,
            uint8_t * ptr_stat_dat )
```

Standard Device Request on Default Pipe - GET_STATUS.

Parameters

in	*device*	index of USB Device.
in	*recipient*	recipient.
in	*index*	interface or endpoint index.
out	*ptr_stat_dat*	pointer to where status data should be received.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Kneron Confidential

6.102.3.21 USBH_DeviceRequest_SetAddress()

```
usbStatus USBH_DeviceRequest_SetAddress (
            uint8_t device,
            uint8_t device_address )
```

Standard Device Request on Default Pipe - SET_ADDRESS.

Parameters

in	*device*	index of USB Device.
in	*device_address*	device address.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.22 USBH_DeviceRequest_SetConfiguration()

```
usbStatus USBH_DeviceRequest_SetConfiguration (
            uint8_t device,
            uint8_t configuration )
```

Standard Device Request on Default Pipe - SET_CONFIGURATION.

Parameters

in	*device*	index of USB Device.
in	*configuration*	configuration.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.23 USBH_DeviceRequest_SetDescriptor()

```
usbStatus USBH_DeviceRequest_SetDescriptor (
            uint8_t device,
            uint8_t recipient,
            uint8_t descriptor_type,
            uint8_t descriptor_index,
            uint16_t language_id,
            const uint8_t * descriptor_data,
            uint16_t descriptor_length )
```

Standard Device Request on Default Pipe - SET_DESCRIPTOR.

Kneron Confidential

Parameters

in	*device*	index of USB Device.
in	*recipient*	recipient.
in	*descriptor_type*	descriptor type.
in	*descriptor_index*	descriptor index.
in	*language_id*	language ID.
in	*descriptor_data*	pointer to descriptor data to be written.
in	*descriptor_length*	descriptor length.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.24 USBH_DeviceRequest_SetFeature()

```
usbStatus USBH_DeviceRequest_SetFeature (
            uint8_t device,
            uint8_t recipient,
            uint8_t index,
            uint8_t feature_selector )
```

Standard Device Request on Default Pipe - SET_FEATURE.

Parameters

in	*device*	index of USB Device.
in	*recipient*	recipient.
in	*index*	interface or endpoint index.
in	*feature_selector*	feature selector.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.25 USBH_DeviceRequest_SetInterface()

```
usbStatus USBH_DeviceRequest_SetInterface (
            uint8_t device,
            uint8_t index,
            uint8_t alternate )
```

Standard Device Request on Default Pipe - SET_INTERFACE.

Kneron Confidential

Parameters

in	*device*	index of USB Device.
in	*index*	interface index.
in	*alternate*	alternate setting.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:

6.102.3.26 USBH_DeviceRequest_SynchFrame()

```
usbStatus USBH_DeviceRequest_SynchFrame (
            uint8_t device,
            uint8_t index,
            uint8_t * ptr_frame_number )
```

Standard Device Request on Default Pipe - SYNCH_FRAME.

Parameters

in	*device*	index of USB Device.
in	*index*	interface or endpoint index.
out	*ptr_frame_number*	pointer to where frame number data will be read.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.27 USBH_GetVersion()

```
uint32_t USBH_GetVersion (
            void  )
```

Get version of USB Host stack.

Kneron Confidential

Returns

version (major.minor.revision : mmnnnrrrr decimal)

6.102.3.28 USBH_Initialize()

```
usbStatus USBH_Initialize (
            uint8_t ctrl )
```

Initialize USB Host stack and controller.

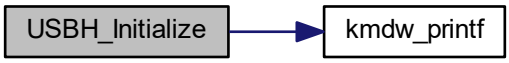
Parameters

in	*ctrl*	index of USB Host controller.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



6.102.3.29 USBH_Notify()

```
void USBH_Notify (
            uint8_t ctrl,
            uint8_t port,
            uint8_t device,
            USBH_NOTIFY notify )
```

Callback function called when some event has happened on corresponding controller and port.

Parameters

in	*ctrl*	index of USB Host controller.
in	*port*	index of Root HUB port.

Kneron Confidential

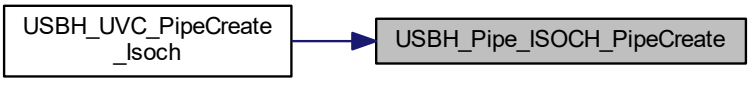
Parameters

in	*device*	index of USB Device : • value <= 127: index of of USB Device for device notifications • value 255: no device information for port notifications
in	*notify*	notification : • USBH_NOTIFY_CONNECT = Port connection happened • USBH_NOTIFY_DISCONNECT = Port disconnection happened • USBH_NOTIFY_OVERCURRENT = Port overcurrent happened • USBH_NOTIFY_REMOTE_WAKEUP = Port remote wakeup signaling happened • USBH_NOTIFY_READY = Device was successfully enumerated, initialized and is ready for communication • USBH_NOTIFY_UNKNOWN_DEVICE = Device was successfully enumerated but there is no driver for it • USBH_NOTIFY_INSUFFICIENT_POWER = Device requires more power consumption than available • USBH_NOTIFY_CONFIGURATION_FAILED = Device was not successfully configured (not enough resources) • USBH_NOTIFY_INITIALIZATION_FAILED = Device was not successfully initialized

6.102.3.30 USBH_Pipe_ISOCH_PipeCreate()

```
USBH_PIPE_HANDLE USBH_Pipe_ISOCH_PipeCreate (
            uint8_t device,
            uint8_t ep_addr,
            uint32_t wMaxPacketSize,
            uint8_t bInterval,
            uint8_t * buf,
            uint32_t buf_size )
```

Here is the caller graph for this function:



Kneron Confidential

6.102.3.31 USBH_Pipe_ISOCH_Start()

```
usbStatus USBH_Pipe_ISOCH_Start (
            USBH_PIPE_HANDLE pipe_hndl,
            USBH_CB_ISR_Isoch_transfer isoch_cb )
```

6.102.3.32 USBH_Pipe_ISOCH_Stop()

```
usbStatus USBH_Pipe_ISOCH_Stop (
            USBH_PIPE_HANDLE pipe_hndl )
```

6.102.3.33 USBH_PipeAbort()

```
usbStatus USBH_PipeAbort (
            USBH_PIPE_HANDLE pipe_hndl )
```

Abort send/receive operation on Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.34 USBH_PipeCreate()

```
USBH_PIPE_HANDLE USBH_PipeCreate (
            uint8_t device,
            uint8_t ep_addr,
            uint8_t ep_type,
            uint16_t ep_max_packet_size,
            uint8_t ep_interval )
```

Create Pipe.

Parameters

in	*device*	index of USB Device.
in	*ep_addr*	endpoint address : • ep_addr.0..3 : address • ep_addr.7 : direction
in	*ep_type*	endpoint type.
in	*ep_max_packet_size*	endpoint maximum packet size.
in	*ep_interval*	endpoint polling interval.

Kneron Confidential

Returns

pipe handle or pipe creation failed :

- value $>$ 0 : pipe handle
- value 0 : pipe creation failed

6.102.3.35 USBH_PipeDelete()

```
usbStatus USBH_PipeDelete (
            USBH_PIPE_HANDLE pipe_hndl )
```

Delete Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.36 USBH_PipeReceive()

```
usbStatus USBH_PipeReceive (
            USBH_PIPE_HANDLE pipe_hndl,
            uint8_t * buf,
            uint32_t len )
```

Receive data on Pipe.

Parameters

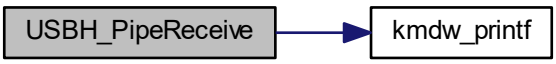
in	*pipe_hndl*	pipe handle.
out	*buf*	buffer that receives data.
in	*len*	maximum number of bytes to receive.

Kneron Confidential

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



6.102.3.37 USBH_PipeReceiveGetResult()

```
uint32_t USBH_PipeReceiveGetResult (
            USBH_PIPE_HANDLE pipe_hndl )
```

Get result of receive data operation on Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

number of successfully received data bytes.

6.102.3.38 USBH_PipeReset()

```
usbStatus USBH_PipeReset (
            USBH_PIPE_HANDLE pipe_hndl )
```

Reset Pipe (reset data toggle)

Parameters

in	*pipe_hndl*	pipe handle.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Kneron Confidential

6.102.3.39 USBH_PipeSend()

```
usbStatus USBH_PipeSend (
            USBH_PIPE_HANDLE pipe_hndl,
            const uint8_t * buf,
            uint32_t len )
```

Send data on Pipe.

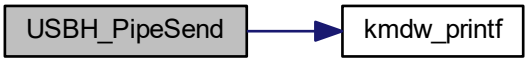
Parameters

in	*pipe_hndl*	pipe handle.
in	*buf*	buffer containing data bytes to send.
in	*len*	number of bytes to send.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:



6.102.3.40 USBH_PipeSendGetResult()

```
uint32_t USBH_PipeSendGetResult (
            USBH_PIPE_HANDLE pipe_hndl )
```

Get result of send data operation on Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

number of successfully sent data bytes.

Kneron Confidential

6.102.3.41 USBH_PipeUpdate()

```
usbStatus USBH_PipeUpdate (
            USBH_PIPE_HANDLE pipe_hndl )
```

Update Pipe.

Parameters

in	*pipe_hndl*	pipe handle.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.42 USBH_Port_Resume()

```
usbStatus USBH_Port_Resume (
            uint8_t ctrl,
            uint8_t port )
```

Resume a root HUB port on specified controller.

Parameters

in	*ctrl*	index of USB Host controller.
in	*port*	root HUB port.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.43 USBH_Port_Suspend()

```
usbStatus USBH_Port_Suspend (
            uint8_t ctrl,
            uint8_t port )
```

Suspend a root HUB port on specified controller.

Kneron Confidential

Parameters

in	*ctrl*	index of USB Host controller.
in	*port*	root HUB port.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.102.3.44 USBH_Uninitialize()

```
usbStatus USBH_Uninitialize (
            uint8_t ctrl )
```

De-initialize USB Host stack and controller.

Parameters

in	*ctrl*	index of USB Host controller.

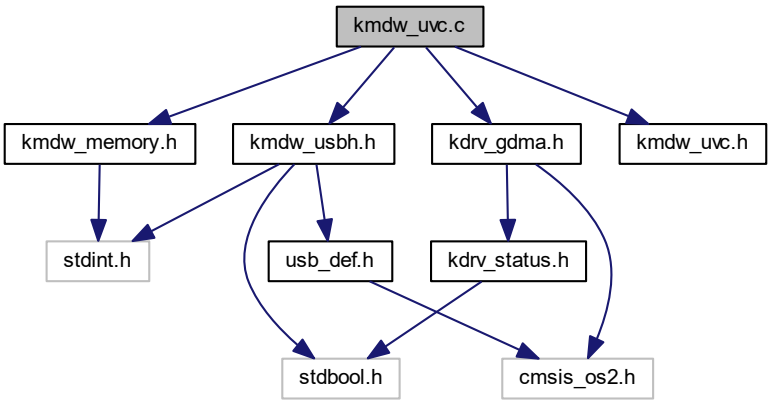
Returns

status code that indicates the execution status of the function as defined with usbStatus.

6.103 kmdw_uvc.c File Reference

```
#include "kmdw_usbh.h"
#include "kmdw_uvc.h"
#include "kmdw_memory.h"
#include "kdrv_gdma.h"
```

Include dependency graph for kmdw_uvc.c:



Kneron Confidential

Data Structures

- struct _UVC_FRAME_LINK

Macros

- #define UVC_MAX_FRAME 10
- #define ITD_BUF_SIZE (4 ∗ 1024 ∗ 1024)
- #define UVC_HEADER_SIZE 12
- #define USBH_UVC_Send_Frame USBH_UVC_Get_Frame

Typedefs

- typedef struct _UVC_FRAME_LINK UVC_FRAME_LINK

Functions

- __weak uint8_t USBH_UVC_Configure (uint8_t device, const USB_DEVICE_DESCRIPTOR ∗ptr_dev_desc, const USB_CONFIGURATION_DESCRIPTOR ∗ptr_cfg_desc)
- __weak usbStatus USBH_UVC_Initialize (uint8_t instance)
- __weak void USBH_UVC_Get_Frame (uint32_t ∗frame_ptr, uint32_t frame_size)
- uint8_t USBH_CustomClass_Configure (uint8_t device, const USB_DEVICE_DESCRIPTOR ∗ptr_dev_desc, const USB_CONFIGURATION_DESCRIPTOR ∗ptr_cfg_desc)

  Callback function called when custom class device is connected and needs to configure resources used by custom class device instance.
- usbStatus USBH_CustomClass_Initialize (uint8_t instance)

  Callback function called when custom class device is connected and needs to initialize custom class device instance.
- usbStatus USBH_UVC_VS_Control (uint8_t device, UVC_VS_Request_t vs_req, UVC_VS_ControlSelector_t cs, UVC_PROBE_COMMIT_CONTROL ∗upc_ctrl)
- USBH_PIPE_HANDLE USBH_UVC_PipeCreate_Isoch (uint8_t device, uint8_t ep_addr, uint32_t wMaxPacketSize, uint8_t bInterval)
- void uvc_isoch_cb (uint32_t ∗payload, uint32_t length)
- usbStatus USBH_UVC_PipeStart_Isoch (USBH_PIPE_HANDLE pipe_hndl)
- usbStatus USBH_UVC_PipeStop_Isoch (USBH_PIPE_HANDLE pipe_hndl)
- usbStatus USBH_UVC_Queue_Frame (USBH_PIPE_HANDLE pipe, uint32_t ∗frame_ptr, uint32_t size)

6.103.1 Macro Definition Documentation

6.103.1.1 ITD_BUF_SIZE

```
#define ITD_BUF_SIZE (4 * 1024 * 1024)
```

Kneron Confidential

6.103.1.2 USBH_UVC_Send_Frame

```
#define USBH_UVC_Send_Frame USBH_UVC_Get_Frame
```

6.103.1.3 UVC_HEADER_SIZE

```
#define UVC_HEADER_SIZE 12
```

6.103.1.4 UVC_MAX_FRAME

```
#define UVC_MAX_FRAME 10
```

6.103.2 Typedef Documentation

6.103.2.1 UVC_FRAME_LINK

```
typedef struct _UVC_FRAME_LINK UVC_FRAME_LINK
```

6.103.3 Function Documentation

6.103.3.1 USBH_CustomClass_Configure()

```
uint8_t USBH_CustomClass_Configure (
        uint8_t device,
        const USB_DEVICE_DESCRIPTOR * ptr_dev_desc,
        const USB_CONFIGURATION_DESCRIPTOR * ptr_cfg_desc )
```

Callback function called when custom class device is connected and needs to configure resources used by custom class device instance.

Parameters

in	*device*	index of USB Device.
in	*ptr_dev_desc*	pointer to device descriptor.
in	*ptr_cfg_desc*	pointer to configuration descriptor.

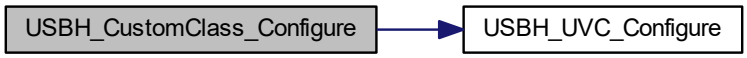
Kneron Confidential

Returns

index of configured custom class device instance or configuration failed :

- value <= 127 : index of configured custom class device instance
- value 255 : configuration failed

Here is the call graph for this function:



6.103.3.2 USBH_CustomClass_Initialize()

```
usbStatus USBH_CustomClass_Initialize (
            uint8_t instance )
```

Callback function called when custom class device is connected and needs to initialize custom class device instance.

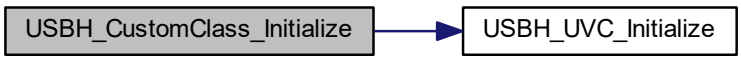
Parameters

in	*instance*	index of custom class device instance.

Returns

status code that indicates the execution status of the function as defined with usbStatus.

Here is the call graph for this function:

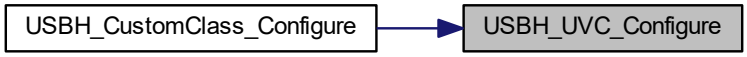


Kneron Confidential

6.103.3.3 USBH_UVC_Configure()

```
__weak uint8_t USBH_UVC_Configure (
             uint8_t device,
             const USB_DEVICE_DESCRIPTOR * ptr_dev_desc,
             const USB_CONFIGURATION_DESCRIPTOR * ptr_cfg_desc )
```

Here is the caller graph for this function:



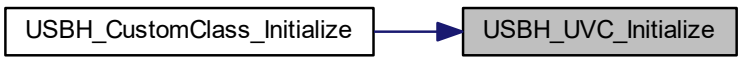
6.103.3.4 USBH_UVC_Get_Frame()

```
__weak void USBH_UVC_Get_Frame (
             uint32_t * frame_ptr,
             uint32_t frame_size )
```

6.103.3.5 USBH_UVC_Initialize()

```
__weak usbStatus USBH_UVC_Initialize (
             uint8_t instance )
```

Here is the caller graph for this function:

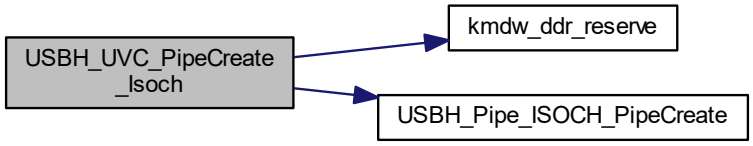


Kneron Confidential

6.103.3.6 USBH_UVC_PipeCreate_Isoch()

```
USBH_PIPE_HANDLE USBH_UVC_PipeCreate_Isoch (
            uint8_t device,
            uint8_t ep_addr,
            uint32_t wMaxPacketSize,
            uint8_t bInterval )
```

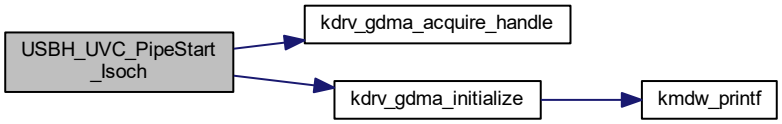
Here is the call graph for this function:



6.103.3.7 USBH_UVC_PipeStart_Isoch()

```
usbStatus USBH_UVC_PipeStart_Isoch (
            USBH_PIPE_HANDLE pipe_hndl )
```

Here is the call graph for this function:



6.103.3.8 USBH_UVC_PipeStop_Isoch()

```
usbStatus USBH_UVC_PipeStop_Isoch (
            USBH_PIPE_HANDLE pipe_hndl )
```

Kneron Confidential

Here is the call graph for this function:

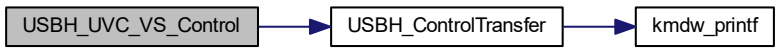
6.103.3.9 USBH_UVC_Queue_Frame()

```
usbStatus USBH_UVC_Queue_Frame (
            USBH_PIPE_HANDLE pipe,
            uint32_t * frame_ptr,
            uint32_t size )
```

6.103.3.10 USBH_UVC_VS_Control()

```
usbStatus USBH_UVC_VS_Control (
            uint8_t device,
            UVC_VS_Request_t vs_req,
            UVC_VS_ControlSelector_t cs,
            UVC_PROBE_COMMIT_CONTROL * upc_ctrl )
```

Here is the call graph for this function:



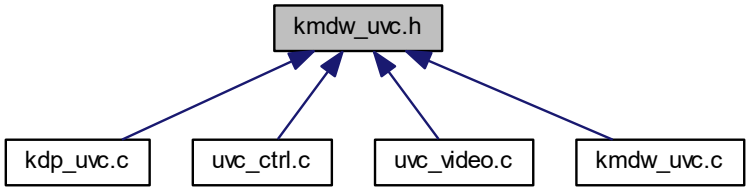
6.103.3.11 uvc_isoch_cb()

```
void uvc_isoch_cb (
            uint32_t * payload,
            uint32_t length )
```

Kneron Confidential

6.104 kmdw_uvc.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum UVC_VS_Request_t { SET_CUR = 0x01, GET_CUR = 0x81 }
- enum UVC_VS_ControlSelector_t { VS_PROBE_CONTROL = 0x100, VS_COMMIT_CONTROL = 0x200 }

Functions

- struct __attribute__ ((__packed__))
  8-byte setup packet struct
- uint8_t USBH_UVC_Configure (uint8_t device, const USB_DEVICE_DESCRIPTOR *ptr_dev_desc, const USB_CONFIGURATION_DESCRIPTOR *ptr_cfg_desc)
- USBH_PIPE_HANDLE USBH_UVC_PipeCreate_Isoch (uint8_t device, uint8_t ep_addr, uint32_t wMaxPacketSize, uint8_t bInterval)
- usbStatus USBH_UVC_PipeStart_Isoch (USBH_PIPE_HANDLE pipe_hndl)
- usbStatus USBH_UVC_PipeStop_Isoch (USBH_PIPE_HANDLE pipe_hndl)
- usbStatus USBH_UVC_Initialize (uint8_t instance)
- usbStatus USBH_UVC_VS_Control (uint8_t device, UVC_VS_Request_t vs_req, UVC_VS_ControlSelector_t cs, UVC_PROBE_COMMIT_CONTROL *upc_ctrl)
- usbStatus USBH_UVC_Queue_Frame (USBH_PIPE_HANDLE pipe, uint32_t *frame_ptr, uint32_t size)
- void USBH_UVC_Get_Frame (uint32_t *frame_ptr, uint32_t frame_size)

Variables

- UVC_PROBE_COMMIT_CONTROL

6.104.1 Enumeration Type Documentation

6.104.1.1 UVC_VS_ControlSelector_t

```
enum UVC_VS_ControlSelector_t
```

Kneron Confidential

Enumerator

VS_PROBE_CONTROL	
VS_COMMIT_CONTROL	

6.104.1.2 UVC_VS_Request_t

```
enum UVC_VS_Request_t
```

Enumerator

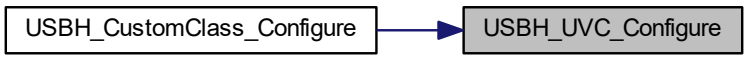
SET_CUR	
GET_CUR	

6.104.2 Function Documentation

6.104.2.1 USBH_UVC_Configure()

```
uint8_t USBH_UVC_Configure (
            uint8_t device,
            const USB_DEVICE_DESCRIPTOR * ptr_dev_desc,
            const USB_CONFIGURATION_DESCRIPTOR * ptr_cfg_desc )
```

Here is the caller graph for this function:



6.104.2.2 USBH_UVC_Get_Frame()

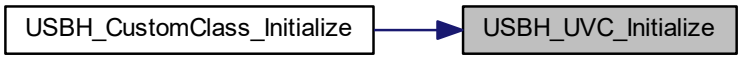
```
void USBH_UVC_Get_Frame (
            uint32_t * frame_ptr,
            uint32_t frame_size )
```

Kneron Confidential

6.104.2.3 USBH_UVC_Initialize()

```
usbStatus USBH_UVC_Initialize (
            uint8_t instance )
```

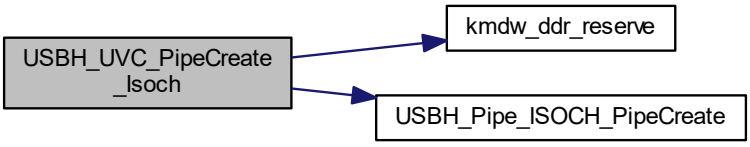
Here is the caller graph for this function:



6.104.2.4 USBH_UVC_PipeCreate_Isoch()

```
USBH_PIPE_HANDLE USBH_UVC_PipeCreate_Isoch (
            uint8_t device,
            uint8_t ep_addr,
            uint32_t wMaxPacketSize,
            uint8_t bInterval )
```

Here is the call graph for this function:

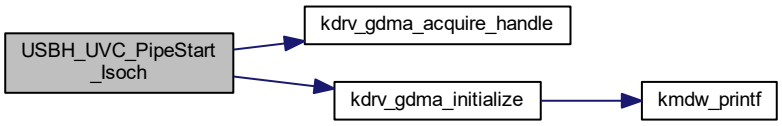


6.104.2.5 USBH_UVC_PipeStart_Isoch()

```
usbStatus USBH_UVC_PipeStart_Isoch (
            USBH_PIPE_HANDLE pipe_hndl )
```

Kneron Confidential

Here is the call graph for this function:



6.104.2.6 USBH_UVC_PipeStop_Isoch()

```
usbStatus USBH_UVC_PipeStop_Isoch (
            USBH_PIPE_HANDLE pipe_hndl )
```

Here is the call graph for this function:

6.104.2.7 USBH_UVC_Queue_Frame()

```
usbStatus USBH_UVC_Queue_Frame (
            USBH_PIPE_HANDLE pipe,
            uint32_t * frame_ptr,
            uint32_t size )
```

6.104.2.8 USBH_UVC_VS_Control()

```
usbStatus USBH_UVC_VS_Control (
            uint8_t device,
            UVC_VS_Request_t vs_req,
            UVC_VS_ControlSelector_t cs,
            UVC_PROBE_COMMIT_CONTROL * upc_ctrl )
```

Kneron Confidential

Here is the call graph for this function:

6.104.3 Variable Documentation

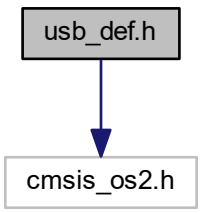
6.104.3.1 UVC_PROBE_COMMIT_CONTROL

```
UVC_PROBE_COMMIT_CONTROL
```

6.105 usb_def.h File Reference

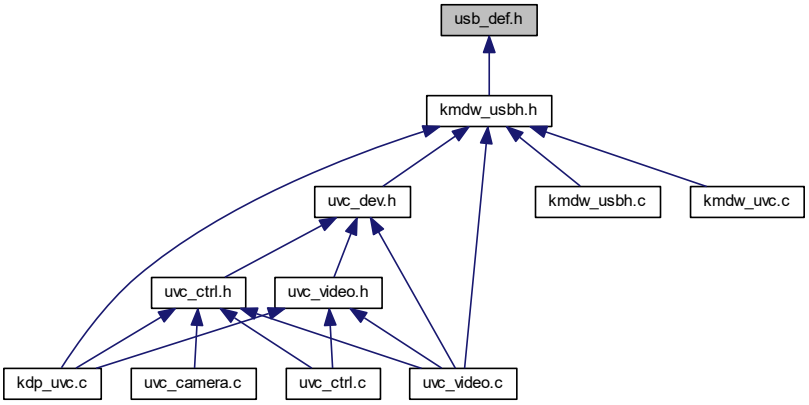
```
#include "cmsis_os2.h"
```

Include dependency graph for usb_def.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Data Structures

- struct USB_REQUEST_TYPE

  bmRequestType Definition
- struct USB_SETUP_PACKET

  USB Default Control Pipe Setup Packet.

Macros

- #define USB_SPEED_LOW 0U
- #define USB_SPEED_FULL 1U
- #define USB_SPEED_HIGH 2U
- #define USB_PID_RESERVED 0U
- #define USB_PID_OUT 1U
- #define USB_PID_ACK 2U
- #define USB_PID_DATA0 3U
- #define USB_PID_PING 4U
- #define USB_PID_SOF 5U
- #define USB_PID_DATA2 7U
- #define USB_PID_NYET 6U
- #define USB_PID_SPLIT 8U
- #define USB_PID_IN 9U
- #define USB_PID_NAK 10U
- #define USB_PID_DATA1 11U
- #define USB_PID_PRE 12U
- #define USB_PID_ERR 12U
- #define USB_PID_SETUP 13U
- #define USB_PID_STALL 14U
- #define USB_PID_MDATA 15U
- #define USB_REQUEST_HOST_TO_DEVICE 0U
- #define USB_REQUEST_DEVICE_TO_HOST 1U
- #define USB_REQUEST_STANDARD 0U
- #define USB_REQUEST_CLASS 1U

Kneron Confidential

- #define USB_REQUEST_VENDOR 2U
- #define USB_REQUEST_RESERVED 3U
- #define USB_REQUEST_TO_DEVICE 0U
- #define USB_REQUEST_TO_INTERFACE 1U
- #define USB_REQUEST_TO_ENDPOINT 2U
- #define USB_REQUEST_TO_OTHER 3U
- #define USB_REQUEST_GET_STATUS 0U
- #define USB_REQUEST_CLEAR_FEATURE 1U
- #define USB_REQUEST_SET_FEATURE 3U
- #define USB_REQUEST_SET_ADDRESS 5U
- #define USB_REQUEST_GET_DESCRIPTOR 6U
- #define USB_REQUEST_SET_DESCRIPTOR 7U
- #define USB_REQUEST_GET_CONFIGURATION 8U
- #define USB_REQUEST_SET_CONFIGURATION 9U
- #define USB_REQUEST_GET_INTERFACE 10U
- #define USB_REQUEST_SET_INTERFACE 11U
- #define USB_REQUEST_SYNC_FRAME 12U
- #define USB_GETSTATUS_SELF_POWERED 0x01U
- #define USB_GETSTATUS_REMOTE_WAKEUP 0x02U
- #define USB_GETSTATUS_ENDPOINT_STALL 0x01U
- #define USB_FEATURE_ENDPOINT_STALL 0U
- #define USB_FEATURE_REMOTE_WAKEUP 1U
- #define USB_DEVICE_DESCRIPTOR_TYPE 1U
- #define USB_CONFIGURATION_DESCRIPTOR_TYPE 2U
- #define USB_STRING_DESCRIPTOR_TYPE 3U
- #define USB_INTERFACE_DESCRIPTOR_TYPE 4U
- #define USB_ENDPOINT_DESCRIPTOR_TYPE 5U
- #define USB_DEVICE_QUALIFIER_DESCRIPTOR_TYPE 6U
- #define USB_OTHER_SPEED_CONFIG_DESCRIPTOR_TYPE 7U
- #define USB_INTERFACE_POWER_DESCRIPTOR_TYPE 8U
- #define USB_OTG_DESCRIPTOR_TYPE 9U
- #define USB_DEBUG_DESCRIPTOR_TYPE 10U
- #define USB_INTERFACE_ASSOCIATION_DESCRIPTOR_TYPE 11U
- #define USB_DEVICE_CLASS_RESERVED 0x00U
- #define USB_DEVICE_CLASS_AUDIO 0x01U
- #define USB_DEVICE_CLASS_COMMUNICATIONS 0x02U
- #define USB_DEVICE_CLASS_HUMAN_INTERFACE 0x03U
- #define USB_DEVICE_CLASS_MONITOR 0x04U
- #define USB_DEVICE_CLASS_PHYSICAL_INTERFACE 0x05U
- #define USB_DEVICE_CLASS_POWER 0x06U
- #define USB_DEVICE_CLASS_PRINTER 0x07U
- #define USB_DEVICE_CLASS_STORAGE 0x08U
- #define USB_DEVICE_CLASS_HUB 0x09U
- #define USB_DEVICE_CLASS_MISCELLANEOUS 0xEFU
- #define USB_DEVICE_CLASS_VENDOR_SPECIFIC 0xFFU
- #define USB_CONFIG_POWERED_MASK 0x40U
- #define USB_CONFIG_BUS_POWERED 0x80U
- #define USB_CONFIG_SELF_POWERED 0xC0U
- #define USB_CONFIG_REMOTE_WAKEUP 0x20U
- #define USB_CONFIG_POWER_MA(mA) ((mA)/2)
- #define USB_ENDPOINT_DIRECTION_MASK 0x80U
- #define USB_ENDPOINT_OUT(addr) (addr)
- #define USB_ENDPOINT_IN(addr) ((addr) | 0x80U)
- #define USB_ENDPOINT_TYPE_MASK 0x03U
- #define USB_ENDPOINT_TYPE_CONTROL 0x00U

Kneron Confidential

- #define USB_ENDPOINT_TYPE_ISOCHRONOUS 0x01U
- #define USB_ENDPOINT_TYPE_BULK 0x02U
- #define USB_ENDPOINT_TYPE_INTERRUPT 0x03U
- #define USB_ENDPOINT_SYNC_MASK 0x0CU
- #define USB_ENDPOINT_SYNC_NO_SYNCHRONIZATION 0x00U
- #define USB_ENDPOINT_SYNC_ASYNCHRONOUS 0x04U
- #define USB_ENDPOINT_SYNC_ADAPTIVE 0x08U
- #define USB_ENDPOINT_SYNC_SYNCHRONOUS 0x0CU
- #define USB_ENDPOINT_USAGE_MASK 0x30U
- #define USB_ENDPOINT_USAGE_DATA 0x00U
- #define USB_ENDPOINT_USAGE_FEEDBACK 0x10U
- #define USB_ENDPOINT_USAGE_IMPLICIT_FEEDBACK 0x20U
- #define USB_ENDPOINT_USAGE_RESERVED 0x30U

Functions

- struct __attribute__ ((__packed__))

  USB Standard Device Descriptor.

Variables

- USB_DEVICE_DESCRIPTOR
- USB_DEVICE_QUALIFIER_DESCRIPTOR
- USB_CONFIGURATION_DESCRIPTOR
- USB_INTERFACE_DESCRIPTOR
- USB_ENDPOINT_DESCRIPTOR
- USB_STRING_DESCRIPTOR
- USB_COMMON_DESCRIPTOR
- USB_INTERFACE_ASSOCIATION_DESCRIPTOR

6.105.1 Macro Definition Documentation

6.105.1.1 USB_CONFIG_BUS_POWERED

```
#define USB_CONFIG_BUS_POWERED 0x80U
```

6.105.1.2 USB_CONFIG_POWER_MA

```
#define USB_CONFIG_POWER_MA(
                mA ) ((mA)/2)
```

Kneron Confidential

6.105.1.3 USB_CONFIG_POWERED_MASK

```
#define USB_CONFIG_POWERED_MASK 0x40U
```

6.105.1.4 USB_CONFIG_REMOTE_WAKEUP

```
#define USB_CONFIG_REMOTE_WAKEUP 0x20U
```

6.105.1.5 USB_CONFIG_SELF_POWERED

```
#define USB_CONFIG_SELF_POWERED 0xC0U
```

6.105.1.6 USB_CONFIGURATION_DESCRIPTOR_TYPE

```
#define USB_CONFIGURATION_DESCRIPTOR_TYPE 2U
```

6.105.1.7 USB_DEBUG_DESCRIPTOR_TYPE

```
#define USB_DEBUG_DESCRIPTOR_TYPE 10U
```

6.105.1.8 USB_DEVICE_CLASS_AUDIO

```
#define USB_DEVICE_CLASS_AUDIO 0x01U
```

6.105.1.9 USB_DEVICE_CLASS_COMMUNICATIONS

```
#define USB_DEVICE_CLASS_COMMUNICATIONS 0x02U
```

6.105.1.10 USB_DEVICE_CLASS_HUB

```
#define USB_DEVICE_CLASS_HUB 0x09U
```

Kneron Confidential

6.105.1.11 USB_DEVICE_CLASS_HUMAN_INTERFACE

```
#define USB_DEVICE_CLASS_HUMAN_INTERFACE 0x03U
```

6.105.1.12 USB_DEVICE_CLASS_MISCELLANEOUS

```
#define USB_DEVICE_CLASS_MISCELLANEOUS 0xEFU
```

6.105.1.13 USB_DEVICE_CLASS_MONITOR

```
#define USB_DEVICE_CLASS_MONITOR 0x04U
```

6.105.1.14 USB_DEVICE_CLASS_PHYSICAL_INTERFACE

```
#define USB_DEVICE_CLASS_PHYSICAL_INTERFACE 0x05U
```

6.105.1.15 USB_DEVICE_CLASS_POWER

```
#define USB_DEVICE_CLASS_POWER 0x06U
```

6.105.1.16 USB_DEVICE_CLASS_PRINTER

```
#define USB_DEVICE_CLASS_PRINTER 0x07U
```

6.105.1.17 USB_DEVICE_CLASS_RESERVED

```
#define USB_DEVICE_CLASS_RESERVED 0x00U
```

6.105.1.18 USB_DEVICE_CLASS_STORAGE

```
#define USB_DEVICE_CLASS_STORAGE 0x08U
```

Kneron Confidential

6.105.1.19 USB_DEVICE_CLASS_VENDOR_SPECIFIC

