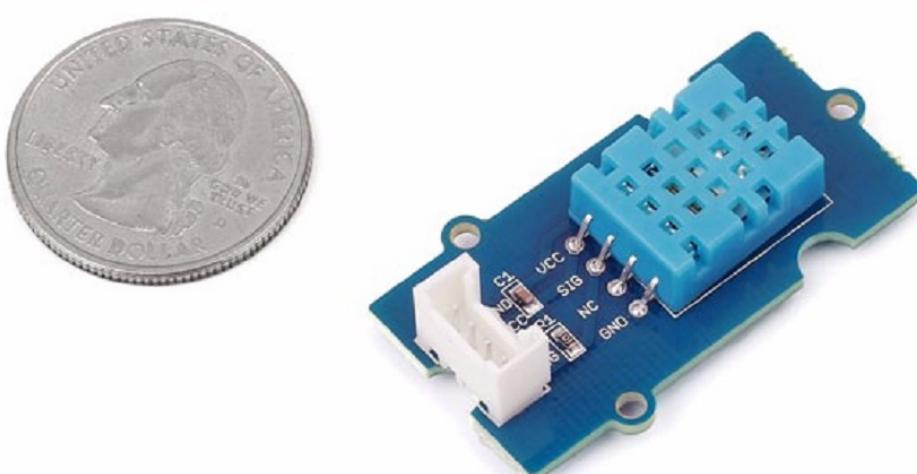




Grove - Temperature&Humidity Sensor (DHT11)



This Temperature&Humidity sensor provides a pre-calibrated digital output. A unique capacitive sensor element measures relative humidity and the temperature is measured by a negative temperature coefficient (NTC) thermistor. It has excellent reliability and long term stability. Please note that this sensor will not work for temperatures below 0 degree.

Get One Now

[<https://www.seeedstudio.com/Grove-Temperature-Humidity-Sensor-DHT11-p-745.html>]

Features

- Relative Humidity and temperature measurement
- Full range temperature compensation Calibrated
- Digital signal
- Long term stability
- Long transmission distance(>20m)
- Low power consumption

**Tip**

More details about Grove modules please refer to [Grove System](#) [https://wiki.seeedstudio.com/Grove_System/]

Applications Ideas

- Consumption product
- Weather station
- Humidity regulator
- Air conditioner

Specifications

Key Specifications

Items	Min
PCB Size	2.0cm*4.0cm
Interface	2.0mm pitch pin header
IO Structure	SIG,VCC,GND,NC
ROHS	YES

Electronic Characterstics

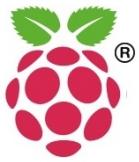
Items	Conditions	Min	Norm	Max	Unit
VCC	-	3.3	-	5	Volts
Measuring Current Supply	-	1.3	-	2.1	mA
Average Current Supply	-	0.5	-	1.1	mA
Measuring Range	Humidity	20%	-	90%	RH
	Temperature	0	-	50	°C
Accuracy	Humidity	-	-	±5%	RH
	Temperature			±2	°C
Sensitivity	Humidity		-	1%	RH
	Temperature			1	°C
Repeatability	Humidity			±1%	RH
	Temperature			±1	°C
Long-term Stability				±1%	RH/year
Signal Collecting Period			2		s

Platforms Supported

Arduino



Raspberry Pi

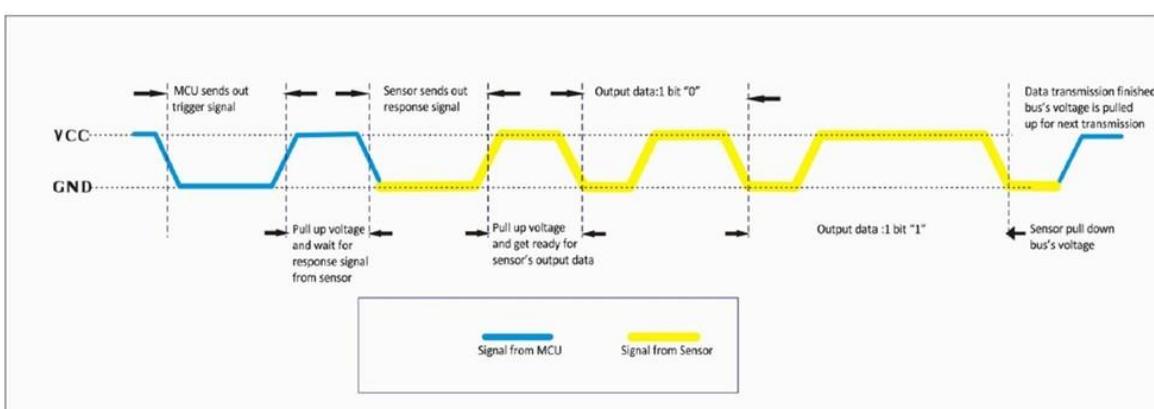


Note

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.

Getting Started

When MCU sends a trigger signal, sensor will change from low power consumption mode to active mode. After the trigger signal sensor will send a response signal back to MCU, then 40 bit collected data is sent out and a new signal collecting is triggered.(Note that the 40 bit collected data which is sent from sensor to MCU is already collected before the trigger signal comes.) One trigger signal receives one time 40 bit response data from sensor. Single-bus data is used for communication between MCU and sensor. The communication process is shown below:



It costs 5ms for single time communication. The high-order bit of data sends out first. Signal Data is 40 bit, comprised of 16 bit humidity data, 16 bit temperature data and 8 bit checksum. The data format is:

```

1  8bits integer part of humidity+8bits decimal part of humidity
2  +8bits integer part of temperature+8bits decimal part of temperature
3  +8bits checksum.

