



Grove - Temperature and Humidity Sensor

Release date : 9/20/2015

Version : 1.0

Wiki: [http://www.seeedstudio.com/wiki/Grove- Temperature and Humidity Sensor](http://www.seeedstudio.com/wiki/Grove-_Temperature_and_Humidity_Sensor)

Bazaar: <http://www.seeedstudio.com/depot/Grove-TempHumi-Sensor-p-745.html>

Document Revision History

Revision	Date	Author	Description
1.0	Sep 21, 2015	Victor.He	Create file

Contents

Document Revision History	2
1. Introduction	2
2. Specification	3
3. Usage	4
3.1 With Arduino	4
3.2 With TI LaunchPad	6
3.3 With Raspberry Pi	9
4. Resources	12

Disclaimer

For physical injuries and possessions loss caused by those reasons which are not related to product quality, such as operating without following manual guide, natural disasters or force majeure, we take no responsibility for that.

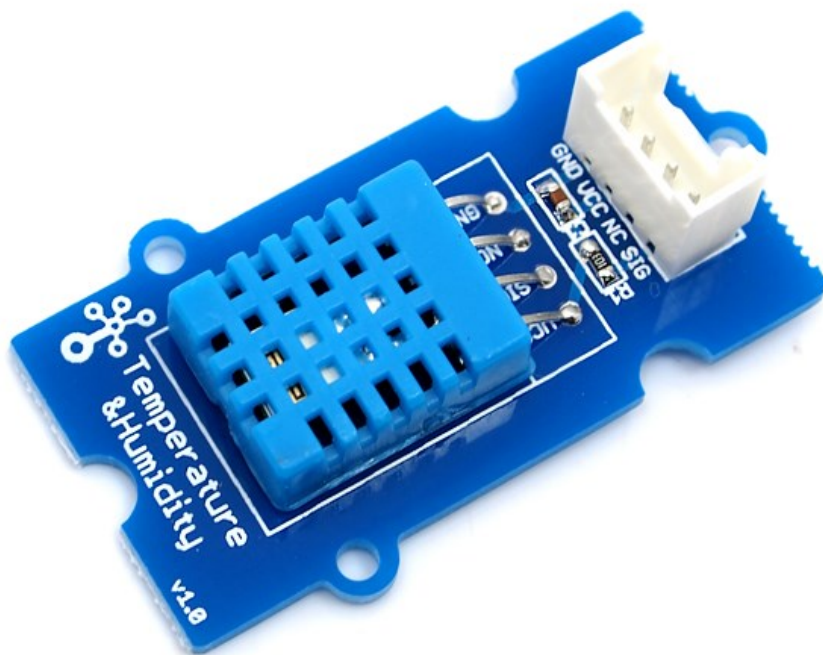
Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

Copyright

The design of this product (including software) and its accessories is under tutelage of laws. Any action to violate relevant right of our product will be penalized through law. Please consciously observe relevant local laws in the use of this product.

1. Introduction

This is a multifunctional sensor that gives you temperature and relative humidity information at the same time. It utilizes a DHT11 sensor that can meet measurement needs of general purposes. It provides reliable readings when environment humidity condition in between 20% RH and 90% RH, and temperature condition in between 0°C and 50°C, covering needs in most home and daily applications that don't contain extreme conditions.



