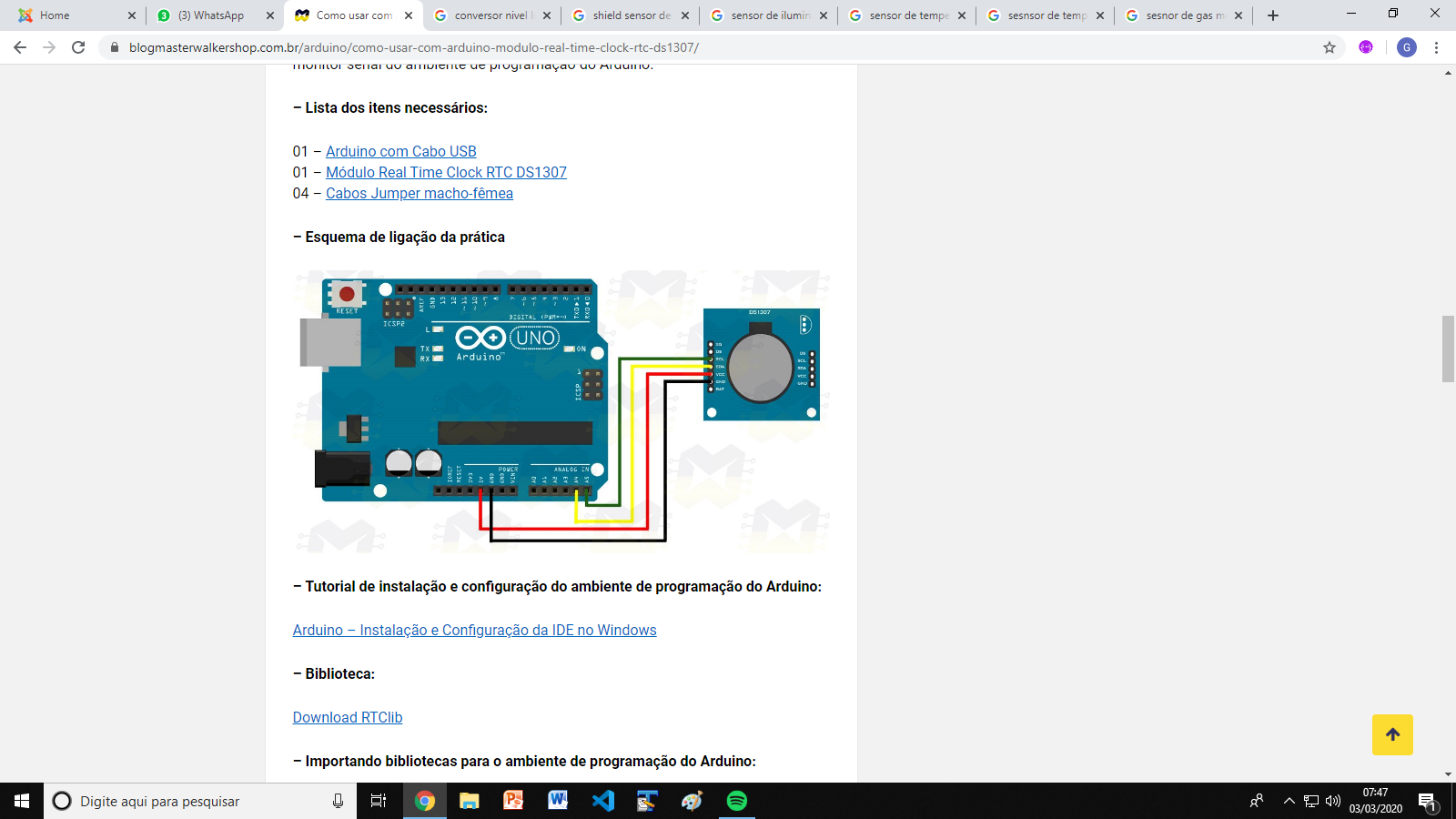
1. RELÓGIO



#include <Wire.h>

#include "RTClib.h"

RTC\_DS1307 rtc; //OBJETO DO TIPO RTC\_DS1307

//DECLARAÇÃO DOS DIAS DA SEMANA

char daysOfTheWeek[7][12] = {"Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado"};

void setup () {

  Serial.begin(9600); //INICIALIZA A SERIAL

  if (! rtc.begin()) // SE O RTC NÃO FOR INICIALIZADO

{

    Serial.println("DS1307 não encontrado");

    while(1); //SEMPRE ENTRE NO LOOP **?? QUE**

  }

  if (! rtc.isrunning()) { //SE RTC NÃO ESTIVER SENDO EXECUTADO

    Serial.println("DS1307 rodando!");

    rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_))); //CAPTURA A DATA E HORA EM QUE O SKETCH É COMPILADO

    //rtc.adjust(DateTime(2018, 7, 5, 15, 33, 15)); //(ANO), (MÊS), (DIA), (HORA), (MINUTOS), (SEGUNDOS)

