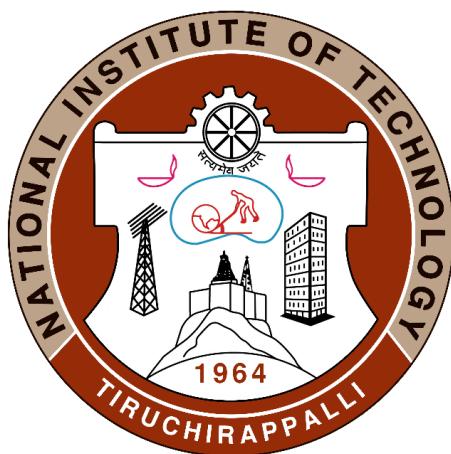


# CSLR61 - Embedded Systems Lab



## Lab Report

Name: **Rajneesh Pandey**

Roll No.: **106119100**

Class: **CSE-B**

Date: **02/05/2022**



<b>S. No.</b>	<b>Topic</b>	<b>Date</b>	<b>Lab</b>	<b>Page No.</b>
1	Blink LEDs in alternate order.	31-01-2022	1	5
2	Blink LEDs - count from 1 to 15			
3	Blink LEDs - composite number			
4	Blink LEDs - to count even numbers			
5	Switch between hex counter and decade counter using switches.	07-02-2022	2	9
6	Display the octal counter using the onboard LEDs using Ticker Object			
7	Display the odd counter using the onboard LEDs and screen			
8	Increase the intensity of one LED while decreasing the other LED's intensity at the same unit	14-02-2022	3	14
9	Interface potentiometer with mbed board and based on the value of potentiometer, adjust the intensity of an external LED.			
10	Interface switch, 2 LEDs, potentiometer with the mbed board. If the switch is on, control the intensity of the LEDs using potentiometer. If the switch is off, blink LEDs one at increasing rate.			
11	Write a program to increase the intensity of external LED and sound intensity of speaker.	28-02-2022	4	18
12	Design an alarm system, which makes double beep sound for every 5s using a speaker.			
13	Have a switch, 2 LEDs and a speaker interfaced with mbed board. Upon switching			

	on, the intensity of the sound should be increasing and the upon switching off, should make the intensity of the sound decreasing.			
14	Design a program to make ambulance sound.			
15	Blink LEDs in alternate order - 1 and 3 together and 2 and 4 together.			23
16	Blink LEDs - count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec.	07-03-2022	5	
17	Blink LEDs - for all composite number below			
18	Blink LEDs - to count even numbers			
19	Interface an ultrasonic sensor and display the distance measured in a screen.	28-03-2022	6	29
20	Interface a PIR sensor with Arduino and blink a led if any movement is captured by the sensor.			
21	Interface two 7-segment display with Arduino Uno board. a)odd counter and b)even counter			35
22	Interface the ambient light sensor with Arduino Uno board. Check the light value from the sensor and switch on/off the bulb.	11-04-2022	7	
23	Interface the temperature and gas sensor with Arduino Uno board. Check the temperature and the gas value.			
24	Interface force sensor with Arduino board and display the amount of force given as input to sensor in LCD screen.	18-04-2022	8	43
25	Interface the keypad and tilt sensor with Arduino board, if the sensor is being tilted, then take input from keypad and print it in the LCD.			

## Questions

Lab 1

Date: 31-01-2022

1. Blink LEDs in alternate order - 1 and 3 together and 2 and 4 together.
2. Blink LEDs - count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec
3. Blink LEDs - for all composite number below 15
4. Blink LEDs - to count even numbers

Lab 2

Date: 07-02-2022

1. Switch between hex counter and decade counter using switches. Display the current value of the counter with the help of 4 LEDs and the screen.
2. Display the octal counter using the onboard LEDs using Ticker Object; After 10 second, reset all the values of the onboard LED using Timeout Object.
3. Display the odd counter using the onboard LEDs and screen; After 10 seconds, reset all the values of the onboard LED and display even counter using Timeout Object.

Lab 3

Date: 14-02-2022

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

Lab 4

Date: 28-02-2022

1. Write a program to increase the intensity of external LED. At the same rate, increase the sound intensity of speaker.
2. Design an alarm system, which makes double beep sound for every 5s using a speaker.
3. Have a switch, 2 LEDs and a speaker interfaced with mbed board. Upon switching on, the intensity of the sound should be increasing and the upon switching off, should make the intensity of the sound decreasing. Proportionally increase or decrease the intensity of two LEDs.
4. Design a program to make ambulance sound.

Lab 5

Date: 07-03-2022

1. Blink LEDs in alternate order - 1 and 3 together and 2 and 4 together.
2. Blink LEDs - count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec
3. Blink LEDs - for all composite number below 15

4. Blink LEDs - to count even numbers

#### Lab 6

Date: 28-03-2022

1. Interface an ultrasonic sensor and display the distance measured in a screen and glow red, blue, and green led if the distance is close, moderate, and far respectively. Play different tones for different cases.
2. Interface a PIR sensor with Arduino and blink a led if any movement is captured by the sensor and count the number of movements. Make a buzzer when movement detected.

#### Lab 7

Date: 11-04-2022

1. Interface two 7-segment display with Arduino Uno board.
2. Implement the a)odd counter and b)even counter and display the values in the seven-segment display.
3. Interface the ambient light sensor with Arduino Uno board. Check the light value from the sensor and switch on/off the bulb (based on the threshold value).
4. Interface the temperature and gas sensor with Arduino Uno board. Check the temperature and the gas value, if the limit is beyond the threshold, switch on the bulb and make alarm using buzzer.

#### Lab 8

Date: 18-04-2022

1. Interface force sensor with Arduino board and display the amount of force given as input to sensor in LCD screen. Also, interface LED RGB and change colour of the LED based on some threshold Values.
2. Interface the keypad and tilt sensor with Arduino board, if the sensor is being tilted, then take input from keypad and print it in the LCD

# Lab 1

Date: 31-01-2022

1. Blink LEDs in alternate order – 1 and 3 together and 2 and 4 together.

```
#include "mbed.h"

DigitalOut led1(LED1);
DigitalOut led2(LED2);
DigitalOut led3(LED3);
DigitalOut led4(LED4);

int main(){
    while (1){
        led1 = !led1;
        printf("Blink! LED1 is now %d\n", led1.read());
        led3 = !led3;
        printf("Blink! LED3 is now %d\n", led3.read());
        wait_ms(1000);
        led2 = !led2;
        printf("Blink! LED2 is now %d\n", led2.read());
        led4 = !led4;
        printf("Blink! LED4 is now %d\n", led4.read());
    }
}
```

The screenshot shows the arm mbed online compiler interface. On the left, the code for 'Blink' is displayed in a text editor. The code initializes four digital outputs (led1, led2, led3, led4) and enters a loop where it toggles led1 and led3, then led2 and led4, with a 1-second delay between each group. In the center, a pinout diagram of an mbed board is shown, highlighting the pins used: Vout, GND, USB, Ethernet, SPI, I2C, Serial, Analog In, and Microcontroller. On the right, the 'Serial output' window displays the real-time console logs of the program's execution.

```
Blink! LED4 is now 0
Blink! LED1 is now 1
Blink! LED3 is now 1
Blink! LED2 is now 1
Blink! LED4 is now 1
Blink! LED1 is now 0
Blink! LED3 is now 0
Blink! LED2 is now 0
Blink! LED4 is now 0
Blink! LED1 is now 1
Blink! LED3 is now 1
Blink! LED2 is now 1
Blink! LED4 is now 1
Blink! LED1 is now 0
```

2. Blink LEDs - count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec

```
#include "mbed.h"

BusOut myleds(LED1, LED2, LED3, LED4);

int main(){
    while (1){
        for (int i = 1; i < 16; i++){
            myleds = i;
            printf("%d ", i);
            if (i & 1){
                printf("odd count\n");
                wait(1);
            }
            else{
                printf("even count\n");
                wait(2);
            }
        }
    }
}
```

arm MBED

Blinky with BusOut Load demo Run

+ Add component

The screenshot shows the arm MBED software interface. On the left is the code editor with the provided C code for a mbed application. In the center is the component palette showing a breadboard with various pins and components like SPI, I2C, Serial, and Analog In. On the right is the serial output terminal displaying the printed text from the code.

Serial output

```
1 odd count
2 even count
3 odd count
4 even count
5 odd count
6 even count
7 odd count
8 even count
9 odd count
10 even count
11 odd count
12 even count
13 odd count
14 even count
```

### 3. Blink LEDs - for all composite number below 15

```
#include "mbed.h"

BusOut myleds(LED1, LED2, LED3, LED4);

bool isComposite(int n){
    if (n == 1)
        return true;
    for (int i = 2; i < n; i++){
        if (n % i == 0)
            return true;
    }
    return false;
}

int main(){
    while (1){
        for (int i = 1; i < 16; i++){
            if (isComposite(i)){
                myleds = i;
                printf("%d is Composite Number\n", i);
                wait(1);
            }
        }
    }
}
```

arm MBED

Blinky with BusOut + Load demo Run + Add component

The screenshot shows the arm MBED IDE interface. On the left is the code editor with the provided C code. In the center is the component palette showing various hardware components like a breadboard, servo, motor, and various connectors. On the right is the serial output terminal displaying the results of the program execution.

Serial output

```
1 is Composite Number
4 is Composite Number
6 is Composite Number
8 is Composite Number
9 is Composite Number
10 is Composite Number
12 is Composite Number
14 is Composite Number
15 is Composite Number
1 is Composite Number
4 is Composite Number
6 is Composite Number
8 is Composite Number
9 is Composite Number
```

#### 4. Blink LEDs - to count even numbers

```
#include "mbed.h"

BusOut leds(LED1, LED2, LED3, LED4);

int main()
{
    while (1)
    {
        for (int i = 0; i < 16; i++)
        {
            if (i % 2 == 0)
            {
                leds = i;
                printf("count %d\n", leds.read());
                wait(2);
            }
        }
    }
}
```

The screenshot shows the arm MBED online compiler interface. On the left, the code for 'Blinky' is displayed in a text editor. In the center, a pinout diagram of an Mbed microcontroller is shown, with pins labeled from 1 to 26. On the right, a terminal window displays the 'Serial output' showing the counts of even numbers from 0 to 14.

```
1 #include "mbed.h"
2
3 BusOut leds(LED1, LED2, LED3, LED4);
4
5 int main(){
6     while (1){
7         for(int i = 0; i < 16; i++){
8             if(i%2==0){
9                 leds = i;
10                printf("count %d\n", leds.read());
11                wait(2);
12            }
13        }
14    }
15 }
```

Serial output

```
count 0
count 2
count 4
count 6
count 8
count 10
count 12
count 14
count 0
count 2
```

# Lab 2

Date: 07-02-2022

1. Switch between hex counter and decade counter using switches. Display the current value of the counter with the help of 4 LEDs and the screen.

```
#include "mbed.h"
BusOut leds(LED1, LED2, LED3, LED4);
InterruptIn swt(p5);

void hexCounter()
{
    while (1)
    {
        for (int i = 0; i < 16; i++)
        {
            leds = i;
            printf("Hex count : %d\n", i);
            wait(1);
        }
    }
}

void decCounter()
{
    while (1)
    {
        for (int i = 0; i < 10; i++)
        {
            leds = i;
            printf("Dec count : %d\n", i);
            wait(1);
        }
    }
}

int main()
{
    swt.rise(&hexCounter);
    swt.fall(&decCounter);
}
```

