Experiment 1: Measuring Temperature:

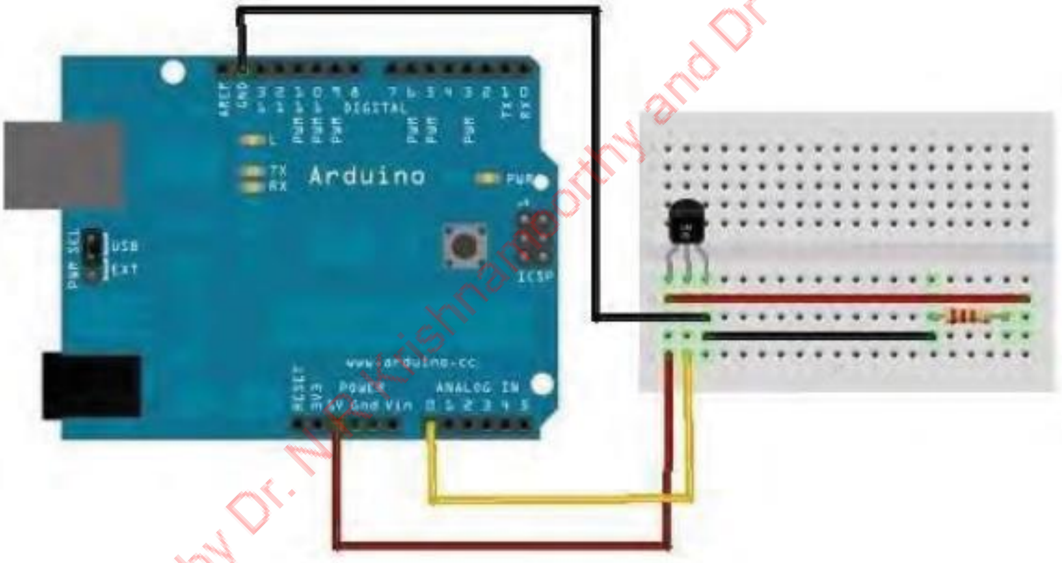
Aim:

To measure the room temperature of a place through Arduino using Lm-35.

Components Used:

1. Resistor - 470Ω
2. Temperature resistor (LM35)
3. Arduino Uno Rev 2
4. Breadboard
5. Connecting wires

Circuit Diagram:



Theory:

Lm-35 is an absolute temperature sensor which can measure the temperature of the surroundings with in 100 to 500 feet. Lm-35 output voltage is proportional to the Celsius/centigrade temperature which increments the output by 1 on every 10-mV change in temperature. Lm-35 can measure from - 50 to 150 degree Celsius.

x Arduino analog pins work normally on +5 volts.

x Resolution of analog pin starts from 0 to 1023.

x Maximum voltage of Lm35 is 1.5 voltage.

x Formula for converting the voltage into system input number =

(V/1024)*5

Procedure:

1. Make connections as per circuit diagram
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to arduino.
5. Now see the temperature through serial monitor

Arduino IDE code to display in serial monitor:

```
const int sensor=A0;
float tempc;
float tempf;
float vout;
void setup() {
pinMode(sensor,INPUT);
Serial.begin(9600);
}
void loop() {
vout=analogRead(sensor);
vout=(vout/1024.0)*5.0;
tempc = (vout - 0.5)*100;
tempf=(tempc*1.8)+32;
Serial.print("in DegreeC=");
Serial.print("\t");
Serial.print(tempc);
Serial.print(" °C ");
Serial.print("in Fahrenheit=");
Serial.print("\t");
Serial.print(tempf);
Serial.println(" °F ");
delay(5000);
}
```

Precautions:

1. Connections should be made properly.
2. Avoid loose connections.
3. Verify the code before uploading.

Result:

Temperature is measured with temperature sensor LM35.

Experiment 2. Distance of the object

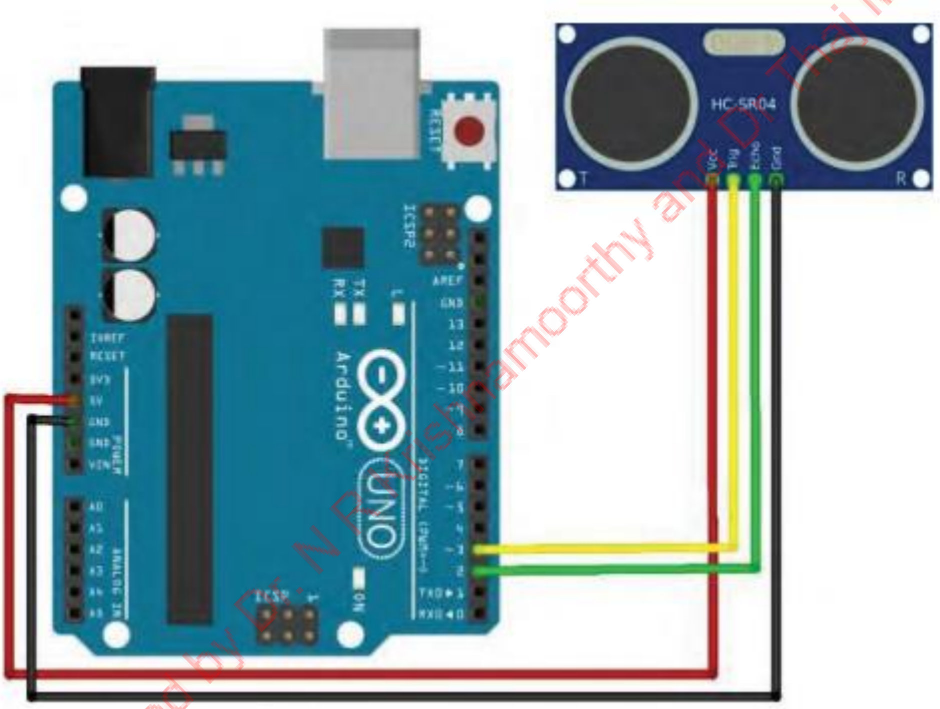
Aim:

To measure the distance of the object using ultrasonic sensor.

