

Practical No: 1

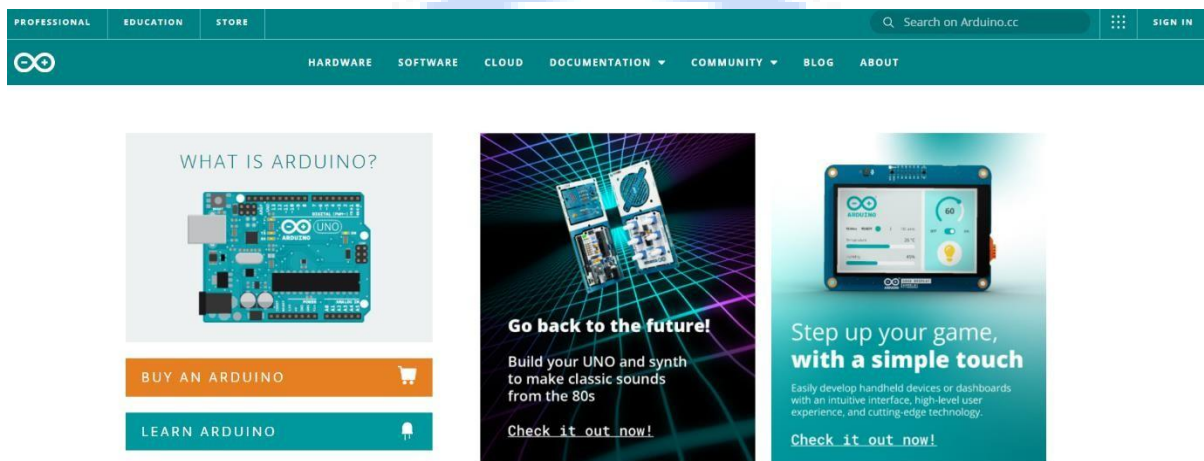
AIM: Installation of Arduino IDE.

Requirements

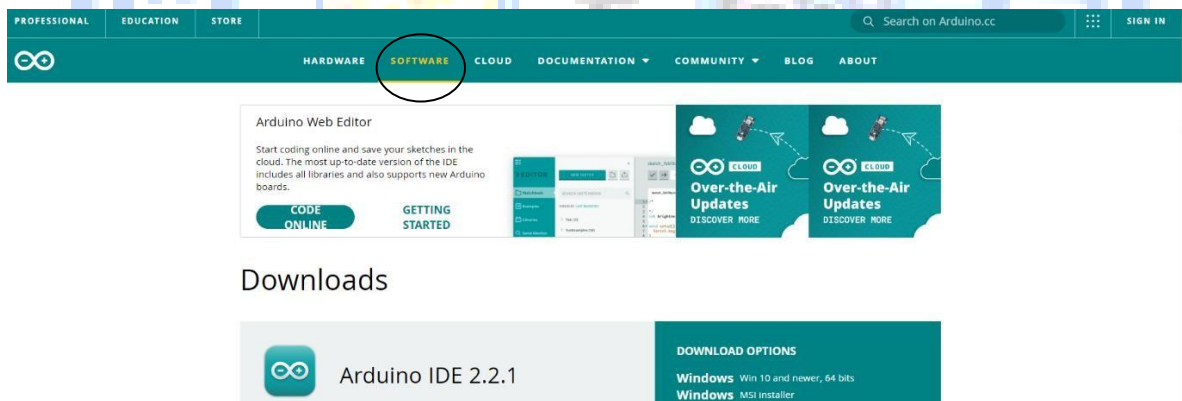
Laptop/ Desktop, Internet.