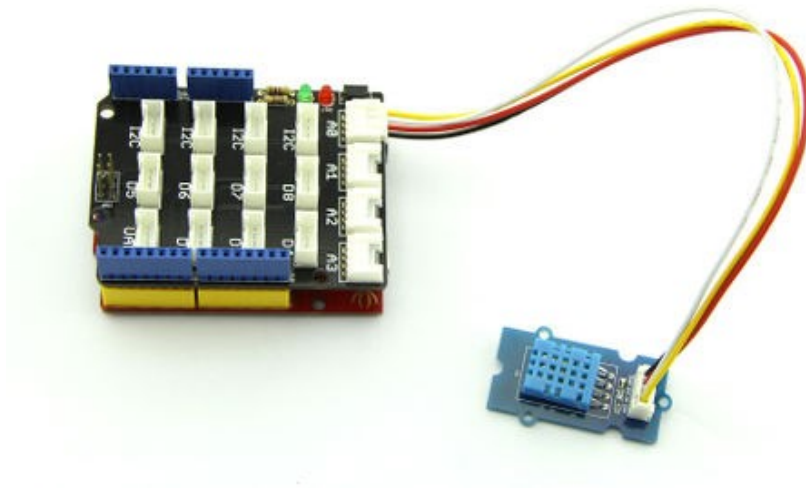
2. Specification

Work Voltage: 3.3V ~ 5V	
Measuring Range:	
Humidity: 20% - 90% RH	
Temperature: 0 ~ 50 ° C	
Accuracy:	
Humidity: ±5% RH	
Temperature: ±2° C	
Sensitivity:	
Humidity: ±1% RH	
Temperature: 1° C	
Signal Collecting Period: 2S	

3. Usage

3.1 With [Arduino](#)

This demo is going to show you how to read temperature and humidity information from this Grove - Temperature and Humidity Sensor.



The Temperature and Humidity sensor is connecting to analog port A0 of Grove - Base Shield.

- Download [Humidity Temperature Sensor library](#) for Arduino boards with 16MHz XTAL; Unzip and put it in the libraries file of Arduino IDE by the path: ..\arduino-1.0\libraries; Another library - [DHTlib](#) is also available instead of Humidity_Temperature_Sensor library. This library supports both 16MHz and 8MHz Arduino(eg. Seeeduino Stalker)
- Open “DHTtester” example via the path: File --> Examples --> Humidity_Temperature_Sensor --> DHTtester.

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

#include "DHT.h"

#define DHTPIN A0 // what pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302)
```

```
//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the
sensor

DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}

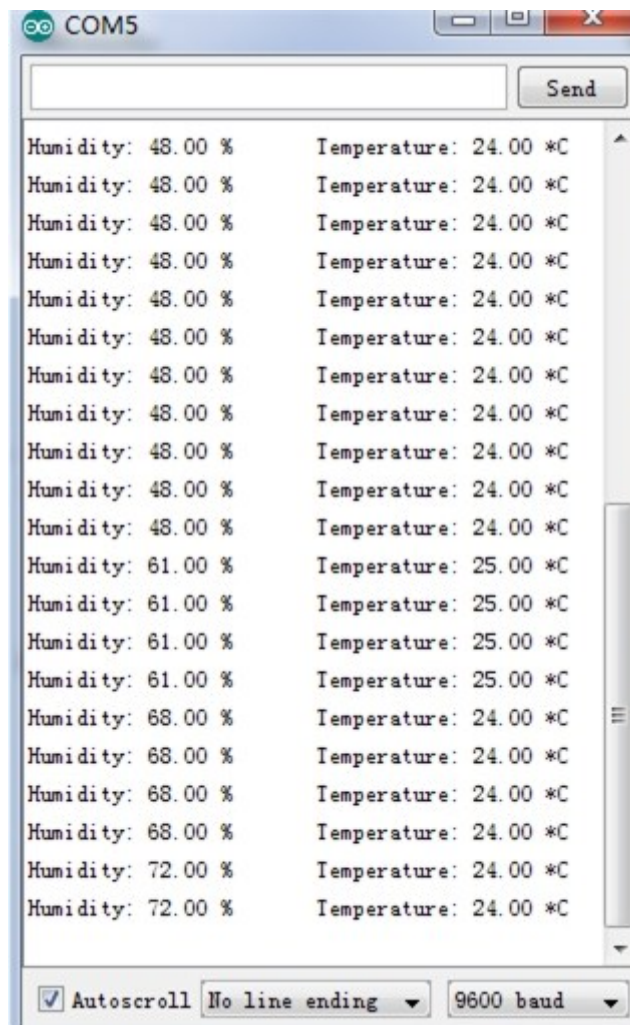
void loop()
{
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to A0 seconds 'old' (its a very
slow sensor)
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  // check if returns are valid, if they are NaN (not a number) then
something went wrong!
  if (isnan(t) || isnan(h))
  {
    Serial.println("Failed to read from DHT");
  }
  else
  {
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.println(" *C");
  }
}
```


Note: This Grove - Temperature and Humidity Sensor and our another product [Grove - Temperature and Humidity Sensor Pro](#) are sharing this library. No matter which product you are using, make sure that you have made the definition line of the sensor of your board into effect and commented out the definition lines of other specs. For example, the sensor used on Grove - Temperature and Humidity Sensor is DHT 11. So the definition part of the sensor spec should be:

```
#define DHTTYPE DHT11 // DHT 11
//#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
```

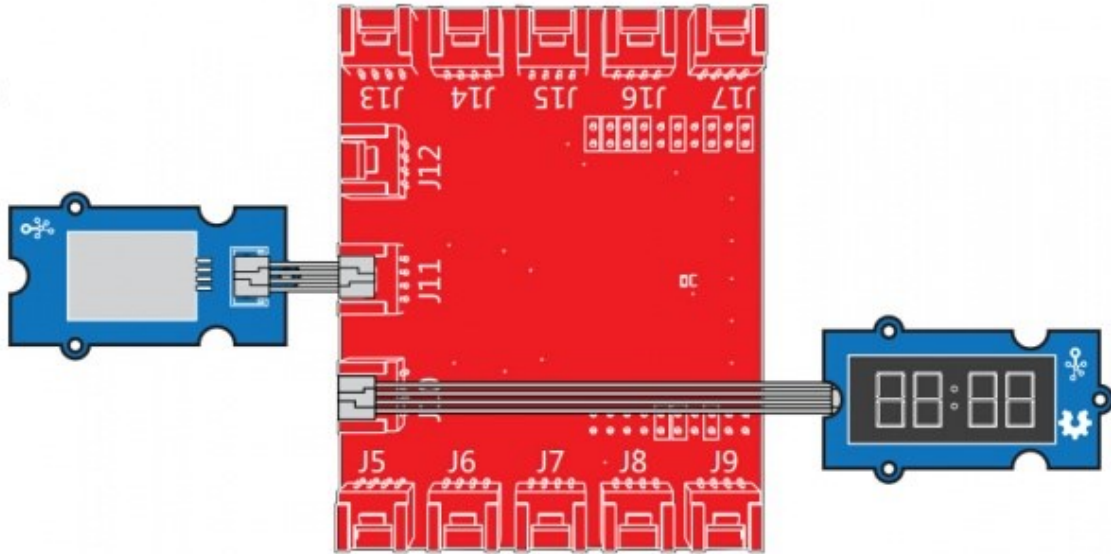
- Upload it into your Arduino board and open the serial monitor to observe the temperature and relative humidity information of the environment



3.2 With [TI LaunchPad](#)

Feeling the Environments (Temperature and humidity)

This example allows you to read the value from grove-temperature-humidity sensor. The temperature & humidity values are displayed on the Grove 4 digital display.



```

/*
Grove-Temperature-Humidity Sensor
Read value from grove-temperature-humidity sensor then display on grove-
4-digital-display

```

The circuit:

- * 4-digital-display attached to pin38 & pin39 (J14 plug on Grove Base BoosterPack)
- * sig pin of the Grove-Temperature-Humidity Sensor to the analog pin 24 (Grove connector J6)
- * one side pin (either one) to ground
- * the other side pin to +VCC
- * LED anode (long leg) attached to RED_LED
- * LED cathode (short leg) attached to ground

* Note: Put your hands on Grove-Temperature-Humidity Sensor, both of the value will rise.

4-digital-display:

```

|-----|
| temperature : humidity |
|-----|

```

This example code is in the public domain.

[http://www.seeedstudio.com/wiki/Grove -
 Temperature and Humidity Sensor Pro](http://www.seeedstudio.com/wiki/Grove_-_Temperature_and_Humidity_Sensor_Pro)

```

*/

```

```

#include "TM1637.h"
#include "DHT.h"

/* Macro Define */
#define CLK          39          /* 4-digital display clock pin
*/
#define DIO          38          /* 4-digital display data pin */
#define BLINK_LED    RED_LED    /* blink led */
#define TEMP_HUMI_PIN 24        /* pin of temperature&humidity
sensor */

