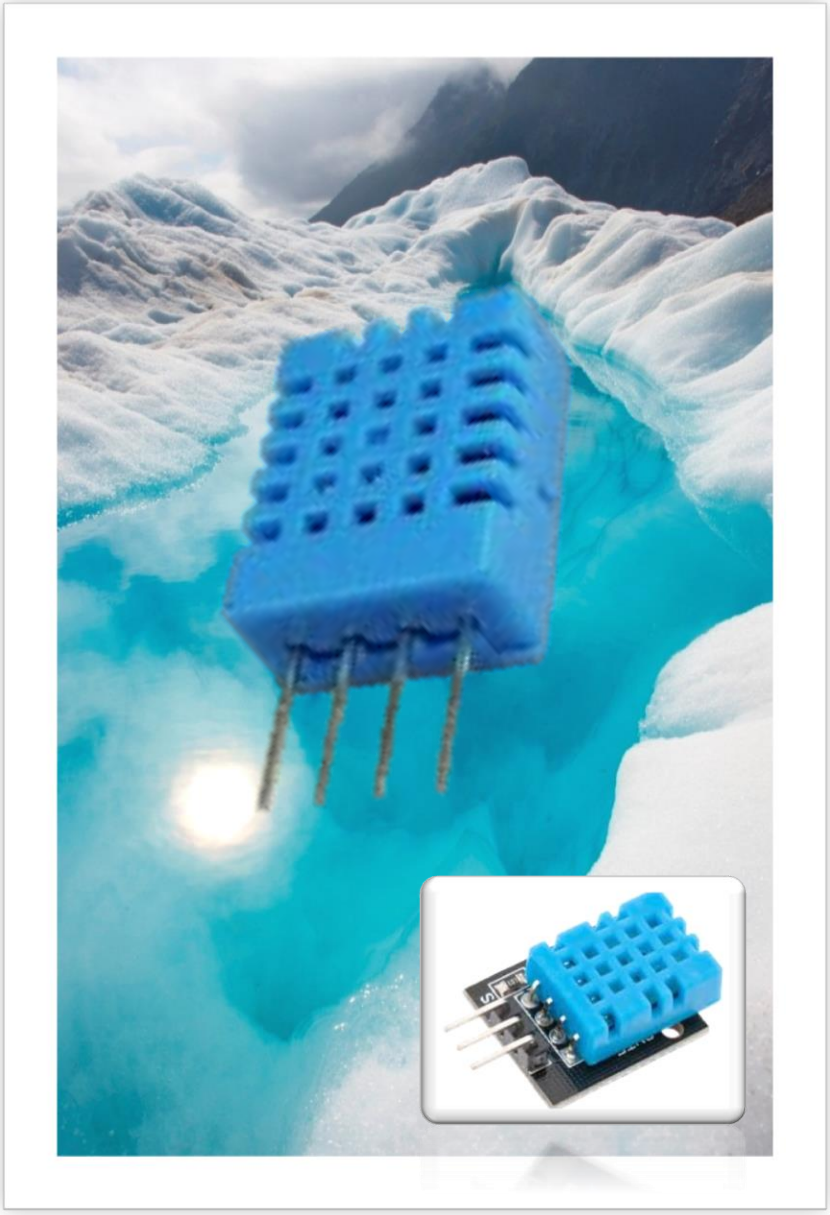


Sensor de DHT 11

ESTUDO E TESTE DAS SUAS CAPACIDADES

Nelson Proença | DHT11 | November 24, 2019

Conteúdo

Sensor de temperatura e humidade – DHT 11 3
Estrutura do Sensor – Recursos 3
As características do DHT11 são: 3
Esquema do Sensor 3
Aplicação do DHT 11 4
Exemplo de teste deste Sensor 4
Material usado 5
Circuito utilizado 6
Código usado 7
Explicação do código 9
Conclusão 11

Sensor de temperatura e humidade – DHT 11

Humidade e temperatura são duas grandezas que estão muito relacionadas quando o assunto é climatização.

Sendo assim, realizou-se este estudo para aprender a medir essas duas grandezas através do **sensor de humidade e temperatura DHT11** em conjunto com um Arduíno.

