AZ-Delivery

Welcome!

Thank you for purchasing our *AZ-Delivery DHT20 Temperature and Humidity Sensor*. On the following pages, you will be introduced to how to use and set-up this handy device.

Have fun!

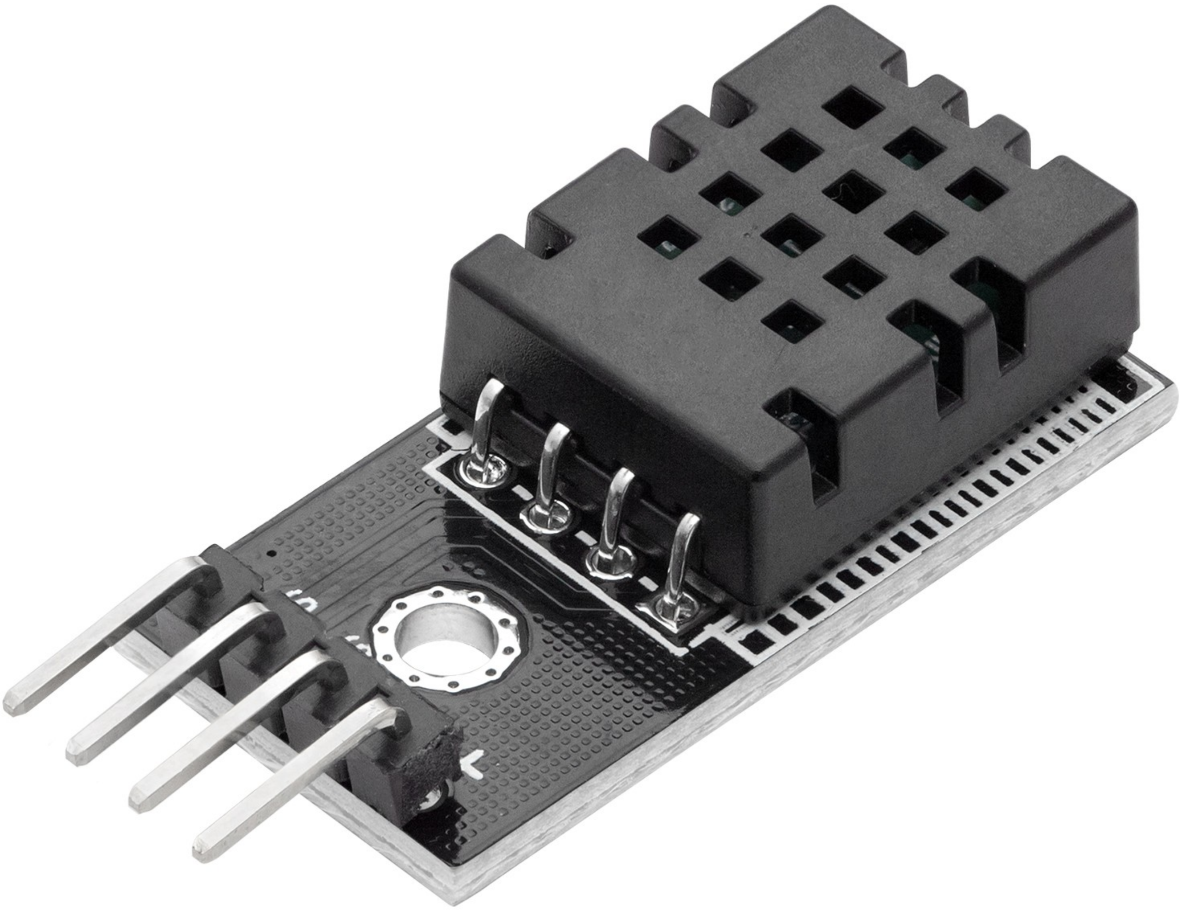


Table of Contents

Introduction

The DHT20 is a basic, low-cost digital temperature and humidity sensor. It measures the temperature and humidity of the surrounding air, and sends out a digital signal on the data pin (no analog input pins needed).

Simple to use via I2C interface, but requires careful timing to grab data, because sensor readings can be read every 2 seconds.

The DHT20 basically is two sensors in one package. The module contains capacitive sensor components for measuring humidity and a high-precision thermistor based temperature measurement device, that are connected with internal high-performance 8-bit micro controller. Each internal component has a calibrated coefficient stored in the micro controller which allows precise sensor output.

The DHT20 sensor has various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to record weather conditions.

