EENG 5600 Microcomputer Engineering

Microcontroller Lab 1 Thursday Afternoon Lab

Part 1

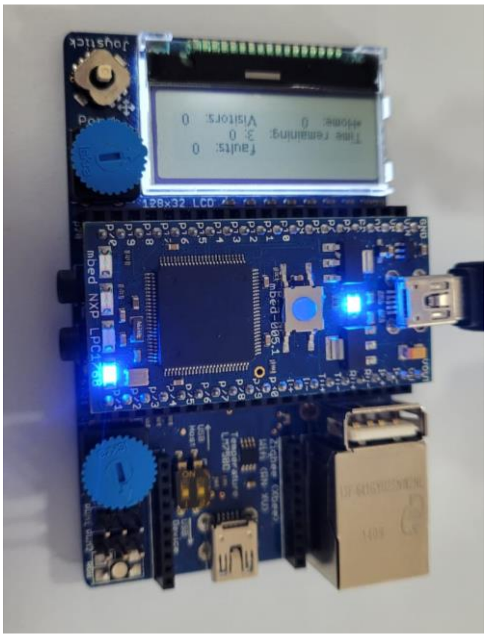
Flashing LED 1:

Uploading the code on LPC 1768

```
#include "mbed.h"
DigitalOut myled(LED1);// assigning code to led 1

int main()
{
  while(1)
 {
    myled = 1; // LED 1 to turn on
    wait(0.2); // Time delay 0.2 sec
    myled = 0; // LED off
    wait(0.2);
  }

}
```

Modifying code to flash LED 4 for 2 seconds

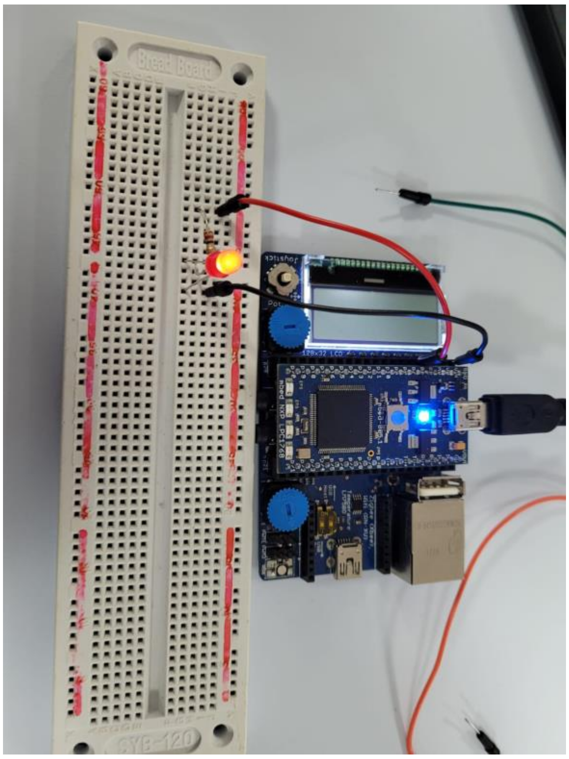
```
#include "mbed.h"
DigitalOut myled(LED4); //Setting code for LED 4

int main() {
  while(1) {
    myled = 1; //LED on
    wait(2); // Time delay 2 seconds
    myled = 0; // LED off
    wait(2);
  }
}
```



Part 2

Connecting LED to breadboard and controlling it from microcontroller.

```
#include "mbed.h"

DigitalOut myled(p5); //LED pin set at pin 5 rather than controlling the in built LED

int main() {

   while(1) {

      myled = 1; // LED on

      wait(2); //time delay 2 seconds

      myled = 0; //led off

      wait(2);

   }

}
```