Procedure

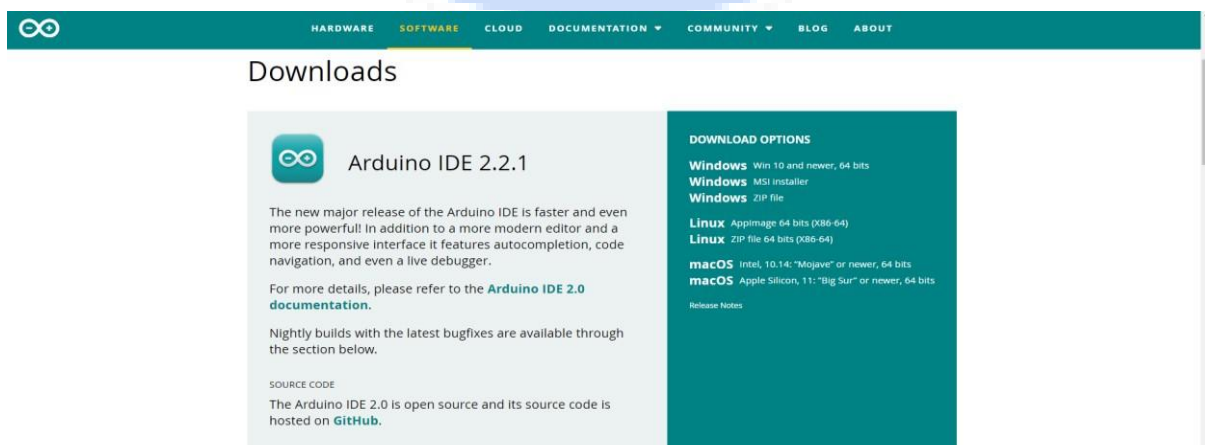
1. First of all, go to the official website of Arduino <https://www.arduino.cc> from your computer

