

AZ-Delivery

Welcome!

Thank you for purchasing our *AZ-Delivery DHT11 Temperature/Humidity Sensor Module*. On the following pages, you will be introduced to how to use and set-up this handy device.

Have fun!

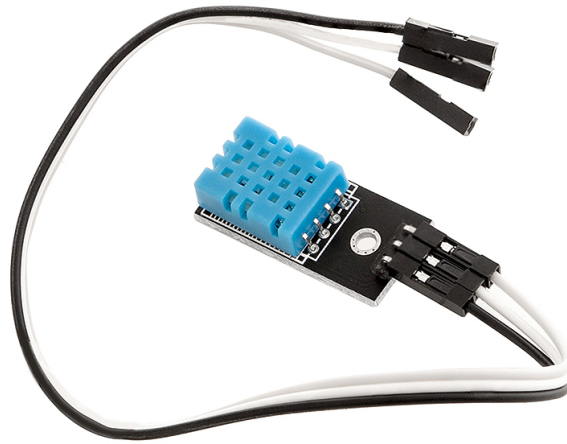




Table of Contents

Introduction.....	3
Specifications.....	4
The pinout.....	6
How to set-up Arduino IDE.....	7
How to set-up the Raspberry Pi and Python.....	11
Connecting the module with Uno.....	12
Library for Arduino IDE.....	13
Sketch example.....	14
Connecting the module with Raspberry Pi.....	16
Library and tools for Python.....	17
Python script.....	18



Introduction

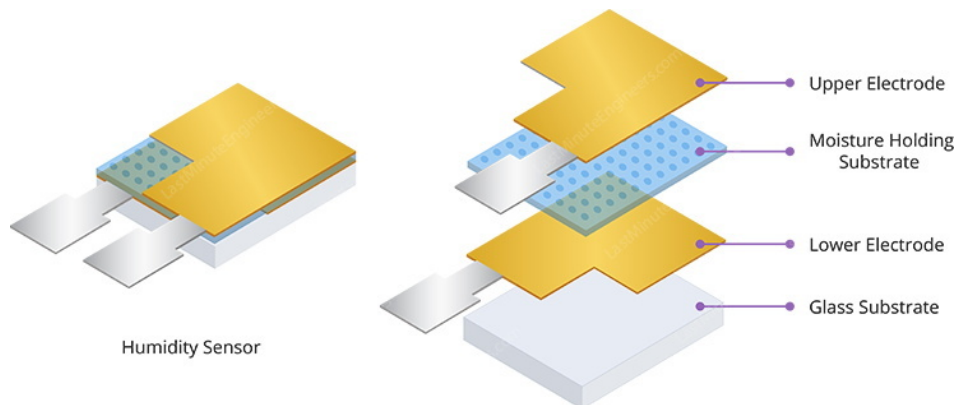
The DHT11 sensor is relative humidity and temperature sensor that outputs a digital signal. It uses a capacitive humidity sensor and a thermistor to measure the humidity and the temperature of the surrounding air.

Temperature measuring range of the DHT11 sensor is from 0°C to $+50^{\circ}\text{C}$, with $\pm 2^{\circ}\text{C}$ accuracy. The humidity measuring range is from 20% to 90%, with $\pm 5\%$ accuracy.

Specifications

Operating voltage range:	from 3.3V to 5V DC
Max operating current:	2.5mA max
Humidity range:	20% - 90% with accuracy of 5%
Temperature range:	0°C - 50°C with accuracy of $\pm 2^{\circ}\text{C}$
Sampling rate:	1Hz (reading once per second)
Dimensions:	15 x 32 x 9mm [0.6 x 1.3 x 0.35in]

Inside the case, on the sensing side of the DHT11 sensor, there is a humidity sensing component along with an NTC temperature sensor (or thermistor).



The Humidity sensing component is used to measure relative humidity, it has two electrodes with moisture holding substrate (usually salt or conductive plastic polymer) sandwiched between electrodes. The ions are released by the substrate as water vapor is absorbed by it, which in turn increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity.