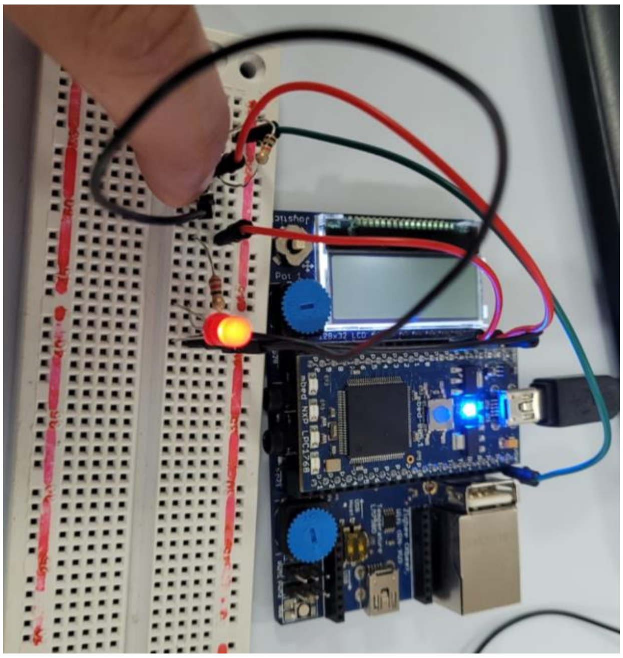


Maximum voltage without fusing the LED typically is 2.5V and Mbed supplies 3.3V. Using a resistor reduces the voltage and manages current within a safe range of 20mA to 40mA.

Part 3

VDD connection and button connection

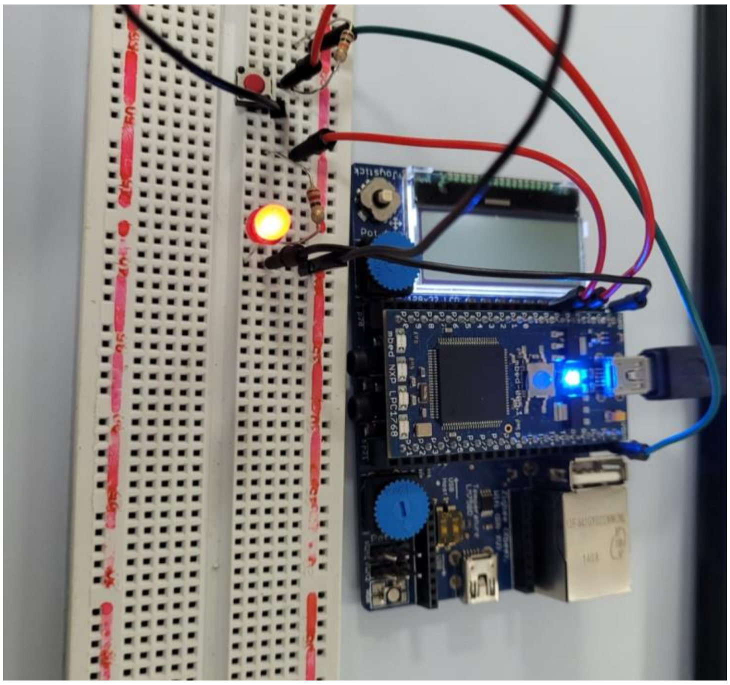
```
#include "mbed.h"

DigitalIn switchinput(p5);//Button acting as input

DigitalOut light(p6); //LED connection


int main() {

   while(1)

   {

      if (switchinput == 1) //short circuit so no current

      {

         light = 0; //light off
```



```
        }
        else
        light = 1;  //current on


    }
```

Leaving light on for 2 seconds after pressing the button

```
}
#include "mbed.h"


DigitalIn switchinput(p5);
DigitalOut light(p6);


int main() {
    while(1)
    {
        if (switchinput == 1)
        {
            light = 0; /* light on */
        }
        else
      {
        light = 1; //light on
        wait (2.0);// time delay 2 seconds
      }
    }
}
```