Estrutura do Sensor – Recursos

O sensor DHT11 e o sensor DHT22 são sensores básicos e de baixo custo que utilizam um termistor e um sensor capacitivo para medir a temperatura e a umidade do ar ambiente.

Esses sensores são bastante simples de usar, mas requer cuidado com o tempo entre duas leituras consecutivas, uma vez que ***é necessário um intervalo de, no mínimo, 1 segundo entre uma leitura e outra***.

Existem diferentes versões do DHT, similares na aparência e na pinagem, porém com características diferentes.

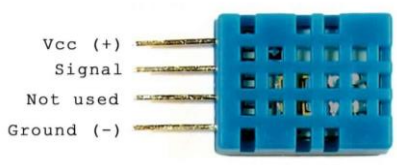
As características do DHT11 são:

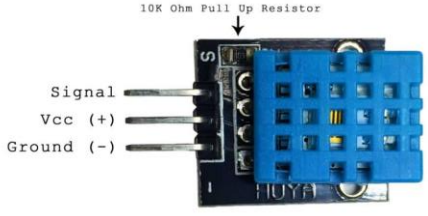
- Baixo custo
- Tensão de alimentação de 3V a 5V
- 2.5mA de corrente máxima durante a conversão
- Bom para medir humidade entre 20% e 80%, com 5% de precisão
- Bom para medir temperaturas entre 0 e 50°C, com ±2°C de precisão
- Taxa de amostragem de até 1Hz (1 leitura por segundo)
- Dimensões: 15.5mm x 12mm x 5.5mm
- 4 pinos com 0.1” de espaçamento entre eles

Esquema do Sensor

O sensor DHT11 tem 4 pinos, mas o pino 3 não é utilizado e deve permanecer desconectado quando se fizer o circuito.

O mais comum é encontrá-lo em forma de módulo, onde temos apenas 3 pinos: Vcc, Data e Gnd.





Aplicação do DHT 11

Este sensor poder ser aplicado em diversos equipamentos eletrónicos:

- Medições e monitorização de temperatura e humidade ambiente.

Exemplo de teste deste Sensor

Efetuou-se um teste a este sensor, afim de verificar os seus valores e perceber a sua aplicação.

Estudo do Sensor DHT11. Projeto para analise de temperatura e humidade, com o modulo DHT11, criação de Nelson, em colaboração com Gonçalo e Francisco.

Este projeto tem por finalidade reproduzir a ideia de previsão de tempo, feita pelo galo do tempo.

Assim, mediante as condições climatéricas registadas pelo DTH, se liga os leds RGB, com cores pré-estabelecidas:

- ✓ Azul: Sol;
- ✓ Ardosia: nuvens + sol;
- ✓ Violeta: nuvens;
- ✓ Plum: nuvens + aguaceiros;
- ✓ Rosa: nuvens + chuva + neve.

Esta escala poderá ser modificada, na configuração das cores.

Criação a 07 de novembro 2019.

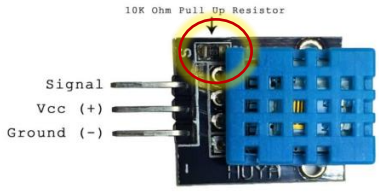
********** Revisão a 11 de novembro de 2019, por Nelson.**********

Este sensor utiliza uma biblioteca especifica a "DHT.h", assim as correções dos dados recolhidos já estão efetuadas, contudo tera de ser sempre testado.

Utiliza a porta digital (2), e será necessário verificar o tipo de sensor, esta biblioteca esta disponível para vários modelos de sensores: DHT11, DHT12, DHT22 e DHT21.

Sendo assim, definir o que esta a usar. Deve ter em atenção o tipo de Sensor e a sua estrutura quanto aos pinos, o usado foi o DHT11:

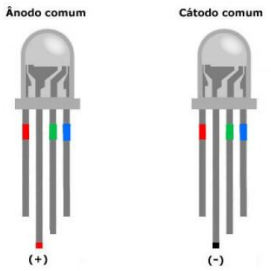
- ✓ pino 1 = Sinal (D2);
- ✓ pino 2 = Vcc (+);
- ✓ pino 3 = GND (-).



Utilizou-se um modulo que já trás uma resistência de 10K, tendo o seu próprio circuito, dai a configuração dos pinos anterior.