Components Used:

1. Ultrasonic sensor
2. Arduino Uno Rev 2
3. Breadboard
4. Connecting wires

Circuit Diagram:



Theory:

The Ultrasonic sensor emits out the very high frequency sound pulse and checks the time taken to reflect the sound back. It has two openings in front one for transmitting and other for receiving the sound waves. The sound travels with the speed of 341 meters per second in air. The time difference between the sending and receiving will help to measure the of the object The mathematical equation: Distance = Time *(speed of sound/2) Ultrasonic sensors will measure the following without any contact with the medium: Distance, Level, Position, Presence and Diameter

Procedure:

1. Make connections as per circuit diagram
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can find out the distance through serial monitor.

Arduino IDE code to display in Serial monitor:

```
const int trigPin = 3;
const int echoPin = 2;
long duration;
int distance;
void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
Serial.print("Distance: ");
Serial.println(distance);
delay(10);
}
```

Precautions:

1. Connections should be made properly.
2. Avoid loose connections.
3. Verify the code before uploading.

Result:

So, the distance of the object is measured with the ultra- sonic sensor.

Experiment 3. Stopwatch in LCD

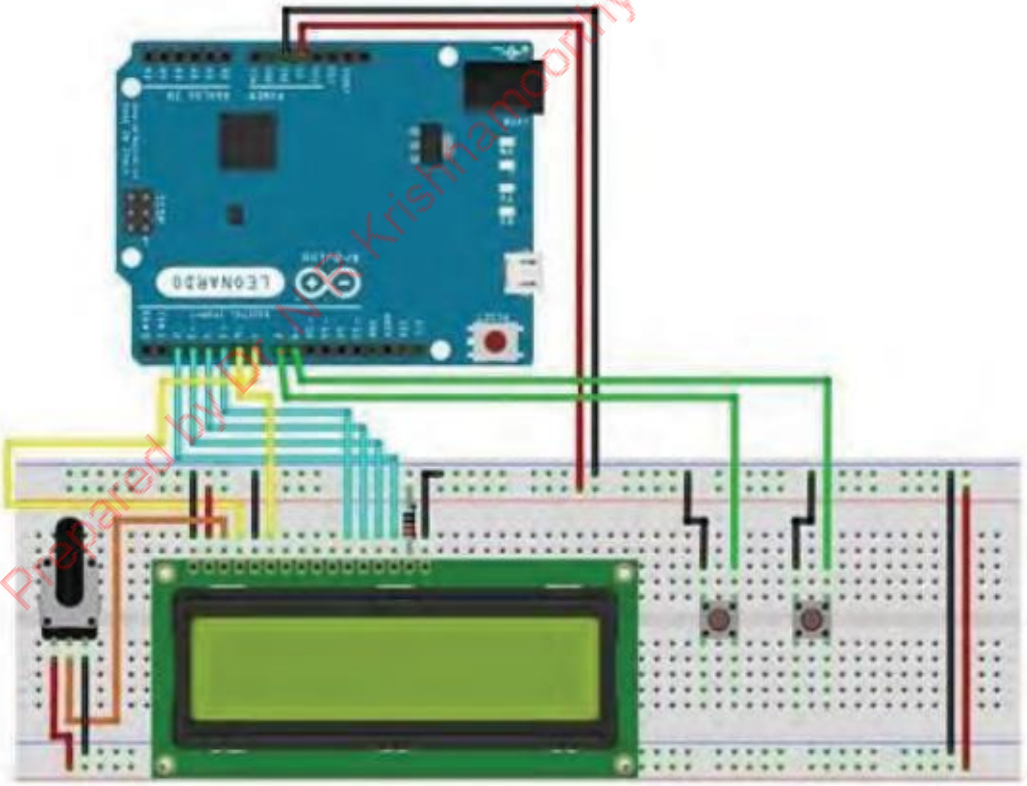
Aim:

To display stopwatch through LCD and control the stopwatch through start/stop button.

Components Used:

1. LCD Screen
2. 220Ω Resistor -(1)
3. 10kΩ potentiometer - (1)
4. Arduino Uno (any board)
5. Breadboard
6. Connecting wires
7. Push button – 2

Circuit Diagram:



Theory:

We come across LCD displays all around the world. Laptops, mobiles calculators, T.V, digital watches use display to display the digits. An L.C.D uses the liquid crystal to show us the visible image which Is an electronic device. In circuits we generally use 16*2 LCD display which is most used. The 16 characters are displayed in 2 lines, so we call it as a 16*2 display. Each character is displayed in 5*7-pixel matrix. So here we now display the stopwatch with the LCD.

Procedure:

1. Make connections as per circuit diagram.
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can find out the stopwatch in the L.C.D.

Arduino IDE code:

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
double i = 0;
double a = millis();
double c ;
void setup()
{
lcd.begin(16, 2);
lcd.clear();
Serial.begin(9600);
pinMode(8, INPUT);
digitalWrite(8, HIGH);
pinMode(9, INPUT);
digitalWrite(9, HIGH);
}
void loop(){
lcd.clear();
lcd.print("press start");
delay(100);
if(digitalRead(8) == LOW)
{
lcd.clear();
a = millis();
while(digitalRead(9) == HIGH)
{
c = millis();
i = (c - a) / 1000;
lcd.print(i);
lcd.setCursor(7,0);
lcd.print("Sec's");
lcd.setCursor(0,0);
Serial.println(c);
Serial.println(a);
Serial.println(i);
Serial.println(" ..... ");
delay(100);
}
if(digitalRead(9) == LOW)
{
while(digitalRead(8) == HIGH)
{
lcd.setCursor(0,0);
lcd.print(i);
lcd.setCursor(11,0);
lcd.print("");
lcd.setCursor(0,0);
delay(100);
}
}
}
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

The Stopwatch is displayed in the Liquid Crystal display with the start/stop button.

