



## **Práctica 2 Semáforo**

**Fecha:** 29/01/22

**Alumno:**

Sosa Zepeda Yessica

## **PRÁCTICA 2.**

### **Semáforo**

#### **INTRODUCCIÓN**

GPIO (General Purpose Input/Output) es un término utilizado en electrónica para describir un tipo de puerto de entrada/salida (I/O) en un microcontrolador o sistema de control de computadora.

Un puerto GPIO es un pin digital que puede ser programado para funcionar como una entrada o una salida. Cuando se utiliza como entrada, puede leer un estado lógico (por ejemplo, alto o bajo) y tomar una acción en consecuencia. Cuando se utiliza como salida, puede enviar una señal lógica para controlar dispositivos externos, como LED, relés, sensores, etc.

Los pines GPIO son muy versátiles y se utilizan en una amplia gama de aplicaciones, desde el control de motores hasta la lectura de sensores y la comunicación con otros dispositivos.

#### **MARCO TEÓRICO**

##### **TIVA TM4C123GH6PM**

Es un microcontrolador de 32 bits basado en el procesador ARM Cortex-M4 y ofrece una amplia gama de características para aplicaciones embebidas, incluyendo una memoria flash de 256 KB y una memoria RAM de 32 KB.

El TM4C123GH6PM es un microcontrolador de bajo costo y de baja potencia que es adecuado para una amplia gama de aplicaciones, incluyendo sistemas de control de motores, sistemas de automatización industrial, sistemas de monitoreo de sensores y sistemas de control de procesos.

El microcontrolador incluye una amplia gama de periféricos integrados, como

UART, SPI, I2C, ADC, PWM, timers, entre otros. Además, también ofrece soporte para una amplia gama de sistemas operativos embebidos, como FreeRTOS, embOS y Keil RTX5.

TM4C123GH6PM es un microcontrolador versátil y de bajo costo que ofrece una amplia gama de características y funcionalidades para aplicaciones embebidas.

## Máquina de estados

La máquina de estados es un modelo de programación utilizado para describir el comportamiento de un sistema dinámico que puede encontrarse en diferentes estados y cambiar entre ellos en respuesta a eventos externos.

Un estado es una representación de un conjunto de condiciones o valores que describen el comportamiento o funcionamiento del sistema en un momento dado. La máquina de estados describe cómo el sistema reacciona a los eventos y cambia de un estado a otro.

La máquina de estados es un modelo útil para describir sistemas complejos y se utiliza en una amplia gama de aplicaciones, incluyendo sistemas de control de procesos, robótica, sistemas de automatización y programación de microcontroladores.

## METODOLOGÍA

Se implementó el siguiente código de programación en Visual Studio:

```
#include "lib/include.h"
extern void Configurar_GPIO(void)
{
    volatile uint32_t delay= SYSCTL->RCGCGPIO; .....// no need to unlock;
    SYSCTL->RCGCGPIO |= 0x12; .....// Puertos b y E (00010010)
    //GPIO se hace para cada puerto
    //Entradas 0, salidas 1. DIR--DATA DIRECTION
    GPIOB->DIR |= 0x3F; .....// salidas en el puerto b, desde PB0-PB5 (00111111=0x3F)
    GPIOE->DIR &= ~0x03; .....// Entradas en el puerto E, desde PE0-PE1 (00000011=0x03) Y va n

    //DEN--digital enable 0-funcion digital deshabilitada 1-func dig habilitada
    GPIOB->DEN |= 0x3F; .....// enable digital on PB5-0 (00111111=0x3F)
    GPIOE->DEN |= 0x03; .....// enable digital on PE1-0 (00000011=0x03)

}
```

