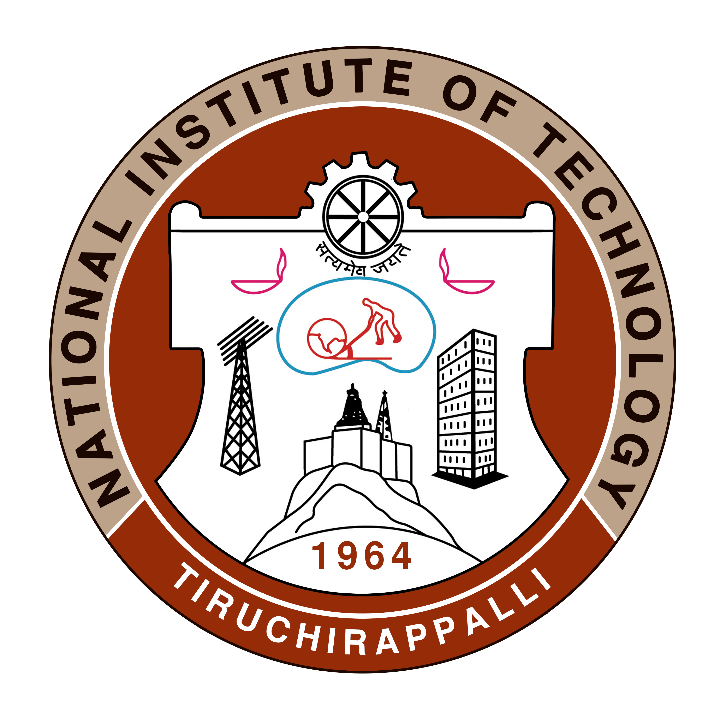
****

**EMBEDDED LAB REPORT**

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Topic** | **Lab** | **Date** | **Page**  **No** |
| **1** | **Blink LEDs in alternate order** | **1** | **31/01/2022** |  |
| **2** | **Blink LEDs – Alternate Wait** |  |
| **3** | **Blink LEDs – Composite Numbers** |  |
| **4** | **Blink LEDS – Even Numbers** |  |
| **5** | **Hex and Decade Counters – Switch** | **2** | **07/02/2022** |  |
| **6** | **Octal Counter – Ticker and Timeout** |  |
| **7** | **Odd and Even Counter – Ticker, Timeout and Switch** |  |
| **8** | **Intensity of LED - Pwmout** | **3** | **14/02/2022** |  |
| **9** | **Intensity of LED - Potentiometer** |  |
| **10** | **Intensity/Blinking of LEDs – Potentiometer, Switch** |  |
| **11** | **LED and Speaker – Pwmout** | **4** | **28/02/2022** |  |
| **12** | **Alarm System – Double Beep every 5s** |  |
| **13** | **LEDs and Speaker – Increase and Decrease intensity/frequency using switch** |  |
| **14** | **Simulate Ambulance sound** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Topic** | **Lab** | **Date** | **Page**  **No** |
| **15** | **Tinkercad – Blink LEDs in alternate order** | **5** | **07/03/2022** |  |
| **16** | **Tinkercad – Blink LEDs – Alternate Wait** |  |
| **17** | **Tinkercad – Blink LEDs – Composite** |  |
| **18** | **Tinkercad – Blink LEDs – Even Numbers** |  |
| **19** | **Ultrasonic Sensor** | **6** | **28/03/2022** |  |
| **20** | **PIR Sensor** |  |
| **21** | **7-Segment Display**  **Odd and Even Counter** | **7** | **11/04/2022** |  |
| **22** | **Ambient Light Sensor** |  |
| **23** | **Temperature and Gas Sensor** |  |
| **24** | **Force Sensor** | **8** | **18/04/2022** |  |
| **25** | **Keypad and Tilt Sensor** |  |

**Questions**

**MBED**

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) 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.

6) 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.

7) 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.

8) Increase the intensity of one LED while decreasing the other LED's intensity at the same unit. Display the intensity value of the LEDs.

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 (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.

11) Write a program to increase the intensity of external LED. At the same rate, increase the sound intensity of speaker.Libraries Used: pwmout

12) Design an alarm system, which makes double beep sound for every 5s using a speaker. Libraries To Be Used: pwmout, analogin

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. Proportionally increase or decrease the intensity of two LEDs.

14) Simulate ambulance sound using a speaker.

**Tinkercad**

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

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

17) Blink LEDs – for all composite number below 15.

18) Blink LEDs – to count even numbers.

19) 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.

20) 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.

21) 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.

22) 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).

23) 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.

24) 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 some threshold values.

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.

**Experiment 1**

**Aim**

To blink LEDs in alternate order – 1 and 3 should blink together and 2 and 4 should blink together.

**Code**

#include "mbed.h"

DigitalOut led1(*LED1*);

DigitalOut led2(*LED2*);

DigitalOut led3(*LED3*);

DigitalOut led4(*LED4*);

int main() {

    led1 = led3 = 1;

    led2 = led4 = 0;

    while (1) {

        led1 = !led1;

        led3 = led1;

        printf("Blink! LED is now *%d*\n", led1.read());

        led2 = !led2;

        led4 = led2;

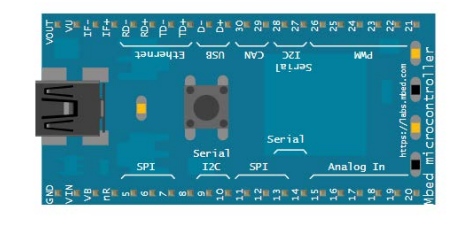
        printf("Blink! LED is now *%d*\n", led2.read());

        wait\_ms(500);

    }

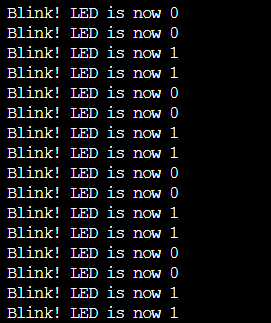
}

**Output**

****

****

**Serial Output**

****

**Experiment 2**

**Aim**

To blink LEDs and count from 1 to 15. If the board is counting odd value, it should wait for 1 sec, else it should wait for 2 sec.

**Code**

#include "mbed.h"

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

int main() {

    leds = 0;

    for(int i=1; i<=15; i++) {

        leds = i;

        if(i % 2 == 1) {

            wait\_ms(1000);

        } else {

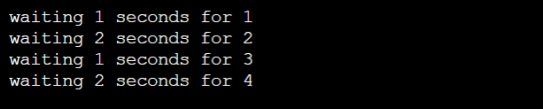
            wait\_ms(2000);

        }

    }

}

**Serial Output**

****

**Experiment 3**

**Aim**

To blink LEDs for all composite numbers below 15.

**Code**

#include "mbed.h"

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

bool isPrime(int *k*) {

    if(*k* <= 1) {

        return false;

    }

    for(int i=2; i<=sqrt(*k*); i++) {

        if(*k*%i == 0)

            return false;

    }

    return true;

}

int main() {

    leds = 0;

    for(int i=1; i<=15; i++) {

        if(!isPrime(i))

            leds=i;

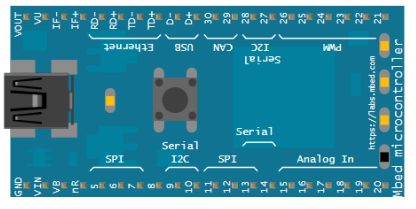
        printf("*%d* \n", i);

        wait\_ms(1000);

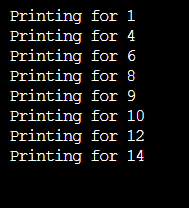
    }

}

**Output**

****

**Serial Output**

****

**Experiment 4**

**Aim**

To blink LEDs to count even numbers

**Code**

#include "mbed.h"

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

int main() {

    leds = 0;

    for(int i=1; i<=15; i++) {

        if(i%2 == 0) {

            leds=i;

            printf("Printing for *%d*\n", i);

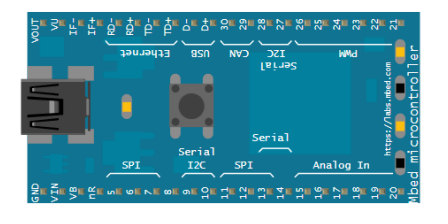
        }

        wait\_ms(1000);

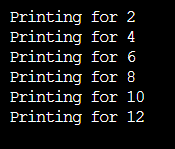
    }

}

**Output**

****

**Serial Output**

****

**Experiment 5**

**Aim**

To switch between hex counter and decade counter using switches and display the current value of the counter with the help of 4 LEDs and the screen using the InterruptIn and BusOut libraries.

**Code**

#include "mbed.h"

DigitalOut led(*LED1*);

InterruptIn btn(*p5*);

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

void hex() {

    while(1) {

        for(int i=0; i<16; i++) {

            leds = i;

            printf("Hex: *%d* \n", i);

            wait(1);

        }

    }

}

void dec() {

    while(1) {

        for(int i=0; i<10; i++) {

            leds = i;

            printf("Dec: *%d* \n", i);

            wait(1);

        }

    }

}

int main() {

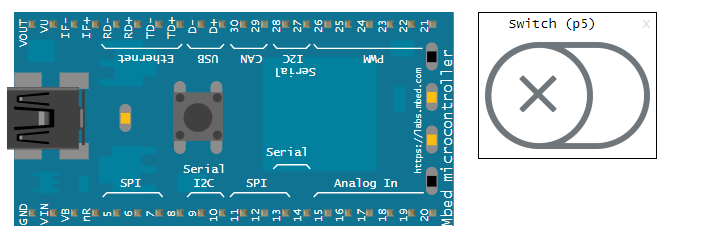
    btn.rise(&hex);

    btn.fall(&dec);

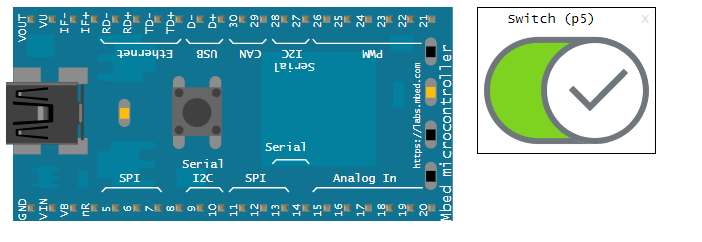
}

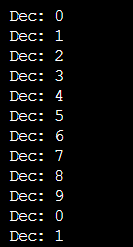
**Output**

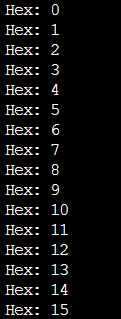
**Dec:**

****

**Hex:**

****

**Serial Output:**

**Dec: Hex:**

**2.**

**Experiment 6**

• 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.

• Libraries Used: TimeOut, Ticker

**Aim**

To display the octal counter using the onboard LEDs using Ticker Object and after 10 seconds reset all the values of the onboard LED using Timeout Object.

