REAL-TIME EMBEDDED SYSTEM

EXPERIMENT NO. 3

WEATHER MONITORING

Name: NAVARRO, ROD GERYK C.

Course/Section: CPE161P-4/C1

Group No.: N/A

Date of Performance: 12/17/2024
Date of Submission: 01/30/2025

CYREL O. MANLISES, PH.D.
Instructor

DISCUSSION

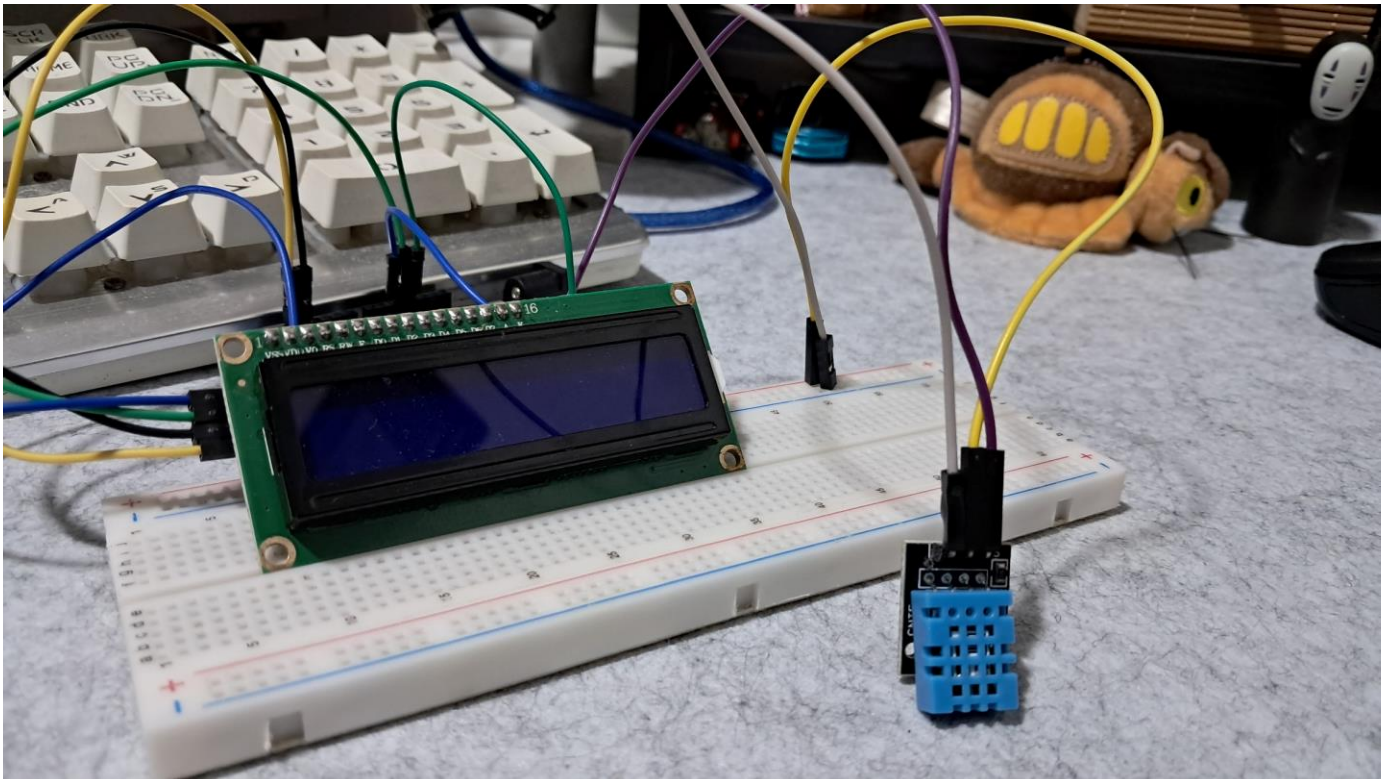
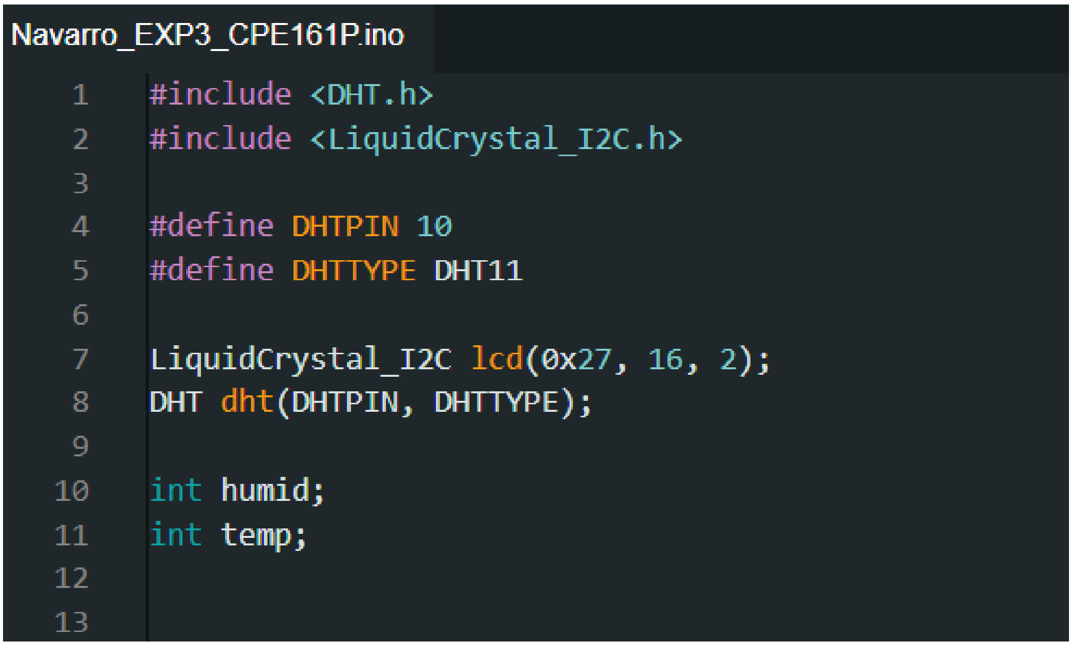


