

Configurando display

Transcrição

Conhecendo melhor o sensor e o display, vamos reuni-los no nosso código, deixando-o mais próximo do código final.

Para o organizar melhor o nosso código, vamos criar uma nova função, chamada `medirTemperaturaUmidade()` e mover **todo** o código da função `loop()` para dentro dela:

```
#include <DHT.h>

#define DHTPIN D3
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

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

void loop() {

}

void medirTemperaturaUmidade() {
  int umidade = dht.readHumidity();
  int temperatura = dht.readTemperature(false);
  Serial.print("Temperatura: ");
  Serial.print(temperatura);
  Serial.print("°C");
  Serial.print(" ");
  Serial.print("Umidade: ");
  Serial.println(umidade);
  delay(5000);
}
```

E para o display ter acesso aos valores das variáveis `temperatura` e `umidade`, vamos torná-las globais:

```
#include <DHT.h>

#define DHTPIN D3
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
int umidade;
int temperatura;

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

void loop() {
```

```

}

void medirTemperaturaUmidade() {
  umidade = dht.readHumidity();
  temperatura = dht.readTemperature(false);
  Serial.print("Temperatura: ");
  Serial.print(temperatura);
  Serial.print("°C");
  Serial.print(" ");
  Serial.print("Umidade: ");
  Serial.println(umidade);
  delay(5000);
}

```

Agora, importamos a biblioteca **Adafruit_SSD1306.h**, que instalamos no último vídeo:

```

// --- DHT ---
#include <DHT.h>
#define DHTPIN D3
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
int umidade;
int temperatura;

// --- DISPLAY ---
#include <Adafruit_SSD1306.h>

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

void loop() {

}

void medirTemperaturaUmidade() {
  // código da função omitido
}

```

Agora, para criar o objeto do display, no momento da criação, devemos passar um pino de *reset*, no nosso caso vamos utilizar o pino do LED interno do NodeMCU, o **LED_BUILTIN** (**EXPLICAR O PORQUÊ DISSO**):

```

// --- DHT ---
#include <DHT.h>
#define DHTPIN D3
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
int umidade;
int temperatura;

// --- DISPLAY ---
#include <Adafruit_SSD1306.h>
#define OLED_RESET LED_BUILTIN

```

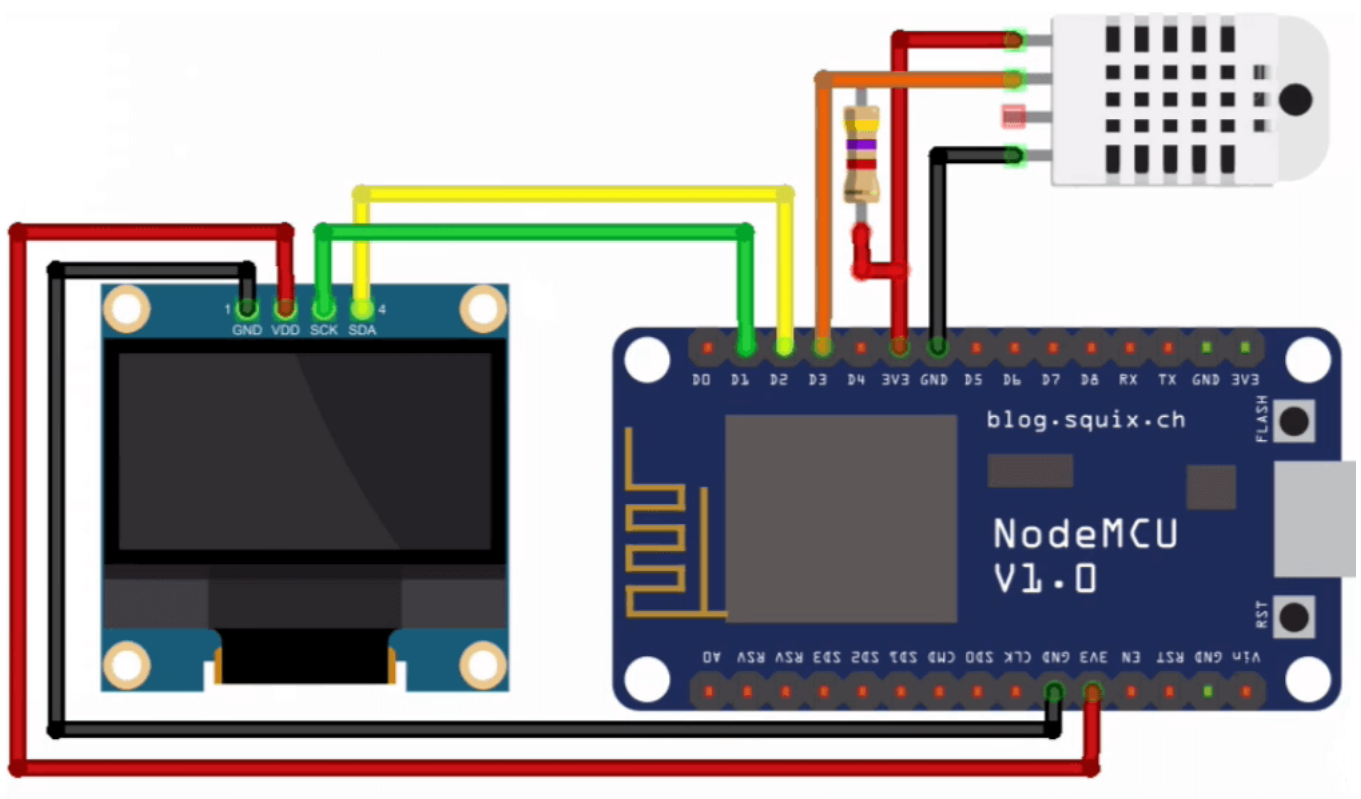
```
Adafruit_SSD1306 display(OLED_RESET);
```

