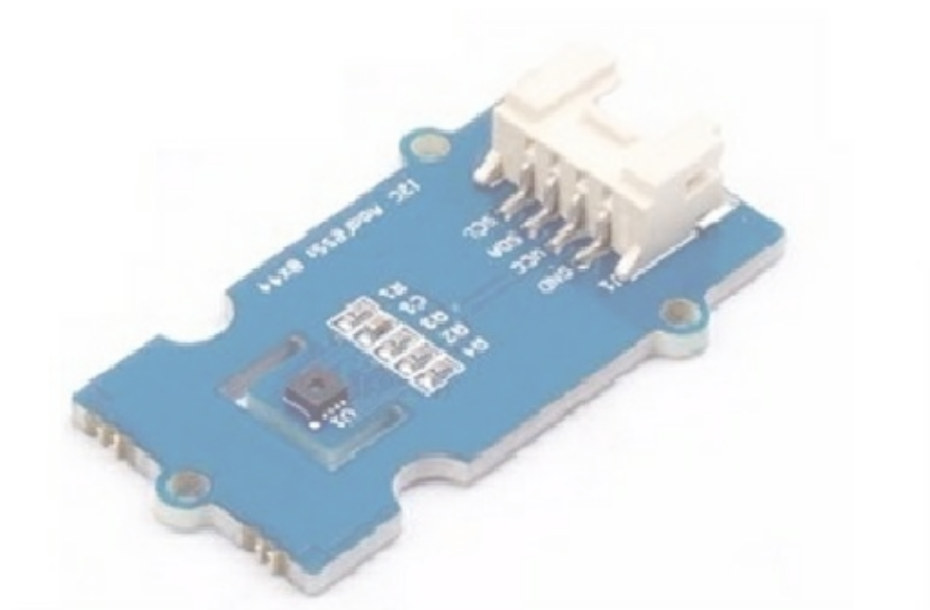


Grove - Temp and Humi Sensor(SHT31)



Grove - Temp&Humi Sensor(SHT31) is a highly reliable, accurate, quick response and integrated temperature & humidity sensor. The sensor(chip) used in the module is designed with Sensirion's CMOSens[®] technology. The chip is well calibrated, linearized and compensated for digital output.

The typical accuracy of this module can be $\pm 2\%$ RH (for relative humidity) and $\pm 0.3^{\circ}\text{C}$ (for temperature). This module is compatible with 3.3 Volts and 5 Volts and hence does not require a voltage level shifter. This module communicates using with I²C serial bus and can work up to 1 MHz speed. We also have provided a highly abstracted library to make this product more easier to use.

Using the sensor is easy. For [Seeeduino](#) (compliant with Arduino), just connect this breakout board with the main control board via [Grove cable](#). Then use the provided library and example/demo code available at [GitHub](#) to get your data. If you're using an Arduino without a Base Shield, simply connect the VIN pin to the 5V voltage pin, GND to ground, SCL to I2C Clock (Analog 5) and SDA to I2C Data (Analog 4).

Version

Product Version	Changes	Released Date
Grove - Temperature&Humidity Sensor(SHT31) V1.0	Initial	Jan 2016

Features

- Highly reliable, accurate and quick response time
- Grove compatible and easy to use
- Well calibrated, linearized, compensated for digital output
- Highly abstracted development library
- I2C Address 0x44

Note

If you want to use multiple I2C devices, please refer to [Software I2C](#).


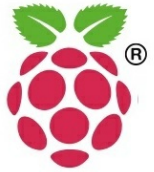
Tip

More details about Grove modules please refer to [Grove System](#)

Specifications

Parameter	Value
Input voltage (VCC)	3.3 volts or 5 volts
I/O Logic Level	3.3 volts or 5 volts based on VCC
Operating Current	100 μ A
Operating Temperature	-40–125 $^{\circ}$ C
Temperature Sensor Range	-40–125 $^{\circ}$ C, with $\pm 0.3^{\circ}$ C accuracy
Humidity Sensor Range	0% - 100%(Relative Humidity), with $\pm 2\%$ accuracy
Sensor Chip	SHT31(Datasheet)
Port	I ² C
Weight	4 g (for breakout board), 9 g for whole package each piece
Dimensions	40(length) \times 20(width) mm