/* Global Variables */
TM1637 tm1637(CLK, DIO);        /* 4-digital display object */
DHT dht(TEMP_HUMI_PIN, DHT22); /* temperature&humidity sensor
object */

int8_t t_bits[2] = {0};        /* array to store the single bits
of the temperature */
int8_t h_bits[2] = {0};        /* array to store the single bits
of the humidity */

/* the setup() method runs once, when the sketch starts */
void setup() {

    tm1637.init();              /* initialize 4-digital display */
    tm1637.set(BRIGHT_TYPICAL); /* set the brightness */
    tm1637.point(POINT_ON);     /* light the clock point ":" */

    dht.begin();               /* initialize temperature humidity
sensor */

    pinMode(RED_LED, OUTPUT);   /* declare the red_led pin as an
OUTPUT */
}

/* the loop() method runs over and over again */
void loop() {

    int _temperature = dht.readTemperature(); /* read the
temperature value from the sensor */
    int _humidity = dht.readHumidity();      /* read the
humidity value from the sensor */

```

```
    memset(t_bits, 0, 2);           /* reset array when
we use it */
    memset(h_bits, 0, 2);

    /* 4-digital-display [0,1] is used to display temperature */
    t_bits[0] = _temperature % 10;
    _temperature /= 10;
    t_bits[1] = _temperature % 10;

    /* 4-digital-display [2,3] is used to display humidity */
    h_bits[0] = _humidity % 10;
    _humidity /= 10;
    h_bits[1] = _humidity % 10;

    /* show it */
    tm1637.display(1, t_bits[0]);
    tm1637.display(0, t_bits[1]);

    tm1637.display(3, h_bits[0]);
    tm1637.display(2, h_bits[1]);

}
```

3.3 With [Raspberry Pi](#)

This example can Return temperature and humidity value,And show them on [Grove - LCD RGB Backlight](#).Also print them on PC terminal.



Notice: To use the method `dht(pin,module_type)` you need to change parameter `module_type` to 1 when using DHT pro and to 0 when using DHT.

```
# grovepi_lcd_dht.py

# dht(pin,module_type), change module_type number to use other kind of
dht
# module_type:
#           DHT11 0
#           DHT22 1
#           DHT21 2
#           DHT2301 3

from grovepi import *
from grove_rgb_lcd import *

dht_sensor_port = 7    # Connect the DHT sensor to port 7

while True:
    try:
        [ temp,hum ] = dht(dht_sensor_port,1)    #Get the temperature
and Humidity from the DHT sensor
        print "temp =", temp, "C\thumadity =", hum,"%"
        t = str(temp)
        h = str(hum)

        setRGB(0,128,64)
        setRGB(0,255,0)
        setText("Temp:" + t + "C    " + "Humidity :" + h + "%")
```

```
except (IOError,TypeError) as e:  
    print "Error"
```

Run the program

- Find the path to the file(According to your own path)

```
cd Desktop/Applications/GrovePi/GrovePi LCD DHT/
```

- Run Program

```
sudo python grovepi_lcd_dht.py
```

▪

4. Resources

[Temperature Humidity Eagle File](#)

[library](#)

[DHT11 datasheet](#)

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- Конкурентоспособные цены и скидки постоянным клиентам.
- Специальные условия для постоянных клиентов.
- Подбор аналогов.
- Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.
- Приемлемые сроки поставки, возможна ускоренная поставка.
- Доставку товара в любую точку России и стран СНГ.
- Комплексную поставку.
- Работу по проектам и поставку образцов.
- Формирование склада под заказчика.
- Сертификаты соответствия на поставляемую продукцию (по желанию клиента).
- Тестирование поставляемой продукции.
- Поставку компонентов, требующих военную и космическую приемку.
- Входной контроль качества.
- Наличие сертификата ISO.

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- Регистрацию проекта у производителя компонентов.
- Техническую поддержку проекта.
- Защиту от снятия компонента с производства.
- Оценку стоимости проекта по компонентам.
- Изготовление тестовой платы монтаж и пусконаладочные работы.



Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru