

NodeMCU ESP8266 – Termômetro digital com DHT22



Por Fernando Koyanagi

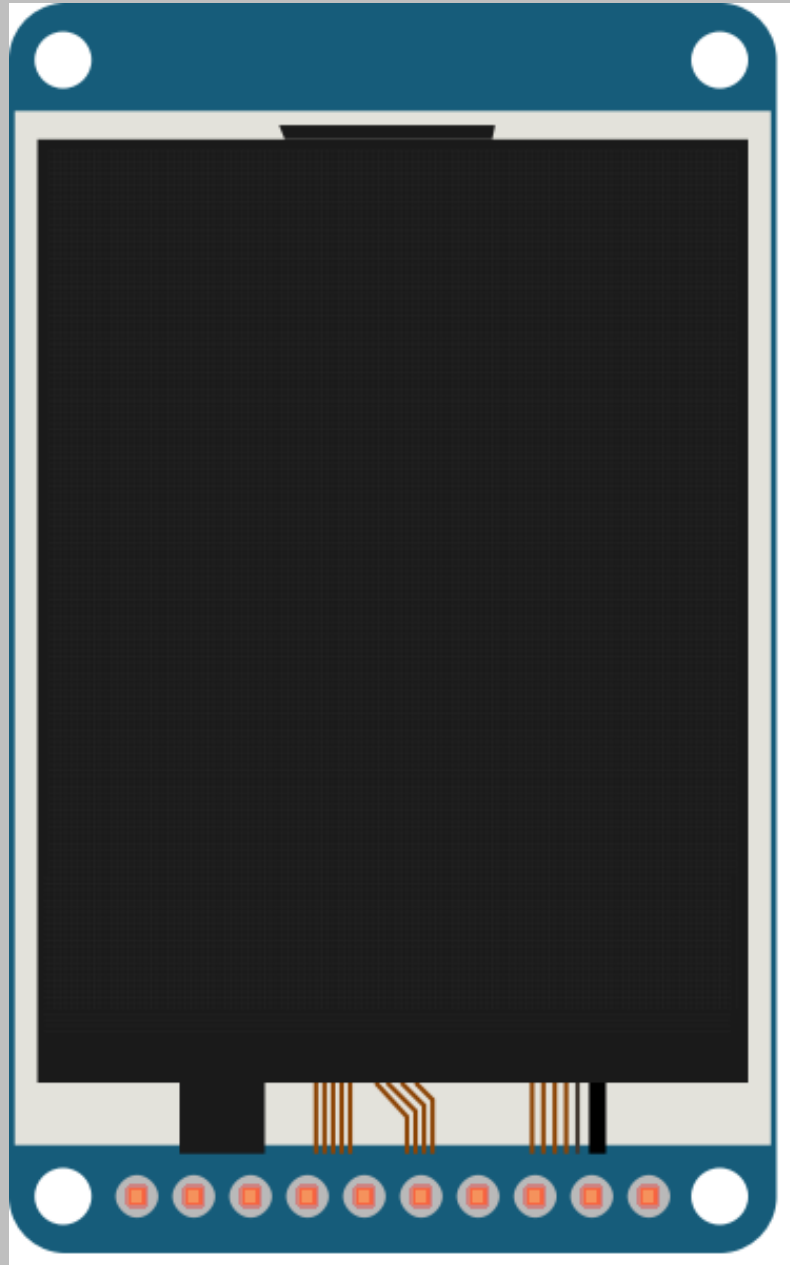
Objetivo

Faremos um termômetro digital utilizando um NodeMCU Esp8266, um módulo gráfico LCD e um sensor de umidade e temperatura AM2302 DHT22.

No módulo LCD mostraremos a temperatura e a umidade do ambiente em tempo real.



