CPE 325: Intro to Embedded Computer System

Lab03

Digital I/O on Experimenter's Board: LEDs and Switches

Submitted by: Anshika Sinha

Date of Experiment:___01/22/2025___________________

Report Deadline:_____01/28/2025____________________

Demonstration Deadline: __02/03/2025________________

Theory

Topic 1: Debouncing

a) Debouncing is a technique used to ensure that a function is executed only once after a specific period of time, regardless of how many times it is triggered within that interval.

b) Debouncing prevents unnecessary executions of a function when an event is triggered multiple times in quick succession, improving performance by reducing the frequency of expensive operations.

c) It is usually applied in scenarios where events occur rapidly, such as keystrokes and mouse movements.

d) It is implemented by setting a timer when the event is triggered. If the event is triggered again before the timer expires, the previous timer is cleared and restarted. The function executes only after the timer expires without further triggers.

Topic 2: Software Delay

a) A software delay introduces a time delay between operations without the use of hardware timers in an embedded system where precise timing is not critical.

b) This is implemented using an empty for-loop. The delay duration depends on the number of iterations, the clock speed of the microcontroller, and the instructions used in the loop.

c) Software delays are useful for making LEDs blink, debouncing, and allowing hardware to initialize during startup.

d) Ex. :

```
unsigned int i = 0;
for (i = 0; i < 2000; i++);          //20 ms delay
```

Results & Observation

Program 1:

Program Description:

This C program starts with LED1 off and LED2 on. If Switch 1 is held, LED1 blinks at 11 Hz and LED2 turns off. If Switch 2 is held, LED1 turns on and LED2 blinks at 6 Hz. When none of the switches are pressed, it returns to its original state with LED1 off and LED2 on.

Report Questions:

1. **How do you handle debouncing?**
   Debouncing is handled with an empty for loop under a condition to create a delay of 20 ms, before checking the condition once again to determine whether the switch is actually pressed.

2. **How do you create the required delay?**
   The required delay is implemented using empty for loops.

   To blink LED1 at 11 Hz,

   $$11\,Hz \;=\; \frac{1000\,ms}{t}$$

   $$t \;\approx\; 90\,ms$$

   To blink LED2 at 6 Hz,

   $$6\,Hz \;=\; \frac{1000\,ms}{t}$$

   $$t \;\approx\; 166\,ms$$

   Therefore, the empty loops create a delay of 90 ms and 166 ms for when SW1 and SW2 are held respectively.

3. **How does your code handle both the switches?**
   While the program is running, the code checks if Switch 1 is held. If it is, it executes the instructions for that condition. If Switch 1 is not held, it checks if Switch 2 is held. If Switch 2 is held, it executes the respective instructions for that condition. Otherwise, it remains in its original state.

Program Flowchart:

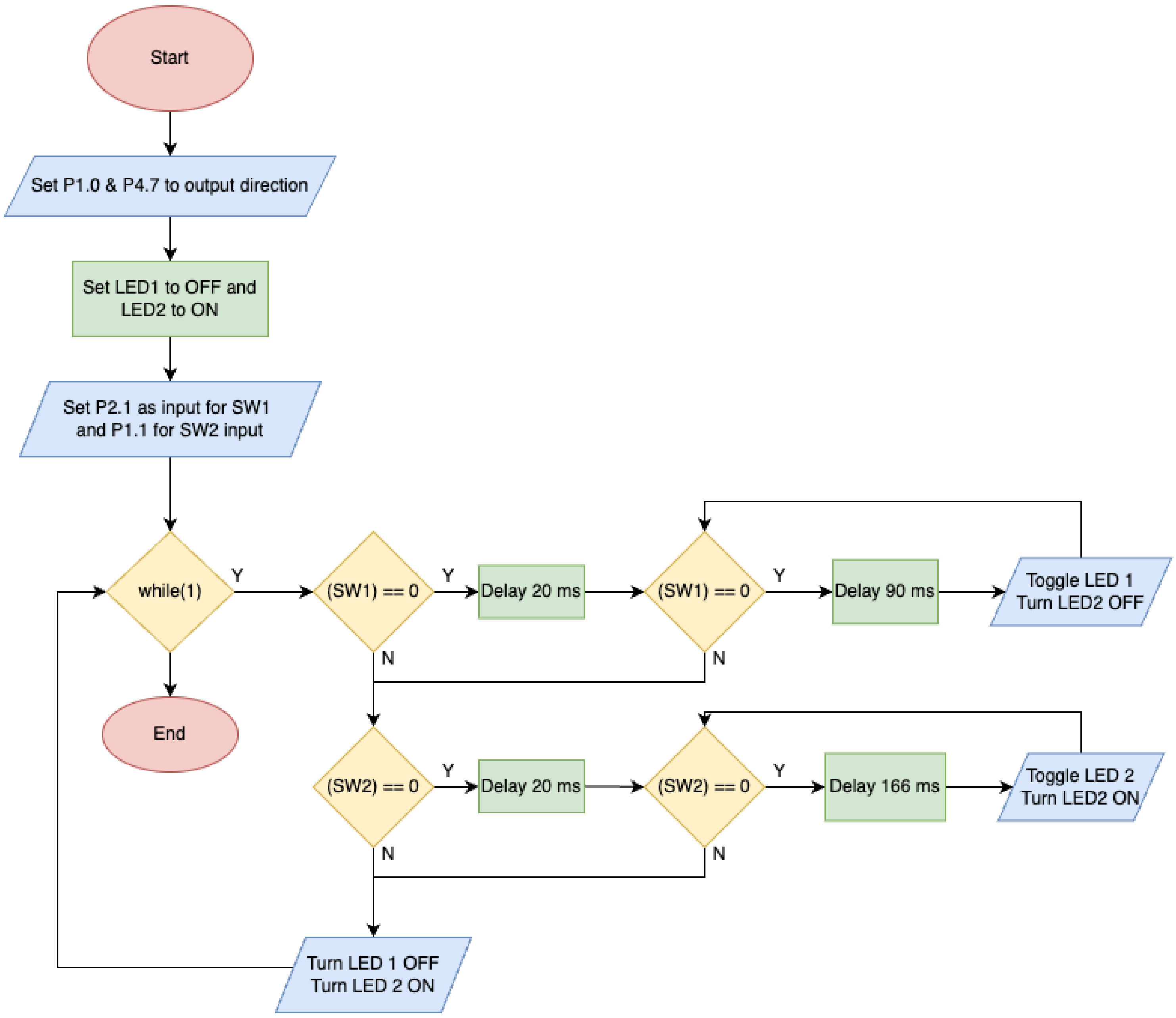
Start
Set P1.0 & P4.7 to output direction
Set LED1 to OFF and LED2 to ON
Set P2.1 as input for SW1 and P1.1 for SW2 input
while(1)
Y
End
(SW1) == 0
Y
N
Delay 20 ms
(SW1) == 0
Y
N
Delay 90 ms
Toggle LED 1 Turn LED2 OFF
(SW2) == 0
Y
N
Delay 20 ms
(SW2) == 0
Y
N
Delay 166 ms
Toggle LED 2 Turn LED2 ON
Turn LED 1 OFF Turn LED 2 ON