Figure 1: *The Connection of the 16x2 LCD, and the DHT11 Sensor.*

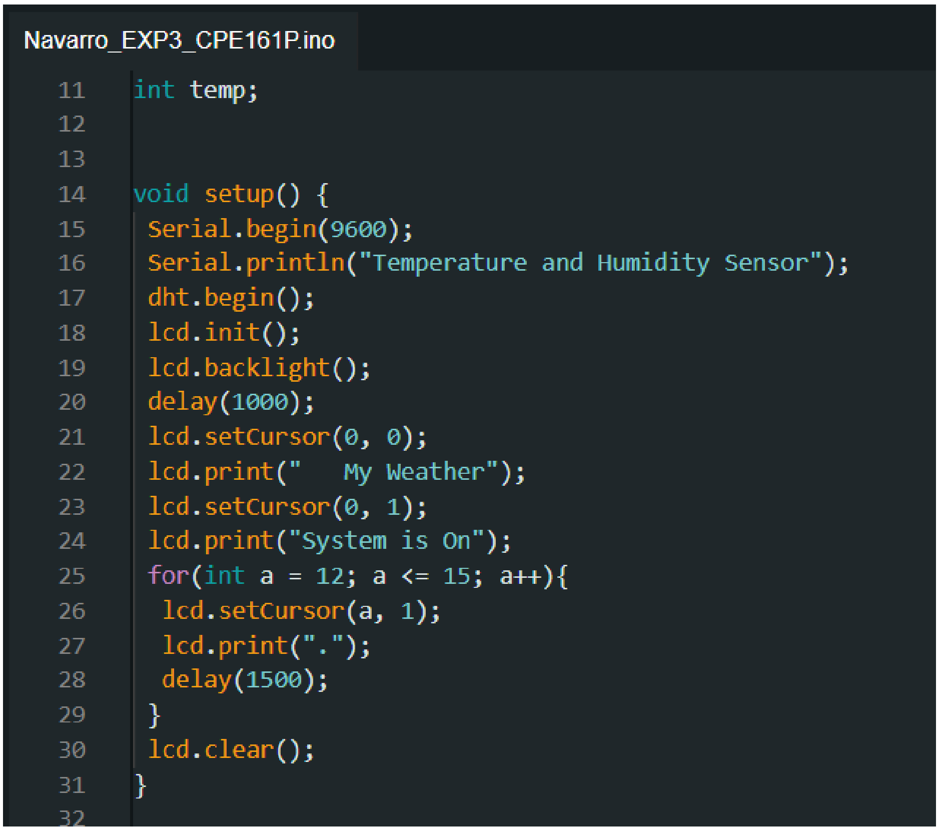
This experiment focuses on monitoring the weather, specifically by measuring temperature. However, as an additional feature, I also included humidity monitoring. To achieve this, I used a DHT11 sensor, which can measure both temperature and humidity. In this experiment, I used an LCD screen to display the readings from the DHT11 sensor. The sensor detects changes in temperature and humidity in the surroundings and provides this data as input. To begin, I first built the circuit by connecting the LCD screen and the DHT11 sensor.