The screenshot shows the arm MBED online simulator interface. On the left, there is a code editor with the following C code:

```

1 #include "mbed.h"
2
3 BusOut leds(LED1, LED2, LED3, LED4);
4 InterruptIn swt(p5);
5
6 void hexCounter(){
7     while(1){
8         for(int i = 0; i < 16; i++) {
9             leds = i;
10            printf("Hex count : %d\n",i);
11            wait(1);
12        }
13    }
14 }
15 void decCounter(){
16     while(1){
17         for(int i = 0; i < 10; i++) {
18             leds = i;
19             printf("Dec count : %d\n",i);
20             wait(1);
21         }
22     }
23 }
24 int main() {
25     swt.rise(&hexCounter);
26     swt.fall(&decCounter);
27 }

```

In the center, there is a component palette showing a 'Switch (p5)' component. On the right, the 'Serial output' window displays the following text:

```

Hex count : 1
Hex count : 2
Hex count : 3
Dec count : 0
Dec count : 1
Hex count : 0
Hex count : 1
Hex count : 2
Hex count : 3

```

2. Display the octal counter using the onboard LEDs using Ticker Object; After 10 second, reset all the values of the onboard LED using Timeout Object.

```

#include "mbed.h"

BusOut leds(LED1, LED2, LED3, LED4);
Ticker tck;
Timeout to;

int octCount = 0;

void octalCounter()
{
    leds = octCount;
    printf("counter %d\n", leds.read());
    octCount = (octCount + 1) % 8;
}

void reset()
{
    printf("resetting...");
    octCount = 0;
    leds = 0;
    printf("Value after resetting %d\n", leds.read());
}

```

```

int main()
{
    tck.attach(&octalCounter, 2);
    to.attach(&reset, 10);
}

```

Blinky

Load demo

Run

+ Add component

Serial output

```

counter 0
counter 1
counter 2
counter 3
counter 4
resetting...Value after resetting 0
counter 0
counter 1
counter 2
counter 3

```

3. Display the odd counter using the onboard LEDs and screen;  
After 10 seconds, reset all the values of the onboard LED and display even counter using Timeout Object.

```

#include "mbed.h"

BusOut leds(LED1, LED2, LED3, LED4);
Ticker tck;
Timeout to;
InterruptIn swt(p5);

int state = 1;
int odd = 1;

void Counter()
{
    while (1)
    {
        for (int i = 0; i < 16; i++)
        {

```

```

        if (i % 2 == state)
    {
        leds = i;
        if (state)
            printf("Odd Counter %d\n", leds.read());
        else
        {
            printf("Even Counter %d\n", leds.read());
        }
        wait(1);
    }
}

void Toggle()
{
    state = !state;
    printf("Toggle using switch...\n");
    Counter();
}

void reset()
{
    printf("resetting Odd Counter...\n");
    state = 0;
    leds = 0;
    tck.detach();
    printf("Starting Even Counter...\n");
    Counter();
}
void oddCounter()
{
    leds = odd;
    odd = (odd + 2) % 16;
    printf("Odd Counter %d\n", leds.read());
}

int main()
{
    tck.attach(&oddCounter, 2.0);
    to.attach(&reset, 10);
    swt.rise(&Toggle);
}

```

## arm MBED

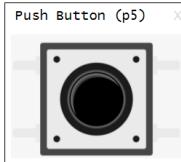
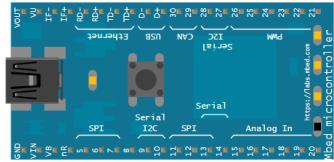
Blinky

Load demo

Run

+ Add component

```
1 #include "mbed.h"
2
3 BusOut leds(LED1, LED2, LED3, LED4);
4 Ticker tck;
5 Timeout to;
6 InterruptIn swt(p5);
7
8 int state = 1;
9 int odd = 1;
10
11 void Counter(){
12     while(1){
13         for(int i = 0; i < 16; i++){
14             if (i % 2 == state){
15                 leds = i;
16                 if (state)
17                     printf("Odd Counter %d\n", leds.read());
18                 else{
19                     printf("Even Counter %d\n", leds.read());
20                 }
21                 wait(1);
22             }
23         }
24     }
25 }
26
27 void Toggle(){
28     state = !state;
29     printf("Toggle using switch...\n");
30     Counter();
31 }
32
33 void reset(){
34     printf("resetting Odd Counter...\n");
35     state = 0;
```



## Serial output

```
Odd Counter 9
resetting Odd Counter...
Starting Even Counter...
Even Counter 0
Even Counter 2
Even Counter 4
Even Counter 6
Even Counter 8
Even Counter 10
Even Counter 12
Even Counter 14
Even Counter 0
Toggle using switch...
Odd Counter 1
```

# Lab 3

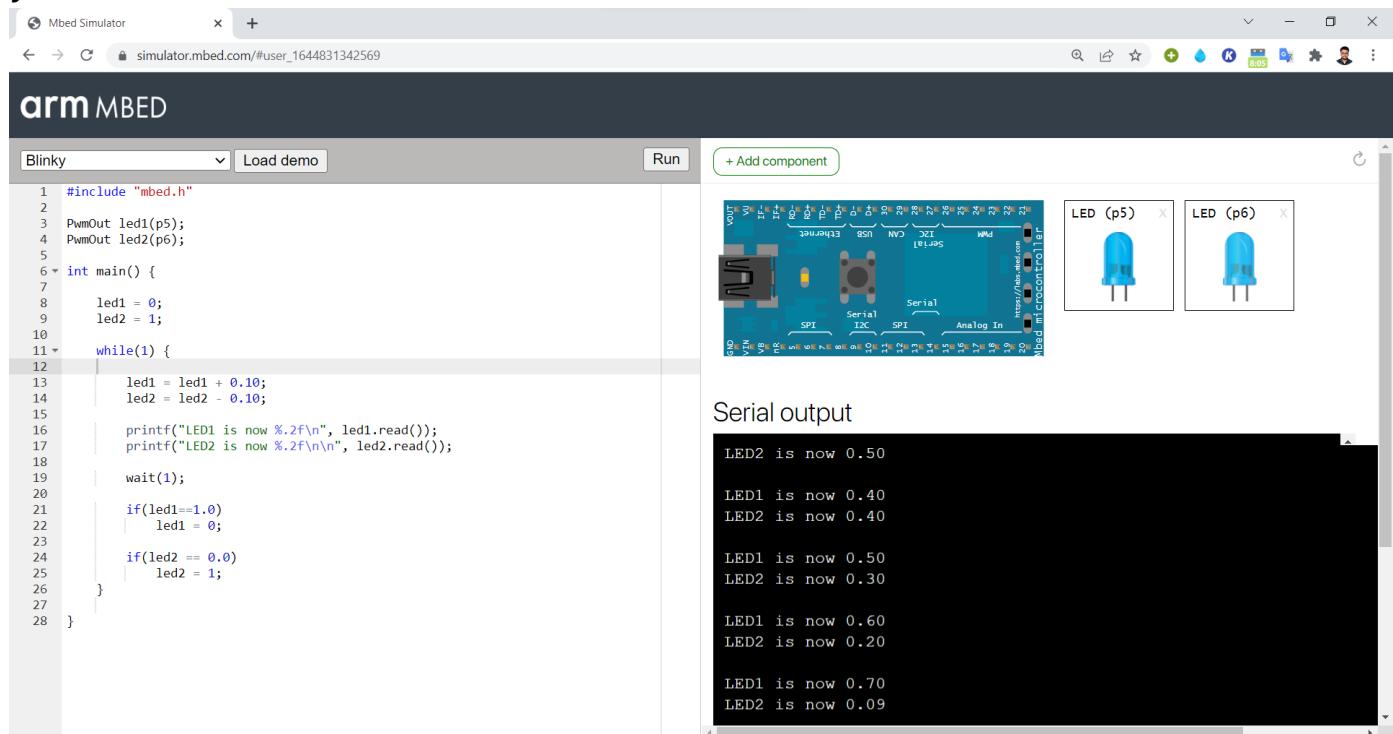
Date: 14-02-2022

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.

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

```
PwmOut led1(p5), 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;
    }
}
```

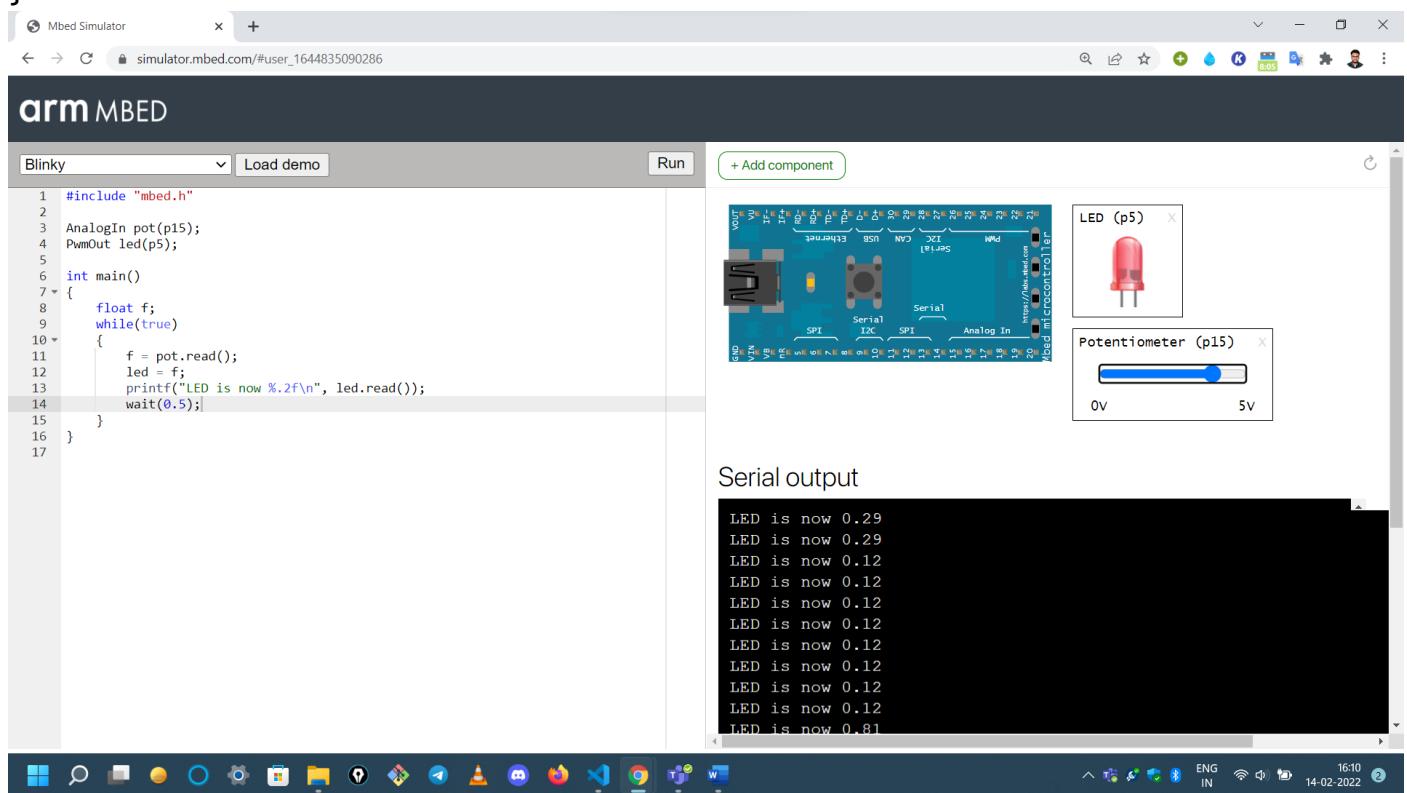


## 2. Interface potentiometer with mbed board and based on the value of potentiometer, adjust the intensity of an external LED.

```
#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.

