

## ESP32 GPIO Drivers

Generado por Doxygen 1.9.4



<b>1 GPIO DRIVERS ESP32</b>	<b>1</b>
1.1 Instalación	1
1.2 Documentacion	1
<b>2 Índice de estructura de datos</b>	<b>3</b>
2.1 Estructura de datos	3
<b>3 Indice de archivos</b>	<b>5</b>
3.1 Lista de archivos	5
<b>4 Documentación de las estructuras de datos</b>	<b>7</b>
4.1 Referencia de la Estructura GPIO_FUNC_OUT_SEL_GENERIC	7
4.1.1 Descripción detallada	7
4.2 Referencia de la Estructura GPIO_GENERIC_I_0	7
4.2.1 Descripción detallada	7
4.3 Referencia de la Estructura GPIO_GENERIC_I_1	8
4.3.1 Descripción detallada	8
4.4 Referencia de la Estructura GPIO_GENERIC_O_0	8
4.4.1 Descripción detallada	8
4.5 Referencia de la Estructura GPIO_GENERIC_O_1	8
4.5.1 Descripción detallada	9
4.6 Referencia de la Unión GPIO_PIN_GENERIC	9
4.6.1 Descripción detallada	9
4.7 Referencia de la Unión IO_MUX	10
<b>5 Documentación de archivos</b>	<b>11</b>
5.1 Referencia del Archivo /home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio↵ _drivers.h	11
5.1.1 Descripción detallada	11
5.1.2 Documentación de las funciones	12
5.1.2.1 GPIO_INPUT_DISABLE()	12
5.1.2.2 GPIO_INPUT_ENABLE()	12
5.1.2.3 GPIO_INPUT_READ()	12
5.1.2.4 GPIO_INTERRUPTION_SET()	12
5.1.2.5 GPIO_OUTPUT_DISABLE()	13
5.1.2.6 GPIO_OUTPUT_ENABLE()	13
5.1.2.7 GPIO_OUTPUT_SET()	13
5.2 gpio_drivers.h	13
5.3 Referencia del Archivo /home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio↵ _iomatrix.h	14
5.3.1 Descripción detallada	15
5.3.2 Documentación de los 'defines'	15
5.3.2.1 _I	15
5.3.2.2 IO_MUX_GPIO0_REG	15

---

5.3.3 Documentación de las variables	16
5.3.3.1 ioMuxDirections	16
5.4 gpio_iomatrix.h	16
5.5 Referencia del Archivo /home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio↔ _matrix.h	17
5.5.1 Descripción detallada	19
5.5.2 Documentación de los 'defines'	19
5.5.2.1 _I	19
5.5.2.2 APP_CPU	19
5.5.2.3 GPIO_OUT	20
5.5.2.4 IO0	20
5.5.2.5 PERIFERAL_BASE	20
5.6 gpio_matrix.h	20
<b>6 Documentación de ejemplos</b>	<b>23</b>
6.1 main.c	23

# Capítulo 1

## GPIO DRIVERS ESP32

### 1.1. Instalación

En la carpeta de tu proyecto usar el siguiente comando:

```
git clone https://github.com/Javier20060904/DRIVERS\_GPIO\_ESP32.git
```

Después en el archivo CMakeLists.txt añadir lo siguiente:

```
set(EXTRA_COMPONENT_DIRS ./DRIVERS_GPIO_ESP32)
```

### 1.2. Documentacion

En el mismo repositorio existe un [pdf](#) donde se encuentran todas las funciones y su uso.



## Capítulo 2

# Índice de estructura de datos

### 2.1. Estructura de datos

Lista de estructuras con una breve descripción:

GPIO_FUNC_OUT_SEL_GENERIC . . . . .	7
GPIO_GENERIC_I_0 . . . . .	7
GPIO_GENERIC_I_1 . . . . .	8
GPIO_GENERIC_O_0 . . . . .	8
GPIO_GENERIC_O_1 . . . . .	8
GPIO_PIN_GENERIC . . . . .	9
IO_MUX . . . . .	10





## Capítulo 3

# Indice de archivos

### 3.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/ <a href="#">gpio_drivers.h</a>	
Funciones y declaraciones para el manejo del GPIO . . . . .	11
/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/ <a href="#">gpio_iomatrix.h</a>	
Declaraciones de registros de IOMUX . . . . .	14
/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/ <a href="#">gpio_matrix.h</a>	
Declaraciones de registros del GPIO . . . . .	17



## Capítulo 4

# Documentación de las estructuras de datos

### 4.1. Referencia de la Estructura GPIO\_FUNC\_OUT\_SEL\_GENERIC

```
#include <gpio_matrix.h>
```

#### Campos de datos

- `_IO uint32_t REG`

#### 4.1.1. Descripción detallada

Declaracion generica del GPIO\_FUNC\_OUT

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_matrix.h`

### 4.2. Referencia de la Estructura GPIO\_GENERIC\_I\_0

```
#include <gpio_matrix.h>
```

#### Campos de datos

- `_I uint32_t REG_IO`

#### 4.2.1. Descripción detallada

Declaracion generica del registro IN del puerto 0 hasta el 31

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_matrix.h`

### 4.3. Referencia de la Estructura GPIO\_GENERIC\_I\_1

```
#include <gpio_matrix.h>
```

#### Campos de datos

- `uint32_t REG_IO:8`
- `uint32_t RESERVED:24`

#### 4.3.1. Descripción detallada

Declaracion generica del registro IN del puerto 32 hasta el 39

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/gpio\\_matrix.h](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_matrix.h)

