Digital Watch

Generated by Doxygen 1.8.18

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 DMA_ChannelCfg_t Struct Reference

`#include <DMA.h>`

Data Fields

- u8 peri
- u8 channel
- u16 data_count
- u32 data_direction
- void * peripheral_address
- void * memory_address
- u32 channel_priority
- u32 circular_mode_control
- u32 mem2mem_mode_control
- u32 peripheral_inc_mode_control
- u32 memory_inc_mode_control
- u32 peripheral_size
- u32 memory_size
- u32 transfer_error_int_control
- u32 half_transfer_int_control
- u32 transfer_complete_int_control
- DMA_Callback_t transfer_error_callback
- DMA_Callback_t half_transfer_callback
- DMA_Callback_t transfer_complete_callback

3.1.1 Detailed Description

Definition at line 11 of file DMA.h.

3.1.2 Field Documentation

3.1.2.1 channel

```
u8 DMA_ChannelCfg_t::channel
```

DMA channel select:
DMA_CHANNEL1,
DMA_CHANNEL2,
DMA_CHANNEL3,
DMA_CHANNEL4,
DMA_CHANNEL5,
DMA_CHANNEL6,
DMA_CHANNEL7

Definition at line 30 of file DMA.h.

3.1.2.2 channel_priority

```
u32 DMA_ChannelCfg_t::channel_priority
```

DMA channel priority:
DMA_CHANNEL_PRIORITY_LOW,
DMA_CHANNEL_PRIORITY_MEDIUM,
DMA_CHANNEL_PRIORITY_HIGH,
DMA_CHANNEL_PRIORITY_VERY_HIGH

Definition at line 64 of file DMA.h.

3.1.2.3 circular_mode_control

```
u32 DMA_ChannelCfg_t::circular_mode_control
```

DMA channel circular mode enable/disable:
DMA_CIRCULAR_MODE_CONTROL_OFF,
DMA_CIRCULAR_MODE_CONTROL_ON

Definition at line 72 of file DMA.h.

3.1.2.4 data_count

```
u16 DMA_ChannelCfg_t::data_count
```

DMA channel transfer data unit count/length: [0 -> 65535]

Definition at line 36 of file DMA.h.

3.1.2.5 data_direction

```
u32 DMA_ChannelCfg_t::data_direction
```

DMA channel transfer direction:
DMA_DATA_DIRECTION_PERI2MEM,
DMA_DATA_DIRECTION_MEM2PERI

Definition at line 43 of file DMA.h.

3.1.2.6 half_transfer_callback

```
DMA_Callback_t DMA_ChannelCfg_t::half_transfer_callback
```

DMA channel half-transfer event callback

Definition at line 144 of file DMA.h.

3.1.2.7 half_transfer_int_control

```
u32 DMA_ChannelCfg_t::half_transfer_int_control
```

DMA channel half-transfer event interrupt enable/disable:
DMA_HALF_TRANSFER_INT_CONTROL_OFF,
DMA_HALF_TRANSFER_INT_CONTROL_ON

Definition at line 126 of file DMA.h.

3.1.2.8 mem2mem_mode_control

```
u32 DMA_ChannelCfg_t::mem2mem_mode_control
```

DMA channel memory-to-memory mode enable/disable:
DMA_MEM2MEM_MODE_CONTROL_OFF,
DMA_MEM2MEM_MODE_CONTROL_ON

Definition at line 79 of file DMA.h.

3.1.2.9 memory_address

```
void* DMA_ChannelCfg_t::memory_address
```

DMA channel memory address

Definition at line 54 of file DMA.h.

3.1.2.10 memory_inc_mode_control

```
u32 DMA_ChannelCfg_t::memory_inc_mode_control
```

DMA channel memory address increment mode enable/disable:
DMA_MEMORY_INC_MODE_CONTROL_OFF,
DMA_MEMORY_INC_MODE_CONTROL_ON

Definition at line 94 of file DMA.h.

3.1.2.11 memory_size

```
u32 DMA_ChannelCfg_t::memory_size
```

DMA channel memory data unit size:
DMA_MEMORY_SIZE_8_BIT,
DMA_MEMORY_SIZE_16_BIT,
DMA_MEMORY_SIZE_32_BIT

Definition at line 111 of file DMA.h.

3.1.2.12 peri

```
u8 DMA_ChannelCfg_t::peri
```

DMA peripheral select:
DMA_PERI1,
DMA_PERI2

Definition at line 18 of file DMA.h.

3.1.2.13 peripheral_address

```
void* DMA_ChannelCfg_t::peripheral_address
```

DMA channel peripheral address

Definition at line 49 of file DMA.h.

3.1.2.14 peripheral_inc_mode_control

```
u32 DMA_ChannelCfg_t::peripheral_inc_mode_control
```

DMA channel peripheral address increment mode enable/disable:
DMA_PERIPHERAL_INC_MODE_CONTROL_OFF,
DMA_PERIPHERAL_INC_MODE_CONTROL_ON

Definition at line 87 of file DMA.h.

3.1.2.15 peripheral_size

```
u32 DMA_ChannelCfg_t::peripheral_size
```

DMA channel peripheral data unit size:
DMA_PERIPHERAL_SIZE_8_BIT,
DMA_PERIPHERAL_SIZE_16_BIT,
DMA_PERIPHERAL_SIZE_32_BIT

Definition at line 103 of file DMA.h.

3.1.2.16 transfer_complete_callback

```
DMA_Callback_t DMA_ChannelCfg_t::transfer_complete_callback
```

DMA channel transfer-complete event callback

Definition at line 149 of file DMA.h.

3.1.2.17 transfer_complete_int_control

```
u32 DMA_ChannelCfg_t::transfer_complete_int_control
```

DMA channel transfer-complete event interrupt enable/disable:
DMA_TRANSFER_COMPLETE_INT_CONTROL_OFF,
DMA_TRANSFER_COMPLETE_INT_CONTROL_ON

Definition at line 133 of file DMA.h.

3.1.2.18 transfer_error_callback

```
DMA_Callback_t DMA_ChannelCfg_t::transfer_error_callback
```

DMA channel transfer-error event callback

Definition at line 139 of file DMA.h.

3.1.2.19 transfer_error_int_control

```
u32 DMA_ChannelCfg_t::transfer_error_int_control
```

DMA channel transfer-error event interrupt enable/disable:
DMA_TRANSFER_ERROR_INT_CONTROL_OFF,
DMA_TRANSFER_ERROR_INT_CONTROL_ON

Definition at line 119 of file DMA.h.

The documentation for this struct was generated from the following file:

- include/DMA.h

3.2 GPIO_t Struct Reference

```
#include <GPIO.h>
```

Data Fields

- u64 pin
- u64 mode
- u64 speed
- u64 port

3.2.1 Detailed Description

Definition at line 360 of file GPIO.h.

3.2.2 Field Documentation

3.2.2.1 mode

```
u64 GPIO_t::mode
```

Definition at line 363 of file GPIO.h.

3.2.2.2 pin

```
u64 GPIO_t::pin
```

Definition at line 362 of file GPIO.h.

3.2.2.3 port

```
u64 GPIO_t::port
```

Definition at line 365 of file GPIO.h.

3.2.2.4 speed

```
u64 GPIO_t::speed
```

Definition at line 364 of file GPIO.h.

The documentation for this struct was generated from the following file:

- include/GPIO.h

3.3 LCD_cfg_t Struct Reference

```
#include <LCD_cfg.h>
```

Data Fields

- GPIO_t lcd_IO_DB [4]
- GPIO_t lcd_IO_RS
- GPIO_t lcd_IO_RW
- GPIO_t lcd_IO_E

3.3.1 Detailed Description

Definition at line 18 of file LCD_cfg.h.

3.3.2 Field Documentation

3.3.2.1 lcd_IO_DB

```
GPIO_t LCD_cfg_t::lcd_IO_DB[4]
```

Definition at line 20 of file LCD_cfg.h.

3.3.2.2 lcd_IO_E

```
GPIO_t LCD_cfg_t::lcd_IO_E
```

Definition at line 23 of file LCD_cfg.h.

3.3.2.3 lcd_IO_RS

```
GPIO_t LCD_cfg_t::lcd_IO_RS
```

Definition at line 21 of file LCD_cfg.h.

3.3.2.4 lcd_IO_RW

```
GPIO_t LCD_cfg_t::lcd_IO_RW
```

Definition at line 22 of file LCD_cfg.h.

The documentation for this struct was generated from the following file:

- include/LCD_cfg.h

3.4 SCHED_systask_info_t Struct Reference

```
#include <sched_config.h>
```

Data Fields

- SCHED_task_t const * apptask
- u32 delayMs

3.4.1 Detailed Description

Definition at line 21 of file sched_config.h.

3.4.2 Field Documentation

3.4.2.1 apptask

```
SCHED_task_t const* SCHED_systask_info_t::apptask
```

Definition at line 23 of file sched_config.h.

3.4.2.2 delayMs

```
u32 SCHED_systask_info_t::delayMs
```

Definition at line 24 of file sched_config.h.

The documentation for this struct was generated from the following file:

- include/sched_config.h

3.5 SCHED_task_t Struct Reference

```
#include <sched_interface.h>
```

Data Fields

- SCHED_task_runnable_t runnable
- u32 periodicTimeMs

3.5.1 Detailed Description

Definition at line 15 of file sched_interface.h.

3.5.2 Field Documentation

3.5.2.1 periodicTimeMs

```
u32 SCHED_task_t::periodicTimeMs
```

Definition at line 18 of file sched_interface.h.

3.5.2.2 runnable

```
SCHED_task_runnable_t SCHED_task_t::runnable
```

Definition at line 17 of file sched_interface.h.

The documentation for this struct was generated from the following file:

- include/sched_interface.h

3.6 Switch_cfg_t Struct Reference

```
#include <switch_config.h>
```

Data Fields

- GPIO_t switchIO
- u8 activeState

3.6.1 Detailed Description

Definition at line 16 of file switch_config.h.

3.6.2 Field Documentation

3.6.2.1 activeState

```
u8 Switch_cfg_t::activeState
```

Definition at line 19 of file switch_config.h.

3.6.2.2 switchIO

```
GPIO_t Switch_cfg_t::switchIO
```

Definition at line 18 of file switch_config.h.

The documentation for this struct was generated from the following file:

- include/switch_config.h

Chapter 4

File Documentation

4.1 include/DMA.h File Reference

Data Structures

- struct DMA_ChannelCfg_t

Macros

- #define DMA_PERI1 0

  Selector for DMA peripheral 1.
- #define DMA_PERI2 1

  Selector for DMA peripheral 2.
- #define DMA_CHANNEL1 0

  Selector for DMA channel 1.
- #define DMA_CHANNEL2 1

  Selector for DMA channel 2.
- #define DMA_CHANNEL3 2

  Selector for DMA channel 3.
- #define DMA_CHANNEL4 3

  Selector for DMA channel 4.
- #define DMA_CHANNEL5 4

  Selector for DMA channel 5.
- #define DMA_CHANNEL6 5

  Selector for DMA channel 6.
- #define DMA_CHANNEL7 6

  Selector for DMA channel 7.
- #define DMA_CHANNEL_PRIORITY_LOW 0x00000000

  Selector for channel priority: Low.
- #define DMA_CHANNEL_PRIORITY_MEDIUM 0x00001000

  Selector for channel priority: Medium.
- #define DMA_CHANNEL_PRIORITY_HIGH 0x00002000

  Selector for channel priority: High.
- #define DMA_CHANNEL_PRIORITY_VERY_HIGH 0x00003000

  Selector for channel priority: Very High.

- #define DMA_MEMORY_SIZE_8_BIT 0x00000000
  Selector for the size of the memory data unit: 8-bit (1 byte)
- #define DMA_MEMORY_SIZE_16_BIT 0x00000400
  Selector for the size of the memory data unit: 16-bit (2 bytes)
- #define DMA_MEMORY_SIZE_32_BIT 0x00000800
  Selector for the size of the memory data unit: 32-bit (4 bytes)
- #define DMA_PERIPHERAL_SIZE_8_BIT 0x00000000
  Selector for the size of the peripheral data unit: 8-bit (1 byte)
- #define DMA_PERIPHERAL_SIZE_16_BIT 0x00000100
  Selector for the size of the peripheral data unit: 16-bit (2 byte)
- #define DMA_PERIPHERAL_SIZE_32_BIT 0x00000200
  Selector for the size of the peripheral data unit: 32-bit (4 byte)
- #define DMA_DATA_DIRECTION_PERI2MEM 0x00000000
  Select the channel transfer direction as: peripheral -> memory.
- #define DMA_DATA_DIRECTION_MEM2PERI 0x00000010
  Select the channel transfer direction as: memory -> peripheral.
- #define DMA_MEM2MEM_MODE_CONTROL_OFF 0x00000000
  Control selector for the memory-to-memory mode: off.
- #define DMA_MEM2MEM_MODE_CONTROL_ON 0x00004000
  Control selector for the memory-to-memory mode: on.
- #define DMA_MEMORY_INC_MODE_CONTROL_OFF 0x00000000
  Control selector for the memory increment mode: off.
- #define DMA_MEMORY_INC_MODE_CONTROL_ON 0x00000080
  Control selector for the memory increment mode: on.
- #define DMA_PERIPHERAL_INC_MODE_CONTROL_OFF 0x00000000
  Control selector for the peripheral increment mode: off.
- #define DMA_PERIPHERAL_INC_MODE_CONTROL_ON 0x00000040
  Control selector for the peripheral increment mode: on.
- #define DMA_CIRCULAR_MODE_CONTROL_OFF 0x00000000
  Control selector for the circular mode: off.
- #define DMA_CIRCULAR_MODE_CONTROL_ON 0x00000020
  Control selector for the circular mode: on.
- #define DMA_TRANSFER_ERROR_INT_CONTROL_OFF 0x00000000
  Control selector for the transfer-error event interrupt: off.
- #define DMA_TRANSFER_ERROR_INT_CONTROL_ON 0x00000008
  Control selector for the transfer-error event interrupt: on.
- #define DMA_HALF_TRANSFER_INT_CONTROL_OFF 0x00000000
  Control selector for the half-transfer event interrupt: off.
- #define DMA_HALF_TRANSFER_INT_CONTROL_ON 0x00000004
  Control selector for the half-transfer event interrupt: on.
- #define DMA_TRANSFER_COMPLETE_INT_CONTROL_OFF 0x00000000
  Control selector for the transfer-complete event interrupt: off.
- #define DMA_TRANSFER_COMPLETE_INT_CONTROL_ON 0x00000002
  Control selector for the transfer-complete event interrupt: on.
- #define DMA_CHANNEL_CONTROL_OFF 0x00000000
  Control selector for the channel state: off.
- #define DMA_CHANNEL_CONTROL_ON 0x00000001
  Control selector for the channel state: on.
- #define DMA_CALLBACK_TYPE_TRANSFER_ERROR 0
  Selector for the callback trigger event type: transfer-error.
- #define DMA_CALLBACK_TYPE_TRANSFER_COMPLETE 1
  Selector for the callback trigger event type: transfer-complete.
- #define DMA_CALLBACK_TYPE_HALF_TRANSFER 2
  Selector for the callback trigger event type: half-transfer.

Typedefs

- typedef void(∗ DMA_Callback_t) (void)

  DMA callback type, used when registering a channel event callback.

Functions

- void DMA_InitChannel (const DMA_ChannelCfg_t ∗cfg)

  Initialize a DMA channel and reset its configurations.
- u8 DMA_GetIsErrorTransfer (const DMA_ChannelCfg_t ∗cfg)

  Get the state of the transfer-error channel flag.
- u8 DMA_GetIsHalfTransfer (const DMA_ChannelCfg_t ∗cfg)

  Get the state of the half-transfer channel flag.
- u8 DMA_GetIsTransferComplete (const DMA_ChannelCfg_t ∗cfg)

  Get the state of the transfer-complete channel flag.
- u8 DMA_GetIsGlobalInterrupt (const DMA_ChannelCfg_t ∗cfg)

  Get the state of the global channel flag.
- void DMA_ConfigChannel (const DMA_ChannelCfg_t ∗cfg)
- void DMA_ControlChannel (const DMA_ChannelCfg_t ∗cfg, u32 channel_control)

  Enable/Disable a channel.

4.1.1 Macro Definition Documentation

4.1.1.1 DMA_CALLBACK_TYPE_HALF_TRANSFER

```
#define DMA_CALLBACK_TYPE_HALF_TRANSFER 2
```

Selector for the callback trigger event type: half-transfer.

Definition at line 403 of file DMA.h.

4.1.1.2 DMA_CALLBACK_TYPE_TRANSFER_COMPLETE

```
#define DMA_CALLBACK_TYPE_TRANSFER_COMPLETE 1
```

Selector for the callback trigger event type: transfer-complete.

Definition at line 397 of file DMA.h.

4.1.1.3 DMA_CALLBACK_TYPE_TRANSFER_ERROR

```
#define DMA_CALLBACK_TYPE_TRANSFER_ERROR 0
```

Selector for the callback trigger event type: transfer-error.

Definition at line 391 of file DMA.h.

4.1.1.4 DMA_CHANNEL1

```
#define DMA_CHANNEL1 0
```

Selector for DMA channel 1.

Definition at line 171 of file DMA.h.

4.1.1.5 DMA_CHANNEL2

```
#define DMA_CHANNEL2 1
```

Selector for DMA channel 2.

Definition at line 177 of file DMA.h.

4.1.1.6 DMA_CHANNEL3

```
#define DMA_CHANNEL3 2
```

Selector for DMA channel 3.

Definition at line 183 of file DMA.h.

4.1.1.7 DMA_CHANNEL4

```
#define DMA_CHANNEL4 3
```

Selector for DMA channel 4.

Definition at line 189 of file DMA.h.

4.1.1.8 DMA_CHANNEL5

```
#define DMA_CHANNEL5 4
```

Selector for DMA channel 5.

Definition at line 195 of file DMA.h.

4.1.1.9 DMA_CHANNEL6

```
#define DMA_CHANNEL6 5
```

Selector for DMA channel 6.

Definition at line 201 of file DMA.h.

4.1.1.10 DMA_CHANNEL7

```
#define DMA_CHANNEL7 6
```

Selector for DMA channel 7.

Definition at line 207 of file DMA.h.

4.1.1.11 DMA_CHANNEL_CONTROL_OFF

```
#define DMA_CHANNEL_CONTROL_OFF 0x00000000
```

Control selector for the channel state: off.

Definition at line 377 of file DMA.h.

4.1.1.12 DMA_CHANNEL_CONTROL_ON

```
#define DMA_CHANNEL_CONTROL_ON 0x00000001
```

Control selector for the channel state: on.

Definition at line 383 of file DMA.h.

4.1.1.13 DMA_CHANNEL_PRIORITY_HIGH

```
#define DMA_CHANNEL_PRIORITY_HIGH 0x00002000
```

Selector for channel priority: High.

Definition at line 227 of file DMA.h.

4.1.1.14 DMA_CHANNEL_PRIORITY_LOW

```
#define DMA_CHANNEL_PRIORITY_LOW 0x00000000
```

Selector for channel priority: Low.

Definition at line 215 of file DMA.h.

4.1.1.15 DMA_CHANNEL_PRIORITY_MEDIUM

```
#define DMA_CHANNEL_PRIORITY_MEDIUM 0x00001000
```

Selector for channel priority: Medium.

Definition at line 221 of file DMA.h.

4.1.1.16 DMA_CHANNEL_PRIORITY_VERY_HIGH

```
#define DMA_CHANNEL_PRIORITY_VERY_HIGH 0x00003000
```

Selector for channel priority: Very High.

Definition at line 233 of file DMA.h.

4.1.1.17 DMA_CIRCULAR_MODE_CONTROL_OFF

```
#define DMA_CIRCULAR_MODE_CONTROL_OFF 0x00000000
```

Control selector for the circular mode: off.

Definition at line 329 of file DMA.h.

4.1.1.18 DMA_CIRCULAR_MODE_CONTROL_ON

```
#define DMA_CIRCULAR_MODE_CONTROL_ON 0x00000020
```

Control selector for the circular mode: on.

Definition at line 335 of file DMA.h.

4.1.1.19 DMA_DATA_DIRECTION_MEM2PERI

```
#define DMA_DATA_DIRECTION_MEM2PERI 0x00000010
```

Select the channel transfer direction as: memory -> peripheral.

Definition at line 285 of file DMA.h.

4.1.1.20 DMA_DATA_DIRECTION_PERI2MEM

```
#define DMA_DATA_DIRECTION_PERI2MEM 0x00000000
```

Select the channel transfer direction as: peripheral -> memory.

Definition at line 279 of file DMA.h.

4.1.1.21 DMA_HALF_TRANSFER_INT_CONTROL_OFF

```
#define DMA_HALF_TRANSFER_INT_CONTROL_OFF 0x00000000
```

Control selector for the half-transfer event interrupt: off.

Definition at line 353 of file DMA.h.

4.1.1.22 DMA_HALF_TRANSFER_INT_CONTROL_ON

```
#define DMA_HALF_TRANSFER_INT_CONTROL_ON 0x00000004
```

Control selector for the half-transfer event interrupt: on.

Definition at line 359 of file DMA.h.

4.1.1.23 DMA_MEM2MEM_MODE_CONTROL_OFF

```
#define DMA_MEM2MEM_MODE_CONTROL_OFF 0x00000000
```

Control selector for the memory-to-memory mode: off.

Definition at line 293 of file DMA.h.

4.1.1.24 DMA_MEM2MEM_MODE_CONTROL_ON

```
#define DMA_MEM2MEM_MODE_CONTROL_ON 0x00004000
```

Control selector for the memory-to-memory mode: on.

Definition at line 299 of file DMA.h.

4.1.1.25 DMA_MEMORY_INC_MODE_CONTROL_OFF

```
#define DMA_MEMORY_INC_MODE_CONTROL_OFF 0x00000000
```

Control selector for the memory increment mode: off.

Definition at line 305 of file DMA.h.

4.1.1.26 DMA_MEMORY_INC_MODE_CONTROL_ON

```
#define DMA_MEMORY_INC_MODE_CONTROL_ON 0x00000080
```

Control selector for the memory increment mode: on.

Definition at line 311 of file DMA.h.

