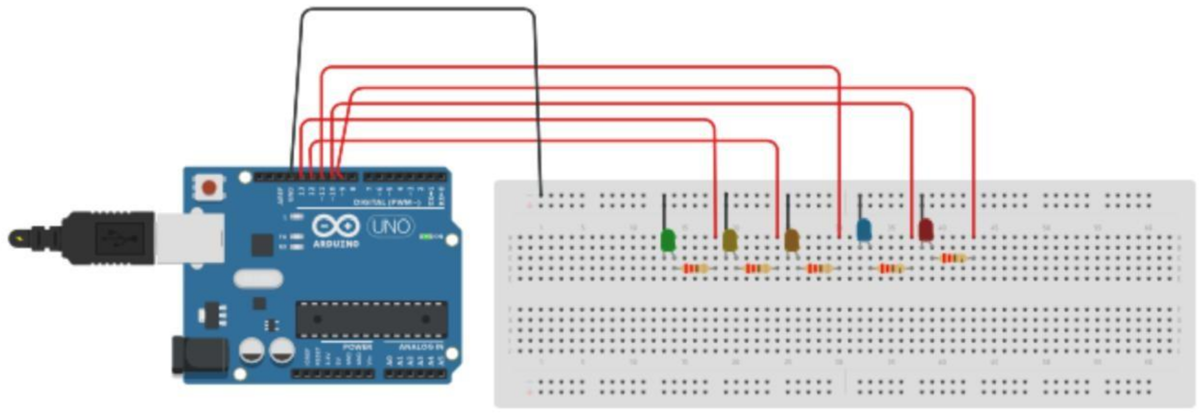
INTRENET OF THINGS (IPCC)

LAB MANUAL

Ex. No: 1	**Develop a program to blink 5 LEDs back and forth.**

Aim: Develop a program to blink 5 LEDs back and forth.



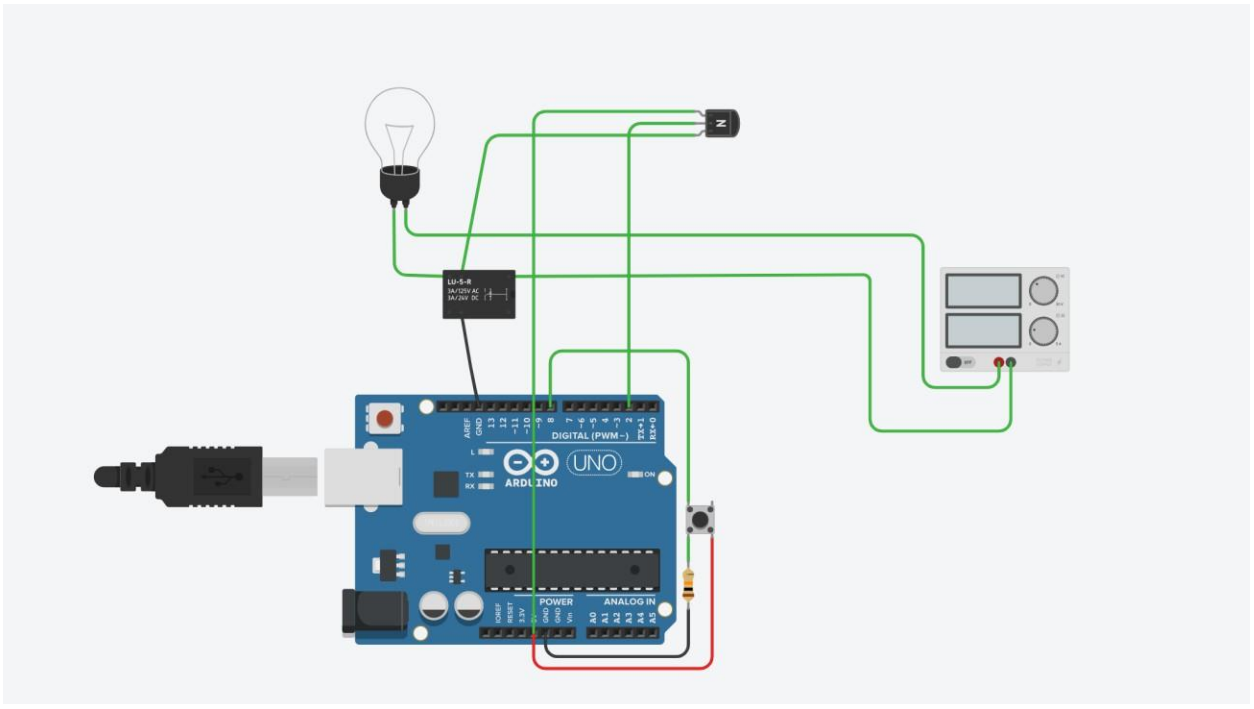
Coding

```
void setup()
{
  pinMode(13,   OUTPUT);
  pinMode(12,   OUTPUT);
  pinMode(11,   OUTPUT);
  pinMode(10,   OUTPUT);
  pinMode(9, OUTPUT);
}
void loop()
{
 digitalWrite(13, HIGH);// Wait for 1000 millisecond(s)
 delay(1000);
 digitalWrite(13, LOW);
 delay(1000);
 digitalWrite(12, HIGH);
 delay(1000);
 digitalWrite(12, LOW);
 delay(1000);
 digitalWrite(11, HIGH);
 delay(1000);
 digitalWrite(11, LOW);
 delay(1000);
 digitalWrite(10, HIGH);
 delay(1000);
 digitalWrite(10, LOW);
 delay(1000);
 digitalWrite(9, HIGH);
 delay(1000);
 digitalWrite(9, LOW);
 delay(1000);
}
```

