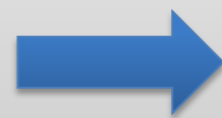


NodeMCU ESP8266 – Gráfico de temperatura e umidade com DHT22



ARDUINO
IDE



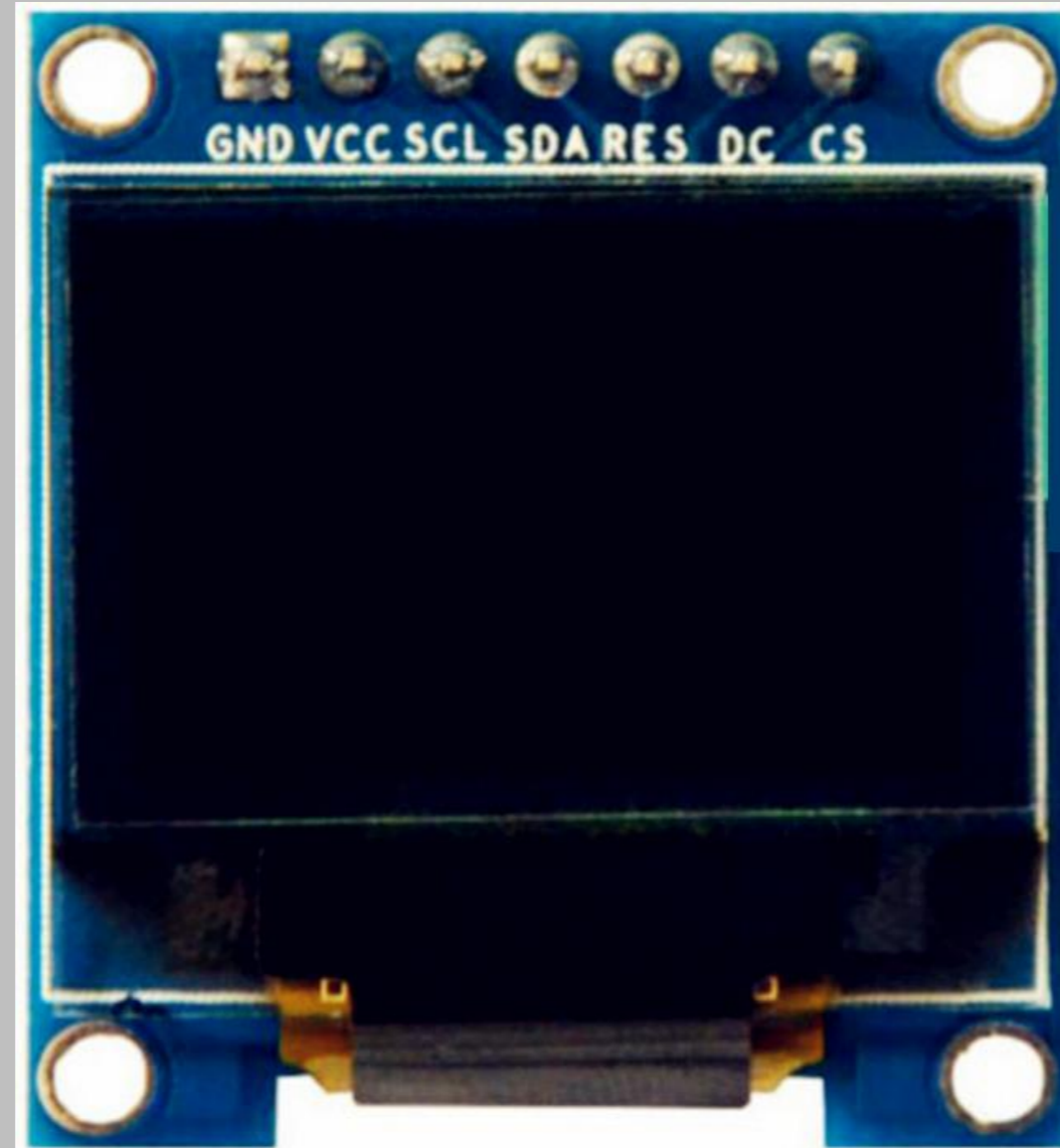
Por Fernando Koyanagi

Objetivo

Faremos um termômetro digital utilizando um NodeMCU Esp8266, um display OLED e um sensor de umidade e temperatura AM2302 DHT22. No display mostraremos a temperatura e a umidade lida pelo sensor e um gráfico em tempo real.



