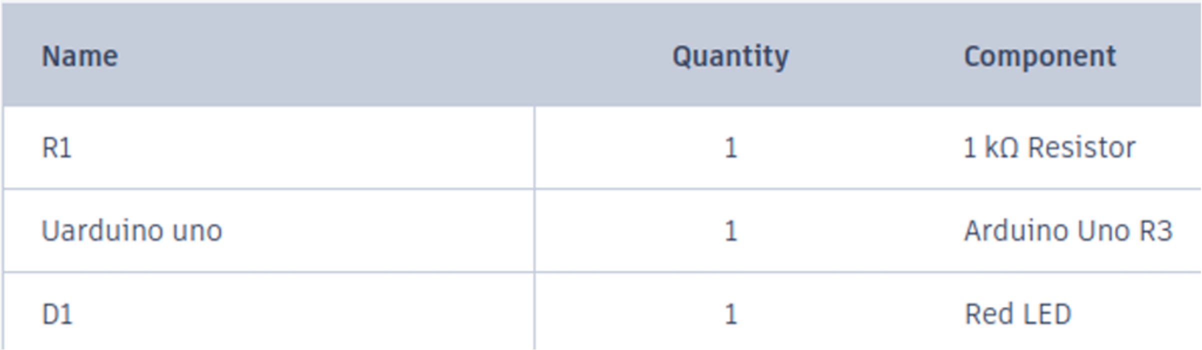
Experiment-1: LED Interfacing

AIM : To implement and verify LED interfacing using Arduino Uno.

a) Single Led
b)Multiple Led’s

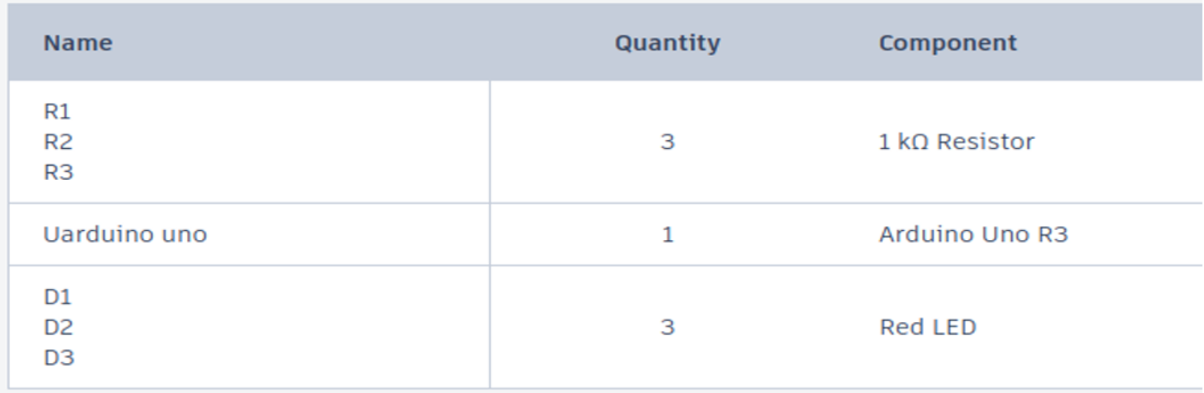
Software Used: Arduino IDE Apparatus:

a) Single Led



Name	Quantity	Component
R1	1	1 kΩ Resistor
Uarduino uno	1	Arduino Uno R3
D1	1	Red LED

b)Multiple Led’s



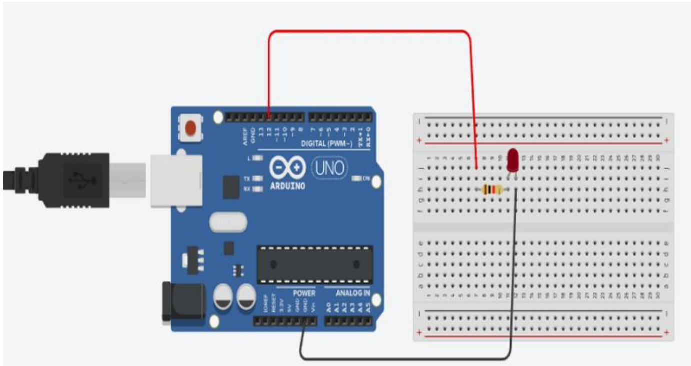
Name	Quantity	Component
R1 R2 R3	3	1 kΩ Resistor
Uarduino uno	1	Arduino Uno R3
D1 D2 D3	3	Red LED

Program:

a) Single Led

```
#define led=12
void setup() {
pinMode(led,OUTPUT);
}
void loop() {
digitalWrite(led,HIGH);
delay(50);
digitalWrite(led,LOW);
delay(50);}
```

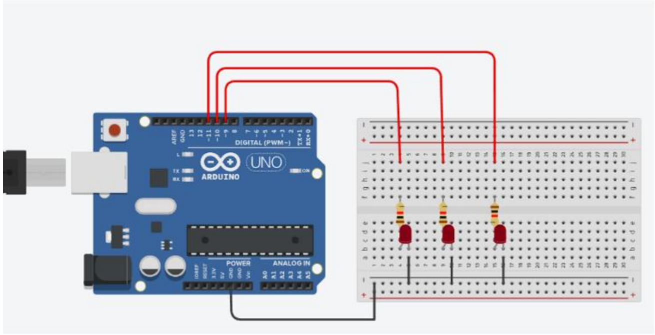
Circuit Diagram: a)Single LED



b)Multiple Led's

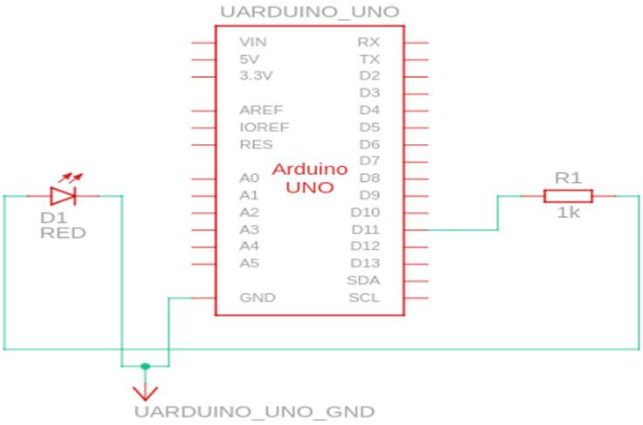
```
int L1=9;
int L2=10;
int L3=11;
int del=100;
void setup() {
pinMode(L1,OUTPUT);
pinMode(L2,OUTPUT);
pinMode(L3,OUTPUT); }
void loop() {
digitalWrite(L1,HIGH);
digitalWrite(L2,LOW);
digitalWrite(L3,LOW);
delay(del);
digitalWrite(L1,LOW);
digitalWrite(L2,HIGH);
digitalWrite(L3,LOW);
delay(del);
digitalWrite(L1,LOW);
digitalWrite(L2,LOW);
digitalWrite(L3,HIGH);
delay(del);
}
```

b)Multiple Led's

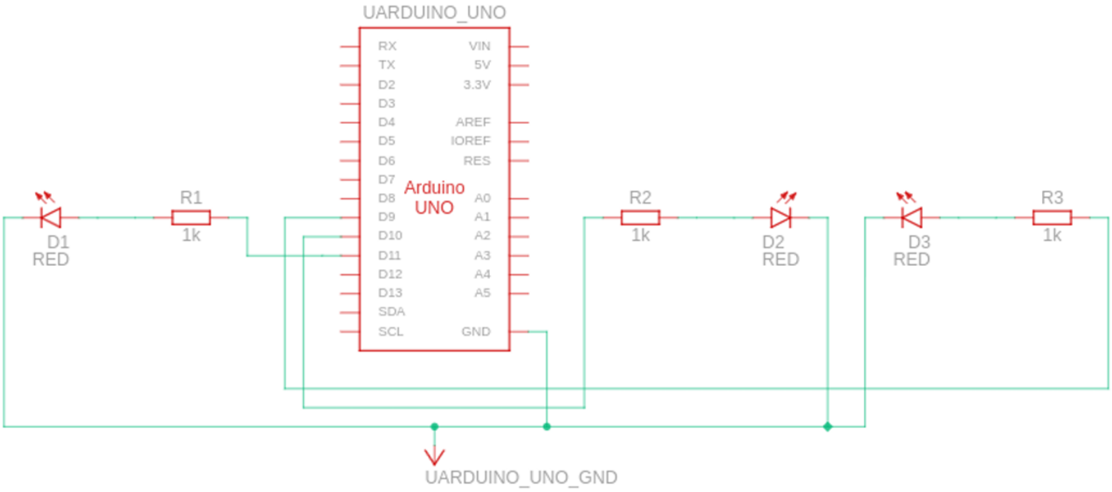


Schematic Diagram:

a)Single Led



b)Multiple Led's



Procedure:

Step 1: Connect the LED Connect the components on the breadboard as follows:

- Connect the longer leg of the LED (anode) to digital pin 12 on the Arduino Uno.
- Connect the shorter leg of the LED (cathode) to the ground (GND) on the Arduino.
- Connect one end of the resistor to the cathode of the LED.
- Connect the other end of the resistor to a different ground pin on the Arduino.
- For multiple LEDs connect the longer leg of the LEDs(anode) to digital pins 9,10,11.

Step 2: Write the Arduino Code Open the Arduino IDE on your computer and write a simple code to turn the LED on and off.

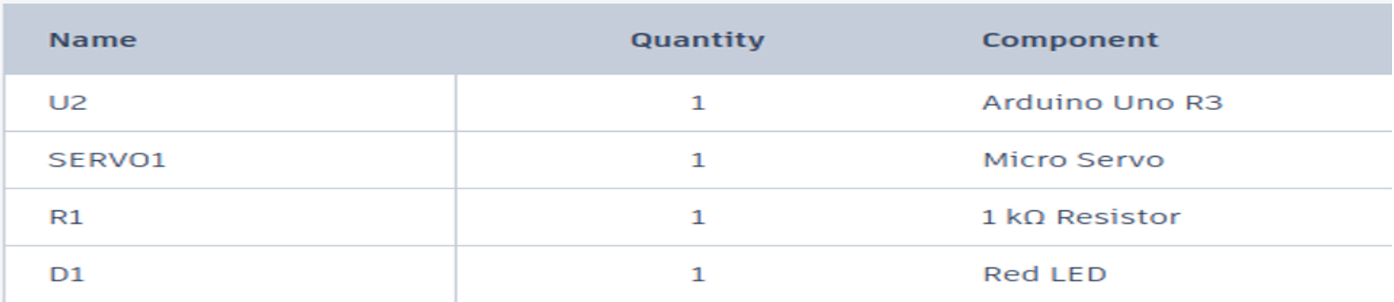
Step 3: Upload the Code Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 4: Verify the LED Operation After uploading the code, the LED should start blinking, turning on for 1 second and then off for 1 second. If the LED is not blinking, double-check your connections and code for any errors

AIM: a)To control speed of servo motor using arduino.

b)To glow led for each 45 degree rotation of the servo motor using arduino.

Software Used: Arduino IDE Apparatus:

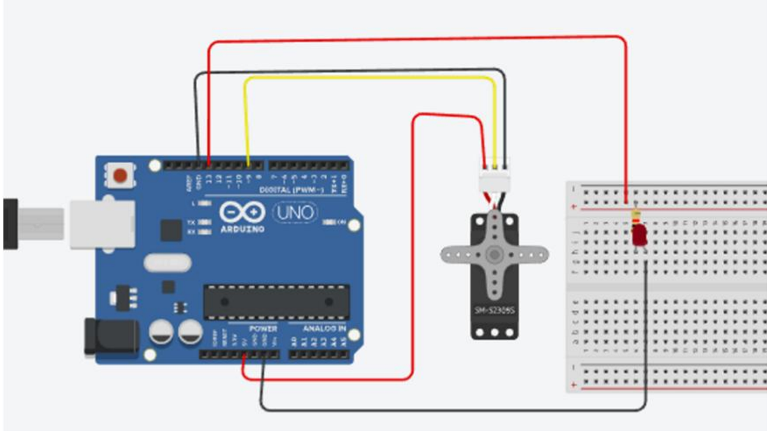


Name	Quantity	Component
U2	1	Arduino Uno R3
SERVO1	1	Micro Servo
R1	1	1 kΩ Resistor
D1	1	Red LED

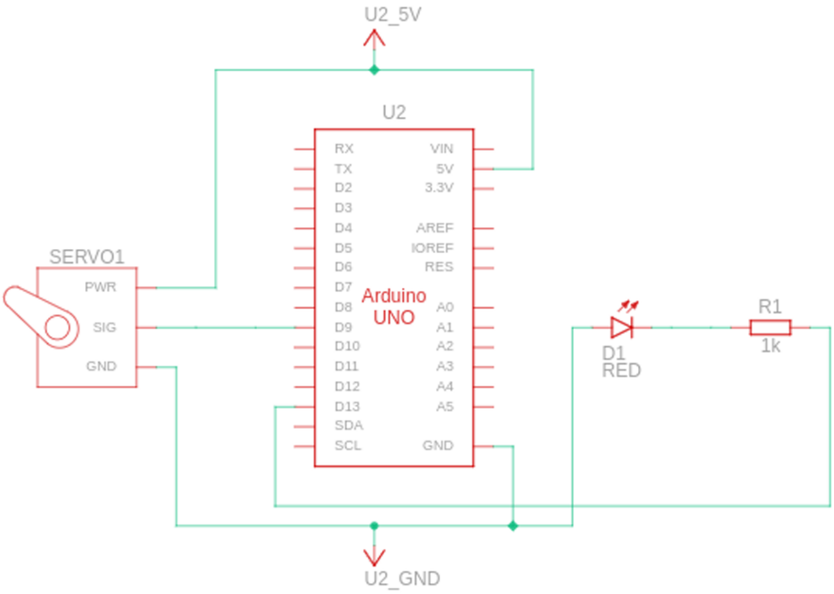
Program:

```
#include int
del=1000;
int led=13;
int servo=9;
int k; Servo myservo;
void setup(){
myservo.attach(servo);
pinMode(led,OUTPUT); }
void loop(){
k=myservo.read();
myservo.write(0);
delay(1000);
myservo.write(45);
digitalWrite(led,HIGH);
delay(1000);
digitalWrite(led,LOW);
delay(1000); .
myservo.write(90);
digitalWrite(led,HIGH);
delay(1000);
digitalWrite(led,LOW);
delay(1000);
myservo.write(135);
digitalWrite(led,HIGH);
```

Circuit Diagram:



Schematic Diagram:



```
delay(1000);
digitalWrite(led,LOW);
delay(1000);
myservo.write(180);
digitalWrite(led,HIGH);
delay(1000);
digitalWrite(led,LOW);
}
```

Procedure:

Step 1: Connect the Servo Motor Connect the servo motor to the Arduino as follows:

- Connect the servo's power wire (usually red) to the 5V output on the Arduino.
- Connect the servo's ground wire (usually brown) to any ground (GND) on the Arduino.
- Connect the servo's signal wire (usually yellow or orange) to a PWM pin on the Arduino, e.g., pin 9.
- For connection of the LED connect longer leg of the LED(anode) to digital pin 11 ,shorter leg of th eLED(cathode) to GND and connect one end of the resistor to the cathode of LED.