  }

  delay(100);

}

void loop () {

    DateTime now = rtc.now(); //CHAMADA DE FUNÇÃO

    Serial.print("Data: ");

    Serial.print(now.day(), DEC);

    Serial.print('/');

    Serial.print(now.month(), DEC);

    Serial.print('/');

    Serial.print(now.year(), DEC);

    Serial.print(" / Dia: ");

    Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);

    Serial.print(" / Horas: ");

    Serial.print(now.hour(), DEC);

    Serial.print(':');

    Serial.print(now.minute(), DEC);

    Serial.print(':');

    Serial.print(now.second(), DEC);

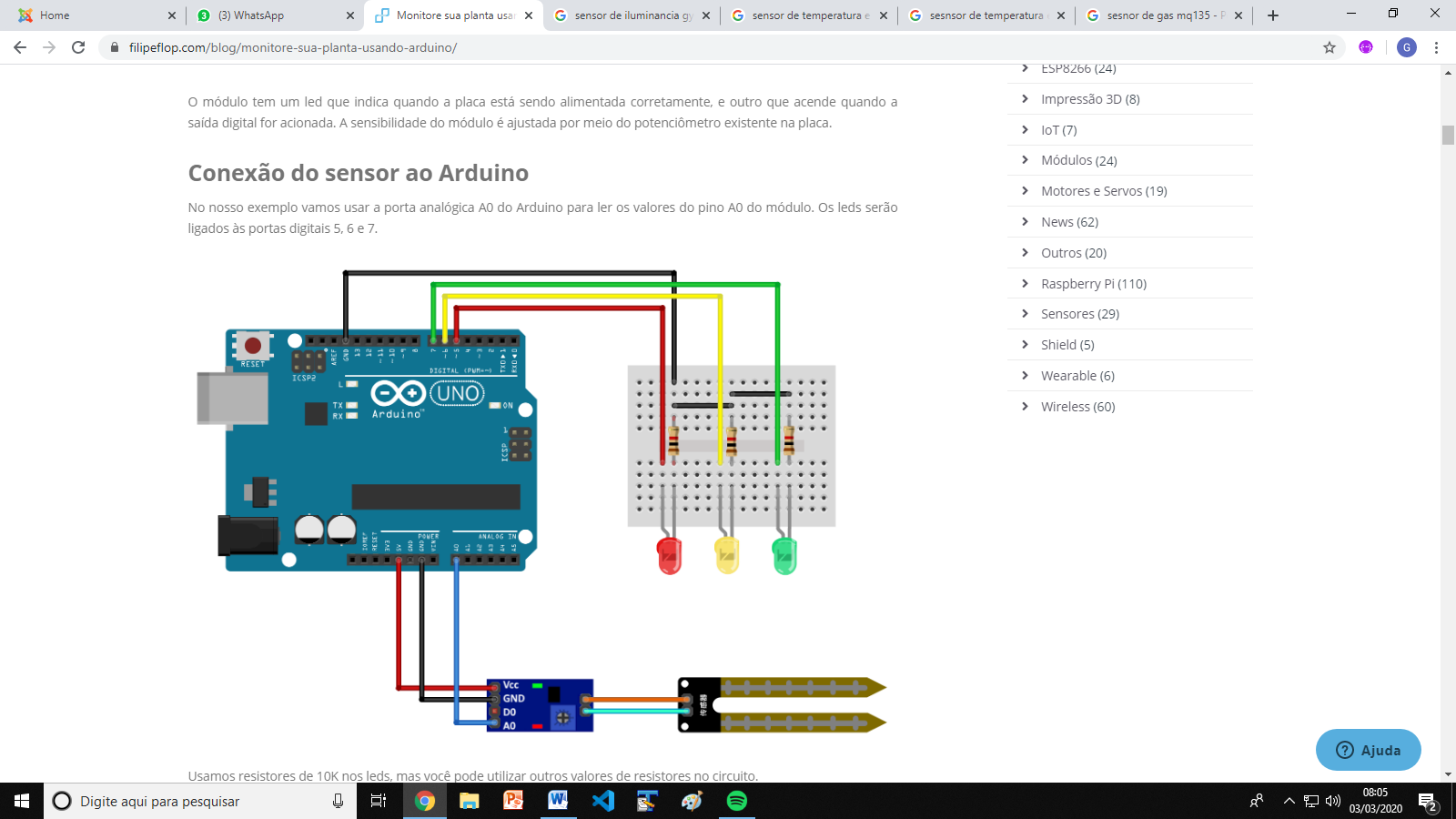
    Serial.println();

    delay(1000); //INTERVALO DE 1 SEGUNDO

}

<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-real-time-clock-rtc-ds1307/>

1. SENSOR DE UMIDADE DO SOLO



//Programa: Monitoracao de planta usando Arduino

//Autor: FILIPEFLOP

#define pino\_sinal\_analogico A0

#define pino\_led\_vermelho 5

#define pino\_led\_amarelo 6

#define pino\_led\_verde 7

int valor\_analogico;

void setup()