Display OLED 96"



Sensor de Umidade e Temperatura AM2302 DHT22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Vídeo do projeto funcionando



EXEMPLO FUNCIONANDO

Em www.fernandok.com

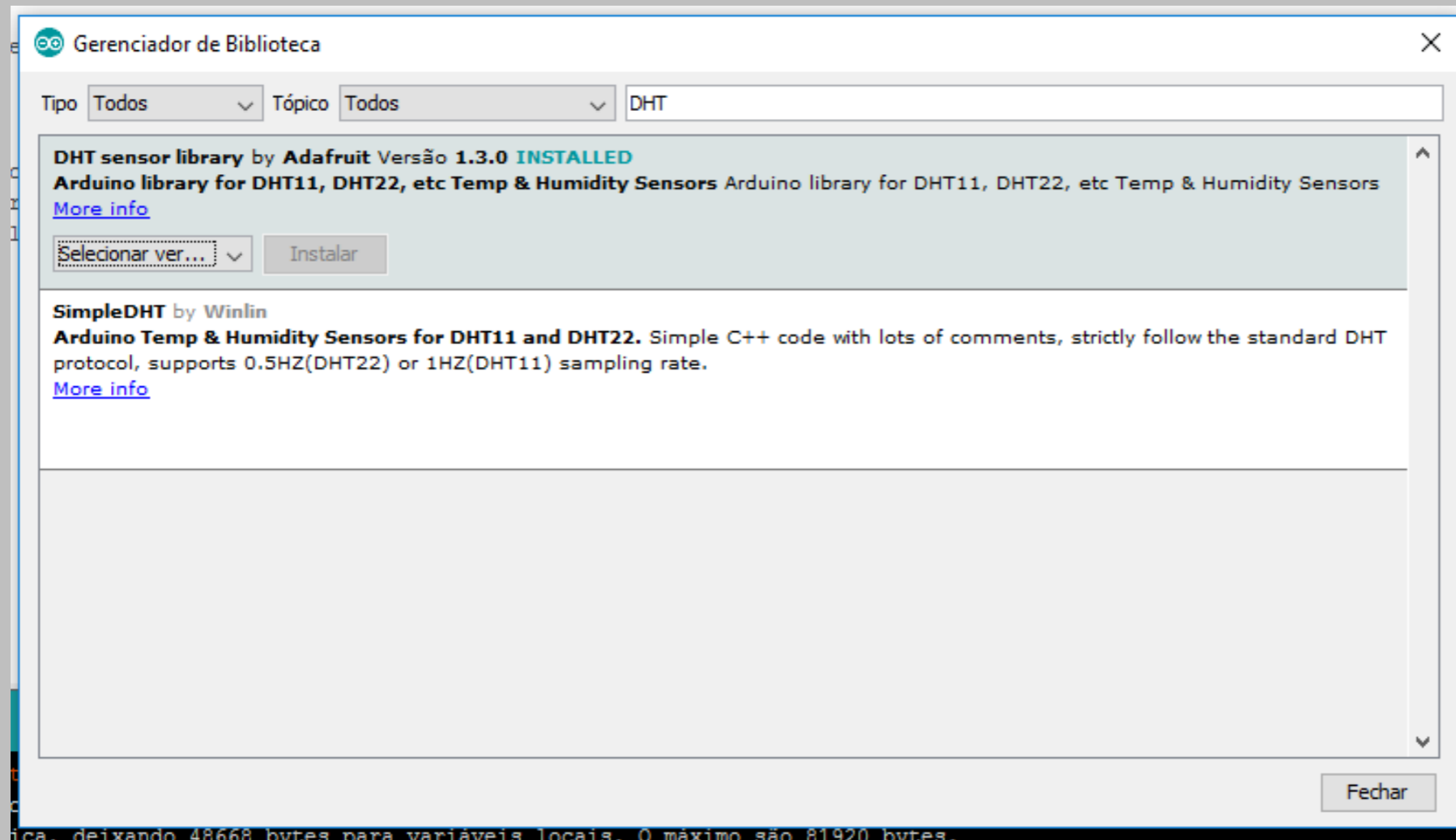
Download arquivo **PDF** dos diagramas
Download arquivo **INO** do código fonte