Step 2: Write the Arduino Code Open the Arduino IDE on your computer and write a simple code to control the speed of the servo motor

Step 3: Upload the Code Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

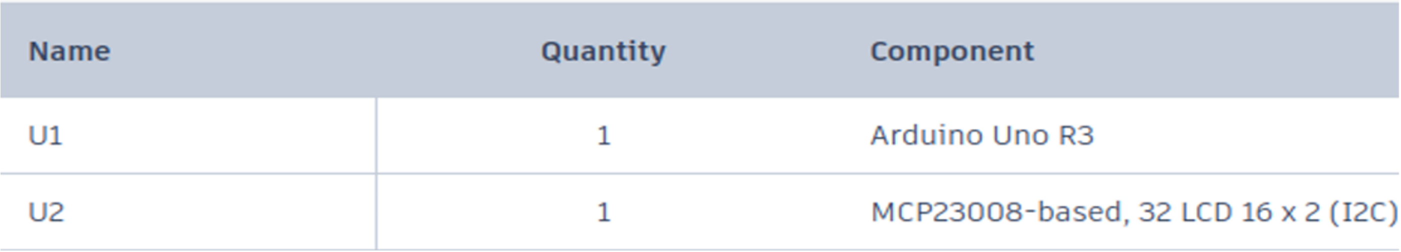
Step 4: Verify Servo Motor Operation Turn the potentiometer and observe the servo motor's rotation speed. The servo should move based on the potentiometer's position, adjusting the speed accordingly.

Experiment 3: LCD Interfacing

Aim: To interface the LCD using Arduino

Software Used: Arduino IDE

Apparatus:

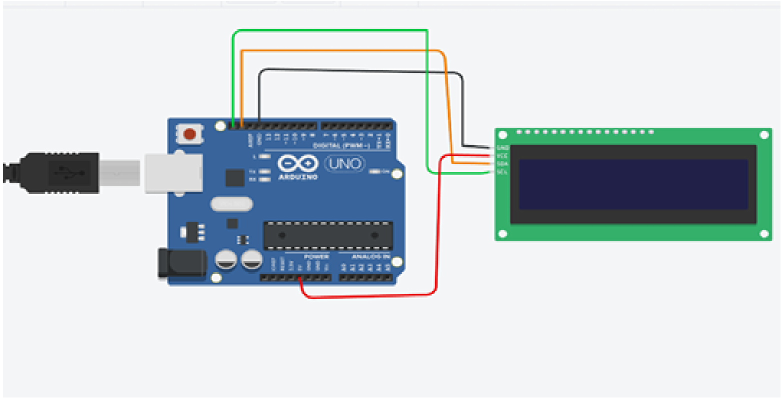


Name	Quantity	Component
U1	1	Arduino Uno R3
U2	1	MCP23008-based, 32 LCD 16 x 2 (I2C)

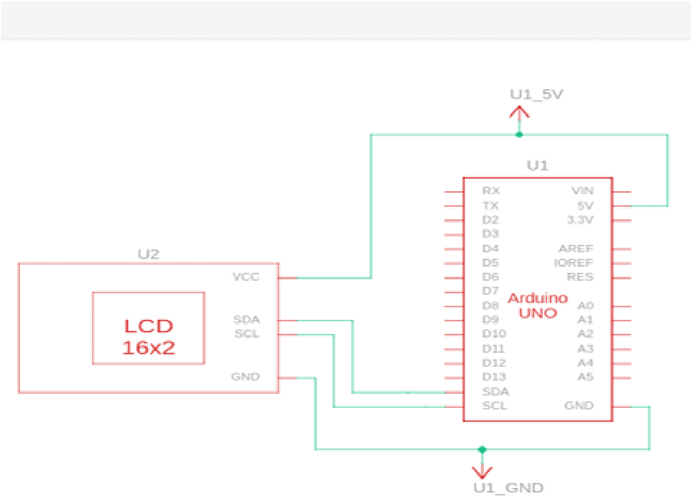
Program:

```
#include <wire.h>
#include LiquidCrystal_I2C
lcd(0X27,16,2);
void setup() {
lcd.begin(16,2);
lcd.backlight();
lcd.print(“.....Hello World...!”);
lcd.setCursor(“Welcome”); }
void loop() { }
```

Circuit Diagram:



Schematic Diagram:



Procedure:

Step 1: Connect the LCD to the Arduino Connect the LCD to the Arduino following the instructions below.

- LCD VSS (Pin 1): Connect to GND on Arduino
- LCD VDD (Pin 2): Connect to 5V on Arduino
- LCD V0 (Pin 3): Connect to one end of the fixed resistor
- LCD RS (Pin 4): Connect to digital pin 12 on Arduino
- LCD RW (Pin 5): Connect to GND on Arduino
- LCD E (Pin 6): Connect to digital pin 11 on Arduino
- LCD D4 (Pin 11): Connect to digital pin 5 on Arduino
- LCD D5 (Pin 12): Connect to digital pin 4 on Arduino
- LCD D6 (Pin 13): Connect to digital pin 3 on Arduino
- LCD D7 (Pin 14): Connect to digital pin 2 on Arduino
- LCD A (Pin 15): Connect to +5V on Arduino
- LCD K (Pin 16): Connect to GND on Arduino
- Other end of the fixed resistor (V0): Connect to GND on Arduino

Step 2: Write the Arduino Code Write a simple code to display text on the LCD. The contrast may vary depending on the resistor value; you may need to adjust it in the code.

Step 3: Upload the Code Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 4: Verify LCD Operation After uploading the code, you should see the text "Hello, Arduino!" on the first line and "LCD Interfacing" on the second line of the LCD. If the contrast is not suitable, you may need to experiment with different resistor values to get a clear display.

Experiment 4: PIR Sensor Interfacing

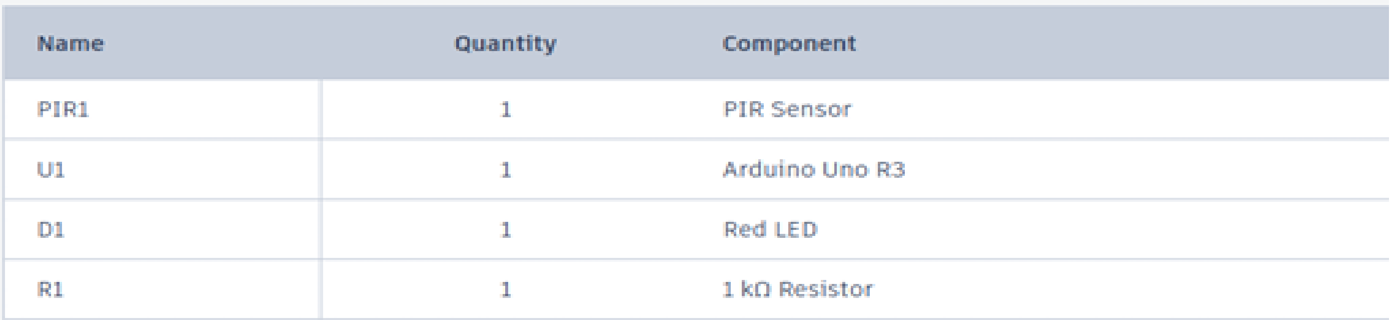
Aim: To Interface PIR sensor and glow led by using arduino and print information on :

a)Serial Monitor

b)LCD

Software Used: Arduino IDE

Apparatus:

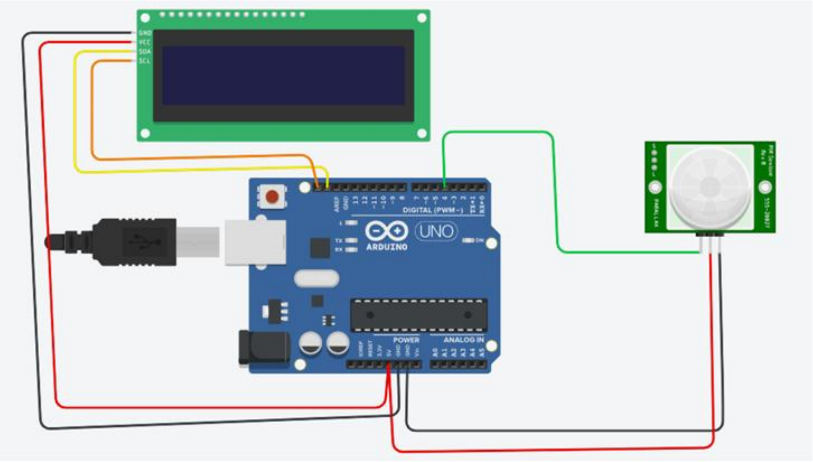


Name	Quantity	Component
PIR1	1	PIR Sensor
U1	1	Arduino Uno R3
D1	1	Red LED
R1	1	1 kΩ Resistor