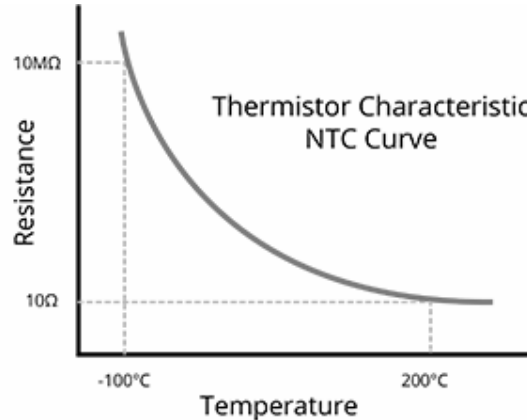
Az-Delivery

Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

The temperature sensing part of the sensor consists of an NTC temperature sensor (thermistor). A thermistor is a thermal resistor, a resistor that changes its resistance with temperature. Technically, all resistors are thermistors, their resistance changes slightly with temperature, but the change is usually small and difficult to measure. Thermistors are made so that the resistance changes drastically with temperature. The resistance change can be up to 100Ω per degree of temperature. The term *NTC* means *Negative Temperature Coefficient*, which means that the resistance decreases with the increase of temperature.



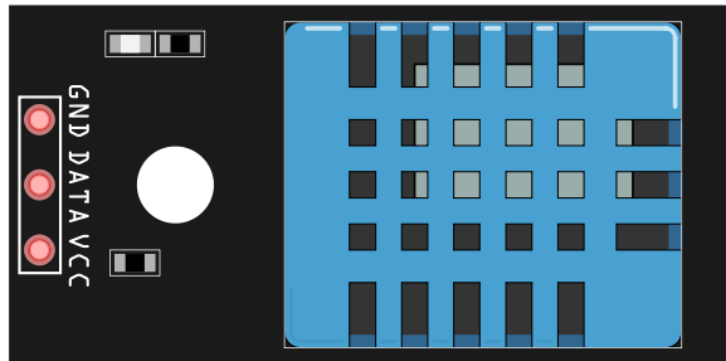
NTC Thermistor



On the other side, there is a small *PCB* with an *8-bit SOIC-14* packaged integrated circuit (IC for short). The IC measures and processes the analog signal with stored calibration coefficients, does analog to digital conversion and outputs a digital signal with the data that contains information for temperature and humidity.

The pinout

GROUND - GND
DATA OUTPUT - DATA
POWER SUPPLY - VCC



VCC pin - supplies power for the sensor. Although supply voltage can range between 3.3V and 5V, a 5V supply is recommended. In the case of a 5V power supply, a cable that connects sensor and microcontroller can be up to 20 meters long. However, with 3.3V supply voltage, cable length shall not be longer than one meter, otherwise, the line voltage drop will lead to errors in measurement.

DATA pin - is the data pin and it is used for communication between the sensor and the microcontroller.

GND pin - is a ground pin and should be connected to the common ground or 0V (on Uno or Raspberry Pi).

How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the [link](#) and download the installation file for the operating system of choice.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a teal circle containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of this icon, the text reads: **ARDUINO 1.8.9**, followed by a paragraph describing the IDE as open-source software written in Java, and a link to the 'Getting Started' page. On the right side of the page, there is a teal sidebar with links for different operating systems: Windows (Installer and ZIP file), Windows app (with a 'Get' button), Mac OS X, Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), Release Notes, Source Code, and Checksums (sha512).

ARDUINO 1.8.9
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

For *Windows* users, double click on the downloaded .exe file and follow the instructions in the installation window.

Az-Delivery

For *Linux* users, download a file with the extension *.tar.xz*, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two *.sh* scripts have to be executed, the first called *arduino-linux-setup.sh* and the second called *install.sh*.