Bibliotecas

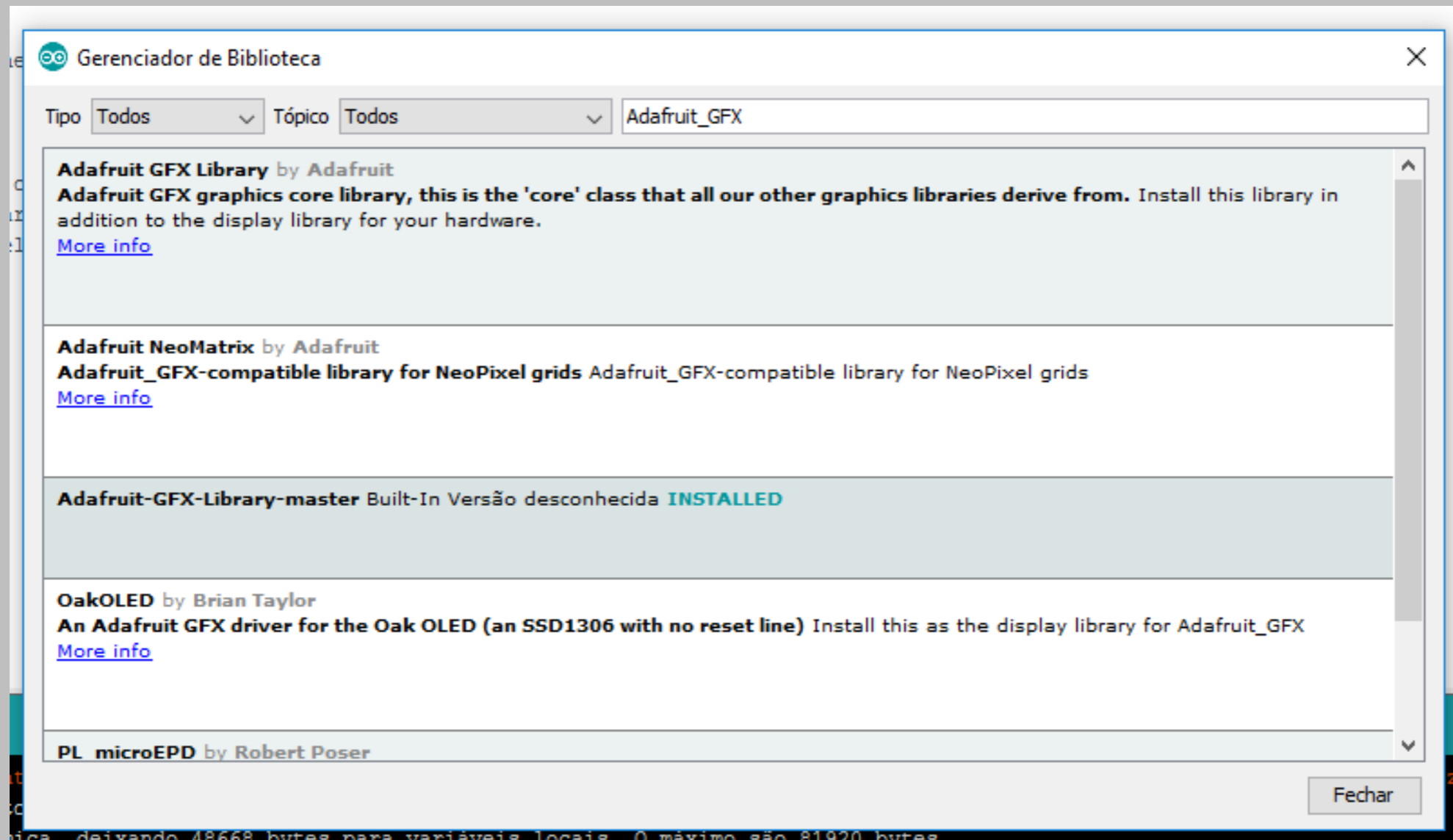
Adicione a seguinte biblioteca “DHT sensor library” para comunicação com o sensor de umidade e temperatura.

