

# **Lab Experiment 4#**

**ENGR 478**

**Submitted by:**

**Jochelle Viray**

**Edgar Torres**

**Brandyn Byrnes**

**March 5, 2023**

## Introduction:

The purpose of this experiment is to familiarize with the microcontroller TM4C123G by Texas Instrument. Before getting started with the experiment, we proceed with the lab procedures to serve its purpose. Since the previous lab experiment was assigned to use a low-level language in software Keil  $\mu$ Vision5. In this lab, we would need to use C-programming to complete the task.

In this lab, we are observing the Tiva board and its LED by running the program in switch and toggle. The aim of this experiment is to program the microcontroller; hence, C-programming would help us learn how to change the color of the LED and control the switch. This lab experiment has given us a task to rebuild switch and toggle function.

In the switch function, we are tasked to switch the color of the LED light from blue to red and let the light on when the SW2 is not pressed, otherwise, it will be off. While the toggle function has to be delayed for 5 times slower than the original function. This lab has been demonstrated to the lab instructor.

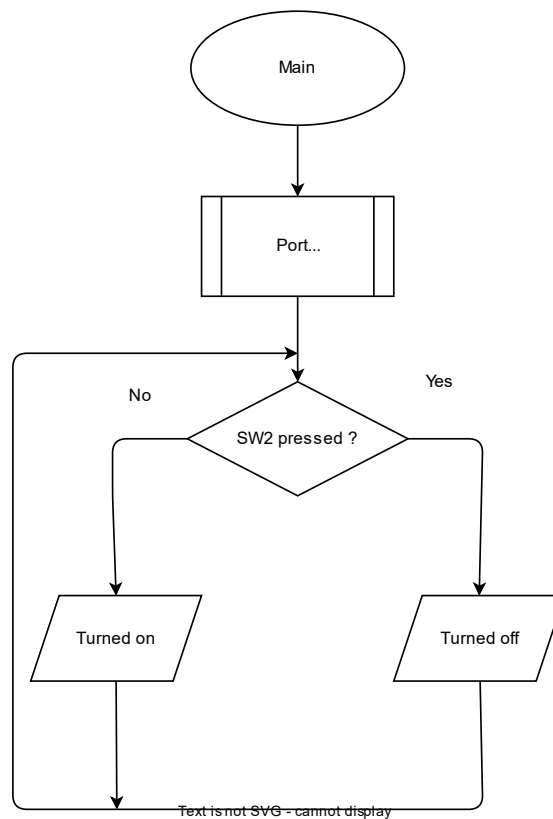
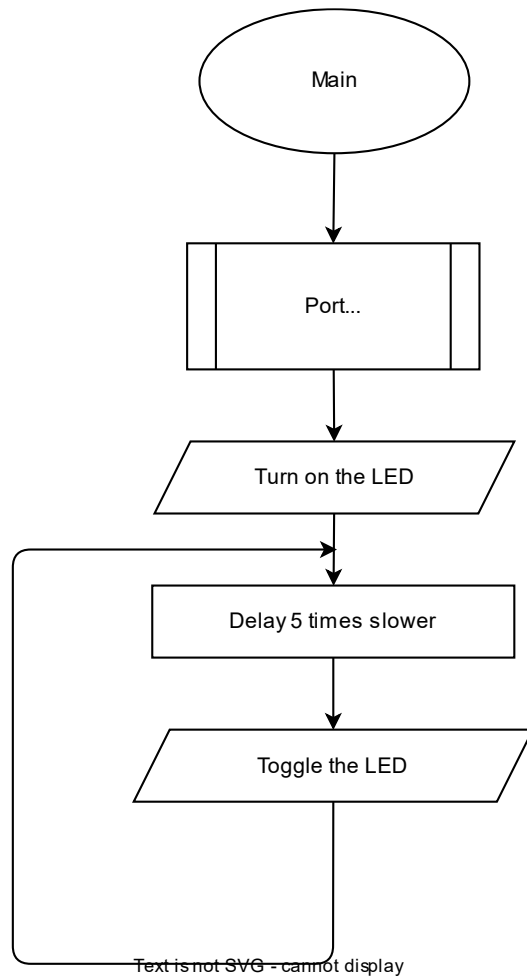


Figure 1: work of switch function



Text is not SVG - cannot display  
Figure 2: workflow of toggle

Figure 1 and Figure 2 show the workflow of my\_switch and my\_toggle. The following statement shows the modification that we have been assigned to create.