Experiment 4. Traffic lights using LED’s

Aim:

To control the traffic lights by switching LED’s with time.

Components Used:

1.Red, orange, green LED – 1No

2. 470Ω – 3No

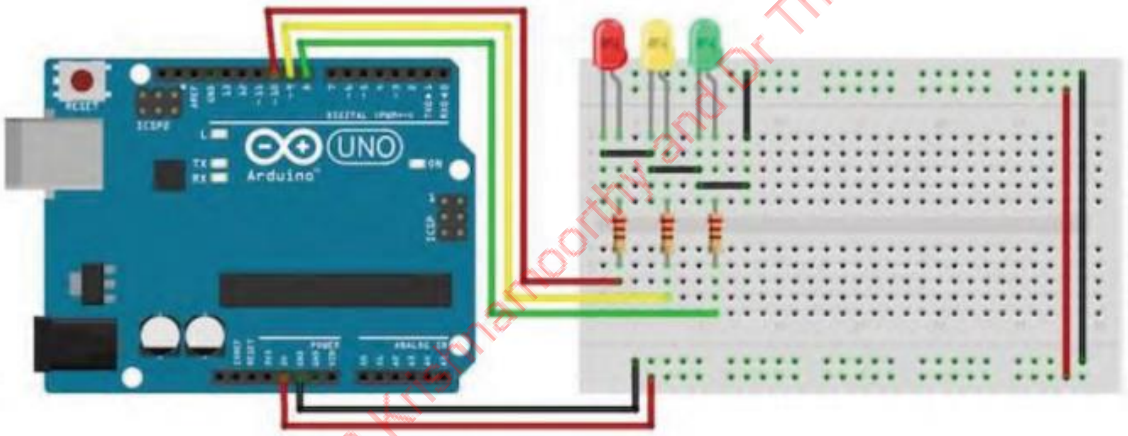
3. Arduino Uno (any board)

4. Push button

5. Breadboard

6. Connecting wires

Circuit Diagram:



Theory:

An L.E.D(Light emitting diode) is a small light which works with very little power. The digital pin 13 is built in the Arduino board for the L.E.D. LEDs have polarity will work only the legs are oriented in the proper way. The two legs of the L.E.D one leg goes for the positive and other leg is for the ground. The L.E.D has flat edge on one side of the bulb. To protect the L.E.D we should use resistors in series to protect it from burning.

Procedure:

1. Make connections as per circuit diagram.

2. Connect the Arduino to pc and configure Arduino IDE.

3. Copy and verify the code in Arduino IDE.

4. Upload the code to Arduino.

5. See the result, three LED are glowing with specified time interval

Arduino IDE code:

```
int red = 10;
int yellow = 9;
int green = 8;
void setup() {
// put your setup code here, to run once:
pinMode(red, OUTPUT);
pinMode(yellow, OUTPUT);
pinMode(green, OUTPUT);
}
void loop(){
// put your main code here, to run repeatedly:
digitalWrite(green, HIGH);
digitalWrite(yellow, LOW);
digitalWrite(red, LOW);
delay(5000);
digitalWrite(green, LOW);
digitalWrite(yellow, HIGH);
digitalWrite(red, LOW);
delay(5000);
digitalWrite(green, LOW);
digitalWrite(yellow, LOW);
digitalWrite(red, HIGH);
delay(5000);
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

Traffic lights with LED’s are achieved with time difference.

Experiment 5. Dark sensing light

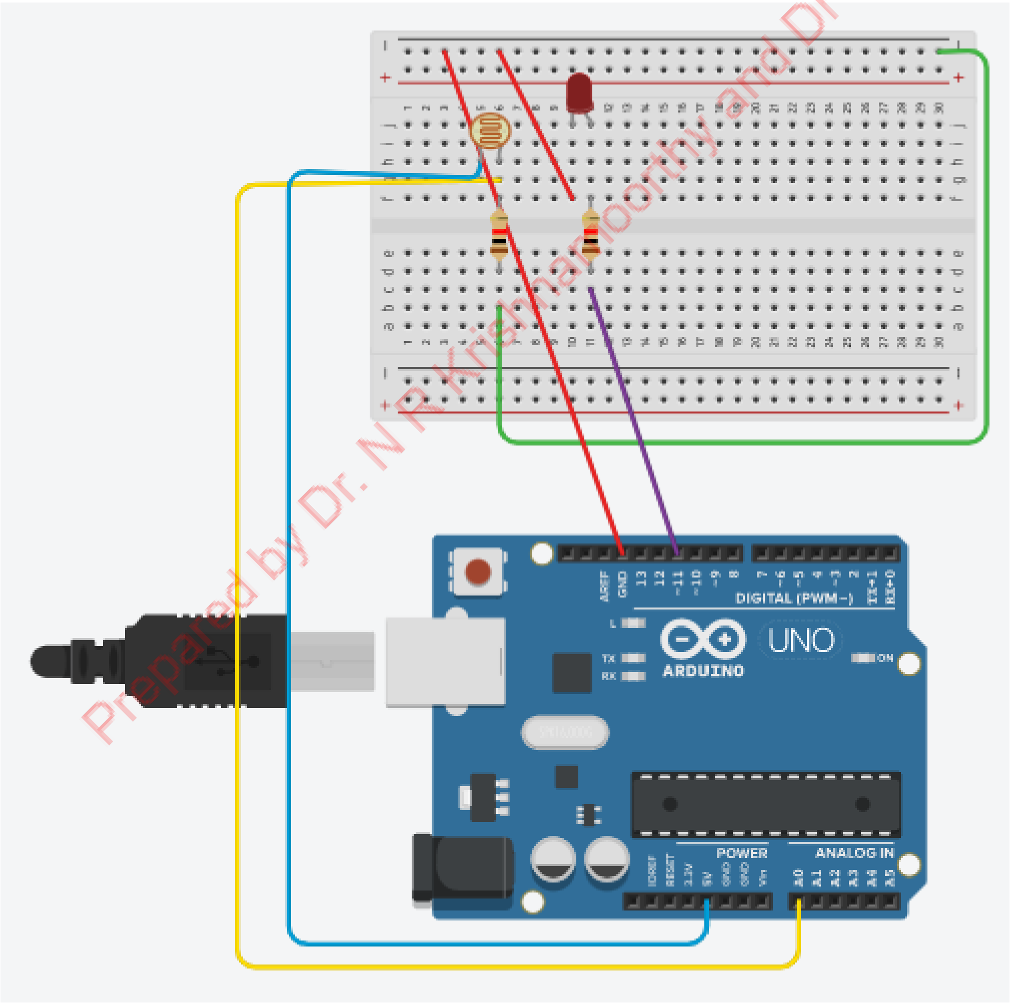
Aim:

To light up an LED automatically if the room has no light present in it.

Components Used:

1. L.E.D
2. Resistors 470Ω - 1
3. Photo Resistor
4. Arduino Uno (any board)
5. Breadboard
6. Connecting wires

Circuit Diagram:



Theory:

An LDR (Light Dependent Resistor) is a component which changes its resistance depending upon the light intensity falling on it. These components are used in light sensing circuits. In most common LDR if the light intensity is more falling on it then the resistance will gradually decrease. The resistance of the LDR have the following

Daylight = 5000Ω

Dark = 20000000Ω

Some of the application of LDR:

1. Lighting Switch 2. Camera shutter control

An L.E.D(Light emitting diode) is a small light which works with very little power. The digital pin 13 is built in the Arduino board for the L.E.D. LEDs have polarity will work only the legs are oriented in the proper way. The two legs of the L.E.D one leg goes for the positive and other leg is for the ground. The L.E.D has flat edge on one side of the bulb. To protect the L.E.D we should use resistors in series to protect it from burning.

Arduino IDE code:

```
int value=0;
void setup()
{
  Serial.begin(9600);
  pinMode(11, OUTPUT);
  pinMode(A0, INPUT);
}
void loop()
{
  value=analogRead(A0);
  if (value<150){
    digitalWrite(11,HIGH);
    Serial.println("Light ON");
    Serial.println(value);
    delay(1000);
  }
  else{
    digitalWrite(11,LOW);
    Serial.println("Light OFF");
    Serial.println(value);
    delay(1000);
  }
}
```

Procedure:

1. Make connections as per circuit diagram.

2. Connect the Arduino to pc and configure Arduino IDE.

3. Design the code with the help of the instructor.

4. Upload the code to Arduino.

5. Now we can see the L.E.D lights up when the light intensity is low.

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

We can see the LED lighting up in the dark.

Experiment 6. LCD Rolling display

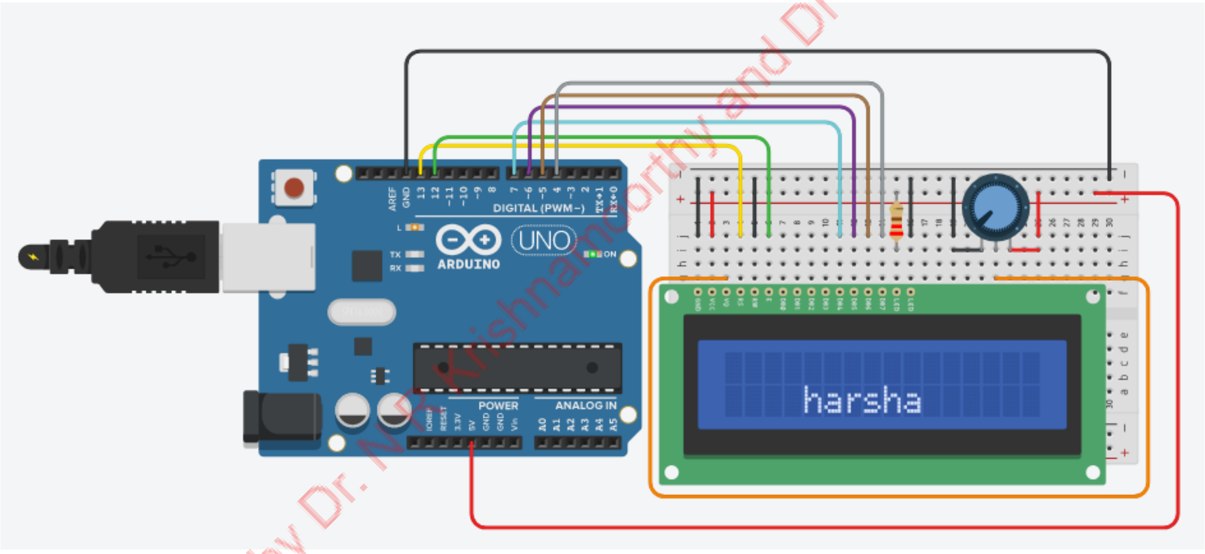
Aim:

To preform rolling of given text in the LCD display

Components Used:

1. L.C.D (16 x 2) - 1
2. Resistors 220Ω - 1
3. Arduino Uno (any board)
4. Breadboard
5. Connecting wires
6. Potentoiometer (250 k ohm) - 1

Circuit Diagram:



Theory:

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

A 16×2 LCD has two registers like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as command register. Similarly, when the register set is ‘1’, then it is known as data register.

Procedure:

1. Make connections as per circuit diagram.
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can see that the text is scrolling in the 16x2 LCD Display.

Arduino IDE code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,7,6,5,4);
int i;
void setup()
{
 lcd.begin(16,2);
}
void loop()
{
 for(i=0 ; i<10 ; i++)
 {
  lcd.setCursor(i,0);
  lcd.print("harsha");
  delay(100);
  lcd.clear();
 }
 for(i=15 ; i>=0 ; i--)
 {
  lcd.setCursor(i,1);
  lcd.print("harsha");
  delay(100);
  lcd.clear();
 }
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

Thus the given text was scrolled in the 16x2 LCD Display using Arduino board

Experiment 7 Pressure Measurement

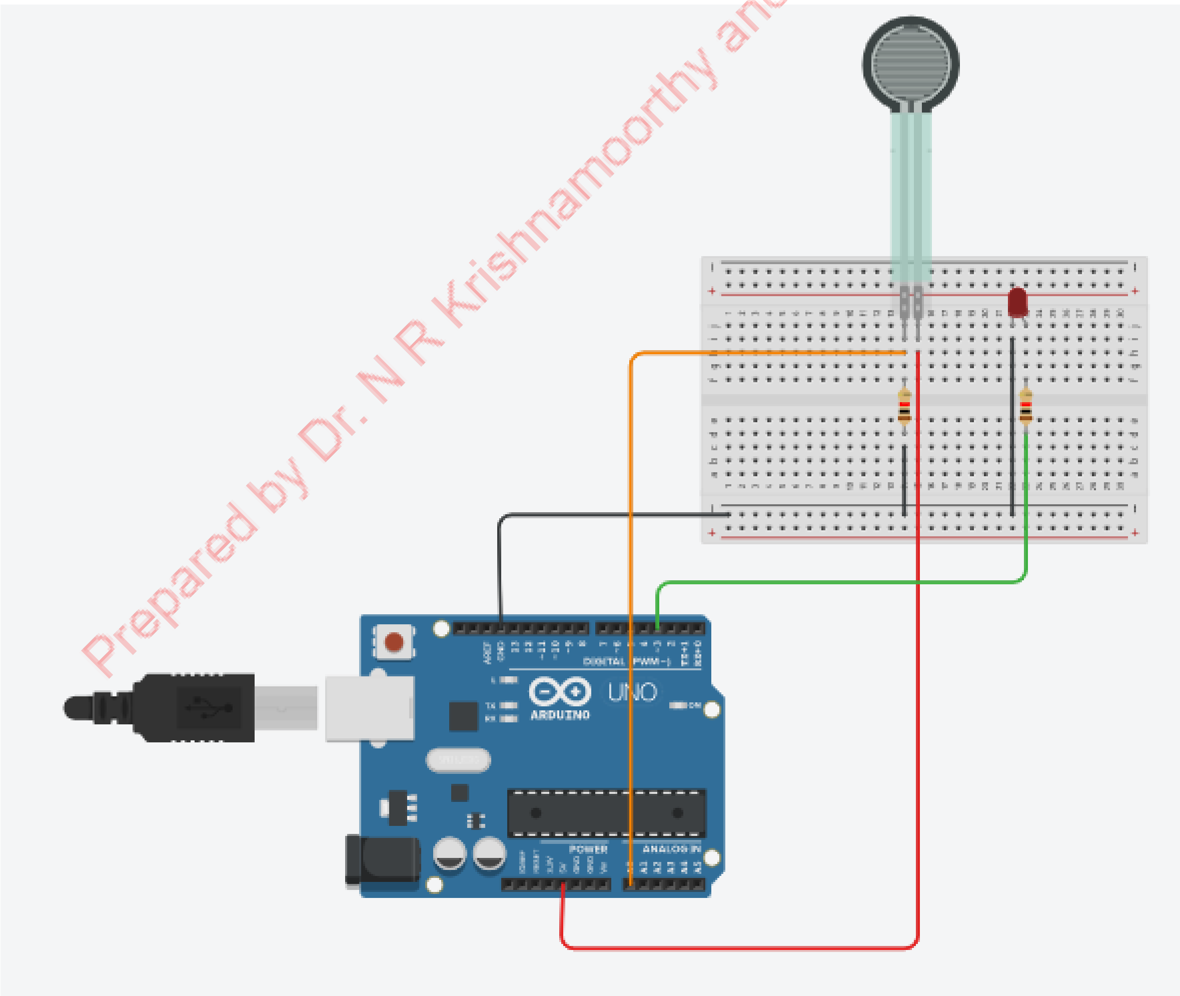
Aim:

To measure and display the pressure by the Arduino using Force sensor

Components Used:

1. Force Sensor - 1
2. Resistors 1kΩ - 2
3. Arduino Uno (any board)
4. Breadboard
5. Connecting wires
6. RED LED - 1

Circuit Diagram:



Theory:

A Force Sensor is a sensor that helps in measuring the amount of force applied to an object. By observing the amount of change in the resistance values of force-sensing resistors, the applied force can be calculated. The general working principle of Force Sensors is that they respond to the applied force and convert the value into a measurable quantity. There are various types of Force Sensors available in the market based on various sensing elements. Most of the Force Sensors are designed using Force-Sensing Resistors. These sensors consist of a sensing film and electrodes. Some of the applications of Force sensor that uses force-sensing resistors includes pressure-sensing buttons, in musical instruments, as car-occupancy sensors, in artificial limbs, in foot-pronation systems, augmented reality,etc…

Procedure:

1. Make connections as per circuit diagram.
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can see the pressure value by the sensor in the serial monitor.

Arduino IDE code:

```
float force_val = 0;
void setup()
{
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  force_val = analogRead(A0);
  if(force_val>=200)
  {
  analogWrite(3,HIGH);
   Serial.println ("Alert !!");
  }else
  {analogWrite(3,LOW);
  }
  Serial.println(force_val);
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

Thus coding was written and uploaded to Arduino and measurement of pressure was carried out successfully.

Experiment 8 Proximity Sensor

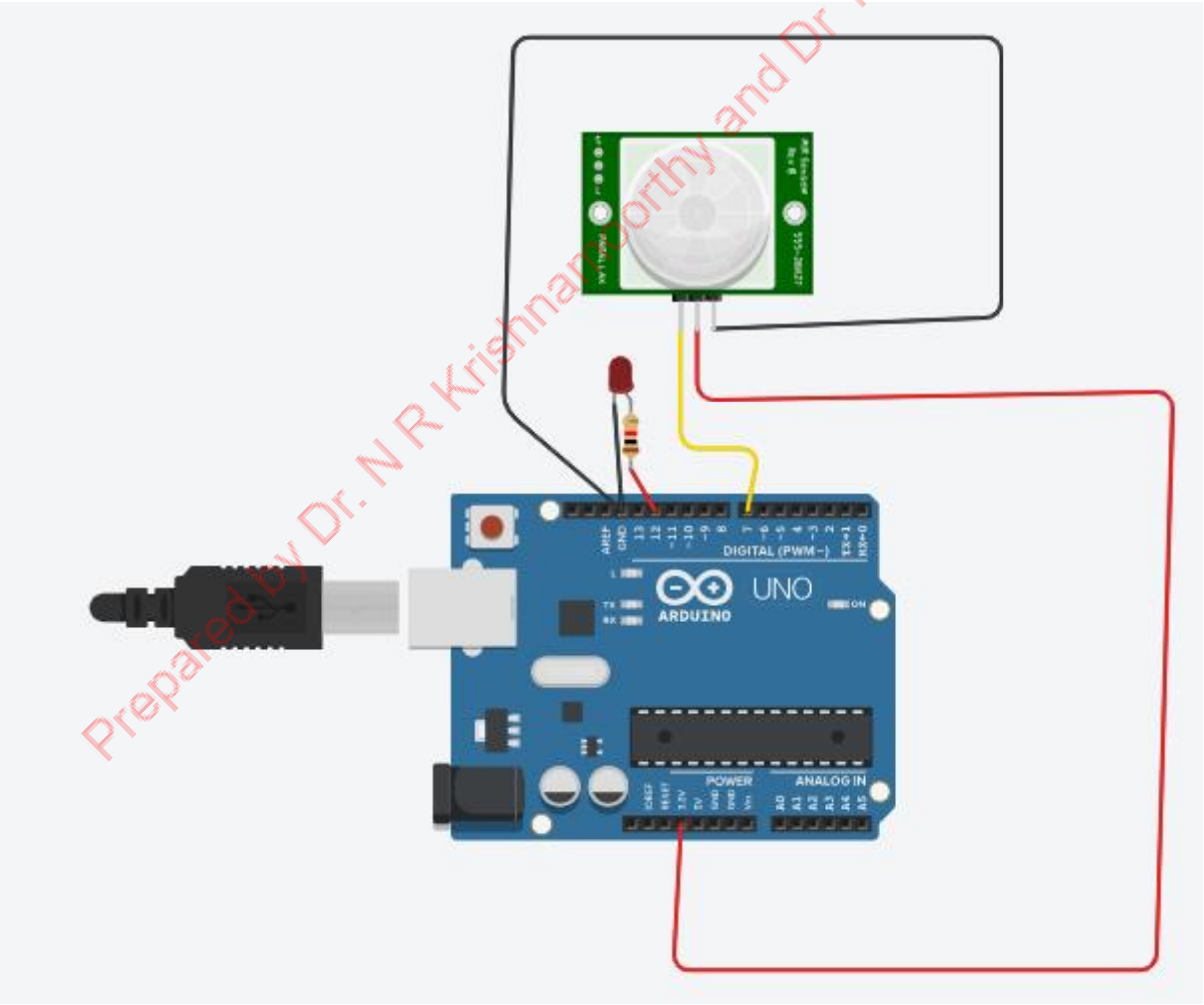
Aim:

To detect or sense the presence of nearby objects by the Arduino using Proximity sensor

Components Used:

1. PIR Sensor - 1
2. Resistors 1kΩ - 1
3. Arduino Uno (any board)
4. Breadboard
5. Connecting wires
6. RED LED - 1

Circuit Diagram:



Theory:

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors

PIRs are basically made of a pyroelectric sensor (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Procedure:

1. Make connections as per circuit diagram.
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can see the presence of a object within specific region.

