TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

Ayala Boulevard, Ermita, Manila

CIT-ELECTRONICS DEPARTMENT

CPET11L-M – Microprocessor and Microcontroller Systems, Lab

1st Semester, SY 2-24-2025

Name: Joseph C. Arenas	**Instructor:** Prof. Tristan Mopas
Course & Section: BET-CPET- 3A	**Date Submitted:**

Activity 3

Topic 1: Basic LCD Output

Topic 2: Temperature with LM35

Topic 3: Humidity and Temperature with DHT11

I. OBJECTIVES

- To apply practical knowledge in using the Arduino Mega 2560
- To explain the functions and components of the related topics
- To implement the LCD output, DHT11 and LM35 components using code and circuit diagrams
- To develop and enhance problem-solving skills related to the topics

II. EQUIPMENT AND MATERIALS

HARDWARE

- Arduino Uno R3 / Arduino Mega 2560
- Breadboard
- Jumper Wires
- Laptop
- DHT11

- LM 35
- 20x4 or 16x2 LCD

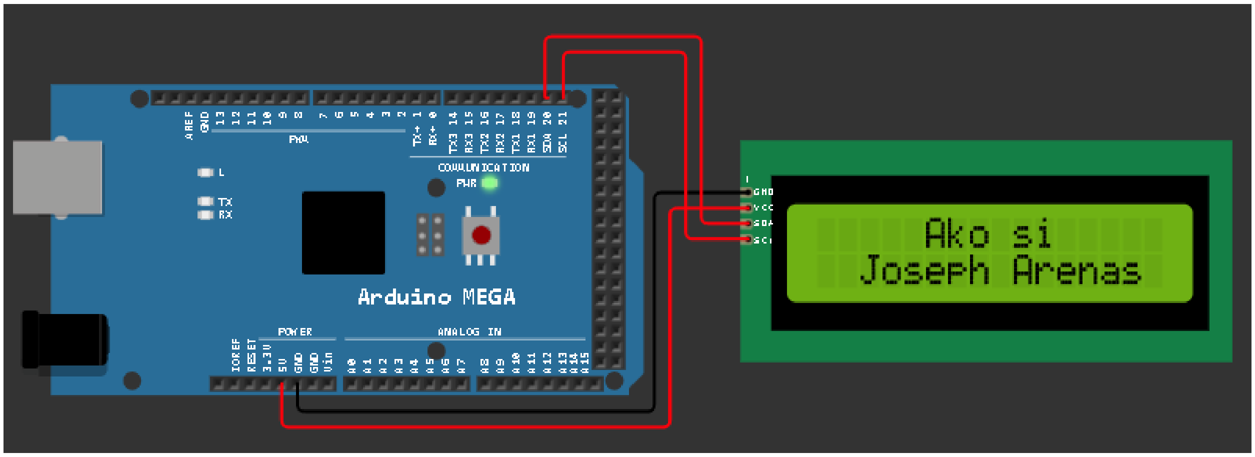
SOFTWARE

- Arduino IDE with libraries for specified components.
- MS Word
- Wokwi & TinkerCAD Simulator