Platforms Supported

Arduino	Raspberry Pi
	




Caution

The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Play With Arduino

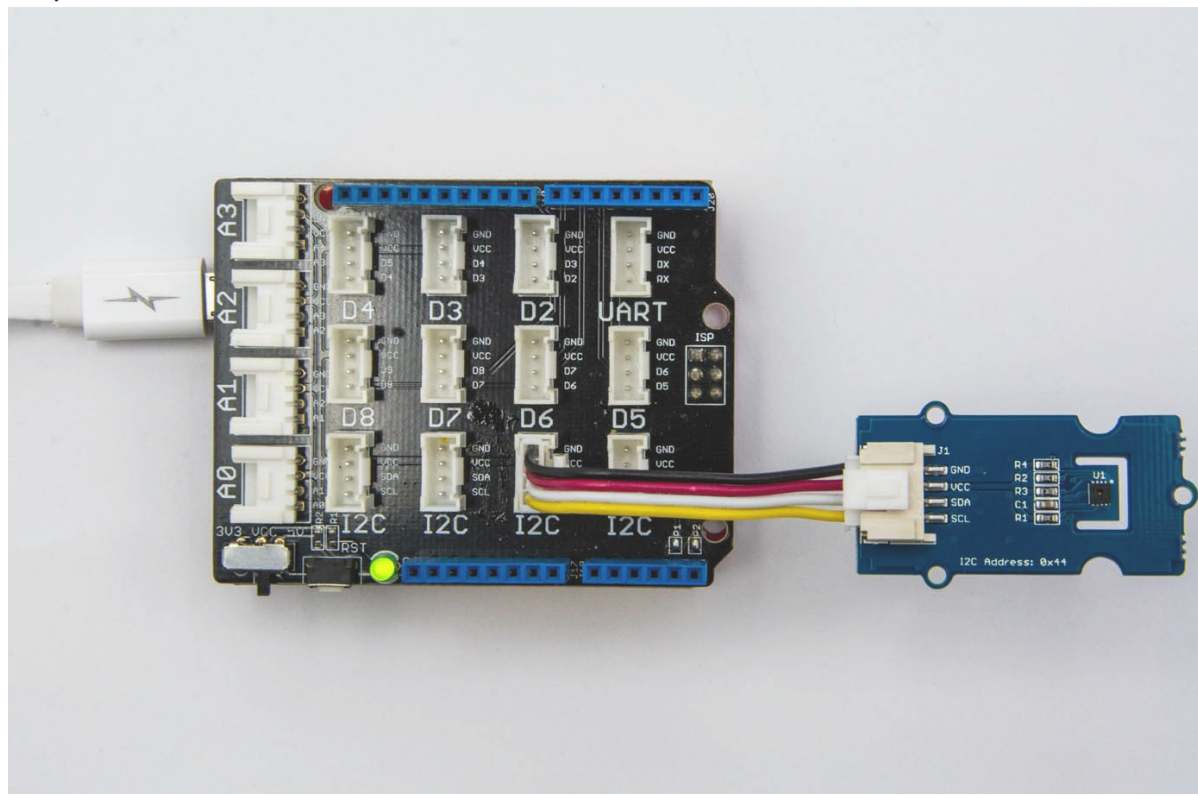
Hardware

- Step 1. Prepare the below stuffs:

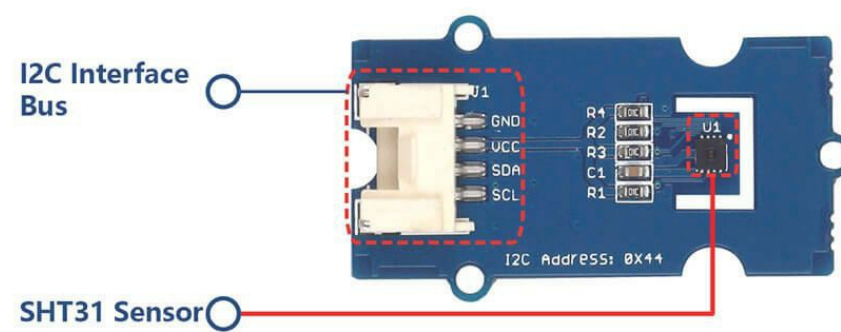
Seeeduino V4.2	Base Shield	Grove - Temp&Hum Sensor(SHT31)
		
Get One Now	Get One Now	Get One Now

- Step 2. Connect Grove - Temperature&Humidity Sensor(SHT31) to I2C port of Grove-Base Shield.
- Step 3. Plug Grove - Base Shield into Seeeduino.