Arduino IDE code:

```
void setup()
{
 pinMode(7,INPUT);
 pinMode(12,OUTPUT);
}
void loop()
{
 int sensvalue=digitalRead(7);
 if(sensvalue>0)
 {
  digitalWrite(12,HIGH);
  delay(100);
 }
 else
 {
  digitalWrite(12,LOW);
  delay(100);
 }
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

Thus the detection of nearby objects by the Arduino using Proximity sensor was demonstrated successfully

Experiment 9 Soil Moisture Sensor

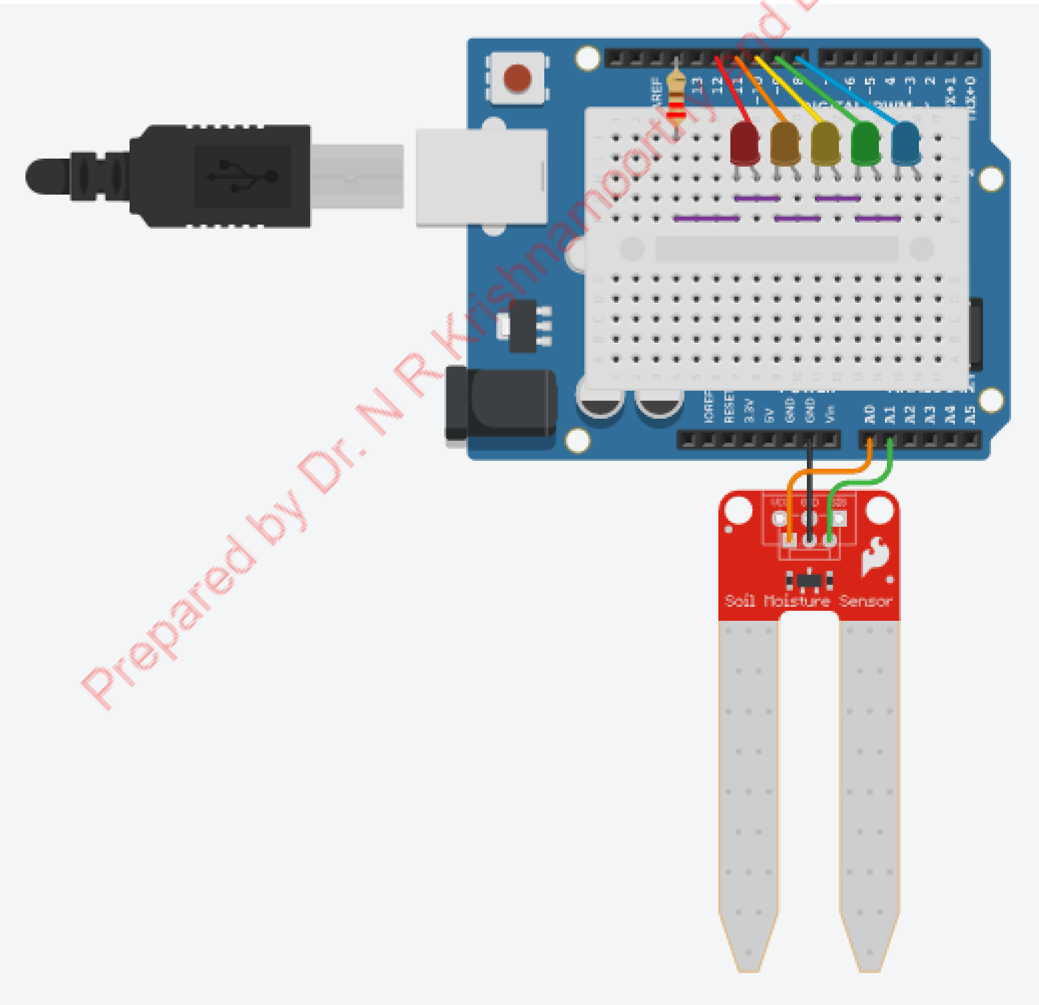
Aim:

To measure or estimate the amount of water in the soil by the Arduino using soil moisture sensor

Components Used:

1. Soil Moisture Sensor - 1
2. Resistors 220 Ω - 1
3. Arduino Uno (any board)
4. Breadboard
5. Connecting wires
6. Red, Orange, Yellow, Green and Blue LED - 1

Circuit Diagram:



Theory:

The soil moisture sensor is one kind of sensor used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content.

The relation among the calculated property as well as moisture of soil should be adjusted & may change based on ecological factors like temperature, type of soil, otherwise electric conductivity. The microwave emission which is reflected can be influenced by the moisture of soil as well as mainly used in agriculture and remote sensing within hydrology. These sensors normally used to check volumetric water content, and another group of sensors calculates a new property of moisture within soils named water potential. Generally, these sensors are named as soil water potential sensors which include gypsum blocks and tensiometer.