Módulo gráfico LCD 128x128 RGB TFT ILI 9163C

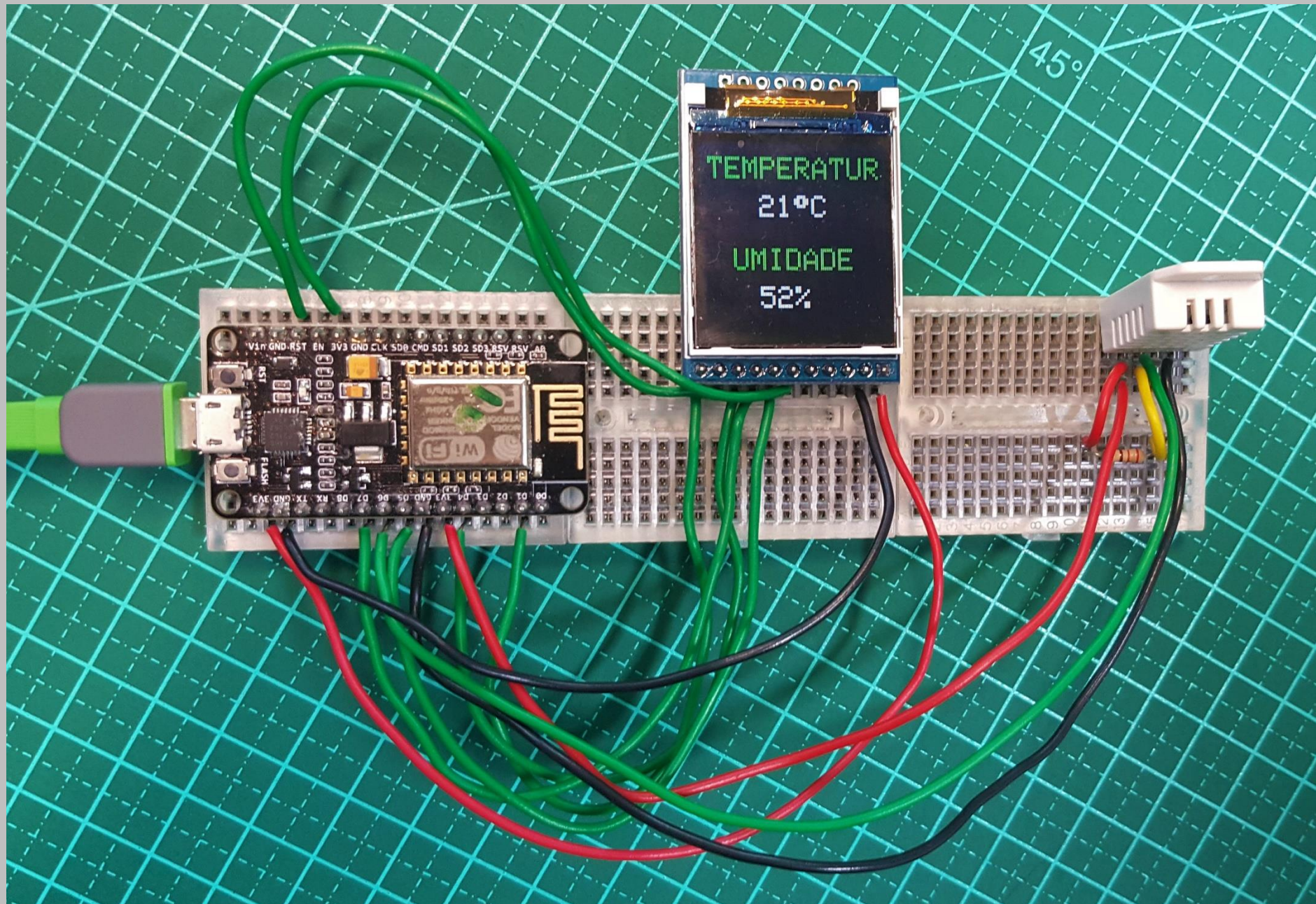


Sensor de Umidade e Temperatura AM2302 DHT22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Foto do projeto funcionando



