

Universidad Autónoma de Baja California  
Facultad de Ciencias Químicas e Ingeniería



**MICROCONTROLADORES**

**Taller 3 - GPIOs**

**Docente:** Aguilar Noriega Leocundo

**Alumno:** Gómez Cárdenas Emmanuel Alberto

**Matricula:** 01261509

**Actividades:**

1. Incluya la visualización del registro GPIO\_OUT inmediatamente después de presentar el mensaje del estado del LED. Para ello haga uso de la función **printf()** con el formateo correspondiente a la visualización de un valor de 32bits en formato hexadecimal.
  - a) **Compile y cargue el proyecto a la placa para realizar la pruebas necesarias de visualización, para ello verifique el valor del bit 2 del registro pues es el bit correspondiente a la terminal GPIO2.**
  - b) **¿Qué valor tiene GPIO\_OUT cuando se enciende el LED y qué valor tiene el bit 2?**
    - GPIO\_OUT vale 0x00000004h y el bit 2 tiene valor 1 cuando el led se enciende.
  - c) **¿Qué valor tiene GPIO\_OUT cuando se apaga el LED y qué valor tiene el bit 2?**
    - GPIO\_OUT vale 0x00000000h y el bit 2 tiene valor 0 cuando el led se apaga.

---

```
Value of GPIO Output Register at address 0x3ff44004 when ON:  0x00000004h
Bit 2 = 1
```

```
Value of GPIO Output Register at address 0x3ff44004 when OFF: 0x00000000h
Bit 2 = 0
```

---

2. Incluya la visualización del registro GPIO\_IN inmediatamente después de presentar el mensaje del estado del LED. Para ello es necesario definir un nuevo apuntador GPIO\_IN con la dirección correspondiente a ese registro. Luego haga uso de la función **printf()** con el formateo correspondiente a la visualización de un valor de 32bits en formato hexadecimal.

a) **Compile y cargue el proyecto a la placa para realizar la pruebas necesarias de visualización, para ello verifique el valor del bit 2 del registro pues es el bit correspondiente a la terminal GPIO2.**

b) **¿Qué valor tiene GPIO\_IN cuando se enciende el LED y qué valor tiene el bit 2?**

- GPIO\_IN vale 0x0002000eh y el bit 2 tiene valor 1 cuando el led se enciende.

c) **¿Qué valor tiene GPIO\_IN cuando se apaga el LED y qué valor tiene el bit 2?**

- GPIO\_IN vale 0x0002000ah y el bit 2 tiene valor 0 cuando el led se apaga.

---

```
Value of GPIO Input Register at address 0x3ff4403c when ON: 0x0002000eh
Bit 2 = 1
```

```
Value of GPIO Output Register at address 0x3ff44004 when ON: 0x00000004h
Bit 2 = 1
```

```
Value of GPIO Input Register at address 0x3ff4403c when OFF: 0x0002000ah
Bit 2 = 0
```

```
Value of GPIO Output Register at address 0x3ff44004 when OFF: 0x00000000h
Bit 2 = 0
```

---

3. Realice el siguiente cambio en la configuración de GPIO2 para configurar de la siguiente manera:

```
gpio_set_direction( LED_GPIO, GPIO_MODE_INPUT_OUTPUT );
```

- a) **Compile y cargue el proyecto a la placa para realizar la pruebas necesarias de visualización, para ello verifique el valor del bit 2 del registro pues es el bit correspondiente a la terminal GPIO2.**
- b) **¿Qué valor tiene GPIO\_IN cuando se enciende el LED y qué valor tiene el bit 2?**
  - GPIO\_IN vale 0x0002000eh y el bit 2 tiene valor 1 cuando el led se enciende.
- c) **¿Qué valor tiene GPIO\_IN cuando se apaga el LED y qué valor tiene el bit 2?**
  - GPIO\_IN vale 0x0002000ah y el bit 2 tiene valor 0 cuando el led se apaga.

-----

Value of GPIO Input Register at address 0x3ff4403c when ON: 0x0002000eh  
Bit 2 = 1

Value of GPIO Output Register at address 0x3ff44004 when ON: 0x00000004h  
Bit 2 = 1

Value of GPIO Input Register at address 0x3ff4403c when OFF: 0x0002000ah  
Bit 2 = 0

Value of GPIO Output Register at address 0x3ff44004 when OFF: 0x00000000h  
Bit 2 = 0

-----

## Conclusiones

En este taller se trabajó con los registros de entrada y salida GPIO del ESP32 así como sus configuraciones, realizando escritura y lectura de los registros. Este tipo de ejercicios nos ayudan a familiarizarnos con el ESP32 y entender el funcionamiento de los diferentes configuraciones es fundamental para el desarrollo de proyectos mas complejos, tambien es importante comprender que cada modo de configuración de los pines tiene un propósito específico y debe ser utilizado según sea necesario en cada situación.

## Comentarios

Es importante destacar que a pesar de que GPIO\_MODE\_INPUT\_OUTPUT y GPIO\_MODE\_OUTPUT nos dieron el mismo resultado, no se debe asumir que siempre nos dará el mismo resultado, por ejemplo, si configuramos un pin en el modo GPIO\_MODE\_OUTPUT y se espera leer un valor de entrada en ese pin, es posible que el valor leído no sea el esperado. Del mismo modo, si configuramos un pin en el modo GPIO\_MODE\_INPUT\_OUTPUT y se espera obtener una señal de salida precisa, es posible que la señal enviada no sea exactamente la esperada debido a posibles interferencias de la señal de entrada.

## Código

```
/*
 * Autor: Emmanuel Alberto Gómez Cárdenas
 * Fecha: 12/03/2023
 * Descripcion: Taller 3 - GPIOs
 * Taller de la clase Microcontroladores
 */

/* Blink Example */
#include <stdio.h>
#include "driver/gpio.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "sdkconfig.h"

#define LED_GPIO 2

void app_main() {
    /* Configure the peripheral according to the LED type */
    gpio_reset_pin(LED_GPIO);
    /* Set the GPIO as a push/pull output */
    gpio_set_direction(LED_GPIO, GPIO_MODE_INPUT_OUTPUT);

    volatile uint32_t *GPIO_OUT = (uint32_t)0x3FF44004;
    volatile uint32_t *GPIO_IN = (uint32_t)0x3FF4403C;

    while (1) {
        gpio_set_level(LED_GPIO, 0);
        printf("Value of GPIO Input Register at address %p when OFF: 0x%08x\n",
            GPIO_IN, *GPIO_IN);
        printf("Bit 2 = %d\n", (*GPIO_IN & (1 << 2)) ? 1 : 0);
        printf("Value of GPIO Output Register at address %p when OFF: 0x%08x\n",
            GPIO_OUT, *GPIO_OUT);
        printf("Bit 2 = %d\n", (*GPIO_OUT & (1 << 2)) ? 1 : 0);
        vTaskDelay(50 / portTICK_PERIOD_MS);

        gpio_set_level(LED_GPIO, 1);
        printf("Value of GPIO Input Register at address %p when ON: 0x%08x\n",
            GPIO_IN, *GPIO_IN);
        printf("Bit 2 = %d\n", (*GPIO_IN & (1 << 2)) ? 1 : 0);
        printf("Value of GPIO Output Register at address %p when ON: 0x%08x\n",
            GPIO_OUT, *GPIO_OUT);
        printf("Bit 2 = %d\n", (*GPIO_OUT & (1 << 2)) ? 1 : 0);
        printf(
            "\n-----"
            "-----\n\n");
        vTaskDelay(3000 / portTICK_PERIOD_MS);
    }
}
```