Mapúa University

School of Electrical, Electronics and Computer Engineering

Real Time Embedded System

COE187P/ E01

Temperature Sensor

Lab: #04

Submitted By:

Sardina, Kent Johnric M.

Submitted To:

Engr. Jocelyn Villaverde

I. Introduction

DHT11 Humidity and Temperature sensor that measures the humidity and temperature of the surrounding was used in all parts of the experiment. Temperature sensors are used in many industries like agriculture, medical, and ect. There are many types of temperature sensors these are: Thermocouples, thermistor, DHT11, DS18B20 and etc. Temperatures are very useful many automation or robotics prototypes and projects are using a temperature sensor the measure the temperature of the surrounding or a specific surface they are also used as a part of a temperature controller which monitors the temperature.

II. Objectives

1. Familiarize the basic connection and functions of Arduino
2. Familiarize on how to code Arduino
3. Familirize on using different libraries for the sensors and other modules
4. Familiarize on using the Serial Monitor
5. Use DHT11 as a temperature sensor

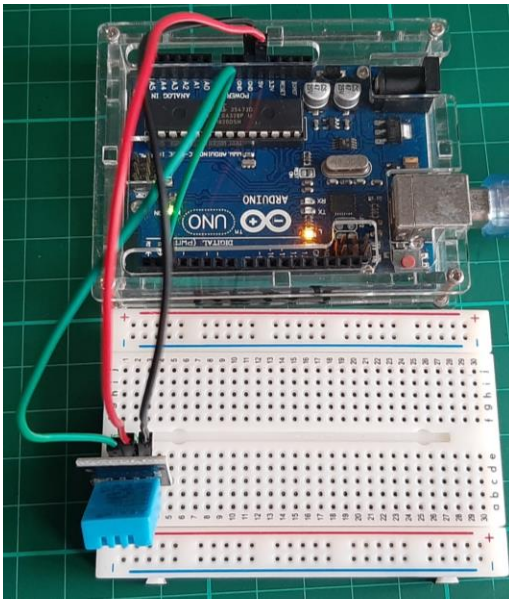
III. Materials and Components

- Arduino UNO and peripherals
- Jumper Wires
- Breadboard
- Arduino IDE / Visual Studio with Arduino IDE plugin
- DHT11 Humidity and Temperature Sensor
- Buzzer

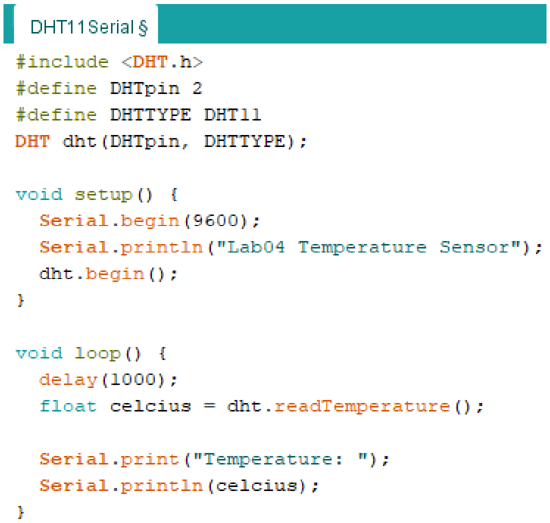
IV. PROCEDURE

Part 01: Using DHT11 with Serial Monitor

1. Follow the Connection of the DHT11 to the Arduino



2. Connect the VCC to 5 volts and GND to Ground of the Arduino
3. Connect the Signal pin of the DHT11 to pin 2 of the Arduino
4. Open Arduino IDE or Visual Studio (with Arduino IDE plugin)
5. Enter the Code Below



```
#include <DHT.h>
#define DHTpin 2
#define DHTTYPE DHT11
DHT dht(DHTpin, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("Lab04 Temperature Sensor");
  dht.begin();
}

void loop() {
  delay(1000);
  float celcius = dht.readTemperature();

  Serial.print("Temperature: ");
  Serial.println(celcius);
}
```