{

  Serial.begin(9600);

  pinMode(pino\_sinal\_analogico, INPUT);

  pinMode(pino\_led\_vermelho, OUTPUT);

  pinMode(pino\_led\_amarelo, OUTPUT);

  pinMode(pino\_led\_verde, OUTPUT);

}

void loop()

{

  //Le o valor do pino A0 do sensor

  valor\_analogico = analogRead(pino\_sinal\_analogico);

  //Mostra o valor da porta analogica no serial monitor

  Serial.print("Porta analogica: ");

  Serial.print(valor\_analogico);

  //Solo umido, acende o led verde

  if (valor\_analogico > 0 && valor\_analogico < 400)

  {

    Serial.println(" Status: Solo umido");

    apagaleds();

    digitalWrite(pino\_led\_verde, HIGH);

  }

  //Solo com umidade moderada, acende led amarelo

  if (valor\_analogico > 400 && valor\_analogico < 800)

  {

    Serial.println(" Status: Umidade moderada");

    apagaleds();

    digitalWrite(pino\_led\_amarelo, HIGH);

  }

  //Solo seco, acende led vermelho

  if (valor\_analogico > 800 && valor\_analogico < 1024)

  {

    Serial.println(" Status: Solo seco");

    apagaleds();

    digitalWrite(pino\_led\_vermelho, HIGH);

  }

  delay(100);

}

void apagaleds()

{

  digitalWrite(pino\_led\_vermelho, LOW);