```
#define USB_DEVICE_CLASS_VENDOR_SPECIFIC 0xFFU
```

6.105.1.20 USB_DEVICE_DESCRIPTOR_TYPE

```
#define USB_DEVICE_DESCRIPTOR_TYPE 1U
```

6.105.1.21 USB_DEVICE_QUALIFIER_DESCRIPTOR_TYPE

```
#define USB_DEVICE_QUALIFIER_DESCRIPTOR_TYPE 6U
```

6.105.1.22 USB_ENDPOINT_DESCRIPTOR_TYPE

```
#define USB_ENDPOINT_DESCRIPTOR_TYPE 5U
```

6.105.1.23 USB_ENDPOINT_DIRECTION_MASK

```
#define USB_ENDPOINT_DIRECTION_MASK 0x80U
```

6.105.1.24 USB_ENDPOINT_IN

```
#define USB_ENDPOINT_IN(
            addr ) ((addr) | 0x80U)
```

6.105.1.25 USB_ENDPOINT_OUT

```
#define USB_ENDPOINT_OUT(
             addr ) ( addr )
```

Kneron Confidential

6.105.1.26 USB_ENDPOINT_SYNC_ADAPTIVE

```
#define USB_ENDPOINT_SYNC_ADAPTIVE 0x08U
```

6.105.1.27 USB_ENDPOINT_SYNC_ASYNCHRONOUS

```
#define USB_ENDPOINT_SYNC_ASYNCHRONOUS 0x04U
```

6.105.1.28 USB_ENDPOINT_SYNC_MASK

```
#define USB_ENDPOINT_SYNC_MASK 0x0CU
```

6.105.1.29 USB_ENDPOINT_SYNC_NO_SYNCHRONIZATION

```
#define USB_ENDPOINT_SYNC_NO_SYNCHRONIZATION 0x00U
```

6.105.1.30 USB_ENDPOINT_SYNC_SYNCHRONOUS

```
#define USB_ENDPOINT_SYNC_SYNCHRONOUS 0x0CU
```

6.105.1.31 USB_ENDPOINT_TYPE_BULK

```
#define USB_ENDPOINT_TYPE_BULK 0x02U
```

6.105.1.32 USB_ENDPOINT_TYPE_CONTROL

```
#define USB_ENDPOINT_TYPE_CONTROL 0x00U
```

6.105.1.33 USB_ENDPOINT_TYPE_INTERRUPT

```
#define USB_ENDPOINT_TYPE_INTERRUPT 0x03U
```

Kneron Confidential

6.105.1.34 USB_ENDPOINT_TYPE_ISOCHRONOUS

```
#define USB_ENDPOINT_TYPE_ISOCHRONOUS 0x01U
```

6.105.1.35 USB_ENDPOINT_TYPE_MASK

```
#define USB_ENDPOINT_TYPE_MASK 0x03U
```

6.105.1.36 USB_ENDPOINT_USAGE_DATA

```
#define USB_ENDPOINT_USAGE_DATA 0x00U
```

6.105.1.37 USB_ENDPOINT_USAGE_FEEDBACK

```
#define USB_ENDPOINT_USAGE_FEEDBACK 0x10U
```

6.105.1.38 USB_ENDPOINT_USAGE_IMPLICIT_FEEDBACK

```
#define USB_ENDPOINT_USAGE_IMPLICIT_FEEDBACK 0x20U
```

6.105.1.39 USB_ENDPOINT_USAGE_MASK

```
#define USB_ENDPOINT_USAGE_MASK 0x30U
```

6.105.1.40 USB_ENDPOINT_USAGE_RESERVED

```
#define USB_ENDPOINT_USAGE_RESERVED 0x30U
```

6.105.1.41 USB_FEATURE_ENDPOINT_STALL

```
#define USB_FEATURE_ENDPOINT_STALL 0U
```

Kneron Confidential

6.105.1.42 USB_FEATURE_REMOTE_WAKEUP

```
#define USB_FEATURE_REMOTE_WAKEUP 1U
```

6.105.1.43 USB_GETSTATUS_ENDPOINT_STALL

```
#define USB_GETSTATUS_ENDPOINT_STALL 0x01U
```

6.105.1.44 USB_GETSTATUS_REMOTE_WAKEUP

```
#define USB_GETSTATUS_REMOTE_WAKEUP 0x02U
```

6.105.1.45 USB_GETSTATUS_SELF_POWERED

```
#define USB_GETSTATUS_SELF_POWERED 0x01U
```

6.105.1.46 USB_INTERFACE_ASSOCIATION_DESCRIPTOR_TYPE

```
#define USB_INTERFACE_ASSOCIATION_DESCRIPTOR_TYPE 11U
```

6.105.1.47 USB_INTERFACE_DESCRIPTOR_TYPE

```
#define USB_INTERFACE_DESCRIPTOR_TYPE 4U
```

6.105.1.48 USB_INTERFACE_POWER_DESCRIPTOR_TYPE

```
#define USB_INTERFACE_POWER_DESCRIPTOR_TYPE 8U
```

6.105.1.49 USB_OTG_DESCRIPTOR_TYPE

```
#define USB_OTG_DESCRIPTOR_TYPE 9U
```

Kneron Confidential

6.105.1.50 USB_OTHER_SPEED_CONFIG_DESCRIPTOR_TYPE

```
#define USB_OTHER_SPEED_CONFIG_DESCRIPTOR_TYPE 7U
```

6.105.1.51 USB_PID_ACK

```
#define USB_PID_ACK 2U
```

6.105.1.52 USB_PID_DATA0

```
#define USB_PID_DATA0 3U
```

6.105.1.53 USB_PID_DATA1

```
#define USB_PID_DATA1 11U
```

6.105.1.54 USB_PID_DATA2

```
#define USB_PID_DATA2 7U
```

6.105.1.55 USB_PID_ERR

```
#define USB_PID_ERR 12U
```

6.105.1.56 USB_PID_IN

```
#define USB_PID_IN 9U
```

6.105.1.57 USB_PID_MDATA

```
#define USB_PID_MDATA 15U
```

Kneron Confidential

6.105.1.58 USB_PID_NAK

```
#define USB_PID_NAK 10U
```

6.105.1.59 USB_PID_NYET

```
#define USB_PID_NYET 6U
```

6.105.1.60 USB_PID_OUT

```
#define USB_PID_OUT 1U
```

6.105.1.61 USB_PID_PING

```
#define USB_PID_PING 4U
```

6.105.1.62 USB_PID_PRE

```
#define USB_PID_PRE 12U
```

6.105.1.63 USB_PID_RESERVED

```
#define USB_PID_RESERVED 0U
```

6.105.1.64 USB_PID_SETUP

```
#define USB_PID_SETUP 13U
```

6.105.1.65 USB_PID_SOF

```
#define USB_PID_SOF 5U
```

Kneron Confidential

6.105.1.66 USB_PID_SPLIT

```
#define USB_PID_SPLIT 8U
```

6.105.1.67 USB_PID_STALL

```
#define USB_PID_STALL 14U
```

6.105.1.68 USB_REQUEST_CLASS

```
#define USB_REQUEST_CLASS 1U
```

6.105.1.69 USB_REQUEST_CLEAR_FEATURE

```
#define USB_REQUEST_CLEAR_FEATURE 1U
```

6.105.1.70 USB_REQUEST_DEVICE_TO_HOST

```
#define USB_REQUEST_DEVICE_TO_HOST 1U
```

6.105.1.71 USB_REQUEST_GET_CONFIGURATION

```
#define USB_REQUEST_GET_CONFIGURATION 8U
```

6.105.1.72 USB_REQUEST_GET_DESCRIPTOR

```
#define USB_REQUEST_GET_DESCRIPTOR 6U
```

6.105.1.73 USB_REQUEST_GET_INTERFACE

```
#define USB_REQUEST_GET_INTERFACE 10U
```

Kneron Confidential

6.105.1.74 USB_REQUEST_GET_STATUS

```
#define USB_REQUEST_GET_STATUS 0U
```

6.105.1.75 USB_REQUEST_HOST_TO_DEVICE

```
#define USB_REQUEST_HOST_TO_DEVICE 0U
```

6.105.1.76 USB_REQUEST_RESERVED

```
#define USB_REQUEST_RESERVED 3U
```

6.105.1.77 USB_REQUEST_SET_ADDRESS

```
#define USB_REQUEST_SET_ADDRESS 5U
```

6.105.1.78 USB_REQUEST_SET_CONFIGURATION

```
#define USB_REQUEST_SET_CONFIGURATION 9U
```

6.105.1.79 USB_REQUEST_SET_DESCRIPTOR

```
#define USB_REQUEST_SET_DESCRIPTOR 7U
```

6.105.1.80 USB_REQUEST_SET_FEATURE

```
#define USB_REQUEST_SET_FEATURE 3U
```

6.105.1.81 USB_REQUEST_SET_INTERFACE

```
#define USB_REQUEST_SET_INTERFACE 11U
```

Kneron Confidential

6.105.1.82 USB_REQUEST_STANDARD

```
#define USB_REQUEST_STANDARD 0U
```

6.105.1.83 USB_REQUEST_SYNC_FRAME

```
#define USB_REQUEST_SYNC_FRAME 12U
```

6.105.1.84 USB_REQUEST_TO_DEVICE

```
#define USB_REQUEST_TO_DEVICE 0U
```

6.105.1.85 USB_REQUEST_TO_ENDPOINT

```
#define USB_REQUEST_TO_ENDPOINT 2U
```

6.105.1.86 USB_REQUEST_TO_INTERFACE

```
#define USB_REQUEST_TO_INTERFACE 1U
```

6.105.1.87 USB_REQUEST_TO_OTHER

```
#define USB_REQUEST_TO_OTHER 3U
```

6.105.1.88 USB_REQUEST_VENDOR

```
#define USB_REQUEST_VENDOR 2U
```

6.105.1.89 USB_SPEED_FULL

```
#define USB_SPEED_FULL 1U
```

Kneron Confidential

6.105.1.90 USB_SPEED_HIGH

```
#define USB_SPEED_HIGH 2U
```

6.105.1.91 USB_SPEED_LOW

```
#define USB_SPEED_LOW 0U
```

6.105.1.92 USB_STRING_DESCRIPTOR_TYPE

```
#define USB_STRING_DESCRIPTOR_TYPE 3U
```

6.105.2 Variable Documentation

6.105.2.1 USB_COMMON_DESCRIPTOR

```
USB_COMMON_DESCRIPTOR
```

6.105.2.2 USB_CONFIGURATION_DESCRIPTOR

```
USB_CONFIGURATION_DESCRIPTOR
```

6.105.2.3 USB_DEVICE_DESCRIPTOR

```
USB_DEVICE_DESCRIPTOR
```

6.105.2.4 USB_DEVICE_QUALIFIER_DESCRIPTOR

```
USB_DEVICE_QUALIFIER_DESCRIPTOR
```

Kneron Confidential

6.105.2.5 USB_ENDPOINT_DESCRIPTOR

```
USB_ENDPOINT_DESCRIPTOR
```

6.105.2.6 USB_INTERFACE_ASSOCIATION_DESCRIPTOR

```
USB_INTERFACE_ASSOCIATION_DESCRIPTOR
```

6.105.2.7 USB_INTERFACE_DESCRIPTOR

```
USB_INTERFACE_DESCRIPTOR
```

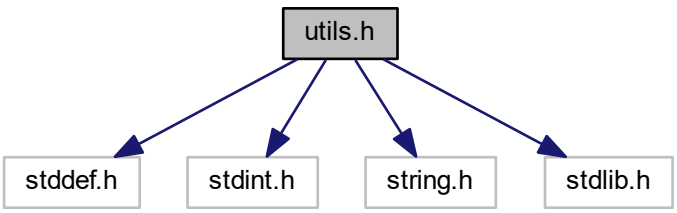
6.105.2.8 USB_STRING_DESCRIPTOR

```
USB_STRING_DESCRIPTOR
```

6.106 utils.h File Reference

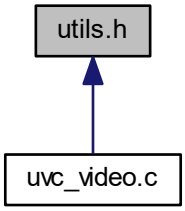
```
#include <stddef.h>
#include <stdint.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for utils.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Macros

- #define PAGE_SHIFT 12
- #define PAGE_SIZE (1 << PAGE_SHIFT)
- #define PAGE_MASK (~(PAGE_SIZE-1))
- #define ___PASTE(a, b) a##b
- #define __UNIQUE_ID(prefix) __PASTE(__PASTE(__UNIQUE_ID_, prefix), __COUNTER__)
- #define __min(t1, t2, min1, min2, x, y)
- #define min(x, y)
- #define __max(t1, t2, max1, max2, x, y)
- #define max(x, y)
- #define min3(x, y, z) min((typeof(x))min(x, y), z)
- #define max3(x, y, z) max((typeof(x))max(x, y), z)
- #define min_not_zero(x, y)
- #define clamp(val, lo, hi) min((typeof(val))max(val, lo), hi)
- #define min_t(type, x, y)
- #define max_t(type, x, y)
- #define clamp_t(type, val, lo, hi) min_t(type, max_t(type, val, lo), hi)
- #define clamp_val(val, lo, hi) clamp_t(typeof(val), val, lo, hi)
- #define swap(a, b) do { typeof(a) __tmp = (a); (a) = (b); (b) = __tmp; } while (0)
- #define container_of(ptr, type, member)
- #define ARRAY_SIZE(arr) (sizeof(arr) / sizeof((arr)[0]))
- #define BITS_PER_LONG 32
- #define UINT_MAX (~0U)

6.106.1 Macro Definition Documentation

6.106.1.1 ___PASTE

```
#define ___PASTE(
                a,
                b ) a##b
```

Kneron Confidential

6.106.1.2 __max

```
#define __max(
              t1,
              t2,
              max1,
              max2,
              x,
              y )
```

Value:

```
({      \
    t1 max1 = (x);                  \
    t2 max2 = (y);                  \
    (void) (&max1 == &max2);            \
    max1 > max2 ? max1 : max2; })
```

6.106.1.3 __min

```
#define __min(
              t1,
              t2,
              min1,
              min2,
              x,
              y )
```

Value:

```
({      \
    t1 min1 = (x);                  \
    t2 min2 = (y);                  \
    (void) (&min1 == &min2);            \
    min1 < min2 ? min1 : min2; })
```

6.106.1.4 __UNIQUE_ID

```
#define __UNIQUE_ID(
              prefix ) __PASTE(__PASTE(__UNIQUE_ID_, prefix), __COUNTER__)
```

6.106.1.5 ARRAY_SIZE

```
#define ARRAY_SIZE(
              arr ) (sizeof(arr) / sizeof((arr)[0]))
```

6.106.1.6 BITS_PER_LONG

```
#define BITS_PER_LONG 32
```

Kneron Confidential

6.106.1.7 clamp

```
#define clamp(
              val,
              lo,
              hi ) min((typeof(val))max(val, lo), hi)
```

clamp - return a value clamped to a given range with strict typechecking @val: current value @lo: lowest allowable value @hi: highest allowable value

This macro does strict typechecking of lo/hi to make sure they are of the same type as val. See the unnecessary pointer comparisons.

6.106.1.8 clamp_t

```
#define clamp_t(
              type,
              val,
              lo,
              hi ) min_t(type, max_t(type, val, lo), hi)
```

clamp_t - return a value clamped to a given range using a given type @type: the type of variable to use @val: current value @lo: minimum allowable value @hi: maximum allowable value

This macro does no typechecking and uses temporary variables of type 'type' to make all the comparisons.

6.106.1.9 clamp_val

```
#define clamp_val(
              val,
              lo,
              hi ) clamp_t(typeof(val), val, lo, hi)
```

clamp_val - return a value clamped to a given range using val's type @val: current value @lo: minimum allowable value @hi: maximum allowable value

This macro does no typechecking and uses temporary variables of whatever type the input argument 'val' is. This is useful when val is an unsigned type and min and max are literals that will otherwise be assigned a signed integer type.

6.106.1.10 container_of

```
#define container_of(
              ptr,
              type,
              member )
```

Value:

```
({          \
    const typeof( ((type *)0)->member ) *__mptr = (ptr);    \
    (type *)( (char *)__mptr - offsetof(type,member) );})
```

container_of - cast a member of a structure out to the containing structure @ptr: the pointer to the member. @type: the type of the container struct this is embedded in. @member: the name of the member within the struct.

Kneron Confidential

6.106.1.11 max

```
#define max(
                x,
                y )
```

Value:

```
__max(typeof(x), typeof(y),           \
          __UNIQUE_ID(max1_), __UNIQUE_ID(max2_), \
          x, y)
```

6.106.1.12 max3

```
#define max3(
                x,
                y,
                z ) max((typeof(x))max(x, y), z)
```

6.106.1.13 max_t

```
#define max_t(
                type,
                x,
                y )
```

Value:

```
__max(type, type,             \
          __UNIQUE_ID(min1_), __UNIQUE_ID(min2_), \
          x, y)
```

6.106.1.14 min

```
#define min(
                x,
                y )
```

Value:

```
__min(typeof(x), typeof(y),           \
          __UNIQUE_ID(min1_), __UNIQUE_ID(min2_), \
          x, y)
```

6.106.1.15 min3

```
#define min3(
                x,
                y,
                z ) min((typeof(x))min(x, y), z)
```

Kneron Confidential

6.106.1.16 min_not_zero

```
#define min_not_zero(
                x,
                y )
```

Value:

```
({          \
    typeof(x) __x = (x);            \
    typeof(y) __y = (y);            \
    __x == 0 ? __y : ((__y == 0) ? __x : min(__x, __y)); })
```

min_not_zero - return the minimum that is *not* zero, unless both are zero @x: value1 @y: value2

6.106.1.17 min_t

```
#define min_t(
                type,
                x,
                y )
```

Value:

```
__min(type, type,               \
          __UNIQUE_ID(min1_), __UNIQUE_ID(min2_), \
          x, y)
```

6.106.1.18 PAGE_MASK

```
#define PAGE_MASK (~(PAGE_SIZE-1))
```

6.106.1.19 PAGE_SHIFT

```
#define PAGE_SHIFT 12
```

6.106.1.20 PAGE_SIZE

```
#define PAGE_SIZE (1 << PAGE_SHIFT)
```

6.106.1.21 swap

```
#define swap(
                a,
                b ) do { typeof(a) __tmp = (a); (a) = (b); (b) = __tmp; } while (0)
```

Kneron Confidential

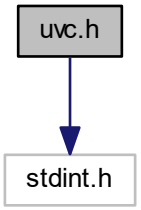
6.106.1.22 UINT_MAX

```
#define UINT_MAX (~0U)
```

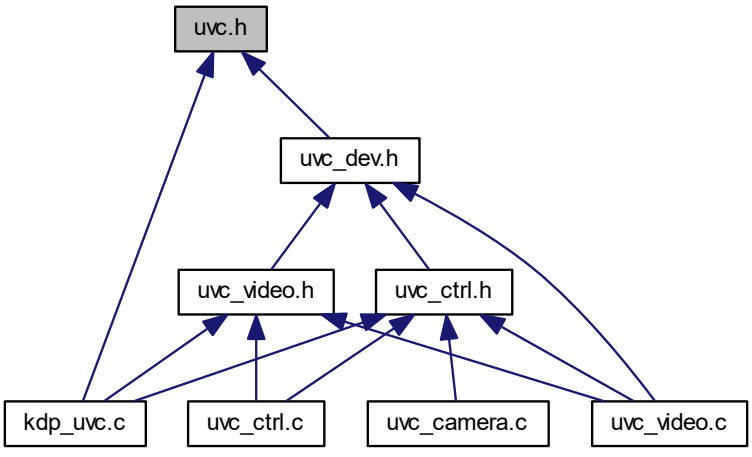
6.107 uvc.h File Reference

```
#include <stdint.h>
```

Include dependency graph for uvc.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct uvc_inf_assoc_descriptor
- struct uvc_descriptor_header
- struct uvc_vc_if_header_descriptor

- struct uvc_input_terminal_descriptor
- struct uvc_output_terminal_descriptor
- struct uvc_camera_terminal_descriptor
- struct uvc_selector_unit_descriptor
- struct uvc_processing_unit_descriptor
- struct uvc_encoding_unit_descriptor
- struct uvc_extension_unit_descriptor
- struct uvc_control_endpoint_descriptor
- struct uvc_input_header_descriptor
- struct uvc_ET_Head_descriptor
- struct uvc_output_header_descriptor
- struct uvc_still_image_frame_descriptor
- struct uvc_color_matching_descriptor
- struct uvc_streaming_control_data
- struct uvc_vs_still_control_data
- struct uvc_format_desc_head
- struct uvc_frame_desc_head
- struct uvc_format_uncompressed
- struct uvc_frame_uncompressed
- struct uvc_format_mjpeg
- struct uvc_frame_mjpeg
- struct uvc_format_frame_based

Macros

- #define UVC_CC_VIDEO 0x0E
- #define UVC_SC_UNDEFINED 0x00
- #define UVC_SC_VIDEOCONTROL 0x01
- #define UVC_SC_VIDEOSTREAMING 0x02
- #define UVC_SC_VIDEO_INTERFACE_COLLECTION 0x03
- #define UVC_PC_PROTOCOL_UNDEFINED 0x00
- #define UVC_PC_PROTOCOL_15 0x01
- #define CS_UNDEFINED 0x20
- #define CS_DEVICE 0x21
- #define CS_CONFIGURATION 0x22
- #define CS_STRING 0x23
- #define CS_INTERFACE 0x24
- #define CS_ENDPOINT 0x25
- #define UVC_VC_DESCRIPTOR_UNDEFINED 0x00
- #define UVC_VC_HEADER 0x01
- #define UVC_VC_INPUT_TERMINAL 0x02
- #define UVC_VC_OUTPUT_TERMINAL 0x03
- #define UVC_VC_SELECTOR_UNIT 0x04
- #define UVC_VC_PROCESSING_UNIT 0x05
- #define UVC_VC_EXTENSION_UNIT 0x06
- #define UVC_VC_ENCODING_UNIT 0x07
- #define UVC_VS_UNDEFINED 0x00
- #define UVC_VS_INPUT_HEADER 0x01
- #define UVC_VS_OUTPUT_HEADER 0x02
- #define UVC_VS_STILL_IMAGE_FRAME 0x03
- #define UVC_VS_FORMAT_UNCOMPRESSED 0x04
- #define UVC_VS_FRAME_UNCOMPRESSED 0x05
- #define UVC_VS_FORMAT_MJPEG 0x06
- #define UVC_VS_FRAME_MJPEG 0x07

Kneron Confidential

- #define UVC_VS_FORMAT_MPEG2TS 0x0a
- #define UVC_VS_FORMAT_DV 0x0c
- #define UVC_VS_COLORFORMAT 0x0d
- #define UVC_VS_FORMAT_FRAME_BASED 0x10
- #define UVC_VS_FRAME_FRAME_BASED 0x11
- #define UVC_VS_FORMAT_STREAM_BASED 0x12
- #define VS_FORMAT_H264 0x13
- #define VS_FRAME_H264 0x14
- #define VS_FORMAT_H264_SIMULCAST 0x15
- #define VS_FORMAT_VP8 0x16
- #define VS_FRAME_VP8 0x17
- #define VS_FORMAT_VP8_SIMULCAST 0x18
- #define UVC_EP_UNDEFINED 0x00
- #define UVC_EP_GENERAL 0x01
- #define UVC_EP_ENDPOINT 0x02
- #define UVC_EP_INTERRUPT 0x03
- #define UVC_RC_UNDEFINED 0x00
- #define UVC_SET_CUR 0x01
- #define UVC_SET_CUR_ALL 0x11
- #define UVC_GET_CUR 0x81
- #define UVC_GET_MIN 0x82
- #define UVC_GET_MAX 0x83
- #define UVC_GET_RES 0x84
- #define UVC_GET_LEN 0x85
- #define UVC_GET_INFO 0x86
- #define UVC_GET_DEF 0x87
- #define UVC_GET_CUR_ALL 0x91
- #define UVC_GET_MIN_ALL 0x92
- #define UVC_GET_MAX_ALL 0x93
- #define UVC_GET_RES_ALL 0x94
- #define UVC_GET_DEF_ALL 0x97
- #define UVC_VC_CONTROL_UNDEFINED 0x00
- #define UVC_VC_VIDEO_POWER_MODE_CONTROL 0x01
- #define UVC_VC_REQUEST_ERROR_CODE_CONTROL 0x02
- #define UVC_TE_CONTROL_UNDEFINED 0x00
- #define UVC_SU_CONTROL_UNDEFINED 0x00
- #define UVC_SU_INPUT_SELECT_CONTROL 0x01
- #define UVC_CT_CONTROL_UNDEFINED 0x00
- #define UVC_CT_SCANNING_MODE_CONTROL 0x01
- #define UVC_CT_AE_MODE_CONTROL 0x02
- #define UVC_CT_AE_PRIORITY_CONTROL 0x03
- #define UVC_CT_EXPOSURE_TIME_ABSOLUTE_CONTROL 0x04
- #define UVC_CT_EXPOSURE_TIME_RELATIVE_CONTROL 0x05
- #define UVC_CT_FOCUS_ABSOLUTE_CONTROL 0x06
- #define UVC_CT_FOCUS_RELATIVE_CONTROL 0x07
- #define UVC_CT_FOCUS_AUTO_CONTROL 0x08
- #define UVC_CT_IRIS_ABSOLUTE_CONTROL 0x09
- #define UVC_CT_IRIS_RELATIVE_CONTROL 0x0a
- #define UVC_CT_ZOOM_ABSOLUTE_CONTROL 0x0b
- #define UVC_CT_ZOOM_RELATIVE_CONTROL 0x0c
- #define UVC_CT_PANTILT_ABSOLUTE_CONTROL 0x0d
- #define UVC_CT_PANTILT_RELATIVE_CONTROL 0x0e
- #define UVC_CT_ROLL_ABSOLUTE_CONTROL 0x0f
- #define UVC_CT_ROLL_RELATIVE_CONTROL 0x10
- #define UVC_CT_PRIVACY_CONTROL 0x11

Kneron Confidential

- #define UVC_CT_FOCUS_SIMPLE_CONTROL 0x12
- #define UVC_CT_WINDOW_CONTROL 0x13
- #define UVC_CT_REGION_OF_INTEREST_CONTROL 0x14
- #define UVC_PU_CONTROL_UNDEFINED 0x00
- #define UVC_PU_BACKLIGHT_COMPENSATION_CONTROL 0x01
- #define UVC_PU_BRIGHTNESS_CONTROL 0x02
- #define UVC_PU_CONTRAST_CONTROL 0x03
- #define UVC_PU_GAIN_CONTROL 0x04
- #define UVC_PU_POWER_LINE_FREQUENCY_CONTROL 0x05
- #define UVC_PU_HUE_CONTROL 0x06
- #define UVC_PU_SATURATION_CONTROL 0x07
- #define UVC_PU_SHARPNESS_CONTROL 0x08
- #define UVC_PU_GAMMA_CONTROL 0x09
- #define UVC_PU_WHITE_BALANCE_TEMPERATURE_CONTROL 0x0a
- #define UVC_PU_WHITE_BALANCE_TEMPERATURE_AUTO_CONTROL 0x0b
- #define UVC_PU_WHITE_BALANCE_COMPONENT_CONTROL 0x0c
- #define UVC_PU_WHITE_BALANCE_COMPONENT_AUTO_CONTROL 0x0d
- #define UVC_PU_DIGITAL_MULTIPLIER_CONTROL 0x0e
- #define UVC_PU_DIGITAL_MULTIPLIER_LIMIT_CONTROL 0x0f
- #define UVC_PU_HUE_AUTO_CONTROL 0x10
- #define UVC_PU_ANALOG_VIDEO_STANDARD_CONTROL 0x11
- #define UVC_PU_ANALOG_LOCK_STATUS_CONTROL 0x12
- #define UVC_PU_CONTRAST_AUTO_CONTROL 0x13
- #define UVC_EU_CONTROL_UNDEFINED 0x00
- #define UVC_EU_SELECT_LAYER_CONTROL 0x01
- #define UVC_EU_PROFILE_TOOLSET_CONTROL 0x02
- #define UVC_EU_VIDEO_RESOLUTION_CONTROL 0x03
- #define UVC_EU_MIN_FRAME_INTERVAL_CONTROL 0x04
- #define UVC_EU_SLICE_MODE_CONTROL 0x05
- #define UVC_EU_RATE_CONTROL_MODE_CONTROL 0x06
- #define UVC_EU_AVERAGE_BITRATE_CONTROL 0x07
- #define UVC_EU_CPB_SIZE_CONTROL 0x08
- #define UVC_EU_PEAK_BIT_RATE_CONTROL 0x09
- #define UVC_EU_QUANTIZATION_PARAMS_CONTROL 0x0A
- #define UVC_EU_SYNC_REF_FRAME_CONTROL 0x0B
- #define UVC_EU_LTR_BUFFER_CONTROL 0x0C
- #define UVC_EU_LTR_PICTURE_CONTROL 0x0D
- #define UVC_EU_LTR_VALIDATION_CONTROL 0x0E
- #define UVC_EU_LEVEL_IDC_LIMIT_CONTROL 0x0F
- #define UVC_EU_SEI_PAYLOADTYPE_CONTROL 0x10
- #define UVC_EU_QP_RANGE_CONTROL 0x11
- #define UVC_EU_PRIORITY_CONTROL 0x12
- #define UVC_EU_START_OR_STOP_LAYER_CONTROL 0x13
- #define UVC_EU_ERROR_RESILIENCY_CONTROL 0x14
- #define UVC_XU_CONTROL_UNDEFINED 0x00
- #define UVC_VS_CONTROL_UNDEFINED 0x00
- #define UVC_VS_PROBE_CONTROL 0x01
- #define UVC_VS_COMMIT_CONTROL 0x02
- #define UVC_VS_STILL_PROBE_CONTROL 0x03
- #define UVC_VS_STILL_COMMIT_CONTROL 0x04
- #define UVC_VS_STILL_IMAGE_TRIGGER_CONTROL 0x05
- #define UVC_VS_STREAM_ERROR_CODE_CONTROL 0x06
- #define UVC_VS_GENERATE_KEY_FRAME_CONTROL 0x07
- #define UVC_VS_UPDATE_FRAME_SEGMENT_CONTROL 0x08
- #define UVC_VS_SYNC_DELAY_CONTROL 0x09

Kneron Confidential

- #define UVC_TT_VENDOR_SPECIFIC 0x0100
- #define UVC_TT_STREAMING 0x0101
- #define UVC_ITT_VENDOR_SPECIFIC 0x0200
- #define UVC_ITT_CAMERA 0x0201
- #define UVC_ITT_MEDIA_TRANSPORT_INPUT 0x0202
- #define UVC_OTT_VENDOR_SPECIFIC 0x0300
- #define UVC_OTT_DISPLAY 0x0301
- #define UVC_OTT_MEDIA_TRANSPORT_OUTPUT 0x0302
- #define UVC_EXTERNAL_VENDOR_SPECIFIC 0x0400
- #define UVC_COMPOSITE_CONNECTOR 0x0401
- #define UVC_SVIDEO_CONNECTOR 0x0402
- #define UVC_COMPONENT_CONNECTOR 0x0403
- #define UVC_STATUS_TYPE_CONTROL 1
- #define UVC_STATUS_TYPE_STREAMING 2
- #define UVC_STREAM_EOH (1 << 7)
- #define UVC_STREAM_ERR (1 << 6)
- #define UVC_STREAM_STI (1 << 5)
- #define UVC_STREAM_RES (1 << 4)
- #define UVC_STREAM_SCR (1 << 3)
- #define UVC_STREAM_PTS (1 << 2)
- #define UVC_STREAM_EOF (1 << 1)
- #define UVC_STREAM_FID (1 << 0)
- #define UVC_CONTROL_CAP_GET (1 << 0)
- #define UVC_CONTROL_CAP_SET (1 << 1)
- #define UVC_CONTROL_CAP_DISABLED (1 << 2)
- #define UVC_CONTROL_CAP_AUTOUPDATE (1 << 3)
- #define UVC_CONTROL_CAP_ASYNCHRONOUS (1 << 4)
- #define UVC_DT_HEADER_SIZE(n) (12+(n))
- #define UVC_HEADER_DESCRIPTOR(n) uvc_header_descriptor_##n
- #define DECLARE_UVC_HEADER_DESCRIPTOR(n)
- #define UVC_DT_INPUT_TERMINAL_SIZE 8
- #define UVC_DT_OUTPUT_TERMINAL_SIZE 9
- #define UVC_DT_CT_CONST_LEN 15
- #define UVC_DT_CAMERA_TERMINAL_SIZE(n) (15+(n))
- #define UVC_SU_CONST_LEN 5
- #define UVC_DT_SELECTOR_UNIT_SIZE(n) (6+(n))
- #define UVC_SELECTOR_UNIT_DESCRIPTOR(n) uvc_selector_unit_descriptor_##n
- #define DECLARE_UVC_SELECTOR_UNIT_DESCRIPTOR(n)
- #define UVC_PU_CONST_LEN 8
- #define UVC_DT_PROCESSING_UNIT_SIZE(n) (9+(n))
- #define UVC_EU_CONST_LEN 7
- #define UVC_XU_CONST_LEN 22
- #define UVC_DT_EXTENSION_UNIT_SIZE(p, n) (24+(p)+(n))
- #define UVC_EXTENSION_UNIT_DESCRIPTOR(p, n) uvc_extension_unit_descriptor_##p_##n
- #define DECLARE_UVC_EXTENSION_UNIT_DESCRIPTOR(p, n)
- #define UVC_DT_CONTROL_ENDPOINT_SIZE 5
- #define UVC_DT_INPUT_HEADER_SIZE(n, p) (13+(n*p))
- #define UVC_INPUT_HEADER_DESCRIPTOR(n, p) uvc_input_header_descriptor_##n_##p
- #define DECLARE_UVC_INPUT_HEADER_DESCRIPTOR(n, p)
- #define UVC_DT_OUTPUT_HEADER_SIZE(n, p) (9+(n*p))
- #define UVC_OUTPUT_HEADER_DESCRIPTOR(n, p) uvc_output_header_descriptor_##n_##p
- #define DECLARE_UVC_OUTPUT_HEADER_DESCRIPTOR(n, p)
- #define UVC_STILL_IMAGE_FRAME_DESCRIPTOR_SIZE(l, n) (6+(4*l + 2*n))
- #define UVC_STILL_IMAGE_FRAME_DESCRIPTOR(l, n) uvc_still_image_frame_descriptor_##n_##p
- #define DECLARE_UVC_STILL_IMAGE_FRAME_DESCRIPTOR(l, n)

Kneron Confidential

- #define UVC_DT_COLOR_MATCHING_SIZE 6
- #define UVC_DT_FORMAT_UNCOMPRESSED_SIZE 27
- #define UVC_DT_FRAME_UNCOMPRESSED_SIZE(n) (26+4∗(n))
- #define UVC_FRAME_UNCOMPRESSED(n) uvc_frame_uncompressed_##n
- #define DECLARE_UVC_FRAME_UNCOMPRESSED(n)
- #define UVC_DT_FORMAT_MJPEG_SIZE 11
- #define UVC_DT_FRAME_MJPEG_SIZE(n) (26+4∗(n))
- #define UVC_FRAME_MJPEG(n) uvc_frame_mjpeg_##n
- #define DECLARE_UVC_FRAME_MJPEG(n)

Functions

- struct uvc_inf_assoc_descriptor __attribute__ ((__packed__))