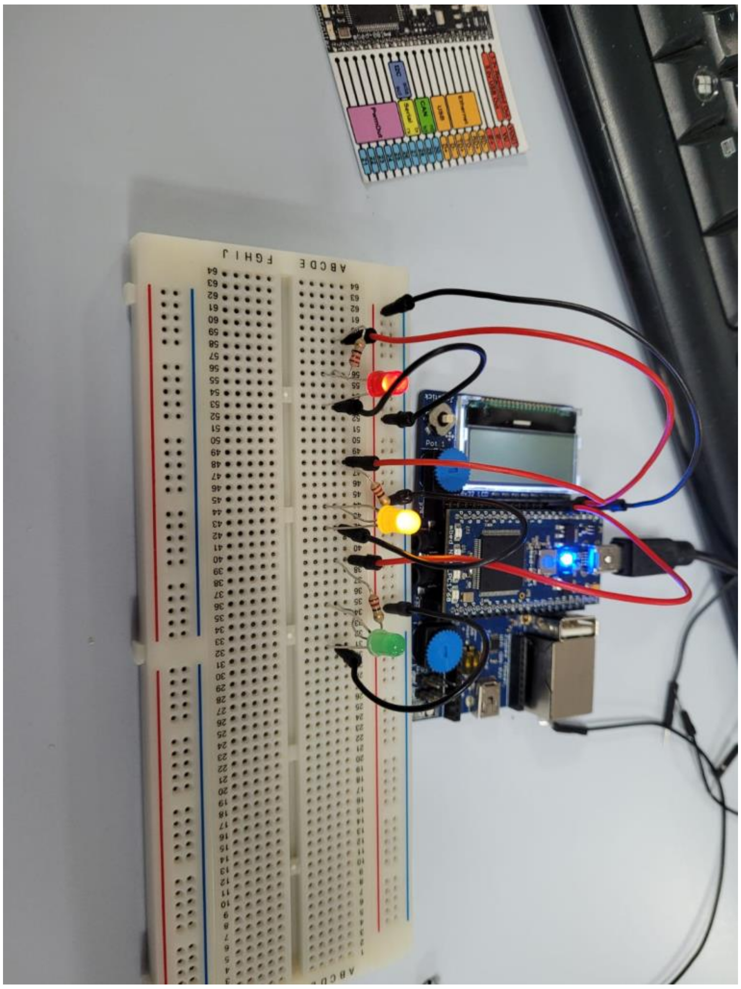
Part 4

Creating traffic lights with times mentioned on the table

- I have kept the connections as simple as possible. All LEDs are connected to pin 5, 6 and 7 consecutively. I used the top rail as a common ground from Mbed. Each LED has a 220 ohms resistor connected in series.

```
#include "mbed.h"
```

```
DigitalOut red(p5);
DigitalOut yellow(p6);
DigitalOut green(p7);

int main() {
  while(1)
  {
    red = 1;// only red on
    wait(2.5);
    //red = 1; red on from before
    yellow = 1; //yellow on with red
    wait (1);
    red = 0;
    yellow = 0;
    green = 1; // only green light on
    wait (2.5);
    green = 0;
    yellow = 1;
    wait (1.0);
    yellow = 0;
  }
}
```

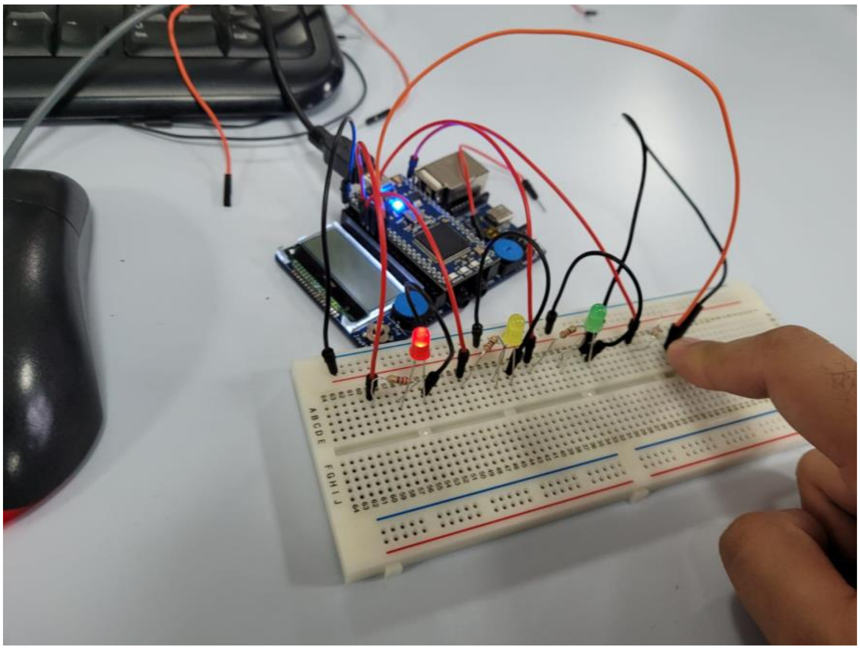


Adding a switch to stop the traffic lights and move to red lights

- For this circuit I added a switch to pin 9 as an input with 10K ohm resistor connected on VDD or VOUT pin on MBed. So once the Mbed is turned on the circuit acts like the traffic light like above. However, once you press the switch it stops LED yellow or green and only allows red light to stay on. However once button is released circuit goes back to traffic light.

```
#include "mbed.h"
DigitalOut red(p5);
DigitalOut yellow(p6);
DigitalOut green(p7);
DigitalIn switchinput(p9);// adding switch in as an input
```