```



Note

If this is the first time you work with Arduino, we firmly recommend you to see [Getting Started with Arduino](#) [https://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

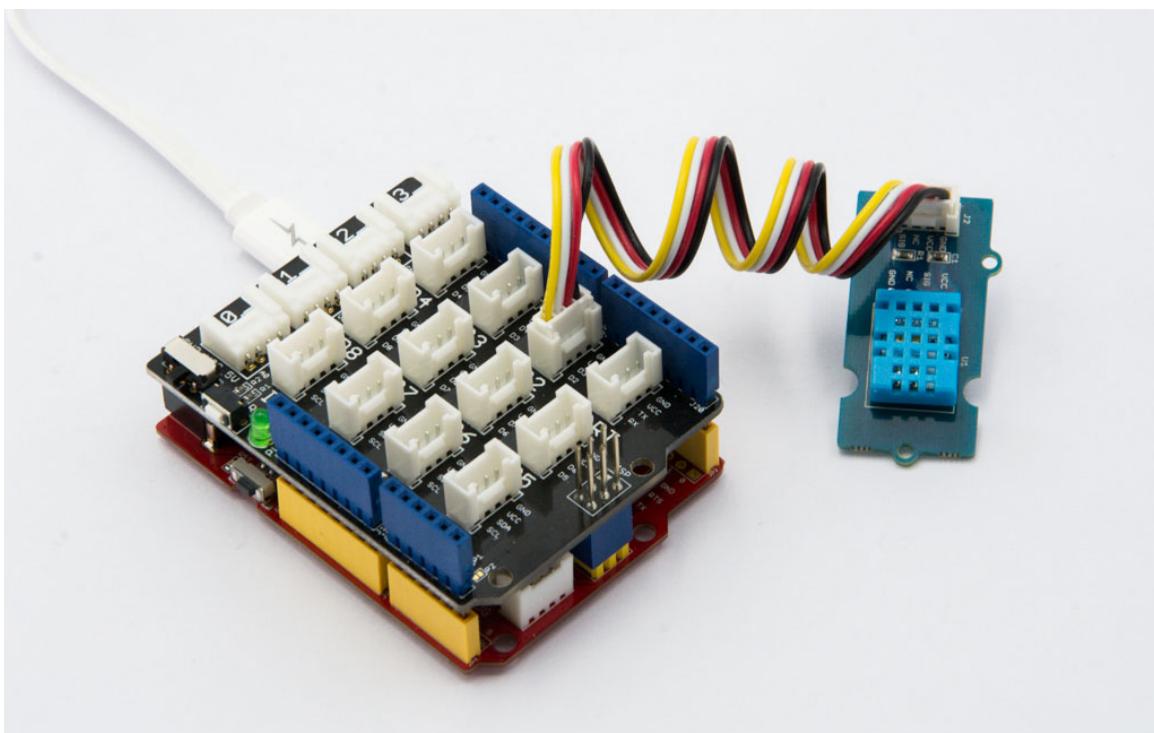
Play With Arduino

Hardware

- **Step 1.** Prepare the below stuffs:

Seeeduino V4.2	Base Shield	Temperature&Humidity Sensor
		
Get One Now [https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]	Get One Now [https://www.seeedstudio.com/Base-Shield-V2-p-1378.html]	Get One Now [https://www.seeedstudio.com/Grove-Temp%26Humi-Sensor-p-745.html]

- **Step 2.** Connect Grove - Temperature&Humidity Sensor to port **D2** of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.

**Note**

If we don't have Grove Base Shield, We also can directly connect Grove - Temperature and Humidity Sensor Pro to Seeeduino as below.

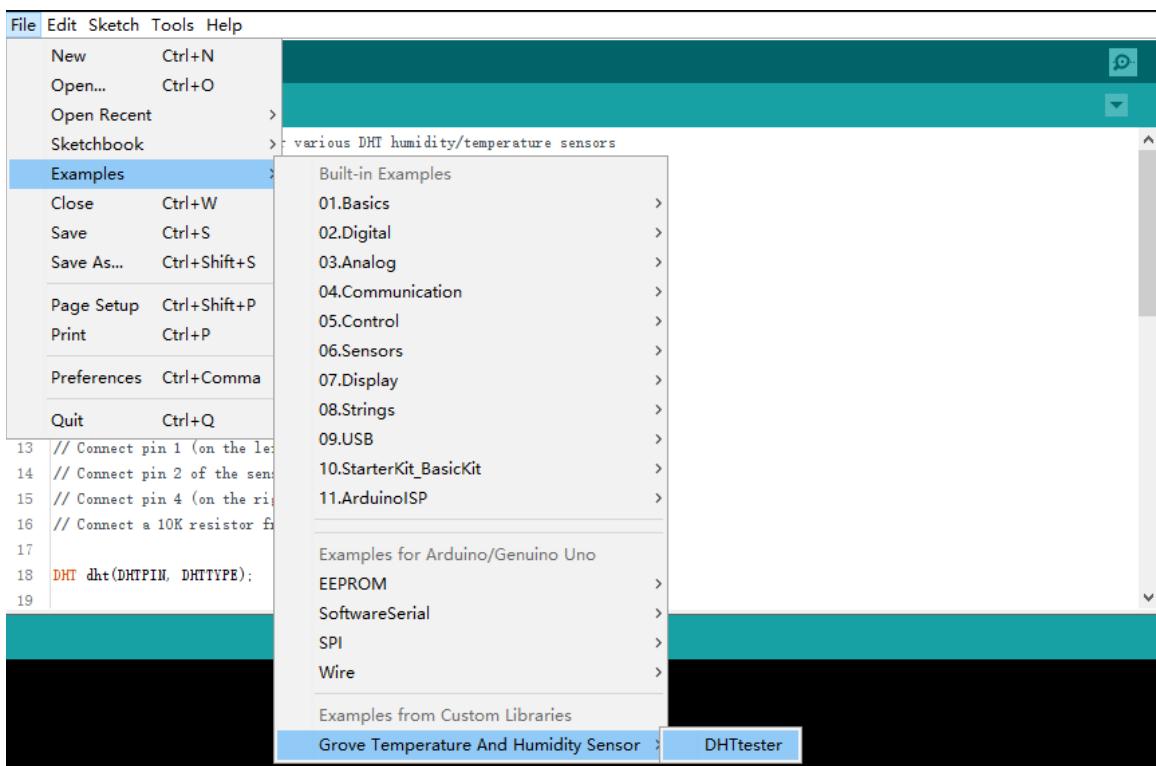
Seeeduino	Temperature&Humidity Sensor
5V	Red
GND	Black
Not Conencted	White
D2	Yellow