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

```
void loop() {  
  
}
```

```
void medirTemperaturaUmidade() {  
  // código da função omitido  
}
```

O objeto recebe mais dois pinos, mas não passamo-os para ele, pois vamos utilizar os padrões da biblioteca, isto é, conectar o **SCL**, ou **SCK** (sigla de **Serial Clock**), do display ao pino **D1** do NodeMCU, e conectar o **SDA** (sigla de **Serial Data**) do display ao pino **D2** do NodeMCU, conforme a imagem abaixo:



Esses outros dois pinos são responsáveis pela comunicação com a placa, via protocolo I2C. É importante ter atenção à ordem das conexões, pois uma conexão errada pode causar a queima do seu display.

Configurando o display

Com o objeto criado, devemos configurá-lo, para começar a utilizá-lo. Para isso, vamos criar a função `configurarDisplay()`, e já chamá-la na função `setup()`:

```
/* Exibindo somente as funções  
setup(), loop() e configurarDisplay(),  
omitindo o restante do código */  
  
void setup() {
```

```
Serial.begin(115200);
configurarDisplay();
}

void loop() {

}

void configurarDisplay() {

}
```

Agora, precisamos inicializar o nosso display, através da função `begin`. Essa função recebe dois valores, o primeiro é o **(PESQUISAR, NÃO FAÇO IDEIA DO QUE É)**, e o segundo é endereço I2C da nossa placa. Para descobrir esse endereço, podemos executar um programa, que pode ser visto no [site do Arduino \(https://playground.arduino.cc/Main/I2cScanner\)](https://playground.arduino.cc/Main/I2cScanner), ou pode ser baixado [aqui \(https://s3.amazonaws.com/caelum-online-public/iot-nodemcu/files/i2c-scanner.ino\)](https://s3.amazonaws.com/caelum-online-public/iot-nodemcu/files/i2c-scanner.ino). Basta realizar a conexões do display no NodeMCU e executar esse programa na placa, que descobriremos seu endereço I2C, por exemplo o endereço `0x3C`:

```
/* Exibindo somente a função
configurarDisplay(), omitindo o
restante do código */

void configurarDisplay() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
}
```

Como o nosso display é monocromático, vamos deixar a cor do seu texto branca e limpá-lo:

```
/* Exibindo somente a função
configurarDisplay(), omitindo o
restante do código */

void configurarDisplay() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.setTextColor(WHITE);
  display.clearDisplay();
}
```

Com o display configurado, vamos ver como exibir as informações nele no próximo vídeo.