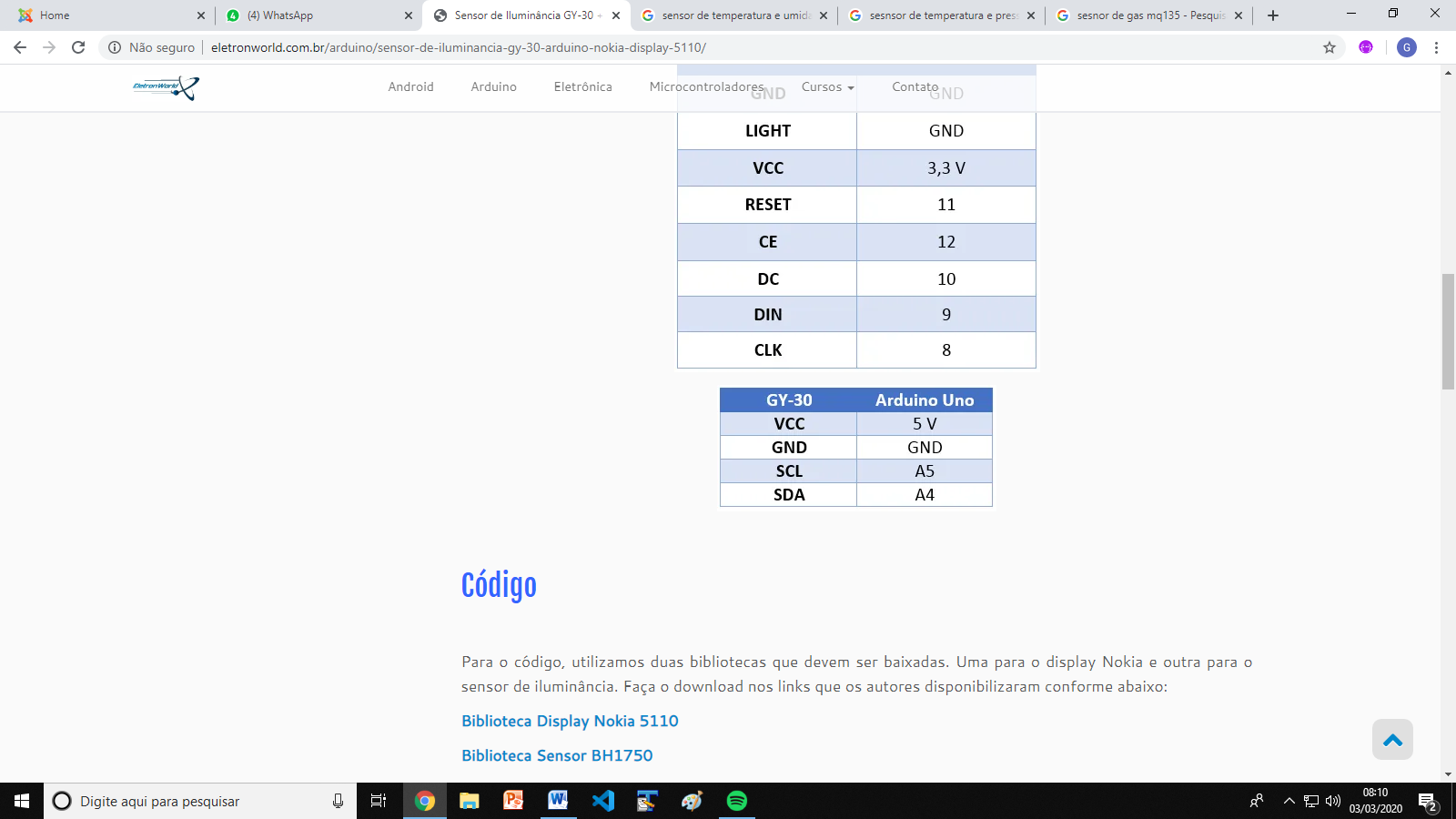
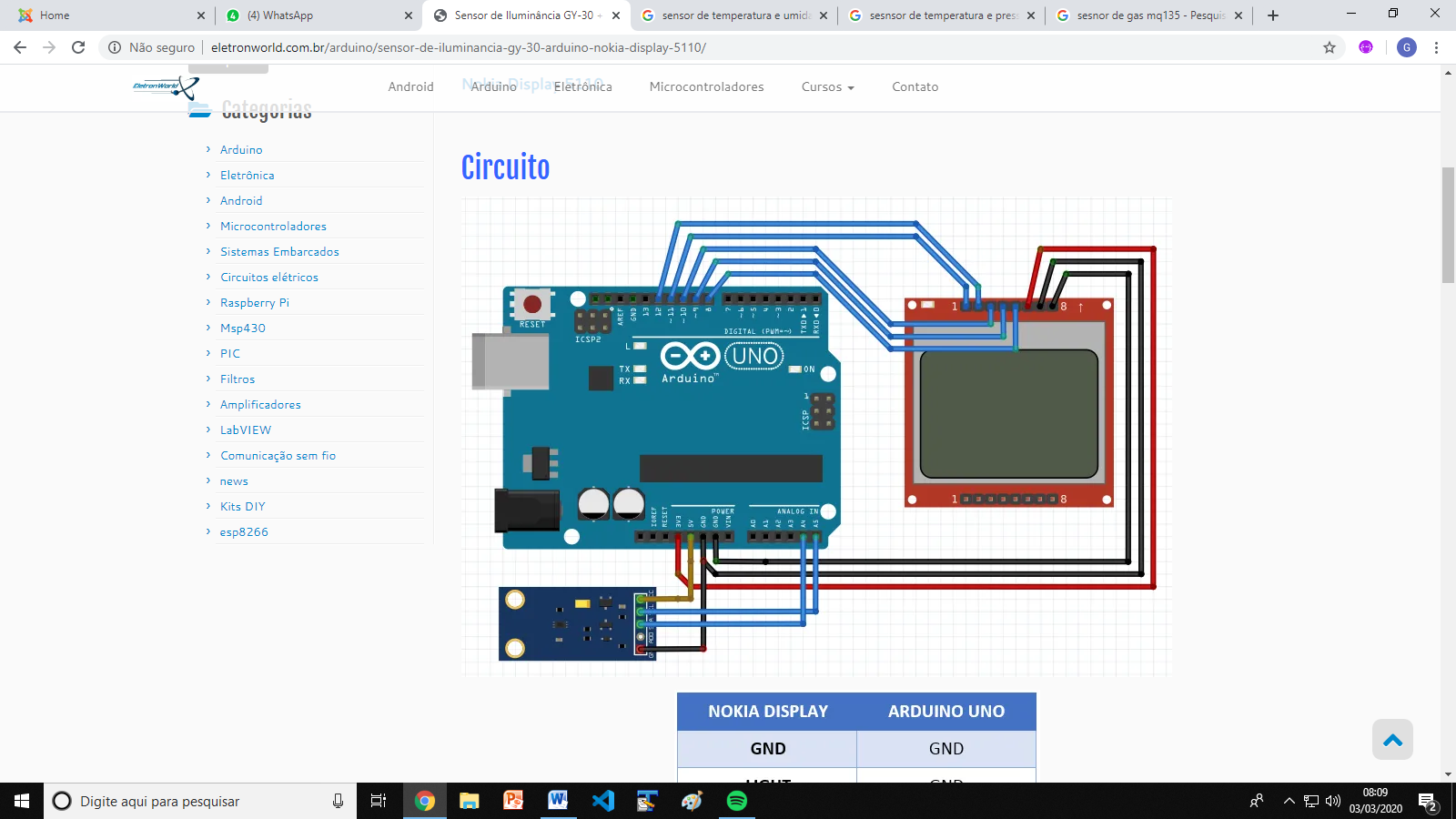
  digitalWrite(pino\_led\_amarelo, LOW);

  digitalWrite(pino\_led\_verde, LOW);

}

<https://www.filipeflop.com/blog/monitore-sua-planta-usando-arduino/>

1. SENSOR DE ILUMINANCIA GY-30 FUNFOU



Baixar - <https://github.com/claws/BH1750>

// Bibliotecas Utilizadas

#include <BH1750.h>

#include <Wire.h>

// Instância e variável para o sensor

BH1750 lightSensor;

String luz;

void setup() {

Serial.begin(9600);

