

# Cristopher Sagastume 18640

## Experimento 1: Tiva C TivaWare

### Utilizando la plataforma Tiva C

Este experimento servirá para que el estudiante se familiarice con la configuración de los pines GPIO de la tiva. La idea es que utilice los Leds y Switches que trae la Tiva utilizando programación en C con la ayuda de la librería DriverLib de TivaWare. Si no tuviera la Tiva C, podrá simularlo observando el comportamiento de los registros correspondientes.

#### Parte1

Siga todos los pasos de la Guía de Instalación de Code Composer y TivaWare, si ya tiene instalado Eclipse sáltese los pasos de instalación de Code Composer.

#### Parte 2.

#### Configure el reloj del sistema.

Para más información sobre el reloj puede irse a partir de la página 219 de la hoja de datos de la Tiva C en la sección del Reloj y verifique las tablas 5-4, 5-5 y 5-6 para ver las posibles opciones del reloj del sistema. Pruebe configurar el reloj del sistema utilizando el PLL a varias frecuencias.

#### Configure el reloj para habilitar el puerto F

Para más información puede irse a la página 340 de la hoja de datos y observe las opciones del registro RCGCGPIO. Observe también que otros registros necesitaría para habilitar los siguientes periféricos.

Periféricos	Registro necesario
Timers	SRTIMER
UARTs	SRUART
USB	SRUSB
PWM	SRPWM
ADC	RCGCADC ó SCGCADC

Configure los pines de los Leds Rojo, Verde y Azul como salidas.

### Parte 3.

De primero pruebe encender y lograr hacer combinación de colores con los 3 Leds. Haga una rutina en la cual pueda simular el comportamiento de un semáforo. (Rojo - Amarillo - Verde - Verde parpadeante). Pruebe utilizar máscaras para el encendido y apagado de cada Led utilizando operadores Bit operands.

Programe una rutina que genere un delay para poder observar el cambio de cada uno de los comportamientos.

Observe si existe alguna función que pueda utilizar para generar el delay sin utilizar el algoritmo que acaba de crear anteriormente.

Función	Parámetros
SysCtlDelay();	El número de loops que debe realizar

### Parte 4.

Configure un botón para iniciar el semáforo, empezando desde Verde, pase a verde parpadeante, amarillo y por último quede en el color rojo y se pueda reiniciar con el mismo botón.

Tome en cuenta que puede configurar el pin de entrada para utilizar los weak pull-ups. Configure el botón de esta forma, e implemente un algoritmo de anti debounce para ese botón. Puede encontrar más información en la página 264 del API de la librería Driverlib de TivaWare.

### Pseudocodigo:

- Configurar Reloj del sistema

- Configurar el puerto F, tanto los botones como los led
- Verificar si se presiono el botón
  - Si se presiono el botón iniciar el semáforo de Verde-Verde parpadeante-Amarillo-Rojo.
  - Si no se presiono no pasara nada.

### Código Comentado

```
//Cristopher Sagastume 18640

#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/pin_map.h"
#include "driverlib/debug.h"
#include "driverlib/gpio.h"

volatile uint32_t i;

int main (void){
    //Se configura el reloj a 40Mhz
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);

    //Se habilita el boton izquierdo de la tiva como input pull-up a 2mA
    GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_4);
    GPIOPadConfigSet(GPIO_PORTF_BASE, GPIO_PIN_4, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU);
    //Se habilitan los led como salidas (Puerto F)
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_3);

    while(1){
        //Se verifica si el botón fue presionado
        if (!GPIOPinRead(GPIO_PORTF_BASE,GPIO_PIN_4)){
            //empieza el ciclo encendiendo el led verde con la mascara 0x08

            GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x08);
            //Se realiza una iteracion como sustituto a un delay
            for (i=0;i<2000000;i++){

                //se apaga el led verde

            }
            GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x00);
        }
    }
}
```

```
        for (i=0;i<2000000;i++){  
  
            //Se enciende el led verde para que parpadee  
  
            GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x08);  
            for (i=0;i<1000000;i++){  
  
                GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x00);  
                for (i=0;i<1000000;i++){  
  
                    GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x08);  
                    for (i=0;i<2000000;i++){  
  
                        GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x00);  
                        for (i=0;i<2000000;i++){  
  
                            //Se enciende el led amarillo con la mascara 0x0A  
  
                            GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x0A);  
                            for (i=0;i<2000000;i++){  
  
                                //Se apaga el led amarillo  
  
                                GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x00);  
                                for (i=0;i<2000000;i++){  
  
                                    //Se enciende el led rojo con la mascara 0x02  
  
                                    GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3|GPIO_PIN_2|GPIO_PIN_1,0x02);  
                                    for (i=0;i<2000000;i++){  
  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

### Repositorio:

[https://github.com/sag18640/Electronica\\_Digital\\_2.git](https://github.com/sag18640/Electronica_Digital_2.git)