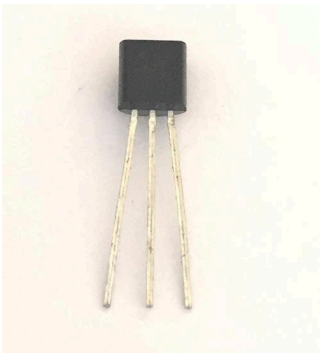
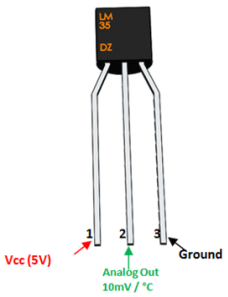
EXPERIMENT NO. 7 (Group B)

- **Aim:** Write a program read the temperature sensor and send the values to the serial monitor on the computer
- **Outcome:** Understanding working principle of DHT11, LM35 temperature sensor, Relationship between different temperature scales
- **Hardware Requirement:** Arduino, LED, LM35, DHT11, etc
- **Software Requirement**: Arduino IDE
- **Theory:**

LM35 Temperature Sensor





LM35 Temperature Sensor Pinout
LM35 Sensor Pinout Configuration

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground of circuit

- **LM35 Sensor Features**
  - Minimum and Maximum Input Voltage is 35V and -2V respectively. Typically 5V.
  - Can measure temperature ranging from -55°C to 150°C
  - Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature.
  - ±0.5°C Accuracy
  - Drain current is less than 60uA

- Low cost temperature sensor
- Small and hence suitable for remote applications
- Available in TO-92, TO-220, TO-CAN and SOIC package

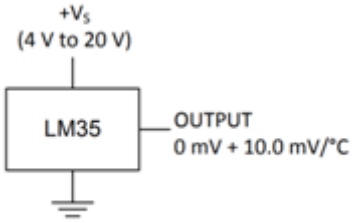
- **LM35 Temperature Sensor Equivalent**

LM34, DS18B20, DS1620, LM94022

How to use LM35 Temperature Sensor:

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.

Power the IC by applying a regulated voltage like +5V (VS) to the input pin and connected the ground pin to the ground of the circuit. Now, you can measure the temperature in form of voltage as shown below.



If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can converted into temperature using the below formulae.

$$V_{OUT} = 10\ mv/°C \times T$$

where

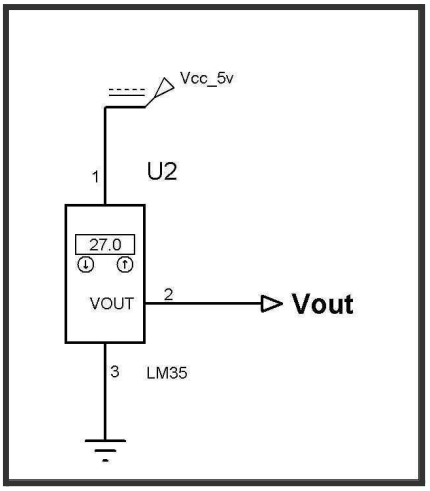
- V_{OUT} is the LM35 output voltage
- T is the temperature in °C

- **LM35 Temperature Sensor Applications**
  - Measuring temperature of a particular environment
  - Providing thermal shutdown for a circuit/component
  - Monitoring Battery Temperature
  - Measuring Temperatures for HVAC applications.

- **How Does LM35 Sensor Work?**

Main advantage of LM35 is that it is linear i.e. 10mv/°C which means for every degree rise in temperature the output of LM35 will rise by 10mv. So if the output of LM35 is 220mv/0.22V the temperature will be 22°C. So if room temperature is 32°C then the output of LM35 will be 320mv i.e. 0.32V.

- **LM35 Interfacing Circuit**



As such no extra components required to interface LM35 to ADC as the output of LM35 is linear with 10mv/degree scale. It can be directly interfaced to any 10 or 12 bit ADC. But if you are using an 8-bit ADC like ADC0808 or ADC0804 an amplifier section will be needed if you require to measure 1°C change.

LM35 can also be directly connected to Arduino. The output of LM35 temperature can also be given to the comparator circuit and can be used for over temperature indication or by using a simple relay can be used as a temperature controller.

Conclusion: -

_____

_____

_____

_____

_____

_____

Experiment no.7

```
// Define the analog pin where the LM35 sensor is connected
const int lm35Pin = A0;
void setup() {
  // Initialize serial communication at 9600 baud rate
  Serial.begin(9600);
}
void loop() {
  // Read the analog value from LM35 sensor
  int sensorValue = analogRead(lm35Pin);

  // Convert the analog reading to voltage (in millivolts)
  float voltage = sensorValue * (5000.0 / 1023.0);

  // Convert the voltage to temperature in degrees Celsius
  float temperatureC = voltage / 10.0;
if (temperatureC >= -50 && temperatureC <= 150) {
     // Print the temperature to the Serial Monitor
     Serial.print("Temperature: ");
     Serial.print(temperatureC);
     Serial.println(" °C");
   } else {
     // Handle unreasonable temperature readings
     Serial.println("Error: Unreasonable temperature reading.");
   }
  // Delay for 1 second before taking the next reading
  delay(4000);
}
```