### 4.4. Referencia de la Estructura GPIO\_GENERIC\_O\_0

```
#include <gpio_matrix.h>
```

#### Campos de datos

- `_IO uint32_t REG_IO`

#### 4.4.1. Descripción detallada

Declaracion generica del registro OUT del puerto 0 hasta el 31

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/gpio\\_matrix.h](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_matrix.h)

### 4.5. Referencia de la Estructura GPIO\_GENERIC\_O\_1

```
#include <gpio_matrix.h>
```

#### Campos de datos

- `_IO uint32_t REG_IO:8`
- `uint32_t RESERVED:24`

### 4.5.1. Descripción detallada

Declaracion generica del registro OUT del puerto 32 hasta el 39

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/gpio\\_matrix.h](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_matrix.h)

## 4.6. Referencia de la Unión GPIO\_PIN\_GENERIC

```
#include <gpio_matrix.h>
```

### Campos de datos

- uint32\_t **reg**
- 

```
struct {  
    uint32_t RESERVED1:2  
    _IO uint32_t PAD_DRIVER:1  
    uint32_t RESERVED2:4  
    _IO uint32_t INT_TYPE:3  
    _IO uint32_t WAKEUP_ENABLE:1  
    uint32_t RESERVED3:2  
    _IO uint32_t INT_ENA:5  
    uint32_t RESERVED4:13  
} campo
```

### 4.6.1. Descripción detallada

Declaracion generica del GPIO\_PIN

La documentación para esta unión fue generada a partir del siguiente fichero:

- [/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/gpio\\_matrix.h](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_matrix.h)

## 4.7. Referencia de la Unión IO\_MUX

### Campos de datos

- uint32\_t **reg**

- 

```
struct {  
    _IO uint32_t MCU_OE:1  
    _IO uint32_t SLP_SEL:1  
    _IO uint32_t MCU_WPD:1  
    _IO uint32_t MCU_WPU:1  
    _IO uint32_t MCU_IE:1  
    _IO uint32_t MCU_DRV:2  
    _IO uint32_t FUN_WPD:1  
    _IO uint32_t FUN_WPU:1  
    _IO uint32_t FUN_IE:1  
    _IO uint32_t FUN_DRV:2  
    _IO uint32_t MCU_SEL:3  
    uint32_t RESERVED:17  
} confirguration
```

La documentación para esta unión fue generada a partir del siguiente fichero:

- [/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/gpio\\_iomatrix.h](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/gpio_iomatrix.h)

## Capítulo 5

# Documentación de archivos

### 5.1. Referencia del Archivo /home/jamep/Documents/PlatformIO/Projects/DRIVERS\_GPIO\_ESP32/gpio\_drivers.h ↩

Funciones y declaraciones para el manejo del GPIO.

```
#include "gpio_matrix.h"
#include <stdbool.h>
```

#### defines

- `#define IN 0`
- `#define OUT 1`
- `#define ON 1`
- `#define OFF 0`
- `#define PULLUP OFF`
- `#define PULLDOWN ON`

#### Funciones

- void `GPIO_OUTPUT_ENABLE` (uint8\_t port)
- void `GPIO_OUTPUT_DISABLE` (uint8\_t port)
- void `GPIO_OUTPUT_SET` (uint8\_t port, bool value)
- void `GPIO_INPUT_ENABLE` (uint8\_t port, bool resistor)
- void `GPIO_INPUT_DISABLE` (uint8\_t port)
- bool `GPIO_INPUT_READ` (uint8\_t port)
- void `GPIO_INTERRUPTION_SET` (uint8\_t port, uint8\_t int\_type, uint8\_t signal\_type)

#### 5.1.1. Descripción detallada

Funciones y declaraciones para el manejo del GPIO.

## 5.1.2. Documentación de las funciones

### 5.1.2.1. GPIO\_INPUT\_DISABLE()

```
void GPIO_INPUT_DISABLE (
    uint8_t port )
```

Funcion: GPIO\_INPUT\_DISABLE Pre-condiciones: Ninguna Descripcion: Deshabilita el puerto seleccionado Valores de entrada: Puerto Valores de salida: Ninguno

### 5.1.2.2. GPIO\_INPUT\_ENABLE()

```
void GPIO_INPUT_ENABLE (
    uint8_t port,
    bool resistor )
```

Funcion: GPIO\_INPUT\_ENABLE Pre-condiciones: Ninguna Descripcion: Habilita la entrada del puerto seleccionado Valores de entrada: Puerto, tipo de resistencia Valores de salida: Ninguno

#### Ejemplos

[/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/main.c](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/main.c).

