

CPE316 – Embedded Systems Final Project Report  
[Temperature Humidity Data Logger]  
Semester II (2021-2022)

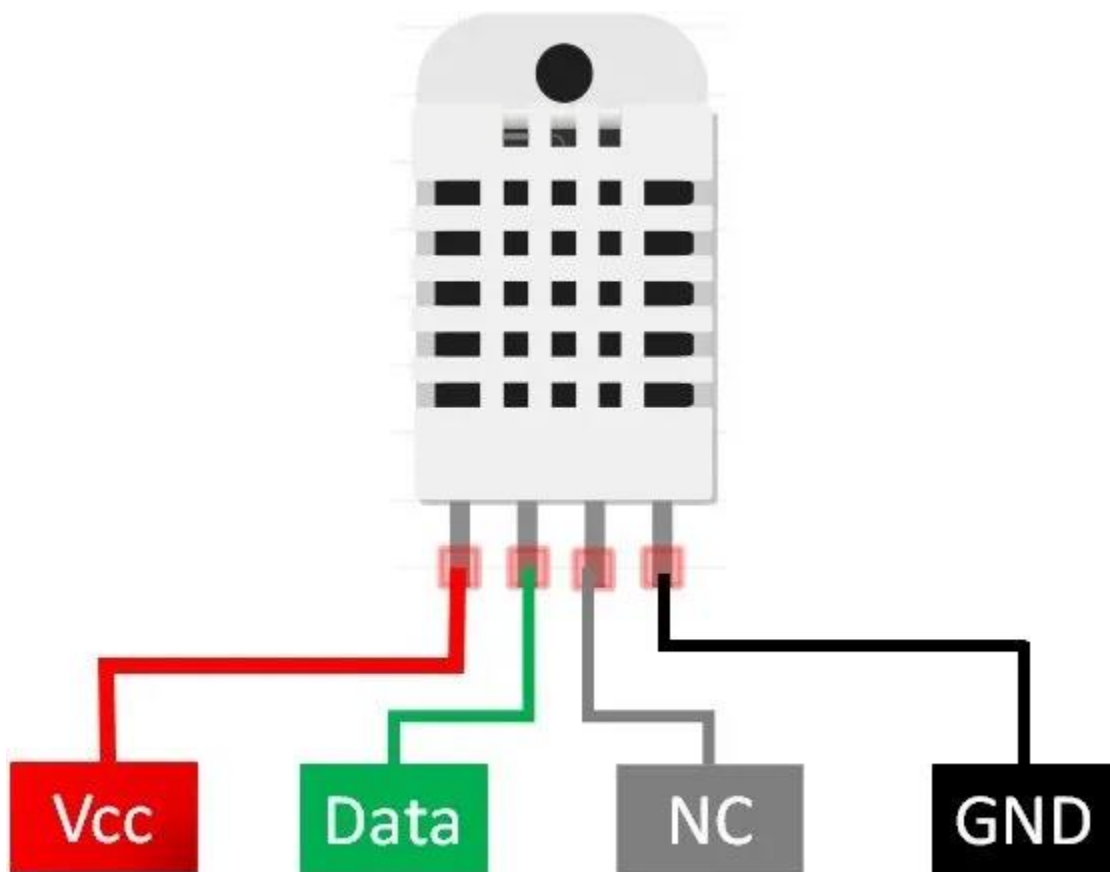
Course supervisors: Dr. Iman Elawady ENG. Michael B.Khani	Team Rocket Team members: Zafer Manar(2017010206029@ogrenci.karabuk.edu.tr) Muhammet Sarıcan(2017010206042@ogrenci.karabuk.edu.tr) Batıkan Burak Taş(2017010206039@ogrenci.karabuk.edu.tr) Uğurcan Tiryaki(2017010206035@ogrenci.karabuk.edu.tr)
---	--

