

Laboratorio GPIO

Miguel Angel Gonzalez (7003847), Laura
Yuleissy Gutierrez (7003739)

Resumen - A través del cambio de estado en el pulsador de USUARIO se realizó el incremento de un contador de 4 bits, cuyo valor se refleja en 1 display de 7 segmentos y se muestra de forma sucesiva los nombres de los integrantes del grupo de manera automática. Los 30 LEDS deben estar distribuidos en al menos 3 puertos diferentes y las condiciones se determinan dependiendo el estado lógico de la tabla de valores con una frecuencia de 0.1 Hz, y en corrimientos de una posición con una frecuencia de 0.5 Hz.

I. INTRODUCCIÓN

En este documento se encuentran los resultados obtenidos a partir del uso de los registros GPIO con la placa STM32F747ZG, se presentan los pines seleccionados y el análisis lógico de su funcionamiento.

II. MÉTODOS Y MATERIALES

El procedimiento realizado en la práctica de laboratorio, se desarrolló por medio del uso de los registros GPIO que permitieron programar el funcionamiento propuesto durante la práctica, de esta manera se mostrará los tipos de registros utilizados en el programa y los materiales implementados.

A continuación, para hacer más claro la lógica del programa, se presentan los registros utilizados:

A. GPIO Registros

GPIO port mode register (GPIOx_MODER) (x=A to K):
Se presenta el registro encargado de declarar las señales de entrada, salida y como función análoga.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]	MODER14[1:0]	MODER13[1:0]	MODER12[1:0]	MODER11[1:0]	MODER10[1:0]	MODER9[1:0]	MODER8[1:0]	MODER7[1:0]	MODER6[1:0]	MODER5[1:0]	MODER4[1:0]	MODER3[1:0]	MODER2[1:0]	MODER1[1:0]	MODER0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **MODER[15:0][1:0]**: Port x configuration I/O pin y (y = 15 to 0)
These bits are written by software to configure the I/O mode.
00: Input mode (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

Ilustración 1. GPIO Moder Register

GPIO port input data register (GPIOx_IDR) (x = A to K):
Se presenta el registro encargado de leer los pines que se declararon anteriormente.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDR[15:0]**: Port x input data I/O pin y (y = 15 to 0)
These bits are read-only. They contain the input value of the corresponding I/O port.

Ilustración 2. GPIO IDR Register

GPIO port output data register (GPIOx_ODR) (x = A to K):
Se presenta el registro encargado de escritura, de esta manera se enciende y/o apaga el LED.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODR[15:0]**: Port output data I/O pin y (y = 15 to 0)
These bits can be read and written by software.

Note: For atomic bit set/reset, the ODR bits can be individually set and/or reset by writing to the GPIOx_BSRR or GPIOx_BRR registers (x = A..F).

Ilustración 3. GPIO ODR Register

GPIO port output data register (GPIOx_OSPEEDR) (x = A to K):

Se presenta el registro encargado de velocidad, para medir el tiempo en que se encuentra funcionando el programa.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15[1:0]	OSPEEDR14[1:0]	OSPEEDR13[1:0]	OSPEEDR12[1:0]	OSPEEDR11[1:0]	OSPEEDR10[1:0]	OSPEEDR9[1:0]	OSPEEDR8[1:0]	OSPEEDR7[1:0]	OSPEEDR6[1:0]	OSPEEDR5[1:0]	OSPEEDR4[1:0]	OSPEEDR3[1:0]	OSPEEDR2[1:0]	OSPEEDR1[1:0]	OSPEEDR0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **OSPEEDR[15:0][1:0]**: Port x configuration I/O pin y (y = 15 to 0)

These bits are written by software to configure the I/O output speed.

00: Low speed
01: Medium speed
10: High speed
11: Very high speed

Note: Refer to the product datasheets for the values of OSPEEDRy bits versus V_{DD} range and external load.

Ilustración 4. GPIO OSPEEDR Register

B. Materiales

- Software uVision Keil.
- Tarjeta de desarrollo STM32F746ZG
- Cable mini USB.
- LEDs varios colores

- Pulsadores
- Display 7 segmentos
- Protoboard

C. Actividad de laboratorio

A continuación, se muestra la tabla de verdad con las siguientes condiciones a realizar en el programa Uvision Keil :

Pin de Control #1	Pin de Control #2	Acción con el pulsador de usuario
0	0	El pulsador de usuario no debe funcionar, y todos los LEDS disponibles en el montaje deben encender y apagar un tiempo determinado por el estudiante.
0	1	Por medio del estado lógico del pulsador DE USUARIO, se debe ejecutar una secuencia nombrada como "SECUENCIA SETUP GAMING" explicada en el numeral 11.5
1	0	A través del cambio de estado en el pulsador de USUARIO realicen el incremento de un contador de 4 bits, cuyo valor se refleje en 1 display de 7 segmentos. (No hay que usar interrupciones, pura lógica)
1	1	En 1 display de 7 segmentos ir mostrando de forma sucesiva los nombres de los integrantes del grupo de manera automática. Se debe usar el abecedario de la ilustración 2.