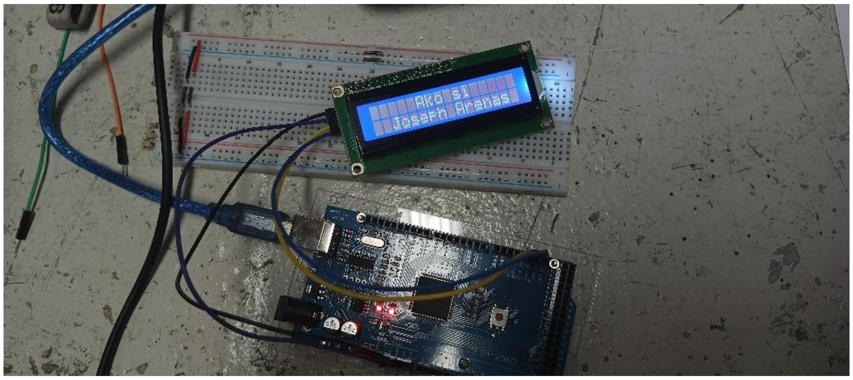
III. DIAGRAM

===== TOPIC 1: Basic LCD Output =====

A. Wokwi Simulation



B. Breadboard

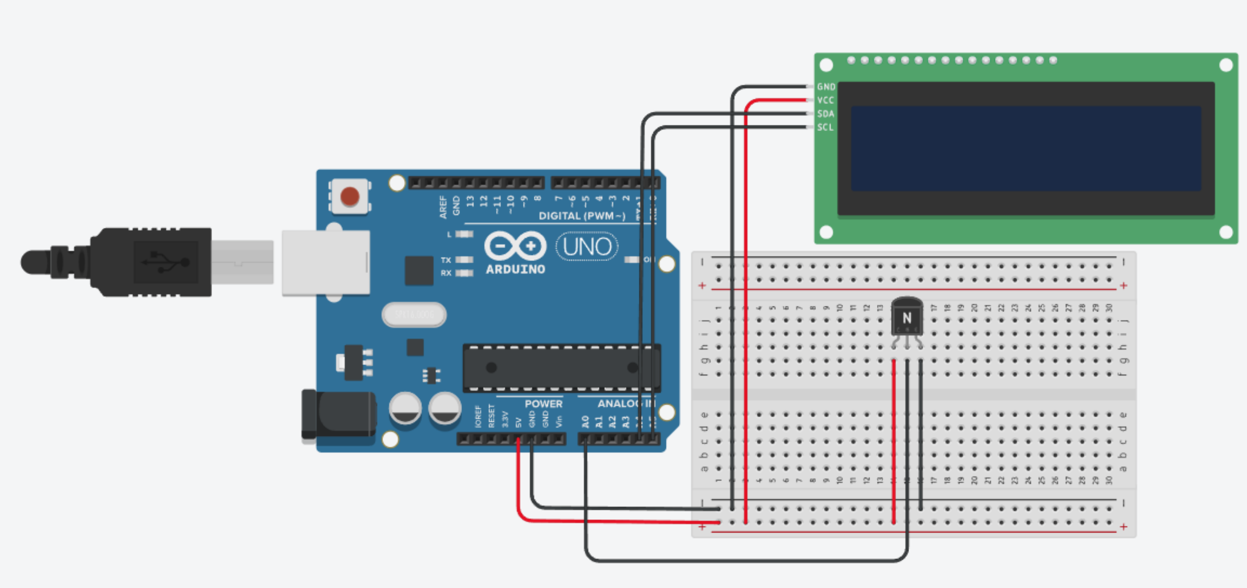


C. Source Code

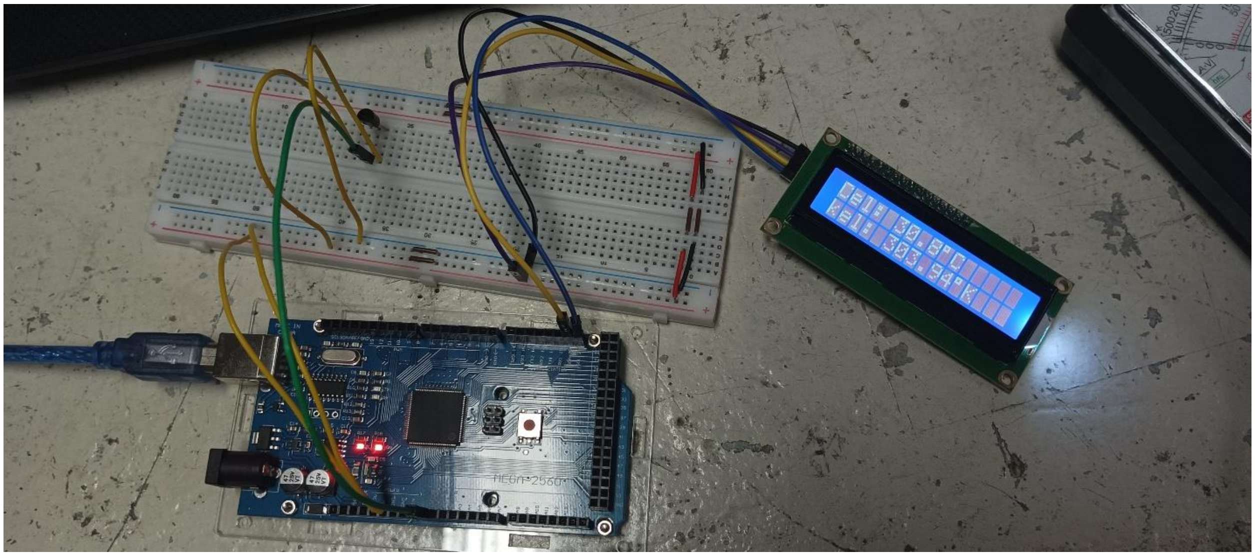
```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup(){
  lcd.init();
  lcd.clear();
  lcd.backlight();

  lcd.setCursor(5, 0);
  lcd.print("Ako si");

  lcd.setCursor(2, 1);
  lcd.print("Joseph Arenas");
}

void loop(){
}
```

===== Topic 2: Temperature with LM35 =====

A. TinkerCad Simulation



B. Breadboard



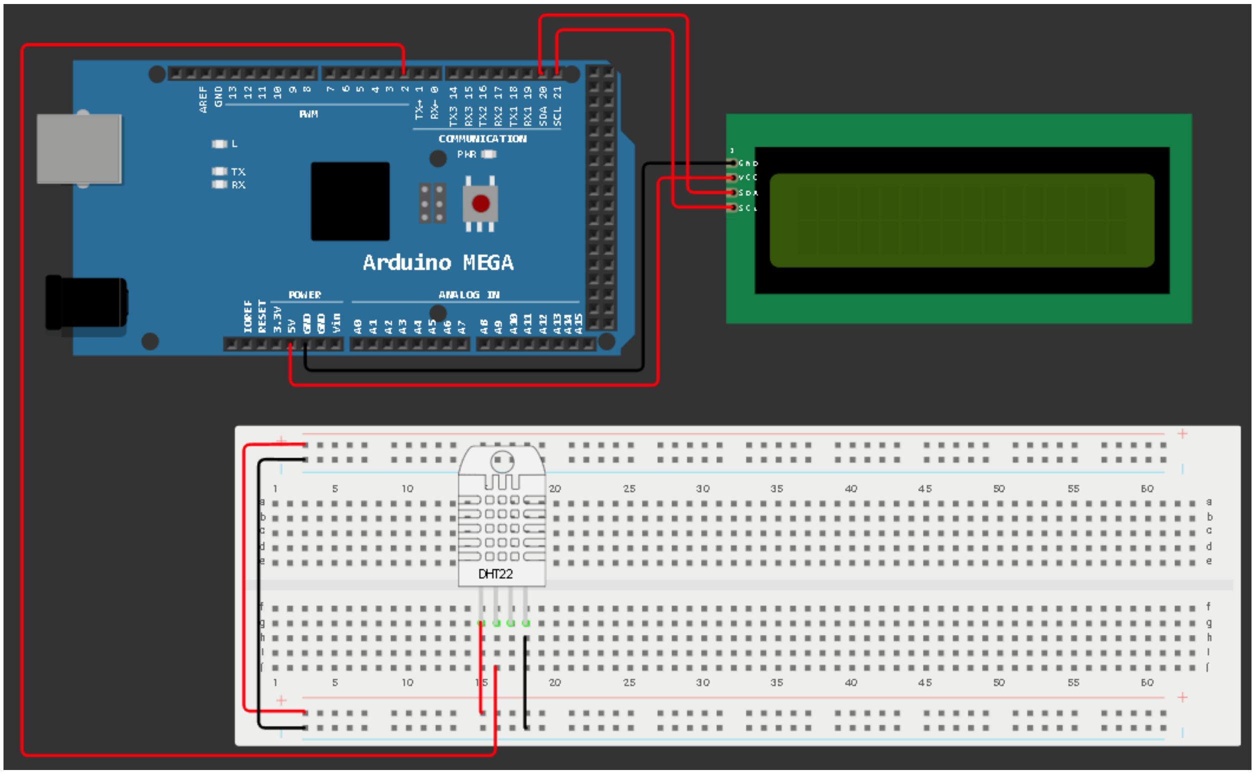
C. Source Code

```
#define USE_I2C 1
#if USE_I2C
  #include <Wire.h>
  #include <LiquidCrystal_I2C.h>
  LiquidCrystal_I2C lcd(0x27, 16, 2); // Change address if needed (0x27 or 0x3F)
#else
  #include <LiquidCrystal.h>
  LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // RS, EN, D4, D5, D6, D7
#endi
// === LM35 Settings ===
#define LM35_PIN A0  // LM35 OUT pin connected to analog pin A0
void setup() {
  // Start serial monitor (optional)
  Serial.begin(9600);
  // Initialize LCD
#if USE_I2C
  lcd.init();
  lcd.backlight();
#else
  lcd.begin(16, 2);
#endif
  // Startup message
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("LM35 Temp Sensor");
  lcd.setCursor(0, 1);
  lcd.print("Initializing...");
  delay(2000);
  lcd.clear();
}
```