### 5.1.2.3. GPIO\_INPUT\_READ()

```
bool GPIO_INPUT_READ (
    uint8_t port )
```

Funcion: GPIO\_INPUT\_READ Pre-condiciones: Ninguna Descripcion: Lee el puerto seleccionado Valores de entrada: Puerto Valores de salida: Valor de la lectura

#### Ejemplos

[/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/main.c](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/main.c).

### 5.1.2.4. GPIO\_INTERRUPTION\_SET()

```
void GPIO_INTERRUPTION_SET (
    uint8_t port,
    uint8_t ena_type,
    uint8_t int_type )
```

Funcion: GPIO\_INTERRUPTION\_SET Pre-condiciones: Que el puerto seleccionado sea una entrada Descripcion: Habilita la interrupcion de la entrada (Sin terminar) Valores de entrada: Puerto, tipo de habilitacion, tipo de interrupcion Valores de salida: Ninguno



**5.1.2.5. GPIO\_OUTPUT\_DISABLE()**

```
void GPIO_OUTPUT_DISABLE (
    uint8_t port )
```

Funcion: GPIO\_OUTPUT\_DISABLE Pre-condiciones: Ninguna Descripcion: Deshabilita la salida del puerto seleccionado Valores de entrada: Puerto Valores de salida: Ninguno

**5.1.2.6. GPIO\_OUTPUT\_ENABLE()**

```
void GPIO_OUTPUT_ENABLE (
    uint8_t port )
```

Funcion: GPIO\_OUTPUT\_ENABLE Pre-condiciones: Ninguna Descripcion: Habilita la salida del puerto seleccionado Valores de entrada: Puerto Valores de salida: Ninguno

**Ejemplos**

[/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/main.c](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/main.c).

**5.1.2.7. GPIO\_OUTPUT\_SET()**

```
void GPIO_OUTPUT_SET (
    uint8_t port,
    bool value )
```

Funcion: GPIO\_OUTPUT\_SET Pre-condiciones: El puerto debe estar habilitado Descripcion: Asigna un valor de salida (0 o 1) al puerto seleccionado Valores de entrada: Puerto, Valor de salida Valores de salida: Ninguno

**Ejemplos**

[/home/jamep/Documents/PlatformIO/Projects/DRIVERS\\_GPIO\\_ESP32/main.c](/home/jamep/Documents/PlatformIO/Projects/DRIVERS_GPIO_ESP32/main.c).

**5.2. gpio\_drivers.h**

[Ir a la documentación de este archivo.](#)

```
1 /*****
2 * Title           : gpio drivers
3 * Filename        : gpio_drivers.h
4 * Author         : Javier Perez Macias, Marco Antonio Calderón Macías, Alejandro Morales Holguín
5 * Origin Date    : 04/12/2023
6 * Version        : 1.0.0
7 * Compiler       : CMAKE
8 * Target         : ESP32
9 *
10 *****/
11
12 #ifndef GPIO_DRIVER_H
13 #define GPIO_DRIVER_H
14
15 /*****
16 * Includes
17 *****/
18
19 #include "gpio_matrix.h"
20 #include <stdbool.h>
21 //Uso del tipo de dato bool
```

```

23
24 /*****
25 * Constantes del preprocesador
26 *****/
27 #define IN 0
28 #define OUT 1
29 #define ON 1
30 #define OFF 0
31 #define PULLUP OFF
32 #define PULLDOWN ON
33
34
35 /*****
36 * Prototipo de funciones
37 *****/
38 extern void GPIO_OUTPUT_ENABLE(uint8_t port);
39 extern void GPIO_OUTPUT_DISABLE(uint8_t port);
40 extern void GPIO_OUTPUT_SET(uint8_t port, bool value);
41 extern void GPIO_INPUT_ENABLE(uint8_t port, bool resistor);
42 extern void GPIO_INPUT_DISABLE(uint8_t port);
43 extern bool GPIO_INPUT_READ(uint8_t port);
44 extern void GPIO_INTERRUPT_SET(uint8_t port, uint8_t int_type, uint8_t signal_type);
45 #endif
46 /***** FIN DEL ARCHIVO *****/

```

### 5.3. Referencia del Archivo /home/jamep/Documents/PlatformIO/Projects/DRIVERS\_GPIO\_ESP32/gpio\_iomatrix.h

Declaraciones de registros de IOMUX.

```
#include <stdint.h>
```

#### Estructuras de datos

- union [IO\\_MUX](#)

