

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.

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	RCGCWTIMER
UARTs	RCGCUART
USB	RCGCUSB
PWM	RCGCPWM
ADC	RCGCADC

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.

Programa 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()	uint32_t, ui32Count

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.

Código de ejemplo:

```
#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"

int main(void)
{
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|
SYSCTL_OSC_MAIN);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|
GPIO_PIN_3);
    while(1){
        GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x08);
        SysCtlDelay(20000000);
        GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(20000000);
    }
}
```