/\*

\* Rochester Institute of Technology

\* Department of Computer Engineering

\* CMPE 460 Interfacing Digital Electronics

\* LJBeato

\* 1/14/2021

\*

\* Filename: main\_timer\_template.c

\*/

#include <stdio.h>

#include <stdlib.h>

#include "msp.h"

#include "uart.h"

#include "Lab1.c"

#include "Timer32.h"

#include "CortexM.h"

#include "Common.h"

extern uint32\_t SystemCoreClock;

// these are not used by the timer

BOOLEAN g\_sendData = FALSE;

uint16\_t line[128];

int colorIndex = 0;

#define RED BIT0

#define GREEN BIT1

#define BLUE BIT2

#define CYAN BIT2 | BIT1

#define MAGENTA BIT0 | BIT2

#define YELLOW BIT1 | BIT2

#define WHITE BIT0 | BIT1 | BIT2

BYTE colors[7] = { RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, WHITE };

BOOLEAN Timer1RunningFlag = FALSE;

BOOLEAN Timer2RunningFlag = FALSE;

unsigned long MillisecondCounter = 0;

//

// I/O interrupt pin setup

//

// DIR SEL0/SEL1 IE IES Port Mode

// 0 00 0 0 Input, rising edge trigger

// 0 00 0 1 Input, falling edge trigger, interrupt

// 0 00 1 0 Input, rising edge trigger, interrupt

// 0 00 1 1 Input, falling edge trigger, interrupt

//

void Switch1\_Interrupt\_Init(void)

{

// disable interrupts

DisableInterrupts();

// initialize the Switch as per previous lab

Switch1\_Init();

//7-0 PxIFG RW 0h Port X interrupt flag

//0b = No interrupt is pending.

//1b = Interrupt is pending.

// clear flag1 (reduce possibility of extra interrupt)

P1->IFG=0;

//7-0 PxIE RW 0h Port X interrupt enable

//0b = Corresponding port interrupt disabled

//1b = Corresponding port interrupt enabled

// arm interrupt on P1.1

P1->IE=BIT1;

//7-0 PxIES RW Undefined Port X interrupt edge select

//0b = PxIFG flag is set with a low-to-high transition.

//1b = PxIFG flag is set with a high-to-low transition

// now set the pin to cause falling edge interrupt event

// P1.1 is falling edge event

P1->IES=BIT1;

// now set the pin to cause falling edge interrupt event

NVIC\_IPR8 = (NVIC\_IPR8 & 0x00FFFFFF)|0x40000000; // priority 2

// enable Port 1 - interrupt 35 in NVIC

NVIC\_ISER1 = 0x00000008;

// enable interrupts (// clear the I bit )

EnableInterrupts();

}

void Switch2\_Interrupt\_Init(void)

{

// disable interrupts

DisableInterrupts();

// initialize the Switch as per previous lab

Switch2\_Init();

// now set the pin to cause falling edge interrupt event

// P1.4 is falling edge event

P1->IE=BIT4;

// clear flag4 (reduce possibility of extra interrupt)

;

// arm interrupt on P1.4

;

// now set the pin to cause falling edge interrupt event

NVIC\_IPR8 = (NVIC\_IPR8&0x00FFFFFF)|0x40000000; // priority 2

// enable Port 1 - interrupt 35 in NVIC

NVIC\_ISER1 = 0x00000008;

// enable interrupts (// clear the I bit )

EnableInterrupts();

}

// PORT 1 IRQ Handler

// LJBeato

// Will be triggered if any pin on the port causes interrupt

//

// Derived From: Jonathan Valvano

void PORT1\_IRQHandler(void)

{

float numSeconds = 0.0;

char temp[32];

// First we check if it came from Switch1 ?

if(P1->IFG & BIT1) // we start a timer to toggle the LED1 1 second ON and 1 second OFF

{

// acknowledge P1.1 is pressed, by setting BIT1 to zero - remember P1.1 is switch 1

// clear flag, acknowledge

;

}

// Now check to see if it came from Switch2 ?

if(P1->IFG & BIT4)

{

// acknowledge P1.4 is pressed, by setting BIT4 to zero - remember P1.4 is switch 2

; // clear flag4, acknowledge

}

}

//

// Interrupt Service Routine for Timer32-1

//

//

//

void Timer32\_1\_ISR(void)

{

if (LED1\_State() == FALSE )

{

LED1\_On();

}

else LED1\_Off();

}

//

// Interrupt Service Routine

//

//

//

void Timer32\_2\_ISR(void)

{

MillisecondCounter++;

}

//

// main

//

//

//

int main(void){

//initializations

uart0\_init();

uart0\_put("\r\nLab5 Timer demo\r\n");

// Set the Timer32-2 to 2Hz (0.5 sec between interrupts)

//TIMER32\_2->VALUE=.5;

//Timer32\_1\_Init(&Timer32\_1\_ISR, SystemCoreClock/2, T32DIV1); // initialize Timer A32-1;

;

// Setup Timer32-2 with a .001 second timeout.

// So use DEFAULT\_CLOCK\_SPEED/(1/0.001) = SystemCoreClock/1000

//Timer32\_2\_Init(&Timer32\_2\_ISR, SystemCoreClock/1000, T32DIV1); // initialize Timer A32-1;

;

Switch1\_Interrupt\_Init();

Switch2\_Interrupt\_Init();

LED1\_Init();

LED2\_Init();

EnableInterrupts();

while(1)

{

WaitForInterrupt();

}

}