Software

- **Step 1.** Download the [Seeed DHT library](https://github.com/Seeed-Studio/Grove_Temperature_and_Humidity_Sensor) [https://github.com/Seeed-Studio/Grove_Temperature_and_Humidity_Sensor] from Github.
- **Step 2.** Refer [How to install library](https://wiki.seeedstudio.com) [<https://wiki.seeedstudio.com>]

/How_to_install_Arduino_Library] to install library for Arduino.

- **Step 3.** Restart the Arduino IDE. Open “ DHTtester” example via the path: **File → Examples → Grove_Humidity_Temperature_Sensor-master → DHTtester**. Through this demo, we can read the temperature and relative humidity information of the environment.



Note

This Grove - Temperature&Humidity Sensor and our another product [Grove-Temperature&Humidity Sensor pro](#) [https://wiki.seeedstudio.com/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 we used on Grove - Temperature&Humidity Sensor is DHT 11. So the definition part of the sensor spec should be:

```

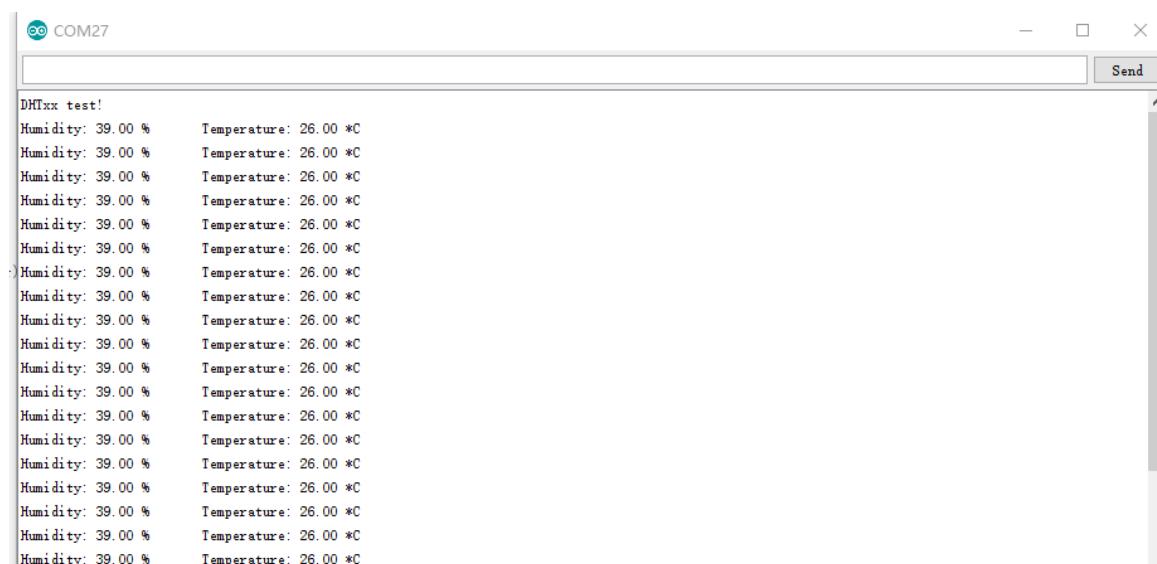
1 #define DHTTYPE DHT11 // DHT 11
2 // #define DHTTYPE DHT22 // DHT 22 (AM2302)
3 // #define DHTTYPE DHT21 // DHT 21 (AM2301)

```

The default setting of the library is DHT 22 , so you need to change it into DHT 11 manually.

- **Step 4.** Upload the demo. If you do not know how to upload the code, please check [how to upload code](#) [https://wiki.seeedstudio.com/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. if every thing goes well, you will get the temperature.

The result should be like:



The screenshot shows the Arduino Serial Monitor window titled "COM27". The window has a "Send" button in the top right corner. The text area displays multiple lines of sensor data, all showing 39.00% humidity and 26.00 *C temperature, indicating a loop or test mode.

```
DMTx test!
Humidity: 39.00 % Temperature: 26.00 *C
```

Play with Codecraft

Hardware

Step 1. Connect a Grove - Temperature&Humidity Sensor to port D2 a Base Shield.