- Step 4. Connect Seeeduino to PC via a USB cable.



Hardware Overview



Caution

Do not touch, shake or let this product vibrate while using. Otherwise, it will affect the accuracy of data measured.

Note

If we don't have Grove Base Shield, We also can directly Grove - Temperature&Humidity Sensor(SHT31) to Seeeduino as below.

Seeeduino	Grove - Temperature&Humidity Sensor(SHT31)
5V	Red
GND	Black
SDA	White
SCL	Yellow

Software

- Step 1. Download the [Library](#) from Github.
- Step 2. Refer to [How to install library](#) to install library for Arduino.
- Step 3. Restart the Arduino IDE. Open a new sketch, and copy the following code into the new sketch.

```

#include <Arduino.h>
#include <Wire.h>
#include "SHT31.h"

SHT31 sht31 = SHT31();

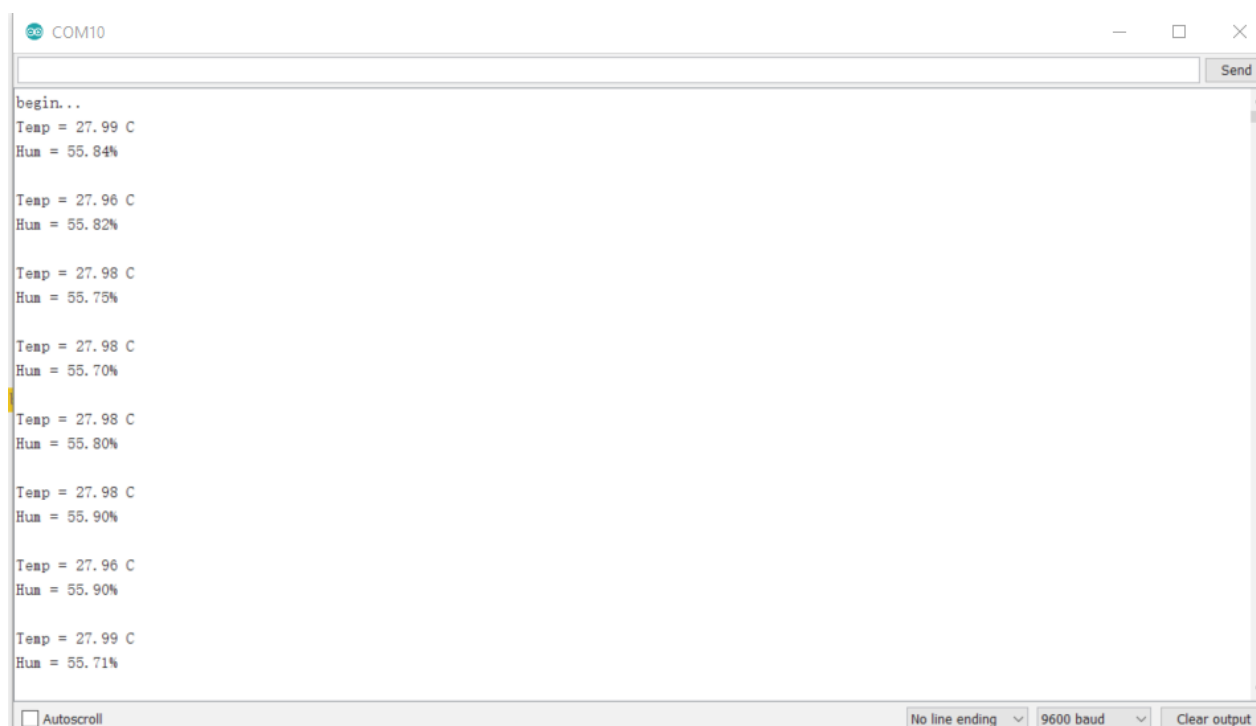
void setup() {
  Serial.begin(9600);
  while(!Serial);
  Serial.println("begin...");
  sht31.begin();
}

void loop() {
  float temp = sht31.getTemperature();
  float hum = sht31.getHumidity();
  Serial.print("Temp = ");
  Serial.print(temp);
  Serial.println(" C"); //The unit for Celsius because original arduino don't support special symbols
  Serial.print("Hum = ");
  Serial.print(hum);
  Serial.println("%");
  Serial.println();
  delay(1000);
}