**Code**

#include "mbed.h"

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

Ticker tk;

Timeout to;

int current = 0;

void octal() {

    leds = (current++)%8;

    printf("Octal: *%d* \n", leds.read());

}

void reset() {

    printf("Reset Done \n");

    current=0;

    leds=0;

}

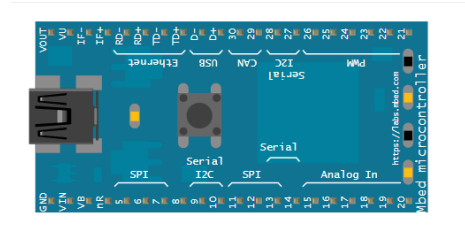
int main() {

    tk.attach(&octal, 1);

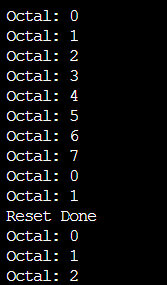
    to.attach(&reset, 10);

}

**Output**

****

**Serial Output**

****

**Experiment 7**

**Aim**

To display the odd counter using the onboard LEDs and screen and after 10 seconds, reset all the values of the onboard LED and display even counter using Timeout Object with a switch option to manually toggle between two counters.

**Code**

#include "mbed.h"

InterruptIn my\_switch(*p5*);

Timeout my\_timeout;

Ticker my\_ticker;

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

int i = 0;

bool is\_odd = 0;

void switch\_odd() {

    printf("switching to odd\n");

    i = 1;

    is\_odd = true;

}

void switch\_even() {

    printf("switching to even\n");

    i = 0;

    is\_odd = false;

}

void increase\_count() {

    myleds = i % 16;

    i += 2;

    printf("count: *%d*\n", myleds.read());

}

void reset() {

    printf("resetting count\n");

    switch\_even();

}

int main() {

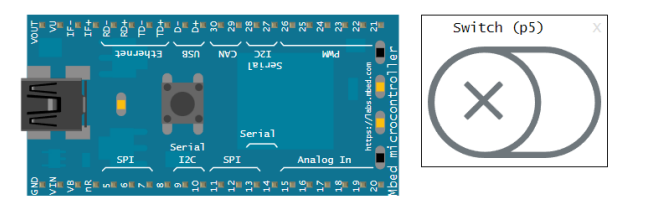
    my\_ticker.attach(&increase\_count, 0.5);

    my\_timeout.attach(&reset, 10.0);

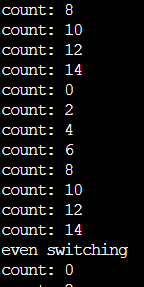
    my\_switch.rise(&switch\_odd);

    my\_switch.fall(&switch\_even);

}

**Output **

**Serial Output**

****

**Experiment 8**

**Aim**

To Increase the intensity of one LED while descreasing the other LED's intensity at the same unit and display the intensity value of the LEDs using pwmout library.

**Code**

#include "mbed.h"

PwmOut led1(*p5*);

PwmOut led2(*p6*);

int main() {

    led1 = 0;

    led2 = 1;

    int flag = 1;

    while (1) {

        if(flag == 1) {

            led1 = led1 + 0.10;

            led2 = led2 - 0.10;

        } else {

            led1 = led1 - 0.10;

            led2 = led2 + 0.10;

        }

        printf("Blink! LED1 is now *%0.2f* \n", led1.read());

        printf("Blink! LED2 is now *%0.2f* \n", led2.read());

        wait(0.1);

        if(led1 == 1.0) {

            flag = 0;

        }

        if(led1 == 0.0) {

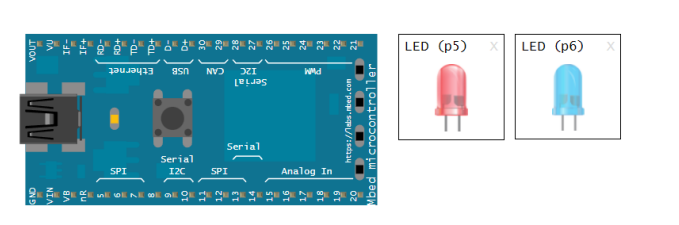
            flag = 1;

        }

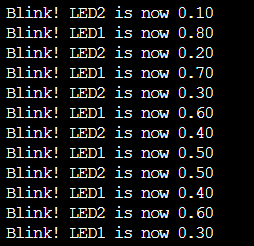
    }

}

**Output**

****

**Serial Output**

****

**Experiment 9**

**Aim**

To Interface potentiometer with mbed board and based on the value of potentiometer, adjust the intensity of an external LED using pwmOut and analogIn libraries.

**Code**

#include "mbed.h"

AnalogIn p(*p15*);

PwmOut led(*p5*);

int main() {

    float var;

    while (1) {

        var = p.read();

        led = var;

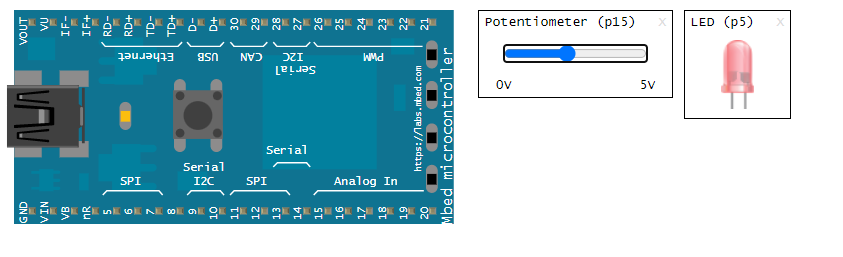
        printf("LED is now *%0.2f* \n", led.read());

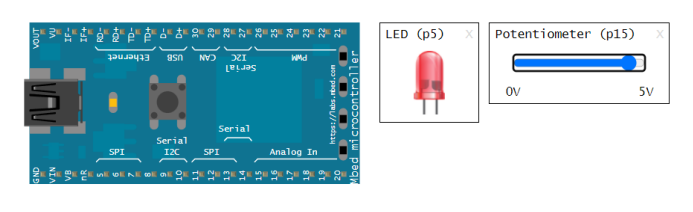
        wait(1);

    }

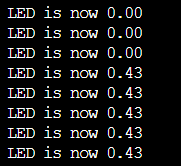
}

**Output**

****

****

**Serial Output**

****

**Experiment 10**

**Aim**

To Interface switch, 2 LEDs, potentiometer with the mbed board and If the switch is on, control the intensity of the LEDs using potentiometer (with one LED increasing and the other decreasing at the rate given via potentiometer) and 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 p(*p15*);

PwmOut led1(*p5*);

PwmOut led2(*p6*);

InterruptIn btn(*p7*);

void on() {

    float var;

    while(1) {

        var = p.read();

        led1 = var;

        led2 = 1-var;

        printf("LED1 is *%0.2f*. LED2 is *%0.2f* \n", led1.read(), led2.read());

        wait(0.2);

    }

}

void off() {

    led1 = 0;

    led2 = 1;

    int flag = 1;

    while (1) {

        if(flag == 1) {

            led1 = led1 + 0.10;

            led2 = led2 - 0.10;

        } else {

            led1 = led1 - 0.10;

            led2 = led2 + 0.10;

        }

        printf("Blink! LED1 is now *%0.2f* \n", led1.read());

        printf("Blink! LED2 is now *%0.2f* \n", led2.read());

        wait(0.1);

        if(led1 == 1.0) {

            flag = 0;

        }

        if(led1 == 0.0) {

            flag = 1;

        }

    }

}

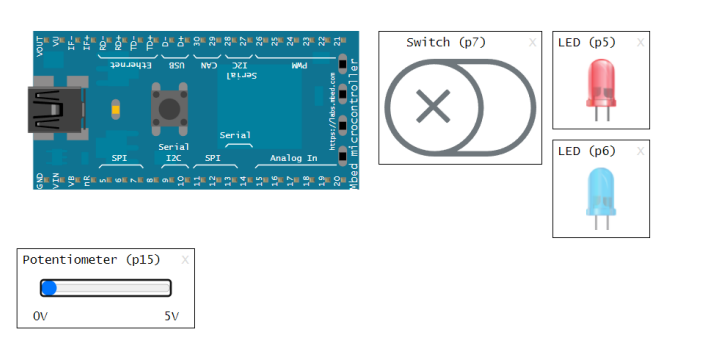
int main() {

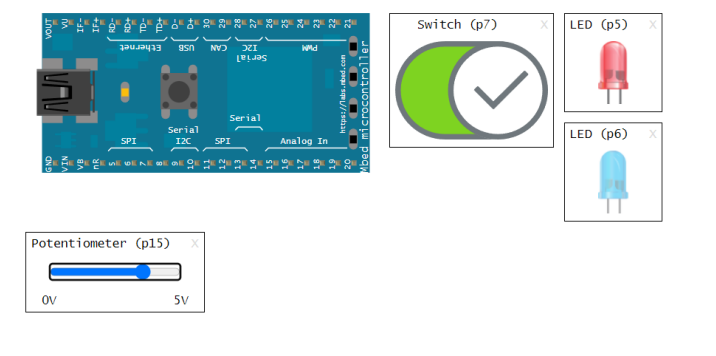
    btn.rise(&on);

    btn.fall(&off);

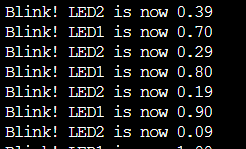
}

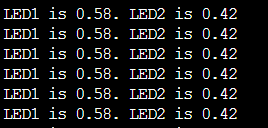
**Output**

****

****

**Serial Output**

****

****

**Experiment 11**

**Aim**

To increase the intensity of external LED and at the same rate, increase the sound intensity of speaker using pwmOut library.

**Code**

#include "mbed.h"

PwmOut led(*p5*);

PwmOut speaker(*p21*);

void play\_tone(float *frequency*, float *volume*, int *interval*, int *rest*) {

    speaker.period(1.0 / *frequency*);

    speaker = *volume*;

    wait(*interval*);

    speaker = 0.0;

    wait(*rest*);

}

int main()

{

    led=0.0;

    float vol=0.0;

    while(1) {

        led = led + 0.1;

        vol += 0.1;

        printf("led=*%f*\n", led.read());

        printf("vol=*%f*\n", vol);

        play\_tone(200.0, vol, 1, 0);

        wait(1);

        if(led==1.0){

            led=0.0;

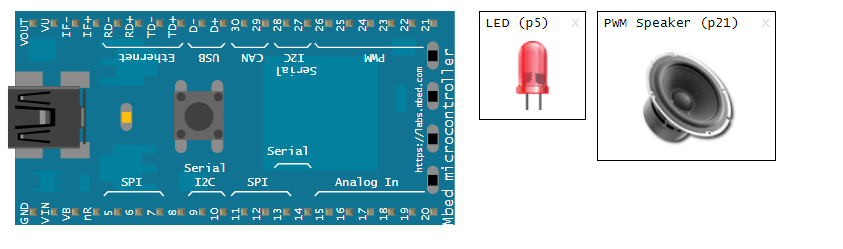
            vol=0.0;

        }

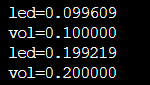
    }

}

**Output**

****

**Serial Output**

****

**Experiment 12**

**Aim**

To design an alarm system, which makes double beep sound for every 5s using a speaker using pwmout and analogin libraries.