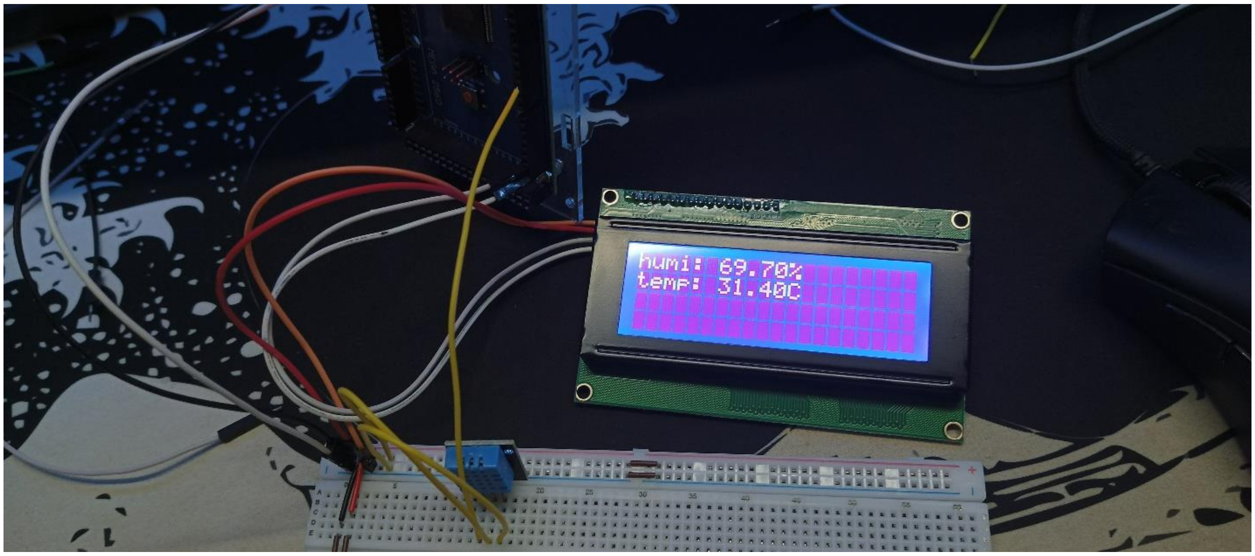
```
void loop() {
  // === Read LM35 Sensor ===
  int rawValue = analogRead(LM35_PIN);

  // LM35 formula: 10mV per °C
  // Temp(°C) = (ADC * 500) / 1023
  float celsius = (rawValue * 500.0) / 1023.0;
  float kelvin = celsius + 273.15;

  // === Display on LCD ===
  lcd.setCursor(0, 0);
  lcd.print("Cel: ");
  lcd.print(celsius, 1);   // 1 decimal place
  lcd.print((char)223);    // Degree symbol
  lcd.print("C");

  lcd.setCursor(0, 1);
  lcd.print("kel: ");
  lcd.print(kelvin, 2);
  lcd.print((char)223);    // 2 decimal places
  lcd.print("K");

  // === Print to Serial Monitor (optional) ===
  Serial.print("Temp: ");
  Serial.print(celsius, 1);
  Serial.print(" °C | ");
  Serial.print(kelvin, 2);
  Serial.println(" °K");

  delay(1000); // Update every 1 second
}
```

