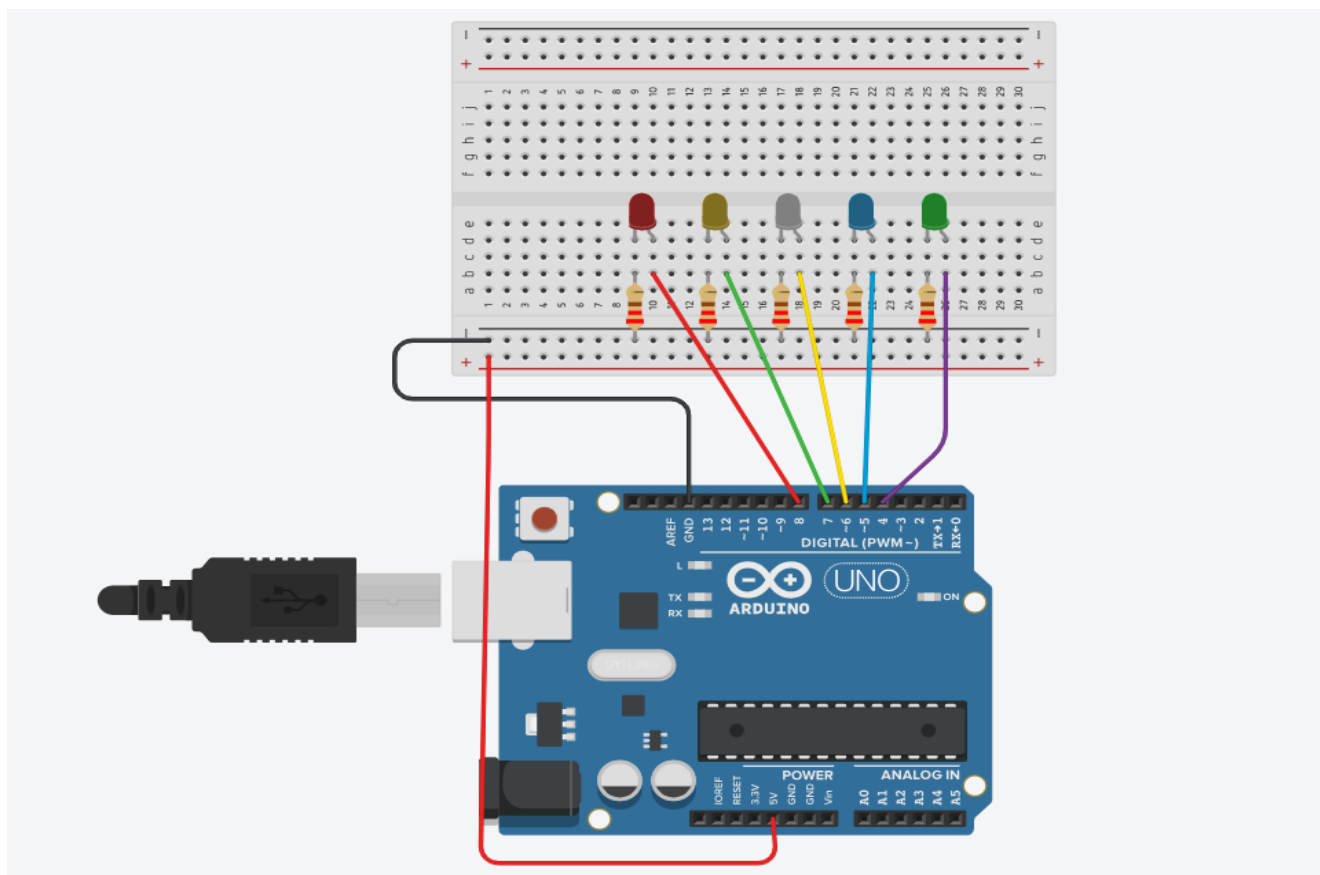


Experiment No: 1

Develop a program to blink 5 LEDs back and forth.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	7
3.	Bread Board	1
4.	LED	5
5.	Resistance (220 Ω)	5

CIRCUIT DIAGRAM:**SET UP:**

- Connect the components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components from search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.

CODE:

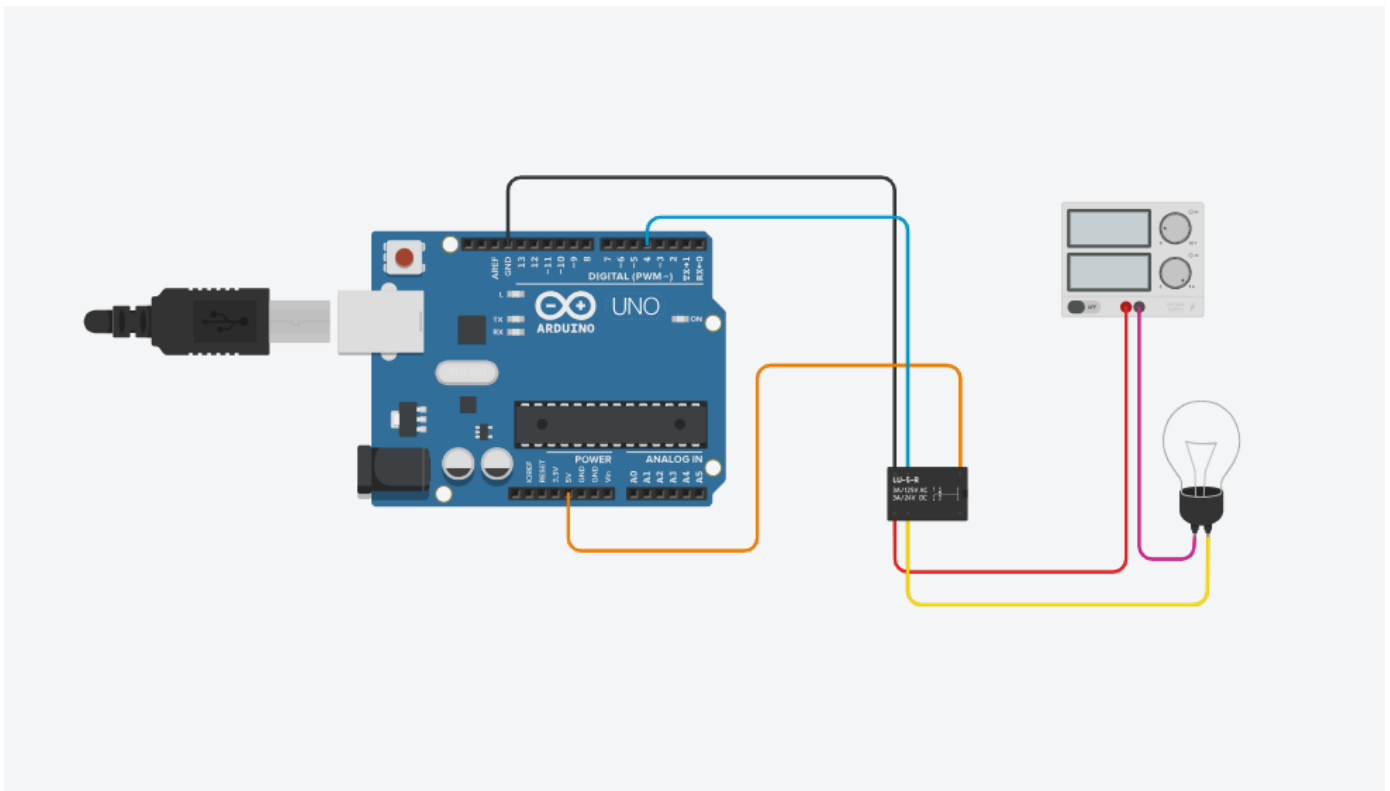
```
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
void loop()
{
  digitalWrite(4, HIGH);
  delay(1000);
  digitalWrite(5, HIGH);
  delay(1000);
  digitalWrite(6, HIGH);
  delay(1000);
  digitalWrite(7, HIGH);
  delay(1000);
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000);
  digitalWrite(7, LOW);
  delay(1000);
  digitalWrite(6, LOW);
  delay(1000);
  digitalWrite(5, LOW);
  delay(1000);
  digitalWrite(4, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Experiment No: 2

Develop a program to interface a relay with Arduino board.

COMPONENT:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	6
3.	Relay SPDT	1
4.	Power Supply	1
5.	Bulb	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.
- To get appropriate output click on Start Simulation tab.

CODE:

```
int relaypin = 4;

void setup()
{
    pinMode(relaypin, OUTPUT);
}

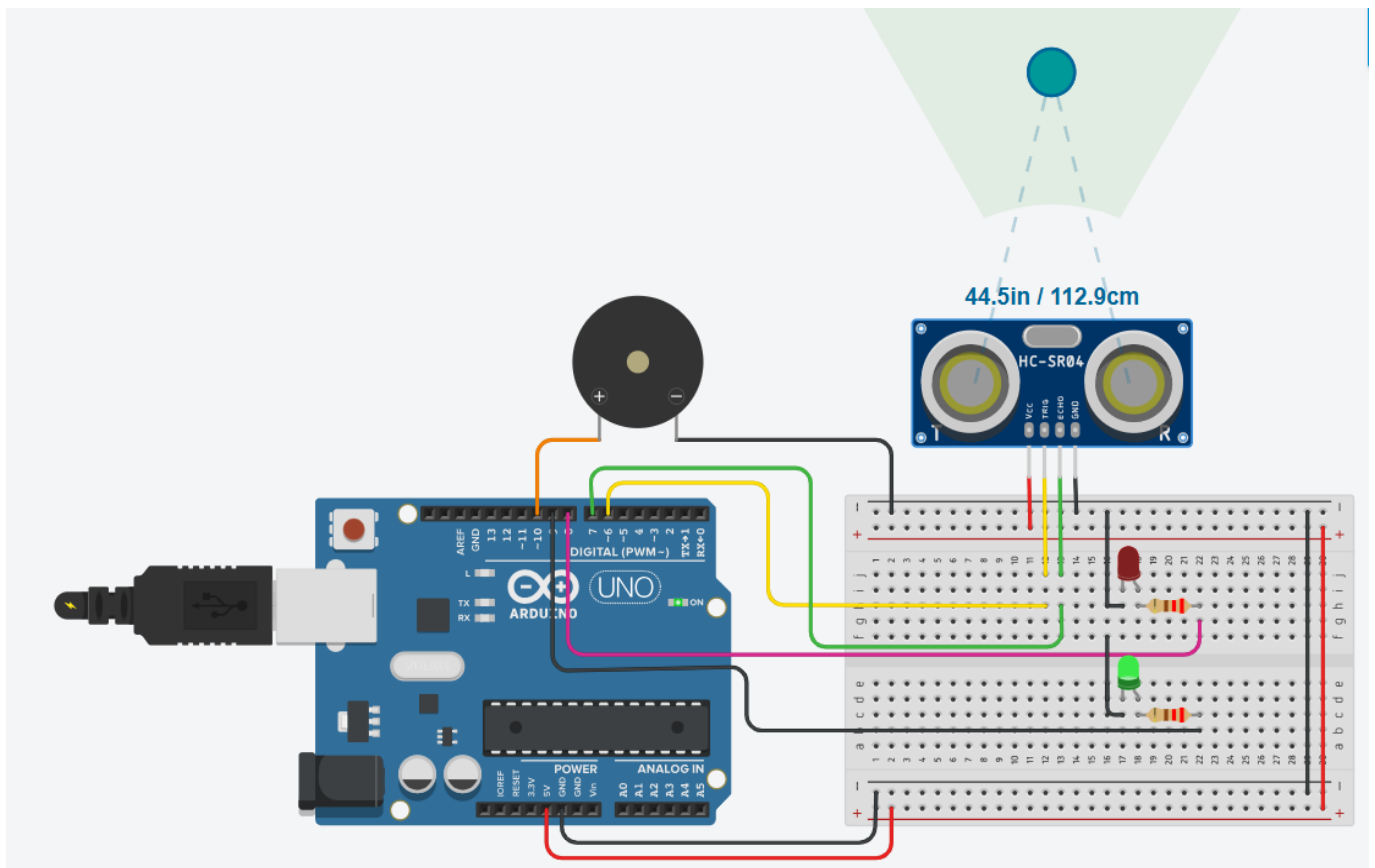
void loop()
{
    digitalWrite(relaypin, LOW);
    delay(1000);
    digitalWrite(relaypin, HIGH);
    delay(1000);
}
```

Experiment No: 3

Develop a program to deploy an intrusion detection system using Ultrasonic and sound sensors.

COMPONENT:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	6
3.	Bread Board	1
4.	LED	2
5.	Resistance (220 Ω)	2
6.	Ultrasonic Distance Sensor (4-pin)	1
7.	Piezo / Buzzer	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.

- f) To get appropriate output click on Start Simulation tab.
- g) Click on the Ultrasonic Distance Sensor to get the desired Output.
- h) Also verify the values in the Serial Monitor Section.

CODE:

```
#define echo 2

#define trig 3

#define outA 8

#define outB 9

#define outC 10

float duration;

float distance;

const int intruderDistance = 100;

void setup() {

  pinMode(trig, OUTPUT);

  pinMode(echo, INPUT);

  pinMode(outA, OUTPUT);

  digitalWrite(outA, LOW);

  pinMode(outB, OUTPUT);

  digitalWrite(outB, LOW);

  pinMode(outC, OUTPUT);

  digitalWrite(outC, LOW);

  Serial.begin(9600)

}

void loop()

{

  time_Measurement();

  distance = (float)duration * (0.0343) / 2;

  Serial.println(distance);

  alarm_condition();