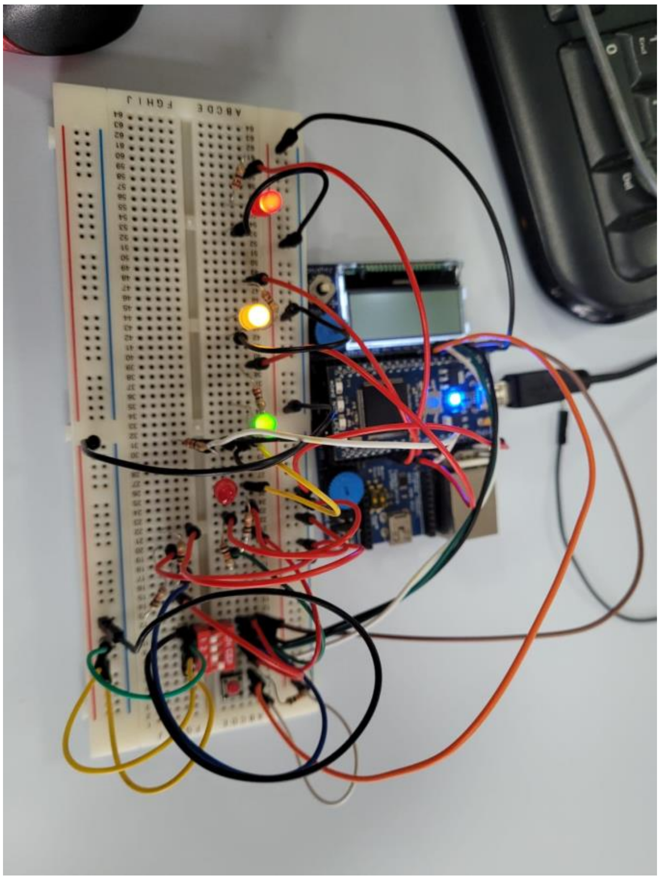
```
int main() {
  while(1)
  {

    red = 1;
    wait(2.5);
    //red = 1;
    yellow = 1;
    wait (1);
    red = 0;
    yellow = 0;
    green = 1;
    wait (2.5);
    green = 0;
    yellow = 1;
    wait (1.0);
    yellow = 0;
//adding while loop with the switch to turn lights red only
     while (switchinput == 0)
    {
      red = 1; /* light on */
      yellow = 0;
      green = 0;
    }

  }
}
```



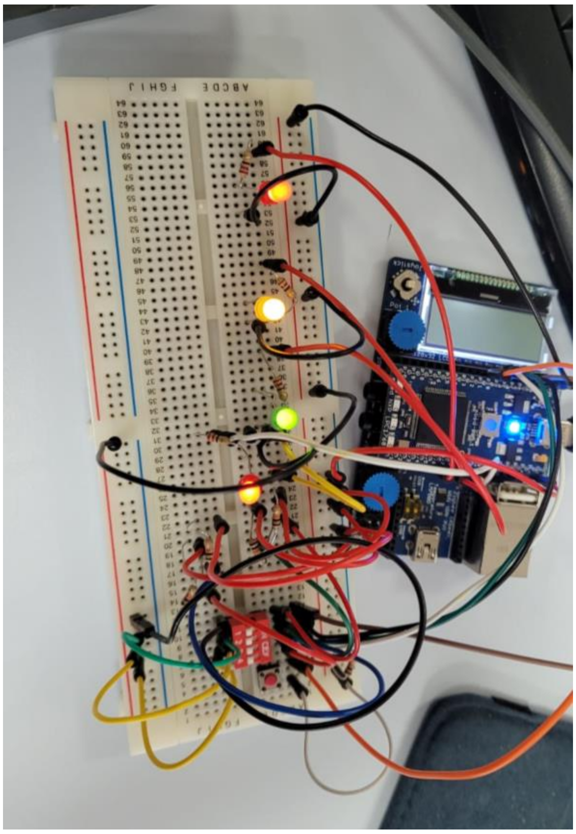
Part 5

- For this circuit I connected the pins as suggested in the brief. The Dip switch I assigned as a bus in controlling LEDs as Bus out in consecutive pins. Due to this the program is quite straight forward.

```
#include "mbed.h"


DigitalIn switchinput(p9);

BusIn switches (p5, p6, p7, p8);//Dip switch connections

BusOut LEDS (p27, p28, p29, p30);//LED connections


int main() {

    while(1)

    {

        LEDS = switches;//LEDs controlled by Dip switch

    }

}
```



- For this part of the circuit, I programmed it such that once the push button is pressed only then Dip switch changes would be accepted and LEDs would react accordingly. Although photo of this part is not essential I have attached one.