Ex. No: 2	Develop a program to interface a relay with Arduino board.

Aim: Develop a program to interface a relay with Arduino board.

Circuit Diagram:

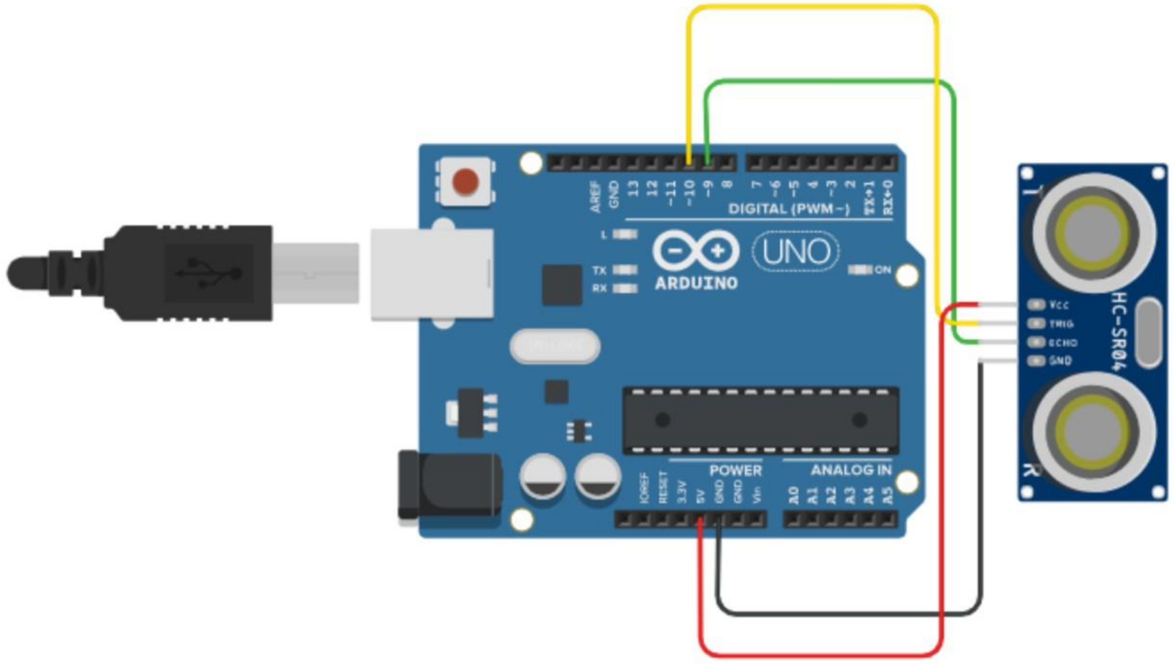


Arduino Code:

```
void setup()
{
  pinMode(3, OUTPUT);
}
void loop()
{
  digitalWrite(3, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(3, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Ex.No: 3 &11	**Develop a program to deploy an intrusion detection system using Ultrasonic and sound sensors.**
Date:	

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



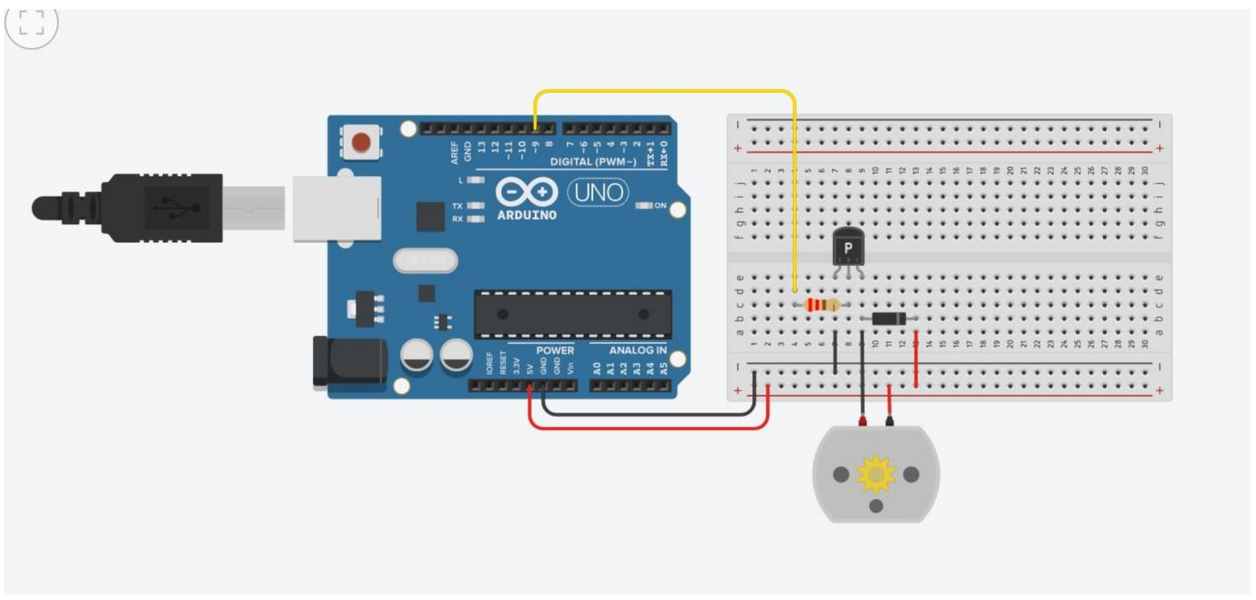
PSUEDOCode :