  8-byte setup packet struct

Variables

- uint8_t bLength
- uint8_t bDescriptorType
- uint8_t bFirstInterface
- uint8_t bInterfaceCount
- uint8_t bFunctionClass
- uint8_t bFunctionSubClass
- uint8_t bFunctionProtocol
- uint8_t iFunction
- uint8_t bDescriptorSubType
- uint16_t bcdUVC
- uint16_t wTotalLength
- uint32_t dwClockFrequency
- uint8_t bInCollection
- uint8_t baInterfaceNr [ ]
- uint8_t bTerminalID
- uint16_t wTerminalType
- uint8_t bAssocTerminal
- uint8_t iTerminal
- uint8_t bSourceID
- uint16_t wObjectiveFocalLengthMin
- uint16_t wObjectiveFocalLengthMax
- uint16_t wOcularFocalLength
- uint8_t bControlSize
- uint8_t bmControls [0]
- uint8_t bUnitID
- uint8_t bNrInPins
- uint8_t baSourceID [0]
- uint8_t iSelector
- uint16_t wMaxMultiplier
- uint8_t iProcessing
- uint8_t bmVideoStandards
- uint8_t iEncoding
- uint8_t bmControlsRuntime [0]
- uint8_t guidExtensionCode [16]
- uint8_t bNumControls

Kneron Confidential

- uint8_t iExtension
- uint16_t wMaxTransferSize
- uint8_t bNumFormats
- uint8_t bEndpointAddress
- uint8_t bmInfo
- uint8_t bTerminalLink
- uint8_t bStillCaptureMethod
- uint8_t bTriggerSupport
- uint8_t bTriggerUsage
- uint8_t bmaControls [ ]
- uint8_t bNumImageSizePatterns
- uint16_t wWidth [0]
- uint16_t wHeight [0]
- uint8_t bNumCompressionPattern
- uint8_t bCompression [ ]
- uint8_t bColorPrimaries
- uint8_t bTransferCharacteristics
- uint8_t bMatrixCoefficients
- uint16_t wmHint
- uint8_t bFormatIndex
- uint8_t bFrameIndex
- uint32_t dwFrameInterval
- uint16_t wKeyFrameRate
- uint16_t wPFrameRate
- uint16_t wCompQuality
- uint16_t wCompWindowSize
- uint16_t wDelay
- uint32_t dwMaxVideoFrameSize
- uint32_t dwMaxPayloadTransferSize
- uint8_t bmFramingInfo
- uint8_t bPreferedVersion
- uint8_t bMinVersion
- uint8_t bMaxVersion
- uint8_t bUsage
- uint8_t bBitDrpthLuma
- uint8_t bmSetting
- uint8_t bMaxNumberOfRefFeamesPlus1
- uint16_t wRateControlModes
- uint8_t bmLayoutPerStream [8]
- uint8_t bCompressionIndex
- uint8_t bDescriptorSubtype
- uint8_t bNumFrameDescriptors
- uint8_t guidFormat [16]
- uint8_t bBitsPerPixel
- uint8_t bDefaultFrameIndex
- uint8_t bAspectRatioX
- uint8_t bAspectRatioY
- uint8_t bmInterfaceFlags
- uint8_t bCopyProtect
- uint8_t bmCapabilities
- uint32_t dwMinBitRate
- uint32_t dwMaxBitRate
- uint32_t dwMaxVideoFrameBufferSize
- uint32_t dwDefaultFrameInterval
- uint8_t bFrameIntervalType
- uint8_t bmFlags
- uint8_t bVariableSize

Kneron Confidential

6.107.1 Macro Definition Documentation

6.107.1.1 CS_CONFIGURATION

```
#define CS_CONFIGURATION 0x22
```

6.107.1.2 CS_DEVICE

```
#define CS_DEVICE 0x21
```

6.107.1.3 CS_ENDPOINT

```
#define CS_ENDPOINT 0x25
```

6.107.1.4 CS_INTERFACE

```
#define CS_INTERFACE 0x24
```

6.107.1.5 CS_STRING

```
#define CS_STRING 0x23
```

6.107.1.6 CS_UNDEFINED

```
#define CS_UNDEFINED 0x20
```

Kneron Confidential

6.107.1.7 DECLARE_UVC_EXTENSION_UNIT_DESCRIPTOR

```
#define DECLARE_UVC_EXTENSION_UNIT_DESCRIPTOR(
                p,
                n )
```

Value:

```
struct UVC_EXTENSION_UNIT_DESCRIPTOR(p, n) {        \
    uint8_t  bLength;                    \
    uint8_t  bDescriptorType;               \
    uint8_t  bDescriptorSubType;            \
    uint8_t  bUnitID;                    \
    uint8_t  guidExtensionCode[16];            \
    uint8_t  bNumControls;                \
    uint8_t  bNrInPins;                \
    uint8_t  baSourceID[p];                \
    uint8_t  bControlSize;                \
    uint8_t  bmControls[n];                \
    uint8_t  iExtension;                \
} __attribute__ ((packed))
```

6.107.1.8 DECLARE_UVC_FRAME_MJPEG

```
#define DECLARE_UVC_FRAME_MJPEG(
                n )
```

Value:

```
struct UVC_FRAME_MJPEG(n) {                \
    uint8_t  bLength;                \
    uint8_t  bDescriptorType;               \
    uint8_t  bDescriptorSubType;            \
    uint8_t  bFrameIndex;                \
    uint8_t  bmCapabilities;                \
    uint16_t wWidth;                \
    uint16_t wHeight;                \
    uint32_t dwMinBitRate;                \
    uint32_t dwMaxBitRate;                \
    uint32_t dwMaxVideoFrameBufferSize;            \
    uint32_t dwDefaultFrameInterval;            \
    uint8_t  bFrameIntervalType;               \
    uint32_t dwFrameInterval[n];               \
} __attribute__ ((packed))
```

6.107.1.9 DECLARE_UVC_FRAME_UNCOMPRESSED

```
#define DECLARE_UVC_FRAME_UNCOMPRESSED(
                n )
```

Value:

```
struct UVC_FRAME_UNCOMPRESSED(n) {            \
    uint8_t  bLength;                \
    uint8_t  bDescriptorType;               \
    uint8_t  bDescriptorSubType;            \
    uint8_t  bFrameIndex;                \
    uint8_t  bmCapabilities;                \
    uint16_t wWidth;                \
    uint16_t wHeight;                \
    uint32_t dwMinBitRate;                \
    uint32_t dwMaxBitRate;                \
    uint32_t dwMaxVideoFrameBufferSize;            \
    uint32_t dwDefaultFrameInterval;            \
    uint8_t  bFrameIntervalType;               \
    uint32_t dwFrameInterval[n];               \
} __attribute__ ((__packed__))
```

Kneron Confidential

6.107.1.10 DECLARE_UVC_HEADER_DESCRIPTOR

```
#define DECLARE_UVC_HEADER_DESCRIPTOR(
                n )
```

Value:

```
struct UVC_HEADER_DESCRIPTOR(n) {            \
    uint8_t  bLength;                   \
    uint8_t  bDescriptorType;              \
    uint8_t  bDescriptorSubType;           \
    uint16_t bcdUVC;                    \
    uint16_t wTotalLength;                 \
    uint32_t dwClockFrequency;             \
    uint8_t  bInCollection;                \
    uint8_t  baInterfaceNr[n];             \
} __attribute__ ((packed))
```

6.107.1.11 DECLARE_UVC_INPUT_HEADER_DESCRIPTOR

```
#define DECLARE_UVC_INPUT_HEADER_DESCRIPTOR(
                n,
                p )
```

Value:

```
struct UVC_INPUT_HEADER_DESCRIPTOR(n, p) {        \
    uint8_t  bLength;                   \
    uint8_t  bDescriptorType;              \
    uint8_t  bDescriptorSubType;           \
    uint8_t  bNumFormats;               \
    uint16_t wTotalLength;                 \
    uint8_t  bEndpointAddress;             \
    uint8_t  bmInfo;                    \
    uint8_t  bTerminalLink;                \
    uint8_t  bStillCaptureMethod;          \
    uint8_t  bTriggerSupport;              \
    uint8_t  bTriggerUsage;                \
    uint8_t  bControlSize;                 \
    uint8_t  bmaControls[p][n];            \
} __attribute__ ((packed))
```

6.107.1.12 DECLARE_UVC_OUTPUT_HEADER_DESCRIPTOR

```
#define DECLARE_UVC_OUTPUT_HEADER_DESCRIPTOR(
                n,
                p )
```

Value:

```
struct UVC_OUTPUT_HEADER_DESCRIPTOR(n, p) {        \
    uint8_t  bLength;                   \
    uint8_t  bDescriptorType;              \
    uint8_t  bDescriptorSubType;           \
    uint8_t  bNumFormats;               \
    uint16_t wTotalLength;                 \
    uint8_t  bEndpointAddress;             \
    uint8_t  bTerminalLink;                \
    uint8_t  bControlSize;                 \
    uint8_t  bmaControls[p][n];            \
} __attribute__ ((packed))
```

Kneron Confidential

6.107.1.13 DECLARE_UVC_SELECTOR_UNIT_DESCRIPTOR

```
#define DECLARE_UVC_SELECTOR_UNIT_DESCRIPTOR(
              n )
```

Value:

```
struct UVC_SELECTOR_UNIT_DESCRIPTOR(n) {        \
    uint8_t  bLength;                    \
    uint8_t  bDescriptorType;               \
    uint8_t  bDescriptorSubType;             \
    uint8_t  bUnitID;                    \
    uint8_t  bNrInPins;                  \
    uint8_t  baSourceID[n];                 \
    uint8_t  iSelector;                  \
} __attribute__ ((packed))
```

6.107.1.14 DECLARE_UVC_STILL_IMAGE_FRAME_DESCRIPTOR

```
#define DECLARE_UVC_STILL_IMAGE_FRAME_DESCRIPTOR(
              l,
              n )
```

Value:

```
struct UVC_STILL_IMAGE_FRAME_DESCRIPTOR(l, n) {        \
    uint8_t  bLength;                   \
    uint8_t  bDescriptorType;           \
    uint8_t  bDescriptorSubType;        \
    uint8_t  bEndpointAddress;          \
    uint8_t  bNumImageSizePatterns;     \
    uint16_t wWidth[l];                  \
    uint16_t wHeight[n];                 \
    uint8_t bNumCompressionPattern;     \
    uint8_t bCompression[l];             \
} __attribute__ ((packed))
```

6.107.1.15 UVC_CC_VIDEO

```
#define UVC_CC_VIDEO 0x0E
```

6.107.1.16 UVC_COMPONENT_CONNECTOR

```
#define UVC_COMPONENT_CONNECTOR 0x0403
```

6.107.1.17 UVC_COMPOSITE_CONNECTOR

```
#define UVC_COMPOSITE_CONNECTOR 0x0401
```

Kneron Confidential

6.107.1.18 UVC_CONTROL_CAP_ASYNCHRONOUS

```
#define UVC_CONTROL_CAP_ASYNCHRONOUS (1 << 4)
```

6.107.1.19 UVC_CONTROL_CAP_AUTOUPDATE

```
#define UVC_CONTROL_CAP_AUTOUPDATE (1 << 3)
```

6.107.1.20 UVC_CONTROL_CAP_DISABLED

```
#define UVC_CONTROL_CAP_DISABLED (1 << 2)
```

6.107.1.21 UVC_CONTROL_CAP_GET

```
#define UVC_CONTROL_CAP_GET (1 << 0)
```

6.107.1.22 UVC_CONTROL_CAP_SET

```
#define UVC_CONTROL_CAP_SET (1 << 1)
```

6.107.1.23 UVC_CT_AE_MODE_CONTROL

```
#define UVC_CT_AE_MODE_CONTROL 0x02
```

6.107.1.24 UVC_CT_AE_PRIORITY_CONTROL

```
#define UVC_CT_AE_PRIORITY_CONTROL 0x03
```

6.107.1.25 UVC_CT_CONTROL_UNDEFINED

```
#define UVC_CT_CONTROL_UNDEFINED 0x00
```

Kneron Confidential

6.107.1.26 UVC_CT_EXPOSURE_TIME_ABSOLUTE_CONTROL

```
#define UVC_CT_EXPOSURE_TIME_ABSOLUTE_CONTROL 0x04
```

6.107.1.27 UVC_CT_EXPOSURE_TIME_RELATIVE_CONTROL

```
#define UVC_CT_EXPOSURE_TIME_RELATIVE_CONTROL 0x05
```

6.107.1.28 UVC_CT_FOCUS_ABSOLUTE_CONTROL

```
#define UVC_CT_FOCUS_ABSOLUTE_CONTROL 0x06
```

6.107.1.29 UVC_CT_FOCUS_AUTO_CONTROL

```
#define UVC_CT_FOCUS_AUTO_CONTROL 0x08
```

6.107.1.30 UVC_CT_FOCUS_RELATIVE_CONTROL

```
#define UVC_CT_FOCUS_RELATIVE_CONTROL 0x07
```

6.107.1.31 UVC_CT_FOCUS_SIMPLE_CONTROL

```
#define UVC_CT_FOCUS_SIMPLE_CONTROL 0x12
```

6.107.1.32 UVC_CT_IRIS_ABSOLUTE_CONTROL

```
#define UVC_CT_IRIS_ABSOLUTE_CONTROL 0x09
```

6.107.1.33 UVC_CT_IRIS_RELATIVE_CONTROL

```
#define UVC_CT_IRIS_RELATIVE_CONTROL 0x0a
```

Kneron Confidential

6.107.1.34 UVC_CT_PANTILT_ABSOLUTE_CONTROL

```
#define UVC_CT_PANTILT_ABSOLUTE_CONTROL 0x0d
```

6.107.1.35 UVC_CT_PANTILT_RELATIVE_CONTROL

```
#define UVC_CT_PANTILT_RELATIVE_CONTROL 0x0e
```

6.107.1.36 UVC_CT_PRIVACY_CONTROL

```
#define UVC_CT_PRIVACY_CONTROL 0x11
```

6.107.1.37 UVC_CT_REGION_OF_INTEREST_CONTROL

```
#define UVC_CT_REGION_OF_INTEREST_CONTROL 0x14
```

6.107.1.38 UVC_CT_ROLL_ABSOLUTE_CONTROL

```
#define UVC_CT_ROLL_ABSOLUTE_CONTROL 0x0f
```

6.107.1.39 UVC_CT_ROLL_RELATIVE_CONTROL

```
#define UVC_CT_ROLL_RELATIVE_CONTROL 0x10
```

6.107.1.40 UVC_CT_SCANNING_MODE_CONTROL

```
#define UVC_CT_SCANNING_MODE_CONTROL 0x01
```

6.107.1.41 UVC_CT_WINDOW_CONTROL

```
#define UVC_CT_WINDOW_CONTROL 0x13
```

Kneron Confidential

6.107.1.42 UVC_CT_ZOOM_ABSOLUTE_CONTROL

```
#define UVC_CT_ZOOM_ABSOLUTE_CONTROL 0x0b
```

6.107.1.43 UVC_CT_ZOOM_RELATIVE_CONTROL

```
#define UVC_CT_ZOOM_RELATIVE_CONTROL 0x0c
```

6.107.1.44 UVC_DT_CAMERA_TERMINAL_SIZE

```
#define UVC_DT_CAMERA_TERMINAL_SIZE(
              n ) (15+(n))
```

6.107.1.45 UVC_DT_COLOR_MATCHING_SIZE

```
#define UVC_DT_COLOR_MATCHING_SIZE 6
```

6.107.1.46 UVC_DT_CONTROL_ENDPOINT_SIZE

```
#define UVC_DT_CONTROL_ENDPOINT_SIZE 5
```

6.107.1.47 UVC_DT_CT_CONST_LEN

```
#define UVC_DT_CT_CONST_LEN 15
```

6.107.1.48 UVC_DT_EXTENSION_UNIT_SIZE

```
#define UVC_DT_EXTENSION_UNIT_SIZE(
              p,
              n ) (24+(p)+(n))
```

Kneron Confidential

6.107.1.49 UVC_DT_FORMAT_MJPEG_SIZE

```
#define UVC_DT_FORMAT_MJPEG_SIZE 11
```

6.107.1.50 UVC_DT_FORMAT_UNCOMPRESSED_SIZE

```
#define UVC_DT_FORMAT_UNCOMPRESSED_SIZE 27
```

6.107.1.51 UVC_DT_FRAME_MJPEG_SIZE

```
#define UVC_DT_FRAME_MJPEG_SIZE(
              n ) (26+4*(n))
```

6.107.1.52 UVC_DT_FRAME_UNCOMPRESSED_SIZE

```
#define UVC_DT_FRAME_UNCOMPRESSED_SIZE(
              n ) (26+4*(n))
```

6.107.1.53 UVC_DT_HEADER_SIZE

```
#define UVC_DT_HEADER_SIZE(
              n ) (12+(n))
```

6.107.1.54 UVC_DT_INPUT_HEADER_SIZE

```
#define UVC_DT_INPUT_HEADER_SIZE(
              n,
              p ) (13+(n*p))
```

6.107.1.55 UVC_DT_INPUT_TERMINAL_SIZE

```
#define UVC_DT_INPUT_TERMINAL_SIZE 8
```

Kneron Confidential

6.107.1.56 UVC_DT_OUTPUT_HEADER_SIZE

```
#define UVC_DT_OUTPUT_HEADER_SIZE(
              n,
              p ) (9+(n*p))
```

6.107.1.57 UVC_DT_OUTPUT_TERMINAL_SIZE

```
#define UVC_DT_OUTPUT_TERMINAL_SIZE 9
```

6.107.1.58 UVC_DT_PROCESSING_UNIT_SIZE

```
#define UVC_DT_PROCESSING_UNIT_SIZE(
              n ) (9+(n))
```

6.107.1.59 UVC_DT_SELECTOR_UNIT_SIZE

```
#define UVC_DT_SELECTOR_UNIT_SIZE(
              n ) (6+(n))
```

6.107.1.60 UVC_EP_ENDPOINT

```
#define UVC_EP_ENDPOINT 0x02
```

6.107.1.61 UVC_EP_GENERAL

```
#define UVC_EP_GENERAL 0x01
```

6.107.1.62 UVC_EP_INTERRUPT

```
#define UVC_EP_INTERRUPT 0x03
```

Kneron Confidential

6.107.1.63 UVC_EP_UNDEFINED

```
#define UVC_EP_UNDEFINED 0x00
```

6.107.1.64 UVC_EU_AVERAGE_BITRATE_CONTROL

```
#define UVC_EU_AVERAGE_BITRATE_CONTROL 0x07
```

6.107.1.65 UVC_EU_CONST_LEN

```
#define UVC_EU_CONST_LEN 7
```

6.107.1.66 UVC_EU_CONTROL_UNDEFINED

```
#define UVC_EU_CONTROL_UNDEFINED 0x00
```

6.107.1.67 UVC_EU_CPB_SIZE_CONTROL

```
#define UVC_EU_CPB_SIZE_CONTROL 0x08
```

6.107.1.68 UVC_EU_ERROR_RESILIENCY_CONTROL

```
#define UVC_EU_ERROR_RESILIENCY_CONTROL 0x14
```

6.107.1.69 UVC_EU_LEVEL_IDC_LIMIT_CONTROL

```
#define UVC_EU_LEVEL_IDC_LIMIT_CONTROL 0x0F
```

6.107.1.70 UVC_EU_LTR_BUFFER_CONTROL

```
#define UVC_EU_LTR_BUFFER_CONTROL 0x0C
```

Kneron Confidential

6.107.1.71 UVC_EU_LTR_PICTURE_CONTROL

```
#define UVC_EU_LTR_PICTURE_CONTROL 0x0D
```

6.107.1.72 UVC_EU_LTR_VALIDATION_CONTROL

```
#define UVC_EU_LTR_VALIDATION_CONTROL 0x0E
```

6.107.1.73 UVC_EU_MIN_FRAME_INTERVAL_CONTROL

```
#define UVC_EU_MIN_FRAME_INTERVAL_CONTROL 0x04
```

6.107.1.74 UVC_EU_PEAK_BIT_RATE_CONTROL

```
#define UVC_EU_PEAK_BIT_RATE_CONTROL 0x09
```

6.107.1.75 UVC_EU_PRIORITY_CONTROL

```
#define UVC_EU_PRIORITY_CONTROL 0x12
```

6.107.1.76 UVC_EU_PROFILE_TOOLSET_CONTROL

```
#define UVC_EU_PROFILE_TOOLSET_CONTROL 0x02
```

6.107.1.77 UVC_EU_QP_RANGE_CONTROL

```
#define UVC_EU_QP_RANGE_CONTROL 0x11
```

6.107.1.78 UVC_EU_QUANTIZATION_PARAMS_CONTROL

```
#define UVC_EU_QUANTIZATION_PARAMS_CONTROL 0x0A
```

Kneron Confidential

6.107.1.79 UVC_EU_RATE_CONTROL_MODE_CONTROL

```
#define UVC_EU_RATE_CONTROL_MODE_CONTROL 0x06
```

6.107.1.80 UVC_EU_SEI_PAYLOADTYPE_CONTROL

```
#define UVC_EU_SEI_PAYLOADTYPE_CONTROL 0x10
```

6.107.1.81 UVC_EU_SELECT_LAYER_CONTROL

```
#define UVC_EU_SELECT_LAYER_CONTROL 0x01
```

6.107.1.82 UVC_EU_SLICE_MODE_CONTROL

```
#define UVC_EU_SLICE_MODE_CONTROL 0x05
```

6.107.1.83 UVC_EU_START_OR_STOP_LAYER_CONTROL

```
#define UVC_EU_START_OR_STOP_LAYER_CONTROL 0x13
```

6.107.1.84 UVC_EU_SYNC_REF_FRAME_CONTROL

```
#define UVC_EU_SYNC_REF_FRAME_CONTROL 0x0B
```

6.107.1.85 UVC_EU_VIDEO_RESOLUTION_CONTROL

```
#define UVC_EU_VIDEO_RESOLUTION_CONTROL 0x03
```

Kneron Confidential

6.107.1.86 UVC_EXTENSION_UNIT_DESCRIPTOR

```
#define UVC_EXTENSION_UNIT_DESCRIPTOR(
              p,
              n ) uvc_extension_unit_descriptor_##p_##n
```

6.107.1.87 UVC_EXTERNAL_VENDOR_SPECIFIC

```
#define UVC_EXTERNAL_VENDOR_SPECIFIC 0x0400
```

6.107.1.88 UVC_FRAME_MJPEG

```
#define UVC_FRAME_MJPEG(
              n ) uvc_frame_mjpeg_##n
```

6.107.1.89 UVC_FRAME_UNCOMPRESSED

```
#define UVC_FRAME_UNCOMPRESSED(
              n ) uvc_frame_uncompressed_##n
```

6.107.1.90 UVC_GET_CUR

```
#define UVC_GET_CUR 0x81
```

6.107.1.91 UVC_GET_CUR_ALL

```
#define UVC_GET_CUR_ALL 0x91
```

6.107.1.92 UVC_GET_DEF

```
#define UVC_GET_DEF 0x87
```

Kneron Confidential

6.107.1.93 UVC_GET_DEF_ALL

```
#define UVC_GET_DEF_ALL 0x97
```

6.107.1.94 UVC_GET_INFO

```
#define UVC_GET_INFO 0x86
```

6.107.1.95 UVC_GET_LEN

```
#define UVC_GET_LEN 0x85
```

6.107.1.96 UVC_GET_MAX

```
#define UVC_GET_MAX 0x83
```

6.107.1.97 UVC_GET_MAX_ALL

```
#define UVC_GET_MAX_ALL 0x93
```

6.107.1.98 UVC_GET_MIN

```
#define UVC_GET_MIN 0x82
```

6.107.1.99 UVC_GET_MIN_ALL

```
#define UVC_GET_MIN_ALL 0x92
```

6.107.1.100 UVC_GET_RES

```
#define UVC_GET_RES 0x84
```

Kneron Confidential

6.107.1.101 UVC_GET_RES_ALL

```
#define UVC_GET_RES_ALL 0x94
```

6.107.1.102 UVC_HEADER_DESCRIPTOR

```
#define UVC_HEADER_DESCRIPTOR(
             n ) uvc_header_descriptor_##n
```

6.107.1.103 UVC_INPUT_HEADER_DESCRIPTOR

```
#define UVC_INPUT_HEADER_DESCRIPTOR(
             n,
             p ) uvc_input_header_descriptor_##n_##p
```

6.107.1.104 UVC_ITT_CAMERA

```
#define UVC_ITT_CAMERA 0x0201
```

6.107.1.105 UVC_ITT_MEDIA_TRANSPORT_INPUT

```
#define UVC_ITT_MEDIA_TRANSPORT_INPUT 0x0202
```

6.107.1.106 UVC_ITT_VENDOR_SPECIFIC

```
#define UVC_ITT_VENDOR_SPECIFIC 0x0200
```

6.107.1.107 UVC_OTT_DISPLAY

```
#define UVC_OTT_DISPLAY 0x0301
```

Kneron Confidential

6.107.1.108 UVC_OTT_MEDIA_TRANSPORT_OUTPUT

```
#define UVC_OTT_MEDIA_TRANSPORT_OUTPUT 0x0302
```

6.107.1.109 UVC_OTT_VENDOR_SPECIFIC

```
#define UVC_OTT_VENDOR_SPECIFIC 0x0300
```

6.107.1.110 UVC_OUTPUT_HEADER_DESCRIPTOR

```
#define UVC_OUTPUT_HEADER_DESCRIPTOR(
             n,
             p ) uvc_output_header_descriptor_##n_##p
```

6.107.1.111 UVC_PC_PROTOCOL_15

```
#define UVC_PC_PROTOCOL_15 0x01
```

6.107.1.112 UVC_PC_PROTOCOL_UNDEFINED

```
#define UVC_PC_PROTOCOL_UNDEFINED 0x00
```

6.107.1.113 UVC_PU_ANALOG_LOCK_STATUS_CONTROL

```
#define UVC_PU_ANALOG_LOCK_STATUS_CONTROL 0x12
```

6.107.1.114 UVC_PU_ANALOG_VIDEO_STANDARD_CONTROL

```
#define UVC_PU_ANALOG_VIDEO_STANDARD_CONTROL 0x11
```

Kneron Confidential

6.107.1.115 UVC_PU_BACKLIGHT_COMPENSATION_CONTROL

```
#define UVC_PU_BACKLIGHT_COMPENSATION_CONTROL 0x01
```

6.107.1.116 UVC_PU_BRIGHTNESS_CONTROL

```
#define UVC_PU_BRIGHTNESS_CONTROL 0x02
```

6.107.1.117 UVC_PU_CONST_LEN

```
#define UVC_PU_CONST_LEN 8
```

6.107.1.118 UVC_PU_CONTRAST_AUTO_CONTROL

```
#define UVC_PU_CONTRAST_AUTO_CONTROL 0x13
```

6.107.1.119 UVC_PU_CONTRAST_CONTROL

```
#define UVC_PU_CONTRAST_CONTROL 0x03
```

6.107.1.120 UVC_PU_CONTROL_UNDEFINED

```
#define UVC_PU_CONTROL_UNDEFINED 0x00
```

6.107.1.121 UVC_PU_DIGITAL_MULTIPLIER_CONTROL

```
#define UVC_PU_DIGITAL_MULTIPLIER_CONTROL 0x0e
```

6.107.1.122 UVC_PU_DIGITAL_MULTIPLIER_LIMIT_CONTROL

```
#define UVC_PU_DIGITAL_MULTIPLIER_LIMIT_CONTROL 0x0f
```

Kneron Confidential

6.107.1.123 UVC_PU_GAIN_CONTROL

```
#define UVC_PU_GAIN_CONTROL 0x04
```

6.107.1.124 UVC_PU_GAMMA_CONTROL

```
#define UVC_PU_GAMMA_CONTROL 0x09
```

6.107.1.125 UVC_PU_HUE_AUTO_CONTROL

```
#define UVC_PU_HUE_AUTO_CONTROL 0x10
```

6.107.1.126 UVC_PU_HUE_CONTROL

```
#define UVC_PU_HUE_CONTROL 0x06
```

6.107.1.127 UVC_PU_POWER_LINE_FREQUENCY_CONTROL

```
#define UVC_PU_POWER_LINE_FREQUENCY_CONTROL 0x05
```

6.107.1.128 UVC_PU_SATURATION_CONTROL

```
#define UVC_PU_SATURATION_CONTROL 0x07
```

6.107.1.129 UVC_PU_SHARPNESS_CONTROL

```
#define UVC_PU_SHARPNESS_CONTROL 0x08
```

6.107.1.130 UVC_PU_WHITE_BALANCE_COMPONENT_AUTO_CONTROL

```
#define UVC_PU_WHITE_BALANCE_COMPONENT_AUTO_CONTROL 0x0d
```

Kneron Confidential

6.107.1.131 UVC_PU_WHITE_BALANCE_COMPONENT_CONTROL

```
#define UVC_PU_WHITE_BALANCE_COMPONENT_CONTROL 0x0c
```

6.107.1.132 UVC_PU_WHITE_BALANCE_TEMPERATURE_AUTO_CONTROL

```
#define UVC_PU_WHITE_BALANCE_TEMPERATURE_AUTO_CONTROL 0x0b
```

6.107.1.133 UVC_PU_WHITE_BALANCE_TEMPERATURE_CONTROL

```
#define UVC_PU_WHITE_BALANCE_TEMPERATURE_CONTROL 0x0a
```

6.107.1.134 UVC_RC_UNDEFINED

```
#define UVC_RC_UNDEFINED 0x00
```

6.107.1.135 UVC_SC_UNDEFINED

```
#define UVC_SC_UNDEFINED 0x00
```

6.107.1.136 UVC_SC_VIDEO_INTERFACE_COLLECTION

```
#define UVC_SC_VIDEO_INTERFACE_COLLECTION 0x03
```

6.107.1.137 UVC_SC_VIDEOCONTROL

```
#define UVC_SC_VIDEOCONTROL 0x01
```

6.107.1.138 UVC_SC_VIDEOSTREAMING

```
#define UVC_SC_VIDEOSTREAMING 0x02
```

Kneron Confidential

6.107.1.139 UVC_SELECTOR_UNIT_DESCRIPTOR

```
#define UVC_SELECTOR_UNIT_DESCRIPTOR(
              n ) uvc_selector_unit_descriptor_##n
```

6.107.1.140 UVC_SET_CUR

```
#define UVC_SET_CUR 0x01
```

6.107.1.141 UVC_SET_CUR_ALL

```
#define UVC_SET_CUR_ALL 0x11
```

6.107.1.142 UVC_STATUS_TYPE_CONTROL

```
#define UVC_STATUS_TYPE_CONTROL 1
```

6.107.1.143 UVC_STATUS_TYPE_STREAMING

```
#define UVC_STATUS_TYPE_STREAMING 2
```

6.107.1.144 UVC_STILL_IMAGE_FRAME_DESCRIPTOR

```
#define UVC_STILL_IMAGE_FRAME_DESCRIPTOR(
              l,
              n ) uvc_still_image_frame_descriptor_##n_##p
```

6.107.1.145 UVC_STILL_IMAGE_FRAME_DESCRIPTOR_SIZE

```
#define UVC_STILL_IMAGE_FRAME_DESCRIPTOR_SIZE(
              l,
              n ) (6+(4*l + 2*n))
```