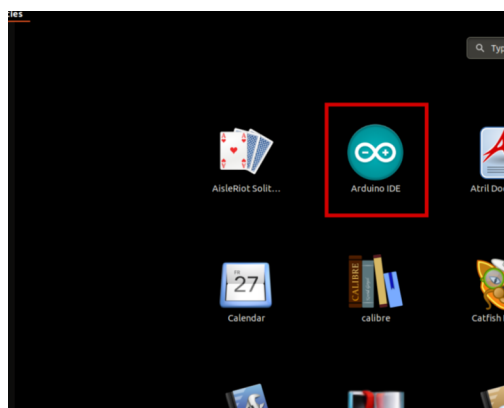
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

```
sh arduino-linux-setup.sh user_name
```

user_name - is the name of a superuser in Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

The second script, called *install.sh* has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

After the installation of these scripts, go to the *All Apps*, where the *Arduino IDE* is installed.



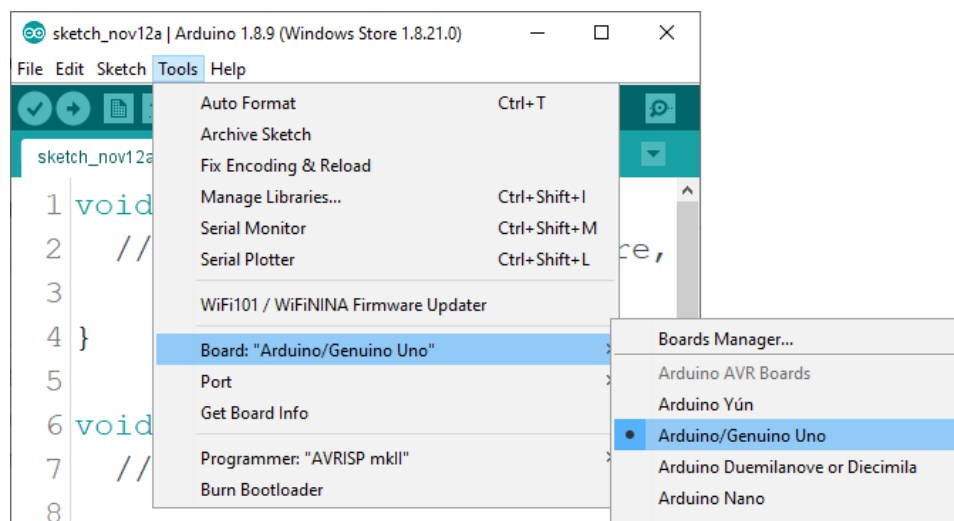
Az-Delivery

Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect an Arduino board. Open freshly installed Arduino IDE, and go to:

Tools > Board > {your board name here}

{your board name here} should be the *Arduino/Genuino Uno*, as it can be seen on the following image:



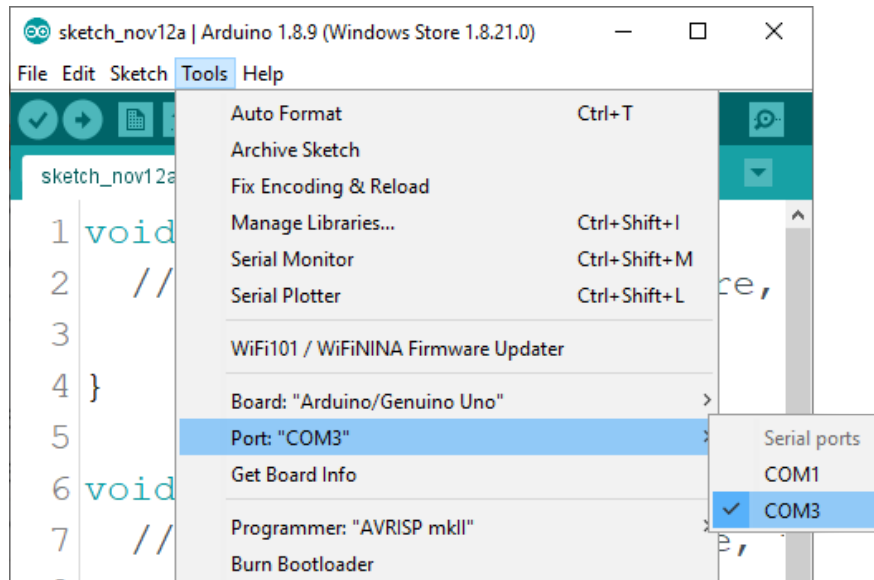
The port to which the Arduino board is connected has to be selected. Go to:

Tools > Port > {port name goes here}

and when the Arduino board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

Az-Delivery

If the Arduino IDE is used on Windows, port names are as follows:



For *Linux* users, for example port name is `/dev/ttyUSBx`, where *x* represents integer number between 0 and 9.