  // Inicialização sensor de luz

  lightSensor.begin();

  }

void loop() {

  // Faz a leitura no sensor e atribui os valores

  uint16\_t lux = lightSensor.readLightLevel();

  luz = String(lux);

  Serial.print(“Luz: ”);

Serial.print(luz);

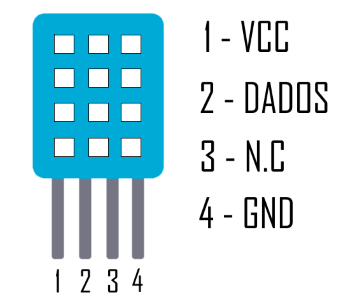
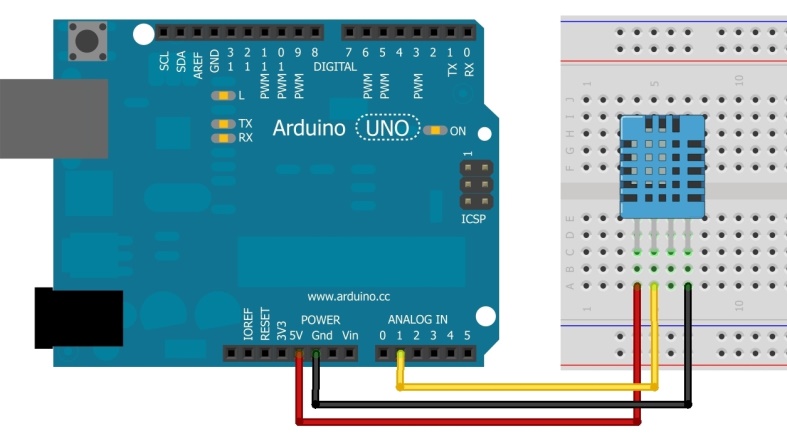
Serial.println();

  delay(150);

}

<http://eletronworld.com.br/arduino/sensor-de-iluminancia-gy-30-arduino-nokia-display-5110/>

1. SENSOR TEMPERATURA E UMIDADE DHT11 FUNFOU



Formato dos dados: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum = 40 bits.

Baixar - <https://github.com/adafruit/DHT-sensor-library>

Após o download descompacte o arquivo .zip e mova-o para a pasta arduinosketchfolder/libraries/ e reinicie a IDE do Arduino. Não retire o arquivo dht.cpp.  e não esqueça de renomear a pasta para “**DHT**”. Talvez será necessário criar uma sub-pasta da biblioteca caso não exista.

#include "DHT.h"

#define DHTPIN A1 // pino que estamos conectado

#define DHTTYPE DHT11 // DHT 11

// Conecte pino 1 do sensor (esquerda) ao +5V

// Conecte pino 2 do sensor ao pino de dados definido em seu Arduino

// Conecte pino 4 do sensor ao GND

// Conecte o resistor de 10K entre pin 2 (dados)

// e ao pino 1 (VCC) do sensor

DHT dht(DHTPIN, DHTTYPE);

void setup()

{

  Serial.begin(9600);

  //Serial.println("DHTxx test!");

  dht.begin();

}

void loop()

{

  // A leitura da temperatura e umidade pode levar 250ms!

  // O atraso do sensor pode chegar a 2 segundos.

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  // testa se retorno é valido, caso contrário algo está errado.

  if (isnan(t) || isnan(h))

  {

    Serial.println("Failed to read from DHT");

  }

  else

  {

    Serial.print("Umidade: ");

    Serial.print(h);

    Serial.print(" %t");

    Serial.print("Temperatura: ");

    Serial.print(t);