EXEMPLO FUNCIONANDO

Em www.fernandok.com

Download arquivo **PDF** dos diagramas
Download arquivo **INO** do código fonte



Montagem

NodeMCU

LCD

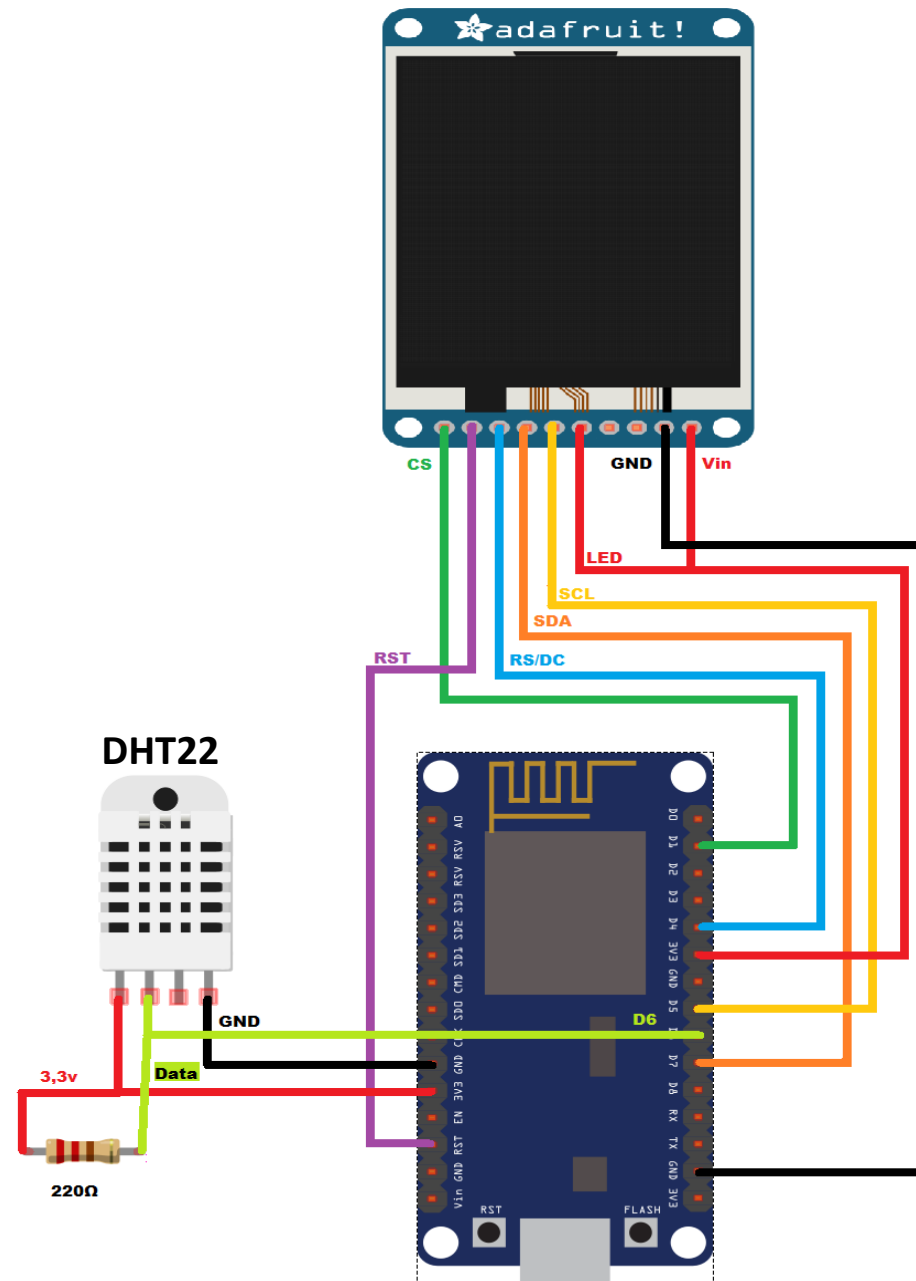
D5	SCL
D7	DAS
D1	CS
RST	RST
D4	RS
3,3v	LED
3,3v	Vin
GND	GND