Program:

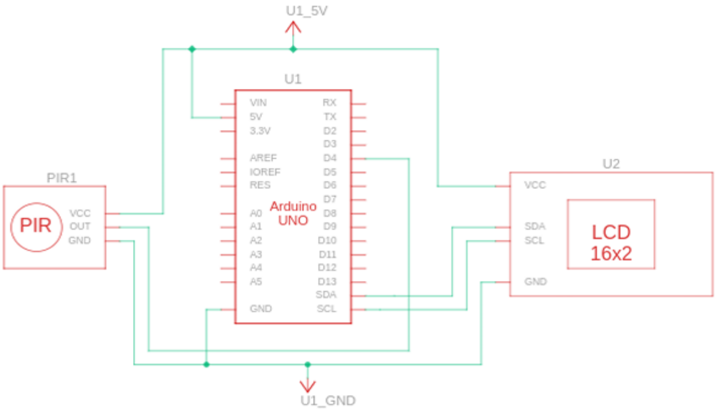
```
#include<LiquidCrystal>
#include<wire.h>
#include <LiquidCrystal
_I2C lcd(0X27,16,2)>
int pirpin=5;
int led=10;
void setup() {
lcd.begin(0,0);
lcd.setCursor(1,0);
pinMode(pirpin,INPUT);
pinMode(led,OUTPUT);
Serial.begin(9600);
lcd.backlight(); }
void loop() {
int pirval=digitalRead(pirpin);
```

Circuit Diagram:



```
if (pirval==HIGH) {
Serial.println("Moving");
lcd.print("ss");  delay(1000);
digitalWrite(led,HIGH);
delay(1000);
digitalWrite(led,LOW);
// delay(1000); }
else { lcd.print("no"); }
 delay(1000); }
```

Schematic Diagram:



Procedure:

Step 1: Connect the PIR Sensor

Connect the PIR sensor, LED, and 16x2 LCD to the Arduino as follows:

PIR Sensor:

Connect the VCC pin to 5V on Arduino.

Connect the GND pin to GND on Arduino.

Connect the OUT pin to digital pin 2 on Arduino.

LED:

Connect the anode (longer leg) of the LED to digital pin 13 on Arduino through a resistor (220-330 ohms).

Connect the cathode (shorter leg) of the LED to GND on Arduino.

16x2 LCD:

Follow the wiring instructions from the earlier response for interfacing the 16x2 LCD with Arduino.

Step 2: Write the Arduino Code

Write a code to read the PIR sensor input, control the LED based on motion detection, and print information on the Serial Monitor.

Step 3: Upload the Code

Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 4: Open the Serial Monitor

After uploading the code, open the Serial Monitor in the Arduino IDE

(Tools > Serial Monitor) to view the information about motion detection.

Step 5: Verify Operation

Move in front of the PIR sensor, and the LED should turn on while the Serial Monitor displays "Moving object found" .When there is no motion, the LED should turn off, and the Serial Monitor will display "no."

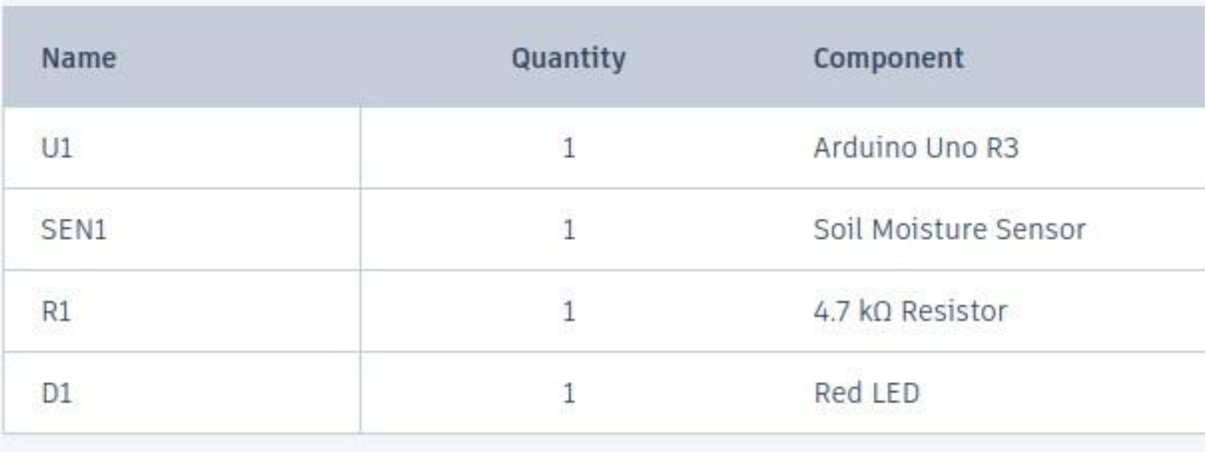
Experiment 5: Moisture Sensor Interfacing

Aim: To interface moisture sensor using arduino.   a) To glow led when there is moisture

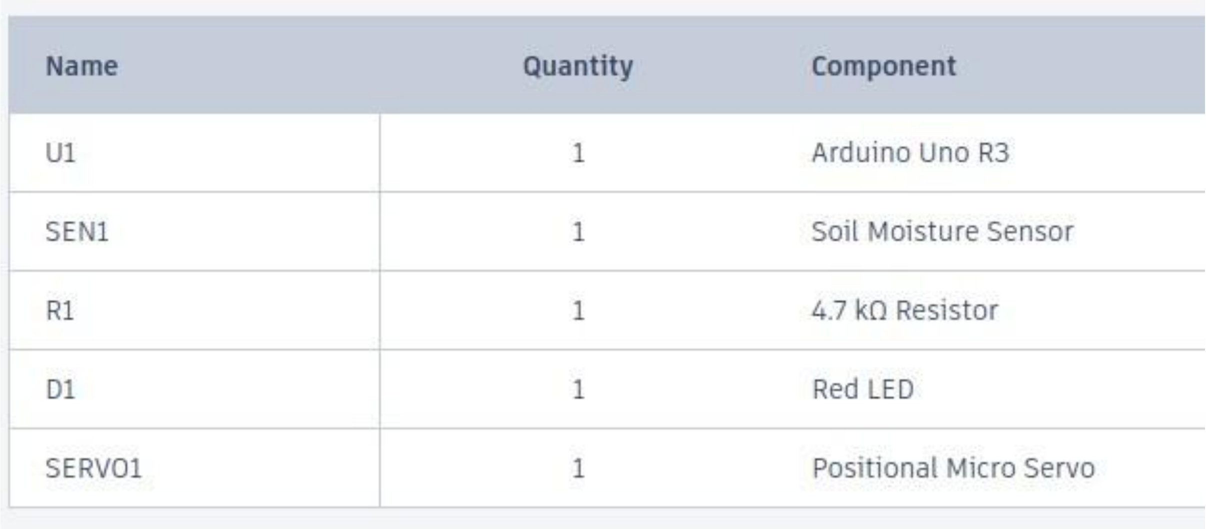
b)To rotate servo motor when analog moisture value is >500

Software Used: Arduino IDE

Apparatus:

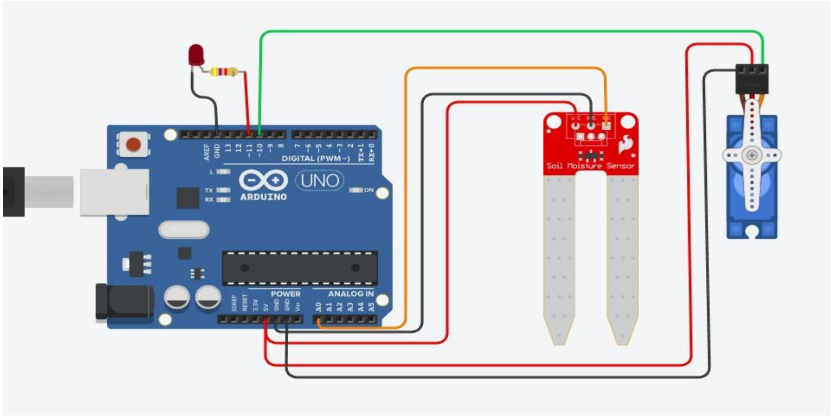


Name	Quantity	Component
U1	1	Arduino Uno R3
SEN1	1	Soil Moisture Sensor
R1	1	4.7 kΩ Resistor
D1	1	Red LED



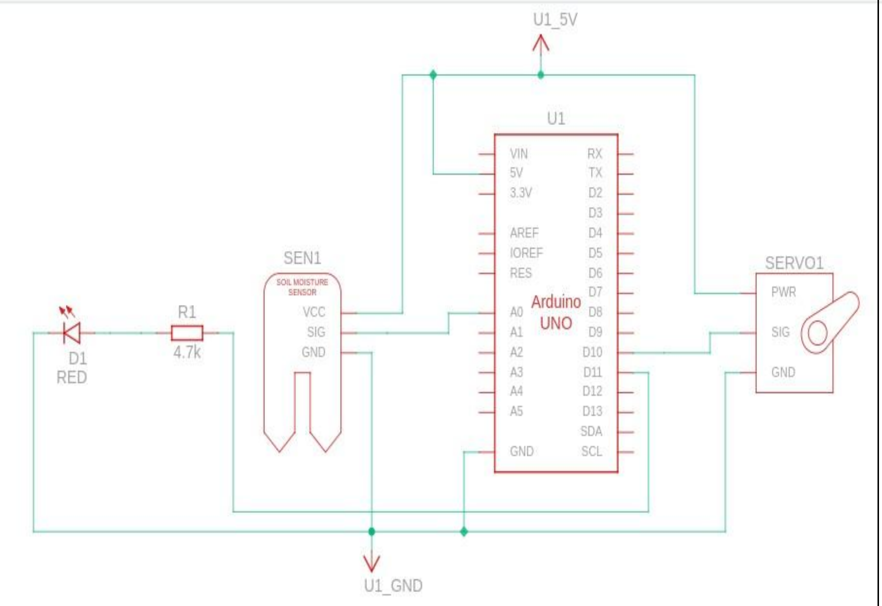
Name	Quantity	Component
U1	1	Arduino Uno R3
SEN1	1	Soil Moisture Sensor
R1	1	4.7 kΩ Resistor
D1	1	Red LED
SERVO1	1	Positional Micro Servo

```
#include<Servo.h>
int moisture;
int moist=A0;
int led=11;
Servo s;
void setup()
{
  pinMode(moist,INPUT);
```



```
Serial.begin(9600);
pinMode(led,OUTPUT);
 s.attach(10); }
void loop() {
moisture=analogRead(A0);
Serial.println(moisture);
 if (moisture>500)
 {
  digitalWrite(led,HIGH);
s.write(90);   delay(1000);
s.write(180);   delay(1000);
s.write(90);   delay(1000);
s.write(0);   delay(1000);
 }
else
 {
  s.write(0);   digitalWrite(led,LOW)
 }
```

Schematic Diagram:



Procedure:

Step 1: Connect the Moisture Sensor Connect the moisture sensor to the Arduino:

- **Moisture Sensor:**
- Connect the VCC pin to 5V on Arduino.
- Connect the GND pin to GND on Arduino.
- Connect the signal (SIG) pin to digiral pin 6 on Arduino.

Step 2: Connect the LED Connect the LED to the Arduino:

- **LED:**
- Connect the anode (longer leg) to digital pin 11 on Arduino through a resistor (220-330 ohms).
- Connect the cathode (shorter leg) to GND on Arduino.

Step 3: Connect the Servo Motor Connect the servo motor to the Arduino:

- **Servo Motor:**

- Connect the signal wire to digital pin 10 on Arduino.
- Connect the power wire to 5V on Arduino.
- Connect the ground wire to GND on Arduino.

Step 4: Write the Arduino Code

Write a code to read the moisture level, control the LED and servo motor based on moisture conditions.

Step 5: Upload the Code

Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 6: Open the Serial Monitor

After uploading the code, open the Serial Monitor in the Arduino IDE (Tools > Serial Monitor) to view the moisture level information.

Step 7: Verify LED and Servo Motor Operation

When there is moisture (moisture level greater than the threshold), the LED should glow.

Experiment 6: Ultrasonic Sensor Interfacing

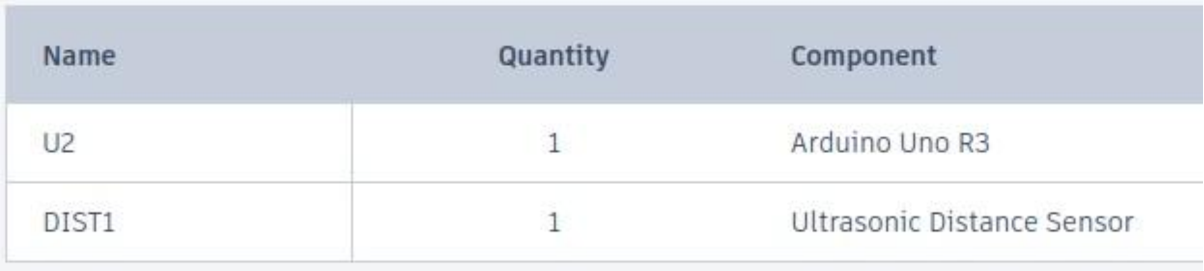
Aim: To measure distance using Ultrasonic sensor

a)print distance in serial monitor

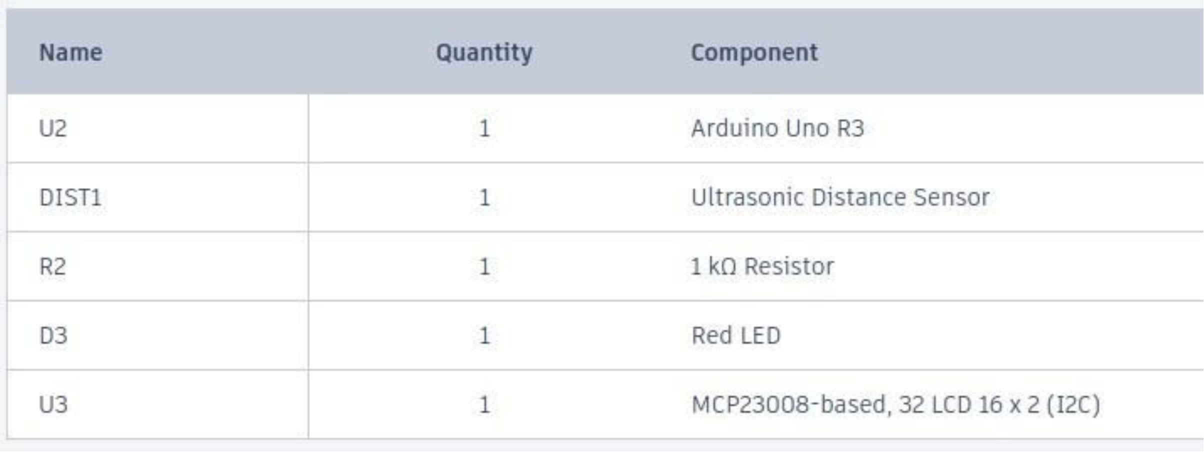
b)print distance in lcd display and glow led when distance is less