**Code**

#include "mbed.h"

Ticker tk;

PwmOut speaker(*p21*);

void beep(){

    speaker=0.3;

    wait(0.1);

    speaker=0;

    wait(0.1);

    speaker=0.3;

    wait(0.1);

    speaker=0;

}

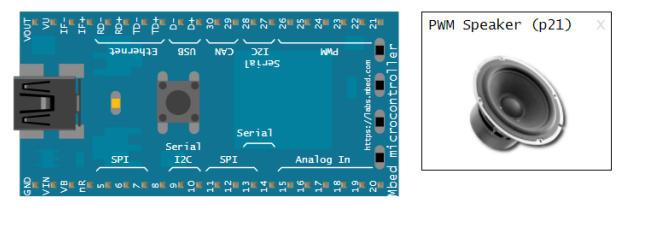
int main()

{

    tk.attach(&beep,5.0);

}

**Output**

****

**Serial Output**

****

**Experiment 13**

**Aim**

To have a switch, 2 LEDs and a speaker interfaced with mbed board and upon switching on, increase the intensity of the sound and the upon switching off decrease the intensity of the sound and proportionally increase or decrease the intensity of two LEDs.

**Code**

#include "mbed.h"

PwmOut speaker(*p21*);

PwmOut led1(*p5*);

PwmOut led2(*p6*);

InterruptIn btn(*p7*);

float sign = -0.1;

void inc(){

    sign = 0.1;

}

void dec(){

    sign = -0.1;

}

int main()

{

    btn.rise(&inc);

    btn.fall(&dec);

    speaker=0;

    while(1) {

        if(sign>0 && speaker>=1)

            speaker=0;

        if(sign<0 && speaker<=0)

            speaker=1;

        led1 = speaker;

        led2 = speaker;

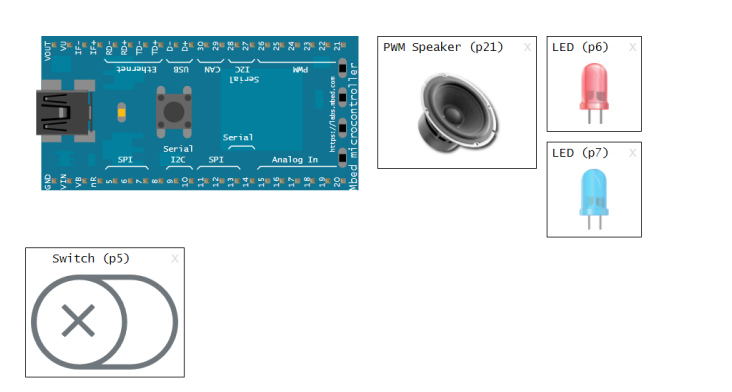
        speaker = speaker + sign;

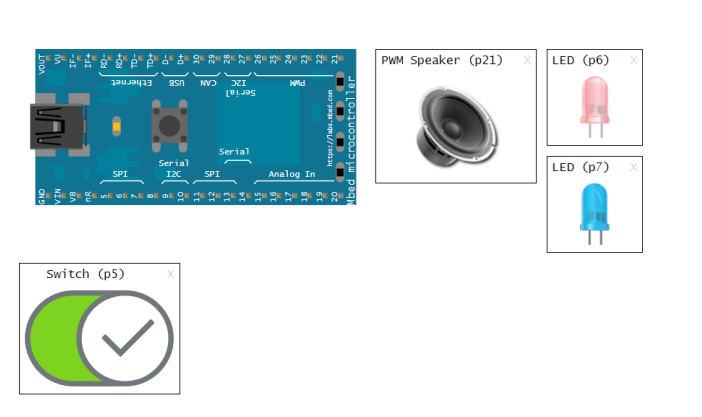
        wait(0.2);

    }

}

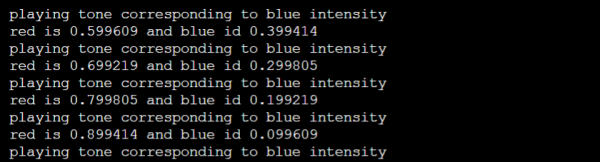
**Output**

****

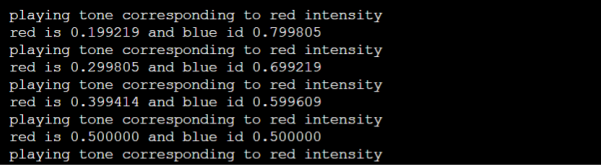
****

**Serial Output**

**Case 1**

****

**Case 2**

****

**Experiment 14**

**Aim**

To simulate the ambulance sound using the PwmOut library.

**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);

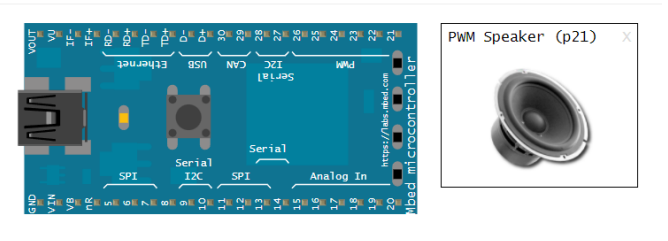
            wait(0.5);

        }

    }

}

**Output**

****

**Experiment 15**

**Aim**

To blink LEDs in alternate order – 1 and 3 should blink together and 2 and 4 should blink together.

**Tinkercad Link**

<https://www.tinkercad.com/things/1PwYTabHSMP>

**Code**

*// C++ code*

*//*

void setup()

{

  pinMode(8, OUTPUT);

  pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(11, OUTPUT);

}

void loop()

{

  digitalWrite(8, HIGH);

  digitalWrite(10, HIGH);

  digitalWrite(9, LOW);

  digitalWrite(11, LOW);

  delay(1000); *// Wait for 1000 millisecond(s)*

  digitalWrite(8, LOW);

  digitalWrite(10, LOW);

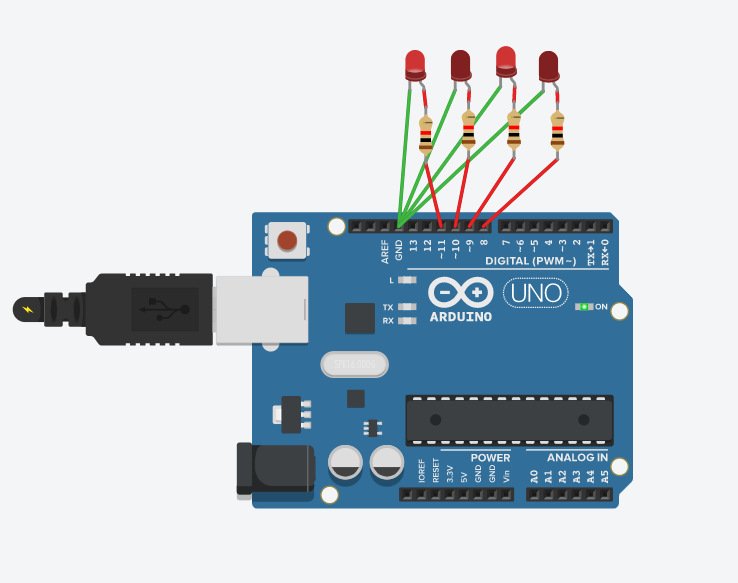
  digitalWrite(9, HIGH);

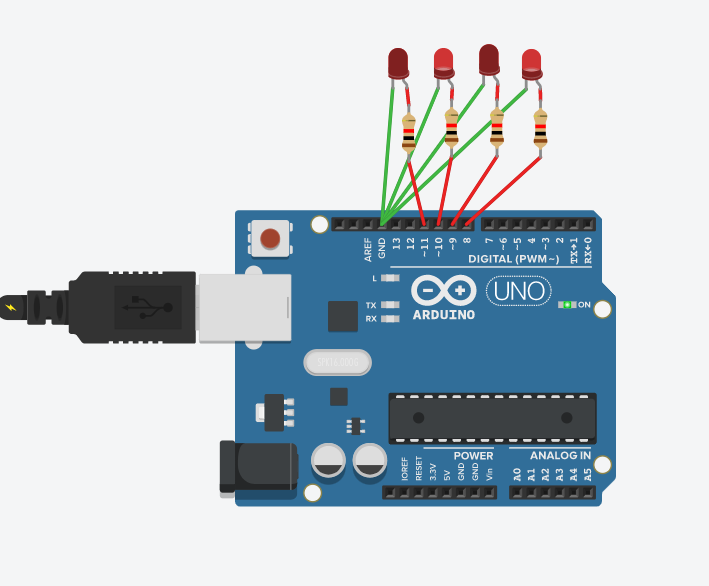
  digitalWrite(11, HIGH);

  delay(1000); *// Wait for 1000 millisecond(s)*

}

**Output**

****

****

**Experiment 16**

**Aim**

To blink LEDs and count from 1 to 15. If the board is counting odd value, it should wait for 1 sec, else it should wait for 2 secs.

**Tinkercad Link**

<https://www.tinkercad.com/things/fDj0RiOP8Hf>

**Code**

*// C++ code*

*//*

void setup()

{

  pinMode(8, OUTPUT);

  pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(11, OUTPUT);

}

void display(int *n*)

{

  digitalWrite(8, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(9, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(10, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(11, *n*&1?HIGH:LOW);

}

void loop(){

   for(int i=0;i<16;i++)

   {

     display(i);

      Serial.print("printing ");

      Serial.print(i);

      Serial.print(" with a delay of ");

     if(i%2){

       Serial.println("1 second");

       delay(1000);

     }

     else{

       Serial.println("2 seconds");

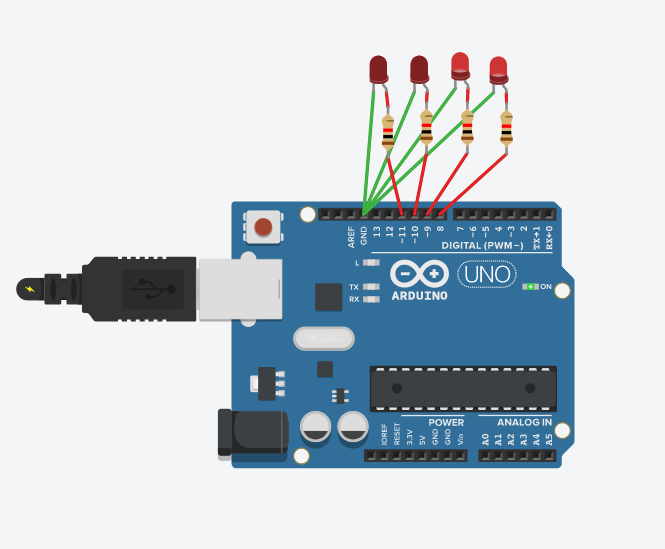
       delay(2000);

     }

   }

}

**Output**

****

**Experiment 17**

**Aim**

To blink LEDs for all composite numbers below 15.