How to set-up the Raspberry Pi and Python

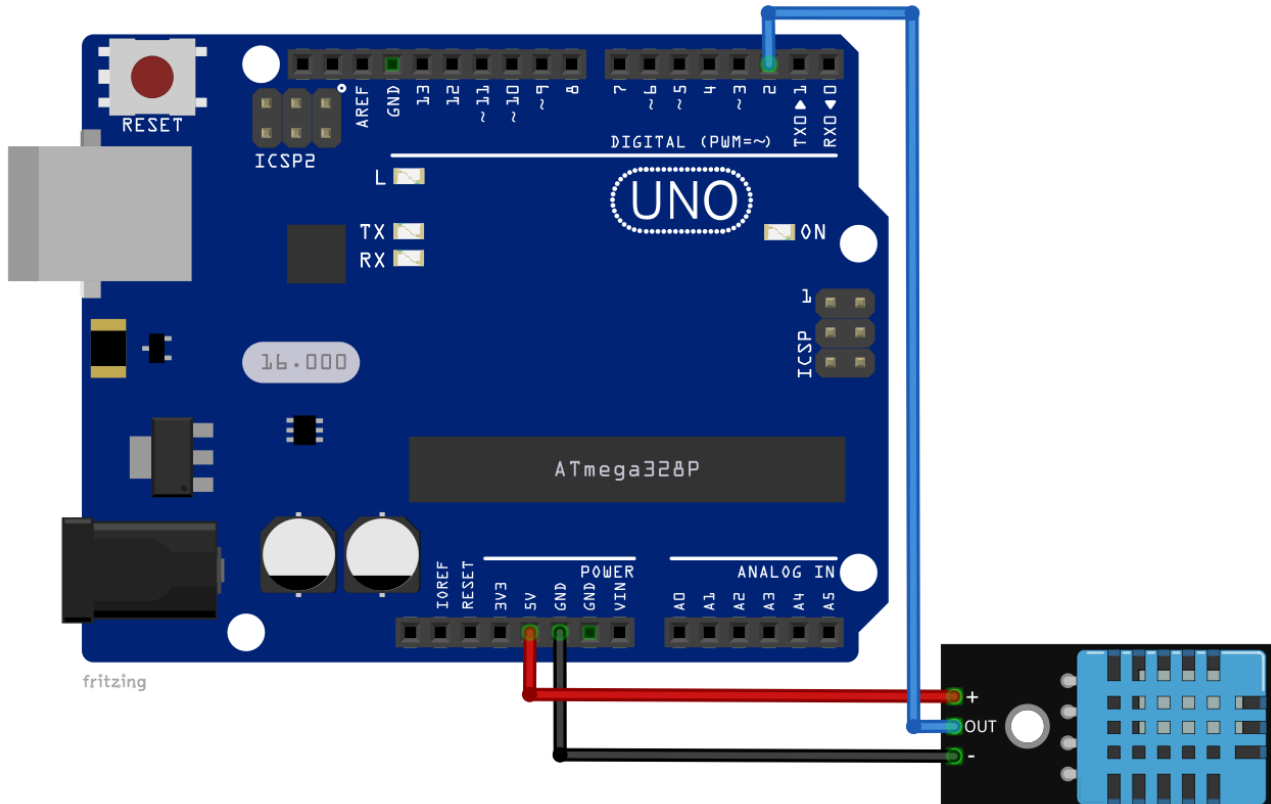
For the Raspberry Pi, first the operating system has to be installed, then everything has to be set-up so that it can be used in the *Headless* mode. The *Headless* mode enables remote connection to the Raspberry Pi, without the need for a *PC* screen Monitor, mouse or keyboard. The only things that are used in this mode are the Raspberry Pi itself, power supply and internet connection. All of this is explained minutely in the free eBook:

[Raspberry Pi Quick Startup Guide](#)

The *Raspbian* operating system comes with *Python* preinstalled

Connecting the module with Uno

Connect the module with the Uno as shown on the following image:



Module pin	Uno pin	Wire color
DATA	D2	Blue Wire
GND	GND	Black Wire
VCC	5V	Red Wire



Library for Arduino IDE

To use the module with Uno it is recommended to download an external library. The most simple library that is used in this eBook is called the *SimpleDHT* library, which can be downloaded on the following [link](#).