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



Bibliotecas

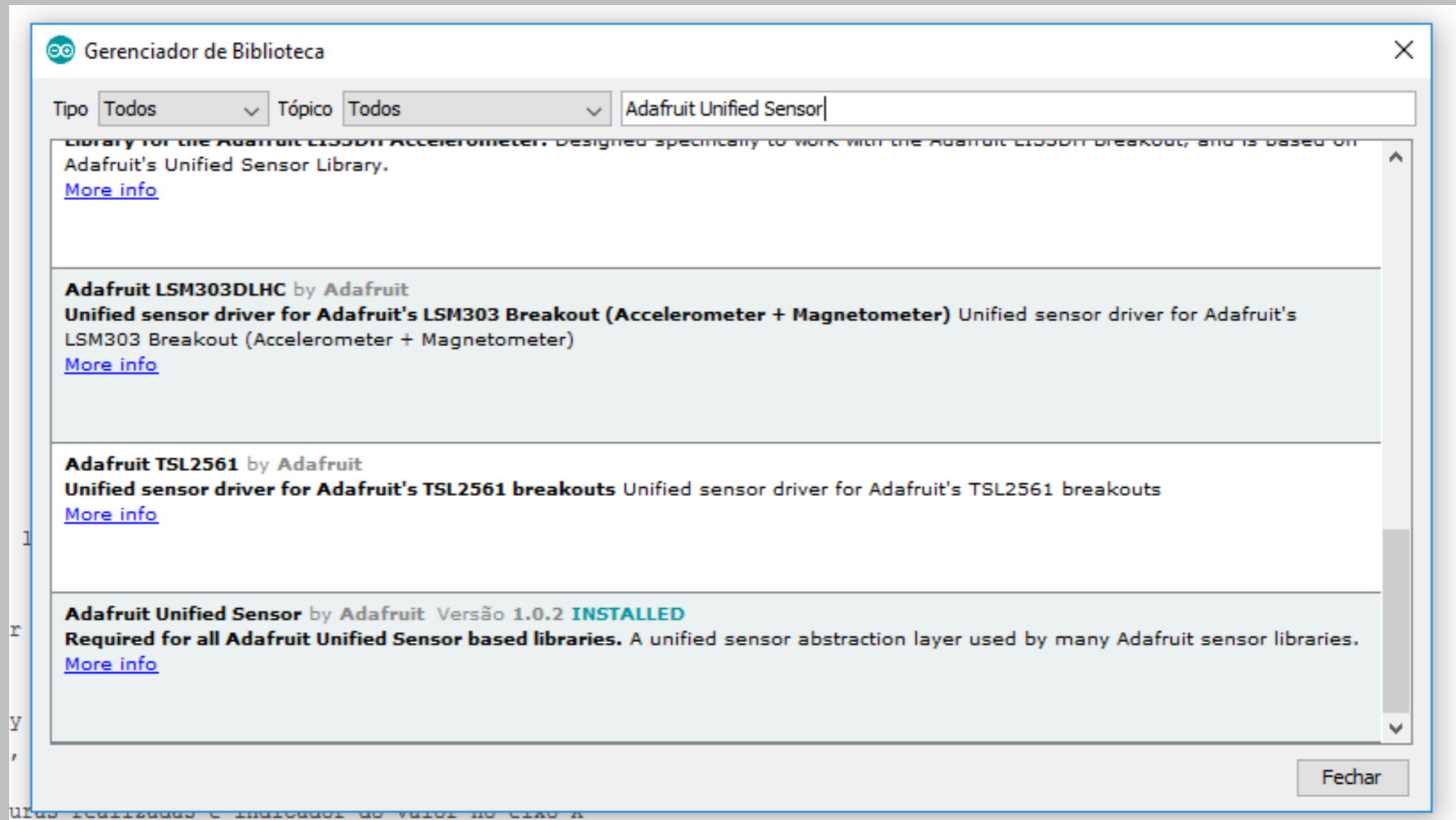
Adicione a seguinte biblioteca “Adafruit-GFX-Library-master”.
Basta acessar “Sketch >> Incluir Bibliotecas >> Gerenciar Bibliotecas...”