6. Select the correct COM port for the Arduino

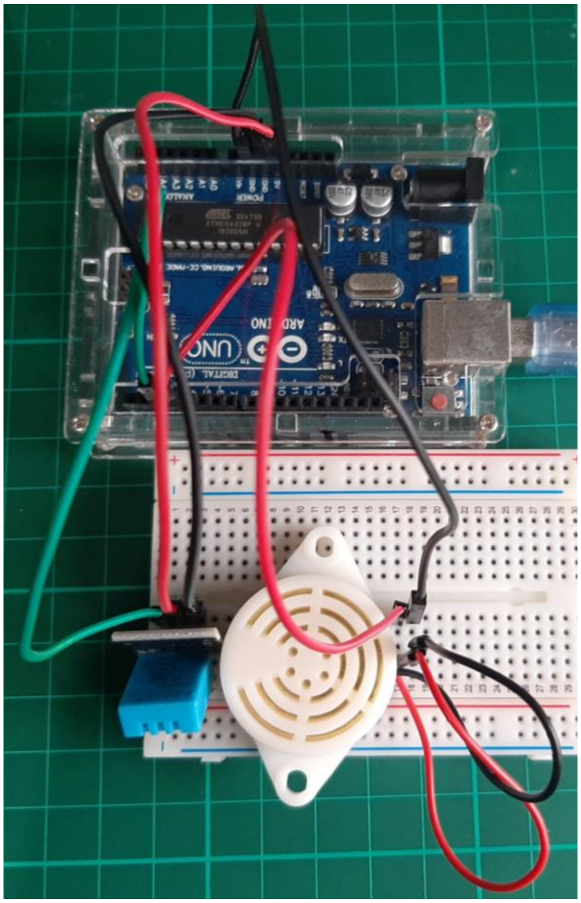
7. Verify, Save, and Upload the code to the Arduino UNO

8. Open the Serial Monitor of the Arduino IDE to determine the temperature read by the DHT11

9. Test and observe the Resuslts

Part 02: Temperature Sensor (with Serial Monitor) and Buzzer

1. Follow the connection of the DHT11 and Buzzer to the Arduino



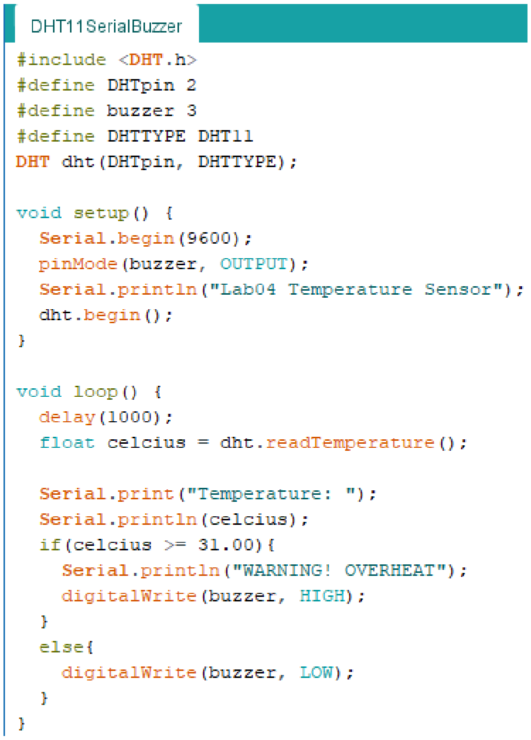
2. The connection of the DHT11 is the same in part 01

3. Connect the GND of the Buzzer to Ground of the Arduino

4. Connect the Red wire of the buzzer to pin 3 of the Arduino

5. Open Arduino IDE or Visual Studio(with Arduino IDE plugin)

6. Enter the Code below:



```
#include <DHT.h>
#define DHTpin 2
#define buzzer 3
#define DHTTYPE DHT11
DHT dht(DHTpin, DHTTYPE);

void setup() {
  Serial.begin(9600);
  pinMode(buzzer, OUTPUT);
  Serial.println("Lab04 Temperature Sensor");
  dht.begin();
}

void loop() {
  delay(1000);
  float celcius = dht.readTemperature();

  Serial.print("Temperature: ");
  Serial.println(celcius);
  if(celcius >= 31.00){
    Serial.println("WARNING! OVERHEAT");
    digitalWrite(buzzer, HIGH);
  }
  else{
    digitalWrite(buzzer, LOW);
  }
}
```

7. Select the correct COM port for the Arduino
8. Verify, Save, and Upload the code to the Arduino
9. Open the Serial Monitor of the Arduino IDE to determine the temperature and the Overheat Warning
10. Test and observe the results

V. Results and Discussion

In this experiment I used DHT11 as the temperature sensor and for this to work you need to download and include that library in your Arduino UNO, DHT11 needs 2 library files these are DHT11 library master and Adafruit Sensor library master without these libraries the code will have an error and have do define the library in the code. By using the Serial monitor of the Arduino IDE you need to add Serial.begin(9600) in the code with default baud rate at 9600 and using Serial.print() or Serial.println() functions if you want to display something on the serial monitor; by using the libraries you have included you need to know the defined variables and functions of the library like the readTemperature() function of library in order to read the temperature.

VI. Conclusion

After the parts and activity, all the objectives in this experiment are achieved. By using the functions of the library for the DHT11 Humidity and Temperature sensor, I was able to create, run, and test the functionality of the sensor. If you you want to a specific library for a specific component you first need to read the library code (a file with an extension name of .h) to know its variables and code functions for you to use it in your code and run that specific component.