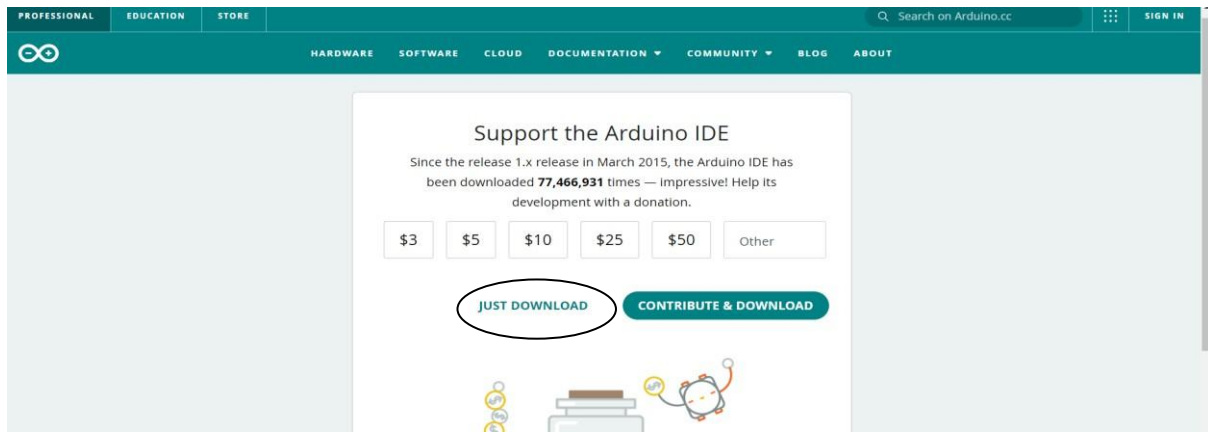


2. After that go to the download page in the software tab

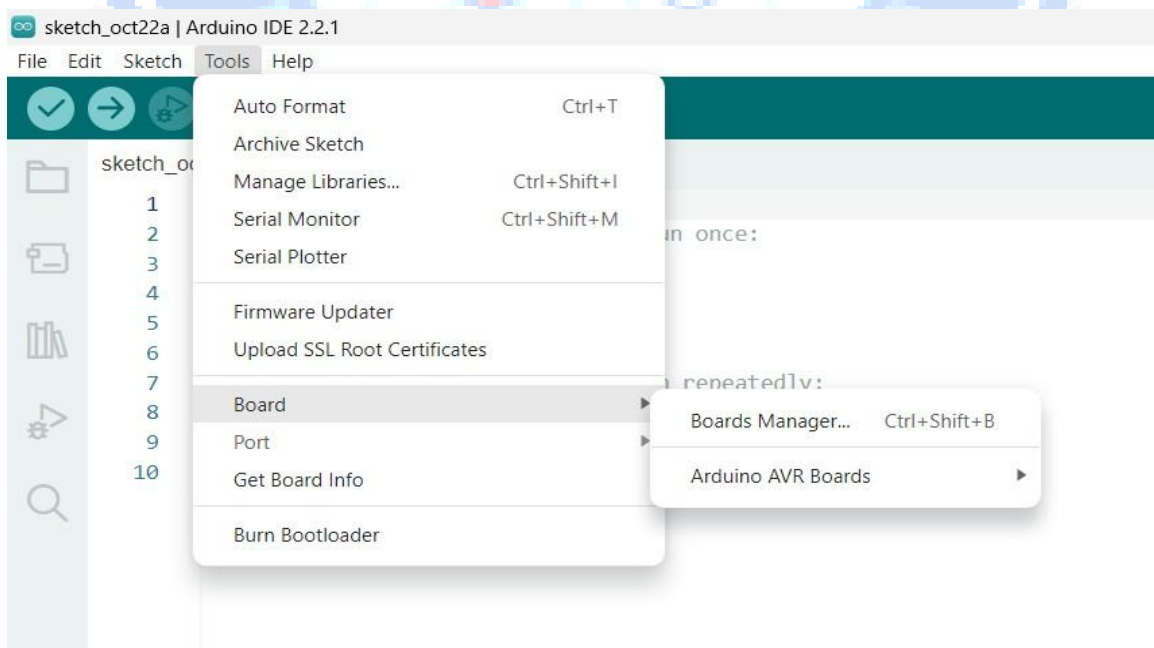


3. Select and download the latest version of Arduino IDE compatible with your system from the option given on the download page





4. Clicking on just download will start downloading the file.
5. Go to the folder in which you have saved this file and install it by double clicking on the arduino-1.8.15-windows file.
6. Click on "I Agree to the terms of the License Agreement."
7. Click on Arduino setup window and Next.
8. Clicking on install on the next page will start the installation.
9. Unzip the downloaded file, save it and open it and install it.
10. After the Arduino IDE software is downloaded, you can start the Arduino IDE by unzipping the Arduino folder and installing it by double clicking the Arduino icon inside it.
11. After the Arduino IDE is open, you have to select your board in it, then by going to Tools → Serial Port, you have to select the port from which the Arduino board is connected to the computer system, it can be COM3 or COM3 or COM2. This is usually reserved for hardware serial ports. To find out, you can disconnect your Arduino board and reopen the menu, the entry that disappears should be that of the Arduino board. Reconnect the board and can select that serial port.



Result In this way Arduino IDE will be installed in your system.

Practical No: 2

AIM :- Interfacing Light Emitting Diode (LED) - Blinking LED.

Requirements

Breadboard, 1 x LED, 1xResistance, Arduino UNO, 2x jumper wires,

Procedure

1. First of all connect the LED to the breadboard.
2. After connecting the LED, connect the first terminal of the Resistance to the positive terminal of the LED. Will connect the other terminal of the resistance to the output on the digital pin (13).
3. Connect the negative terminal of the LED to the ground of the Arduino.
4. After making the connection, we will write a program to make the LED blink in the Arduino IDE.
5. Now with the help of USB port, we will connect Arduino to laptop or PC.
6. After this we will upload the program written in IDE to Arduino. After this the LED will start blinking.