4.1.1.27 DMA_MEMORY_SIZE_16_BIT

```
#define DMA_MEMORY_SIZE_16_BIT 0x00000400
```

Selector for the size of the memory data unit: 16-bit (2 bytes)

Definition at line 247 of file DMA.h.

4.1.1.28 DMA_MEMORY_SIZE_32_BIT

```
#define DMA_MEMORY_SIZE_32_BIT 0x00000800
```

Selector for the size of the memory data unit: 32-bit (4 bytes)

Definition at line 253 of file DMA.h.

4.1.1.29 DMA_MEMORY_SIZE_8_BIT

```
#define DMA_MEMORY_SIZE_8_BIT 0x00000000
```

Selector for the size of the memory data unit: 8-bit (1 byte)

Definition at line 241 of file DMA.h.

4.1.1.30 DMA_PERI1

```
#define DMA_PERI1 0
```

Selector for DMA peripheral 1.

Definition at line 157 of file DMA.h.

4.1.1.31 DMA_PERI2

```
#define DMA_PERI2 1
```

Selector for DMA peripheral 2.

Definition at line 163 of file DMA.h.

4.1.1.32 DMA_PERIPHERAL_INC_MODE_CONTROL_OFF

```
#define DMA_PERIPHERAL_INC_MODE_CONTROL_OFF 0x00000000
```

Control selector for the peripheral increment mode: off.

Definition at line 317 of file DMA.h.

4.1.1.33 DMA_PERIPHERAL_INC_MODE_CONTROL_ON

```
#define DMA_PERIPHERAL_INC_MODE_CONTROL_ON 0x00000040
```

Control selector for the peripheral increment mode: on.

Definition at line 323 of file DMA.h.

4.1.1.34 DMA_PERIPHERAL_SIZE_16_BIT

```
#define DMA_PERIPHERAL_SIZE_16_BIT 0x00000100
```

Selector for the size of the peripheral data unit: 16-bit (2 byte)

Definition at line 265 of file DMA.h.

4.1.1.35 DMA_PERIPHERAL_SIZE_32_BIT

```
#define DMA_PERIPHERAL_SIZE_32_BIT 0x00000200
```

Selector for the size of the peripheral data unit: 32-bit (4 byte)

Definition at line 271 of file DMA.h.

4.1.1.36 DMA_PERIPHERAL_SIZE_8_BIT

```
#define DMA_PERIPHERAL_SIZE_8_BIT 0x00000000
```

Selector for the size of the peripheral data unit: 8-bit (1 byte)

Definition at line 259 of file DMA.h.

4.1.1.37 DMA_TRANSFER_COMPLETE_INT_CONTROL_OFF

```
#define DMA_TRANSFER_COMPLETE_INT_CONTROL_OFF 0x00000000
```

Control selector for the transfer-complete event interrupt: off.

Definition at line 365 of file DMA.h.

4.1.1.38 DMA_TRANSFER_COMPLETE_INT_CONTROL_ON

```
#define DMA_TRANSFER_COMPLETE_INT_CONTROL_ON 0x00000002
```

Control selector for the transfer-complete event interrupt: on.

Definition at line 371 of file DMA.h.

4.1.1.39 DMA_TRANSFER_ERROR_INT_CONTROL_OFF

```
#define DMA_TRANSFER_ERROR_INT_CONTROL_OFF 0x00000000
```

Control selector for the transfer-error event interrupt: off.

Definition at line 341 of file DMA.h.

4.1.1.40 DMA_TRANSFER_ERROR_INT_CONTROL_ON

```
#define DMA_TRANSFER_ERROR_INT_CONTROL_ON 0x00000008
```

Control selector for the transfer-error event interrupt: on.

Definition at line 347 of file DMA.h.

4.1.2 Typedef Documentation

4.1.2.1 DMA_Callback_t

```
typedef void(* DMA_Callback_t) (void)
```

DMA callback type, used when registering a channel event callback.

Definition at line 9 of file DMA.h.

4.1.3 Function Documentation

4.1.3.1 DMA_ConfigChannel()

```
void DMA_ConfigChannel (
            const DMA_ChannelCfg_t * cfg )
```

4.1.3.2 DMA_ControlChannel()

```
void DMA_ControlChannel (
            const DMA_ChannelCfg_t * cfg,
            u32 channel_control )
```

Enable/Disable a channel.

Parameters

cfg	Reference to an initialized DMA_ChannelCfg_t object that holds all the channel configurations
channel_control	The state of the channel: DMA_CHANNEL_CONTROL_OFF DMA_CHANNEL_CONTROL_ON

4.1.3.3 DMA_GetIsErrorTransfer()

```
u8 DMA_GetIsErrorTransfer (
            const DMA_ChannelCfg_t * cfg )
```

Get the state of the transfer-error channel flag.

Parameters

cfg	Reference to an initialized DMA_ChannelCfg_t object that holds all the channel configurations

Returns

Boolean value: 1 => a transfer-error has occurred, 0 => no transfer-error occurred

4.1.3.4 DMA_GetIsGlobalInterrupt()

```
u8 DMA_GetIsGlobalInterrupt (
            const DMA_ChannelCfg_t * cfg )
```

Get the state of the global channel flag.

Parameters

cfg	Reference to an initialized DMA_ChannelCfg_t object that holds all the channel configurations

Returns

Boolean value: 1 => an event has happened on the channel:

- Transfer error event (use DMA_GetIsErrorTransfer),
- Half transfer complete event (use DMA_GetIsHalfTransfer),
- Transfer complete event (use DMA_GetIsTransferComplete) 0 => no event happened on the channel

4.1.3.5 DMA_GetIsHalfTransfer()

```
u8 DMA_GetIsHalfTransfer (
            const DMA_ChannelCfg_t * cfg )
```

Get the state of the half-transfer channel flag.

Parameters

cfg	Reference to an initialized DMA_ChannelCfg_t object that holds all the channel configurations

Returns

Boolean value: 1 => half-transfer has finished, 0 => half-transfer didn't finish

4.1.3.6 DMA_GetIsTransferComplete()

```
u8 DMA_GetIsTransferComplete (
            const DMA_ChannelCfg_t * cfg )
```

Get the state of the transfer-complete channel flag.

Parameters

cfg	Reference to an initialized DMA_ChannelCfg_t object that holds all the channel configurations

Returns

Boolean value: 1 => transfer has finished completely, 0 => transfer didn't finish completely

4.1.3.7 DMA_InitChannel()

```
void DMA_InitChannel (
            const DMA_ChannelCfg_t * cfg )
```

Initialize a DMA channel and reset its configurations.

Parameters

cfg	Reference to an initialized DMA_ChannelCfg_t object that holds all the channel configurations

4.2 include/GPIO.h File Reference

This file is to be used as an implementation of the GPIO driver.

Data Structures

- struct GPIO_t

Macros

- #define GPIO_PIN0_VALUE_HIGH 0x000000000000000F
- #define GPIO_PIN0_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN1_VALUE_HIGH 0x00000000000000F0
- #define GPIO_PIN1_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN2_VALUE_HIGH 0x0000000000000F00
- #define GPIO_PIN2_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN3_VALUE_HIGH 0x000000000000F000
- #define GPIO_PIN3_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN4_VALUE_HIGH 0x00000000000F0000
- #define GPIO_PIN4_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN5_VALUE_HIGH 0x0000000000F00000
- #define GPIO_PIN5_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN6_VALUE_HIGH 0x000000000F000000
- #define GPIO_PIN6_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN7_VALUE_HIGH 0x00000000F0000000
- #define GPIO_PIN7_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN8_VALUE_HIGH 0x0000000F00000000
- #define GPIO_PIN8_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN9_VALUE_HIGH 0x000000F000000000
- #define GPIO_PIN9_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN10_VALUE_HIGH 0x00000F0000000000
- #define GPIO_PIN10_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN11_VALUE_HIGH 0x0000F00000000000
- #define GPIO_PIN11_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN12_VALUE_HIGH 0x000F000000000000
- #define GPIO_PIN12_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN13_VALUE_HIGH 0x00F0000000000000
- #define GPIO_PIN13_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN14_VALUE_HIGH 0x0F00000000000000
- #define GPIO_PIN14_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN15_VALUE_HIGH 0xF000000000000000
- #define GPIO_PIN15_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN_ALL_VALUE_HIGH 0xFFFFFFFFFFFFFFFF
- #define GPIO_PIN_ALL_VALUE_LOW 0x0000000000000000
- #define GPIO_PIN0_SELECT 0x000000000000000F
- #define GPIO_PIN1_SELECT 0x00000000000000F0
- #define GPIO_PIN2_SELECT 0x0000000000000F00
- #define GPIO_PIN3_SELECT 0x000000000000F000
- #define GPIO_PIN4_SELECT 0x00000000000F0000
- #define GPIO_PIN5_SELECT 0x0000000000F00000
- #define GPIO_PIN6_SELECT 0x000000000F000000
- #define GPIO_PIN7_SELECT 0x00000000F0000000

- #define GPIO_PIN8_SELECT 0x0000000F00000000
- #define GPIO_PIN9_SELECT 0x000000F000000000
- #define GPIO_PIN10_SELECT 0x00000F0000000000
- #define GPIO_PIN11_SELECT 0x0000F00000000000
- #define GPIO_PIN12_SELECT 0x000F000000000000
- #define GPIO_PIN13_SELECT 0x00F0000000000000
- #define GPIO_PIN14_SELECT 0x0F00000000000000
- #define GPIO_PIN15_SELECT 0xF000000000000000
- #define GPIO_PIN0_PORTA 0x0000000000000001
- #define GPIO_PIN0_PORTB 0x0000000000000002
- #define GPIO_PIN0_PORTC 0x0000000000000003
- #define GPIO_PIN1_PORTA 0x0000000000000010
- #define GPIO_PIN1_PORTB 0x0000000000000020
- #define GPIO_PIN1_PORTC 0x0000000000000030
- #define GPIO_PIN2_PORTA 0x0000000000000100
- #define GPIO_PIN2_PORTB 0x0000000000000200
- #define GPIO_PIN2_PORTC 0x0000000000000300
- #define GPIO_PIN3_PORTA 0x0000000000001000
- #define GPIO_PIN3_PORTB 0x0000000000002000
- #define GPIO_PIN3_PORTC 0x0000000000003000
- #define GPIO_PIN4_PORTA 0x0000000000010000
- #define GPIO_PIN4_PORTB 0x0000000000020000
- #define GPIO_PIN4_PORTC 0x0000000000030000
- #define GPIO_PIN5_PORTA 0x0000000000100000
- #define GPIO_PIN5_PORTB 0x0000000000200000
- #define GPIO_PIN5_PORTC 0x0000000000300000
- #define GPIO_PIN6_PORTA 0x0000000001000000
- #define GPIO_PIN6_PORTB 0x0000000002000000
- #define GPIO_PIN6_PORTC 0x0000000003000000
- #define GPIO_PIN7_PORTA 0x0000000010000000
- #define GPIO_PIN7_PORTB 0x0000000020000000
- #define GPIO_PIN7_PORTC 0x0000000030000000
- #define GPIO_PIN8_PORTA 0x0000000100000000
- #define GPIO_PIN8_PORTB 0x0000000200000000
- #define GPIO_PIN8_PORTC 0x0000000300000000
- #define GPIO_PIN9_PORTA 0x0000001000000000
- #define GPIO_PIN9_PORTB 0x0000002000000000
- #define GPIO_PIN9_PORTC 0x0000003000000000
- #define GPIO_PIN10_PORTA 0x0000010000000000
- #define GPIO_PIN10_PORTB 0x0000020000000000
- #define GPIO_PIN10_PORTC 0x0000030000000000
- #define GPIO_PIN11_PORTA 0x0000100000000000
- #define GPIO_PIN11_PORTB 0x0000200000000000
- #define GPIO_PIN11_PORTC 0x0000300000000000
- #define GPIO_PIN12_PORTA 0x0001000000000000
- #define GPIO_PIN12_PORTB 0x0002000000000000
- #define GPIO_PIN12_PORTC 0x0003000000000000
- #define GPIO_PIN13_PORTA 0x0010000000000000
- #define GPIO_PIN13_PORTB 0x0020000000000000
- #define GPIO_PIN13_PORTC 0x0030000000000000
- #define GPIO_PIN14_PORTA 0x0100000000000000
- #define GPIO_PIN14_PORTB 0x0200000000000000
- #define GPIO_PIN14_PORTC 0x0300000000000000
- #define GPIO_PIN15_PORTA 0x1000000000000000
- #define GPIO_PIN15_PORTB 0x2000000000000000

- #define GPIO_PIN15_PORTC 0x3000000000000000
- #define GPIO_PIN_ALL_PORTA 0x1111111111111111
- #define GPIO_PIN_ALL_PORTB 0x2222222222222222
- #define GPIO_PIN_ALL_PORTC 0x3333333333333333
- #define GPIO_PIN0_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN0_MODE_INPUT_FLOATING 0x0000000000000004
- #define GPIO_PIN0_MODE_INPUT_PULL_UP_DOWN 0x0000000000000008
- #define GPIO_PIN0_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN0_MODE_OUTPUT_OPEN_DRAIN 0x0000000000000004
- #define GPIO_PIN0_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000000008
- #define GPIO_PIN0_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000000000000C
- #define GPIO_PIN1_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN1_MODE_INPUT_FLOATING 0x0000000000000040
- #define GPIO_PIN1_MODE_INPUT_PULL_UP_DOWN 0x0000000000000080
- #define GPIO_PIN1_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN1_MODE_OUTPUT_OPEN_DRAIN 0x0000000000000040
- #define GPIO_PIN1_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000000080
- #define GPIO_PIN1_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000000000000C0
- #define GPIO_PIN2_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN2_MODE_INPUT_FLOATING 0x0000000000000400
- #define GPIO_PIN2_MODE_INPUT_PULL_UP_DOWN 0x0000000000000800
- #define GPIO_PIN2_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN2_MODE_OUTPUT_OPEN_DRAIN 0x0000000000000400
- #define GPIO_PIN2_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000000800
- #define GPIO_PIN2_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000000000000C00
- #define GPIO_PIN3_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN3_MODE_INPUT_FLOATING 0x0000000000004000
- #define GPIO_PIN3_MODE_INPUT_PULL_UP_DOWN 0x0000000000008000
- #define GPIO_PIN3_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN3_MODE_OUTPUT_OPEN_DRAIN 0x0000000000004000
- #define GPIO_PIN3_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000008000
- #define GPIO_PIN3_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000000000C000
- #define GPIO_PIN4_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN4_MODE_INPUT_FLOATING 0x0000000000040000
- #define GPIO_PIN4_MODE_INPUT_PULL_UP_DOWN 0x0000000000080000
- #define GPIO_PIN4_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN4_MODE_OUTPUT_OPEN_DRAIN 0x0000000000040000
- #define GPIO_PIN4_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000080000
- #define GPIO_PIN4_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000000000C0000
- #define GPIO_PIN5_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN5_MODE_INPUT_FLOATING 0x0000000000400000
- #define GPIO_PIN5_MODE_INPUT_PULL_UP_DOWN 0x0000000000800000
- #define GPIO_PIN5_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN5_MODE_OUTPUT_OPEN_DRAIN 0x0000000000400000
- #define GPIO_PIN5_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000800000
- #define GPIO_PIN5_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000000000C00000
- #define GPIO_PIN6_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN6_MODE_INPUT_FLOATING 0x0000000004000000
- #define GPIO_PIN6_MODE_INPUT_PULL_UP_DOWN 0x0000000008000000
- #define GPIO_PIN6_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN6_MODE_OUTPUT_OPEN_DRAIN 0x0000000004000000
- #define GPIO_PIN6_MODE_OUTPUT_AF_PUSH_PULL 0x0000000008000000
- #define GPIO_PIN6_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000000C000000
- #define GPIO_PIN7_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN7_MODE_INPUT_FLOATING 0x0000000040000000

- #define GPIO_PIN7_MODE_INPUT_PULL_UP_DOWN 0x0000000080000000
- #define GPIO_PIN7_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN7_MODE_OUTPUT_OPEN_DRAIN 0x0000000040000000
- #define GPIO_PIN7_MODE_OUTPUT_AF_PUSH_PULL 0x0000000080000000
- #define GPIO_PIN7_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000000C0000000
- #define GPIO_PIN8_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN8_MODE_INPUT_FLOATING 0x0000000400000000
- #define GPIO_PIN8_MODE_INPUT_PULL_UP_DOWN 0x0000000800000000
- #define GPIO_PIN8_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN8_MODE_OUTPUT_OPEN_DRAIN 0x0000000400000000
- #define GPIO_PIN8_MODE_OUTPUT_AF_PUSH_PULL 0x0000000800000000
- #define GPIO_PIN8_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000000C00000000
- #define GPIO_PIN9_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN9_MODE_INPUT_FLOATING 0x0000004000000000
- #define GPIO_PIN9_MODE_INPUT_PULL_UP_DOWN 0x0000008000000000
- #define GPIO_PIN9_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN9_MODE_OUTPUT_OPEN_DRAIN 0x0000004000000000
- #define GPIO_PIN9_MODE_OUTPUT_AF_PUSH_PULL 0x0000008000000000
- #define GPIO_PIN9_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000C000000000
- #define GPIO_PIN10_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN10_MODE_INPUT_FLOATING 0x0000040000000000
- #define GPIO_PIN10_MODE_INPUT_PULL_UP_DOWN 0x0000080000000000
- #define GPIO_PIN10_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN10_MODE_OUTPUT_OPEN_DRAIN 0x0000040000000000
- #define GPIO_PIN10_MODE_OUTPUT_AF_PUSH_PULL 0x0000080000000000
- #define GPIO_PIN10_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000C0000000000
- #define GPIO_PIN11_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN11_MODE_INPUT_FLOATING 0x0000400000000000
- #define GPIO_PIN11_MODE_INPUT_PULL_UP_DOWN 0x0000800000000000
- #define GPIO_PIN11_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN11_MODE_OUTPUT_OPEN_DRAIN 0x0000400000000000
- #define GPIO_PIN11_MODE_OUTPUT_AF_PUSH_PULL 0x0000800000000000
- #define GPIO_PIN11_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000C00000000000
- #define GPIO_PIN12_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN12_MODE_INPUT_FLOATING 0x0004000000000000
- #define GPIO_PIN12_MODE_INPUT_PULL_UP_DOWN 0x0008000000000000
- #define GPIO_PIN12_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN12_MODE_OUTPUT_OPEN_DRAIN 0x0004000000000000
- #define GPIO_PIN12_MODE_OUTPUT_AF_PUSH_PULL 0x0008000000000000
- #define GPIO_PIN12_MODE_OUTPUT_AF_OPEN_DRAIN 0x000C000000000000
- #define GPIO_PIN13_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN13_MODE_INPUT_FLOATING 0x0040000000000000
- #define GPIO_PIN13_MODE_INPUT_PULL_UP_DOWN 0x0080000000000000
- #define GPIO_PIN13_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN13_MODE_OUTPUT_OPEN_DRAIN 0x0040000000000000
- #define GPIO_PIN13_MODE_OUTPUT_AF_PUSH_PULL 0x0080000000000000
- #define GPIO_PIN13_MODE_OUTPUT_AF_OPEN_DRAIN 0x00C0000000000000
- #define GPIO_PIN14_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN14_MODE_INPUT_FLOATING 0x0400000000000000
- #define GPIO_PIN14_MODE_INPUT_PULL_UP_DOWN 0x0800000000000000
- #define GPIO_PIN14_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN14_MODE_OUTPUT_OPEN_DRAIN 0x0400000000000000
- #define GPIO_PIN14_MODE_OUTPUT_AF_PUSH_PULL 0x0800000000000000
- #define GPIO_PIN14_MODE_OUTPUT_AF_OPEN_DRAIN 0x0C00000000000000
- #define GPIO_PIN15_MODE_INPUT_ANALOG 0x0000000000000000

- #define GPIO_PIN15_MODE_INPUT_FLOATING 0x4000000000000000
- #define GPIO_PIN15_MODE_INPUT_PULL_UP_DOWN 0x8000000000000000
- #define GPIO_PIN15_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN15_MODE_OUTPUT_OPEN_DRAIN 0x4000000000000000
- #define GPIO_PIN15_MODE_OUTPUT_AF_PUSH_PULL 0x8000000000000000
- #define GPIO_PIN15_MODE_OUTPUT_AF_OPEN_DRAIN 0xC000000000000000
- #define GPIO_PIN_ALL_MODE_INPUT_ANALOG 0x0000000000000000
- #define GPIO_PIN_ALL_MODE_INPUT_FLOATING 0x4444444444444444
- #define GPIO_PIN_ALL_MODE_INPUT_PULL_UP_DOWN 0x8888888888888888
- #define GPIO_PIN_ALL_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
- #define GPIO_PIN_ALL_MODE_OUTPUT_OPEN_DRAIN 0x4444444444444444
- #define GPIO_PIN_ALL_MODE_OUTPUT_AF_PUSH_PULL 0x8888888888888888
- #define GPIO_PIN_ALL_MODE_OUTPUT_AF_OPEN_DRAIN 0xCCCCCCCCCCCCCCCC
- #define GPIO_PIN0_SPEED_10MHZ 0x0000000000000001
- #define GPIO_PIN0_SPEED_2MHZ 0x0000000000000002
- #define GPIO_PIN0_SPEED_50MHZ 0x0000000000000003
- #define GPIO_PIN0_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN1_SPEED_10MHZ 0x0000000000000010
- #define GPIO_PIN1_SPEED_2MHZ 0x0000000000000020
- #define GPIO_PIN1_SPEED_50MHZ 0x0000000000000030
- #define GPIO_PIN1_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN2_SPEED_10MHZ 0x0000000000000100
- #define GPIO_PIN2_SPEED_2MHZ 0x0000000000000200
- #define GPIO_PIN2_SPEED_50MHZ 0x0000000000000300
- #define GPIO_PIN2_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN3_SPEED_10MHZ 0x0000000000001000
- #define GPIO_PIN3_SPEED_2MHZ 0x0000000000002000
- #define GPIO_PIN3_SPEED_50MHZ 0x0000000000003000
- #define GPIO_PIN3_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN4_SPEED_10MHZ 0x0000000000010000
- #define GPIO_PIN4_SPEED_2MHZ 0x0000000000020000
- #define GPIO_PIN4_SPEED_50MHZ 0x0000000000030000
- #define GPIO_PIN4_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN5_SPEED_10MHZ 0x0000000000100000
- #define GPIO_PIN5_SPEED_2MHZ 0x0000000000200000
- #define GPIO_PIN5_SPEED_50MHZ 0x0000000000300000
- #define GPIO_PIN5_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN6_SPEED_10MHZ 0x0000000001000000
- #define GPIO_PIN6_SPEED_2MHZ 0x0000000002000000
- #define GPIO_PIN6_SPEED_50MHZ 0x0000000003000000
- #define GPIO_PIN6_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN7_SPEED_10MHZ 0x0000000010000000
- #define GPIO_PIN7_SPEED_2MHZ 0x0000000020000000
- #define GPIO_PIN7_SPEED_50MHZ 0x0000000030000000
- #define GPIO_PIN7_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN8_SPEED_10MHZ 0x0000000100000000
- #define GPIO_PIN8_SPEED_2MHZ 0x0000000200000000
- #define GPIO_PIN8_SPEED_50MHZ 0x0000000300000000
- #define GPIO_PIN8_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN9_SPEED_10MHZ 0x0000001000000000
- #define GPIO_PIN9_SPEED_2MHZ 0x0000002000000000
- #define GPIO_PIN9_SPEED_50MHZ 0x0000003000000000
- #define GPIO_PIN9_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN10_SPEED_10MHZ 0x0000010000000000
- #define GPIO_PIN10_SPEED_2MHZ 0x0000020000000000