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

#define DHTPIN 10
#define DHTTYPE DHT11

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

int humid;
int temp;


```

Figure 2: The Initialization of Variables and Libraries.

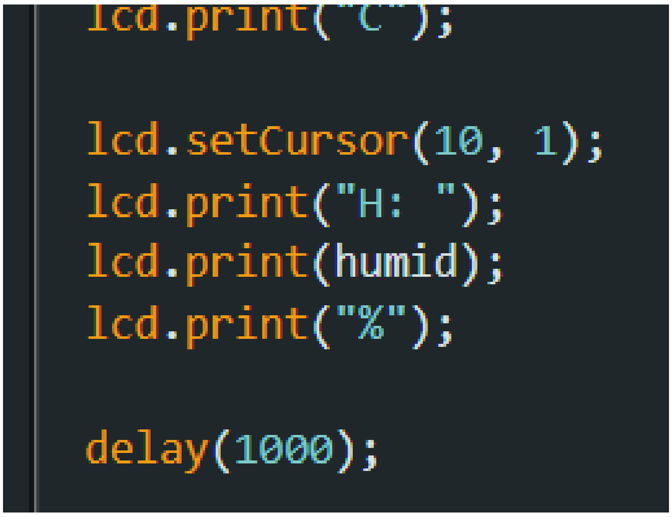
In this experiment, I focused on monitoring weather conditions by measuring temperature and humidity using the DHT11 sensor. To display the readings, I used an LCD screen. Before setting up the components, I first included the necessary libraries: DHT.h for handling the sensor and LiquidCrystal_I2C.h for controlling the LCD. I then defined DHTPIN as 10, which is where the sensor is connected, and specified the sensor type as DHT11. Next, I created objects for both the LCD and the sensor using LiquidCrystal_I2C lcd(0x27, 16, 2); and DHT dht(DHTPIN, DHTTYPE);. Lastly, I declared two integer variables, humid and temp, to store the humidity and temperature readings.



```
int temp;


void setup() {
 Serial.begin(9600);
 Serial.println("Temperature and Humidity Sensor");
 dht.begin();
 lcd.init();
 lcd.backlight();
 delay(1000);
 lcd.setCursor(0, 0);
 lcd.print("   My Weather");
 lcd.setCursor(0, 1);
 lcd.print("System is On");
 for(int a = 12; a <= 15; a++){
  lcd.setCursor(a, 1);
  lcd.print(".");
  delay(1500);
 }
 lcd.clear();
}
```

Figure 3: *The Setup of the Components.*

In the setup() function, I initialized the serial monitor with Serial.begin(9600); to display sensor data for debugging. Then, I started the DHT sensor using dht.begin(); and initialized the LCD with lcd.init();. To make the display more user-friendly, I turned on the LCD backlight and displayed a startup message: "My Weather System is On." A simple loading effect was created using a loop that printed dots on the screen. After a short delay, I cleared the screen, preparing it for real-time monitoring.



```
lcd.print("C");

lcd.setCursor(10, 1);
lcd.print("H: ");
lcd.print(humid);
lcd.print("%");

