Andrew Pham, 9179-19969

Moshe Stanylov

Problem #1

Port

Initiation

SW2 Not pressed?

Yes No

LED Off

LED On

define MASK 0x02 //to change to diff LED color, just change mask value.

// Set the direction of PF1 (red LED) as output

GPIO\_PORTF\_DIR\_R |= MASK;

// Enable both PF1 and PF0 for digital function.

GPIO\_PORTF\_DEN\_R |= MASK + 0x01;

Change loop 🡪 if SW2 not pressed

Yes 🡪 LED on

No 🡪 LED off

#include <stdint.h>

#include "inc/tm4c123gh6pm.h"

//variable MASK = LED pin position. 0x02 = RED, 0x04=BLUE and 0x08 = GREEN

#define MASK 0x02

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 LED as output

GPIO\_PORTF\_DIR\_R |= MASK;

// Set the direction of PF0 (SW2) as input by clearing the bit

GPIO\_PORTF\_DIR\_R &= ~0x01;

// Enable both LED and PF0 for digital function.

GPIO\_PORTF\_DEN\_R |= MASK + 0x01;

//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)==0x01) //SW2 is NOT pressed

{

// Turn on the LED.

GPIO\_PORTF\_DATA\_R |= MASK;

}

else

{

// Turn off the LED.

GPIO\_PORTF\_DATA\_R &= ~MASK;

}

}

}

//Observation: Program works as expected. By using var MASK for LED, changing LED selection is much simpler.

Problem #2

Activate clock of PortF

Configure PF3 (green LED) as output

PF3 = 1 🡪 ON (positive logic)

Delay for 5X slower

Toggle PF3

#include <stdint.h>

#include <stdbool.h>

#include "inc/tm4c123gh6pm.h"

#include "driverlib/sysctl.h"

//Define MASK as LED variable. In this case it's 0x08 or GREEn LED

#define MASK 0x08

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 red LED (PF1). Set the direction as output, and

// enable the GPIO pin for digital function.

GPIO\_PORTF\_DIR\_R |= MASK;

GPIO\_PORTF\_DEN\_R |= MASK;

}

int main(void)

{

//initialize the GPIO ports

PortFunctionInit();

// Turn on the LED.

GPIO\_PORTF\_DATA\_R |= MASK;

// Loop forever.

while(1)

{

// Delay for a bit.

SysCtlDelay(10000000);

// Toggle the LED.

GPIO\_PORTF\_DATA\_R ^= MASK;

}

}

Observation:

Void SysCtlDelay(uint32\_t ui32Count)

Where ui32Count is the number of delay loop iterations to perform. This function provides a means of generating a delay by executing a simple 3 instruction cycle loop a given number of times.

The previous delay count was 2,000,000. To increase the delay by 5X, we changed it to 10,000,000.

The TM4C123G is 80MHz microcontroller, so each clock cycle is 10ns. For delay count of 10,000,000 and assuming that each instruction takes 1 clock cycle, the LED on/off is delayed by 10,000,000 x 3 x 10E-9 = 0.375 sec

Similar to problem #1, we defined MASK as LED variable to allow us to change to different LED by just changing the MASK’s value.