**Tinkercad Link**

<https://www.tinkercad.com/things/cpcjoc7UaTt>

**Code**

*// C++ code*

*//*

void setup()

{

  pinMode(8, OUTPUT);

  pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(11, OUTPUT);

}

void display(int *n*)

{

  digitalWrite(8, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(9, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(10, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(11, *n*&1?HIGH:LOW);

}

bool isPrime(int *n*) {

  for(int i=2; i<=*n*/2; i++) {

    if(*n* % i == 0)

      return false;

  }

  return true;

}

void loop(){

   for(int i=1; i<=15;i++)

   {

     if(!isPrime(i)) {

       display(i);

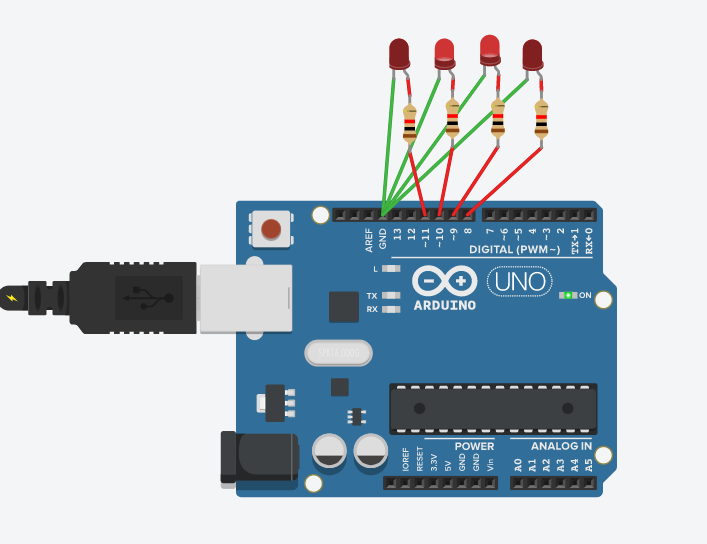
       delay(2000);

     }

   }

}

**Output**

****

**Experiment 18**

**Aim**

To blink LEDs to count even numbers.

**Tinkercad Link**

<https://www.tinkercad.com/things/0F0UdPUgjv8>

**Code**

*// C++ code*

*//*

void setup()

{

  pinMode(8, OUTPUT);

  pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

  pinMode(11, OUTPUT);

}

void display(int *n*)

{

  digitalWrite(8, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(9, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(10, *n*&1?HIGH:LOW);

*n* >>= 1;

  digitalWrite(11, *n*&1?HIGH:LOW);

}

void loop(){

   for(int i=1; i<=15;i++)

   {

     if(i%2 == 0) {

       display(i);

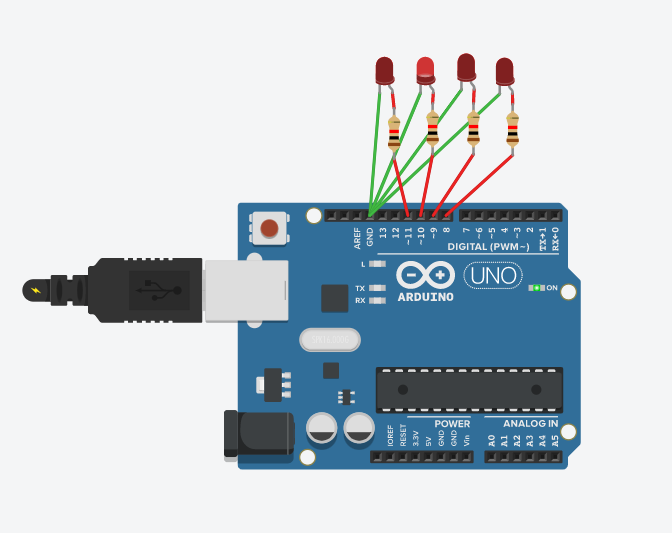
       delay(2000);

     }

   }

}

**Output**

****

**Experiment 19**

**Aim**

To 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 and to play different tones for

different cases.

**Tinkercad Link**

<https://www.tinkercad.com/things/gP7RMdMhAuJ>

**Code**

*// C++ code*

*//*

const int redPin = 3;

const int bluePin = 4;

const int greenPin = 5;

const int speakerPin = 10;

const int pingPin = 7;

void setup() {

*// initialize serial communication:*

  Serial.begin(9600);

  pinMode(redPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  pinMode(speakerPin, OUTPUT);

}

void displayLight(int *code*) {

  digitalWrite(redPin, *code* == 0 ? HIGH : LOW);

  digitalWrite(bluePin, *code* == 1 ? HIGH : LOW);

  digitalWrite(greenPin, *code* == 2 ? HIGH : LOW);

}

void playTone(long *duration*, int *freq*) {

*duration* \*= 1000;

  int period = (1.0 / *freq*) \* 100000;

  long elapsed\_time = 0;

  while (elapsed\_time < *duration*) {

    digitalWrite(speakerPin, HIGH);

    delayMicroseconds(period / 2);

    digitalWrite(speakerPin, LOW);

    delayMicroseconds(period / 2);

    elapsed\_time += (period);

  }

}

void loop() {

  long duration, inches, cm;

  pinMode(pingPin, OUTPUT);

  digitalWrite(pingPin, LOW);

  delayMicroseconds(2);

  digitalWrite(pingPin, HIGH);

  delayMicroseconds(5);

  digitalWrite(pingPin, LOW);

  pinMode(pingPin, INPUT);

  duration = pulseIn(pingPin, 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;

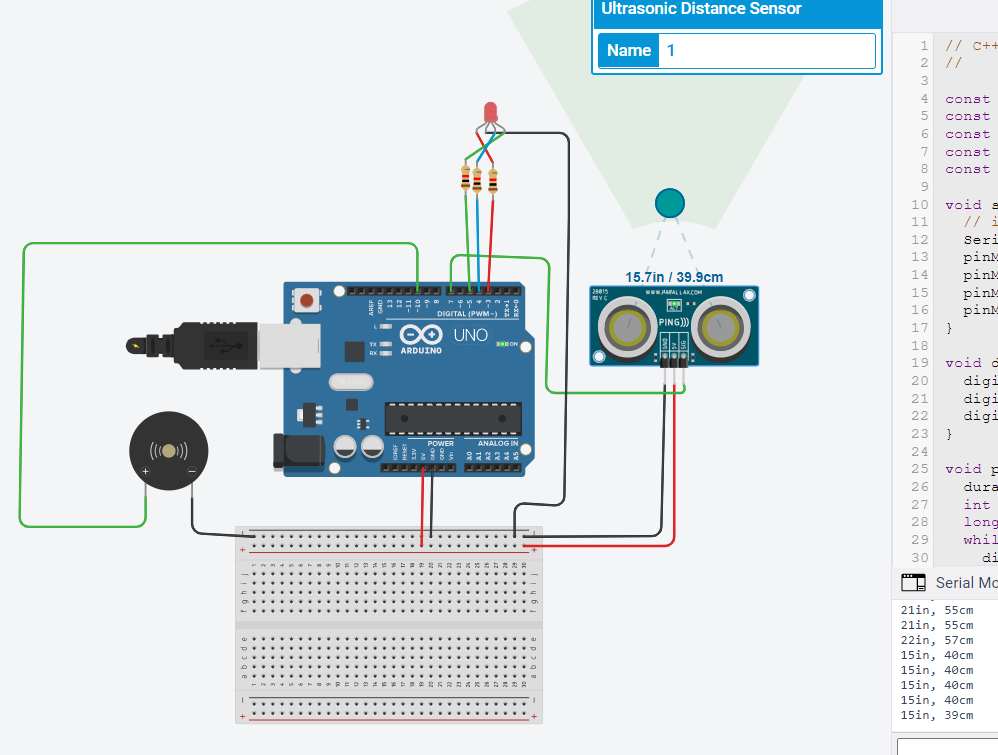
}

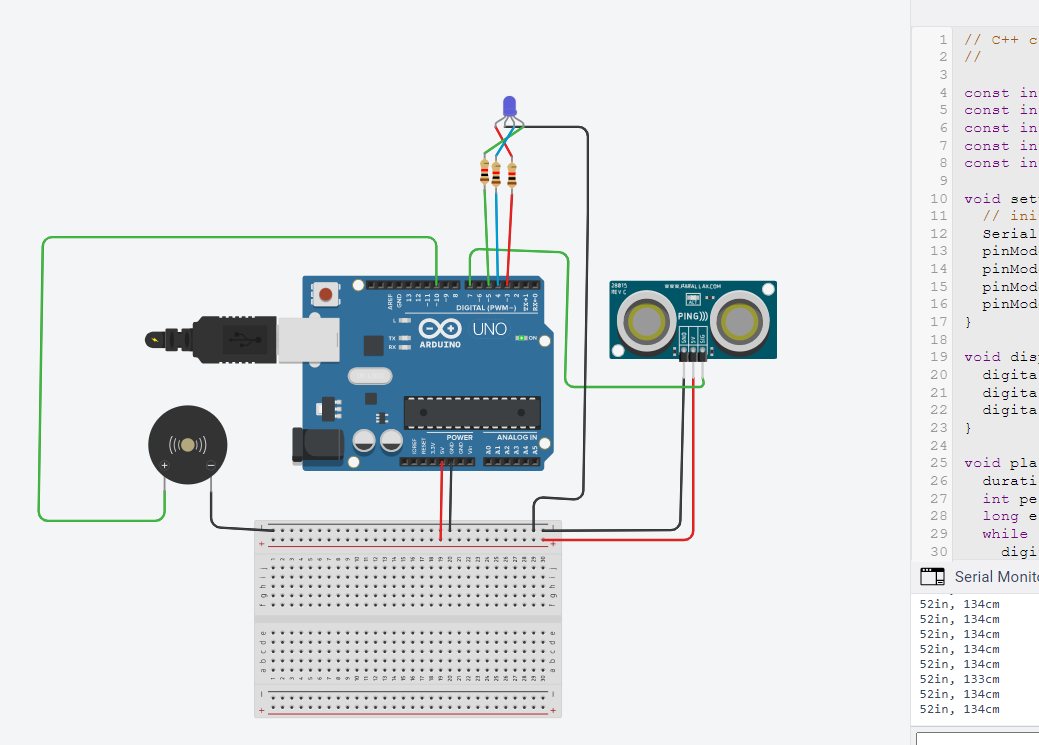
long microsecondsToCentimeters(long *microseconds*) {