```
// C++ code

//

int trigPin =
10;        int
echoPin = 9;
long time;
```

```
float
distance;
void setup()
{

  pinMode(trigPin, OUTPUT); // SETTING OUTPUT PIN
  pinMode(echoPin, INPUT); // SETTING INPUT PIN
  Serial.begin(9600); // INITIALISING THE COMMUNICATION
}

void loop()

{

  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  // transmitting sound for 10
  microseconds digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(10, LOW);

 // calculating distance
  time=pulseIn(echoPin ,
  HIGH); Serial.print("time: ");
  Serial.println(time);
  distance = time * 0.0343/2;

// Printing out the final output => distance
  Serial.print("Distance:");
  Serial.println(distance);
}
```

Ex. No: 4	Develop a program to control a DC motor with Arduino board.

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



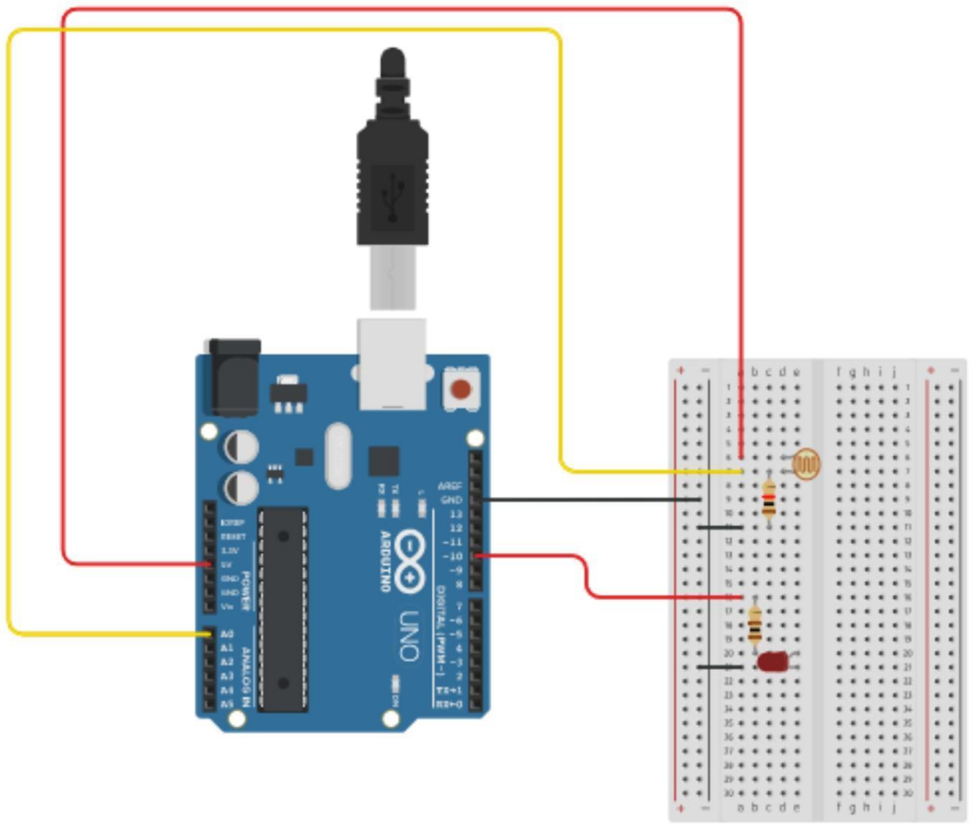
PSUEDO Code:

```
void setup()

{

  pinMode(13, OUTPUT);

  pinMode(12, OUTPUT);

}


void loop()

{

  digitalWrite(13, HIGH);

