

## CAP 6133 Assignment-05

### Solution Proposed by Anmol Sureshkumar Panchal; UID:4446829

The DHT11 humidity and temperature sensor helps us to detect surrounding's humidity and temperature data when combined with Arduino Uno kit. It's perfect for remote weather stations, home environmental control systems, and farm or garden monitoring systems.

The DHT11 measures relative humidity. Relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in air.

The formula to calculate relative humidity is:

$$RH = \left( \frac{\rho_w}{\rho_s} \right) \times 100\%$$

*RH : Relative Humidity*

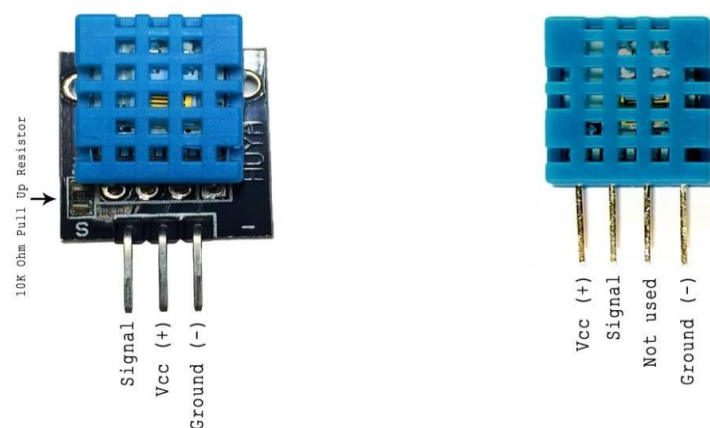
*$\rho_w$  : Density of water vapor*

*$\rho_s$  : Density of water vapor at saturation*

Relative humidity is expressed as a percentage. At 100% RH, condensation occurs, and at 0% RH, the air is completely dry.

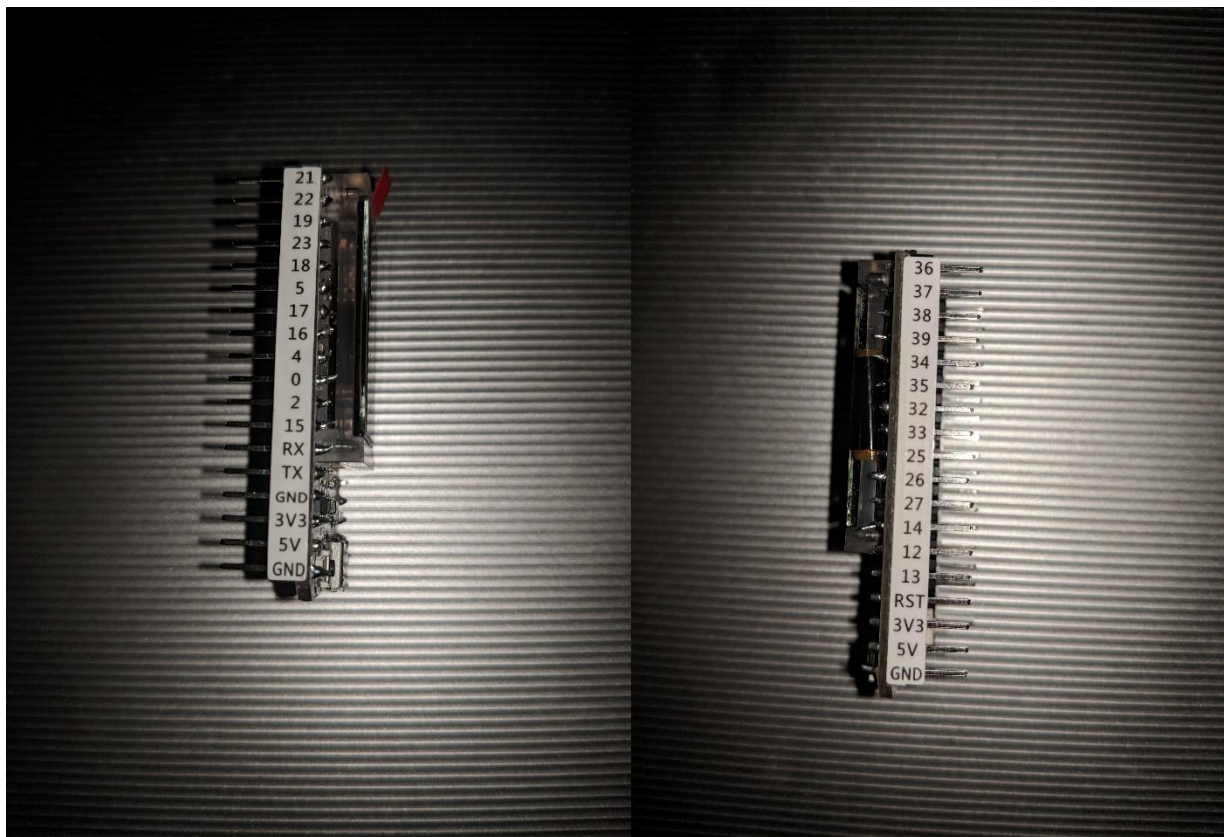
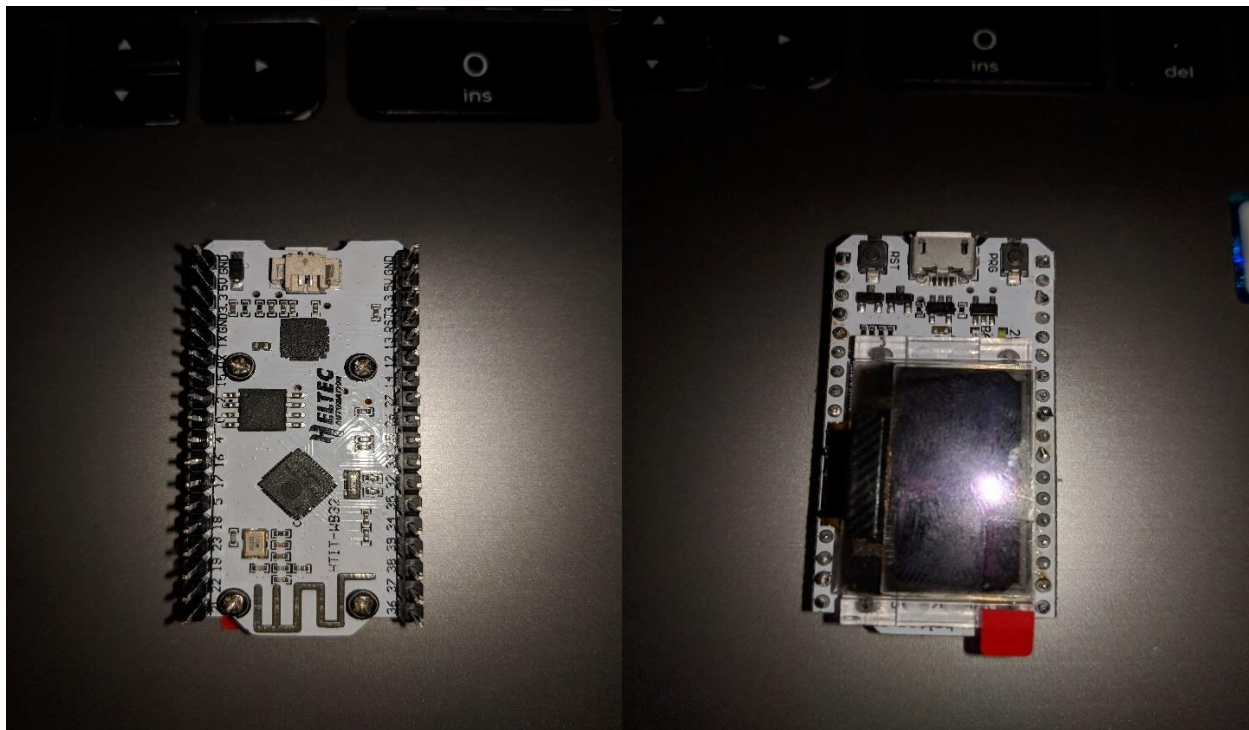
The DHT11 uses just one signal wire to transmit data to the Arduino. Power comes from separate 5V and ground wires. A 10K Ohm pull-up resistor is needed between the signal line and 5V line to make sure the signal level stays high by default (see the datasheet for more info).