}

void time_Measurement()
```

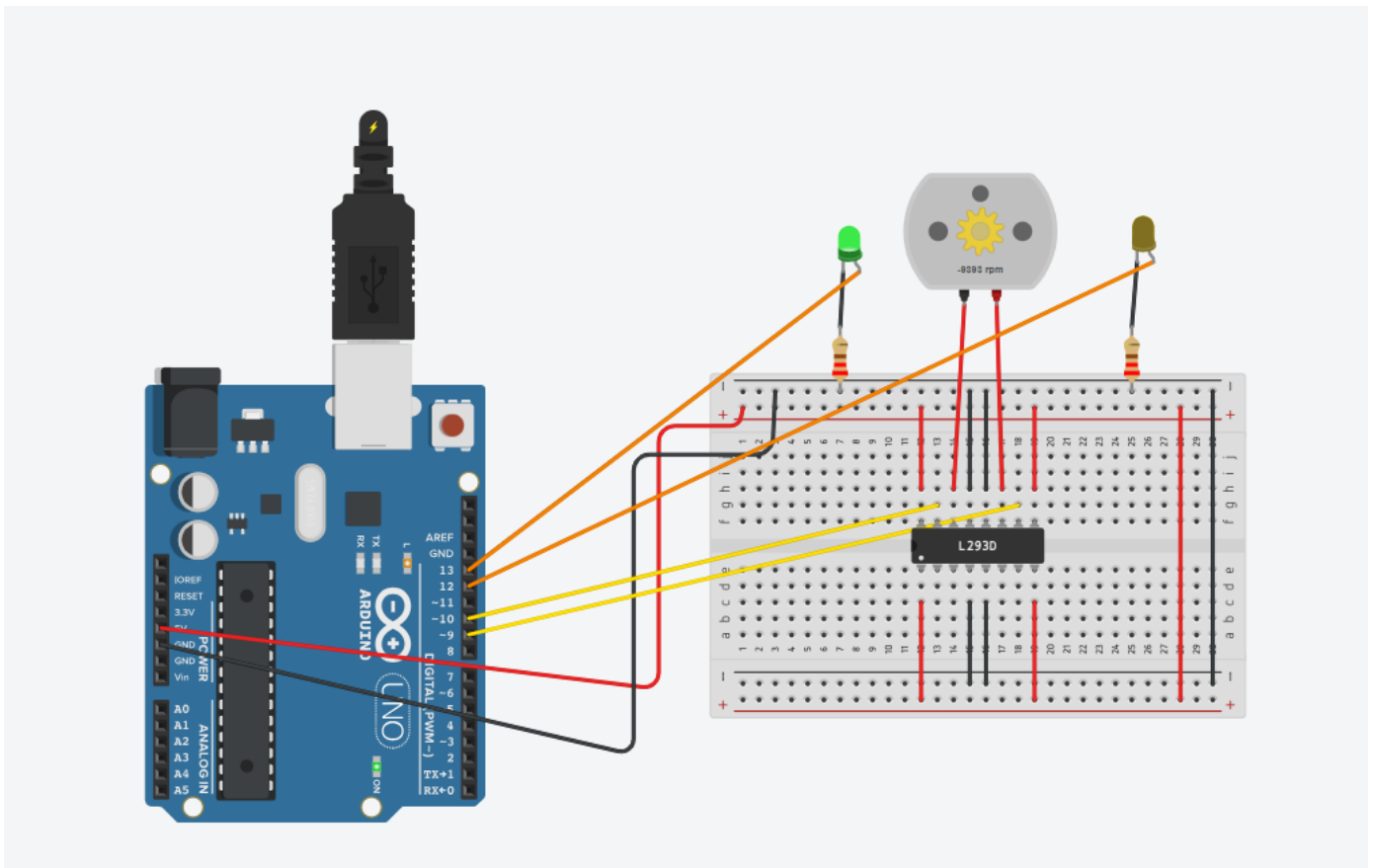
```
{  
    digitalWrite(trig, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trig, LOW);  
    duration = pulseIn(echo, HIGH);}  
void alarm_condition()  
{  
    if(distance<=intruderDistance)  
    {  
        digitalWrite(outA,HIGH);  
        digitalWrite(outB,LOW);  
        analogWrite(outC,100);}  
    else  
    {  
        digitalWrite(outA,LOW);  
        digitalWrite (outB, HIGH);  
        analogWrite (outC,0);  
    }  
}
```

Experiment No: 4

Develop a program to control a DC motor with Arduino board.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	6
3.	Bread Board	1
4.	LED's	2
5.	Resistance (220 Ω)	2
6.	DC Motor	1
7.	L293D Motor Driver	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.
- To get appropriate output click on Start Simulation tab.

CODE:

```
int ledgreen = 13;
int ledyellow = 12;
int inputpin1 = 10;
int inputpin2 = 9;

void setup()
{
    pinMode(10, INPUT);
    pinMode(9, INPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
}

void loop()
{
    digitalWrite(10, HIGH);
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    digitalWrite(12, LOW);
    delay(1000);

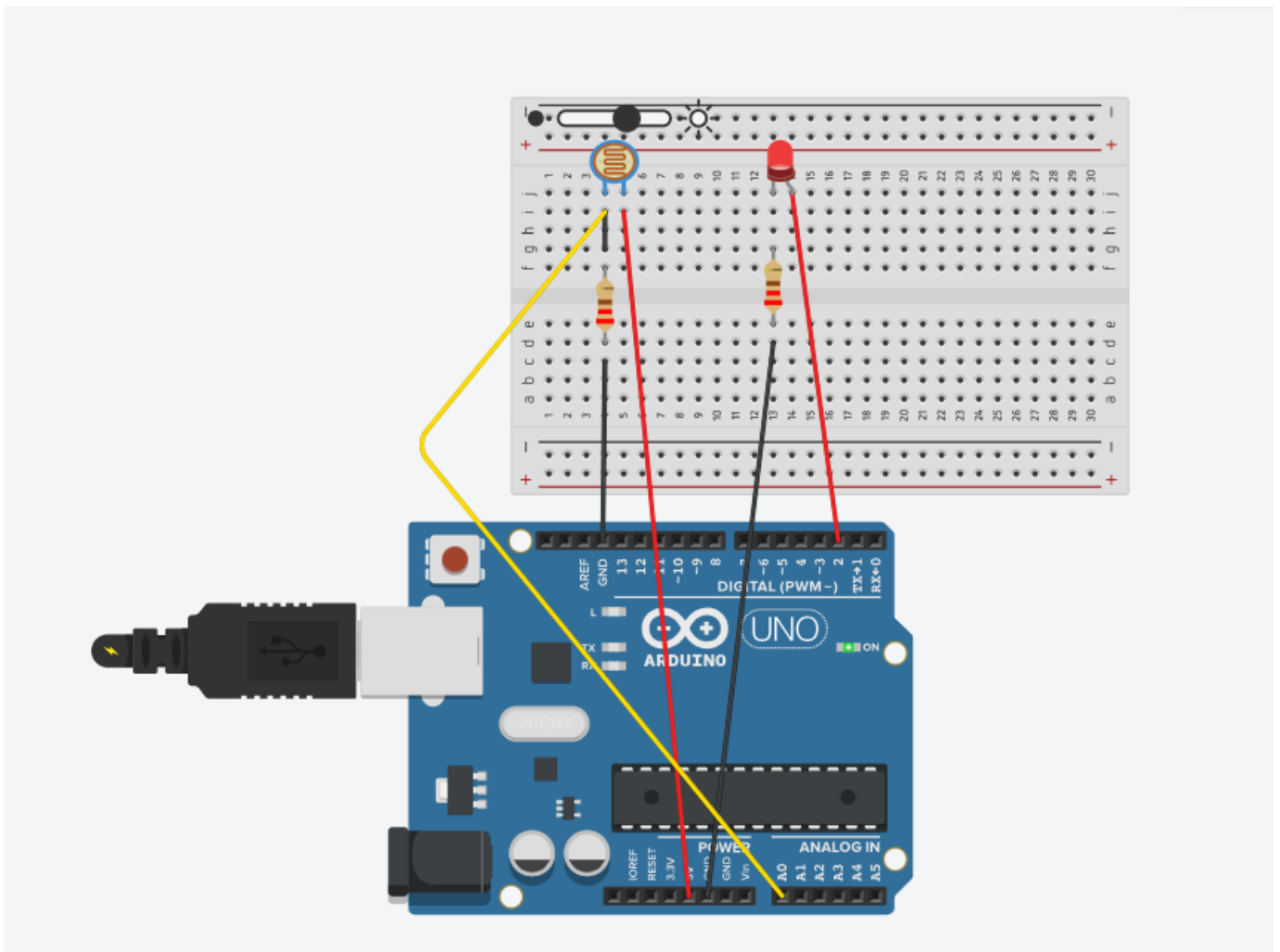
    digitalWrite(9, HIGH);
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(10, LOW);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Experiment No: 5

Develop a program to deploy smart street light system using LDR sensor.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	4
3.	Bread Board	1
4.	LED	1
5.	Resistance (850 Ω)	2
6.	LDR	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the

figure above.

