



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL

INTERNET DAS COISAS – 2024.1 – PROF. WENDLEY S. SILVA

AULA PRÁTICA 02

THINGSPEAK: ARMAZENANDO NA INTERNET OS DADOS COLETADOS

Para mais informações sobre o NodeMCU ESP-12E e a instalação do módulo, reveja o material da aula prática 01.

INTRODUÇÃO

Este material orienta para a prática de como registrar na nuvem os dados coletados pelos sensores. Será utilizado o serviço do site ThingSpeak.com como provedor de serviços de nuvem e o sensor de temperatura e umidade DHT11/DHT22 será usado para medir dados de temperatura e umidade. Caso não tenha esse sensor, pode adaptar o código para outro tipo de sensor.

Dessa forma, com o programa aqui desenvolvido, você poderá monitorar, de qualquer lugar do mundo, a temperatura e umidade relativa do ar de qualquer local que você deseje, desde que tal local possua conectividade Wi-Fi à Internet.

INSTALANDO A BIBLIOTECA DO DHT11/DHT22

Inicialmente, vá no Gerenciador de Bibliotecas da Arduino IDE e instale os módulos para DHT e Adafruit Unified Sensor.

MONTAGEM DO CIRCUITO (SUGESTÃO)

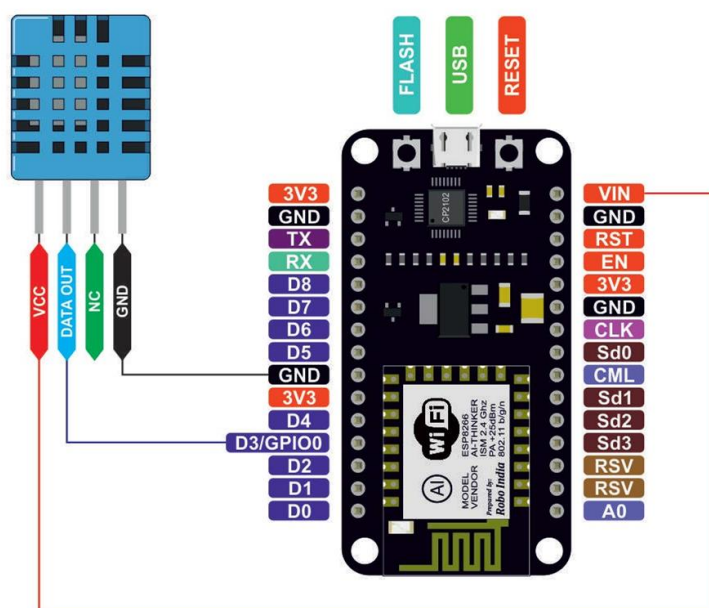


Figura 1 – Sugestão de montagem do circuito do hardware. Também pode ligar VCC do DHT em 3V3 do NodeMCU

ACESSO AO THINGSPEAK

O ThingSpeak é uma plataforma IoT que permite, sem custo algum, o upload de dados numéricos, os quais serão plotados ao longo do tempo na forma de gráficos.

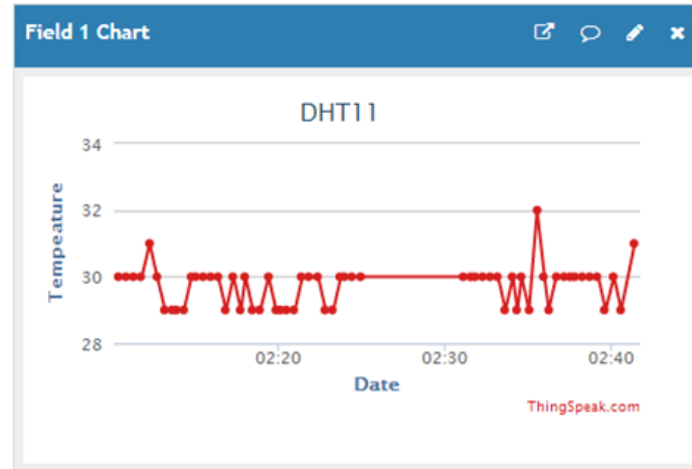


Figura 2 – Exemplo de um monitoramento salvo no ThingSpeak

Crie a sua conta (caso não possua) no site www.thingspeak.com. Crie um novo canal (Channel) e preencha com os dados relativos à prática. Vá para a aba **API Keys** e anote/copie as chaves, você precisará delas.

