Lab Report

EECS 3100:046

Embedded Systems Lab

Lab 5:

Switch and LED Interfacing in Hardware

Landon Jackson & Logan Crawfis

Professor Brent Nowlin

June 24th, 2021

Lab Objective

The purpose of this lab is to learn how to interface a switch and an LED. We will perform explicit measurements on the circuits to verify they are operational and to improve our understanding of how they work. We will also familiarize ourselves with electrical circuit schematic and the software to design them.

Introduction

Schematics and calculations performed when designing a system are important steps taken before transferring the final design into a hardware implementation. Switches and LEDs are both common components used in circuits for hardware interfacing. Hardware interfacing allows us to transfer signals between multiple devices, to allow control from external devices.

Project Description

A positive logic switch operating an LED will be designed, and debugged using the Tiva microcontroller and schematic drawing software. The switch will be connected to PE0 as input, and the LED will be connected to PE1 as an output. The switch will be used to toggle the LED causing it to blink at a rate of ~8Hz.

Design / Analysis Procedure

The specifications given to us initially served as a basis for the simulated circuit wiring. The schematic assisted us in building the circuit. Programming the board to utilize portE was simple since most of the program was completed in lab 3.Changes that were necessary were changing the clock, setting PE1 and PE0 as inputs and outputs, and digitally enabling the ports that were necessary. The handout specified that the LED should remain on while the button is not pressed, we achieved this by setting the 0th bit of PE0. The last change that needed to be made was decreasing the timing of the delay function, this was achieved by increasing the value of our delay function to achieve a delay of 62ms. When building the circuit we followed the circuit schematic, and built the circuit using a breadboard.

Discussion

The First part of the lab was to design a schematic for the LED and switch that was to be used for the output and input respectively. Using the data sheet provided for the LED the proper resistance was chosen for a forward voltage of 1.9 V and an operating current of 2 mA. The schematic can be seen below in Figure 1.

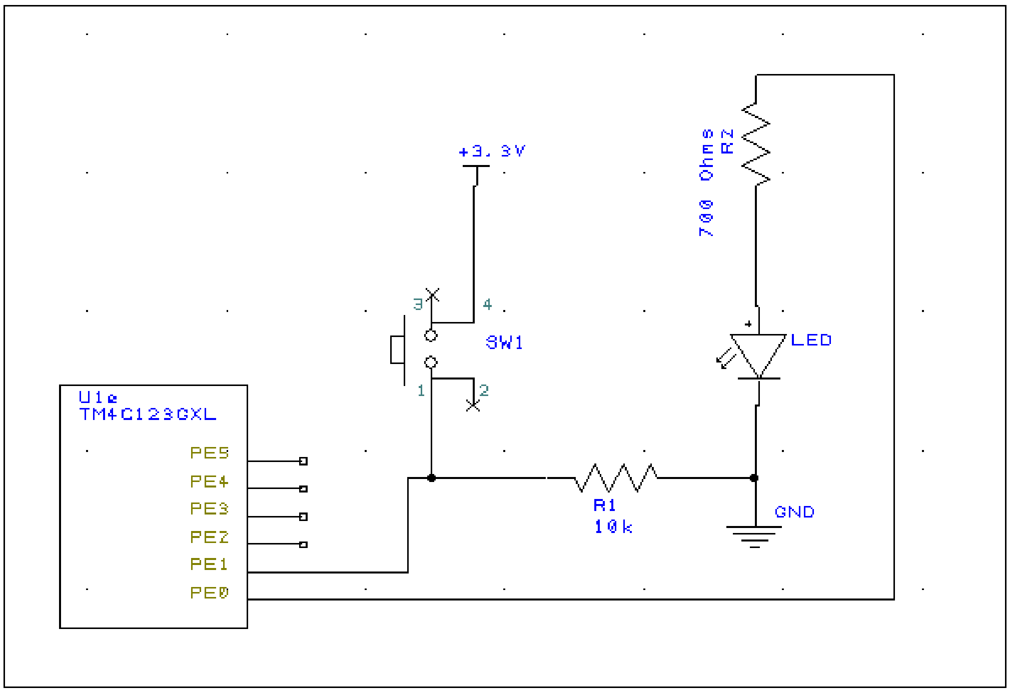

Figure 1. Schematic of LED and switch circuit

To interface with the LED and switch, most of the same software from lab 4 was used with a few minor changes: The LED was to be initialized high at the start of the program, the input and output were on PE1 and PE0 respectively, the delay for the toggle on the LED was now around 62 ms, as well as the change from negative to positive logic on the input. Prior to actually running the new program on the hardware, values from the circuit were measured using a multimeter. These values can be seen below in Tables 1 and 2.

Parameter	Measured Value	Units	Measurement Conditions
Resistance, R1	9800	ohms	With power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.29	volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}	0	volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor Current	0	mAs	Powered, but switch not pressed I=VPE1/R1 (calculated and measured with an ammeter)
Input Voltage, V_{PE1}	3.29	volts	Powered and with switch pressed (measured with voltmeter)
Resistor Current	0.336	mAs	Powered and switch pressed I=VPE1/R1 (calculated and measured with an ammeter)

Table 1. Switch measurements

Parameter	Measured Value	Units	Measurement Conditions
Resistance, R2	984	ohms	With power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$	3.29	volts	Powered (measured with voltmeter)
TM4C123 Output: V_{PE0} to LED k-/GND	0	volts	with PE0 = 0 (measured with voltmeter relative to ground)
LED voltage V_{a+}: a+ or bottom side of R2 to GND	0	volts	with PE0 = 0 (measured with voltmeter relative to ground)

R2 voltage	0	volts	calculated as V_{PE0} - V_{a+}
LED current (junction cut off)	0	mAs	calculated as (VPE0 - Va+)/R2 & measured with an ammeter
TM4C123 Output: V_{PE0} to LED k-/GND	3.15	volts	with PE0 = 1 (measured with voltmeter relative to ground)
LED voltage V_{a+}: a+ or bottom side of R2 to GND	1.89	volts	with PE0 = 1 (measured with voltmeter relative to ground)
R2 voltage	1.26	volts	calculated as V_{PE0} - V_{a+}
LED forward bias current	1.28	mAs	calculated as (V_{PE0} - V_{a+})/R2
LED forward bias current	2.99	mAs	measured with an ammeter

Table 2. LED measurements

```
led_on

    LDR     R4, [R3]
    ORR     R4,  R4, #0x01
    STR     R4, [R3]
    B        loop
```

Fig. 2: LED_ON Function

```
    MOV         R5, #3875
    B         Delay

Delay

    MOVS      R2, #0

sub_loop

    ADD      R2, #1
    CMP      R2, #255
    BNE          sub_loop
    SUB      R5, #1
    CMP      R5, #0
    BNE      Delay
    B          loop
```

Fig. 3: Delay Function

```
    LDR         R0, =SYSCTL_RCGCGPIO_R
    MOV         R1, #0x10
    STR         R1, [R0]

    NOP
    NOP

    LDR         R0, =GPIO_PORTE_AMSEL_R
    MOV         R1, #0x00
    STR         R1, [R0]

    LDR         R0, =GPIO_PORTE_DIR_R
    MOV         R1, #0x01
    STR         R1, [R0]

    LDR      R0, =GPIO_PORTE_AFSEL_R
    MOV      R1, #0x00
    STR         R1, [R0]

    LDR      R0, =GPIO_PORTE_DEN_R
    MOV      R1, #0x03
    STR         R1, [R0]

    LDR      R0, =GPIO_PORTE_PCTL_R
    MOV      R1, #0x00
    STR         R1, [R0]
```

Fig. 4: Port E initialization

Some of the major changes that needed to be implemented in this interfacing are shown in the figures above and below. Fig 3. shows the assembly code used to initialize Port E for use with the switch and LED. The function shown in Fig 4. Is the led_on function. The specifications required the LED to remain on until the button was pressed, by setting the 0th bit on PE0 the bit is now on and the light will remain on, before the led was to remain off, this was achieved by clearing the 0th bit on PE0. Figure 4 shows the improved implementation of our timing delay. The specifications changed the time of the delay from 120ms used in lab 4 to 62ms. Essentially the loop is a nested for-loop that creates the delay by manipulating arithmetic based clock cycles. The results of the program can be seen in Fig. 5, The logic analyzer clearly shows that when PE0 is low, PE1. Also when PE0 is high, PE1 can be seen toggling with a delay of 62ms. The lab was completed in its entirety, we had no issues completing any portion of the lab.

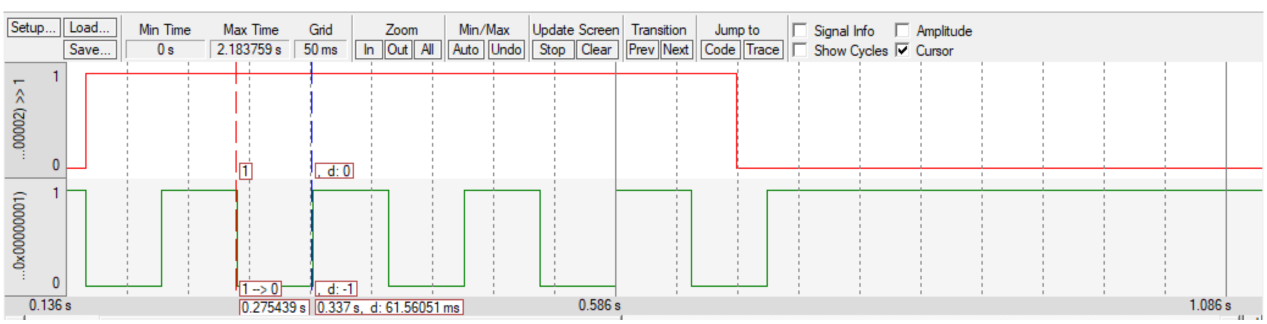

Figure 5. Logic Analyzer I/O

Conclusion

In lab 5, there was not as much programming required compared to previous lab experiments. The few changes were fairly easy to make and were executed without issues. Most of the work of this lab experiment was in the hardware. Using the provided switch and LED along with some resistors we had to make an external LED and switch interface. After some calculations the circuit was designed. Using the multimeter allowed for some verification that the circuit should work and not cause any damage to the electronics. This experiment was more interesting as it allowed for our inner EE to truly shine and allowed for a bit of a step away from the normal programming projects that we have been doing.

The work for this lab was evenly split between both partners. We worked together on the programming changes as well as the circuit design over discord, each making our own circuit and program, although the programs are identical. Then we split portions of the report between each other and wrote it together using Google Docs.

```
;****************************************************************
; main.s
; Author: Logan Crawfis & Landon Jackson
; Date Created: 06/22/2021
; Last Modified: 06/22/2021
; Section Number: 046
; Instructor: ***update this***
; Lab number: 5
; Brief description of the program
;   If the switch is presses, the LED toggles at 8 Hz
; Hardware connections
;  PE1 is switch input  (1 means pressed, 0 means not pressed)
;  PE0 is LED output (1 activates external LED on protoboard)
; Overall functionality is similar to Lab 4, with six changes:
;   1) the pin to which we connect the switch is moved to PE1,
;   2) you will have to remove the PUR initialization because
;      pull up is no longer needed.
;   3) the pin to which we connect the LED is moved to PE0,
;   4) the switch is changed from negative to positive logic, and
;   5) you should increase the delay so it flashes about 8 Hz.
;   6) the LED should be on when the switch is not pressed
; Operation
;   1) Make PE0 an output and make PE1 an input.
;   2) The system starts with the LED on (make PE0 =1).
;   3) Wait about 62 ms
;   4) If the switch is pressed (PE1 is 1), then toggle the LED
;      once, else turn the LED on.
;   5) Steps 3 and 4 are repeated over and over
;****************************************************************

GPIO_PORTE_DATA_R       EQU   0x400243FC
GPIO_PORTE_DIR_R        EQU   0x40024400
GPIO_PORTE_AFSEL_R      EQU   0x40024420
GPIO_PORTE_DEN_R        EQU   0x4002451C
GPIO_PORTE_AMSEL_R      EQU   0x40024528
GPIO_PORTE_PCTL_R       EQU   0x4002452C
SYSCTL_RCGCGPIO_R       EQU   0x400FE608
PE0                     EQU         0x40024004
PE1                     EQU         0x40024008

       IMPORT  TExaS_Init

       AREA    |.text|, CODE, READONLY, ALIGN=2
       THUMB
       EXPORT  Start
Start
        ; TExaS_Init sets bus clock at 80 MHz
      BL  TExaS_Init ; voltmeter, scope on PD3
      ; you initialize PE1 PE0

       LDR              R0, =SYSCTL_RCGCGPIO_R     ; set clock for PortE
       MOV              R1, #0x10
       STR        R1, [R0]
```

```
    NOP
    NOP

    LDR           R0, =GPIO_PORTE_AMSEL_R   ; clear amsel for normal operation
    MOV           R1, #0x00
    STR       R1, [R0]

    LDR           R0, =GPIO_PORTE_DIR_R        ; set PE1 as input and PE0 as output
    MOV           R1, #0x01
    STR       R1, [R0]

    LDR    R0, =GPIO_PORTE_AFSEL_R             ; disable analog functionality
  MOV    R1, #0x00
    STR       R1, [R0]

    LDR    R0, =GPIO_PORTE_DEN_R         ; enable digital I/O on PE0 and 1
  MOV    R1, #0x03
    STR       R1, [R0]

    LDR    R0, =GPIO_PORTE_PCTL_R        ; clear PCTL for PE
  MOV    R1, #0x00
    STR       R1, [R0]

  CPSIE  I    ; TExaS voltmeter, scope runs on interrupts

    LDR           R3, =PE0
led_on

  LDR    R4, [R3]                        ; load PE0 data into R4
  ORR    R4,  R4, #0x01                       ; set PE0
  STR    R4, [R3]                        ; store data onto PE0
    B         loop

led_toggle

    LDR           R4, [R3]                       ; load PE0 data into R4
    EOR           R4, R4, #0x01                  ; toggle PE0
    STR       R4,  [R3]                     ; store toggled data into PE0
    MOV       R5, #3875                          ; start the delay process
    B         Delay

Delay

    MOVS    R2, #0                               ;reset count for timer

sub_loop

    ADD       R2, #1                             ; add 1 to R2 until it hits 255
    CMP       R2, #255                      ; after hitting 255 subtract 1 from R5
    BNE           sub_loop
    SUB       R5, #1                             ; sub 1 from R5 until it hits zero
    CMP       R5, #0
    BNE       Delay                              ; go back until first portion of sub loop has been run 1447 times
```

```
      B          loop
check                                                   ; check that the led is already on

      LDR          R4, [R3]
      CMP          R4, #0x00
      BEQ          led_on


loop
    LDR    R0, =PE1
      LDR          R7, [R0]
      CMP       R7, #0x02
      BNE       check                                   ; turns on led if PE1 is not pressed
      B         led_toggle                              ;toggles led if PE1 is high
    B       loop



     ALIGN     ; make sure the end of this section is aligned
     END       ; end of file
```