Step 2. Plug the Base Shield to your Seeeduino/Arduino.

Step 3. Link Seeeduino/Arduino to your PC via an USB cable.

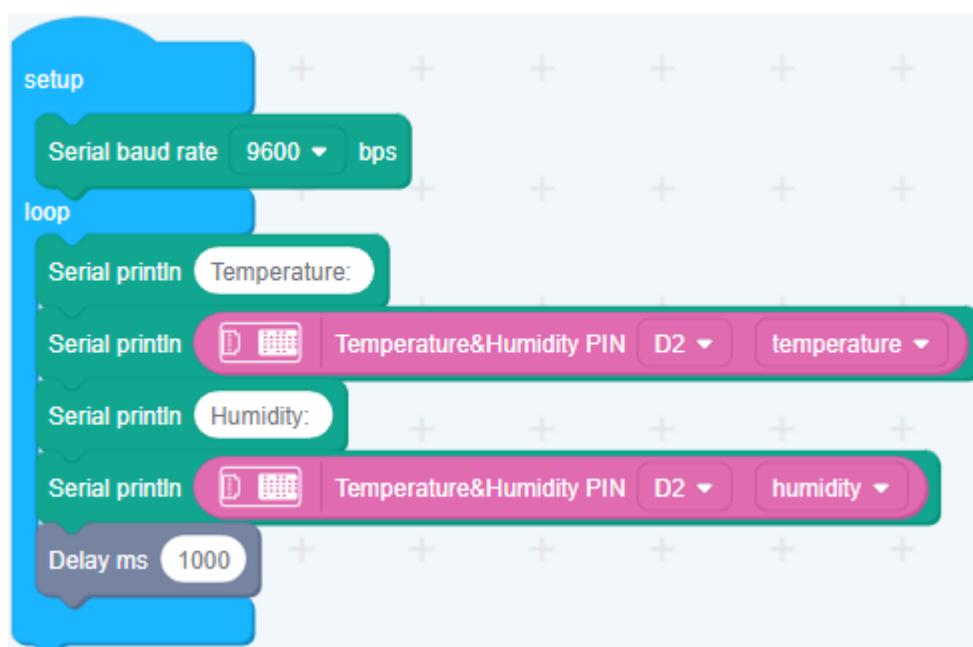
Software

Step 1. Open [Codecraft](#) [<https://ide.chmakered.com/>], add Arduino support, and drag a main procedure to working area.

**Note**

If this is your first time using Codecraft, see also [Guide for Codecraft using Arduino](#) [https://wiki.seeedstudio.com/Guide_for_Codecraft_using_Arduino/].

Step 2. Drag blocks as picture below or open the cdc file which can be downloaded at the end of this page.



Upload the program to your Arduino/Seeeduino.

**Success**

When the code finishes uploaded, you will see temperature and humidity displayed in the Serial Monitor.

Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

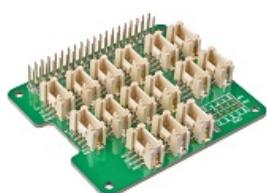
Hardware

- **Step 1.** Things used in this project:

Raspberry pi



Grove Base Hat for RasPi



Grove - Temp & Hum Sensor

**Get ONE Now**

[<https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html>]

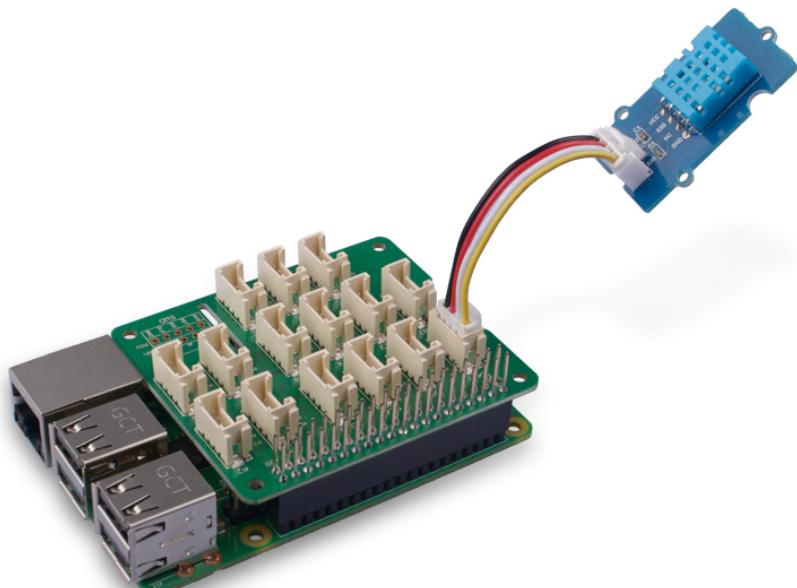
Get ONE Now

[<https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi-p-3186.html>]

Get ONE Now

[<https://www.seeedstudio.com/Grove-Temp%26Humi-Sensor-p-745.html>]

- **Step 2.** Plug the Grove Base Hat into Raspberry.
- **Step 3.** Connect the temperature and humidity sensor to Port 12 of the Base Hat.
- **Step 4.** Connect the Raspberry Pi to PC through USB cable.



Note

For step 3 you are able to connect the temperature and humidity sensor to **any GPIO Port** but make sure you change the command with the corresponding port number.

