CSLR61

Embedded Systems Architecture

Laboratory Record

Anurag Goyal

106119014

CSE - B

Contents

1 Lab 1 - 31/01/2022 **1**

1.1 1

1.2 2

1.3 3

1.4 4

2 Lab 2 - 07/02/2022 **5**

2.1 5

2.2 7

2.3 8

3 Lab 3 - 14/02/2022 **10**

3.1 10

3.2 12

3.3 13

4 Lab 4 - 28/02/2022 **15**

4.1 15

4.2 17

4.3 18

4.4 19

5 Lab 5 - 07/03/2022 **20**

5.1 20

5.2 21

5.3 22

5.4 23

6 Lab 6 - 28/03/2022 **24**

6.1 24

6.2 27

7 Lab 7 - 11/04/2022 **29**

7.1 29

7.2 33

7.3 35

8 Lab 8 - 18/04/2022 **37**

8.1 37

8.2 39

1 Lab 1 - 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

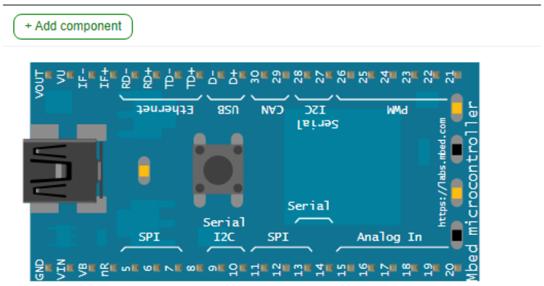
1.1

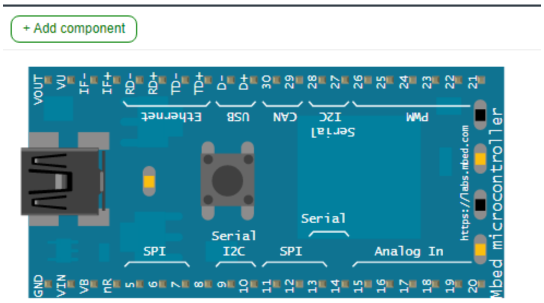
Aim Blink LEDs in alternate order – 1 and 3 together and 2 and 4 together.

Code

```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
int main()
{
    while (1)
    {
        // 1010
        // 0101
        int arr[] = {10, 5};
        for (int i = 0; i < sizeof(arr) / sizeof(int); i++)
        {
            myleds = arr[i];
            wait(1);
        }
    }
}
```

Circuit Design / Output


+ Add component
VOUT VU IF- IF+ RD- RD+ TD- TD+ D- D+ 30 29 28 27 26 25 24 23 22 21
Ethernet USB CAN I2C PWM
Serial
Serial
SPI Serial I2C SPI Analog In
GND VIN VB nR 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
https://labs.mbed.com
mbed microcontroller


+ Add component
VOUT VU IF- IF+ RD- RD+ TD- TD+ D- D+ 30 29 28 27 26 25 24 23 22 21
Ethernet USB CAN I2C PWM
Serial
Serial
SPI Serial I2C SPI Analog In
GND VIN VB nR 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
https://labs.mbed.com
mbed microcontroller

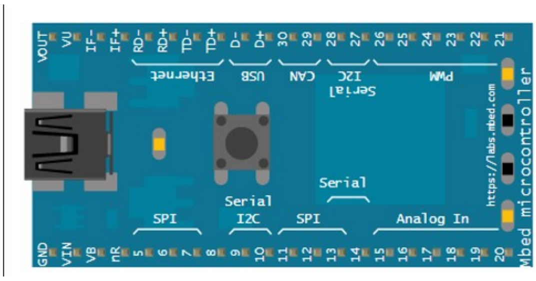
1.2

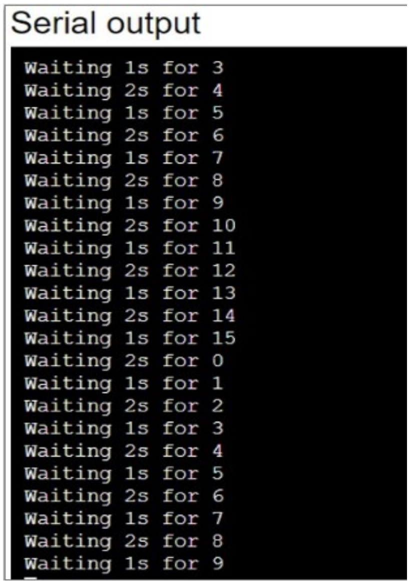
Aim Blink LEDs – count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec.

Code

```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
int main()
{
    while (1)
    {
        for (int i = 0; i < 16; i++)
        {
            myleds = i;
            if (i % 2 == 1)
            {
                printf("Waiting 1s for %d\n", i);
                wait(1);
            }
            else
            {
                printf("Waiting 2s for %d\n", i);
                wait(2);
            }
        }
    }
}
```

Circuit Design / Output





Serial output

```
Waiting 1s for 3
Waiting 2s for 4
Waiting 1s for 5
Waiting 2s for 6
Waiting 1s for 7
Waiting 2s for 8
Waiting 1s for 9
Waiting 2s for 10
Waiting 1s for 11
Waiting 2s for 12
Waiting 1s for 13
Waiting 2s for 14
Waiting 1s for 15
Waiting 2s for 0
Waiting 1s for 1
Waiting 2s for 2
Waiting 1s for 3
Waiting 2s for 4
Waiting 1s for 5
Waiting 2s for 6
Waiting 1s for 7
Waiting 2s for 8
Waiting 1s for 9
```

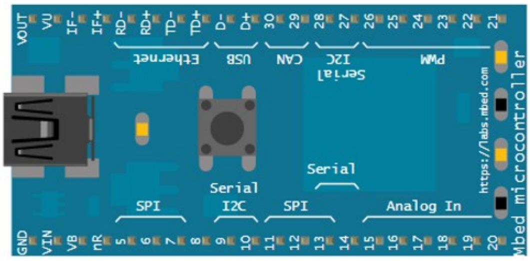
1.3

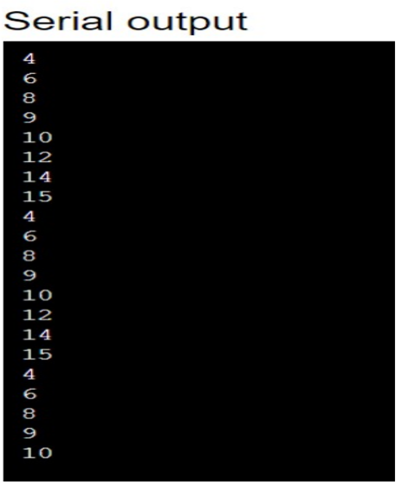
Aim Blink LEDs – for all composite number below 15.

Code

```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
bool isComposite(int n)
{
    if (n == 1)
        return false;
    for (int i = 2; i * 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))
            {
                printf("%d\n", i);
                myleds = i;
                wait(1);
            }
        }
    }
}
```

Circuit Design / Output




Serial output

```
4
6
8
9
10
12
14
15
4
6
8
9
10
12
14
15
4
6
8
9
10
```

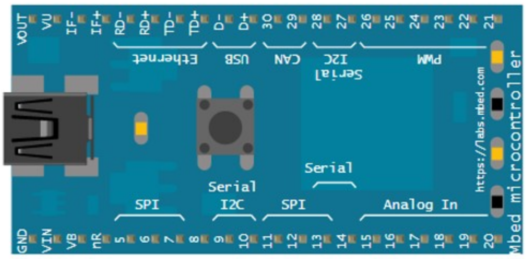
1.4

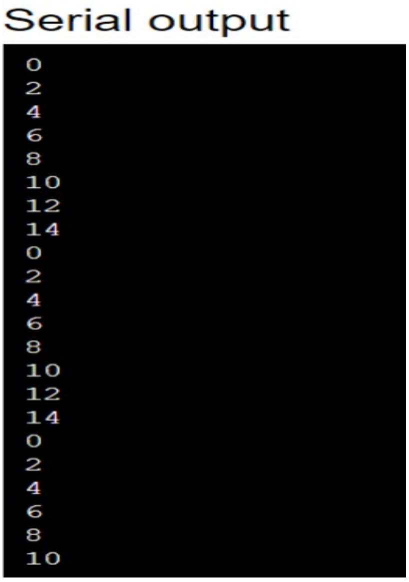
Aim Blink LEDs – to count even numbers.

Code

```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
int main()
{
    while (1)
    {
        for (int i = 0; i < 16; i += 2)
        {
            printf("%d\n", i);
            myleds = i;
            wait(0.5);
        }
    }
}
```

Circuit Design / Output





Serial output

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

2 Lab 2 - 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. Give switch option to manually toggle between two counters.

Libraries Used: InterruptIn, TimeOut, BusOut, Ticker

2.1

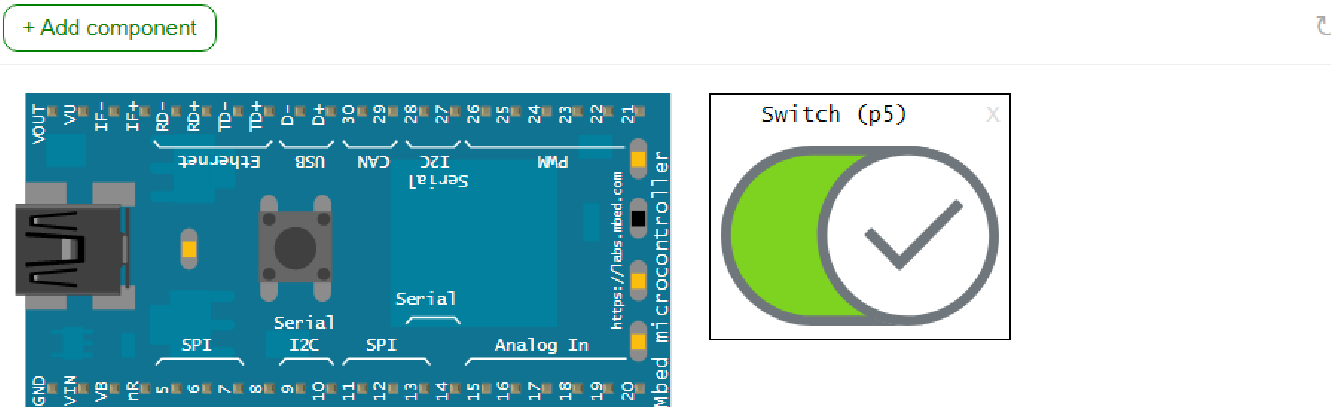
Aim Switch between hex counter and decade counter using switches. Display the current value of the counter with the help of 4 LEDs and the screen.

Code

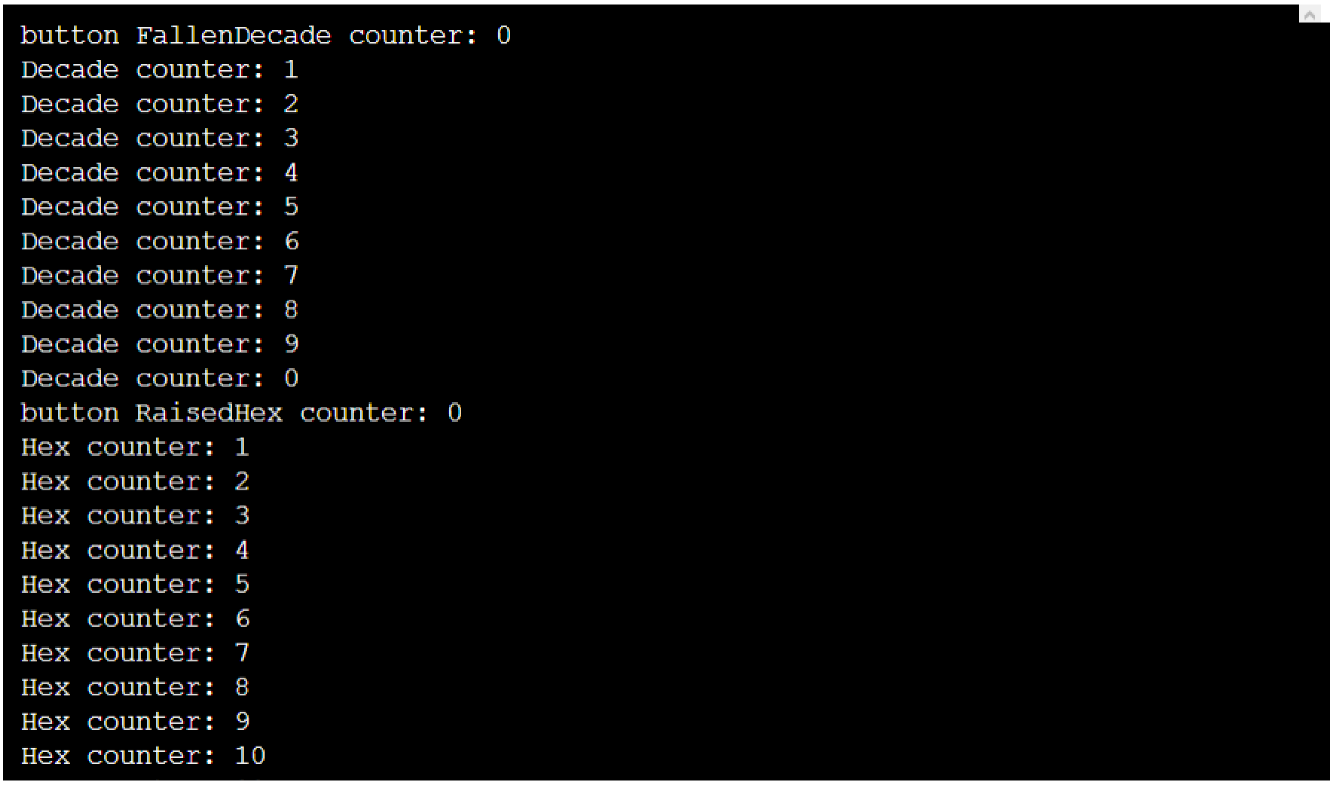
```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
InterruptIn button(p5);
void hex_counter()
{
    printf("button Raised");
    while (1)
    {
        for (int i = 0; i < 16; i++)
        {
            myleds = i;
            printf("Hex counter: %d\n", i);
            wait(0.5);
        }
    }
}
void dec_counter()
{
    printf("button Fallen");
    while (1)
    {
        for (int i = 0; i < 10; i++)
        {
            myleds = i;
            printf("Decade counter: %d\n", i);
            wait(0.5);
        }
    }
}
int main()
```

```
{
    button.rise(&hex_counter);
    button.fall(&dec_counter);
}
```

