



SD Card Module With ESP8266



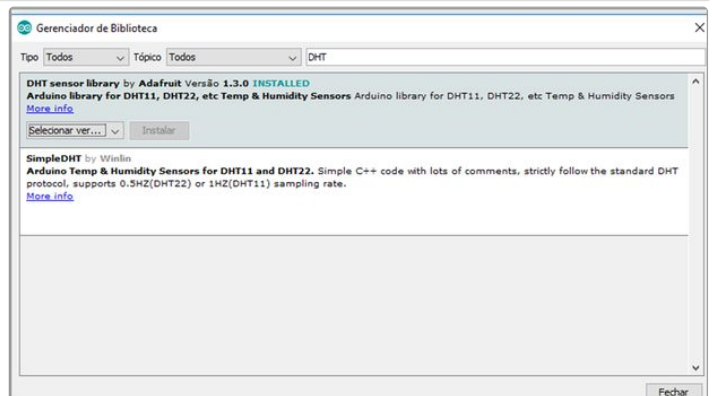
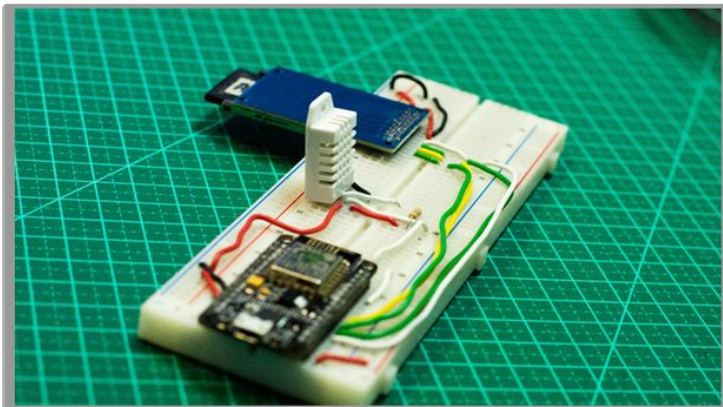
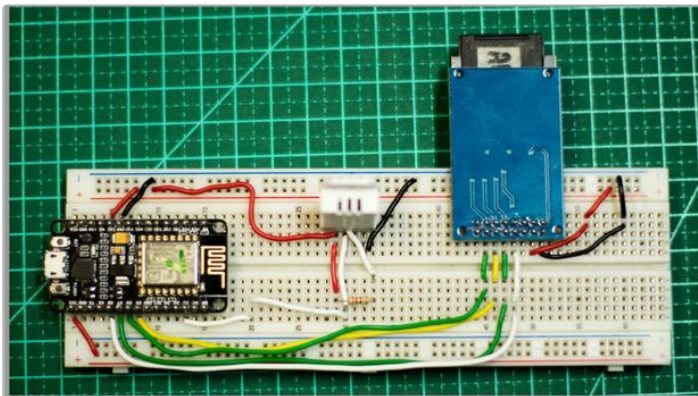
by Fernando Koyanagi

In this assembly, we have an SD Card connected to the ESP8266. We put a DHT22, which measures temperature and humidity and sends this information to the SD card.

On the circuit, it shows humidity of 43.40 and a temperature of 26.80. Every time that it shows the

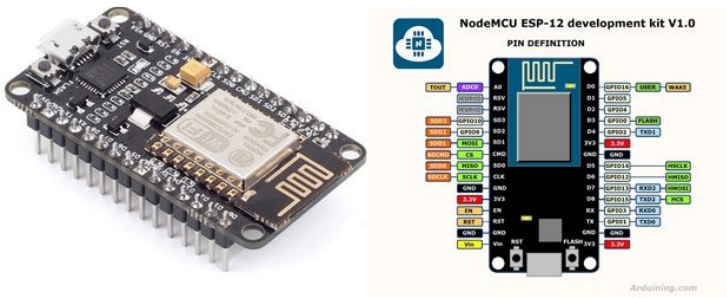
message "opening the file successfully," it's because it ran once in the loop. What occurs in this scenario is as follows: only the values are being written to the log file, and thus, the message "opening the file successfully" is only an advisory, and it is not recorded.