A ligação do led no Arduíno é feita em relação as cores, se for um led RGB atenção ao pino GND:

a) 1: Vermelho - pino 9
b) 3: Verde - pino 10
c) 4: Azul - pino 11
d) 2: Negativo - pino GND



Entre os pinos 9, 10, 11 e os pinos do led aplicar uma resistência conforme o esquema elétrico a baixo exemplificado.

Pode ser ligado um led de cada cor mas o efeito não será tão visível.

Ao usar a biblioteca não esquecer de identificar os seus parâmetros: " DHT dht(DHTPIN, DHTTYPE);"

Assim como a sua inicialização: "dht.begin();"

Lista de comandos desta biblioteca:

****Syntax Coloring Map For DHT-sensor-library*******

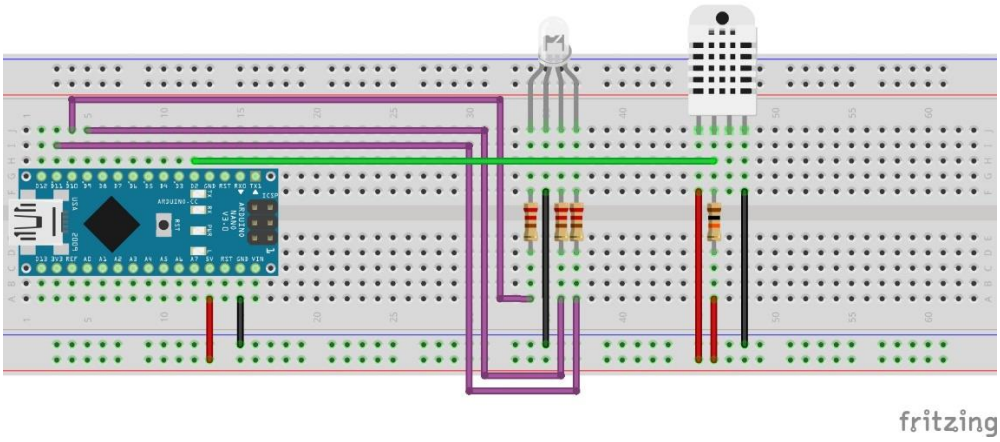
✓ Datatypes	KEYWORD1
✓ DHT	KEYWORD1
✓ Methods and Functions	KEYWORD2
✓ begin	KEYWORD2
✓ readTemperature	KEYWORD2
✓ convertCtoF	KEYWORD2
✓ convertFtoC	KEYWORD2
✓ computeHeatIndex	KEYWORD2
✓ readHumidity	KEYWORD2
✓ read	KEYWORD2

