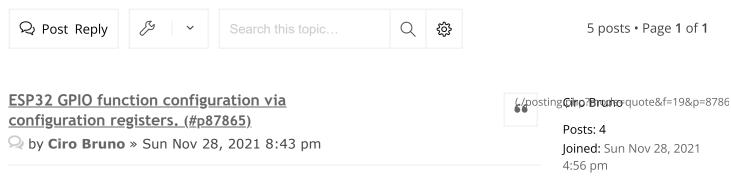
## ESP32 GPIO function configuration via configuration registers.



#### Hello all

I've been using ESP32 for a couple of years, mostly through Arduino IDE. Since July I've been using exclusively VSCode + PlatformIO with Arduino framework. My reason for changing has been the interruption of support to SPIFFS in the latest Arduino IDE versions.

Nevertheless, now I'm facing difficulties for setting up GPIO 12 and 13 as INPUT\_PULLUP in a ESP32-Pico-D4. I've been using

```
CODE: SELECT ALL (#)
pinMode( 12, INPUT PULLUP);
```

, but this always return 0 (false), as well as trying to set it as INPUT or INPUT\_PULLDOWN. Browsing some discussions in the internet I found out that the default Arduino framework version used by VSCode is outdated and has issues regarding some GPIO pins configuration.

So, I'd like to use the safest solution for this issue, that is configuring these GPIOs directly through configuration registers.

I've been reading chapter 5 of esp32\_technical\_reference\_manual\_en.pdf, but I didn't

understand what to do. Maybe I need to understand better the microprocessor's architecture itself.

Anyway, I'd like to count on your help for:

item 1) setting up GPIO12 and GPIO13 as inputs with pullup, through writing on configuration registers;

item 2) understanding better how attribute functions to GPIO pads.

I'd like to start by item 1, making GPIO 12 and 13 behave as INPUT\_PULLUP by writing on the proper configuration registers. Then, I'd like to play with other basic IO functions, like INPUT\_PULLDOWN, INPUT and OUTPUT.

Later (item 2, I'd like to go through an overview of the GPIOs related registers, from a single GPIO pad toward a specific embedded function.

Starting from Figure 7 ("IO\_MUX, RTC IO\_MUX and GPIO Matrix Overview") on page 47, I'd like to understand where I should find the simple digital IOs, such as those accessible through "digitalRead" or "digitalWrite".

- 2.1) Are these ones are considered as "peripherals" or not? I can't identify them in the diagram.
- 2.2) Should they be treated attached directly to IO\_MUX or via GPIO Matrix?
- 2.3) This Figure also shows "34 GPIOs" between "GPIO Matrix" and "IO\_MUX". Isn't 34 the same number of GPIO pads? Is it a coincidence or is this label ("34 GPIOs") misplaced in the diagram?



Thank you. Best regards, Ciro.

#### **ATTACHMENTS**

Figure 7.png (43.97 KiB) Viewed 7043 times

## Re: ESP32 GPIO function configuration via configuration registers. (#p87896)

Q by **boarchuz** » Mon Nov 29, 2021 6:32 am

(/postinghop?chode=quote&f=19&p=8789

Posts: 470

Joined: Tue Aug 21, 2018

5:28 am

66 <u>Ciro Bruno (./memberlist.php?</u>

<u>mode=viewprofile&u=22668&sid=2c03cbaa2417a6da6f19d44b5588631e}</u>**wrote**: <u>↑ (,/viewtopic.php?</u> p=87865&sid=2c03cbaa2417a6da6f19d44b5588631e#p87865)

Sun Nov 28, 2021 8:43 pm

I've been using



```
CODE: SELECT ALL (#)
```

pinMode( 12, INPUT PULLUP);

, but this always return 0 (false)

pinMode is void:

https://github.com/espressif/arduino-es ... gpio.h#L88

(https://github.com/espressif/arduino-

esp32/blob/399f4ecbb3a4cef21e2bffa37adb6190356dfb76/cores/esp32/esp32-hal-gpio.h#L88)\_

Do you mean that reading the pin level (ie. with digitalRead(12)) afterwards always returns 0?