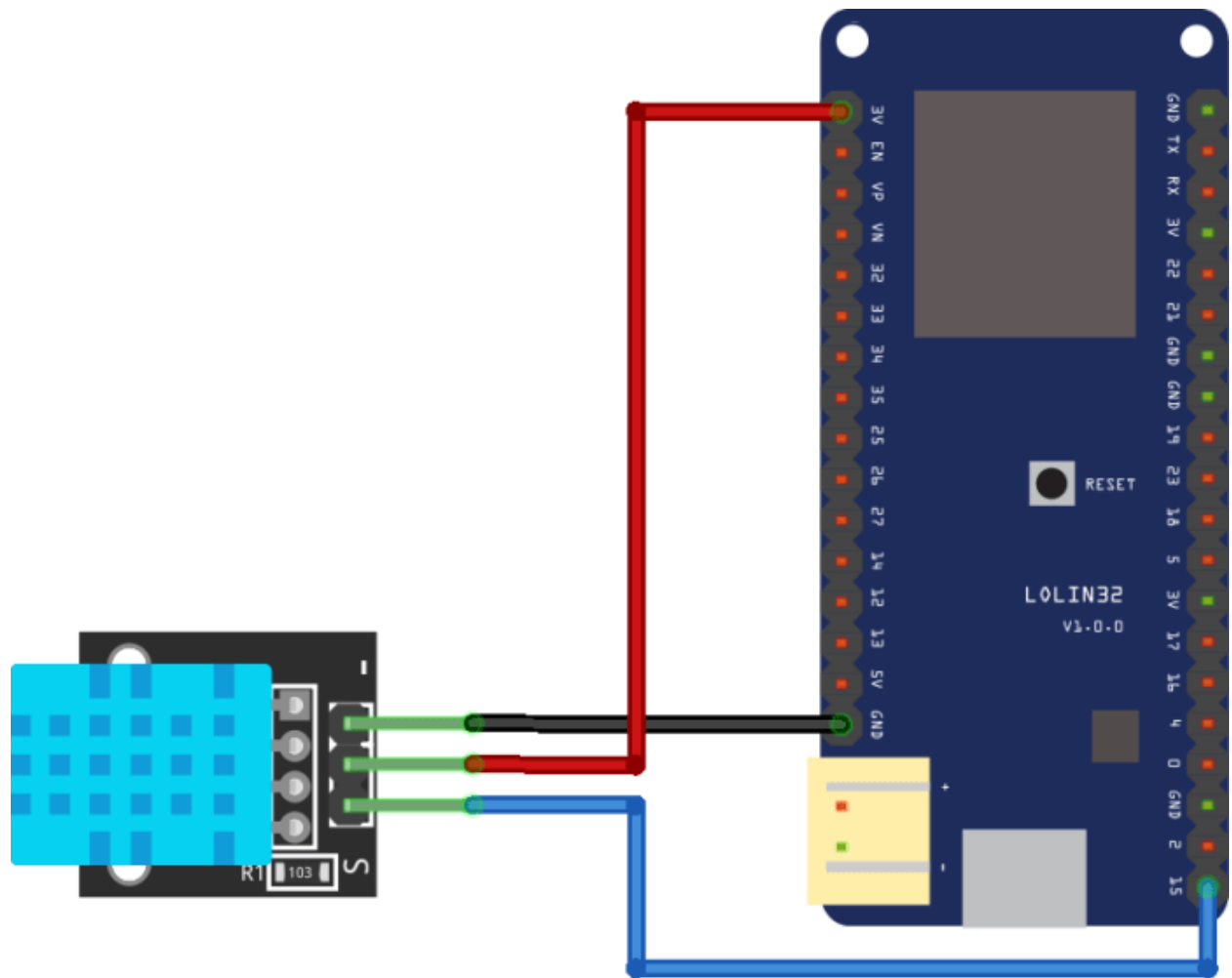
There are two different versions of the DHT11 you might come across. One type has four pins, and the other type has three pins and is mounted to a small PCB. We have a four pin DHT11 sensor in our kit. Here are the pin outs for both versions:



[Source: <http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>]

1. Show the picture of the finished ESP32 with the headers and labels. (1 point).

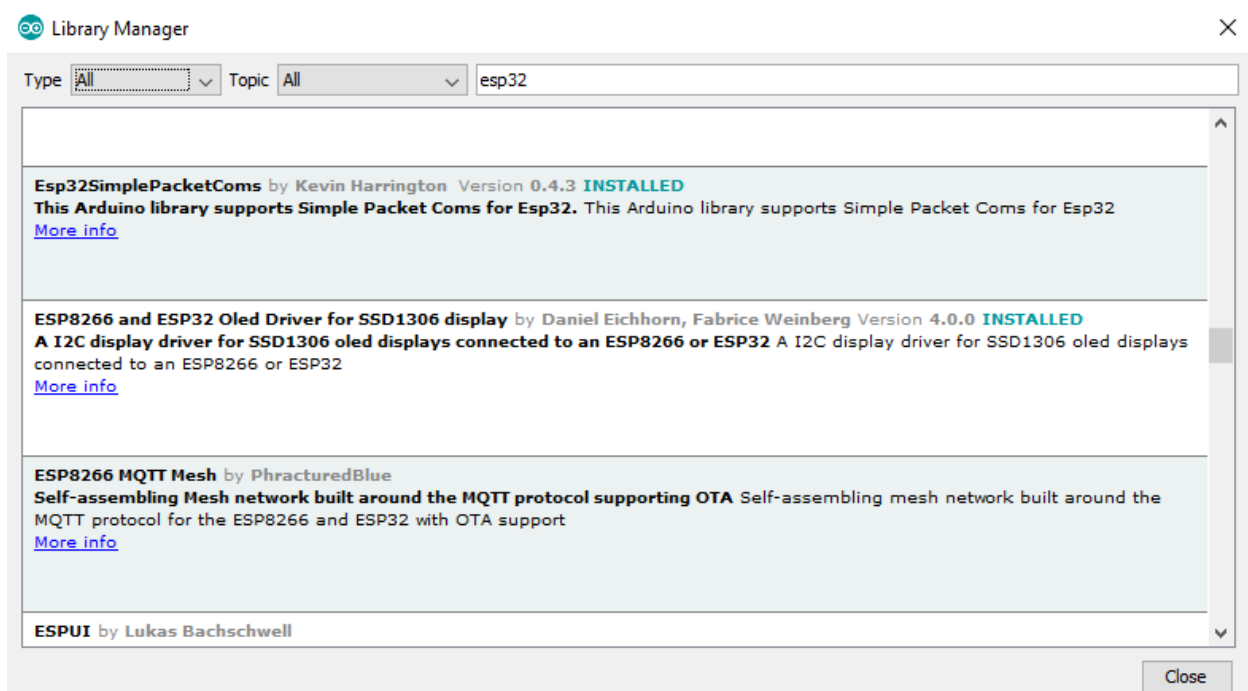
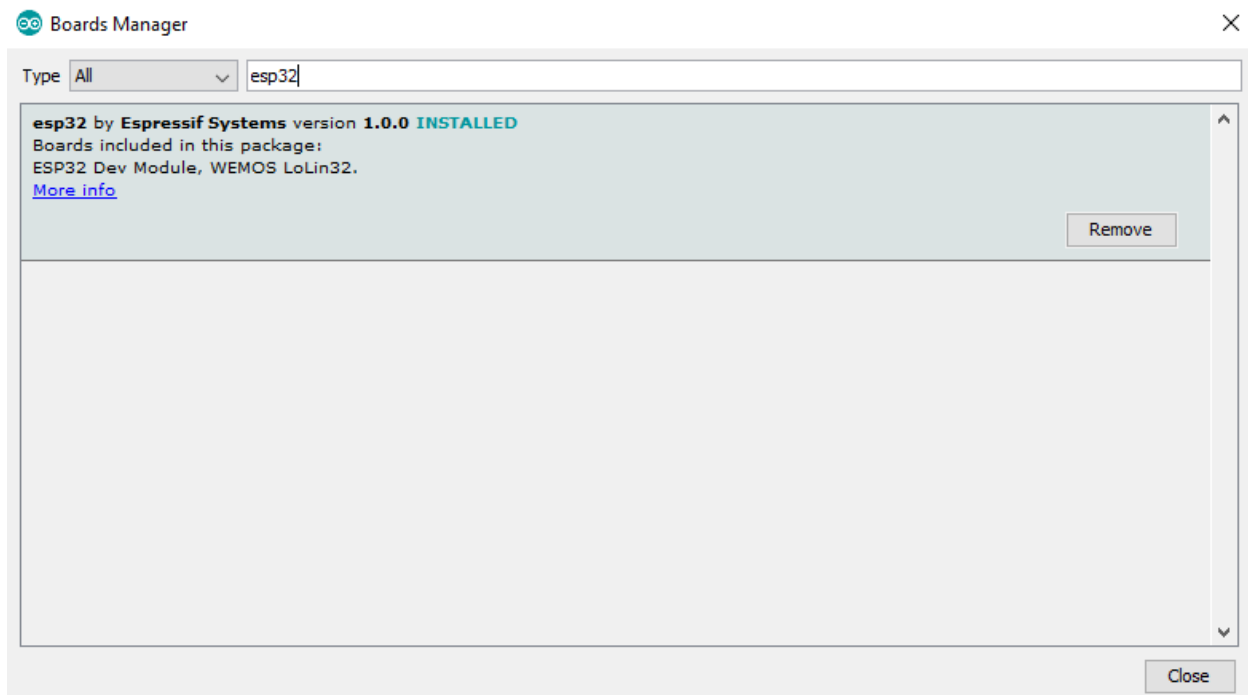


