Lab 5: GPIOs, Switches and LED interfaces

Farnam Adelkhani

[FAdelkha@mail.sfsu.edu](mailto:FAdelkha@mail.sfsu.edu)

915815724

**System Specifications:**

The system setup by our group utilizes a switch to turn the blue led on and off while simultaneously toggling the red led. The system is going to require configuration of ports SW1(PF4), PF2 (blue LED) and PF1(Red LED). When SW1(PF4) is pressed the red LED(PF1) will turn on and toggle based on the specified delay, the length of the delay and customization is something we will discuss in the next paragraph. When the switch is released the red LED should turn off and the blue LED should begin to toggle instead.

We will now discuss how to generate a specific delay. First the user must take into account the clock frequency being used, in this situation the default of 16 Mhz is being used. In order to calculate a specific delay we use the SysCtlDelay() command and within the parenthesis we specify a factor to multiply or divide the system delay by, to get a desired delay time. Each loop takes three cpu cycles. If we wanted to generate a specific delay we could use the following formula.

(time)s = [1/(Default Clock)] x (# of loops) x 2000000 loops.

**Flow Chart:**

**Code:**

#include <stdint.h>

#include <stdbool.h>

#include "lab5proj.h"

#include "inc/hw\_types.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_gpio.h"

#include "driverlib/sysctl.h"

#include "driverlib/pin\_map.h"

#include "driverlib/gpio.h"

#include "inc/tm4c123gh6pm.h"

//!need to define the led masks and hex values

#define     RED\_MASK     0x02

#define     BLUE\_MASK      0x04

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void

PortFunctionInit(void)

{

//

// Enable Peripheral Clocks

//

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

//

// Enable pin PF2 for GPIOOutput

//

GPIOPinTypeGPIOOutput(GPIO\_PORTF\_BASE, GPIO\_PIN\_2);

//

// Enable pin PF1 for GPIOOutput

//

GPIOPinTypeGPIOOutput(GPIO\_PORTF\_BASE, GPIO\_PIN\_1);

//

// Enable pin PF4 for GPIOInput

//

GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE, GPIO\_PIN\_4);

   //!Enable pull-up on PF4

   //!since the switches are negative logic, it requires

   //!activation of the internal pull-up resistors

   GPIO\_PORTF\_PUR\_R |= 0x10;

}

int main(void)

{

   PortFunctionInit();

   while (1) //necessary to keep checking for actions

   {

   // Turn on the BLUE LED.

GPIO\_PORTF\_DATA\_R |= 0x04;

   // Turn off the RED LED.

GPIO\_PORTF\_DATA\_R &=~ 0x02;

   if((GPIO\_PORTF\_DATA\_R&0x10)==0x00) //checks if SW1 is pressed

   {

   while(1) //necessary to check for actions while the SW1 button is pressed

   {

   if((GPIO\_PORTF\_DATA\_R&0x10)==0x00)

//the if-else statement that verifies the action

   //that is taking place with our toggle functions

   {

   // Turn off the BLUE LED.

   GPIO\_PORTF\_DATA\_R &=~ 0x04;

   //SysCtlDelay(2000000\*(4/3));

   // Toggle the RED LED.

   GPIO\_PORTF\_DATA\_R ^=RED\_MASK;

   SysCtlDelay(2000000\*(4/3));

   }

   else

   {

   // Turn off the RED LED.

   GPIO\_PORTF\_DATA\_R &= ~0x02;

   //SysCtlDelay(2000000\*(4/3));

   // Toggle the BLUE LED.

   GPIO\_PORTF\_DATA\_R ^=BLUE\_MASK;

   SysCtlDelay(2000000\*(4/3));

   }

   }

   }

   }

}

**The execution results of program:**

As expected, the results of the program are as follow:

1 - The system begins w/ the blue LED ON and the red LED OFF.

2 - When SW1 is pressed, then the blue LED will turn off and the red LED will begin to

toggle about every ½ second

3 - When SW1 is released then the red LED will be off and the blue LED will be toggled

@ roughly every ½ second

… in order to verify the correctness of the system our group members both, individually, tested the program on their ti launchpads and were able to verify that the system is responding as expected. No supplemental imagining has been provided since a photo wouldn’t be able to capture the fact that the light is blinking anyways.

**Discussion and Suggestions:**

The purpose and goal of lab assignment five is to use the knowledge we have gained over the past few experiments to develop our own project from scratch to implement the required i/o settings indicated by the lab assignment. First step for our group members is to create a new project and then to use the tiva pinmux software in order to set and import the desired inputs and outputs. In previous lab assignments we used SW2 (PF0) to control the system but now we are asked to use SW1 (PF4) as the input to the system. The other task this lab is asking the user to master is combining the projects, toggle and switch into one project capable of both feats.

In this system we start with the blue led on and the red led off and so the SW1 button being pressed will result in the blue LED being turned off and the red LED will be toggled every half second. When SW1 is not pressed, then the red LED will be off and the blue LED will instead toggle at a rate of roughly half a second. This lab requires us to figure out which IO ports you need to configure and how to configure those ones in particular.

We also learned how to implement a specific delay by understanding that this device is working using a clock. If we can figure out how long one clock cycle takes then we are able to set a delay for the proper span of time.

This laboratory assignment is straightforward and we do not have any suggestions to improve this lab besides including more information about how to manipulate the device to delay for a specified amount of time.