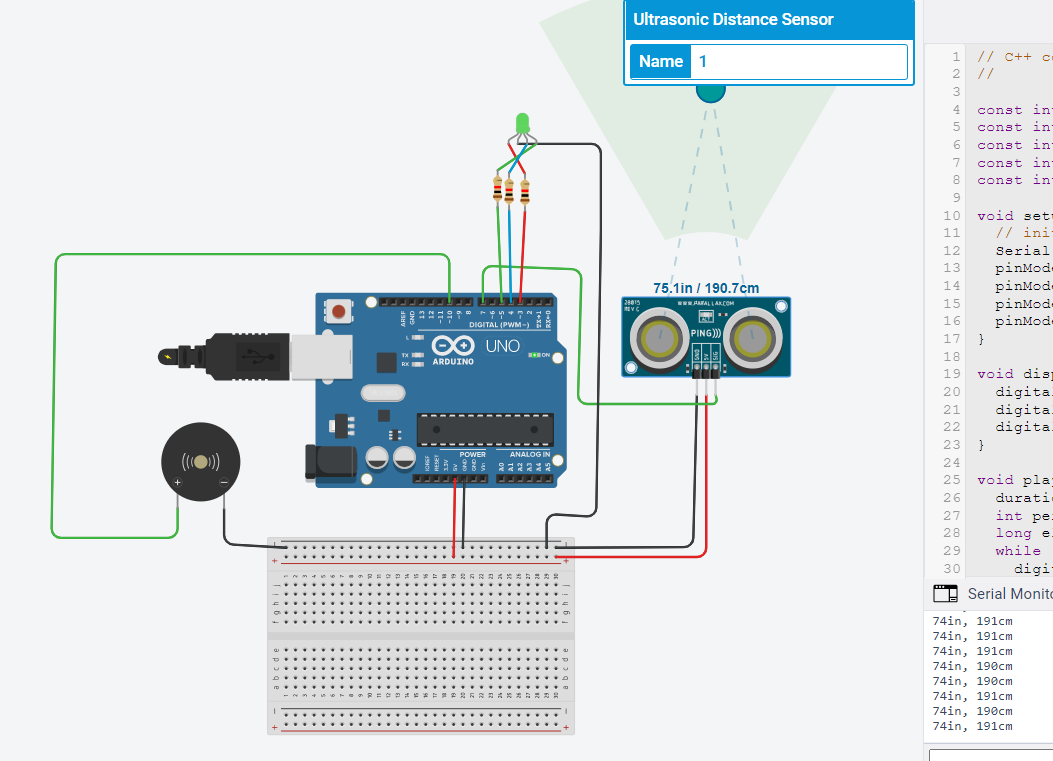
  return *microseconds* / 29 / 2;

}

**Output**

****

****

****

**Experiment 20**

**Aim**

To interface a PIR sensor with Arduino and blink a led if any movement is captured by the sensor and count the number of movements and make a buzzer when movement detected.

**Tinkercad Link**

<https://www.tinkercad.com/things/0Luze9yHsJT>

**Code**

*// C++ code*

*//*

const int pirPin = 3;

const int ledPin = 4;

const int speakerPin = 8;

int totalMovements = 0;

bool isMoving = false;

void setup()

{

  pinMode(ledPin, OUTPUT);

  pinMode(pirPin, INPUT);

  pinMode(speakerPin, OUTPUT);

  Serial.begin(9600);

}

void playTone(long *duration*, int *freq*) {

*duration* \*= 1000;

  int period = (1.0 / *freq*) \* 100000;

  long elapsed\_time = 0;

  while (elapsed\_time < *duration*) {

    digitalWrite(speakerPin, HIGH);

    delayMicroseconds(period / 2);

    digitalWrite(speakerPin, LOW);

    delayMicroseconds(period / 2);

    elapsed\_time += (period);

  }

}

void loop()

{

  int pirStat = digitalRead(pirPin);

  if (pirStat == HIGH) {

    if (!isMoving) {

      playTone(300, 300);

      totalMovements += 1;

      Serial.print("total movements: ");

      Serial.print(totalMovements);

      Serial.print("\n");

      isMoving = true;

    }

    digitalWrite(ledPin, HIGH);

  } else {

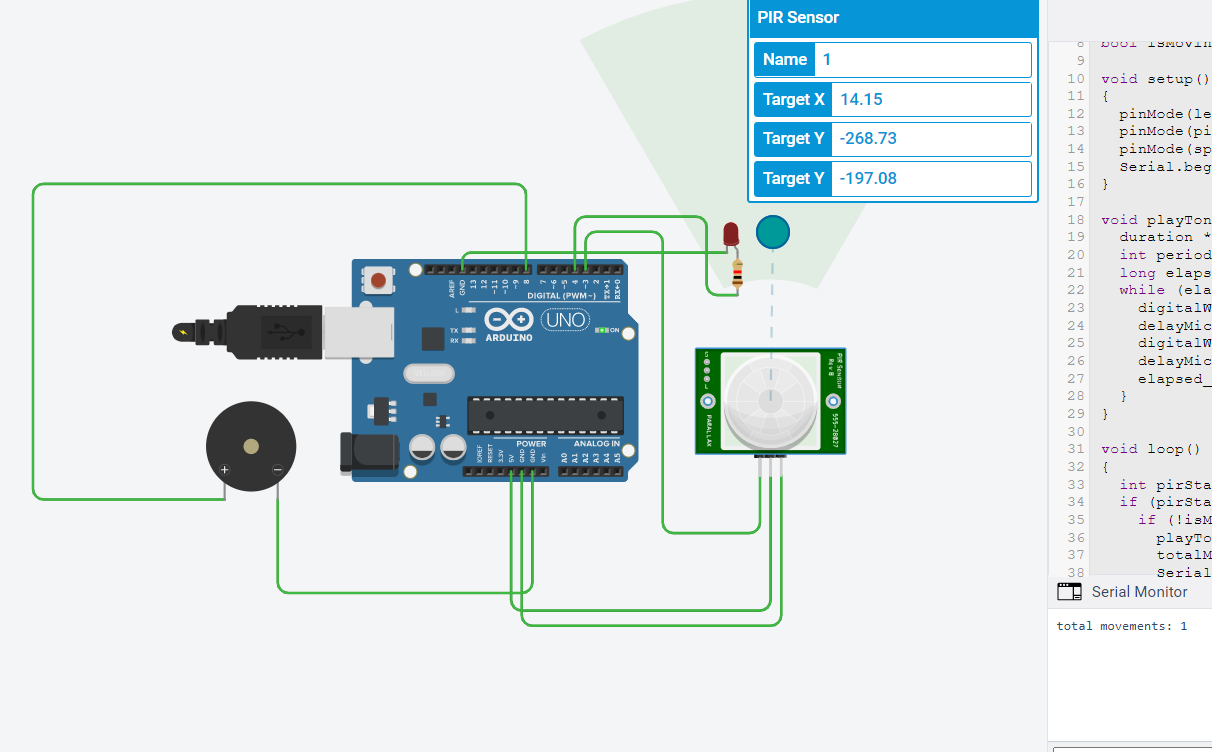
      isMoving = false;

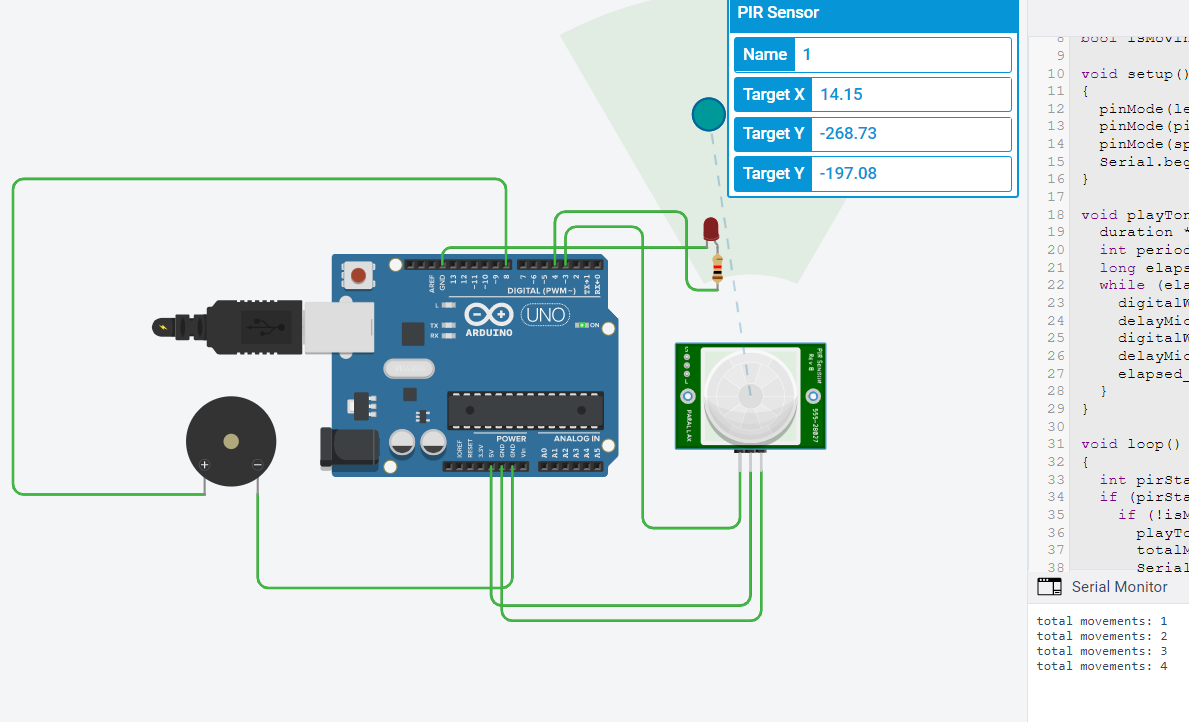
    digitalWrite(ledPin, LOW);

  }

}

**Output**

****

****

**Experiment 21**

1.

**Question**

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.

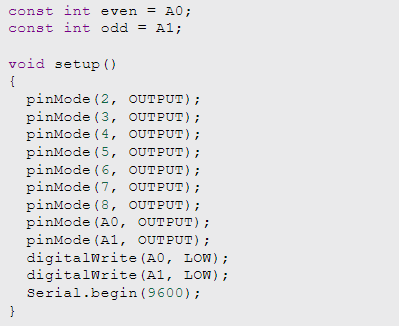
**Aim**

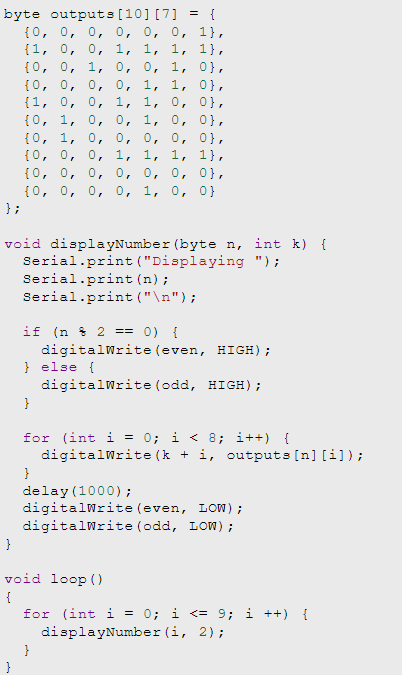
To interface two 7-segment display with Arduino Uno board and to implement odd counter and even counter and display the values in the seven-segment display.