Software Used: Arduino IDE

Apparatus:



Name	Quantity	Component
U2	1	Arduino Uno R3
DIST1	1	Ultrasonic Distance Sensor

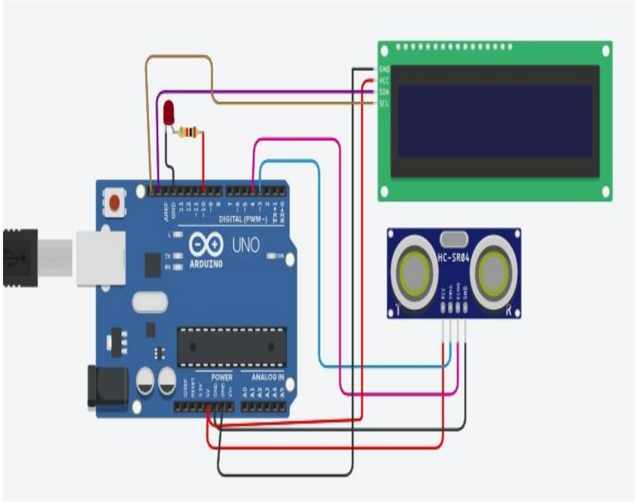


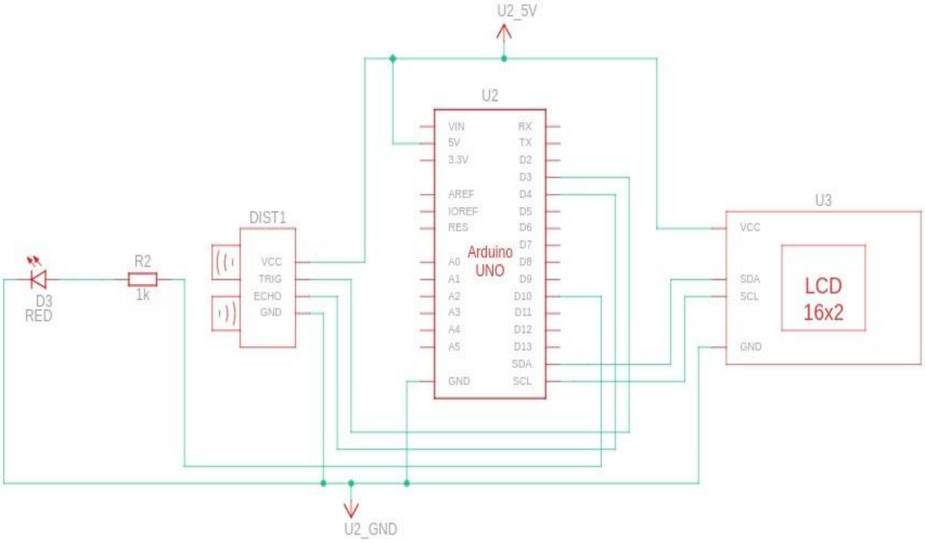
Name	Quantity	Component
U2	1	Arduino Uno R3
DIST1	1	Ultrasonic Distance Sensor
R2	1	1 kΩ Resistor
D3	1	Red LED
U3	1	MCP23008-based, 32 LCD 16 x 2 (I2C)

```
#include <LiquidCrystal_I2C.h>
const int trigPin = 3;
const int echoPin = 4;
int led=10;
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(led,OUTPUT);
```

```
 lcd.begin(16,2);
 lcd.print("Distance(inches and
cm");
lcd.setCursor(0, 1);
 lcd.backlight();
void loop() {
delay(1000);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
float duration = pulseIn(echoPin, HIGH);
float inches = duration * 0.0133 / 2;
float cm =
duration * 0.034 /
2;
if ( inches
 lcd.setCursor(0, 1);
lcd.print("        ");
lcd.setCursor(0, 1);
lcd.print(inches);
lcd.setCursor(8, 1);
lcd.print("     ");
lcd.setCursor(8, 1);
lcd.print(cm);
  delay(1000);

}
```





Procedure:

Step 1: Connect the Ultrasonic Sensor Connect the ultrasonic sensor to the Arduino:

- **Ultrasonic Sensor:**
- Connect the VCC pin to 5V on Arduino.
- Connect the GND pin to GND on Arduino.
- Connect the Trig pin to digital pin 3 on Arduino.
- Connect the Echo pin to digital pin 4 on Arduino.

Step 2: Connect the LCD Connect the LCD to the Arduino:

- **16x2 LCD:**
- Follow the wiring instructions from the earlier responses for interfacing the 16x2 LCD with Arduino.

Step 3: Connect the LED Connect the LED to the Arduino:

- **LED:**
- Connect the anode (longer leg) to digital pin 10 on Arduino through a resistor (220-330 ohms).
- Connect the cathode (shorter leg) to GND on Arduino.

Step 4: Write the Arduino Code

Write a code to measure distance using the ultrasonic sensor, and display the distance on both the Serial Monitor and the LCD. Additionally, glow the LED when the distance is less than a specified threshold.

Step 5: Upload the Code

Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 6: Open the Serial Monitor

After uploading the code, open the Serial Monitor in the Arduino IDE (Tools > Serial Monitor) to view the distance information.

Step 7: Verify LCD Display and LED Operation

- The LCD should display the distance measured by the ultrasonic sensor.
- The LED should glow when the distance is less than the specified threshold (in this example, less than 10 cm)