CÓDIGO INICIAL

Uma vez que o circuito está corretamente montado, é necessário programar o NodeMCU. A seguir está o código básico para executar este circuito conforme a descrição da prática, com algumas lacunas para vocês preencherem. No site da disciplina também tem o código em arquivo .ino para baixar.

```
#include <DHT.h> // Including library for dht
#include <ESP8266WiFi.h>

String apiKey = "Your API of thingspeak"; // 2 seu Write API key do site ThingSpeak

const char *ssid = "Your wifi Network name"; // 2 substitua com o ssid e senha da rede Wifi
const char *pass = "Network password";
const char* server = "api.thingspeak.com";

#define DHTPIN 0 // 2 pin onde o dht11 está conectado

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()
{
    Serial.begin(115200);
    delay(10);
    dht.begin();
}
```

```

Serial.println("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print("");
}
Serial.println("");
Serial.println("WiFi connected");
}

void loop()
{

    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t))
    {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com
    {

        String postStr = apiKey;
        postStr += "&field1="; //⚠️ atenção, esse é o campo 1 que você escolheu no canal do ThingSpeak
        postStr += String(t);
        postStr += "&field2=";
        postStr += String(h);
        postStr += "\r\n\r\n";

        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(postStr.length());
        client.print("\n\n");
        client.print(postStr);

        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.print(" degrees Celcius, Humidity: ");
        Serial.print(h);
        Serial.println("% Send to Thingspeak.");
    }
    client.stop();

    Serial.println("Waiting...");

    // thingspeak needs minimum 15 sec delay between updates, i've set it to 20 seconds
    delay(20000);
}

```

Nesse ponto, é importante lembrar que os pinos do NodeMCU e Arduino possuem o seguinte mapeamento:

D0 = 16;
D1 = 5;
D2 = 4;
D3 = 0;
D4 = 2;
D5 = 14;

D6 = 12;
D7 = 13;
D8 = 15;
D9 = 3;
D10 = 1;

EXECUÇÃO E PÓS-PROCESSAMENTO

Execute e acompanhe a saída no monitor serial e no site do seu canal no ThingSpeak. Os gráficos online são atualizados com poucos segundos (~3 seg.) de atraso depois da leitura realizada pelo sensor. Deixe o seu projeto em execução por aprox. 4 min. Em seguida, utilize o recurso do site chamado Matlab Analysis e calcule a média dos valores de temperatura e da umidade. Também exporte os dados em formato .CSV para o seu computador.

ATIVIDADE FINAL

Por fim, adicione um botão, ou chave seletora, na montagem do seu hardware, e exiba os seus valores (pressionado/solto) no seu canal do ThingSpeak.

O QUE ENVIAR PELO SIGAA?

Enviar um arquivo .ZIP, contendo:

- Código-fonte (geralmente um ou vários arquivos .ino)
- ~~Link para um vídeo de até 3 minutos mostrando o funcionamento da prática (hospedar em um serviço da nuvem, enviar apenas o link)~~
- Relatório em PDF de até 3 pág. contendo os nomes dos membros da equipe, o que cada um fez, como a prática foi desenvolvida (equipamentos, versão do software, sistema operacional utilizado), resultados gerais, observações e link para o vídeo.

LITERATURA CONSULTADA

NodeMCU Temperature, Humidity data upload on Thingspeak on Arduino IDE | Robo Indi. Disponível em: <<https://roboindia.com/tutorials/nodeMCU-dht11-thingspeak-data-upload>>. Acesso em: 02 mai. 2024.

THOMSEN, Adilson. **Como programar o módulo ESP8266 NodeMCU.** Disponível em: <<https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>>. Acesso em: 18 abr. 2024.