Figura 1. Código en Visual Studio

```
#include "lib/include.h"

#define SENSOR ((volatile uint32_t *)0x4002400C)
#define LIGHT ((volatile uint32_t *)0x400050FC)
// Linked data structure

struct State {
    uint32_t Out;
    uint32_t Time;
    uint32_t Next[4];
};

typedef const struct State State_t;
#define goN 0
#define waitN 1
#define goE 2
#define waitE 3

State_t FSM[4]={
    {0x21, 300, {goN, waitN, goN, waitN}},
    {0x22, 50, {goE, goE, goE, goE}},
    {0x0C, 300, {goE, goE, waitE, waitE}},
    {0x14, 50, {goN, goN, goN, goN}};
uint32_t S; // index to the current state
uint32_t Input;

int main(void){
    PLL_Config(); // 80-MHz, Program 10.1
    SysTick_Init(); // Program 10.2
    Configurar_GPIO();
    S = goN;
    while(1){
        LIGHT = FSM[S].Out; // set lights
```

```
31     while(1){
32         LIGHT = FSM[S].Out; // set lights
33         SysTick_Wait10ms(FSM[S].Time);
34         Input = SENSOR; // read sensors
35         S = FSM[S].Next[Input];
36     }
37 }
```

Figura 2. Código en Visual Studio

Se implementó el siguiente circuito:

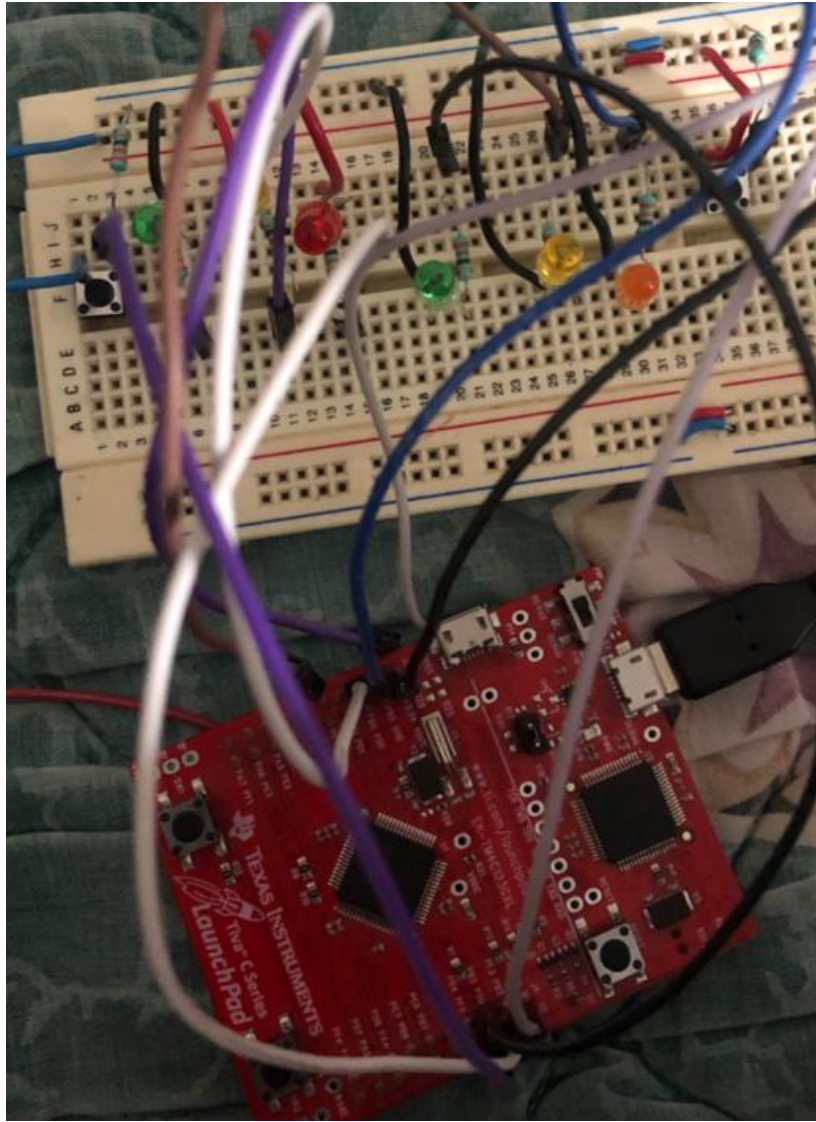


Figura 3. Circuito

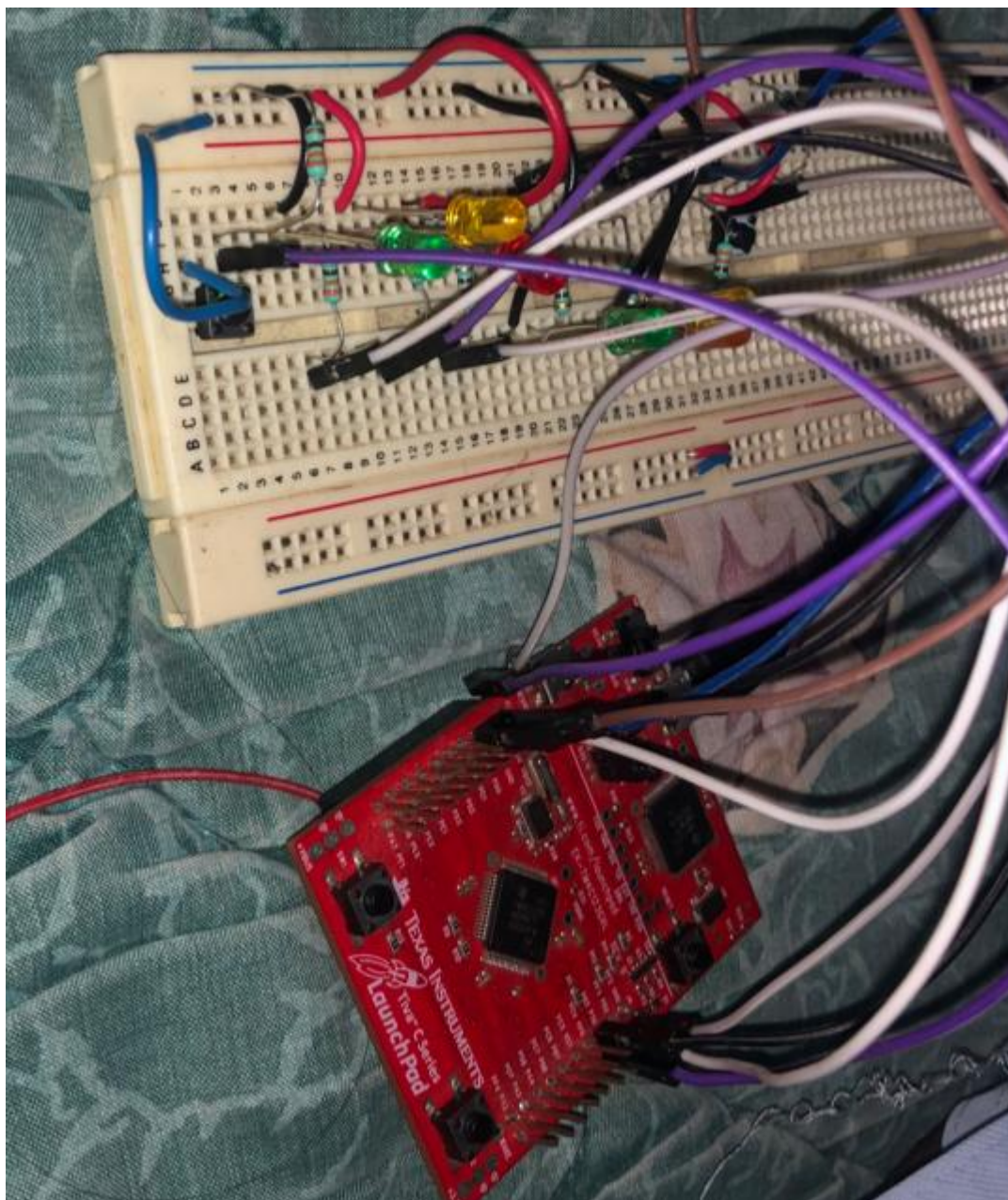


Figura 4. Circuito



## **CONCLUSIÓN**

La implementación del GPIO es importante ya que los pines GPIO permiten que un sistema electrónico interactúe con el mundo exterior a través de entradas y salidas digitales. Pueden utilizarse para controlar dispositivos externos como LED, motores, sensores, etc. o para recibir información de estos dispositivos.

Además, los pines GPIO pueden ser programados para cumplir una amplia variedad de funciones y ajustarse a diferentes requisitos, lo que los hace muy versátiles y útiles en una amplia gama de aplicaciones.