



Faculty of Engineering
Ain Shams University
CSE 211s: Introduction to Embedded
Systems

REPORT (6)

Name: Habiba El-Sayed Mowafy

Sec.: 1

Program: CSE

ID: 2100792

Question 3:

Q3. Write Embedded C program to increment “seconds” variable each 1 sec and toggle Green LED in SysTick_Handler(). The program also pauses the SysTick timer and RED LED is turned on when SW1 is pressed, while it resumes the SysTick timer and the RED LED is turned off when SW2 is pressed.

Upon starting the program, all the LEDs should be turned off. Assume the SysTick timer operates on 16 MHZ and its interrupt has priority of 1, while the priority of GPIOF interrupt has priority of 2.

C code:

```
#include "tm4c123gh6pm.h"
#include <stdint.h>
#define LED_RED (1U << 1)
#define LED_GREEN (1U << 3)
#define CLOCK 16000000

int seconds=0;

void SystemInit (void) {
    SYSTCTL_RCGCGPIO_R |= 0x20; // PortF clock enable
    while ((SYSTCTL_PRGPIO_R & 0x20)==0); //Delay
    GPIO_PORTF_LOCK_R= GPIO_LOCK_KEY; // Unlock PortF Commit register
    GPIO_PORTF_CR_R = 0x1F;
    GPIO_PORTF_AMSEL_R = 0x00; // Disable analog function
    GPIO_PORTF_PCTL_R = 0x00000000; // GPIO clear bit PCTL
    GPIO_PORTF_AFSEL_R = 0x00; // No alternate function
    GPIO_PORTF_DIR_R = 0x0E; // PF321 output
    GPIO_PORTF_DEN_R = 0x1F; // Enable digital pins PF4-PFO
    GPIO_PORTF_PUR_R = 0x11;
    GPIO_PORTF_IS_R = ~0x11;
    GPIO_PORTF_IBE_R = ~0x11;
    GPIO_PORTF_IEV_R = ~0x11;
    GPIO_PORTF_IM_R = 0x11;
    NVIC_PRI7_R|=0x00400000;
    NVIC_EN0_R |=0x40000000;
}

void GPIOF_Handler() {
    if (GPIO_PORTF_MIS_R & 0x01) {
        GPIO_PORTF_DATA_R &= ~LED_RED;
        GPIO_PORTF_ICR_R |= 0x01;
        NVIC_ST_CTRL_R=0x7;
    }
}
```

```
        else if (GPIO_PORTF_MIS_R & 0x10) {
            GPIO_PORTF_DATA_R |= LED_RED;
            GPIO_PORTF_ICR_R |= 0x10;
            NVIC_ST_CTRL_R=0x6;
        }
    }

void SysTick_Handler() {
    seconds++;
    GPIO_PORTF_DATA_R ^= LED_GREEN;
}

int main() {
    SystemInit();
    NVIC_ST_CURRENT_R=0;
    NVIC_ST_RELOAD_R=CLOCK-1;
    NVIC_ST_CTRL_R=0x7;
    __enable_irq();
    while (1){}
```

Snapshots:

Program Started, all LEDs are OFF

The screenshot shows the TI-RTOS IDE with the following components:

- Registers:** Shows the Core registers. The PC register (R15) is at 0x00000047.
- Disassembly:** Shows the main function starting at address 0x00000047. The code is:

```
46 int main() {  
47     SystemInit();  
48     NVIC_ST_CURRENT_R=0;  
49     NVIC_ST_RELOAD_R=CLOCK-1;  
50     NVIC_ST_CTRL_R=0x7;  
51     enable_irq();  
52     while (1){  
53     }  
54 }
```
- GPIOF:** Shows the initial values of the GPIO registers. The DIR register is 0x00000000.
- Port F Hardware:** Shows the initial state of the Port F registers. The DIR register is 0x00000000.
- System Tick Timer:** Shows the initial state of the timer. The ST_CURRENT register is 0x00000000.

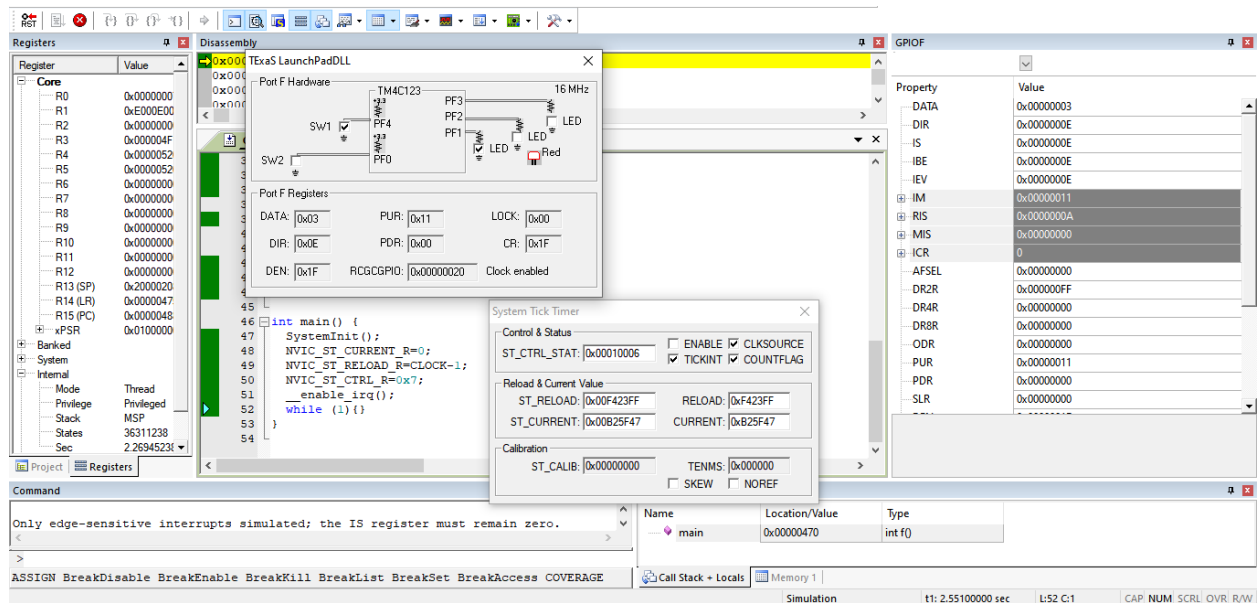
First Toggle and green LED turned ON

The screenshot shows the TI-RTOS IDE with the following components:

- Registers:** Shows the Core registers. The PC register (R15) is at 0x00000047.
- Disassembly:** Shows the main function starting at address 0x00000047. The code is:

```
46 int main() {  
47     SystemInit();  
48     NVIC_ST_CURRENT_R=0;  
49     NVIC_ST_RELOAD_R=CLOCK-1;  
50     NVIC_ST_CTRL_R=0x7;  
51     enable_irq();  
52     while (1){  
53     }  
54 }
```
- GPIOF:** Shows the updated values of the GPIO registers. The DIR register is 0x00000000.
- Port F Hardware:** Shows the updated state of the Port F registers. The DIR register is 0x00000000.
- System Tick Timer:** Shows the updated state of the timer. The ST_CURRENT register is 0x00000000.

Red LED is ON & timer is disabled after clicking switch 1



Red LED turned OFF and timer enabled again after clicking switch 2