Kneron Confidential

6.107.1.146 UVC_STREAM_EOF

```
#define UVC_STREAM_EOF (1 << 1)
```

6.107.1.147 UVC_STREAM_EOH

```
#define UVC_STREAM_EOH (1 << 7)
```

6.107.1.148 UVC_STREAM_ERR

```
#define UVC_STREAM_ERR (1 << 6)
```

6.107.1.149 UVC_STREAM_FID

```
#define UVC_STREAM_FID (1 << 0)
```

6.107.1.150 UVC_STREAM_PTS

```
#define UVC_STREAM_PTS (1 << 2)
```

6.107.1.151 UVC_STREAM_RES

```
#define UVC_STREAM_RES (1 << 4)
```

6.107.1.152 UVC_STREAM_SCR

```
#define UVC_STREAM_SCR (1 << 3)
```

6.107.1.153 UVC_STREAM_STI

```
#define UVC_STREAM_STI (1 << 5)
```

Kneron Confidential

6.107.1.154 UVC_SU_CONST_LEN

```
#define UVC_SU_CONST_LEN 5
```

6.107.1.155 UVC_SU_CONTROL_UNDEFINED

```
#define UVC_SU_CONTROL_UNDEFINED 0x00
```

6.107.1.156 UVC_SU_INPUT_SELECT_CONTROL

```
#define UVC_SU_INPUT_SELECT_CONTROL 0x01
```

6.107.1.157 UVC_SVIDEO_CONNECTOR

```
#define UVC_SVIDEO_CONNECTOR 0x0402
```

6.107.1.158 UVC_TE_CONTROL_UNDEFINED

```
#define UVC_TE_CONTROL_UNDEFINED 0x00
```

6.107.1.159 UVC_TT_STREAMING

```
#define UVC_TT_STREAMING 0x0101
```

6.107.1.160 UVC_TT_VENDOR_SPECIFIC

```
#define UVC_TT_VENDOR_SPECIFIC 0x0100
```

6.107.1.161 UVC_VC_CONTROL_UNDEFINED

```
#define UVC_VC_CONTROL_UNDEFINED 0x00
```

Kneron Confidential

6.107.1.162 UVC_VC_DESCRIPTOR_UNDEFINED

```
#define UVC_VC_DESCRIPTOR_UNDEFINED 0x00
```

6.107.1.163 UVC_VC_ENCODING_UNIT

```
#define UVC_VC_ENCODING_UNIT 0x07
```

6.107.1.164 UVC_VC_EXTENSION_UNIT

```
#define UVC_VC_EXTENSION_UNIT 0x06
```

6.107.1.165 UVC_VC_HEADER

```
#define UVC_VC_HEADER 0x01
```

6.107.1.166 UVC_VC_INPUT_TERMINAL

```
#define UVC_VC_INPUT_TERMINAL 0x02
```

6.107.1.167 UVC_VC_OUTPUT_TERMINAL

```
#define UVC_VC_OUTPUT_TERMINAL 0x03
```

6.107.1.168 UVC_VC_PROCESSING_UNIT

```
#define UVC_VC_PROCESSING_UNIT 0x05
```

6.107.1.169 UVC_VC_REQUEST_ERROR_CODE_CONTROL

```
#define UVC_VC_REQUEST_ERROR_CODE_CONTROL 0x02
```

Kneron Confidential

6.107.1.170 UVC_VC_SELECTOR_UNIT

```
#define UVC_VC_SELECTOR_UNIT 0x04
```

6.107.1.171 UVC_VC_VIDEO_POWER_MODE_CONTROL

```
#define UVC_VC_VIDEO_POWER_MODE_CONTROL 0x01
```

6.107.1.172 UVC_VS_COLORFORMAT

```
#define UVC_VS_COLORFORMAT 0x0d
```

6.107.1.173 UVC_VS_COMMIT_CONTROL

```
#define UVC_VS_COMMIT_CONTROL 0x02
```

6.107.1.174 UVC_VS_CONTROL_UNDEFINED

```
#define UVC_VS_CONTROL_UNDEFINED 0x00
```

6.107.1.175 UVC_VS_FORMAT_DV

```
#define UVC_VS_FORMAT_DV 0x0c
```

6.107.1.176 UVC_VS_FORMAT_FRAME_BASED

```
#define UVC_VS_FORMAT_FRAME_BASED 0x10
```

6.107.1.177 UVC_VS_FORMAT_MJPEG

```
#define UVC_VS_FORMAT_MJPEG 0x06
```

Kneron Confidential

6.107.1.178 UVC_VS_FORMAT_MPEG2TS

```
#define UVC_VS_FORMAT_MPEG2TS 0x0a
```

6.107.1.179 UVC_VS_FORMAT_STREAM_BASED

```
#define UVC_VS_FORMAT_STREAM_BASED 0x12
```

6.107.1.180 UVC_VS_FORMAT_UNCOMPRESSED

```
#define UVC_VS_FORMAT_UNCOMPRESSED 0x04
```

6.107.1.181 UVC_VS_FRAME_FRAME_BASED

```
#define UVC_VS_FRAME_FRAME_BASED 0x11
```

6.107.1.182 UVC_VS_FRAME_MJPEG

```
#define UVC_VS_FRAME_MJPEG 0x07
```

6.107.1.183 UVC_VS_FRAME_UNCOMPRESSED

```
#define UVC_VS_FRAME_UNCOMPRESSED 0x05
```

6.107.1.184 UVC_VS_GENERATE_KEY_FRAME_CONTROL

```
#define UVC_VS_GENERATE_KEY_FRAME_CONTROL 0x07
```

6.107.1.185 UVC_VS_INPUT_HEADER

```
#define UVC_VS_INPUT_HEADER 0x01
```

Kneron Confidential

6.107.1.186 UVC_VS_OUTPUT_HEADER

```
#define UVC_VS_OUTPUT_HEADER 0x02
```

6.107.1.187 UVC_VS_PROBE_CONTROL

```
#define UVC_VS_PROBE_CONTROL 0x01
```

6.107.1.188 UVC_VS_STILL_COMMIT_CONTROL

```
#define UVC_VS_STILL_COMMIT_CONTROL 0x04
```

6.107.1.189 UVC_VS_STILL_IMAGE_FRAME

```
#define UVC_VS_STILL_IMAGE_FRAME 0x03
```

6.107.1.190 UVC_VS_STILL_IMAGE_TRIGGER_CONTROL

```
#define UVC_VS_STILL_IMAGE_TRIGGER_CONTROL 0x05
```

6.107.1.191 UVC_VS_STILL_PROBE_CONTROL

```
#define UVC_VS_STILL_PROBE_CONTROL 0x03
```

6.107.1.192 UVC_VS_STREAM_ERROR_CODE_CONTROL

```
#define UVC_VS_STREAM_ERROR_CODE_CONTROL 0x06
```

6.107.1.193 UVC_VS_SYNC_DELAY_CONTROL

```
#define UVC_VS_SYNC_DELAY_CONTROL 0x09
```

Kneron Confidential

6.107.1.194 UVC_VS_UNDEFINED

```
#define UVC_VS_UNDEFINED 0x00
```

6.107.1.195 UVC_VS_UPDATE_FRAME_SEGMENT_CONTROL

```
#define UVC_VS_UPDATE_FRAME_SEGMENT_CONTROL 0x08
```

6.107.1.196 UVC_XU_CONST_LEN

```
#define UVC_XU_CONST_LEN 22
```

6.107.1.197 UVC_XU_CONTROL_UNDEFINED

```
#define UVC_XU_CONTROL_UNDEFINED 0x00
```

6.107.1.198 VS_FORMAT_H264

```
#define VS_FORMAT_H264 0x13
```

6.107.1.199 VS_FORMAT_H264_SIMULCAST

```
#define VS_FORMAT_H264_SIMULCAST 0x15
```

6.107.1.200 VS_FORMAT_VP8

```
#define VS_FORMAT_VP8 0x16
```

6.107.1.201 VS_FORMAT_VP8_SIMULCAST

```
#define VS_FORMAT_VP8_SIMULCAST 0x18
```

Kneron Confidential

6.107.1.202 VS_FRAME_H264

```
#define VS_FRAME_H264 0x14
```

6.107.1.203 VS_FRAME_VP8

```
#define VS_FRAME_VP8 0x17
```

6.107.2 Variable Documentation

6.107.2.1 baInterfaceNr

```
uint8_t baInterfaceNr[]
```

6.107.2.2 baSourceID

```
uint8_t baSourceID
```

6.107.2.3 bAspectRatioX

```
uint8_t bAspectRatioX
```

6.107.2.4 bAspectRatioY

```
uint8_t bAspectRatioY
```

6.107.2.5 bAssocTerminal

```
uint8_t bAssocTerminal
```

Kneron Confidential

6.107.2.6 bBitDrpthLuma

```
uint8_t bBitDrpthLuma
```

6.107.2.7 bBitsPerPixel

```
uint8_t bBitsPerPixel
```

6.107.2.8 bcdUVC

```
uint16_t bcdUVC
```

6.107.2.9 bColorPrimaries

```
uint8_t bColorPrimaries
```

6.107.2.10 bCompression

```
uint8_t bCompression[]
```

6.107.2.11 bCompressionIndex

```
uint8_t bCompressionIndex
```

6.107.2.12 bControlSize

```
uint8_t bControlSize
```

6.107.2.13 bCopyProtect

```
uint8_t bCopyProtect
```

Kneron Confidential

6.107.2.14 bDefaultFrameIndex

```
uint8_t bDefaultFrameIndex
```

6.107.2.15 bDescriptorSubType

```
uint8_t bDescriptorSubType
```

6.107.2.16 bDescriptorSubtype

```
uint8_t bDescriptorSubtype
```

6.107.2.17 bDescriptorType

```
uint8_t bDescriptorType
```

6.107.2.18 bEndpointAddress

```
uint8_t bEndpointAddress
```

6.107.2.19 bFirstInterface

```
uint8_t bFirstInterface
```

6.107.2.20 bFormatIndex

```
uint8_t bFormatIndex
```

6.107.2.21 bFrameIndex

```
uint8_t bFrameIndex
```

Kneron Confidential

6.107.2.22 bFrameIntervalType

```
uint8_t bFrameIntervalType
```

6.107.2.23 bFunctionClass

```
uint8_t bFunctionClass
```

6.107.2.24 bFunctionProtocol

```
uint8_t bFunctionProtocol
```

6.107.2.25 bFunctionSubClass

```
uint8_t bFunctionSubClass
```

6.107.2.26 bInCollection

```
uint8_t bInCollection
```

6.107.2.27 bInterfaceCount

```
uint8_t bInterfaceCount
```

6.107.2.28 bLength

```
uint8_t bLength
```

6.107.2.29 bmaControls

```
uint8_t bmaControls
```

Kneron Confidential

6.107.2.30 bMatrixCoefficients

```
uint8_t bMatrixCoefficients
```

6.107.2.31 bMaxNumberOfRefFeamesPlus1

```
uint8_t bMaxNumberOfRefFeamesPlus1
```

6.107.2.32 bMaxVersion

```
uint8_t bMaxVersion
```

6.107.2.33 bmCapabilities

```
uint8_t bmCapabilities
```

6.107.2.34 bmControls

```
uint8_t bmControls
```

6.107.2.35 bmControlsRuntime

```
uint8_t bmControlsRuntime[0]
```

6.107.2.36 bmFlags

```
uint8_t bmFlags
```

6.107.2.37 bmFramingInfo

```
uint8_t bmFramingInfo
```

Kneron Confidential

6.107.2.38 bmInfo

```
uint8_t bmInfo
```

6.107.2.39 bmInterfaceFlags

```
uint8_t bmInterfaceFlags
```

6.107.2.40 bMinVersion

```
uint8_t bMinVersion
```

6.107.2.41 bmLayoutPerStream

```
uint8_t bmLayoutPerStream[8]
```

6.107.2.42 bmSetting

```
uint8_t bmSetting
```

6.107.2.43 bmVideoStandards

```
uint8_t bmVideoStandards
```

6.107.2.44 bNrInPins

```
uint8_t bNrInPins
```

6.107.2.45 bNumCompressionPattern

```
uint8_t bNumCompressionPattern
```

Kneron Confidential

6.107.2.46 bNumControls

```
uint8_t bNumControls
```

6.107.2.47 bNumFormats

```
uint8_t bNumFormats
```

6.107.2.48 bNumFrameDescriptors

```
uint8_t bNumFrameDescriptors
```

6.107.2.49 bNumImageSizePatterns

```
uint8_t bNumImageSizePatterns
```

6.107.2.50 bPreferedVersion

```
uint8_t bPreferedVersion
```

6.107.2.51 bSourceID

```
uint8_t bSourceID
```

6.107.2.52 bStillCaptureMethod

```
uint8_t bStillCaptureMethod
```

6.107.2.53 bTerminalID

```
uint8_t bTerminalID
```

Kneron Confidential

6.107.2.54 bTerminalLink

```
uint8_t bTerminalLink
```

6.107.2.55 bTransferCharacteristics

```
uint8_t bTransferCharacteristics
```

6.107.2.56 bTriggerSupport

```
uint8_t bTriggerSupport
```

6.107.2.57 bTriggerUsage

```
uint8_t bTriggerUsage
```

6.107.2.58 bUnitID

```
uint8_t bUnitID
```

6.107.2.59 bUsage

```
uint8_t bUsage
```

6.107.2.60 bVariableSize

```
uint8_t bVariableSize
```

6.107.2.61 dwClockFrequency

```
uint32_t dwClockFrequency
```

Kneron Confidential

6.107.2.62 dwDefaultFrameInterval

```
uint32_t dwDefaultFrameInterval
```

6.107.2.63 dwFrameInterval

```
uint32_t dwFrameInterval
```

6.107.2.64 dwMaxBitRate

```
uint32_t dwMaxBitRate
```

6.107.2.65 dwMaxPayloadTransferSize

```
uint32_t dwMaxPayloadTransferSize
```

6.107.2.66 dwMaxVideoFrameBufferSize

```
uint32_t dwMaxVideoFrameBufferSize
```

6.107.2.67 dwMaxVideoFrameSize

```
uint32_t dwMaxVideoFrameSize
```

6.107.2.68 dwMinBitRate

```
uint32_t dwMinBitRate
```

6.107.2.69 guidExtensionCode

```
uint8_t guidExtensionCode[16]
```

Kneron Confidential

6.107.2.70 guidFormat

```
uint8_t guidFormat
```

6.107.2.71 iEncoding

```
uint8_t iEncoding
```

6.107.2.72 iExtension

```
uint8_t iExtension
```

6.107.2.73 iFunction

```
uint8_t iFunction
```

6.107.2.74 iProcessing

```
uint8_t iProcessing
```

6.107.2.75 iSelector

```
uint8_t iSelector
```

6.107.2.76 iTerminal

```
uint8_t iTerminal
```

6.107.2.77 wCompQuality

```
uint16_t wCompQuality
```

Kneron Confidential

6.107.2.78 wCompWindowSize

```
uint16_t wCompWindowSize
```

6.107.2.79 wDelay

```
uint16_t wDelay
```

6.107.2.80 wHeight

```
uint16_t wHeight
```

6.107.2.81 wKeyFrameRate

```
uint16_t wKeyFrameRate
```

6.107.2.82 wMaxMultiplier

```
uint16_t wMaxMultiplier
```

6.107.2.83 wMaxTransferSize

```
uint16_t wMaxTransferSize
```

6.107.2.84 wmHint

```
uint16_t wmHint
```

6.107.2.85 wObjectiveFocalLengthMax

```
uint16_t wObjectiveFocalLengthMax
```

Kneron Confidential

6.107.2.86 wObjectiveFocalLengthMin

```
uint16_t wObjectiveFocalLengthMin
```

6.107.2.87 wOcularFocalLength

```
uint16_t wOcularFocalLength
```

6.107.2.88 wPFrameRate

```
uint16_t wPFrameRate
```

6.107.2.89 wRateControlModes

```
uint16_t wRateControlModes
```

6.107.2.90 wTerminalType

```
uint16_t wTerminalType
```

6.107.2.91 wTotalLength

```
uint16_t wTotalLength
```

6.107.2.92 wWidth

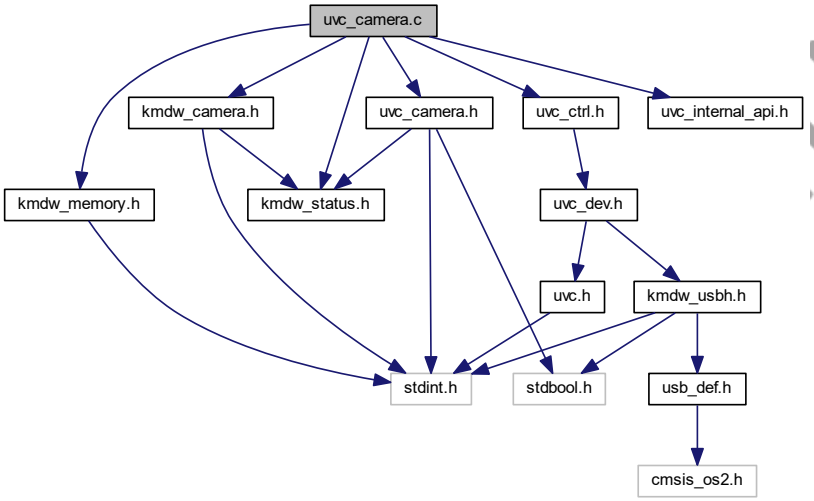
```
uint16_t wWidth
```

Kneron Confidential

6.108 uvc_camera.c File Reference

```
#include "uvc_camera.h"
#include "kmdw_memory.h"
#include "uvc_internal_api.h"
#include "uvc_ctrl.h"
#include "kmdw_camera.h"
#include "kmdw_status.h"
```

Include dependency graph for uvc_camera.c:



Functions

- int kdp_uvc_get_ctl_list (uint32_t *ctl_list)

Variables

- struct uvc_device * uvc_video_device

6.108.1 Function Documentation

6.108.1.1 kdp_uvc_get_ctl_list()

```
int kdp_uvc_get_ctl_list (
            uint32_t * ctl_list )
```

6.108.2 Variable Documentation

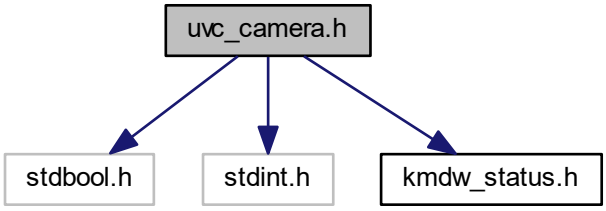
6.108.2.1 uvc_video_device

```
struct uvc_device* uvc_video_device
```

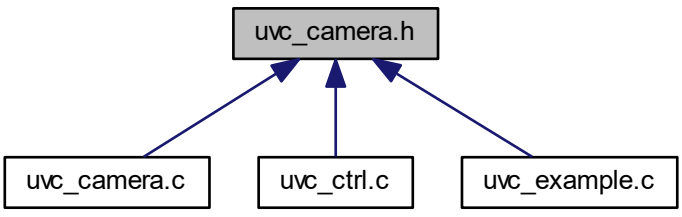
6.109 uvc_camera.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "kmdw_status.h"
```

Include dependency graph for uvc_camera.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct ct_scm
- struct ct_aem
- struct ct_aep
- struct ct_eta
- struct ct_etr
- struct ct_focus_a
- struct ct_focus_r_data
- struct ct_focus_r
- struct ct_focus_sr
- struct ct_fauto
- struct ct_iris_a
- struct ct_iris_r
- struct ct_zoom_a
- struct ct_zoomr_data
- struct ct_zoom_r
- struct ct_pan_tilta_data
- struct ct_pan_tilt_a
- struct ct_pan_tiltr_data
- struct ct_pan_tilt_r
- struct ct_roll_a
- struct ct_rollr_data
- struct ct_roll_r
- struct ct_privacy_shutter
- struct ct_dwindow_data
- struct ct_dwindow
- struct ct_roi_data
- struct ct_roi
- struct pu_backlight
- struct pu_brightness
- struct pu_contrast
- struct pu_contrast_auto
- struct pu_gain
- struct pu_power_line_frequency
- struct pu_hue
- struct pu_hue_auto
- struct pu_saturation
- struct pu_sharpness
- struct pu_gamma
- struct pu_white_balance_temp
- struct pu_white_balance_temp_auto
- struct pu_whitebalance_comp_data
- struct pu_whitebalance_comp
- struct pu_wbc_auto
- struct pu_dmultiplier
- struct pu_dmultiplierlimit

Kneron Confidential

Macros

- #define SCANNING_MODE 0x1
- #define AUTO_EXPOSURE_MODE 0x2
- #define AUTO_EXPOSURE_PRIORITY 0x4
- #define EXPOSURE_TIME_ABSOLUTE 0x8
- #define EXPOSURE_TIME_RELATIVE 0x10
- #define FOCUS_ABSOLUTE 0x20
- #define FOCUS_RELATIVE 0x40
- #define IRIS_ABSOLUTE 0x80
- #define IRIS_RELATIVE 0x100
- #define ZOOM_ABSOLUTE 0x200
- #define ZOOM_RELATIVE 0x400
- #define PANTILT_ABSOLUTE 0x800
- #define PANTILT_RELATIVE 0x1000
- #define ROLL_ABSOLUTE 0x2000
- #define ROLL_RELATIVE 0x4000
- #define FOCUS_AUTO 0x20000
- #define PRIVACY 0x40000
- #define FOCUS_SIMPLE 0x80000
- #define WINDOW 0x100000
- #define REGION_OF_INTEREST 0x200000
- #define BRIGHTNESS 0x1
- #define CONTRAST 0x2
- #define HUE 0x4
- #define SATURATION 0x8
- #define SHARPNESS 0x10
- #define GAMMA 0x20
- #define WHITE_BALANCE_TEMPERATURE 0x40
- #define WHITE_BALANCE_COMPONENT 0x80
- #define BACKLIGHT_COMPENSATION 0x100
- #define GAIN 0x200
- #define POWER_LINE_FREQUENCY 0x400
- #define HUE_AUTO 0x800
- #define WHITE_BALANCE_TEMPERATURE_AUTO 0x1000
- #define WHITE_BALANCE_COMPONENT_AUTO 0x2000
- #define DIGITAL_MULTIPLIER 0x4000
- #define DIGITAL_MULTIPLIER_LIMIT 0x8000
- #define ANALOG_VIDEO_STANDARD 0x10000
- #define ANALOG_VIDEO_LOCK_STATUS 0x20000
- #define CONTRAST_AUTO 0x40000
- #define UVC_SET_CUR 0x01
- #define UVC_GET_CUR 0x81
- #define UVC_GET_MIN 0x82
- #define UVC_GET_MAX 0x83
- #define UVC_GET_RES 0x84
- #define UVC_GET_LEN 0x85
- #define UVC_GET_INFO 0x86
- #define UVC_GET_DEF 0x87
- #define SCANNING_MODE_CTL_INTERLACED 0x0
- #define SCANNING_MODE_CTL_PROGRESSIVE 0x1
- #define EXPOSURE_MANUAL_MODE 0x1
- #define EXPOSURE_AUTO_MODE 0x2
- #define EXPOSURE_SHUTTER_PRIORITY_MODE 0x4
- #define EXPOSURE_APERTURE_PRIORITY_MODE 0x8

Kneron Confidential

- #define EXPOSURE_FRAME_RATE_CONSTANT 0x0
- #define EXPOSURE_FRAME_RATE_VARIED 0x1
- #define FSR_FULL_RANGE 0x1
- #define FSR_MACRO 0x2
- #define FSR_PEOPLE 0x3
- #define FSR_SCENE 0x4
- #define ZOOM_STOP 0x0
- #define ZOOM_TELE_DIR 0x1
- #define ZOOM_WIDE_AGLE_DIR 0xFF
- #define DIGITAL_ZOOM_OFF 0x0
- #define DIGITAL_ZOOM_ON 0x1
- #define PAN_STOP 0x0
- #define PAN_CLOSEWISE_DIR 0x1
- #define PAN_COUNTER_CLOSEWISE_DIR 0xFF
- #define TILT_STOP 0x0
- #define TILT_POINT_IMG_UP 0x1
- #define TILT_POINT_IMG_DOWN 0xFF
- #define SHUTTER_OPEN 0x0
- #define SHUTTER_CLOSE 0x1
- #define PLF_DSIABLE 0x0
- #define PLF_50HZ 0x1
- #define PLF_60HZ 0x2
- #define PLF_AUTO 0x3

Enumerations

- enum scm_req { SCM_SET_CUR = 0x01, SCM_GET_CUR = 0x81, SCM_GET_CAP = 0x86 }
- enum aem_req {
  AEM_SET_CUR = 0x01, AEM_GET_CUR = 0x81, AEM_GET_RES = 0x84, AEM_GET_CAP = 0x86,
  AEM_GET_DEF = 0x87 }
- enum aep_req { AEP_SET_CUR = 0x01, AEP_GET_CUR = 0x81, AEP_GET_CAP = 0x86 }
- enum eta_req {
  ETA_SET_CUR = 0x01, ETA_GET_CUR = 0x81, ETA_GET_MIN = 0x82, ETA_GET_MAX = 0x83,
  ETA_GET_RES = 0x84, ETA_GET_CAP = 0x86, ETA_GET_DEF = 0x87 }
- enum etr_req { ETR_SET_CUR = 0x01, ETR_GET_CUR = 0x81, ETR_GET_CAP = 0x86 }
- enum focus_a_req {
  FA_SET_CUR = 0x01, FA_GET_CUR = 0x81, FA_GET_MIN = 0x82, FA_GET_MAX = 0x83,
  FA_GET_RES = 0x84, FA_GET_CAP = 0x86, FA_GET_DEF = 0x87 }
- enum fr_req {
  FR_SET_CUR = 0x01, FR_GET_CUR = 0x81, FR_GET_MIN = 0x82, FR_GET_MAX = 0x83,
  FR_GET_RES = 0x84, FR_GET_CAP = 0x86, FR_GET_DEF = 0x87 }
- enum fsr_req { FSR_SET_CUR = 0x01, FSR_GET_CUR = 0x81, FSR_GET_CAP = 0x86, FSR_GET_DEF = 0x87 }
- enum fauto_req { FAUTO_SET_CUR = 0x01, FAUTO_GET_CUR = 0x81, FAUTO_GET_CAP = 0x86, FAUTO_GET_DEF = 0x87 }
- enum irisa_req {
  IRISA_SET_CUR = 0x01, IRISA_GET_CUR = 0x81, IRISA_GET_MIN = 0x82, IRISA_GET_MAX = 0x83,
  IRISA_GET_RES = 0x84, IRISA_GET_CAP = 0x86, IRISA_GET_DEF = 0x87 }
- enum irisr_req { IRISR_SET_CUR = 0x01, IRISR_GET_CUR = 0x81, IRISR_GET_CAP = 0x86 }
- enum zooma_req {
  ZOOMA_SET_CUR = 0x01, ZOOMA_GET_CUR = 0x81, ZOOMA_GET_MIN = 0x82, ZOOMA_GET_MAX = 0x83,
  ZOOMA_GET_RES = 0x84, ZOOMA_GET_CAP = 0x86, ZOOMA_GET_DEF = 0x87 }

Kneron Confidential

- enum zoomr_req {
  ZOOMR_SET_CUR = 0x01, ZOOMR_GET_CUR = 0x81, ZOOMR_GET_MIN = 0x82, ZOOMR_GET_MAX = 0x83,
  ZOOMR_GET_RES = 0x84, ZOOMR_GET_CAP = 0x86, ZOOMR_GET_DEF = 0x87 }
- enum pantilta_req {
  TILTA_SET_CUR = 0x01, TILTA_GET_CUR = 0x81, TILTA_GET_MIN = 0x82, TILTA_GET_MAX = 0x83,
  TILTA_GET_RES = 0x84, TILTA_GET_CAP = 0x86, TILTA_GET_DEF = 0x87 }
- enum pantiltr_req {
  TILTR_SET_CUR = 0x01, TILTR_GET_CUR = 0x81, TILTR_GET_MIN = 0x82, TILTR_GET_MAX = 0x83,
  TILTR_GET_RES = 0x84, TILTR_GET_CAP = 0x86, TILTR_GET_DEF = 0x87 }
- enum rolla_req {
  ROLLA_SET_CUR = 0x01, ROLLA_GET_CUR = 0x81, ROLLA_GET_MIN = 0x82, ROLLA_GET_MAX = 0x83,
  ROLLA_GET_RES = 0x84, ROLLA_GET_CAP = 0x86, ROLLA_GET_DEF = 0x87 }
- enum rollr_req {
  ROLLR_SET_CUR = 0x01, ROLLR_GET_CUR = 0x81, ROLLR_GET_MIN = 0x82, ROLLR_GET_MAX = 0x83,
  ROLLR_GET_RES = 0x84, ROLLR_GET_CAP = 0x86, ROLLR_GET_DEF = 0x87 }
- enum ps_req { PS_SET_CUR = 0x01, PS_GET_CUR = 0x81, PS_GET_CAP = 0x86 }
- enum dwindow_req {
  DWINDOW_SET_CUR = 0x01, DWINDOW_GET_CUR = 0x81, DWINDOW_GET_MIN = 0x82,
  DWINDOW_GET_MAX = 0x83,
  DWINDOW_GET_DEF = 0x87 }
- enum roi_req {
  ROI_SET_CUR = 0x01, ROI_GET_CUR = 0x81, ROI_GET_MIN = 0x82, ROI_GET_MAX = 0x83,
  ROI_GET_DEF = 0x87 }
- enum backlight_req {
  BKC_SET_CUR = 0x01, BKC_GET_CUR = 0x81, BKC_GET_MIN = 0x82, BKC_GET_MAX = 0x83,
  BKC_GET_RES = 0x84, BKC_GET_CAP = 0x86, BKC_GET_DEF = 0x87 }
- enum brightness_req {
  BRIGHTNESS_SET_CUR = 0x01, BRIGHTNESS_GET_CUR = 0x81, BRIGHTNESS_GET_MIN = 0x82,
  BRIGHTNESS_GET_MAX = 0x83,
  BRIGHTNESS_GET_RES = 0x84, BRIGHTNESS_GET_CAP = 0x86, BRIGHTNESS_GET_DEF = 0x87 }
- enum contrast_req {
  CONTRAST_SET_CUR = 0x01, CONTRAST_GET_CUR = 0x81, CONTRAST_GET_MIN = 0x82,
  CONTRAST_GET_MAX = 0x83,
  CONTRAST_GET_RES = 0x84, CONTRAST_GET_CAP = 0x86, CONTRAST_GET_DEF = 0x87 }
- enum contrast_auto_req { CONTRASTA_SET_CUR = 0x01, CONTRASTA_GET_CUR = 0x81,
  CONTRASTA_GET_CAP = 0x86, CONTRASTA_GET_DEF = 0x87 }
- enum gain_req {
  GAIN_SET_CUR = 0x01, GAIN_GET_CUR = 0x81, GAIN_GET_MIN = 0x82, GAIN_GET_MAX = 0x83,
  GAIN_GET_RES = 0x84, GAIN_GET_CAP = 0x86, GAIN_GET_DEF = 0x87 }
- enum power_line_freq_req { POWER_LINE_FREQUENCY_SET_CUR = 0x01, POWER_LINE_FREQUENCY_GET_CUR = 0x81, POWER_LINE_FREQUENCY_GET_CAP = 0x86, POWER_LINE_FREQUENCY_GET_DEF = 0x87
  }
- enum hue_req {
  HUE_SET_CUR = 0x01, HUE_GET_CUR = 0x81, HUE_GET_MIN = 0x82, HUE_GET_MAX = 0x83,
  HUE_GET_RES = 0x84, HUE_GET_CAP = 0x86, HUE_GET_DEF = 0x87 }
- enum hue_auto_req { HUEA_SET_CUR = 0x01, HUEA_GET_CUR = 0x81, HUEA_GET_CAP = 0x86,
  HUEA_GET_DEF = 0x87 }
- enum saturation_req {
  SATURATION_SET_CUR = 0x01, SATURATION_GET_CUR = 0x81, SATURATION_GET_MIN = 0x82,
  SATURATION_GET_MAX = 0x83,
  SATURATION_GET_RES = 0x84, SATURATION_GET_CAP = 0x86, SATURATION_GET_DEF = 0x87 }
- enum sharpness_req {
  SHARPNESS_SET_CUR = 0x01, SHARPNESS_GET_CUR = 0x81, SHARPNESS_GET_MIN = 0x82,
  SHARPNESS_GET_MAX = 0x83,
  SHARPNESS_GET_RES = 0x84, SHARPNESS_GET_CAP = 0x86, SHARPNESS_GET_DEF = 0x87 }

Kneron Confidential

- enum gamma_req {
  GAMMA_SET_CUR = 0x01, GAMMA_GET_CUR = 0x81, GAMMA_GET_MIN = 0x82, GAMMA_GET_MAX = 0x83,
  GAMMA_GET_RES = 0x84, GAMMA_GET_CAP = 0x86, GAMMA_GET_DEF = 0x87 }
- enum wbt_req {
  WBT_SET_CUR = 0x01, WBT_GET_CUR = 0x81, WBT_GET_MIN = 0x82, WBT_GET_MAX = 0x83,
  WBT_GET_RES = 0x84, WBT_GET_CAP = 0x86, WBT_GET_DEF = 0x87 }
- enum wbt_auto_req { WBTA_SET_CUR = 0x01, WBTA_GET_CUR = 0x81, WBTA_GET_CAP = 0x86, WBTA_GET_DEF = 0x87 }
- enum whitebalance_comp_req {
  WBC_SET_CUR = 0x01, WBC_GET_CUR = 0x81, WBC_GET_MIN = 0x82, WBC_GET_MAX = 0x83,
  WBC_GET_RES = 0x84, WBC_GET_CAP = 0x86, WBC_GET_DEF = 0x87 }
- enum wbc_auto_req { WBCA_SET_CUR = 0x01, WBCA_GET_CUR = 0x81, WBCA_GET_CAP = 0x86, WBCA_GET_DEF = 0x87 }
- enum dmpl_req {
  MPL_SET_CUR = 0x01, MPL_GET_CUR = 0x81, MPL_GET_MIN = 0x82, MPL_GET_MAX = 0x83,
  MPL_GET_RES = 0x84, MPL_GET_CAP = 0x86, MPL_GET_DEF = 0x87 }
- enum dmpl_limit_req {
  DMPL_SET_CUR = 0x01, DMPL_GET_CUR = 0x81, DMPL_GET_MIN = 0x82, DMPL_GET_MAX = 0x83,
  DMPL_GET_RES = 0x84, DMPL_GET_CAP = 0x86, DMPL_GET_DEF = 0x87 }

Variables

- enum scm_req __attribute__

6.109.1 Macro Definition Documentation

6.109.1.1 ANALOG_VIDEO_LOCK_STATUS

```
#define ANALOG_VIDEO_LOCK_STATUS 0x20000
```

6.109.1.2 ANALOG_VIDEO_STANDARD

```
#define ANALOG_VIDEO_STANDARD 0x10000
```

6.109.1.3 AUTO_EXPOSURE_MODE

```
#define AUTO_EXPOSURE_MODE 0x2
```

Kneron Confidential

6.109.1.4 AUTO_EXPOSURE_PRIORITY

```
#define AUTO_EXPOSURE_PRIORITY 0x4
```

6.109.1.5 BACKLIGHT_COMPENSATION

```
#define BACKLIGHT_COMPENSATION 0x100
```

6.109.1.6 BRIGHTNESS

```
#define BRIGHTNESS 0x1
```

6.109.1.7 CONTRAST

```
#define CONTRAST 0x2
```

6.109.1.8 CONTRAST_AUTO

```
#define CONTRAST_AUTO 0x40000
```

6.109.1.9 DIGITAL_MULTIPLIER

```
#define DIGITAL_MULTIPLIER 0x4000
```

6.109.1.10 DIGITAL_MULTIPLIER_LIMIT

```
#define DIGITAL_MULTIPLIER_LIMIT 0x8000
```

6.109.1.11 DIGITAL_ZOOM_OFF

```
#define DIGITAL_ZOOM_OFF 0x0
```

Kneron Confidential

6.109.1.12 DIGITAL_ZOOM_ON

```
#define DIGITAL_ZOOM_ON 0x1
```

6.109.1.13 EXPOSURE_APERTURE_PRIORITY_MODE

```
#define EXPOSURE_APERTURE_PRIORITY_MODE 0x8
```

6.109.1.14 EXPOSURE_AUTO_MODE

```
#define EXPOSURE_AUTO_MODE 0x2
```

6.109.1.15 EXPOSURE_FRAME_RATE_CONSTANT

```
#define EXPOSURE_FRAME_RATE_CONSTANT 0x0
```

6.109.1.16 EXPOSURE_FRAME_RATE_VARIED

```
#define EXPOSURE_FRAME_RATE_VARIED 0x1
```

6.109.1.17 EXPOSURE_MANUAL_MODE

```
#define EXPOSURE_MANUAL_MODE 0x1
```

6.109.1.18 EXPOSURE_SHUTTER_PRIORITY_MODE

```
#define EXPOSURE_SHUTTER_PRIORITY_MODE 0x4
```

6.109.1.19 EXPOSURE_TIME_ABSOLUTE

```
#define EXPOSURE_TIME_ABSOLUTE 0x8
```

Kneron Confidential

6.109.1.20 EXPOSURE_TIME_RELATIVE

```
#define EXPOSURE_TIME_RELATIVE 0x10
```

6.109.1.21 FOCUS_ABSOLUTE

```
#define FOCUS_ABSOLUTE 0x20
```

6.109.1.22 FOCUS_AUTO

```
#define FOCUS_AUTO 0x20000
```

6.109.1.23 FOCUS_RELATIVE

```
#define FOCUS_RELATIVE 0x40
```

6.109.1.24 FOCUS_SIMPLE

```
#define FOCUS_SIMPLE 0x80000
```

6.109.1.25 FSR_FULL_RANGE

```
#define FSR_FULL_RANGE 0x1
```

6.109.1.26 FSR_MACRO

```
#define FSR_MACRO 0x2
```

6.109.1.27 FSR_PEOPLE

```
#define FSR_PEOPLE 0x3
```

Kneron Confidential

6.109.1.28 FSR_SCENE

```
#define FSR_SCENE 0x4
```

6.109.1.29 GAIN

```
#define GAIN 0x200
```

6.109.1.30 GAMMA

```
#define GAMMA 0x20
```

6.109.1.31 HUE

```
#define HUE 0x4
```

6.109.1.32 HUE_AUTO

```
#define HUE_AUTO 0x800
```

6.109.1.33 IRIS_ABSOLUTE

```
#define IRIS_ABSOLUTE 0x80
```

6.109.1.34 IRIS_RELATIVE

```
#define IRIS_RELATIVE 0x100
```

6.109.1.35 PAN_CLOSEWISE_DIR

```
#define PAN_CLOSEWISE_DIR 0x1
```

Kneron Confidential

6.109.1.36 PAN_COUNTER_CLOSEWISE_DIR

```
#define PAN_COUNTER_CLOSEWISE_DIR 0xFF
```

6.109.1.37 PAN_STOP

```
#define PAN_STOP 0x0
```

6.109.1.38 PANTILT_ABSOLUTE

```
#define PANTILT_ABSOLUTE 0x800
```

6.109.1.39 PANTILT_RELATIVE

```
#define PANTILT_RELATIVE 0x1000
```

6.109.1.40 PLF_50HZ

```
#define PLF_50HZ 0x1
```

6.109.1.41 PLF_60HZ

```
#define PLF_60HZ 0x2
```

6.109.1.42 PLF_AUTO

```
#define PLF_AUTO 0x3
```

6.109.1.43 PLF_DSIABLE

```
#define PLF_DSIABLE 0x0
```

Kneron Confidential

6.109.1.44 POWER_LINE_FREQUENCY

```
#define POWER_LINE_FREQUENCY 0x400
```

6.109.1.45 PRIVACY

```
#define PRIVACY 0x40000
```

6.109.1.46 REGION_OF_INTEREST

```
#define REGION_OF_INTEREST 0x200000
```

6.109.1.47 ROLL_ABSOLUTE

```
#define ROLL_ABSOLUTE 0x2000
```

6.109.1.48 ROLL_RELATIVE

```
#define ROLL_RELATIVE 0x4000
```

6.109.1.49 SATURATION

```
#define SATURATION 0x8
```

6.109.1.50 SCANNING_MODE

```
#define SCANNING_MODE 0x1
```

6.109.1.51 SCANNING_MODE_CTL_INTERLACED

```
#define SCANNING_MODE_CTL_INTERLACED 0x0
```

Kneron Confidential

6.109.1.52 SCANNING_MODE_CTL_PROGRESSIVE