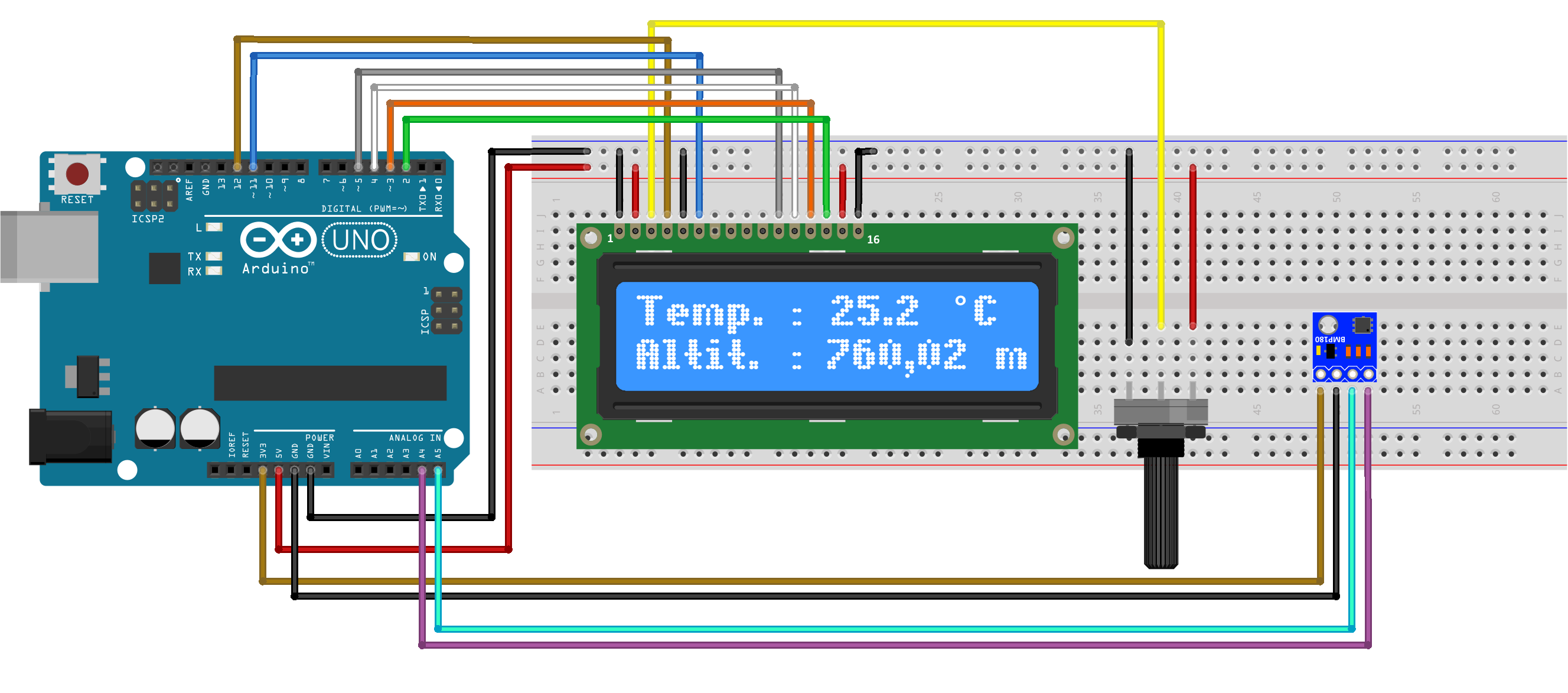
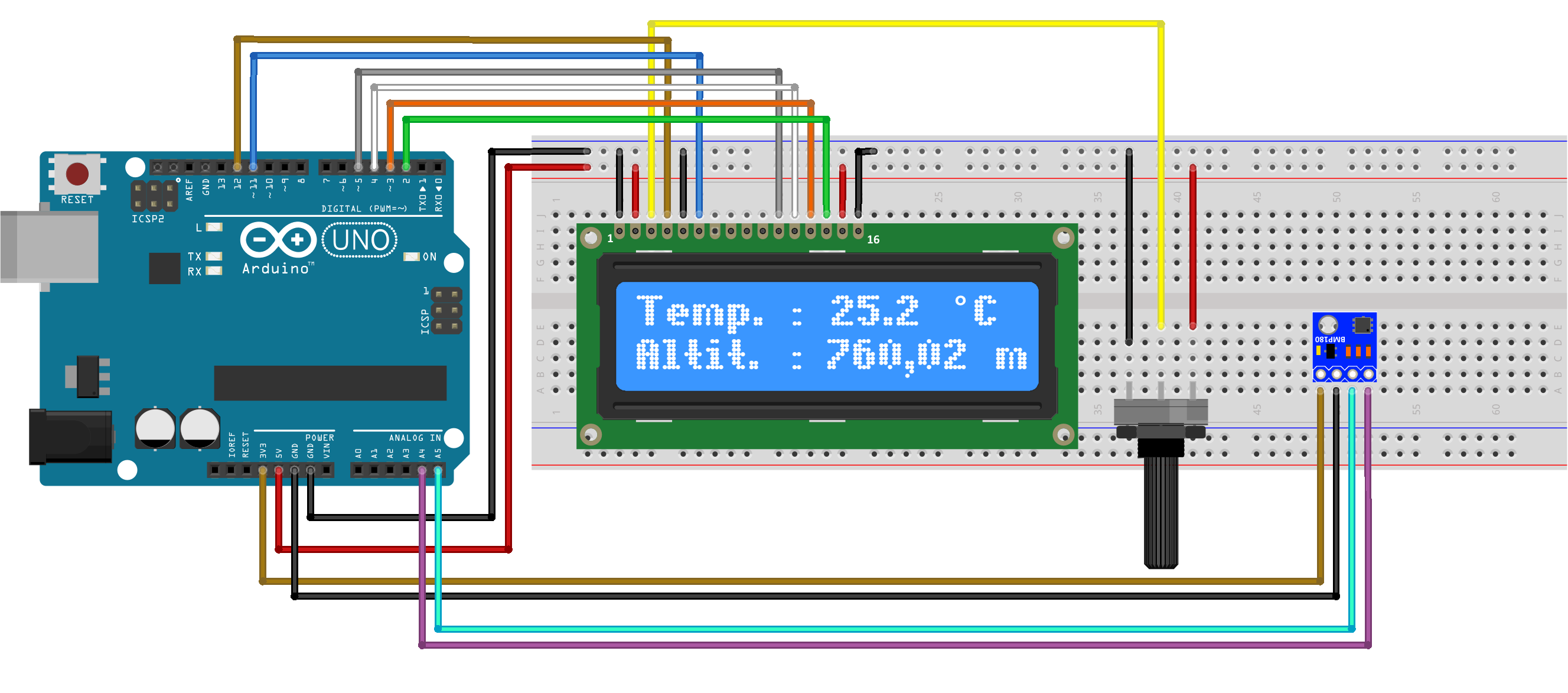
    Serial.println(" \*C");

