CPE 325: Intro to Embedded Computer System

Lab03

Digital I/O on Experimenter Board: LEDs and Switches

Submitted by: Caleb Keller

Date of Experiment:_______09/06/2022______________

Report Deadline:__________09/07/2022______________

Demonstration Deadline: _________09/08/2022_________________

Introduction

Write a brief discussion on what the lab is about. (Use the tutorial and write in your own words. DO NOT copy text).

In this lab, I am writing a C program that manipulates components on a MSP430 board. This program assigns LEDs to flash at different speeds (in Hertz) and utilizes switches to cause the LEDs to respond with unique behaviors.

Theory

Write short notes on each topic discussed in lab.

Topic 1: Debouncing

a.) Debouncing refers to the implementation of code to confirm that a switch has been pressed. A delay is added to detect that a switch is pressed verses a faulty detection caused by noise.

b.) Debouncing is necessary to ensure that the communication between hardware and software is accurate.

c.) The "noise" being referred to consists electrical pulses that happen right before a switch is pressed. An Oscilloscope reading can represent the period of noise right before a switch is pressed.

Topic 2: Software Delay

a.) Software delays can be implemented in code by creating a "for loop" to pause for a specified time before executing a task. Delays are used in debouncing.

b.) Software delays are also used in creating timed gaps between actions in programming.

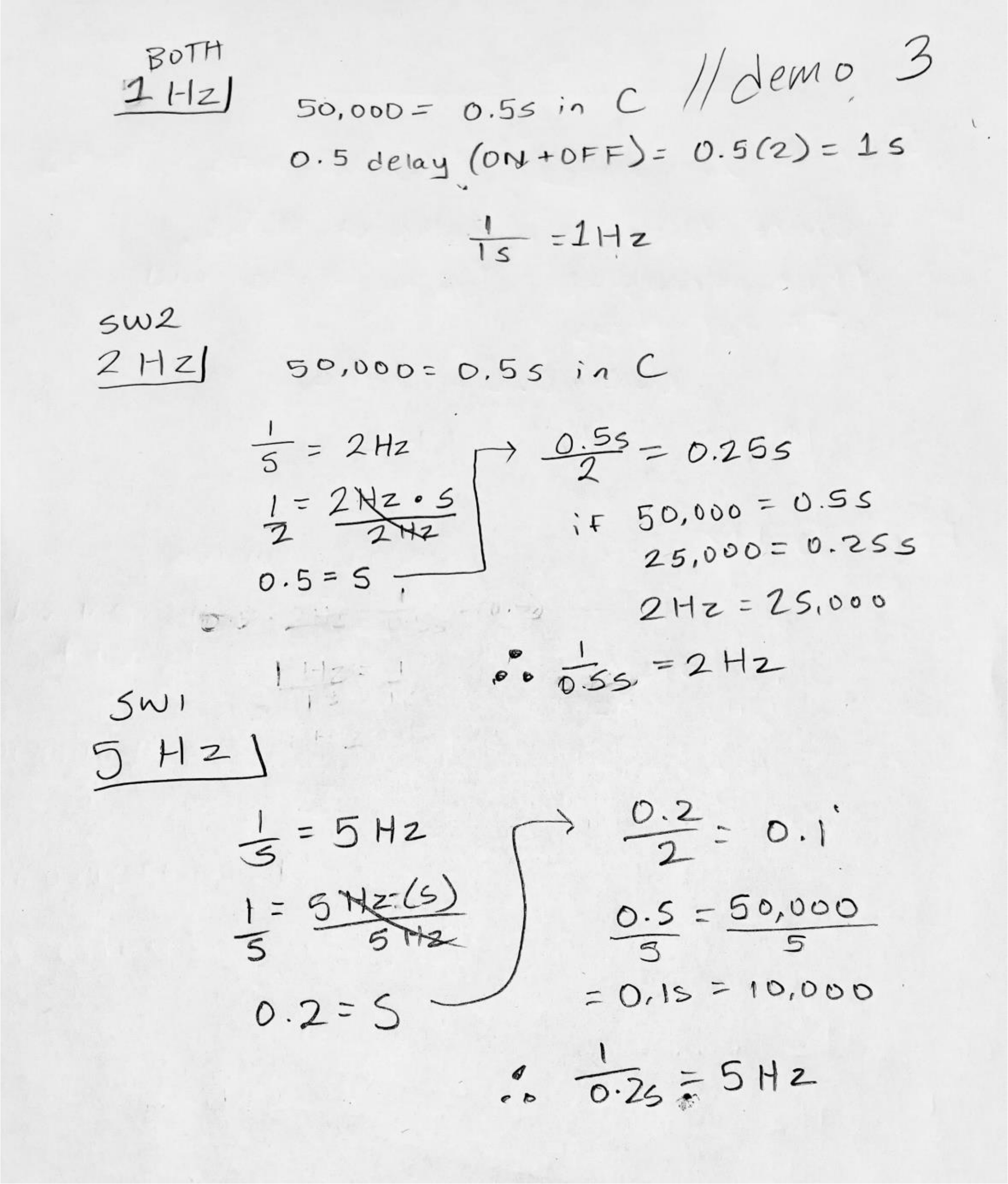
Results & Observation

Program:

Program Description:

Explain your approach in solving the problem.