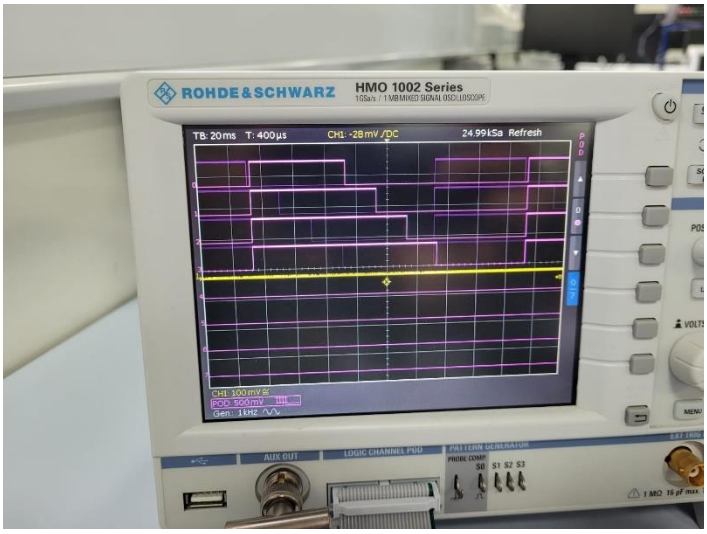
```
#include "mbed.h"


DigitalIn switchinput(p9);

BusIn switches (p5, p6, p7, p8);

BusOut LEDS (p27, p28, p29, p30);

//DigitalOut red(p27);

int main() {

    while(1)

    {
```

```
    // LEDS = switches;
     while (switchinput ==0)
     {
     LEDS = switches;
     wait(0.25);
     }

  }
}
```

Part 6

- Initial code without switch button to make it easy to understand

```
#include "mbed.h"
BusOut output (p5,p6,p7,p8); //An API for declaring a 4-bit Output
int main()
 {
int values[9]={0xF,0xF,0xF,0x7,0x3,0x1,0x0,0x0,0x0}; // using hexadecimal 4 bit input from the picture in the question
int durations[9]={20,20,20,20,20,20,20,20,20}; //Time duration between input
int n;
for(;;) {
for(n=0; n<9; n++) //loops to display data on the output
{
output = values[n]; // Output Pattern
wait_ms(durations[n]); // Delay till next Pattern
}
}
}
```

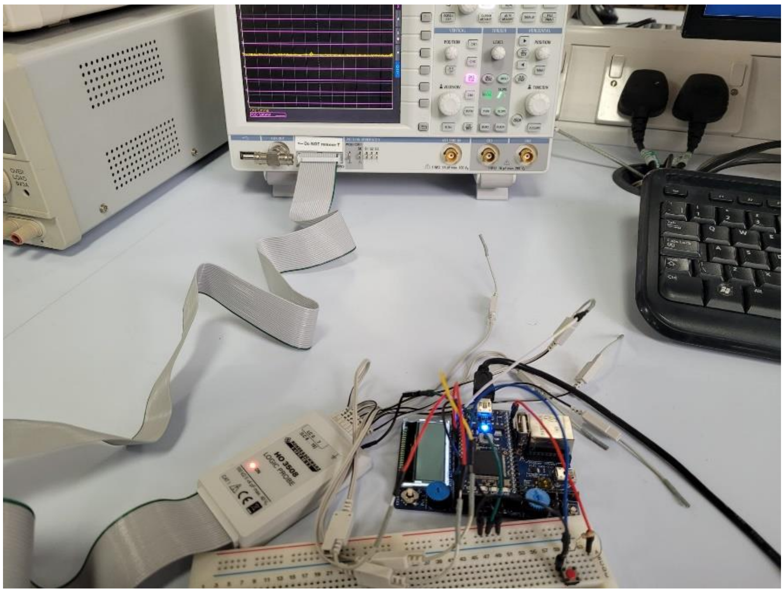


- Using the switch to display the output from above

```
#include "mbed.h"
```

```
BusOut output (p5,p6,p7,p8); //An API for declaring a 4-bit Output
DigitalIn switchinput(p20);
int main()
 {


    int values[9]={0xF,0xF,0xF,0x7,0x3,0x1,0x0,0x0,0x0};// input 4 bit
    int durations[9]={20,20,20,20,20,20,20,20,20}; //time delay between data points
    int n;



        for(;;)
        {
            while (switchinput ==0) //once switch is on display the desired loop
    {
            for(n=0; n<9; n++)
            {
                output = values[n]; // Output Pattern
                wait_ms(durations[n]); // Delay till next Pattern
            }
        }
    }
}
```

- When button not pressed



- When button is pressed the output