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

  digitalWrite(12, LOW);

  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);

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

  digitalWrite(12, HIGH);

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

}
```

Ex. No: 5	Develop a program to deploy smart street light system using LDR sensor.

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

.



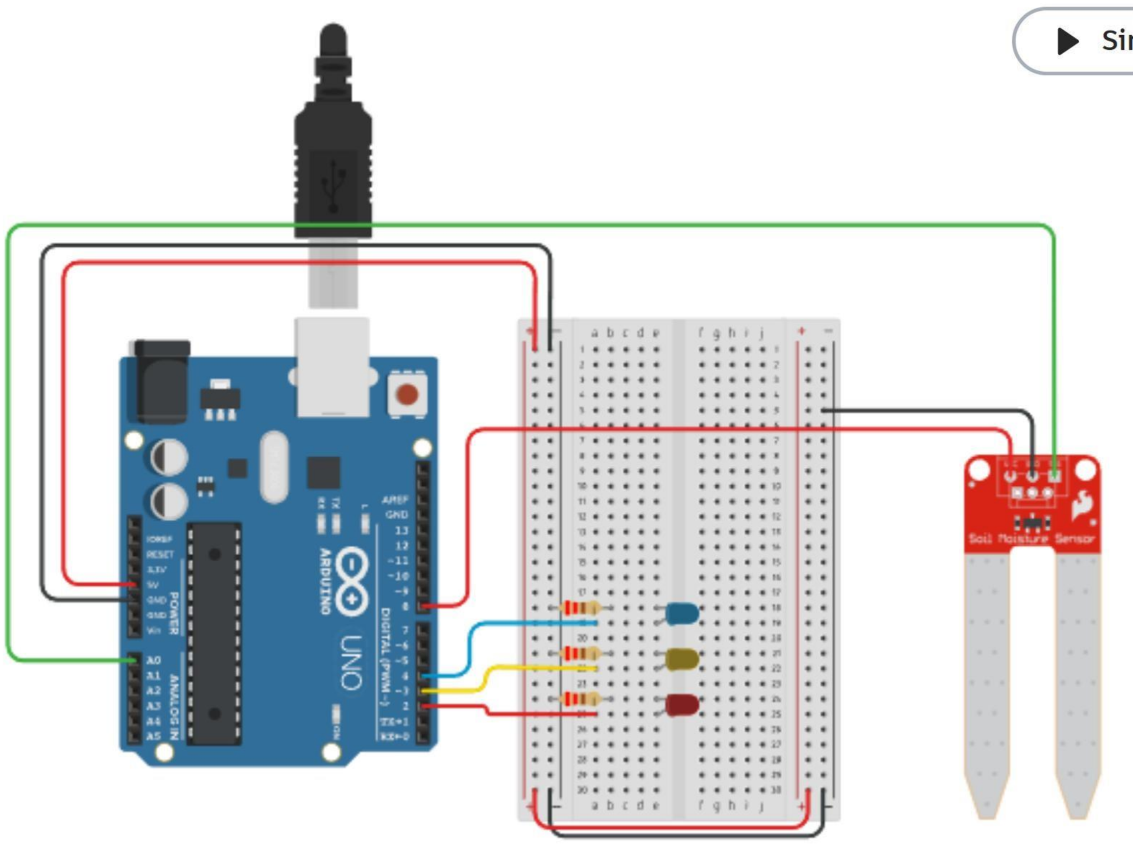
Arduino Code

```
// C++ code

//

int LDR_VAL = 0;


void setup()
```

```
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}
void loop()
{
  LDR_VAL = analogRead(A0);
  Serial.println(LDR_VAL);
  if (LDR_VAL > 500) {
    digitalWrite(8, HIGH);
  }
  digitalWrite(8, LOW);
  delay(10); // Delay a little bit to improve simulation performance
}
```

Ex. No: 6	Develop a program to classify dry and wet waste with the Moisture sensor (DHT22).

Aim: Develop a program to classify dry and wet waste with the Moisture sensor .



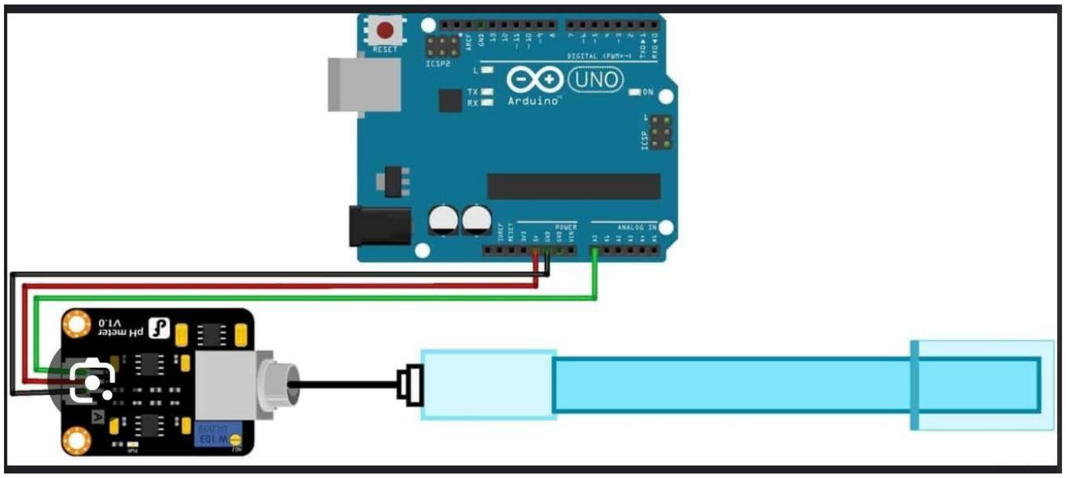
Arduino Code:

```
// declare variables for pins
const int sensorpin = A0;
const int sensorpower = 8;
const int LED1 = 2;
const int LED2 = 3;
const int LED3 = 4;

// variable for sensor reading
int sensor;
```

```
const int delayTime = 1000;

// "wet" and "dry" thresholds - these require calibration
int wet = 800;
int dry = 500;

void setup(){ // code that only runs once
  // set pins as outputs
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  pinMode(sensorpower,OUTPUT);

  // initialize serial communication
  Serial.begin(9600);
}

void loop(){ // code that loops forever
  // power on sensor and wait briefly
  digitalWrite(sensorpower,HIGH);
  delay(10);
  // take reading from sensor
  sensor = analogRead(sensorpin);
  // turn sensor off to help prevent corrosion
  digitalWrite(sensorpower,LOW);

  // print sensor reading
  Serial.println(sensor);

  // If sensor reading is greater than "wet" threshold,
  // turn on the blue LED. If it is less than the "dry"
  // threshold, turn on the red LED. If it is in between
  // the two values, turn on the yellow LED.
  if(sensor>wet){ digitalWrite(LED1,LOW);
  digitalWrite(LED2,LOW);
  digitalWrite(LED3,HIGH);
  }
  else if(sensor<dry){
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,LOW);
    digitalWrite(LED3,LOW);
  }
  else{ digitalWrite(LED1,LOW);
    digitalWrite(LED2,HIGH);
    digitalWrite(LED3,LOW);
  }

  // wait before taking next reading
  delay(delayTime);

}
```

Ex. No: 7	**Develop a program to read the pH value of a various substances like milk, lime and water.**

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

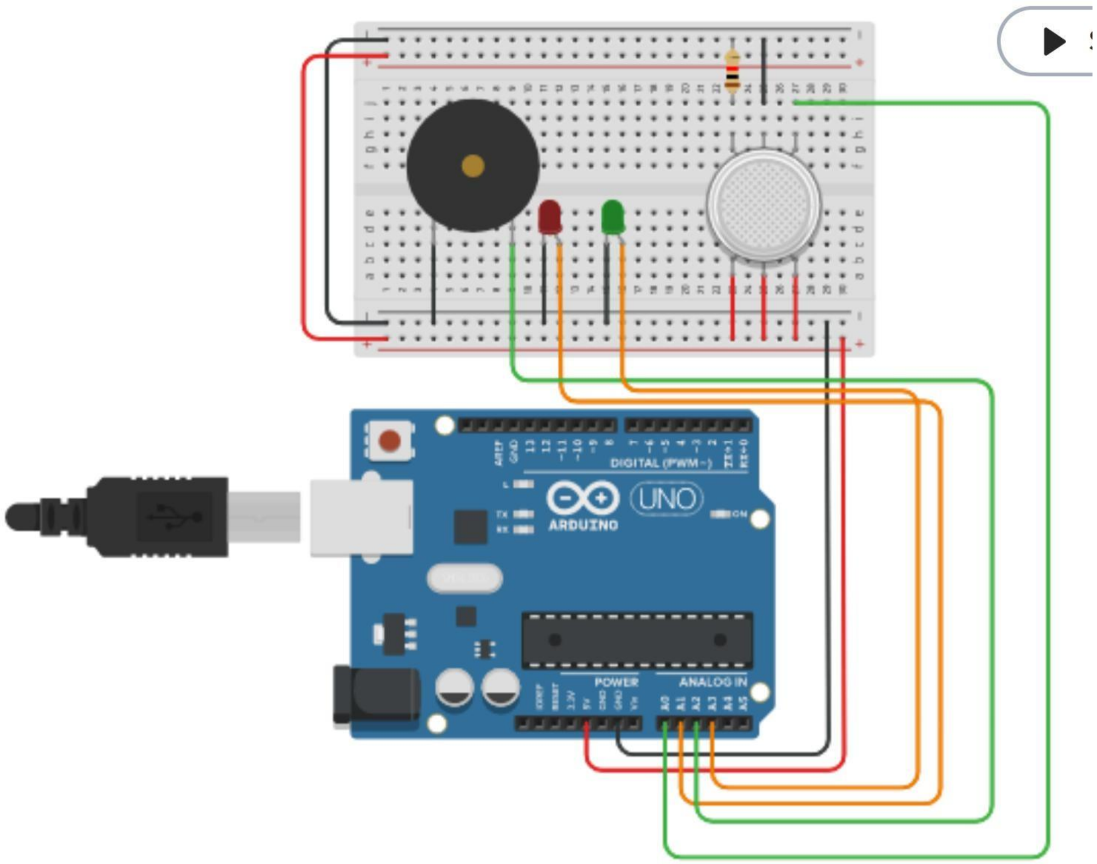


Arduino Code:

```
/*
  # This sample code is used to test the pH meter V1.0. # Editor :
  YouYou
  # Ver        : 1.0
  # Product: analog pH meter #
  SKU          : SEN0161
*/
#define SensorPin A0                    //pH meter Analog output to Arduino Analog Input 0
#define Offset 0.00                     //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth  40            //times of collection
int  pHArray[ArrayLenth];     //Store the average value of the sensor feedback
int pHArrayIndex=0;
void setup(void)
{
    pinMode(LED,OUTPUT);
    Serial.begin(9600);
    Serial.println("pH meter experiment!");                    //Test the serial monitor
}
void loop(void)
{
    static unsigned long samplingTime = millis(); static
    unsigned long printTime = millis();
    static float pHValue,voltage;
    if(millis()-samplingTime > samplingInterval)
  {
            pHArray[pHArrayIndex++]=analogRead(SensorPin);
            if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
            voltage =
            pHValue =  avergearray(pHArray, ArrayLenth)*5.0/1024; 3.5*voltage+Offset;
            samplingTime=millis();
  }
    if(millis() - printTime > printInterval)                   //Every 800 milliseconds, print a
numerical, convert the state of the LED indicator
  {
        Serial.print("Voltage:");
                Serial.print(voltage,2); Serial.print("
                                                    pH value: ");
        Serial.println(pHValue,2);
                digitalWrite(LED,digitalRead(LED)^1);
                printTime=millis();
  }
}
double avergearray(int* arr, int number){ int i;
    int max,min;
    double avg; long
    amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/n");
        return 0;
  }
    if(number<5){        //less than 5, calculated directly statistics
        for(i=0;i<number;i++){
            amount+=arr[i];
        }
        avg = amount/number;
        return avg;
  }else{
        if(arr[0]<arr[1]){
            min = arr[0];max=arr[1];
        }
        else{
            min=arr[1];max=arr[0];
        }
        for(i=2;i<number;i++){ if(arr[i]<min){
                amount+=min;               //arr<min
                min=arr[i];
            }else {
                if(arr[i]>max){
                    amount+=max;      //arr>max
                    max=arr[i];
                }else{
                    amount+=arr[i]; //min<=arr<=max
                }
            }//if
        }//for
        avg = (double)amount/(number-2);
  }//if return
    avg;

}
```

Ex. No: 8	Develop a program to detect the gas leakage in the surrounding environment.

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

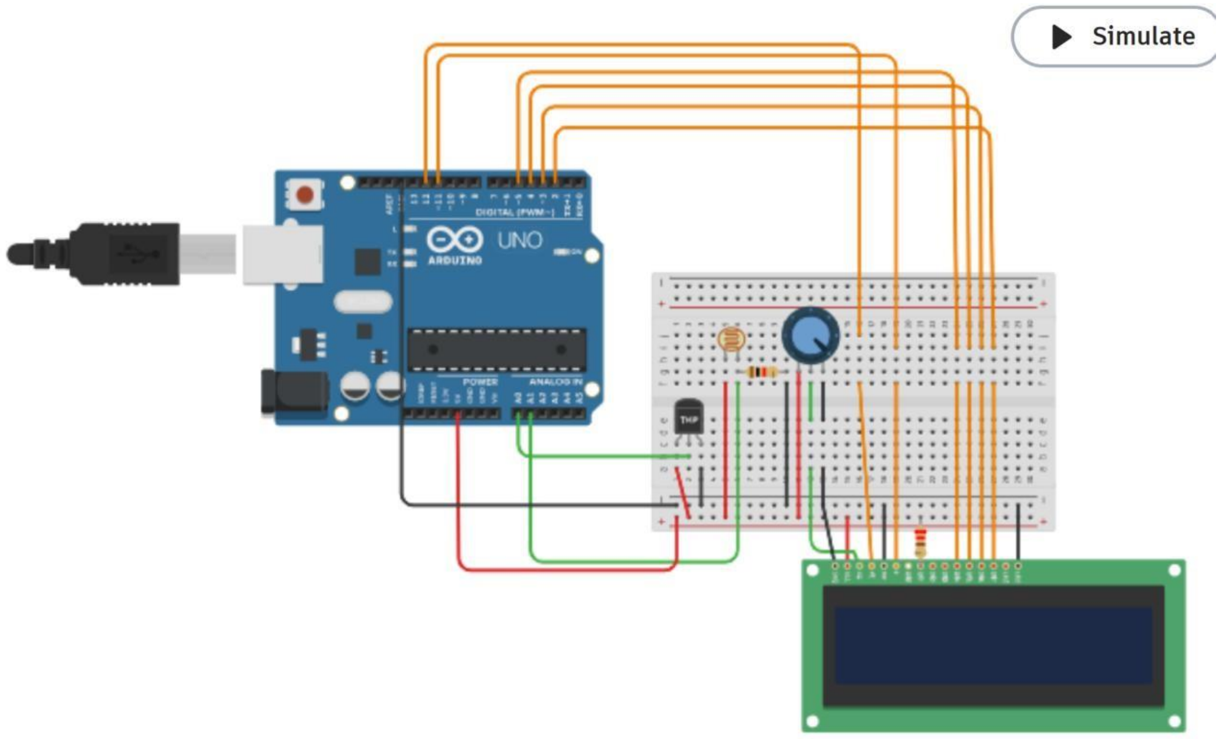


Arduino Code:

```
int LED = A1;                   //Red LED
int LED1 = A3;                  //Green LED
int gas_pin = A0;                 // For Gas Sensor int
buzzer_pin = A2;                // For Buzzer

void setup()
{
   Serial.begin(9600);
   pinMode (buzzer_pin, OUTPUT);
   pinMode (gas_pin, INPUT);
}

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


   if(sensorValue >= 300)
   {
       digitalWrite(LED,HIGH);
       digitalWrite(LED1,LOW);

       digitalWrite (buzzer_pin, HIGH);
       //Serial.println(); Serial.print(sensorValue);
       Serial.println(" |SMOKE DETECTED|");
   }

   else
   {
            digitalWrite(LED,LOW);
       digitalWrite(LED1,HIGH);

       digitalWrite (buzzer_pin, LOW);
       Serial.println(); Serial.println("Sensor
       Value: "); Serial.print(sensorValue);
       //Serial.print(" |Safe Mode|");
   }

   delay(1000);

}
```

Ex. No: 9	Develop a program to demonstrate weather station readings using Arduino.

Aim: Develop a program to demonstrate weather station readings using Arduino.



Arduino Code:

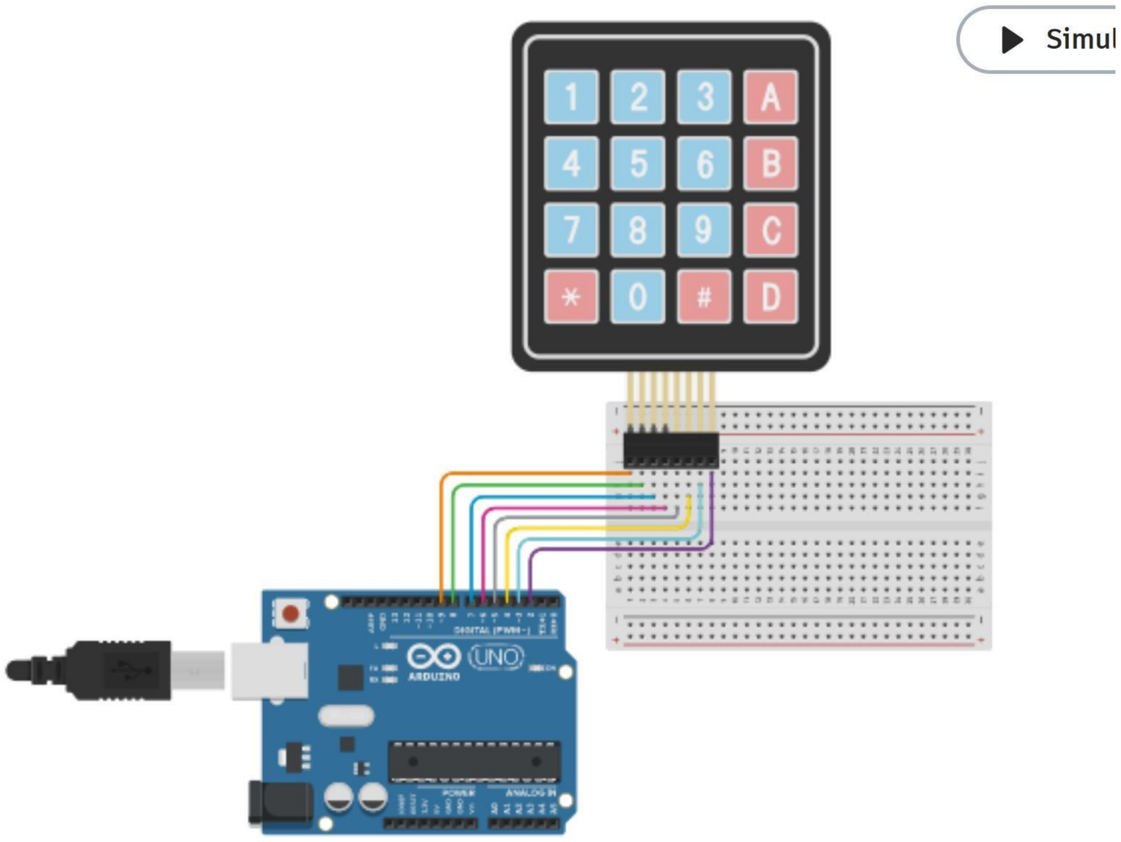
```
#include <LiquidCrystal.h> // including LiquidCrystal Library