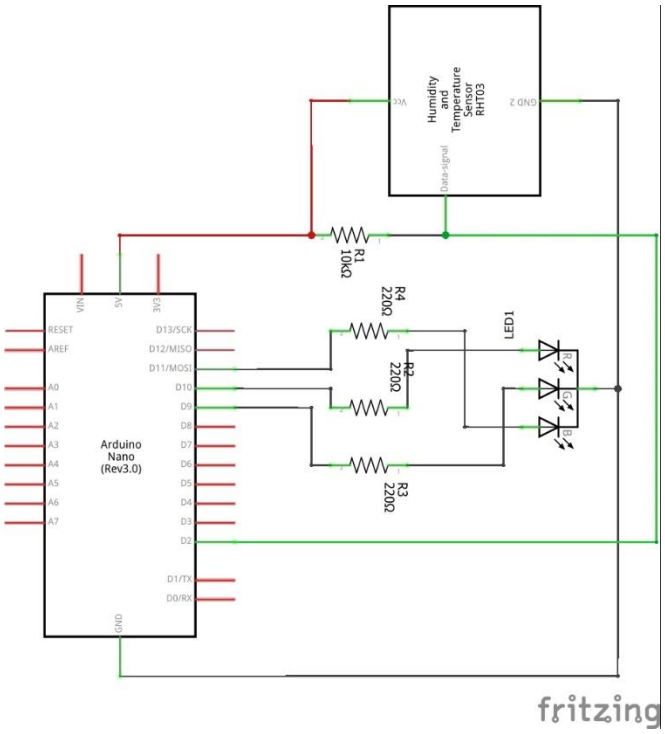
****

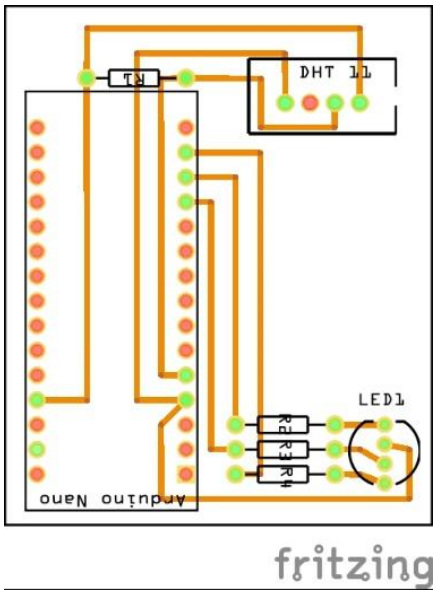
Material usado

1- Arduíno Nano
2- Sensor DHT11 (modulo)
3- Led RGB
4- 1 resistência de 10KΩ (no caso do Modulo não é necessário esta resistência)
5- 3 resistências, de 220Ω
6- Alguns fios de ligação
7- 1 Breadboard

Circuito utilizado







Neste esquema podemos verificar que foi usado o Arduíno nano, ligado ao sensor e às resistências ligadas aos respetivos pinos do led RGB.

O Arduíno fornece a alimentação ao Sensor, assim como ao led.

A fonte de alimentação do Arduíno foi a própria porta Usb.

Assim, o led é ligado pelas portas D9, D10 e D11 do Arduíno.

O sensor foi ligado a porta D2, afim de enviar os respetivos dados recolhidos.

O esquema de ligação do sensor e do led está exemplificado anteriormente.

Código usado

Para tornar o esquema mais apelativo criou-se o seguinte código:

```
#include "DHT.h"

#define DHTPIN 2    // pino digital para DHT sensor
#define DHTTYPE DHT11   // identificar o tipo de sensor utilizado: DHT 11

DHT dht(DHTPIN, DHTTYPE);

//identificar os pinos do led rgb
#define R 10 //pino vermelho
#define G 9 //pino verde
#define B 11 //pino azul

// Variaveis gerais
float h;
float t;

//Configuracao das cores RGB, a utilizar no led rgb
// Programar a cor RGB
void rosa(void)
{
  analogWrite(10, 255);
  analogWrite(9, 192);
  analogWrite(11, 203);
}
// Programar a cor RGB
void plum(void)
{
  analogWrite(10, 255);
  analogWrite(9, 192);
  analogWrite(11, 203);
}
// Programar a cor RGB
void violeta(void)
{
  analogWrite(10, 147);
  analogWrite(9, 112);
  analogWrite(11, 219);
}
// Programar a cor RGB
void ardosia()
{
  analogWrite(10, 0);
  analogWrite(9, 127);
  analogWrite(11, 255);
}
// Programar a cor RGB
void azul(void)
{
  analogWrite(10, 0);
  analogWrite(9, 0);
  analogWrite(11, 255);
}
// Inicializacao do programa
void setup()
{
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
  pinMode(R, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(B, OUTPUT);
}
```

```
void loop()
{
 // tempo de espera entre leituras do sensor
 delay(2000);
 // ler a humidade
 h = dht.readHumidity();
 // ler a temperatura em Celsius (por defeito)
 t = dht.readTemperature();

 // executar um teste ao sensor, para calibacao.
 if (isnan(h) || isnan(t))
 {
  Serial.println(F("Não é possível ler o sensor DHT! Verificar ligações..."));
  return;
 }
 Serial.print(F("Humidade: "));
 Serial.print(h);
 Serial.print(F("%  Temperatura: "));
 Serial.print(t);
 Serial.println(F("°C "));

 if ( h <= 25 && t >= 30)
 {
  azul();
  delay(1000);
 }

 if ((h > 25 && h < 38) && (t < 30 && t > 24))
 {
  ardosia();
  delay(1000);
 }

 if ((h > 38 && h < 50) && (t < 24 && t > 18))
 {
  violeta();
  delay(1000);
 }

 if ((h > 50 && h < 75) && (t < 18 && t > 6))
 {
  plum();
  delay(1000);
 }

 if ( h >= 75 && t <= 6)
 {
  rosa();
  delay(1000);
 }
}
```