Program Code:-

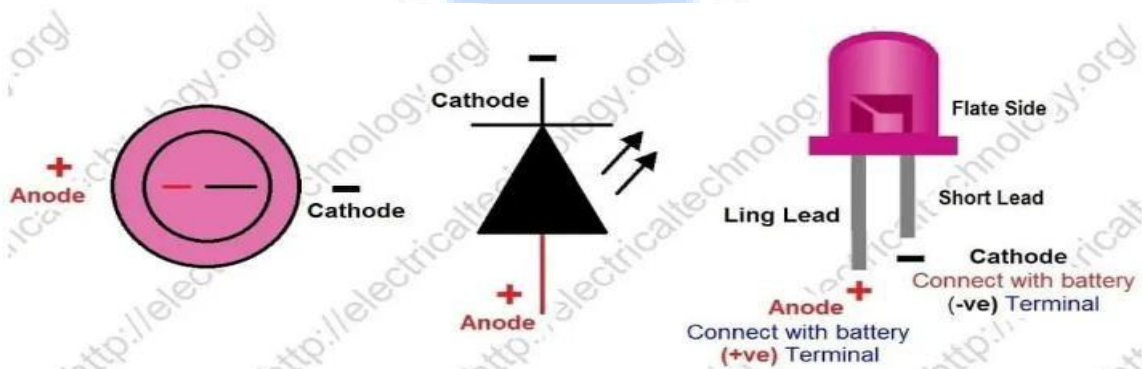
```
int ledpin = 13; // Set pin number for the LED

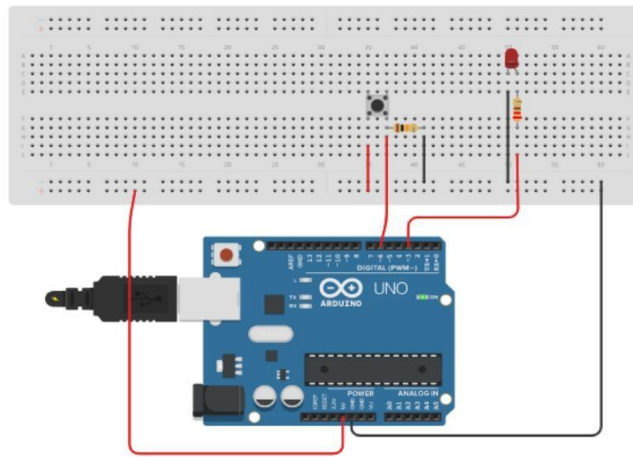
void setup() {
  pinMode(ledpin, OUTPUT); // Initialize digital pin LED_BUILTIN as an
  output
}

void loop() {
  digitalWrite(ledpin, LOW); // Turn the LED off (LOW voltage)
  delay(1000);               // Wait for 1 second (1000 milliseconds)

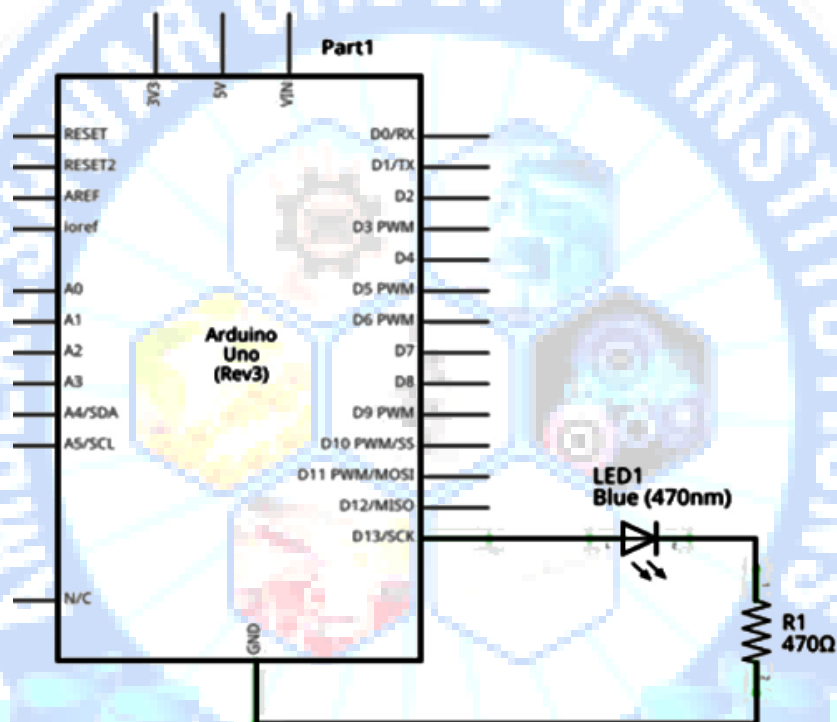
  digitalWrite(ledpin, HIGH); // Turn the LED on (HIGH voltage)
  delay(1000);               // Wait for 1 second
}
```

LED Connection Identification:-





Pin Diagram:



Result:

The LED is blinking when the power supply is turned on.

Practical No: 3

AIM -: Interfacing button and led: led blinking when button is pressed.

Requirements

1x Breadboard, 1xLED, 1xButton, 1x Resistance, 6 x jumper wires.

Procedure

1. First of all we will do LED and Button from breadboard.
2. After connection, the positive terminal of the LED will be connected to the digital pin 13 of the Arduino board for output and the negative terminal will be connected to the ground of the Arduino.
3. Connect one terminal of the button to the digital pin (2) of Arduino for input and connect the other terminal to the power pin of 5V.
4. After this, connect one terminal of resistance to digital pin (2) and connect the other terminal to GND (Arduino).
5. Now connect Arduino to laptop or PC.
6. After uploading the program written in Arduino IDC, we will check the output.