- #define GPIO_PIN10_SPEED_50MHZ 0x0000030000000000
- #define GPIO_PIN10_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN11_SPEED_10MHZ 0x0000100000000000
- #define GPIO_PIN11_SPEED_2MHZ 0x0000200000000000
- #define GPIO_PIN11_SPEED_50MHZ 0x0000300000000000
- #define GPIO_PIN11_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN12_SPEED_10MHZ 0x0001000000000000
- #define GPIO_PIN12_SPEED_2MHZ 0x0002000000000000
- #define GPIO_PIN12_SPEED_50MHZ 0x0003000000000000
- #define GPIO_PIN12_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN13_SPEED_10MHZ 0x0010000000000000
- #define GPIO_PIN13_SPEED_2MHZ 0x0020000000000000
- #define GPIO_PIN13_SPEED_50MHZ 0x0030000000000000
- #define GPIO_PIN13_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN14_SPEED_10MHZ 0x0100000000000000
- #define GPIO_PIN14_SPEED_2MHZ 0x0200000000000000
- #define GPIO_PIN14_SPEED_50MHZ 0x0300000000000000
- #define GPIO_PIN14_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN15_SPEED_10MHZ 0x1000000000000000
- #define GPIO_PIN15_SPEED_2MHZ 0x2000000000000000
- #define GPIO_PIN15_SPEED_50MHZ 0x3000000000000000
- #define GPIO_PIN15_SPEED_NONE 0x0000000000000000
- #define GPIO_PIN_ALL_SPEED_10MHZ 0x1111111111111111
- #define GPIO_PIN_ALL_SPEED_2MHZ 0x2222222222222222
- #define GPIO_PIN_ALL_SPEED_50MHZ 0x3333333333333333
- #define GPIO_PIN_ALL_SPEED_NONE 0x0000000000000000

Functions

- void GPIO_InitPin (const GPIO_t *gpio)

  Initialize a GPIO object: mode, speed, direction.
- void GPIO_WritePin (const GPIO_t *gpio, u64 state)

  Write multiple value on a GPIO object.
- u64 GPIO_ReadPin (const GPIO_t *gpio)

  Read multiple value from a GPIO object.

4.2.1 Detailed Description

This file is to be used as an implementation of the GPIO driver.

Author

MSN

Date

Mar 31, 2020

4.2.2 Macro Definition Documentation

4.2.2.1 GPIO_PIN0_MODE_INPUT_ANALOG

```
#define GPIO_PIN0_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 139 of file GPIO.h.

4.2.2.2 GPIO_PIN0_MODE_INPUT_FLOATING

```
#define GPIO_PIN0_MODE_INPUT_FLOATING 0x0000000000000004
```

Definition at line 140 of file GPIO.h.

4.2.2.3 GPIO_PIN0_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN0_MODE_INPUT_PULL_UP_DOWN 0x0000000000000008
```

Definition at line 141 of file GPIO.h.

4.2.2.4 GPIO_PIN0_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN0_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000000000000C
```

Definition at line 145 of file GPIO.h.

4.2.2.5 GPIO_PIN0_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN0_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000000008
```

Definition at line 144 of file GPIO.h.

4.2.2.6 GPIO_PIN0_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN0_MODE_OUTPUT_OPEN_DRAIN 0x0000000000000004
```

Definition at line 143 of file GPIO.h.

4.2.2.7 GPIO_PIN0_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN0_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 142 of file GPIO.h.

4.2.2.8 GPIO_PIN0_PORTA

```
#define GPIO_PIN0_PORTA 0x0000000000000001
```

Definition at line 71 of file GPIO.h.

4.2.2.9 GPIO_PIN0_PORTB

```
#define GPIO_PIN0_PORTB 0x0000000000000002
```

Definition at line 72 of file GPIO.h.

4.2.2.10 GPIO_PIN0_PORTC

```
#define GPIO_PIN0_PORTC 0x0000000000000003
```

Definition at line 73 of file GPIO.h.

4.2.2.11 GPIO_PIN0_SELECT

```
#define GPIO_PIN0_SELECT 0x000000000000000F
```

Definition at line 51 of file GPIO.h.

4.2.2.12 GPIO_PIN0_SPEED_10MHZ

```
#define GPIO_PIN0_SPEED_10MHZ 0x0000000000000001
```

Definition at line 275 of file GPIO.h.

4.2.2.13 GPIO_PIN0_SPEED_2MHZ

```
#define GPIO_PIN0_SPEED_2MHZ 0x0000000000000002
```

Definition at line 276 of file GPIO.h.

4.2.2.14 GPIO_PIN0_SPEED_50MHZ

```
#define GPIO_PIN0_SPEED_50MHZ 0x0000000000000003
```

Definition at line 277 of file GPIO.h.

4.2.2.15 GPIO_PIN0_SPEED_NONE

```
#define GPIO_PIN0_SPEED_NONE 0x0000000000000000
```

Definition at line 278 of file GPIO.h.

4.2.2.16 GPIO_PIN0_VALUE_HIGH

```
#define GPIO_PIN0_VALUE_HIGH 0x000000000000000F
```

Definition at line 14 of file GPIO.h.

4.2.2.17 GPIO_PIN0_VALUE_LOW

```
#define GPIO_PIN0_VALUE_LOW 0x0000000000000000
```

Definition at line 15 of file GPIO.h.

4.2.2.18 GPIO_PIN10_MODE_INPUT_ANALOG

```
#define GPIO_PIN10_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 219 of file GPIO.h.

4.2.2.19 GPIO_PIN10_MODE_INPUT_FLOATING

```
#define GPIO_PIN10_MODE_INPUT_FLOATING 0x0000040000000000
```

Definition at line 220 of file GPIO.h.

4.2.2.20 GPIO_PIN10_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN10_MODE_INPUT_PULL_UP_DOWN 0x0000080000000000
```

Definition at line 221 of file GPIO.h.

4.2.2.21 GPIO_PIN10_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN10_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000C0000000000
```

Definition at line 225 of file GPIO.h.

4.2.2.22 GPIO_PIN10_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN10_MODE_OUTPUT_AF_PUSH_PULL 0x0000080000000000
```

Definition at line 224 of file GPIO.h.

4.2.2.23 GPIO_PIN10_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN10_MODE_OUTPUT_OPEN_DRAIN 0x0000040000000000
```

Definition at line 223 of file GPIO.h.

4.2.2.24 GPIO_PIN10_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN10_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 222 of file GPIO.h.

4.2.2.25 GPIO_PIN10_PORTA

```
#define GPIO_PIN10_PORTA 0x0000010000000000
```

Definition at line 111 of file GPIO.h.

4.2.2.26 GPIO_PIN10_PORTB

```
#define GPIO_PIN10_PORTB 0x0000020000000000
```

Definition at line 112 of file GPIO.h.

4.2.2.27 GPIO_PIN10_PORTC

```
#define GPIO_PIN10_PORTC 0x0000030000000000
```

Definition at line 113 of file GPIO.h.

4.2.2.28 GPIO_PIN10_SELECT

```
#define GPIO_PIN10_SELECT 0x00000F0000000000
```

Definition at line 61 of file GPIO.h.

4.2.2.29 GPIO_PIN10_SPEED_10MHZ

```
#define GPIO_PIN10_SPEED_10MHZ 0x0000010000000000
```

Definition at line 325 of file GPIO.h.

4.2.2.30 GPIO_PIN10_SPEED_2MHZ

```
#define GPIO_PIN10_SPEED_2MHZ 0x0000020000000000
```

Definition at line 326 of file GPIO.h.

4.2.2.31 GPIO_PIN10_SPEED_50MHZ

```
#define GPIO_PIN10_SPEED_50MHZ 0x0000030000000000
```

Definition at line 327 of file GPIO.h.

4.2.2.32 GPIO_PIN10_SPEED_NONE

```
#define GPIO_PIN10_SPEED_NONE 0x0000000000000000
```

Definition at line 328 of file GPIO.h.

4.2.2.33 GPIO_PIN10_VALUE_HIGH

```
#define GPIO_PIN10_VALUE_HIGH 0x00000F0000000000
```

Definition at line 34 of file GPIO.h.

4.2.2.34 GPIO_PIN10_VALUE_LOW

```
#define GPIO_PIN10_VALUE_LOW 0x0000000000000000
```

Definition at line 35 of file GPIO.h.

4.2.2.35 GPIO_PIN11_MODE_INPUT_ANALOG

```
#define GPIO_PIN11_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 227 of file GPIO.h.

4.2.2.36 GPIO_PIN11_MODE_INPUT_FLOATING

```
#define GPIO_PIN11_MODE_INPUT_FLOATING 0x0000400000000000
```

Definition at line 228 of file GPIO.h.

4.2.2.37 GPIO_PIN11_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN11_MODE_INPUT_PULL_UP_DOWN 0x0000800000000000
```

Definition at line 229 of file GPIO.h.

4.2.2.38 GPIO_PIN11_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN11_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000C00000000000
```

Definition at line 233 of file GPIO.h.

4.2.2.39 GPIO_PIN11_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN11_MODE_OUTPUT_AF_PUSH_PULL 0x0000800000000000
```

Definition at line 232 of file GPIO.h.

4.2.2.40 GPIO_PIN11_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN11_MODE_OUTPUT_OPEN_DRAIN 0x0000400000000000
```

Definition at line 231 of file GPIO.h.

4.2.2.41 GPIO_PIN11_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN11_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 230 of file GPIO.h.

4.2.2.42 GPIO_PIN11_PORTA

```
#define GPIO_PIN11_PORTA 0x0000100000000000
```

Definition at line 115 of file GPIO.h.

4.2.2.43 GPIO_PIN11_PORTB

```
#define GPIO_PIN11_PORTB 0x0000200000000000
```

Definition at line 116 of file GPIO.h.

4.2.2.44 GPIO_PIN11_PORTC

```
#define GPIO_PIN11_PORTC 0x0000300000000000
```

Definition at line 117 of file GPIO.h.

4.2.2.45 GPIO_PIN11_SELECT

```
#define GPIO_PIN11_SELECT 0x0000F00000000000
```

Definition at line 62 of file GPIO.h.

4.2.2.46 GPIO_PIN11_SPEED_10MHZ

```
#define GPIO_PIN11_SPEED_10MHZ 0x0000100000000000
```

Definition at line 330 of file GPIO.h.

4.2.2.47 GPIO_PIN11_SPEED_2MHZ

```
#define GPIO_PIN11_SPEED_2MHZ 0x0000200000000000
```

Definition at line 331 of file GPIO.h.

4.2.2.48 GPIO_PIN11_SPEED_50MHZ

```
#define GPIO_PIN11_SPEED_50MHZ 0x0000300000000000
```

Definition at line 332 of file GPIO.h.

4.2.2.49 GPIO_PIN11_SPEED_NONE

```
#define GPIO_PIN11_SPEED_NONE 0x0000000000000000
```

Definition at line 333 of file GPIO.h.

4.2.2.50 GPIO_PIN11_VALUE_HIGH

```
#define GPIO_PIN11_VALUE_HIGH 0x0000F00000000000
```

Definition at line 36 of file GPIO.h.

4.2.2.51 GPIO_PIN11_VALUE_LOW

```
#define GPIO_PIN11_VALUE_LOW 0x0000000000000000
```

Definition at line 37 of file GPIO.h.

4.2.2.52 GPIO_PIN12_MODE_INPUT_ANALOG

```
#define GPIO_PIN12_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 235 of file GPIO.h.

4.2.2.53 GPIO_PIN12_MODE_INPUT_FLOATING

```
#define GPIO_PIN12_MODE_INPUT_FLOATING 0x0004000000000000
```

Definition at line 236 of file GPIO.h.

4.2.2.54 GPIO_PIN12_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN12_MODE_INPUT_PULL_UP_DOWN 0x0008000000000000
```

Definition at line 237 of file GPIO.h.

4.2.2.55 GPIO_PIN12_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN12_MODE_OUTPUT_AF_OPEN_DRAIN 0x000C000000000000
```

Definition at line 241 of file GPIO.h.

4.2.2.56 GPIO_PIN12_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN12_MODE_OUTPUT_AF_PUSH_PULL 0x0008000000000000
```

Definition at line 240 of file GPIO.h.

4.2.2.57 GPIO_PIN12_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN12_MODE_OUTPUT_OPEN_DRAIN 0x0004000000000000
```

Definition at line 239 of file GPIO.h.

4.2.2.58 GPIO_PIN12_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN12_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 238 of file GPIO.h.

4.2.2.59 GPIO_PIN12_PORTA

```
#define GPIO_PIN12_PORTA 0x0001000000000000
```

Definition at line 119 of file GPIO.h.

4.2.2.60 GPIO_PIN12_PORTB

```
#define GPIO_PIN12_PORTB 0x0002000000000000
```

Definition at line 120 of file GPIO.h.

4.2.2.61 GPIO_PIN12_PORTC

```
#define GPIO_PIN12_PORTC 0x0003000000000000
```

Definition at line 121 of file GPIO.h.

4.2.2.62 GPIO_PIN12_SELECT

```
#define GPIO_PIN12_SELECT 0x000F000000000000
```

Definition at line 63 of file GPIO.h.

4.2.2.63 GPIO_PIN12_SPEED_10MHZ

```
#define GPIO_PIN12_SPEED_10MHZ 0x0001000000000000
```

Definition at line 335 of file GPIO.h.

4.2.2.64 GPIO_PIN12_SPEED_2MHZ

```
#define GPIO_PIN12_SPEED_2MHZ 0x0002000000000000
```

Definition at line 336 of file GPIO.h.

4.2.2.65 GPIO_PIN12_SPEED_50MHZ

```
#define GPIO_PIN12_SPEED_50MHZ 0x0003000000000000
```

Definition at line 337 of file GPIO.h.

4.2.2.66 GPIO_PIN12_SPEED_NONE

```
#define GPIO_PIN12_SPEED_NONE 0x0000000000000000
```

Definition at line 338 of file GPIO.h.

4.2.2.67 GPIO_PIN12_VALUE_HIGH

```
#define GPIO_PIN12_VALUE_HIGH 0x000F000000000000
```

Definition at line 38 of file GPIO.h.

4.2.2.68 GPIO_PIN12_VALUE_LOW

```
#define GPIO_PIN12_VALUE_LOW 0x0000000000000000
```

Definition at line 39 of file GPIO.h.

4.2.2.69 GPIO_PIN13_MODE_INPUT_ANALOG

```
#define GPIO_PIN13_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 243 of file GPIO.h.

4.2.2.70 GPIO_PIN13_MODE_INPUT_FLOATING

```
#define GPIO_PIN13_MODE_INPUT_FLOATING 0x0040000000000000
```

Definition at line 244 of file GPIO.h.

4.2.2.71 GPIO_PIN13_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN13_MODE_INPUT_PULL_UP_DOWN 0x0080000000000000
```

Definition at line 245 of file GPIO.h.

4.2.2.72 GPIO_PIN13_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN13_MODE_OUTPUT_AF_OPEN_DRAIN 0x00C0000000000000
```

Definition at line 249 of file GPIO.h.

4.2.2.73 GPIO_PIN13_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN13_MODE_OUTPUT_AF_PUSH_PULL 0x0080000000000000
```

Definition at line 248 of file GPIO.h.

4.2.2.74 GPIO_PIN13_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN13_MODE_OUTPUT_OPEN_DRAIN 0x0040000000000000
```

Definition at line 247 of file GPIO.h.

4.2.2.75 GPIO_PIN13_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN13_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 246 of file GPIO.h.

4.2.2.76 GPIO_PIN13_PORTA

```
#define GPIO_PIN13_PORTA 0x0010000000000000
```

Definition at line 123 of file GPIO.h.

4.2.2.77 GPIO_PIN13_PORTB

```
#define GPIO_PIN13_PORTB 0x0020000000000000
```

Definition at line 124 of file GPIO.h.

4.2.2.78 GPIO_PIN13_PORTC

```
#define GPIO_PIN13_PORTC 0x0030000000000000
```

Definition at line 125 of file GPIO.h.

4.2.2.79 GPIO_PIN13_SELECT

```
#define GPIO_PIN13_SELECT 0x00F0000000000000
```

Definition at line 64 of file GPIO.h.

4.2.2.80 GPIO_PIN13_SPEED_10MHZ

```
#define GPIO_PIN13_SPEED_10MHZ 0x0010000000000000
```

Definition at line 340 of file GPIO.h.

4.2.2.81 GPIO_PIN13_SPEED_2MHZ

```
#define GPIO_PIN13_SPEED_2MHZ 0x0020000000000000
```

Definition at line 341 of file GPIO.h.

4.2.2.82 GPIO_PIN13_SPEED_50MHZ

```
#define GPIO_PIN13_SPEED_50MHZ 0x0030000000000000
```

Definition at line 342 of file GPIO.h.

4.2.2.83 GPIO_PIN13_SPEED_NONE

```
#define GPIO_PIN13_SPEED_NONE 0x0000000000000000
```

Definition at line 343 of file GPIO.h.

4.2.2.84 GPIO_PIN13_VALUE_HIGH

```
#define GPIO_PIN13_VALUE_HIGH 0x00F0000000000000
```

Definition at line 40 of file GPIO.h.

4.2.2.85 GPIO_PIN13_VALUE_LOW

```
#define GPIO_PIN13_VALUE_LOW 0x0000000000000000
```

Definition at line 41 of file GPIO.h.

4.2.2.86 GPIO_PIN14_MODE_INPUT_ANALOG

```
#define GPIO_PIN14_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 251 of file GPIO.h.

4.2.2.87 GPIO_PIN14_MODE_INPUT_FLOATING

```
#define GPIO_PIN14_MODE_INPUT_FLOATING 0x0400000000000000
```

Definition at line 252 of file GPIO.h.

4.2.2.88 GPIO_PIN14_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN14_MODE_INPUT_PULL_UP_DOWN 0x0800000000000000
```

Definition at line 253 of file GPIO.h.

4.2.2.89 GPIO_PIN14_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN14_MODE_OUTPUT_AF_OPEN_DRAIN 0x0C000000000000000
```

Definition at line 257 of file GPIO.h.

4.2.2.90 GPIO_PIN14_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN14_MODE_OUTPUT_AF_PUSH_PULL 0x0800000000000000
```

Definition at line 256 of file GPIO.h.

4.2.2.91 GPIO_PIN14_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN14_MODE_OUTPUT_OPEN_DRAIN 0x0400000000000000
```

Definition at line 255 of file GPIO.h.

4.2.2.92 GPIO_PIN14_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN14_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 254 of file GPIO.h.

4.2.2.93 GPIO_PIN14_PORTA

```
#define GPIO_PIN14_PORTA 0x0100000000000000
```

Definition at line 127 of file GPIO.h.

4.2.2.94 GPIO_PIN14_PORTB

```
#define GPIO_PIN14_PORTB 0x0200000000000000
```

Definition at line 128 of file GPIO.h.

4.2.2.95 GPIO_PIN14_PORTC

```
#define GPIO_PIN14_PORTC 0x0300000000000000
```

Definition at line 129 of file GPIO.h.

4.2.2.96 GPIO_PIN14_SELECT

```
#define GPIO_PIN14_SELECT 0x0F00000000000000
```

Definition at line 65 of file GPIO.h.

4.2.2.97 GPIO_PIN14_SPEED_10MHZ

```
#define GPIO_PIN14_SPEED_10MHZ 0x0100000000000000
```

Definition at line 345 of file GPIO.h.

4.2.2.98 GPIO_PIN14_SPEED_2MHZ

```
#define GPIO_PIN14_SPEED_2MHZ 0x0200000000000000
```

Definition at line 346 of file GPIO.h.

4.2.2.99 GPIO_PIN14_SPEED_50MHZ

```
#define GPIO_PIN14_SPEED_50MHZ 0x0300000000000000
```

Definition at line 347 of file GPIO.h.

4.2.2.100 GPIO_PIN14_SPEED_NONE

```
#define GPIO_PIN14_SPEED_NONE 0x0000000000000000
```

Definition at line 348 of file GPIO.h.

4.2.2.101 GPIO_PIN14_VALUE_HIGH

```
#define GPIO_PIN14_VALUE_HIGH 0x0F00000000000000
```

Definition at line 42 of file GPIO.h.

4.2.2.102 GPIO_PIN14_VALUE_LOW

```
#define GPIO_PIN14_VALUE_LOW 0x0000000000000000
```

Definition at line 43 of file GPIO.h.

4.2.2.103 GPIO_PIN15_MODE_INPUT_ANALOG

```
#define GPIO_PIN15_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 259 of file GPIO.h.

4.2.2.104 GPIO_PIN15_MODE_INPUT_FLOATING

```
#define GPIO_PIN15_MODE_INPUT_FLOATING 0x4000000000000000
```

Definition at line 260 of file GPIO.h.

4.2.2.105 GPIO_PIN15_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN15_MODE_INPUT_PULL_UP_DOWN 0x8000000000000000
```

Definition at line 261 of file GPIO.h.