Tabla 1. Tabla de verdad con las funciones

De la misma manera, se muestra una guía de cómo se vería reflejado la secuencia "setup gaming":



Ilustración 5. Configuración Display 7 segmentos

Los 30 LEDS deben estar distribuidos en al menos 3 puertos diferentes:

- Si el estado lógico proveniente del pulsador es alto, se deben ver encendidos dos diodos LED, moviéndose tres posiciones desde el centro hacia afuera con una frecuencia de 0.1 Hz, una vez alcance los extremos debe regresar al centro, pero en corrimientos de una posición con una frecuencia de 0.5 Hz.
- Si el estado lógico proveniente del pulsador es bajo, se deben ver encendidos cuatro diodos LED a la derecha de la trama. Se

deben mover dos espacios a la izquierda hasta alcanzar el otro extremo y devolverse de la misma manera. La frecuencia del movimiento en este caso es de 0.25 Hz.

D. Algoritmia

Se presenta el diagrama de bloques de la lógica de programación que se aplicó para crear las condiciones dadas por medio del uso de registros y condicionales en el programa KEIL haciendo uso de 3 puertos para 30 pines que se reflejarán en el siguiente diagrama:

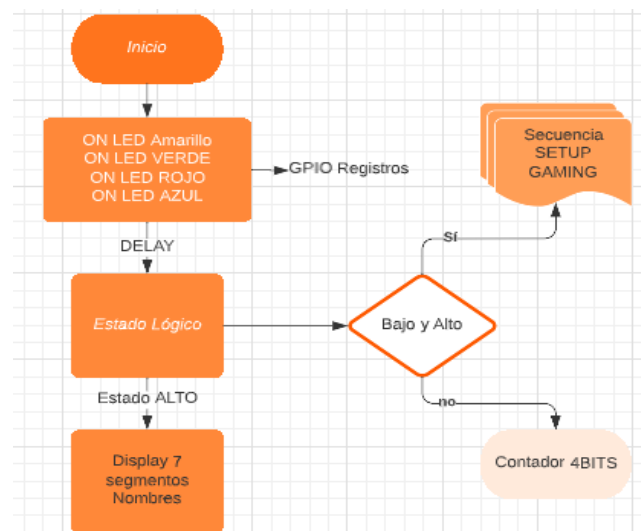


Fig 2. Diagrama de bloques.

III. RESULTADOS

A continuación, se presenta la primera condición donde todos los LEDS usados en el puerto C, D & F, en estado BAJO se deben encender y apagar oscilando un tiempo de 1s:

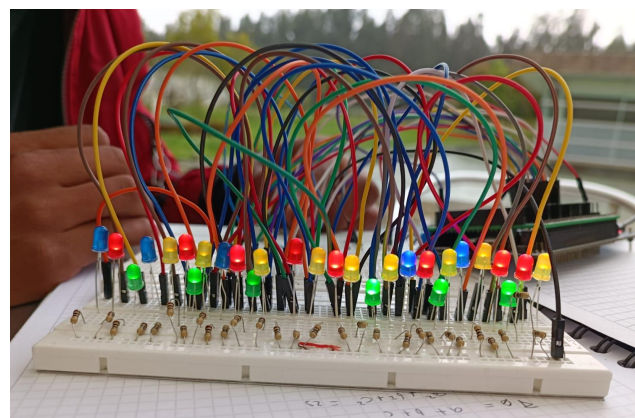


Ilustración 6. Montaje LEDS

De la misma manera se ven las conexiones de los pulsadores y el display donde se realiza las pulsaciones en alto o bajo para cumplir las condiciones SETUP GAMING dadas por medio del uso de condicionales y registros.

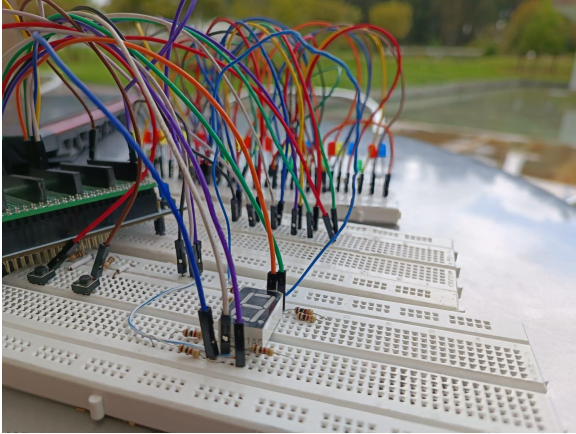


Ilustración 7. Montaje Display 7 segmentos

Finalmente, se ve reflejado el display 7 segmentos donde se cumple la condición del estado ALTO con éxito para mostrar los nombres de los integrantes del laboratorio, y así mismo, realizar la programación para el contador de 4 bits por cada pulsación, para esto, se requirió hacer uso de un tercer pulsador..

IV. CONCLUSIONES

1. Se comprendió el uso de los distintos tipos de registros GPIO que permitió cumplir las condiciones del comportamiento de los estados lógicos.
2. Se conocieron los diferentes tipos de puertos y pines en la placa para el desarrollo de la programación.
3. Se establecieron conocimientos previos por medio del lenguaje de programación C++ haciendo uso de registros y condicionales.

V. REFERENCIAS

- [1] STMicroelectronics, *RM0385 Reference manual*. 2018.
- [2] STMicroelectronics, *UM1974 User manual*. 2020.
- [3] STMicroelectronics, *STM32F745xx STM32F746xx*, 2016