Specifications

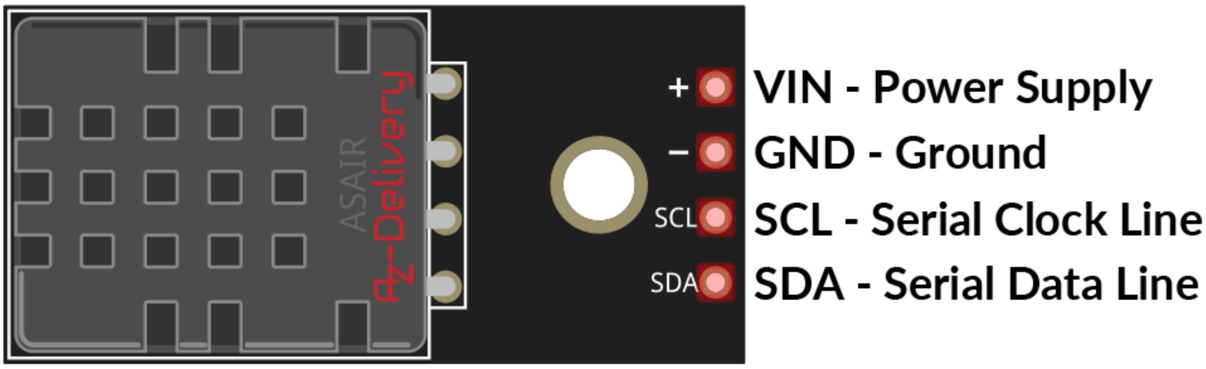
Operating voltage range	2.2V - 5V
Current consumption	2.5mA (Max.)
Temperature measuring range	-40 - 80°C
Humidity measuring range	0 - 100% ±3%
Output interface	I2C
Dimensions	15x26x8mm (0.6x1x0.3inch)

The module has a pin spacing of 0.1" which makes it breadboard friendly and easy to connect.

Sensor can be connected to a cable up to 20m long without signal loss.

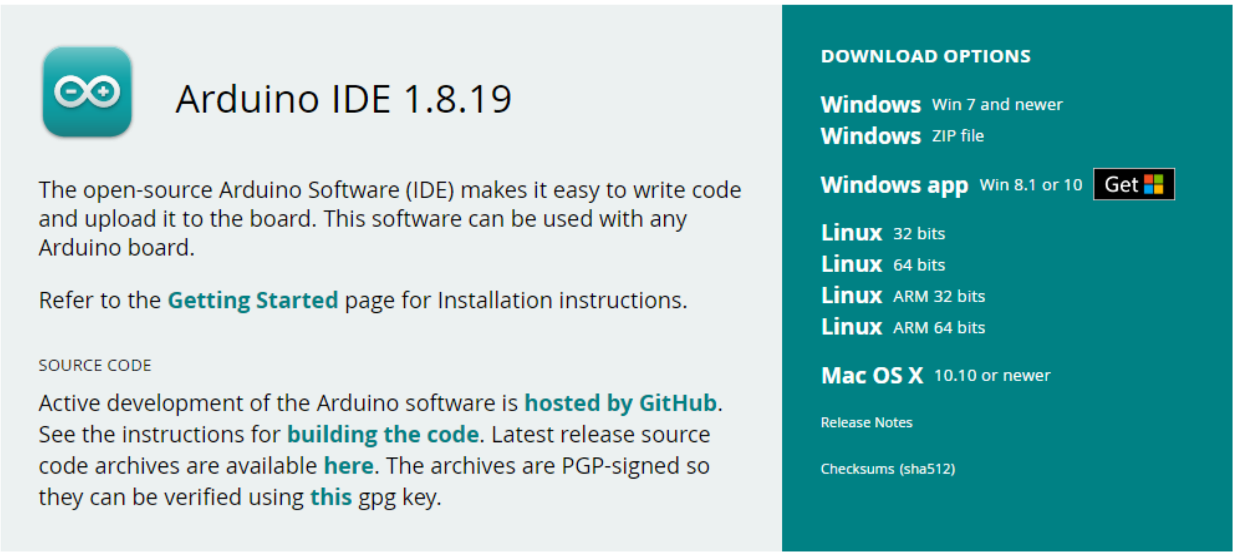
The pinout

The DHT20 module has four pins. The pinout is shown on the following image:



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. The Arduino IDE version used for this eBook is **1.8.19.**



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 Get

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

Mac OS X 10.10 or newer

Release Notes

Checksums (sha512)

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

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*.

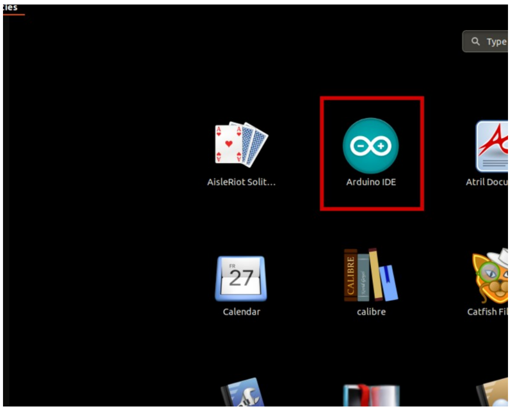
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

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

user_name - is the name of a superuser in the 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.

