Experiment 2.3

Student Name: Saumyamani Bhardwaz **UID: 20BCS1682**
Branch: BE CSE **Section/Group: 701_A**
Semester: 6th **Date of Performance:**
Subject Name: IoT Lab **Subject Code: 20CSP-358**

1. **Aim:** To display data generated by sensor on LCD using Arduino/Raspberry Pi

2. **Components Required:**
   - BreadBoard
   - Arduino UNO
   - Jumper Wire
   - Connecting Cabel
   - LCD Display
   - Temprature Sensor

3. **Objective:**
   - To Interface sensor with Arduino/Raspberry Pi for data acquisition.
   - To Program Arduino/Raspberry Pi to display sensor data on LCD.

4. **Script:**

About LCD:

A Liquid Crystal Display commonly abbreviated as LCD is basically a display unit built using Liquid Crystal technology. When we build real life/real world electronics-based projects, we need a medium/device to display output values and messages. Other commonly used LCD displays are 20×4 Character LCD, Nokia 5110 LCD module, 128×64 Graphical LCD Display and 2.4-inch TFT Touch screen LCD display.

Interfacing 16×2 LCD to Arduino uno:

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again.

5. Code:

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  // sets the interfacing pins

void setup()
{
 lcd.begin(16, 2);  // initializes the 16x2 LCD
}
void loop()
{
  lcd.setCursor(0,0);                  //sets the cursor at row 0 column 0
  lcd.print("16x2 LCD MODULE");      // prints 16x2 LCD MODULE
  lcd.setCursor(2,1);                //sets the cursor at row 1 column 2
  lcd.print("TEMPRATURE");      // prints TEMPRATURE
}

#include <LiquidCrystal.h>
int vcc=A0;
int sensor=A1;
int gnd=A2;
float temp;
float tempf;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
  pinMode(vcc,OUTPUT);
  pinMode(sensor,INPUT);
  pinMode(gnd,OUTPUT);
  digitalWrite(vcc,HIGH);   // Vcc for LM35
  digitalWrite(gnd,LOW);    // Ground for LM35
  lcd.begin(16, 2);         // initializes the 16x2 LCD
  lcd.setCursor(2,0);       // sets the cursor at column 2 row 0
  lcd.print("TEMPERATURE"); // prints temperature
}
void loop()
{
  temp=analogRead(sensor); // reads the sensor output
  temp=temp*5;             // converts the sensor reading to temperature
  temp=temp/10;            // adds the decimal point
  tempf=(temp*1.8)+32;     // converts to Fahrenheit

  lcd.setCursor(0,1);   // sets cursor at column 0 row 1
  lcd.print(temp);      // prints temperature in degree Celsius
  lcd.print((char)223); // prints degree sign
  lcd.print("C");       // prints letter c
  lcd.setCursor(8,1);   // sets cursor at column 8 row 1
  lcd.print(tempf);     // prints temperature in degree Fahrenheit
  lcd.print((char)223); // prints degree sign
  lcd.print("F");       // prints letter F
  delay(1000);          // 1 second delay
}
```

6. Figure/Screenshots:

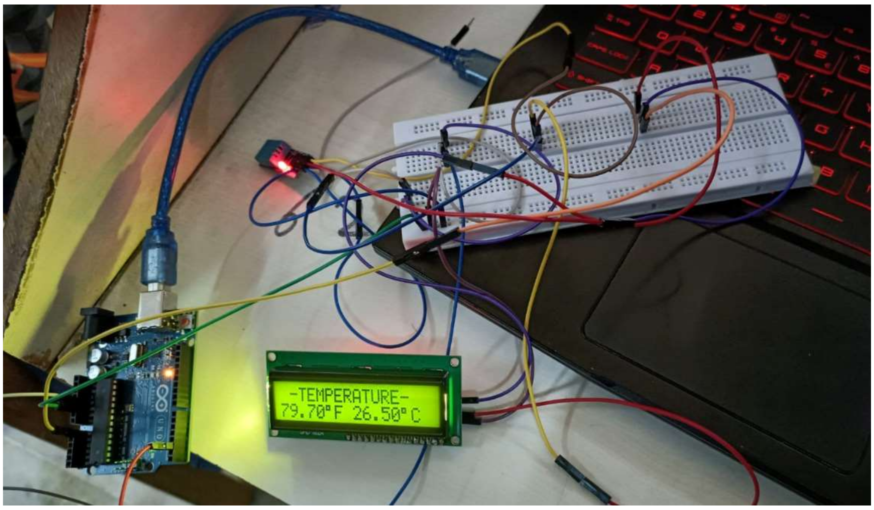

Fig. 1: Working of Temperature sensor using LCD display breadboard and Arduino.

7. Result/Output:

The experiment aimed to display temperature data on an LCD screen using Arduino. The temperature sensor was interfaced with Arduino, and a program was written to read data from the sensor and display it on the LCD screen. The experiment was successful in displaying real-time temperature data on the LCD, demonstrating the capabilities of using Arduino

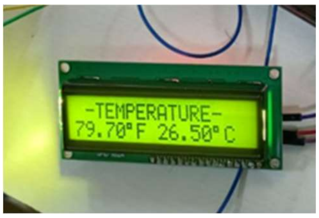

Fig.2: Output on LCD display

8. Analysis of the experiment:

The experiment aimed to display temperature data on an LCD screen using an Arduino microcontroller. To achieve this, a temperature sensor was connected to the Arduino, and a program was written to read the data from the sensor and display it on the LCD screen. The experiment was successful in displaying real-time temperature data on the LCD screen, which demonstrated the ability of the Arduino microcontroller to interface with sensors and display data on output devices.

The analysis of this experiment highlights the importance of microcontrollers in sensor-based projects. The use of an Arduino allowed for a simple and cost-effective solution for interfacing with the temperature sensor and displaying data on the LCD screen. The experiment also demonstrated the importance of programming skills in sensor-based projects. The program that was written to read and display the data on the LCD screen required a good understanding of programming concepts and the ability to translate them into working code. Overall, this experiment showcased the potential of using microcontrollers like Arduino for sensor-based projects and the importance of programming skills in developing such projects.