4.2.2.106 GPIO_PIN15_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN15_MODE_OUTPUT_AF_OPEN_DRAIN 0xC000000000000000
```

Definition at line 265 of file GPIO.h.

4.2.2.107 GPIO_PIN15_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN15_MODE_OUTPUT_AF_PUSH_PULL 0x8000000000000000
```

Definition at line 264 of file GPIO.h.

4.2.2.108 GPIO_PIN15_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN15_MODE_OUTPUT_OPEN_DRAIN 0x4000000000000000
```

Definition at line 263 of file GPIO.h.

4.2.2.109 GPIO_PIN15_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN15_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 262 of file GPIO.h.

4.2.2.110 GPIO_PIN15_PORTA

```
#define GPIO_PIN15_PORTA 0x1000000000000000
```

Definition at line 131 of file GPIO.h.

4.2.2.111 GPIO_PIN15_PORTB

```
#define GPIO_PIN15_PORTB 0x2000000000000000
```

Definition at line 132 of file GPIO.h.

4.2.2.112 GPIO_PIN15_PORTC

```
#define GPIO_PIN15_PORTC 0x3000000000000000
```

Definition at line 133 of file GPIO.h.

4.2.2.113 GPIO_PIN15_SELECT

```
#define GPIO_PIN15_SELECT 0xF000000000000000
```

Definition at line 66 of file GPIO.h.

4.2.2.114 GPIO_PIN15_SPEED_10MHZ

```
#define GPIO_PIN15_SPEED_10MHZ 0x1000000000000000
```

Definition at line 350 of file GPIO.h.

4.2.2.115 GPIO_PIN15_SPEED_2MHZ

```
#define GPIO_PIN15_SPEED_2MHZ 0x2000000000000000
```

Definition at line 351 of file GPIO.h.

4.2.2.116 GPIO_PIN15_SPEED_50MHZ

```
#define GPIO_PIN15_SPEED_50MHZ 0x3000000000000000
```

Definition at line 352 of file GPIO.h.

4.2.2.117 GPIO_PIN15_SPEED_NONE

```
#define GPIO_PIN15_SPEED_NONE 0x0000000000000000
```

Definition at line 353 of file GPIO.h.

4.2.2.118 GPIO_PIN15_VALUE_HIGH

```
#define GPIO_PIN15_VALUE_HIGH 0xF000000000000000
```

Definition at line 44 of file GPIO.h.

4.2.2.119 GPIO_PIN15_VALUE_LOW

```
#define GPIO_PIN15_VALUE_LOW 0x0000000000000000
```

Definition at line 45 of file GPIO.h.

4.2.2.120 GPIO_PIN1_MODE_INPUT_ANALOG

```
#define GPIO_PIN1_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 147 of file GPIO.h.

4.2.2.121 GPIO_PIN1_MODE_INPUT_FLOATING

```
#define GPIO_PIN1_MODE_INPUT_FLOATING 0x0000000000000040
```

Definition at line 148 of file GPIO.h.

4.2.2.122 GPIO_PIN1_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN1_MODE_INPUT_PULL_UP_DOWN 0x0000000000000080
```

Definition at line 149 of file GPIO.h.

4.2.2.123 GPIO_PIN1_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN1_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000000000000C0
```

Definition at line 153 of file GPIO.h.

4.2.2.124 GPIO_PIN1_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN1_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000000080
```

Definition at line 152 of file GPIO.h.

4.2.2.125 GPIO_PIN1_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN1_MODE_OUTPUT_OPEN_DRAIN 0x0000000000000040
```

Definition at line 151 of file GPIO.h.

4.2.2.126 GPIO_PIN1_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN1_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 150 of file GPIO.h.

4.2.2.127 GPIO_PIN1_PORTA

```
#define GPIO_PIN1_PORTA 0x0000000000000010
```

Definition at line 75 of file GPIO.h.

4.2.2.128 GPIO_PIN1_PORTB

```
#define GPIO_PIN1_PORTB 0x0000000000000020
```

Definition at line 76 of file GPIO.h.

4.2.2.129 GPIO_PIN1_PORTC

```
#define GPIO_PIN1_PORTC 0x0000000000000030
```

Definition at line 77 of file GPIO.h.

4.2.2.130 GPIO_PIN1_SELECT

```
#define GPIO_PIN1_SELECT 0x00000000000000F0
```

Definition at line 52 of file GPIO.h.

4.2.2.131 GPIO_PIN1_SPEED_10MHZ

```
#define GPIO_PIN1_SPEED_10MHZ 0x0000000000000010
```

Definition at line 280 of file GPIO.h.

4.2.2.132 GPIO_PIN1_SPEED_2MHZ

```
#define GPIO_PIN1_SPEED_2MHZ 0x0000000000000020
```

Definition at line 281 of file GPIO.h.

4.2.2.133 GPIO_PIN1_SPEED_50MHZ

```
#define GPIO_PIN1_SPEED_50MHZ 0x0000000000000030
```

Definition at line 282 of file GPIO.h.

4.2.2.134 GPIO_PIN1_SPEED_NONE

```
#define GPIO_PIN1_SPEED_NONE 0x0000000000000000
```

Definition at line 283 of file GPIO.h.

4.2.2.135 GPIO_PIN1_VALUE_HIGH

```
#define GPIO_PIN1_VALUE_HIGH 0x00000000000000F0
```

Definition at line 16 of file GPIO.h.

4.2.2.136 GPIO_PIN1_VALUE_LOW

```
#define GPIO_PIN1_VALUE_LOW 0x0000000000000000
```

Definition at line 17 of file GPIO.h.

4.2.2.137 GPIO_PIN2_MODE_INPUT_ANALOG

```
#define GPIO_PIN2_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 155 of file GPIO.h.

4.2.2.138 GPIO_PIN2_MODE_INPUT_FLOATING

```
#define GPIO_PIN2_MODE_INPUT_FLOATING 0x0000000000000400
```

Definition at line 156 of file GPIO.h.

4.2.2.139 GPIO_PIN2_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN2_MODE_INPUT_PULL_UP_DOWN 0x0000000000000800
```

Definition at line 157 of file GPIO.h.

4.2.2.140 GPIO_PIN2_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN2_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000000000000C00
```

Definition at line 161 of file GPIO.h.

4.2.2.141 GPIO_PIN2_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN2_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000000800
```

Definition at line 160 of file GPIO.h.

4.2.2.142 GPIO_PIN2_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN2_MODE_OUTPUT_OPEN_DRAIN 0x0000000000000400
```

Definition at line 159 of file GPIO.h.

4.2.2.143 GPIO_PIN2_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN2_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 158 of file GPIO.h.

4.2.2.144 GPIO_PIN2_PORTA

```
#define GPIO_PIN2_PORTA 0x0000000000000100
```

Definition at line 79 of file GPIO.h.

4.2.2.145 GPIO_PIN2_PORTB

```
#define GPIO_PIN2_PORTB 0x0000000000000200
```

Definition at line 80 of file GPIO.h.

4.2.2.146 GPIO_PIN2_PORTC

```
#define GPIO_PIN2_PORTC 0x0000000000000300
```

Definition at line 81 of file GPIO.h.

4.2.2.147 GPIO_PIN2_SELECT

```
#define GPIO_PIN2_SELECT 0x0000000000000F00
```

Definition at line 53 of file GPIO.h.

4.2.2.148 GPIO_PIN2_SPEED_10MHZ

```
#define GPIO_PIN2_SPEED_10MHZ 0x0000000000000100
```

Definition at line 285 of file GPIO.h.

4.2.2.149 GPIO_PIN2_SPEED_2MHZ

```
#define GPIO_PIN2_SPEED_2MHZ 0x0000000000000200
```

Definition at line 286 of file GPIO.h.

4.2.2.150 GPIO_PIN2_SPEED_50MHZ

```
#define GPIO_PIN2_SPEED_50MHZ 0x0000000000000300
```

Definition at line 287 of file GPIO.h.

4.2.2.151 GPIO_PIN2_SPEED_NONE

```
#define GPIO_PIN2_SPEED_NONE 0x0000000000000000
```

Definition at line 288 of file GPIO.h.

4.2.2.152 GPIO_PIN2_VALUE_HIGH

```
#define GPIO_PIN2_VALUE_HIGH 0x0000000000000F00
```

Definition at line 18 of file GPIO.h.

4.2.2.153 GPIO_PIN2_VALUE_LOW

```
#define GPIO_PIN2_VALUE_LOW 0x0000000000000000
```

Definition at line 19 of file GPIO.h.

4.2.2.154 GPIO_PIN3_MODE_INPUT_ANALOG

```
#define GPIO_PIN3_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 163 of file GPIO.h.

4.2.2.155 GPIO_PIN3_MODE_INPUT_FLOATING

```
#define GPIO_PIN3_MODE_INPUT_FLOATING 0x0000000000004000
```

Definition at line 164 of file GPIO.h.

4.2.2.156 GPIO_PIN3_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN3_MODE_INPUT_PULL_UP_DOWN 0x0000000000008000
```

Definition at line 165 of file GPIO.h.

4.2.2.157 GPIO_PIN3_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN3_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000000000C000
```

Definition at line 169 of file GPIO.h.

4.2.2.158 GPIO_PIN3_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN3_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000008000
```

Definition at line 168 of file GPIO.h.

4.2.2.159 GPIO_PIN3_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN3_MODE_OUTPUT_OPEN_DRAIN 0x0000000000004000
```

Definition at line 167 of file GPIO.h.

4.2.2.160 GPIO_PIN3_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN3_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 166 of file GPIO.h.

4.2.2.161 GPIO_PIN3_PORTA

```
#define GPIO_PIN3_PORTA 0x0000000000001000
```

Definition at line 83 of file GPIO.h.

4.2.2.162 GPIO_PIN3_PORTB

```
#define GPIO_PIN3_PORTB 0x0000000000002000
```

Definition at line 84 of file GPIO.h.

4.2.2.163 GPIO_PIN3_PORTC

```
#define GPIO_PIN3_PORTC 0x0000000000003000
```

Definition at line 85 of file GPIO.h.

4.2.2.164 GPIO_PIN3_SELECT

```
#define GPIO_PIN3_SELECT 0x000000000000F000
```

Definition at line 54 of file GPIO.h.

4.2.2.165 GPIO_PIN3_SPEED_10MHZ

```
#define GPIO_PIN3_SPEED_10MHZ 0x0000000000001000
```

Definition at line 290 of file GPIO.h.

4.2.2.166 GPIO_PIN3_SPEED_2MHZ

```
#define GPIO_PIN3_SPEED_2MHZ 0x0000000000002000
```

Definition at line 291 of file GPIO.h.

4.2.2.167 GPIO_PIN3_SPEED_50MHZ

```
#define GPIO_PIN3_SPEED_50MHZ 0x0000000000003000
```

Definition at line 292 of file GPIO.h.

4.2.2.168 GPIO_PIN3_SPEED_NONE

```
#define GPIO_PIN3_SPEED_NONE 0x0000000000000000
```

Definition at line 293 of file GPIO.h.

4.2.2.169 GPIO_PIN3_VALUE_HIGH

```
#define GPIO_PIN3_VALUE_HIGH 0x000000000000F000
```

Definition at line 20 of file GPIO.h.

4.2.2.170 GPIO_PIN3_VALUE_LOW

```
#define GPIO_PIN3_VALUE_LOW 0x0000000000000000
```

Definition at line 21 of file GPIO.h.

4.2.2.171 GPIO_PIN4_MODE_INPUT_ANALOG

```
#define GPIO_PIN4_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 171 of file GPIO.h.

4.2.2.172 GPIO_PIN4_MODE_INPUT_FLOATING

```
#define GPIO_PIN4_MODE_INPUT_FLOATING 0x0000000000040000
```

Definition at line 172 of file GPIO.h.

4.2.2.173 GPIO_PIN4_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN4_MODE_INPUT_PULL_UP_DOWN 0x0000000000080000
```

Definition at line 173 of file GPIO.h.

4.2.2.174 GPIO_PIN4_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN4_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000000000C0000
```

Definition at line 177 of file GPIO.h.

4.2.2.175 GPIO_PIN4_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN4_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000080000
```

Definition at line 176 of file GPIO.h.

4.2.2.176 GPIO_PIN4_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN4_MODE_OUTPUT_OPEN_DRAIN 0x0000000000040000
```

Definition at line 175 of file GPIO.h.

4.2.2.177 GPIO_PIN4_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN4_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 174 of file GPIO.h.

4.2.2.178 GPIO_PIN4_PORTA

```
#define GPIO_PIN4_PORTA 0x0000000000010000
```

Definition at line 87 of file GPIO.h.

4.2.2.179 GPIO_PIN4_PORTB

```
#define GPIO_PIN4_PORTB 0x0000000000020000
```

Definition at line 88 of file GPIO.h.

4.2.2.180 GPIO_PIN4_PORTC

```
#define GPIO_PIN4_PORTC 0x0000000000030000
```

Definition at line 89 of file GPIO.h.

4.2.2.181 GPIO_PIN4_SELECT

```
#define GPIO_PIN4_SELECT 0x00000000000F0000
```

Definition at line 55 of file GPIO.h.

4.2.2.182 GPIO_PIN4_SPEED_10MHZ

```
#define GPIO_PIN4_SPEED_10MHZ 0x0000000000010000
```

Definition at line 295 of file GPIO.h.

4.2.2.183 GPIO_PIN4_SPEED_2MHZ

```
#define GPIO_PIN4_SPEED_2MHZ 0x0000000000020000
```

Definition at line 296 of file GPIO.h.

4.2.2.184 GPIO_PIN4_SPEED_50MHZ

```
#define GPIO_PIN4_SPEED_50MHZ 0x0000000000030000
```

Definition at line 297 of file GPIO.h.

4.2.2.185 GPIO_PIN4_SPEED_NONE

```
#define GPIO_PIN4_SPEED_NONE 0x0000000000000000
```

Definition at line 298 of file GPIO.h.

4.2.2.186 GPIO_PIN4_VALUE_HIGH

```
#define GPIO_PIN4_VALUE_HIGH 0x00000000000F0000
```

Definition at line 22 of file GPIO.h.

4.2.2.187 GPIO_PIN4_VALUE_LOW

```
#define GPIO_PIN4_VALUE_LOW 0x0000000000000000
```

Definition at line 23 of file GPIO.h.

4.2.2.188 GPIO_PIN5_MODE_INPUT_ANALOG

```
#define GPIO_PIN5_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 179 of file GPIO.h.

4.2.2.189 GPIO_PIN5_MODE_INPUT_FLOATING

```
#define GPIO_PIN5_MODE_INPUT_FLOATING 0x0000000000400000
```

Definition at line 180 of file GPIO.h.

4.2.2.190 GPIO_PIN5_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN5_MODE_INPUT_PULL_UP_DOWN 0x0000000000800000
```

Definition at line 181 of file GPIO.h.

4.2.2.191 GPIO_PIN5_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN5_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000000000C00000
```

Definition at line 185 of file GPIO.h.

4.2.2.192 GPIO_PIN5_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN5_MODE_OUTPUT_AF_PUSH_PULL 0x0000000000800000
```

Definition at line 184 of file GPIO.h.

4.2.2.193 GPIO_PIN5_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN5_MODE_OUTPUT_OPEN_DRAIN 0x0000000000400000
```

Definition at line 183 of file GPIO.h.

4.2.2.194 GPIO_PIN5_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN5_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 182 of file GPIO.h.

4.2.2.195 GPIO_PIN5_PORTA

```
#define GPIO_PIN5_PORTA 0x0000000000100000
```

Definition at line 91 of file GPIO.h.

4.2.2.196 GPIO_PIN5_PORTB

```
#define GPIO_PIN5_PORTB 0x0000000000200000
```

Definition at line 92 of file GPIO.h.

4.2.2.197 GPIO_PIN5_PORTC

```
#define GPIO_PIN5_PORTC 0x0000000000300000
```

Definition at line 93 of file GPIO.h.

4.2.2.198 GPIO_PIN5_SELECT

```
#define GPIO_PIN5_SELECT 0x0000000000F00000
```

Definition at line 56 of file GPIO.h.

4.2.2.199 GPIO_PIN5_SPEED_10MHZ

```
#define GPIO_PIN5_SPEED_10MHZ 0x0000000000100000
```

Definition at line 300 of file GPIO.h.

4.2.2.200 GPIO_PIN5_SPEED_2MHZ

```
#define GPIO_PIN5_SPEED_2MHZ 0x0000000000200000
```

Definition at line 301 of file GPIO.h.

4.2.2.201 GPIO_PIN5_SPEED_50MHZ

```
#define GPIO_PIN5_SPEED_50MHZ 0x0000000000300000
```

Definition at line 302 of file GPIO.h.

4.2.2.202 GPIO_PIN5_SPEED_NONE

```
#define GPIO_PIN5_SPEED_NONE 0x0000000000000000
```

Definition at line 303 of file GPIO.h.

4.2.2.203 GPIO_PIN5_VALUE_HIGH

```
#define GPIO_PIN5_VALUE_HIGH 0x0000000000F00000
```

Definition at line 24 of file GPIO.h.

4.2.2.204 GPIO_PIN5_VALUE_LOW

```
#define GPIO_PIN5_VALUE_LOW 0x0000000000000000
```

Definition at line 25 of file GPIO.h.

4.2.2.205 GPIO_PIN6_MODE_INPUT_ANALOG

```
#define GPIO_PIN6_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 187 of file GPIO.h.

4.2.2.206 GPIO_PIN6_MODE_INPUT_FLOATING

```
#define GPIO_PIN6_MODE_INPUT_FLOATING 0x0000000004000000
```

Definition at line 188 of file GPIO.h.

4.2.2.207 GPIO_PIN6_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN6_MODE_INPUT_PULL_UP_DOWN 0x0000000008000000
```

Definition at line 189 of file GPIO.h.

4.2.2.208 GPIO_PIN6_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN6_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000000C000000
```

Definition at line 193 of file GPIO.h.

4.2.2.209 GPIO_PIN6_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN6_MODE_OUTPUT_AF_PUSH_PULL 0x0000000008000000
```

Definition at line 192 of file GPIO.h.

4.2.2.210 GPIO_PIN6_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN6_MODE_OUTPUT_OPEN_DRAIN 0x0000000004000000
```

Definition at line 191 of file GPIO.h.

4.2.2.211 GPIO_PIN6_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN6_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 190 of file GPIO.h.

4.2.2.212 GPIO_PIN6_PORTA

```
#define GPIO_PIN6_PORTA 0x0000000001000000
```

Definition at line 95 of file GPIO.h.

4.2.2.213 GPIO_PIN6_PORTB

```
#define GPIO_PIN6_PORTB 0x0000000002000000
```

Definition at line 96 of file GPIO.h.

4.2.2.214 GPIO_PIN6_PORTC

```
#define GPIO_PIN6_PORTC 0x0000000003000000
```

Definition at line 97 of file GPIO.h.

4.2.2.215 GPIO_PIN6_SELECT

```
#define GPIO_PIN6_SELECT 0x000000000F000000
```

Definition at line 57 of file GPIO.h.

4.2.2.216 GPIO_PIN6_SPEED_10MHZ

```
#define GPIO_PIN6_SPEED_10MHZ 0x0000000001000000
```

Definition at line 305 of file GPIO.h.

4.2.2.217 GPIO_PIN6_SPEED_2MHZ

```
#define GPIO_PIN6_SPEED_2MHZ 0x0000000002000000
```

Definition at line 306 of file GPIO.h.

4.2.2.218 GPIO_PIN6_SPEED_50MHZ

```
#define GPIO_PIN6_SPEED_50MHZ 0x0000000003000000
```

Definition at line 307 of file GPIO.h.

4.2.2.219 GPIO_PIN6_SPEED_NONE

```
#define GPIO_PIN6_SPEED_NONE 0x0000000000000000
```

Definition at line 308 of file GPIO.h.

4.2.2.220 GPIO_PIN6_VALUE_HIGH

```
#define GPIO_PIN6_VALUE_HIGH 0x000000000F000000
```

Definition at line 26 of file GPIO.h.

4.2.2.221 GPIO_PIN6_VALUE_LOW

```
#define GPIO_PIN6_VALUE_LOW 0x0000000000000000
```

Definition at line 27 of file GPIO.h.

4.2.2.222 GPIO_PIN7_MODE_INPUT_ANALOG

```
#define GPIO_PIN7_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 195 of file GPIO.h.

4.2.2.223 GPIO_PIN7_MODE_INPUT_FLOATING

```
#define GPIO_PIN7_MODE_INPUT_FLOATING 0x0000000040000000
```

Definition at line 196 of file GPIO.h.

4.2.2.224 GPIO_PIN7_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN7_MODE_INPUT_PULL_UP_DOWN 0x0000000080000000
```

Definition at line 197 of file GPIO.h.

4.2.2.225 GPIO_PIN7_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN7_MODE_OUTPUT_AF_OPEN_DRAIN 0x00000000C0000000
```

Definition at line 201 of file GPIO.h.

4.2.2.226 GPIO_PIN7_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN7_MODE_OUTPUT_AF_PUSH_PULL 0x0000000080000000
```

Definition at line 200 of file GPIO.h.

4.2.2.227 GPIO_PIN7_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN7_MODE_OUTPUT_OPEN_DRAIN 0x0000000040000000
```

Definition at line 199 of file GPIO.h.

4.2.2.228 GPIO_PIN7_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN7_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 198 of file GPIO.h.

4.2.2.229 GPIO_PIN7_PORTA

```
#define GPIO_PIN7_PORTA 0x0000000010000000
```

Definition at line 99 of file GPIO.h.

4.2.2.230 GPIO_PIN7_PORTB

```
#define GPIO_PIN7_PORTB 0x0000000020000000
```

Definition at line 100 of file GPIO.h.

4.2.2.231 GPIO_PIN7_PORTC

```
#define GPIO_PIN7_PORTC 0x0000000030000000
```

Definition at line 101 of file GPIO.h.

4.2.2.232 GPIO_PIN7_SELECT

```
#define GPIO_PIN7_SELECT 0x00000000F0000000
```

Definition at line 58 of file GPIO.h.

4.2.2.233 GPIO_PIN7_SPEED_10MHZ

```
#define GPIO_PIN7_SPEED_10MHZ 0x0000000010000000
```

Definition at line 310 of file GPIO.h.

4.2.2.234 GPIO_PIN7_SPEED_2MHZ

```
#define GPIO_PIN7_SPEED_2MHZ 0x0000000020000000
```

Definition at line 311 of file GPIO.h.