```
#define SCANNING_MODE_CTL_PROGRESSIVE 0x1
```

6.109.1.53 SHARPNESS

```
#define SHARPNESS 0x10
```

6.109.1.54 SHUTTER_CLOSE

```
#define SHUTTER_CLOSE 0x1
```

6.109.1.55 SHUTTER_OPEN

```
#define SHUTTER_OPEN 0x0
```

6.109.1.56 TILT_POINT_IMG_DOWN

```
#define TILT_POINT_IMG_DOWN 0xFF
```

6.109.1.57 TILT_POINT_IMG_UP

```
#define TILT_POINT_IMG_UP 0x1
```

6.109.1.58 TILT_STOP

```
#define TILT_STOP 0x0
```

6.109.1.59 UVC_GET_CUR

```
#define UVC_GET_CUR 0x81
```

Kneron Confidential

6.109.1.60 UVC_GET_DEF

```
#define UVC_GET_DEF 0x87
```

6.109.1.61 UVC_GET_INFO

```
#define UVC_GET_INFO 0x86
```

6.109.1.62 UVC_GET_LEN

```
#define UVC_GET_LEN 0x85
```

6.109.1.63 UVC_GET_MAX

```
#define UVC_GET_MAX 0x83
```

6.109.1.64 UVC_GET_MIN

```
#define UVC_GET_MIN 0x82
```

6.109.1.65 UVC_GET_RES

```
#define UVC_GET_RES 0x84
```

6.109.1.66 UVC_SET_CUR

```
#define UVC_SET_CUR 0x01
```

6.109.1.67 WHITE_BALANCE_COMPONENT

```
#define WHITE_BALANCE_COMPONENT 0x80
```

Kneron Confidential

6.109.1.68 WHITE_BALANCE_COMPONENT_AUTO

```
#define WHITE_BALANCE_COMPONENT_AUTO 0x2000
```

6.109.1.69 WHITE_BALANCE_TEMPERATURE

```
#define WHITE_BALANCE_TEMPERATURE 0x40
```

6.109.1.70 WHITE_BALANCE_TEMPERATURE_AUTO

```
#define WHITE_BALANCE_TEMPERATURE_AUTO 0x1000
```

6.109.1.71 WINDOW

```
#define WINDOW 0x100000
```

6.109.1.72 ZOOM_ABSOLUTE

```
#define ZOOM_ABSOLUTE 0x200
```

6.109.1.73 ZOOM_RELATIVE

```
#define ZOOM_RELATIVE 0x400
```

6.109.1.74 ZOOM_STOP

```
#define ZOOM_STOP 0x0
```

6.109.1.75 ZOOM_TELE_DIR

```
#define ZOOM_TELE_DIR 0x1
```

Kneron Confidential

6.109.1.76 ZOOM_WIDE_AGLE_DIR

```
#define ZOOM_WIDE_AGLE_DIR 0xFF
```

6.109.2 Enumeration Type Documentation

6.109.2.1 aem_req

```
enum aem_req
```

Enumerator

AEM_SET_CUR	
AEM_GET_CUR	
AEM_GET_RES	
AEM_GET_CAP	
AEM_GET_DEF	

6.109.2.2 aep_req

```
enum aep_req
```

Enumerator

AEP_SET_CUR	
AEP_GET_CUR	
AEP_GET_CAP	

6.109.2.3 backlight_req

```
enum backlight_req
```

Enumerator

BKC_SET_CUR	
BKC_GET_CUR	
BKC_GET_MIN	
BKC_GET_MAX	
BKC_GET_RES	
BKC_GET_CAP	
BKC_GET_DEF	

Kneron Confidential

6.109.2.4 brightness_req

```
enum brightness_req
```

Enumerator

BRIGHTNESS_SET_CUR	
BRIGHTNESS_GET_CUR	
BRIGHTNESS_GET_MIN	
BRIGHTNESS_GET_MAX	
BRIGHTNESS_GET_RES	
BRIGHTNESS_GET_CAP	
BRIGHTNESS_GET_DEF	

6.109.2.5 contrast_auto_req

```
enum contrast_auto_req
```

Enumerator

CONTRASTA_SET_CUR	
CONTRASTA_GET_CUR	
CONTRASTA_GET_CAP	
CONTRASTA_GET_DEF	

6.109.2.6 contrast_req

```
enum contrast_req
```

Enumerator

CONTRAST_SET_CUR	
CONTRAST_GET_CUR	
CONTRAST_GET_MIN	
CONTRAST_GET_MAX	
CONTRAST_GET_RES	
CONTRAST_GET_CAP	
CONTRAST_GET_DEF	

Kneron Confidential

6.109.2.7 dmpl_limit_req

enum dmpl_limit_req

Enumerator

DMPL_SET_CUR	
DMPL_GET_CUR	
DMPL_GET_MIN	
DMPL_GET_MAX	
DMPL_GET_RES	
DMPL_GET_CAP	
DMPL_GET_DEF	

6.109.2.8 dmpl_req

enum dmpl_req

Enumerator

MPL_SET_CUR	
MPL_GET_CUR	
MPL_GET_MIN	
MPL_GET_MAX	
MPL_GET_RES	
MPL_GET_CAP	
MPL_GET_DEF	

6.109.2.9 dwindow_req

enum dwindow_req

Enumerator

DWINDOW_SET_CUR	
DWINDOW_GET_CUR	
DWINDOW_GET_MIN	
DWINDOW_GET_MAX	
DWINDOW_GET_DEF	

Kneron Confidential

6.109.2.10 eta_req

enum eta_req

Enumerator

ETA_SET_CUR	
ETA_GET_CUR	
ETA_GET_MIN	
ETA_GET_MAX	
ETA_GET_RES	
ETA_GET_CAP	
ETA_GET_DEF	

6.109.2.11 etr_req

enum etr_req

Enumerator

ETR_SET_CUR	
ETR_GET_CUR	
ETR_GET_CAP	

6.109.2.12 fauto_req

enum fauto_req

Enumerator

FAUTO_SET_CUR	
FAUTO_GET_CUR	
FAUTO_GET_CAP	
FAUTO_GET_DEF	

6.109.2.13 focus_a_req

enum focus_a_req

Kneron Confidential

Enumerator

FA_SET_CUR	
FA_GET_CUR	
FA_GET_MIN	
FA_GET_MAX	
FA_GET_RES	
FA_GET_CAP	
FA_GET_DEF	

6.109.2.14 fr_req

```
enum fr_req
```

Enumerator

FR_SET_CUR	
FR_GET_CUR	
FR_GET_MIN	
FR_GET_MAX	
FR_GET_RES	
FR_GET_CAP	
FR_GET_DEF	

6.109.2.15 fsr_req

```
enum fsr_req
```

Enumerator

FSR_SET_CUR	
FSR_GET_CUR	
FSR_GET_CAP	
FSR_GET_DEF	

6.109.2.16 gain_req

```
enum gain_req
```

Kneron Confidential

Enumerator

GAIN_SET_CUR	
GAIN_GET_CUR	
GAIN_GET_MIN	
GAIN_GET_MAX	
GAIN_GET_RES	
GAIN_GET_CAP	
GAIN_GET_DEF	

6.109.2.17 gamma_req

```
enum gamma_req
```

Enumerator

GAMMA_SET_CUR	
GAMMA_GET_CUR	
GAMMA_GET_MIN	
GAMMA_GET_MAX	
GAMMA_GET_RES	
GAMMA_GET_CAP	
GAMMA_GET_DEF	

6.109.2.18 hue_auto_req

```
enum hue_auto_req
```

Enumerator

HUEA_SET_CUR	
HUEA_GET_CUR	
HUEA_GET_CAP	
HUEA_GET_DEF	

6.109.2.19 hue_req

```
enum hue_req
```

Kneron Confidential

Enumerator

HUE_SET_CUR	
HUE_GET_CUR	
HUE_GET_MIN	
HUE_GET_MAX	
HUE_GET_RES	
HUE_GET_CAP	
HUE_GET_DEF	

6.109.2.20 irisa_req

```
enum irisa_req
```

Enumerator

IRISA_SET_CUR	
IRISA_GET_CUR	
IRISA_GET_MIN	
IRISA_GET_MAX	
IRISA_GET_RES	
IRISA_GET_CAP	
IRISA_GET_DEF	

6.109.2.21 irisr_req

```
enum irisr_req
```

Enumerator

IRISR_SET_CUR	
IRISR_GET_CUR	
IRISR_GET_CAP	

6.109.2.22 pantilta_req

```
enum pantilta_req
```

Enumerator

TILTA_SET_CUR	

Kneron Confidential

Enumerator

TILTA_GET_CUR	
TILTA_GET_MIN	
TILTA_GET_MAX	
TILTA_GET_RES	
TILTA_GET_CAP	
TILTA_GET_DEF	

6.109.2.23 pantiltr_req

`enum pantiltr_req`

Enumerator

TILTR_SET_CUR	
TILTR_GET_CUR	
TILTR_GET_MIN	
TILTR_GET_MAX	
TILTR_GET_RES	
TILTR_GET_CAP	
TILTR_GET_DEF	

6.109.2.24 power_line_freq_req

`enum power_line_freq_req`

Enumerator

POWER_LINE_FREQUENCY_SET_CUR	
POWER_LINE_FREQUENCY_GET_CUR	
POWER_LINE_FREQUENCY_GET_CAP	
POWER_LINE_FREQUENCY_GET_DEF	

6.109.2.25 ps_req

`enum ps_req`

Enumerator

PS_SET_CUR	
PS_GET_CUR	
PS_GET_CAP	

Kneron Confidential

6.109.2.26 roi_req

enum roi_req

Enumerator

ROI_SET_CUR	
ROI_GET_CUR	
ROI_GET_MIN	
ROI_GET_MAX	
ROI_GET_DEF	

6.109.2.27 rolla_req

enum rolla_req

Enumerator

ROLLA_SET_CUR	
ROLLA_GET_CUR	
ROLLA_GET_MIN	
ROLLA_GET_MAX	
ROLLA_GET_RES	
ROLLA_GET_CAP	
ROLLA_GET_DEF	

6.109.2.28 rollr_req

enum rollr_req

Enumerator

ROLLR_SET_CUR	
ROLLR_GET_CUR	
ROLLR_GET_MIN	
ROLLR_GET_MAX	
ROLLR_GET_RES	
ROLLR_GET_CAP	
ROLLR_GET_DEF	

Kneron Confidential

6.109.2.29 saturation_req

enum saturation_req

Enumerator

SATURATION_SET_CUR	
SATURATION_GET_CUR	
SATURATION_GET_MIN	
SATURATION_GET_MAX	
SATURATION_GET_RES	
SATURATION_GET_CAP	
SATURATION_GET_DEF	

6.109.2.30 scm_req

enum scm_req

Enumerator

SCM_SET_CUR	
SCM_GET_CUR	
SCM_GET_CAP	

6.109.2.31 sharpness_req

enum sharpness_req

Enumerator

SHARPNESS_SET_CUR	
SHARPNESS_GET_CUR	
SHARPNESS_GET_MIN	
SHARPNESS_GET_MAX	
SHARPNESS_GET_RES	
SHARPNESS_GET_CAP	
SHARPNESS_GET_DEF	

6.109.2.32 wbc_auto_req

enum wbc_auto_req

Kneron Confidential

Enumerator

WBCA_SET_CUR	
WBCA_GET_CUR	
WBCA_GET_CAP	
WBCA_GET_DEF	

6.109.2.33 wbt_auto_req

```
enum wbt_auto_req
```

Enumerator

WBTA_SET_CUR	
WBTA_GET_CUR	
WBTA_GET_CAP	
WBTA_GET_DEF	

6.109.2.34 wbt_req

```
enum wbt_req
```

Enumerator

WBT_SET_CUR	
WBT_GET_CUR	
WBT_GET_MIN	
WBT_GET_MAX	
WBT_GET_RES	
WBT_GET_CAP	
WBT_GET_DEF	

Kneron Confidential

6.109.2.35 whitebalance_comp_req

```
enum whitebalance_comp_req
```

Enumerator

WBC_SET_CUR	
WBC_GET_CUR	
WBC_GET_MIN	

Enumerator

WBC_GET_MAX	
WBC_GET_RES	
WBC_GET_CAP	
WBC_GET_DEF	

6.109.2.36 zooma_req

```
enum zooma_req
```

Enumerator

ZOOMA_SET_CUR	
ZOOMA_GET_CUR	
ZOOMA_GET_MIN	
ZOOMA_GET_MAX	
ZOOMA_GET_RES	
ZOOMA_GET_CAP	
ZOOMA_GET_DEF	

6.109.2.37 zoomr_req

```
enum zoomr_req
```

Enumerator

ZOOMR_SET_CUR	
ZOOMR_GET_CUR	
ZOOMR_GET_MIN	
ZOOMR_GET_MAX	
ZOOMR_GET_RES	
ZOOMR_GET_CAP	
ZOOMR_GET_DEF	

6.109.3 Variable Documentation

6.109.3.1 __attribute__

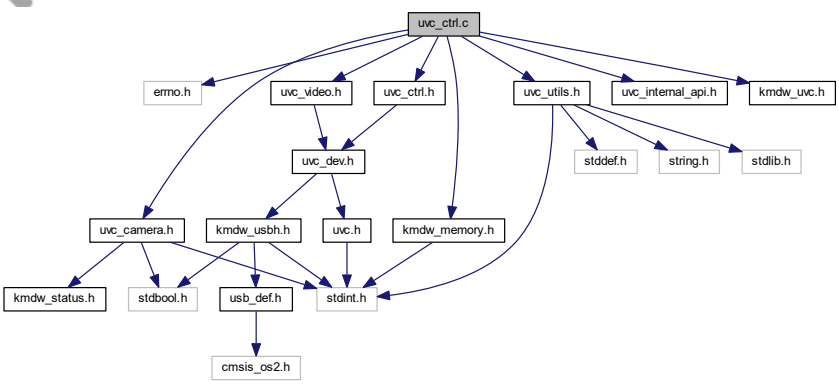
```
enum scm_req __attribute__
```

Kneron Confidential

6.110 uvc_ctrl.c File Reference

```
#include "errno.h"
#include "uvc_ctrl.h"
#include "uvc_video.h"
#include "uvc_utils.h"
#include "uvc_internal_api.h"
#include "kmdw_memory.h"
#include "kmdw_uvc.h"
#include "uvc_camera.h"
```

Include dependency graph for uvc_ctrl.c:

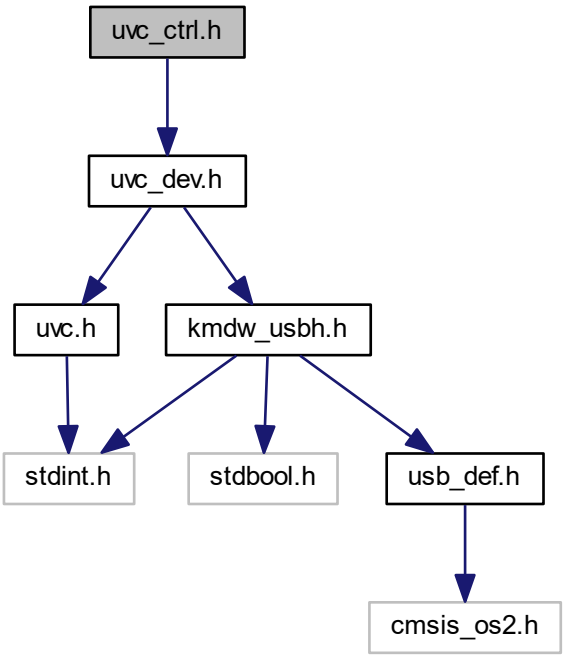


Kneron Confidential

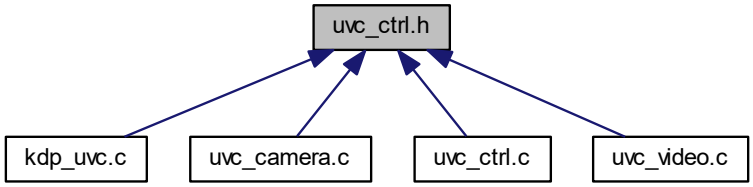
6.111 uvc_ctrl.h File Reference

```
#include "uvc_dev.h"
```

Include dependency graph for uvc_ctrl.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct uvc_xu_control_query
- struct uvc_usb_ctrlreq
- struct eu_select_layer

- struct eu_profile
- struct eu_videoresolution
- struct eu_min_frame_interval
- struct eu_slicemode
- struct eu_ratecontrolmode
- struct eu_qpprime
- struct eu_syncframe
- struct eu_ltrbuffers
- struct eu_ltrpicture
- struct eu_ltrvalidation
- struct eu_levelidc
- struct eu_seimessages
- struct eu_range
- struct eu_priority
- struct start_stop_layer
- struct ErrorResiliencyFeatures
- struct vs_synch_delay
- struct vs_streamerrorcode
- struct vs_generatekeyframe
- struct vs_updateframe
- struct vs_still_image_trigger

Macros

- #define UVC_CTRL_FLAG_SET_CUR (1 << 0)
- #define UVC_CTRL_FLAG_GET_CUR (1 << 1)
- #define UVC_CTRL_FLAG_GET_MIN (1 << 2)
- #define UVC_CTRL_FLAG_GET_MAX (1 << 3)
- #define UVC_CTRL_FLAG_GET_RES (1 << 4)
- #define UVC_CTRL_FLAG_GET_INFO (1 << 5)
- #define UVC_CTRL_FLAG_GET_DEF (1 << 6)
- #define UVC_CTRL_FLAG_GET_LEN (1 << 7)
- #define UVC_CTRL_FLAG_RESTORE (1 << 9)
- #define UVC_CTRL_FLAG_AUTO_UPDATE (1 << 10)
- #define UVC_CTRL_FLAG_GET_RANGE
- #define VC_CTRL_NUM 2
- #define DEVICE_POWER_MODE_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define REQUEST_ERROR_CODE_FLAG UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define CT_CTRL_NUM 22
- #define SCANNING_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define AUTO_EXPOSURE_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_RES | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define AUTO_EXPOSURE_PRIORITY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define EXPOSURE_TIME_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define EXPOSURE_TIME_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define FOCUS_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define FOCUS_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define IRIS_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define IRIS_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO

Kneron Confidential

- #define ZOOM_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define ZOOM_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define PANTILT_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define PANTILT_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define ROLL_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define ROLL_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define FOCUS_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define PRIVACY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INF
- #define FOCUS_SIMPLE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define WINDOW_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_DEF
- #define REGION_OF_INTEREST_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_DEF
- #define PU_CTRL_NUM 19
- #define BRIGHTNESS_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define CONTRAST_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define HUE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define SATURATION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define SHARPNESS_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define GAMMA_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define WHITE_BALANCE_TEMPERATURE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define WHITE_BALANCE_COMPONENT_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define BACKLIGHT_COMPENSATION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define GAIN_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define POWER_LINE_FREQUENCY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define HUE_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define WHITE_BALANCE_TEMPERATURE_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define WHITE_BALANCE_COMPONENT_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CU | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define DIGITAL_MULTIPLIER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define DIGITAL_MULTIPLIER_LIMIT_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
- #define ANALOG_VIDEO_STANDARD_CTL_FLAG UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define ANALOG_VIDEO_LOCK_STATUS_CTL_FLAG UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
- #define CONTRAST_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
- #define EU_CTRL_NUM 20
- #define SELECT_LAYER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define PROFILE_AND_TOOLSET_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define VIDEO_RESOLUTION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define MINIMUM_FRAME_INTERVAL_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define SLICE_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN

Kneron Confidential

- #define RATE_CONTROL_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define AVERAGE_BIT_RATE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define CPB_SIZE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define PEAK_BIT_RATE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define QUANTIZATION_PARAMETER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN | UVC_CTRL_FLAG_GET_RES
- #define SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define LONG_TERM_BUFFER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define PICTURE_LONG_TERM_REFERENCE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define LTR_VALIDATION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define LEVEL_IDC_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF
- #define SEI_MESSAGE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_INFO
- #define QP_RANGE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN | UVC_CTRL_FLAG_GET_RES
- #define PRIORITY_ID_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define START_OR_STOP_LAYER_VIEW_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
- #define ERROR_RESILIENCY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_RES
- #define HOST_SET_CLASS_INTF 0x21
- #define HOST_GET_CLASS_INTF 0xA1

Functions

- int uvc_ctrl_init_device (struct uvc_device *dev)
- struct uvc_usb_ctrlreq __attribute__ ((packed))
- int uvc_init_stream_ctl (struct uvc_device *udev, struct uvc_streaming *stream)
- int uvc_vs_probe (struct uvc_streaming *stream, struct uvc_streaming_control_data *probe)
- int uvc_ctrl_get (struct ctrl_info *ctrl, struct uvc_device *dev, uint8_t req_num)
- int uvc_ctrl_set (struct uvc_device *udev, struct ctrl_info *info)
- void set_ct_ctrl_flag (struct ctrl_info *info)
- void set_pu_ctrl_flag (struct ctrl_info *info)
- int uvc_send_ctl (struct uvc_device *udev, uint32_t cid, uint16_t req, uint8_t *para, uint8_t len)
- int uvc_init_device_ctrl (struct uvc_device *udev)

Kneron Confidential

Variables

- uint8_t bRequestType
- uint8_t bRequest
- uint16_t wValue
- uint16_t wIndex
- uint16_t wLength
- uint8_t * data
- struct eu_select_layer __attribute__

6.111.1 Macro Definition Documentation

6.111.1.1 ANALOG_VIDEO_LOCK_STATUS_CTL_FLAG

```
#define ANALOG_VIDEO_LOCK_STATUS_CTL_FLAG UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.2 ANALOG_VIDEO_STANDARD_CTL_FLAG

```
#define ANALOG_VIDEO_STANDARD_CTL_FLAG UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.3 AUTO_EXPOSURE_MODE_CTL_FLAG

```
#define AUTO_EXPOSURE_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_RES
| UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
```

6.111.1.4 AUTO_EXPOSURE_PRIORITY_CTL_FLAG

```
#define AUTO_EXPOSURE_PRIORITY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.5 AVERAGE_BIT_RATE_CTL_FLAG

```
#define AVERAGE_BIT_RATE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

Kneron Confidential

6.111.1.6 BACKLIGHT_COMPENSATION_CTL_FLAG

```
#define BACKLIGHT_COMPENSATION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.7 BRIGHTNESS_CTL_FLAG

```
#define BRIGHTNESS_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.8 CONTRAST_AUTO_CTL_FLAG

```
#define CONTRAST_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
| UVC_CTRL_FLAG_GET_DEF
```

6.111.1.9 CONTRAST_CTL_FLAG

```
#define CONTRAST_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.10 CPB_SIZE_CTL_FLAG

```
#define CPB_SIZE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.11 CT_CTRL_NUM

```
#define CT_CTRL_NUM 22
```

6.111.1.12 DEVICE_POWER_MODE_FLAG

```
#define DEVICE_POWER_MODE_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

Kneron Confidential

6.111.1.13 DIGITAL_MULTIPLIER_CTL_FLAG

```
#define DIGITAL_MULTIPLIER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.14 DIGITAL_MULTIPLIER_LIMIT_CTL_FLAG

```
#define DIGITAL_MULTIPLIER_LIMIT_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.15 ERROR_RESILIENCY_CTL_FLAG

```
#define ERROR_RESILIENCY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF
| UVC_CTRL_FLAG_GET_RES
```

6.111.1.16 EU_CTRL_NUM

```
#define EU_CTRL_NUM 20
```

6.111.1.17 EXPOSURE_TIME_ABSOLUTE_CTL_FLAG

```
#define EXPOSURE_TIME_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.18 EXPOSURE_TIME_RELATIVE_CTL_FLAG

```
#define EXPOSURE_TIME_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.19 FOCUS_ABSOLUTE_CTL_FLAG

```
#define FOCUS_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

Kneron Confidential

6.111.1.20 FOCUS_AUTO_CTL_FLAG

```
#define FOCUS_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
| UVC_CTRL_FLAG_GET_DEF
```

6.111.1.21 FOCUS_RELATIVE_CTL_FLAG

```
#define FOCUS_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.22 FOCUS_SIMPLE_CTL_FLAG

```
#define FOCUS_SIMPLE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
| UVC_CTRL_FLAG_GET_DEF
```

6.111.1.23 GAIN_CTL_FLAG

```
#define GAIN_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.24 GAMMA_CTL_FLAG

```
#define GAMMA_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.25 HOST_GET_CLASS_INTF

```
#define HOST_GET_CLASS_INTF 0xA1
```

6.111.1.26 HOST_SET_CLASS_INTF

```
#define HOST_SET_CLASS_INTF 0x21
```

Kneron Confidential

6.111.1.27 HUE_AUTO_CTL_FLAG

```
#define HUE_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
| UVC_CTRL_FLAG_GET_DEF
```

6.111.1.28 HUE_CTL_FLAG

```
#define HUE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.29 IRIS_ABSOLUTE_CTL_FLAG

```
#define IRIS_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.30 IRIS_RELATIVE_CTL_FLAG

```
#define IRIS_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.31 LEVEL_IDC_CTL_FLAG

```
#define LEVEL_IDC_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF
```

6.111.1.32 LONG_TERM_BUFFER_CTL_FLAG

```
#define LONG_TERM_BUFFER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF
| UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.33 LTR_VALIDATION_CTL_FLAG

```
#define LTR_VALIDATION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF
| UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

Kneron Confidential

6.111.1.34 MINIMUM_FRAME_INTERVAL_CTL_FLAG

```
#define MINIMUM_FRAME_INTERVAL_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.35 PANTILT_ABSOLUTE_CTL_FLAG

```
#define PANTILT_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.36 PANTILT_RELATIVE_CTL_FLAG

```
#define PANTILT_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.37 PEAK_BIT_RATE_CTL_FLAG

```
#define PEAK_BIT_RATE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.38 PICTURE_LONG_TERM_REFERENCE_CTL_FLAG

```
#define PICTURE_LONG_TERM_REFERENCE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR |
UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.39 POWER_LINE_FREQUENCY_CTL_FLAG

```
#define POWER_LINE_FREQUENCY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
| UVC_CTRL_FLAG_GET_DEF
```

6.111.1.40 PRIORITY_ID_CTL_FLAG

```
#define PRIORITY_ID_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
| UVC_CTRL_FLAG_GET_LEN
```

Kneron Confidential

6.111.1.41 PRIVACY_CTL_FLAG

```
#define PRIVACY_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.42 PROFILE_AND_TOOLSET_CTL_FLAG

```
#define PROFILE_AND_TOOLSET_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF
| UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.43 PU_CTRL_NUM

```
#define PU_CTRL_NUM 19
```

6.111.1.44 QP_RANGE_CTL_FLAG

```
#define QP_RANGE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
| UVC_CTRL_FLAG_GET_RES
```

6.111.1.45 QUANTIZATION_PARAMETER_CTL_FLAG

```
#define QUANTIZATION_PARAMETER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
| UVC_CTRL_FLAG_GET_RES
```

6.111.1.46 RATE_CONTROL_MODE_CTL_FLAG

```
#define RATE_CONTROL_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF
| UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.47 REGION_OF_INTEREST_CTL_FLAG

```
#define REGION_OF_INTEREST_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MIN
| UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_DEF
```

Kneron Confidential

6.111.1.48 REQUEST_ERROR_CODE_FLAG

```
#define REQUEST_ERROR_CODE_FLAG UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.49 ROLL_ABSOLUTE_CTL_FLAG

```
#define ROLL_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.50 ROLL_RELATIVE_CTL_FLAG

```
#define ROLL_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.51 SATURATION_CTL_FLAG

```
#define SATURATION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.52 SCANNING_MODE_CTL_FLAG

```
#define SCANNING_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.53 SEI_MESSAGE_CTL_FLAG

```
#define SEI_MESSAGE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_DEF
| UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_INFO
```

6.111.1.54 SELECT_LAYER_CTL_FLAG

```
#define SELECT_LAYER_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

Kneron Confidential

6.111.1.55 SHARPNESS_CTL_FLAG

```
#define SHARPNESS_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.56 SLICE_MODE_CTL_FLAG

```
#define SLICE_MODE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.57 START_OR_STOP_LAYER_VIEW_CTL_FLAG

```
#define START_OR_STOP_LAYER_VIEW_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_INF
| UVC_CTRL_FLAG_GET_LEN
```

6.111.1.58 SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME_CTL_FLAG

```
#define SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.59 UVC_CTRL_FLAG_AUTO_UPDATE

```
#define UVC_CTRL_FLAG_AUTO_UPDATE (1 << 10)
```

6.111.1.60 UVC_CTRL_FLAG_GET_CUR

```
#define UVC_CTRL_FLAG_GET_CUR (1 << 1)
```

6.111.1.61 UVC_CTRL_FLAG_GET_DEF

```
#define UVC_CTRL_FLAG_GET_DEF (1 << 6)
```

Kneron Confidential

6.111.1.62 UVC_CTRL_FLAG_GET_INFO

```
#define UVC_CTRL_FLAG_GET_INFO (1 << 5)
```

6.111.1.63 UVC_CTRL_FLAG_GET_LEN

```
#define UVC_CTRL_FLAG_GET_LEN (1 << 7)
```

6.111.1.64 UVC_CTRL_FLAG_GET_MAX

```
#define UVC_CTRL_FLAG_GET_MAX (1 << 3)
```

6.111.1.65 UVC_CTRL_FLAG_GET_MIN

```
#define UVC_CTRL_FLAG_GET_MIN (1 << 2)
```

6.111.1.66 UVC_CTRL_FLAG_GET_RANGE

```
#define UVC_CTRL_FLAG_GET_RANGE
```

Value:

```
(UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MIN | \
     UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_RES | \
     UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO)
```

6.111.1.67 UVC_CTRL_FLAG_GET_RES

```
#define UVC_CTRL_FLAG_GET_RES (1 << 4)
```

6.111.1.68 UVC_CTRL_FLAG_RESTORE

```
#define UVC_CTRL_FLAG_RESTORE (1 << 9)
```

Kneron Confidential

6.111.1.69 UVC_CTRL_FLAG_SET_CUR

```
#define UVC_CTRL_FLAG_SET_CUR (1 << 0)
```

6.111.1.70 VC_CTRL_NUM

```
#define VC_CTRL_NUM 2
```

6.111.1.71 VIDEO_RESOLUTION_CTL_FLAG

```
#define VIDEO_RESOLUTION_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MAX
| UVC_CTRL_FLAG_GET_MIN | UVC_CTRL_FLAG_GET_DEF | UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_LEN
```

6.111.1.72 WHITE_BALANCE_COMPONENT_AUTO_CTL_FLAG

```
#define WHITE_BALANCE_COMPONENT_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR |
UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
```

6.111.1.73 WHITE_BALANCE_COMPONENT_CTL_FLAG

```
#define WHITE_BALANCE_COMPONENT_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.74 WHITE_BALANCE_TEMPERATURE_AUTO_CTL_FLAG

```
#define WHITE_BALANCE_TEMPERATURE_AUTO_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR |
UVC_CTRL_FLAG_GET_INFO | UVC_CTRL_FLAG_GET_DEF
```

6.111.1.75 WHITE_BALANCE_TEMPERATURE_CTL_FLAG

```
#define WHITE_BALANCE_TEMPERATURE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

Kneron Confidential

6.111.1.76 WINDOW_CTL_FLAG

```
#define WINDOW_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_CUR | UVC_CTRL_FLAG_GET_MIN |
UVC_CTRL_FLAG_GET_MAX | UVC_CTRL_FLAG_GET_DEF
```

6.111.1.77 ZOOM_ABSOLUTE_CTL_FLAG

```
#define ZOOM_ABSOLUTE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.1.78 ZOOM_RELATIVE_CTL_FLAG

```
#define ZOOM_RELATIVE_CTL_FLAG UVC_CTRL_FLAG_SET_CUR | UVC_CTRL_FLAG_GET_RANGE
```

6.111.2 Function Documentation

6.111.2.1 __attribute__()

```
struct uvc_usb_ctrlreq __attribute__ (
            (packed)  )
```

6.111.2.2 set_ct_ctrl_flag()

```
void set_ct_ctrl_flag (
            struct ctrl_info * info )
```

6.111.2.3 set_pu_ctrl_flag()

```
void set_pu_ctrl_flag (
            struct ctrl_info * info )
```

Kneron Confidential

6.111.2.4 uvc_ctrl_get()

```
int uvc_ctrl_get (
            struct ctrl_info * ctrl,
            struct uvc_device * dev,
            uint8_t req_num )
```

6.111.2.5 uvc_ctrl_init_device()

```
int uvc_ctrl_init_device (
            struct uvc_device * dev )
```

6.111.2.6 uvc_ctrl_set()

```
int uvc_ctrl_set (
            struct uvc_device * udev,
            struct ctrl_info * info )
```

6.111.2.7 uvc_init_device_ctrl()

```
int uvc_init_device_ctrl (
            struct uvc_device * udev )
```

6.111.2.8 uvc_init_stream_ctl()

```
int uvc_init_stream_ctl (
            struct uvc_device * udev,
            struct uvc_streaming * stream )
```

6.111.2.9 uvc_send_ctl()

```
int uvc_send_ctl (
            struct uvc_device * udev,
            uint32_t cid,
            uint16_t req,
            uint8_t * para,
            uint8_t len )
```

Kneron Confidential

6.111.2.10 uvc_vs_probe()

```
int uvc_vs_probe (
            struct uvc_streaming * stream,
            struct uvc_streaming_control_data * probe )
```

6.111.3 Variable Documentation

6.111.3.1 __attribute__

```
struct eu_select_layer __attribute__
```

6.111.3.2 bRequest

```
uint8_t bRequest
```

6.111.3.3 bRequestType

```
uint8_t bRequestType
```

6.111.3.4 data

```
uint8_t* data
```

6.111.3.5 wIndex

```
uint16_t wIndex
```

6.111.3.6 wLength

```
uint16_t wLength
```

Kneron Confidential

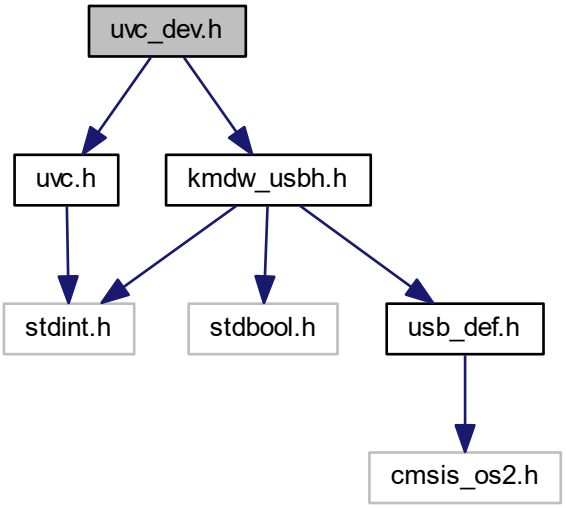
6.111.3.7 wValue

```
uint16_t wValue
```

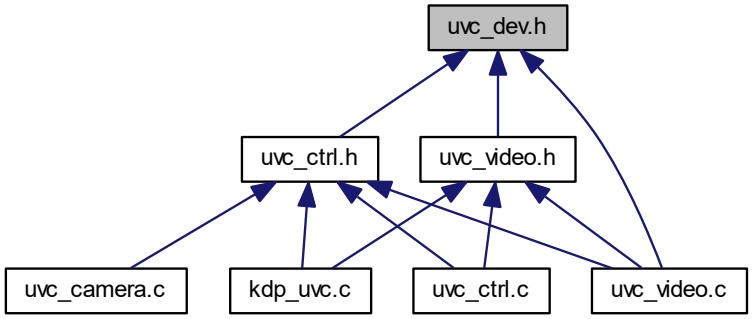
6.112 uvc_dev.h File Reference