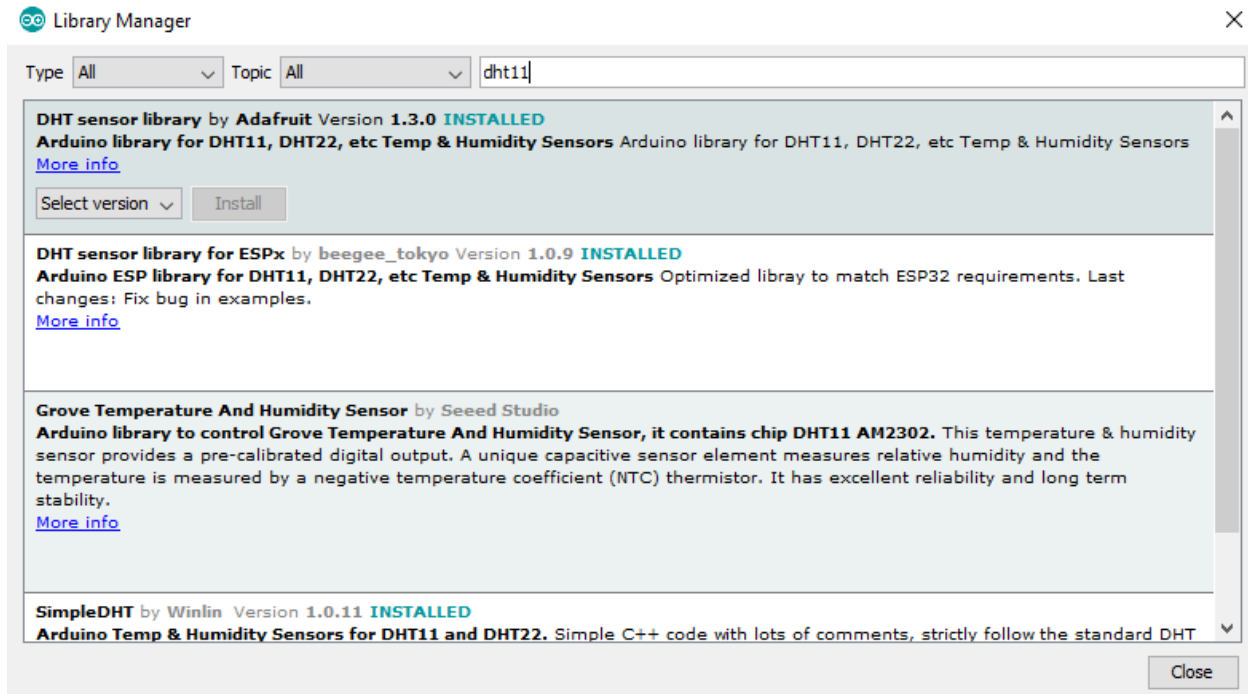


fritzing

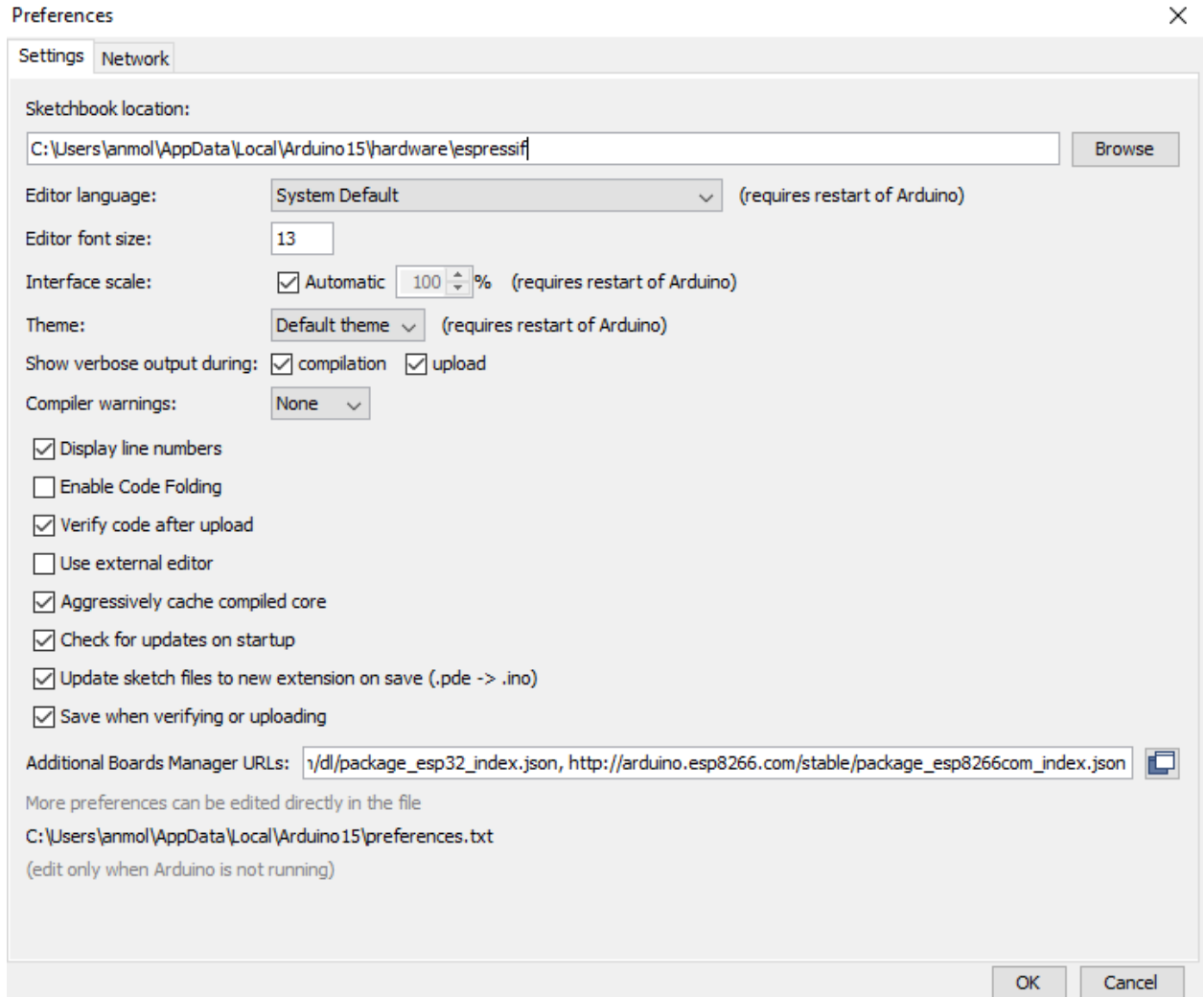
[Source: <http://www.esp32learning.com/code/esp32-and-dht11-sensor-example.php>]