#### defines

- #define [\\_I](#) volatile const
- #define [\\_IO](#) volatile
- #define [\\_O](#) \_IO
- #define [IO\\_MUX\\_GPIO0\\_REG](#) (0x3FF49044)
- #define [IO\\_MUX\\_GPIO1\\_REG](#) (0x3FF49088)
- #define [IO\\_MUX\\_GPIO2\\_REG](#) (0x3FF49040)
- #define [IO\\_MUX\\_GPIO3\\_REG](#) (0x3FF49084)
- #define [IO\\_MUX\\_GPIO4\\_REG](#) (0x3FF49048)
- #define [IO\\_MUX\\_GPIO5\\_REG](#) (0x3FF4906C)
- #define [IO\\_MUX\\_GPIO6\\_REG](#) (0x3FF49060)
- #define [IO\\_MUX\\_GPIO7\\_REG](#) (0x3FF49064)
- #define [IO\\_MUX\\_GPIO8\\_REG](#) (0x3FF49068)
- #define [IO\\_MUX\\_GPIO9\\_REG](#) (0x3FF49054)
- #define [IO\\_MUX\\_GPIO10\\_REG](#) (0x3FF49058)
- #define [IO\\_MUX\\_GPIO11\\_REG](#) (0x3FF4905C)
- #define [IO\\_MUX\\_GPIO12\\_REG](#) (0x3FF49034)
- #define [IO\\_MUX\\_GPIO13\\_REG](#) (0x3FF49038)
- #define [IO\\_MUX\\_GPIO14\\_REG](#) (0x3FF49030)
- #define [IO\\_MUX\\_GPIO15\\_REG](#) (0x3FF4903C)

- #define **IO\_MUX\_GPIO16\_REG** (0x3FF4904C)
- #define **IO\_MUX\_GPIO17\_REG** (0x3FF49050)
- #define **IO\_MUX\_GPIO18\_REG** (0x3FF49070)
- #define **IO\_MUX\_GPIO19\_REG** (0x3FF49074)
- #define **IO\_MUX\_GPIO20\_REG** (0x3FF49078)
- #define **IO\_MUX\_GPIO21\_REG** (0x3FF4907C)
- #define **IO\_MUX\_GPIO22\_REG** (0x3FF49080)
- #define **IO\_MUX\_GPIO23\_REG** (0x3FF4908C)
- #define **IO\_MUX\_GPIO24\_REG** (0x3FF49090)
- #define **IO\_MUX\_GPIO25\_REG** (0x3FF49024)
- #define **IO\_MUX\_GPIO26\_REG** (0x3FF49028)
- #define **IO\_MUX\_GPIO27\_REG** (0x3FF4902C)
- #define **IO\_MUX\_GPIO32\_REG** (0x3FF4901C)
- #define **IO\_MUX\_GPIO33\_REG** (0x3FF49020)
- #define **IO\_MUX\_GPIO34\_REG** (0x3FF49014)
- #define **IO\_MUX\_GPIO35\_REG** (0x3FF49018)
- #define **IO\_MUX\_GPIO36\_REG** (0x3FF49004)
- #define **IO\_MUX\_GPIO37\_REG** (0x3FF49008)
- #define **IO\_MUX\_GPIO38\_REG** (0x3FF4900C)
- #define **IO\_MUX\_GPIO39\_REG** (0x3FF49010)

## Variables

- const uint32\_t **ioMuxDirections** []

### 5.3.1. Descripción detallada

Declaraciones de registros de IOMUX.

### 5.3.2. Documentación de los 'defines'

#### 5.3.2.1. \_I

```
#define _I volatile const
```

Tipo de registro

#### 5.3.2.2. IO\_MUX\_GPIO0\_REG

```
#define IO_MUX_GPIO0_REG (0x3FF49044)
```

Direcciones de registros

### 5.3.3. Documentación de las variables

#### 5.3.3.1. ioMuxDirections

```
const uint32_t ioMuxDirections[]
```

Valor inicial:

```
= {
    IO_MUX_GPIO0_REG, IO_MUX_GPIO1_REG, IO_MUX_GPIO2_REG, IO_MUX_GPIO3_REG,
    IO_MUX_GPIO4_REG, IO_MUX_GPIO5_REG, IO_MUX_GPIO6_REG, IO_MUX_GPIO7_REG,
    IO_MUX_GPIO8_REG, IO_MUX_GPIO9_REG, IO_MUX_GPIO10_REG, IO_MUX_GPIO11_REG,
    IO_MUX_GPIO12_REG, IO_MUX_GPIO13_REG, IO_MUX_GPIO14_REG, IO_MUX_GPIO15_REG,
    IO_MUX_GPIO16_REG, IO_MUX_GPIO17_REG, IO_MUX_GPIO18_REG, IO_MUX_GPIO19_REG,
    0x00, IO_MUX_GPIO21_REG, IO_MUX_GPIO22_REG, IO_MUX_GPIO23_REG,
    IO_MUX_GPIO24_REG, IO_MUX_GPIO25_REG, IO_MUX_GPIO26_REG, IO_MUX_GPIO27_REG,
    0x00, 0x00, 0x00, 0x00,
    IO_MUX_GPIO32_REG, IO_MUX_GPIO33_REG, IO_MUX_GPIO34_REG, IO_MUX_GPIO35_REG,
    IO_MUX_GPIO36_REG, IO_MUX_GPIO37_REG, IO_MUX_GPIO38_REG, IO_MUX_GPIO39_REG
}
```

