EMBEDDED SYSTEM AND DESIGN
EXPERIMENT NO. 4

Kettle

Name: NAVARRO, ROD GERYK C.

Course/Section: CPE160P-4/A1

Group No.:

Date of Performance: 09/05/2024
Date of Submission: 09/07/2024

CYREL O. MANLISES, PH.D.
Instructor

DISCUSSION

In the experiment, I first created the circuit to connect all the components in their respective ports of the Arduino. As shown in the figure below that I've used two breadboards in the experiment, the purpose of this is to separate the DHT11 sensor and the servo motor in the other breadboard in order to extend the reach of the sensor and motor to connect to the kettle. I have connected the output pin of the servo motor to the pin 9 of the Arduino, which control the pulse of the servo motor that control its speed. For the sensor, DHT11 sensor is connected to the pin 10 for gathering data for the measurement of the temperature and humidity of the water, and a buzzer that is connected to pin 3, that serves as an alarm.

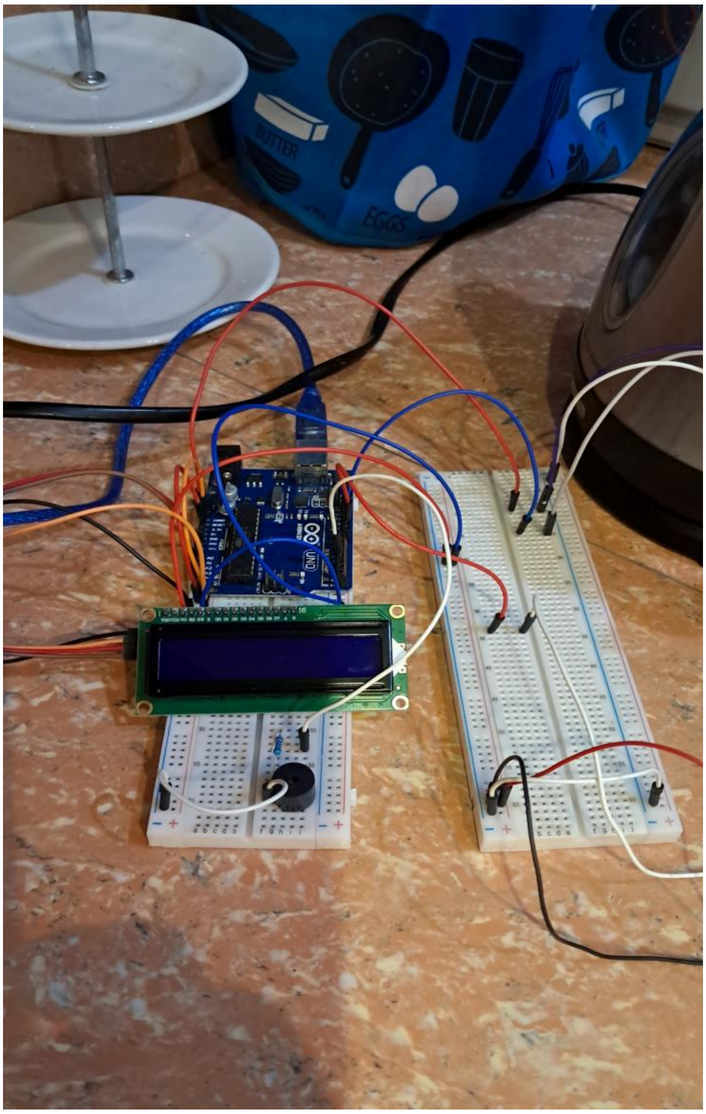

Figure 1: *The connection of the LCD, DHT11 sensor, servo motor, and the active buzzer to the Arduino board.*

In this part of the experiment, I begin to test the design of the whole circuit with the electric kettle, so taped the motor around the handle of the kettle to turn off the kettle if it reaches the boiling point temperature of the water, and also tested if the DHT11 sensor can reach the mouth of the kettle.