delay(1000);
```

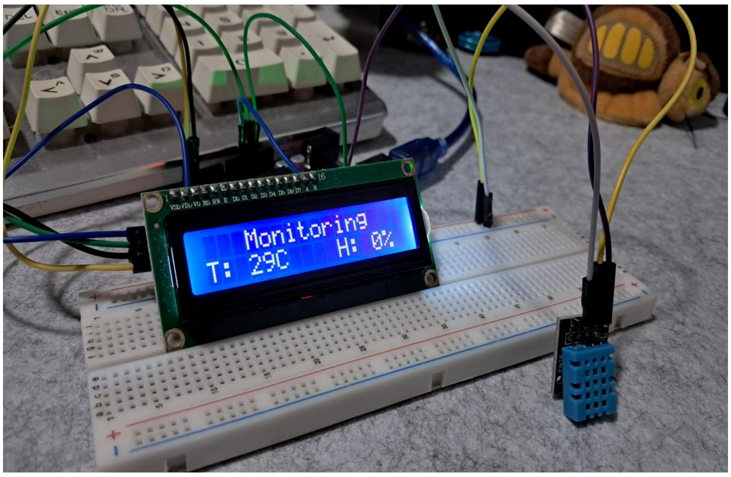
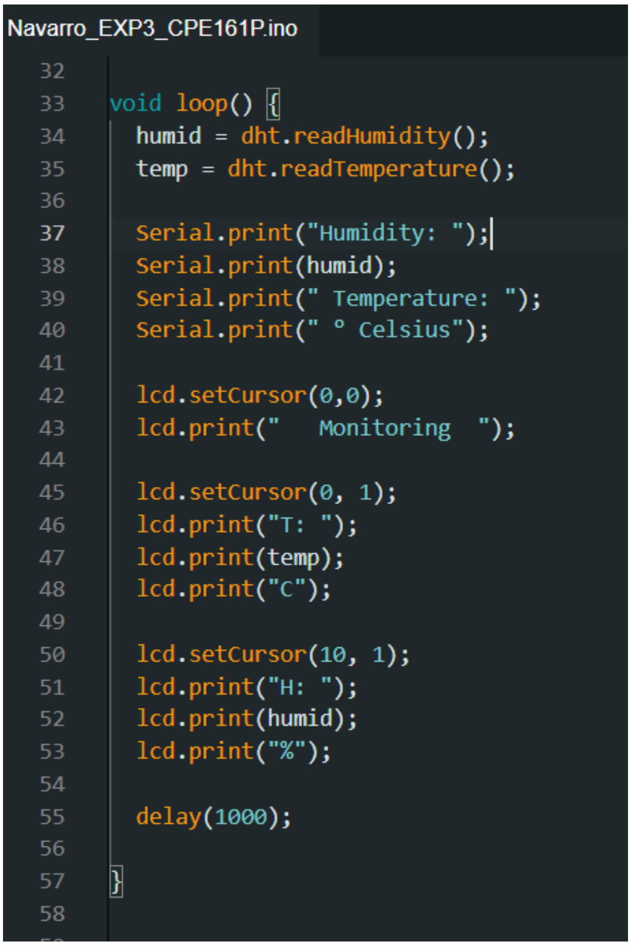


***Figure 4:** Fixing the Errors.*

During testing, I noticed that the temperature values were not displaying correctly and occasional incorrect readings from the sensor. To solve this, I ensured that the DHT sensor was properly connected and added a slight delay between readings to give the sensor enough time to update. Also, the sensor that I have used is inaccurate when it comes to monitoring the real values of temperature and humidity in my place or location specially the humidity values. However, I still manage to meet the main requirement which is it works in real-time. The system works by monitoring the changes in temperature and humidity, which is very important in a weather monitoring system.



Navarro_EXP3_CPE161P.ino

```

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

  Serial.print("Humidity: ");
  Serial.print(humid);
  Serial.print(" Temperature: ");
  Serial.print(" ° Celsius");

  lcd.setCursor(0,0);
  lcd.print("   Monitoring  ");

  lcd.setCursor(0, 1);
  lcd.print("T: ");
  lcd.print(temp);
  lcd.print("C");

  lcd.setCursor(10, 1);
  lcd.print("H: ");
  lcd.print(humid);
  lcd.print("%");

  delay(1000);