4.2.2.235 GPIO_PIN7_SPEED_50MHZ

```
#define GPIO_PIN7_SPEED_50MHZ 0x0000000030000000
```

Definition at line 312 of file GPIO.h.

4.2.2.236 GPIO_PIN7_SPEED_NONE

```
#define GPIO_PIN7_SPEED_NONE 0x0000000000000000
```

Definition at line 313 of file GPIO.h.

4.2.2.237 GPIO_PIN7_VALUE_HIGH

```
#define GPIO_PIN7_VALUE_HIGH 0x00000000F0000000
```

Definition at line 28 of file GPIO.h.

4.2.2.238 GPIO_PIN7_VALUE_LOW

```
#define GPIO_PIN7_VALUE_LOW 0x0000000000000000
```

Definition at line 29 of file GPIO.h.

4.2.2.239 GPIO_PIN8_MODE_INPUT_ANALOG

```
#define GPIO_PIN8_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 203 of file GPIO.h.

4.2.2.240 GPIO_PIN8_MODE_INPUT_FLOATING

```
#define GPIO_PIN8_MODE_INPUT_FLOATING 0x0000000400000000
```

Definition at line 204 of file GPIO.h.

4.2.2.241 GPIO_PIN8_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN8_MODE_INPUT_PULL_UP_DOWN 0x0000000800000000
```

Definition at line 205 of file GPIO.h.

4.2.2.242 GPIO_PIN8_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN8_MODE_OUTPUT_AF_OPEN_DRAIN 0x0000000C00000000
```

Definition at line 209 of file GPIO.h.

4.2.2.243 GPIO_PIN8_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN8_MODE_OUTPUT_AF_PUSH_PULL 0x0000000800000000
```

Definition at line 208 of file GPIO.h.

4.2.2.244 GPIO_PIN8_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN8_MODE_OUTPUT_OPEN_DRAIN 0x0000000400000000
```

Definition at line 207 of file GPIO.h.

4.2.2.245 GPIO_PIN8_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN8_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 206 of file GPIO.h.

4.2.2.246 GPIO_PIN8_PORTA

```
#define GPIO_PIN8_PORTA 0x0000000100000000
```

Definition at line 103 of file GPIO.h.

4.2.2.247 GPIO_PIN8_PORTB

```
#define GPIO_PIN8_PORTB 0x0000000200000000
```

Definition at line 104 of file GPIO.h.

4.2.2.248 GPIO_PIN8_PORTC

```
#define GPIO_PIN8_PORTC 0x0000000300000000
```

Definition at line 105 of file GPIO.h.

4.2.2.249 GPIO_PIN8_SELECT

```
#define GPIO_PIN8_SELECT 0x0000000F00000000
```

Definition at line 59 of file GPIO.h.

4.2.2.250 GPIO_PIN8_SPEED_10MHZ

```
#define GPIO_PIN8_SPEED_10MHZ 0x0000000100000000
```

Definition at line 315 of file GPIO.h.

4.2.2.251 GPIO_PIN8_SPEED_2MHZ

```
#define GPIO_PIN8_SPEED_2MHZ 0x0000000200000000
```

Definition at line 316 of file GPIO.h.

4.2.2.252 GPIO_PIN8_SPEED_50MHZ

```
#define GPIO_PIN8_SPEED_50MHZ 0x0000000300000000
```

Definition at line 317 of file GPIO.h.

4.2.2.253 GPIO_PIN8_SPEED_NONE

```
#define GPIO_PIN8_SPEED_NONE 0x0000000000000000
```

Definition at line 318 of file GPIO.h.

4.2.2.254 GPIO_PIN8_VALUE_HIGH

```
#define GPIO_PIN8_VALUE_HIGH 0x0000000F00000000
```

Definition at line 30 of file GPIO.h.

4.2.2.255 GPIO_PIN8_VALUE_LOW

```
#define GPIO_PIN8_VALUE_LOW 0x0000000000000000
```

Definition at line 31 of file GPIO.h.

4.2.2.256 GPIO_PIN9_MODE_INPUT_ANALOG

```
#define GPIO_PIN9_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 211 of file GPIO.h.

4.2.2.257 GPIO_PIN9_MODE_INPUT_FLOATING

```
#define GPIO_PIN9_MODE_INPUT_FLOATING 0x0000004000000000
```

Definition at line 212 of file GPIO.h.

4.2.2.258 GPIO_PIN9_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN9_MODE_INPUT_PULL_UP_DOWN 0x0000008000000000
```

Definition at line 213 of file GPIO.h.

4.2.2.259 GPIO_PIN9_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN9_MODE_OUTPUT_AF_OPEN_DRAIN 0x000000C000000000
```

Definition at line 217 of file GPIO.h.

4.2.2.260 GPIO_PIN9_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN9_MODE_OUTPUT_AF_PUSH_PULL 0x0000008000000000
```

Definition at line 216 of file GPIO.h.

4.2.2.261 GPIO_PIN9_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN9_MODE_OUTPUT_OPEN_DRAIN 0x0000004000000000
```

Definition at line 215 of file GPIO.h.

4.2.2.262 GPIO_PIN9_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN9_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 214 of file GPIO.h.

4.2.2.263 GPIO_PIN9_PORTA

```
#define GPIO_PIN9_PORTA 0x0000001000000000
```

Definition at line 107 of file GPIO.h.

4.2.2.264 GPIO_PIN9_PORTB

```
#define GPIO_PIN9_PORTB 0x0000002000000000
```

Definition at line 108 of file GPIO.h.

4.2.2.265 GPIO_PIN9_PORTC

```
#define GPIO_PIN9_PORTC 0x0000003000000000
```

Definition at line 109 of file GPIO.h.

4.2.2.266 GPIO_PIN9_SELECT

```
#define GPIO_PIN9_SELECT 0x000000F000000000
```

Definition at line 60 of file GPIO.h.

4.2.2.267 GPIO_PIN9_SPEED_10MHZ

```
#define GPIO_PIN9_SPEED_10MHZ 0x0000001000000000
```

Definition at line 320 of file GPIO.h.

4.2.2.268 GPIO_PIN9_SPEED_2MHZ

```
#define GPIO_PIN9_SPEED_2MHZ 0x0000002000000000
```

Definition at line 321 of file GPIO.h.

4.2.2.269 GPIO_PIN9_SPEED_50MHZ

```
#define GPIO_PIN9_SPEED_50MHZ 0x0000003000000000
```

Definition at line 322 of file GPIO.h.

4.2.2.270 GPIO_PIN9_SPEED_NONE

```
#define GPIO_PIN9_SPEED_NONE 0x0000000000000000
```

Definition at line 323 of file GPIO.h.

4.2.2.271 GPIO_PIN9_VALUE_HIGH

```
#define GPIO_PIN9_VALUE_HIGH 0x000000F000000000
```

Definition at line 32 of file GPIO.h.

4.2.2.272 GPIO_PIN9_VALUE_LOW

```
#define GPIO_PIN9_VALUE_LOW 0x0000000000000000
```

Definition at line 33 of file GPIO.h.

4.2.2.273 GPIO_PIN_ALL_MODE_INPUT_ANALOG

```
#define GPIO_PIN_ALL_MODE_INPUT_ANALOG 0x0000000000000000
```

Definition at line 267 of file GPIO.h.

4.2.2.274 GPIO_PIN_ALL_MODE_INPUT_FLOATING

```
#define GPIO_PIN_ALL_MODE_INPUT_FLOATING 0x4444444444444444
```

Definition at line 268 of file GPIO.h.

4.2.2.275 GPIO_PIN_ALL_MODE_INPUT_PULL_UP_DOWN

```
#define GPIO_PIN_ALL_MODE_INPUT_PULL_UP_DOWN 0x8888888888888888
```

Definition at line 269 of file GPIO.h.

4.2.2.276 GPIO_PIN_ALL_MODE_OUTPUT_AF_OPEN_DRAIN

```
#define GPIO_PIN_ALL_MODE_OUTPUT_AF_OPEN_DRAIN 0xCCCCCCCCCCCCCCCC
```

Definition at line 273 of file GPIO.h.

4.2.2.277 GPIO_PIN_ALL_MODE_OUTPUT_AF_PUSH_PULL

```
#define GPIO_PIN_ALL_MODE_OUTPUT_AF_PUSH_PULL 0x8888888888888888
```

Definition at line 272 of file GPIO.h.

4.2.2.278 GPIO_PIN_ALL_MODE_OUTPUT_OPEN_DRAIN

```
#define GPIO_PIN_ALL_MODE_OUTPUT_OPEN_DRAIN 0x4444444444444444
```

Definition at line 271 of file GPIO.h.

4.2.2.279 GPIO_PIN_ALL_MODE_OUTPUT_PUSH_PULL

```
#define GPIO_PIN_ALL_MODE_OUTPUT_PUSH_PULL 0x0000000000000000
```

Definition at line 270 of file GPIO.h.

4.2.2.280 GPIO_PIN_ALL_PORTA

```
#define GPIO_PIN_ALL_PORTA 0x1111111111111111
```

Definition at line 135 of file GPIO.h.

4.2.2.281 GPIO_PIN_ALL_PORTB

```
#define GPIO_PIN_ALL_PORTB 0x2222222222222222
```

Definition at line 136 of file GPIO.h.

4.2.2.282 GPIO_PIN_ALL_PORTC

```
#define GPIO_PIN_ALL_PORTC 0x3333333333333333
```

Definition at line 137 of file GPIO.h.

4.2.2.283 GPIO_PIN_ALL_SPEED_10MHZ

```
#define GPIO_PIN_ALL_SPEED_10MHZ 0x1111111111111111
```

Definition at line 355 of file GPIO.h.

4.2.2.284 GPIO_PIN_ALL_SPEED_2MHZ

```
#define GPIO_PIN_ALL_SPEED_2MHZ 0x2222222222222222
```

Definition at line 356 of file GPIO.h.

4.2.2.285 GPIO_PIN_ALL_SPEED_50MHZ

```
#define GPIO_PIN_ALL_SPEED_50MHZ 0x3333333333333333
```

Definition at line 357 of file GPIO.h.

4.2.2.286 GPIO_PIN_ALL_SPEED_NONE

```
#define GPIO_PIN_ALL_SPEED_NONE 0x0000000000000000
```

Definition at line 358 of file GPIO.h.

4.2.2.287 GPIO_PIN_ALL_VALUE_HIGH

```
#define GPIO_PIN_ALL_VALUE_HIGH 0xFFFFFFFFFFFFFFFF
```

Definition at line 47 of file GPIO.h.

4.2.2.288 GPIO_PIN_ALL_VALUE_LOW

```
#define GPIO_PIN_ALL_VALUE_LOW 0x0000000000000000
```

Definition at line 48 of file GPIO.h.

4.2.3 Function Documentation

4.2.3.1 GPIO_InitPin()

```
void GPIO_InitPin (
            const GPIO_t * gpio )
```

Initialize a GPIO object: mode, speed, direction.

Parameters

gpio	constant GPIO_t object reference

4.2.3.2 GPIO_ReadPin()

```
u64 GPIO_ReadPin (
            const GPIO_t * gpio )
```

Read multiple value from a GPIO object.

Parameters

gpio	constant GPIO_t object reference

Returns

GPIO object pins readings

4.2.3.3 GPIO_WritePin()

```
void GPIO_WritePin (
            const GPIO_t * gpio,
            u64 state )
```

Write multiple value on a GPIO object.

Parameters

gpio	constant GPIO_t object reference
state	GPIO_PINx_VALUE_x

4.3 include/LCD.h File Reference

Macros

- #define LCD_STATUS_OK 0
- #define LCD_STATUS_NOK 1
- #define LCD_CMD_Clear_Display 0x01
- #define LCD_CMD_Return_Home 0x02
- #define LCD_CMD_Entry_DecCursor_NoDisplayShift 0x04
- #define LCD_CMD_Entry_DecCursor_DisplayShift 0x05

- #define LCD_CMD_Entry_IncCursor_NoDisplayShift 0x06
- #define LCD_CMD_Entry_IncCursor_DisplayShift 0x07
- #define LCD_CMD_DisplayOff_CursorOff_BlinkOff 0x08
- #define LCD_CMD_DisplayOff_CursorOff_BlinkOn 0x09
- #define LCD_CMD_DisplayOff_CursorOn_BlinkOff 0x0A
- #define LCD_CMD_DisplayOff_CursorOn_BlinkOn 0x0B
- #define LCD_CMD_DisplayOn_CursorOff_BlinkOff 0x0C
- #define LCD_CMD_DisplayOn_CursorOff_BlinkOn 0x0D
- #define LCD_CMD_DisplayOn_CursorOn_BlinkOff 0x0E
- #define LCD_CMD_DisplayOn_CursorOn_BlinkOn 0x0F
- #define LCD_CMD_ShiftCursor_Left 0x10
- #define LCD_CMD_ShiftCursor_Right 0x14
- #define LCD_CMD_ShiftDisplay_Left 0x18
- #define LCD_CMD_ShiftDisplay_Right 0x1C
- #define LCD_CMD_Function_4BitMode_1Line_5x7Dots 0x20
- #define LCD_CMD_Function_4BitMode_1Line_5x10Dots 0x24
- #define LCD_CMD_Function_4BitMode_2Line_5x7Dots 0x28
- #define LCD_CMD_Function_4BitMode_2Line_5x10Dots 0x2C
- #define LCD_CMD_Function_8BitMode_1Line_5x7Dots 0x30
- #define LCD_CMD_Function_8BitMode_1Line_5x10Dots 0x34
- #define LCD_CMD_Function_8BitMode_2Line_5x7Dots 0x38
- #define LCD_CMD_Function_8BitMode_2Line_5x10Dots 0x3C
- #define LCD_CMD_Set_CGRAM_Addr 0x40
- #define LCD_CMD_Set_DDRAM_Addr 0x80

Typedefs

- typedef void(∗ LCD_CMD_CB_t) (void)

  LCD command callback type.
- typedef void(∗ LCD_Data_CB_t) (void)

  LCD data callback type.

Functions

- void LCD_RegisterCMD_Callback (LCD_CMD_CB_t cmdCB)
- void LCD_RegisterData_Callback (LCD_Data_CB_t dataCB)
- u8 LCD_WriteCMD (u8 cmd)
- u8 LCD_WriteData (u8 ∗data, u8 len)

4.3.1 Macro Definition Documentation

4.3.1.1 LCD_CMD_Clear_Display

```
#define LCD_CMD_Clear_Display 0x01
```

Definition at line 14 of file LCD.h.

4.3.1.2 LCD_CMD_DisplayOff_CursorOff_BlinkOff

```
#define LCD_CMD_DisplayOff_CursorOff_BlinkOff 0x08
```

Definition at line 23 of file LCD.h.

4.3.1.3 LCD_CMD_DisplayOff_CursorOff_BlinkOn

```
#define LCD_CMD_DisplayOff_CursorOff_BlinkOn 0x09
```

Definition at line 24 of file LCD.h.

4.3.1.4 LCD_CMD_DisplayOff_CursorOn_BlinkOff

```
#define LCD_CMD_DisplayOff_CursorOn_BlinkOff 0x0A
```

Definition at line 25 of file LCD.h.

4.3.1.5 LCD_CMD_DisplayOff_CursorOn_BlinkOn

```
#define LCD_CMD_DisplayOff_CursorOn_BlinkOn 0x0B
```

Definition at line 26 of file LCD.h.

4.3.1.6 LCD_CMD_DisplayOn_CursorOff_BlinkOff

```
#define LCD_CMD_DisplayOn_CursorOff_BlinkOff 0x0C
```

Definition at line 27 of file LCD.h.

4.3.1.7 LCD_CMD_DisplayOn_CursorOff_BlinkOn

```
#define LCD_CMD_DisplayOn_CursorOff_BlinkOn 0x0D
```

Definition at line 28 of file LCD.h.

4.3.1.8 LCD_CMD_DisplayOn_CursorOn_BlinkOff

```
#define LCD_CMD_DisplayOn_CursorOn_BlinkOff 0x0E
```

Definition at line 29 of file LCD.h.

4.3.1.9 LCD_CMD_DisplayOn_CursorOn_BlinkOn

```
#define LCD_CMD_DisplayOn_CursorOn_BlinkOn 0x0F
```

Definition at line 30 of file LCD.h.

4.3.1.10 LCD_CMD_Entry_DecCursor_DisplayShift

```
#define LCD_CMD_Entry_DecCursor_DisplayShift 0x05
```

Definition at line 19 of file LCD.h.

4.3.1.11 LCD_CMD_Entry_DecCursor_NoDisplayShift

```
#define LCD_CMD_Entry_DecCursor_NoDisplayShift 0x04
```

Definition at line 18 of file LCD.h.

4.3.1.12 LCD_CMD_Entry_IncCursor_DisplayShift

```
#define LCD_CMD_Entry_IncCursor_DisplayShift 0x07
```

Definition at line 21 of file LCD.h.

4.3.1.13 LCD_CMD_Entry_IncCursor_NoDisplayShift

```
#define LCD_CMD_Entry_IncCursor_NoDisplayShift 0x06
```

Definition at line 20 of file LCD.h.

4.3.1.14 LCD_CMD_Function_4BitMode_1Line_5x10Dots

```
#define LCD_CMD_Function_4BitMode_1Line_5x10Dots 0x24
```

Definition at line 38 of file LCD.h.

4.3.1.15 LCD_CMD_Function_4BitMode_1Line_5x7Dots

```
#define LCD_CMD_Function_4BitMode_1Line_5x7Dots 0x20
```

Definition at line 37 of file LCD.h.

4.3.1.16 LCD_CMD_Function_4BitMode_2Line_5x10Dots

```
#define LCD_CMD_Function_4BitMode_2Line_5x10Dots 0x2C
```

Definition at line 40 of file LCD.h.

4.3.1.17 LCD_CMD_Function_4BitMode_2Line_5x7Dots

```
#define LCD_CMD_Function_4BitMode_2Line_5x7Dots 0x28
```

Definition at line 39 of file LCD.h.

4.3.1.18 LCD_CMD_Function_8BitMode_1Line_5x10Dots

```
#define LCD_CMD_Function_8BitMode_1Line_5x10Dots 0x34
```

Definition at line 42 of file LCD.h.

4.3.1.19 LCD_CMD_Function_8BitMode_1Line_5x7Dots

```
#define LCD_CMD_Function_8BitMode_1Line_5x7Dots 0x30
```

Definition at line 41 of file LCD.h.

4.3.1.20 LCD_CMD_Function_8BitMode_2Line_5x10Dots

```
#define LCD_CMD_Function_8BitMode_2Line_5x10Dots 0x3C
```

Definition at line 44 of file LCD.h.

4.3.1.21 LCD_CMD_Function_8BitMode_2Line_5x7Dots

```
#define LCD_CMD_Function_8BitMode_2Line_5x7Dots 0x38
```

Definition at line 43 of file LCD.h.

4.3.1.22 LCD_CMD_Return_Home

```
#define LCD_CMD_Return_Home 0x02
```

Definition at line 16 of file LCD.h.

4.3.1.23 LCD_CMD_Set_CGRAM_Addr

```
#define LCD_CMD_Set_CGRAM_Addr 0x40
```

Definition at line 47 of file LCD.h.

4.3.1.24 LCD_CMD_Set_DDRAM_Addr

```
#define LCD_CMD_Set_DDRAM_Addr 0x80
```

Definition at line 48 of file LCD.h.

4.3.1.25 LCD_CMD_ShiftCursor_Left

```
#define LCD_CMD_ShiftCursor_Left 0x10
```

Definition at line 32 of file LCD.h.

4.3.1.26 LCD_CMD_ShiftCursor_Right

```
#define LCD_CMD_ShiftCursor_Right 0x14
```

Definition at line 33 of file LCD.h.

4.3.1.27 LCD_CMD_ShiftDisplay_Left

```
#define LCD_CMD_ShiftDisplay_Left 0x18
```

Definition at line 34 of file LCD.h.

4.3.1.28 LCD_CMD_ShiftDisplay_Right

```
#define LCD_CMD_ShiftDisplay_Right 0x1C
```

Definition at line 35 of file LCD.h.

4.3.1.29 LCD_STATUS_NOK

```
#define LCD_STATUS_NOK 1
```

Definition at line 12 of file LCD.h.

4.3.1.30 LCD_STATUS_OK

```
#define LCD_STATUS_OK 0
```

Definition at line 11 of file LCD.h.

4.3.2 Typedef Documentation

4.3.2.1 LCD_CMD_CB_t

```
typedef void(* LCD_CMD_CB_t) (void)
```

LCD command callback type.

Definition at line 54 of file LCD.h.

4.3.2.2 LCD_Data_CB_t

```
typedef void(* LCD_Data_CB_t) (void)
```

LCD data callback type.

Definition at line 60 of file LCD.h.

4.3.3 Function Documentation

4.3.3.1 LCD_RegisterCMD_Callback()

```
void LCD_RegisterCMD_Callback (
            LCD_CMD_CB_t cmdCB )
```

4.3.3.2 LCD_RegisterData_Callback()

```
void LCD_RegisterData_Callback (
            LCD_Data_CB_t dataCB )
```

4.3.3.3 LCD_WriteCMD()

```
u8 LCD_WriteCMD (
            u8 cmd )
```

4.3.3.4 LCD_WriteData()

```
u8 LCD_WriteData (
            u8 * data,
            u8 len )
```

4.4 include/LCD_cfg.h File Reference

Data Structures

- struct LCD_cfg_t

Macros

- #define LCD_QUEUE_MAX_LEN 10
- #define LCD_DATA_MAX_LEN 16
- #define LCD_COUNT 1

4.4.1 Macro Definition Documentation

4.4.1.1 LCD_COUNT

```
#define LCD_COUNT 1
```

Definition at line 17 of file LCD_cfg.h.

4.4.1.2 LCD_DATA_MAX_LEN

```
#define LCD_DATA_MAX_LEN 16
```

Definition at line 15 of file LCD_cfg.h.

4.4.1.3 LCD_QUEUE_MAX_LEN

```
#define LCD_QUEUE_MAX_LEN 10
```

Definition at line 12 of file LCD_cfg.h.

4.5 include/NVIC.h File Reference

This file is to be used as an implementation of the NVIC driver.