NodeMCU

DHT22

D6	Data
GND	GND
3,3v	VCC

VCC → Resistor 220Ω → Data



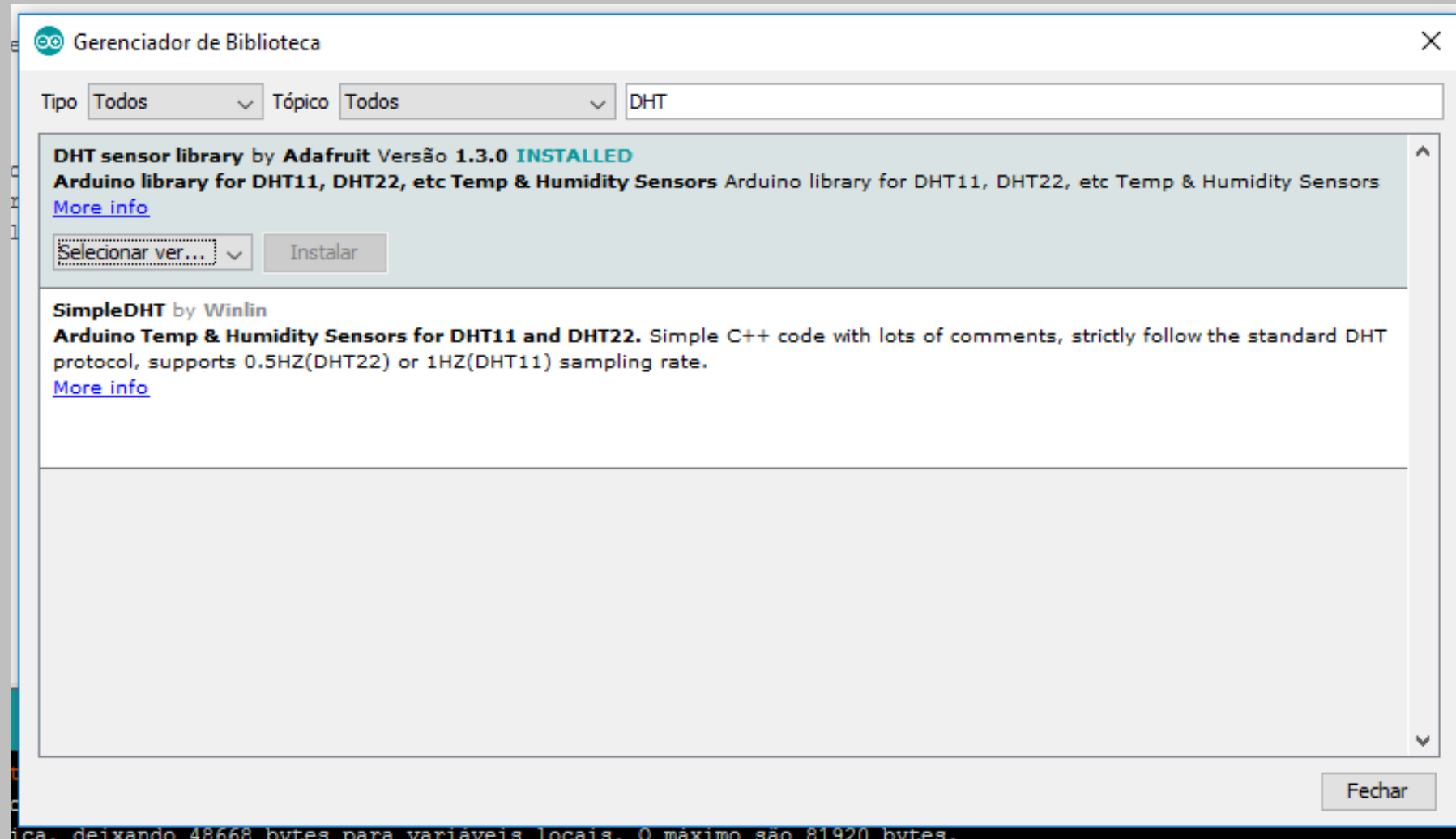
NodeMCU Esp8266



Bibliotecas

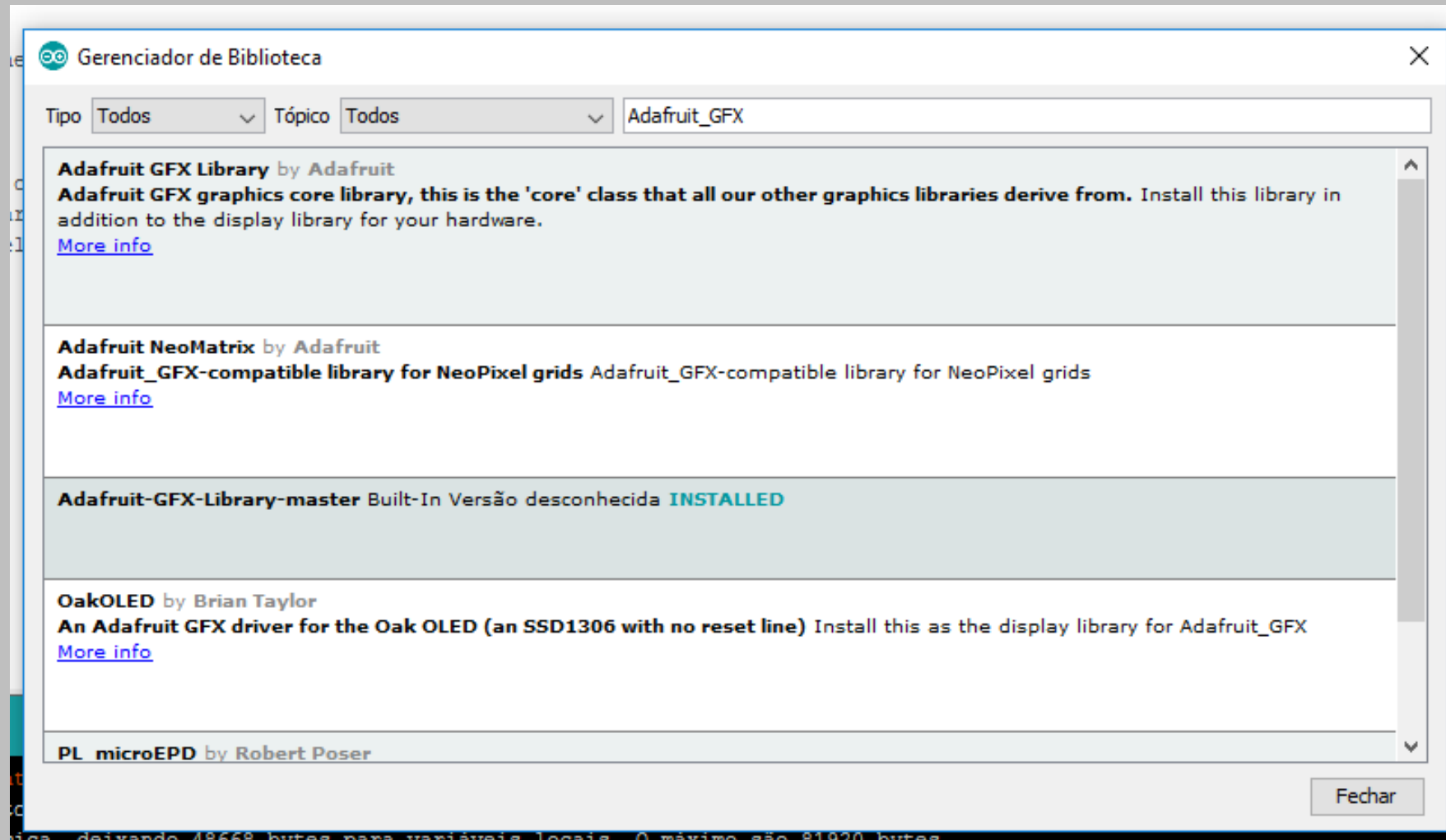
Adicione a seguinte biblioteca “DHT sensor library” para comunicação com o sensor de umidade e temperatura.

Basta acessar “Sketch >> Incluir Bibliotecas >> Gerenciar Bibliotecas...”



Bibliotecas

**Adicione a seguinte biblioteca “Adafruit-GFX-Library-master”.
Basta acessar “Sketch >> Incluir Bibliotecas >> Gerenciar Bibliotecas...”**



Bibliotecas

Adicione também a biblioteca “TFT_ILI9163C” para comunicação com o módulo gráfico LCD.

Acesse o [link](#) e faça download da biblioteca.

Descompacte o arquivo e cole na pasta de bibliotecas da IDE do arduino.

C:/Program Files (x86)/Arduino/libraries



Código

Primeiramente vamos adicionar as bibliotecas que serão utilizadas em nosso código.

```
#include <TFT_ILI9163C.h> //utilizada para se comunicar com o módulo LCD  
#include <DHT.h> //utilizada para se comunicar com o sensor de umidade e temperatura
```



Definições

Veremos abaixo as variáveis que utilizaremos durante o programa e a instância dos objetos.

```
#define DHTPIN D6 // pino que conectaremos o sensor DHT22
#define DHTTYPE DHT22 // DHT22 é o tipo do sensor que utilizaremos (importante para o construtor)

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

// definição das cores utilizadas (hexadecimal)
#define BLACK 0x0000
#define GREEN 0x07E0
#define WHITE 0xFFFF

#define __CS D1 // pino que conectaremos o CS do módulo LCD
#define __DC D4 // pino que conectaremos o RS do módulo LCD

TFT_ILI9163C display = TFT_ILI9163C(__CS, __DC); // construtor do objeto que utilizaremos para se
comunicar com o módulo LCD
```



Setup

Na função setup(), faremos a inicialização da variável “dht”, que é responsável pela comunicação com o sensor de umidade e temperatura, inicializaremos também a variável “display” que utilizaremos para se comunicar com o módulo LCD.

Também faremos a configuração do objeto para começar a desenhar em tela.

```
void setup()
{
    dht.begin(); // inicialização para se comunicar com o sensor

    display.begin(); // inicialização para se comunicar com o módulo LCD

    display.clearScreen(); // limpa a tela, removendo todos os desenhos
    display.fillScreen(BLACK); // pinta a tela toda de preto
    display.setTextSize(2); // configura o tamanho do texto com o tamanho 2
    display.setTextColor(GREEN); // configura a cor do texto como verde
    display.setCursor(5,10); // posiciona o cursor para começar a escrita a partir do (x,y)
    display.print("TEMPERATUR"); // escreve em tela
    display.setCursor(22,70); // reposiciona o cursor
    display.print("UMIDADE"); // escreve em tela
    display.setTextColor(WHITE); // configura a cor do texto como branco (a partir de agora)

    delay(1000); // espera de 1 segundo
}
```



loop

Na função loop(), vamos recuperar a umidade e temperatura lida pelo sensor e escrever na tela no local específico. A cada intervalo de 5 segundos o valor é lido do sensor e escrito em tela.

```
void loop()
{
    int h = dht.readHumidity(); // faz a leitura da umidade do sensor
    int t = dht.readTemperature(); // faz a leitura da temperatura do sensor

    //as 2 linhas seguintes utilizando o método "fillRect", são para fazer a limpeza do local onde
    //escreveremos a umidade e a temperatura, apagaremos o valor atual para escrever novamente atualizado.
    display.fillRect(5,32,120,20,BLACK); // fillRect(x,y, width, height, color);
    display.fillRect(5,92,120,20,BLACK);

    display.setCursor(40,35); // reposiciona o cursor para escrever
    display.print(t); // escreve a temperatura em tela
    display.print((char)247); // escreve o símbolo de grau ° através de código
    display.print("C"); // coloca o "C" para indicar que é graus Celcius

    display.setCursor(40,95); // reposiciona o cursor para escrever
    display.print(h); // escreve a umidade em tela
    display.print("%"); // escreve o símbolo de "porcentagem" para indicar a umidade

    delay(5000);
}
```



Algumas outras funções interessantes

//rotaciona o conteúdo da tela (parâmetros 0,1,2 ou 3)

display.setRotation(uint8_t);

//inverte as cores do display (faz um negativo)

display.invertDisplay(boolean);

//desenha um único pixel na tela na posição (x,y)

display.drawPixel(x, y, color);

//desenha uma linha vertical na posição

display.drawFastVLine(x, y, width, color);

//desenha uma linha vertical na posição especificada

display.drawFastHLine(x, y, width, color);

//desenha uma linha horizontal na posição especificada

display.drawRect(x, y, width, heigh, color);

//desenha um círculo na posição especificada

display.drawCircle(x, y, radius, color);

Em www.fernandok.com

Download arquivo **PDF** dos diagramas
Download arquivo **INO** do código fonte