## 2- Install the Arduino ESP32 and DHT libraries.

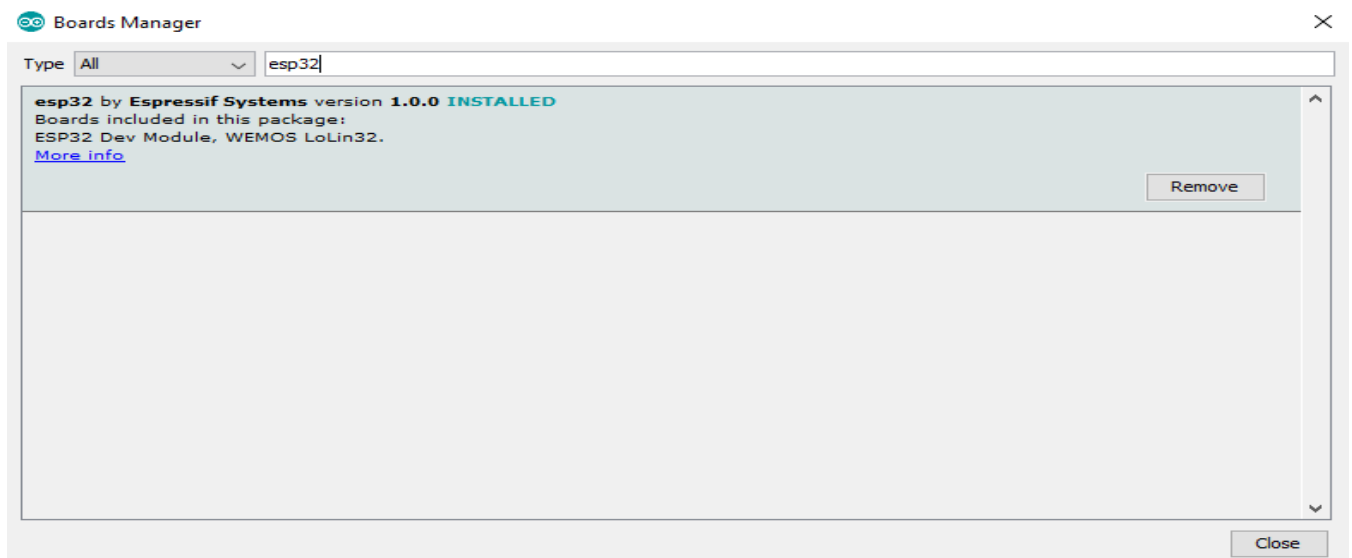


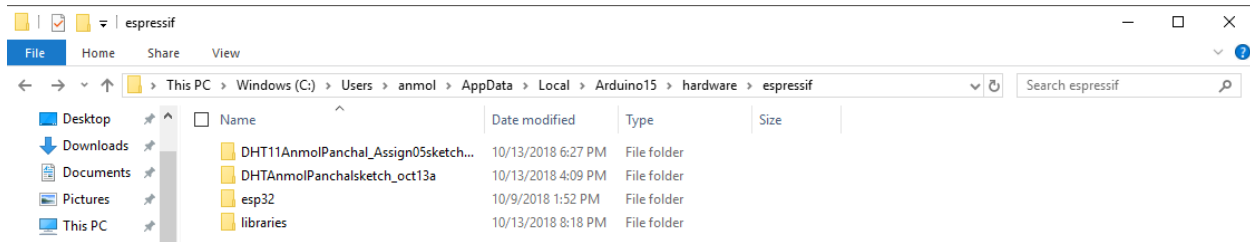


It will be installed and you can then set the path in preferences option of Arduino



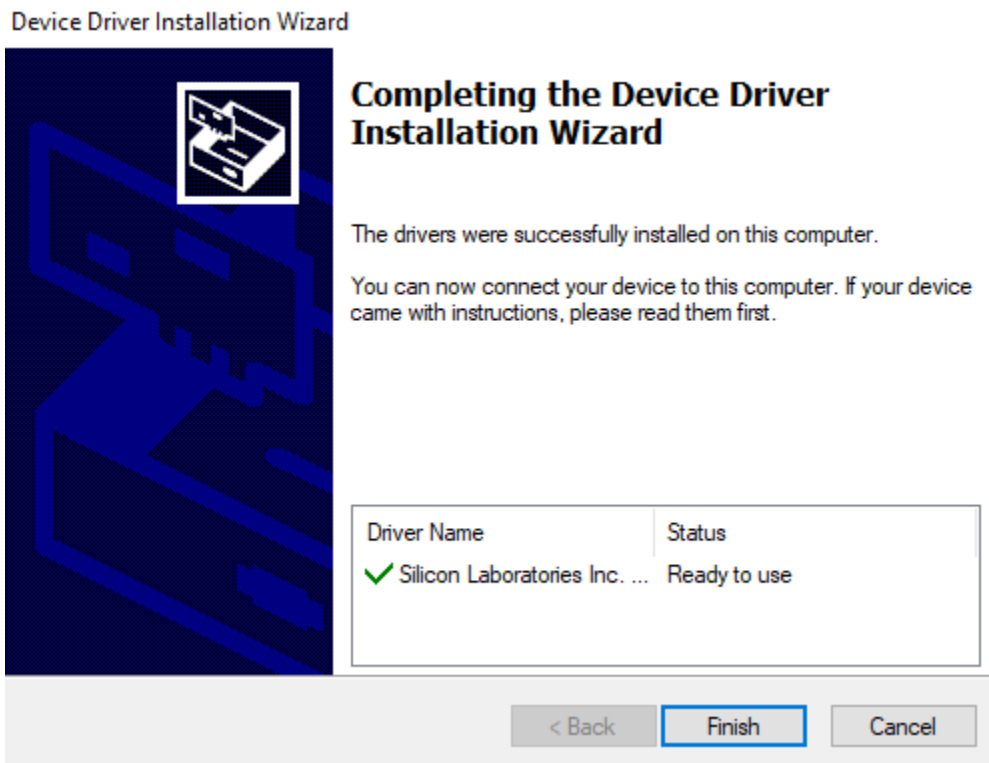
3- Show the picture of the installed ESP32 folder.





#### 4- Install the USB to UART driver.

Download USB to UART driver from <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers> and then extract the folder and install the CP210x driver.



## 5- Installing the latest Arduino IDE.

© DHT11AnmolPanchal\_Assign05sketch\_oct13b | Arduino 1.8.7

File Edit Sketch Tools Help

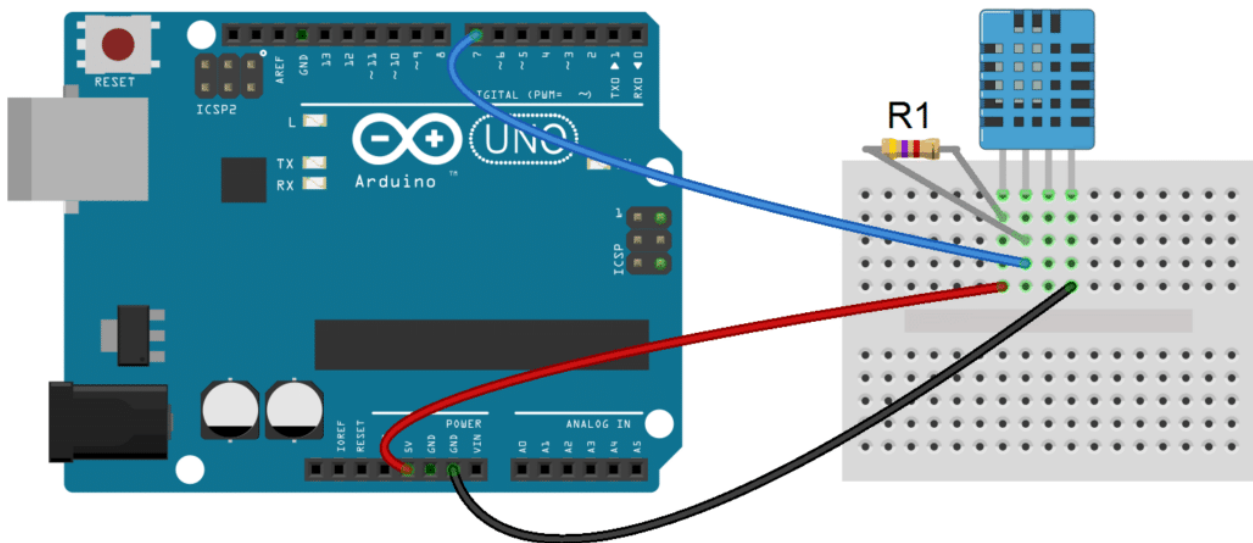
```
DHT11AnmolPanchal_Assign05sketch_oct13b

1 #include <SimpleDHT.h>
2
3 // for DHT11,
4 //     VCC: 5V or 3V
5 //     GND: GND
6 //     DATA: 2
7 int pinDHT11 = 2;
8 SimpleDHT11 dht11(pinDHT11);
9
10 void setup() {
11     Serial.begin(115200);
12 }
13
14 void loop() {
```

6.