}

```

Figure 5: Loop Function.

The loop() function continuously reads temperature and humidity values from the DHT11 sensor. It first assigns the humidity and temperature readings to humid and temp using dht.readHumidity(); and dht.readTemperature();. These values are then printed on both the serial monitor and the LCD screen. The LCD displays ”Monitoring” on the first row, while the second row shows the temperature (T:) and humidity (H:) values. A delay(1000); ensures that the readings update every second, preventing flickering on the display. This setup allows real-time weather monitoring with clear and updated readings.

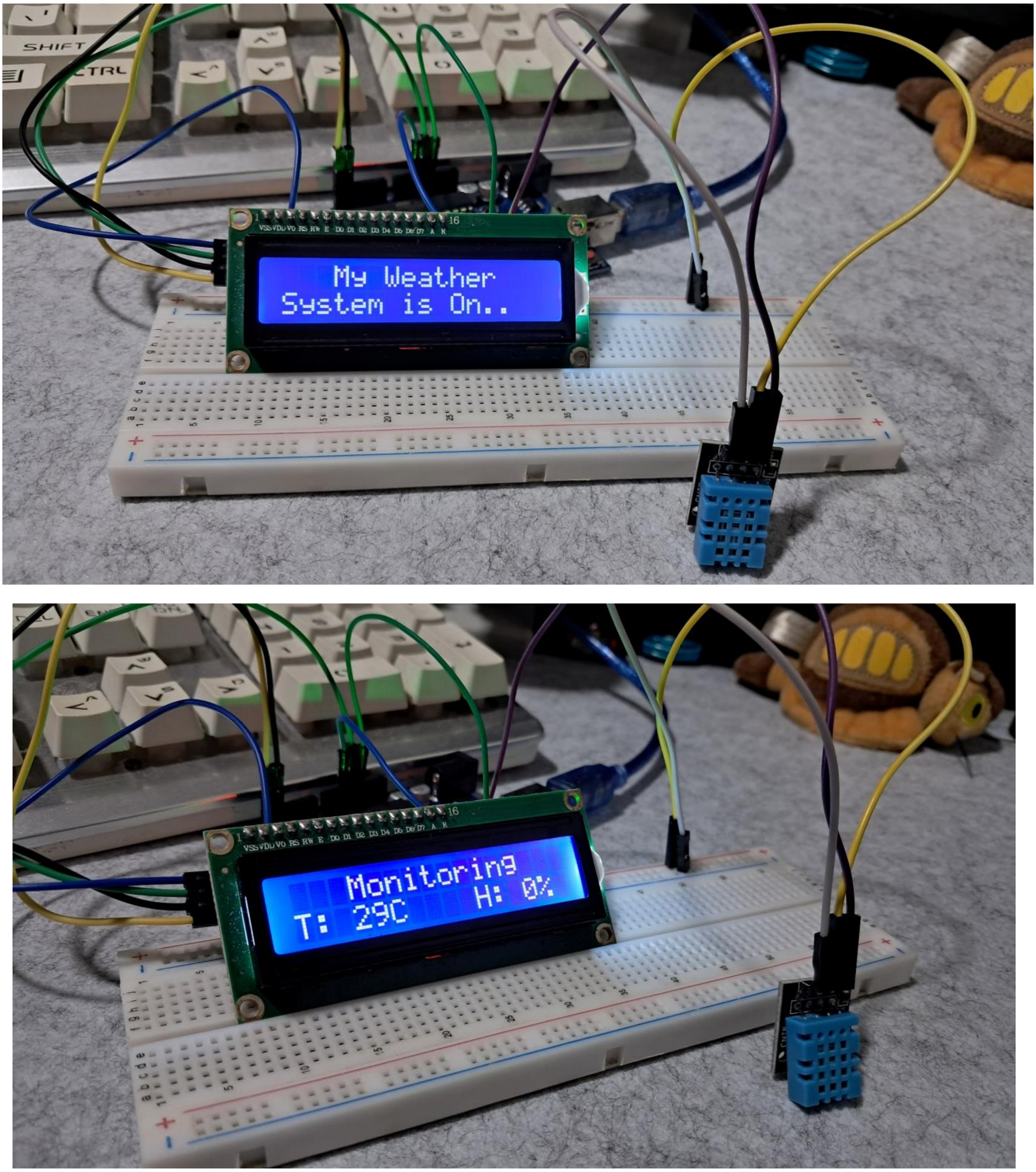


Figure 6: *Working Prototype.*

This is the final working prototype of the weather monitoring system experiment, which successfully meets all the requirements.

CONCLUSION

In this experiment, I successfully built a real-time weather monitoring system using a DHT11 sensor and an LCD screen. The system can measure both temperature and humidity, displaying the readings in real-time. I started by setting up the necessary components, including initializing the sensor and LCD, and writing code to read and display the values. Throughout the process, I encountered some issues, such as incorrect readings, but I managed to fix them by adjusting the sensor connection and adding a delay between readings.

Although the DHT11 sensor is not highly accurate, especially for humidity measurements, the system still functions as expected and provides real-time monitoring. The LCD screen effectively displays the temperature and humidity changes, making it a useful tool for basic weather tracking. This experiment helped me understand how sensors collect data and how real-time systems work, which can be useful for future projects involving environmental monitoring.