

PIC32 SDK

Generated by Doxygen 1.8.17

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 drivers/cpu.c File Reference	3
2.1.1 Function Documentation	3
2.1.1.1 cpu_get_peripheral_clock()	3
2.2 drivers/gpio.c File Reference	4
2.2.1 Function Documentation	4
2.2.1.1 gpio_lock_pps()	4
2.2.1.2 gpio_read()	4
2.2.1.3 gpio_set_function()	5
2.2.1.4 gpio_set_mode()	5
2.2.1.5 gpio_unlock_pps()	5
2.2.1.6 gpio_write()	5
2.3 drivers/uart.c File Reference	5
2.3.1 Function Documentation	6
2.3.1.1 uart_close()	6
2.3.1.2 uart_open()	6
2.3.1.3 uart_set_baud()	6
2.3.1.4 uart_set_format()	7
2.3.1.5 uart_set_rx_pin()	7
2.3.1.6 uart_set_tx_pin()	8
Index	9

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

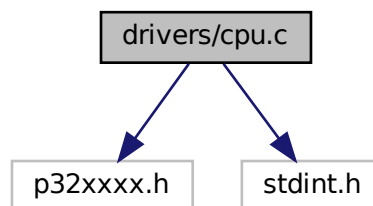
drivers/ cpu.c	3
drivers/ gpio.c	4
drivers/ uart.c	5

Chapter 2

File Documentation

2.1 drivers/cpu.c File Reference

```
#include <p32xxxx.h>
#include <stdint.h>
Include dependency graph for cpu.c:
```



Functions

- `uint32_t` [cpu_get_peripheral_clock](#) ()

2.1.1 Function Documentation

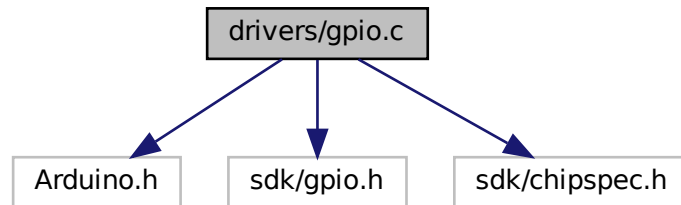
2.1.1.1 `cpu_get_peripheral_clock()`

```
uint32_t cpu_get_peripheral_clock ( )
```

2.2 drivers/gpio.c File Reference

```
#include <Arduino.h>
#include "sdk/gpio.h"
#include "sdk/chipspec.h"
```

Include dependency graph for gpio.c:



Functions

- void [gpio_set_mode](#) (uint8_t pin, uint8_t mode)
- uint8_t [gpio_read](#) (uint8_t pin)
- void [gpio_write](#) (uint8_t pin, uint8_t val)
- void [gpio_set_function](#) (uint8_t pin, uint8_t function)
- void [gpio_unlock_pps](#) ()
- void [gpio_lock_pps](#) ()

2.2.1 Function Documentation

2.2.1.1 gpio_lock_pps()

```
void gpio_lock_pps ( )
```

2.2.1.2 gpio_read()

```
uint8_t gpio_read (
    uint8_t pin )
```


2.2.1.3 gpio_set_function()

```
void gpio_set_function (
    uint8_t pin,
    uint8_t function )
```

2.2.1.4 gpio_set_mode()

```
void gpio_set_mode (
    uint8_t pin,
    uint8_t mode )
```

2.2.1.5 gpio_unlock_pps()

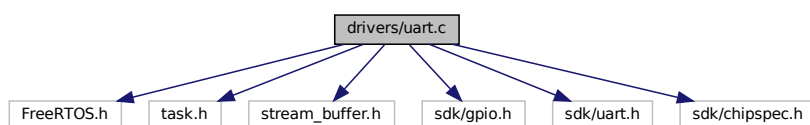
```
void gpio_unlock_pps ( )
```

2.2.1.6 gpio_write()

```
void gpio_write (
    uint8_t pin,
    uint8_t val )
```

2.3 drivers/uart.c File Reference

```
#include "FreeRTOS.h"
#include "task.h"
#include "stream_buffer.h"
#include "sdk/gpio.h"
#include "sdk/uart.h"
#include "sdk/chipspec.h"
Include dependency graph for uart.c:
```