I did the temperature and humidity detection using DHT sensor in two ways:

1. Arduino Kit
2. ESP32



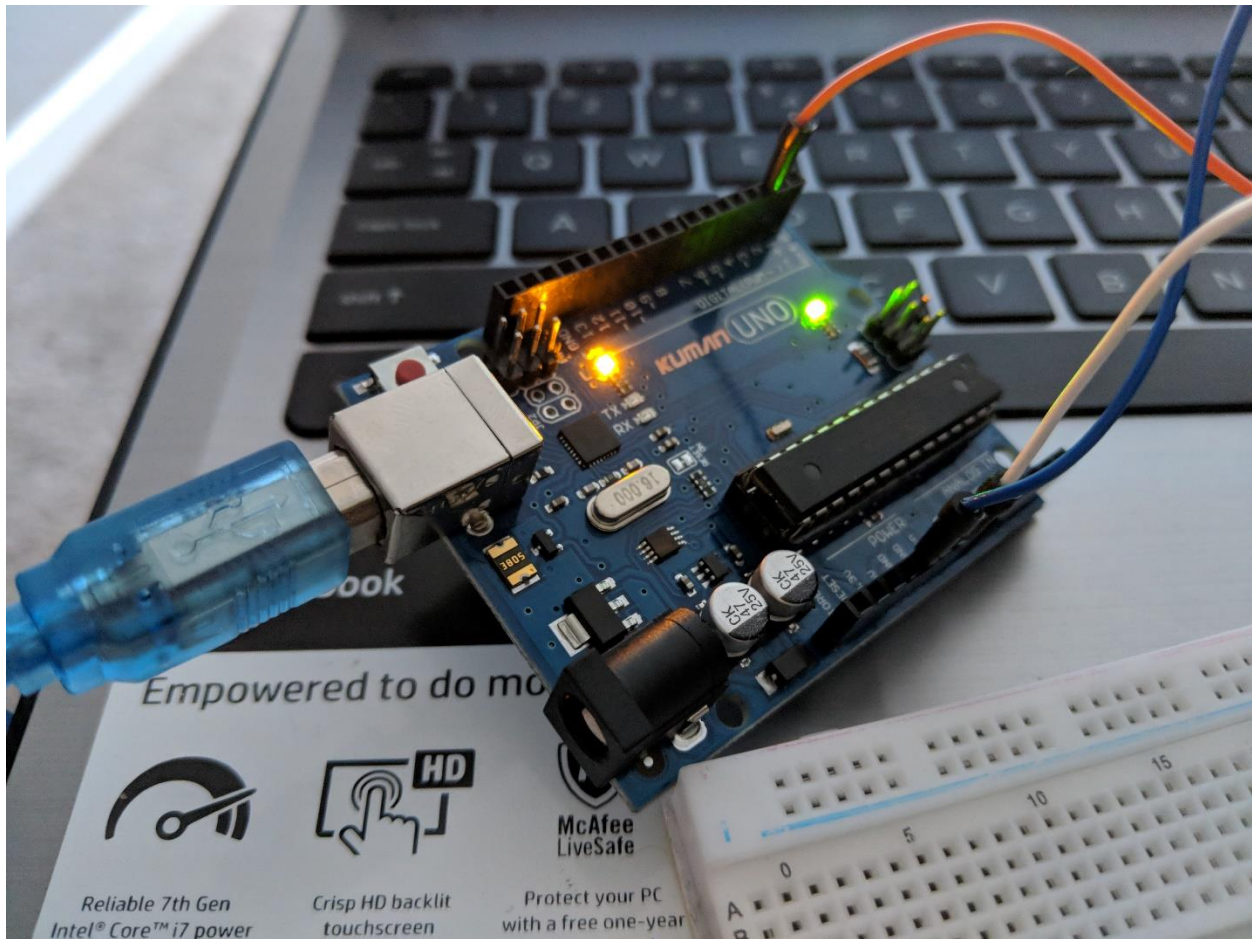
[Source: <http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>]

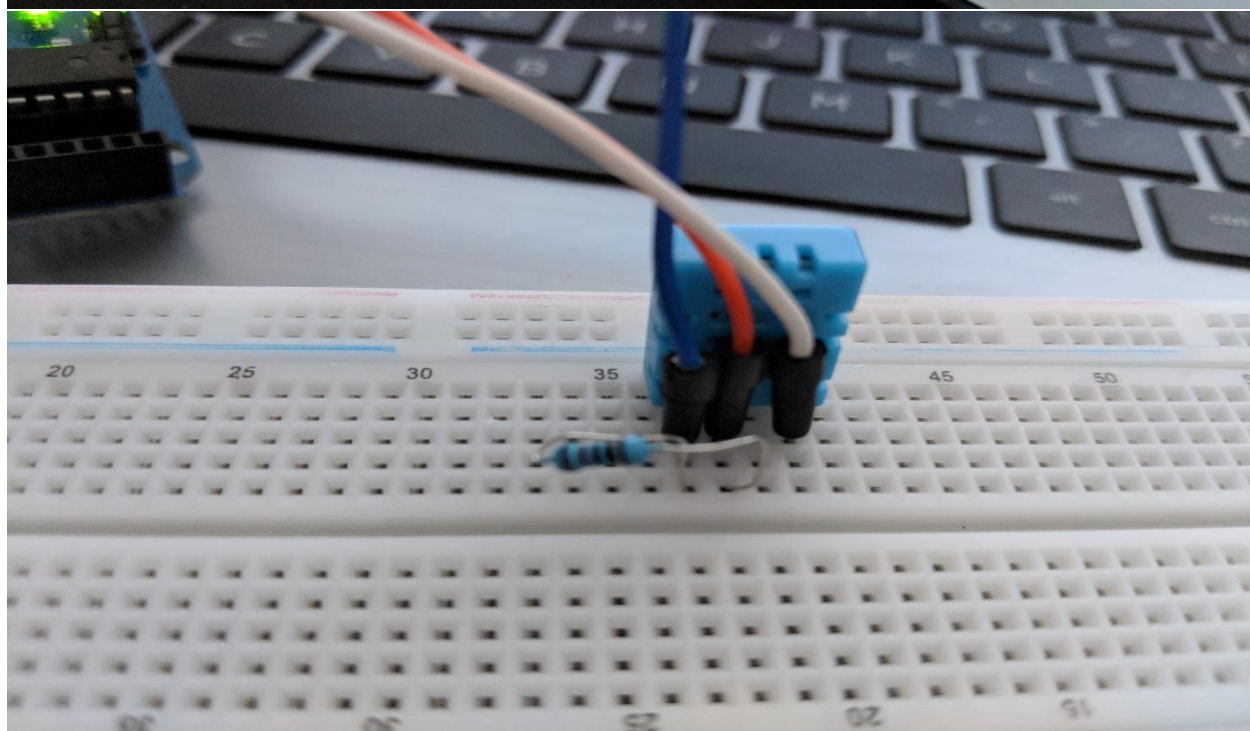
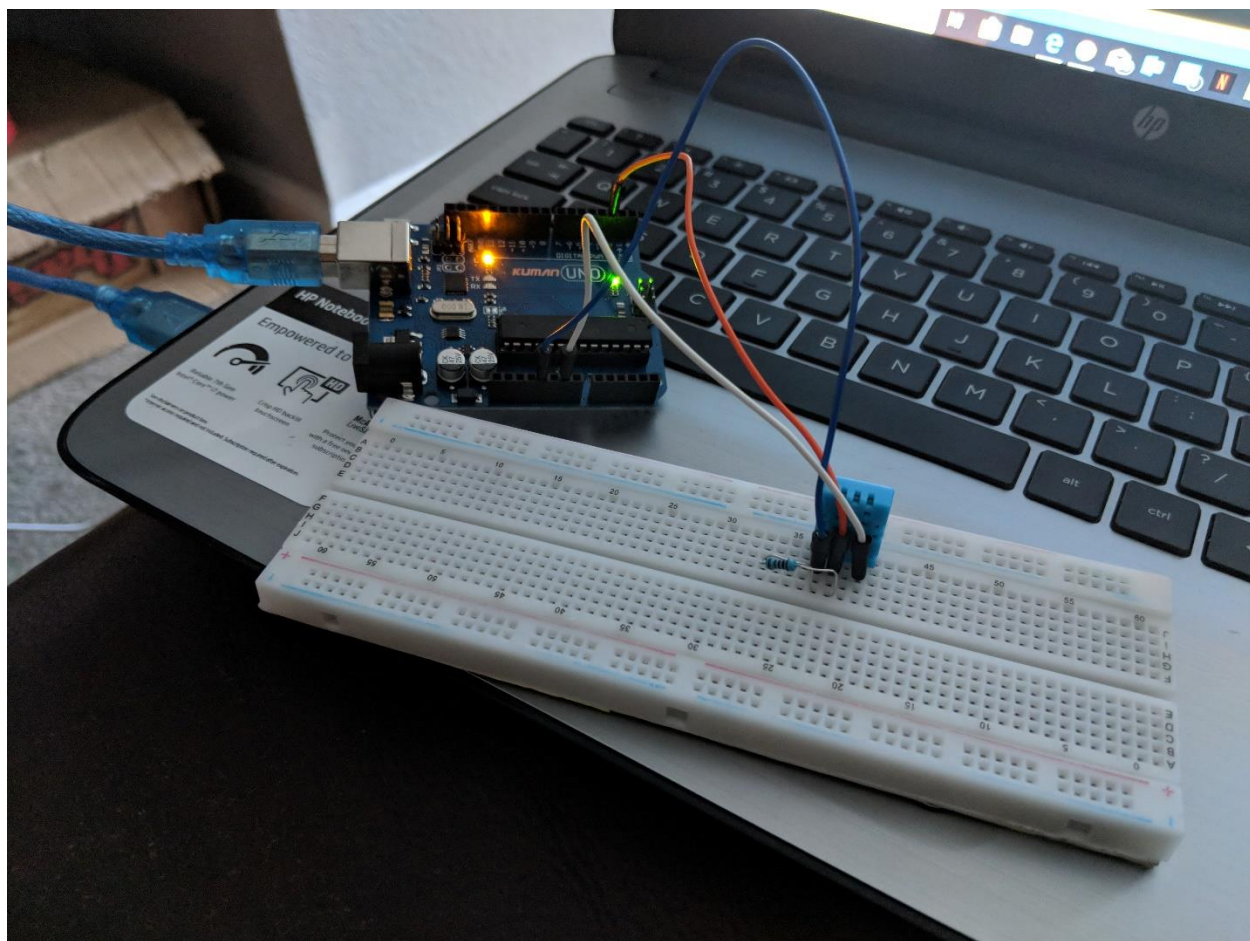


Here we have connected a resistor R1 which is a 10K Ohm pull up resistor.