Procedure:

1. Make connections as per circuit diagram.
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can see the amount of water in the soil was measured.

Arduino IDE code:

```
int moisture = 0;
void setup()
{
  pinMode(A0, OUTPUT);
  pinMode(A1, INPUT);
  Serial.begin(9600);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}
```

```
void loop()
{
 digitalWrite(A0, HIGH);
 delay(10);
 moisture = analogRead(A1);
 digitalWrite(A0, LOW);
 Serial.println(moisture);
 digitalWrite(8, LOW);
 digitalWrite(9, LOW);
 digitalWrite(10, LOW);
 digitalWrite(11, LOW);
 digitalWrite(12, LOW);
 if (moisture < 200)
 {
  digitalWrite(12, HIGH);
 }
 else
 {
  if (moisture < 400)
  {
   digitalWrite(11, HIGH);
  }
  else
  {
   if (moisture < 600)
   {
    digitalWrite(10, HIGH);
   }
   else
   {
    if (moisture < 800)
    {
     digitalWrite(9, HIGH);
    }
    else
    {
     digitalWrite(8, HIGH);
    }
   }
  }
 }
 delay(100);
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

Thus a arduino program was written to measure the amount of water in the soil by soil moisture sensor

Experiment 10 Water Level Sensor

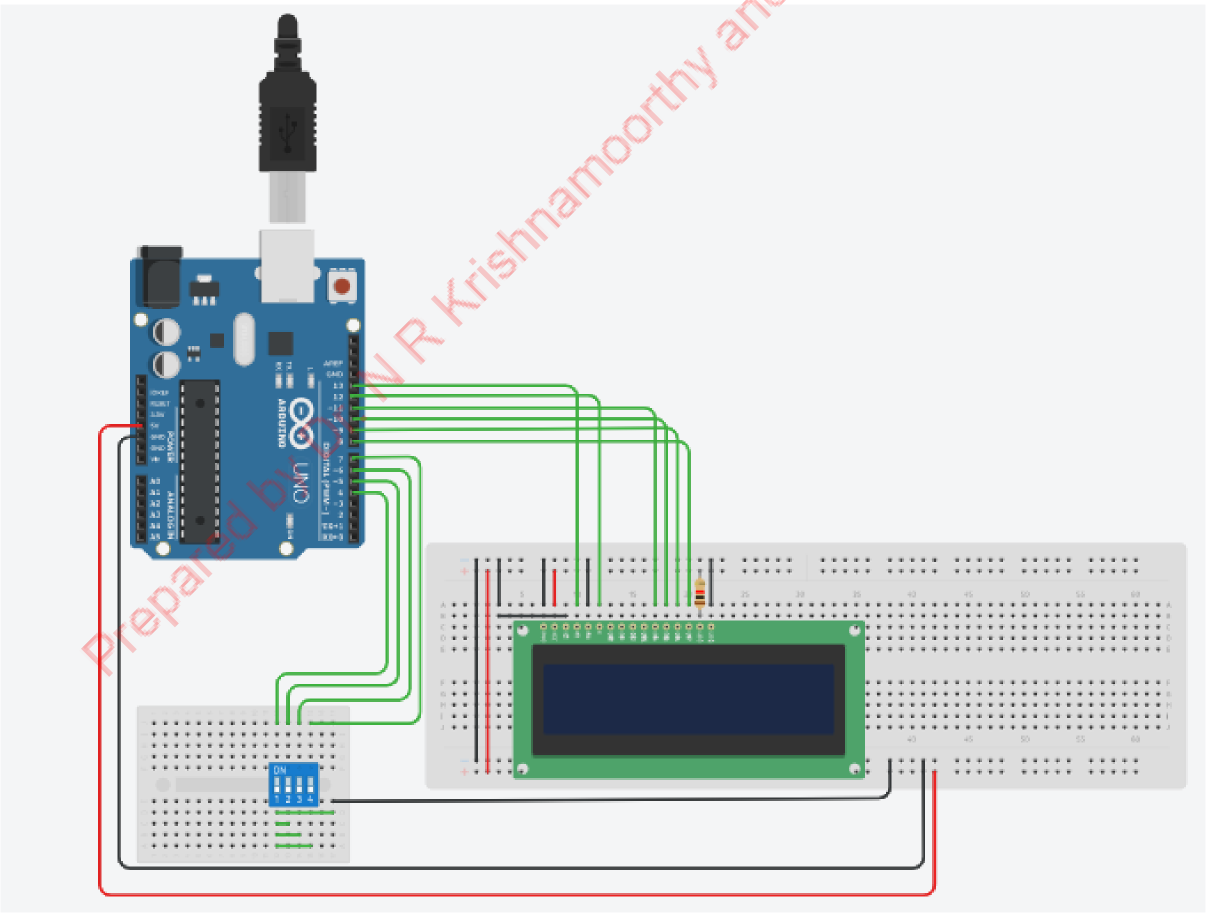
Aim:

To detect the level of substance that can flow by the Arduino using water level sensor

Components Used:

1. DIP Switch SPST x 4 - 1
2. Resistors 1k Ω - 1
3. Arduino Uno (any board)
4. Breadboard
5. Connecting wires
6. LCD 16 x 2 - 1

Circuit Diagram:



Theory:

The water level sensor is a device that measures the liquid level in a fixed container that is too high or too low. According to the method of measuring the liquid level, it can be divided into two types: contact type and non-contact type. The input type water level transmitter we call is a contact measurement, which converts the height of the liquid level into an electrical signal for output. It is currently a widely used water level transmitter.

The working principle of the water level sensor is that when it is put into a certain depth in the liquid to be measured, the pressure on the sensor's front surface is converted into the liquid level height. The calculation formula is P=ρ.g.H+Po, in the formula P is the pressure on the liquid surface of the sensor, ρ is the density of the liquid to be measured, g is the local acceleration of gravity, Po is the atmospheric pressure on the liquid surface, and H is the depth at which the sensor drops into the liquid.

The level sensor is a device designed to monitor and measure liquid (and sometimes solid) levels. When the liquid level is detected, the sensor converts the sensed data into an electrical signal. Level sensors are mainly used for monitoring reservoirs, oil tanks or rivers

Procedure:

1. Make connections as per circuit diagram.
2. Connect the Arduino to pc and configure Arduino IDE.
3. Copy and verify the code in Arduino IDE.
4. Upload the code to Arduino.
5. Now we can see the level of water in the LCD Display.

Arduino IDE code:

```
#include <LiquidCrystal.h>

// Input pins
int pin1 = 4;
int pin2 = 5;
int pin3 = 6;
int pin4 = 7;

// Control vars
int num = 0;
int counter = 0;
bool blink = true;
bool done = false;
```

```
// Block char
byte blockChar[] = {
  B00000,
  B00000,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B00000
};

// Water drop icon
byte iconChar[] = {
  B00100,
  B00100,
  B01110,
  B01110,
  B11111,
  B11111,
  B11111,
  B01110
};

// Lightning bolt icon
/* byte iconChar[] = {
  B00111,
  B01110,
  B01100,
  B11111,
  B00110,
  B01100,
  B01000,
  B10000
}; */

// Empty char
byte clearChar[] = {
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000
};

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
```

```
void setup()
{
  // Set output pins
  lcd.begin(16, 2);
  lcd.createChar(0, blockChar);
  lcd.createChar(1, iconChar);
  lcd.createChar(2, clearChar);
  // Set input pins
  pinMode(pin1, INPUT_PULLUP);
  pinMode(pin2, INPUT_PULLUP);
  pinMode(pin3, INPUT_PULLUP);
  pinMode(pin4, INPUT_PULLUP);
}

void loop()
{
  int max = 0;
  // Read the four "sensors"
  int level25  = digitalRead(pin1);
  int level50  = digitalRead(pin2);
  int level75  = digitalRead(pin3);
  int level100 = digitalRead(pin4);
  // Set the number to display on the BCD display
  if (level25 == HIGH) {
    max = 25;
    if (level50 == HIGH) {
      max = 50;
      if (level75 == HIGH) {
        max = 75;
        if (level100 == HIGH) {
          max = 100;
        }
      }
    }
  }
  int incDec = min(10, abs(num - max));
  if (num < max) num += incDec;
  if (num > max) num -= incDec;
  // Draw only if required
  if (!done) {
    //
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Level " + String(num) + "%");
    //
    lcd.setCursor(0, 1);
    int percent = (16 * num) / 100;
    for (int i = 0; i < percent; i++) {
      lcd.write(byte(0));
    }
    // Set flag
    if (num == max) {
      done = true;
    }
  } else {
    // Reset flag
    if (num != max) {
      done = false;
    }
  }
  // Draw the icon
  lcd.setCursor(15, 0);
  if (blink) {
    lcd.write(byte(1));
  } else {
    lcd.write(byte(2));
  }
  counter++;
  if (counter >= 2) {
    blink = !blink;
    counter = 0;
  }
  // Delay
  delay(250);
}
```

Precautions:

1.Connections should be made properly.

2.Avoid loose connections.

3.Verify the code before uploading.

Result:

Thus the detection of the level of substance that can flow by the Arduino was carried out using water level sensor