Classes

- struct **uartControlDataStruct**

Functions

- int [uart_set_tx_pin](#) (uint8_t uart, uint8_t pin)
- int [uart_set_rx_pin](#) (uint8_t uart, uint8_t pin)
- int [uart_set_baud](#) (uint8_t uart, uint32_t baud)
- int [uart_set_format](#) (uint8_t uart, uint8_t format)
- int [uart_open](#) (uint8_t uart, StreamBufferHandle_t txBuffer, StreamBufferHandle_t rxBuffer)
- int [uart_close](#) (uint8_t uart)

2.3.1 Function Documentation

2.3.1.1 [uart_close\(\)](#)

```
int uart_close (
    uint8_t uart )
```

2.3.1.2 [uart_open\(\)](#)

```
int uart_open (
    uint8_t uart,
    StreamBufferHandle_t txBuffer,
    StreamBufferHandle_t rxBuffer )
```

2.3.1.3 [uart_set_baud\(\)](#)

```
int uart_set_baud (
    uint8_t uart,
    uint32_t baud )
```

Configure the baud rate of the selected UART

Parameters

<i>uart</i>	The index of the UART
<i>baud</i>	The baud rate to configure.

Returns

1 on success, 0 on failure

2.3.1.4 uart_set_format()

```
int uart_set_format (
    uint8_t uart,
    uint8_t format )
```

Configure the data format for the selected UART. Formats are specified in convenient macros:

- `uart8N1`
- `uart8N2`
- `uart8O1`
- `uart8O2`
- `uart8E1`
- `uart8E2`
- `uart9N1`
- `uart9N2`

Parameters

<i>uart</i>	The index of the UART
<i>format</i>	The format to use.

Returns

1 on success, 0 on failure

2.3.1.5 uart_set_rx_pin()

```
int uart_set_rx_pin (
    uint8_t uart,
    uint8_t pin )
```

Configure the RX pin of the selected UART through PPS

Parameters

<i>uart</i>	The index of the UART
<i>pin</i>	The index of the pin to assign the RX function to

Returns

1 on success, 0 on failure

2.3.1.6 uart_set_tx_pin()

```
int uart_set_tx_pin (
    uint8_t uart,
    uint8_t pin )
```

Configure the TX pin of the selected UART through PPS

Parameters

<i>uart</i>	The index of the UART
<i>pin</i>	The index of the pin to assign the TX function to

Returns

1 on success, 0 on failure

Index

cpu.c

- `cpu_get_peripheral_clock`, [3](#)
- `cpu_get_peripheral_clock`
 - `cpu.c`, [3](#)

- `drivers/cpu.c`, [3](#)

- `drivers/gpio.c`, [4](#)

- `drivers/uart.c`, [5](#)

gpio.c

- `gpio_lock_pps`, [4](#)
- `gpio_read`, [4](#)
- `gpio_set_function`, [4](#)
- `gpio_set_mode`, [5](#)
- `gpio_unlock_pps`, [5](#)
- `gpio_write`, [5](#)
- `gpio_lock_pps`
 - `gpio.c`, [4](#)
- `gpio_read`
 - `gpio.c`, [4](#)
- `gpio_set_function`
 - `gpio.c`, [4](#)
- `gpio_set_mode`
 - `gpio.c`, [5](#)
- `gpio_unlock_pps`
 - `gpio.c`, [5](#)
- `gpio_write`
 - `gpio.c`, [5](#)

uart.c

- `uart_close`, [6](#)
- `uart_open`, [6](#)
- `uart_set_baud`, [6](#)
- `uart_set_format`, [7](#)
- `uart_set_rx_pin`, [7](#)
- `uart_set_tx_pin`, [8](#)
- `uart_close`
 - `uart.c`, [6](#)
- `uart_open`
 - `uart.c`, [6](#)
- `uart_set_baud`
 - `uart.c`, [6](#)
- `uart_set_format`
 - `uart.c`, [7](#)
- `uart_set_rx_pin`
 - `uart.c`, [7](#)
- `uart_set_tx_pin`
 - `uart.c`, [8](#)