## 5.4. gpio\_iomatrix.h

[Ir a la documentación de este archivo.](#)

```
1 /*****
2 * Title           : gpio matrix
3 * Filename        : gpio_matrix.h
4 * Author          : Javier Perez Macias, Marco Antonio Calderón Macías, Alejandro Morales Holguín
5 * Origin Date     : 04/12/2023
6 * Version         : 1.0.0
7 * Compiler        : CMAKE
8 * Target          : ESP32
9 *
10 *****/
14 #ifndef GPIO_IOMATRIX_H
15 #define GPIO_IOMATRIX_H
16
17 #include <stdint.h>
18
19 /*****
20 * Constantes del preprocesador
21 *****/
25 #define _I volatile const //Solo lectura (RO)
26 #define _IO volatile //Escritura y lectura (R/W)
27 #define _O _IO //Solo escritura (WO)
28
32 #define IO_MUX_GPIO0_REG (0x3FF49044)
33 #define IO_MUX_GPIO1_REG (0x3FF49088)
34 #define IO_MUX_GPIO2_REG (0x3FF49040)
35 #define IO_MUX_GPIO3_REG (0x3FF49084)
36 #define IO_MUX_GPIO4_REG (0x3FF49048)
37 #define IO_MUX_GPIO5_REG (0x3FF4906C)
38 #define IO_MUX_GPIO6_REG (0x3FF49060)
39 #define IO_MUX_GPIO7_REG (0x3FF49064)
40 #define IO_MUX_GPIO8_REG (0x3FF49068)
41 #define IO_MUX_GPIO9_REG (0x3FF49054)
42 #define IO_MUX_GPIO10_REG (0x3FF49058)
43 #define IO_MUX_GPIO11_REG (0x3FF4905C)
44 #define IO_MUX_GPIO12_REG (0x3FF49034)
45 #define IO_MUX_GPIO13_REG (0x3FF49038)
46 #define IO_MUX_GPIO14_REG (0x3FF49030)
47 #define IO_MUX_GPIO15_REG (0x3FF4903C)
48 #define IO_MUX_GPIO16_REG (0x3FF4904C)
49 #define IO_MUX_GPIO17_REG (0x3FF49050)
50 #define IO_MUX_GPIO18_REG (0x3FF49070)
51 #define IO_MUX_GPIO19_REG (0x3FF49074)
52 #define IO_MUX_GPIO20_REG (0x3FF49078)
53 #define IO_MUX_GPIO21_REG (0x3FF4907C)
54 #define IO_MUX_GPIO22_REG (0x3FF49080)
55 #define IO_MUX_GPIO23_REG (0x3FF4908C)
56 #define IO_MUX_GPIO24_REG (0x3FF49090)
57 #define IO_MUX_GPIO25_REG (0x3FF49024)
58 #define IO_MUX_GPIO26_REG (0x3FF49028)
```

```

59 #define IO_MUX_GPIO27_REG    (0x3FF4902C)
60 #define IO_MUX_GPIO32_REG    (0x3FF4901C)
61 #define IO_MUX_GPIO33_REG    (0x3FF49020)
62 #define IO_MUX_GPIO34_REG    (0x3FF49014)
63 #define IO_MUX_GPIO35_REG    (0x3FF49018)
64 #define IO_MUX_GPIO36_REG    (0x3FF49004)
65 #define IO_MUX_GPIO37_REG    (0x3FF49008)
66 #define IO_MUX_GPIO38_REG    (0x3FF4900C)
67 #define IO_MUX_GPIO39_REG    (0x3FF49010)
68
69 /*****
70 * Variables
71 *****/
72 const uint32_t ioMuxDirections[] = {
73     IO_MUX_GPIO0_REG, IO_MUX_GPIO1_REG, IO_MUX_GPIO2_REG, IO_MUX_GPIO3_REG,
74     IO_MUX_GPIO4_REG, IO_MUX_GPIO5_REG, IO_MUX_GPIO6_REG, IO_MUX_GPIO7_REG,
75     IO_MUX_GPIO8_REG, IO_MUX_GPIO9_REG, IO_MUX_GPIO10_REG, IO_MUX_GPIO11_REG,
76     IO_MUX_GPIO12_REG, IO_MUX_GPIO13_REG, IO_MUX_GPIO14_REG, IO_MUX_GPIO15_REG,
77     IO_MUX_GPIO16_REG, IO_MUX_GPIO17_REG, IO_MUX_GPIO18_REG, IO_MUX_GPIO19_REG,
78     0x00, IO_MUX_GPIO21_REG, IO_MUX_GPIO22_REG, IO_MUX_GPIO23_REG,
79     IO_MUX_GPIO24_REG, IO_MUX_GPIO25_REG, IO_MUX_GPIO26_REG, IO_MUX_GPIO27_REG,
80     0x00, 0x00, 0x00, 0x00,
81     IO_MUX_GPIO32_REG, IO_MUX_GPIO33_REG, IO_MUX_GPIO34_REG, IO_MUX_GPIO35_REG,
82     IO_MUX_GPIO36_REG, IO_MUX_GPIO37_REG, IO_MUX_GPIO38_REG, IO_MUX_GPIO39_REG
83 };
84
85 /*****
86 * Typedefs
87 *****/
88 typedef union
89 {
90     uint32_t reg;
91     struct
92     {
93         _IO uint32_t MCU_OE :1;
94         _IO uint32_t SLP_SEL :1;
95         _IO uint32_t MCU_WPD :1;
96         _IO uint32_t MCU_WPU :1;
97         _IO uint32_t MCU_IE :1;
98         _IO uint32_t MCU_DRV :2;
99         _IO uint32_t FUN_WPD :1;
100        _IO uint32_t FUN_WPU :1;
101        _IO uint32_t FUN_IE :1;
102        _IO uint32_t FUN_DRV :2;
103        _IO uint32_t MCU_SEL :3;
104        uint32_t RESERVED :17;
105    } configuration;
106 } IO_MUX;
107
108
109
110 #endif /*gpio_iomatrix*/
111
112 /***** FIN DEL ARCHIVO *****/