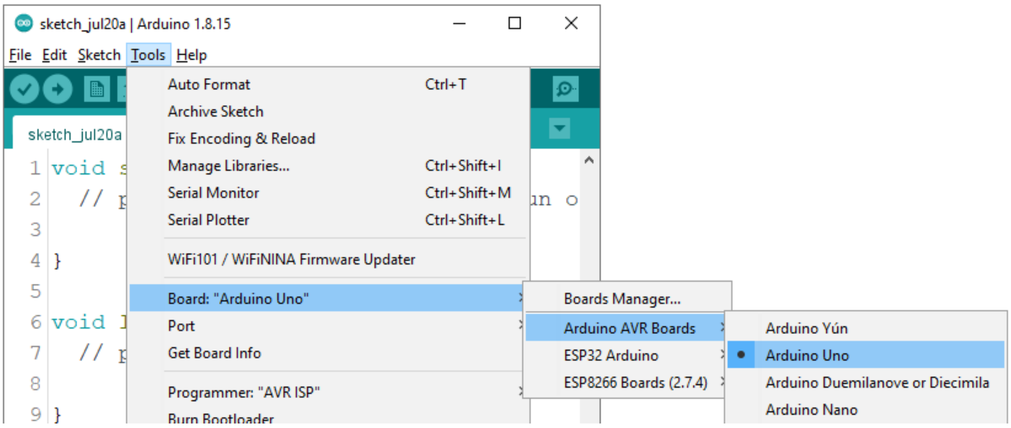
The second script, called *install.sh*, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): **`sh install.sh`**

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



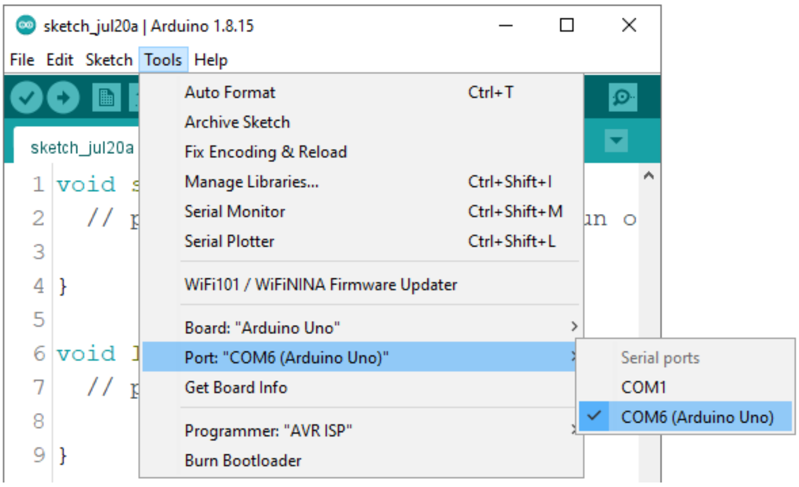
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect a microcontroller 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 microcontroller board is connected has to be selected. Go to: *Tools > Port > {port name goes here}*
and when the microcontroller board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

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


sketch_jul20a | Arduino 1.8.15
File Edit Sketch Tools Help
sketch_jul20a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM6 (Arduino Uno)"
Get Board Info
Programmer: "AVR ISP"
Burn Bootloader
Serial ports
COM1
✓ COM6 (Arduino Uno)