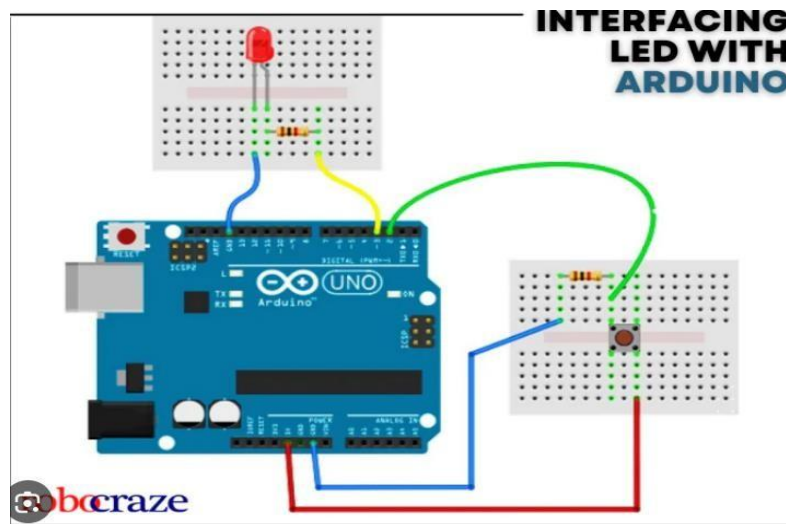
Program Code

```
const int BUTTON = 2;  // Pin number for the button
const int LED = 3;     // Pin number for the LED
int BUTTONState = 0;   // Variable to store button state

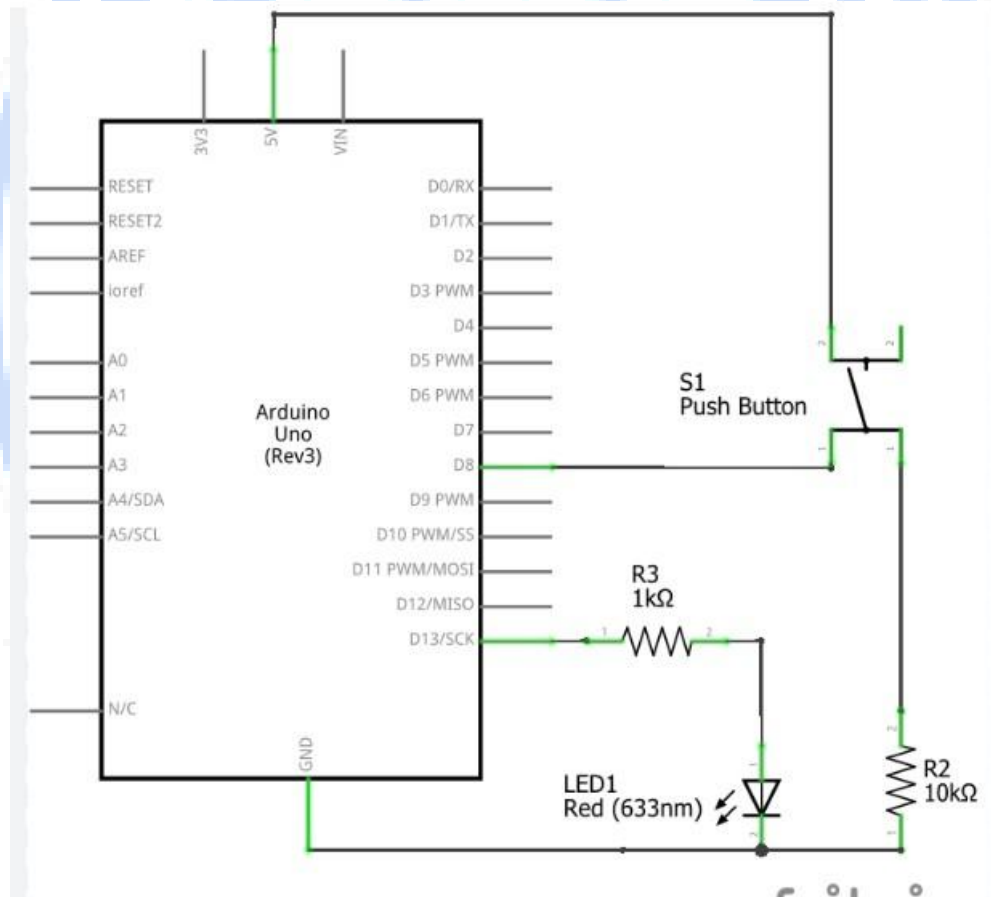
void setup() {
  pinMode(BUTTON, INPUT);  // Set the button pin as input
  pinMode(LED, OUTPUT);    // Set the LED pin as output
}

void loop() {
  BUTTONState = digitalRead(BUTTON);  // Read the button state and store it
  in BUTTONState

  if (BUTTONState == HIGH) { // If the button is pressed
    digitalWrite(LED, HIGH); // Turn the LED on
  } else {
    digitalWrite(LED, LOW);  // Turn the LED off
  }
}
```



Pin Diagram



Practical No: 4

AIM -: Interfacing Light Dependent Resistor (LDR) and LED displaying automatic light lamp.