- e) Type the code in code section tab.
- f) To get appropriate output click on Start Simulation tab.
- g) Click on the LDR Sensor to and adjust the values to get the desired output.
- h) Also verify the values in the Serial Monitor Section.

CODE:

```
int ldr = A0;

int led = 2;

int ldrValue = 0;

void setup()
{
    pinMode(led, OUTPUT);
    pinMode(ldr, INPUT);
    Serial.begin(9600);
}

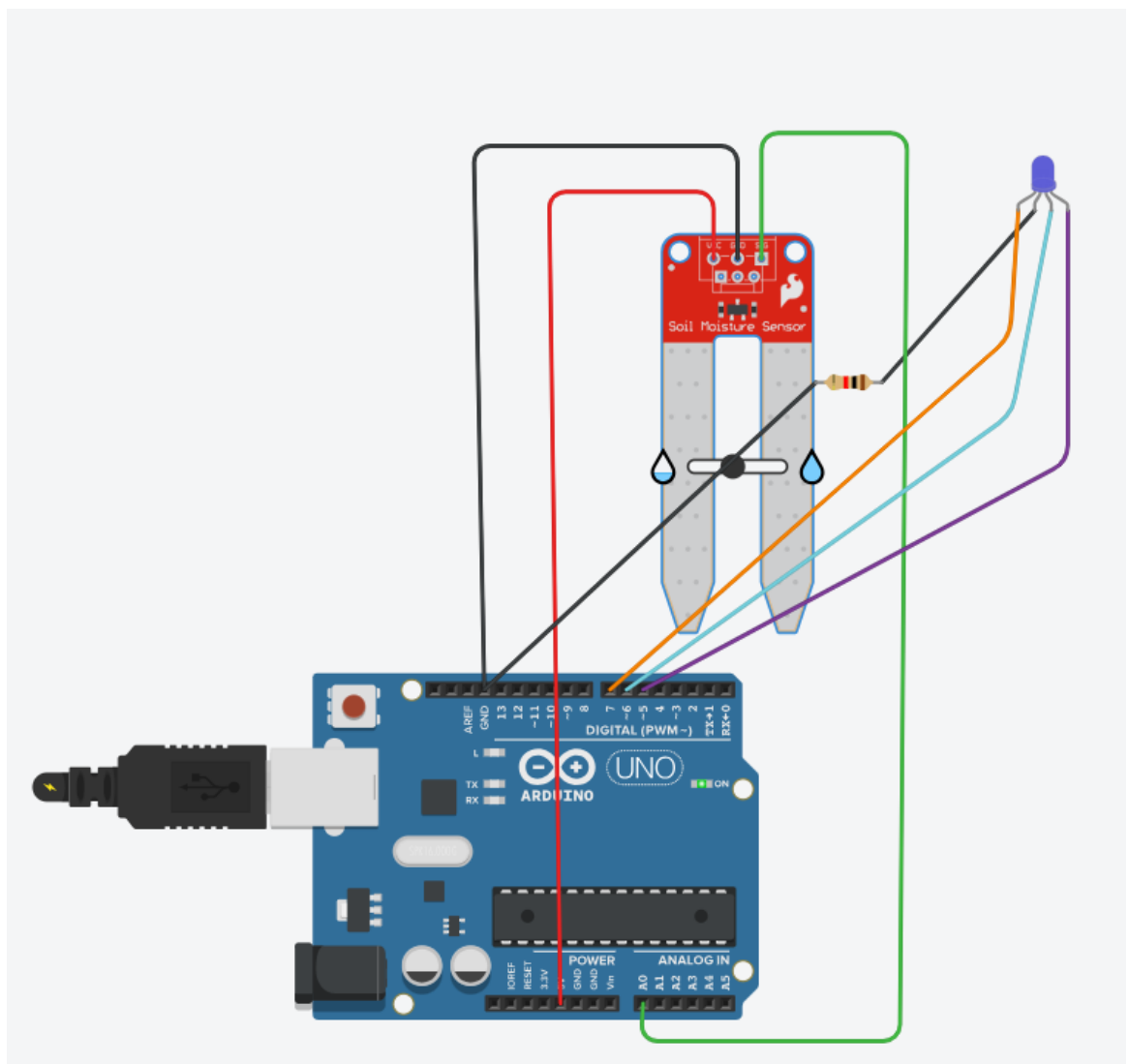
void loop()
{
    ldrValue=analogRead(ldr);
    Serial.println(ldrValue);
    if(ldrValue>=100)
        digitalWrite(led, HIGH);
    else
        digitalWrite(led, LOW);
}
```

Experiment No: 6

Develop a program to classify dry and wet waste with the Moisture sensor (DHT22).

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	8
3.	Bread Board	1
4.	LED	1
5.	Resistance (1K Ω)	1
6.	Moisture sensor (DHT22)	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.

- d) Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- e) Type the code in code section tab.
- f) To get appropriate output click on Start Simulation tab.
- g) Adjust the values by clicking on the Moisture sensor to get the desired output.
- h) Also verify the values in the Serial Monitor Section.

CODE:

```
int moisture_Value=0;

float moisture_percentage=0;

void setup()
{
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  moisture_Value = analogRead(A0);
  moisture_percentage = ((moisture_Value/539.00)*100);
  if ( moisture_percentage>0 && moisture_percentage<25)
  {
    digitalWrite(7,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
  }

  if (moisture_percentage>25 && moisture_percentage<80)
  {
    digitalWrite(7,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(5,LOW);
  }
}
```