## Contents

1. Introduction .....	3
Schematic.....	6
Related Works .....	6
3. Project Design .....	6
3.1. Project Layout .....	6
4. Conclusion and Results .....	6
5. Lesson Learnt .....	7
6. References .....	7
7. Attachments.....	8
7.1. Project Images.....	8
7.2. Code Modules .....	9

## 1. Introduction

In this user guide, we will learn how to log temperature and humidity readings acquired from DHT22 sensor to a microSD card using Arduino and Arduino IDE. We will create a .txt file in our microSD card through programming our Arduino board and consequently log temperature and humidity readings to that file after every seconds. Users can use this tutorial, to learn how to acquire data from the DHT22



DHT22 Pins	Arduino UNO
1 (VCC)	This is the power supply pin (3.3V-5V). We will connect with 5V.
2 (Data)	Any digital pin of Arduino board along with 10k ohm pull-up resistor. We will use GPIO2
3 (NC)	Not used
4 (GND)	Ground

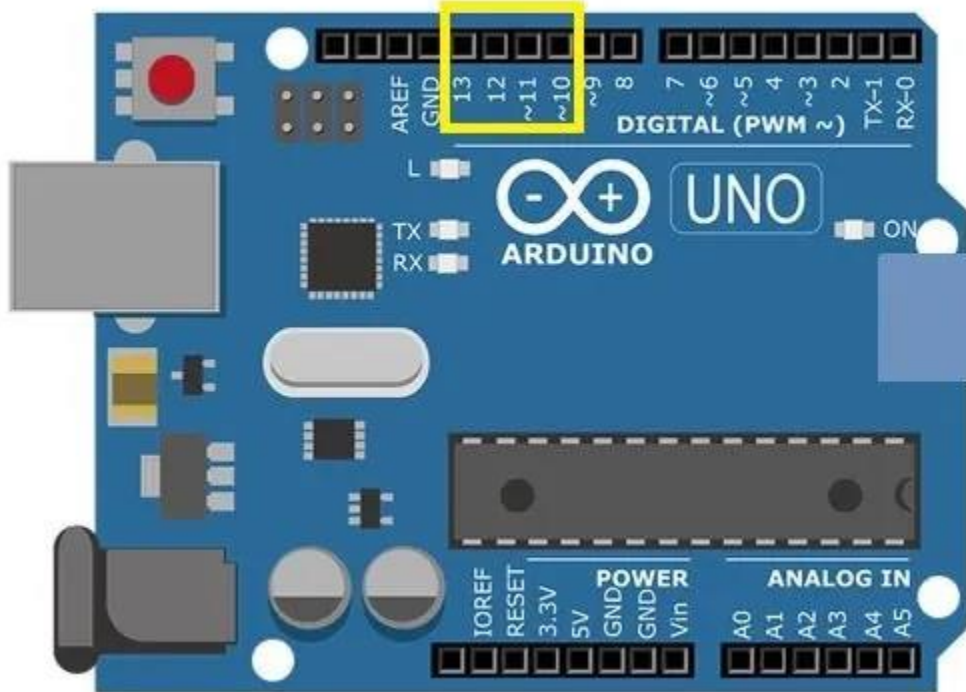
## 2. Arduino Connection Diagram with DHT22 and microSD card module

This section shows how to connect Arduino UNO with DHT22 sensor and the microSD card module.

The DHT22 sensor has 4 terminals which we will connect with the Arduino board. As it is 5V tolerant, hence we will connect the VCC terminal with 5V of the Arduino UNO. The data out pin will be connected with GPIO2 with 10k ohm pull-up resistor. The resistor is not required if you are using the DHT22 module instead. You can also choose any other appropriate digital pin to connect with data out. The third pin is not used.