Requirements

1xLED, 1x220 12 Resistor, 1 x 10 k2 Resistor, breadboard, 6 x jumper wires.

Theory

An LDR (Light Dependent Resistor) is a component that has a variable resistance that changes with the intensity of light falling on it. This allows them to be used in light sensing circuits. A photoresistor is made of a high resistance semiconductor. In the dark, a photoresistor's resistance can be as high as several mega ohms (m2), while in the light, a photoresistor's resistance can be as low as a few hundred ohms. If the light incident on a photoresistor exceeds a certain frequency, the photons absorbed by the semiconductor give enough energy to the bound electrons to jump into the conduction band. The resulting free electrons (and their hole partners) conduct electricity, reducing resistance."

Procedure

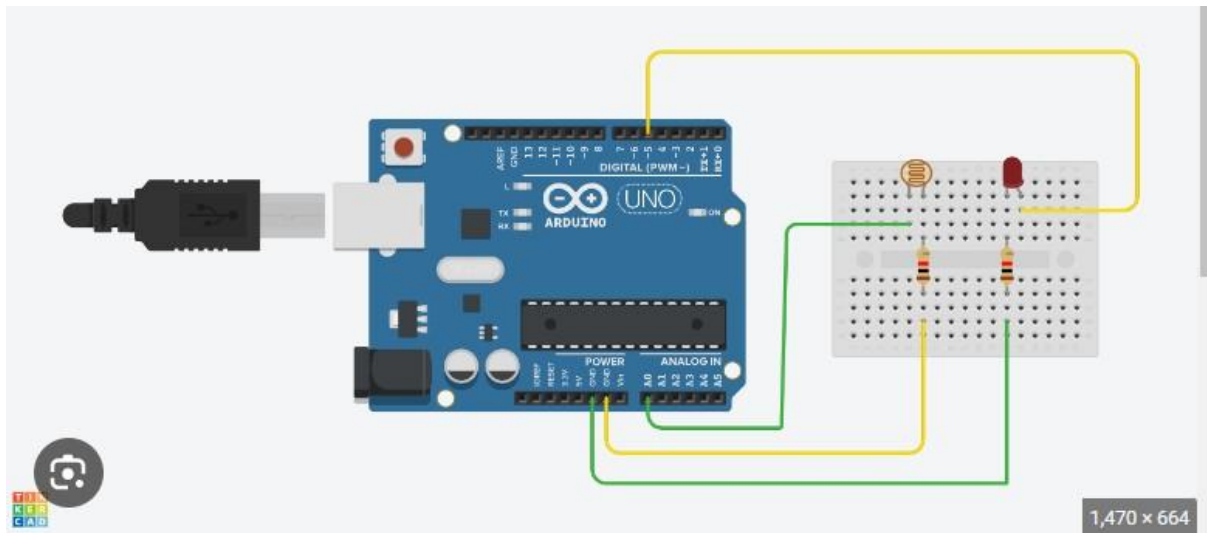
1. First of all connect LDR and LED to breadboard.
2. Connect one terminal of LDR to +5 V pin and connect the other terminal to analog pin A0 of Arduino and connect the other terminal of Resistor to ground.
3. After connection will write program in Arduino IDC for automatic light lamp.
4. Now connect the Arduino board to laptop and upload the program written in arduino IDE to Arduino. After this will check the output.