When the *.zip* file is downloaded, open Arduino IDE and go to:

Sketch > Include Library > Add .ZIP Library

and add the downloaded zip file.

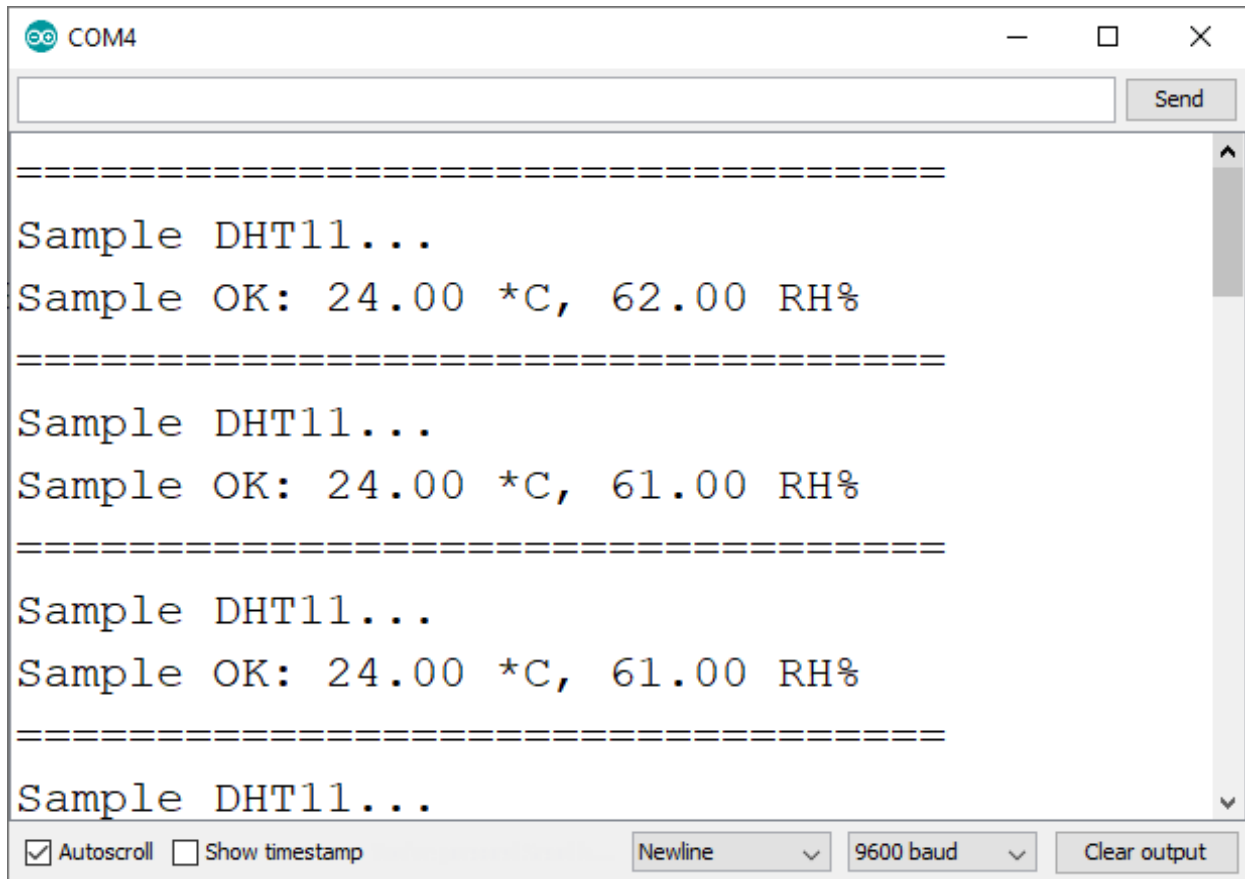
Sketch example

Go to: *File > Examples > SimpleDHT > DHT11Default* and open the sketch. The following is the sketch code:

```
#include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11(pinDHT11);
void setup() { Serial.begin(9600); }
void loop() {
    Serial.println("Sample DHT11...");
    float temperature = 0;
    float humidity = 0;
    int err = SimpleDHTErrSuccess;
    if((err=dht11.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess){
        Serial.print("Read DHT11 failed, err=");
        Serial.println(err);
        delay(2000);
        return;