```
#include <uvc.h>
#include <kmdw_usbh.h>
```

Include dependency graph for uvc_dev.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Data Structures

- struct uvc_format_desc
- struct ctrl_info
- struct uvc_ct
- struct uvc_it
- struct uvc_ot
- struct uvc_su
- struct uvc_pu
- struct uvc_xu
- struct uvc_eu
- struct uvc_control
- struct uvc_streaming_header
- struct uvc_decode_op
- struct uvc_color_mat_desc
- struct uvc_vs_format
- struct cont_frame_intervals
- struct uvc_frame
- struct uvc_format
- struct uvc_vs_alt_intf
- struct uvc_vs_ctl_data
- struct ctrl_vs_info
- struct uvc_streaming
- struct uvc_vc_int_ep
- struct uvc_device

Macros

- #define UVC_TERM_INPUT 0x0000
- #define UVC_TERM_OUTPUT 0x8000
- #define UVC_TERM_DIRECTION(term) ((term)->type & 0x8000)
- #define UVC_ENTITY_TYPE(entity) ((entity)->type & 0x7fff)
- #define UVC_ENTITY_IS_UNIT(entity) (((entity)->type & 0xff00) == 0)
- #define UVC_ENTITY_IS_TERM(entity) (((entity)->type & 0xff00) != 0)
- #define UVC_ENTITY_IS_ITERM(entity)
- #define UVC_ENTITY_IS_OTERM(entity)
- #define UVC_ET_IT UVC_TERM_INPUT
- #define UVC_ET_OT UVC_TERM_OUTPUT
- #define UVC_ET_CT 0x1
- #define UVC_ET_PU 0x2
- #define UVC_ET_SU 0x3
- #define UVC_ET_EU 0x4
- #define UVC_ET_XU 0x5
- #define UVC_DEV_INF 0x6
- #define UVC_DEF 0
- #define UVC_CUR 1
- #define UVC_MIN 2
- #define UVC_MAX 3
- #define UVC_RES 4
- #define DRIVER_VERSION "1.0.1"
- #define UVC_CTRL_CONTROL_TIMEOUT 30
- #define UVC_CTRL_STREAMING_TIMEOUT 5000
- #define UVC_FMT_FLAG_COMPRESSED 0x00000001
- #define UVC_FMT_FLAG_STREAM 0x00000002

Kneron Confidential

- #define CAP_SUPPORT_GET 0x0
- #define CAP_SUPPORT_SET 0x1
- #define STATE_DISABLED_AUTO_MODE_STATE 0x2
- #define CAP_SUPPORT_AUTOUPDATE_CTRL 0x4
- #define CAP_SUPPORT_ASYNC_CTRL 0x8
- #define STATE_DISABLED_INCOMP_COMMIT_STATE 0x10
- #define UVC_ENTITY_FLAG_DEFAULT (1 << 0)
- #define PARAM_ARRAY_SIZE 5

Functions

- struct uvc_color_mat_desc __attribute__ ((__packed__))

  8-byte setup packet struct
- struct uvc_entity * uvc_entity_by_id (struct uvc_device *dev, int id)
- struct uvc_device * video_dev (const char *pname)
- int kdp_uvc_parse_config (uint8_t *dev)
- void usbh_mdw_uvc_init (void)
- int kdp_uvc_init (void)

Variables

- uint8_t bColorPrimaries
- uint8_t bTransferCharacteristics
- uint8_t bMatrixCoefficients
- struct uvc_vs_format __attribute__

6.112.1 Macro Definition Documentation

6.112.1.1 CAP_SUPPORT_ASYNC_CTRL

```
#define CAP_SUPPORT_ASYNC_CTRL 0x8
```

6.112.1.2 CAP_SUPPORT_AUTOUPDATE_CTRL

```
#define CAP_SUPPORT_AUTOUPDATE_CTRL 0x4
```

6.112.1.3 CAP_SUPPORT_GET

```
#define CAP_SUPPORT_GET 0x0
```

Kneron Confidential

6.112.1.4 CAP_SUPPORT_SET

```
#define CAP_SUPPORT_SET 0x1
```

6.112.1.5 DRIVER_VERSION

```
#define DRIVER_VERSION "1.0.1"
```

6.112.1.6 PARAM_ARRAY_SIZE

```
#define PARAM_ARRAY_SIZE 5
```

6.112.1.7 STATE_DISABLED_AUTO_MODE_STATE

```
#define STATE_DISABLED_AUTO_MODE_STATE 0x2
```

6.112.1.8 STATE_DISABLED_INCOMP_COMMIT_STATE

```
#define STATE_DISABLED_INCOMP_COMMIT_STATE 0x10
```

6.112.1.9 UVC_CTRL_CONTROL_TIMEOUT

```
#define UVC_CTRL_CONTROL_TIMEOUT 30
```

6.112.1.10 UVC_CTRL_STREAMING_TIMEOUT

```
#define UVC_CTRL_STREAMING_TIMEOUT 5000
```

6.112.1.11 UVC_CUR

```
#define UVC_CUR 1
```

Kneron Confidential

6.112.1.12 UVC_DEF

```
#define UVC_DEF 0
```

6.112.1.13 UVC_DEV_INF

```
#define UVC_DEV_INF 0x6
```

6.112.1.14 UVC_ENTITY_FLAG_DEFAULT

```
#define UVC_ENTITY_FLAG_DEFAULT (1 << 0)
```

6.112.1.15 UVC_ENTITY_IS_ITERM

```
#define UVC_ENTITY_IS_ITERM(
               entity )
```

Value:

```
(UVC_ENTITY_IS_TERM(entity) && \
    ((entity)->type & 0x8000) == UVC_TERM_INPUT)
```

6.112.1.16 UVC_ENTITY_IS_OTERM

```
#define UVC_ENTITY_IS_OTERM(
               entity )
```

Value:

```
(UVC_ENTITY_IS_TERM(entity) && \
    ((entity)->type & 0x8000) == UVC_TERM_OUTPUT)
```

6.112.1.17 UVC_ENTITY_IS_TERM

```
#define UVC_ENTITY_IS_TERM(
               entity ) (((entity)->type & 0xff00) != 0)
```

Kneron Confidential

6.112.1.18 UVC_ENTITY_IS_UNIT

```
#define UVC_ENTITY_IS_UNIT(
              entity ) (((entity)->type & 0xff00) == 0)
```

6.112.1.19 UVC_ENTITY_TYPE

```
#define UVC_ENTITY_TYPE(
              entity ) ((entity)->type & 0x7fff)
```

6.112.1.20 UVC_ET_CT

```
#define UVC_ET_CT 0x1
```

6.112.1.21 UVC_ET_EU

```
#define UVC_ET_EU 0x4
```

6.112.1.22 UVC_ET_IT

```
#define UVC_ET_IT UVC_TERM_INPUT
```

6.112.1.23 UVC_ET_OT

```
#define UVC_ET_OT UVC_TERM_OUTPUT
```

6.112.1.24 UVC_ET_PU

```
#define UVC_ET_PU 0x2
```

Kneron Confidential

6.112.1.25 UVC_ET_SU

```
#define UVC_ET_SU 0x3
```

6.112.1.26 UVC_ET_XU

```
#define UVC_ET_XU 0x5
```

6.112.1.27 UVC_FMT_FLAG_COMPRESSED

```
#define UVC_FMT_FLAG_COMPRESSED 0x00000001
```

6.112.1.28 UVC_FMT_FLAG_STREAM

```
#define UVC_FMT_FLAG_STREAM 0x00000002
```

6.112.1.29 UVC_MAX

```
#define UVC_MAX 3
```

6.112.1.30 UVC_MIN

```
#define UVC_MIN 2
```

6.112.1.31 UVC_RES

```
#define UVC_RES 4
```

6.112.1.32 UVC_TERM_DIRECTION

```
#define UVC_TERM_DIRECTION(
             term ) ((term)->type & 0x8000)
```

Kneron Confidential

6.112.1.33 UVC_TERM_INPUT

```
#define UVC_TERM_INPUT 0x0000
```

6.112.1.34 UVC_TERM_OUTPUT

```
#define UVC_TERM_OUTPUT 0x8000
```

6.112.2 Function Documentation

6.112.2.1 kdp_uvc_init()

```
int kdp_uvc_init (
             void  )
```

6.112.2.2 kdp_uvc_parse_config()

```
int kdp_uvc_parse_config (
             uint8_t * dev )
```

6.112.2.3 usbh_mdw_uvc_init()

```
void usbh_mdw_uvc_init (
             void  )
```

Here is the caller graph for this function:

Kneron Confidential

6.112.2.4 uvc_entity_by_id()

```
struct uvc_entity* uvc_entity_by_id (
            struct uvc_device * dev,
            int id )
```

6.112.2.5 video_dev()

```
struct uvc_device* video_dev (
            const char * pname )
```

6.112.3 Variable Documentation

6.112.3.1 __attribute__

```
struct uvc_vs_format __attribute__
```

6.112.3.2 bColorPrimaries

```
uint8_t bColorPrimaries
```

6.112.3.3 bMatrixCoefficients

```
uint8_t bMatrixCoefficients
```

6.112.3.4 bTransferCharacteristics

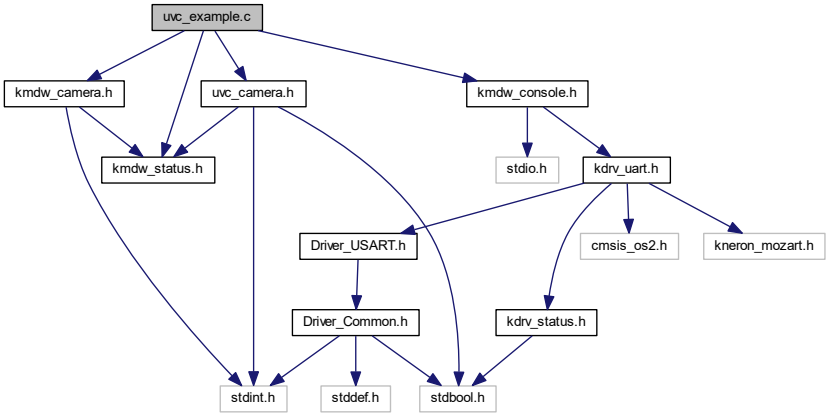
```
uint8_t bTransferCharacteristics
```

Kneron Confidential

6.113 uvc_example.c File Reference

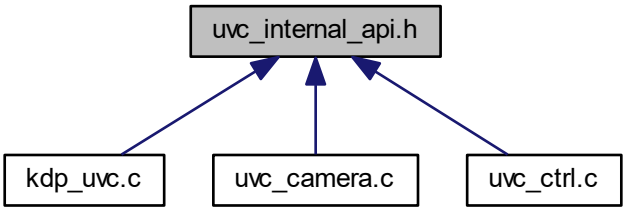
```
#include "kmdw_camera.h"
#include "kmdw_console.h"
#include "uvc_camera.h"
#include "kmdw_status.h"
```

Include dependency graph for uvc_example.c:



6.114 uvc_internal_api.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define VIDEO_MAX_FRAME 32
- #define ET_POS 28
- #define UVC_CID_DEV_INF_CLASS_BASE (UVC_DEV_INF << ET_POS)
- #define DEVICE_POWER_MODE_LEN 0x1
- #define DEVICE_POWER_MODE_LEN 0x1
- #define UVC_CID_DEVICE_POWER_MODE UVC_CID_DEV_INF_CLASS_BASE | UVC_VC_VIDEO_POWER_MODE_CON

Kneron Confidential

- #define POWER_MODECTL_FULL_POWER 0000b
- #define POWER_MODECTL_DEVICE_DEP_POWER 0001b
- #define POWER_MODECTL_DEVICE_DEP_POWER_SUPPORTED 0x10
- #define POWER_MODECTL_POWER_BY_USB 0x20
- #define POWER_MODECTL_POWER_BY_BATTERY 0x40
- #define POWER_MODECTL_POWER_BY_BY_AC 0x80
- #define DEVICE_REQUEST_ERROR_CODE_LEN 0x1
- #define UVC_CID_REQUEST_ERROR_CODE UVC_CID_DEV_INF_CLASS_BASE | UVC_VC_REQUEST_ERROR_CODE_ << 24
- #define REQUEST_ERROR_CTL
- #define UVC_CID_CAMERA_CLASS_BASE (UVC_ET_CT << ET_POS)
- #define SCANNING_MODE 0x1
- #define AUTO_EXPOSURE_MODE 0x2
- #define AUTO_EXPOSURE_PRIORITY 0x4
- #define EXPOSURE_TIME_ABSOLUTE 0x8
- #define EXPOSURE_TIME_RELATIVE 0x10
- #define FOCUS_ABSOLUTE 0x20
- #define FOCUS_RELATIVE 0x40
- #define IRIS_ABSOLUTE 0x80
- #define IRIS_RELATIVE 0x100
- #define ZOOM_ABSOLUTE 0x200
- #define ZOOM_RELATIVE 0x400
- #define PANTILT_ABSOLUTE 0x800
- #define PANTILT_RELATIVE 0x1000
- #define ROLL_ABSOLUTE 0x2000
- #define ROLL_RELATIVE 0x4000
- #define FOCUS_AUTO 0x20000
- #define PRIVACY 0x40000
- #define FOCUS_SIMPLE 0x80000
- #define WINDOW 0x100000
- #define REGION_OF_INTEREST 0x200000
- #define SCANNING_MODE_LEN 0x1
- #define AUTO_EXPOSURE_MODE_LEN 0x1
- #define AUTO_EXPOSURE_PRIORITY_LEN 0x1
- #define EXPOSURE_TIME_ABSOLUTE_LEN 0x4
- #define EXPOSURE_TIME_RELATIVE_LEN 0x1
- #define FOCUS_ABSOLUTE_LEN 0x2
- #define FOCUS_RELATIVE_LEN 0x2
- #define IRIS_ABSOLUTE_LEN 0x2
- #define IRIS_RELATIVE_LEN 0x2
- #define ZOOM_ABSOLUTE_LEN 0x2
- #define ZOOM_RELATIVE_LEN 0x3
- #define PANTILT_ABSOLUTE_LEN 0x8
- #define PANTILT_RELATIVE_LEN 0x4
- #define ROLL_ABSOLUTE_LEN 0x2
- #define ROLL_RELATIVE_LEN 0x2
- #define FOCUS_AUTO_LEN 0x1
- #define PRIVACY_LEN 0x1
- #define FOCUS_SIMPLE_LEN 0x1
- #define WINDOW_LEN 0xc
- #define REGION_OF_INTEREST_LEN 0xa
- #define UVC_CID_SCANNING_MODE UVC_CID_CAMERA_CLASS_BASE | SCANNING_MODE
- #define UVC_CID_AUTO_EXPOSURE_MODE UVC_CID_CAMERA_CLASS_BASE | AUTO_EXPOSURE_MODE
- #define UVC_CID_AUTO_EXPOSURE_PRIORITY UVC_CID_CAMERA_CLASS_BASE | AUTO_EXPOSURE_PRIORITY
- #define UVC_CID_EXPOSURE_TIME_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | EXPOSURE_TIME_ABSOLUTE

Kneron Confidential

- #define UVC_CID_EXPOSURE_TIME_RELATIVE UVC_CID_CAMERA_CLASS_BASE | EXPOSURE_TIME_RELATIVE
- #define UVC_CID_FOCUS_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | FOCUS_ABSOLUTE
- #define UVC_CID_FOCUS_RELATIVE UVC_CID_CAMERA_CLASS_BASE | FOCUS_RELATIVE
- #define UVC_CID_IRIS_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | IRIS_ABSOLUTE
- #define UVC_CID_IRIS_RELATIVE UVC_CID_CAMERA_CLASS_BASE | IRIS_RELATIVE
- #define UVC_CID_ZOOM_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | ZOOM_ABSOLUTE
- #define UVC_CID_ZOOM_RELATIVE UVC_CID_CAMERA_CLASS_BASE | ZOOM_RELATIVE
- #define UVC_CID_PANTILT_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | PANTILT_ABSOLUTE
- #define UVC_CID_PANTILT_RELATIVE UVC_CID_CAMERA_CLASS_BASE | PANTILT_RELATIVE
- #define UVC_CID_ROLL_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | ROLL_ABSOLUTE
- #define UVC_CID_ROLL_RELATIVE UVC_CID_CAMERA_CLASS_BASE | ROLL_RELATIVE
- #define UVC_CID_FOCUS_AUTO UVC_CID_CAMERA_CLASS_BASE | FOCUS_AUTO
- #define UVC_CID_PRIVACY UVC_CID_CAMERA_CLASS_BASE | PRIVACY
- #define UVC_CID_FOCUS_SIMPLE UVC_CID_CAMERA_CLASS_BASE | FOCUS_SIMPLE
- #define UVC_CID_WINDOW UVC_CID_CAMERA_CLASS_BASE | WINDOW
- #define UVC_CID_REGION_OF_INTEREST UVC_CID_CAMERA_CLASS_BASE | REGION_OF_INTEREST
- #define SCANNING_MODE_CTL_INTERLACED 0x0
- #define SCANNING_MODE_CTL_PROGRESSIVE 0x1
- #define AUTO_EXPOSURE_MODE_CTL_MODE_MANUAL 0x1
- #define AUTO_EXPOSURE_MODE_CTL_MODE_AUTO 0x2
- #define AUTO_EXPOSURE_MODE_CTL_MODE_SHUTTER_PRIO 0x4
- #define AUTO_EXPOSURE_MODE_CTL_MODE_APERTURE_PRIO 0x8
- #define AUTO_EXPOSURE_PRI_CTL_FRAMERAT_CONST 0x0
- #define AUTO_EXPOSURE_PRI_CTL_FRAMERAT_VARIED 0x1
- #define EXPOSURE_TIME_REL_CTL_DEF 0x0
- #define EXPOSURE_TIME_REL_CTL_INC_BY_1STEP 0x1
- #define EXPOSURE_TIME_REL_CTL_DEC_BY_1_STEP 0xFF
- #define FOCUS_REL_CTL_REL_STOP 0x0
- #define FOCUS_REL_CTL_REL_NEAR_DIR 0x1
- #define FOCUS_REL_CTL_REL_INFINITE_DIR 0xFF
- #define FOCUS_SIMPLE_CTL_FULL_RANGE 0x00
- #define FOCUS_SIMPLE_CTL_MACRO_LESS_THAN_0_3M 0x01
- #define FOCUS_SIMPLE_CTL_PEOPLE_0_3M_TO_3M 0x02
- #define FOCUS_SIMPLE_CTL_SCENE_3M_TO_INFINITY 0x03
- #define IRIS_REL_CTL_REL_DEF 0x0
- #define IRIS_REL_CTL_REL_OPEN_BY_1STEP 0x1
- #define IRIS_REL_CTL_REL_CLOSE_BY_1STEP 0xFF
- #define ZOOM_REL_CTL_ZOOM_STOP 0x0
- #define ZOOM_REL_CTL_ZOOM_TO_TELEPHOTO_DIR 0x1
- #define ZOOM_REL_CTL_ZOOM_TO_WIDEANGLE_DIR 0xFF
- #define ZOOM_REL_CTL_DIGITALZOOM_OFF 0x0
- #define ZOOM_REL_CTL_DIGITALZOOM_ON 0x1
- #define PANTILT_REL_CTL_STOP 0x0
- #define PANTILT_REL_CTL_PLANE_UP 0x1
- #define PANTILT_REL_CTL_PLANE_DOWN 0xFF
- #define ROLL_RELATIVE_CTL_STOP 0x0
- #define ROLL_RELATIVE_CTL_CLOCKWISE_ROTATION 0x1
- #define ROLL_RELATIVE_CTL_COUNTER_CLOCKWISE_ROTATION 0xFF
- #define PRIVACY_CTL_OPEN 0x0
- #define PRIVACY_CTL_CLOSE 0x1
- #define WINDOWS_CTL_STEPUNIT_VIDEOFRAME 0x1
- #define WINDOWS_CTL_STEPUNIT_MILLISECONDS 0x2
- #define ROI_CTL_AUTOCTL_EXPOSURE 0x1
- #define ROI_CTL_AUTOCTL_IRIS 0x2
- #define ROI_CTL_AUTOCTL_WHITE_BALANCE 0x4

Kneron Confidential

- #define ROI_CTL_AUTOCTL_FOCUS 0x8
- #define ROI_CTL_AUTOCTL_FACE_DETECT 0x10
- #define ROI_CTL_AUTOCTL_AUTO_DETECT_TRACK 0x20
- #define ROI_CTL_AUTOCTL_IMAGE_STABILIZATION 0x40
- #define ROI_CTL_AUTOCTL_HIGHER_QUALITY 0x80
- #define BRIGHTNESS 0x1
- #define CONTRAST 0x2
- #define HUE 0x4
- #define SATURATION 0x8
- #define SHARPNESS 0x10
- #define GAMMA 0x20
- #define WHITE_BALANCE_TEMPERATURE 0x40
- #define WHITE_BALANCE_COMPONENT 0x80
- #define BACKLIGHT_COMPENSATION 0x100
- #define GAIN 0x200
- #define POWER_LINE_FREQUENCY 0x400
- #define HUE_AUTO 0x800
- #define WHITE_BALANCE_TEMPERATURE_AUTO 0x1000
- #define WHITE_BALANCE_COMPONENT_AUTO 0x2000
- #define DIGITAL_MULTIPLIER 0x4000
- #define DIGITAL_MULTIPLIER_LIMIT 0x8000
- #define ANALOG_VIDEO_STANDARD 0x10000
- #define ANALOG_VIDEO_LOCK_STATUS 0x20000
- #define CONTRAST_AUTO 0x40000
- #define BRIGHTNESS_LEN 0x2
- #define CONTRAST_LEN 0x2
- #define HUE_LEN 0x2
- #define SATURATION_LEN 0x2
- #define SHARPNESS_LEN 0x2
- #define GAMMA_LEN 0x2
- #define WHITE_BALANCE_TEMPERATURE_LEN 0x2
- #define WHITE_BALANCE_COMPONENT_LEN 0x2
- #define BACKLIGHT_COMPENSATION_LEN 0x2
- #define GAIN_LEN 0x2
- #define POWER_LINE_FREQUENCY_LEN 0x1
- #define HUE_AUTO_LEN 0x1
- #define WHITE_BALANCE_TEMPERATURE_AUTO_LEN 0x1
- #define WHITE_BALANCE_COMPONENT_AUTO_LEN 0x1
- #define DIGITAL_MULTIPLIER_LEN 0x2
- #define DIGITAL_MULTIPLIER_LIMIT_LEN 0x2
- #define ANALOG_VIDEO_STANDARD_LEN 0x1
- #define ANALOG_VIDEO_LOCK_STATUS_LEN 0x1
- #define CONTRAST_AUTO_LEN 0x1
- #define UVC_CID_PU_CLASS_BASE (UVC_ET_PU << ET_POS)
- #define UVC_CID_BRIGHTNESS UVC_CID_PU_CLASS_BASE | BRIGHTNESS
- #define UVC_CID_CONTRAST UVC_CID_PU_CLASS_BASE | CONTRAST
- #define UVC_CID_HUE UVC_CID_PU_CLASS_BASE | HUE
- #define UVC_CID_SATURATION UVC_CID_PU_CLASS_BASE | SATURATION
- #define UVC_CID_SHARPNESS UVC_CID_PU_CLASS_BASE | SHARPNESS
- #define UVC_CID_GAMMA UVC_CID_PU_CLASS_BASE | GAMMA
- #define UVC_CID_WHITE_BALANCE_TEMPERATURE UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_TEMPERATURE
- #define UVC_CID_WHITE_BALANCE_COMPONENT UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_COMPONENT
- #define UVC_CID_BACKLIGHT_COMPENSATION UVC_CID_PU_CLASS_BASE | BACKLIGHT_COMPENSATION
- #define UVC_CID_GAIN UVC_CID_PU_CLASS_BASE | GAIN
- #define UVC_CID_POWER_LINE_FREQUENCY UVC_CID_PU_CLASS_BASE | POWER_LINE_FREQUENCY

Kneron Confidential

- #define UVC_CID_HUE_AUTO UVC_CID_PU_CLASS_BASE | HUE_AUTO
- #define UVC_CID_WHITE_BALANCE_TEMPERATURE_AUTO UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_TEMPER
- #define UVC_CID_WHITE_BALANCE_COMPONENT_AUTO UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_COMPONE
- #define UVC_CID_DIGITAL_MULTIPLIER UVC_CID_PU_CLASS_BASE | DIGITAL_MULTIPLIER
- #define UVC_CID_DIGITAL_MULTIPLIER_LIMIT UVC_CID_PU_CLASS_BASE | DIGITAL_MULTIPLIER_LIMIT
- #define UVC_CID_ANALOG_VIDEO_STANDARD UVC_CID_PU_CLASS_BASE | ANALOG_VIDEO_STANDARD
- #define UVC_CID_ANALOG_VIDEO_LOCK_STATUS UVC_CID_PU_CLASS_BASE | ANALOG_VIDEO_LOCK_STATUS
- #define UVC_CID_CONTRAST_AUTO UVC_CID_PU_CLASS_BASE | CONTRAST_AUTO
- #define BACKLIGHT_COMPENSATION_CTL_DISABLED 0x0
- #define CONTRAST_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
- #define POWER_LINE_FREQUENCY_CTL_DISABLED 0
- #define POWER_LINE_FREQUENCY_CTL_50HZ 1
- #define POWER_LINE_FREQUENCY_CTL_60HZ 2
- #define HUE_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
- #define SATURATION_CTL_GRAYSCALE 0x0
- #define GAMMA_CTL_GAMMA_MIN 1
- #define GAMMA_CTL_GAMMA_MAX 500
- #define WHITE_BALANCE_TEMPERATURE_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
- #define WHITE_BALANCE_COMPONENT_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
- #define UVC_CID_EU_CLASS_BASE (UVC_ET_EU << ET_POS)
- #define SELECT_LAYER 0x1
- #define PROFILE_AND_TOOLSET 0x2
- #define VIDEO_RESOLUTION 0x4
- #define MINIMUM_FRAME_INTERVAL 0x8
- #define SLICE_MODE 0x10
- #define RATE_CONTROL_MODE 0x20
- #define AVERAGE_BIT_RATE 0x40
- #define CPB_SIZE 0x80
- #define PEAK_BIT_RATE 0x100
- #define QUANTIZATION_PARAMETER 0x200
- #define SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME 0x400
- #define LONG_TERM_BUFFER 0x800
- #define PICTURE_LONG_TERM_REFERENCE 0x1000
- #define LTR_VALIDATION 0x2000
- #define LEVEL_IDC 0x4000
- #define SEI_MESSAGE 0x8000
- #define QP_RANGE 0x10000
- #define PRIORITY_ID 0x20000
- #define START_OR_STOP_LAYER_VIEW 0x40000
- #define ERROR_RESILIENCY 0x80000
- #define SELECT_LAYERL_LEN 0x2
- #define PROFILE_AND_TOOLSET_LEN 0x5
- #define VIDEO_RESOLUTION_LEN 0x4
- #define MINIMUM_FRAME_INTERVAL_LEN 0x4
- #define SLICE_MODE_LEN 0x4
- #define RATE_CONTROL_MODE_LEN 0x1
- #define AVERAGE_BIT_RATE_LEN 0x4
- #define CPB_SIZE_LEN 0x4
- #define PEAK_BIT_RATE_LEN 0x4
- #define QUANTIZATION_PARAMETER_LEN 0x6
- #define SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME_LEN 0x4
- #define LONG_TERM_BUFFER_LEN 0x2
- #define PICTURE_LONG_TERM_REFERENCE_LEN 0x2
- #define LTR_VALIDATION_LEN 0x2
- #define LEVEL_IDC_LEN 0x1

Kneron Confidential

- #define SEI_MESSAGE_LEN 0x8
- #define QP_RANGE_LEN 0x2
- #define PRIORITY_ID_LEN 0x1
- #define START_OR_STOP_LAYER_VIEW_LEN 0x1
- #define ERROR_RESILIENCY_LEN 0x2
- #define UVC_CID_SELECT_LAYER UVC_CID_EU_CLASS_BASE | SELECT_LAYER
- #define UVC_CID_PROFILE_AND_TOOLSET UVC_CID_EU_CLASS_BASE | PROFILE_AND_TOOLSET
- #define UVC_CID_VIDEO_RESOLUTION UVC_CID_EU_CLASS_BASE | VIDEO_RESOLUTION
- #define UVC_CID_MINIMUM_FRAME_INTERVAL UVC_CID_EU_CLASS_BASE | MINIMUM_FRAME_INTERVAL
- #define UVC_CID_SLICE_MODE UVC_CID_EU_CLASS_BASE | SLICE_MODE
- #define UVC_CID_RATE_CONTROL_MODE UVC_CID_EU_CLASS_BASE | RATE_CONTROL_MODE
- #define UVC_CID_AVERAGE_BIT_RATE UVC_CID_EU_CLASS_BASE | AVERAGE_BIT_RATE
- #define UVC_CID_CPB_SIZE UVC_CID_EU_CLASS_BASE | CPB_SIZE
- #define UVC_CID_PEAK_BIT_RATE UVC_CID_EU_CLASS_BASE | PEAK_BIT_RATE
- #define UVC_CID_QUANTIZATION_PARAMETER UVC_CID_EU_CLASS_BASE | QUANTIZATION_PARAMETER
- #define UVC_CID_SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME UVC_CID_EU_CLASS_BASE | SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME
- #define UVC_CID_LONG_TERM_BUFFER UVC_CID_EU_CLASS_BASE | LONG_TERM_BUFFER
- #define UVC_CID_PICTURE_LONG_TERM_REFERENCE UVC_CID_EU_CLASS_BASE | PICTURE_LONG_TERM_REFER
- #define UVC_CID_LTR_VALIDATION UVC_CID_EU_CLASS_BASE | LTR_VALIDATION
- #define UVC_CID_LEVEL_IDC UVC_CID_EU_CLASS_BASE | LEVEL_IDC
- #define UVC_CID_SEI_MESSAGE UVC_CID_EU_CLASS_BASE | SEI_MESSAGE
- #define UVC_CID_QP_RANGE UVC_CID_EU_CLASS_BASE | QP_RANGE
- #define UVC_CID_PRIORITY_ID UVC_CID_EU_CLASS_BASE | PRIORITY_ID
- #define UVC_CID_START_OR_STOP_LAYER_VIEW UVC_CID_EU_CLASS_BASE | START_OR_STOP_LAYER_VIEW
- #define UVC_CID_ERROR_RESILIENCY UVC_CID_EU_CLASS_BASE | ERROR_RESILIENCY
- #define RATE_CONTROL_MODE_CTL_MODE_VBR 0x1
- #define RATE_CONTROL_MODE_CTL_MODE_CBR 0x2
- #define RATE_CONTROL_MODE_CTL_MODE_Constant_QP 0x3
- #define RATE_CONTROL_MODE_CTL_MODE_GVBR 0x4
- #define RATE_CONTROL_MODE_CTL_MODE_VBRN 0x5
- #define RATE_CONTROL_MODE_CTL_MODE_GVBRN 0x6
- #define PROFILE_AND_TOOLSET_CTL_CONSTRAINED_BASELINE_PROFILE 0x4240
- #define PROFILE_AND_TOOLSET_CTL_BASELINE_PROFILE 0x4200
- #define PROFILE_AND_TOOLSET_CTL_MAIN_PROFILE 0x4D00
- #define PROFILE_AND_TOOLSET_CTL_CONSTRAINED_HIGH_PROFILE 0x640C
- #define PROFILE_AND_TOOLSET_CTL_HIGH_PROFILE 0x6400
- #define PROFILE_AND_TOOLSET_CTL_SCALABLE_CONSTRAINED_BASELINE_PROFILE 0x5304
- #define PROFILE_AND_TOOLSET_CTL_SCALABLE_BASELINE_PROFILE 0x5300
- #define PROFILE_AND_TOOLSET_CTL_SCALABLE_CONSTRAINED_HIGH_PROFILE 0x5604
- #define PROFILE_AND_TOOLSET_CTL_SCALABLE_HIGH_PROFILE 0x5600
- #define PROFILE_AND_TOOLSET_CTL_MULTIVIEW_HIGH_PROFILE 0x7600
- #define PROFILE_AND_TOOLSET_CTL_STEREO_HIGH_PROFILE 0x8000
- #define SLICE_MODE_CTL_MODE_MAX_MBS_PER_SLICE 0x0
- #define SLICE_MODE_CTL_MODE_TARGET_COMPRESSED_SIZE_PER_SLICE 0x1
- #define SLICE_MODE_CTL_MODE_NUMBER_OF_SLICES_PER_FRAME 0x2
- #define SLICE_MODE_CTL_MODE_NUMBER_OF_MACROBLOCK_ROWS_PER_SLICE 0x3
- #define SYNC_AND_LTR_FRAME_CTL_RESET 0x0
- #define SYNC_AND_LTR_FRAME_CTL_IDR 0x1
- #define SYNC_AND_LTR_FRAME_CTL_LONG_TERM_REF_IDR 0x2
- #define SYNC_AND_LTR_FRAME_CTL_NON_IDR_RAND_I_FRAME 0x3
- #define SYNC_AND_LTR_FRAME_CTL_LTR_NON_IDR_RAND_I_FRAME 0x4
- #define SYNC_AND_LTR_FRAME_CTL_LTR_P_FRAME 0x5
- #define LONG_TERM_BUFFER_CTL_TRUSTMODE_NOT_TRUST 0x0
- #define LONG_TERM_BUFFER_CTL_TRUSTMODE_TRUST 0x1

Kneron Confidential

- #define SEI_MESSAGE_CTL_BUFFERING_PERIOD 0x1
- #define SEI_MESSAGE_CTL_PIC_TIMING 0x2
- #define SEI_MESSAGE_CTL_PAN_SCAN_RECT 0x4
- #define SEI_MESSAGE_CTL_FILLER_PAYLOAD 0x8
- #define SEI_MESSAGE_CTL_USER_DATA_REGISTERED_ITU_T_T35 0x10
- #define SEI_MESSAGE_CTL_USER_DATA_UNREGISTERED 0x20
- #define SEI_MESSAGE_CTL_RECOVERY_POINT 0x40
- #define SEI_MESSAGE_CTL_DEC_REF_PIC_MARKING_REPETITION 0x80
- #define SEI_MESSAGE_CTL_SPARE_PIC 0x100
- #define SEI_MESSAGE_CTL_SCENE_INFO 0x200
- #define SEI_MESSAGE_CTL_SUB_SEQ_INFO 0x400
- #define SEI_MESSAGE_CTL_SUB_SEQ_LAYER_CHARACTERISTICS 0x800
- #define SEI_MESSAGE_CTL_SUB_SEQ_CHARACTERISTICS 0x1000
- #define SEI_MESSAGE_CTL_FULL_FRAME_FREEZE 0x2000
- #define SEI_MESSAGE_CTL_FULL_FRAME_FREEZE_RELEASE 0x4000
- #define SEI_MESSAGE_CTL_FULL_FRAME_SNAPSHOT 0x8000
- #define SEI_MESSAGE_CTL_PROGRESSIVE_REFINEMENT_SEGMENT_START 0x10000
- #define SEI_MESSAGE_CTL_PROGRESSIVE_REFINEMENT_SEGMENT_END 0x20000
- #define SEI_MESSAGE_CTL_MOTION_CONSTRAINED_SLICE_GROUP_SET 0x40000
- #define SEI_MESSAGE_CTL_FILM_GRAIN_CHARACTERISTICS 0x80000
- #define SEI_MESSAGE_CTL_DEBLOCKING_FILTER_DISPLAY_PREFERENCE 0x100000
- #define SEI_MESSAGE_CTL_STEREO_VIDEO_INFO 0x200000
- #define SEI_MESSAGE_CTL_POST_FILTER_HINT 0x400000
- #define SEI_MESSAGE_CTL_TONE_MAPPING_INFO 0x800000
- #define SEI_MESSAGE_CTL_SCALABILITY_INFO 0x1000000
- #define SEI_MESSAGE_CTL_SUB_PIC_SCALABLE_LAYER 0x2000000
- #define SEI_MESSAGE_CTL_NON_REQUIRED_LAYER_REP 0x4000000
- #define SEI_MESSAGE_CTL_PRIORITY_LAYER_INFO 0x8000000
- #define SEI_MESSAGE_CTL_LAYERS_NOT_PRESENT 0x10000000
- #define SEI_MESSAGE_CTL_LAYER_DEPENDENCY_CHANGE 0x20000000
- #define SEI_MESSAGE_CTL_SCALABLE_NESTING 0x40000000
- #define SEI_MESSAGE_CTL_BASE_LAYER_TEMPORAL_HRD 0x80000000
- #define SEI_MESSAGE_CTL_QUALITY_LAYER_INTEGRITY_CHECK 0x100000000
- #define SEI_MESSAGE_CTL_REDUNDANT_PIC_PROPERTY 0x200000000
- #define SEI_MESSAGE_CTL_TL0_DEP_REP_INDEX 0x400000000
- #define SEI_MESSAGE_CTL_TL_SWITCHING_POINT 0x800000000
- #define SEI_MESSAGE_CTL_PARALLEL_DECODING_INFO 0x1000000000
- #define SEI_MESSAGE_CTL_MVC_SCALABLE_NESTING 0x2000000000
- #define SEI_MESSAGE_CTL_VIEW_SCALABILITY_INFO 0x4000000000
- #define SEI_MESSAGE_CTL_MULTIVIEW_SCENE_INFO 0x8000000000
- #define SEI_MESSAGE_CTL_MULTIVIEW_ACQUISITION_INFO 0x10000000000
- #define SEI_MESSAGE_CTL_NON_REQUIRED_VIEW_COMPONENT 0x20000000000
- #define SEI_MESSAGE_CTL_VIEW_DEPENDENCY_CHANGE 0x40000000000
- #define SEI_MESSAGE_CTL_OPERATION_POINTS_NOT_PRESENT 0x80000000000
- #define SEI_MESSAGE_CTL_BASE_VIEW_TEMPORAL_HRD 0x100000000000
- #define SEI_MESSAGE_CTL_FRAME_PACKING_ARRANGEMENT 0x200000000000
- #define UVC_CID_XU_CLASS_BASE (UVC_ET_XU << ET_POS)
- #define UVC_VSCID_PROBE_CONTROL UVC_VS_PROBE_CONTROL
- #define UVC_VSCID_COMMIT_CONTROL UVC_VS_COMMIT_CONTROL
- #define UVC_VSCID_STILL_PROBE UVC_VS_STILL_PROBE_CONTROL
- #define UVC_VSCID_STILL_COMMIT UVC_VS_STILL_COMMIT_CONTROL
- #define UVC_VSCID_STILL_IMAGE_TRIGGER UVC_VS_STILL_IMAGE_TRIGGER_CONTROL
- #define UVC_VSCID_STREAM_ERROR_CODE UVC_VS_STREAM_ERROR_CODE_CONTROL
- #define UVC_VSCID_GENERATE_KEY_FRAME UVC_VS_GENERATE_KEY_FRAME_CONTROL
- #define UVC_VSCID_UPDATE_FRAME_SEGMENT UVC_VS_UPDATE_FRAME_SEGMENT_CONTROL

Kneron Confidential

- #define UVC_VSCID_SYNC_DELAY UVC_VS_SYNC_DELAY_CONTROL
- #define PROBE_LEN 48
- #define PROFILE_AND_TOOLSET_LEN 0x5
- #define UVC_VSCID_PROBE UVC_VS_PROBE_CONTROL << 8
- #define UVC_VSCID_COMMIT UVC_VS_COMMIT_CONTROL << 8

6.114.1 Macro Definition Documentation

6.114.1.1 ANALOG_VIDEO_LOCK_STATUS

```
#define ANALOG_VIDEO_LOCK_STATUS 0x20000
```

6.114.1.2 ANALOG_VIDEO_LOCK_STATUS_LEN

```
#define ANALOG_VIDEO_LOCK_STATUS_LEN 0x1
```

6.114.1.3 ANALOG_VIDEO_STANDARD

```
#define ANALOG_VIDEO_STANDARD 0x10000
```

6.114.1.4 ANALOG_VIDEO_STANDARD_LEN

```
#define ANALOG_VIDEO_STANDARD_LEN 0x1
```

6.114.1.5 AUTO_EXPOSURE_MODE

```
#define AUTO_EXPOSURE_MODE 0x2
```

6.114.1.6 AUTO_EXPOSURE_MODE_CTL_MODE_APERTURE_PRIO

```
#define AUTO_EXPOSURE_MODE_CTL_MODE_APERTURE_PRIO 0x8
```

Kneron Confidential

6.114.1.7 AUTO_EXPOSURE_MODE_CTL_MODE_AUTO

```
#define AUTO_EXPOSURE_MODE_CTL_MODE_AUTO 0x2
```

6.114.1.8 AUTO_EXPOSURE_MODE_CTL_MODE_MANUAL

```
#define AUTO_EXPOSURE_MODE_CTL_MODE_MANUAL 0x1
```

6.114.1.9 AUTO_EXPOSURE_MODE_CTL_MODE_SHUTTER_PRIO

```
#define AUTO_EXPOSURE_MODE_CTL_MODE_SHUTTER_PRIO 0x4
```

6.114.1.10 AUTO_EXPOSURE_MODE_LEN

```
#define AUTO_EXPOSURE_MODE_LEN 0x1
```

6.114.1.11 AUTO_EXPOSURE_PRI_CTL_FRAMERAT_CONST

```
#define AUTO_EXPOSURE_PRI_CTL_FRAMERAT_CONST 0x0
```

6.114.1.12 AUTO_EXPOSURE_PRI_CTL_FRAMERAT_VARIED

```
#define AUTO_EXPOSURE_PRI_CTL_FRAMERAT_VARIED 0x1
```

6.114.1.13 AUTO_EXPOSURE_PRIORITY

```
#define AUTO_EXPOSURE_PRIORITY 0x4
```

6.114.1.14 AUTO_EXPOSURE_PRIORITY_LEN

```
#define AUTO_EXPOSURE_PRIORITY_LEN 0x1
```

Kneron Confidential

6.114.1.15 AVERAGE_BIT_RATE

```
#define AVERAGE_BIT_RATE 0x40
```

6.114.1.16 AVERAGE_BIT_RATE_LEN

```
#define AVERAGE_BIT_RATE_LEN 0x4
```

6.114.1.17 BACKLIGHT_COMPENSATION

```
#define BACKLIGHT_COMPENSATION 0x100
```

6.114.1.18 BACKLIGHT_COMPENSATION_CTL_DISABLED

```
#define BACKLIGHT_COMPENSATION_CTL_DISABLED 0x0
```

6.114.1.19 BACKLIGHT_COMPENSATION_LEN

```
#define BACKLIGHT_COMPENSATION_LEN 0x2
```

6.114.1.20 BRIGHTNESS

```
#define BRIGHTNESS 0x1
```

6.114.1.21 BRIGHTNESS_LEN

```
#define BRIGHTNESS_LEN 0x2
```

6.114.1.22 CONTRAST

```
#define CONTRAST 0x2
```

Kneron Confidential

6.114.1.23 CONTRAST_AUTO

```
#define CONTRAST_AUTO 0x40000
```

6.114.1.24 CONTRAST_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED

```
#define CONTRAST_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
```

6.114.1.25 CONTRAST_AUTO_LEN

```
#define CONTRAST_AUTO_LEN 0x1
```

6.114.1.26 CONTRAST_LEN

```
#define CONTRAST_LEN 0x2
```

6.114.1.27 CPB_SIZE

```
#define CPB_SIZE 0x80
```

6.114.1.28 CPB_SIZE_LEN

```
#define CPB_SIZE_LEN 0x4
```

6.114.1.29 DEVICE_POWER_MODE_LEN [1/2]

```
#define DEVICE_POWER_MODE_LEN 0x1
```

6.114.1.30 DEVICE_POWER_MODE_LEN [2/2]

```
#define DEVICE_POWER_MODE_LEN 0x1
```

Kneron Confidential

6.114.1.31 DEVICE_REQUEST_ERROR_CODE_LEN

```
#define DEVICE_REQUEST_ERROR_CODE_LEN 0x1
```

6.114.1.32 DIGITAL_MULTIPLIER

```
#define DIGITAL_MULTIPLIER 0x4000
```

6.114.1.33 DIGITAL_MULTIPLIER_LEN

```
#define DIGITAL_MULTIPLIER_LEN 0x2
```

6.114.1.34 DIGITAL_MULTIPLIER_LIMIT

```
#define DIGITAL_MULTIPLIER_LIMIT 0x8000
```

6.114.1.35 DIGITAL_MULTIPLIER_LIMIT_LEN

```
#define DIGITAL_MULTIPLIER_LIMIT_LEN 0x2
```

6.114.1.36 ERROR_RESILIENCY

```
#define ERROR_RESILIENCY 0x80000
```

6.114.1.37 ERROR_RESILIENCY_LEN

```
#define ERROR_RESILIENCY_LEN 0x2
```

6.114.1.38 ET_POS

```
#define ET_POS 28
```

Kneron Confidential

6.114.1.39 EXPOSURE_TIME_ABSOLUTE

```
#define EXPOSURE_TIME_ABSOLUTE 0x8
```

6.114.1.40 EXPOSURE_TIME_ABSOLUTE_LEN

```
#define EXPOSURE_TIME_ABSOLUTE_LEN 0x4
```

6.114.1.41 EXPOSURE_TIME_REL_CTL_DEC_BY_1_STEP

```
#define EXPOSURE_TIME_REL_CTL_DEC_BY_1_STEP 0xFF
```

6.114.1.42 EXPOSURE_TIME_REL_CTL_DEF

```
#define EXPOSURE_TIME_REL_CTL_DEF 0x0
```

6.114.1.43 EXPOSURE_TIME_REL_CTL_INC_BY_1STEP

```
#define EXPOSURE_TIME_REL_CTL_INC_BY_1STEP 0x1
```

6.114.1.44 EXPOSURE_TIME_RELATIVE

```
#define EXPOSURE_TIME_RELATIVE 0x10
```

6.114.1.45 EXPOSURE_TIME_RELATIVE_LEN

```
#define EXPOSURE_TIME_RELATIVE_LEN 0x1
```

6.114.1.46 FOCUS_ABSOLUTE

```
#define FOCUS_ABSOLUTE 0x20
```

Kneron Confidential

6.114.1.47 FOCUS_ABSOLUTE_LEN

```
#define FOCUS_ABSOLUTE_LEN 0x2
```

6.114.1.48 FOCUS_AUTO

```
#define FOCUS_AUTO 0x20000
```

6.114.1.49 FOCUS_AUTO_LEN

```
#define FOCUS_AUTO_LEN 0x1
```

6.114.1.50 FOCUS_REL_CTL_REL_INFINITE_DIR

```
#define FOCUS_REL_CTL_REL_INFINITE_DIR 0xFF
```

6.114.1.51 FOCUS_REL_CTL_REL_NEAR_DIR

```
#define FOCUS_REL_CTL_REL_NEAR_DIR 0x1
```

6.114.1.52 FOCUS_REL_CTL_REL_STOP

```
#define FOCUS_REL_CTL_REL_STOP 0x0
```

6.114.1.53 FOCUS_RELATIVE

```
#define FOCUS_RELATIVE 0x40
```

6.114.1.54 FOCUS_RELATIVE_LEN

```
#define FOCUS_RELATIVE_LEN 0x2
```

Kneron Confidential

6.114.1.55 FOCUS_SIMPLE

```
#define FOCUS_SIMPLE 0x80000
```

6.114.1.56 FOCUS_SIMPLE_CTL_FULL_RANGE

```
#define FOCUS_SIMPLE_CTL_FULL_RANGE 0x00
```

6.114.1.57 FOCUS_SIMPLE_CTL_MACRO_LESS_THAN_0_3M

```
#define FOCUS_SIMPLE_CTL_MACRO_LESS_THAN_0_3M 0x01
```

6.114.1.58 FOCUS_SIMPLE_CTL_PEOPLE_0_3M_TO_3M

```
#define FOCUS_SIMPLE_CTL_PEOPLE_0_3M_TO_3M 0x02
```

6.114.1.59 FOCUS_SIMPLE_CTL_SCENE_3M_TO_INFINITY

```
#define FOCUS_SIMPLE_CTL_SCENE_3M_TO_INFINITY 0x03
```

6.114.1.60 FOCUS_SIMPLE_LEN

```
#define FOCUS_SIMPLE_LEN 0x1
```

6.114.1.61 GAIN

```
#define GAIN 0x200
```

6.114.1.62 GAIN_LEN

```
#define GAIN_LEN 0x2
```

Kneron Confidential

6.114.1.63 GAMMA

```
#define GAMMA 0x20
```

6.114.1.64 GAMMA_CTL_GAMMA_MAX

```
#define GAMMA_CTL_GAMMA_MAX 500
```

6.114.1.65 GAMMA_CTL_GAMMA_MIN

```
#define GAMMA_CTL_GAMMA_MIN 1
```

6.114.1.66 GAMMA_LEN

```
#define GAMMA_LEN 0x2
```

6.114.1.67 HUE

```
#define HUE 0x4
```

6.114.1.68 HUE_AUTO

```
#define HUE_AUTO 0x800
```

6.114.1.69 HUE_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED

```
#define HUE_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
```

6.114.1.70 HUE_AUTO_LEN

```
#define HUE_AUTO_LEN 0x1
```

Kneron Confidential

6.114.1.71 HUE_LEN

```
#define HUE_LEN 0x2
```

6.114.1.72 IRIS_ABSOLUTE

```
#define IRIS_ABSOLUTE 0x80
```

6.114.1.73 IRIS_ABSOLUTE_LEN

```
#define IRIS_ABSOLUTE_LEN 0x2
```

6.114.1.74 IRIS_REL_CTL_REL_CLOSE_BY_1STEP

```
#define IRIS_REL_CTL_REL_CLOSE_BY_1STEP 0xFF
```

6.114.1.75 IRIS_REL_CTL_REL_DEF

```
#define IRIS_REL_CTL_REL_DEF 0x0
```

6.114.1.76 IRIS_REL_CTL_REL_OPEN_BY_1STEP

```
#define IRIS_REL_CTL_REL_OPEN_BY_1STEP 0x1
```

6.114.1.77 IRIS_RELATIVE

```
#define IRIS_RELATIVE 0x100
```

6.114.1.78 IRIS_RELATIVE_LEN

```
#define IRIS_RELATIVE_LEN 0x2
```

Kneron Confidential

6.114.1.79 LEVEL_IDC

```
#define LEVEL_IDC 0x4000
```

6.114.1.80 LEVEL_IDC_LEN

```
#define LEVEL_IDC_LEN 0x1
```

6.114.1.81 LONG_TERM_BUFFER

```
#define LONG_TERM_BUFFER 0x800
```

6.114.1.82 LONG_TERM_BUFFER_CTL_TRUSTMODE_NOT_TRUST

```
#define LONG_TERM_BUFFER_CTL_TRUSTMODE_NOT_TRUST 0x0
```

6.114.1.83 LONG_TERM_BUFFER_CTL_TRUSTMODE_TRUST

```
#define LONG_TERM_BUFFER_CTL_TRUSTMODE_TRUST 0x1
```

6.114.1.84 LONG_TERM_BUFFER_LEN

```
#define LONG_TERM_BUFFER_LEN 0x2
```

6.114.1.85 LTR_VALIDATION

```
#define LTR_VALIDATION 0x2000
```

6.114.1.86 LTR_VALIDATION_LEN

```
#define LTR_VALIDATION_LEN 0x2
```

Kneron Confidential

6.114.1.87 MINIMUM_FRAME_INTERVAL

```
#define MINIMUM_FRAME_INTERVAL 0x8
```

6.114.1.88 MINIMUM_FRAME_INTERVAL_LEN

```
#define MINIMUM_FRAME_INTERVAL_LEN 0x4
```

6.114.1.89 PANTILT_ABSOLUTE

```
#define PANTILT_ABSOLUTE 0x800
```

6.114.1.90 PANTILT_ABSOLUTE_LEN

```
#define PANTILT_ABSOLUTE_LEN 0x8
```

6.114.1.91 PANTILT_REL_CTL_PLANE_DOWN

```
#define PANTILT_REL_CTL_PLANE_DOWN 0xFF
```

6.114.1.92 PANTILT_REL_CTL_PLANE_UP

```
#define PANTILT_REL_CTL_PLANE_UP 0x1
```

6.114.1.93 PANTILT_REL_CTL_STOP

```
#define PANTILT_REL_CTL_STOP 0x0
```

6.114.1.94 PANTILT_RELATIVE

```
#define PANTILT_RELATIVE 0x1000
```

Kneron Confidential

6.114.1.95 PANTILT_RELATIVE_LEN

```
#define PANTILT_RELATIVE_LEN 0x4
```

6.114.1.96 PEAK_BIT_RATE

```
#define PEAK_BIT_RATE 0x100
```

6.114.1.97 PEAK_BIT_RATE_LEN

```
#define PEAK_BIT_RATE_LEN 0x4
```

6.114.1.98 PICTURE_LONG_TERM_REFERENCE

```
#define PICTURE_LONG_TERM_REFERENCE 0x1000
```

6.114.1.99 PICTURE_LONG_TERM_REFERENCE_LEN

```
#define PICTURE_LONG_TERM_REFERENCE_LEN 0x2
```

6.114.1.100 POWER_LINE_FREQUENCY

```
#define POWER_LINE_FREQUENCY 0x400
```

6.114.1.101 POWER_LINE_FREQUENCY_CTL_50HZ

```
#define POWER_LINE_FREQUENCY_CTL_50HZ 1
```

6.114.1.102 POWER_LINE_FREQUENCY_CTL_60HZ

```
#define POWER_LINE_FREQUENCY_CTL_60HZ 2
```

Kneron Confidential

6.114.1.103 POWER_LINE_FREQUENCY_CTL_DISABLED

```
#define POWER_LINE_FREQUENCY_CTL_DISABLED 0
```

6.114.1.104 POWER_LINE_FREQUENCY_LEN

```
#define POWER_LINE_FREQUENCY_LEN 0x1
```

6.114.1.105 POWER_MODECTL_DEVICE_DEP_POWER

```
#define POWER_MODECTL_DEVICE_DEP_POWER 0001b
```

6.114.1.106 POWER_MODECTL_DEVICE_DEP_POWER_SUPPORTED

```
#define POWER_MODECTL_DEVICE_DEP_POWER_SUPPORTED 0x10
```

6.114.1.107 POWER_MODECTL_FULL_POWER

```
#define POWER_MODECTL_FULL_POWER 0000b
```

6.114.1.108 POWER_MODECTL_POWER_BY_BATTERY

```
#define POWER_MODECTL_POWER_BY_BATTERY 0x40
```

6.114.1.109 POWER_MODECTL_POWER_BY_BY_AC

```
#define POWER_MODECTL_POWER_BY_BY_AC 0x80
```

6.114.1.110 POWER_MODECTL_POWER_BY_USB

```
#define POWER_MODECTL_POWER_BY_USB 0x20
```

Kneron Confidential

6.114.1.111 PRIORITY_ID

```
#define PRIORITY_ID 0x20000
```

6.114.1.112 PRIORITY_ID_LEN

```
#define PRIORITY_ID_LEN 0x1
```

6.114.1.113 PRIVACY

```
#define PRIVACY 0x40000
```

6.114.1.114 PRIVACY_CTL_CLOSE

```
#define PRIVACY_CTL_CLOSE 0x1
```

6.114.1.115 PRIVACY_CTL_OPEN

```
#define PRIVACY_CTL_OPEN 0x0
```

6.114.1.116 PRIVACY_LEN

```
#define PRIVACY_LEN 0x1
```

6.114.1.117 PROBE_LEN

```
#define PROBE_LEN 48
```

6.114.1.118 PROFILE_AND_TOOLSET

```
#define PROFILE_AND_TOOLSET 0x2
```

Kneron Confidential

6.114.1.119 PROFILE_AND_TOOLSET_CTL_BASELINE_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_BASELINE_PROFILE 0x4200
```

6.114.1.120 PROFILE_AND_TOOLSET_CTL_CONSTRAINED_BASELINE_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_CONSTRAINED_BASELINE_PROFILE 0x4240
```

6.114.1.121 PROFILE_AND_TOOLSET_CTL_CONSTRAINED_HIGH_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_CONSTRAINED_HIGH_PROFILE 0x640C
```

6.114.1.122 PROFILE_AND_TOOLSET_CTL_HIGH_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_HIGH_PROFILE 0x6400
```

6.114.1.123 PROFILE_AND_TOOLSET_CTL_MAIN_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_MAIN_PROFILE 0x4D00
```

6.114.1.124 PROFILE_AND_TOOLSET_CTL_MULTIVIEW_HIGH_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_MULTIVIEW_HIGH_PROFILE 0x7600
```

6.114.1.125 PROFILE_AND_TOOLSET_CTL_SCALABLE_BASELINE_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_SCALABLE_BASELINE_PROFILE 0x5300
```

6.114.1.126 PROFILE_AND_TOOLSET_CTL_SCALABLE_CONSTRAINED_BASELINE_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_SCALABLE_CONSTRAINED_BASELINE_PROFILE 0x5304
```

Kneron Confidential

6.114.1.127 PROFILE_AND_TOOLSET_CTL_SCALABLE_CONSTRAINED_HIGH_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_SCALABLE_CONSTRAINED_HIGH_PROFILE 0x5604
```

6.114.1.128 PROFILE_AND_TOOLSET_CTL_SCALABLE_HIGH_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_SCALABLE_HIGH_PROFILE 0x5600
```

6.114.1.129 PROFILE_AND_TOOLSET_CTL_STEREO_HIGH_PROFILE

```
#define PROFILE_AND_TOOLSET_CTL_STEREO_HIGH_PROFILE 0x8000
```

6.114.1.130 PROFILE_AND_TOOLSET_LEN [1/2]

```
#define PROFILE_AND_TOOLSET_LEN 0x5
```

6.114.1.131 PROFILE_AND_TOOLSET_LEN [2/2]

```
#define PROFILE_AND_TOOLSET_LEN 0x5
```

6.114.1.132 QP_RANGE

```
#define QP_RANGE 0x10000
```

6.114.1.133 QP_RANGE_LEN

```
#define QP_RANGE_LEN 0x2
```

6.114.1.134 QUANTIZATION_PARAMETER

```
#define QUANTIZATION_PARAMETER 0x200
```

Kneron Confidential

6.114.1.135 QUANTIZATION_PARAMETER_LEN

```
#define QUANTIZATION_PARAMETER_LEN 0x6
```

6.114.1.136 RATE_CONTROL_MODE

```
#define RATE_CONTROL_MODE 0x20
```

6.114.1.137 RATE_CONTROL_MODE_CTL_MODE_CBR

```
#define RATE_CONTROL_MODE_CTL_MODE_CBR 0x2
```

6.114.1.138 RATE_CONTROL_MODE_CTL_MODE_Constant_QP

```
#define RATE_CONTROL_MODE_CTL_MODE_Constant_QP 0x3
```

6.114.1.139 RATE_CONTROL_MODE_CTL_MODE_GVBR

```
#define RATE_CONTROL_MODE_CTL_MODE_GVBR 0x4
```

6.114.1.140 RATE_CONTROL_MODE_CTL_MODE_GVBRN

```
#define RATE_CONTROL_MODE_CTL_MODE_GVBRN 0x6
```

6.114.1.141 RATE_CONTROL_MODE_CTL_MODE_VBR

```
#define RATE_CONTROL_MODE_CTL_MODE_VBR 0x1
```

6.114.1.142 RATE_CONTROL_MODE_CTL_MODE_VBRN

```
#define RATE_CONTROL_MODE_CTL_MODE_VBRN 0x5
```

Kneron Confidential

6.114.1.143 RATE_CONTROL_MODE_LEN

```
#define RATE_CONTROL_MODE_LEN 0x1
```

6.114.1.144 REGION_OF_INTEREST

```
#define REGION_OF_INTEREST 0x200000
```

6.114.1.145 REGION_OF_INTEREST_LEN

```
#define REGION_OF_INTEREST_LEN 0xa
```

6.114.1.146 REQUEST_ERROR_CTL

```
#define REQUEST_ERROR_CTL
```

6.114.1.147 ROI_CTL_AUTOCTL_AUTO_DETECT_TRACK

```
#define ROI_CTL_AUTOCTL_AUTO_DETECT_TRACK 0x20
```

6.114.1.148 ROI_CTL_AUTOCTL_EXPOSURE

```
#define ROI_CTL_AUTOCTL_EXPOSURE 0x1
```

6.114.1.149 ROI_CTL_AUTOCTL_FACE_DETECT

```
#define ROI_CTL_AUTOCTL_FACE_DETECT 0x10
```

6.114.1.150 ROI_CTL_AUTOCTL_FOCUS

```
#define ROI_CTL_AUTOCTL_FOCUS 0x8
```

Kneron Confidential

6.114.1.151 ROI_CTL_AUTOCTL_HIGHER_QUALITY

```
#define ROI_CTL_AUTOCTL_HIGHER_QUALITY 0x80
```

6.114.1.152 ROI_CTL_AUTOCTL_IMAGE_STABILIZATION

```
#define ROI_CTL_AUTOCTL_IMAGE_STABILIZATION 0x40
```

6.114.1.153 ROI_CTL_AUTOCTL_IRIS

```
#define ROI_CTL_AUTOCTL_IRIS 0x2
```

6.114.1.154 ROI_CTL_AUTOCTL_WHITE_BALANCE

```
#define ROI_CTL_AUTOCTL_WHITE_BALANCE 0x4
```

6.114.1.155 ROLL_ABSOLUTE

```
#define ROLL_ABSOLUTE 0x2000
```

6.114.1.156 ROLL_ABSOLUTE_LEN

```
#define ROLL_ABSOLUTE_LEN 0x2
```

6.114.1.157 ROLL_RELATIVE

```
#define ROLL_RELATIVE 0x4000
```

6.114.1.158 ROLL_RELATIVE_CTL_CLOCKWISE_ROTATION

```
#define ROLL_RELATIVE_CTL_CLOCKWISE_ROTATION 0x1
```

Kneron Confidential

6.114.1.159 ROLL_RELATIVE_CTL_COUNTER_CLOCKWISE_ROTATION

```
#define ROLL_RELATIVE_CTL_COUNTER_CLOCKWISE_ROTATION 0xFF
```

6.114.1.160 ROLL_RELATIVE_CTL_STOP

```
#define ROLL_RELATIVE_CTL_STOP 0x0
```

6.114.1.161 ROLL_RELATIVE_LEN

```
#define ROLL_RELATIVE_LEN 0x2
```

6.114.1.162 SATURATION

```
#define SATURATION 0x8
```

6.114.1.163 SATURATION_CTL_GRAYSCALE

```
#define SATURATION_CTL_GRAYSCALE 0x0
```

6.114.1.164 SATURATION_LEN

```
#define SATURATION_LEN 0x2
```

6.114.1.165 SCANNING_MODE

```
#define SCANNING_MODE 0x1
```

6.114.1.166 SCANNING_MODE_CTL_INTERLACED

```
#define SCANNING_MODE_CTL_INTERLACED 0x0
```

Kneron Confidential

6.114.1.167 SCANNING_MODE_CTL_PROGRESSIVE

```
#define SCANNING_MODE_CTL_PROGRESSIVE 0x1
```

6.114.1.168 SCANNING_MODE_LEN

```
#define SCANNING_MODE_LEN 0x1
```

6.114.1.169 SEI_MESSAGE

```
#define SEI_MESSAGE 0x8000
```

6.114.1.170 SEI_MESSAGE_CTL_BASE_LAYER_TEMPORAL_HRD

```
#define SEI_MESSAGE_CTL_BASE_LAYER_TEMPORAL_HRD 0x80000000
```

6.114.1.171 SEI_MESSAGE_CTL_BASE_VIEW_TEMPORAL_HRD

```
#define SEI_MESSAGE_CTL_BASE_VIEW_TEMPORAL_HRD 0x100000000000
```

6.114.1.172 SEI_MESSAGE_CTL_BUFFERING_PERIOD

```
#define SEI_MESSAGE_CTL_BUFFERING_PERIOD 0x1
```

6.114.1.173 SEI_MESSAGE_CTL_DEBLOCKING_FILTER_DISPLAY_PREFERENCE

```
#define SEI_MESSAGE_CTL_DEBLOCKING_FILTER_DISPLAY_PREFERENCE 0x100000
```

6.114.1.174 SEI_MESSAGE_CTL_DEC_REF_PIC_MARKING_REPETITION

```
#define SEI_MESSAGE_CTL_DEC_REF_PIC_MARKING_REPETITION 0x80
```

Kneron Confidential

6.114.1.175 SEI_MESSAGE_CTL_FILLER_PAYLOAD

```
#define SEI_MESSAGE_CTL_FILLER_PAYLOAD 0x8
```

6.114.1.176 SEI_MESSAGE_CTL_FILM_GRAIN_CHARACTERISTICS

```
#define SEI_MESSAGE_CTL_FILM_GRAIN_CHARACTERISTICS 0x80000
```

6.114.1.177 SEI_MESSAGE_CTL_FRAME_PACKING_ARRANGEMENT

```
#define SEI_MESSAGE_CTL_FRAME_PACKING_ARRANGEMENT 0x200000000000
```

6.114.1.178 SEI_MESSAGE_CTL_FULL_FRAME_FREEZE

```
#define SEI_MESSAGE_CTL_FULL_FRAME_FREEZE 0x2000
```

6.114.1.179 SEI_MESSAGE_CTL_FULL_FRAME_FREEZE_RELEASE

```
#define SEI_MESSAGE_CTL_FULL_FRAME_FREEZE_RELEASE 0x4000
```

6.114.1.180 SEI_MESSAGE_CTL_FULL_FRAME_SNAPSHOT

```
#define SEI_MESSAGE_CTL_FULL_FRAME_SNAPSHOT 0x8000
```

6.114.1.181 SEI_MESSAGE_CTL_LAYER_DEPENDENCY_CHANGE

```
#define SEI_MESSAGE_CTL_LAYER_DEPENDENCY_CHANGE 0x20000000
```

6.114.1.182 SEI_MESSAGE_CTL_LAYERS_NOT_PRESENT

```
#define SEI_MESSAGE_CTL_LAYERS_NOT_PRESENT 0x10000000
```

Kneron Confidential

6.114.1.183 SEI_MESSAGE_CTL_MOTION_CONSTRAINED_SLICE_GROUP_SET

```
#define SEI_MESSAGE_CTL_MOTION_CONSTRAINED_SLICE_GROUP_SET 0x40000
```

6.114.1.184 SEI_MESSAGE_CTL_MULTIVIEW_ACQUISITION_INFO

```
#define SEI_MESSAGE_CTL_MULTIVIEW_ACQUISITION_INFO 0x10000000000
```

6.114.1.185 SEI_MESSAGE_CTL_MULTIVIEW_SCENE_INFO

```
#define SEI_MESSAGE_CTL_MULTIVIEW_SCENE_INFO 0x8000000000
```

6.114.1.186 SEI_MESSAGE_CTL_MVC_SCALABLE_NESTING

```
#define SEI_MESSAGE_CTL_MVC_SCALABLE_NESTING 0x2000000000
```

6.114.1.187 SEI_MESSAGE_CTL_NON_REQUIRED_LAYER_REP

```
#define SEI_MESSAGE_CTL_NON_REQUIRED_LAYER_REP 0x4000000
```

6.114.1.188 SEI_MESSAGE_CTL_NON_REQUIRED_VIEW_COMPONENT

```
#define SEI_MESSAGE_CTL_NON_REQUIRED_VIEW_COMPONENT 0x20000000000
```

6.114.1.189 SEI_MESSAGE_CTL_OPERATION_POINTS_NOT_PRESENT

```
#define SEI_MESSAGE_CTL_OPERATION_POINTS_NOT_PRESENT 0x80000000000
```

6.114.1.190 SEI_MESSAGE_CTL_PAN_SCAN_RECT

```
#define SEI_MESSAGE_CTL_PAN_SCAN_RECT 0x4
```

Kneron Confidential

6.114.1.191 SEI_MESSAGE_CTL_PARALLEL_DECODING_INFO

```
#define SEI_MESSAGE_CTL_PARALLEL_DECODING_INFO 0x1000000000
```

6.114.1.192 SEI_MESSAGE_CTL_PIC_TIMING

```
#define SEI_MESSAGE_CTL_PIC_TIMING 0x2
```

6.114.1.193 SEI_MESSAGE_CTL_POST_FILTER_HINT

```
#define SEI_MESSAGE_CTL_POST_FILTER_HINT 0x400000
```

6.114.1.194 SEI_MESSAGE_CTL_PRIORITY_LAYER_INFO

```
#define SEI_MESSAGE_CTL_PRIORITY_LAYER_INFO 0x8000000
```

6.114.1.195 SEI_MESSAGE_CTL_PROGRESSIVE_REFINEMENT_SEGMENT_END

```
#define SEI_MESSAGE_CTL_PROGRESSIVE_REFINEMENT_SEGMENT_END 0x20000
```

6.114.1.196 SEI_MESSAGE_CTL_PROGRESSIVE_REFINEMENT_SEGMENT_START

```
#define SEI_MESSAGE_CTL_PROGRESSIVE_REFINEMENT_SEGMENT_START 0x10000
```

6.114.1.197 SEI_MESSAGE_CTL_QUALITY_LAYER_INTEGRITY_CHECK

```
#define SEI_MESSAGE_CTL_QUALITY_LAYER_INTEGRITY_CHECK 0x100000000
```

6.114.1.198 SEI_MESSAGE_CTL_RECOVERY_POINT

```
#define SEI_MESSAGE_CTL_RECOVERY_POINT 0x40
```

Kneron Confidential

6.114.1.199 SEI_MESSAGE_CTL_REDUNDANT_PIC_PROPERTY

```
#define SEI_MESSAGE_CTL_REDUNDANT_PIC_PROPERTY 0x200000000
```

6.114.1.200 SEI_MESSAGE_CTL_SCALABILITY_INFO

```
#define SEI_MESSAGE_CTL_SCALABILITY_INFO 0x1000000
```

6.114.1.201 SEI_MESSAGE_CTL_SCALABLE_NESTING

```
#define SEI_MESSAGE_CTL_SCALABLE_NESTING 0x40000000
```

6.114.1.202 SEI_MESSAGE_CTL_SCENE_INFO

```
#define SEI_MESSAGE_CTL_SCENE_INFO 0x200
```

6.114.1.203 SEI_MESSAGE_CTL_SPARE_PIC

```
#define SEI_MESSAGE_CTL_SPARE_PIC 0x100
```

6.114.1.204 SEI_MESSAGE_CTL_STEREO_VIDEO_INFO

```
#define SEI_MESSAGE_CTL_STEREO_VIDEO_INFO 0x200000
```

6.114.1.205 SEI_MESSAGE_CTL_SUB_PIC_SCALABLE_LAYER

```
#define SEI_MESSAGE_CTL_SUB_PIC_SCALABLE_LAYER 0x2000000
```

6.114.1.206 SEI_MESSAGE_CTL_SUB_SEQ_CHARACTERISTICS

```
#define SEI_MESSAGE_CTL_SUB_SEQ_CHARACTERISTICS 0x1000
```

Kneron Confidential

6.114.1.207 SEI_MESSAGE_CTL_SUB_SEQ_INFO