```
#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;

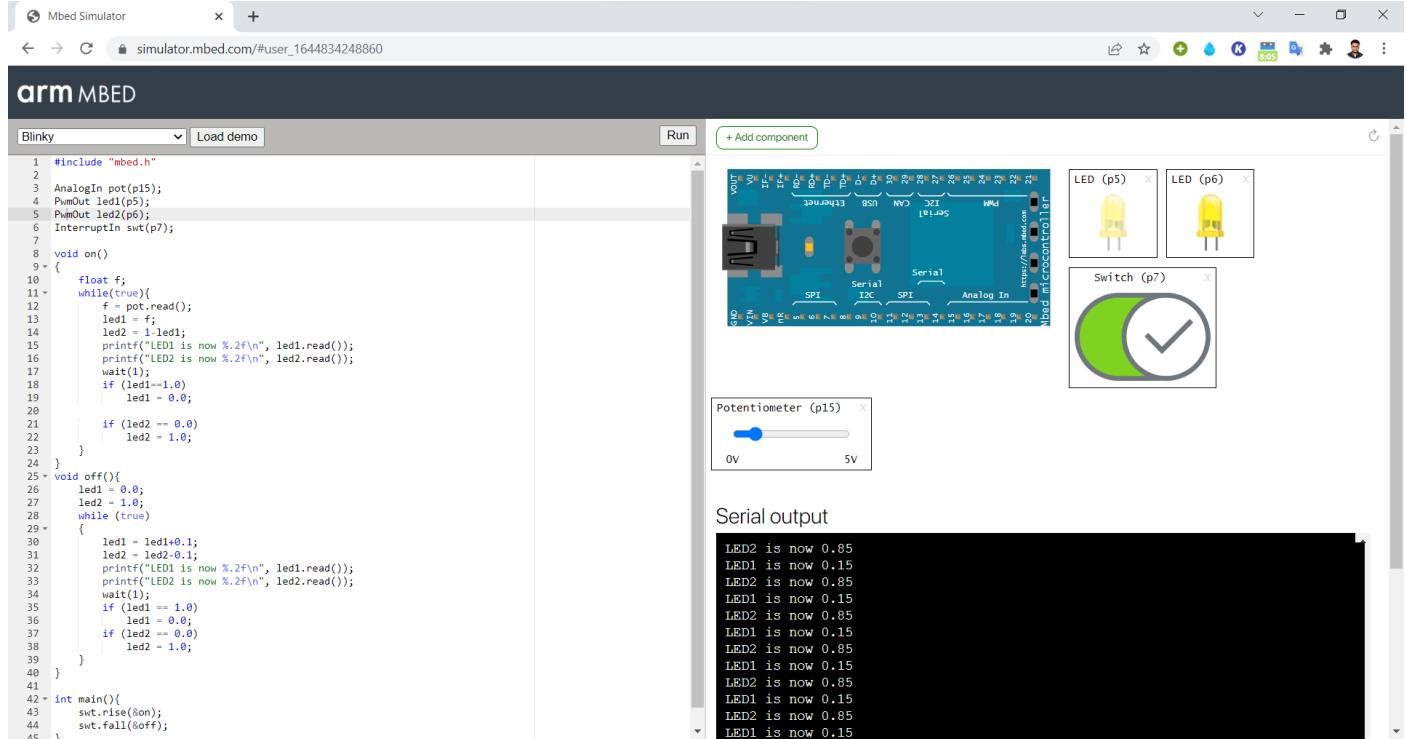
        if (led2 == 0.0)
            led2 = 1.0;
    }
}

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);
}
```



# Lab 4

Date: 28-02-2022

1. Write a program to increase the intensity of external LED. At the same rate, increase the sound intensity of speaker.

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

```
PwmOut led(p5);
PwmOut spker(p6);

int main()
{
    led = 0.0;
    spker = led;
    while (true)
    {

        led = led + 0.1;
        spker = led;
        printf("LED is now %.2f\n", led.read());
        printf("SPEAKER is now %.2f\n", spker.read());
        wait(1);
    }
}
```

The screenshot shows the Mbed Simulator interface. On the left, the code editor displays the provided C++ code for a 'Blinky' application. On the right, the component palette lists an 'LED (p5)' and a 'PWM Speaker (p21)'. The central workspace shows a breadboard with these components connected to pins p5 and p21 respectively. Below the workspace, a terminal window titled 'Serial output' displays the real-time logs of the program's execution, showing the increasing values of the LED and speaker intensities over time.

```
1 #include "mbed.h"
2
3 PwmOut led(p5);
4 PwmOut spker(p21);
5
6 int main(){
7     led = 0.0;
8     spker = led;
9     while (true)
10    {
11         led = led+0.1;
12         spker = led;
13         printf("LED is now %.2f\n", led.read());
14         printf("SPEAKER is now %.2f\n", spker.read());
15         wait(1);
16     }
17 }
```

Serial output

```
SPEAKER is now 0.50
LED is now 0.60
SPEAKER is now 0.60
LED is now 0.70
SPEAKER is now 0.70
LED is now 0.80
SPEAKER is now 0.80
LED is now 0.90
SPEAKER is now 0.90
LED is now 1.00
SPEAKER is now 1.00
```

## 2. Design an alarm system, which makes double beep sound for every 5s using a speaker.

```
#include "mbed.h"

PwmOut speaker(p21);
Ticker tck;

void beepTwice(){
    speaker = 0.0;
    speaker = 1.0;
    printf("Beep");
    wait(0.2);
    speaker = 0.0;
    wait(0.2);
    speaker = 1.0;
    printf("Beep\n");
    wait(0.2);
    speaker = 0.0;
}

int main()
{
    beepTwice();
    tck.attach(&beepTwice, 5);
}
```

The screenshot shows the Mbed Simulator interface. On the left, the code editor displays the provided C++ code for a PWM speaker. In the center, the component library shows a speaker icon labeled "PWM Speaker (p21)". On the right, the serial output window shows the text "BeepBeep" repeated five times, indicating the speaker is beeping at 5-second intervals.

```
Pwm Speaker      Load demo Run + Add component
1 #include "mbed.h"
2
3 PwmOut speaker(p21);
4 Ticker tck;
5
6 void beepTwice()
7 {
8     speaker = 0.0;
9     speaker = 1.0;
10    printf("Beep");
11    wait(0.2);
12    speaker = 0.0;
13    wait(0.2);
14    speaker = 1.0;
15    printf("Beep\n");
16    wait(0.2);
17    speaker = 0.0;
18 }
19
20 int main()
21 {
22     beepTwice();
23     tck.attach(&beepTwice, 5);
24 }
```

Serial output

```
BeepBeep
BeepBeep
BeepBeep
BeepBeep
BeepBeep
```

3. Have a switch, 2 LEDs and a speaker interfaced with mbed board. Upon switching on, the intensity of the sound should be increasing and the upon switching off, should make the intensity of the sound decreasing. Proportionally increase or decrease the intensity of two LEDs.

```
#include "mbed.h"

PwmOut led1(p5);
PwmOut led2(p6);
PwmOut spker(p21);

InterruptIn swt(p7);

void on()
{
    while (true)
    {
        spker = spker + 0.1;
        led1 = spker;
        led2 = spker;
        printf("Speaker is now %.2f\n", spker.read());
        printf("Led1 is now %.2f\n", led1.read());
        printf("Led2 is now %.2f\n", led2.read());
        wait(2);
    }
}

void off()
{
    while (true)
    {
        spker = spker - 0.1;
        led1 = spker;
        led2 = spker;
        printf("Speaker is now %.2f\n", spker.read());
        printf("Led1 is now %.2f\n", led1.read());
        printf("Led2 is now %.2f\n", led2.read());
        wait(2);
    }
}

int main()
{
```

```

spker = 0.5;
swt.rise(&on);
swt.fall(&off);
}

```

The screenshot shows the arm MBED online simulator interface. On the left, the code for "Blinky" is displayed:

```

1 #include "mbed.h"
2
3 PwmOut led1(p5);
4 PwmOut led2(p6);
5 PwmOut spker(p21);
6
7 InterruptIn swt(p7);
8
9 void on()
10 {
11     while (true)
12     {
13         spker = spker + 0.1;
14         led1 = spker;
15         led2 = spker;
16         printf("Speaker is now %.2f\n", spker.read());
17         printf("Led1 is now %.2f\n", led1.read());
18         printf("Led2 is now %.2f\n", led2.read());
19         wait(2);
20     }
21 }
22 void off()
23 {
24     while (true)
25     {
26         spker = spker - 0.1;
27         led1 = spker;
28         led2 = spker;
29         printf("Speaker is now %.2f\n", spker.read());
30         printf("Led1 is now %.2f\n", led1.read());
31         printf("Led2 is now %.2f\n", led2.read());
32         wait(2);
33     }
34 }
35
36 int main()
37 {
38     spker = 0.5;
39     swt.rise(&on);
40     swt.fall(&off);
41 }
42

```

The breadboard diagram shows a standard breadboard setup with a speaker connected to pin p21, two LEDs (one red, one blue) connected to pins p5 and p6, and a switch connected to pin p7. The right side of the interface shows component libraries for a PWM Speaker, LED (p5), LED (p6), and a Switch (p7).

The Serial output window displays the following text:

```

Led1 is now 0.60
Led2 is now 0.60
Speaker is now 0.50
Led1 is now 0.50
Led2 is now 0.50
Speaker is now 0.40
Led1 is now 0.40
Led2 is now 0.40
Speaker is now 0.50
Led1 is now 0.50

```

#### 4. Design a program to make ambulance sound.

