San Francisco State University

Engr 478: Design with Microprocessors

Lab 3

By Raymond Reed

Date: March 04 2020

Code:

Question 1

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"

#define                 red_MASK        0x02
//*****************************************************************************
//
//!
//! A very simple example that interfaces with the blud LED (PF2) and SW2 (PF0)
//! using direct register access. When SW2 is pressed, the LED is turned on. When
//! SW2 is released, the LED is turned off.
//
//*****************************************************************************


void
PortFunctionInit(void)
{

                volatile uint32_t ui32Loop;


                // Enable the clock of the GPIO port that is used for the on-board
LED and switch.
   SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;
```

```
    //
    // Do a dummy read to insert a few cycles after enabling the peripheral.
    //
    ui32Loop = SYSCTL_RCGC2_R;

                                    // Unlock GPIO Port F
                                    GPIO_PORTF_LOCK_R = 0x4C4F434B;
                                    GPIO_PORTF_CR_R |= 0x01;          // allow changes to PF0

    // Set the direction of PF2 (blue LED) as output
                                    // a change
    GPIO_PORTF_DIR_R |= 0x02;



                                    // Set the direction of PF0 (SW2) as input by clearing the bit
    GPIO_PORTF_DIR_R &= ~0x01;

    // Enable both PF2 and PF0 for digital function.
    GPIO_PORTF_DEN_R |= 0x03;

                                    //Enable pull-up on PF0
                                    GPIO_PORTF_PUR_R |= 0x01;

}



int main(void)
{
```

```
                    //initialize the GPIO ports
                    PortFunctionInit();

    //
    // Loop forever.
    //
    while(1)
    {

        if((GPIO_PORTF_DATA_R&0x01)==0x00) //SW2 is pressed
                                            {
                                                    // Turn off the LED.
                                                    GPIO_PORTF_DATA_R &=
~0x02;
                                            }
                                            else
                                            {

                                                // Turn on the LED.
                                                    GPIO_PORTF_DATA_R |= 0x02;
                                            }
    }
}
```

Question 2

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/tm4c123gh6pm.h"
#include "driverlib/sysctl.h"

#define                         Green_MASK              0x08
//*****************************************************************************
*
//
//!
//! A very simple example that toggles the on-board red LED using direct register
//! access.
//
//*****************************************************************************
*



void
PortFunctionInit(void)
{
//
                                    volatile uint32_t ui32Loop;
                                   // Enable the GPIO port that is used for the on-board LED.
    //
    SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;


    //
    // Do a dummy read to insert a few cycles after enabling the peripheral.
```

```
    //
    ui32Loop = SYSCTL_RCGC2_R;

    //
    // Enable the GPIO pin for the green LED (PF1).  Set the direction as output, and
    // enable the GPIO pin for digital function.
    //
    GPIO_PORTF_DIR_R |= 0x08;
    GPIO_PORTF_DEN_R |= 0x08;

}


int main(void)
{
                                    //initialize the GPIO ports
                                    PortFunctionInit();

    // Turn on the LED.
    GPIO_PORTF_DATA_R |= 0x08;


    //
    // Loop forever.
    //
    while(1)
    {
```

```
        // Delay for a bit.
                                                SysCtlDelay(10000000);

        // Toggle the LED.
        GPIO_PORTF_DATA_R ^=Green_MASK;
    }
}
```

Block Diagram

Diagram#1

Diagram#2