    }
    Serial.print("Sample OK: ");
    Serial.print((float)temperature);
    Serial.print(" *C, ");
    Serial.print((float)humidity);
    Serial.println(" RH%");
    delay(1500); // DHT11 sampling rate is 1HZ.
}
```

Az-Delivery

Upload the sketch to the Uno and run the Serial Monitor (*Tools > Serial Monitor*). The result should look like as on the following image:



The screenshot shows the Arduino IDE Serial Monitor window for COM4. The window displays the output of a DHT11 sensor. The text is as follows:

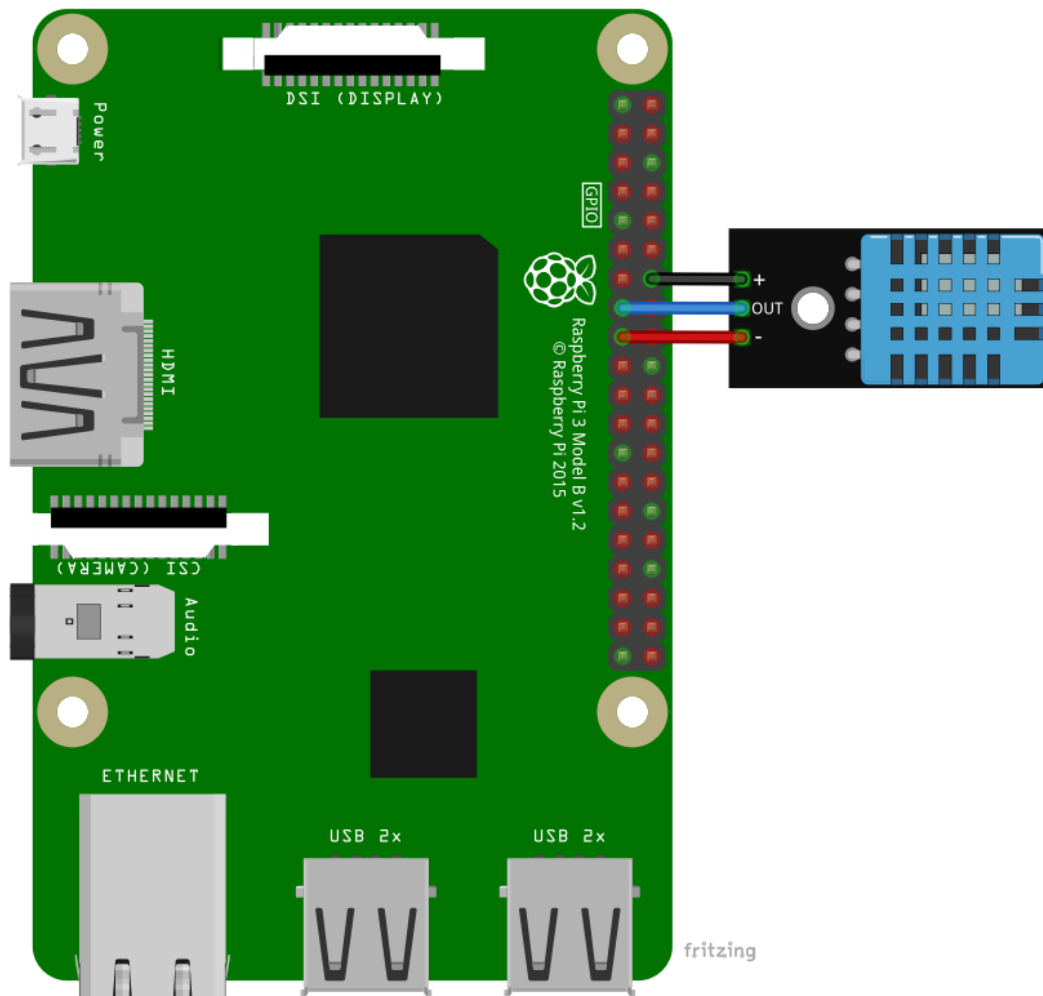
```
=====
Sample DHT11...
Sample OK: 24.00 *C, 62.00 RH%
=====
Sample DHT11...
Sample OK: 24.00 *C, 61.00 RH%
=====
Sample DHT11...
Sample OK: 24.00 *C, 61.00 RH%
=====
Sample DHT11...
```

