

# **Assignment 1 Report**

EE615 : Embedded Systems Lab

Atharv Gade (210020004)  
Yash Meshram (210020053)  
23 August 2024

## Aim:

1. Blink the LED of the Tiva microcontroller with a delay of precisely 1 second.
2. Generate various colors by combining various LEDs by changing the PORTF\_DATA register.

## Procedure:

- Connect the Tiva Microcontroller through the microUSB port and set the switch to debug mode. Create a new project in Code Composer Studio on your system. The Project Parameters are:

Target device - Tiva C series | Tiva TM4C123GH6PM

Compiler version - TI Clang v3.2.1.LTS

Connection - Stellaris In-Circuit Debug Interface

- Add the 'tm4c123gh6pm.h' header file to the project and edit your code in the 'main.c' file.
- To dump the code on the board, build the file and then debug & run it.

## Theory:

1. To generate a delay of precisely 1 second, we can calculate the number of iterations of the delay loop that will cause a delay of 1 second.

The number of iterations can be calculated by dividing the total number of clock cycles in a second by the number of clock cycles it takes per iteration.

We can calculate the number of iterations by the following formula:

$$\text{Iterations} = (\text{Total \# of clock cycles}) / (\text{\# of clock cycles per iteration}).$$

We can also use a stopwatch to approximate the delay and adjust the value accordingly.

For this experiment, we used a stopwatch to obtain the value 700,000 generates a delay of approximately 1 second.

2. To generate various colors, we can set different values of the GPIO\_PORTF\_DATA\_R register. We have red(bit[1]), blue(bit[2]) and

green(bit[3]) color LEDs on the board. The combination of these is used to generate different colors.

Red	-	0001 0010
Yellow	-	0001 1010 (Green + Red)
Green	-	0001 1000
Blue	-	0001 0100
Purple	-	0001 0110 (Blue + Red)

## Code:

### 1. One Second Delay

```
#include <stdint.h>
#include <stdbool.h>
#include "tm4c123gh6pm.h"

int main(void)
{
    SYSCTL_RCGC2_R |= 0x00000020;          /* enable clock to
GPIOF */
    GPIO_PORTF_LOCK_R = 0x4C4F434B;        /* unlock commit
register */
    GPIO_PORTF_CR_R = 0x1F;                /* make PORTF0
configurable */
    GPIO_PORTF_DEN_R = 0x1E;               /* set PORTF pins 4
pin */
    GPIO_PORTF_DIR_R = 0x0E;               /* set PORTF4 pin as
input user switch pin */
    GPIO_PORTF_PUR_R = 0x10;               /* PORTF4 is pulled
up */

    register long int x = 0;
    while(1)
    {
        GPIO_PORTF_DATA_R = 0x12;
        x = 0;
        while(x<700000)
        {
            x++;
        }
        GPIO_PORTF_DATA_R = 0x00;
        x = 0;
        while(x<700000)
        {
            x++;
        }
    }
}
```

## 2. Color Combination

```
#include <stdint.h>
#include <stdbool.h>
#include "tm4c123gh6pm.h"

int main(void)
{
    SYSCTL_RCGC2_R |= 0x00000020;      /* enable clock to
GPIOF */
    GPIO_PORTF_LOCK_R = 0x4C4F434B;    /* unlock commit
register */
    GPIO_PORTF_CR_R = 0x1F;            /* make PORTF0
configurable */
    GPIO_PORTF_DEN_R = 0x1E;           /* set PORTF pins 4
pin */
    GPIO_PORTF_DIR_R = 0x0E;           /* set PORTF4 pin as
input user switch pin */
    GPIO_PORTF_PUR_R = 0x10;           /* PORTF4 is pulled
up */

    register int x = 0;
    while(1)
    {
        GPIO_PORTF_DATA_R = 0x12;
        x = 0;
        while(x<900000)
        {
            x++;
        }
        GPIO_PORTF_DATA_R = 0x1A;
        x = 0;
        while(x<900000)
        {
            x++;
        }
        GPIO_PORTF_DATA_R = 0x18;
        x = 0;
        while(x<900000)
        {
            x++;
        }
        GPIO_PORTF_DATA_R = 0x14;
        x = 0;
        while(x<900000)
        {
            x++;
        }
        GPIO_PORTF_DATA_R = 0x16;
```

```

        x = 0;
        while(x<900000)
        {
            x++;
        }
    }
}

```

## Output:



Fig. 1 : Blinking LED



Fig. 2 : Color Changing

## Result:

The project demonstrated that a simple delay loop can be used to generate approximate delays without the use of hardware timers. However, the accuracy of such delays is limited by the instruction execution time and the number of cycles per iteration.

The combination of different bits in the 'GPIO\_PORTF\_DATA\_R' register allowed for the creation of multiple colors using the onboard RGB LED. We can understand how bitwise operations can be used to control individual pins of the microcontroller.

## **Conclusion:**

We were successful in demonstrating an approximate 1 second delay and the generation of multiple colors using the Tiva TM4C123GH6PM microcontroller and the experiment provided a valuable insight into the fundamentals of embedded programming and GPIO manipulation.