Macros

- #define EXTI0_IRQNUMBER 6U
- #define EXTI1_IRQNUMBER 7U
- #define EXTI2_IRQNUMBER 8U
- #define EXTI3_IRQNUMBER 9U
- #define EXTI4_IRQNUMBER 10U
- #define USART1_IRQNUMBER 37U
- #define USART2_IRQNUMBER 38U
- #define USART3_IRQNUMBER 39U
- #define UART4_IRQNUMBER 52U
- #define UART5_IRQNUMBER 53U
- #define DMA1_Channel4_IRQHandler 14U
- #define DMA1_Channel5_IRQHandler 15U

Functions

- Std_ReturnType NVIC_EnableIRQ (u8 IRQNumber)

  Function to enable interrupt.
- Std_ReturnType NVIC_DisableIRQ (u8 IRQNumber)

  Function to disable interrupt.
- Std_ReturnType NVIC_SetPendingIRQ (u8 IRQNumber)

  Function to changes interrupt state to pending.
- Std_ReturnType NVIC_ClearPendingIRQ (u8 IRQNumber)

  Function to removes the pending state of an interrupt.
- Std_ReturnType NVIC_ISActive (u8 IRQNumber)

  Function return status if the corresponding interrupt is active or not.
- Std_ReturnType NVIC_SetPriority (u8 IRQNumber, u8 Priority)

  Function return status if the corresponding interrupt is active or not.
- Std_ReturnType NVIC_SoftwareInterrupt (u8 IRQNumber)

  Function to generate interrupt software.
- void NVIC_EnableAllInterrupt (void)

  Function to enable IRQ interrupts.
- void NVIC_DisableAllInterrupt (void)

  Function to disable IRQ interrupts.
- void NVIC_DisableAllFaults (void)

  Function to disable all fault exceptions.
- void NVIC_SetPriorityGrouping (u32 priority_grouping)

  Set priority group.

4.5.1 Detailed Description

This file is to be used as an implementation of the NVIC driver.

Author

MSN

Date

Mar 31, 2020

4.5.2 Macro Definition Documentation

4.5.2.1 DMA1_Channel4_IRQHandler

```
#define DMA1_Channel4_IRQHandler 14U
```

Definition at line 24 of file NVIC.h.

4.5.2.2 DMA1_Channel5_IRQHandler

```
#define DMA1_Channel5_IRQHandler 15U
```

Definition at line 25 of file NVIC.h.

4.5.2.3 EXTI0_IRQNUMBER

```
#define EXTI0_IRQNUMBER 6U
```

Definition at line 14 of file NVIC.h.

4.5.2.4 EXTI1_IRQNUMBER

```
#define EXTI1_IRQNUMBER 7U
```

Definition at line 15 of file NVIC.h.

4.5.2.5 EXTI2_IRQNUMBER

```
#define EXTI2_IRQNUMBER 8U
```

Definition at line 16 of file NVIC.h.

4.5.2.6 EXTI3_IRQNUMBER

```
#define EXTI3_IRQNUMBER 9U
```

Definition at line 17 of file NVIC.h.

4.5.2.7 EXTI4_IRQNUMBER

```
#define EXTI4_IRQNUMBER 10U
```

Definition at line 18 of file NVIC.h.

4.5.2.8 UART4_IRQNUMBER

```
#define UART4_IRQNUMBER 52U
```

Definition at line 22 of file NVIC.h.

4.5.2.9 UART5_IRQNUMBER

```
#define UART5_IRQNUMBER 53U
```

Definition at line 23 of file NVIC.h.

4.5.2.10 USART1_IRQNUMBER

```
#define USART1_IRQNUMBER 37U
```

Definition at line 19 of file NVIC.h.

4.5.2.11 USART2_IRQNUMBER

```
#define USART2_IRQNUMBER 38U
```

Definition at line 20 of file NVIC.h.

4.5.2.12 USART3_IRQNUMBER

```
#define USART3_IRQNUMBER 39U
```

Definition at line 21 of file NVIC.h.

4.5.3 Function Documentation

4.5.3.1 NVIC_ClearPendingIRQ()

```
Std_ReturnType NVIC_ClearPendingIRQ (
            u8 IRQNumber )
```

Function to removes the pending state of an interrupt.

Parameters

IRQNumber	interrupt request number from 0 to 80

Returns

Std_ReturnType
E_OK: request accepted
E_NOT_OK: request not accepted

4.5.3.2 NVIC_DisableAllFaults()

```
void NVIC_DisableAllFaults (
            void  )
```

Function to disable all fault exceptions.

Parameters

void	

Returns

void

4.5.3.3 NVIC_DisableAllInterrupt()

```
void NVIC_DisableAllInterrupt (
            void  )
```

Function to disable IRQ interrupts.

Parameters

void	

Returns

void

4.5.3.4 NVIC_DisableIRQ()

```
Std_ReturnType NVIC_DisableIRQ (
            u8 IRQNumber )
```

Function to disable interrupt.

Parameters

IRQNumber	interrupt request number from 0 to 80

Returns

Std_ReturnType
E_OK: request accepted
E_NOT_OK: request not accepted

4.5.3.5 NVIC_EnableAllInterrupt()

```
void NVIC_EnableAllInterrupt (
            void  )
```

Function to enable IRQ interrupts.

Parameters

void	

Returns

void

4.5.3.6 NVIC_EnableIRQ()

```
Std_ReturnType NVIC_EnableIRQ (
            u8 IRQNumber )
```

Function to enable interrupt.

Parameters

IRQNumber	interrupt request number from 0 to 80

Returns

Std_ReturnType:
E_OK: request accepted
E_NOT_OK: request not accepted

4.5.3.7 NVIC_ISActive()

```
Std_ReturnType NVIC_ISActive (
            u8 IRQNumber )
```

Function return status if the corresponding interrupt is active or not.

Parameters

IRQNumber	interrupt request number from 0 to 80

Returns

Std_ReturnType
E_OK: request accepted
E_NOT_OK: request not accepted

4.5.3.8 NVIC_SetPendingIRQ()

```
Std_ReturnType NVIC_SetPendingIRQ (
            u8 IRQNumber )
```

Function to changes interrupt state to pending.

Parameters

IRQNumber	interrupt request number from 0 to 80

Returns

Std_ReturnType
E_OK: request accepted
E_NOT_OK: request not accepted

4.5.3.9 NVIC_SetPriority()

```
Std_ReturnType NVIC_SetPriority (
            u8 IRQNumber,
            u8 Priority )
```

Function return status if the corresponding interrupt is active or not.

Parameters

IRQNumber	interrupt request number from 0 to 80
Priority	interrupt priority number

Returns

Std_ReturnType
E_OK: request accepted
E_NOT_OK: request not accepted

4.5.3.10 NVIC_SetPriorityGrouping()

```
void NVIC_SetPriorityGrouping (
            u32 priority_grouping )
```

Set priority group.

Parameters

priority_grouping	priority group

Returns

void

4.5.3.11 NVIC_SoftwareInterrupt()

```
Std_ReturnType NVIC_SoftwareInterrupt (
            u8 IRQNumber )
```

Function to generate interrupt software.

Parameters

IRQNumber	interrupt request number from 0 to 80

Returns

Std_ReturnType
E_OK: request accepted
E_NOT_OK: request not accepted

4.6 include/RCC.h File Reference

```
#include "STD_TYPES.h"
```

Macros

- #define ClockSourceType (u32)
- #define RCC_CR_HSI (u32)0x00000001
- #define RCC_CR_HSIRDY (u32)0x00000002
- #define RCC_CR_HSE (u32)0x00010000
- #define RCC_CR_HSERDY (u32)0x00020000
- #define RCC_CR_HSEBYP (u32)0x00030000
- #define RCC_CR_CSSON (u32)0x00040000
- #define RCC_CR_PLL (u32)0x01000000
- #define RCC_CR_PLLRDY (u32)0x02010000
- #define RCC_CFGR_SW_HSI (u32)0x00000000
- #define RCC_CFGR_SW_HSE (u32)0x00000001
- #define RCC_CFGR_SW_PLL (u32)0x00000002
- #define RCC_CFGR_SWS_HSI (u32)0x0
- #define RCC_CFGR_SWS_HSE (u32)0x4
- #define RCC_CFGR_SWS_PLL (u32)0x8
- #define RCC_CFGR_PLLMUL_2 (u32)0x00000000
- #define RCC_CFGR_PLLMUL_3 (u32)0x00040000
- #define RCC_CFGR_PLLMUL_4 (u32)0x00080000
- #define RCC_CFGR_PLLMUL_5 (u32)0x000C0000
- #define RCC_CFGR_PLLMUL_6 (u32)0x00100000
- #define RCC_CFGR_PLLMUL_7 (u32)0x00140000
- #define RCC_CFGR_PLLMUL_8 (u32)0x00180000
- #define RCC_CFGR_PLLMUL_9 (u32)0x001C0000
- #define RCC_CFGR_PLLMUL_10 (u32)0x00200000
- #define RCC_CFGR_PLLMUL_11 (u32)0x00240000
- #define RCC_CFGR_PLLMUL_12 (u32)0x00280000
- #define RCC_CFGR_PLLMUL_13 (u32)0x002C0000
- #define RCC_CFGR_PLLMUL_14 (u32)0x00300000
- #define RCC_CFGR_PLLMUL_15 (u32)0x00340000
- #define RCC_CFGR_PLLMUL_16 (u32)0x00380000
- #define RCC_CFGR_PLLXTPRESRC_HSE_divided_1 (u32)0x00010000
- #define RCC_CFGR_PLLXTPRESRC_HSE_divided_2 (u32)0x00030000
- #define RCC_CFGR_PLLXTPRESRC_HSI_divided_2 (u32)0x00000000
- #define RCC_CFGR_PPRE1_div_1 (u32)0x00000000
- #define RCC_CFGR_PPRE1_div_2 (u32)0x00000400
- #define RCC_CFGR_PPRE1_div_4 (u32)0x00000500
- #define RCC_CFGR_PPRE1_div_8 (u32)0x00000600
- #define RCC_CFGR_PPRE1_div_16 (u32)0x00000700
- #define RCC_CFGR_PPRE2_div_1 (u32)0x00000000
- #define RCC_CFGR_PPRE2_div_2 (u32)0x00002000
- #define RCC_CFGR_PPRE2_div_4 (u32)0x00002800
- #define RCC_CFGR_PPRE2_div_8 (u32)0x00003000
- #define RCC_CFGR_PPRE2_div_16 (u32)0x00003800
- #define RCC_CFGR_HPRE_div_1 (u32)0x00000000
- #define RCC_CFGR_HPRE_div_2 (u32)0x00000080
- #define RCC_CFGR_HPRE_div_4 (u32)0x00000090

- #define RCC_CFGR_HPRE_div_8 (u32)0x000000A0
- #define RCC_CFGR_HPRE_div_16 (u32)0x000000B0
- #define RCC_CFGR_HPRE_div_64 (u32)0x000000C0
- #define RCC_CFGR_HPRE_div_128 (u32)0x000000D0
- #define RCC_CFGR_HPRE_div_256 (u32)0x000000E0
- #define RCC_CFGR_HPRE_div_512 (u32)0x000000F0
- #define RCC_CFGR_ADCPRE_div_2 (u32)0x0
- #define RCC_CFGR_ADCPRE_div_4 (u32)0x4000
- #define RCC_CFGR_ADCPRE_div_6 (u32)0x8000
- #define RCC_CFGR_ADCPRE_div_8 (u32)0xc000
- #define RCC_CFGR_MCO_NOCLK (u32)0x00000000
- #define RCC_CFGR_MCO_SYSCLK (u32)0x4000000
- #define RCC_CFGR_MCO_HSI (u32)0x5000000
- #define RCC_CFGR_MCO_HSE (u32)0x6000000
- #define RCC_CFGR_MCO_PLL (u32)0x7000000
- #define RCC_APB2ENR_AFIOEN_Enable (u32)0x1
- #define RCC_APB2ENR_AFIOEN_Disable (u32)0x0
- #define RCC_APB2ENR_IOPAEN_PORTA (u32)0x4
- #define RCC_APB2ENR_IOPBEN_PORTB (u32)0x8
- #define RCC_APB2ENR_IOPCEN_PORTC (u32)0x10
- #define RCC_APB2ENR_IOPDEN_PORTD (u32)0x20
- #define RCC_APB2ENR_IOPEEN_PORTE (u32)0x40
- #define RCC_APB2ENR_IOPFEN_PORTF (u32)0x80
- #define RCC_APB2ENR_IOPGEN_PORTG (u32)0x100
- #define RCC_APB2ENR_ADC1EN (u32)0x200
- #define RCC_APB2ENR_ADC2EN (u32)0x400
- #define RCC_APB2ENR_TIM1EN (u32)0x800
- #define RCC_APB2ENR_SPI1EN (u32)0x1000
- #define RCC_APB2ENR_TIM8EN (u32)0x2000
- #define RCC_APB2ENR_TIM9EN (u32)0x80000
- #define RCC_APB2ENR_TIM10EN (u32)0x100000
- #define RCC_APB2ENR_TIM11EN (u32)0x200000
- #define RCC_APB2ENR_USART1EN (u32)0x4000
- #define RCC_APB2ENR_ADC3EN (u32)0x8000
- #define RCC_APB2ENR_ADC3EN (u32)0x8000
- #define RCC_APB1ENR_USART2EN (u32)0x20000
- #define RCC_APB1ENR_USART3EN (u32)0x40000
- #define RCC_APB1ENR_UART4EN (u32)0x80000
- #define RCC_APB1ENR_UART5EN (u32)0x100000
- #define RCC_AHBENR_DMA1EN (u32)0x01
- #define RCC_AHBENR_DMA2EN (u32)0x02
- #define ON 1
- #define OFF 0

Typedefs

- typedef u32 ClockSource_ReturnType

Functions

- void Select_SystemClock (u32 SystemClock)
- void RCC_SetClock (u32 Clock, u8 State)
- ClockSource_ReturnType RCC_CheckSystemClock (void)
- void RCC_PLLConfiguration (u32 RCC_PLLSource, u32 RCC_PLLMUL)
- void RCC_PPRE2_SetPrescaler (u32 Prescaler)
- void RCC_PPRE1_SetPrescaler (u32 Prescaler)
- void RCC_HPRE_SetPrescaler (u32 Prescaler)
- void RCC_ADCPRE_SetPrescaler (u32 Prescaler)
- void RCC_SelectMCO (u32 Clock)
- void RCC_EnablePeripheral_APB2 (u32 Peripheral)
- void RCC_EnablePeripheral_APB1 (u32 Peripheral)
- void RCC_EnablePeripheral_AHB (u32 Peripheral)

4.6.1 Macro Definition Documentation

4.6.1.1 ClockSourceType

```
#define ClockSourceType (u32)
```

Definition at line 7 of file RCC.h.

4.6.1.2 OFF

```
#define OFF 0
```

Definition at line 139 of file RCC.h.

4.6.1.3 ON

```
#define ON 1
```

Definition at line 138 of file RCC.h.

4.6.1.4 RCC_AHBENR_DMA1EN

```
#define RCC_AHBENR_DMA1EN (u32)0x01
```

Definition at line 134 of file RCC.h.

4.6.1.5 RCC_AHBENR_DMA2EN

```
#define RCC_AHBENR_DMA2EN (u32)0x02
```

Definition at line 135 of file RCC.h.

4.6.1.6 RCC_APB1ENR_UART4EN

```
#define RCC_APB1ENR_UART4EN (u32)0x80000
```

Definition at line 129 of file RCC.h.

4.6.1.7 RCC_APB1ENR_UART5EN

```
#define RCC_APB1ENR_UART5EN (u32)0x100000
```

Definition at line 130 of file RCC.h.

4.6.1.8 RCC_APB1ENR_USART2EN

```
#define RCC_APB1ENR_USART2EN (u32)0x20000
```

Definition at line 127 of file RCC.h.

4.6.1.9 RCC_APB1ENR_USART3EN

```
#define RCC_APB1ENR_USART3EN (u32)0x40000
```

Definition at line 128 of file RCC.h.

4.6.1.10 RCC_APB2ENR_ADC1EN

```
#define RCC_APB2ENR_ADC1EN (u32)0x200
```

Definition at line 114 of file RCC.h.

4.6.1.11 RCC_APB2ENR_ADC2EN

```
#define RCC_APB2ENR_ADC2EN (u32)0x400
```

Definition at line 115 of file RCC.h.

4.6.1.12 RCC_APB2ENR_ADC3EN [1/2]

```
#define RCC_APB2ENR_ADC3EN (u32)0x8000
```

Definition at line 124 of file RCC.h.

4.6.1.13 RCC_APB2ENR_ADC3EN [2/2]

```
#define RCC_APB2ENR_ADC3EN (u32)0x8000
```

Definition at line 124 of file RCC.h.

4.6.1.14 RCC_APB2ENR_AFIOEN_Disable

```
#define RCC_APB2ENR_AFIOEN_Disable (u32)0x0
```

Definition at line 105 of file RCC.h.

4.6.1.15 RCC_APB2ENR_AFIOEN_Enable

```
#define RCC_APB2ENR_AFIOEN_Enable (u32)0x1
```

Definition at line 104 of file RCC.h.

4.6.1.16 RCC_APB2ENR_IOPAEN_PORTA

```
#define RCC_APB2ENR_IOPAEN_PORTA (u32)0x4
```

Definition at line 107 of file RCC.h.

4.6.1.17 RCC_APB2ENR_IOPBEN_PORTB

```
#define RCC_APB2ENR_IOPBEN_PORTB (u32)0x8
```

Definition at line 108 of file RCC.h.

4.6.1.18 RCC_APB2ENR_IOPCEN_PORTC

```
#define RCC_APB2ENR_IOPCEN_PORTC (u32)0x10
```

Definition at line 109 of file RCC.h.

4.6.1.19 RCC_APB2ENR_IOPDEN_PORTD

```
#define RCC_APB2ENR_IOPDEN_PORTD (u32)0x20
```

Definition at line 110 of file RCC.h.

4.6.1.20 RCC_APB2ENR_IOPEEN_PORTE

```
#define RCC_APB2ENR_IOPEEN_PORTE (u32)0x40
```

Definition at line 111 of file RCC.h.

4.6.1.21 RCC_APB2ENR_IOPFEN_PORTF

```
#define RCC_APB2ENR_IOPFEN_PORTF (u32)0x80
```

Definition at line 112 of file RCC.h.

4.6.1.22 RCC_APB2ENR_IOPGEN_PORTG

```
#define RCC_APB2ENR_IOPGEN_PORTG (u32)0x100
```

Definition at line 113 of file RCC.h.

4.6.1.23 RCC_APB2ENR_SPI1EN

```
#define RCC_APB2ENR_SPI1EN (u32)0x1000
```

Definition at line 117 of file RCC.h.

4.6.1.24 RCC_APB2ENR_TIM10EN

```
#define RCC_APB2ENR_TIM10EN (u32)0x100000
```

Definition at line 120 of file RCC.h.

4.6.1.25 RCC_APB2ENR_TIM11EN

```
#define RCC_APB2ENR_TIM11EN (u32)0x200000
```

Definition at line 121 of file RCC.h.

4.6.1.26 RCC_APB2ENR_TIM1EN

```
#define RCC_APB2ENR_TIM1EN (u32)0x800
```

Definition at line 116 of file RCC.h.

4.6.1.27 RCC_APB2ENR_TIM8EN

```
#define RCC_APB2ENR_TIM8EN (u32)0x2000
```

Definition at line 118 of file RCC.h.

4.6.1.28 RCC_APB2ENR_TIM9EN

```
#define RCC_APB2ENR_TIM9EN (u32)0x80000
```

Definition at line 119 of file RCC.h.

4.6.1.29 RCC_APB2ENR_USART1EN

```
#define RCC_APB2ENR_USART1EN (u32)0x4000
```

Definition at line 122 of file RCC.h.

4.6.1.30 RCC_CFGR_ADCPRE_div_2

```
#define RCC_CFGR_ADCPRE_div_2 (u32)0x0
```

Definition at line 89 of file RCC.h.

4.6.1.31 RCC_CFGR_ADCPRE_div_4

```
#define RCC_CFGR_ADCPRE_div_4 (u32)0x4000
```

Definition at line 90 of file RCC.h.

4.6.1.32 RCC_CFGR_ADCPRE_div_6

```
#define RCC_CFGR_ADCPRE_div_6 (u32)0x8000
```

Definition at line 91 of file RCC.h.

4.6.1.33 RCC_CFGR_ADCPRE_div_8

```
#define RCC_CFGR_ADCPRE_div_8 (u32)0xc000
```

Definition at line 92 of file RCC.h.

4.6.1.34 RCC_CFGR_HPRE_div_1

```
#define RCC_CFGR_HPRE_div_1 (u32)0x00000000
```

Definition at line 78 of file RCC.h.

4.6.1.35 RCC_CFGR_HPRE_div_128

```
#define RCC_CFGR_HPRE_div_128 (u32)0x000000D0
```

Definition at line 84 of file RCC.h.

4.6.1.36 RCC_CFGR_HPRE_div_16

```
#define RCC_CFGR_HPRE_div_16 (u32)0x000000B0
```

Definition at line 82 of file RCC.h.

4.6.1.37 RCC_CFGR_HPRE_div_2

```
#define RCC_CFGR_HPRE_div_2 (u32)0x00000080
```

Definition at line 79 of file RCC.h.

4.6.1.38 RCC_CFGR_HPRE_div_256

```
#define RCC_CFGR_HPRE_div_256 (u32)0x000000E0
```

Definition at line 85 of file RCC.h.

4.6.1.39 RCC_CFGR_HPRE_div_4

```
#define RCC_CFGR_HPRE_div_4 (u32)0x00000090
```

Definition at line 80 of file RCC.h.

4.6.1.40 RCC_CFGR_HPRE_div_512

```
#define RCC_CFGR_HPRE_div_512 (u32)0x000000F0
```

Definition at line 86 of file RCC.h.

4.6.1.41 RCC_CFGR_HPRE_div_64

```
#define RCC_CFGR_HPRE_div_64 (u32)0x000000C0
```

Definition at line 83 of file RCC.h.

4.6.1.42 RCC_CFGR_HPRE_div_8

```
#define RCC_CFGR_HPRE_div_8 (u32)0x000000A0
```

Definition at line 81 of file RCC.h.

4.6.1.43 RCC_CFGR_MCO_HSE

```
#define RCC_CFGR_MCO_HSE (u32)0x6000000
```

Definition at line 99 of file RCC.h.