=== Topic 3: Humidity and Temperature with DHT11===

A. Wokwi Simulation



B. Breadboard



C. Source Code

```
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11

LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(DHTPIN, DHTTYPE);

void setup(){
  lcd.init();
  lcd.backlight();
  dht.begin();
}

void loop(){
  float humi = dht.readHumidity();
  float temp = dht.readTemperature();

  lcd.setCursor(0, 0);
  lcd.print("humi: ");
  lcd.print(humi);
  lcd.print("%");

  lcd.setCursor(0, 1);
  lcd.print("temp: ");
  lcd.print(temp);
  lcd.print("C");
  delay(1000);
}
```

IV. PROCEDURE

A. Preparation

- Gather the required components as stated in Chapter 3.
- Prepare the Arduino IDE on your preferred device.
- Review the problem for each topic and formulate or search for a circuit diagram given the problem of each topic.
- Simulate your circuit diagram to a circuit simulator such as TinkerCADor Wokwi.

B. Actual

- For topic 1, connect the 5v VCC and GND pin of the microcontroller to the "+" and "-" of the breadboard. Connect the VCC and GND of the LCD to VCC and ground. Connect the SDA and SCL of the LCD to the SDA and SCL pins of the microcontroller. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.
- For topic 2, connect the 5v VCC and GND pin of the microcontroller to the "+" and "-" of the breadboard. Connect the VCC and GND of the LCD to VCC and ground. Connect the SDA and SCL of the LCD to the SDA and SCL pins of the microcontroller. Attach the LM35 to the breadboard. Connect the VCC and GND pins of the transistor to the "+" and "-" of the breadboard. Afterwards, connect the data pin of the LM35 to A0 of the microcontroller. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.
- For topic 3, connect the 5v VCC and GND pin of the microcontroller to the "+" and "-" of the breadboard. Connect the VCC and GND of the LCD to VCC and ground. Connect the SDA and SCL of the LCD to the SDA and SCL pins of the microcontroller. Attach the DHT11 to the breadboard. Connect the VCC and GND pins of the DHT11 to 5v and ground. Connect the data pin of the DHT11 to digital pin 2. Connect the USBVCC to the Arduino Uno R3 / Arduino Mega 2560 then verify and upload the code.