```

- Step 4. Upload the demo. If you do not know how to upload the code, please check [How to upload code](#).
- Step 5. Open the Serial Monitor of Arduino IDE by click Tool-> Serial Monitor. Or tap the `Ctrl+Shift+M` key at the same time. Set the baud rate to 9600. If every thing goes well, you will get the results.

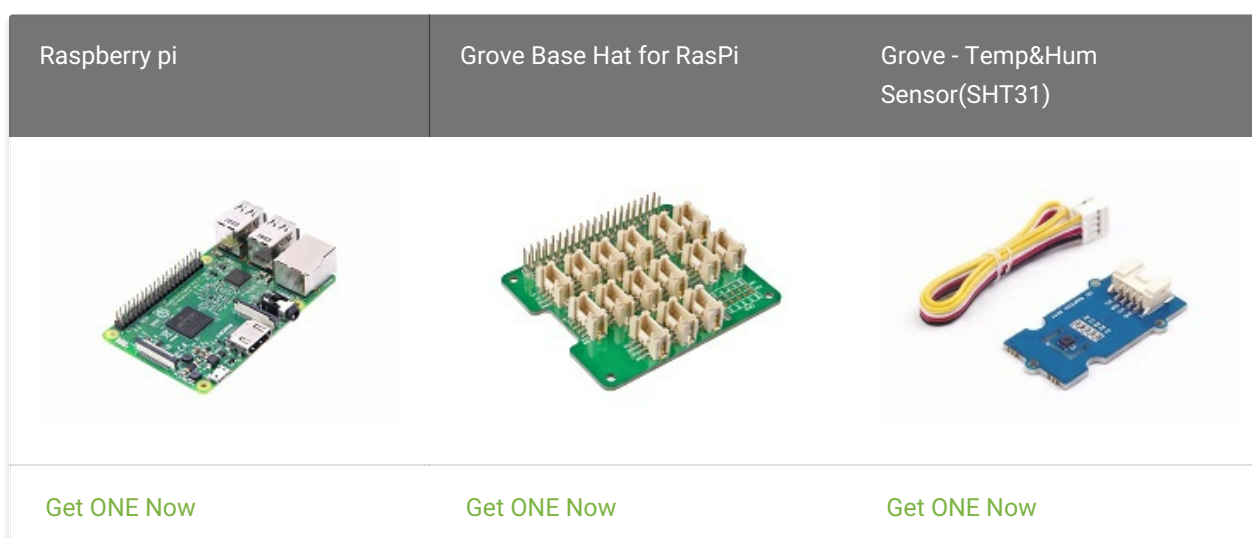
The result should be like:



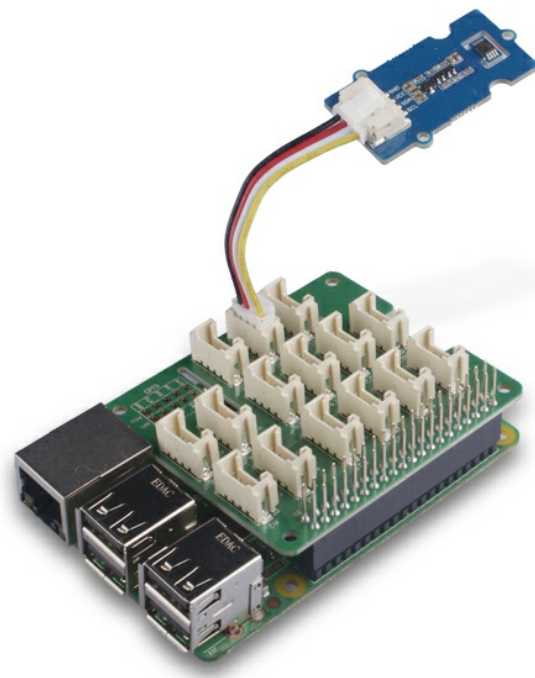
Play With Raspberry Pi

Hardware

- Step 1. Things used in this project:



- Step 2. Plug the Grove Base Hat into Raspberry.
- Step 3. Connect the Grove - Temperature&Humidity Sensor (SHT31) to the I²C port of the Base Hat.
- Step 4. Connect the Raspberry Pi to PC through USB cable.



Software

- Step 1. Follow [Setting Software](#) to configure the development environment.
- Step 2. Download the source file by cloning the grove.py library.

```
cd ~  
git clone https://github.com/Seeed-Studio/grove.py
```

- Step 3. Excute below command to run the code.

```
cd grove.py/grove  
python grove_temperature_humidity_sensor_sht3x.py
```

Following is the grove_temperature_humidity_sensor_sht3x.py code.

```

import time
from grove.i2c import Bus

def CRC(data):
    crc = 0xff
    for s in data:
        crc ^= s
        for _ in range(8):
            if crc & 0x80:
                crc <<= 1
                crc ^= 0x131
            else:
                crc <<= 1
    return crc

class GroveTemperatureHumiditySensorSHT3x(object):

    def __init__(self, address=0x44, bus=None):
        self.address = address

        # I2C bus
        self.bus = Bus(bus)

    def read(self):
        # high repeatability, clock stretching disabled
        self.bus.write_i2c_block_data(self.address, 0x24, [0x00])

        # measurement duration < 16 ms
        time.sleep(0.016)

        # read 6 bytes back
        # Temp MSB, Temp LSB, Temp CRC, Humidity MSB, Humidity LSB, Humidity CR
        data = self.bus.read_i2c_block_data(self.address, 0x00, 6)

        if data[2] != CRC(data[:2]):
            raise ValueError("temperature CRC mismatch")
        if data[5] != CRC(data[3:5]):
            raise ValueError("humidity CRC mismatch")

        temperature = data[0] * 256 + data[1]
        celsius = -45 + (175 * temperature) / 65535.0
        humidity = 100 * (data[3] * 256 + data[4]) / 65535.0

        return celsius, humidity

Grove = GroveTemperatureHumiditySensorSHT3x

def main():
    sensor = GroveTemperatureHumiditySensorSHT3x()
    while True:
        temperature, humidity = sensor.read()

        print("Temperature in Celsius is {:.2f} C".format(temperature))
        print("Relative Humidity is {:.2f} %".format(humidity))

        time.sleep(1)

if __name__ == "__main__":
    main()

```

Success

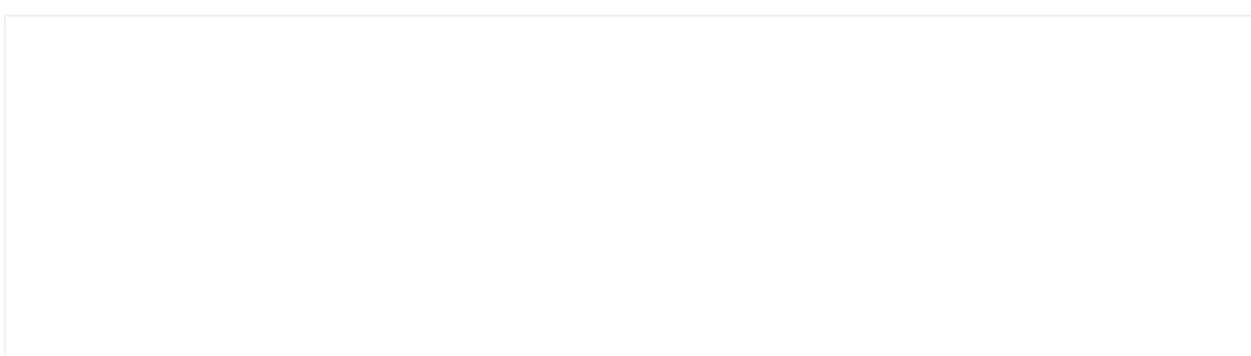
If everything goes well, you will be able to see temperature and humidity