So first we will see implementation using Arduino Uno kit.

After the connections are made properly we connect our Arduino Uno kit to the computer and open Arduino application where we run DHT sensor code from inbuilt DHT library.

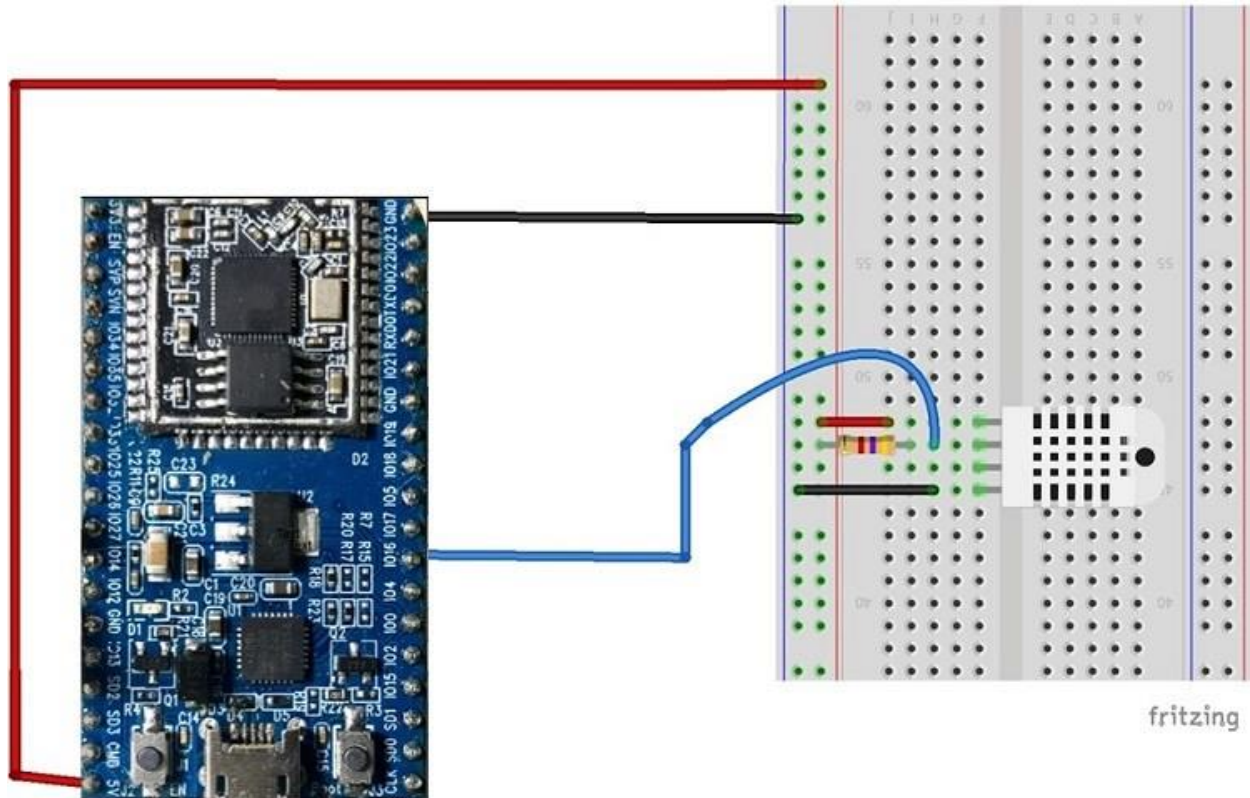




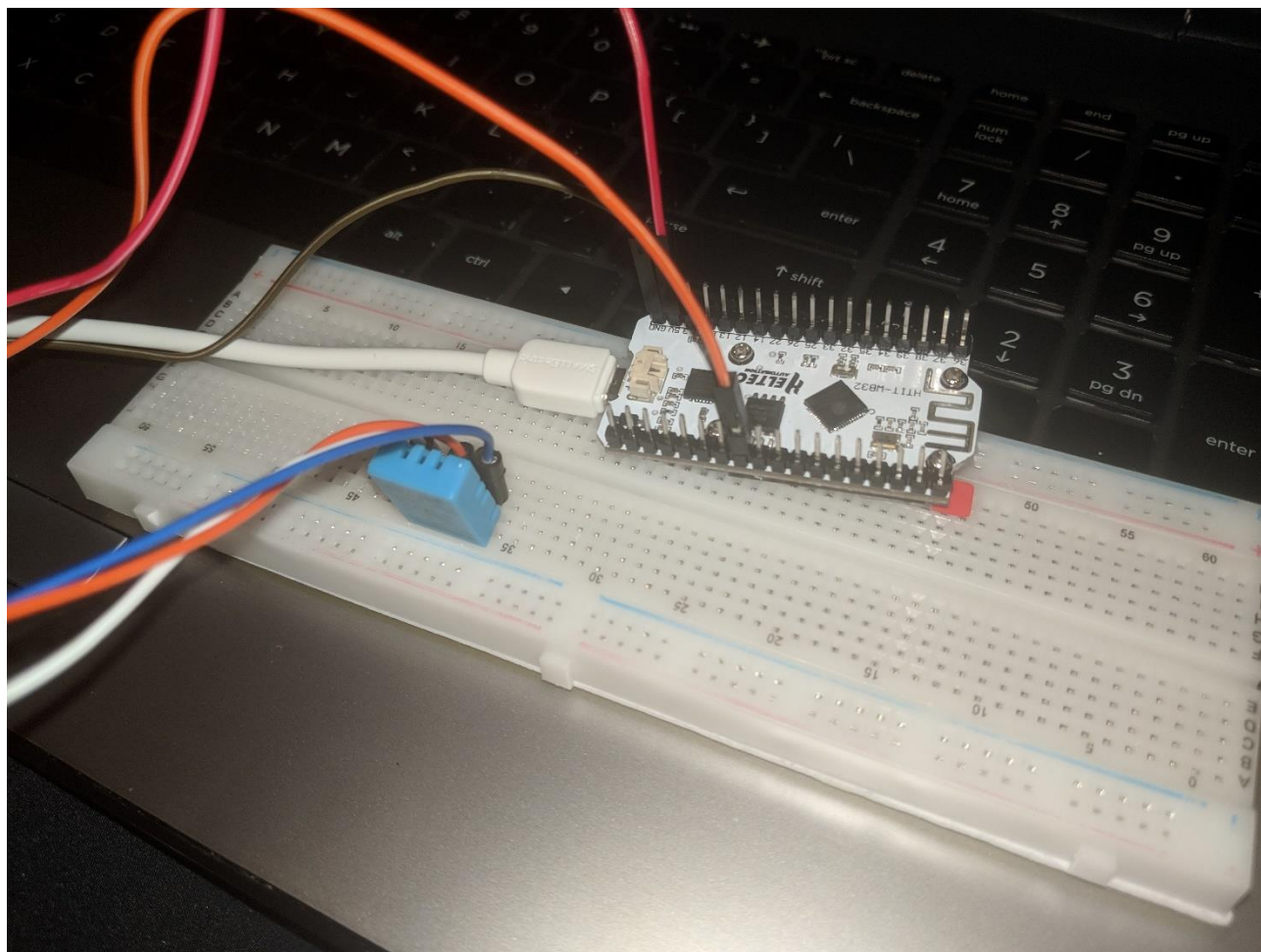


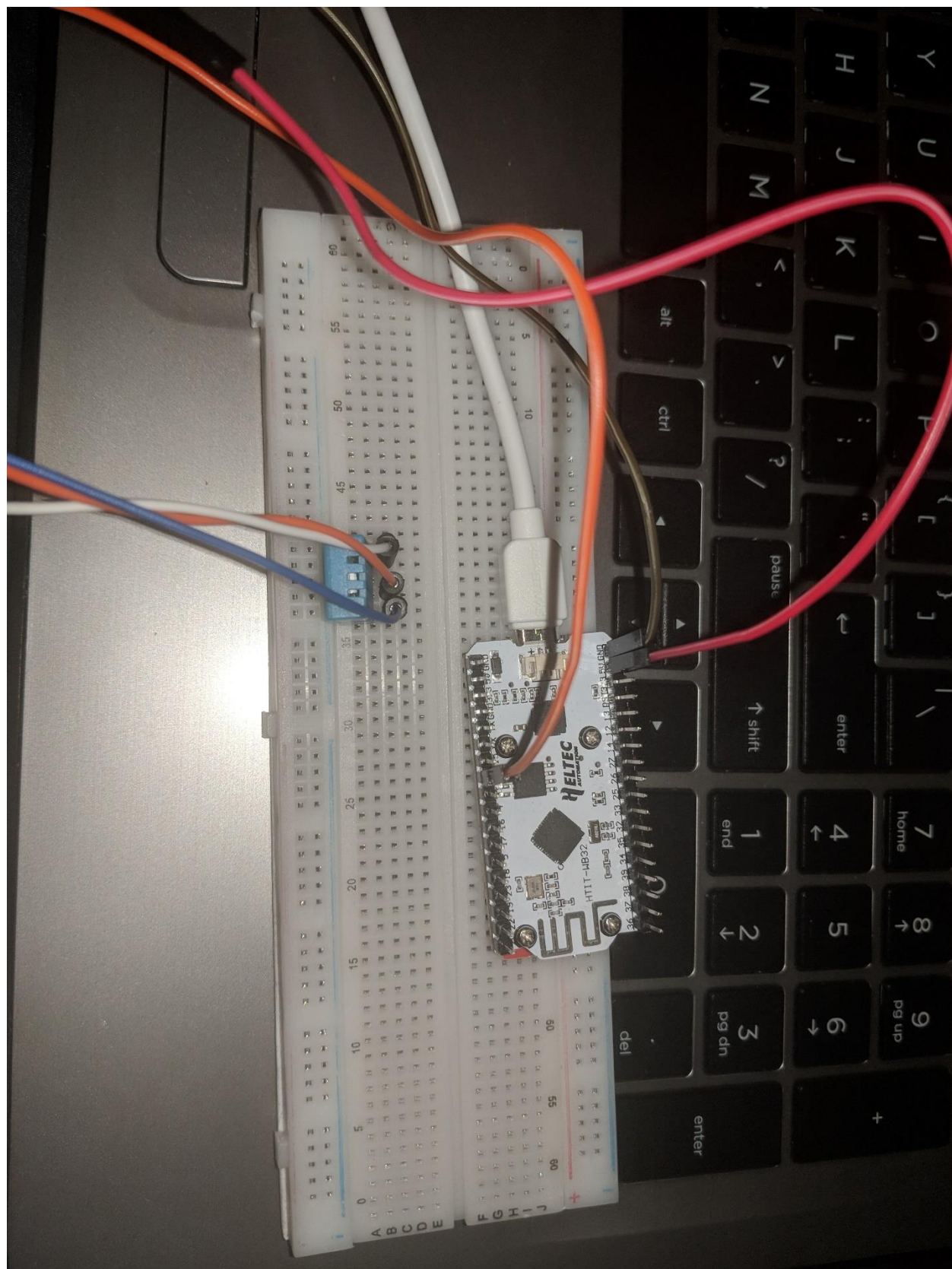
Now we implement using ESP32 kit as shown ahead;

After the connections are made properly we connect our ESP32 kit to the computer and open Arduino application where we run DHT sensor code from inbuilt DHT library.



[Source: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-web-server-arduino-ide/>]





You can see the **code** as below which is used same for both Arduino and ESP32 implementation:

```
#include <SimpleDHT.h>

//Assignment submission for Anmol Sureshkumar Panchal

// for DHT11,