```

## 5.5. Referencia del Archivo /home/jamep/Documents/PlatformIO/Projects/DRIVERS\_GPIO\_ESP32/gpio\_matrix.h

Declaraciones de registros del GPIO.

```
#include <stdint.h>
```

### Estructuras de datos

- struct GPIO\_GENERIC\_O\_0
- struct GPIO\_GENERIC\_O\_1
- struct GPIO\_GENERIC\_I\_0
- struct GPIO\_GENERIC\_I\_1
- union GPIO\_PIN\_GENERIC
- struct GPIO\_FUNC\_OUT\_SEL\_GENERIC

## defines

- `#define _I volatile const`
- `#define _IO volatile`
- `#define _O _IO`
- `#define IO0 0`
- `#define IO1 1`
- `#define IO2 2`
- `#define IO3 3`
- `#define IO4 4`
- `#define IO5 5`
- `#define IO6 6`
- `#define IO7 7`
- `#define IO8 8`
- `#define IO9 9`
- `#define IO10 10`
- `#define IO11 11`
- `#define IO12 12`
- `#define IO13 13`
- `#define IO14 14`
- `#define IO15 15`
- `#define IO16 16`
- `#define IO17 17`
- `#define IO18 18`
- `#define IO19 19`
- `#define IO20 20`
- `#define IO21 21`
- `#define IO22 22`
- `#define IO23 23`
- `#define IO24 24`
- `#define IO25 25`
- `#define IO26 26`
- `#define IO27 27`
- `#define IO28 28`
- `#define IO29 29`
- `#define IO30 30`
- `#define IO31 31`
- `#define IO32 32`
- `#define IO33 33`
- `#define IO34 34`
- `#define IO35 35`
- `#define IO36 36`
- `#define IO37 37`
- `#define IO38 38`
- `#define IO39 39`
- `#define LED4 IO2`
- `#define LED3 IO4`
- `#define LED2 IO16`
- `#define LED1 IO17`
- `#define APP_CPU 0x01`
- `#define APP_CPU_N 0x02`
- `#define PRO_CPU 0x04`
- `#define PRO_CPU_N 0x08`
- `#define DISABLE 0x00`
- `#define RISING_EDGE 0x01`

- #define **FALLING\_EDGE** 0x02
- #define **ANY\_EDGE** 0x03
- #define **LOW\_LEVEL** 0x04
- #define **HIGH\_LEVEL** 0x05
- #define **PERIFERAL\_BASE** ((uint32\_t) 0x3FF00000)
- #define **GPIO\_BASE\_DIR** ((uint32\_t)0x3FF44004)
- #define **GPIO\_OUT\_DIR** (GPIO\_BASE\_DIR)
- #define **GPIO\_OUT\_W1TS\_DIR** (GPIO\_BASE\_DIR + 0x0004)
- #define **GPIO\_OUT\_W1TC\_DIR** (GPIO\_BASE\_DIR + 0x0008)
- #define **GPIO\_OUT\_1\_DIR** (GPIO\_BASE\_DIR + 0x000C)
- #define **GPIO\_OUT\_1\_W1TS\_DIR** (GPIO\_BASE\_DIR + 0x0010)
- #define **GPIO\_OUT\_1\_W1TC\_DIR** (GPIO\_BASE\_DIR + 0x0014)
- #define **GPIO\_ENABLE\_DIR** (GPIO\_BASE\_DIR + 0x001C)
- #define **GPIO\_ENABLE\_1\_DIR** (GPIO\_BASE\_DIR + 0x0028)
- #define **GPIO\_IN\_DIR** (GPIO\_BASE\_DIR + 0x0038)
- #define **GPIO\_IN\_1\_DIR** (GPIO\_BASE\_DIR + 0x003C)
- #define **GPIO\_PIN\_DIR** (GPIO\_BASE\_DIR + 0x0084)
- #define **GPIO\_FUNC\_OUT\_SEL\_DIR** (0x3FF44530)
- #define **GPIO\_OUT** ((**GPIO\_GENERIC\_O\_0\***) GPIO\_OUT\_DIR)
- #define **GPIO\_OUT\_W1TS** ((**GPIO\_GENERIC\_O\_0\***) GPIO\_OUT\_W1TS\_DIR)
- #define **GPIO\_OUT\_W1TC** ((**GPIO\_GENERIC\_O\_0\***) GPIO\_OUT\_W1TC\_DIR)
- #define **GPIO\_OUT\_1** ((**GPIO\_GENERIC\_O\_1\***) GPIO\_OUT\_1\_DIR)
- #define **GPIO\_OUT\_1\_W1TS** ((**GPIO\_GENERIC\_O\_0\***) GPIO\_OUT\_1\_W1TS\_DIR)
- #define **GPIO\_OUT\_1\_W1TC** ((**GPIO\_GENERIC\_O\_0\***) GPIO\_OUT\_1\_W1TC\_DIR)
- #define **GPIO\_IN** ((**GPIO\_GENERIC\_I\_0\***) GPIO\_IN\_DIR)
- #define **GPIO\_IN\_1** ((**GPIO\_GENERIC\_I\_1\***) GPIO\_IN\_1\_DIR)
- #define **GPIO\_ENABLE** ((**GPIO\_GENERIC\_O\_0\***) GPIO\_ENABLE\_DIR)
- #define **GPIO\_ENABLE\_1** ((**GPIO\_GENERIC\_O\_1\***) GPIO\_ENABLE\_1\_DIR)
- #define **GPIO\_PIN**(X) (GPIO\_PIN\_DIR + (4\*X))
- #define **GPIO\_FUNC\_OUT\_SEL**(X) ((**GPIO\_FUNC\_OUT\_SEL\_GENERIC\***)(GPIO\_FUNC\_OUT\_SEL\_DIR + (4\*X)))