```

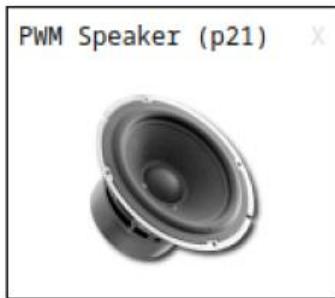
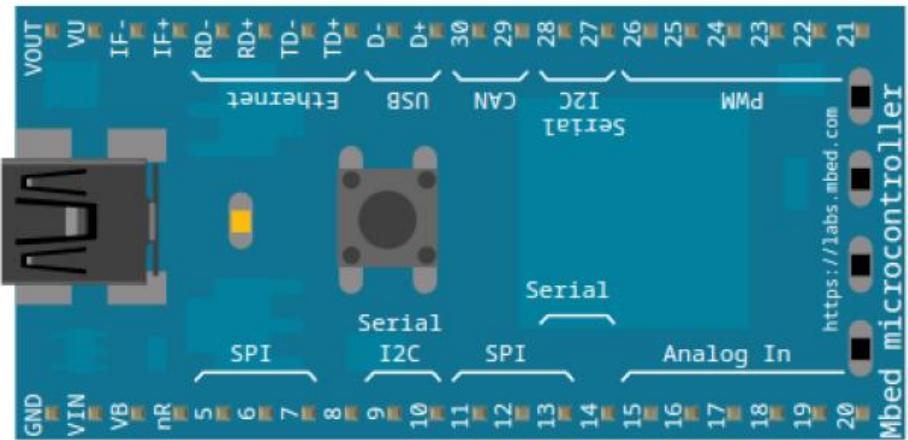
#include "mbed.h"
PwmOut speaker(p21);
int main()
{
    int i;
    while (1)
    {
        for (i = 0; i < 26; i++)
        {
            speaker.period(1.0 / 259.0);
            speaker = float(i) / 500.0;
            wait(0.5);
            speaker.period(1.0 / 340.0);
            wait(0.5);
        }
        for (i = 26; i >= 0; i--)

```

```

    {
        speaker.period(1.0 / 259.0);
        speaker = float(i) / 500.0;
        wait(0.5);
        speaker.period(1.0 / 440.0);
        wait(0.5);
    }
}

```



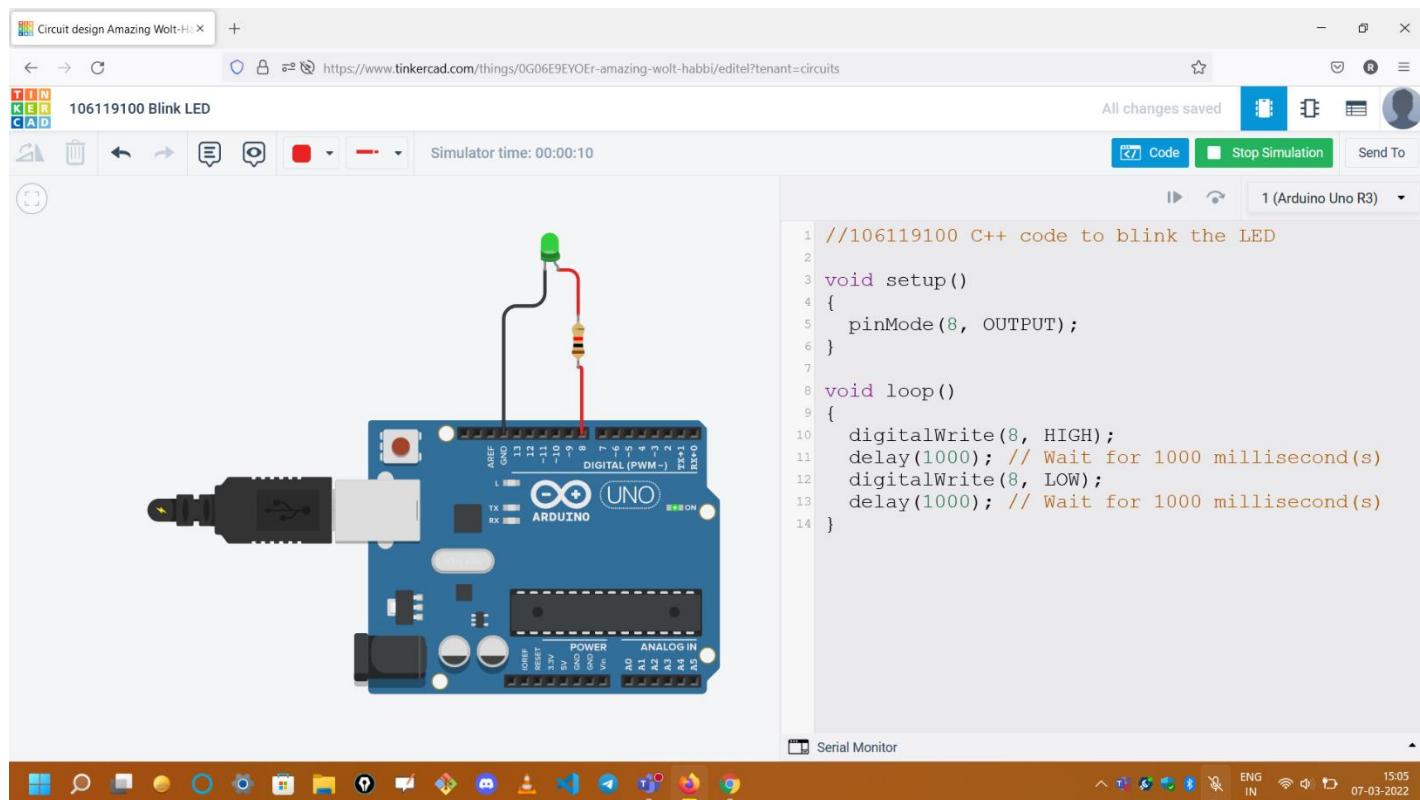
# Lab 5

Date: 07-03-2022

1. Blink LEDs in alternate order - 1 and 3 together and 2 and 4 together.

```
void setup()
{
    pinMode(8, OUTPUT);
}

void loop()
{
    digitalWrite(8, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(8, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
}
```

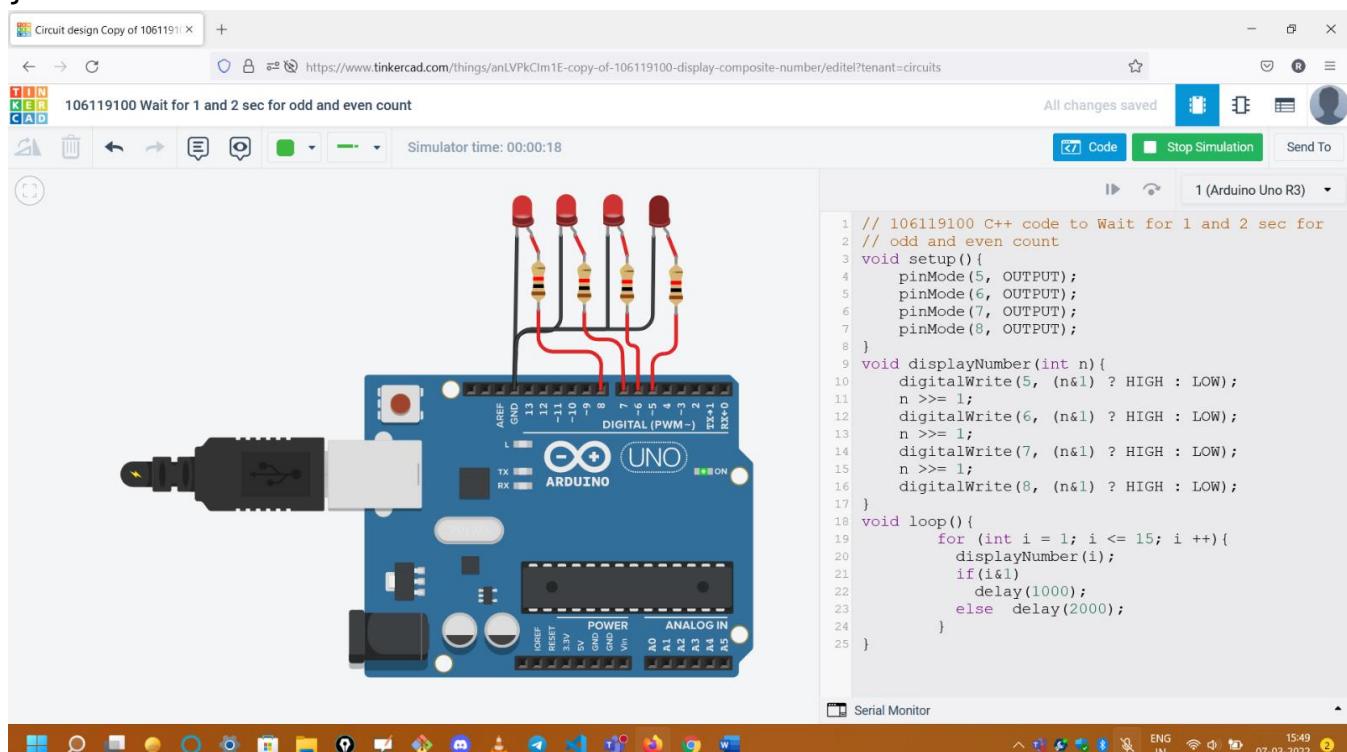


2. Blink LEDs - count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec

```
void setup(){
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}

void displayNumber(int n){
    digitalWrite(5, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(6, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(7, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(8, (n & 1) ? HIGH : LOW);
}

void loop(){
    for (int i = 1; i <= 15; i++){
        displayNumber(i);
        if (i & 1)
            delay(1000);
        else
            delay(2000);
    }
}
```



### 3.Blink Alternate LEDs

```
void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}
void loop()
{
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(6, HIGH);
    delay(2000);
    digitalWrite(8, LOW);
    digitalWrite(6, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(7, HIGH);
    delay(2000);
}
```

The screenshot shows the Tinkercad interface for a project titled "106119100 Blink Alternative LED". The circuit diagram at the top features four red LEDs connected in series. The first LED is connected to digital pin 5, the second to pin 7, the third to pin 8, and the fourth to pin 6. All four LEDs have their common ground connection tied to ground. The Arduino Uno R3 board is shown below the circuit, with its pins labeled. The code editor on the right contains the C++ code for the Arduino Uno R3, which sets up pins 5, 6, 7, and 8 as outputs and then enters a loop where it alternates the state of these pins every 2 seconds. The code is as follows:

```
// 106119100 C++ code to Blink Alternative LED
void setup() {
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}
void loop() {
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(6, HIGH);
    delay(2000);
    digitalWrite(8, LOW);
    digitalWrite(6, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(7, HIGH);
    delay(2000);
}
```

#### 4. Blink LEDs - for all composite number below 15

```
// 106119100 C++ code to count composite number

void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}

void displayNumber(int n)
{
    digitalWrite(5, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(6, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(7, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(8, (n & 1) ? HIGH : LOW);
}