Circuit Design / Output



Serial output



```
button FallenDecade counter: 0
Decade counter: 1
Decade counter: 2
Decade counter: 3
Decade counter: 4
Decade counter: 5
Decade counter: 6
Decade counter: 7
Decade counter: 8
Decade counter: 9
Decade counter: 0
button RaisedHex counter: 0
Hex counter: 1
Hex counter: 2
Hex counter: 3
Hex counter: 4
Hex counter: 5
Hex counter: 6
Hex counter: 7
Hex counter: 8
Hex counter: 9
Hex counter: 10
```

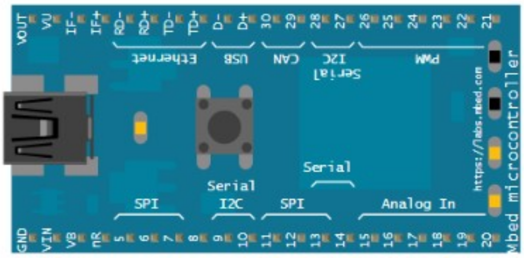
2.2

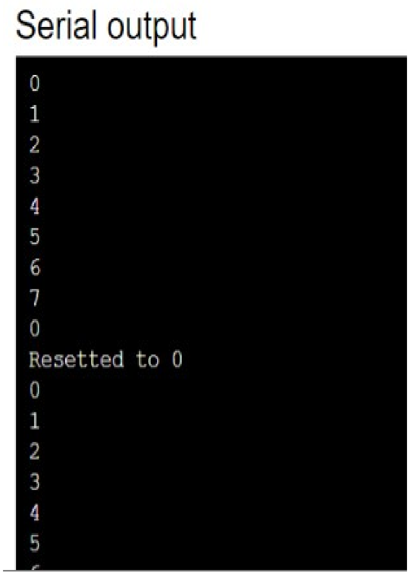
Aim Display the octal counter using the onboard LEDs using Ticker Object; After 10 second, reset all the values of the onboard LED using Timeout Object.

Code

```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
Ticker tk;
Timeout to;
int curr = 0;
void oct_counter()
{
    myleds = curr;
    printf("%d \n", curr);
    curr += 1;
    curr %= 8;
}
void reset()
{
    curr = 0;
    printf("Resetted to 0\n");
}
int main()
{
    tk.attach(&oct_counter, 1.0);
    to.attach(&reset, 10.0);
}
```

Circuit Design / Output





Serial output

```
0
1
2
3
4
5
6
7
0
Resetted to 0
0
1
2
3
4
5
```

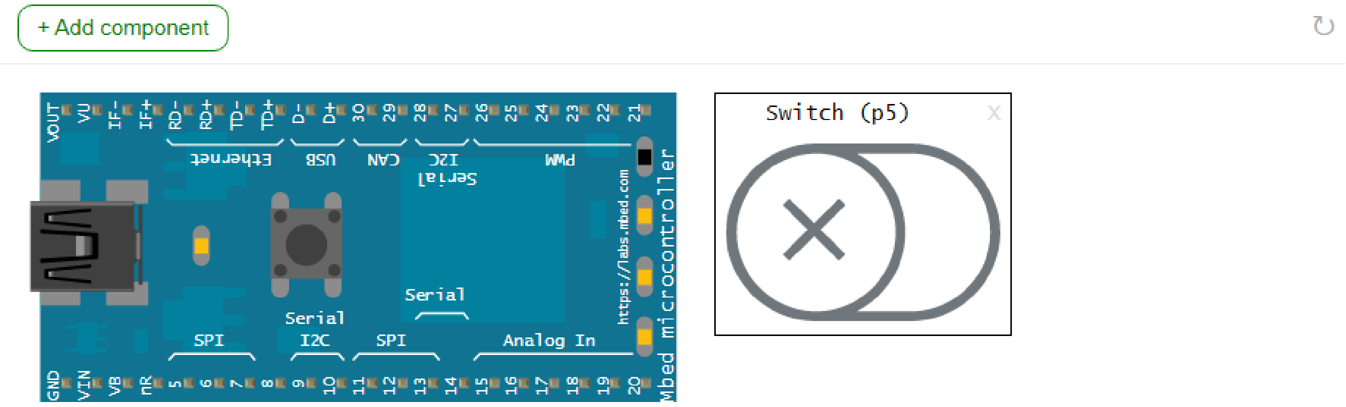
2.3

Aim 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. Give switch option to manually toggle between two counters.

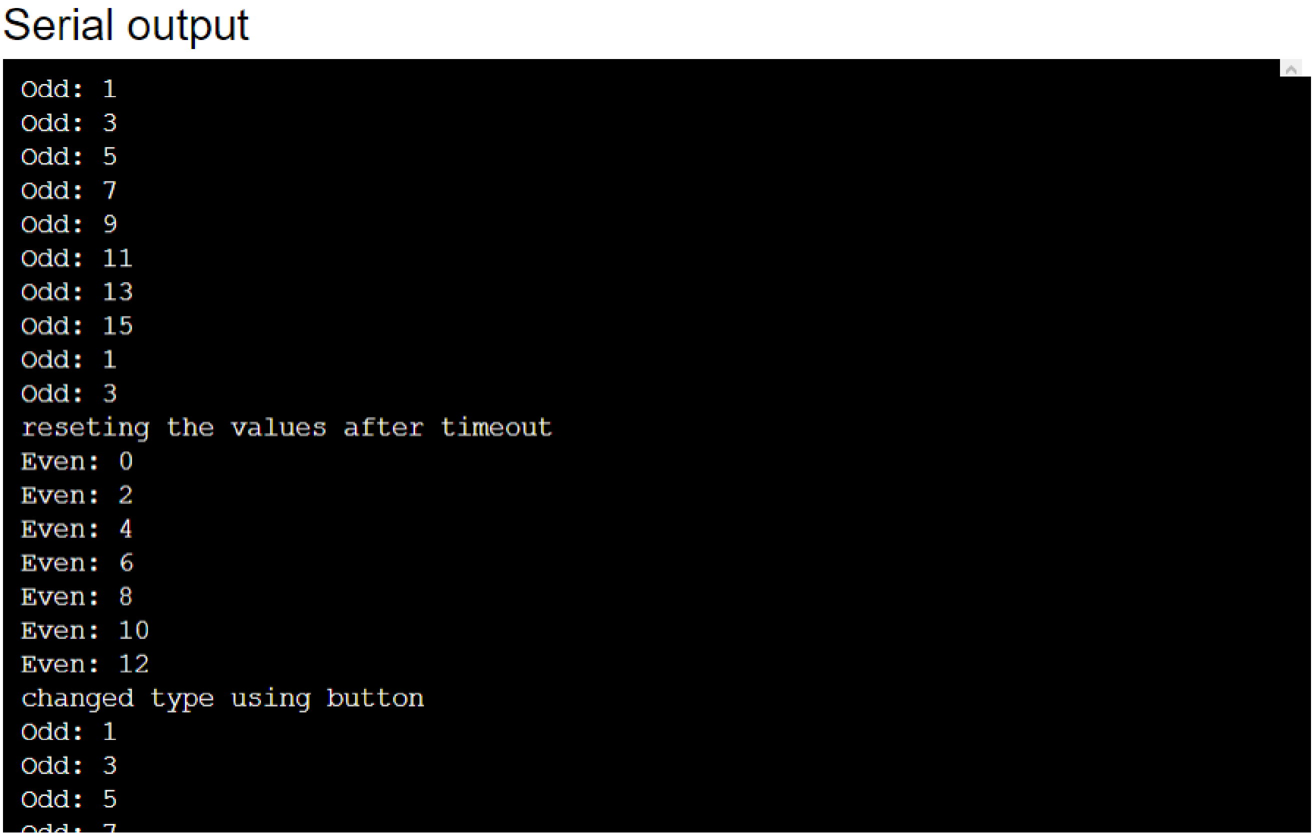
Code

```
#include "mbed.h"
BusOut myleds(LED1, LED2, LED3, LED4);
InterruptIn btn(p5);
Ticker tk;
Timeout to;
int odd = 1;
const char *cntr_type = "Odd";
int val = 1;
void counter()
{
    myleds = val;
    printf("%s: %d \n", cntr_type, val);
    val += 2;
    val %= 16;
}
void change_type()
{
    printf("changed type using button \n");
    if (odd)
    {
        odd = 0;
        val = 0;
        cntr_type = "Even";
    }
    else
    {
        odd = 1;
        val = 1;
        cntr_type = "Odd";
    }
}
void reset()
{
    printf("reseting the values after timeout \n");
    odd = 0;
    val = 0;
    cntr_type = "Even";
}
int main()
{
    tk.attach(&counter, 1.0);
    btn.rise(&change_type);
    btn.fall(&change_type);
    to.attach(&reset, 10.0);
}
```

Circuit Design / Output



Serial output



```
Odd: 1
Odd: 3
Odd: 5
Odd: 7
Odd: 9
Odd: 11
Odd: 13
Odd: 15
Odd: 1
Odd: 3
reseting the values after timeout
Even: 0
Even: 2
Even: 4
Even: 6
Even: 8
Even: 10
Even: 12
changed type using button
Odd: 1
Odd: 3
Odd: 5
Odd: 7
```