### 5.5.1. Descripción detallada

Declaraciones de registros del GPIO.

### 5.5.2. Documentación de los 'defines'

#### 5.5.2.1. **\_I**

```
#define _I volatile const
```

Tipo de registro

#### 5.5.2.2. **APP\_CPU**

```
#define APP_CPU 0x01
```

Interrupciones parametros

### 5.5.2.3. GPIO\_OUT

```
#define GPIO_OUT ((GPIO_GENERIC_O_0*) GPIO_OUT_DIR)
```

Punteros a los registros

### 5.5.2.4. IO0

```
#define IO0 0
```

Puertos

### 5.5.2.5. PERIFERAL\_BASE

```
#define PERIFERAL_BASE ((uint32_t) 0x3FF00000)
```

Direcciones de registros

## 5.6. gpio\_matrix.h

[Ir a la documentación de este archivo.](#)

```
1 /*****
2 * Title           : gpio matrix
3 * Filename        : gpio_matrix.h
4 * Author         : Javier Perez Macias, Marco Antonio Calderón Macías, Alejandro Morales Holguín
5 * Origin Date    : 04/12/2023
6 * Version       : 1.0.0
7 * Compiler      : CMAKE
8 * Target        : ESP32
9 *
10 *****/
14 #ifndef GPIO_MATRIX_H
15 #define GPIO_MATRIX_H
16
17 #include <stdint.h>
18
19 /*****
20 * Constantes del preprocesador
21 *****/
22
26 #define _I volatile const //Solo lectura (RO)
27 #define _IO volatile //Escritura y lectura (R/W)
28 #define _O _IO //Solo escritura (WO)
29
33 #define IO0 0
34 #define IO1 1
35 #define IO2 2
36 #define IO3 3
37 #define IO4 4
38 #define IO5 5
39 #define IO6 6
40 #define IO7 7
41 #define IO8 8
42 #define IO9 9
43 #define IO10 10
44 #define IO11 11
45 #define IO12 12
46 #define IO13 13
47 #define IO14 14
48 #define IO15 15
49 #define IO16 16
50 #define IO17 17
51 #define IO18 18
52 #define IO19 19
53 #define IO20 20
54 #define IO21 21
55 #define IO22 22
```



```

56 #define IO23 23
57 #define IO24 24
58 #define IO25 25
59 #define IO26 26
60 #define IO27 27
61 #define IO28 28
62 #define IO29 29
63 #define IO30 30
64 #define IO31 31
65 #define IO32 32
66 #define IO33 33
67 #define IO34 34
68 #define IO35 35
69 #define IO36 36
70 #define IO37 37
71 #define IO38 38
72 #define IO39 39
73
74 #define LED4 IO2
75 #define LED3 IO4
76 #define LED2 IO16
77 #define LED1 IO17
78
82 #define APP_CPU      0x01
83 #define APP_CPU_N    0x02
84 #define PRO_CPU      0x04
85 #define PRO_CPU_N    0x08
86
87 #define DISABLE      0x00
88 #define RISING_EDGE  0x01
89 #define FALLING_EDGE 0x02
90 #define ANY_EDGE     0x03
91 #define LOW_LEVEL    0x04
92 #define HIGH_LEVEL   0x05
93
94
98 #define PERIPHERAL_BASE      ((uint32_t) 0x3FF00000)
99 #define GPIO_BASE_DIR        ((uint32_t) 0x3FF44004)
100 #define GPIO_OUT_DIR          (GPIO_BASE_DIR)
101 #define GPIO_OUT_W1TS_DIR     (GPIO_BASE_DIR + 0x0004)
102 #define GPIO_OUT_W1TC_DIR     (GPIO_BASE_DIR + 0x0008)
103 #define GPIO_OUT_1_DIR        (GPIO_BASE_DIR + 0x000C)
104 #define GPIO_OUT_1_W1TS_DIR   (GPIO_BASE_DIR + 0x0010)
105 #define GPIO_OUT_1_W1TC_DIR   (GPIO_BASE_DIR + 0x0014)
106 #define GPIO_ENABLE_DIR       (GPIO_BASE_DIR + 0x001C)