**Tinkercad Link**

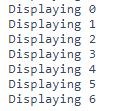
<https://www.tinkercad.com/things/jg8jYolPxuL>

**Code**

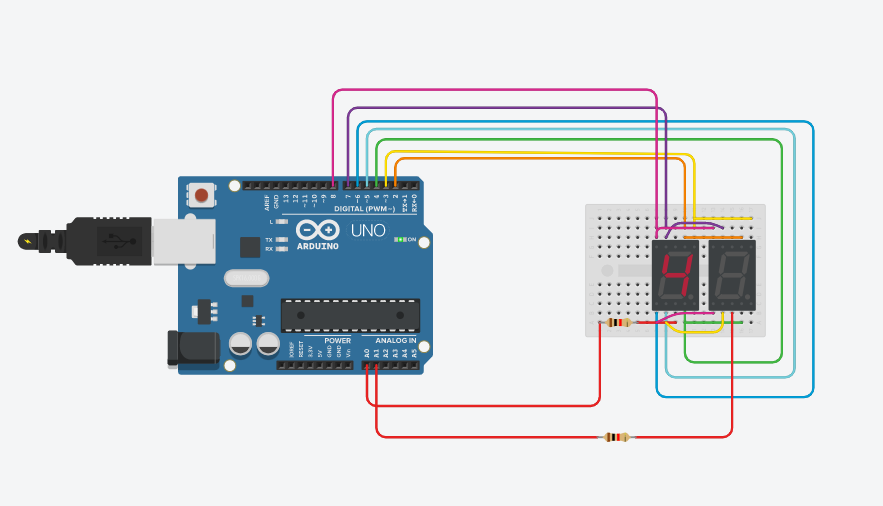
****

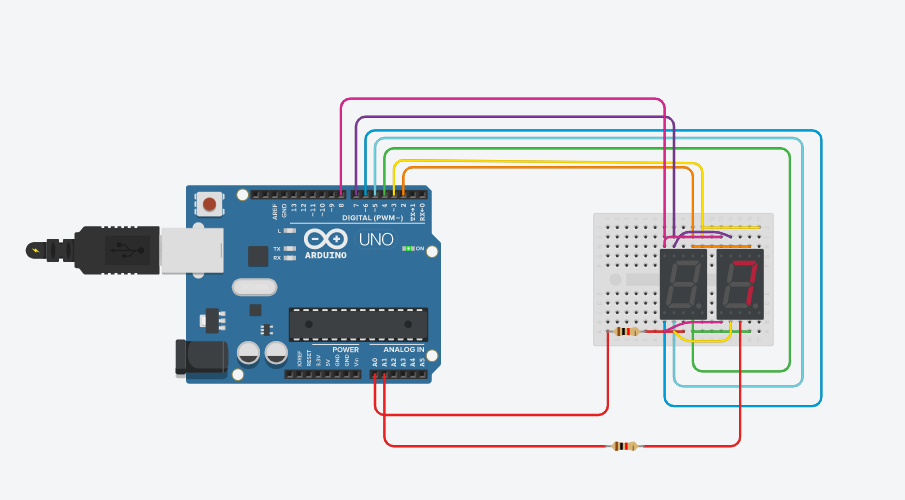
****

**Serial Monitor**

****

**Output**

****

****

**2.**

**Question**

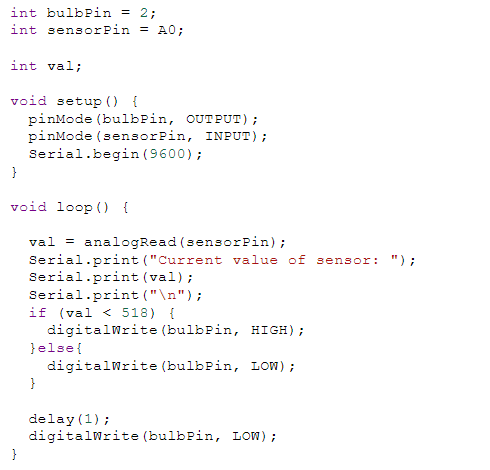
Interface the ambient light sensor with Arduino Uno board. Check the light value from the sensor and switch on/off the bulb (based on the threshold value). **Aim**

To interface the ambient light sensor with Arduino Uno board and to check the light value from the sensor and switch on/off the bulb (based on the threshold value).

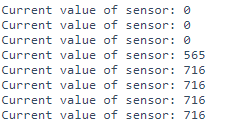
**Tinkercad Link**

<https://www.tinkercad.com/things/8sHMma0csEW>

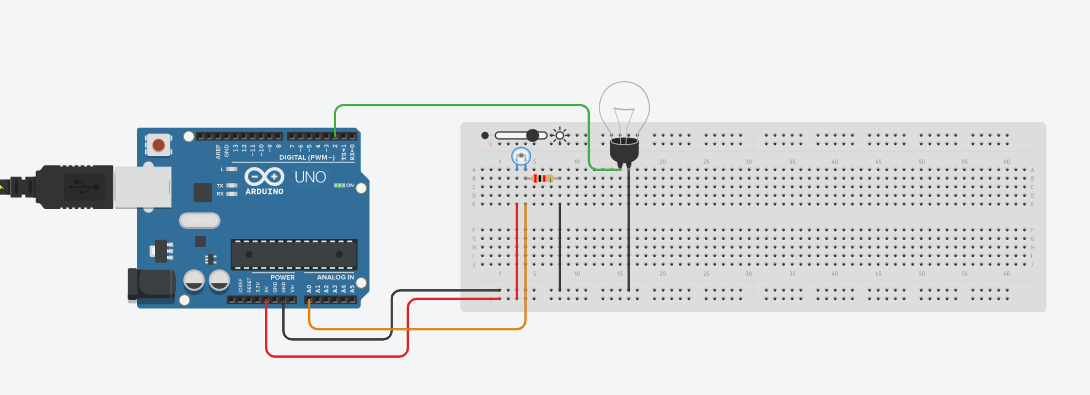
**Code**

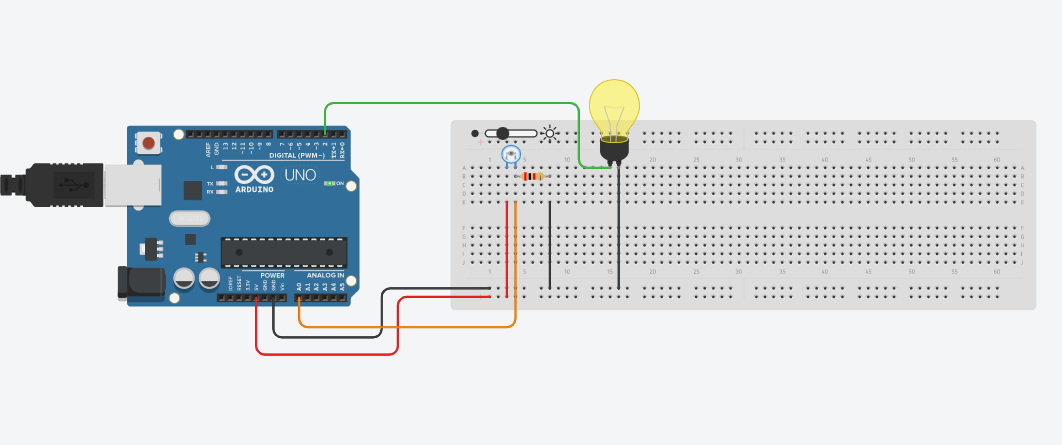
****

**Serial Monitor**

****

**Output**

****

****

3.

**Question**

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.

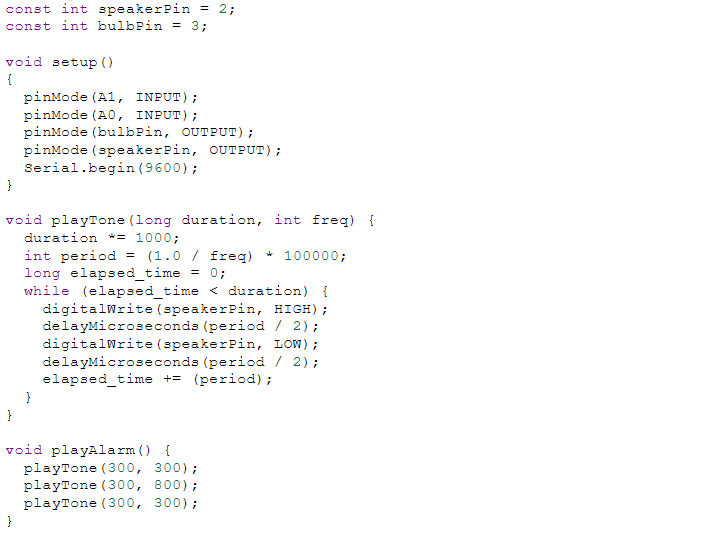
**Aim**

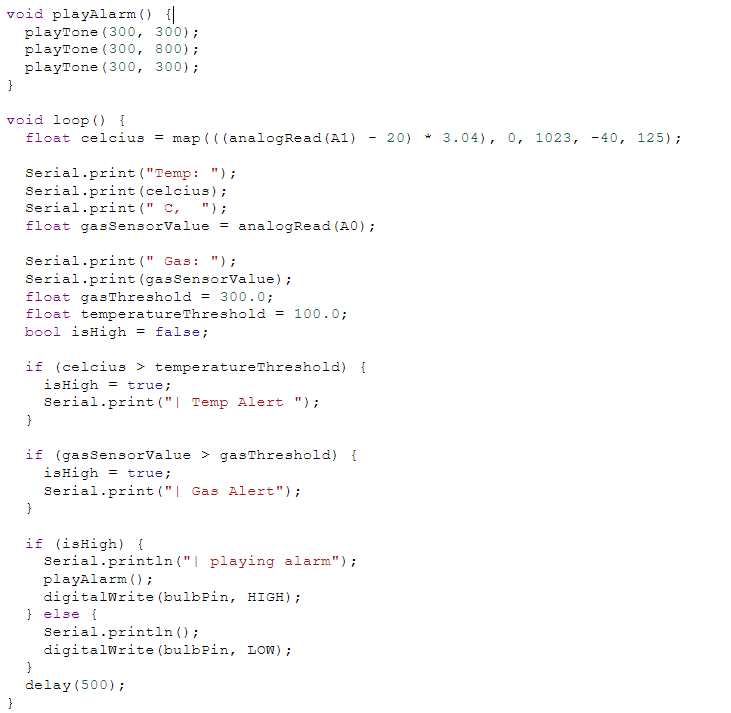
To interface the temperature and gas sensor with Arduino Uno board and check the temperature and the gas value, if the limit is beyond the threshold, switch on the bulb and make alarm using buzzer.

