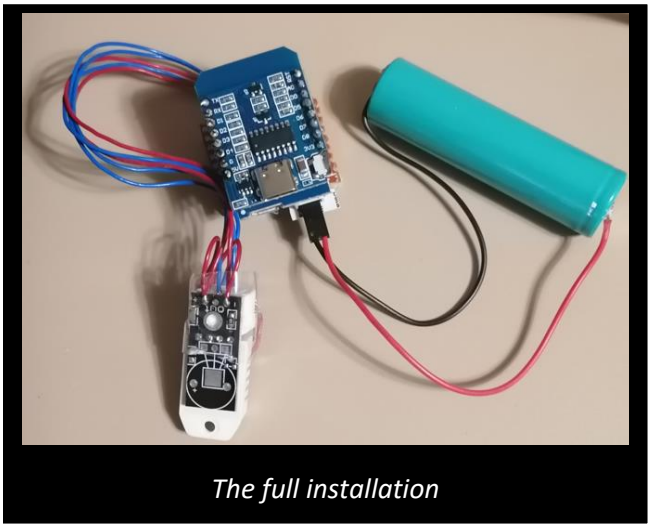
DMP Documentation – Deac Dan Cristian / 30433

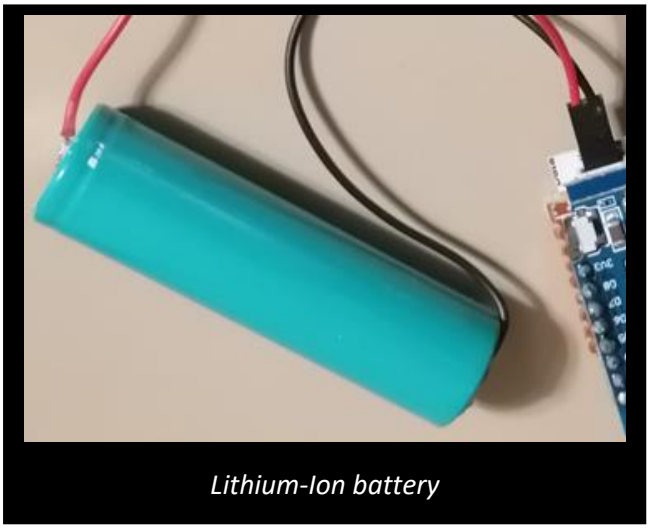
Intro:

Plant is a weather station that I tried to make as useful as possible. It is designed to work on a Lithium-Ion battery that can be charged through a micro usb, and then placed somewhere in your room where it looks good and can be used any time through your network a site you can always connect to and check the Temperature and humidity of the room in which the plant was placed on.

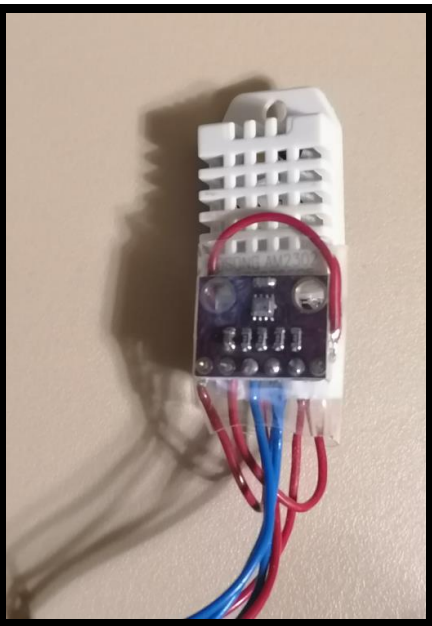
Hardware:



The full installation



Lithium-Ion battery

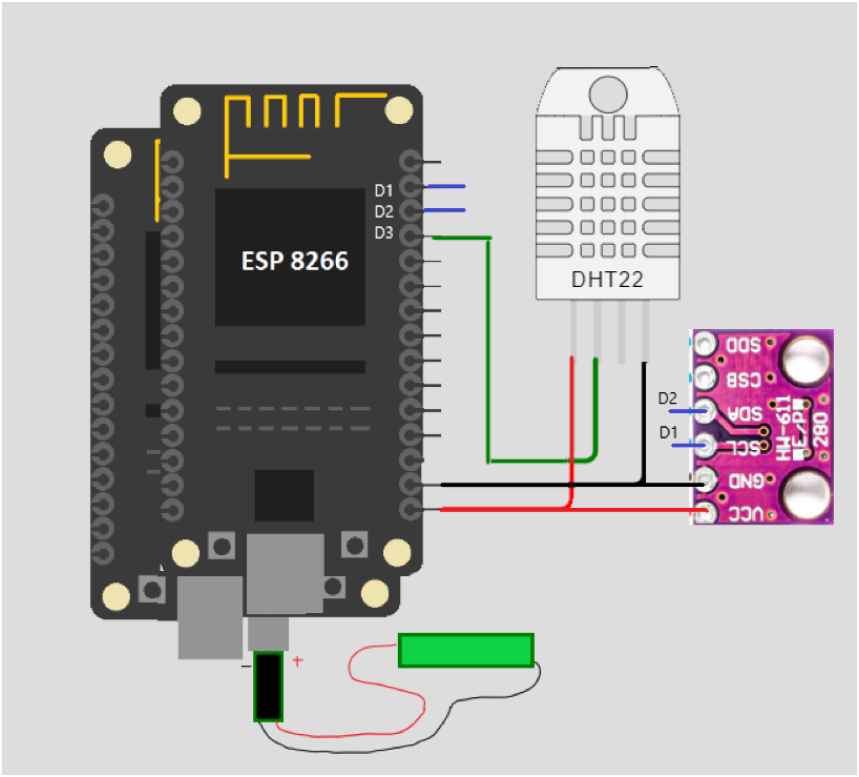


Temperature and humidity – DHT22
Air pressure BMP280

Battery shield

Esp8266 – d1 mini

Schematic:



Schematic

This schematic shows how the all the pieces are connected. Unfortunately, so far, I have not managed to make all the components work together. For now, it works well with the humidity and temperature sensors, though the BMP sensor can get the humidity and air pressure, I figured it wasn't really that important, though I included it either way in the schematic.

Code:

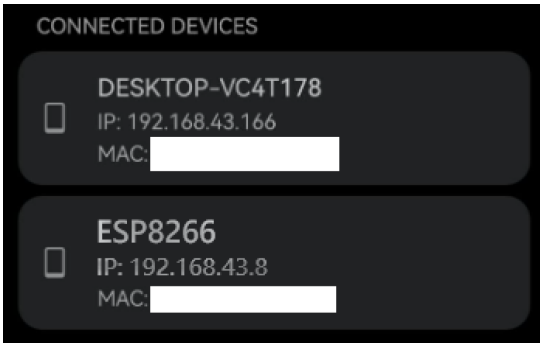
```
#include <ESP8266WiFi.h>
#include <DHT.h>

#define DHTPIN 0  // Pin for connecting the DHT sensor - I believe its the GPIO 0
#define DHTTYPE DHT22
WiFiServer server(80);
DHT dht(DHTPIN, DHTTYPE);

const char* ssid = "HUAWEI-P30-lite";
const char* password = "******";

float t = 0.0, h = 0.0;
unsigned long previousMillis = 0;  // will store last time DHT was updated

// Updates DHT readings every 10 seconds
const long interval = 10000;

String processor(const String& var) {
  if (var == "TEMPERATURE") {
    return String(t);
  } else if (var == "HUMIDITY") {
    return String(h);
  }
  return String();
}

void setup() {
  // Start the serial communication
  Serial.begin(115200);
  dht.begin();
  // Connect to the WiFi network
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
```

Unfortunately I did not manage to get the code working without using the given libraries and because of having to deal with some hardware issues (uploading code to the board was only possible with the shield disconnected) I did not manage to get them working in the time necessary

The following code will be for reading the sensors (**loop**) :

```
void loop() {

  // Print the values to the serial monitor
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    // save the last time you updated the DHT values
    previousMillis = currentMillis;
    // Read temperature as Celsius (the default)
    float newT = dht.readTemperature();

    if (isnan(newT)) {
      Serial.println("Failed to read from DHT sensor!");
    } else {
      t = newT;
    }
    float newH = dht.readHumidity();
    // if humidity read failed, don't change h value
    if (isnan(newH)) {
      Serial.println("Failed to read from DHT sensor!");
    } else {
      h = newH;
    }
  }
```

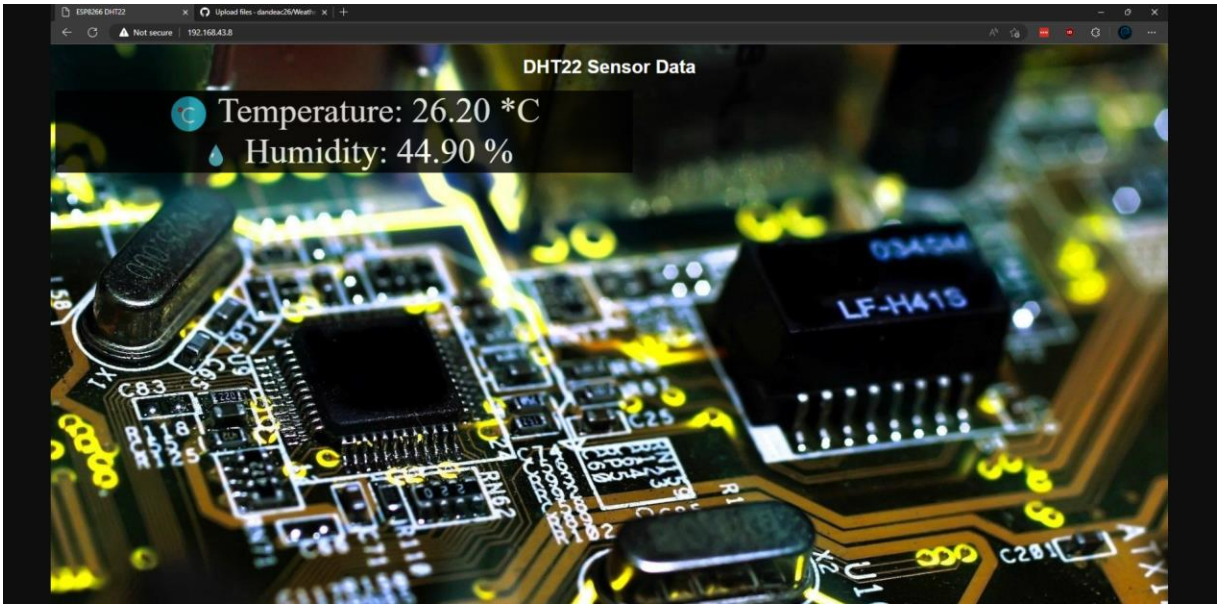
To sum things up before getting to the wepage, I am using the board to connect to the WIFI and then have the board be assigned an IP address where the users connected to the same network have access to the information from the sensors. This can be easily seen on my phone hotspot. I can see the ESP clearly on the network created between them and the IP I need to connect to:



Webpage code :

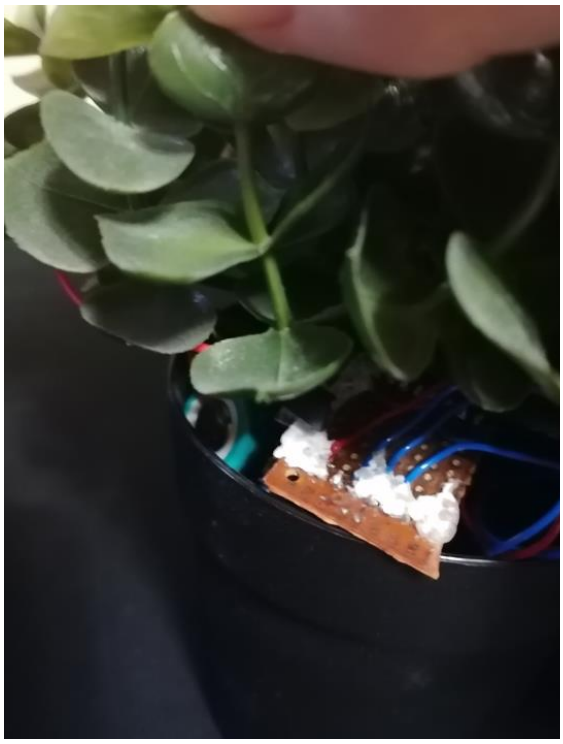
```
  //Handle client connections
  WiFiClient client = server.available();
  if (client) {
    // Print the values to the web page
   client.println("<!DOCTYPE html>");
    client.println("<html>");
    client.println("<head>");
    client.println("<title>ESP8266 DHT22</title>");
    client.println("<style>");
    client.println("body {background-color: #f2f2f2;}");
    client.println("h1 {text-align: center; font-family: Arial, sans-serif;color:white;}");
    client.println(".sensor-value {font-size: 60px; background-color:black; opacity:0.8;width:50%; color:white;text-
align:center;}");
    client.println("img.icon { width: 60px; height: 60px; margin-right: 20px; vertical-align: middle;}");
    client.println(".temperature { background-repeat: no-repeat; padding-left: 40px; }");
    client.println(".humidity { background-repeat: no-repeat; padding-left: 40px;}");
    client.println("body { background-image: url('https://wallpaper.dog/large/427491.jpg'); background-repeat: no-
repeat;background-size: cover;}");
    client.println("</style>");
    client.println("</head>");
    client.println("<body>");
    client.println("<h1>DHT22 Sensor Data</h1>");
    client.println("<div class='sensor-value temperature'><img class='icon'
src='https://icons.iconarchive.com/icons/graphicloads/folded/48/c-temperature-folded-icon.png'><span>Temperature: ");
    client.print(t);
    client.println(" *C</span></div>");
    client.println("<div class='sensor-value humidity'><img class='icon' src='https://icons.iconarchive.com/icons/custom-
icon-design/lovely-weather-2/64/Humidity-icon.png'><span>Humidity: ");
    client.print(h);
    client.println(" %</span></div>");
    client.println("</body>");
    client.println("</html>");
    client.stop();
  }

  delay(5000);
}
```

Here must be noted that I am using several images from the internet to get the background and the icons for the components (icons.iconarchive.com)



Conclusions

Here is the final product, where I introduced the components in a plastic plant that is a great case for this project in my opinion:



In this project I learnt a lot of things about hardware and especially debugging issues. I was hit for example with an issue where I couldn't upload code to the board. After intense debugging and using PlatformIO as well, I figured out in the end that the issue was not with something software related, it was something related to the wiring. I surely did something I should not have, but I found an easy solution. I would just remove the shield and upload the code to the empty ESP and then disconnect the cables, connect the board to the shield and connect again to the pc for power. It was really rewarding after managing to do this but at the same time I lost a lot of time that I didn't have. At least now I have experience and I will be able to debug things faster in the future.

Here is the GitHub link to the project: https://github.com/dandeac26/WeatherStation.git