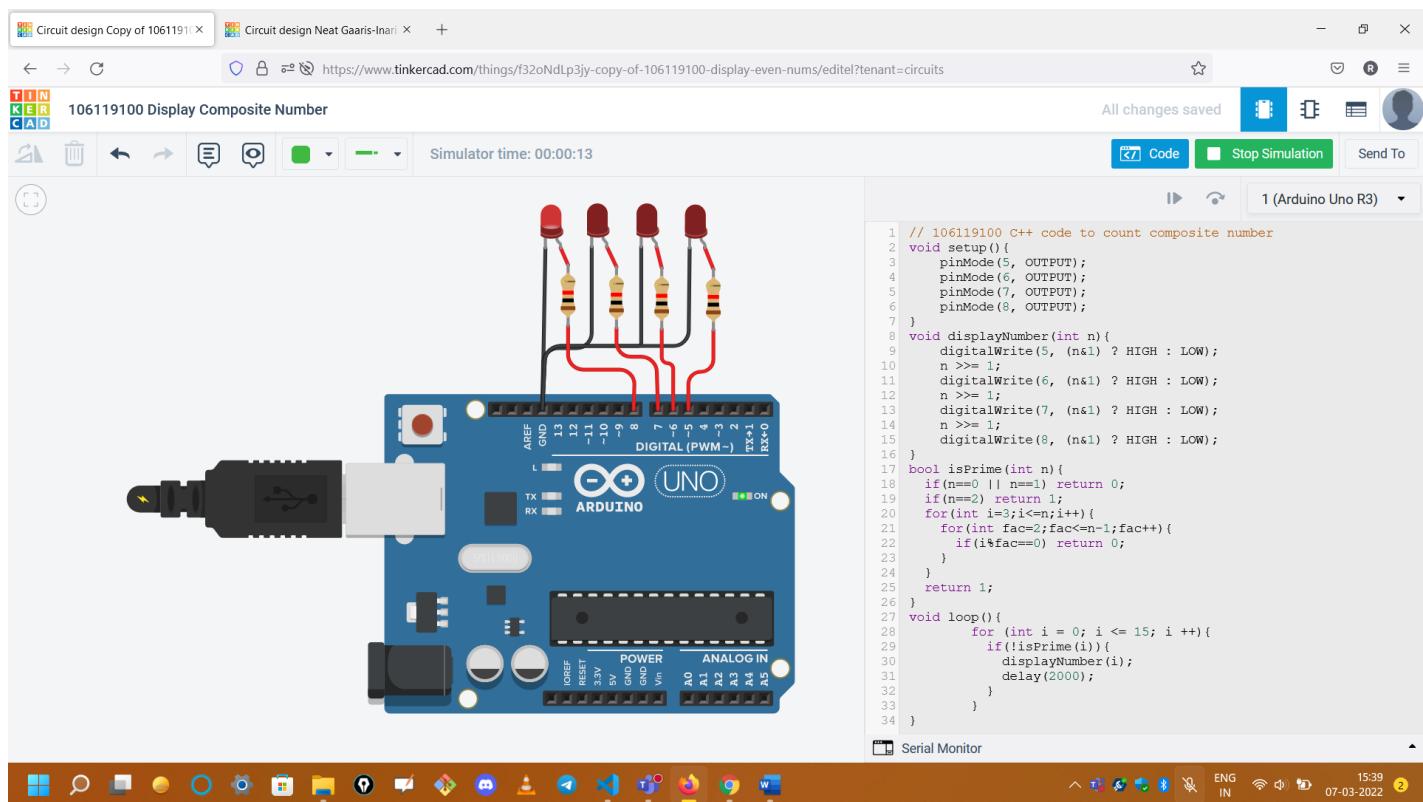
bool isPrime(int n)
{
    if (n == 0 || n == 1)
        return 0;
    if (n == 2)
        return 1;
    for (int i = 3; i <= n; i++)
    {
        for (int fac = 2; fac <= n - 1; fac++)
        {
            if (i % fac == 0)
                return 0;
        }
    }
    return 1;
}

void loop()
{
    for (int i = 0; i <= 15; i++)
    {
        if (!isPrime(i))
        {
            displayNumber(i);
        }
    }
}
```

```

        delay(2000);
    }
}
}

```



## 5. Blink LEDs – to count even numbers

```

// 106119100 C++ code to blink the LED

void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}

void displayNumber(int n)
{
    digitalWrite(5, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(6, (n & 1) ? HIGH : LOW);
    n >>= 1;
}

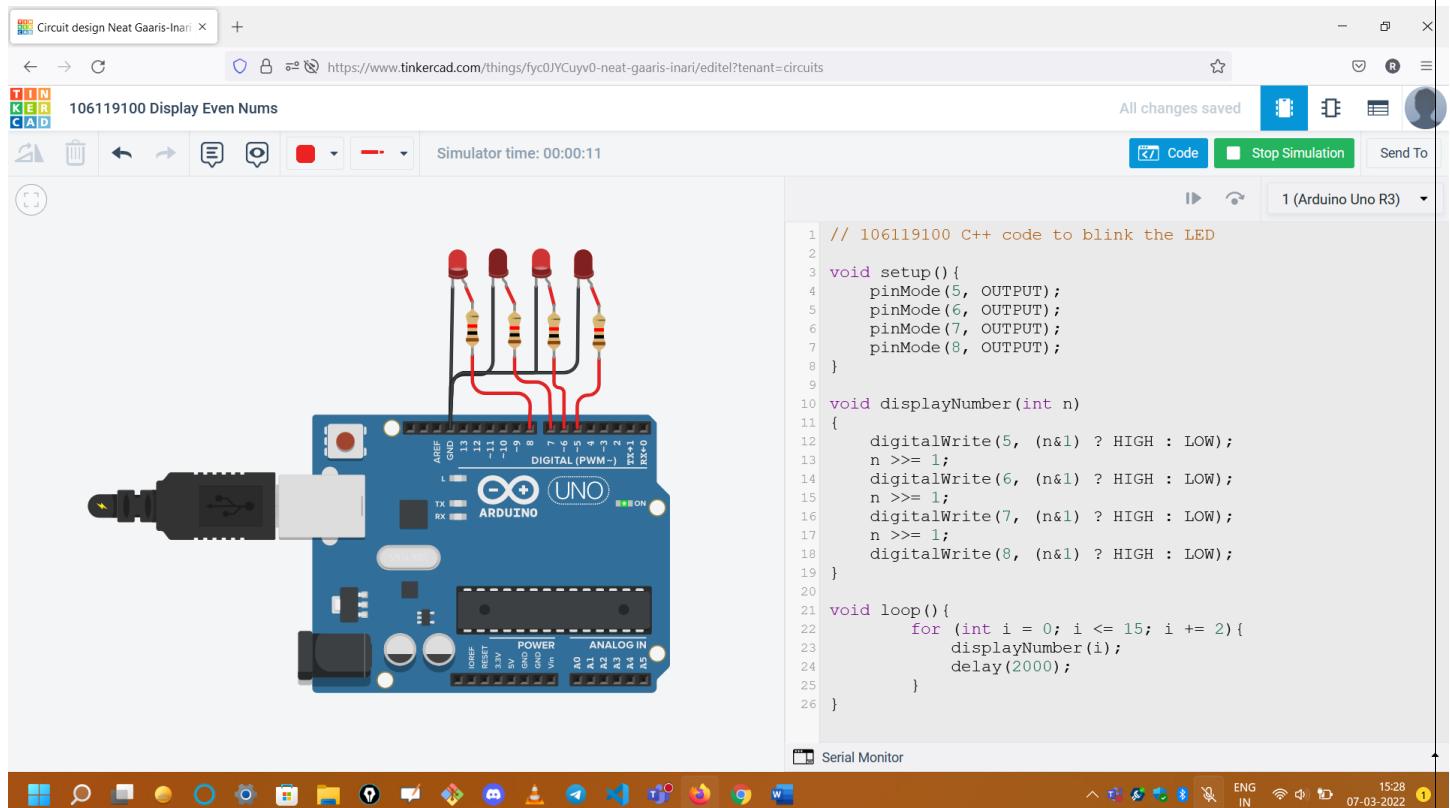
```

```

digitalWrite(7, (n & 1) ? HIGH : LOW);
n >>= 1;
digitalWrite(8, (n & 1) ? HIGH : LOW);
}

void loop()
{
    for (int i = 0; i <= 15; i += 2)
    {
        displayNumber(i);
        delay(2000);
    }
}

```



# Lab 6

Date: 28-03-2022

1. Interface an ultrasonic sensor and display the distance measured in a screen and glow red, blue, and green led if the distance is close, moderate, and far respectively. Play different tones for different cases.

```
#define F(x) x ? HIGH : LOW
const int red = 3;
const int blue = 4;
const int green = 5;
const int speaker = 10;
const int ping = 7;

void setup()
{
    // initialize serial communication:
    Serial.begin(9600);
    pinMode(red, OUTPUT);
    pinMode(blue, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(speaker, OUTPUT);
}

void displayLight(int code)
{
    digitalWrite(red, F(code == 0));
    digitalWrite(blue, F(code == 1));
    digitalWrite(green, F(code == 2));
}

void playTone(long duration, int freq)
{
    duration *= 1000;
    int period = (1.0 / freq) * 100000;
    long elapsed_time = 0;
    while (elapsed_time < duration)
    {
        digitalWrite(speaker, HIGH);
        delayMicroseconds(period / 2);
        digitalWrite(speaker, LOW);
        delayMicroseconds(period / 2);
        elapsed_time += (period);
    }
}
```

```

    }
}

void loop()
{
    long duration, inches, cm;
    pinMode(ping, OUTPUT);
    digitalWrite(ping, LOW);
    delayMicroseconds(2);
    digitalWrite(ping, HIGH);
    delayMicroseconds(5);
    digitalWrite(ping, LOW);
    pinMode(ping, INPUT);
    duration = pulseIn(ping, HIGH);
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);
    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    if (cm < 50)
    {
        displayLight(0);
        playTone(300, 100);
    }
    else if (cm < 150)
    {
        displayLight(1);
        playTone(300, 200);
    }
    else
    {
        displayLight(2);
        playTone(300, 300);
    }
    delay(100);
}
long microsecondsToInches(long microseconds)
{
    return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

{}

Circuit design Frantic Amur-Esb... MUTED

Circuit design Sizzling Bruticus... +

<https://www.tinkercad.com/things/bYCB0XWRnFa-frantic-amur-esboo/editel?tenant=circuits>

All changes saved

TIN KER CAD 106119100 Lab6-1 Interface an ultrasonic sensor

Simulator time: 00:00:21

Code Stop Simulation Send To 1 (Arduino Uno R3)

Ultrasonic Distance Sensor  
Name 1

```

56     {
57         displayLight(0);
58         playTone(300, 100);
59     }
60     else if (cm < 150)
61     {
62         displayLight(1);
63         playTone(300, 200);
64     }
65     else
66     {
67         displayLight(2);
68         playTone(300, 300);
69     }
70     delay(100);
71 }
72 long microsecondsToInches(long microseconds)
73 {
74     return microseconds / 74 / 2;
75 }
76 long microsecondsToCentimeters(long microseconds)
77 {
78     return microseconds / 29 / 2;
79 }

```

Serial Monitor

106in, 271cm  
67in, 173cm  
67in, 173cm  
67in, 173cm  
67in, 173cm  
67in, 173cm  
67in, 173cm  
67in, 173cm

Send Clear

Circuit design Frantic Amur-Esb... MUTED

Circuit design Sizzling Bruticus... +

<https://www.tinkercad.com/things/bYCB0XWRnFa-frantic-amur-esboo/editel?tenant=circuits>

All changes saved

TIN KER CAD 106119100 Lab6-1 Interface an ultrasonic sensor

Simulator time: 00:00:34.135

Code Stop Simulation Send To 1 (Arduino Uno R3)