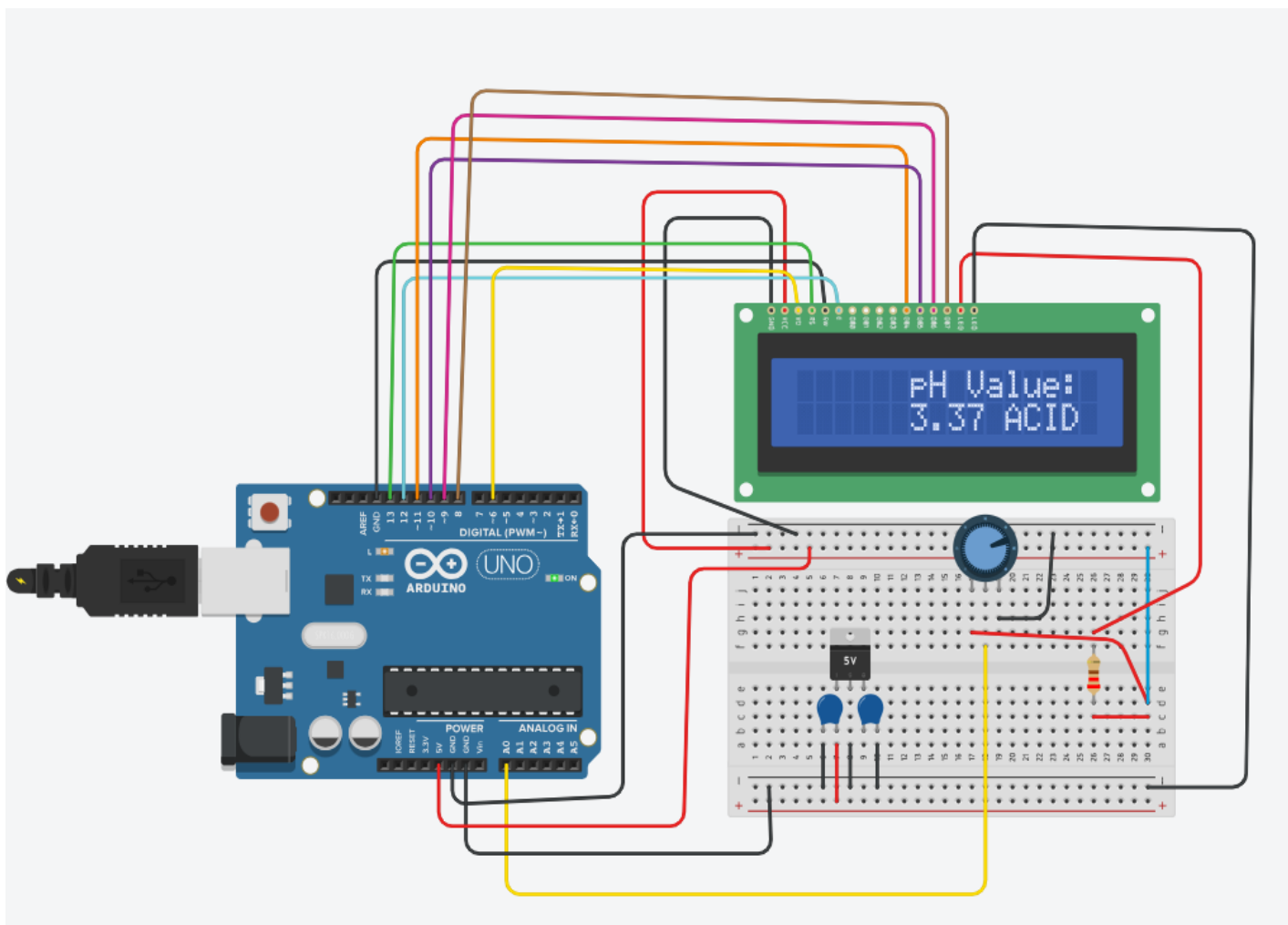
```
    if (moisture_percentage>80 && moisture_percentage<100)
    {
        digitalWrite(7,LOW);
        digitalWrite(6,LOW);
        digitalWrite(5,HIGH);
    }
    Serial.println("Moisture Value : ");
    Serial.println(moisture_percentage);
    delay(1000);
}
```

Experiment No: 7

Develop a program to read the pH value of a various substances like milk, lime and water.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	12
3.	Bread Board	1
4.	Resistance (220 Ω)	1
5.	LCD 16 x 2	1
6.	10 M Ω Potentiometer	1
7.	5V Regulator [LM7805]	1
8.	0.22 uF Capacitor	1
9.	0.1 uF Capacitor	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino

- c) Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- d) Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- e) Type the code in code section tab.
- f) To get appropriate output click on Start Simulation tab.
- g) Click on the Potentiometer and try to adjust the range to get the desired Output on the LCD Display.
- h) Also verify the values in the Serial Monitor Section.

CODE:

```
#include<LiquidCrystal.h>

const int rs =13, en = 12, d4 =11, d5 =10, d6 =9, d7 =8;

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

int Contrast = 0;

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

  analogWrite (6,Contrast);

  lcd.begin(16,2);

  lcd.setCursor(6,0);

  lcd.print("pH Value:");
}

void loop()
{
  int sensorValue = analogRead(A0);

  float ph = sensorValue * (14.0/1023.0);

  Serial.println(ph);

  lcd.setCursor(6,1);

  if (ph>0.0 && ph<5.0)
  {
    lcd.print (ph); lcd.print (" ACID");
  }

  if (ph>5.0 &&ph<7.0)
```



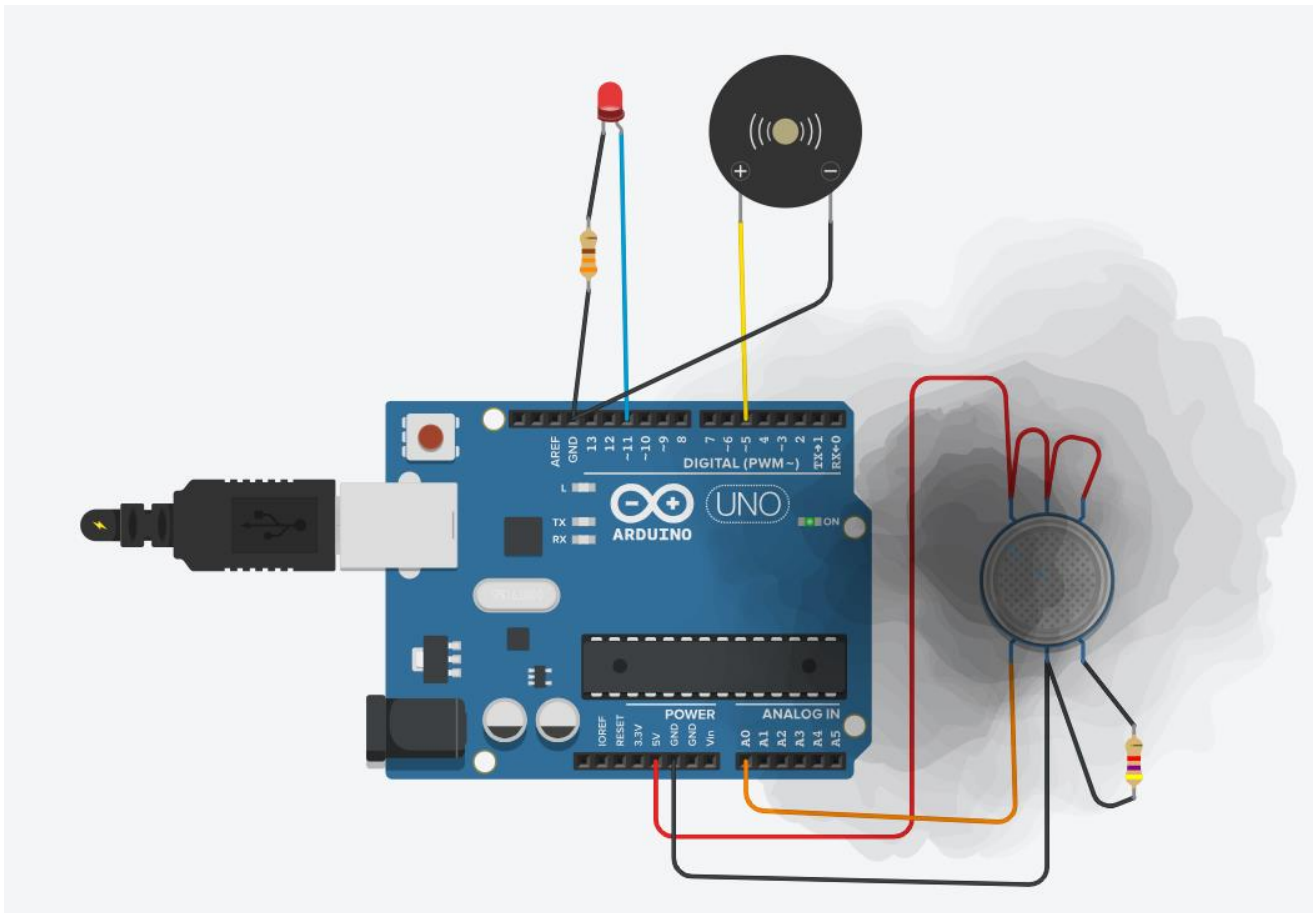
```
{  
  lcd.print (ph);  
  lcd.print (" Normal");  
}  
if (ph>7.0 && ph<14.0)  
  lcd.print (ph);  
{  
  lcd.print (" Base");  
}  
}
```

