Aim:- Write code to read temperature values from the sensor.

1.Theory

The Temperature Sensor LM35 series are precision integrated-circuit temperature devices

with an output voltage linearly proportional to the Centigrade temperature.

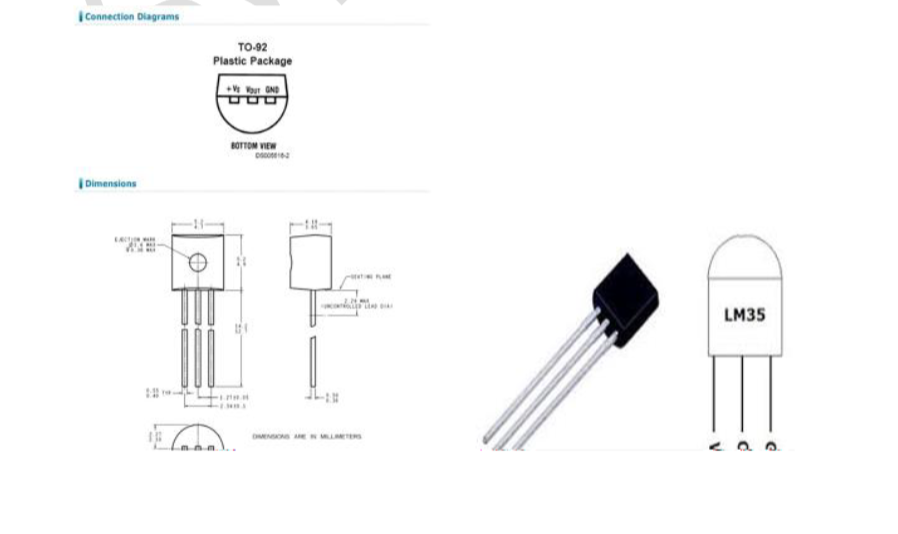
The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin,

as the user is not required to subtract a large constant voltage from the output to obtain

convenient Centigrade scaling. The LM35 device does not require any external

calibration or trimming to provide typical accuracies of ±¼°C at room temperature and

±¾°C over a full −55°C to 150°C temperature range.



DHT11/DHT22 Sensor

The DHT11 and DHT22 are digital sensors used to measure temperature and humidity. The

DHT22 is a more accurate and wider-ranging version of the DHT11. Both sensors provide a

digital output, which can be read by a microcontroller like the Arduino.

Arduino

The Arduino is a microcontroller platform used for interfacing with various electronic

components. It processes sensor data and can control outputs based on sensor inputs.

3. Materials Required

 Arduino Uno board

 DHT11 or DHT22 sensor

 10k-ohm resistor (for DHT22)

 Breadboard

 Jumper wires

 USB cable for Arduino

 Arduino IDE

4. Procedure

1. Connect the Sensor:

1. DHT11 Sensor Connections:

o Connect the VCC pin of the DHT11 sensor to the 5V pin on the Arduino.

o Connect the GND pin of the DHT11 sensor to the GND pin on the Arduino.

o Connect the DATA pin of the DHT11 sensor to digital pin 2 on the Arduino.

2. DHT22 Sensor Connections:

o Connect the VCC pin of the DHT22 sensor to the 5V pin on the Arduino.

o Connect the GND pin of the DHT22 sensor to the GND pin on the Arduino.

o Connect the DATA pin of the DHT22 sensor to digital pin 2 on the Arduino.

o Place a 10k-ohm resistor between the VCC and DATA pins of the DHT22 sensor

(for pull-up functionality).

2. Install the DHT Sensor Library:

 Open the Arduino IDE.

 Navigate to Sketch &gt; Include Library &gt; Manage Libraries.

 Search for &quot;DHT sensor library&quot; and install it by Adafruit.

3. Write and Upload the Code:

Arduino Sketch

cpp

Copy code

#include &lt;DHT.h&gt;

// Define the pin connected to the DHT sensor

#define DHTPIN 2 // Pin where the sensor is connected

// Define the type of DHT sensor used (DHT11 or DHT22)

#define DHTTYPE DHT22 // Change to DHT11 if using DHT11

// Initialize the DHT sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {

// Start serial communication at 9600 baud rate

Serial.begin(9600);

// Initialize the DHT sensor

dht.begin();

}

void loop() {

// Wait a few seconds between readings

delay(2000);

// Read humidity and temperature

float humidity = dht.readHumidity();

float temperature = dht.readTemperature(); // Read temperature in Celsius

// Check if any reads failed and exit early

if (isnan(humidity) || isnan(temperature)) {

Serial.println(&quot;Failed to read from DHT sensor!&quot;);

return;

}

// Print the temperature and humidity values to the Serial Monitor

Serial.print(&quot;Temperature: &quot;);

Serial.print(temperature);

Serial.println(&quot; °C&quot;);

Serial.print(&quot;Humidity: &quot;);

Serial.print(humidity);

Serial.println(&quot; %&quot;);

}

4. Open the Serial Monitor:

 After uploading the code, open the Serial Monitor from Tools &gt; Serial Monitor in the

Arduino IDE.

 Set the baud rate to 9600 to match the Serial.begin(9600) setting in the code.

5. Image



6. Working

Sensor Operation:

The DHT sensor measures temperature and humidity and sends this data to the Arduino as a

digital signal. The Arduino processes this signal and uses the DHT library to interpret the data.

Arduino Code Operation:

The code initializes the sensor and reads temperature and humidity values every 2 seconds. It then prints these values to the Serial Monitor. If the sensor fails to read data, an error message is displayed.

7. Conclusion

The lab successfully demonstrated how to interface a DHT11 or DHT22 sensor with an Arduino to read and display temperature values. Students learned to connect the sensor, install necessary libraries, write a program to read and display data, and view the results on the Serial Monitor. This exercise provides a practical understanding of sensor integration and data acquisition using an Arduino.