Ultrasonic Distance Sensor  
Name 1

```

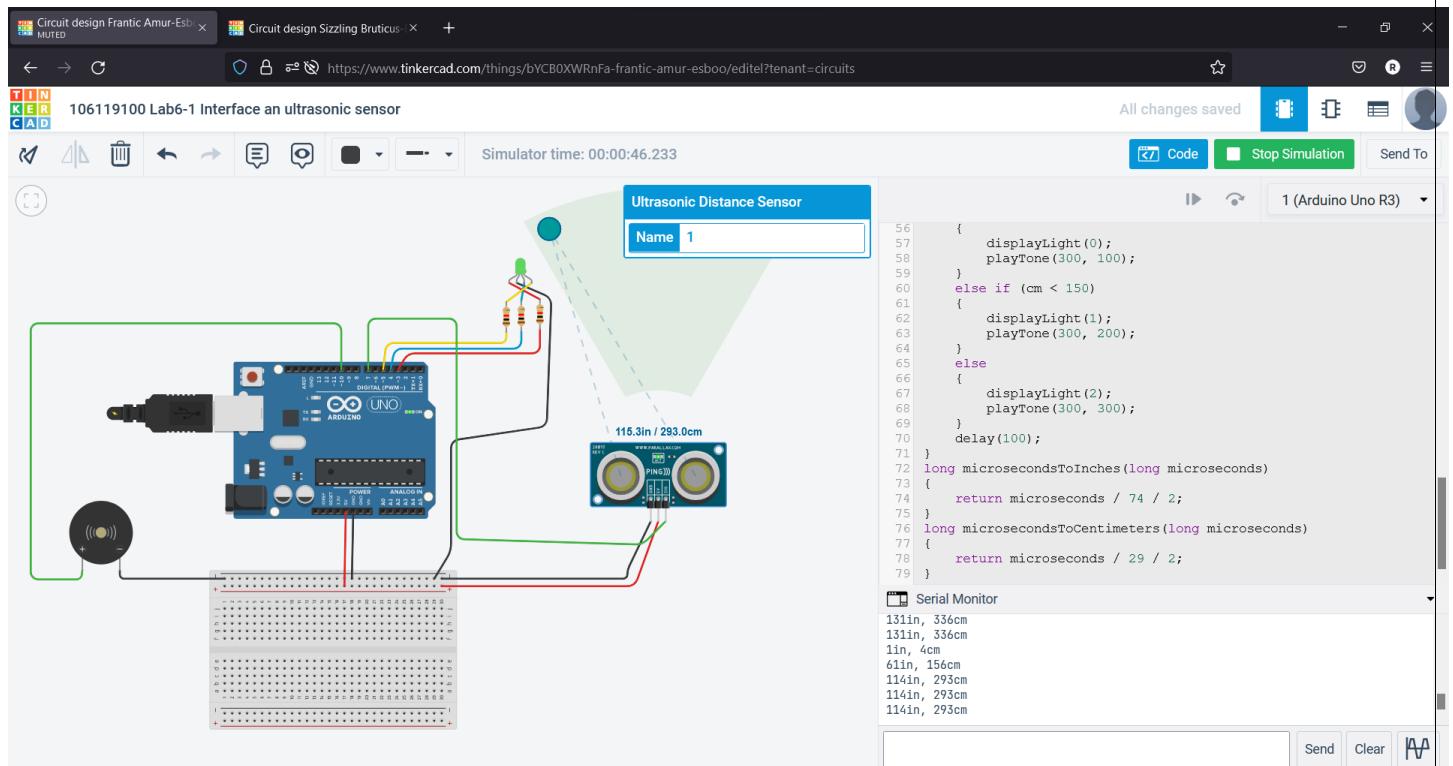
56     {
57         displayLight(0);
58         playTone(300, 100);
59     }
60     else if (cm < 150)
61     {
62         displayLight(1);
63         playTone(300, 200);
64     }
65     else
66     {
67         displayLight(2);
68         playTone(300, 300);
69     }
70     delay(100);
71 }
72 long microsecondsToInches(long microseconds)
73 {
74     return microseconds / 74 / 2;
75 }
76 long microsecondsToCentimeters(long microseconds)
77 {
78     return microseconds / 29 / 2;
79 }

```

Serial Monitor

67in, 173cm  
67in, 173cm  
67in, 173cm  
131in, 336cm  
131in, 336cm  
131in, 336cm  
131in, 336cm

Send Clear



2. Interface a PIR sensor with Arduino and blink a led if any movement is captured by the sensor and count the number of movements. Make a buzzer when movement detected.

```

// 106119100 Interface a PIR sensor with Arduino
#define F(x) x ? HIGH : LOW
const int led = 4;
const int pir = 3;
const int speaker = 10;
int totalMovements = 0;
bool isMoving = false;
void setup()
{
    Serial.begin(9600);
    pinMode(led, OUTPUT);
    pinMode(pir, INPUT);
    pinMode(speaker, OUTPUT);
}
void playTone(long duration, int freq)
{
    duration *= 1000;
}

```

```

int period = (1.0 / freq) * 100000;
long elapsed_time = 0;
while (elapsed_time < duration)
{
    digitalWrite(speaker, HIGH);
    delayMicroseconds(period / 2);
    digitalWrite(speaker, LOW);
    delayMicroseconds(period / 2);
    elapsed_time += (period);
}
}

void loop()
{
    int pirStat = digitalRead(pir);
    if (pirStat == HIGH)
    {
        if (!isMoving)
        {
            playTone(300, 300);
            totalMovements += 1;
            Serial.print("Total movements: ");
            Serial.print(totalMovements);
            Serial.print("\n");
            isMoving = true;
        }
        digitalWrite(led, HIGH);
    }
    else
    {
        isMoving = false;
        digitalWrite(led, LOW);
    }
}

```

Circuit design Copy of 106119100 Lab6 - 2 Interface a PIR sensor with Arduino

PIR Sensor  
Name 1

Simulator time: 00:00:33

```

1 #define F(x) x ? HIGH : LOW
2
3 const int led = 4;
4 const int pir = 3;
5 const int speaker = 10;
6 int totalMovements = 0;
7 bool isMoving = false;
8 void setup()
9 {
10   Serial.begin(9600);
11   pinMode(led, OUTPUT);
12   pinMode(pir, INPUT);
13   pinMode(speaker, OUTPUT);
14 }
15 void playTone(long duration, int freq)
16 {
17   duration *= 1000;
18   int period = (1.0 / freq) * 100000;
19   long elapsed_time = 0;
20   while (elapsed_time < duration)
21   {
22     Serial.print(".");
23     delayMicroseconds(period);
24     elapsed_time += period;
25   }
26 }
27
28 void loop()
29 {
30   if (isMoving)
31   {
32     playTone(1000, 1000);
33     isMoving = false;
34   }
35   if (digitalRead(pir) == F(1))
36   {
37     isMoving = true;
38     totalMovements++;
39     Serial.println("Movement detected!");
40   }
41   if (digitalRead(led) == F(1))
42   {
43     digitalWrite(led, F(0));
44   }
45   else
46   {
47     digitalWrite(led, F(1));
48   }
49   if (digitalRead(speaker) == F(1))
50   {
51     playTone(1000, 1000);
52   }
53 }

```

Serial Monitor

```

Total movements: 1
Total movements: 2
Total movements: 3
Total movements: 4

```

Send Clear

Circuit design Copy of 106119100 Lab6 - 2 Interface a PIR sensor with Arduino

PIR Sensor  
Name 1

Simulator time: 00:00:55.199

```

1 #define F(x) x ? HIGH : LOW
2
3 const int led = 4;
4 const int pir = 3;
5 const int speaker = 10;
6 int totalMovements = 0;
7 bool isMoving = false;
8 void setup()
9 {
10   Serial.begin(9600);
11   pinMode(led, OUTPUT);
12   pinMode(pir, INPUT);
13   pinMode(speaker, OUTPUT);
14 }
15 void playTone(long duration, int freq)
16 {
17   duration *= 1000;
18   int period = (1.0 / freq) * 100000;
19   long elapsed_time = 0;
20   while (elapsed_time < duration)
21   {
22     Serial.print(".");
23     delayMicroseconds(period);
24     elapsed_time += period;
25   }
26 }
27
28 void loop()
29 {
30   if (isMoving)
31   {
32     playTone(1000, 1000);
33     isMoving = false;
34   }
35   if (digitalRead(pir) == F(1))
36   {
37     isMoving = true;
38     totalMovements++;
39     Serial.println("Movement detected!");
40   }
41   if (digitalRead(led) == F(1))
42   {
43     digitalWrite(led, F(0));
44   }
45   else
46   {
47     digitalWrite(led, F(1));
48   }
49   if (digitalRead(speaker) == F(1))
50   {
51     playTone(1000, 1000);
52   }
53 }

```

Serial Monitor

```

Total movements: 1
Total movements: 2
Total movements: 3
Total movements: 4
Total movements: 5
Total movements: 6
Total movements: 7
Total movements: 1
Total movements: 2

```

Send Clear

# Lab 7

Date: 11-04-2022

## 1. Interface two 7-segment display with Arduino Uno board.

```
/* 106119100 Rajneesh */

// a,b,c,d,e,f,g
byte segValue[10][7] = {
    {0, 0, 0, 0, 0, 0, 1}, // 0
    {1, 0, 0, 1, 1, 1, 1}, // 1
    {0, 0, 1, 0, 0, 1, 0}, // 2
    {0, 0, 0, 0, 1, 1, 0}, // 3
    {1, 0, 0, 1, 1, 0, 0}, // 4
    {0, 1, 0, 0, 1, 0, 0}, // 5
    {0, 1, 0, 0, 0, 0, 0}, // 6
    {0, 0, 0, 1, 1, 1, 1}, // 7
    {0, 0, 0, 0, 0, 0, 0}, // 8
    {0, 0, 0, 0, 1, 0, 0} // 9
};

byte segPin[8] = {12, 13, 4, 6, 7, 10, 9, 3}; // {a,b,c,d,e,f,g,dp}
byte digitPin[2] = {A1, A2}; // segment

void setup()
{
    for (int i = 0; i < 10; i++)
    {
        pinMode(segPin[i], OUTPUT);
    }
    pinMode(digitPin[0], OUTPUT);
    pinMode(digitPin[1], OUTPUT);
    digitalWrite(digitPin[0], LOW);
    digitalWrite(digitPin[1], LOW);
}

void loop()
{
    for (int i = 1; i < 100; i += 2)
    {
        display_N(i);
        delay(1);
    }
}
```

```

    }
}

void display_N(int num)
{
    int und = num % 10;
    int dec = (num % 100) / 10;
    for (int i = 0; i < 100; i++)
    {
        segOutput(1, und, 1);
        segOutput(0, dec, 1);
        delay(2);
    }

    Serial.print(dec);
    Serial.println(und);
}

// LED
void segClear()
{
    for (int i = 0; i < 8; i++)
    {
        digitalWrite(segPin[i], HIGH);
    }
}
// LED
void segOutput(int d, int Number, int dp)
{
    segClear();
    digitalWrite(digitPin[d], HIGH);
    for (int i = 0; i < 7; i++)
    {
        digitalWrite(segPin[i], segValue[Number][i]);
    }
    digitalWrite(segPin[7], dp);
    delayMicroseconds(1000);
    digitalWrite(digitPin[d], LOW);
}