Software



Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- **Step 1.** Follow **Setting Software** [https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi/#installation] to configure the development environment.
- **Step 2.** Download the source file by cloning the `grove.py` library.

```
1 cd ~  
2 git clone https://github.com/Seeed-Studio/Seeed_Python_DHT.git
```

- **Step 3.** Execute below commands to run the code.

```
1 cd Seeed_Python_DHT  
2 sudo python3 setup.py install  
3 cd ~/Seeed_Python_DHT/examples  
4 python3 dht_simpleread.py
```

Following is the dht_simpleread.py code.

```
1 import time  
2 import seeed_dht  
3 def main():  
4  
5     # for DHT11/DHT22  
6     sensor = seeed_dht.DHT("11", 12)  
7     # for DHT10  
8     # sensor = seeed_dht.DHT("10")  
9  
10    while True:  
11        humi, temp = sensor.read()  
12        if not humi is None:  
13            print('DHT{0}, humidity {1:.1f}%, temperature {2:.1f}*'.format(sensor.dht_type, humi, temp))  
14        else:  
15            print('DHT{0}, humidity & temperature: {1}'.format(sensor.dht_type, temp))  
16        time.sleep(1)  
17  
18  
19 if __name__ == '__main__':  
20     main()
```



Success

If everything goes well, you will be able to see the following result

```
1 pi@raspberrypi:~/Seeed_Python_DHT/examples $ python3 dht_simpleread.py  
2 DHT11, humidity 39.2%, temperature 29.1*  
3 DHT11, humidity 39.2%, temperature 29.1*  
4 DHT11, humidity 39.2%, temperature 29.1*  
5 DHT11, humidity 39.1%, temperature 29.1*  
6 DHT11, humidity 40.0%, temperature 29.1*
```

```
7 DHT11, humidity 39.9%, temperature 29.1*
8 DHT11, humidity 40.3%, temperature 29.1*
9 DHT11, humidity 42.0%, temperature 29.1*
```

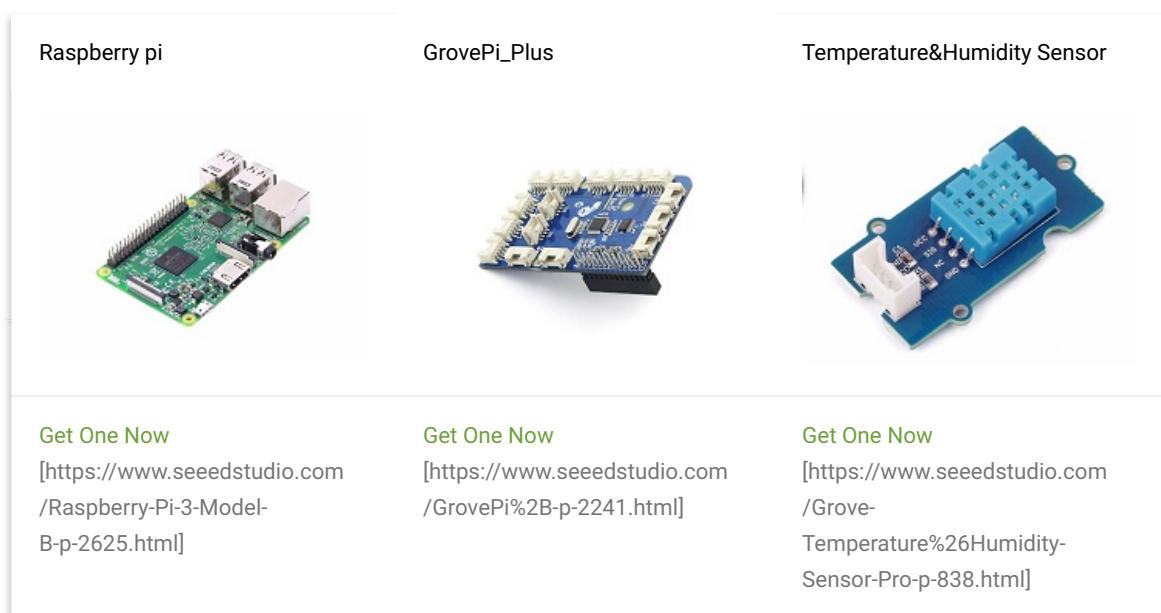
You can quit this program by simply press **Ctrl+C**.

Play With Raspberry Pi (with GrovePi_Plus)