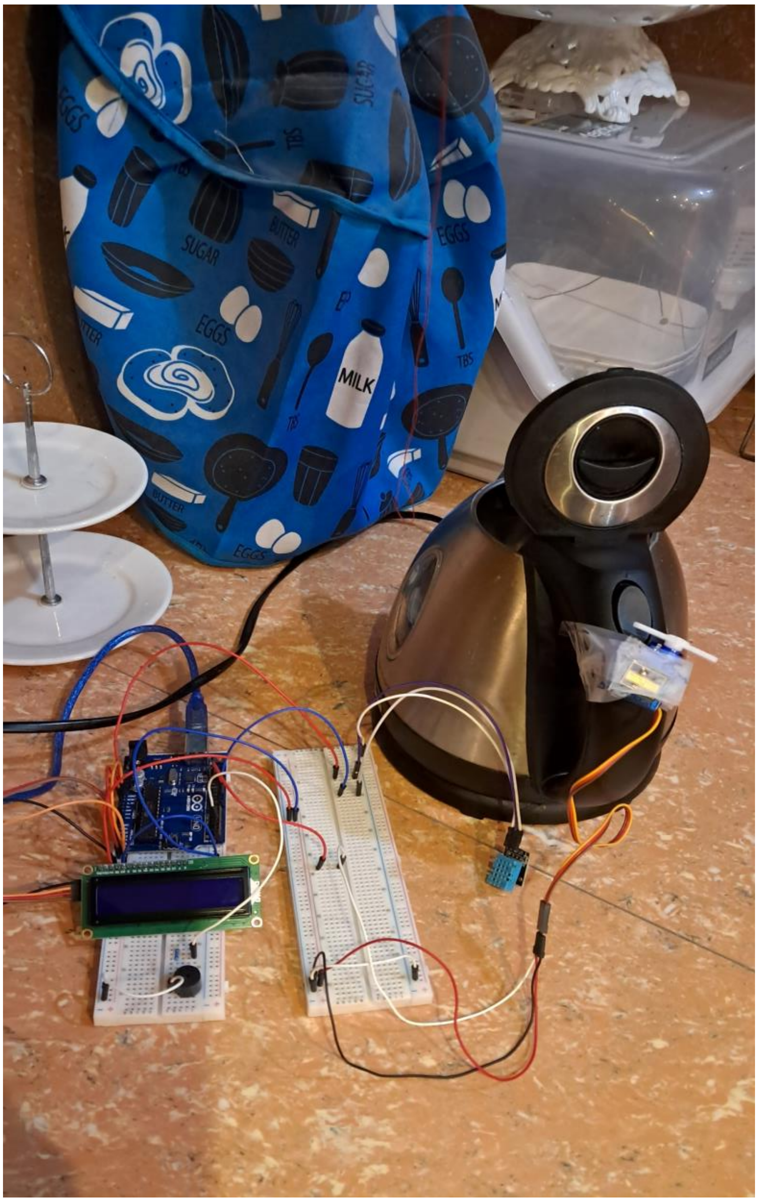

Figure 2: *The connection of the circuit with the kettle.*

For the coding, the code uses 4 libraries, namely the DHT.h, Wire.h, Servo.h, and the LiquidCrystal_I2C.h. The DHT library is used to interface with DHT sensors, the Wire library is used for I2C (Inter-Integrated Circuit) communication in Arduino, the Servo library is used to control the servo motor, and the LiquidCrystal_I2C library is used to control LCD displays that use the I2C communication protocol. After initializing the libraries, I also initialized the angle variable which enables the servo motor to have a controlled angle.