LiquidCrystal lcd(12,11,5,4,3,2); // Creating LC object where parameters:
(RS,E,DB4,DB5,DB6,DB7)

void setup()

{

lcd.begin(16,2); // 16 Characters per line and  2 lines
lcd.setCursor(4,0); // Positioning of the LCD cursor. col=4, row=0
lcd.print("Monitoring"); // Printing text to the LCD
lcd.setCursor(7,1); // Positioning of the LCD cursor. col=7, row=1
lcd.print("System"); // Printing text to the LCD
delay(2000); // time delay

}

void loop()

{

lcd.clear(); // Clearing the LCD screen and positioning the cursor in the upper-left
corner



float Voltage=analogRead(A0)*0.004882814; // Variable declartion and analog input
pin select

float degrees = ( Voltage - 0.5 ) * 100; // Variable declartion and modification



lcd.setCursor(3,0); // Positioning of the LCD cursor. col=3, row=0
lcd.print("Temp:"); // Printing text to the LCD
lcd.setCursor(8,0); // Positioning of the LCD cursor. col=8, row=0
lcd.print(degrees); // Printing text to the LCD
if (degrees<10) // Checking variable value using if function

{

lcd.setCursor(4,1); // Positioning of the LCD cursor. col=4, row=1
lcd.print("-TOO COLD"); // Printing text to the LCD
}

else if(degrees<18 &&degrees>10 ) // Checking variable value using else if function
```

```
{

lcd.setCursor(4,1); // Positioning of the LCD cursor. col=4, row=1
lcd.print("-COLD"); // Printing text to the LCD
}

else if(degrees<30 &&degrees>18 ) // Checking variable value using else if function

{

lcd.setCursor(4,1); // Positioning of the LCD cursor. col=4, row=1
lcd.print("-Normal Temp"); // Printing text to the LCD
}

else if(degrees<45 &&degrees>30 ) // Checking variable value using else if function

{

lcd.setCursor(4,1); // Positioning of the LCD cursor. col=4, row=1
lcd.print("-Hot"); // Printing text to the LCD
}

else if(degrees>45 ) // Checking variable value using else if function

{

lcd.setCursor(4,1); // Positioning of the LCD cursor. col=4, row=1
lcd.print("-TOO HOT"); // Printing text to the LCD
}

delay(2000); // time delay

lcd.clear(); // Clearing the LCD screen and positioning the cursor in the upper-left
int ldr=analogRead(A1); // Variable declartion and analog input pin select
```

```
lcd.setCursor(1,0); // Positioning of the LCD cursor.
lcd.print("Intensity:"); // Printing text to the LCD
lcd.setCursor(12,0); // Positioning of the LCD cursor.
lcd.print(ldr); // Printing text to the LCD
if (ldr<230) // Checking variable value using if function

{

lcd.setCursor(4,1); // Positioning of the LCD cursor.
lcd.print("-TOO BRIGHT"); // Printing text to the LCD
}

else if (ldr<460 && ldr>230 ) // Checking variable value using else if function

{

lcd.setCursor(2,1); // Positioning of the LCD cursor.
lcd.print("-Medium Light"); // Printing text to the LCD
}

else if (ldr>460) // Checking variable value using else if function

{

lcd.setCursor(4,1); // Positioning of the LCD cursor.
lcd.print("-TOO DARK"); // Printing text to the LCD
}

delay(2000); // time delay

lcd.clear(); // Clearing the LCD screen and positioning the cursor in the upper-left
corner

}
```

Ex. No: 12	Develop a program to simulate interfacing with the keypad module to record the keystrokes.

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



CODE:

```
#include <Keypad.h>

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

//keymap defines the key pressed according to the row and columns just as appears on the keypad

char keymap[numRows][numCols]=

{
```

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

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

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

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

};

//Code that shows the the keypad connections to the arduino
terminals byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3

//initializes an instance of the Keypad class

Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);

void setup()

{

Serial.begin(9600);

}

//If key is pressed, this key is stored in 'keypressed' variable

//If key is not equal to 'NO_KEY', then this key is printed out

//if count=17, then count is reset back to 0 (this means no key is pressed during the whole keypad
scan process
void loop()

{

char keypressed =
myKeypad.getKey(); if (keypressed
!= NO_KEY)
{

Serial.println(keypressed);
}

}
```