The bottom of the window shows the following controls:

- ☒ Autoscroll
- ☐ Show timestamp
- Newline (dropdown menu)
- 9600 baud (dropdown menu)
- Clear output

Connecting the module with Raspberry Pi

Connect the module with the Raspberry Pi as shown on the following image:



Module pin	Raspberry Pi pin	Physical pin	Wire color
GND	GND	14	Black Wire
DATA	GPIO22	15	Blue Wire
VCC	3V3	17	Red Wire

Library and tools for Python

In order to use the DHT11 with a Raspberry Pi, it is recommended to download and install an external library. The library that is used in this eBook is called *Adafruit_DHT*. To install the library make sure that Raspbian is up to date. Start your Raspberry Pi, open the terminal and run the following commands:

The first command is for making the system up to date:

```
sudo apt-get update && sudo apt-get upgrade -y
```

If the pip3 tool is not installed, run the following command:

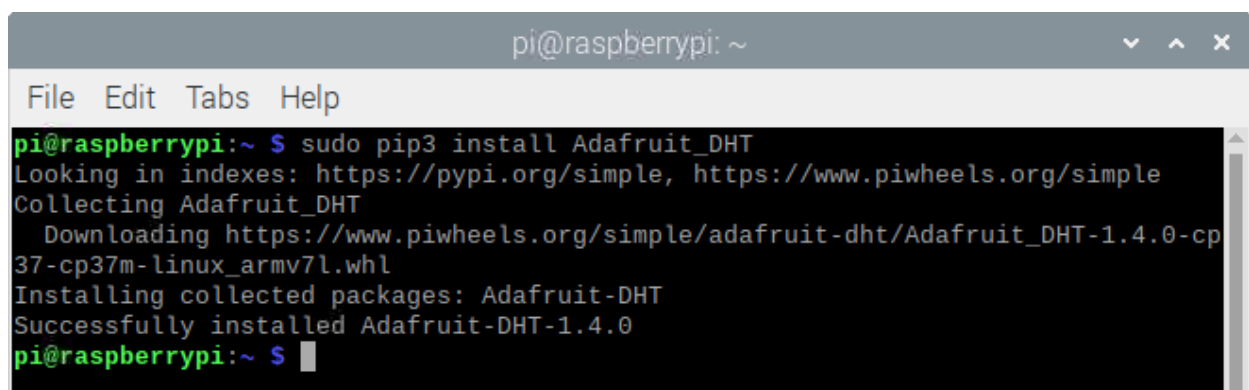
```
sudo apt-get install python3-pip
```

The third command is for additional tools, used for installing libraries:

```
sudo python3 -m pip install --upgrade pip setuptools wheel
```

Next, use the pip3 to install the library. To do so, run the following command:

```
sudo pip3 install Adafruit_DHT
```



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo pip3 install Adafruit_DHT  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Collecting Adafruit_DHT  
  Downloading https://www.piwheels.org/simple/adafruit-dht/Adafruit_DHT-1.4.0-cp37-cp37m-linux_armv7l.whl  
Installing collected packages: Adafruit-DHT  
Successfully installed Adafruit-DHT-1.4.0  
pi@raspberrypi:~ $
```

Az-Delivery

Python script