https://youtu.be/31_YFnHep60



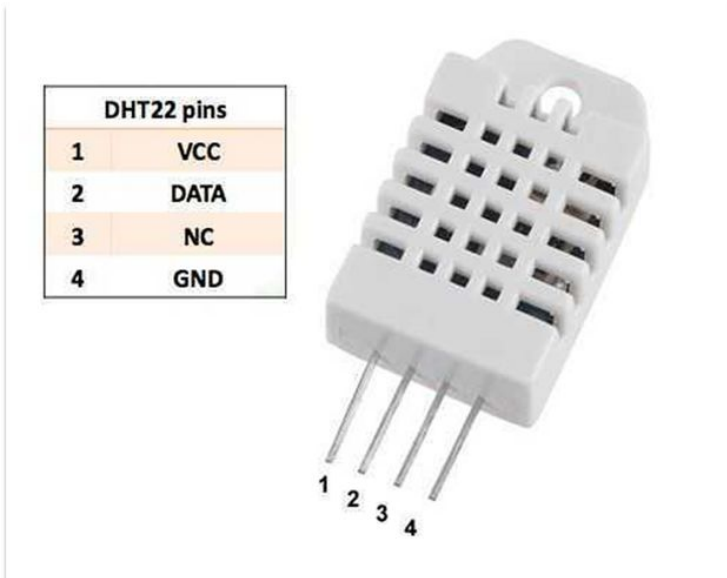
Step 1: WiFi ESP8266 NodeMcu ESP-12E

Here we detail the component we use, in this case the NodeMCU ESP12, along with the datasheet of that device.



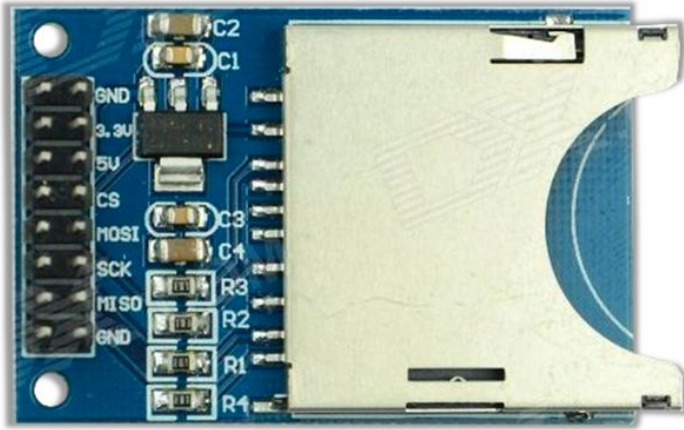
Step 2: Humidity Sensor

In the sequence, I show details about this other component, the DHT22, with the respective pinning.



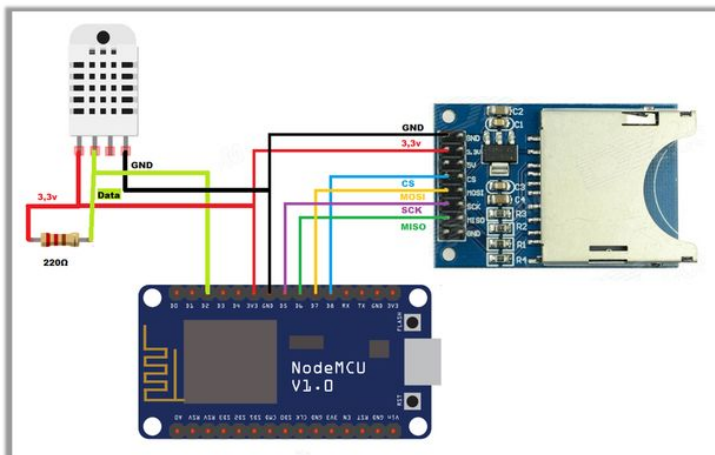
Step 3: SD Card Module

This is our SD Card module. As you can see from the pinout, it is with SPI connection.