Additionally, we will connect the VCC terminal of microSD card module with 5V of Arduino UNO. Both grounds will be common. The default SPI GPIO pins of Arduino UNO are being used to connect with each of the remaining SPI terminals of the microSD card

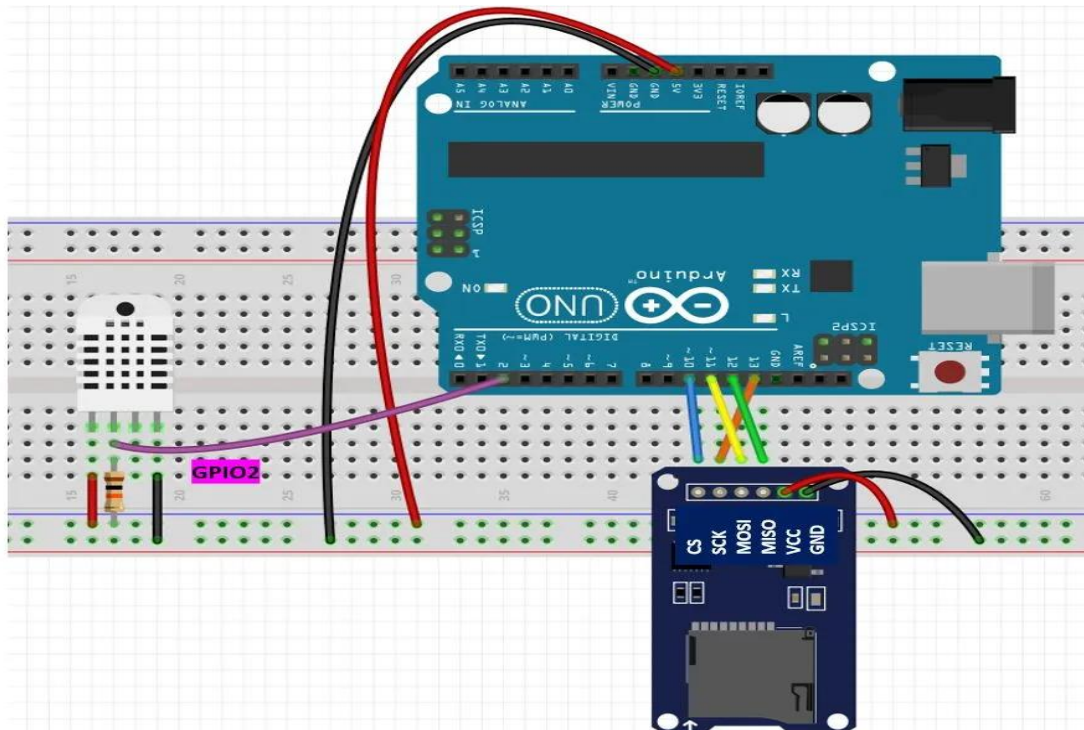
module. The figure below shows the default SPI pins of Arduino Uno.



The table below shows the connections between Arduino UNO and the microSD card module:

Arduino UNO	MicroSD card Module
GND	GND
5V	VCC
GPIO10	CS
GPIO11	MOSI
GPIO13	SCK
GPIO12	MISO

## Schematic



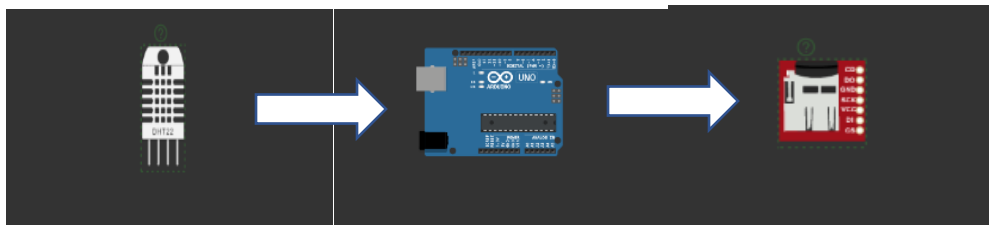
## Related Works

First, research was conducted on sensors that measure temperature and humidity. Learned about DHT22 and sd card reader libraries and needed coding done.