3 Lab 3 - 14/02/2022

1. Increase the intensity of one LED while descreasing 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.

Libraries Used: pwmout, analogin

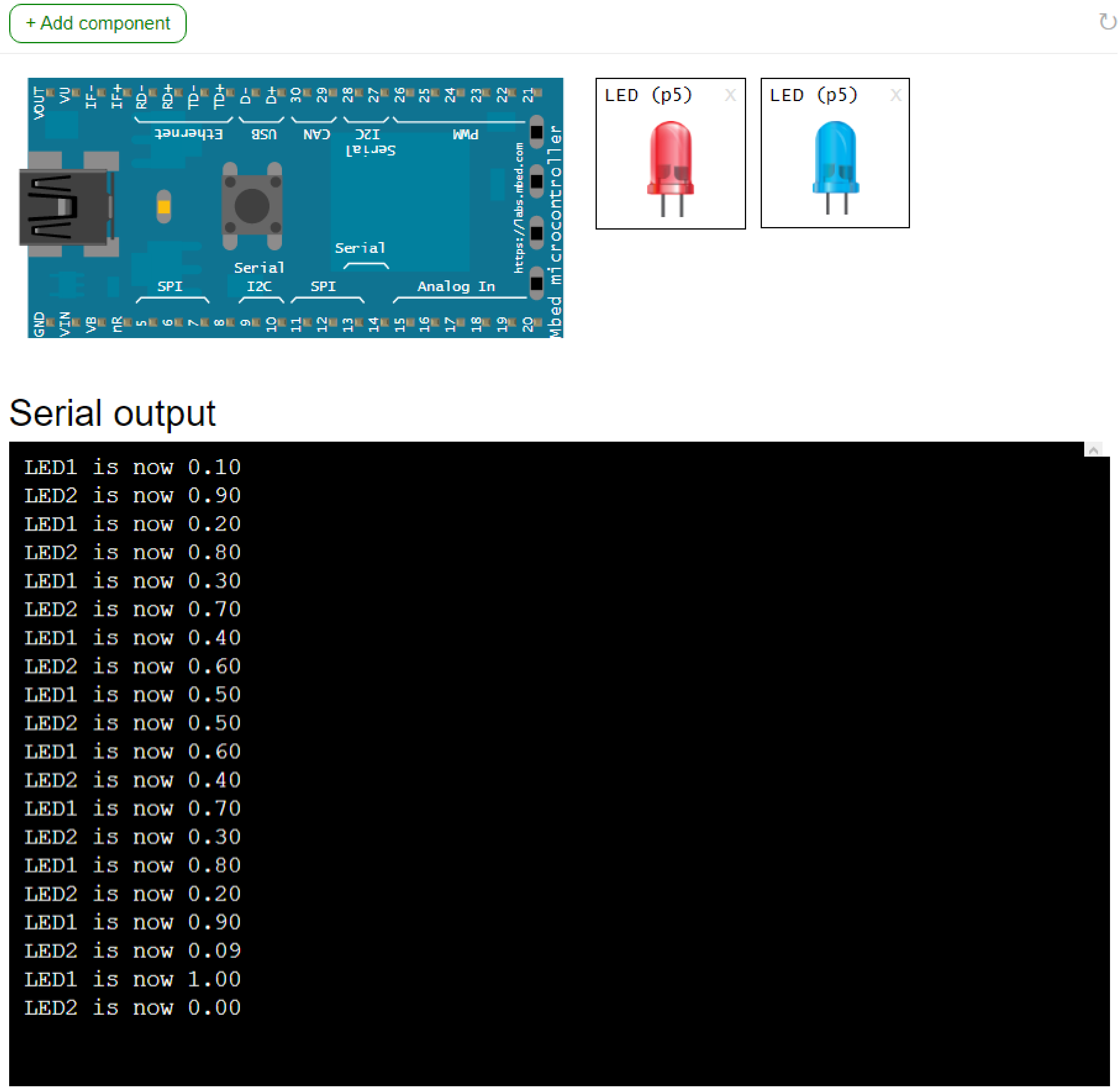
3.1

Aim Increase the intensity of one LED while descreasing the other LED's intensity at the same unit. Display the intensity value of the LEDs.

Code

```
#include "mbed.h"
PwmOut led1(p5);
PwmOut led2(p6);
int main()
{
    led2 = 1;
    led1 = 0;
    while (1)
    {
        led1 = led1 + 0.10;
        led2 = led2 - 0.10;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1.0)
        {
            led1 = 0;
        }
        if (led2 == 0)
        {
            led2 = 1;
        }
    }
}
```

Circuit Design / Output



Serial output

```
LED1 is now 0.10
LED2 is now 0.90
LED1 is now 0.20
LED2 is now 0.80
LED1 is now 0.30
LED2 is now 0.70
LED1 is now 0.40
LED2 is now 0.60
LED1 is now 0.50
LED2 is now 0.50
LED1 is now 0.60
LED2 is now 0.40
LED1 is now 0.70
LED2 is now 0.30
LED1 is now 0.80
LED2 is now 0.20
LED1 is now 0.90
LED2 is now 0.09
LED1 is now 1.00
LED2 is now 0.00
```

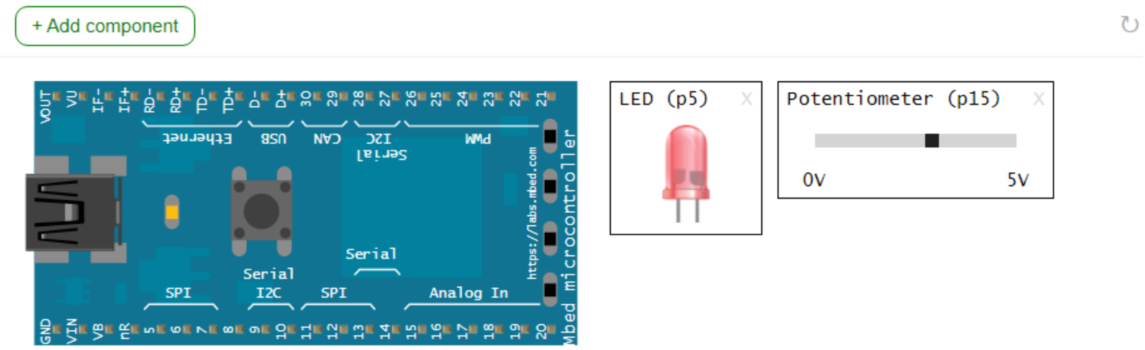
3.2

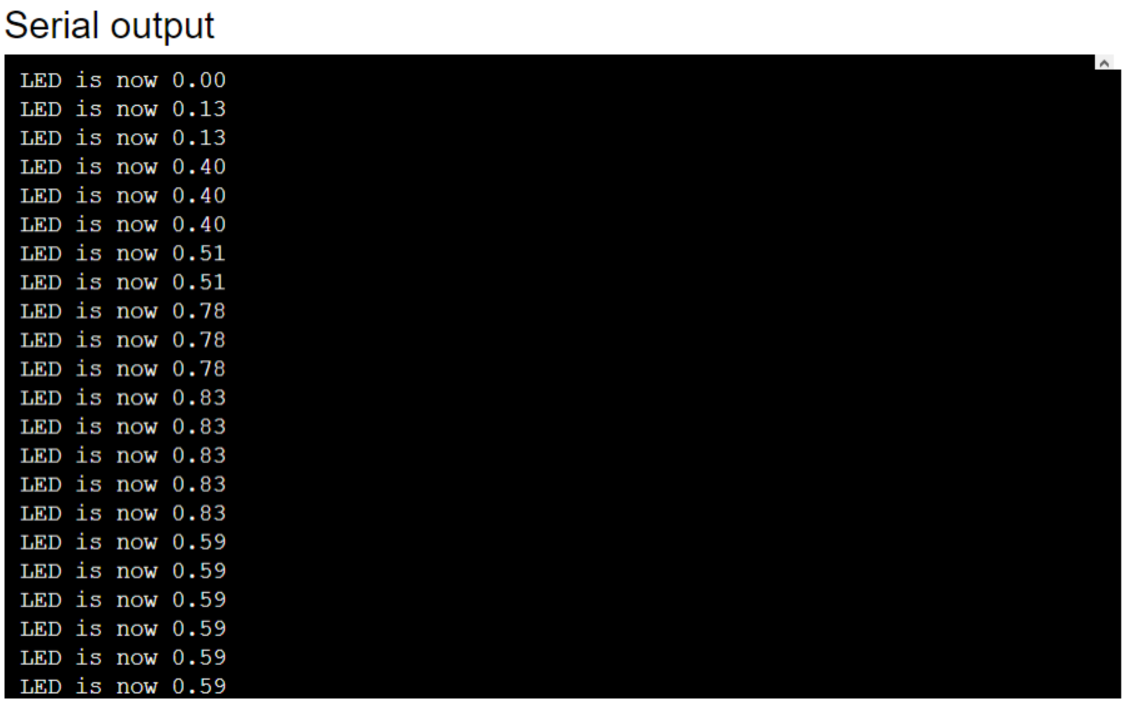
Aim Interface potentiometer with mbed board and based on the value of potentiometer, adjust the intensity of an external LED.

Code

```
#include "mbed.h"
PwmOut led(p5);
AnalogIn p(p15);
int main()
{
    while (1)
    {
        led = p.read();
        printf("LED is now %.2f\n", led.read());
        wait(0.5);
    }
}
```

Circuit Design / Output




Serial output

```
LED is now 0.00
LED is now 0.13
LED is now 0.13
LED is now 0.40
LED is now 0.40
LED is now 0.40
LED is now 0.51
LED is now 0.51
LED is now 0.78
LED is now 0.78
LED is now 0.78
LED is now 0.83
LED is now 0.83
LED is now 0.83
LED is now 0.83
LED is now 0.83
LED is now 0.59
LED is now 0.59
LED is now 0.59
LED is now 0.59
LED is now 0.59
LED is now 0.59
```

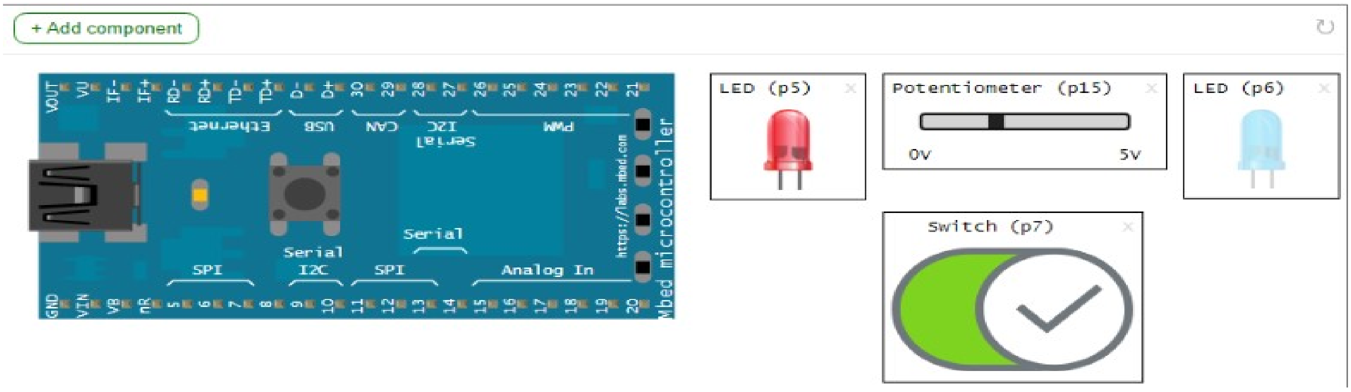
3.3

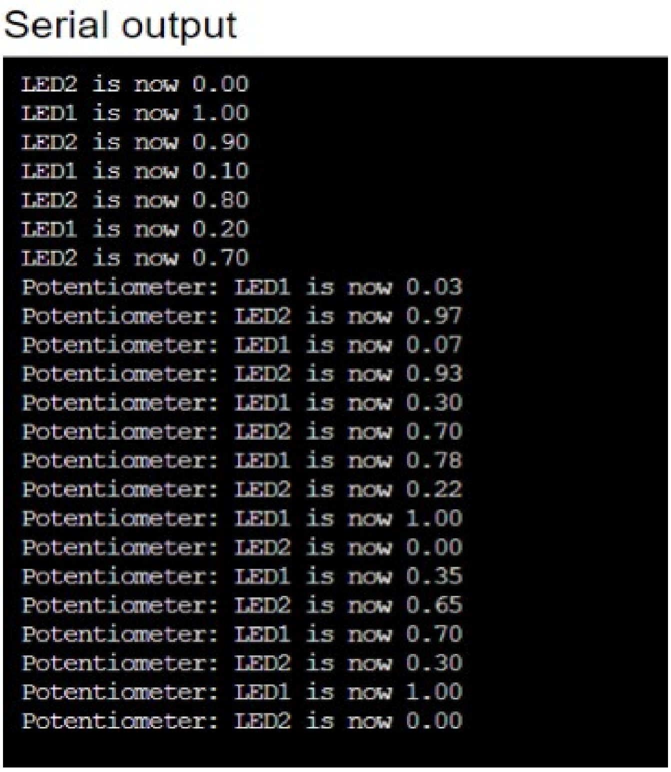
Aim Interface switch, 2 LEDs, potentiometer with the mbed board. If the switch is on, control the intensity of the LEDs using potentiometer (one LED should be increasing and other should be decreasing at the rate given via potentiometer). If the switch is off, blink LEDs one at increasing rate and other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

Code

```
#include "mbed.h"
PwmOut led1(p5);
PwmOut led2(p6);
InterruptIn btn(p7);
AnalogIn p(p15);
#define MAX(x, y) x > y ? x : y
#define MIN(x, y) x > y ? y : x
void inc_dec_by_potentio()
{
    led1 = 0;
    led2 = 1;
    while (1)
    {
        float rate = p.read();
        led1 = MIN(led1 + rate, 1);
        led2 = MAX(led2 - rate, 0);
        printf("Potentiometer: LED1 is now %.2f\n", led1.read());
        printf("Potentiometer: LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1)
        {
            led1 = 0;
        }
        if (led2 == 0)
        {
            led2 = 1;
        }
    }
}
void inc_dec()
{
    led1 = 0;
    led2 = 1;
    while (1)
    {
        led1 = led1 + 0.10;
        led2 = led2 - 0.10;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1.0)
        {
            led1 = 0;
        }
        if (led2 == 0)
        {
            led2 = 1;
        }
    }
}
int main()
{
    btn.rise(inc_dec_by_potentio);
    btn.fall(inc_dec);
}
```