Bibliotecas

Adicione a seguinte biblioteca “Adafruit Unified Sensor”.

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



Bibliotecas

Adicione também a biblioteca “Adafruit-SSD1331-OLED” para comunicação com o display oled.

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 <Adafruit_GFX.h>
#include <Adafruit_SSD1331.h> // comunicação com o display oled
#include <DHT.h> // comunicação com o sensor de umidade e temperatura
```



Definições

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

```
// pinagem para o NodeMCU ESP8266
#define sclk D5
#define mosi D7
#define cs D8
#define rst D3
#define dc D1

// definição das cores que serão utilizadas
#define BLACK 0x0000
#define RED 0xF800
#define CYAN 0x07FF
#define YELLOW 0xFFE0
#define WHITE 0xFFFF
```



Definições

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

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

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

// construtor do objeto para comunicar com o display OLED
Adafruit_SSD1331 display = Adafruit_SSD1331(cs, dc, mosi, sclk, rst);

//variáveis que armazenarão os valores lidos da umidade e temperatura
int umidade = 0;
int temperatura = 0;

//variável que armazenará o valor da coordenada Y para desenharmos uma linha de exemplo
//que varia os valores de 1 em 1
int linhaExemplo = 20;
int fator = 1; //indicará se somaremos ou subtrairemos uma unidade na variável linhaExemplo
```



Definições

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

```
//variável responsável por contar o número de leituras realizadas e indicador do valor no eixo X
int leituraAtual = 1;

//definições do posicionamento dos eixos X e Y
#define POS_X_GRAFICO      5
#define POS_Y_GRAFICO     1
#define ALTURA_GRAFICO   50
#define COMPRIMENTO_GRAFICO 90

//definição da coordenada onde escreveremos os dados de temperatura e umidade
#define POS_X_DADOS      10
#define POS_Y_DADOS      54
```



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 display oled.

```
void setup()  
{  
  //inicializa o objeto para comunicarmos com o sensor DHT  
  dht.begin();  
  
  //inicializa o objeto para comunicarmos com o displa OLED  
  display.begin();  
  
  ...  
}
```



Setup

Vamos a seguir configurar algumas características do display e desenhar os eixos X e Y do gráfico.

```
void setup()
{
    ...

    //pinta a tela toda de preto
    display.fillScreen(BLACK);

    //os comandos a seguir irão desenhar as linhas dos eixos cartesianos na cor branca
    //drawFastVLine(x,y,width,color) --> linha vertical
    display.drawFastVLine(POS_X_GRAFICO, POS_Y_GRAFICO, ALTURA_GRAFICO, WHITE); //eixo Y
    //drawFastHLine(x,y,width,color) --> linha horizontal
    display.drawFastHLine(POS_X_GRAFICO, ALTURA_GRAFICO+1, COMPRIMENTO_GRAFICO, WHITE); //eixo X

    //desenha 2 pixels na tela bem no início do eixo Y para formar uma "seta"
    display.drawPixel(4,2,WHITE);
    display.drawPixel(6,2,WHITE);

    //desenha 2 pixels na tela bem no final do eixo X para formar uma "seta"
    display.drawPixel(POS_X_GRAFICO + COMPRIMENTO_GRAFICO-2, POS_Y_GRAFICO + ALTURA_GRAFICO-1 ,WHITE);
    display.drawPixel(POS_X_GRAFICO + COMPRIMENTO_GRAFICO-2, POS_Y_GRAFICO + ALTURA_GRAFICO+1, WHITE);
    ...
}
```

Setup

Vamos a seguir desenhar em tela no lugar específico onde indicaremos o valor da Temperatura (T) e Umidade (U) em tempo real.

```
void setup()
{
    ...

    //configura o tamanho do texto que escreveremos em tela
    display.setTextSize(1);
    //configura a cor branca para o texto
    display.setTextColor(WHITE);

    //posiciona o cursor para escrita
    display.setCursor(POS_X_DADOS, POS_Y_DADOS);
    display.print("T: "); //indicando a temperatura
    display.setCursor(POS_X_DADOS+35, POS_Y_DADOS);
    display.print(" U: "); //indicando a umidade
}
```



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()
{
    //faz a leitura da umidade
    umidade = dht.readHumidity();
    //faz a leitura da temperatura
    temperatura = dht.readTemperature();

    //limpa a área onde colocamos o valor da temperatura e da umidade
    display.fillRect(POS_X_DADOS+15, POS_Y_DADOS, 20, 10, BLACK);
    display.fillRect(POS_X_DADOS+55, POS_Y_DADOS, 30, 10, BLACK);

    //reposiciona o cursor para escrever a temperatura
    display.setCursor(POS_X_DADOS+15, POS_Y_DADOS);
    display.setTextColor(RED);
    display.print(temperatura);
    display.print((char)247); //escreve o símbolo de grau em tela

    //reposiciona o cursor para escrever a umidade
    display.setCursor(POS_X_DADOS+55, POS_Y_DADOS);
    display.setTextColor(CYAN);
    display.print(umidade);
    display.print("%");
    ...
}
```

loop

Vejam os exemplos abaixo de como colocar os pontos no gráfico de acordo com a leitura.

```
...
//mapeando o valor das variáveis para colocar no gráfico
//necessário pois, o display tem 64px de altura e separamos apenas 50 para o gráfico
//umidade pode ser lida de 0-100
int temperaturaMapeada = map(temperatura,0,100,0,50);
int umidadeMapeada = map(umidade,0,100,0,50);

//desenha na tela o ponto referente aos valores lidos do sensor
display.drawPixel(POS_X_GRAFICO+leituraAtual, ALTURA_GRAFICO-temperaturaMapeada, RED);
display.drawPixel(POS_X_GRAFICO+leituraAtual, ALTURA_GRAFICO-umidadeMapeada, CYAN);

//desenha na tela o ponto referente a nossa linha de exemplo que fica variando
display.drawPixel(POS_X_GRAFICO+leituraAtual, ALTURA_GRAFICO-linhaExemplo, YELLOW);

//aqui controlamos nossa linha de exemplo, quando chega no valor máximo decrementamos o valor
//até um valor mínimo determinado (no nosso caso 10), e a partir daí, incrementa novamente
if(linhaExemplo == 50) fator = -1; //decrementa
else if(linhaExemplo == 10) fator = 1; //incrementa

//soma o valor do fator na linhaExemplo
linhaExemplo += fator;

//incrementa o contador de leituras realizadas
leituraAtual++;
...
```



loop

Por último vamos escrever nossa lógica para limpar a tela do gráfico assim que atingir seu limite

```
void loop()
{
    ...
    //se a leitura chegar em 90 (número máximo do eixo X) então limparemos a área do gráfico para voltarmos
a desenhar.
    if(leituraAtual == 90)
    {
        //limpa a área toda do gráfico
        display.fillRect(POS_X_GRAFICO+1, POS_Y_GRAFICO-1, COMPRIMENTO_GRAFICO, ALTURA_GRAFICO-1, BLACK);
        leituraAtual = 1; //volta o contador de leitura para 1 (nova coordenada X)

        //como limpamos a área do gráfico, temos que redesenhar os pontos da “seta” que ficam na área interna
do gráfico
        display.drawPixel(6,2,WHITE);
        display.drawPixel(POS_X_GRAFICO+COMPRIMENTO_GRAFICO-2,POS_Y_GRAFICO+ALTURA_GRAFICO-1,WHITE);
    }

    //intervalo de tempo para realizarmos nova leitura de dados
    delay(5000);
}
```



Em www.fernandok.com

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

