

Lab 9 - Tiva C en TivaWare

Este experimento servirá para que el estudiante se familiarice con la configuración de los pines GPIO, ISR y Timers de la Tiva C. La idea es que utilice los Leds y Switches que trae la Tiva utilizando programación en C con la ayuda de la librería de Tivaware DriverLib o CMSIS

Pre Lab (20%) - Circuito Físico

*Se debe entregar antes del inicio del laboratorio. Se sube en canvas en formato *.zip con el nombre prelab.*

Intalación de Code Composer

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.

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	RCGCTIMER
UARTs	RCGCUART
USB	RCGCUSB
PWM	RCGCPWM
ADC	RCGCADC

Pruebas

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

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.

Delay

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()	cantidad de ciclos que se ejecutaran

Lab (30%) - Circuito Físico

Se entrega durante el tiempo del laboratorio. Deberá mostrarlo al catedrático o auxiliar para tener una nota.

Semáforo

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.

Configuración con registros

Ahora pruebe realizar el mismo ejercicio, pero ahora utilizando los registros, puede incluir el archivo header "TM4C123GH6PM.h".

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);
    }
}
```

Post Lab (40%) - Circuito Físico

*Se entrega después del tiempo de laboratorio según el portal y debe funcionar con todas las partes anteriores. Deberá subir los entregables en formato *.zip con el nombre entregables.*

Copie las siguientes líneas para incluir los archivos header necesarios

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/tm4c123gh6pm.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "inc/hw_ints.h"
#include "driverlib/sysctl.h"
#include "driverlib/interrupt.h"
#include "driverlib/gpio.h"
#include "driverlib/timer.h"
```

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. Configure el reloj a 40 MHz.

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.

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

Configure el Timer 0

Configure el Timer 0, como un timer de 32-bits, en forma periódica, para ver más información sobre el timer irse a partir de la página 704 de la hoja de datos de la Tiva C.

Toggle

Cree una rutina en la cual haga un toggle de un GPIO a 1Hz con un duty cycle del 50%.

Interrupción

Configure una interrupción a la mitad del periodo de la parte 3, es decir, a 0.5Hz.

Recuerde habilitar todas las interrupciones utilizando la instrucción `IntMasterEnable()`. Cree un Handler para la interrupción del Timer

```
void Timer0IntHandler(void){  
  
}
```

Recuerde irse al archivo `tm4c123gh6pm_startup_ccs.c` y buscar donde está el comentario `// Timer 0 subtimer A`

Reemplazar `IntDefaultHandler`, por el nombre del handler de su función del Timer 0, en este caso sería **`Timer0IntHandler`**,

También tendrá que declarar la función al principio de este archivo de forma externa.

Buscar la línea donde se encuentra
`extern void _c_int00(void);`

y pegue el nombre de su función:

```
extern void Timer0IntHandler(void);
```

Entregables (10%)

Deberá entregar en Canvas un archivo comprimido que contenga:

- El folder completo de su proyecto
- Link a Video mostrando el funcionamiento y explicando su código (lo suben a YouTube/Odysee/Vimeo y lo ponen Unlisted o Público).
- Su código debe estar dividido en secciones y bien comentado
- Link a repositorio en Github