Figure 01: Program 1 Flowchart

Appendix

Table 01: Program 1 source code

```
/*------------------------------------------------------------------------------
 * File:        Lab01_P1.c
 * Function:    Controlling LED1 and LED2 using switches
 * Description: This C program starts with LED1 off and LED2 on. If SW1 is
 *              held, LED1 blinks at 11 hz and LED2 turns off. If SW2 is
 *              held, LED1 turns on and LED2 blinks at 6 hz.
 *
 * Input:       Hold Switch 1 or Switch 2
 * Output:      LED1 blinks at 11 hz and LED2 turns off when Switch1 is held
 *        and LED1 turns on and LED2 blinks at 6 hz when Switch2 is pressed
 *
 * Author(s):   Anshika Sinha
 * Date:        01/22/2025
 *----------------------------------------------------------------------------*/
#include <msp430.h>

// Interface inputs and outputs
#define SW1 P2IN&BIT1                // Switch 1 at P2.1
#define SW2 P1IN&BIT1                // Switch 2 at P1.1
#define LED1 0x01                    // Mask for BIT0 = 0000_0001b
#define LED2 0x80                    // Mask for BIT7 = 1000_0000b

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;        // Stop watchdog timer

    // Original state with LED1 OFF and LED2 ON
    P1DIR |= LED1;                   // Set P1.0 to output direction
    P4DIR |= LED2;                   // Set P4.7 to output direction
    P1OUT &= ~LED1;                  // LED1 is OFF
    P4OUT |= LED2;                   // LED2 is ON

    P2DIR &= ~BIT1;                  // Set P2.1 as input for SW1 input
    P2REN |= BIT1;                   // Enable pull-up register at P2.1
    P2OUT |= BIT1;

    P1DIR &= ~BIT1;                  // Set P1.1 as input for SW2 input
    P1REN |= BIT1;                   // Enable pull-up register at P1.1
    P1OUT |= BIT1;

    unsigned int i = 0;
    while(1)
    {
        if ((SW1) == 0)                      // If SW1 is pressed
        {
            for (i = 2000; i>0; i--);        // Debounce ~20 ms
            while ((SW1) == 0)                  // If SW1 is actually pressed
            {
                for (i = 0; i < 9000; i++);// 90 ms delay
                P1OUT ^= LED1;               // Toggle LED1
                P4OUT &= ~LED2;              // Turn LED2 OFF
            }
        }
        else if ((SW2) == 0)
        {
            for (i = 2000; i>0; i--);       // Debounce ~20 ms
            while ((SW2) == 0){
                for (i = 0; i < 16600; i++);// 166 ms delay
                P4OUT ^= LED2;              // Toggle LED2
                P1OUT |= LED1;              // Turn LED1 on
            }
        }
        else
        {
                P1OUT &= ~LED1;             // Turn LED1 off
                P4OUT |= LED2;              // Turn LED2 on
        }
    }
}
```

Make sure to submit a single pdf file only