Experiment No: 8

Develop a program to detect the gas leakage in the surrounding environment.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	6
3.	Gas Sensor	1
4.	MCP23008-based, 32 (0x20) LCD 16 x 2 (I2C)	1
5.	Resistor (4.7 k Ω)	1
6.	Gas Sensor	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.

- f) To get appropriate output click on Start Simulation tab.
- g) Click on the Gas sensor and try to adjust the values to see the appropriate Output
- h) Also verify the values in the Serial Monitor Section.

CODE:

```
int redled = 11;

int gassensor = A0;

int buzzer = 5;

void setup()
{
    pinMode(A0, INPUT);
    pinMode(11, OUTPUT);
    pinMode(5, OUTPUT);
    Serial.begin(9600);
}

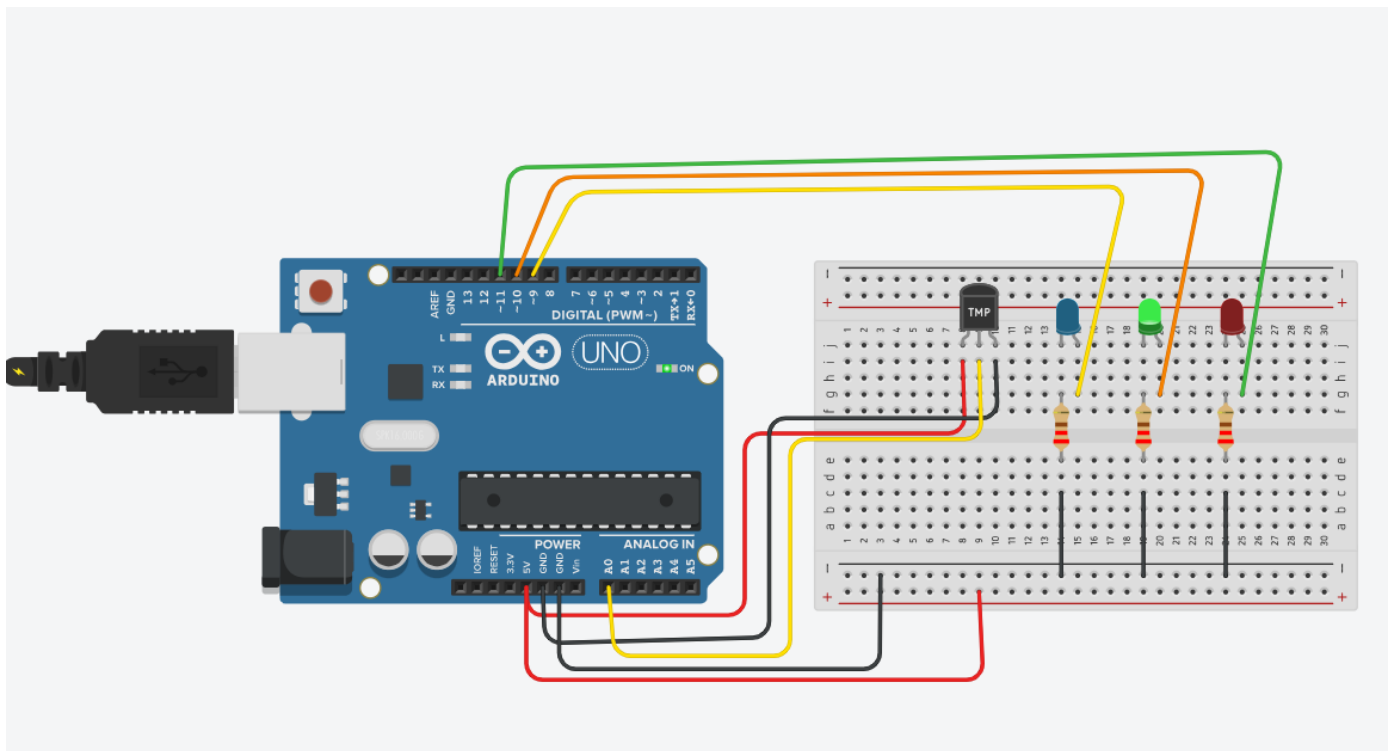
void loop()
{
    gassensor = analogRead(A0);
    Serial.println(gassensor);
    if(gassensor>=100)
    {
        digitalWrite(11,HIGH);
    }
    else    {
        digitalWrite(11,LOW);
        digitalWrite(5,LOW);    }
    if(gassensor>=150)
    {
        digitalWrite(5, HIGH);
        digitalWrite(11, HIGH);
    }
    delay(100);
}
```

Experiment No: 9

Develop a program to demonstrate weather station readings using Arduino.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	2
2.	Cable	11
3.	Resistors (220 Ω)	3
4.	LED's	3
5.	Temperature Sensor (TMP36)	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.
- To get appropriate output click on Start Simulation tab.
- Click on the Temperature sensor and try to adjust the values to see the appropriate Output on the Led's.
- Also verify the values in the Serial Monitor Section.

CODE:

```
const int tmp = A0;

const int blue=9;

const int green=10;

const int red=11;

float volt, heat;

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

    pinMode(blue,OUTPUT);

    pinMode(green,OUTPUT);

    pinMode(red,OUTPUT);
}

void loop()
{
    int tmpValue = analogRead(tmp);

    volt = tmpValue * (5.0 / 1023.0);

    heat = (volt - 0.5) * 100.0;

    if(heat<0)
    {
        digitalWrite(blue,HIGH);

        digitalWrite(green,LOW);

        digitalWrite(red,LOW);

        Serial.println(String(heat) + "C Cold Temperature");
    }

    else if(heat>=0 && heat<=42){

        digitalWrite(green,HIGH);

        digitalWrite(blue,LOW);

        digitalWrite(red,LOW);

        Serial.println(String(heat) + "C Normal Temperature");
    }
}
```