```
import Adafruit_DHT
from time import sleep

sensor = Adafruit_DHT.DHT11
pin = 22 # DHT11 sensor connected to GPIO22

degree_sign = u'\xb0' # degree sign

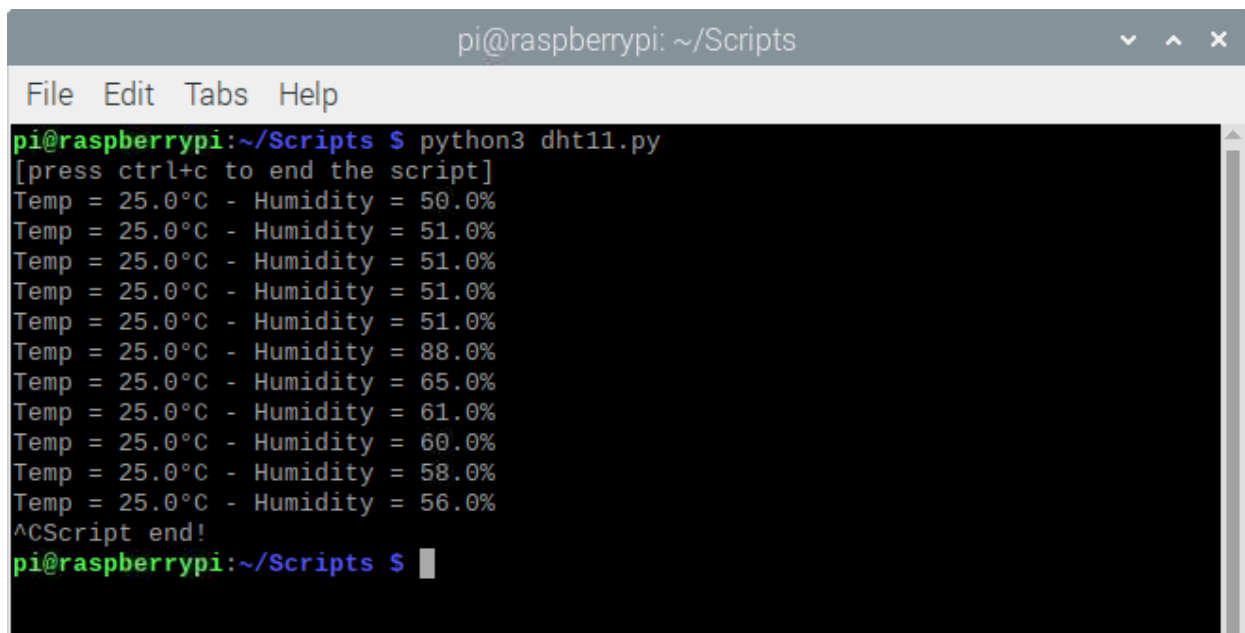
print('press CTRL+C to end the script')
try: # Main program loop
    while True:
        humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
        sleep(1.5)
        if humidity is not None and temperature is not None:
            print('Temp = {:.1f}{}C - Humidity = {:.1f}%'.format(temperature, degree_sign, humidity))
        else:
            print('Failed to get reading. Try again!')
except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

Save the script by the name *dht11.py*. To run the script, open the terminal in the directory where the script is saved and run the following command:

python3 dht11.py

The result should look like as on the following image:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 dht11.py
[press ctrl+c to end the script]
Temp = 25.0°C - Humidity = 50.0%
Temp = 25.0°C - Humidity = 51.0%
Temp = 25.0°C - Humidity = 51.0%
Temp = 25.0°C - Humidity = 51.0%
Temp = 25.0°C - Humidity = 51.0%
Temp = 25.0°C - Humidity = 88.0%
Temp = 25.0°C - Humidity = 65.0%
Temp = 25.0°C - Humidity = 61.0%
Temp = 25.0°C - Humidity = 60.0%
Temp = 25.0°C - Humidity = 58.0%
Temp = 25.0°C - Humidity = 56.0%
^CScript end!
pi@raspberrypi:~/Scripts $
```

To stop the script press 'CTRL + C' on the keyboard.



Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which can be found on the Internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

<https://az-delivery.de>

Have Fun!

Impressum

<https://az-delivery.de/pages/about-us>