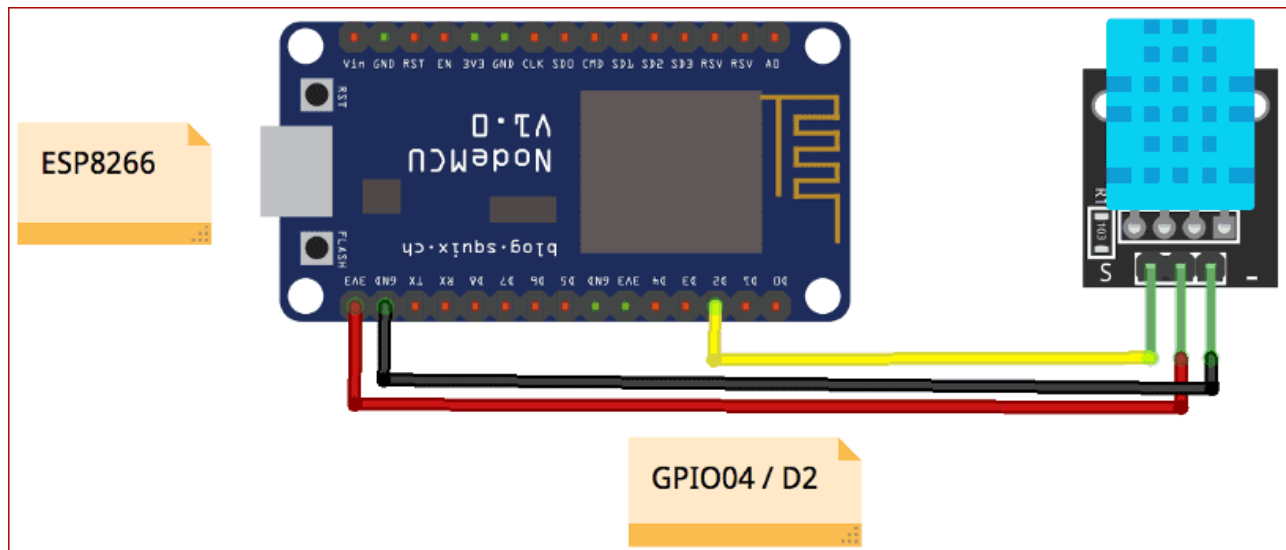


NodeMCU: modificación para enviar datos de un sensor DHT11 (V)

agrportfolioeducativo.blogspot.com/2019/07/nodemcu-modificacion-para-enviar-datos.html



1.- Incorporo biblioteca para DHT11

```
#include "DHTesp.h" // Incluyo la librería DHTesp
```

2.- Creo objeto de la clase DHTesp

```
DHTesp dht; // Objeto de la librería DHTesp, que controla el sensor dht11
```

3.- Inicializo el DataPin correspondiente.

Activo el pin D2, o el GPIO 4, dentro del SETUP.

```
dht.setup(4,DHTesp::DHT11); // data pin 4, que se corresponde con D2
```

4.- Modifico la función readsensor

```
void readSensor() {  
  delay(dht.getMinimumSamplingPeriod()); // Esta función es importante. La lectura del sensor  
  DHT11 (o DHT22) no es instantánea.  
  // La librería calcula el período de tiempo necesario entre lecturas y calculamos ese delay
```

```
  float humidity = dht.getHumidity(); // obtenemos la humedad  
  float temperature = dht.getTemperature(); //obtenemos la temperatura
```

```
  sendDataToGoogleSheets(temperature,humidity); // envío de datos
```

```
  /*  
  Serial.print(dht.getStatusString());  
  Serial.print("\t");  
  Serial.print(humidity, 1);  
  Serial.print("\t\t");
```

```

Serial.print(temperature, 1);
Serial.print("\t\t");
Serial.print(dht.toFahrenheit(temperature), 1);
Serial.print("\t\t");
Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1); // calcula la temperatura
aparente.
Serial.print("\t\t");
Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);
*/
}

```

5.- Modifico en el envío de datos las líneas... , dentro del LOOP

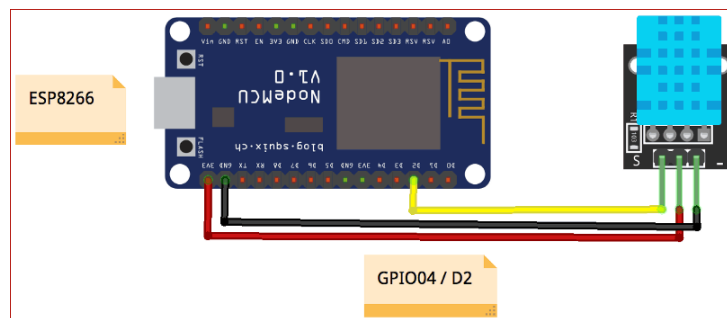
```

String stringTemp = String(temp, 1); //float y lugares decimales.
String stringHum = String(hum, 1);