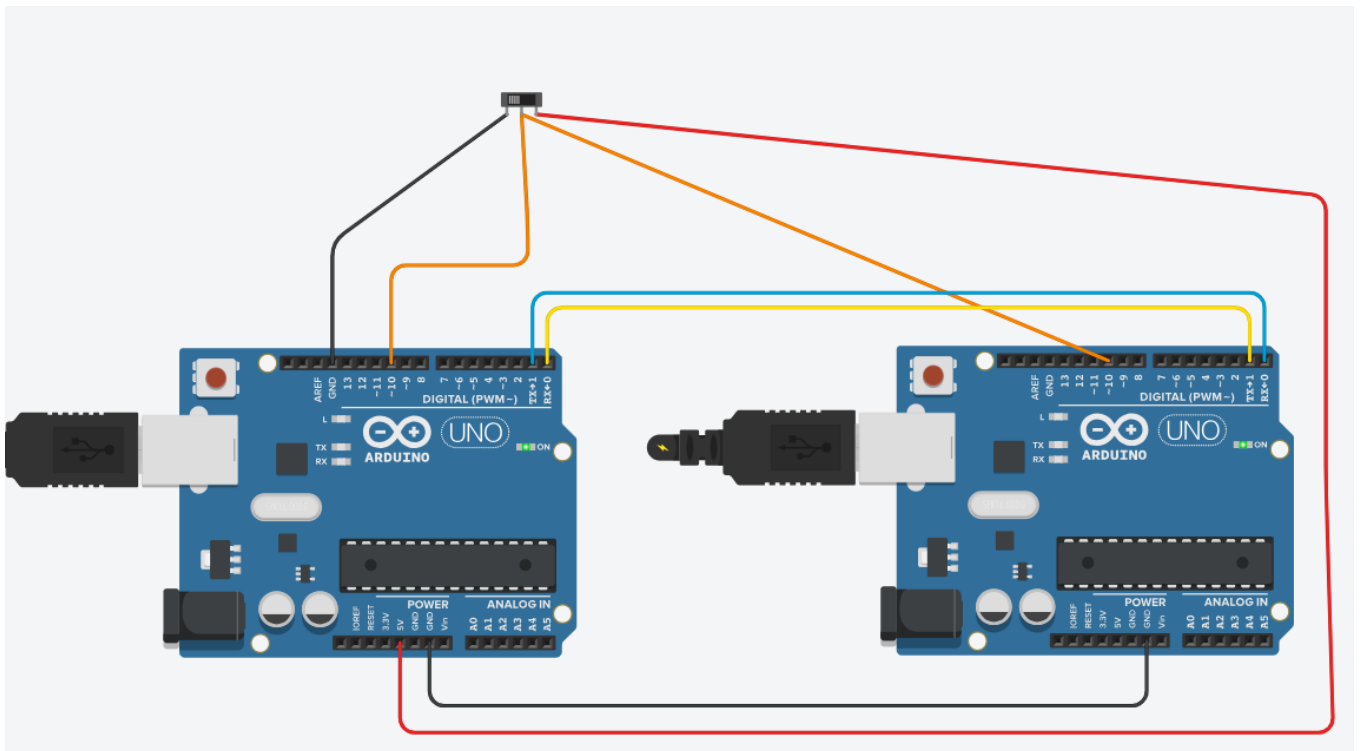
```
else
{
    digitalWrite(red,HIGH);
    digitalWrite(blue,LOW);
    digitalWrite(green,LOW);
    Serial.println(String(heat) + "C Hot Temperature");
}
delay(500);
}
```

Experiment No: 10

Develop a program to setup a UART protocol and pass a string through the protocol.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	2
2.	Cable	7
3.	Slideswitch	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.
- To get appropriate output click on Start Simulation tab.
- With the help of switch, slide the switch button to see the message sent from one Arduino UNO 1 board to another Arduino board UNO 2.
- Also verify the values in the Serial Monitor Section.

CODE:

```
const int MAX_LEN = 100;

char send[MAX_LEN] = "Hello\n";

char recieve[MAX_LEN];

int switch1 = 1;

int switch_pin = 10;

void setup()
{
  Serial.begin(9600);
  pinMode(switch_pin, INPUT);
}

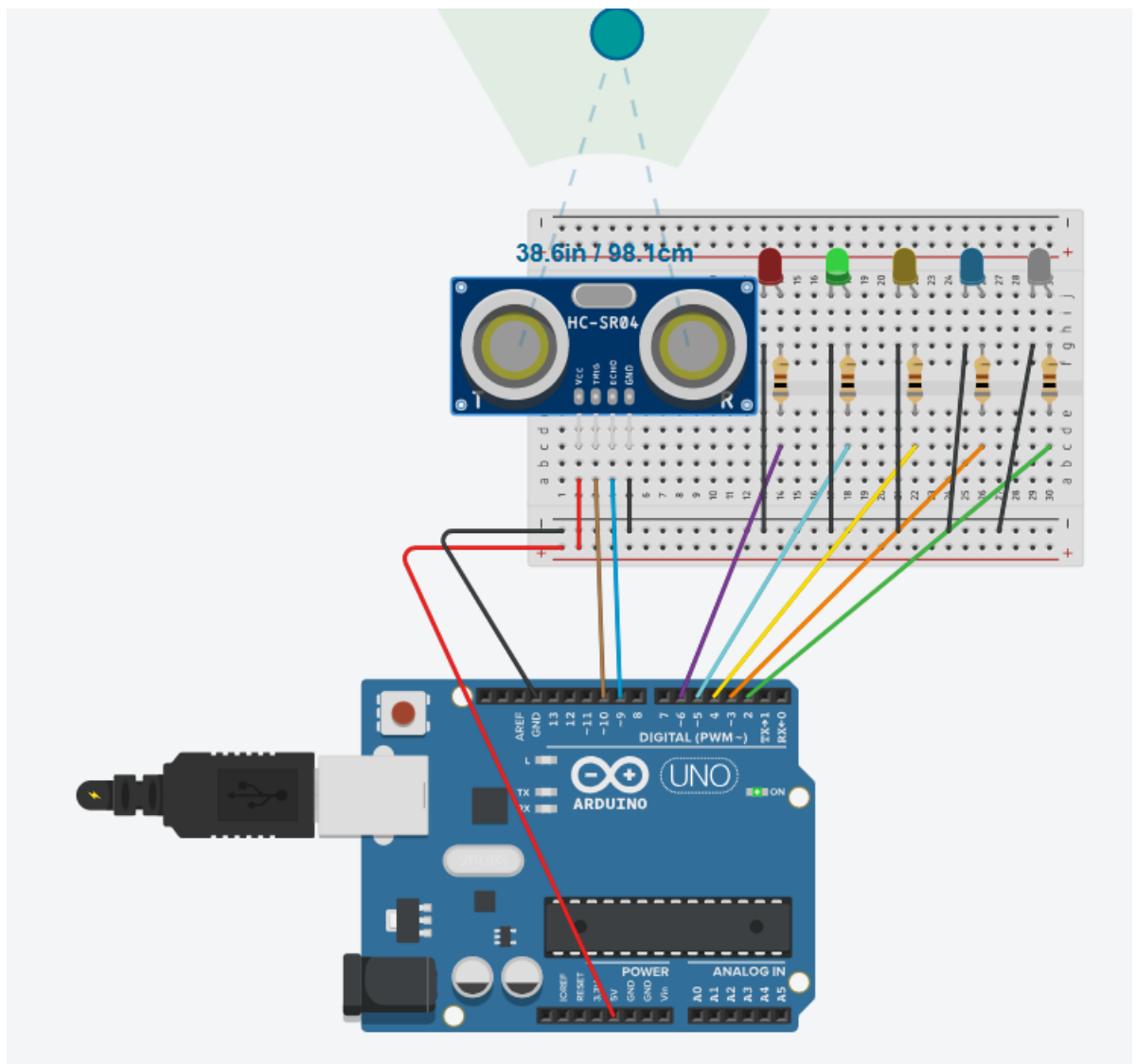
void loop() {
  switch1 = digitalRead(switch_pin);
  if (!switch1)
  {
    int len = Serial.parseInt();
    char garbage = Serial.read();
    Serial.readBytes(recieve, len);
    Serial.print("Length = ");
    Serial.println(len);
    Serial.print("Message = ");
    Serial.println(recieve);
    delay(1000); }
  else {
    int len = strlen(send);
    Serial.println(len);
    Serial.write(send, len);
    delay(1000); }
}
```


