

CSLR61 : EMBEDDED SYSTEMS

LAB-3

Roll no. : 106119100

Name : Rajneesh Pandey

Section : CSE-B

1. Increase the intensity of one LED while decreasing the other LED's intensity at the same unit. Display the intensity value of the LEDs.
 - Libraries Used: pwmout

Code :

```
#include "mbed.h"

PwmOut led1(p5);
PwmOut led2(p6);

int main(){
    led1 = 0.0;
    led2 = 1.0;
    while (true)
    {

        led1 = led1+0.1;
        led2 = led2-0.1;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1.0)
            led1 = 0.0;
        if (led2 == 0.0)
            led2 = 1.0;
    }
}
```

Output :

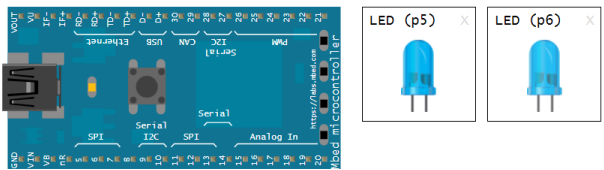
Mbed Simulator

simulator.mbed.com/#user_1644831342569

arm MBED

Blinky Load demo Run + Add component

```
1 #include "mbed.h"
2
3 PwmOut led1(p5);
4 PwmOut led2(p6);
5
6 int main() {
7     led1 = 0;
8     led2 = 1;
9
10
11     while(1) {
12
13         led1 = led1 + 0.10;
14         led2 = led2 - 0.10;
15
16         printf("LED1 is now %.2f\n", led1.read());
17         printf("LED2 is now %.2f\n\n", led2.read());
18
19         wait(1);
20
21         if(led1==1.0)
22             led1 = 0;
23
24         if(led2 == 0.0)
25             led2 = 1;
26     }
27 }
28 }
```



Serial output

```
LED2 is now 0.50
LED1 is now 0.40
LED2 is now 0.40

LED1 is now 0.50
LED2 is now 0.30

LED1 is now 0.60
LED2 is now 0.20

LED1 is now 0.70
LED2 is now 0.09
```

2. Interface potentiometer with mbed board and based on the value of potentiometer, adjust the intensity of an external LED.