**Tinkercad Link**

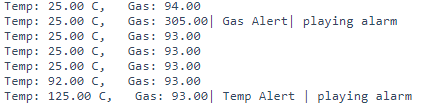
<https://www.tinkercad.com/things/dP0QlrHk0YX>

**Code**

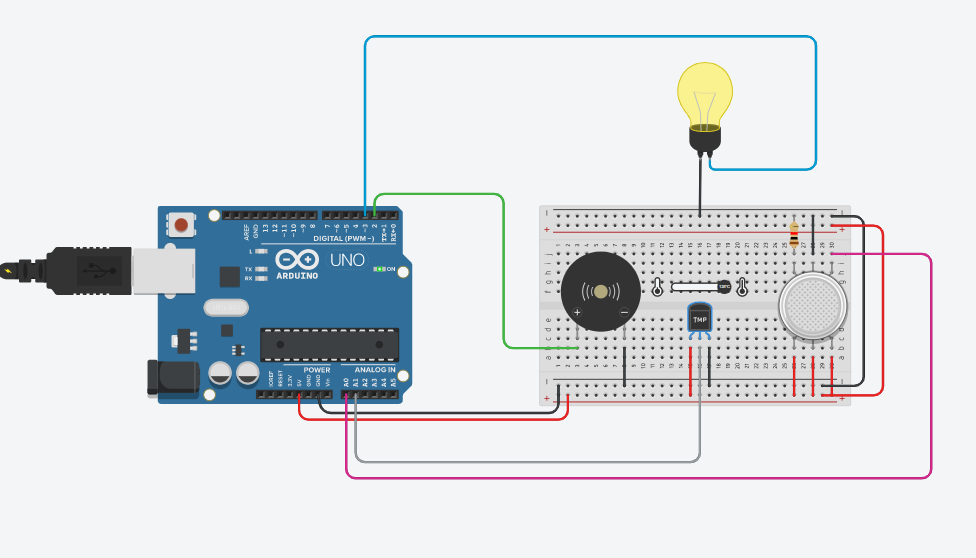
****

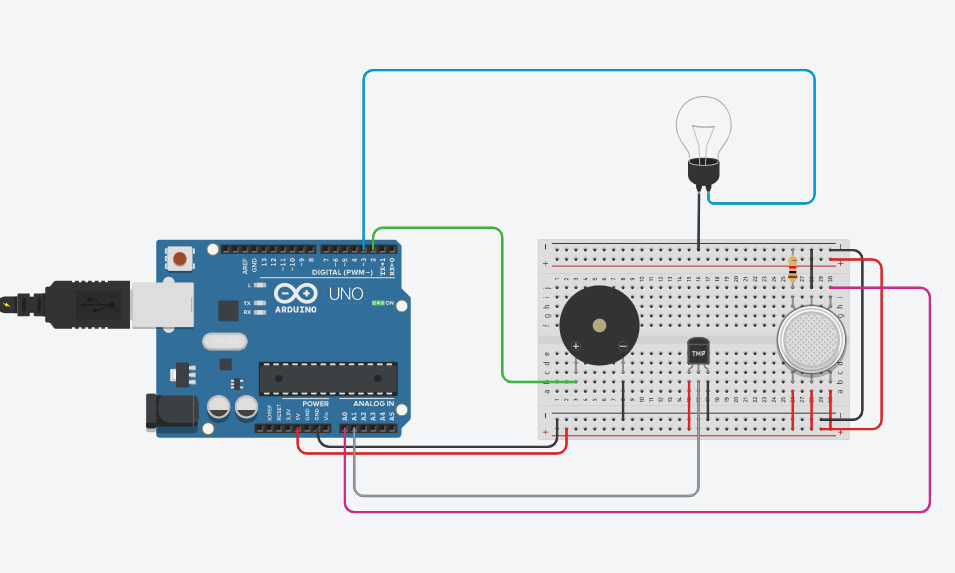
****

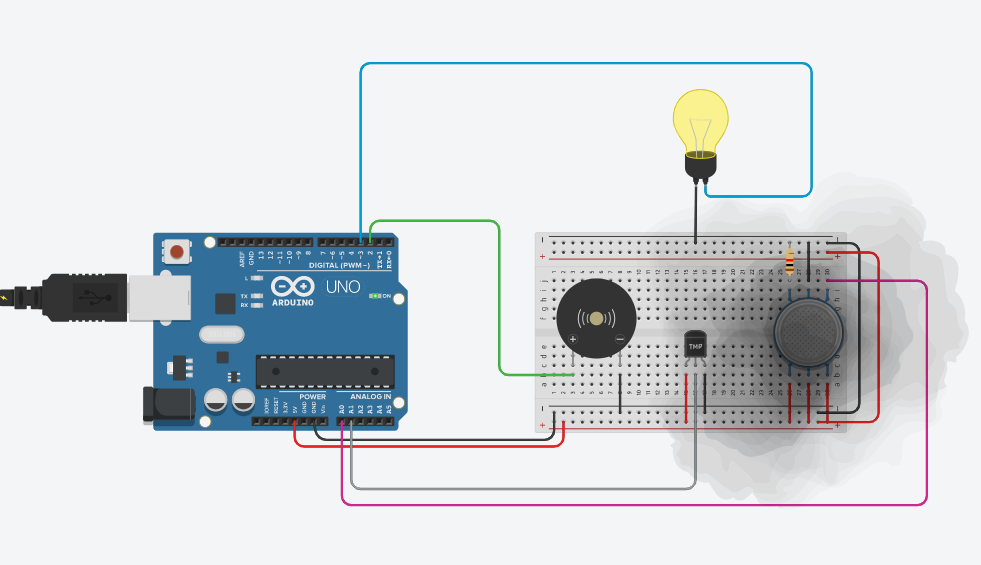
**Serial Monitor**

****

**Output**

****

****

****

106119047 | Hari Rahul V | Embedded Systems Lab Report 9

1.

**Question**

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 some threshold values.

**Aim**

To interface force sensor with Arduino board and display the amount of force given as input to sensor in LCD screen. To interface LED RGB and change color of the LED based on some threshold values.

**Tinkercad Link**

<https://www.tinkercad.com/things/5t3uKnviROj>

**Code**

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int redPin = 8;

const int bluePin = 9;

const int greenPin = 10;

const int forcePin = A0;

const int blueLimit = 125;

const int redLimit = 350;

void blinkLED(int *pinNumber*) {

  digitalWrite(bluePin, LOW);

  digitalWrite(redPin, LOW);

  digitalWrite(greenPin, LOW);

  digitalWrite(*pinNumber*, HIGH);

}

void setup() {

  pinMode(redPin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

  pinMode(forcePin, INPUT);

  lcd.begin(16, 2);

  Serial.begin(9600);

}

void loop() {

  int force = analogRead(forcePin);

  Serial.println(force);

  if (force > redLimit) {

    blinkLED(redPin);

  } else if (force > blueLimit) {

    blinkLED(bluePin);

  } else {

    blinkLED(greenPin);

  }

  lcd.clear();

  lcd.print("Force: ");

  lcd.print(force);

  delay(100);

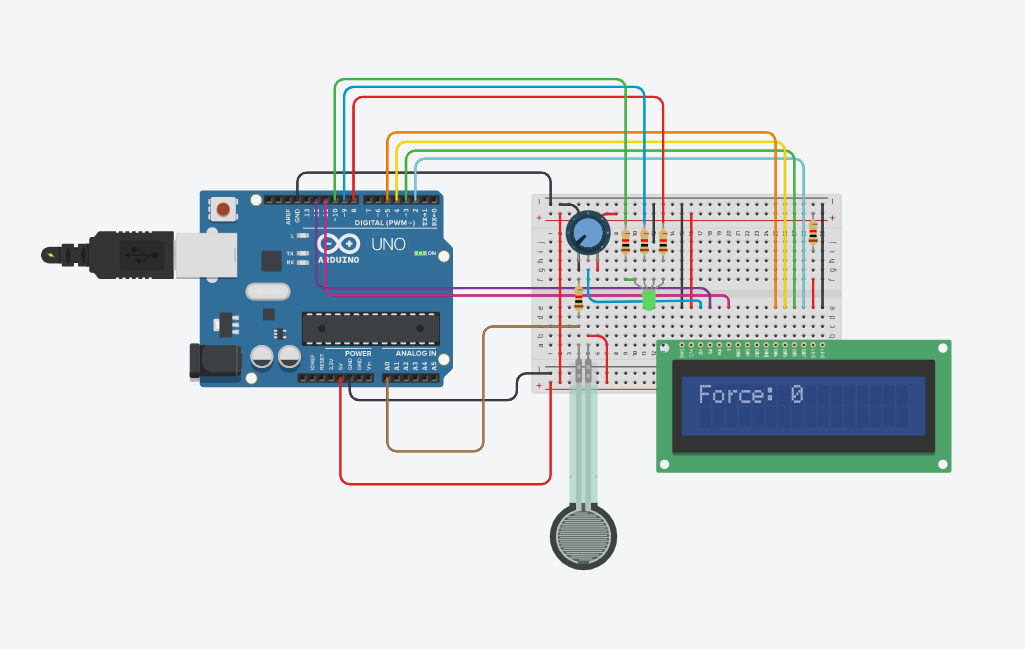
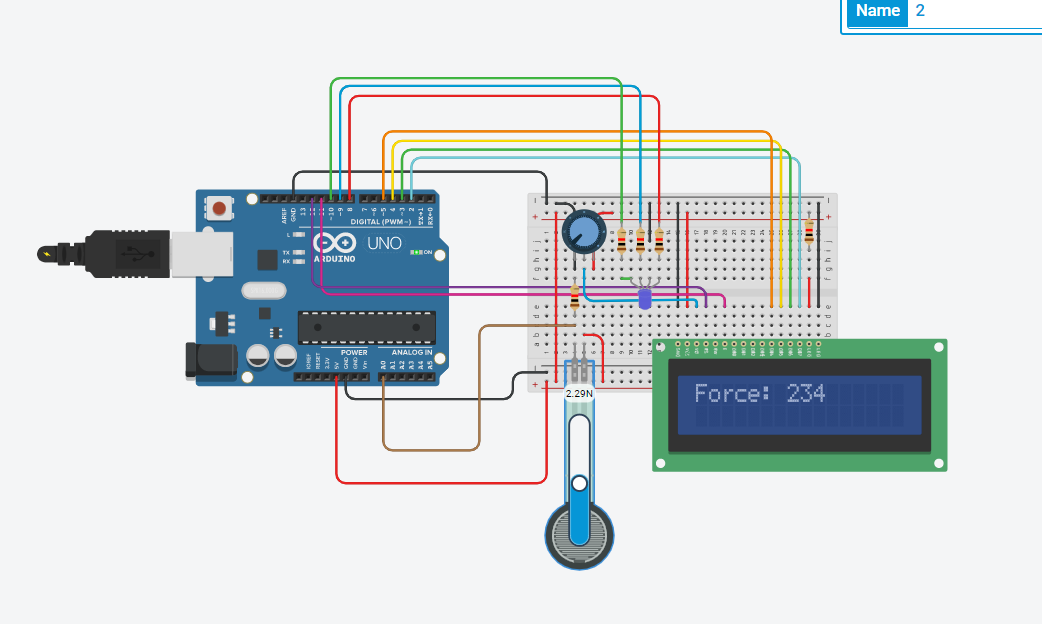
}

**Serial Monitor**

****

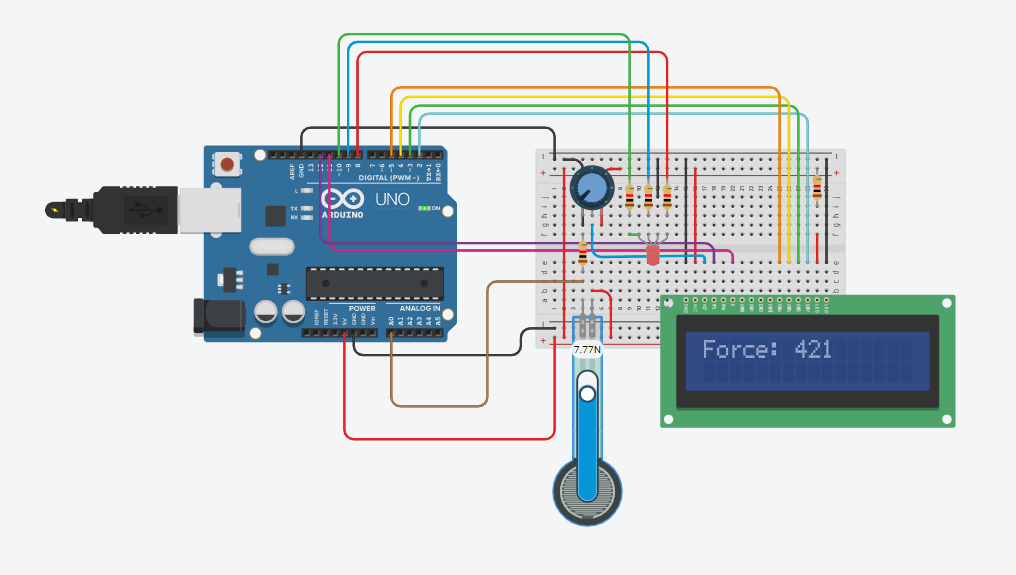
**Output**

i) Low Force

****

ii) Medium Force

iii) High Force

****

**2.**

**Question**

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.

