HOW TO INTERFACE TEMPERATURE SENSOR AND LCD WITH ARDUINO: Digital Thermometer using LCD and Arduino

Components Used:

- Arduino UNO R3
- LCD Display
- Potentiometer
- Temperature Sensor
- Resistor (1KΩ)
- Breadboard
- Jumper Wires

LCD Display (16x2) :

An **LCD display (16x2)** is commonly used with Arduino to show text and sensor readings. It has **two rows and sixteen columns**, allowing it to display **up to 32 characters**. The display operates using a **Hitachi HD44780 driver**, which communicates with the Arduino via either a **parallel interface (8-bit or 4-bit mode)** or an **I2C module** for reduced wiring.

The Arduino sends data to the LCD using **digital pins**. Each character is formed by turning on and off pixels inside the LCD. The **Liquid Crystal Material** inside the screen changes its opacity when an electric signal is applied, making the text visible. The backlight enhances readability in low-light conditions.

To display text, the Arduino first initializes the LCD using the LiquidCrystal library. It then positions the cursor and prints characters using the lcd.print() function. The LCD continuously updates based on new instructions, allowing it to show dynamic data like **sensor values, clock time, or messages**. This makes it useful for real-time monitoring in various projects.

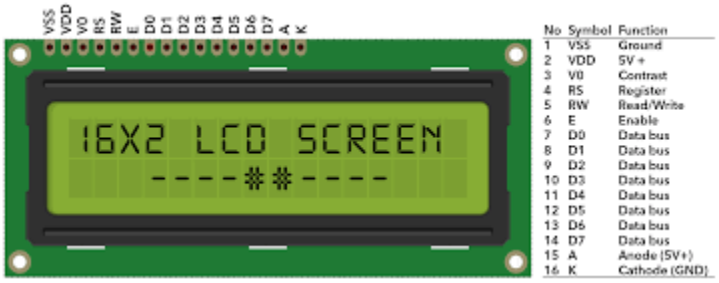

Fig 16.1: 16X2 LCD Display

Temperature Sensor

A **temperature sensor** measures ambient temperature and sends the data to an Arduino. Sensors like **LM35** output an **analog voltage** proportional to temperature, while **DHT11** provides **digital data** for both temperature and humidity.

For the **LM35 sensor**, the Arduino reads the analog voltage from the sensor's output pin using the analogRead() function. Since the LM35 provides **10mV per °C**, the Arduino converts the voltage into temperature using the formula:

Temperature(°C)=Analog Value×5.01024×100

For the **DHT11 sensor**, the Arduino communicates using a **single-wire digital protocol**, receiving temperature and humidity data in a structured format. The DHT library is used to process this data.

The temperature sensor continuously sends updates, allowing real-time monitoring. The measured values can be displayed on an **LCD, Serial Monitor, or IoT-based systems** for further applications such as weather stations, home automation, or safety monitoring.

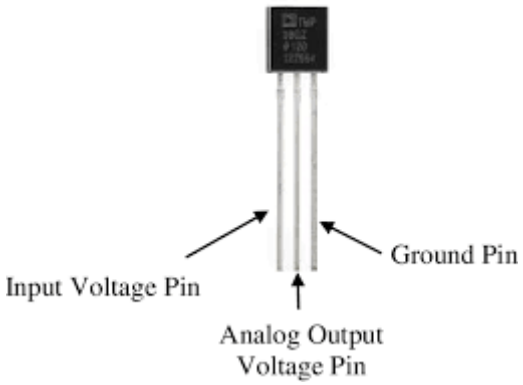

Fig 16.2: Temperature Sensor

Project:

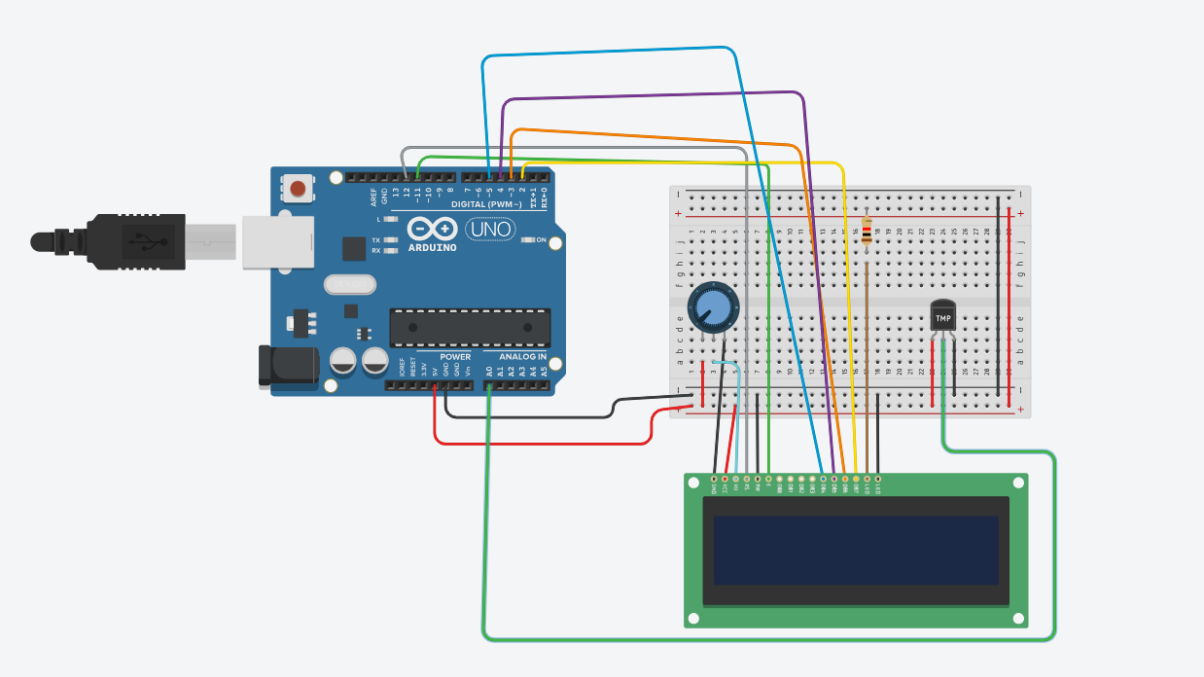

Fig16.3: Digital Thermometer using Arduino

Digital Thermometer Using Arduino and LCD Display

This project is a **digital thermometer** that measures and displays temperature using an **Arduino, an analog temperature sensor, and a 16x2 LCD display**. The system reads temperature data from an **analog temperature sensor** connected to **A0 (analog pin 0)**, processes the value, converts it into **Fahrenheit**, and displays it on the LCD screen.

Pin Connections:

- **LCD (16x2) to Arduino:**
  - RS → **Pin 12**
  - E → **Pin 11**
  - D4 → **Pin 5**
  - D5 → **Pin 4**
  - D6 → **Pin 3**
  - D7 → **Pin 2**
  - VCC & LED+ → **5V**
  - GND & LED- → **GND**
- **Temperature Sensor (LM35) to Arduino:**
  - VCC → **5V**
  - GND → **GND**
  - OUT → **A0**

Signal Flow & Working:

The **temperature sensor** outputs an **analog voltage** proportional to the temperature. The Arduino reads this voltage using analogRead(0), then converts it to temperature in **Celsius** using:

$$T(°C)=(\text{Analog Value}1024\times5V-0.5V)\times100$$

The temperature is then converted to **Fahrenheit** using:

$$T(°F)=95\times T(°C)+32$$

Finally, the calculated temperature is displayed on the **16x2 LCD screen** using lcd.print(). The LCD continuously updates the value in real-time, providing a **simple and efficient digital thermometer** for monitoring temperature changes.