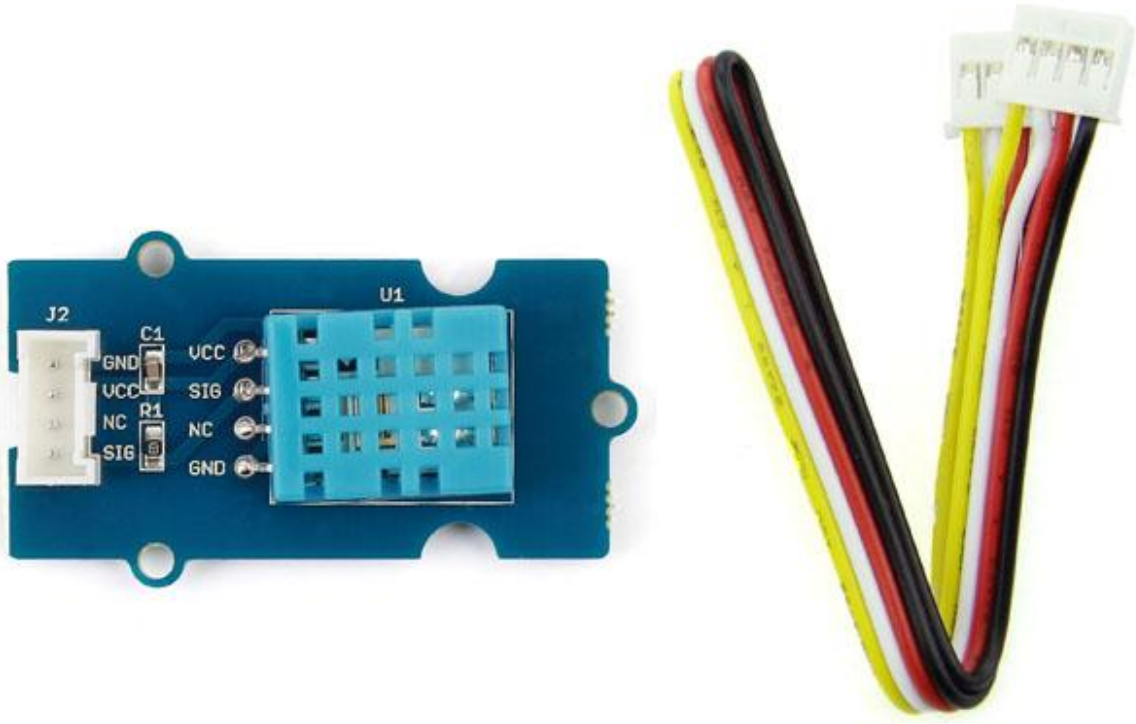
Grove - Temperature and Humidity Sensor SKU: 101020011



该温湿度传感器提供预校准的数字输出。 采用一个独立的电容传感元件测量相对湿度，温度则由负温度系数（NTC）热敏电阻测量。 具有良好的可靠性和长期的稳定性。 请注意，该传感器不适用于低于0度的温度。

产品特性

- 相对湿度和温度测量
- 全范围温度补偿校准
- 输出数字信号
- 能够长期稳定的运行
- 传输距离远（> 20m）
- 低功耗

!!!Tip 关于Grove模块的更多细节请参考 Grove System

创意应用

- 消费产品
- 气象台
- 湿度调节器
- 冷气机

硬件概述

规格参数

主要的规格参数

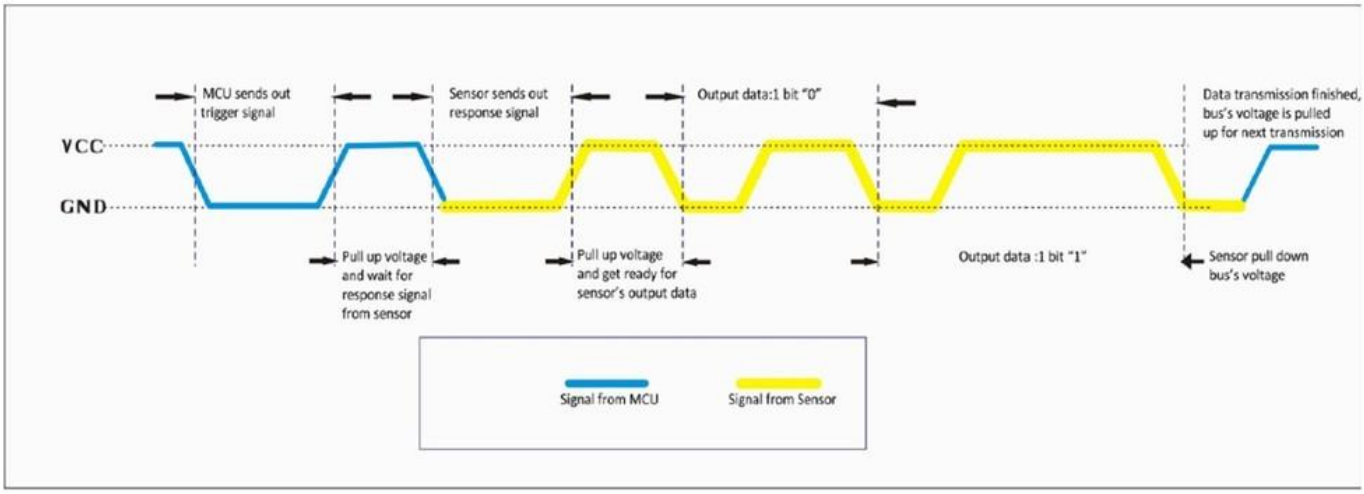
项目	规格
PCB尺寸	2.0cm*4.0cm
接口	2.0mm间距的针头
IO结构	SIG,VCC,GND,NC
ROHS	YES