To perform the given tasks for this program, I began by understanding how bit-wise operations work in the C programming language. I initialized the registers and set the LEDs to their default settings (ON or OFF depending on what the assignment specified). Once all of the initialization process was completed, I began writing the main code. To toggle an LED for a certain amount of time, I used a "for loop" with a value that is translated to match a certain period in Hertz (I will attach a photo below of how I calculated the Hertz values). To determine if a switch was pressed, I used several "if" statements as well as a debouncing loop to confirm the specified switch was pressed. Inside of the "if" statements are "while loops" that perform given tasks according to if the previous "if" statement is true.



BOTH
1 Hz

50,000 = 0.5s in C //demo 3

0.5 delay (ON+OFF) = 0.5(2) = 1s

$\frac{1}{1s} = 1Hz$

SW2
2 Hz

50,000 = 0.5s in C

$\frac{1}{s} = 2Hz$

$\frac{1}{2} = \frac{2\cancel{Hz} \cdot s}{2\cancel{Hz}}$

$0.5 = s \rightarrow \frac{0.5s}{2} = 0.25s$

if 50,000 = 0.5s

25,000 = 0.25s

2Hz = 25,000

$\therefore \frac{1}{0.5s} = 2Hz$

SW1
5 Hz

$\frac{1}{s} = 5 Hz$

$\frac{1}{5} = \frac{5\cancel{Hz}(s)}{5\cancel{Hz}}$

$0.2 = s \rightarrow \frac{0.2}{2} = 0.1$

$\frac{0.5}{5} = \frac{50,000}{5}$

$= 0.1s = 10,000$

$\therefore \frac{1}{0.2s} = 5 Hz$

Figure #1: My Hertz Calculation Method

Program Flowchart

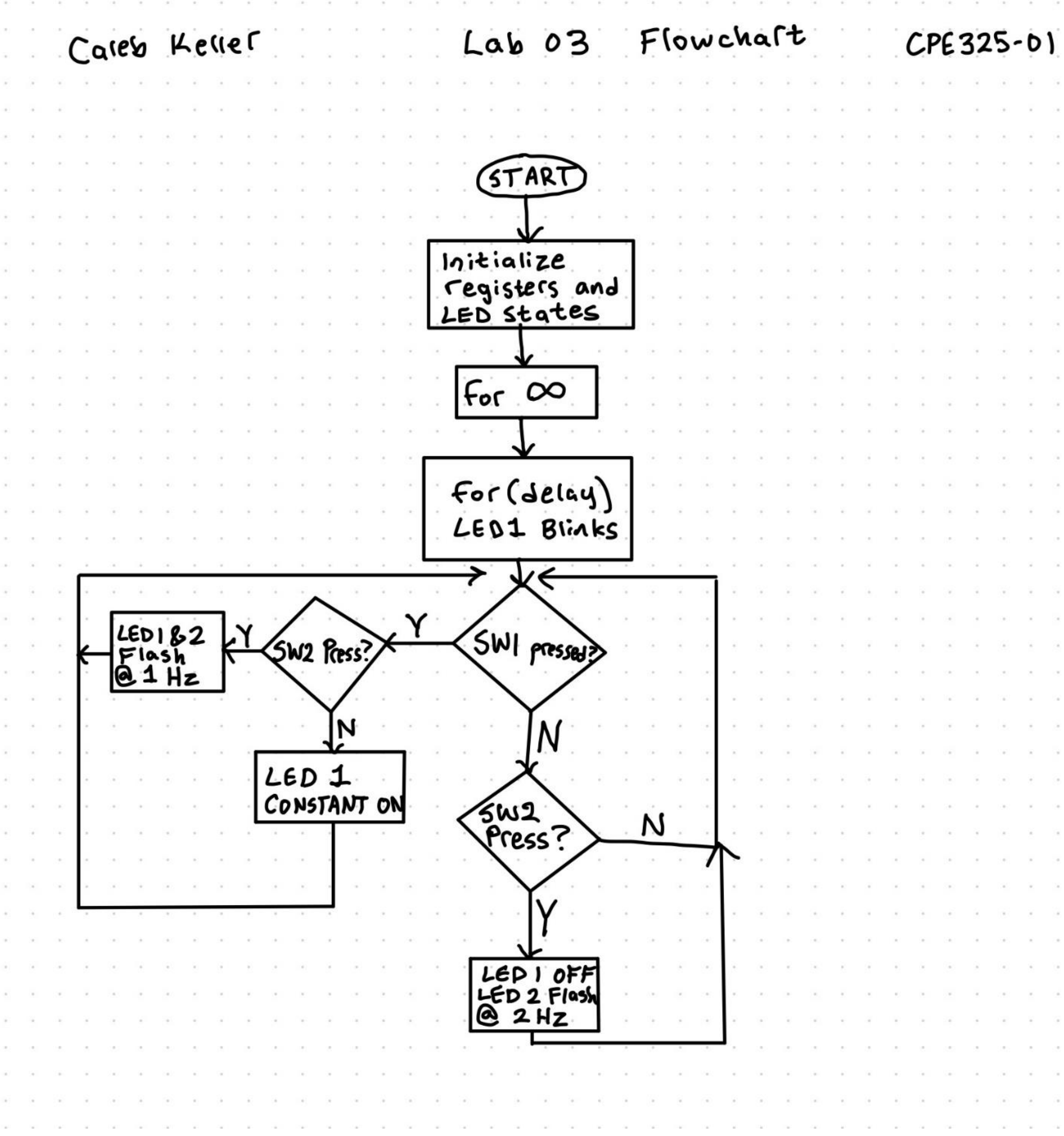


Figure #2: Program Flowchart

Conclusion

While working on this program, I ran into many challenges and issues. My most prevalent challenge was formatting the loops and statements to correctly produce the desired output. The bit-wise operations were also a challenge because this was a new concept to me; however, I believe that I have a much clearer understanding of them after completing this program. An issue I faced was calculating the period to delay in Hertz. I began by converting the value in demo 2 of the lab tutorial to seconds, and I followed that method to get each of the values to place in each loop. In conclusion, I have learned several new topics regarding programming hardware—and might have actually developed a new niche!

Appendix

Your first code goes here, if any. Make sure you use a 1X1 table for this.

(Note: Make sure the code is readable, have comments. Also reduce spacing between lines to avoid lengthy reports.

Table #3: Program source code

```
/*
 * Title: Lab 3
 * Author: Caleb Keller
 * Written: 09/01/2022
 * Description: Programs LED1 and LED2 to turn on and blink at different speeds
whilst using the switches on the board
 */


#include <msp430.h>

#define SW1 P2IN&BIT1 // (0000_00[0]0) matches P2DIR
#define SW2 P1IN&BIT1 // (0000_00[0]0) matches SW1 so P1DIR will also match P2DIR

int main(void)
{
       WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

       P1DIR |= BIT0; // setting P1.0 (LED1) to output direction (0000_000[1])
       P1OUT |= BIT0; // LED1 is ON (0000_000[1])

       P2DIR &= ~BIT1; //Set P2.1 (button) as input for SW1 (0000_00[0]0)
       P2REN |= BIT1; // enable pull up resistor at P2.1
       P2OUT |= BIT1; // required for proper IO


    P4DIR |= BIT7; // setting P4.7 (LED2) to output direction ([1]000_0000)
    P4OUT &= ~BIT7; // LED2 is OFF ([0]000_0000)

    P1DIR &= ~BIT1; //Set P1.1 (button) as input for SW2 (0000_00[0]0)
    P1REN |= BIT1; // enable pull up resistor at P1.1
    P1OUT |= BIT1; // required for proper IO

    unsigned int i = 0;

    for(;;)
    {
        for(i = 0; i < 10000; i++); // Start with LED1 at 5Hz blinking
        P1OUT ^= BIT0; // TOGGLE LED1

// Beginning of bonus (when both switches pressed, both lights flash at 1 Hz
        if((SW1) == 0 && (SW2) == 0)
             {
                for(i = 2000; i > 0; i--); // debouncing for 20ms to confirm no
electrical feedback causing a false button press

                if((SW1) == 0 && (SW2) == 0)
                {
                 P1OUT &= ~BIT0; // turn off LED1
                 P4OUT &= ~BIT7; // turn off LED2

                 while((SW1) == 0 && (SW2) == 0)
                 {
                     for(i = 0; i < 50000; i++); // 1Hz blinking
                     P1OUT ^= BIT0; // toggle LED1
                     P4OUT ^= BIT7; // toggle LED2
                 }

                }while((SW1) == 0 && (SW2) == 0); // as long as both pressed
             }
// End of Bonus

// If switch one is pressed, LED1 stops flashing and is constantly lit
        if((SW1) == 0) // if SW1 is pressed
        {
            for(i = 2000; i > 0; i--); // debouncing for 20ms to confirm no
electrical feedback causing a false button press
            if((SW1) == 0)
            {
                P1OUT |= BIT0; // turn LED1 ON without delay and constant
            }
            while((SW1) == 0); // while SW1 is pressed
        }

// If switch two is pressed, LED1 turns off, LED2 begins flashing at 2Hz
        if((SW2) == 0) // if SW2 is pressed
        {
            for(i = 2000; i > 0; i--); // debouncing for 20ms to confirm no
electrical feedback causing a false button press
            if((SW2) == 0)
            {
                P1OUT &= ~BIT0; // LED 1 IS OFF
                P4OUT |= BIT7; // LED 2 IS ON
                while((SW2) == 0)
                {
                  for(i = 0; i < 25000; i++); // 2Hz blinking
                  P4OUT ^= BIT7; // toggle LED2
                }
            }while((SW2) == 0);

         }
        P4OUT &= ~BIT7; // had an issue where LED2 would stay on randomly after not pressing, so confirmed LED2 is off after letting go


    }



      return 0;
}
```

Make sure to submit a single pdf file only