**Aim**

To 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.

**Tinkercad Link**

<https://www.tinkercad.com/things/5MIIrTOqS2g>

**Code**

#include <Keypad.h>

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int numRows = 4;

const int numColumns = 4;

const int tiltPin = 10;

char keys[numRows][numColumns] = {

  {'1','2','3', 'A'},

  {'4','5','6', 'B'},

  {'7','8','9', 'C'},

  {'\*','0','#', 'D'}

};

byte rowPins[numRows] = {9, 8, A5, A4};

byte columnPins[numColumns] = {A3, A2, A1, A0};

Keypad keypad = Keypad( makeKeymap(keys), rowPins, columnPins, numRows, numColumns );

void setup(){

  Serial.begin(9600);

  lcd.begin(16, 2);

  pinMode(tiltPin, INPUT);

}

void loop(){

  char key = keypad.getKey();

  if(key) {

    Serial.println("Key Pressed");

  }

  int tiltValue = digitalRead(tiltPin);

  if (key && tiltValue == 0){

    lcd.setCursor(0, 0);

    lcd.print(key);

    Serial.println(key);

  }

  if (tiltValue == 1) {

    Serial.println("No Tilt");

  } else {

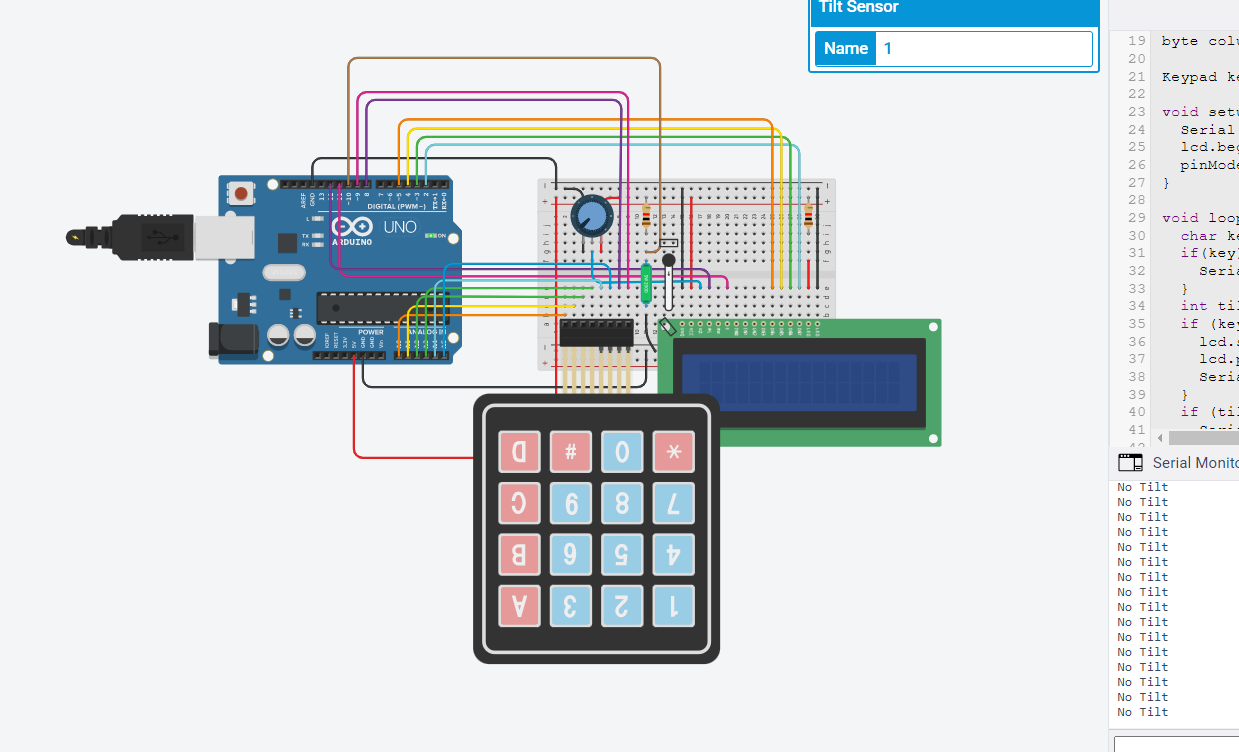
    Serial.println("Tilted");

  }

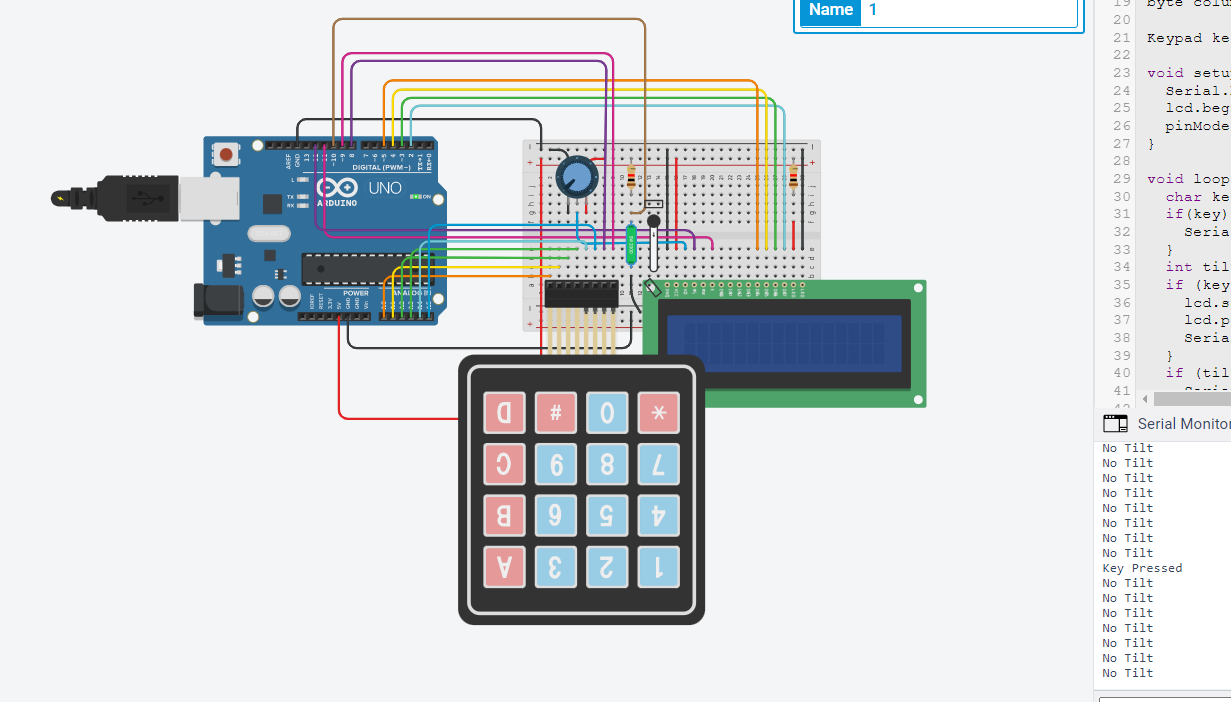
  delay(100);

}

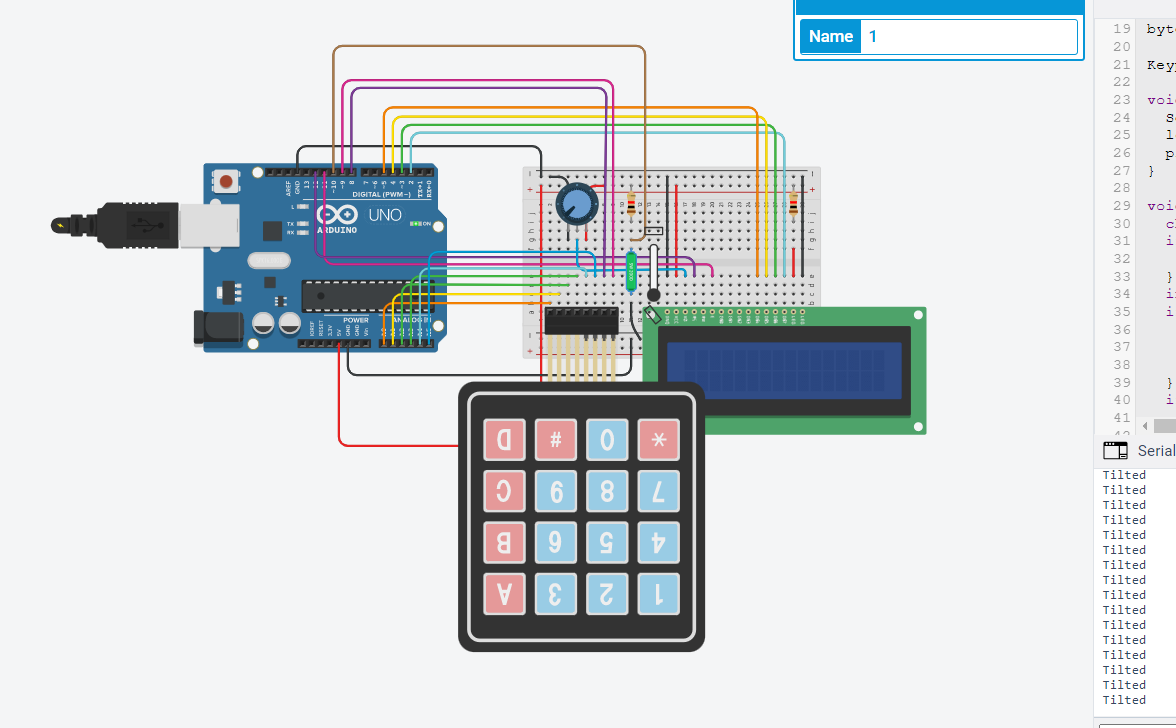
**Output**

i)Key not pressed / No Tilt ****

ii) Key Pressed / No Tilt



iii) Key Not Pressed / Tilt



iv) Key Pressed / Tilt