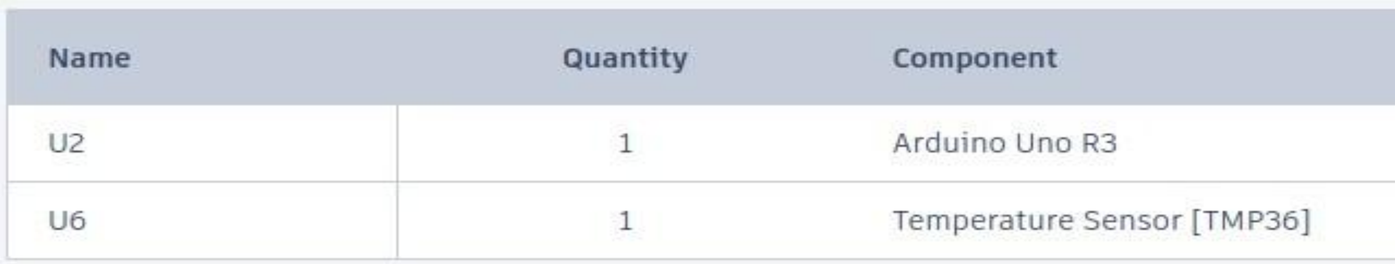
Experiment 7: DHT_11 Sensor Interfacing

Aim: To measure temperature and humidity and print them in a)Serial Monitor

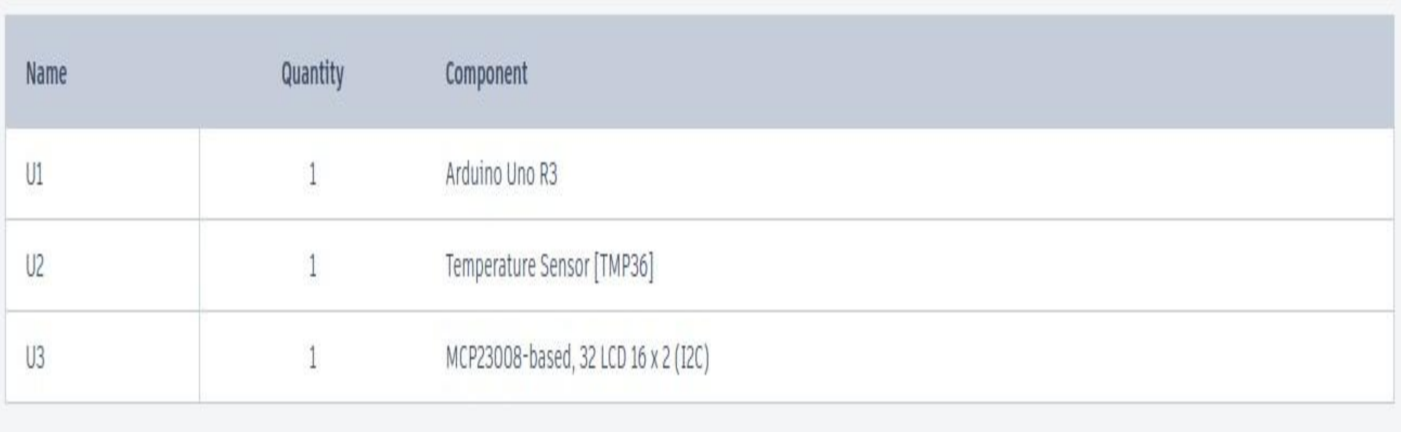
b) LCD

Software Used: Arduino IDE

Apparatus:



Name	Quantity	Component
U2	1	Arduino Uno R3
U6	1	Temperature Sensor [TMP36]

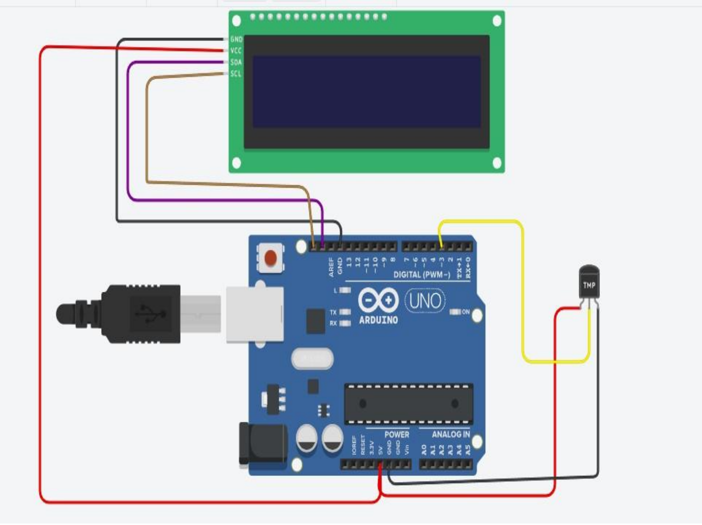


Name	Quantity	Component
U1	1	Arduino Uno R3
U2	1	Temperature Sensor [TMP36]
U3	1	MCP23008-based, 32 LCD 16 x 2 (I2C)

Program:

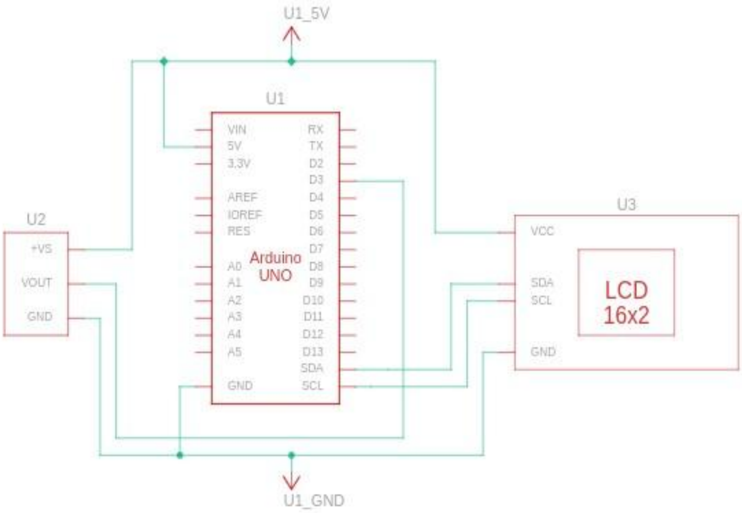
```
#include<DHT.h>
#include<LiquidCrystal_I2C.h>
#define out 2
DHT dht(out, DHT11);
LiquidCrystal_I2C
lcd(0X27,16,2);
void setup() {
  lcd.begin(16,2);
lcd.backlight();
```

Circuit Diagram:



```
dht.begin(); }
void loop() {
delay(2000);
float temperature =
dht.readTemperature();
float humidity =
dht.readHumidity();
lcd.print(temperature);
lcd.print(" °C");
lcd.setCursor(1,2);
lcd.print(humidity);
lcd.println(" %");
}
```

Schematic Diagram:



Procedure:

Step 1: Connect the DHT11 Sensor Connect the DHT11 sensor to the Arduino:

- **DHT11 Sensor:**
- Connect the positive pin (VCC) to 5V on Arduino.
- Connect the negative pin (GND) to GND on Arduino.
- Connect the data pin to digital pin 8 on Arduino.

Step 2: Connect the LCD Connect the LCD to the Arduino:

- **16x2 LCD:**
- Follow the wiring instructions from the earlier responses for interfacing the 16x2 LCD with Arduino.
- Connect the wiper terminal of the potentiometer to the V0 pin on the LCD for contrast adjustment.

Step 3: Write the Arduino Code

Write a code to read temperature and humidity from the DHT11 sensor and display the values on both the Serial Monitor and the LCD.

Step 4: Upload the Code

Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 5: Open the Serial Monitor

After uploading the code, open the Serial Monitor in the Arduino IDE (Tools > Serial Monitor) to view the temperature and humidity information.

Step 6: Verify LCD Display

The LCD should display the temperature and humidity values

Experiment-8: GAS Sensor Interfacing

Aim: Interfacing gas sensor with Arduino UNO.

Software Used: Arduino IDE

Apparatus:

1)Arduino Uno board

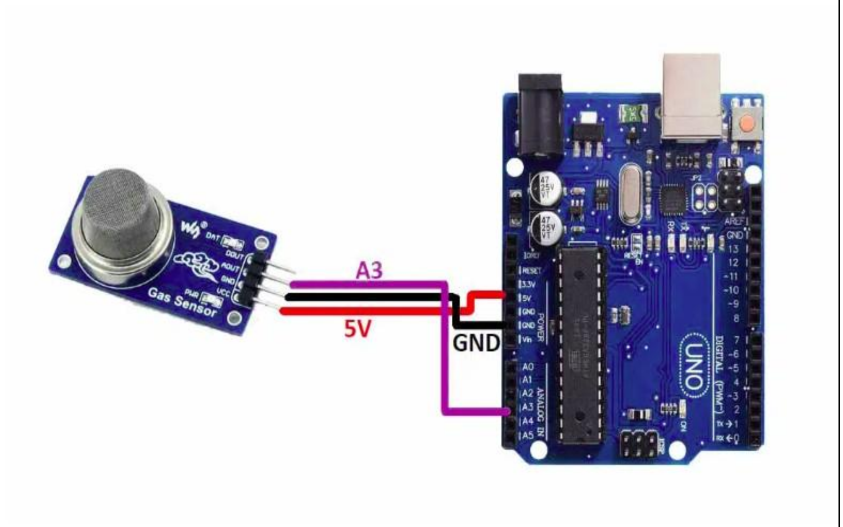
2)Gas Sensor

3) Connecting wires

Program:

```
int gas=A3;
int val;
int del=1000;
void setup()
{
  Serial.begin(9600);
pinMode(gas,INPUT);
} void loop() {
  val=analogRead(gas);
Serial.println(val);
  delay(10000);
}
```

Circuit Diagram:



Procedure:

Step 1: Connect the Gas Sensor

Connect the gas sensor module to the Arduino as follows:

- **Gas Sensor Module:**
- Connect the VCC pin to 5V on Arduino.
- Connect the GND pin to GND on Arduino.
- Connect the AOUT pin to analog pin A3 on Arduino.
- Connect the DOUT pin (if available) to a digital pin on Arduino.