107 #define GPIO_ENABLE_1_DIR     (GPIO_BASE_DIR + 0x0028)
108 #define GPIO_IN_DIR           (GPIO_BASE_DIR + 0x0038)
109 #define GPIO_IN_1_DIR         (GPIO_BASE_DIR + 0x003C)
110 #define GPIO_PIN_DIR          (GPIO_BASE_DIR + 0x0084)
111 #define GPIO_FUNC_OUT_SEL_DIR (0x3FF44530)
112
116 #define GPIO_OUT              ((GPIO_GENERIC_O_0*) GPIO_OUT_DIR)
117 #define GPIO_OUT_W1TS         ((GPIO_GENERIC_O_0*) GPIO_OUT_W1TS_DIR)
118 #define GPIO_OUT_W1TC         ((GPIO_GENERIC_O_0*) GPIO_OUT_W1TC_DIR)
119 #define GPIO_OUT_1            ((GPIO_GENERIC_O_1*) GPIO_OUT_1_DIR)
120 #define GPIO_OUT_1_W1TS       ((GPIO_GENERIC_O_0*) GPIO_OUT_1_W1TS_DIR)
121 #define GPIO_OUT_1_W1TC       ((GPIO_GENERIC_O_0*) GPIO_OUT_1_W1TC_DIR)
122 #define GPIO_IN               ((GPIO_GENERIC_I_0*) GPIO_IN_DIR)
123 #define GPIO_IN_1             ((GPIO_GENERIC_I_1*) GPIO_IN_1_DIR)
124 #define GPIO_ENABLE           ((GPIO_GENERIC_O_0*) GPIO_ENABLE_DIR)
125 #define GPIO_ENABLE_1         ((GPIO_GENERIC_O_1*) GPIO_ENABLE_1_DIR)
126
127
128 /*****
129 * Macros
130 *****/
131 #define GPIO_PIN(X)          (GPIO_PIN_DIR + (4*X))
132 #define GPIO_FUNC_OUT_SEL(X) ((GPIO_FUNC_OUT_SEL_GENERIC*) (GPIO_FUNC_OUT_SEL_DIR + (4*X)))
133
134
135 /*****
136 * Typedefs
137 *****/
138
139 typedef struct
140 {
141     _IO uint32_t    REG_IO;
142 } GPIO_GENERIC_O_0;
143
144 typedef struct
145 {
146     _IO uint32_t    REG_IO :8;
147     uint32_t        RESERVED:24;
148 } GPIO_GENERIC_O_1;
149
150 typedef struct
151 {
152     _IO uint32_t    REG_IO :8;
153     uint32_t        RESERVED:24;
154 } GPIO_GENERIC_O_1;
155
156 typedef struct
157 {

```

```
161     _I uint32_t    REG_IO;
162 } GPIO_GENERIC_I_0;
163
164 typedef struct
165 {
166     _I uint32_t    REG_IO      :8;
167     uint32_t    RESERVED    :24;
168 } GPIO_GENERIC_I_1;
169
170 typedef union
171 {
172     uint32_t reg;
173     struct
174     {
175         uint32_t RESERVED1 :2;
176         _IO uint32_t PAD_DRIVER :1;
177         uint32_t RESERVED2 :4;
178         _IO uint32_t INT_TYPE :3;
179         _IO uint32_t WAKEUP_ENABLE :1;
180         uint32_t RESERVED3 :2;
181         _IO uint32_t INT_ENA :5;
182         uint32_t RESERVED4 :13;
183     } campo;
184 } GPIO_PIN_GENERIC;
185
186 typedef struct
187 {
188     _IO uint32_t REG;
189 } GPIO_FUNC_OUT_SEL_GENERIC;
190
191
192
193 #endif
194 /***** FIN DEL ARCHIVO *****/
```

## Capítulo 6

# Documentación de ejemplos

### 6.1. main.c

Una demostración de un uso básico del GPIO

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "gpio_drivers.h"
void app_main(void)
{
    GPIO_OUTPUT_ENABLE(IO5);
    GPIO_OUTPUT_SET(IO5, ON);
    GPIO_INPUT_ENABLE(IO18, PULLUP);

    while (1)
    {
        GPIO_OUTPUT_SET(IO5, OFF);
        vTaskDelay(pdMS_TO_TICKS(500));
        printf("%d \n", GPIO_INPUT_READ(IO18));
        GPIO_OUTPUT_SET(IO5, ON);
        vTaskDelay(pdMS_TO_TICKS(500));
    }
}
```