4.6.1.44 RCC_CFGR_MCO_HSI

```
#define RCC_CFGR_MCO_HSI (u32)0x5000000
```

Definition at line 98 of file RCC.h.

4.6.1.45 RCC_CFGR_MCO_NOCLK

```
#define RCC_CFGR_MCO_NOCLK (u32)0x00000000
```

Definition at line 96 of file RCC.h.

4.6.1.46 RCC_CFGR_MCO_PLL

```
#define RCC_CFGR_MCO_PLL (u32)0x7000000
```

Definition at line 100 of file RCC.h.

4.6.1.47 RCC_CFGR_MCO_SYSCLK

```
#define RCC_CFGR_MCO_SYSCLK (u32)0x4000000
```

Definition at line 97 of file RCC.h.

4.6.1.48 RCC_CFGR_PLLMUL_10

```
#define RCC_CFGR_PLLMUL_10 (u32)0x00200000
```

Definition at line 49 of file RCC.h.

4.6.1.49 RCC_CFGR_PLLMUL_11

```
#define RCC_CFGR_PLLMUL_11 (u32)0x00240000
```

Definition at line 50 of file RCC.h.

4.6.1.50 RCC_CFGR_PLLMUL_12

```
#define RCC_CFGR_PLLMUL_12 (u32)0x00280000
```

Definition at line 51 of file RCC.h.

4.6.1.51 RCC_CFGR_PLLMUL_13

```
#define RCC_CFGR_PLLMUL_13 (u32)0x002C0000
```

Definition at line 52 of file RCC.h.

4.6.1.52 RCC_CFGR_PLLMUL_14

```
#define RCC_CFGR_PLLMUL_14 (u32)0x00300000
```

Definition at line 53 of file RCC.h.

4.6.1.53 RCC_CFGR_PLLMUL_15

```
#define RCC_CFGR_PLLMUL_15 (u32)0x00340000
```

Definition at line 54 of file RCC.h.

4.6.1.54 RCC_CFGR_PLLMUL_16

```
#define RCC_CFGR_PLLMUL_16 (u32)0x00380000
```

Definition at line 55 of file RCC.h.

4.6.1.55 RCC_CFGR_PLLMUL_2

```
#define RCC_CFGR_PLLMUL_2 (u32)0x00000000
```

Definition at line 41 of file RCC.h.

4.6.1.56 RCC_CFGR_PLLMUL_3

```
#define RCC_CFGR_PLLMUL_3 (u32)0x00040000
```

Definition at line 42 of file RCC.h.

4.6.1.57 RCC_CFGR_PLLMUL_4

```
#define RCC_CFGR_PLLMUL_4 (u32)0x00080000
```

Definition at line 43 of file RCC.h.

4.6.1.58 RCC_CFGR_PLLMUL_5

```
#define RCC_CFGR_PLLMUL_5 (u32)0x000C0000
```

Definition at line 44 of file RCC.h.

4.6.1.59 RCC_CFGR_PLLMUL_6

```
#define RCC_CFGR_PLLMUL_6 (u32)0x00100000
```

Definition at line 45 of file RCC.h.

4.6.1.60 RCC_CFGR_PLLMUL_7

```
#define RCC_CFGR_PLLMUL_7 (u32)0x00140000
```

Definition at line 46 of file RCC.h.

4.6.1.61 RCC_CFGR_PLLMUL_8

```
#define RCC_CFGR_PLLMUL_8 (u32)0x00180000
```

Definition at line 47 of file RCC.h.

4.6.1.62 RCC_CFGR_PLLMUL_9

```
#define RCC_CFGR_PLLMUL_9 (u32)0x001C0000
```

Definition at line 48 of file RCC.h.

4.6.1.63 RCC_CFGR_PLLXTPRESRC_HSE_divided_1

```
#define RCC_CFGR_PLLXTPRESRC_HSE_divided_1 (u32)0x00010000
```

Definition at line 58 of file RCC.h.

4.6.1.64 RCC_CFGR_PLLXTPRESRC_HSE_divided_2

```
#define RCC_CFGR_PLLXTPRESRC_HSE_divided_2 (u32)0x00030000
```

Definition at line 59 of file RCC.h.

4.6.1.65 RCC_CFGR_PLLXTPRESRC_HSI_divided_2

```
#define RCC_CFGR_PLLXTPRESRC_HSI_divided_2 (u32)0x00000000
```

Definition at line 60 of file RCC.h.

4.6.1.66 RCC_CFGR_PPRE1_div_1

```
#define RCC_CFGR_PPRE1_div_1 (u32)0x00000000
```

Definition at line 63 of file RCC.h.

4.6.1.67 RCC_CFGR_PPRE1_div_16

```
#define RCC_CFGR_PPRE1_div_16 (u32)0x00000700
```

Definition at line 67 of file RCC.h.

4.6.1.68 RCC_CFGR_PPRE1_div_2

```
#define RCC_CFGR_PPRE1_div_2 (u32)0x00000400
```

Definition at line 64 of file RCC.h.

4.6.1.69 RCC_CFGR_PPRE1_div_4

```
#define RCC_CFGR_PPRE1_div_4 (u32)0x00000500
```

Definition at line 65 of file RCC.h.

4.6.1.70 RCC_CFGR_PPRE1_div_8

```
#define RCC_CFGR_PPRE1_div_8 (u32)0x00000600
```

Definition at line 66 of file RCC.h.

4.6.1.71 RCC_CFGR_PPRE2_div_1

```
#define RCC_CFGR_PPRE2_div_1 (u32)0x00000000
```

Definition at line 70 of file RCC.h.

4.6.1.72 RCC_CFGR_PPRE2_div_16

```
#define RCC_CFGR_PPRE2_div_16 (u32)0x00003800
```

Definition at line 74 of file RCC.h.

4.6.1.73 RCC_CFGR_PPRE2_div_2

```
#define RCC_CFGR_PPRE2_div_2 (u32)0x00002000
```

Definition at line 71 of file RCC.h.

4.6.1.74 RCC_CFGR_PPRE2_div_4

```
#define RCC_CFGR_PPRE2_div_4 (u32)0x00002800
```

Definition at line 72 of file RCC.h.

4.6.1.75 RCC_CFGR_PPRE2_div_8

```
#define RCC_CFGR_PPRE2_div_8 (u32)0x00003000
```

Definition at line 73 of file RCC.h.

4.6.1.76 RCC_CFGR_SW_HSE

```
#define RCC_CFGR_SW_HSE (u32)0x00000001
```

Definition at line 30 of file RCC.h.

4.6.1.77 RCC_CFGR_SW_HSI

```
#define RCC_CFGR_SW_HSI (u32)0x00000000
```

Definition at line 29 of file RCC.h.

4.6.1.78 RCC_CFGR_SW_PLL

```
#define RCC_CFGR_SW_PLL (u32)0x00000002
```

Definition at line 31 of file RCC.h.

4.6.1.79 RCC_CFGR_SWS_HSE

```
#define RCC_CFGR_SWS_HSE (u32)0x4
```

Definition at line 35 of file RCC.h.

4.6.1.80 RCC_CFGR_SWS_HSI

```
#define RCC_CFGR_SWS_HSI (u32)0x0
```

Definition at line 34 of file RCC.h.

4.6.1.81 RCC_CFGR_SWS_PLL

```
#define RCC_CFGR_SWS_PLL (u32)0x8
```

Definition at line 36 of file RCC.h.

4.6.1.82 RCC_CR_CSSON

```
#define RCC_CR_CSSON (u32)0x00040000
```

Definition at line 20 of file RCC.h.

4.6.1.83 RCC_CR_HSE

```
#define RCC_CR_HSE (u32)0x00010000
```

Definition at line 16 of file RCC.h.

4.6.1.84 RCC_CR_HSEBYP

```
#define RCC_CR_HSEBYP (u32)0x00030000
```

Definition at line 19 of file RCC.h.

4.6.1.85 RCC_CR_HSERDY

```
#define RCC_CR_HSERDY (u32)0x00020000
```

Definition at line 17 of file RCC.h.

4.6.1.86 RCC_CR_HSI

```
#define RCC_CR_HSI (u32)0x00000001
```

Definition at line 13 of file RCC.h.

4.6.1.87 RCC_CR_HSIRDY

```
#define RCC_CR_HSIRDY (u32)0x00000002
```

Definition at line 14 of file RCC.h.

4.6.1.88 RCC_CR_PLL

```
#define RCC_CR_PLL (u32)0x01000000
```

Definition at line 22 of file RCC.h.

4.6.1.89 RCC_CR_PLLRDY

```
#define RCC_CR_PLLRDY (u32)0x02010000
```

Definition at line 23 of file RCC.h.

4.6.2 Typedef Documentation

4.6.2.1 ClockSource_ReturnType

```
typedef u32 ClockSource_ReturnType
```

Definition at line 4 of file RCC.h.

4.6.3 Function Documentation

4.6.3.1 RCC_ADCPRE_SetPrescaler()

```
void RCC_ADCPRE_SetPrescaler (
            u32 Prescaler )
```

4.6.3.2 RCC_CheckSystemClock()

```
ClockSource_ReturnType RCC_CheckSystemClock (
            void  )
```

4.6.3.3 RCC_EnablePeripheral_AHB()

```
void RCC_EnablePeripheral_AHB (
            u32 Peripheral )
```

4.6.3.4 RCC_EnablePeripheral_APB1()

```
void RCC_EnablePeripheral_APB1 (
            u32 Peripheral )
```

4.6.3.5 RCC_EnablePeripheral_APB2()

```
void RCC_EnablePeripheral_APB2 (
            u32 Peripheral )
```

4.6.3.6 RCC_HPRE_SetPrescaler()

```
void RCC_HPRE_SetPrescaler (
            u32 Prescaler )
```

4.6.3.7 RCC_PLLConfiguration()

```
void RCC_PLLConfiguration (
            u32 RCC_PLLSource,
            u32 RCC_PLLMUL )
```

4.6.3.8 RCC_PPRE1_SetPrescaler()

```
void RCC_PPRE1_SetPrescaler (
            u32 Prescaler )
```

4.6.3.9 RCC_PPRE2_SetPrescaler()

```
void RCC_PPRE2_SetPrescaler (
            u32 Prescaler )
```

4.6.3.10 RCC_SelectMCO()

```
void RCC_SelectMCO (
            u32 Clock )
```

4.6.3.11 RCC_SetClock()

```
void RCC_SetClock (
            u32 Clock,
            u8 State )
```

4.6.3.12 Select_SystemClock()

```
void Select_SystemClock (
             u32 SystemClock )
```

4.7 include/sched_config.h File Reference

This file is to be used as an implementation of the scheduler configuration.

Data Structures

- struct SCHED_systask_info_t

Macros

- #define SCHED_MAX_TASKS 3
- #define SCHED_TICK_TIME 2
- #define SCHED_AHB_CLK 8000000

4.7.1 Detailed Description

This file is to be used as an implementation of the scheduler configuration.

Author

MSN

Date

Mar 31, 2020

4.7.2 Macro Definition Documentation

4.7.2.1 SCHED_AHB_CLK

```
#define SCHED_AHB_CLK 8000000
```

Definition at line 19 of file sched_config.h.

4.7.2.2 SCHED_MAX_TASKS

```
#define SCHED_MAX_TASKS 3
```

Definition at line 15 of file sched_config.h.

4.7.2.3 SCHED_TICK_TIME

```
#define SCHED_TICK_TIME 2
```

Definition at line 17 of file sched_config.h.

4.8 include/sched_interface.h File Reference

This file is to be used as an implementation of the scheduler driver.

Data Structures

- struct SCHED_task_t

Typedefs

- typedef void(∗ SCHED_task_runnable_t) (void)

Functions

- void SCHED_Init (void)

  Initialize scheduler.
- void SCHED_Start (void)

  Start the scheduler.

4.8.1 Detailed Description

This file is to be used as an implementation of the scheduler driver.

Author

MSN

Date

Mar 31, 2020

4.8.2 Typedef Documentation

4.8.2.1 SCHED_task_runnable_t

```
typedef void(* SCHED_task_runnable_t) (void)
```

Definition at line 13 of file sched_interface.h.

4.8.3 Function Documentation

4.8.3.1 SCHED_Init()

```
void SCHED_Init (
            void  )
```

Initialize scheduler.

Parameters

void	

Returns

void

4.8.3.2 SCHED_Start()

```
void SCHED_Start (
            void  )
```

Start the scheduler.

Parameters

void	

Returns

void

4.9 include/STD_TYPES.h File Reference

This file is to be used as an implementation of the standard types.

Macros

- #define E_OK (Std_ReturnType)0
- #define E_NOT_OK (Std_ReturnType)1
- #define E_BUSY (Std_ReturnType)2
- #define NULL ((void *)0)

Typedefs

- typedef unsigned char u8
- typedef unsigned short int u16
- typedef unsigned long int u32
- typedef unsigned long long u64
- typedef signed char s8
- typedef signed short int s16
- typedef signed long int s32
- typedef signed long long s64
- typedef float f32
- typedef double f64
- typedef long double f96
- typedef u8 Std_ReturnType

4.9.1 Detailed Description

This file is to be used as an implementation of the standard types.

Author

MSN

Date

Mar 31, 2020

4.9.2 Macro Definition Documentation

4.9.2.1 E_BUSY

```
#define E_BUSY (Std_ReturnType)2
```

Definition at line 33 of file STD_TYPES.h.

4.9.2.2 E_NOT_OK

```
#define E_NOT_OK (Std_ReturnType)1
```

Definition at line 32 of file STD_TYPES.h.

4.9.2.3 E_OK

```
#define E_OK (Std_ReturnType)0
```

Definition at line 31 of file STD_TYPES.h.

4.9.2.4 NULL

```
#define NULL ((void *)0)
```

Definition at line 35 of file STD_TYPES.h.

4.9.3 Typedef Documentation

4.9.3.1 f32

```
typedef float f32
```

Definition at line 23 of file STD_TYPES.h.

4.9.3.2 f64

```
typedef double f64
```

Definition at line 24 of file STD_TYPES.h.

4.9.3.3 f96

```
typedef long double f96
```

Definition at line 25 of file STD_TYPES.h.

4.9.3.4 s16

```
typedef signed short int s16
```

Definition at line 19 of file STD_TYPES.h.

4.9.3.5 s32

```
typedef signed long int s32
```

Definition at line 20 of file STD_TYPES.h.

4.9.3.6 s64

```
typedef signed long long s64
```

Definition at line 21 of file STD_TYPES.h.

4.9.3.7 s8

```
typedef signed char s8
```

Definition at line 18 of file STD_TYPES.h.

4.9.3.8 Std_ReturnType

```
typedef u8 Std_ReturnType
```

Definition at line 28 of file STD_TYPES.h.

4.9.3.9 u16

```
typedef unsigned short int u16
```

Definition at line 14 of file STD_TYPES.h.

4.9.3.10 u32

```
typedef unsigned long int u32
```

Definition at line 15 of file STD_TYPES.h.

4.9.3.11 u64

```
typedef unsigned long long u64
```

Definition at line 16 of file STD_TYPES.h.

4.9.3.12 u8

```
typedef unsigned char u8
```

Definition at line 13 of file STD_TYPES.h.

4.10 include/stm32f10x_conf.h File Reference

```
#include "stm32f10x_adc.h"
#include "stm32f10x_bkp.h"
#include "stm32f10x_can.h"
#include "stm32f10x_cec.h"
#include "stm32f10x_crc.h"
#include "stm32f10x_dac.h"
#include "stm32f10x_dbgmcu.h"
#include "stm32f10x_dma.h"
#include "stm32f10x_exti.h"
#include "stm32f10x_flash.h"
#include "stm32f10x_fsmc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_i2c.h"
#include "stm32f10x_iwdg.h"
#include "stm32f10x_pwr.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_rtc.h"
#include "stm32f10x_sdio.h"
#include "stm32f10x_spi.h"
#include "stm32f10x_tim.h"
#include "stm32f10x_usart.h"
#include "stm32f10x_wwdg.h"
#include "misc.h"
```

Macros

- #define assert_param(expr) ((void)0)

4.10.1 Macro Definition Documentation

4.10.1.1 assert_param

```
#define assert_param(
              expr ) ((void)0)
```

Definition at line 73 of file stm32f10x_conf.h.

4.11 include/switch_config.h File Reference

This file is to be used as an implementation of the switch configuration.

Data Structures

- struct Switch_cfg_t

Macros

- #define SWITCH_COUNT 7

4.11.1 Detailed Description

This file is to be used as an implementation of the switch configuration.

Author

MSN

Date

Mar 31, 2020

4.11.2 Macro Definition Documentation

4.11.2.1 SWITCH_COUNT

```
#define SWITCH_COUNT 7
```

Definition at line 14 of file switch_config.h.

4.12 include/switch_interface.h File Reference

This file is to be used as an implementation of the switch driver.

Macros

- #define SWITCH_PRESSED 1
- #define SWITCH_UNPRESSED 0
- #define SWITCH_ACTIVE_LOW 0
- #define SWITCH_ACTIVE_HIGH 1

Functions

- void Switch_Init (void)

  Initialize all the switches: pin direction, default pull up/down.
- u8 Switch_GetReading (u8 switchNum)

  Return the state of the switch.

4.12.1 Detailed Description

This file is to be used as an implementation of the switch driver.

Author

MSN

Date

Mar 31, 2020

4.12.2 Macro Definition Documentation

4.12.2.1 SWITCH_ACTIVE_HIGH

```
#define SWITCH_ACTIVE_HIGH 1
```

Definition at line 19 of file switch_interface.h.

4.12.2.2 SWITCH_ACTIVE_LOW

```
#define SWITCH_ACTIVE_LOW 0
```

Definition at line 18 of file switch_interface.h.

4.12.2.3 SWITCH_PRESSED

```
#define SWITCH_PRESSED 1
```

Definition at line 14 of file switch_interface.h.

4.12.2.4 SWITCH_UNPRESSED

```
#define SWITCH_UNPRESSED 0
```

Definition at line 15 of file switch_interface.h.

4.12.3 Function Documentation

4.12.3.1 Switch_GetReading()

```
u8 Switch_GetReading (
            u8 switchNum )
```

Return the state of the switch.

Parameters

switchNum	Switch number, this should be <= the amount of configured switches

Returns

the state of the switch:
SWITCH_PRESSED,
SWITCH_UNPRESSED

4.12.3.2 Switch_Init()

```
void Switch_Init (
            void  )
```

Initialize all the switches: pin direction, default pull up/down.

Parameters

void	

Returns

void

4.13 include/SYSTICK.h File Reference

This file is to be used as an implementation of the SysTick driver.

Macros

- #define SYSTICK_ENABLE 1U
- #define SYSTICK_DISABLE 0U
- #define SYSTICK_INTERRUPT_ENABLE (u32)0x00000002
- #define CLOCK_PRESCALER_AHB_DIV_8 (u32)0x00000000
- #define CLOCK_PRESCALER_AHB_DIV_1 (u32)0x00000004
- #define CLOCK_FREQUENCY_8_MHZ (u32)8000000UL
- #define CLOCK_FREQUENCY_8_MHZ_DIV8 (u32)1000000UL

Functions

- void SYSTICK_Init (void)

  Initialize SysTick.
- void SYSTICK_Stop (void)

  To stop SysTick.
- void SYSTICK_Start (void)

  To start SysTick.
- void SYSTICK_SetTimers (u32 Timers)

  Function to set tick time (ms)
- Std_ReturnType SYSTICK_SetCallBack (void(∗Copy_SystickCbf_t)(void))

  Function callback to handle call back function.

4.13.1 Detailed Description

This file is to be used as an implementation of the SysTick driver.

Author

MSN

Date

Mar 31, 2020

4.13.2 Macro Definition Documentation

4.13.2.1 CLOCK_FREQUENCY_8_MHZ

```
#define CLOCK_FREQUENCY_8_MHZ (u32)8000000UL
```

Definition at line 22 of file SYSTICK.h.

4.13.2.2 CLOCK_FREQUENCY_8_MHZ_DIV8

```
#define CLOCK_FREQUENCY_8_MHZ_DIV8 (u32)1000000UL
```

Definition at line 23 of file SYSTICK.h.

4.13.2.3 CLOCK_PRESCALER_AHB_DIV_1

```
#define CLOCK_PRESCALER_AHB_DIV_1 (u32)0x00000004
```

Definition at line 20 of file SYSTICK.h.

4.13.2.4 CLOCK_PRESCALER_AHB_DIV_8

```
#define CLOCK_PRESCALER_AHB_DIV_8 (u32)0x00000000
```

Definition at line 19 of file SYSTICK.h.

4.13.2.5 SYSTICK_DISABLE

```
#define SYSTICK_DISABLE 0U
```

Definition at line 15 of file SYSTICK.h.

4.13.2.6 SYSTICK_ENABLE

```
#define SYSTICK_ENABLE 1U
```

Definition at line 14 of file SYSTICK.h.

4.13.2.7 SYSTICK_INTERRUPT_ENABLE

```
#define SYSTICK_INTERRUPT_ENABLE (u32)0x00000002
```

Definition at line 16 of file SYSTICK.h.

4.13.3 Function Documentation

4.13.3.1 SYSTICK_Init()

```
void SYSTICK_Init (
            void  )
```

Initialize SysTick.

Parameters

void	

Returns

void

4.13.3.2 SYSTICK_SetCallBack()

```
Std_ReturnType SYSTICK_SetCallBack (
            void(*)(void) Copy_SystickCbf_t )
```

Function callback to handle call back function.

Parameters

Copy_Systick↩ Cbf_t	(pointer to function)

Returns

E_OK :- if the input argument is correct .
(if the input pointer to function is valid)
E_NOT_OK :- if there's something wrong with the input argument .
(if the input pointer to function is not valid)

4.13.3.3 SYSTICK_SetTimers()

```
void SYSTICK_SetTimers (
             u32 Timers )
```

Function to set tick time (ms)

Parameters

Timer	time in ms

Returns

void

4.13.3.4 SYSTICK_Start()

```
void SYSTICK_Start (
             void  )
```

To start SysTick.

Parameters

void	

Returns

void

4.13.3.5 SYSTICK_Stop()

```
void SYSTICK_Stop (
             void  )
```

To stop SysTick.

Parameters

void	

Returns

void

4.14 include/SYSTICK_CFG.h File Reference

This file is to be used as an implementation of the SysTick configuration.