Circuit Design / Output





Serial output

```
LED2 is now 0.00
LED1 is now 1.00
LED2 is now 0.90
LED1 is now 0.10
LED2 is now 0.80
LED1 is now 0.20
LED2 is now 0.70
Potentiometer: LED1 is now 0.03
Potentiometer: LED2 is now 0.97
Potentiometer: LED1 is now 0.07
Potentiometer: LED2 is now 0.93
Potentiometer: LED1 is now 0.30
Potentiometer: LED2 is now 0.70
Potentiometer: LED1 is now 0.78
Potentiometer: LED2 is now 0.22
Potentiometer: LED1 is now 1.00
Potentiometer: LED2 is now 0.00
Potentiometer: LED1 is now 0.35
Potentiometer: LED2 is now 0.65
Potentiometer: LED1 is now 0.70
Potentiometer: LED2 is now 0.30
Potentiometer: LED1 is now 1.00
Potentiometer: LED2 is now 0.00
```

4 Lab 4 - 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.

Libraries Used: pwmout, analogin

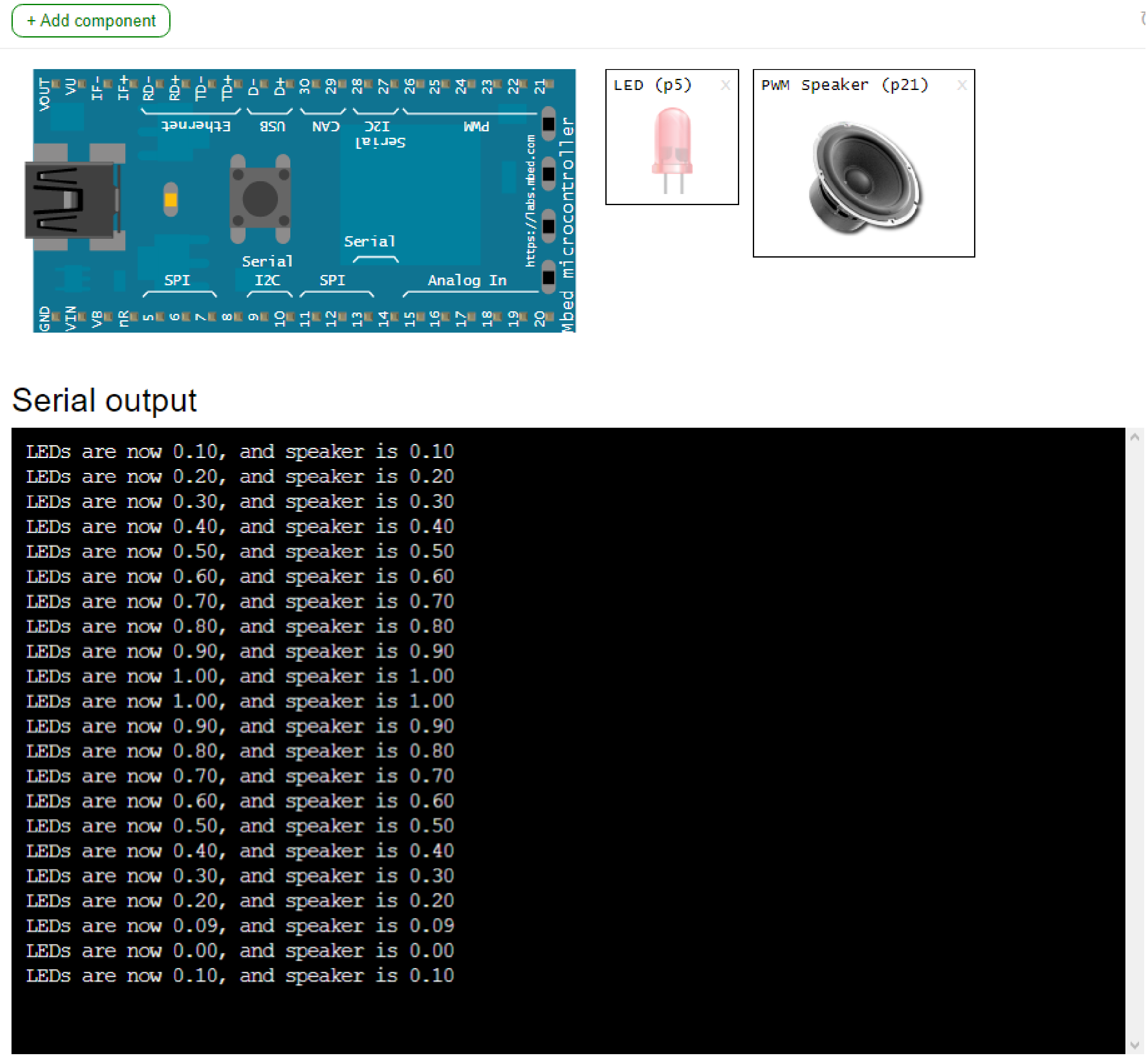
4.1

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

Code

```
#include "mbed.h"
PwmOut led(p5);
PwmOut speaker(p21);
int main()
{
    int inc = 1;
    while (1)
    {
        led = led + inc * 0.10;
        speaker = speaker + inc * 0.10;
        printf("LEDs are now %.2f, and speaker is %.2f\n", led.read(),
               speaker.read());
        wait(0.5);
        if (led == 1.0)
        {
            inc = -1;
        }
        if (led == 0)
        {
            inc = 1;
        }
    }
}
```

Circuit Design / Output


+ Add component
LED (p5)
PWM Speaker (p21)
VOUT VU IF- IF+ RD- RD+ TD- TD+ D- D+ 30 29 28 27 26 25 24 23 22 21
Ethernet USB CAN I2C Serial PWM
Serial
Serial I2C
SPI SPI Analog In
https://labs.mbed.com
mbed microcontroller
GND VIN VB nR 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Serial output

```
LEDs are now 0.10, and speaker is 0.10
LEDs are now 0.20, and speaker is 0.20
LEDs are now 0.30, and speaker is 0.30
LEDs are now 0.40, and speaker is 0.40
LEDs are now 0.50, and speaker is 0.50
LEDs are now 0.60, and speaker is 0.60
LEDs are now 0.70, and speaker is 0.70
LEDs are now 0.80, and speaker is 0.80
LEDs are now 0.90, and speaker is 0.90
LEDs are now 1.00, and speaker is 1.00
LEDs are now 1.00, and speaker is 1.00
LEDs are now 0.90, and speaker is 0.90
LEDs are now 0.80, and speaker is 0.80
LEDs are now 0.70, and speaker is 0.70
LEDs are now 0.60, and speaker is 0.60
LEDs are now 0.50, and speaker is 0.50
LEDs are now 0.40, and speaker is 0.40
LEDs are now 0.30, and speaker is 0.30
LEDs are now 0.20, and speaker is 0.20
LEDs are now 0.09, and speaker is 0.09
LEDs are now 0.00, and speaker is 0.00
LEDs are now 0.10, and speaker is 0.10
```

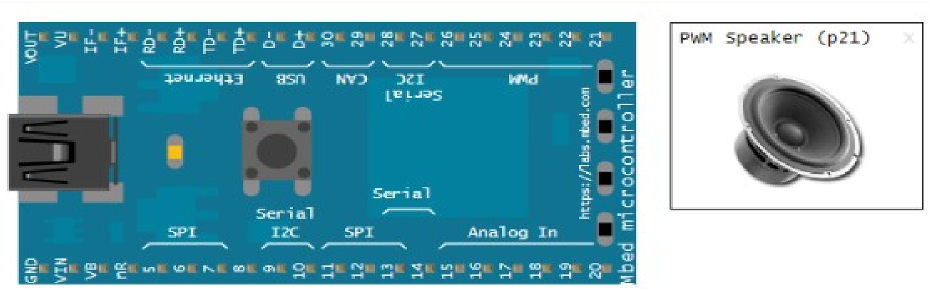
4.2

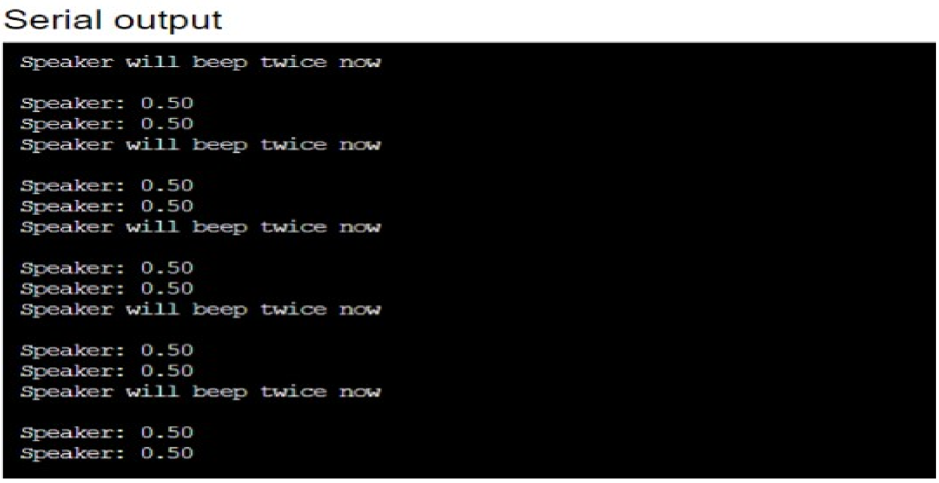
Aim Design an alarm system, which makes double beep sound for every 5s using a speaker.

Code

```
#include "mbed.h"
PwmOut speaker(p21);
int main()
{
    while (1)
    {
        // Can use ticker as well
        wait_ms(5000);
        printf("Speaker will beep twice now \n\n");
        speaker = 0.5;
        wait_ms(50);
        printf("Speaker: %.2f\n", speaker.read());
        speaker = 0;
        wait_ms(50);
        speaker = 0.5;
        wait_ms(50);
        printf("Speaker: %.2f\n", speaker.read());
        speaker = 0;
    }
}
```

Circuit Design / Output




Serial output

```
Speaker will beep twice now

Speaker: 0.50
Speaker: 0.50
Speaker will beep twice now

Speaker: 0.50
Speaker: 0.50
Speaker will beep twice now

Speaker: 0.50
Speaker: 0.50
Speaker will beep twice now

Speaker: 0.50
Speaker: 0.50
Speaker will beep twice now

Speaker: 0.50
Speaker: 0.50
```

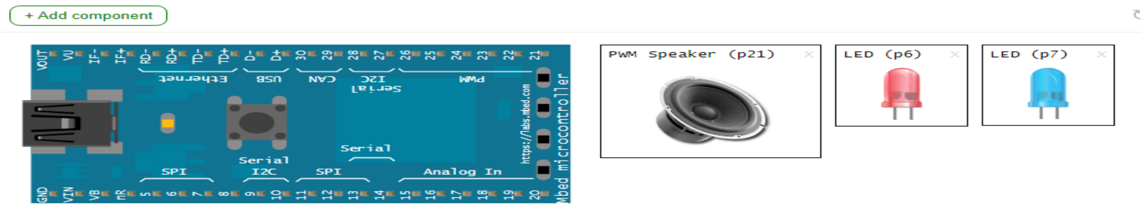
4.3

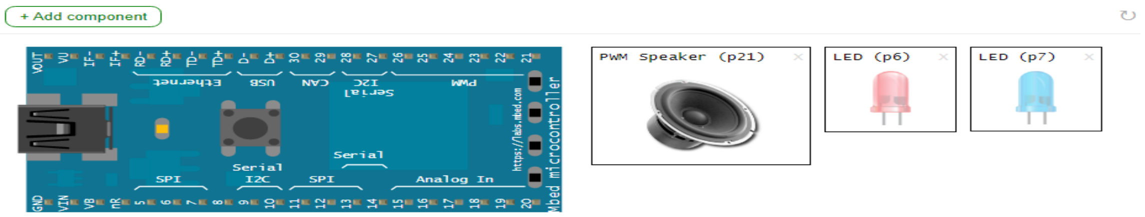
Aim Have a switch, 2 LEDs and a speaker interfaced with mbed board. Upon switching on, the intensity of the sound should be increasing and the upon switching off, should make the intensity of the sound decreasing. Proportionally increase or decrease the intensity of two LEDs.