//   VCC: 5V or 3V
//   GND: GND
//   DATA: 2

int pinDHT11 = 2;

SimpleDHT11 dht11(pinDHT11);

void setup() {
    Serial.begin(115200);
}

void loop() {
    // start working...

    Serial.println("=====");

    Serial.println("Sample DHT11...");

    // read without samples.

    byte temperature = 0;
    byte humidity = 0;

    int err = SimpleDHTErrSuccess;

    if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
        Serial.print("Read DHT11 failed, err="); Serial.println(err);delay(1000);
        return;
    }

    Serial.print("Sample OK: ");

    Serial.print((int)temperature); Serial.print(" *C, ");

    Serial.print((int)humidity); Serial.println(" H");

    // DHT11 sampling rate is 1HZ.

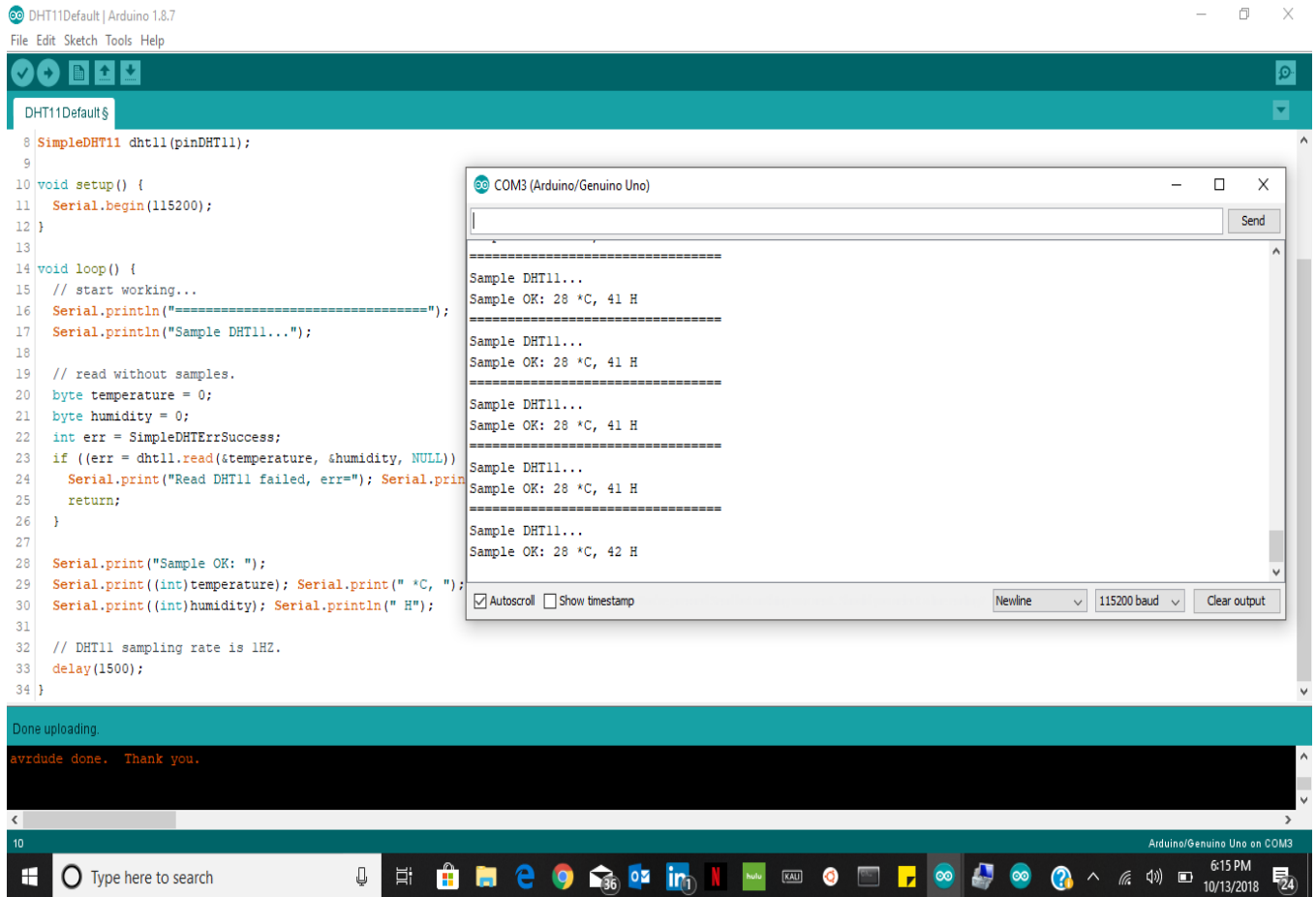
    delay(1500);
}
```

---



We got the desired output as follows:

1- Arduino Uno + DHT11 generated output.