Macros

- #define CLOCK_PRESCALER CLOCK_PRESCALER_AHB_DIV_8
- #define CLOCK_FREQUENCY (CLOCK_FREQUENCY_8_MHZ_DIV8)

4.14.1 Detailed Description

This file is to be used as an implementation of the SysTick configuration.

Author

MSN

Date

Mar 31, 2020

4.14.2 Macro Definition Documentation

4.14.2.1 CLOCK_FREQUENCY

```
#define CLOCK_FREQUENCY (CLOCK_FREQUENCY_8_MHZ_DIV8)
```

Definition at line 16 of file SYSTICK_CFG.h.

4.14.2.2 CLOCK_PRESCALER

```
#define CLOCK_PRESCALER CLOCK_PRESCALER_AHB_DIV_8
```

Definition at line 14 of file SYSTICK_CFG.h.

4.15 include/UART.h File Reference

Macros

- #define UART_PERI_1 0
- #define UART_PERI_2 1
- #define UART_PERI_3 2
- #define UART_STATE_CONTROL_ON 0x00002000
- #define UART_STATE_CONTROL_OFF 0x00000000
- #define UART_DATA_LENGTH_9BIT 0x00001000
- #define UART_DATA_LENGTH_8BIT 0x00000000
- #define UART_PARITY_STATE_ON 0x00000400
- #define UART_PARITY_STATE_OFF 0x00000000
- #define UART_PARITY_SELECT_EVEN 0x00000000
- #define UART_PARITY_SELECT_ODD 0x00000200
- #define UART_PARITY_CONTROL_INT_ON 0x00000100
- #define UART_PARITY_CONTROL_INT_OFF 0x00000000
- #define UART_TX_BUFFER_EMPTY_CONTROL_INT_ON 0x00000080
- #define UART_TX_BUFFER_EMPTY_CONTROL_INT_OFF 0x00000000
- #define UART_TARNSMISSION_COMPLETE_CONTROL_INT_ON 0x00000040
- #define UART_TARNSMISSION_COMPLETE_CONTROL_INT_OFF 0x00000000
- #define UART_RX_BUFFER_FULL_CONTROL_INT_ON 0x00000020
- #define UART_RX_BUFFER_FULL_CONTROL_INT_OFF 0x00000000
- #define UART_TANSMITTER_STATE_ON 0x00000008
- #define UART_TANSMITTER_STATE_OFF 0x00000000
- #define UART_RECEIVER_STATE_ON 0x00000004
- #define UART_RECEIVER_STATE_OFF 0x00000000
- #define UART_LIN_STATE_CONTROL_ON 0x00004000
- #define UART_LIN_STATE_CONTROL_OFF 0x00000000
- #define UART_STOP_BITS_1 0x00000000
- #define UART_STOP_BITS_2 0x00002000
- #define UART_LIN_BREAK_DETECTION_CONTROL_INT_ON 0x00000040
- #define UART_LIN_BREAK_DETECTION_CONTROL_INT_OFF 0x00000000
- #define UART_LIN_BREAK_DETECTION_LENGTH_10 0x00000000
- #define UART_LIN_BREAK_DETECTION_LENGTH_11 0x00000020
- #define UART_DMA_TRANSMITTER_STATE_ON 0x00000080
- #define UART_DMA_TRANSMITTER_STATE_OFF 0x00000000
- #define UART_DMA_RECEIVER_STATE_ON 0x00000040
- #define UART_DMA_RECEIVER_STATE_OFF 0x00000000
- #define UART_DMA_ERROR_CONTROL_INT_ON 0x00000001
- #define UART_DMA_ERROR_CONTROL_INT_OFF 0x00000000

Typedefs

- typedef void(* UART_TxBufferEmptyCallback_t) (void)
- typedef void(* UART_RxBufferFullCallback_t) (u16)
- typedef void(* UART_ParityErrorCallback_t) (void)
- typedef void(* UART_TranssmisonCompleteCallback_t) (void)
- typedef void(* UART_LINBreakCallback_t) (void)
- typedef void(* UART_DMAErrorCallback_t) (void)
- typedef void(* UART_ErrorFrameCallback_t) (void)
- typedef void(* UART_NoiseErrorCallback_t) (void)
- typedef void(* UART_OverRunErrorCallback_t) (void)

Functions

- void UART_Init (u8 peri, u32 baudRate)
- u8 UART_GetIsLIN_BreakDetection (u8 peri)
- u8 UART_GetIsTxBufferEmpty (u8 peri)
- u8 UART_GetIsTansmitComplete (u8 peri)
- u8 UART_GetIsRxBufferFull (u8 peri)
- u8 UART_GetIsOverRunError (u8 peri)
- u8 UART_GetINoiseError (u8 peri)
- u8 UART_GetIsFrameError (u8 peri)
- u8 UART_GetIsParityError (u8 peri)
- u16 UART_GetData (u8 peri)
- void UART_ConfigData (u8 peri, u16 data)
- void UART_ConfigBaudRate (u8 peri, u16 baud_rate)
- void UART_ConfigParityType (u8 peri, u32 parity_select)
- void UART_ConfigStopBits (u8 peri, u32 stop_bits)
- void UART_ConfigLINBreakDetectionLength (u8 peri, u32 lin_break_detection_length)
- void UART_ConfigTxBufferEmptyCallback (u8 peri, UART_TxBufferEmptyCallback_t ptr)
- void UART_ConfigRxBufferFullCallback (u8 peri, UART_RxBufferFullCallback_t ptr)
- void UART_ConfigParityErrorCallback (u8 peri, UART_ParityErrorCallback_t ptr)
- void UART_ConfigTransmissionCompleteCallback (u8 peri, UART_TranssmisonCompleteCallback_t ptr)
- void UART_ConfigLINBreakCallback (u8 peri, UART_LINBreakCallback_t ptr)
- void UART_ConfigNoiseErrorCallBack (u8 peri, UART_NoiseErrorCallback_t ptr)
- void UART_ConfigOverRunErrorCallBack (u8 peri, UART_OverRunErrorCallback_t ptr)
- void UART_ConfigFrameErrorCallBack (u8 peri, UART_ErrorFrameCallback_t ptr)
- void UART_ControlState (u8 peri, u32 state_control)
- void UART_ControlDataLength (u8 peri, u32 data_length)
- void UART_ControlParityState (u8 peri, u32 parity_state)
- void UART_ControlParityErrorINT (u8 peri, u32 parity_control_INT)
- void UART_ControlTxBufferEmptyINT (u8 peri, u32 tx_buffer_empty_control_INT)
- void UART_ControlTransmissionCompleteINT (u8 peri, u32 transmission_complete_control_INT)
- void UART_ControlRxBufferFullINT (u8 peri, u32 rx_buffer_full_control_INT)
- void UART_ControlLINBreakDetectionINT (u8 peri, u32 lin_break_detection_control_INT)
- void UART_ControlDMAErrorINT (u8 peri, u32 dma_error_control_INT)
- void UART_ControlTransmitterState (u8 peri, u32 transmit_state)
- void UART_ControlReceiverState (u8 peri, u32 receive_state)
- void UART_ControlLINState (u8 peri, u32 lin_state_control)
- void UART_ControlDMATransmitterState (u8 peri, u32 dma_transmitter_state)
- void UART_ControlDMAReceiverState (u8 peri, u32 dma_receivier_state)
- void UART_SetBreakCharacter (u8 peri)

4.15.1 Macro Definition Documentation

4.15.1.1 UART_DATA_LENGTH_8BIT

```
#define UART_DATA_LENGTH_8BIT 0x00000000
```

Definition at line 13 of file UART.h.

4.15.1.2 UART_DATA_LENGTH_9BIT

```
#define UART_DATA_LENGTH_9BIT 0x00001000
```

Definition at line 12 of file UART.h.

4.15.1.3 UART_DMA_ERROR_CONTROL_INT_OFF

```
#define UART_DMA_ERROR_CONTROL_INT_OFF 0x00000000
```

Definition at line 58 of file UART.h.

4.15.1.4 UART_DMA_ERROR_CONTROL_INT_ON

```
#define UART_DMA_ERROR_CONTROL_INT_ON 0x00000001
```

Definition at line 57 of file UART.h.

4.15.1.5 UART_DMA_RECEIVER_STATE_OFF

```
#define UART_DMA_RECEIVER_STATE_OFF 0x00000000
```

Definition at line 55 of file UART.h.

4.15.1.6 UART_DMA_RECEIVER_STATE_ON

```
#define UART_DMA_RECEIVER_STATE_ON 0x00000040
```

Definition at line 54 of file UART.h.

4.15.1.7 UART_DMA_TRANSMITTER_STATE_OFF

```
#define UART_DMA_TRANSMITTER_STATE_OFF 0x00000000
```

Definition at line 52 of file UART.h.

4.15.1.8 UART_DMA_TRANSMITTER_STATE_ON

```
#define UART_DMA_TRANSMITTER_STATE_ON 0x00000080
```

Definition at line 51 of file UART.h.

4.15.1.9 UART_LIN_BREAK_DETECTION_CONTROL_INT_OFF

```
#define UART_LIN_BREAK_DETECTION_CONTROL_INT_OFF 0x00000000
```

Definition at line 46 of file UART.h.

4.15.1.10 UART_LIN_BREAK_DETECTION_CONTROL_INT_ON

```
#define UART_LIN_BREAK_DETECTION_CONTROL_INT_ON 0x00000040
```

Definition at line 45 of file UART.h.

4.15.1.11 UART_LIN_BREAK_DETECTION_LENGTH_10

```
#define UART_LIN_BREAK_DETECTION_LENGTH_10 0x00000000
```

Definition at line 48 of file UART.h.

4.15.1.12 UART_LIN_BREAK_DETECTION_LENGTH_11

```
#define UART_LIN_BREAK_DETECTION_LENGTH_11 0x00000020
```

Definition at line 49 of file UART.h.

4.15.1.13 UART_LIN_STATE_CONTROL_OFF

```
#define UART_LIN_STATE_CONTROL_OFF 0x00000000
```

Definition at line 40 of file UART.h.

4.15.1.14 UART_LIN_STATE_CONTROL_ON

```
#define UART_LIN_STATE_CONTROL_ON 0x00004000
```

Definition at line 39 of file UART.h.

4.15.1.15 UART_PARITY_CONTROL_INT_OFF

```
#define UART_PARITY_CONTROL_INT_OFF 0x00000000
```

Definition at line 22 of file UART.h.

4.15.1.16 UART_PARITY_CONTROL_INT_ON

```
#define UART_PARITY_CONTROL_INT_ON 0x00000100
```

Definition at line 21 of file UART.h.

4.15.1.17 UART_PARITY_SELECT_EVEN

```
#define UART_PARITY_SELECT_EVEN 0x00000000
```

Definition at line 18 of file UART.h.

4.15.1.18 UART_PARITY_SELECT_ODD

```
#define UART_PARITY_SELECT_ODD 0x00000200
```

Definition at line 19 of file UART.h.

4.15.1.19 UART_PARITY_STATE_OFF

```
#define UART_PARITY_STATE_OFF 0x00000000
```

Definition at line 16 of file UART.h.

4.15.1.20 UART_PARITY_STATE_ON

```
#define UART_PARITY_STATE_ON 0x00000400
```

Definition at line 15 of file UART.h.

4.15.1.21 UART_PERI_1

```
#define UART_PERI_1 0
```

Definition at line 5 of file UART.h.

4.15.1.22 UART_PERI_2

```
#define UART_PERI_2 1
```

Definition at line 6 of file UART.h.

4.15.1.23 UART_PERI_3

```
#define UART_PERI_3 2
```

Definition at line 7 of file UART.h.

4.15.1.24 UART_RECEIVER_STATE_OFF

```
#define UART_RECEIVER_STATE_OFF 0x00000000
```

Definition at line 37 of file UART.h.

4.15.1.25 UART_RECEIVER_STATE_ON

```
#define UART_RECEIVER_STATE_ON 0x00000004
```

Definition at line 36 of file UART.h.

4.15.1.26 UART_RX_BUFFER_FULL_CONTROL_INT_OFF

```
#define UART_RX_BUFFER_FULL_CONTROL_INT_OFF 0x00000000
```

Definition at line 31 of file UART.h.

4.15.1.27 UART_RX_BUFFER_FULL_CONTROL_INT_ON

```
#define UART_RX_BUFFER_FULL_CONTROL_INT_ON 0x00000020
```

Definition at line 30 of file UART.h.

4.15.1.28 UART_STATE_CONTROL_OFF

```
#define UART_STATE_CONTROL_OFF 0x00000000
```

Definition at line 10 of file UART.h.

4.15.1.29 UART_STATE_CONTROL_ON

```
#define UART_STATE_CONTROL_ON 0x00002000
```

Definition at line 9 of file UART.h.

4.15.1.30 UART_STOP_BITS_1

```
#define UART_STOP_BITS_1 0x00000000
```

Definition at line 42 of file UART.h.

4.15.1.31 UART_STOP_BITS_2

```
#define UART_STOP_BITS_2 0x00002000
```

Definition at line 43 of file UART.h.

4.15.1.32 UART_TANSMITTER_STATE_OFF

```
#define UART_TANSMITTER_STATE_OFF 0x00000000
```

Definition at line 34 of file UART.h.

4.15.1.33 UART_TANSMITTER_STATE_ON

```
#define UART_TANSMITTER_STATE_ON 0x00000008
```

Definition at line 33 of file UART.h.

4.15.1.34 UART_TARNSMISSION_COMPLETE_CONTROL_INT_OFF

```
#define UART_TARNSMISSION_COMPLETE_CONTROL_INT_OFF 0x00000000
```

Definition at line 28 of file UART.h.

4.15.1.35 UART_TARNSMISSION_COMPLETE_CONTROL_INT_ON

```
#define UART_TARNSMISSION_COMPLETE_CONTROL_INT_ON 0x00000040
```

Definition at line 27 of file UART.h.

4.15.1.36 UART_TX_BUFFER_EMPTY_CONTROL_INT_OFF

```
#define UART_TX_BUFFER_EMPTY_CONTROL_INT_OFF 0x00000000
```

Definition at line 25 of file UART.h.

4.15.1.37 UART_TX_BUFFER_EMPTY_CONTROL_INT_ON

```
#define UART_TX_BUFFER_EMPTY_CONTROL_INT_ON 0x00000080
```

Definition at line 24 of file UART.h.

4.15.2 Typedef Documentation

4.15.2.1 UART_DMAErrorCallback_t

```
typedef void(* UART_DMAErrorCallback_t) (void)
```

Definition at line 64 of file UART.h.

4.15.2.2 UART_ErrorFrameCallback_t

```
typedef void(* UART_ErrorFrameCallback_t) (void)
```

Definition at line 65 of file UART.h.

4.15.2.3 UART_LINBreakCallback_t

```
typedef void(* UART_LINBreakCallback_t) (void)
```

Definition at line 63 of file UART.h.

4.15.2.4 UART_NoiseErrorCallback_t

```
typedef void(* UART_NoiseErrorCallback_t) (void)
```

Definition at line 66 of file UART.h.

4.15.2.5 UART_OverRunErrorCallback_t

```
typedef void(* UART_OverRunErrorCallback_t) (void)
```

Definition at line 67 of file UART.h.

4.15.2.6 UART_ParityErrorCallback_t

```
typedef void(* UART_ParityErrorCallback_t) (void)
```

Definition at line 61 of file UART.h.

4.15.2.7 UART_RxBufferFullCallback_t

```
typedef void(* UART_RxBufferFullCallback_t) (u16)
```

Definition at line 60 of file UART.h.

4.15.2.8 UART_TranssmisonCompleteCallback_t

```
typedef void(* UART_TranssmisonCompleteCallback_t) (void)
```

Definition at line 62 of file UART.h.

4.15.2.9 UART_TxBufferEmptyCallback_t

```
typedef void(* UART_TxBufferEmptyCallback_t) (void)
```

Definition at line 59 of file UART.h.

4.15.3 Function Documentation

4.15.3.1 UART_ConfigBaudRate()

```
void UART_ConfigBaudRate (
            u8 peri,
            u16 baud_rate )
```

4.15.3.2 UART_ConfigData()

```
void UART_ConfigData (
            u8 peri,
            u16 data )
```

4.15.3.3 UART_ConfigFrameErrorCallBack()

```
void UART_ConfigFrameErrorCallBack (
            u8 peri,
            UART_ErrorFrameCallback_t ptr )
```

4.15.3.4 UART_ConfigLINBreakCallback()

```
void UART_ConfigLINBreakCallback (
            u8 peri,
            UART_LINBreakCallback_t ptr )
```

4.15.3.5 UART_ConfigLINBreakDetectionLength()

```
void UART_ConfigLINBreakDetectionLength (
            u8 peri,
            u32 lin_break_detection_length )
```

4.15.3.6 UART_ConfigNoiseErrorCallBack()

```
void UART_ConfigNoiseErrorCallBack (
            u8 peri,
            UART_NoiseErrorCallback_t ptr )
```

4.15.3.7 UART_ConfigOverRunErrorCallBack()

```
void UART_ConfigOverRunErrorCallBack (
            u8 peri,
            UART_OverRunErrorCallback_t ptr )
```

4.15.3.8 UART_ConfigParityErrorCallback()

```
void UART_ConfigParityErrorCallback (
            u8 peri,
            UART_ParityErrorCallback_t ptr )
```

4.15.3.9 UART_ConfigParityType()

```
void UART_ConfigParityType (
            u8 peri,
            u32 parity_select )
```

4.15.3.10 UART_ConfigRxBufferFullCallback()

```
void UART_ConfigRxBufferFullCallback (
            u8 peri,
            UART_RxBufferFullCallback_t ptr )
```

4.15.3.11 UART_ConfigStopBits()

```
void UART_ConfigStopBits (
            u8 peri,
            u32 stop_bits )
```

4.15.3.12 UART_ConfigTransmissionCompleteCallback()

```
void UART_ConfigTransmissionCompleteCallback (
            u8 peri,
            UART_TranssmisonCompleteCallback_t ptr )
```

4.15.3.13 UART_ConfigTxBufferEmptyCallback()

```
void UART_ConfigTxBufferEmptyCallback (
            u8 peri,
            UART_TxBufferEmptyCallback_t ptr )
```

4.15.3.14 UART_ControlDataLength()

```
void UART_ControlDataLength (
            u8 peri,
            u32 data_length )
```

4.15.3.15 UART_ControlDMAErrorINT()

```
void UART_ControlDMAErrorINT (
            u8 peri,
            u32 dma_error_control_INT )
```

4.15.3.16 UART_ControlDMAReceiverState()

```
void UART_ControlDMAReceiverState (
            u8 peri,
            u32 dma_receivier_state )
```

4.15.3.17 UART_ControlDMATransmitterState()

```
void UART_ControlDMATransmitterState (
            u8 peri,
            u32 dma_transmitter_state )
```

4.15.3.18 UART_ControlLINBreakDetectionINT()

```
void UART_ControlLINBreakDetectionINT (
            u8 peri,
            u32 lin_break_detection_control_INT )
```

4.15.3.19 UART_ControlLINState()

```
void UART_ControlLINState (
            u8 peri,
            u32 lin_state_control )
```

4.15.3.20 UART_ControlParityErrorINT()

```
void UART_ControlParityErrorINT (
            u8 peri,
            u32 parity_control_INT )
```

4.15.3.21 UART_ControlParityState()

```
void UART_ControlParityState (
            u8 peri,
            u32 parity_state )
```

4.15.3.22 UART_ControlReceiverState()

```
void UART_ControlReceiverState (
            u8 peri,
            u32 receive_state )
```

4.15.3.23 UART_ControlRxBufferFullINT()

```
void UART_ControlRxBufferFullINT (
            u8 peri,
            u32 rx_buffer_full_control_INT )
```

4.15.3.24 UART_ControlState()

```
void UART_ControlState (
            u8 peri,
            u32 state_control )
```

4.15.3.25 UART_ControlTransmissionCompleteINT()

```
void UART_ControlTransmissionCompleteINT (
            u8 peri,
            u32 transmission_complete_control_INT )
```

4.15.3.26 UART_ControlTransmitterState()

```
void UART_ControlTransmitterState (
            u8 peri,
            u32 transmit_state )
```

4.15.3.27 UART_ControlTxBufferEmptyINT()

```
void UART_ControlTxBufferEmptyINT (
            u8 peri,
            u32 tx_buffer_empty_control_INT )
```

4.15.3.28 UART_GetData()

```
u16 UART_GetData (
            u8 peri )
```

4.15.3.29 UART_GetINoiseError()

```
u8 UART_GetINoiseError (
            u8 peri )
```

4.15.3.30 UART_GetIsFrameError()

```
u8 UART_GetIsFrameError (
            u8 peri )
```

4.15.3.31 UART_GetIsLIN_BreakDetection()

```
u8 UART_GetIsLIN_BreakDetection (
            u8 peri )
```

4.15.3.32 UART_GetIsOverRunError()

```
u8 UART_GetIsOverRunError (
            u8 peri )
```

4.15.3.33 UART_GetIsParityError()

```
u8 UART_GetIsParityError (
            u8 peri )
```

4.15.3.34 UART_GetIsRxBufferFull()

```
u8 UART_GetIsRxBufferFull (
            u8 peri )
```

4.15.3.35 UART_GetIsTansmitComplete()

```
u8 UART_GetIsTansmitComplete (
            u8 peri )
```

4.15.3.36 UART_GetIsTxBufferEmpty()

```
u8 UART_GetIsTxBufferEmpty (
            u8 peri )
```

4.15.3.37 UART_Init()

```
void UART_Init (
            u8 peri,
            u32 baudRate )
```

4.15.3.38 UART_SetBreakCharacter()

```
void UART_SetBreakCharacter (
            u8 peri )
```

4.16 include/UART_cfg.h File Reference

Macros

- #define UART_APB1_BUS_FREQ 8000000
- #define UART_APB2_BUS_FREQ 8000000

4.16.1 Macro Definition Documentation

4.16.1.1 UART_APB1_BUS_FREQ

```
#define UART_APB1_BUS_FREQ 8000000
```

Definition at line 5 of file UART_cfg.h.

4.16.1.2 UART_APB2_BUS_FREQ

```
#define UART_APB2_BUS_FREQ 8000000
```

Definition at line 6 of file UART_cfg.h.

Index