```
#define SEI_MESSAGE_CTL_SUB_SEQ_INFO 0x400
```

6.114.1.208 SEI_MESSAGE_CTL_SUB_SEQ_LAYER_CHARACTERISTICS

```
#define SEI_MESSAGE_CTL_SUB_SEQ_LAYER_CHARACTERISTICS 0x800
```

6.114.1.209 SEI_MESSAGE_CTL_TL0_DEP_REP_INDEX

```
#define SEI_MESSAGE_CTL_TL0_DEP_REP_INDEX 0x400000000
```

6.114.1.210 SEI_MESSAGE_CTL_TL_SWITCHING_POINT

```
#define SEI_MESSAGE_CTL_TL_SWITCHING_POINT 0x800000000
```

6.114.1.211 SEI_MESSAGE_CTL_TONE_MAPPING_INFO

```
#define SEI_MESSAGE_CTL_TONE_MAPPING_INFO 0x800000
```

6.114.1.212 SEI_MESSAGE_CTL_USER_DATA_REGISTERED_ITU_T_T35

```
#define SEI_MESSAGE_CTL_USER_DATA_REGISTERED_ITU_T_T35 0x10
```

6.114.1.213 SEI_MESSAGE_CTL_USER_DATA_UNREGISTERED

```
#define SEI_MESSAGE_CTL_USER_DATA_UNREGISTERED 0x20
```

6.114.1.214 SEI_MESSAGE_CTL_VIEW_DEPENDENCY_CHANGE

```
#define SEI_MESSAGE_CTL_VIEW_DEPENDENCY_CHANGE 0x40000000000
```

Kneron Confidential

6.114.1.215 SEI_MESSAGE_CTL_VIEW_SCALABILITY_INFO

```
#define SEI_MESSAGE_CTL_VIEW_SCALABILITY_INFO 0x4000000000
```

6.114.1.216 SEI_MESSAGE_LEN

```
#define SEI_MESSAGE_LEN 0x8
```

6.114.1.217 SELECT_LAYER

```
#define SELECT_LAYER 0x1
```

6.114.1.218 SELECT_LAYERL_LEN

```
#define SELECT_LAYERL_LEN 0x2
```

6.114.1.219 SHARPNESS

```
#define SHARPNESS 0x10
```

6.114.1.220 SHARPNESS_LEN

```
#define SHARPNESS_LEN 0x2
```

6.114.1.221 SLICE_MODE

```
#define SLICE_MODE 0x10
```

6.114.1.222 SLICE_MODE_CTL_MODE_MAX_MBS_PER_SLICE

```
#define SLICE_MODE_CTL_MODE_MAX_MBS_PER_SLICE 0x0
```

Kneron Confidential

6.114.1.223 SLICE_MODE_CTL_MODE_NUMBER_OF_MACROBLOCK_ROWS_PER_SLICE

```
#define SLICE_MODE_CTL_MODE_NUMBER_OF_MACROBLOCK_ROWS_PER_SLICE 0x3
```

6.114.1.224 SLICE_MODE_CTL_MODE_NUMBER_OF_SLICES_PER_FRAME

```
#define SLICE_MODE_CTL_MODE_NUMBER_OF_SLICES_PER_FRAME 0x2
```

6.114.1.225 SLICE_MODE_CTL_MODE_TARGET_COMPRESSED_SIZE_PER_SLICE

```
#define SLICE_MODE_CTL_MODE_TARGET_COMPRESSED_SIZE_PER_SLICE 0x1
```

6.114.1.226 SLICE_MODE_LEN

```
#define SLICE_MODE_LEN 0x4
```

6.114.1.227 START_OR_STOP_LAYER_VIEW

```
#define START_OR_STOP_LAYER_VIEW 0x40000
```

6.114.1.228 START_OR_STOP_LAYER_VIEW_LEN

```
#define START_OR_STOP_LAYER_VIEW_LEN 0x1
```

6.114.1.229 SYNC_AND_LTR_FRAME_CTL_IDR

```
#define SYNC_AND_LTR_FRAME_CTL_IDR 0x1
```

6.114.1.230 SYNC_AND_LTR_FRAME_CTL_LONG_TERM_REF_IDR

```
#define SYNC_AND_LTR_FRAME_CTL_LONG_TERM_REF_IDR 0x2
```

Kneron Confidential

6.114.1.231 SYNC_AND_LTR_FRAME_CTL_LTR_NON_IDR_RAND_I_FRAME

```
#define SYNC_AND_LTR_FRAME_CTL_LTR_NON_IDR_RAND_I_FRAME 0x4
```

6.114.1.232 SYNC_AND_LTR_FRAME_CTL_LTR_P_FRAME

```
#define SYNC_AND_LTR_FRAME_CTL_LTR_P_FRAME 0x5
```

6.114.1.233 SYNC_AND_LTR_FRAME_CTL_NON_IDR_RAND_I_FRAME

```
#define SYNC_AND_LTR_FRAME_CTL_NON_IDR_RAND_I_FRAME 0x3
```

6.114.1.234 SYNC_AND_LTR_FRAME_CTL_RESET

```
#define SYNC_AND_LTR_FRAME_CTL_RESET 0x0
```

6.114.1.235 SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME

```
#define SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME 0x400
```

6.114.1.236 SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME_LEN

```
#define SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME_LEN 0x4
```

6.114.1.237 UVC_CID_ANALOG_VIDEO_LOCK_STATUS

```
#define UVC_CID_ANALOG_VIDEO_LOCK_STATUS UVC_CID_PU_CLASS_BASE | ANALOG_VIDEO_LOCK_STATUS
```

6.114.1.238 UVC_CID_ANALOG_VIDEO_STANDARD

```
#define UVC_CID_ANALOG_VIDEO_STANDARD UVC_CID_PU_CLASS_BASE | ANALOG_VIDEO_STANDARD
```

Kneron Confidential

6.114.1.239 UVC_CID_AUTO_EXPOSURE_MODE

```
#define UVC_CID_AUTO_EXPOSURE_MODE UVC_CID_CAMERA_CLASS_BASE | AUTO_EXPOSURE_MODE
```

6.114.1.240 UVC_CID_AUTO_EXPOSURE_PRIORITY

```
#define UVC_CID_AUTO_EXPOSURE_PRIORITY UVC_CID_CAMERA_CLASS_BASE | AUTO_EXPOSURE_PRIORITY
```

6.114.1.241 UVC_CID_AVERAGE_BIT_RATE

```
#define UVC_CID_AVERAGE_BIT_RATE UVC_CID_EU_CLASS_BASE | AVERAGE_BIT_RATE
```

6.114.1.242 UVC_CID_BACKLIGHT_COMPENSATION

```
#define UVC_CID_BACKLIGHT_COMPENSATION UVC_CID_PU_CLASS_BASE | BACKLIGHT_COMPENSATION
```

6.114.1.243 UVC_CID_BRIGHTNESS

```
#define UVC_CID_BRIGHTNESS UVC_CID_PU_CLASS_BASE | BRIGHTNESS
```

6.114.1.244 UVC_CID_CAMERA_CLASS_BASE

```
#define UVC_CID_CAMERA_CLASS_BASE (UVC_ET_CT << ET_POS)
```

6.114.1.245 UVC_CID_CONTRAST

```
#define UVC_CID_CONTRAST UVC_CID_PU_CLASS_BASE | CONTRAST
```

6.114.1.246 UVC_CID_CONTRAST_AUTO

```
#define UVC_CID_CONTRAST_AUTO UVC_CID_PU_CLASS_BASE | CONTRAST_AUTO
```

Kneron Confidential

6.114.1.247 UVC_CID_CPB_SIZE

```
#define UVC_CID_CPB_SIZE UVC_CID_EU_CLASS_BASE | CPB_SIZE
```

6.114.1.248 UVC_CID_DEV_INF_CLASS_BASE

```
#define UVC_CID_DEV_INF_CLASS_BASE (UVC_DEV_INF << ET_POS)
```

6.114.1.249 UVC_CID_DEVICE_POWER_MODE

```
#define UVC_CID_DEVICE_POWER_MODE UVC_CID_DEV_INF_CLASS_BASE | UVC_VC_VIDEO_POWER_MODE_CONTROL
```

6.114.1.250 UVC_CID_DIGITAL_MULTIPLIER

```
#define UVC_CID_DIGITAL_MULTIPLIER UVC_CID_PU_CLASS_BASE | DIGITAL_MULTIPLIER
```

6.114.1.251 UVC_CID_DIGITAL_MULTIPLIER_LIMIT

```
#define UVC_CID_DIGITAL_MULTIPLIER_LIMIT UVC_CID_PU_CLASS_BASE | DIGITAL_MULTIPLIER_LIMIT
```

6.114.1.252 UVC_CID_ERROR_RESILIENCY

```
#define UVC_CID_ERROR_RESILIENCY UVC_CID_EU_CLASS_BASE | ERROR_RESILIENCY
```

6.114.1.253 UVC_CID_EU_CLASS_BASE

```
#define UVC_CID_EU_CLASS_BASE (UVC_ET_EU << ET_POS)
```

6.114.1.254 UVC_CID_EXPOSURE_TIME_ABSOLUTE

```
#define UVC_CID_EXPOSURE_TIME_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | EXPOSURE_TIME_ABSOLUTE
```

Kneron Confidential

6.114.1.255 UVC_CID_EXPOSURE_TIME_RELATIVE

```
#define UVC_CID_EXPOSURE_TIME_RELATIVE UVC_CID_CAMERA_CLASS_BASE | EXPOSURE_TIME_RELATIVE
```

6.114.1.256 UVC_CID_FOCUS_ABSOLUTE

```
#define UVC_CID_FOCUS_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | FOCUS_ABSOLUTE
```

6.114.1.257 UVC_CID_FOCUS_AUTO

```
#define UVC_CID_FOCUS_AUTO UVC_CID_CAMERA_CLASS_BASE | FOCUS_AUTO
```

6.114.1.258 UVC_CID_FOCUS_RELATIVE

```
#define UVC_CID_FOCUS_RELATIVE UVC_CID_CAMERA_CLASS_BASE | FOCUS_RELATIVE
```

6.114.1.259 UVC_CID_FOCUS_SIMPLE

```
#define UVC_CID_FOCUS_SIMPLE UVC_CID_CAMERA_CLASS_BASE | FOCUS_SIMPLE
```

6.114.1.260 UVC_CID_GAIN

```
#define UVC_CID_GAIN UVC_CID_PU_CLASS_BASE | GAIN
```

6.114.1.261 UVC_CID_GAMMA

```
#define UVC_CID_GAMMA UVC_CID_PU_CLASS_BASE | GAMMA
```

6.114.1.262 UVC_CID_HUE

```
#define UVC_CID_HUE UVC_CID_PU_CLASS_BASE | HUE
```

Kneron Confidential

6.114.1.263 UVC_CID_HUE_AUTO

```
#define UVC_CID_HUE_AUTO UVC_CID_PU_CLASS_BASE | HUE_AUTO
```

6.114.1.264 UVC_CID_IRIS_ABSOLUTE

```
#define UVC_CID_IRIS_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | IRIS_ABSOLUTE
```

6.114.1.265 UVC_CID_IRIS_RELATIVE

```
#define UVC_CID_IRIS_RELATIVE UVC_CID_CAMERA_CLASS_BASE | IRIS_RELATIVE
```

6.114.1.266 UVC_CID_LEVEL_IDC

```
#define UVC_CID_LEVEL_IDC UVC_CID_EU_CLASS_BASE | LEVEL_IDC
```

6.114.1.267 UVC_CID_LONG_TERM_BUFFER

```
#define UVC_CID_LONG_TERM_BUFFER UVC_CID_EU_CLASS_BASE | LONG_TERM_BUFFER
```

6.114.1.268 UVC_CID_LTR_VALIDATION

```
#define UVC_CID_LTR_VALIDATION UVC_CID_EU_CLASS_BASE | LTR_VALIDATION
```

6.114.1.269 UVC_CID_MINIMUM_FRAME_INTERVAL

```
#define UVC_CID_MINIMUM_FRAME_INTERVAL UVC_CID_EU_CLASS_BASE | MINIMUM_FRAME_INTERVAL
```

6.114.1.270 UVC_CID_PANTILT_ABSOLUTE

```
#define UVC_CID_PANTILT_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | PANTILT_ABSOLUTE
```

Kneron Confidential

6.114.1.271 UVC_CID_PANTILT_RELATIVE

```
#define UVC_CID_PANTILT_RELATIVE UVC_CID_CAMERA_CLASS_BASE | PANTILT_RELATIVE
```

6.114.1.272 UVC_CID_PEAK_BIT_RATE

```
#define UVC_CID_PEAK_BIT_RATE UVC_CID_EU_CLASS_BASE | PEAK_BIT_RATE
```

6.114.1.273 UVC_CID_PICTURE_LONG_TERM_REFERENCE

```
#define UVC_CID_PICTURE_LONG_TERM_REFERENCE UVC_CID_EU_CLASS_BASE | PICTURE_LONG_TERM_REFERENCE
```

6.114.1.274 UVC_CID_POWER_LINE_FREQUENCY

```
#define UVC_CID_POWER_LINE_FREQUENCY UVC_CID_PU_CLASS_BASE | POWER_LINE_FREQUENCY
```

6.114.1.275 UVC_CID_PRIORITY_ID

```
#define UVC_CID_PRIORITY_ID UVC_CID_EU_CLASS_BASE | PRIORITY_ID
```

6.114.1.276 UVC_CID_PRIVACY

```
#define UVC_CID_PRIVACY UVC_CID_CAMERA_CLASS_BASE | PRIVACY
```

6.114.1.277 UVC_CID_PROFILE_AND_TOOLSET

```
#define UVC_CID_PROFILE_AND_TOOLSET UVC_CID_EU_CLASS_BASE | PROFILE_AND_TOOLSET
```

6.114.1.278 UVC_CID_PU_CLASS_BASE

```
#define UVC_CID_PU_CLASS_BASE (UVC_ET_PU << ET_POS)
```

Kneron Confidential

6.114.1.279 UVC_CID_QP_RANGE

```
#define UVC_CID_QP_RANGE UVC_CID_EU_CLASS_BASE | QP_RANGE
```

6.114.1.280 UVC_CID_QUANTIZATION_PARAMETER

```
#define UVC_CID_QUANTIZATION_PARAMETER UVC_CID_EU_CLASS_BASE | QUANTIZATION_PARAMETER
```

6.114.1.281 UVC_CID_RATE_CONTROL_MODE

```
#define UVC_CID_RATE_CONTROL_MODE UVC_CID_EU_CLASS_BASE | RATE_CONTROL_MODE
```

6.114.1.282 UVC_CID_REGION_OF_INTEREST

```
#define UVC_CID_REGION_OF_INTEREST UVC_CID_CAMERA_CLASS_BASE | REGION_OF_INTEREST
```

6.114.1.283 UVC_CID_REQUEST_ERROR_CODE

```
#define UVC_CID_REQUEST_ERROR_CODE UVC_CID_DEV_INF_CLASS_BASE | UVC_VC_REQUEST_ERROR_CODE_CONTROL
<< 24
```

6.114.1.284 UVC_CID_ROLL_ABSOLUTE

```
#define UVC_CID_ROLL_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | ROLL_ABSOLUTE
```

6.114.1.285 UVC_CID_ROLL_RELATIVE

```
#define UVC_CID_ROLL_RELATIVE UVC_CID_CAMERA_CLASS_BASE | ROLL_RELATIVE
```

Kneron Confidential

6.114.1.286 UVC_CID_SATURATION

```
#define UVC_CID_SATURATION UVC_CID_PU_CLASS_BASE | SATURATION
```

6.114.1.287 UVC_CID_SCANNING_MODE

```
#define UVC_CID_SCANNING_MODE UVC_CID_CAMERA_CLASS_BASE | SCANNING_MODE
```

6.114.1.288 UVC_CID_SEI_MESSAGE

```
#define UVC_CID_SEI_MESSAGE UVC_CID_EU_CLASS_BASE | SEI_MESSAGE
```

6.114.1.289 UVC_CID_SELECT_LAYER

```
#define UVC_CID_SELECT_LAYER UVC_CID_EU_CLASS_BASE | SELECT_LAYER
```

6.114.1.290 UVC_CID_SHARPNESS

```
#define UVC_CID_SHARPNESS UVC_CID_PU_CLASS_BASE | SHARPNESS
```

6.114.1.291 UVC_CID_SLICE_MODE

```
#define UVC_CID_SLICE_MODE UVC_CID_EU_CLASS_BASE | SLICE_MODE
```

6.114.1.292 UVC_CID_START_OR_STOP_LAYER_VIEW

```
#define UVC_CID_START_OR_STOP_LAYER_VIEW UVC_CID_EU_CLASS_BASE | START_OR_STOP_LAYER_VIEW
```

6.114.1.293 UVC_CID_SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME

```
#define UVC_CID_SYNCHRONIZATION_AND_LONGTERM_REFERENCE_FRAME UVC_CID_EU_CLASS_BASE | SYNCHRONIZATION_AND_LONGT
```

Kneron Confidential

6.114.1.294 UVC_CID_VIDEO_RESOLUTION

```
#define UVC_CID_VIDEO_RESOLUTION UVC_CID_EU_CLASS_BASE | VIDEO_RESOLUTION
```

6.114.1.295 UVC_CID_WHITE_BALANCE_COMPONENT

```
#define UVC_CID_WHITE_BALANCE_COMPONENT UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_COMPONENT
```

6.114.1.296 UVC_CID_WHITE_BALANCE_COMPONENT_AUTO

```
#define UVC_CID_WHITE_BALANCE_COMPONENT_AUTO UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_COMPONENT_AUTO
```

6.114.1.297 UVC_CID_WHITE_BALANCE_TEMPERATURE

```
#define UVC_CID_WHITE_BALANCE_TEMPERATURE UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_TEMPERATURE
```

6.114.1.298 UVC_CID_WHITE_BALANCE_TEMPERATURE_AUTO

```
#define UVC_CID_WHITE_BALANCE_TEMPERATURE_AUTO UVC_CID_PU_CLASS_BASE | WHITE_BALANCE_TEMPERATURE_AUTO
```

6.114.1.299 UVC_CID_WINDOW

```
#define UVC_CID_WINDOW UVC_CID_CAMERA_CLASS_BASE | WINDOW
```

6.114.1.300 UVC_CID_XU_CLASS_BASE

```
#define UVC_CID_XU_CLASS_BASE (UVC_ET_XU << ET_POS)
```

6.114.1.301 UVC_CID_ZOOM_ABSOLUTE

```
#define UVC_CID_ZOOM_ABSOLUTE UVC_CID_CAMERA_CLASS_BASE | ZOOM_ABSOLUTE
```

Kneron Confidential

6.114.1.302 UVC_CID_ZOOM_RELATIVE

```
#define UVC_CID_ZOOM_RELATIVE UVC_CID_CAMERA_CLASS_BASE | ZOOM_RELATIVE
```

6.114.1.303 UVC_VSCID_COMMIT

```
#define UVC_VSCID_COMMIT UVC_VS_COMMIT_CONTROL << 8
```

6.114.1.304 UVC_VSCID_COMMIT_CONTROL

```
#define UVC_VSCID_COMMIT_CONTROL UVC_VS_COMMIT_CONTROL
```

6.114.1.305 UVC_VSCID_GENERATE_KEY_FRAME

```
#define UVC_VSCID_GENERATE_KEY_FRAME UVC_VS_GENERATE_KEY_FRAME_CONTROL
```

6.114.1.306 UVC_VSCID_PROBE

```
#define UVC_VSCID_PROBE UVC_VS_PROBE_CONTROL << 8
```

6.114.1.307 UVC_VSCID_PROBE_CONTROL

```
#define UVC_VSCID_PROBE_CONTROL UVC_VS_PROBE_CONTROL
```

6.114.1.308 UVC_VSCID_STILL_COMMIT

```
#define UVC_VSCID_STILL_COMMIT UVC_VS_STILL_COMMIT_CONTROL
```

6.114.1.309 UVC_VSCID_STILL_IMAGE_TRIGGER

```
#define UVC_VSCID_STILL_IMAGE_TRIGGER UVC_VS_STILL_IMAGE_TRIGGER_CONTROL
```

Kneron Confidential

6.114.1.310 UVC_VSCID_STILL_PROBE

```
#define UVC_VSCID_STILL_PROBE UVC_VS_STILL_PROBE_CONTROL
```

6.114.1.311 UVC_VSCID_STREAM_ERROR_CODE

```
#define UVC_VSCID_STREAM_ERROR_CODE UVC_VS_STREAM_ERROR_CODE_CONTROL
```

6.114.1.312 UVC_VSCID_SYNC_DELAY

```
#define UVC_VSCID_SYNC_DELAY UVC_VS_SYNC_DELAY_CONTROL
```

6.114.1.313 UVC_VSCID_UPDATE_FRAME_SEGMENT

```
#define UVC_VSCID_UPDATE_FRAME_SEGMENT UVC_VS_UPDATE_FRAME_SEGMENT_CONTROL
```

6.114.1.314 VIDEO_MAX_FRAME

```
#define VIDEO_MAX_FRAME 32
```

6.114.1.315 VIDEO_RESOLUTION

```
#define VIDEO_RESOLUTION 0x4
```

6.114.1.316 VIDEO_RESOLUTION_LEN

```
#define VIDEO_RESOLUTION_LEN 0x4
```

6.114.1.317 WHITE_BALANCE_COMPONENT

```
#define WHITE_BALANCE_COMPONENT 0x80
```

Kneron Confidential

6.114.1.318 WHITE_BALANCE_COMPONENT_AUTO

```
#define WHITE_BALANCE_COMPONENT_AUTO 0x2000
```

6.114.1.319 WHITE_BALANCE_COMPONENT_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED

```
#define WHITE_BALANCE_COMPONENT_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
```

6.114.1.320 WHITE_BALANCE_COMPONENT_AUTO_LEN

```
#define WHITE_BALANCE_COMPONENT_AUTO_LEN 0x1
```

6.114.1.321 WHITE_BALANCE_COMPONENT_LEN

```
#define WHITE_BALANCE_COMPONENT_LEN 0x2
```

6.114.1.322 WHITE_BALANCE_TEMPERATURE

```
#define WHITE_BALANCE_TEMPERATURE 0x40
```

6.114.1.323 WHITE_BALANCE_TEMPERATURE_AUTO

```
#define WHITE_BALANCE_TEMPERATURE_AUTO 0x1000
```

6.114.1.324 WHITE_BALANCE_TEMPERATURE_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED

```
#define WHITE_BALANCE_TEMPERATURE_AUTO_CTL_AUTO_ADJUSTMENT_ENABLED 0x1
```

6.114.1.325 WHITE_BALANCE_TEMPERATURE_AUTO_LEN

```
#define WHITE_BALANCE_TEMPERATURE_AUTO_LEN 0x1
```

Kneron Confidential

6.114.1.326 WHITE_BALANCE_TEMPERATURE_LEN

```
#define WHITE_BALANCE_TEMPERATURE_LEN 0x2
```

6.114.1.327 WINDOW

```
#define WINDOW 0x100000
```

6.114.1.328 WINDOW_LEN

```
#define WINDOW_LEN 0xc
```

6.114.1.329 WINDOWS_CTL_STEPUNIT_MILLISECONDS

```
#define WINDOWS_CTL_STEPUNIT_MILLISECONDS 0x2
```

6.114.1.330 WINDOWS_CTL_STEPUNIT_VIDEOFRAME

```
#define WINDOWS_CTL_STEPUNIT_VIDEOFRAME 0x1
```

6.114.1.331 ZOOM_ABSOLUTE

```
#define ZOOM_ABSOLUTE 0x200
```

6.114.1.332 ZOOM_ABSOLUTE_LEN

```
#define ZOOM_ABSOLUTE_LEN 0x2
```

6.114.1.333 ZOOM_REL_CTL_DIGITALZOOM_OFF

```
#define ZOOM_REL_CTL_DIGITALZOOM_OFF 0x0
```

Kneron Confidential

6.114.1.334 ZOOM_REL_CTL_DIGITALZOOM_ON

```
#define ZOOM_REL_CTL_DIGITALZOOM_ON 0x1
```

6.114.1.335 ZOOM_REL_CTL_ZOOM_STOP

```
#define ZOOM_REL_CTL_ZOOM_STOP 0x0
```

6.114.1.336 ZOOM_REL_CTL_ZOOM_TO_TELEPHOTO_DIR

```
#define ZOOM_REL_CTL_ZOOM_TO_TELEPHOTO_DIR 0x1
```

6.114.1.337 ZOOM_REL_CTL_ZOOM_TO_WIDEANGLE_DIR

```
#define ZOOM_REL_CTL_ZOOM_TO_WIDEANGLE_DIR 0xFF
```

6.114.1.338 ZOOM_RELATIVE

```
#define ZOOM_RELATIVE 0x400
```

6.114.1.339 ZOOM_RELATIVE_LEN

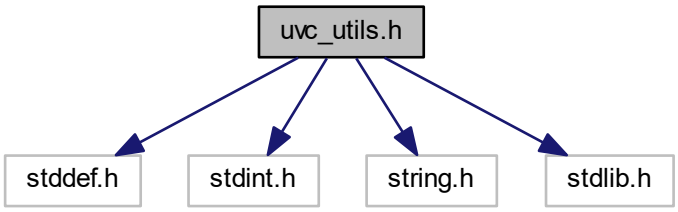
```
#define ZOOM_RELATIVE_LEN 0x3
```

Kneron Confidential

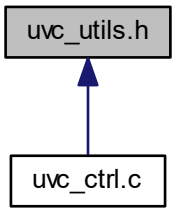
6.115 uvc_utils.h File Reference

```
#include <stddef.h>
#include <stdint.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for uvc_utils.h:



This graph shows which files directly or indirectly include this file:



Kneron Confidential

Macros

- #define PAGE_SHIFT 12
- #define PAGE_SIZE (1 << PAGE_SHIFT)
- #define PAGE_MASK (~(PAGE_SIZE-1))
- #define ___PASTE(a, b) a##b
- #define __UNIQUE_ID(prefix) __PASTE(__PASTE(__UNIQUE_ID_, prefix), __COUNTER__)
- #define __min(t1, t2, min1, min2, x, y)
- #define min(x, y)
- #define __max(t1, t2, max1, max2, x, y)
- #define max(x, y)
- #define min3(x, y, z) min((typeof(x))min(x, y), z)
- #define max3(x, y, z) max((typeof(x))max(x, y), z)
- #define min_not_zero(x, y)

- #define clamp(val, lo, hi) min((typeof(val))max(val, lo), hi)
- #define min_t(type, x, y)
- #define max_t(type, x, y)
- #define clamp_t(type, val, lo, hi) min_t(type, max_t(type, val, lo), hi)
- #define clamp_val(val, lo, hi) clamp_t(typeof(val), val, lo, hi)
- #define swap(a, b) do { typeof(a) __tmp = (a); (a) = (b); (b) = __tmp; } while (0)
- #define container_of(ptr, type, member)
- #define ARRAY_SIZE(arr) (sizeof(arr) / sizeof((arr)[0]))

6.115.1 Macro Definition Documentation

6.115.1.1 ___PASTE

```
#define ___PASTE(
                a,
                b ) a##b
```

6.115.1.2 __max

```
#define __max(
                t1,
                t2,
                max1,
                max2,
                x,
                y )
```

Value:

```
({      \
    t1 max1 = (x);              \
    t2 max2 = (y);              \
    (void) (&max1 == &max2);            \
    max1 > max2 ? max1 : max2; })
```

6.115.1.3 __min

```
#define __min(
                t1,
                t2,
                min1,
                min2,
                x,
                y )
```

Value:

```
({      \
    t1 min1 = (x);                  \
    t2 min2 = (y);                  \
    (void) (&min1 == &min2);                \
    min1 < min2 ? min1 : min2; })
```

Kneron Confidential

6.115.1.4 __UNIQUE_ID

```
#define __UNIQUE_ID(
              prefix ) __PASTE(__PASTE(__UNIQUE_ID_, prefix), __COUNTER__)
```

6.115.1.5 ARRAY_SIZE

```
#define ARRAY_SIZE(
              arr ) (sizeof(arr) / sizeof((arr)[0]))
```

6.115.1.6 clamp

```
#define clamp(
              val,
              lo,
              hi ) min((typeof(val))max(val, lo), hi)
```

clamp - return a value clamped to a given range with strict typechecking @val: current value @lo: lowest allowable value @hi: highest allowable value

This macro does strict typechecking of lo/hi to make sure they are of the same type as val. See the unnecessary pointer comparisons.

6.115.1.7 clamp_t

```
#define clamp_t(
              type,
              val,
              lo,
              hi ) min_t(type, max_t(type, val, lo), hi)
```

clamp_t - return a value clamped to a given range using a given type @type: the type of variable to use @val: current value @lo: minimum allowable value @hi: maximum allowable value

This macro does no typechecking and uses temporary variables of type 'type' to make all the comparisons.

6.115.1.8 clamp_val

```
#define clamp_val(
              val,
              lo,
              hi ) clamp_t(typeof(val), val, lo, hi)
```

clamp_val - return a value clamped to a given range using val's type @val: current value @lo: minimum allowable value @hi: maximum allowable value

This macro does no typechecking and uses temporary variables of whatever type the input argument 'val' is. This is useful when val is an unsigned type and min and max are literals that will otherwise be assigned a signed integer type.

Kneron Confidential

6.115.1.9 container_of

```
#define container_of(
                ptr,
                type,
                member )
```

Value:

```
({          \
    const typeof( ((type *)0)->member ) *__mptr = (ptr);    \
    (type *)( (char *)__mptr - offsetof(type,member) );})
```

container_of - cast a member of a structure out to the containing structure @ptr: the pointer to the member. @type: the type of the container struct this is embedded in. @member: the name of the member within the struct.

6.115.1.10 max

```
#define max(
                x,
                y )
```

Value:

```
__max(typeof(x), typeof(y),           \
          __UNIQUE_ID(max1_), __UNIQUE_ID(max2_), \
          x, y)
```

6.115.1.11 max3

```
#define max3(
                x,
                y,
                z ) max((typeof(x))max(x, y), z)
```

6.115.1.12 max_t

```
#define max_t(
                type,
                x,
                y )
```

Value:

```
__max(type, type,               \
          __UNIQUE_ID(min1_), __UNIQUE_ID(min2_), \
          x, y)
```

Kneron Confidential

6.115.1.13 min

```
#define min(
                x,
                y )
```

Value:

```
__min(typeof(x), typeof(y),           \
          __UNIQUE_ID(min1_), __UNIQUE_ID(min2_), \
          x, y)
```

6.115.1.14 min3

```
#define min3(
                x,
                y,
                z ) min((typeof(x))min(x, y), z)
```

6.115.1.15 min_not_zero

```
#define min_not_zero(
                x,
                y )
```

Value:

```
({          \
    typeof(x) __x = (x);            \
    typeof(y) __y = (y);            \
    __x == 0 ? __y : ((__y == 0) ? __x : min(__x, __y)); })
```

min_not_zero - return the minimum that is *not* zero, unless both are zero @x: value1 @y: value2

6.115.1.16 min_t

```
#define min_t(
                type,
                x,
                y )
```

Value:

```
__min(type, type,             \
          __UNIQUE_ID(min1_), __UNIQUE_ID(min2_), \
          x, y)
```

6.115.1.17 PAGE_MASK

```
#define PAGE_MASK (~(PAGE_SIZE-1))
```

Kneron Confidential

6.115.1.18 PAGE_SHIFT

```
#define PAGE_SHIFT 12
```

6.115.1.19 PAGE_SIZE

```
#define PAGE_SIZE (1 << PAGE_SHIFT)
```

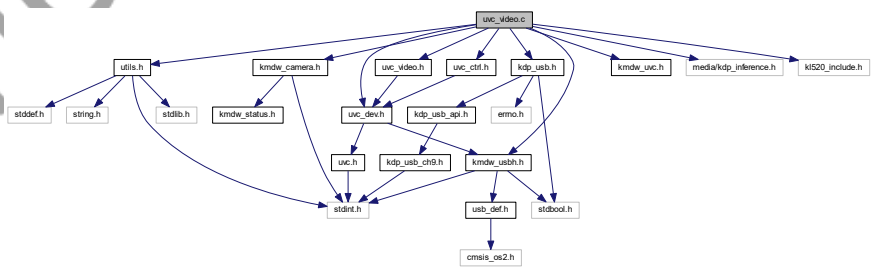
6.115.1.20 swap

```
#define swap(
             a,
             b ) do { typeof(a) __tmp = (a); (a) = (b); (b) = __tmp; } while (0)
```

6.116 uvc_video.c File Reference

```
#include <utils.h>
#include <kdp_usb.h>
#include <uvc_video.h>
#include <uvc_ctrl.h>
#include <uvc_dev.h>
#include "kmdw_usbh.h"
#include "kmdw_uvc.h"
#include "kmdw_camera.h"
#include "media/kdp_inference.h"
#include "kl520_include.h"
```

Include dependency graph for uvc_video.c:



Macros

- #define NSEC_PER_SEC 1000000000L

Kneron Confidential

6.116.1 Macro Definition Documentation

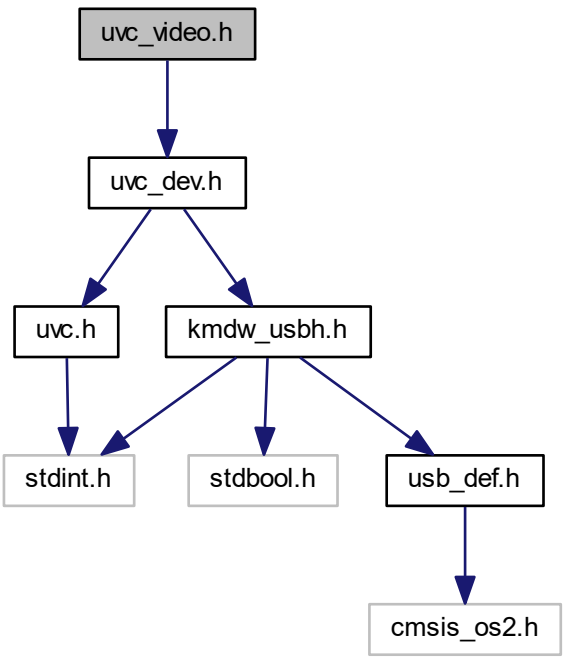
6.116.1.1 NSEC_PER_SEC

```
#define NSEC_PER_SEC 1000000000L
```

6.117 uvc_video.h File Reference

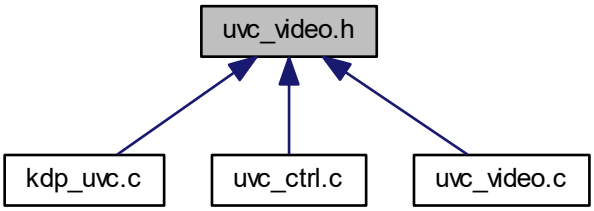
```
#include <uvc_dev.h>
```

Include dependency graph for uvc_video.h:



Kneron Confidential

This graph shows which files directly or indirectly include this file:



Functions

- int uvc_video_init (struct uvc_device ∗dev)
- int uvc_video_enable (struct uvc_streaming ∗stream)
- int uvc_video_disable (struct uvc_streaming ∗stream)

6.117.1 Function Documentation

6.117.1.1 uvc_video_disable()

```
int uvc_video_disable (
            struct uvc_streaming * stream )
```

6.117.1.2 uvc_video_enable()

```
int uvc_video_enable (
            struct uvc_streaming * stream )
```

6.117.1.3 uvc_video_init()

```
int uvc_video_init (
            struct uvc_device * dev )
```

Kneron Confidential

Index

Kneron Confidential

Kneron Confidential