Program Code

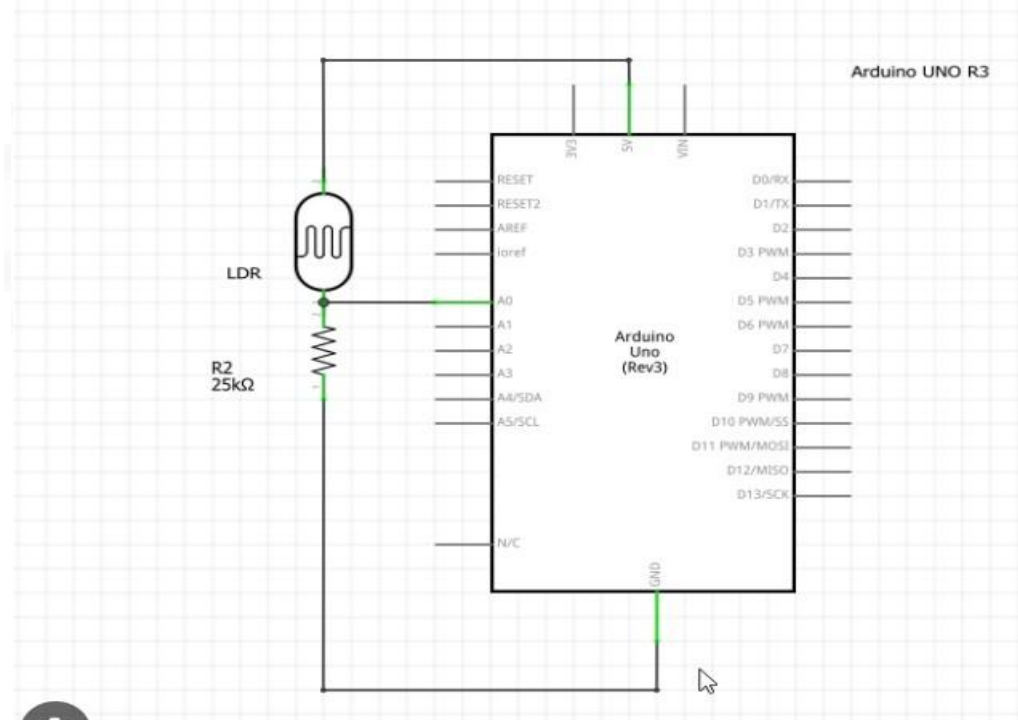
```
int ldr = A0;    // Set A0 (Analog Input) for LDR.
int value = 0;   // Variable to store LDR value.

void setup() {
  Serial.begin(9600); // Start serial communication.
  pinMode(3, OUTPUT); // Set pin 3 as output (LED control).
}

void loop() {
  value = analogRead(ldr); // Read the value from the LDR.
  Serial.println("LDR value is :");
  Serial.println(value);   // Print the LDR value to Serial Monitor.
  if (value < 300) {
    digitalWrite(3, HIGH); // Turn on LED when it's dark.
  } else {
    digitalWrite(3, LOW);  // Turn off LED when it's bright.
  }
  delay(100); // Optional delay for stability.
}
```



Pin Diagram



Result

When there is light on the LDR, the LED turns off and there is no light, the LED turns on.

Practical No: 5

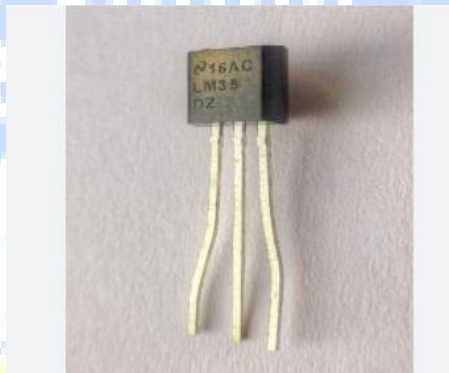
AIM -: Interfacing Temperature Sensor (LM35).

Hardware Required

Temperature Sensor (LM35), Iod display, 1 x Resistor, breadboard, arduino

Theory

LM35 is a temperature sensor which can measure temperature in the range of -55°C to 150°C. It is a 3-terminal device that provides analog voltage proportional to the temperature. Higher the temperature, higher is the output voltage. The output analog voltage can be converted to digital form using ADC so that a microcontroller can process it.



Procedure

1. First of all connected the temperature sensor to arduino board.
2. For this, GND pin of temperature sensor is connected to ground of arduino board, Voc pin of temperature sensor to +5 V pin of arduino and yout pin of sensor to analog pin A1 of arduino board.
3. Will type the program in Arduino Ide.
4. Will upload the program written in Arduino Ide to the arduino board. And will check the output.