C. Checking

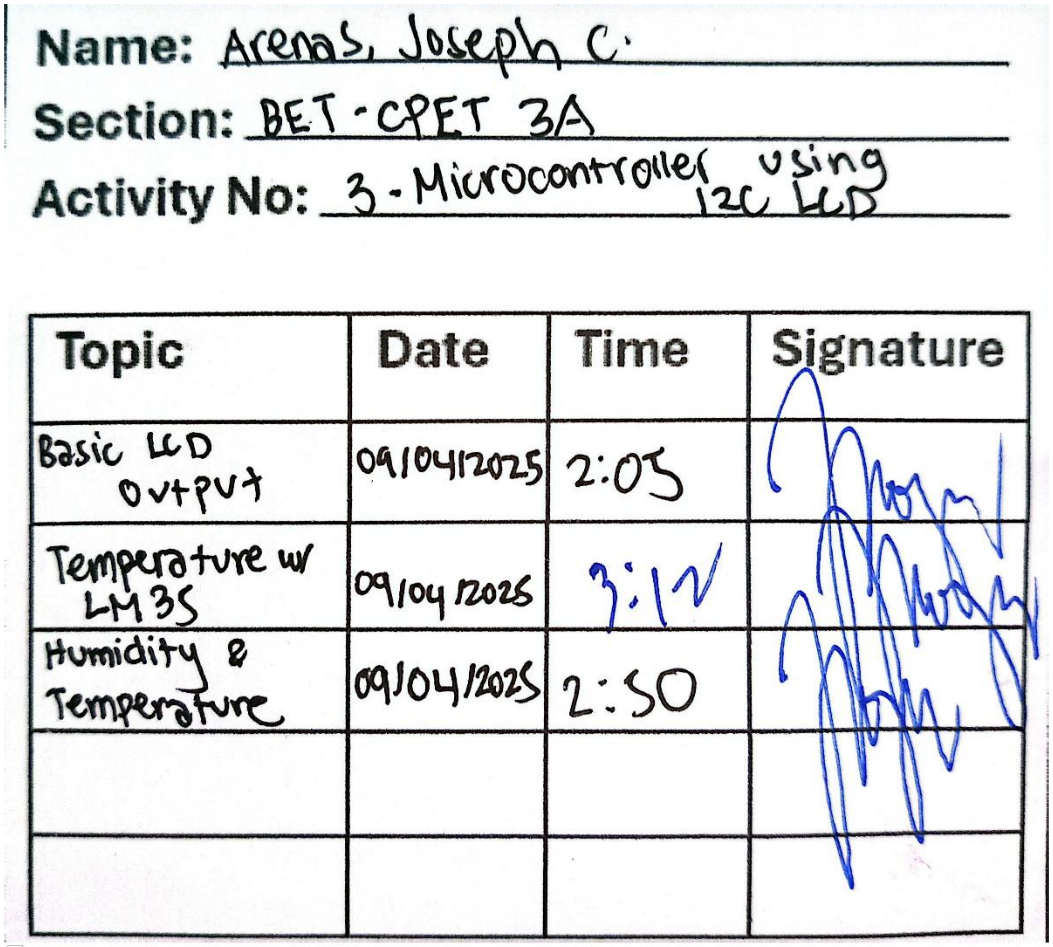
- For topic 1, ensure that the LCD displays the text of your choice based from the code as well as its proper positioning in the display.
- For topic 2, ensure that the LCD displays the temperature of the current environment. Make sure that the temperatures are accurate and realistic.
- For topic 3, ensure that the LCD displays the temperature and the humidity of the current environment. Make sure that the temperature and humidity readings are accurate.
- Adjust values or fix errors if the output is not working as expected.

D. Uploading

- Upload the final and corrected code to the Arduino Uno R3/Arduino Mega 2560.
- Ensure that all three topics work properly.

V. CONCLUSION

The activities provided hands-on experience in interfacing the Arduino with an LCD, LM35, and DHT11 sensors, effectively displaying temperature and humidity data. These exercises deepened understanding of how sensors communicate with microcontrollers and how outputs can be presented through hardware and code. Overall, the work improved both technical knowledge and problem-solving skills in microcontroller applications.



Name: Arenas, Joseph C.

Section: BET-CPET 3A

Activity No: 3 - Microcontroller using I2C LCD

Topic	Date	Time	Signature
Basic LCD Output	09/04/2025	2:05	
Temperature w/ LM35	09/04/2025	3:12	
Humidity & Temperature	09/04/2025	2:50	