  }

}

<https://www.filipeflop.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>

1. SENSOR TEMPERATURA E PRESSÃO DHT11 FUNFOU



- 3.3

- GND

- A5

- A4

Baixar -<https://github.com/adafruit/Adafruit-BMP085-Library>

Faça o download, descompacte o arquivo e coloque a pasta com a biblioteca dentro da pasta **LIBRARIES** da **IDE** do seu Arduino.

// Programa : Sensor de temperatura e pressao BMP180

// Autor : FILIPEFLOP

#include <Wire.h>

#include <Adafruit\_BMP085.h>

Adafruit\_BMP085 bmp180;

int mostrador = 0; **// pra q meu pai?**

void setup()

{

  Serial.begin(9600);

  if (!bmp180.begin())

  {

    Serial.println("Sensor nao encontrado !!");

    while (1) {}

  }

}

void loop()

{

   Serial.print("Temperatura : ");

   if ( bmp180.readTemperature() < 10)

     Serial.print(bmp180.readTemperature());

   else

     Serial.print(bmp180.readTemperature(),1);

Serial.println(" C");

   if (mostrador == 0)

   {

     Serial.print("Altitude : ");

     Serial.print(bmp180.readAltitude());

     Serial.println(" m");

    }

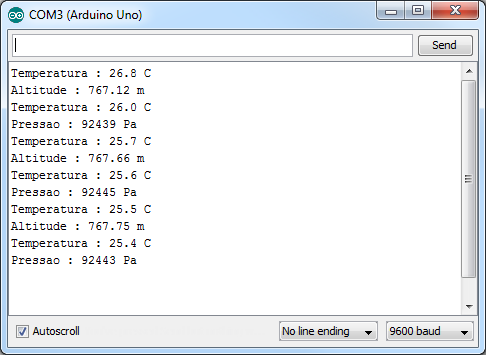
   if (mostrador == 1)

   {

     Serial.print("Pressao : ");

     Serial.print(bmp180.readPressure());

     Serial.println(" Pa");

   }

   delay(3000);