Program Code

```
const int lm35_pin = A1; // LM35 output pin connected to A1.

void setup() {
  Serial.begin(9600); // Start serial communication.
}

void loop() {
  int temp_adc_val;    // Variable to store raw ADC value.
  float temp_val;      // Variable to store temperature in Celsius.

  temp_adc_val = analogRead(lm35_pin); // Read the analog value from LM35.
```

```

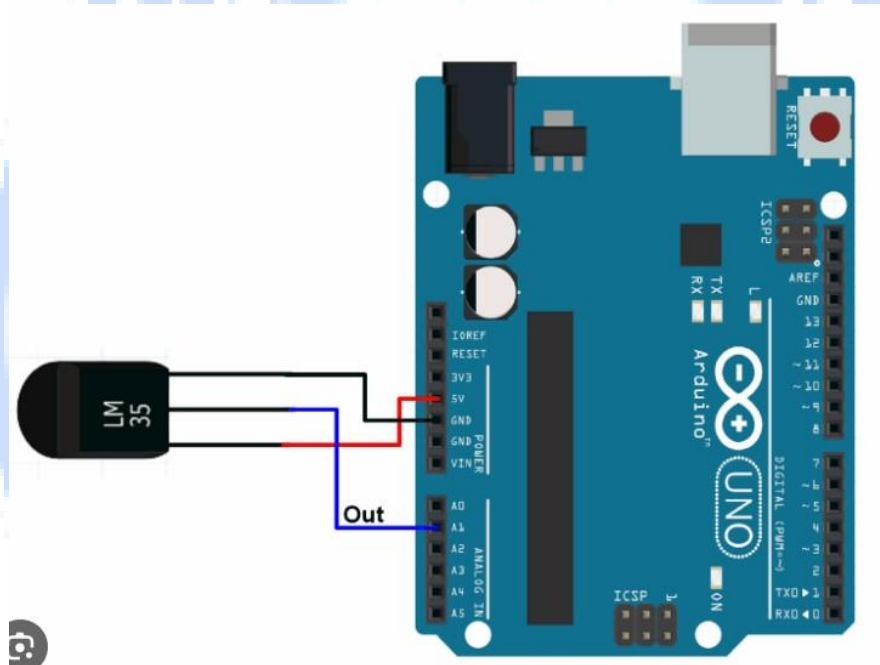
temp_val = temp_adc_val * 4.88;           // Convert ADC value to equivalent
voltage (assuming 5V reference).
temp_val = temp_val / 10;                 // LM35 gives 10mV per degree
Celsius.

Serial.print("Temperature = ");           // Print label for temperature.
Serial.print(temp_val);                   // Print the temperature in
Celsius.
Serial.println(" Degree Celsius");        // Print unit.

delay(1000); // Wait for 1 second before next reading.
}

```

Circuit Diagram



Result

Once uploaded you will see the temperature status like this

Temperature: 29C

Temperature: 28C

Temperature: 99C

Temperature: 27.8C

Practical No: 6

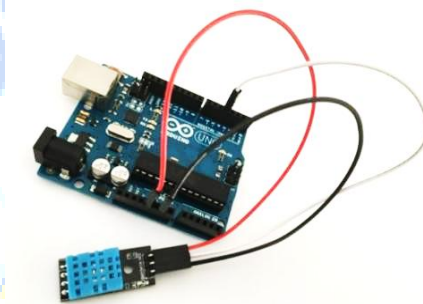
AIM -: Interfacing humidity sensor (DHT11) with arduino.

Requirements

Breadboard, Arduino Uno board, DHT11 (Humidity and Temperature sensor), Jumper wire, DTH library.

Theory

DHT11 sensor is used to measure the temperature and humidity. It has a resistive humidity sensing component and a negative temperature coefficient (NTC). An 8 bit MCU is also connected in it which is responsible for its fast response. It is very inexpensive but it gives values of both temperature and humidity at a time.



Specification of DHT11

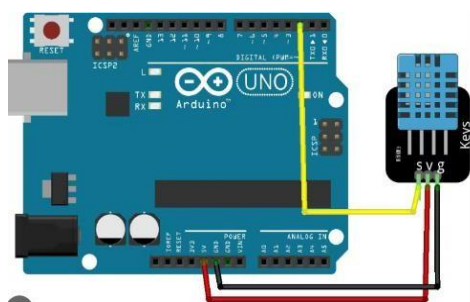
- It has humidity range from 20 to 90%
- RH It has temperature range from 0-50 C
- It has signal transmission range of 20 m
- It is inexpensive
- It has fast response and it is also durable

DHT11 Pin out

- The first pin of the DHT11 is vec pin.
- The second pin of the DHT is Data pin.
- The third pin is not used.
- The fourth pin of the DHT sensor is ground.

DHT11 interfacing with Arduino

First of all, connect the ground and the VCC of the DHT11 temperature and humidity sensor to the ground and 5v of the **Arduino**. Then connect the data pin of the DHT11 sensor to the pin 2 of the Arduino.



Practical No: 7

AIM -: Interfacing Air Quality Sensor pollution (MQ-135)-display data on led, switch on led when data sensed higher than specified value.

Requirement

Arduino board, LCD display, MQ-135 sensor, buzzer & LED (red, green), connecting wires, breadboard, 3 resister.

Theory

The gas sensor module consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it, the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.



Pin Configuration MQ-135 gas sensor

From left to right first pins are as follows:

- | | |
|-------------|-----------------------|
| A0: | Analog output |
| D0: | Digital output |
| GND: | Ground |
| Vcc: | Supply (5V) |