### 3. Project Design

A project design is a strategic organization of ideas, materials and processes for the purpose of achieving a goal.

#### 3.1. Project Layout



## 4. Conclusion and Results

As a result, we have obtained a mechanism where we can keep the temperature and humidity information in our hands. This mechanism can be used as an anathema in a more complex structure in the future, such as air conditioners that control the ambient temperature.

## **5. Lesson Learnt**

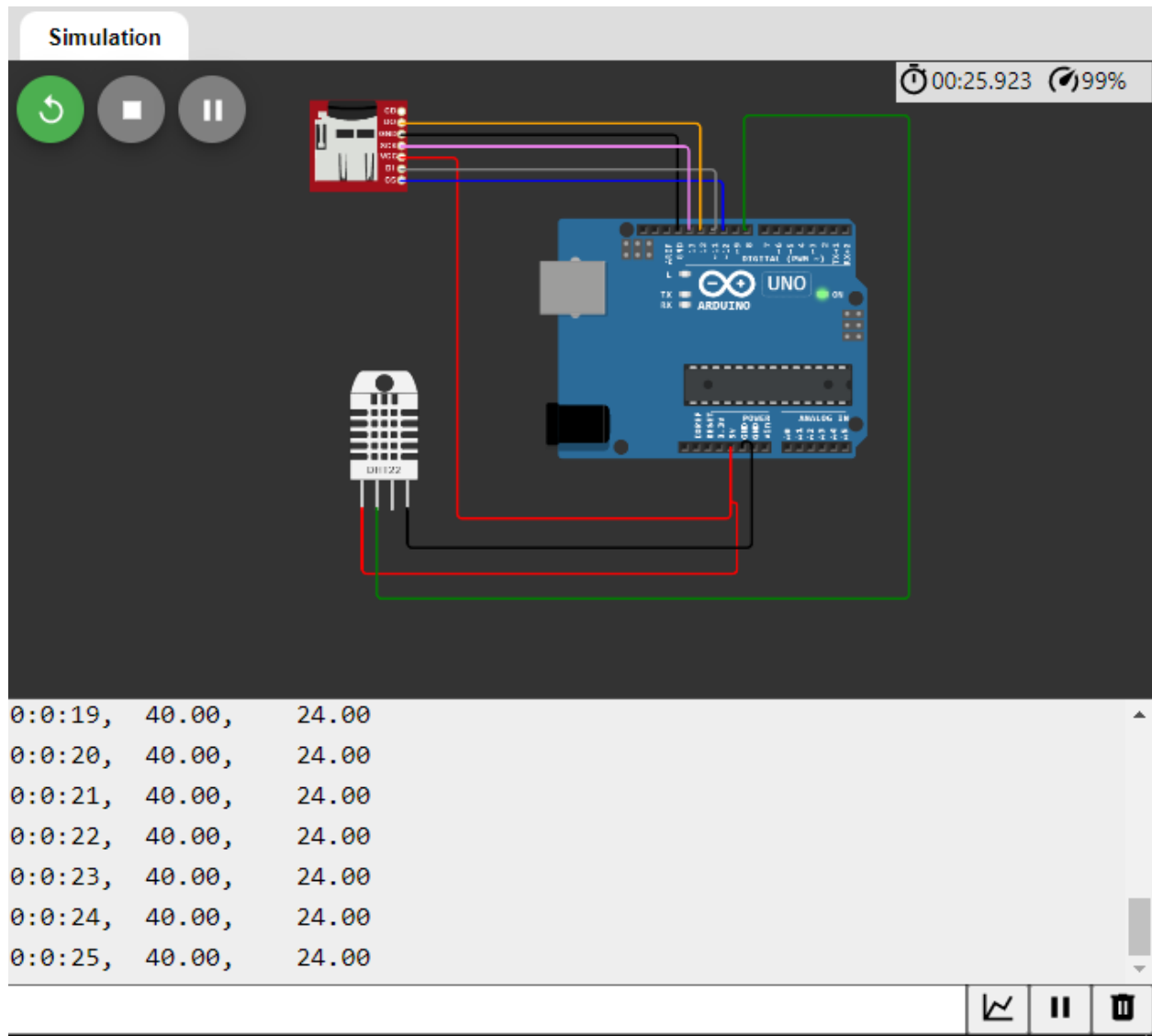
In the course, we first got basic information about embedded systems, then we learned the basic differences between hardware and software, then we learned how to use hardware and software together. With this knowledge, we developed small and basic programs in the course, as a result, in the light of the knowledge we gained in the course, we created this project that we present now.