Step 2: Write the Arduino Code

Write a code to read analog values from the gas sensor and print them on the Serial Monitor.

Step 3: Upload the Code

Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

Step 4: Open the Serial Monitor

After uploading the code, open the Serial Monitor in the Arduino IDE (Tools > Serial Monitor) to view the gas sensor readings.

Experiment-9:COLOR Sensor Interfacing

Aim: Interfacing color sensor with arduino uno.

Software Used: Arduino IDE

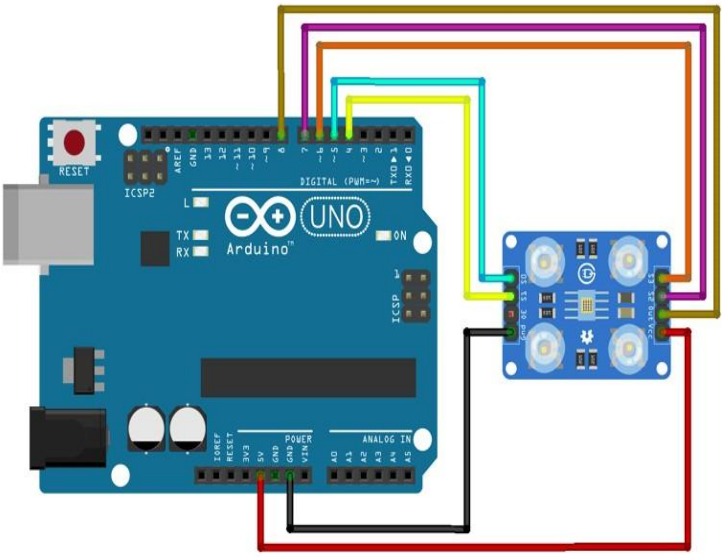
Appatratus:

1)Arduino Uno
2)Color Sensor
3)Connecting Wires

Program:

```
const int S0 = 5;
const int S1 = 4;
const int S2 = 7;
const int S3 = 6;
const int sensorOut = 8;
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
void setup() {
Serial.begin(9600);
pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensorOut, INPUT);
digitalWrite(S0,HIGH);
digitalWrite(S1,LOW);
digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
Serial.println("TCS230 Color Sensor Test");
```

Circuit Diagram



```
}
void loop() {
redFrequency = pulseIn(sensorOut, LOW);
delay(10);
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
greenFrequency = pulseIn(sensorOut, LOW);
delay(10);
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
blueFrequency = pulseIn(sensorOut, LOW);
delay(10);
Serial.print("Red: ");
Serial.print(redFrequency);
Serial.print(" Green: ");
Serial.print(greenFrequency);
Serial.print(" Blue: ");
Serial.println(blueFrequency);
delay(1000);
}
```

Procedure:

Step 1: Connect the Color Sensor

Connect the TCS3200 color sensor to the Arduino as follows:

- **Color Sensor:**
- Connect the S0, S1, S2, and S3 pins on the color sensor to digital pins on Arduino (e.g., 2, 3, 4, 5).
- Connect the OUT pin on the color sensor to digital pin 6 on Arduino.
- Connect the VCC pin on the color sensor to 5V on Arduino.
- Connect the GND pin on the color sensor to GND on Arduino.

- If your sensor has a white LED, connect it to its respective pins following the sensor's datasheet.

  Step 2: Write the Arduino Code

  Write a code to read color data from the TCS3200 color sensor

  Step 3: Upload the Code

  Connect your Arduino Uno to your computer using a USB cable. Select the correct board and port in the Arduino IDE, and click the "Upload" button to upload the code to the Arduino.

  Step 4: Open the Serial Monitor

  After uploading the code, open the Serial Monitor in the Arduino IDE (Tools > Serial Monitor) to view the color sensor readings.

  Step 5: Experiment and Calibrate

  Place different colored objects in front of the color sensor and observe the readings in the Serial Monitor. You may need to calibrate the sensor and adjust the code accordingly for more accurate color detection.

Experiment-9:WiFi module ESP8266 Interfacing

Aim: Interfacing WiFi module ESP8266 with arduino uno.

Software Used: Arduino IDE

Appatratus:

1)Arduino Uno

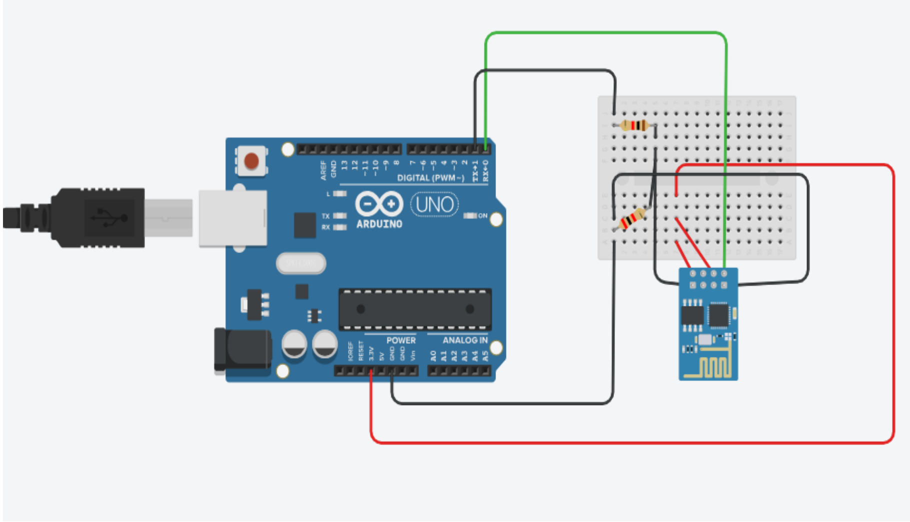
2) ESP8266 Module

3)Connecting Wires

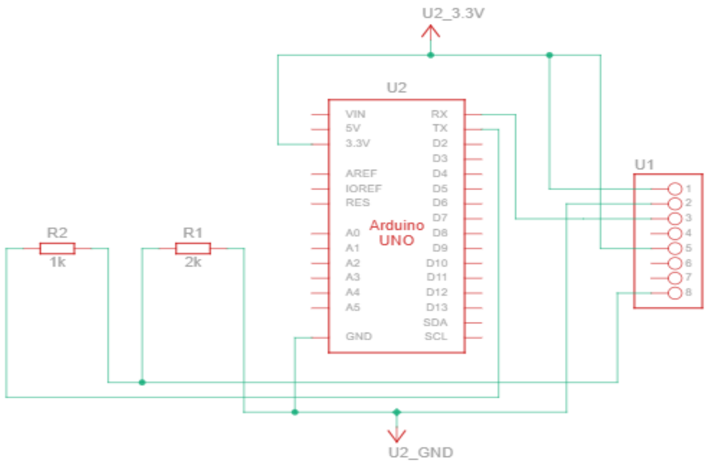
Program

```
#include <SoftwareSerial.h>
SoftwareSerial esp(2, 3);
void setup() {
 Serial.begin(9600);
 esp.begin(9600);
 Serial.println
("ESP8266 WiFi Module
Test");
 esp.println("AT"); }
void loop() {
 if (esp.available()) {
  while (esp.available()) {
     char c = esp.read();
    Serial.write(c);}}
 if (Serial.available()) {
  while (Serial.available()) {
    char c = Serial.read();
    esp.write(c);
  }}}
```

Circuit Diagram



Schematic Diagram:


U2_3.3V
U2
VIN
5V
3.3V
AREF
IOREF
RES
A0
A1
A2
A3
A4
A5
GND
RX
TX
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
SDA
SCL
Arduino
UNO
U1
1
2
3
4
5
6
7
8
R2
1k
R1
2k
U2_GND