Explicação do código

1. Na parte inicial inclui-se a biblioteca e as suas definições, de seguida definem-se as portas, e as variáveis necessárias para o programa:

```
#include "DHT.h"

#define DHTPIN 2     // pino digital para DHT sensor
#define DHTTYPE DHT11   // identificar o tipo de sensor utilizado: DHT 11

DHT dht(DHTPIN, DHTTYPE);

//identificar os pinos do led rgb
#define R 10 //pino vermelho
#define G 9 //pino verde
#define B 11 //pino azul

// Variaveis gerais
float h;
float t;
```

2. De seguida, construímos funções para ajudar a configuração. Assim criou-se cores pretendidas, as quais podem ser configuradas mediante a opção pretendida.
   É uma função simples com o "analogWrite()" , e como se utiliza os pinos PWM podemos usar diferentes valores (0-255):

```
//Configuracao das cores RGB, a utilizar no led rgb
// Programar a cor RGB
void rosa(void)
{
  analogWrite(10, 255);
  analogWrite(9, 192);
  analogWrite(11, 203);
}
```

… e restantes cores.

3. Iniciando o programa, no "setup" inicia-se o serial, inicia-se o programa "dht" e define-se as portas identificadas anteriormente como "OUTPUT":

```
// Inicializacao do programa
void setup()
{
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
  pinMode(R, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(B, OUTPUT);
}
```

4. Agora no "loop" inicia-se as repetições consecutivas:
   a) Configuração da temperatura que terá uma maneira especifica de ser lida, através da opção da biblioteca: `t = dht.readTemperature();` .

b) Configuração da humidade que terá uma maneira especifica de ser lida, através da opção da biblioteca: `h = dht.readHumidity(); .`
c) Faz o teste de calibração do sensor e verifica se esta tudo bem, com a ajuda do “if()” e de “isnan()”.
d) Pode-se verificar os registro das leituras no monitor serial, ou caso haja anomalias resista a respetiva mensagem.
e) Por fim, as funções antes criadas são inicializadas através da condição imposta pelo “if()”, onde dependendo da humidade e da temperatura assim realizará a respetiva função, anteriormente criada.

```
void loop()
{
 // tempo de espera entre leituras do sensor
 delay(2000);
 // ler a humidade
 h = dht.readHumidity();
 // ler a temperatura em Celsius (por defeito)
 t = dht.readTemperature();

 // executar um teste ao sensor, para calibacao.
 if (isnan(h) || isnan(t))
 {
  Serial.println(F("Não é possível ler o sensor DHT! Verificar ligações..."));
  return;
 }
 Serial.print(F("Humidade: "));
 Serial.print(h);
 Serial.print(F("%  Temperatura: "));
 Serial.print(t);
 Serial.println(F("°C "));

 if ( h <= 25 && t >= 30)
 {
  azul();
  delay(1000);
 }
```

Conclusão

- ✓ Estudo do sensor DHT 11 efetuado com sucesso, leituras registadas em diferentes ambientes.
- ✓ Aprendizagem e construção de circuitos elétricos.
- ✓ Verificação e exploração de diferentes funções do código no Arduíno.
    - #include – inserir Biblioteca
    - #define – atribuição de um nome a um valor constante
    - const – variável constante
    - float – variável para números decimais
    - if (x) – verifica uma condição e executa os respetivos procedimentos
    - analogWrite (x) – escreve um valor analógico (PWM)
    - analogRead (x) – faz a leitura analógica da respetiva porta
    - Serial.begin(x) – comunicação entre o Arduíno e o pc
    - Serial.print(x) – imprime a informação no monitor serial
    - Serial.println(x) – imprime a informação com quebra de linha
    - pinMode(x) – configuração do pino como input ou output
    - delay(x) – pausa especifica em milissegundos
    - Funções próprias da Biblioteca:
        - DHT dht(DHTPIN, DHTTYPE);
        - dht.begin();
        - h = dht.readHumidity();
        - t = dht.readTemperature();
        - isnan(x)

Objetivo concluído com sucesso.

Obrigado pela Atenção.

Informações e especificações recolhidas no datasheet do componente.

Para mais informações pode verificar no datasheet o respetivo fabricante do componente.