```

== == == == == == == == ==

CONEXIONADO



Programa completo

```

/*
Envio de datos para Google Sheets
Hardware Utilizado: Nodemcu v1.0, DHT11
Autor: Yhan Christian Souza Silva - data: 27/07/2018
Referências: https://youtu.be/fSoGeaOkNRw
Modificación: Aurelio Gallardo. 06/07/2019
*/

// -- Bibliotecas auxiliares --

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include "DHTesp.h" // Incluyo la librería DHTesp

// -- Hardware --

// -- Variables y constantes --
constchar* ssid = "miSSID"; // Rellena con el nombre de tu red WiFi
constchar* password = "miCONTRASEÑA"; // Rellena con la contraseña de tu red WiFi

```

```

constchar* host ="script.google.com";// Este es el host de los scripts de google.
constint httpsPort =443;

// Huella digital del script de Google:
// D4:9E:40:F4:53:7A:04:93:38:F7:6B:4B:DC:70:02:A9:03:98:C2:DE
constchar* fingerprint ="D4 9E 40 F4 53 7A 04 93 38 F7 6B 4B DC 70 02 A9 03 98 C2 DE";

// const char* fingerprint = "46 B2 C3 44 9C 59 09 8B 01 B6 F8 BD 4C FB 00 74 91 2F EF F6";

String googleSheetsID ="AKfycbzxJZo6IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-
gUBY";// El que me da al implementar una aplicación web en el script.
//
https://script.google.com/macros/s/AKfycbzxJZo6IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-
gUBY/exec?TEMPERATURA=20&HUMEDAD=60

WiFiClientSecure cliente;// Objeto de la librería WiFiClientSecure
DHTesp dht;// Objeto de la librería DHTesp, que controla el sensor dht11

long previusMillis;
constlong interval =30000;//cada 30 segundos
int temperature, humidity;
float lastH, lastT;

// -- Setup --

void setup(){
  dht.setup(4,DHTesp::DHT11);// data pin 4, que se corresponde con GPIO2
  Serial.begin(115200);
  connectToWiFi();
}

// -- LOOP: lectura del sensor y envío de datos según el intervalo--

void loop(){
  if(millis()- previusMillis >= interval){
    readSensor();
    previusMillis = millis();
  }
}

// -- Funciones auxiliares --

// -- Conectando a la red Wifi. Muestra la IP recibida --

void connectToWiFi(){
  Serial.println("Conectando a rede: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while(WiFi.status()!= WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
}

```

```

}
WiFi.mode(WIFI_STA);
Serial.println("");
Serial.println("Conectado!");
Serial.print("IP: ");
Serial.println(WiFi.localIP());
delay(1000);
}

/* Lectura del sensor de temperatura y humedad
*/

void readSensor(){
  delay(dht.getMinimumSamplingPeriod()); // Esta función es importante. La lectura del sensor
  DHT11 (o DHT22) no es instantánea.
  // La librería calcula el período de tiempo necesario entre lecturas y calculamos ese delay

  float humidity = dht.getHumidity(); // obtenemos la humedad
  float temperature = dht.getTemperature(); // obtenemos la temperatura

  sendDataToGoogleSheets(temperature, humidity); // envío de datos

  /*
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.print(dht.toFahrenheit(temperature), 1);
  Serial.print("\t\t");
  Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1); // calcula la temperatura
  aparente.
  Serial.print("\t\t");
  Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);
  */
}

/* Función de conexión. Importante la instrucción cliente.setInsecure(); para conectar de forma
anónima
*/

void sendDataToGoogleSheets(float temp, float hum){

```

```

Serial.print("Conectando a: ");
Serial.println(host);

cliente.setInsecure();

if (!cliente.connect(host, httpsPort)) {
  Serial.println("Falla la conexión a Google Sheets --> " String(host) ": "
String(httpsPort));
return;
}

if(cliente.verify(fingerprint, host)){
  Serial.println("Certificado OK");
}
else{
  Serial.println("Comprobar certificado");
}

  String stringTemp = String(temp, 1); //float y lugares decimales.
  String stringHum = String(hum, 1);
  String url = "/macros/s/" googleSheetsID "/exec?TEMPERATURA=" stringTemp
"&HUMEDAD="

stringHum;

Serial.print("Petición URL ");
Serial.println(url);

cliente.print(String("GET ") url " HTTP/1.1\r\n"
  "Host: " host "\r\n"
  "User-Agent: BuildFailureDetectorESP8266\r\n"

"Connection: close\r\n\r\n");

Serial.println("Request enviada");
while(cliente.connected()){
String line = cliente.readStringUntil('\n');
if(line == "\r"){
  Serial.println("Cabeceras Recibidas");
  Serial.println(line);
break;
}
}

```

```
String line = cliente.readStringUntil('\n');
if(line.startsWith("{\"state\":\"success\"}")){
    Serial.println("Éxito");
}else{
    Serial.println("El envío falló!");
}

Serial.println("Respuesta:");
Serial.println("=====");
Serial.println(line);
Serial.println("=====");
Serial.println("¡Cerrando la conexión!");
}
```