Check your particular board's schematic to ensure 12 isn't used for something else, disconnect any external modules/LEDs/resistors/etc, and check the voltage on 12 with your multimeter. Do the same with a different pin (eg. 25 or 32).

# Re: ESP32 GPIO function configuration via configuration registers. (#p88049)

Q by **Ciro Bruno** » Wed Dec 01, 2021 10:10 pm

(/posting@irp?Brooks-quote&f=19&p=8804

Posts: 4

Joined: Sun Nov 28, 2021

4:56 pm

Hi

Thank you for your attention

Indeed I've already checked all electronic signals, comparing GPIOs 12 and 13 to GPIO25.

Configuring all them as INPUT\_PULLUP, GPIO25 holds close to 3.3v, while GPIO 12 and 13 keep around 0.3v.

Keeping them loose and reading them with "digitalRead()", GPIO25 keeps boolean 1 state, while GPIOs 12 and 13 never leave boolean 0.

When grounding them, one at once, GPIO25 returns boolean 0, as expected.

Reading GPIO with "touchRead()", GPIOs 12 and 13 show values between 22 and 48 when loose, falling to 0 when grounded. This is the only life signal these pins show.

All this happens when compiling and flashing with VSCode.

When compile and flashing with <u>Arduino IDE 2.0.0</u>, these pins <u>work perfectly as expected</u>.

So, I guess there should be no doubt about hardware integrity. That's why I'm looking for solving that through configuration registers.

Regards,

# Re: ESP32 GPIO function configuration via configuration registers. (#p88223)

by Ciro Bruno » Sun Dec 05, 2021 9:06 pm

(/posting@irp?Bronho-quote&f=19&p=8822

Posts: 4

Joined: Sun Nov 28, 2021

4:56 pm

Again, trying to find out how to configure digital inputs and output through ports registers, I've been trying to apply "pinMode" commands to three different GPIOs and check its effect over some registers. But regardless the pin mode selected, it seems I'm watching the wrong registers.

The code I've been using is:

And the results I've been cyclically obtaining are:

This code has run in Arduino 2.0.0-beta.7

Given these results, which might be the registers affected by "pinMode" command?

Please help. Thank you. Regards, Ciro.

Re: ESP32 GPIO function configuration via configuration registers. (#p88783)

by Ciro Bruno » Tue Dec 14, 2021 2:27 pm

(/posting@inp?Brunho-quote&f=19&p=8878

Posts: 4

The matter here it that the compiler doesn't allow the register to be read or written as a variable.

ciro\_bruno:

Serial.print(GPIO\_IN\_REG, BIN); So, doing so I can get just the register's address itself (0b1111111111000010000000111100).

As far as I can remember, in Arduino environment it is/was possible to read and write directly on the registers through their mnemonics.

Anyway, I guess I've got to find out another way.

I've been thinking of doing so via pointers. I've been trying, but I haven't been able to attribute the register's address (a constant) to the pointer . Neither can read the address the pointer points to.

```
CODE: SELECT ALL(#)

uint32_t conteudo;
uint32_t *reg_PTN;

reg_PTN = (uint32_t) GPIO_ENABLE_REG; // #define GPIO_ENABLE_REG (DR_REG conteudo = *reg_PTN;

Serial.print("\nGPIO_ENABLE_REG = 0x"); Serial.print( reg_PTN, HEX);
Serial.print("\tConteudo = 0b"); Serial.print (*reg_PTN, BIN);
```

The error messages are:

So, how could I give this pointer the register's address?

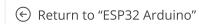
Thank you.

Regards,

Ciro.











## WHO IS ONLINE

Users browsing this forum: No registered users and 16 guests

### **About Us**

Espressif Systems is a fabless semiconductor company providing cutting-edge low power WiFi SoCs and wireless solutions for wireless communications and Internet of Things applications. ESP8266EX and ESP32 are some of our products.

### Extra

- → Espressif Homepage
- → ESP8266EX Official Forum
- **→** ESP8266 Community Forum

### Information

- → Terms of use
- Privacy policy
- → FAQ
- The team
- → Contact us

Espressif ESP32 ... Available now!