## **6. References**

1. [5. DHT22 - Digital Temperature and Humidity Sensor \(readme.io\)](#)
2. [In-Depth Tutorial to Interface Micro SD Card Module with Arduino \(lastminuteengineers.com\)](#)
3. [What is Arduino? | Arduino](#)

## 7. Attachments

### 7.1. Project Images



*Figure 1 – Descriptions*



## 7.2. Code Modules

### 7.2.1. DHT\_Tester.ino

```
1  #include <SD.h>
2  #include "DHT.h"
3
4  #define DHTPIN 8
5  #define DHTTYPE DHT22
6
7  long seconds=00;
8  long minutes=00;
9  long hours=00;
10
11
12  int CS_pin = 10;
13
14  DHT dht(DHTPIN, DHTTYPE);
15  File sd_file;
16
17
18  void setup() {
19      Serial.begin(9600);
20      pinMode(CS_pin, OUTPUT);
21      dht.begin();
22      // SD Card Initialization
23      if (SD.begin()) {
24          Serial.println("SD card is initialized. Ready to go");
25      }
26      else {
27          Serial.println("Failed");
28          return;
29      }
30  }
31
32  File sd_file = SD.open("data.txt", FILE_WRITE);
33
34  if (sd_file) {
35      Serial.print("Time");
36      Serial.print(",");
37      Serial.print("Humidity");
38      Serial.print(",");
39      Serial.print("Temperature_C\n");
40
41      sd_file.print("Time");
42      sd_file.print(",");
43      sd_file.print("Humidity");
```

```

47     sd_file.print(",");
48     sd_file.print("Temperature_C");
49
50 }
51 sd_file.close(); //closing the file
52 }
53
54
55 void loop() {
56     File sd_file = SD.open("data.txt", FILE_WRITE);
57     if (sd_file) {
58         senddata();
59     }
60     // if the file didn't open, print an error:
61     else {
62         Serial.println("error opening file");
63     }
64     delay(1000);
65 }
66
67
68 void senddata() {
69     for(long seconds = 00; seconds < 60; seconds=seconds+1) {
70         float temp = dht.readTemperature(); //Reading the temperature as
71         Celsius and storing in temp
72         float hum = dht.readHumidity();      //Reading the humidity and
73         storing in hum
74
75
76         sd_file.print(hours);
77         sd_file.print(":");
78         sd_file.print(minutes);
79         sd_file.print(":");
80         sd_file.print(seconds);
81         sd_file.print(", ");
82         sd_file.print(hum);
83         sd_file.print(", ");
84         sd_file.print(temp);
85         sd_file.print("\n");
86
87
88         Serial.print(hours);
89         Serial.print(":");
90         Serial.print(minutes);
91         Serial.print(":");
92         Serial.print(seconds);
93         Serial.print(", ");
94         Serial.print(hum);
95
96

```

```
97     Serial.print(", ");
98     Serial.print(temp);
99     Serial.print("\n");
100
101
102     if(seconds>=59) {
103         minutes= minutes + 1;
104         seconds=0;
105     }
106
107
108     if (minutes>59) {
109         hours = hours + 1;
110         minutes = 0;
111     }
112
113     sd_file.flush(); //saving the file
114
115     delay(1000);
116 }
117 sd_file.close(); //closing the file
118 }
119 }
```