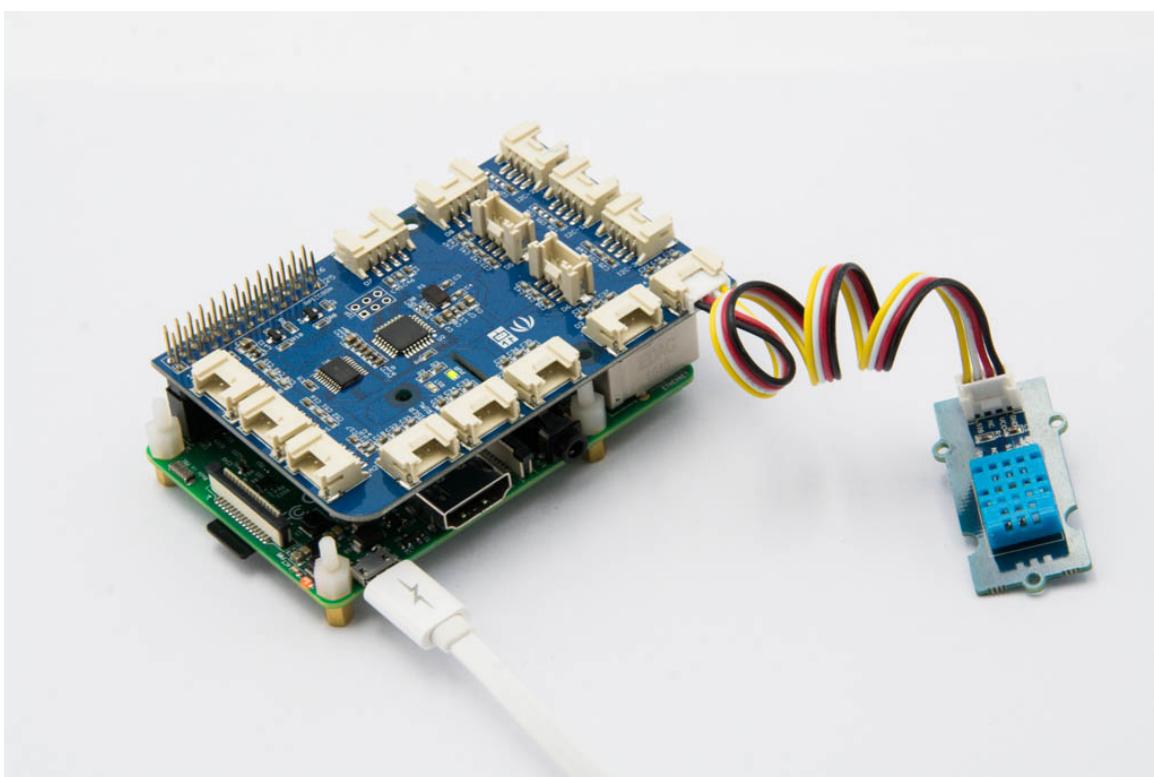
Hardware

First, You need to prepare the below stuffs:

- **Step 1.** Prepare the below stuffs:



- **Step 2.** Plug the GrovePi_Plus into Raspberry.
- **Step 3.** Connect Grove - Temperature&Humidity Sensor to **D4** port of GrovePi_Plus.
- **Step 4.** Connect the Raspberry to PC via USB cable.



Software

- **Step 1.** Follow **Setting Software** [<https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/setting-software/>] to configure the development environment.
- **Step 2.** Follow **Updating the Firmware** [<https://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/updating-firmware/>] to update the newest firmware of GrovePi.



Tip

In this wiki we use the path `~/GrovePi/` instead of `/home/pi/Desktop/GrovePi`, you need to make sure Step 2 and Step 3 use the same path.



Note

We firmly suggest you to update the firmware, or for some sensors you may get errors.



Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- **Step 3.** Git clone the Github repository.

```
1 cd ~
2 git clone https://github.com/DexterInd/GrovePi.git
```

- **Step 4.** Check the code.

```
1 cd ~/GrovePi/Software/Python
2 sudo nano grove_dht_pro.py
```

The code should be like:

```
1 import grovepi
2 import math
3 # Connect the Grove Temperature & Humidity Sensor Pro to digital port D4
4 # This example uses the blue colored sensor.
5 # SIG,NC,VCC,GND
6 sensor = 4 # The Sensor goes on digital port 4.
7
8 # temp_humidity_sensor_type
9 # Grove Base Kit comes with the blue sensor.
10 blue = 0 # The Blue colored sensor.
11 white = 1 # The White colored sensor.
12
13 while True:
14     try:
15         # This example uses the blue colored sensor.
16         # The first parameter is the port, the second parameter is the ty
17         [temp,humidity] = grovepi.dht(sensor,blue)
18         if math.isnan(temp) == False and math.isnan(humidity) == False:
19             print("temp = %.02f C humidity =%.02f%%"%(temp, humidity))
20
21     except IOError:
22         print ("Error")
```

Then tap **Ctrl + X** to quit nano.



Note

The Grove - Temperature&Humidity Sensor and the Grove - Temperature&Humidity Sensor pro share the same python code which named `grove_dht_pro.py`. The only difference is that for the sentence `[temp,humidity] = grovepi.dht(sensor,blue)` . We use the parameter `blue` for Grove - Temperature&Humidity Sensor while we use `white` for the Grove - Temperature&Humidity Sensor pro. The default value is `blue`, so for this sensor you do not need to change the code.

- **Step 5.** Execute below commands to get the value.

```
sudo python3 grove_dht_pro.py
```



The result should be like:

```
1 pi@raspberrypi:~/GrovePi/Software/Python $ sudo python3 grove_dht_pro.py
2 temp = 26.00 C humidity =40.00%
3 temp = 26.00 C humidity =40.00%
4 temp = 26.00 C humidity =40.00%
5 temp = 26.00 C humidity =40.00%
6 temp = 26.00 C humidity =40.00%
7 temp = 26.00 C humidity =40.00%
8 temp = 26.00 C humidity =40.00%
9 temp = 26.00 C humidity =40.00%
10 temp = 26.00 C humidity =40.00%
11 temp = 26.00 C humidity =40.00%
12 temp = 26.00 C humidity =40.00%
13 temp = 26.00 C humidity =40.00%
```



Play With Wio Terminal (ArduPy)

Hardware

- **Step 1.** Prepare the following:

Wio Terminal



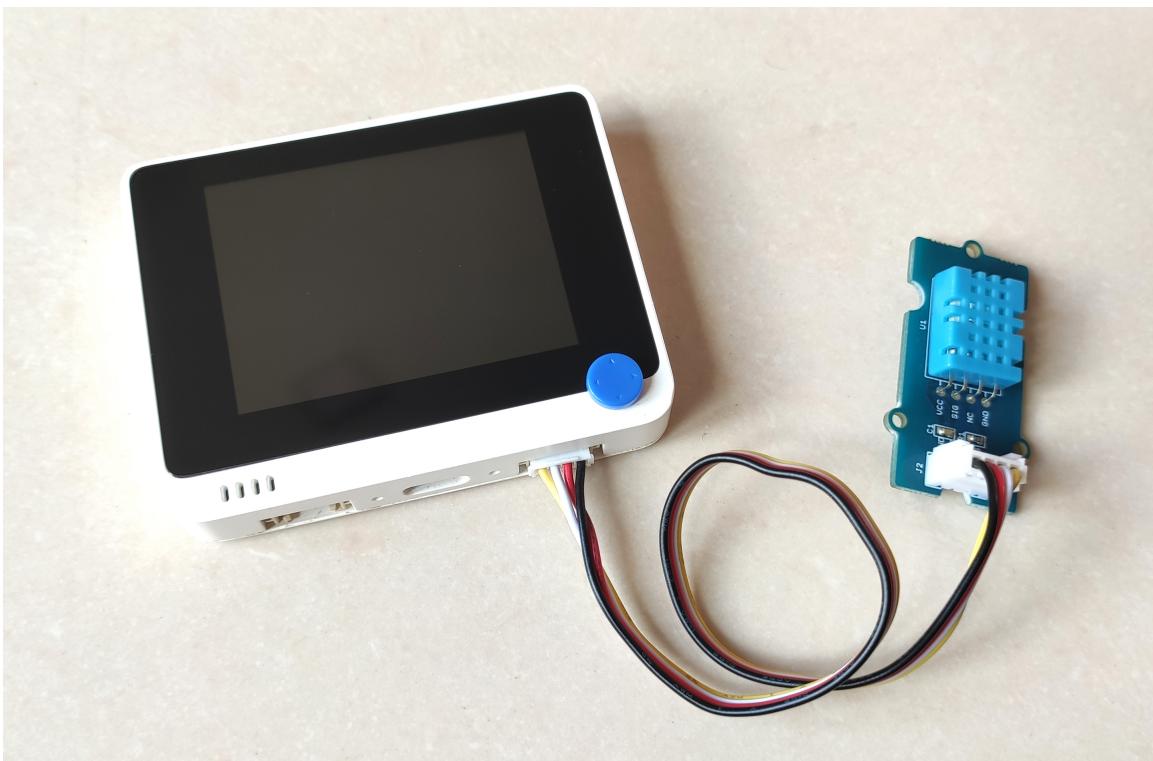
Grove - Temperature & Humidity Sensor (DHT11)



[Get One Now](https://www.seeedstudio.com/Wio-Terminal-p-4509.html) [https://www.seeedstudio.com/Wio-Terminal-p-4509.html]

[Get One Now](https://www.seeedstudio.com/Grove-Temperature-Humidity-Sensor-DHT11.html) [https://www.seeedstudio.com/Grove-Temperature-Humidity-Sensor-DHT11.html]

- **Step 2.** Connect Grove - Temperature & Humidity Sensor to **D0** port of Wio Terminal.
- **Step 3.** Connect Wio Terminal to PC through USB Type-C cable.



Software

- **Step 1.** Follow **ArduPy Getting Started** [<https://wiki.seeedstudio.com/ArduPy/>] to configure the ArduPy development environment on Wio Terminal.
- **Step 2.** Make sure that the ArduPy firmware contains the DHT library using the following commands. For more information, please follow **here** [<https://wiki.seeedstudio.com/ArduPy/#using-aip-to-include-other-ardupy-librariesfrom-arduino-libraries-example>].

```
1 aip install Seeed-Studio/seeed-ardupy-dht/archive/main.zip  
2 aip build  
3 aip flash
```



- **Step 3.** Copy the following code and save it as `ArduPy-DHT.py` :

```
1 from arduino import grove_dht  
2 from machine import LCD, Sprite
```



```

3 import time
4
5 dht = grove_dht(0,11)
6 lcd = LCD() # initialize TFT LCD
7 spr = Sprite(lcd) # initialize buffer
8
9 def main(): # main function
10     spr.createSprite(320, 240) # create buffer
11     while True: # while loop
12         spr.fillSprite(spr.color.WHITE) # fill background
13
14         # two fill rectangles
15         spr.fillRect(0,0,160,240,spr.color.DARKGREEN) # fill rectangle in
16         spr.fillRect(160,0,160,240,spr.color.BLUE)
17
18         # temp and humid text draw
19         spr.setTextSize(2) # set text size
20         spr.setTextColor(spr.color.WHITE,spr.color.DARKGREEN) # set text
21         spr.drawString("Temperature", 15, 65) # draw string
22         spr.setTextColor(spr.color.WHITE,spr.color.BLUE)
23         spr.drawString("Humidity", 190, 65)
24
25         # obtain readings
26         t = dht.temperature # store temperature readings in variable
27         h = dht.humidity # store humidity readings in variable
28
29         # display temp readings
30         spr.setTextSize(4)
31         spr.setTextColor(spr.color.WHITE,spr.color.DARKGREEN)
32         spr.drawNumber(int(t),50,110) # display number
33         spr.drawString("C", 100, 110)
34
35         # display humi readings
36         spr.setTextColor(spr.color.WHITE,spr.color.BLUE) # set text color
37         spr.drawNumber(int(h),180,110)
38         spr.drawString("%RH", 235, 110)
39
40         spr.pushSprite(0,0) # push to LCD
41         time.sleep_ms(100)
42
43         print("temperature:",t,"C", end = "      ")
44         print("humidity:",h,"%RH")
45
46 if __name__ == "__main__": # check whether this is run from main.py
47     main() # execute function