1. #include <stdint.h>
2. #include "inc/tm4c123gh6pm.h"
  
3. #define LED\_MASK 0x02 // Changed to LED\_MASK from BLUE\_MASK
4. //\*\*\*\*\*
5. //
6. //!
7. //! A very simple example that interfaces with the blue LED (PF2) and SW2 (PF0)
8. //! using direct register access. When SW2 is pressed, the LED is turned on.  
When
9. //! SW2 is released, the LED is turned off.
- 10.//

```

11.//*****

12.void
13.PortFunctionInit(void)
14.{

    i. volatile uint32_t ui32Loop;

    ii. // Enable the clock of the GPIO port that is used for the on-board
        LED and switch.
15.SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;

16.//
17.// Do a dummy read to insert a few cycles after enabling the peripheral.
18.//
19.ui32Loop = SYSCTL_RCGC2_R;

    i. // Unlock GPIO Port F
    ii. GPIO_PORTF_LOCK_R = 0x4C4F434B;
    iii. GPIO_PORTF_CR_R |= 0x01;           // allow changes to PF0

20.// Set the direction of PF2 (blue LED) as output
21.GPIO_PORTF_DIR_R |= LED_MASK;

    i. // Set the direction of PF0 (SW2) as input by clearing the bit
22.GPIO_PORTF_DIR_R &= ~0x01;

23.// Enable both PF2 and PF0 for digital function.
24.GPIO_PORTF_DEN_R |= LED_MASK + 0x01; // New modification

    i. //Enable pull-up on PF0
    ii. GPIO_PORTF_PUR_R |= 0x01;

```

```
25.}
```

```
26.int main(void)
```

```
27.{
```

- i. //initialize the GPIO ports
- ii. PortFunctionInit();

```
28.//
```

```
29.// Loop forever.
```

```
30.//
```

```
31.while(1)
```

```
32.{
```

```
    a. if((GPIO_PORTF_DATA_R&0x01)!=0x00) //SW2 is pressed
```

```
        a. {
```

```
            1. // Turn off the LED.
```

```
            ii. GPIO_PORTF_DATA_R |= LED_MASK;
```

```
        b. }
```

```
        c. else
```

```
        d. {
```

```
            1. // Turn on the LED.
```

```
            2. GPIO_PORTF_DATA_R &=
                ~LED_MASK;
```

```
        e. }
```

```
33.}
```

```
34.}
```

```

35.#include <stdint.h>
36.#include <stdbool.h>
37.#include "inc/tm4c123gh6pm.h"
38.#include "driverlib/sysctl.h"

39.#define      GREEN_MASK      0x08
40.//*****
41.//
42.//!
43.//! A very simple example that toggles the on-board red LED using direct register
44.//! access.
45.//
46.//*****

47.void
48.PortFunctionInit(void)
49.{
50.//
51.volatile uint32_t ui32Loop;
52.// Enable the GPIO port that is used for the on-board LED.
53.//
54.SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;

55.//
56.// Do a dummy read to insert a few cycles after enabling the peripheral.
57.//
58.ui32Loop = SYSCTL_RCGC2_R;

59.//
60.// Enable the GPIO pin for the red LED (PF1). Set the direction as output, and
61.// enable the GPIO pin for digital function.
62.//
63.GPIO_PORTF_DIR_R |= GREEN_MASK;
64.GPIO_PORTF_DEN_R |= GREEN_MASK;

```

```
65.}
```

```
66.int main(void)
67.{
```

```
68.//initialize the GPIO ports
69.PortFunctionInit();
```

```
70.// Turn on the LED.
71.GPIO_PORTF_DATA_R |= GREEN_MASK;
```

```
72.//
73.// Loop forever.
74.//
75.while(1)
76.{
77.// Delay for a bit.
78.SysCtlDelay(2000000*5);
```

```
79.// Toggle the LED.
80.GPIO_PORTF_DATA_R ^= GREEN_MASK;
81.}
82.}
```

For a better view of this code, this [link](#) goes to the GitHub where I post my solution for the lab.