

Experimento 2: Tiva C ISRs Timers y UART

Utilizando la plataforma Tiva C

Este experimento servirá para que el estudiante se familiarice con la configuración de las ISR, Timers y el módulo UART. Utilizando programación en C con la ayuda de la librería DriverLib de TivaWare o CMSIS. Si no tuviera la Tiva C, podrá simularlo observando el comportamiento de los registros correspondientes.

Parte1

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"
```

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. 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.

Parte 3

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

Parte 4

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

Configure la Interrupción

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);
```

Parte 5.

Habilitar el módulo UART0 y los periféricos del GPIOA correspondientes al módulo UART.

Inicialice el módulo UART con los siguientes parámetros: 115200, 8 data bits, 1 stop bit, None parity.

Parte 6.

Habilite la interrupción para el módulo UART.

Recuerde irse al archivo *tm4c123gh6pm_startup_ccs.c* y buscar donde está el comentario
// UART0 Rx and Tx

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

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 UARTIntHandler(void);

Parte 7.

Haga una rutina donde recopile un carácter desde el módulo UART utilizando la interrupción.

Con esta rutina, utilice las letras "r", "g" y "b", recibidos desde el módulo UART para habilitar/deshabilitar el toggle de cada uno de los leds respectivos de la Tiva C.

Es decir, si recibe la letra "r", habilita el toggle del led rojo, si vuelve a enviar la letra "r" lo deshabilita.