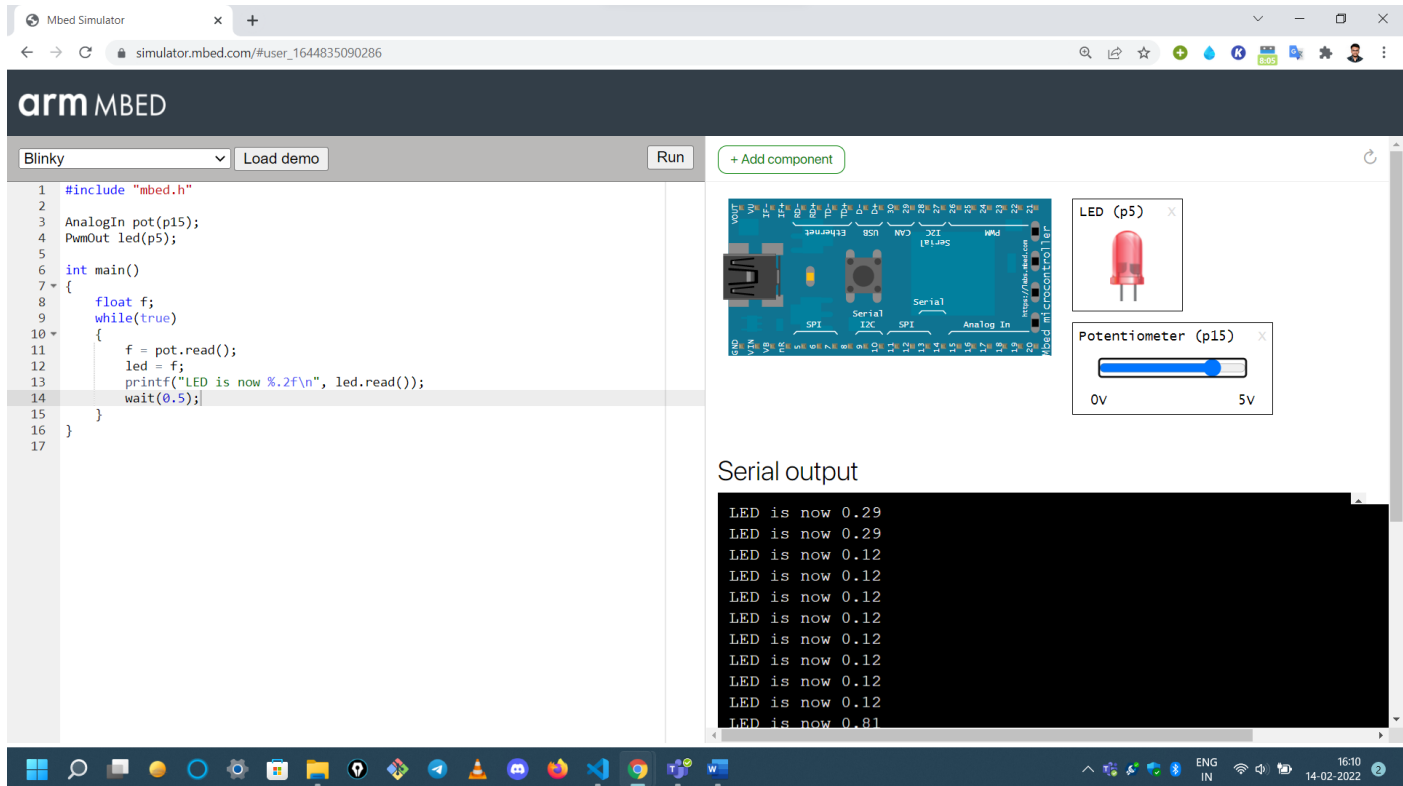
•Libraries To Be Used: pwmout, analogin

Code:

```
#include "mbed.h"

AnalogIn pot(p15);
PwmOut led(p5);

int main(){
    float f;
    while (true)
    {
        f = pot.read();
        led = f;
        printf("LED is now %.2f\n", led.read());
        wait(0.5);
    }
}
```



3. Interface switch, 2 LEDs, potentiometer with the mbed board. If the switch is on, control the intensity of the LEDs using potentiometer (one LED should be increasing and other should be decreasing at the rate given via potentiometer). If the switch is off, blink LEDs one at increasing rate and other at decreasing rate (same rate) and loop back.

Code:

```
#include "mbed.h"

AnalogIn pot(p15);
PwmOut led1(p5);
PwmOut led2(p6);

InterruptIn swt(p7);

void on()
{
    float f;
    while(true){
        f = pot.read();
        led1 = f;
        led2 = 1-led1;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1==1.0)
            led1 = 0.0;

        if (led2 == 0.0)
            led2 = 1.0;
    }
}

void off(){
    led1 = 0.0;
    led2 = 1.0;
    while (true)
    {
```

```

        led1 = led1+0.1;
        led2 = led2-0.1;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1.0)
            led1 = 0.0;
        if (led2 == 0.0)
            led2 = 1.0;
    }
}

int main(){
    swt.rise(&on);
    swt.fall(&off);
}

```

Output :

The screenshot displays the Mbed Simulator interface. On the left, the code editor shows the provided C code. The center pane shows a virtual ARM MBED board with various components connected: a potentiometer (p15), two LEDs (p5 and p6), and a switch (p7). On the right, there are controls for these components: a slider for the potentiometer (p15) ranging from 0V to 5V, and checkboxes for the LEDs (p5) and the switch (p7). The bottom right pane shows the serial output, which displays the following text:

```

LED2 is now 0.85
LED1 is now 0.15
LED2 is now 0.85
LED1 is now 0.15
LED2 is now 0.85
LED1 is now 0.15
LED2 is now 0.85
LED1 is now 0.15
LED2 is now 0.85
LED1 is now 0.15
LED2 is now 0.85
LED1 is now 0.15
LED2 is now 0.85
LED1 is now 0.15

```