GPIO Pin Subsystem

Collaboration diagram for GPIO Pin Subsystem:

Enumerations

enum	**gpio_direction** { **GPIO_INPUT**, **GPIO_OUTPUT** }
enum	**gpio_setting** { **GPIO_OPEN**, **GPIO_PULL_UP**, **GPIO_PULL_DOWN** }
enum	**gpio_interrupt** { **GPIO_FALLING**, **GPIO_RISING**, **GPIO_EITHER** }

Functions

struct gpio_pin *	**gpioPin_init** (struct gpio *gpio, uint32_t pin, enum **gpio_direction** dir, enum **gpio_setting** setting)
int32_t	**gpioPin_deinit** (struct gpio_pin *pin)
int32_t	**gpioPin_enableInterrupt** (struct gpio_pin *pin)
int32_t	**gpioPin_disableInterrupt** (struct gpio_pin *pin)
int32_t	**gpioPin_setCallback** (struct gpio_pin *pin, bool(*callback)(struct gpio_pin *pin, uint32_t pinID, void *data), void *data, enum **gpio_interrupt** inter)
int32_t	**gpioPin_setDirection** (struct gpio_pin *pin, enum **gpio_direction** dir)
int32_t	**gpioPin_setSetting** (struct gpio_pin *pin, enum **gpio_setting** setting)
int32_t	**gpioPin_SchmittTrigger** (struct gpio_pin *pin, bool schmitt)
int32_t	**gpioPin_setValue** (struct gpio_pin *pin, bool value)
int32_t	**gpioPin_setPin** (struct gpio_pin *pin)
int32_t	**gpioPin_clearPin** (struct gpio_pin *pin)
int32_t	**gpioPin_togglePin** (struct gpio_pin *pin)
bool	**gpioPin_getValue** (struct gpio_pin *pin)

Detailed Description

```
#include <gpio.h>
```

This is the Pin GPIO Subsystem for controlling one GPIO pin of a SOC.

Enumeration Type Documentation

◆ gpio_direction

enum **gpio_direction**

GPIO Direction

Enumerator	
GPIO_INPUT	Input
GPIO_OUTPUT	Output

◆ gpio_interrupt

enum **gpio_interrupt**

GPIO as Interrupt

Enumerator	
GPIO_FALLING	Call a Interrupt while Falling Edge
GPIO_RISING	Call a Interrupt while Falling Rising
GPIO_EITHER	Call a Interrupt while Falling and Rising Edge

◆ gpio_setting

enum **gpio_setting**

GPIO Setting

Enumerator	
GPIO_OPEN	Configure Pin without Pull down or Pull up
GPIO_PULL_UP	Configure Pin with Pull Up
GPIO_PULL_DOWN	Configure Pin with Pull Down

Function Documentation

◆ gpioPin_clearPin()

int32_t gpioPin_clearPin (struct gpio_pin * pin)

Set low on Pin

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok

◆ gpioPin_deinit()

int32_t gpioPin_deinit (struct gpio_pin * pin)

Deinit Pin

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok

◆ gpioPin_disableInterrupt()

int32_t gpioPin_disableInterrupt (struct gpio_pin * pin)

Disable Interrupt on pin

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok

◆ gpioPin_enableInterrupt()

int32_t gpioPin_enableInterrupt (struct gpio_pin * pin)

Enable Interrupt on pin

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok

◆ gpioPin_getValue()

bool gpioPin_getValue (struct gpio_pin * pin)

Get Value

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok

◆ gpioPin_init()

```
struct gpio_pin* gpioPin_init ( struct gpio *          gpio,
                                uint32_t              pin,
                                enum gpio_direction   dir,
                                enum gpio_setting     setting
                              )
```

Init one GPIO Pin

Parameters

gpio	GPIO Handle
pin	Pin GPIO pin
dir	Direction
setting	Pin Settings

Returns

GPIO Pin Handle or NULL on error

◆ gpioPin_SchmittTrigger()

```
int32_t gpioPin_SchmittTrigger ( struct gpio_pin *  pin,
                                 bool               schmitt
                               )
```

Set Schmitt Trigger

Parameters

pin	GPIO Pin Handle
schmitt	true = activate Schmitt Trigger false = deactivate Schmitt Trigger

Returns

-1 on Error 0 on ok

◆ gpioPin_setCallback()

```
int32_t gpioPin_setCallback ( struct gpio_pin *                                  pin,
                              bool(*)(struct gpio_pin *pin, uint32_t pinID, void *data)  callback,
                              void *                                             data,
                              enum gpio_interrupt                                inter
                            )
```

Set Interrupt Callback

Parameters

pin	GPIO Pin Handle
callback	Callback
data	Data transmitted to Callback
inter	Interrupt Setting

Returns

-1 on Error 0 on ok

◆ gpioPin_setDirection()

```
int32_t gpioPin_setDirection ( struct gpio_pin *      pin,
                               enum gpio_direction    dir
                             )
```

Set Direction

Parameters

pin	GPIO Pin Handle
dir	Direction

Returns

-1 on Error 0 on ok

◆ gpioPin_setPin()

int32_t gpioPin_setPin (struct gpio_pin * pin)

Set High on Pin

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok

◆ gpioPin_setSetting()

int32_t gpioPin_setSetting (struct gpio_pin * pin,
enum **gpio_setting** setting
)

Set Setting

Parameters

pin GPIO Pin Handle

setting GPIO Pin Setting

Returns

-1 on Error 0 on ok

◆ gpioPin_setValue()

int32_t gpioPin_setValue (struct gpio_pin * pin,
bool value
)

Set Value of Pin

Parameters

pin GPIO Pin Handle

value true = high false = low

Returns

-1 on Error 0 on ok

◆ gpioPin_togglePin()

int32_t gpioPin_togglePin (struct gpio_pin * pin)

Toggle Pin

Parameters

pin GPIO Pin Handle

Returns

-1 on Error 0 on ok