```

```

1  /* 106119100 Rajneesh */
2
3 // a,b,c,d,e,f,g
4
5 byte segValue[10][7] = {
6     {0, 0, 0, 0, 0, 0, 1}, // 0
7     {1, 0, 0, 1, 1, 1, 1}, // 1
8     {0, 0, 1, 0, 0, 1, 0}, // 2
9     {0, 0, 0, 0, 1, 1, 0}, // 3
10    {1, 0, 0, 1, 1, 0, 0}, // 4
11    {0, 1, 0, 0, 1, 0, 0}, // 5
12    {0, 1, 0, 0, 0, 0, 0}, // 6
13    {0, 0, 0, 1, 1, 1, 1}, // 7
14    {0, 0, 0, 0, 0, 0, 0}, // 8
15    {0, 0, 0, 0, 1, 0, 0} // 9
16 };
17
18 byte segPin[8] = {12, 13, 4, 6, 7, 10, 9, 3}; // {a,b,c,d,e,f,g,dp}
19 byte digitPin[2] = {A1, A2}; // segment
20
21 void setup()
22 {
23     for (int i = 0; i < 10; i++)
24     {
25         pinMode(segPin[i], OUTPUT);
26     }
27     pinMode(digitPin[0], OUTPUT);
28     pinMode(digitPin[1], OUTPUT);
29     digitalWrite(digitPin[0], LOW);
30     digitalWrite(digitPin[1], LOW);
31 }
32
33 void loop()
34 {
35     for (int i = 0; i < 10; i++)
36     {
37         for (int j = 0; j < 8; j++)
38         {
39             digitalWrite(segPin[j], segValue[i][j]);
40         }
41         for (int k = 0; k < 2; k++)
42         {
43             digitalWrite(digitPin[k], HIGH);
44             delay(1);
45             digitalWrite(digitPin[k], LOW);
46         }
47     }
48 }

```

2. Implement the a)odd counter and b)even counter and display the values in the seven-segment display.

```

/* 106119100 Rajneesh */

// a,b,c,d,e,f,g
byte segValue[10][7] = {
    {0, 0, 0, 0, 0, 0, 1}, // 0
    {1, 0, 0, 1, 1, 1, 1}, // 1
    {0, 0, 1, 0, 0, 1, 0}, // 2
    {0, 0, 0, 0, 1, 1, 0}, // 3
    {1, 0, 0, 1, 1, 0, 0}, // 4
    {0, 1, 0, 0, 1, 0, 0}, // 5
    {0, 1, 0, 0, 0, 0, 0}, // 6
    {0, 0, 0, 1, 1, 1, 1}, // 7
    {0, 0, 0, 0, 0, 0, 0}, // 8
    {0, 0, 0, 0, 1, 0, 0} // 9
};
byte segPin[8] = {12, 13, 4, 6, 7, 10, 9, 3}; // {a,b,c,d,e,f,g,dp}
byte digitPin[2] = {A1, A2}; // segment

void setup()
{

```

```

    for (int i = 0; i < 10; i++)
    {
        pinMode(segPin[i], OUTPUT);
    }
    pinMode(digitPin[0], OUTPUT);
    pinMode(digitPin[1], OUTPUT);
    digitalWrite(digitPin[0], LOW);
    digitalWrite(digitPin[1], LOW);
}

void loop()
{
    for (int i = 0; i < 100; i += 2)
    {
        display_N(i);
        delay(1);
    }
}

void display_N(int num)
{
    int und = num % 10;
    int dec = (num % 100) / 10;
    for (int i = 0; i < 100; i++)
    {
        segOutput(1, und, 1);
        segOutput(0, dec, 1);
        delay(2);
    }

    Serial.print(dec);
    Serial.println(und);
}

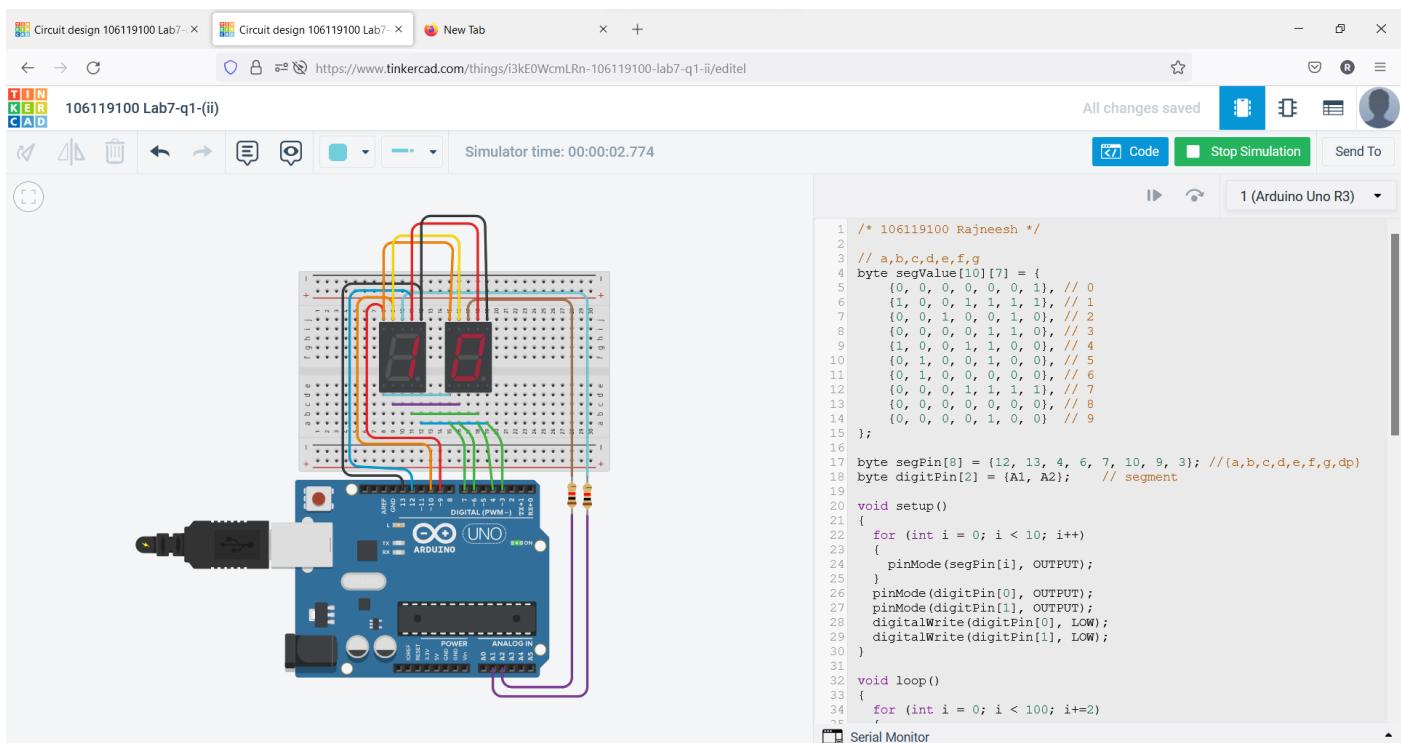
// LED
void segClear()
{
    for (int i = 0; i < 8; i++)
    {
        digitalWrite(segPin[i], HIGH);
    }
}
// LED
void segOutput(int d, int Number, int dp)

```

```

{
    segClear();
    digitalWrite(digitPin[d], HIGH);
    for (int i = 0; i < 7; i++)
    {
        digitalWrite(segPin[i], segValue[Number][i]);
    }
    digitalWrite(segPin[7], dp);
    delayMicroseconds(1000);
    digitalWrite(digitPin[d], LOW);
}

```



3. Interface the ambient light sensor with Arduino Uno board. Check the light value from the sensor and switch on/off the bulb (based on the threshold value).

`/*106119100 Rajneesh*/`

```

int light_sensor = A0;
int light_bulb = 12;
float light;
int light_value;
void setup()
{
    Serial.begin(112500);
}

```

```

pinMode(light_sensor, INPUT);
pinMode(light_bulb, OUTPUT);
}
void loop()
{
    int light_value = analogRead(light_sensor);
    light = light_value * 0.0976;
    Serial.println(light);
    delay(100);
    if (light_value > 200)
    {
        digitalWrite(light_bulb, HIGH);
    }
    else
    {
        digitalWrite(light_bulb, LOW);
    }
}

```

The screenshot shows the Tinkercad interface for a project titled "106119100 Lab7 q2". The circuit diagram on the left features an Arduino Uno at the bottom. A USB cable connects the Uno to a computer. A breadboard is positioned above the Uno, with a light bulb connected to digital pin 12 via a yellow wire. Pin 12 is also connected to ground through a green wire and to a 10k ohm resistor. The other end of the resistor is connected to an analog input pin (labeled A0) on the Uno. A red wire connects the Uno's 5V pin to the breadboard power rail, and a blue wire connects the GND pin to the breadboard ground rail.

The code in the editor is as follows:

```

1  /*106119100 Rajneesh*/
2
3  int light_sensor = A0;
4  int light_bulb = 12;
5  float light;
6  int light_value;
7  void setup()
8  {
9    Serial.begin(112500);
10   pinMode(light_sensor, INPUT);
11   pinMode(light_bulb, OUTPUT);
12 }
13 void loop()
14 {
15   int light_value = analogRead(light_sensor);
16   light = light_value * 0.0976;
17   Serial.println(light);
18   delay(100);
19   if (light_value > 200){
20     digitalWrite(light_bulb, HIGH);
21   }
22   else{
23     digitalWrite(light_bulb, LOW);
24   }
}