Navarro_EXP4.ino

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

#define DHTPIN 10
#define DHTTYPE DHT11
const int buzzAlarm = 3;
int angle = 0;

Servo ServoSwitch;

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

int humid;
int temp;

void setup() {
  Serial.begin(9600);
  Serial.println("Temperature and Humidity Sensor");
  dht.begin();
  lcd.init();
  lcd.backlight();
  ServoSwitch.attach(9);
  pinMode(buzzAlarm, OUTPUT);
}

void loop() {

  humid = dht.readHumidity();
  temp = dht.readTemperature();
```

Figure 3: The insertion of the libraries and other variables needed in the program.

After the initiation of the libraries and variables, the next step that I did is initialized the local variables which are the humid and temp to store the data gathered by the DHT11 sensor. Also, I have configured the display of the LCD. In the LCD, the T symbolizes the temperature, while H for the humidity. I also output a display message to indicate if the kettle is still on if the water in the kettle hasnt reach the boiling point temperature which displays Kettle ON , and if the water has reached or go beyond the set boiling temperature it will display Kettle is OFF which indicates that the kettle is turned off by the servo motor.

Navarro_EXP4.ino

```
void loop() {

  humid = dht.readHumidity();
  temp = dht.readTemperature();

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

  lcd.setCursor(1, 0);
  lcd.print("Kettle is ");
  displayTNH();

  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);

  if(temp >= 60){
    digitalWrite(buzzAlarm, HIGH);
    for(angle = 0; angle <= 180; angle++){
      ServoSwitch.write(angle);
      delay(20);
      break;
      delay(2000);
```

Figure 4: *The initiation of local variables, as well as the configuration of the display.*

The next step would be to set the if conditions to control the servo motor and the buzzer. In this experiment I have realized that the sense data of the DHT11 sensor is inaccurate and not real time since it has a huge delay in measuring the temperature and humidity. So, what I did was to set the trigger value of the temperature to a value that will allow the water in the kettle reach the boiling point temperature, which in my case it is 60 degrees Celsius. However, I know that it is not the original boiling temperature of the water the boiling point of the water is originally 100 degrees Celsius but, in my case, if it reaches to 60 degrees Celsius the water starts to boil in the kettle so I did some adjustments.

Using if conditions, if the temperature of the water reaches or go above 60 degrees Celsius it will trigger the servo motor to rotate a bit from an angle of 180 degrees to click the switch of the kettle to turn off, and also it will trigger the buzzer to set to HIGH as the indication that the water is boiling. On the other hand, if the water is below the set temperature the servo motor will remain at the 180 degrees angle while the buzzer is set to a LOW. In addition, I also created a function that controls the display in the LCD for the indication if the kettle is on or off.

Navarro_EXP4.ino

```
if(temp >= 60){
  digitalWrite(buzzAlarm, HIGH);
  for(angle = 0; angle <= 180; angle++){
    ServoSwitch.write(angle);
    delay(20);
    break;
    delay(2000);

  }
}else if(temp < 60){
  digitalWrite(buzzAlarm, LOW);
  for(angle = 180; angle > 0; angle--){
    ServoSwitch.write(angle);
    delay(20);
    break;
    delay(2000);

  }
}


}

void displayTNH(){
  if(temp >= 60){
  lcd.setCursor(11, 0);
  lcd.print("OFF");

  }else if(temp < 60){
  lcd.setCursor(11, 0);
  lcd.print("ON ");
  }
```

Figure 5: The conditions to control the servo motor and the active buzzer.

CONCLUSION

I successfully connected various components, including a DHT11 sensor, a servo motor, an LCD display, and a buzzer, to an Arduino board to monitor and control the boiling process of water in a kettle. The setup involved using two breadboards to properly position the components for effective operation. By coding the Arduino using the appropriate libraries, I was able to monitor the water temperature and humidity, display this information on the LCD, and use a servo motor to turn off the kettle when the desired temperature was reached.

While the experiment was successful in demonstrating the use of different sensors and actuators to automate a simple task, I encountered limitations with the DHT11 sensor, which had a noticeable delay in providing real-time data. To address this, I adjusted the threshold temperature to 60 degrees Celsius, allowing the setup to function effectively despite the sensor's limitations. Overall, the experiment highlighted both the potential and the challenges of integrating multiple components in an embedded system to achieve a specific task.