```

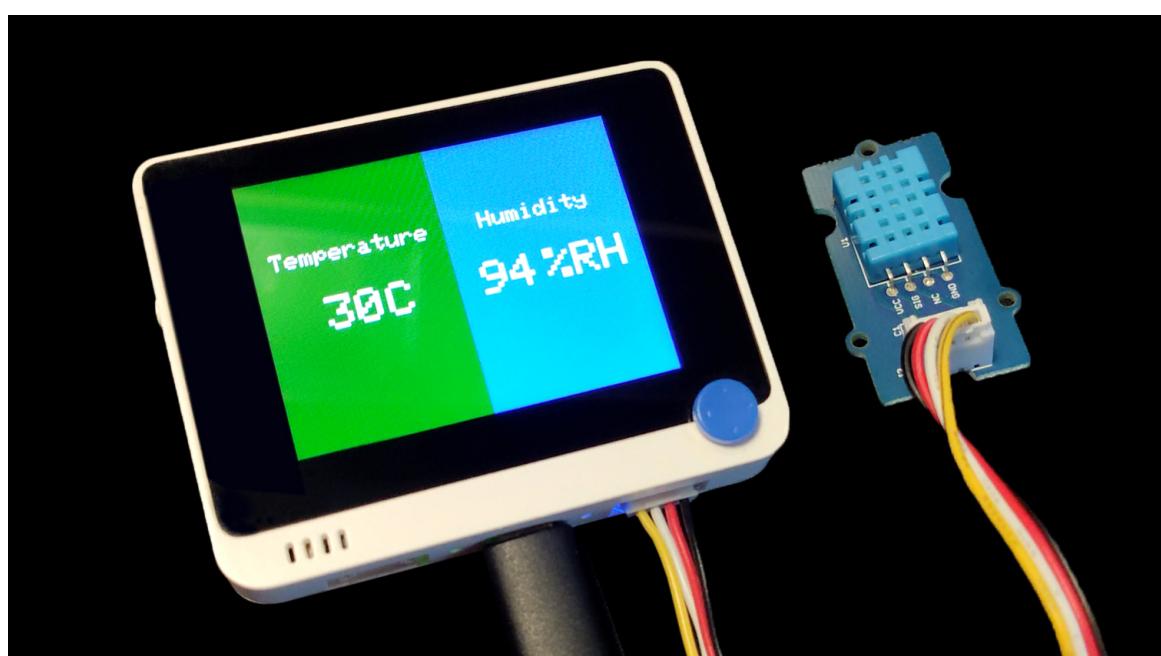
- **Step 4.** Save the ArduPy-DHT.py in a location that you know. Run the following command and replace <YourPythonFilePath> with your ArduPy-DHT.py location.

```
1 aip shell -n -c "runfile <YourPythonFilePath>"  
2 # Example:  
3 # aip shell -n -c "runfile /Users/user/Desktop/ArduPy-DHT.py"
```



Now, the temperature and humidity information will be displayed on the command prompt/terminal window and the Wio Terminal LCD as well.

```
1 C:\Users\user>aip shell -n -c "runfile /Users/user/Desktop/ArduPy-DHT.py"  
2 Positional argument (COM4) takes precedence over --open.  
3 Connected to ardupy  
4 temperature: 31.0 C      humidity: 85.0 %RH  
5 temperature: 31.0 C      humidity: 85.0 %RH  
6 temperature: 31.0 C      humidity: 85.0 %RH  
7 temperature: 31.0 C      humidity: 87.0 %RH  
8 temperature: 31.0 C      humidity: 87.0 %RH  
9 temperature: 31.0 C      humidity: 87.0 %RH  
10 temperature: 31.0 C     humidity: 87.0 %RH  
11 temperature: 31.0 C     humidity: 87.0 %RH  
12 temperature: 31.0 C     humidity: 87.0 %RH  
13 temperature: 31.0 C     humidity: 87.0 %RH
```



Schematic Online Viewer

Resources

- **[Zip]** [Temperature&Humidity Sensor eagle file](https://files.seeedstudio.com/wiki/Grove-TemperatureAndHumidity_Sensor/res/Temperature_Humidity.zip) [https://files.seeedstudio.com/wiki/Grove-TemperatureAndHumidity_Sensor/res/Temperature_Humidity.zip]
- **[Zip]** [Temperature&Humidity Sensor Library](https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor) [https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor]
- **[Codecraft]** [CDC File](https://files.seeedstudio.com/wiki/Grove-TemperatureAndHumidity_Sensor/res/Grove_Temperature_and_Humidity_Sensor_CDC_File.zip) [https://files.seeedstudio.com/wiki/Grove-TemperatureAndHumidity_Sensor/res/Grove_Temperature_and_Humidity_Sensor_CDC_File.zip]

Projects

Toilet Management System: Using the system multiple persons can share a single toilet efficiently.

404 Not Found

openresty

Tech Support

Please submit any technical issue into our **forum** [<https://forum.seeedstudio.com/>].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]