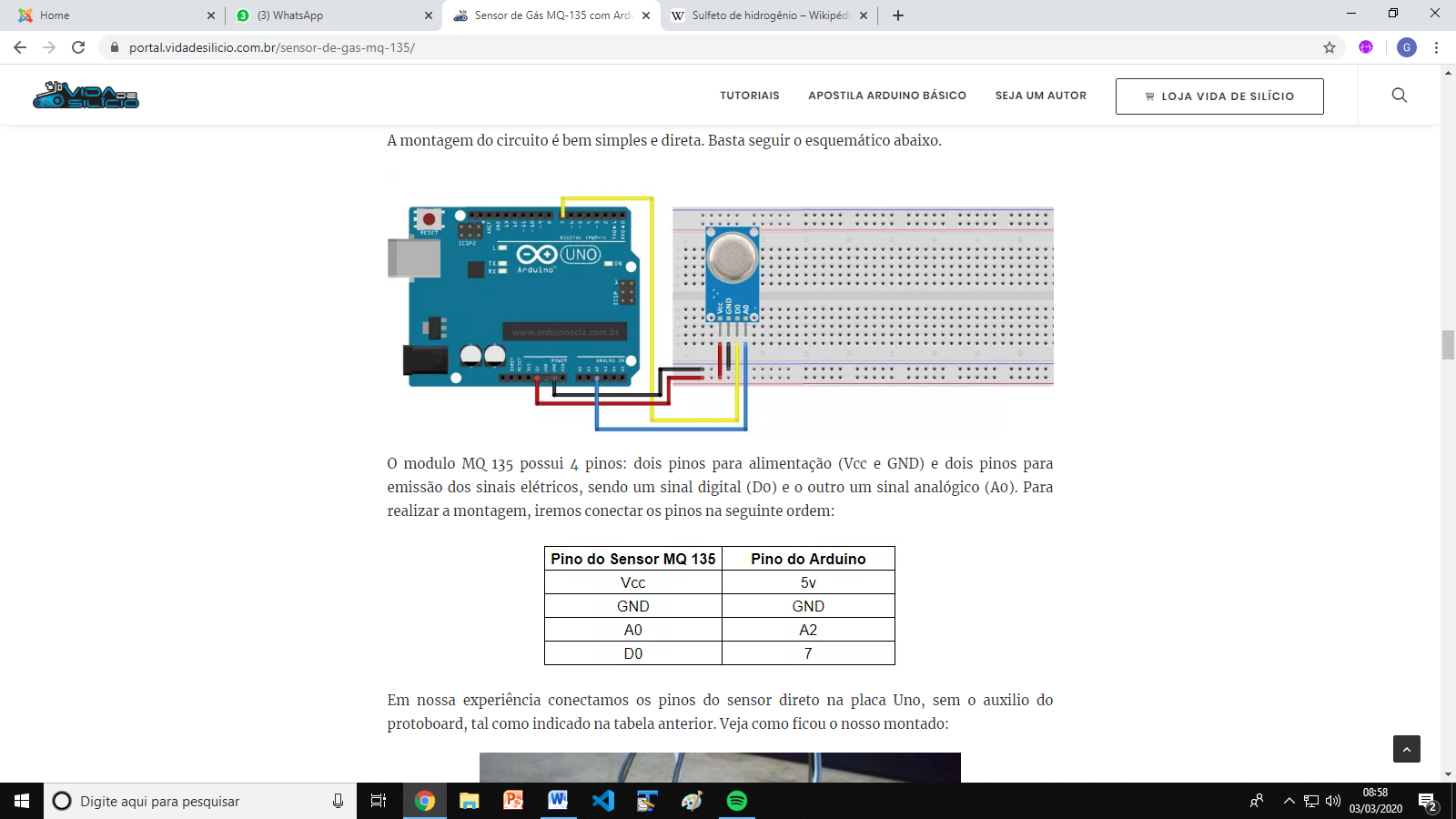
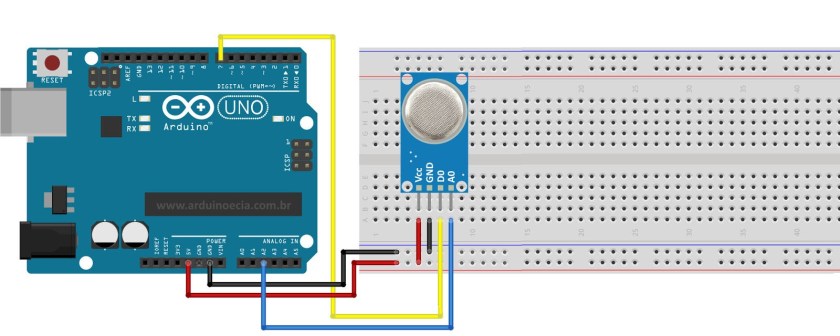
   mostrador = !mostrador;

}

[https://www.filipeflop.com/blog/ temperatura-pressao-bmp180-arduino/](https://www.filipeflop.com/blog/%20temperatura-pressao-bmp180-arduino/)

# SENSOR DE GÁS MQ-135 FUNFOU

- É capaz de detectar a concentração de vários gases tóxicos em um ambiente tais como: **amônia**, dióxido de carbono, **benzeno,** óxido nítrico e também fumaça ou álcool.



#define MQ\_analog A0

#define MQ\_dig 4

int valor\_analog;

int valor\_dig;

void setup() {

   Serial.begin(9600);

   pinMode(MQ\_analog, INPUT);

   pinMode(MQ\_dig, INPUT);

}

void loop() {

   valor\_analog = analogRead(MQ\_analog);

   valor\_dig = digitalRead(MQ\_dig);

   Serial.print(valor\_analog);

   Serial.print(" || ");

   if(valor\_dig == 0)

     Serial.println("GAS DETECTADO !!!");

   else

     Serial.println("GAS AUSENTE !!!");

   delay(500);

}

### – Calibrando o módulo digital

Para isso rode o programa anterior e abra o monitor serial e, com auxilio de uma pequena chave Philips, gire o potenciômetro na parte de trás do módulo até que o você encontre um ponto que altere de 0 pra 1. O valor analógico irá servir como base para mediar a sensibilidade do sensor.

### – Tempo de queima – Burn-in time

Para que o sensor esteja pronto para fazer as medições é necessário aguardar um tempo denominado como Burn-in time (Tempo de queima) ou preheat (pré aquecimento). Esse tempo varia de modelo para modelo, mas para testes mais simples, como para familiarizar com o sensor, podemos aguardar cerca de 3 minutos. Esse é o tempo necessário para que a resistência do sensor aqueça até a temperatura ideal de funcionamento do sensor. Durante esse período as medições podem oscilar muito.

Para aplicações oficiais, tal como implementação de um detector de incêndio, onde precisamos de uma precisão de medição maior e mais confiável, é recomendado pelos fabricantes que o tempo de queima seja de pelo menos 24 horas. Esse é o tempo necessário para que o sensor alcance a plena estabilização das medições entre os terminais internos A e B.

<https://portal.vidadesilicio.com.br/sensor-de-gas-mq-135/>