The screenshot displays the Arduino IDE interface with a sketch named 'DHT11Default' open. The sketch code is as follows:

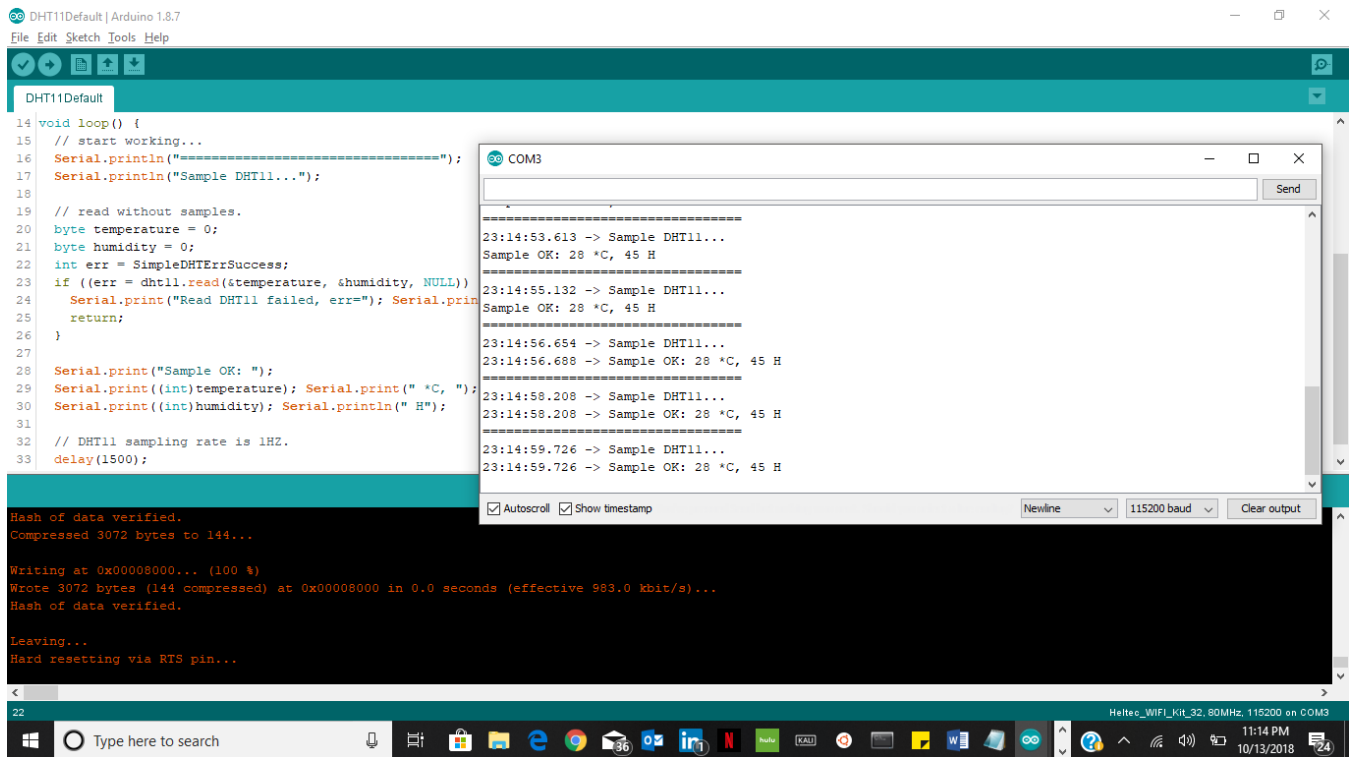
```
8 SimpleDHT11 dht11(pinDHT11);
9
10 void setup() {
11   Serial.begin(115200);
12 }
13
14 void loop() {
15   // start working...
16   Serial.println("=====");
17   Serial.println("Sample DHT11...");
18
19   // read without samples.
20   byte temperature = 0;
21   byte humidity = 0;
22   int err = SimpleDHT11.readSuccess;
23   if ((err = dht11.read(&temperature, &humidity, NULL))
24       Serial.print("Read DHT11 failed, err="); Serial.println(err);
25   return;
26 }
27
28 Serial.print("Sample OK: ");
29 Serial.print((int)temperature); Serial.print(" *C, ");
30 Serial.print((int)humidity); Serial.println(" H");
31
32 // DHT11 sampling rate is 1HZ.
33 delay(1500);
34 }
```

The serial monitor window, titled 'COM3 (Arduino/Genuino Uno)', shows the following output:

```
=====
Sample DHT11...
Sample OK: 28 *C, 41 H
=====
Sample DHT11...
Sample OK: 28 *C, 41 H
=====
Sample DHT11...
Sample OK: 28 *C, 41 H
=====
Sample DHT11...
Sample OK: 28 *C, 41 H
=====
Sample DHT11...
Sample OK: 28 *C, 42 H
```

The IDE status bar at the bottom indicates 'Done uploading.' and 'avrduide done. Thank you.' The Windows taskbar at the very bottom shows the time as 6:15 PM on 10/13/2018.

## 2- ESP32 + DHT11 generated output.



The screenshot displays the Arduino IDE interface. The main window shows the 'DHT11Default' sketch, which is a C++ program for reading data from a DHT11 sensor using the ESP32. The code includes comments and function calls for reading temperature and humidity. The serial monitor window, titled 'COM3', shows the output of the program, which includes timestamps, sample numbers, and the detected temperature and humidity values. The output shows that the temperature is 28 °C and the humidity is 45 H. The IDE status bar at the bottom indicates that the board is 'Heitec\_WIFI\_Kit\_32, 80MHz, 115200 on COM3' and the time is 11:14 PM on 10/13/2018.

```
14 void loop() {
15   // start working...
16   Serial.println("=====");
17   Serial.println("Sample DHT11...");
18
19   // read without samples.
20   byte temperature = 0;
21   byte humidity = 0;
22   int err = SimpleDHTErrSuccess;
23   if ((err = dht11.read(&temperature, &humidity, NULL))
24       Serial.print("Read DHT11 failed, err="); Serial.println(err);
25   return;
26 }
27
28 Serial.print("Sample OK: ");
29 Serial.print((int)temperature); Serial.print(" *C, ");
30 Serial.print((int)humidity); Serial.println(" *H");
31
32 // DHT11 sampling rate is 1HZ.
33 delay(1500);
```

Hash of data verified.  
Compressed 3072 bytes to 144...

Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (144 compressed) at 0x00008000 in 0.0 seconds (effective 983.0 kbit/s)...  
Hash of data verified.

Leaving...  
Hard resetting via RTS pin...

23:14:53.613 -> Sample DHT11...  
Sample OK: 28 \*C, 45 H  
=====

23:14:55.132 -> Sample DHT11...  
Sample OK: 28 \*C, 45 H  
=====

23:14:56.654 -> Sample DHT11...  
23:14:56.688 -> Sample OK: 28 \*C, 45 H  
=====

23:14:58.208 -> Sample DHT11...  
23:14:58.208 -> Sample OK: 28 \*C, 45 H  
=====

23:14:59.726 -> Sample DHT11...  
23:14:59.726 -> Sample OK: 28 \*C, 45 H  
=====

Autoscroll Show timestamp Newline 115200 baud Clear output

The results generated were the outcome of the code and experimental setup with the DHT 11 sensor and Arduino Uno kit performed at my home. We get the output of this code above where I get **Temperature** to be detected as **28 degree Celsius** and **Humidity** as **41 H** in **Arduino** and as **28 degree Celsius** and **Humidity** as **41 H** using **ESP32**. And when compared with current temperature on google.com for accuracy testing as shown in image below you can see we get **100% accuracy** for **temperature** detected but for **humidity** google shows **50%** whereas we got **41 H** that's because I have performed this inside my home where humidity is very more controlled than the outside environment. Hence explaining the difference between the two results. But I can successfully say results were successfully generated accurately.



Search instead for: current temperature in Celsius



US Weather - Current Temperatures Map (Celsius) | WeatherCentral ...

[https://www.weathercentral.com/weather/us/maps/current\\_temperatures\\_c.html](https://www.weathercentral.com/weather/us/maps/current_temperatures_c.html) ▼

US Weather - Current Temperatures Map (Celsius). The US National Current Temperatures map is



Below image shows esp32 kit is connected to my home Wifi named “Vikings”.

The screenshot displays the Arduino IDE interface with a sketch named 'sketch\_oct09a'. The code is as follows:

```
1 #include <WiFi.h>
2
3 const char* ssid = "Vikings";
4 const char* password = "RagnarLothbrok";
5
6 void setup() {
7
8     Serial.begin(115200);
9
10    WiFi.begin(ssid, password);
11
12    while (WiFi.status() != WL_CONNECTED) {
13        delay(500);
14        Serial.println("Connecting to WiFi..");
15    }
16
17    Serial.println("Connected to the WiFi network");
18
19 }
20
21 void loop() {
22
23 }
```

Overlaid on the IDE is a 'COM3' serial monitor window. It shows the following output:

```
Connecting to WiFi..
Connecting to WiFi..
Connecting to WiFi..
Connected to the WiFi network
```

At the bottom of the IDE, a status bar indicates 'Done uploading.' and 'Leaving... Hard resetting via RTS pin...'. The Windows taskbar at the very bottom shows the system clock as 3:51 PM on 10/13/2018.