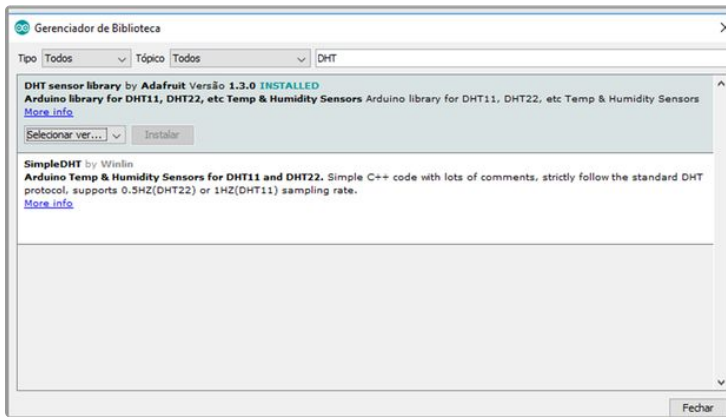
Step 4: Assembly

The assembly diagram relies on the reader, the DHT22, the NodeMCU ESP12. I chose the latter because it needs a reasonable amount of IOs. Thus, the ESP01 would also work for this assembly.



Step 5: Libraries

For this assembly, you need the DHT library of the Arduino IDE itself. Just go to "Sketch> Include Library> Manage Libraries" as you download the DHT. You have to do the same thing for the SD Library.



Step 6: Source Code

The source code used in the assembly is simple, and it is just to show that the SD Card running. You have to insert all the sophistication later, but you can use other innumerable features. However, this doesn't apply to this example.

```
//biblioteca responsável pela comunicação com o Cartão SD
#include <SD.h>
//biblioteca responsável pela comunicação com o sensor DHT22
#include <DHT.h>

// pino de dados do DHT será ligado no D6 do esp
#define DHTPIN D2
// tipo do sensor
#define DHTTYPE DHT22

// construtor do objeto para comunicar com o sensor
DHT dht(DHTPIN, DHTTYPE);

//pino ligado ao CS do módulo SD Card
#define CS_PIN D8;
```

Setup

In the Setup function, we will start our object's communication with the sensor, and also initialize the SD Card.

```

void setup()
{
  Serial.begin(9600);
  Serial.print("Inicializando o cartão SD...");

  //inicializa o objeto para comunicarmos com o sensor DHT
  dht.begin();

  // verifica se o cartão SD está presente e se pode ser inicializado
  if (!SD.begin(CS_PIN)) {
    Serial.println("Falha, verifique se o cartão está presente.");
    //programa encerrado
    return;
  }

  //se chegou aqui é porque o cartão foi inicializado corretamente
  Serial.println("Cartão inicializado.");
}

```

Loop

In the loop, we read moisture, humidity, and temperature. This is very much like the standard C language.

```

//faz a leitura da umidade
float umidade = dht.readHumidity();
Serial.print("Umidade: ");
Serial.println(umidade);
//faz a leitura da temperatura
float temperatura = dht.readTemperature();
Serial.print("Temperatura: ");
Serial.println(temperatura);

File dataFile = SD.open("LOG.txt", FILE_WRITE);
// se o arquivo foi aberto corretamente, escreve os dados nele
if (dataFile) {
  Serial.println("O arquivo foi aberto com sucesso.");
  //formatação no arquivo: linha a linha >> UMIDADE | TEMPERATURA
  dataFile.print(umidade);
  dataFile.print(" | ");
  dataFile.println(temperatura);

  //fecha o arquivo após usá-lo
  dataFile.close();
}
// se o arquivo não pôde ser aberto os dados não serão gravados.
else {
  Serial.println("Falha ao abrir o arquivo LOG.txt");
}

//intervalo de espera para uma nova leitura dos dados.
delay(2000);
}

```