Experiment No: 11

Develop a water level depth detection system using Ultrasonic sensor.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	17
3.	Bread Board	1
4.	LED	5
5.	Resistance (800 Ω)	5
6.	Ultrasonic Distance Sensor (4-pin)	1

CIRCUIT DIAGRAM:

SET UP:

- a) Connect the Components as per circuit.
- b) Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- c) Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- d) Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- e) Type the code in code section tab.
- f) To get appropriate output click on Start Simulation tab.
- g) Click on the Ultrasonic Distance Sensor to adjust the distance and observe the different led's Glow.
- h) Also verify the values in the Serial Monitor Section.

CODE:

```
int led1 =6;
int led2 =5;
int led3 =4;
int led4 =3;
int led5 =2;
int trigPin =10;
int echoPin =9;
int ival;
void setup()
{
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9, INPUT);
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delay(10);
  digitalWrite(trigPin, HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);
  ival=pulseIn(echoPin, HIGH);
  int dist=(ival/2)/29.154;
  Serial.print("dist");
  Serial.println(dist);
  if(dist<=50)
```

```
{
    digitalWrite(6,HIGH);
    digitalWrite(5,LOW);
    digitalWrite(4,LOW);
    digitalWrite(3,LOW);
    digitalWrite(2,LOW);
}
else if(dist<=100)
{
    digitalWrite(6,LOW);
    digitalWrite(5,HIGH);
    digitalWrite(4,LOW);
    digitalWrite(3,LOW);
    digitalWrite(2,LOW);
}
else if(dist<=150)
{
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(4,HIGH);
    digitalWrite(3,LOW);
    digitalWrite(2,LOW);
}

else if(dist<=200)
{
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(4,LOW);
    digitalWrite(3,HIGH);
    digitalWrite(2,LOW);
}

else if(dist<=250)
{
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(4,LOW);
    digitalWrite(3,LOW);
    digitalWrite(2,HIGH);
}
else
{
    digitalWrite(6,HIGH);
    digitalWrite(5,HIGH);
```

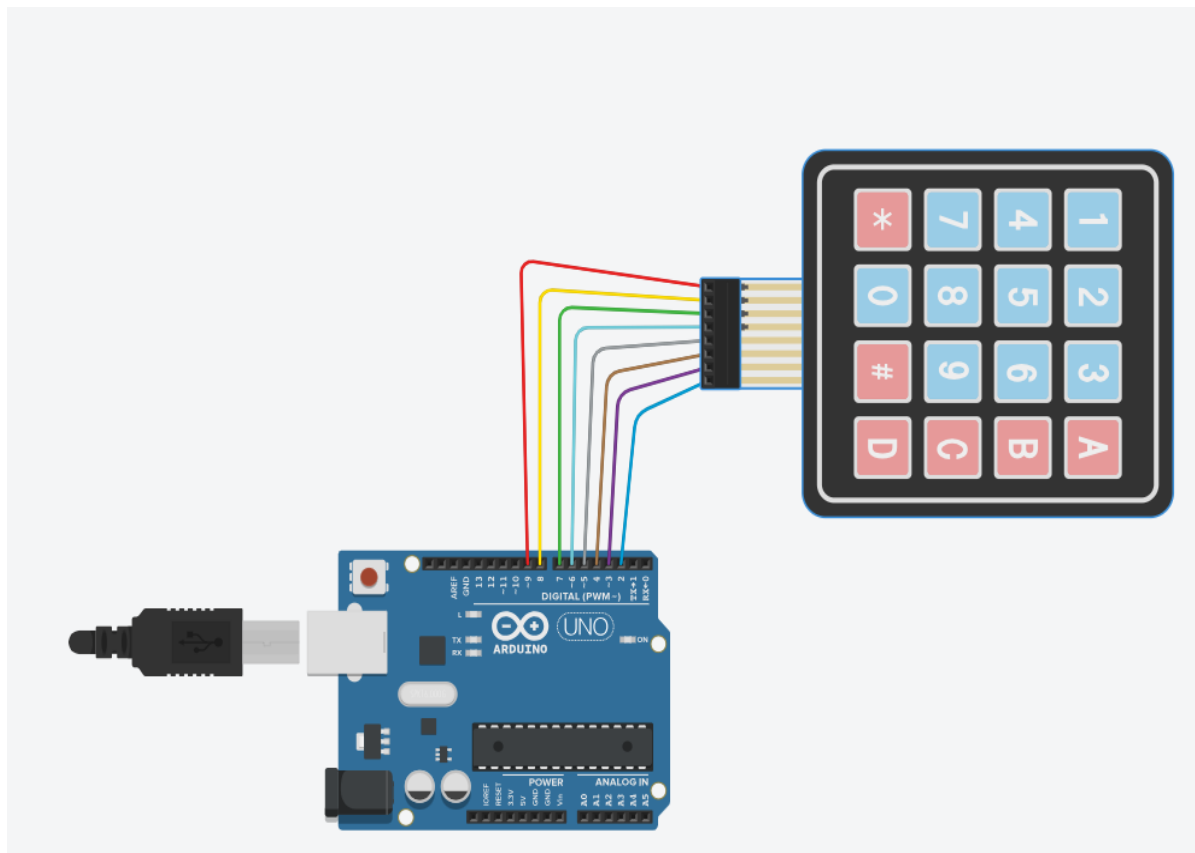
```
digitalWrite(4,HIGH);  
digitalWrite(3,HIGH);  
digitalWrite(2,HIGH);  
}  
delay (50);  
}
```

Experiment No: 12

Develop a program to simulate interfacing with the keypad module to record the keystrokes.

COMPONENTS:

S.NO.	Name	Quantity
1.	Arduino Uno	1
2.	Cable	17
3.	Keypad 4X4	1

CIRCUIT DIAGRAM:**SET UP:**

- Connect the Components as per circuit.
- Make sure VCC and Ground pins connected properly to avoid any damage to Arduino
- Locate the appropriate Components in search for components tab, located at the right-hand side corner of the Tinkercad web-based software.
- Select appropriate components and connect them as per the circuit diagram shown in the figure above.
- Type the code in code section tab.
- To get appropriate output click on Start Simulation tab.
- Click the keys on the Keypad to see the desired output in the Serial Monitor Section

CODE:

```
#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

char hexaKeys[ROWS][COLS] =

{

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'*','0','#','D'},

};

byte rowPins[ROWS] = {9, 8, 7, 6};

byte colPins[COLS] = {5, 4, 3, 2};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

void setup()

{

  Serial.begin(9600);

}

void loop()

{

  char customKey = customKeypad.getKey();

  if (customKey)

  {

    Serial.println(customKey);

    delay(100);

  }

}
```