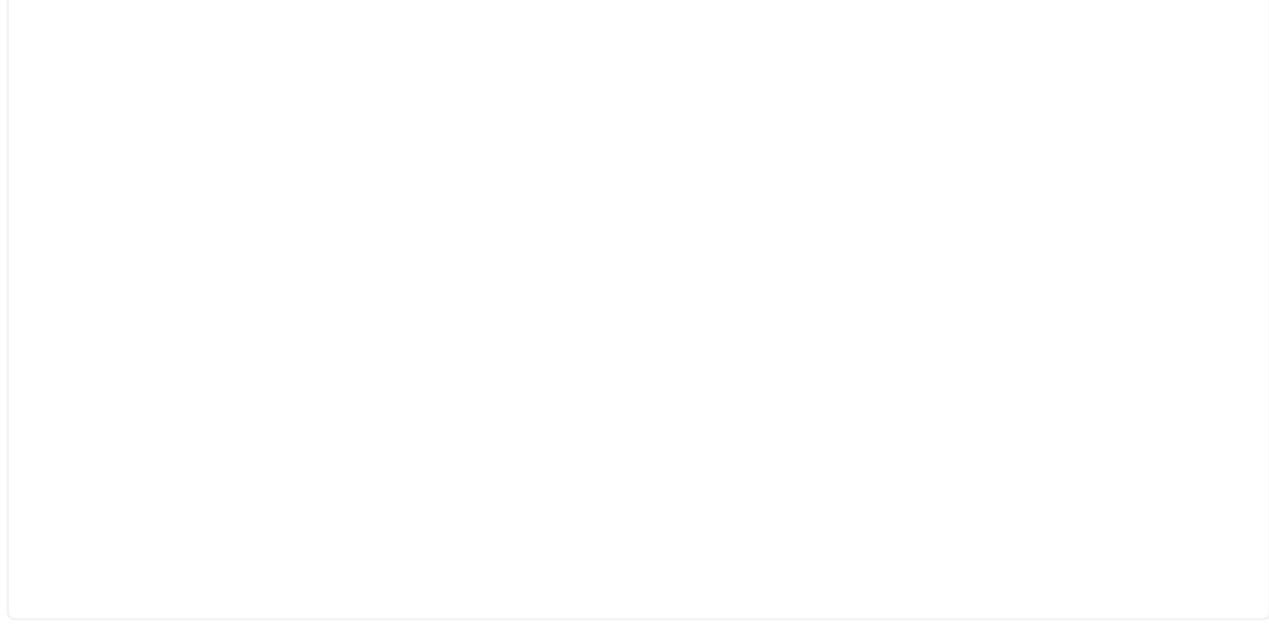
```

pi@raspberrypi:~/grove.py/grove $ python grove_temperature_humidity_sensor_sht3x.py
Temperature in Celsius is 21.48 C
Relative Humidity is 51.32 %
Temperature in Celsius is 21.47 C
Relative Humidity is 51.34 %
Temperature in Celsius is 21.46 C
Relative Humidity is 51.37 %
^CTraceback (most recent call last):
  File "grove_temperature_humidity_sensor_sht3x.py", line 95, in <module>
    main()
  File "grove_temperature_humidity_sensor_sht3x.py", line 91, in main
    time.sleep(1)
KeyboardInterrupt

```

Schematic Online Viewer





We have this part available in [geppetto](#), easy modular electronic design with Seeed and Geppeto. Build it Now. [geppetto.seeedstudio.com](#)

Resources

- [EAGLE] [Grove - Temperature&Humidity Sensor\(SHT31\) PCB files and PDF schematic](#)
- [Datasheet] [SHT31 Sensor Datasheet](#)
- [Library] [Library and example code](#)
- [MoreReading] [I²C How-to for Arduino](#)

Tech Support

Please submit any technical issue into our [forum](#).