Code

```
#include "mbed.h"
PwmOut speaker(p21);
PwmOut led1(p6);
PwmOut led2(p7);
float delta = -0.1f;
InterruptIn btn(p5);
void pos() { delta = 0.1f; }
void neg() { delta = -0.1f; }
int main()
{
    btn.rise(&pos);
    btn.fall(&neg);
    speaker = 0;
    while (1)
    {
        if (delta > 0 && speaker >= 1)
            speaker = 0;
        if (delta < 0 && speaker <= 0)
            speaker = 1;
        speaker = speaker + delta;
        led1 = speaker;
        led2 = speaker;
        wait(0.5);
    }
}
```

Circuit Design / Output





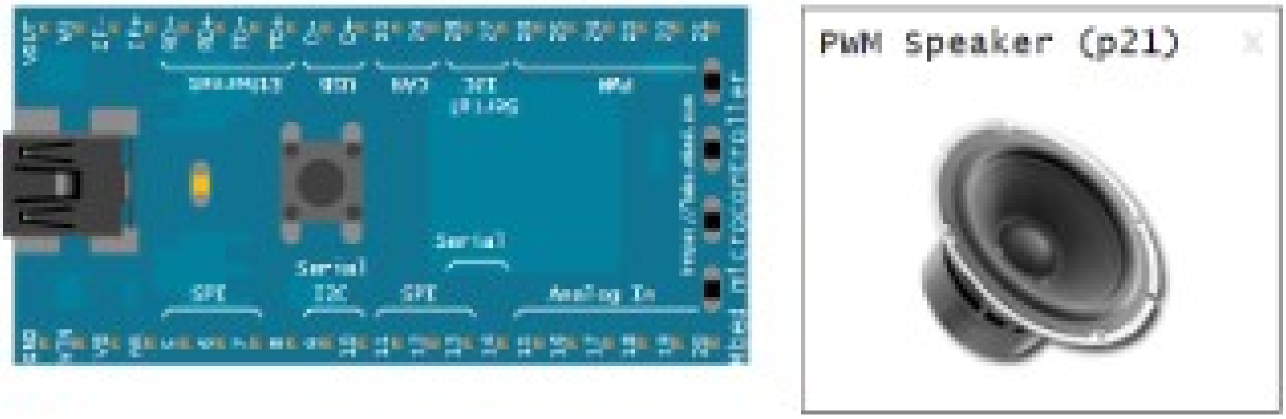
4.4

Aim Design a program to make ambulance sound.

Code

```
#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);
        }
    }
}
```

Circuit Design / Output



5 Lab 5 - 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

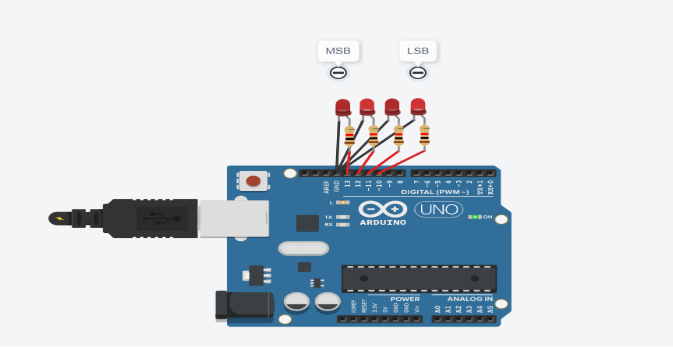
5.1

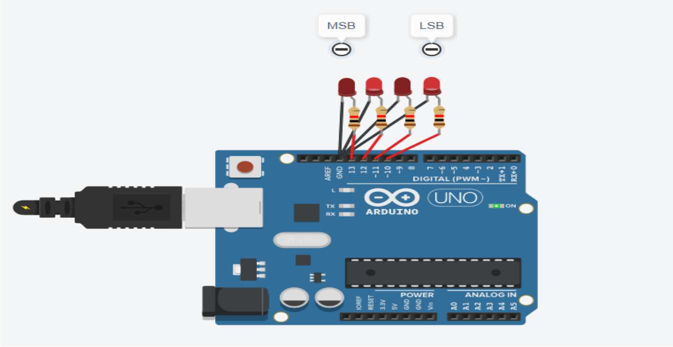
Aim Blink LEDs in alternate order – 1 and 3 together and 2 and 4 together.

Code

```
#define F(x) x ? HIGH : LOW
int pins[] = {13, 12, 11, 10};
void setup()
{
  for (int i = 0; i < 4; i++)
    pinMode(pins[i], OUTPUT);
}
void show(int x)
{
  int vals[] = {F(x & 8), F(x & 4), F(x & 2), F(x & 1)};
  for (int i = 0; i < 4; i++)
  {
    digitalWrite(pins[i], vals[i]);
  }
}

void loop()
{
  show(0b1010);
  delay(1000);
  show(0b0101);
  delay(1000);
}
```

Circuit Design / Output





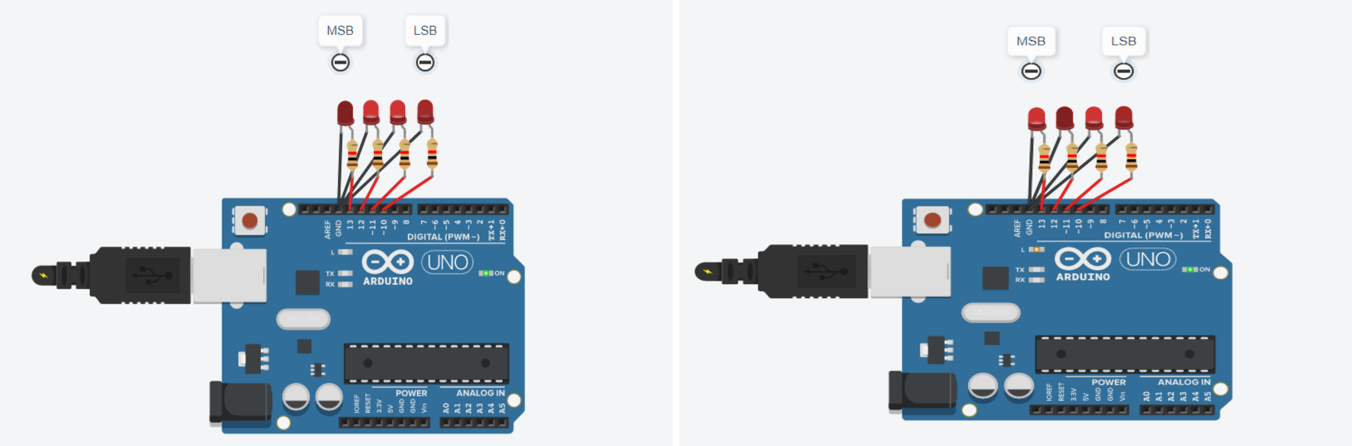
5.2

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

Code

```
#define F(x) x ? HIGH : LOW
int pins[] = {13, 12, 11, 10};
void setup()
{
    for (int i = 0; i < 4; i++)
        pinMode(pins[i], OUTPUT);
}
void show(int x)
{
    int vals[] = {F(x & 8), F(x & 4), F(x & 2), F(x & 1)};
    for (int i = 0; i < 4; i++)
    {
        digitalWrite(pins[i], vals[i]);
    }
}

void loop()
{
    for (int i = 1; i < 16; i++)
    {
        show(i);
        if (i % 2 == 1)
            delay(1000);
        else
            delay(2000);
    }
}
```

Circuit Design / Output



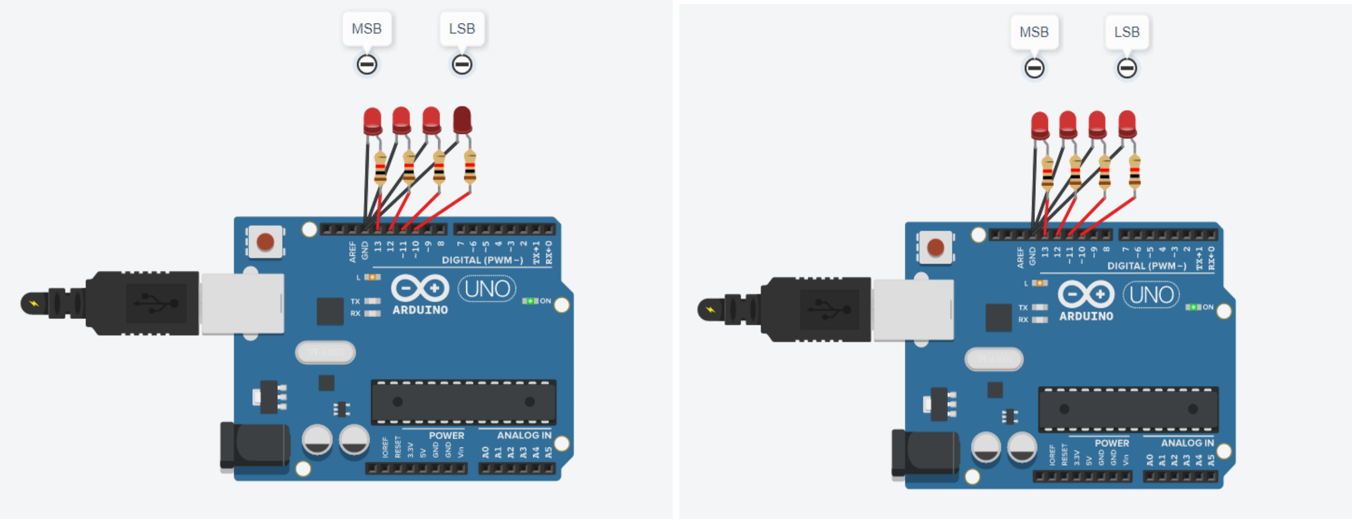
5.3

Aim Blink LEDs – for all composite number below 15.

Code

```
#define F(x) x ? HIGH : LOW
int pins[] = {13, 12, 11, 10};
void setup()
{
    for (int i = 0; i < 4; i++)
        pinMode(pins[i], OUTPUT);
}
void show(int x)
{
    int vals[] = {F(x & 8), F(x & 4), F(x & 2), F(x & 1)};
    for (int i = 0; i < 4; i++)
        digitalWrite(pins[i], vals[i]);
}
bool is_prime(int x)
{
    for (int i = 2; i < x; i++)
        if (x % i == 0)
            return false;
    return true;
}
void loop()
{
    for (int i = 2; i < 16; i += 1)
        if (!is_prime(i))
        {
            show(i);
            delay(1000);
        }
}
```

Circuit Design / Output



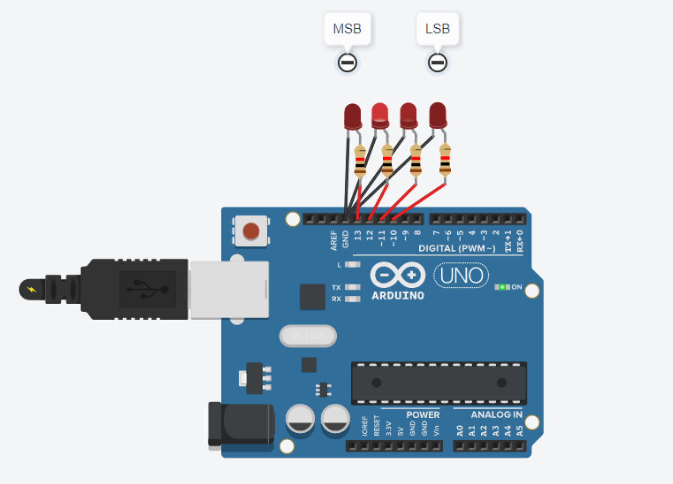
5.4

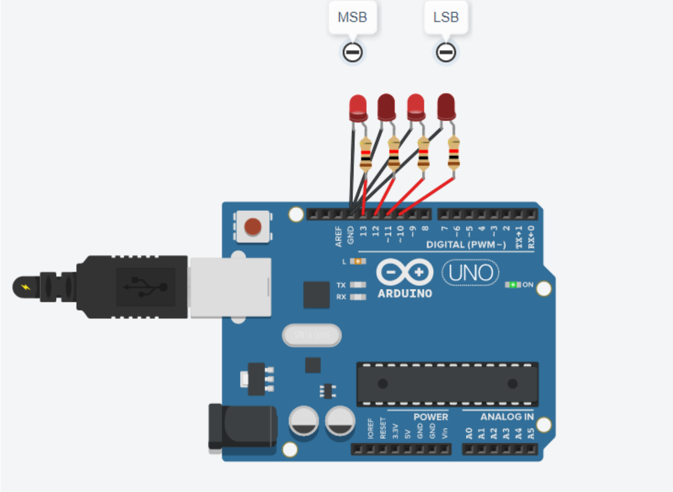
Aim Blink LEDs – to count even numbers

Code

```
#define F(x) x ? HIGH : LOW

int pins[] = {13, 12, 11, 10};

void setup()
{
    for (int i = 0; i < 4; i++)
        pinMode(pins[i], OUTPUT);
}
void show(int x)
{
    int vals[] = {F(x & 8), F(x & 4), F(x & 2), F(x & 1)};
    for (int i = 0; i < 4; i++)
    {
        digitalWrite(pins[i], vals[i]);
    }
}
void loop()
{
    for (int i = 0; i < 16; i += 2)
    {
        show(i);
        delay(1000);
    }
}
```

Circuit Design / Output





6 Lab 6 - 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.

6.1

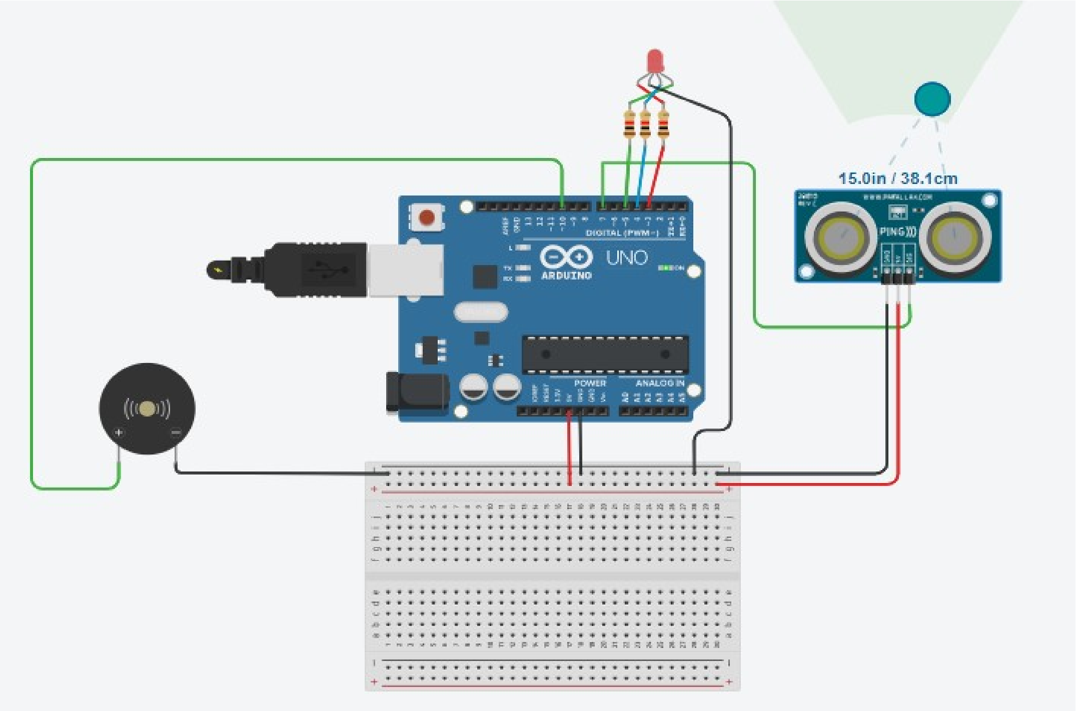
Aim Interface an ultrasonic sensor and display the distance measured in a screen and glow red, blue and green led if the distance is close, moderate and far respectively. Play different tones for different cases.

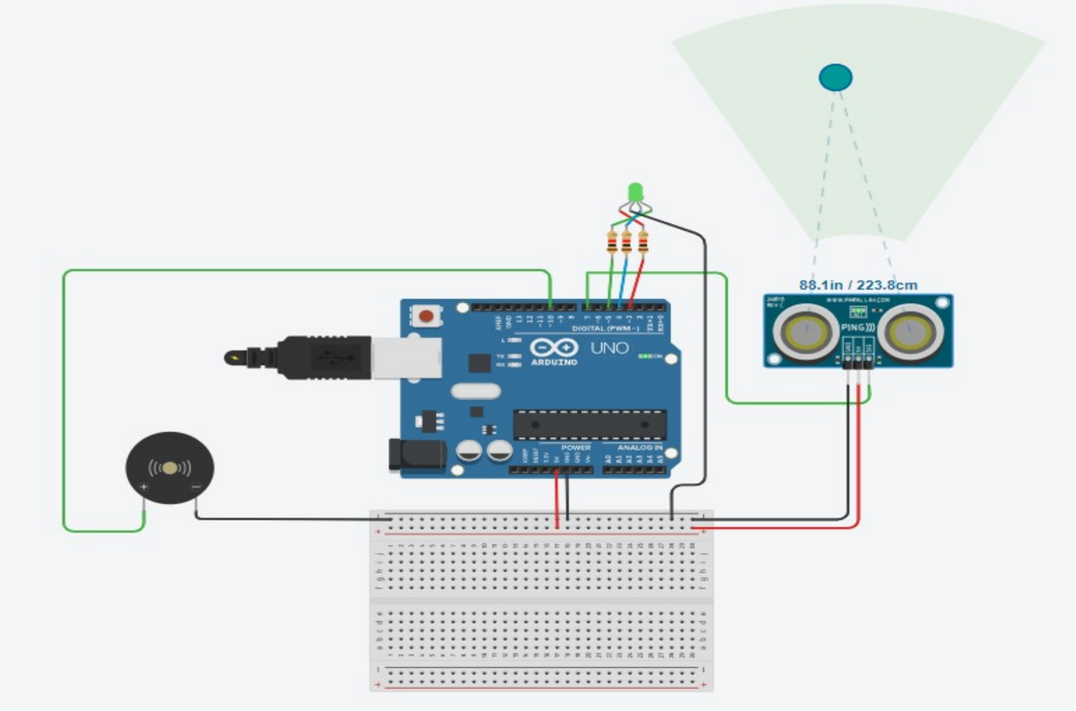
Code

```
#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 / Outputs





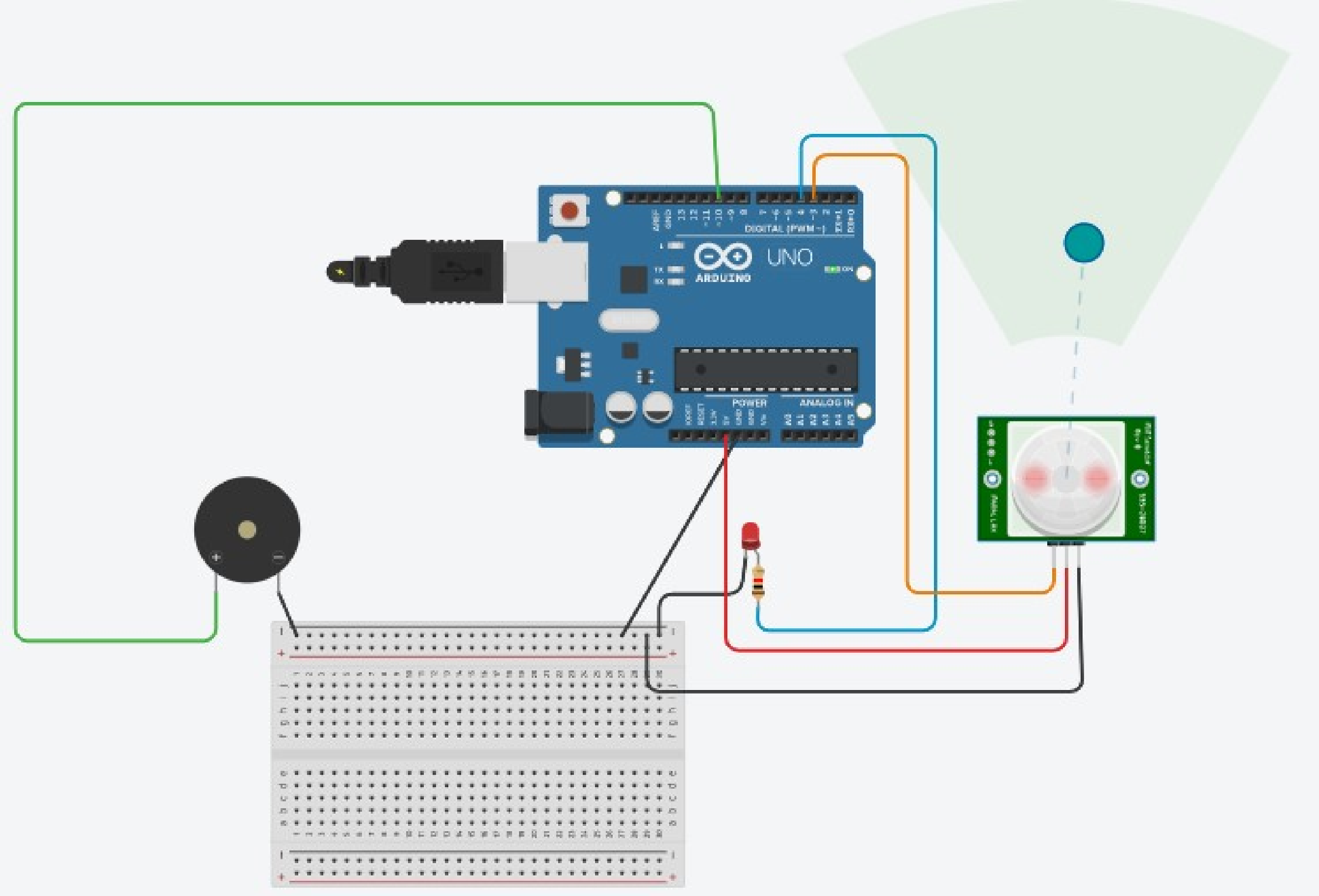
6.2

Aim 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.

Code

```
#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 / Outputs



7 Lab 7 - 11/04/2022

1. Interface two 7-segment display with Arduino Uno board. Implement the a) oddd counter and b) even counter and display the values in the seven segment display.
2. 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).
3. 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.

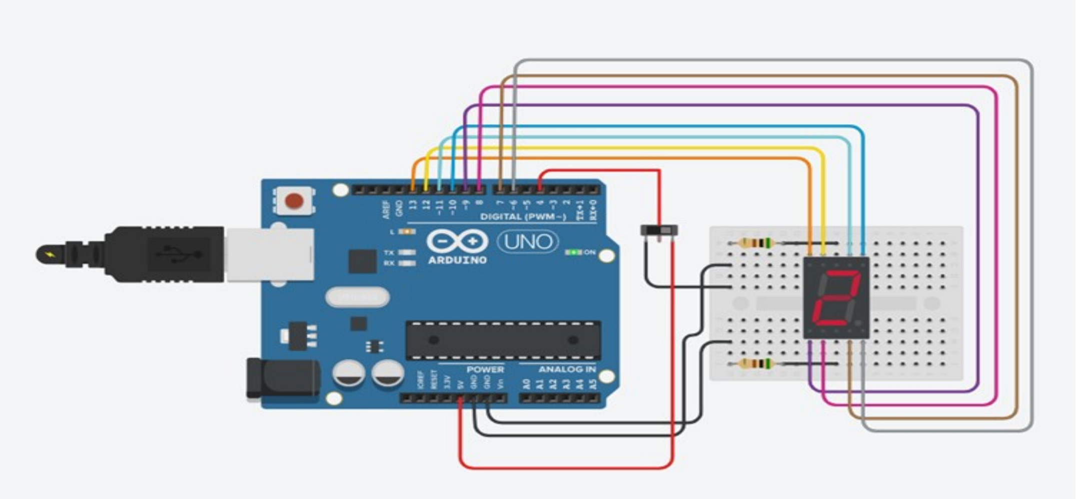
7.1

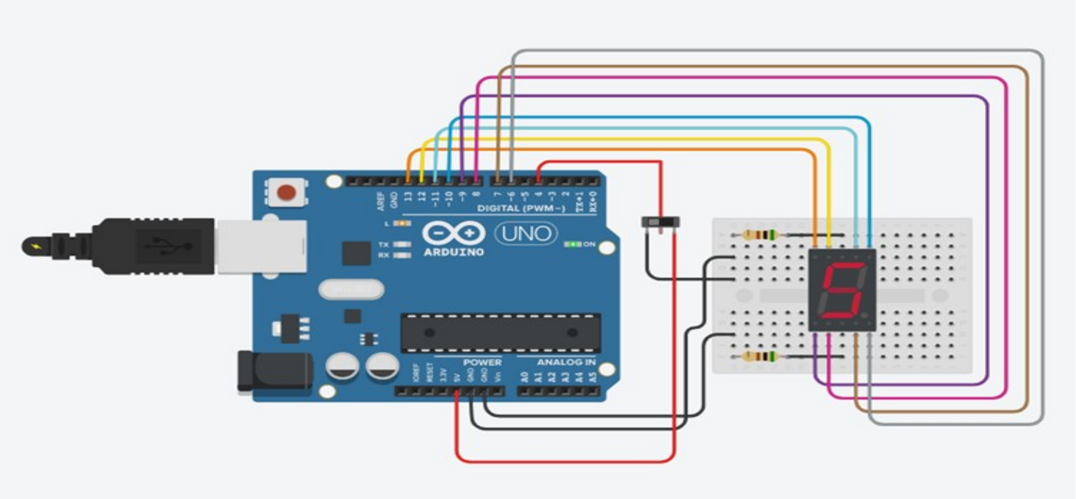
Aim Interface two 7-segment display with Arduino Uno board. Implement the a)odd counter and b)even counter and display the values in the seven segment display.

Code

```
unsigned const int A = 13;
unsigned const int B = 12;
unsigned const int C = 11;
unsigned const int D = 10;
unsigned const int E = 9;
unsigned const int F = 8;
unsigned const int G = 7;
unsigned const int H = 6;
unsigned const int btn = 4;

void setup(void)
{
  pinMode(A, OUTPUT);
  pinMode(B, OUTPUT);
  pinMode(C, OUTPUT);
  pinMode(D, OUTPUT);
  pinMode(E, OUTPUT);
  pinMode(F, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(H, OUTPUT);
  pinMode(btn, INPUT);
}
int pins[] = {A,B,C,D,E,F,G,H};

void zero(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void one(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void two(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

void three(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, LOW);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void four(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void five(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void six(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void seven(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void eight(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void nine(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, LOW);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void (*nums[])() ={zero,one,two,three, four, five,six,seven,eight,nine};

// Start
void loop(void)
{
  if(digitalRead(btn)==HIGH){
    for(int i=0;i<10;i+=2){
      nums[i]();
      delay(1000);
    }
  }else{
    for(int i=1;i<10;i+=2){
          nums[i]();
      delay(1000);
    }
  }
}
```

Circuit Design / Outputs





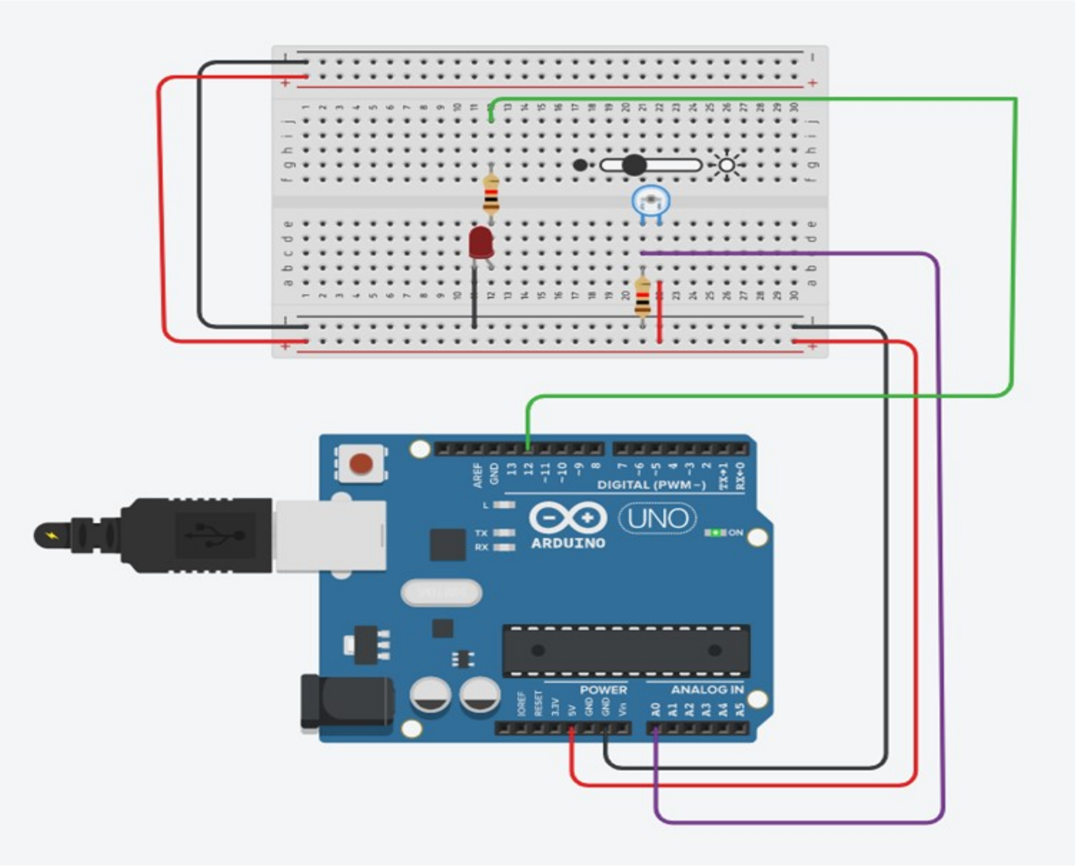
7.2

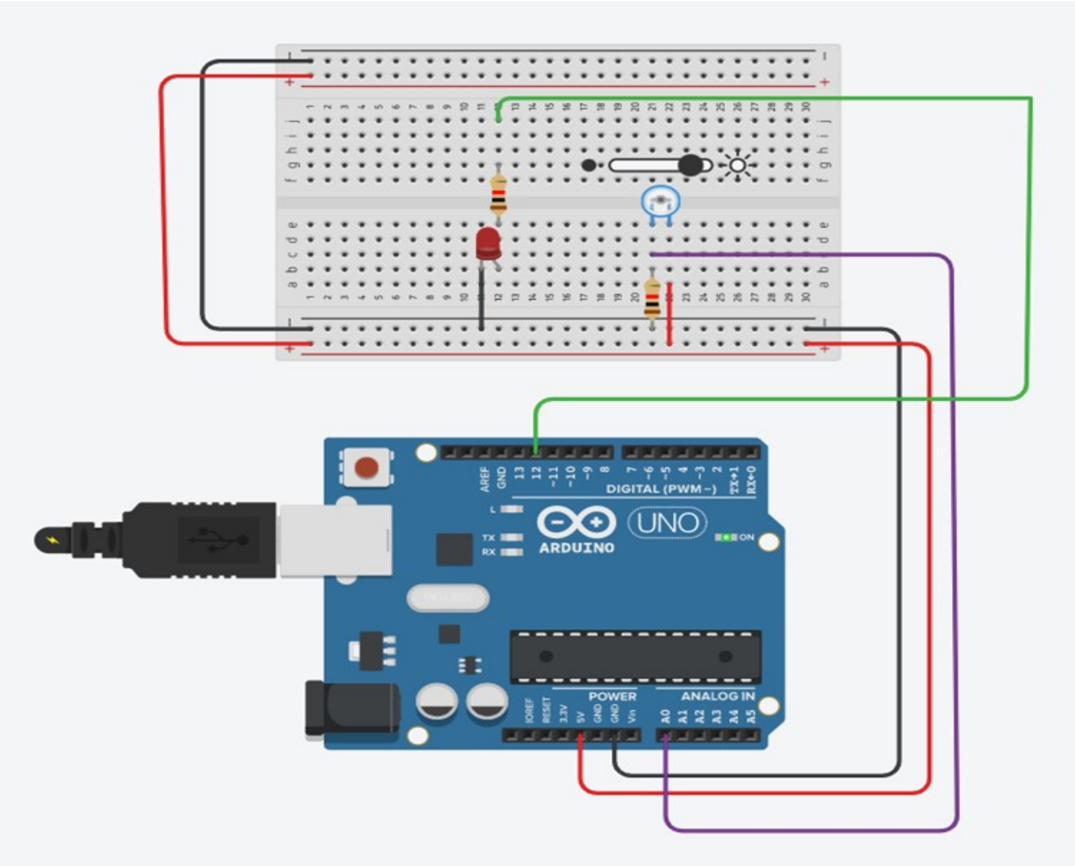
Aim Interface the ambient light sensor with Arduino Uno board. Check the light value from the sensor, and switch on/off the bulb (based on the threshold value).

Code

```
int LED = 12;
int MQ2pin = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  float sensorValue;
  sensorValue = analogRead(MQ2pin); // read analog input pin 0

  if(sensorValue >= 10){
    digitalWrite(LED,HIGH);

    Serial.println("\nSensor Value: ");
    Serial.print(sensorValue);
    Serial.println("\nDANGER\n");
  }
  else{
    digitalWrite(LED,LOW);
    Serial.println("\nSensor Value: ");
    Serial.print(sensorValue);

  }
  delay(1000);
}

float getsensorValue(int pin){
  return (analogRead(pin));
}
```

Circuit Design / Outputs





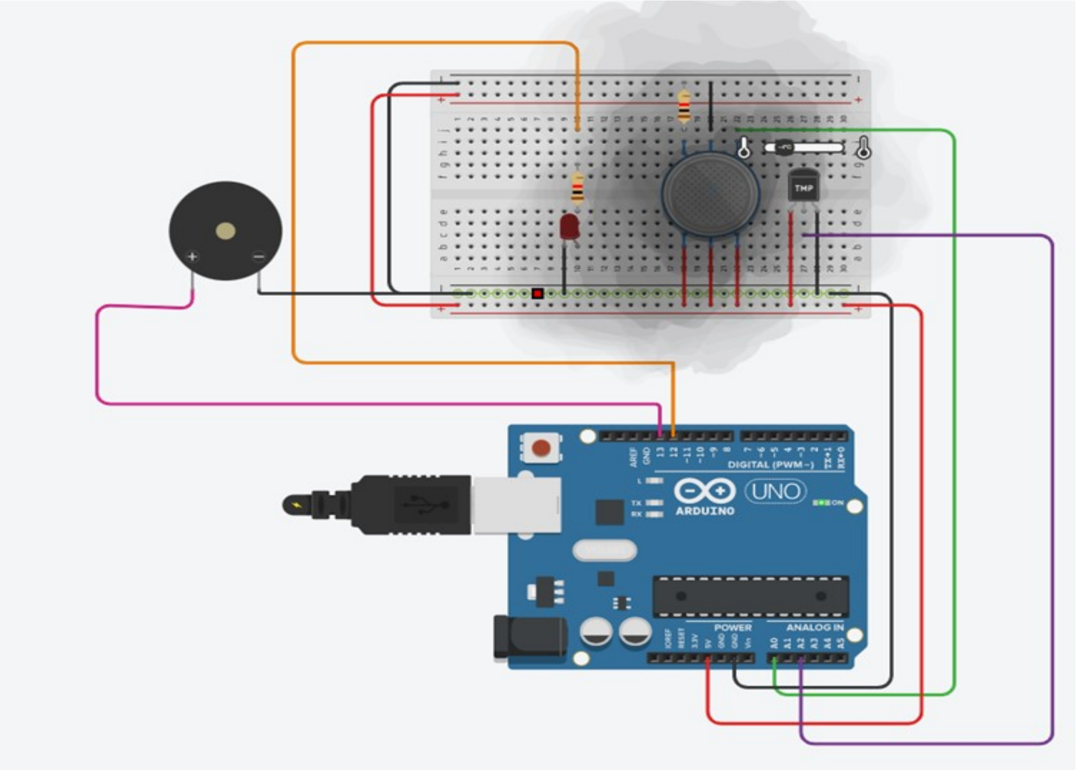
7.3

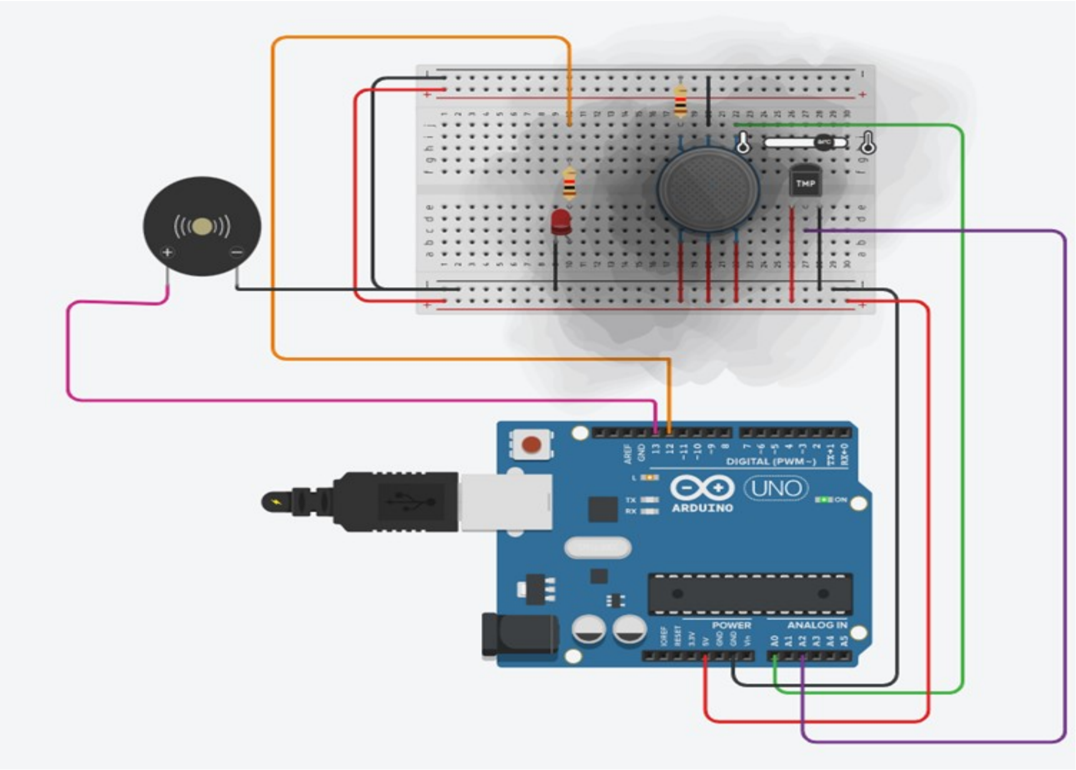
Aim Interface the temperature and gas sensor with Arduino Uno board. Check the temperature and the gas value, if the limit is beyond the threshold, switch on the bulb and make alarm using buzzer.

Code

```
int LED = 12;
int gas = A0;
int tmppin= A2;
int piezo = 13;

void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(piezo, OUTPUT);
  pinMode(gas,INPUT);
  pinMode(tmppin,INPUT);

}

void loop() {
  float sensorValue, tempSensor;
  sensorValue = analogRead(gas); // read analog input pin 0

  tempSensor= analogRead(tmppin);

  bool smoke= sensorValue >= 300;
  bool temp= tempSensor >100;

  if(smoke&&temp){
    digitalWrite(LED,HIGH);
    digitalWrite(piezo, HIGH);

    Serial.println("\nSmoke Sensor Value: ");
    Serial.print(sensorValue);
    Serial.println("\nTemp Sensor Value: ");
    Serial.print(tempSensor);
    Serial.println(" \nDANGER\n");
  }
  else{
    digitalWrite(LED,LOW);
    digitalWrite(piezo, LOW);
    Serial.println("Smoke Sensor Value: ");
    Serial.print(sensorValue);
    Serial.println("\nTemp Sensor Value: ");
    Serial.print(tempSensor);
  }
  delay(1000);
}

float getsensorValue(int pin){
  return (analogRead(pin));
}
```

Circuit Design / Outputs





8 Lab 8 - 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 color of the LED based on the 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.

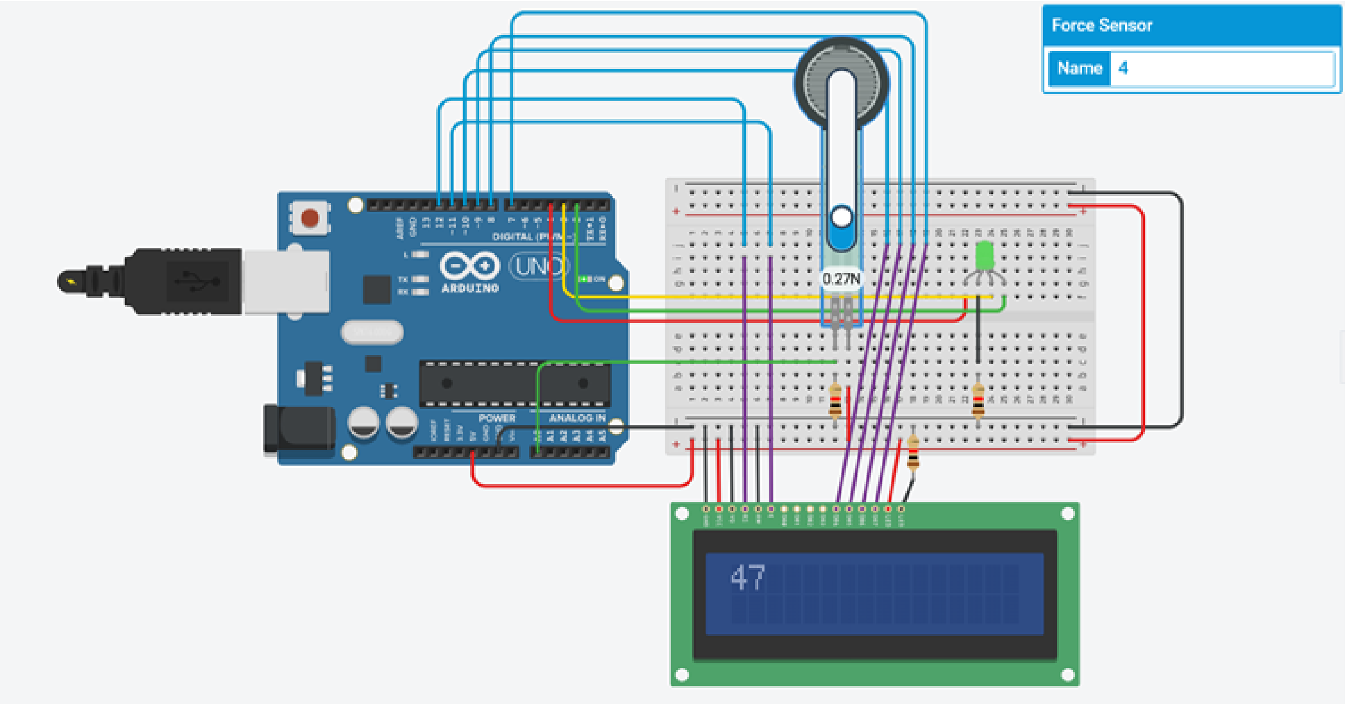
8.1

Aim 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 color of the LED based on the some threshold values.

Code

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

int force = 0;
int red = 4;
int blue = 3;
int green = 2;
void setup()
{
  lcd.begin(16, 2);
  pinMode(A0, INPUT);
  pinMode(green, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(blue, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  force = analogRead(A0);
  Serial.println(force);
  lcd.clear();
  lcd.print(force);
  if(force<100){
           digitalWrite(green, HIGH);
    digitalWrite(blue, LOW);
    digitalWrite(red, LOW);
  }else if(force < 300){
    digitalWrite(green, LOW);
    digitalWrite(blue, HIGH);
    digitalWrite(red, LOW);
  }else{
    digitalWrite(green, LOW);
    digitalWrite(blue, LOW);
    digitalWrite(red, HIGH);
  }
  delay(1000);
}
```

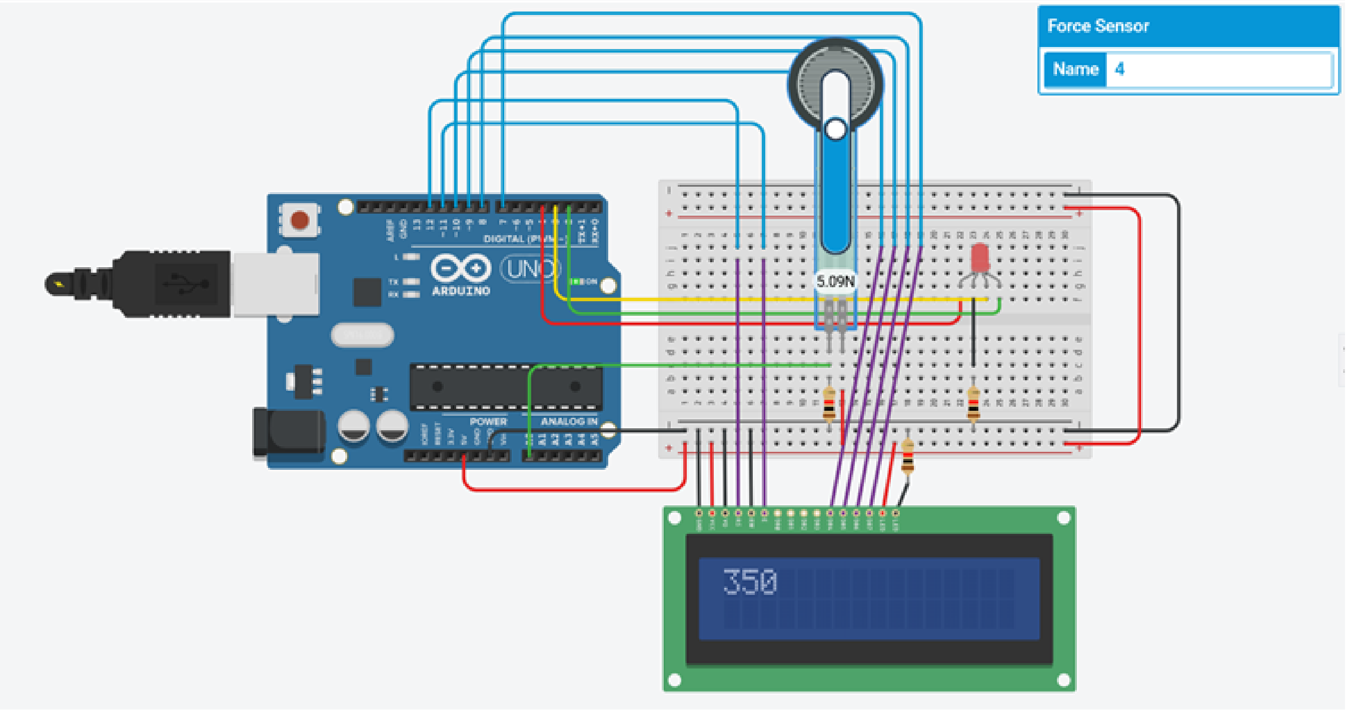
Circuit Design / Outputs



Force Sensor

Name 4

47



Force Sensor

Name 4

350

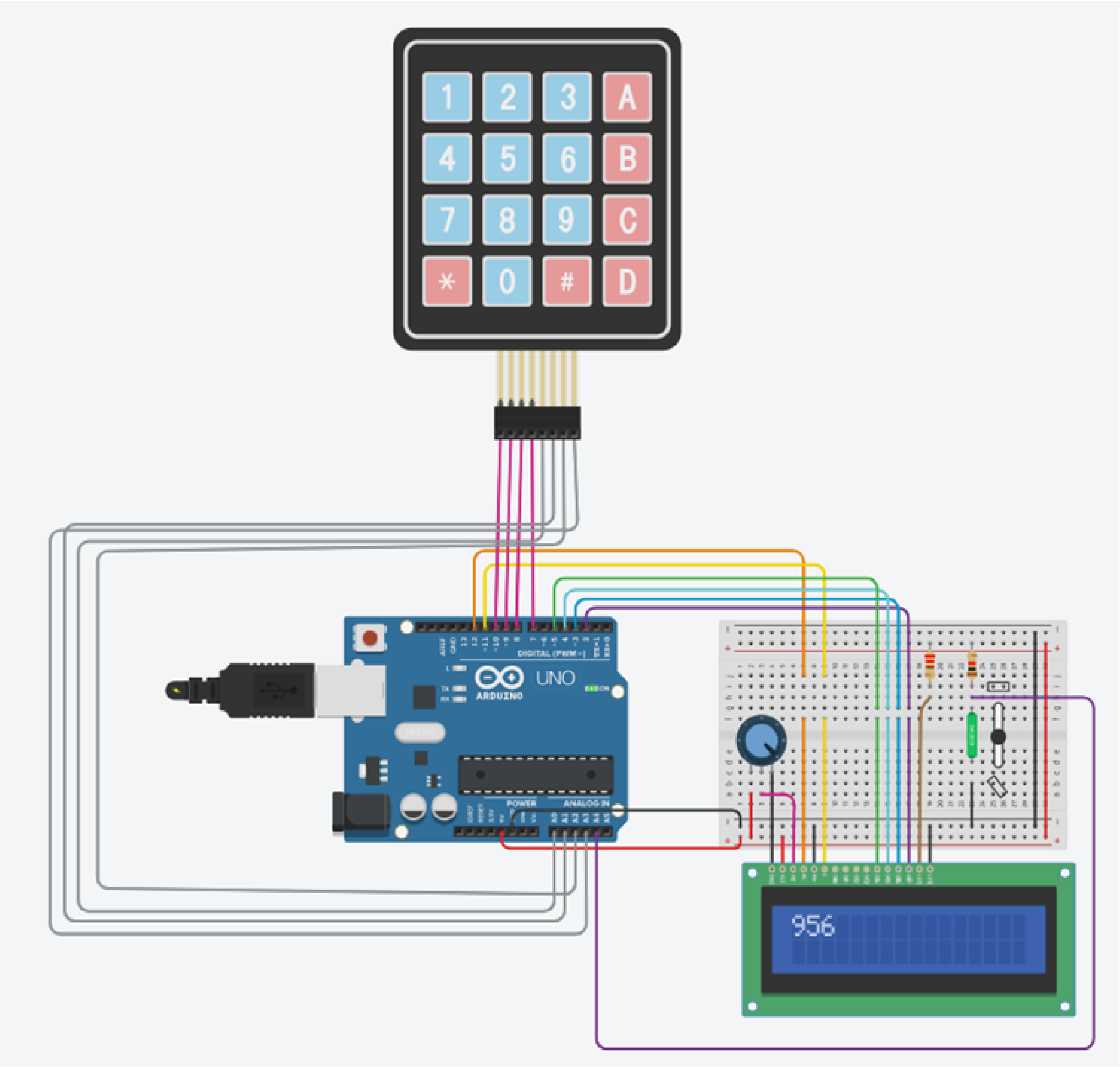
8.2

Aim 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.

Code

```
#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
char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

//Code that shows the 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);
}
```

Circuit Design / Outputs



1
2
3
A
4
5
6
B
7
8
9
C
*
0
#
D
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
956