```

The Serial Monitor window on the right shows the following output:

```

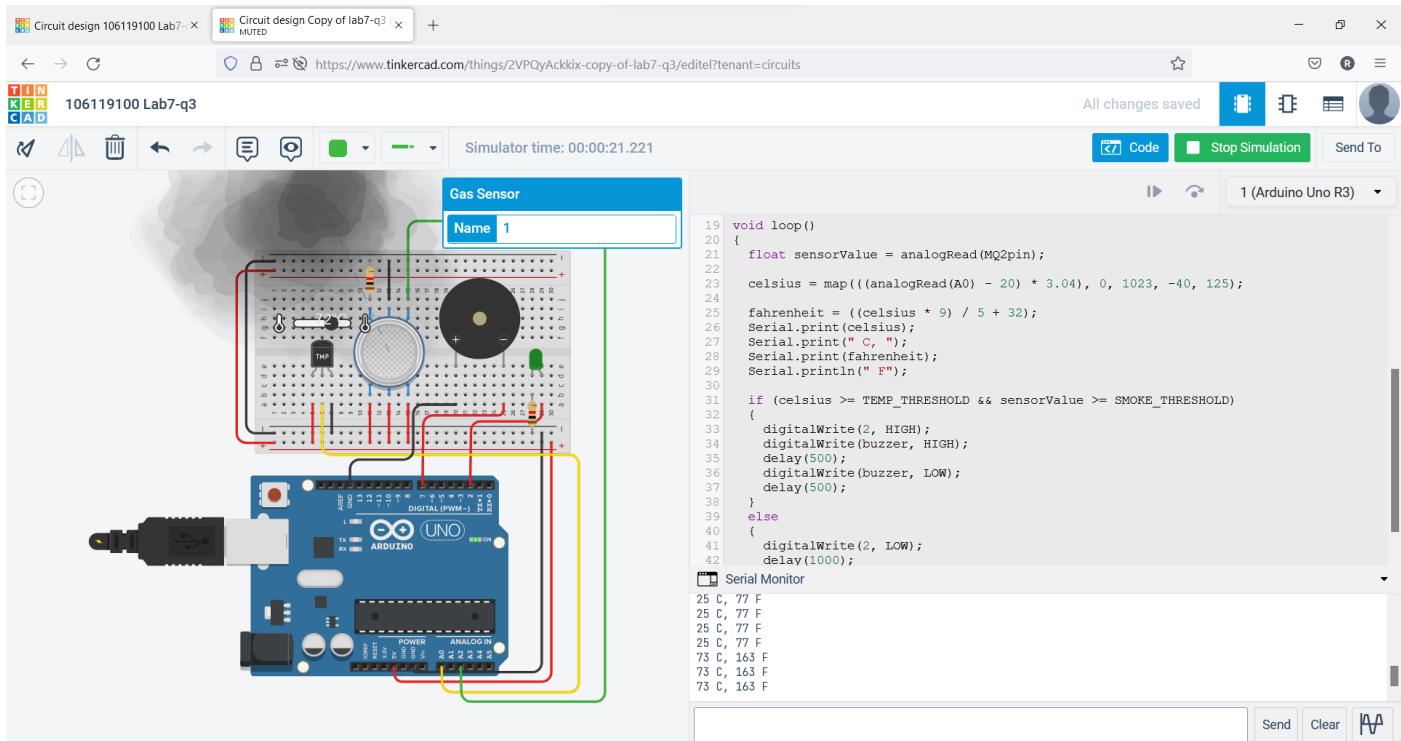
0.00
0.00
5.47
21.18
21.18
21.18
21.18

```

4. Interface the temperature and gas sensor with Arduino Uno board. Check the temperature and the gas value, if the limit is beyond the threshold, switch on the bulb and make alarm using buzzer.

```
/*106119100 Rajneesh*/\n\nint baselineTemp = 0;\nint celsius = 0;\nint fahrenheit = 0;\nint buzzer = 7;\nint MQ2pin = A2;\nconst int TEMP_THRESHOLD = 40, SMOKE_THRESHOLD = 300;\n\nvoid setup()\n{\n    pinMode(A0, INPUT);\n    Serial.begin(9600);\n\n    pinMode(2, OUTPUT);\n    pinMode(buzzer, OUTPUT);\n}\n\nvoid loop()\n{\n    float sensorValue = analogRead(MQ2pin);\n\n    celsius = map((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);\n\n    fahrenheit = ((celsius * 9) / 5 + 32);\n    Serial.print(celsius);\n    Serial.print(" C, ");\n    Serial.print(fahrenheit);\n    Serial.println(" F");\n\n    if (celsius >= TEMP_THRESHOLD && sensorValue >= SMOKE_THRESHOLD)\n    {\n        digitalWrite(2, HIGH);\n        digitalWrite(buzzer, HIGH);\n        delay(500);\n        digitalWrite(buzzer, LOW);\n        delay(500);\n    }\n}
```

```
    else
    {
        digitalWrite(2, LOW);
        delay(1000);
    }
}
```



# Lab 8

Date: 18-04-2022

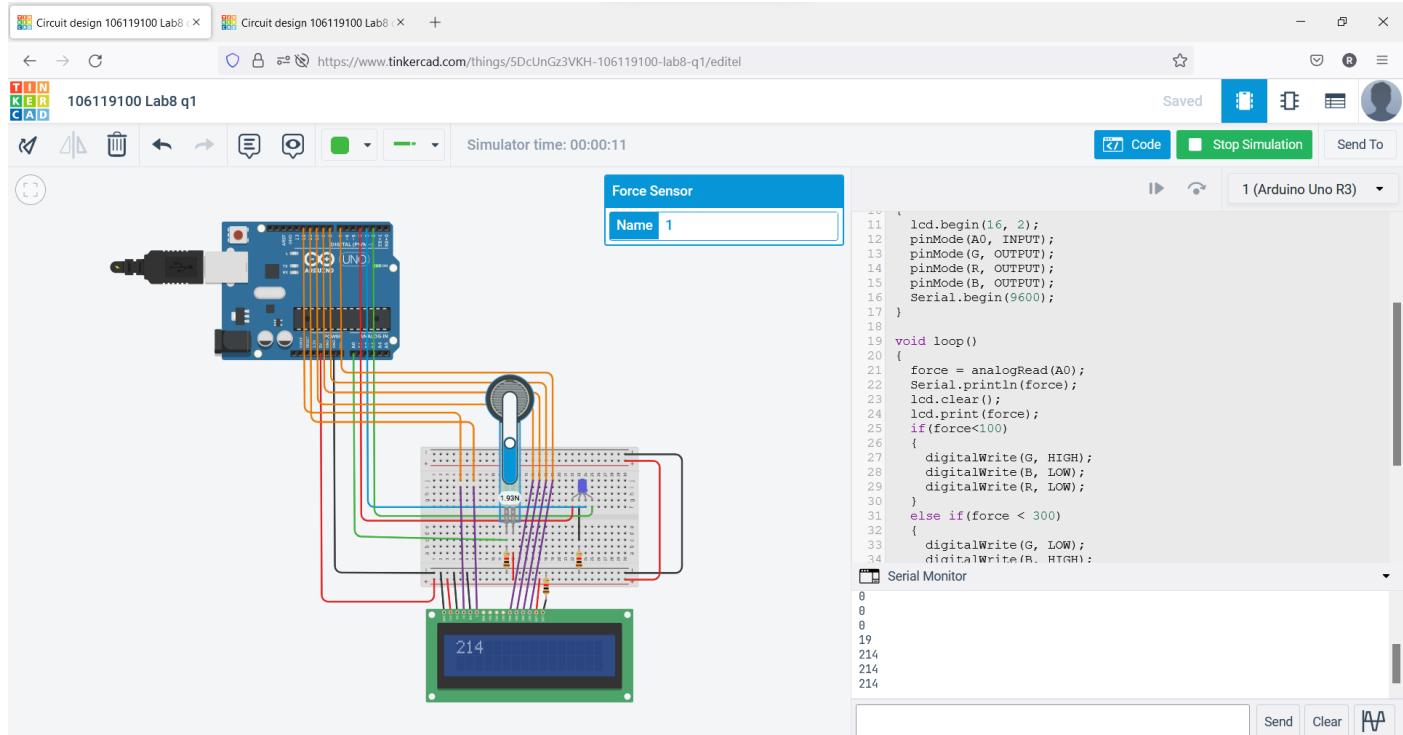
1. Interface force sensor with Arduino board and display the amount of force given as input to sensor in LCD screen. Also, interface LED RGB and change colour of the LED based on some threshold Values.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
int force = 0;
int R = 4;
int B = 3;
int G = 2;
void setup()
{
    lcd.begin(16, 2);
    pinMode(A0, INPUT);
    pinMode(G, OUTPUT);
    pinMode(R, OUTPUT);
    pinMode(B, OUTPUT);
    Serial.begin(9600);
}
void loop()
{
    force = analogRead(A0);
    Serial.println(force);
    lcd.clear();
    lcd.print(force);
    if (force < 100)
    {
        digitalWrite(G, HIGH);
        digitalWrite(B, LOW);
        digitalWrite(R, LOW);
    }
    else if (force < 300)
    {
        digitalWrite(G, LOW);
        digitalWrite(B, HIGH);
        digitalWrite(R, LOW);
    }
    else
```

```

    {
        digitalWrite(R, HIGH);
        digitalWrite(G, LOW);
        digitalWrite(B, LOW);
    }
    delay(1000);
}

```



2. Interface the keypad and tilt sensor with Arduino board, if the sensor is being tilted, then take input from keypad and print it in the LCD

```

#include <LiquidCrystal.h>
#include <Keypad.h>

const byte numRows = 4; // number of rows on the keypad
const byte numCols = 4; // number of columns on the keypad

// keymap defines the key pressed according to the row and columns just as
// appears on the keypad
char keymap[numRows][numCols] =
{
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}};

```

```

// Code that shows the keypad connections to the arduino terminals
byte rowPins[numRows] = {10, 9, 8, 7}; // Rows 0 to 3
byte colPins[numCols] = {A0, A1, A2, A3}; // Columns 0 to 3

// initializes an instance of the Keypad class
Keypad myKeypad = Keypad(makeKeymap(keymap), rowPins, colPins, numRows,
numCols);

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int tilt = A4;

char buf[16] = {0};
int cnt = 0;

void setup()
{
    Serial.begin(9600);
    lcd.begin(16, 2);
    pinMode(tilt, INPUT);
}

void loop()
{
    int reading = analogRead(tilt);

    char keypressed = myKeypad.getKey();

    Serial.println(reading);

    if (reading > 100)
    {

        buf[cnt] = keypressed;
        if (keypressed != 0)
            cnt++;
        cnt %= 16;
        buf[cnt] = 0;
        lcd.clear();
        lcd.print(buf);
    }
    else
    {
        lcd.clear();
        lcd.print("IDLE");
    }

    delay(50);
}

```

The screenshot shows a Tinkercad project titled "Circuit design 106119100 Lab8" with a URL of <https://www.tinkercad.com/things/8yT2Vd8ZaEb-106119100-lab8-q2/editel>. The project includes a keypad component connected to an Arduino Uno and a LiquidCrystal I2C display. The code editor contains the following C++ code for an Arduino Uno R3:

```
1 #include <LiquidCrystal.h>
2 #include <Keypad.h>
3
4 const byte numRows= 4; //number of rows on the keypad
5 const byte numCols= 4; //number of columns on the keypad
6
7 //keymap defines the key pressed according to the row and columns
8 char keymap[numRows][numCols]=
9 {
10 {
11 {'1', '2', '3', 'A'},
12 {'4', '5', '6', 'B'},
13 {'7', '8', '9', 'C'},
14 {'*', '0', '#', 'D'}
15 };
16
17 //Code that shows the keypad connections to the arduino terminal
18 byte rowPins[numRows] = {10,9,8,7}; //Rows 0 to 3
19 byte colPins[numCols] = {A0,A1,A2,A3}; //Columns 0 to 3
20
21 //Initializes an instance of the Keypad class
22 Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, num
23
24 }
```

The Serial Monitor window shows the following output:

```
1023
1023
1023
1023
1023
1023
1023
```

The diagram illustrates a circuit setup for a numeric keypad and a tilt sensor connected to an Arduino Uno. The numeric keypad is represented by a 4x3 grid of buttons. The tilt sensor is shown as a blue cylindrical component. The Arduino Uno is the central microcontroller. A breadboard is used to connect the keypad, tilt sensor, and LCD display. The LCD screen shows the word "IDLE". A blue header box labeled "Tilt Sensor" contains a text input field with the value "1".