电子特性

项目	条件	最小	标准	最大	单位
VCC	-	3.3	-	5	V
测量电流	-	1.3	-	2.1	mA
平均供应电流	-	0.5	-	1.1	mA
测量范围	湿度	20%	-	90%	RH
	温度	0	-	50	°C
准确性	湿度	-	-	±5%	RH
	温度			±2	°C
灵敏度	湿度		-	1%	RH
	温度			1	°C
再现性	湿度			±1%	RH
	温度			±1	°C
长期运行稳定性				±1%	RH/年
信号采集周期			2		S

使用方法

当MCU发送触发信号时，传感器将从低功耗模式切换到激活模式。 trigger signal传感器将响应信号发送回MCU后，它将传输出40位采集到的数据并触发新的信号采集（请注意，在触发信号到来之前MCU已经收集到从传感器发送来的40位采集到的数据）一个触发信号能够从传感器接收一次40位响应的数据。 单总线数据用于MCU和传感器之间的通信。 通信过程如下：



单次通信需要5ms，在高位的数据首先发出。 信号数据为40位，包括16位湿度数据，16位温度数据和8位校验数据。数据格式为：

```
8bits integer part of humidity+8bits decimal part of humidity
+8bits integer part of temperature+8bits decimal part of temperature
+8bits checksum.
```

程序

将温度和湿度传感器连接到模拟端口 **A0** .然后可以使用以下程序获得环境温度和湿度（代码仅适用于seeeduino，如果使用seeeduino mega，则应更改代码，将PINC更改为PINF，将DDRC更改为DDRF并将PORTC更改为PORTF）

```
#define DHT11_PIN 0      // ADC0

byte read_dht11_dat()
{
    byte i = 0;
    byte result=0;
    for(i=0; i< 8; i++){

        while(!(PINC & _BV(DHT11_PIN)));  // wait for 50us
        delayMicroseconds(30);

        if(PINC & _BV(DHT11_PIN))
        result |=(1<<(7-i));
        while((PINC & _BV(DHT11_PIN)));  // wait '1' finish
    }
    return result;
}

void setup()
{
    DDRC |= _BV(DHT11_PIN);
```

```
    PORTC |= _BV(DHT11_PIN);

    Serial.begin(9600);
    Serial.println("Ready");
}

void loop()
{
    byte dht11_dat[5];
    byte dht11_in;
    byte i;
    // start condition
    // 1. pull-down i/o pin from 18ms
    PORTC &= ~_BV(DHT11_PIN);
    delay(18);
    PORTC |= _BV(DHT11_PIN);
    delayMicroseconds(40);

    DDRC &= ~_BV(DHT11_PIN);
    delayMicroseconds(40);

    dht11_in = PINC & _BV(DHT11_PIN);

    if(dht11_in){
        Serial.println("dht11 start condition 1 not met");
        return;
    }
    delayMicroseconds(80);

    dht11_in = PINC & _BV(DHT11_PIN);

    if(!dht11_in){
        Serial.println("dht11 start condition 2 not met");
        return;
    }
    delayMicroseconds(80);
    // now ready for data reception
    for (i=0; i<5; i++)
    dht11_dat[i] = read_dht11_dat();

    DDRC |= _BV(DHT11_PIN);
    PORTC |= _BV(DHT11_PIN);

    byte dht11_check_sum = dht11_dat[0]+dht11_dat[1]+dht11_dat[2]+dht11_dat[3];
    // check check_sum
    if(dht11_dat[4]!= dht11_check_sum)
    {
        Serial.println("DHT11 checksum error");
    }

    Serial.print("Current humdity = ");
    Serial.print(dht11_dat[0], DEC);
    Serial.print(".");
    Serial.print(dht11_dat[1], DEC);
```

```
    Serial.print("%  ");
    Serial.print("temperature = ");
    Serial.print(dht11_dat[2], DEC);
    Serial.print(".");
    Serial.print(dht11_dat[3], DEC);
    Serial.println("C  ");

    delay(2000);
}
```

资源下载

- Temperature Humidity.zip
- Schematic at Easyeda