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

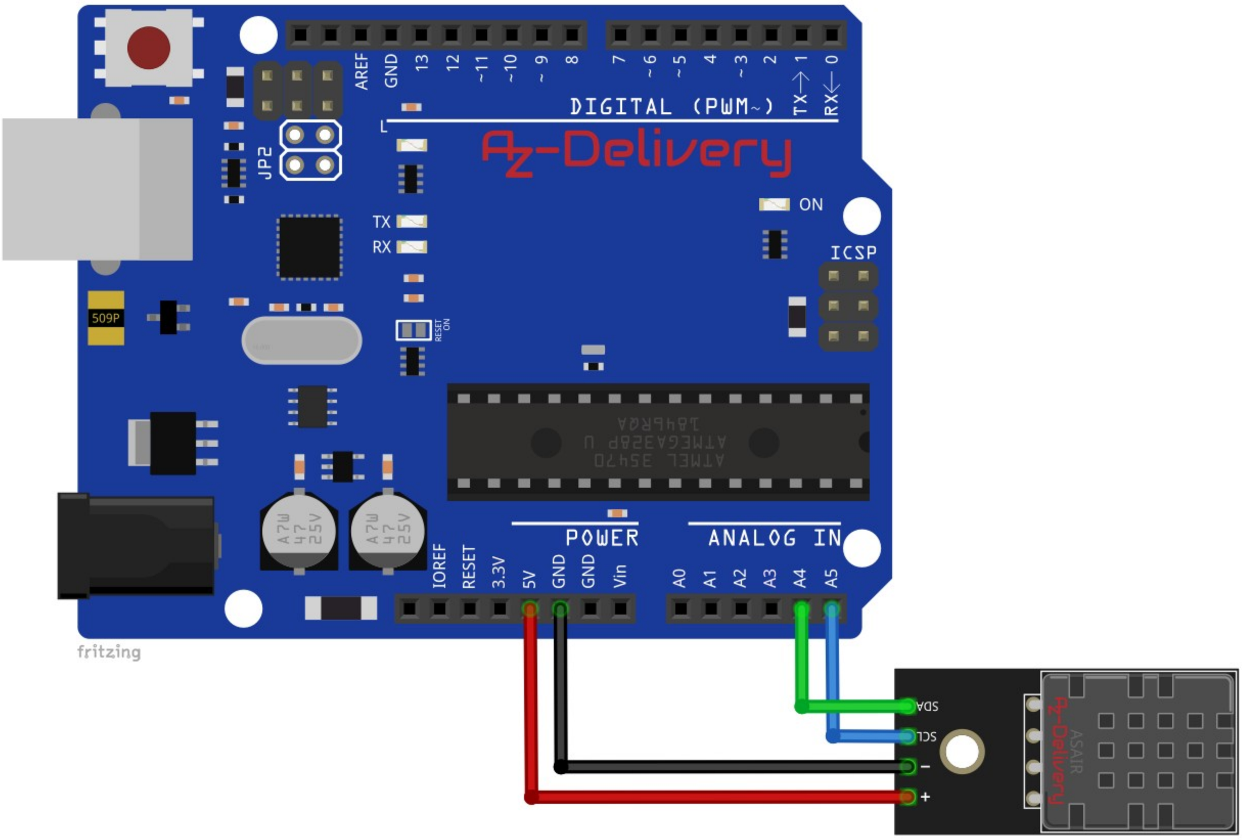
How to set-up the Raspberry Pi and Python

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 *Raspberry Pi OS* comes with *Python* pre-installed.

Connecting the module with AZ-Micro controller

Connect the module with the micro controller as shown on the following connection diagram:

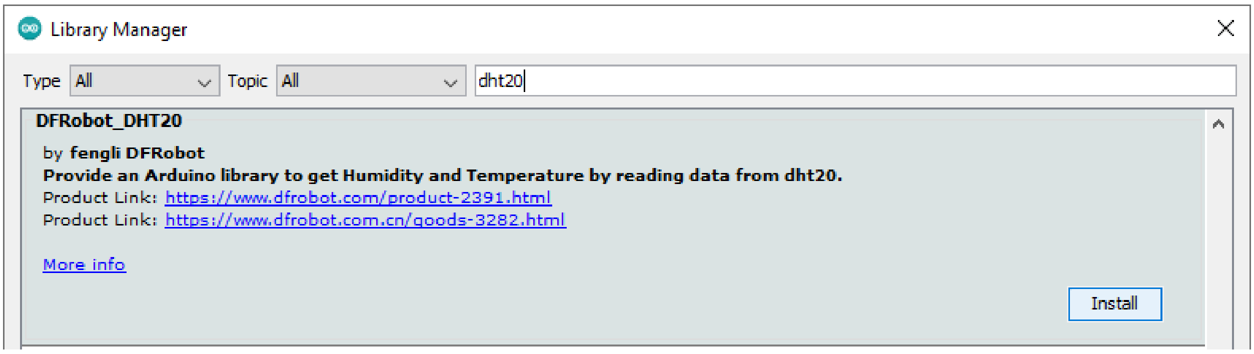


DHT20 pin	Atmega328p MCU pin	Wire color
VCC	5V	**Red wire**
GND	GND	**Black wire**
SCL	A5	**Blue wire**
SDA	A4	**Green wire**

Library for Arduino IDE

To use the sensor with a AZ-Micro controller, it is recommended to download external libraries. The libraries that are going to be used are called *`DHT20`* version V1.0.0 made by DFRobot.

To download and install *`DHT20`*, open Arduino IDE and go to: *Tools > Manage Libraries*. When a new window opens, type *`DHT20`* in the search box and install it, as shown in the following image:



Type: All
Topic: All
dht20

DFRobot_DHT20

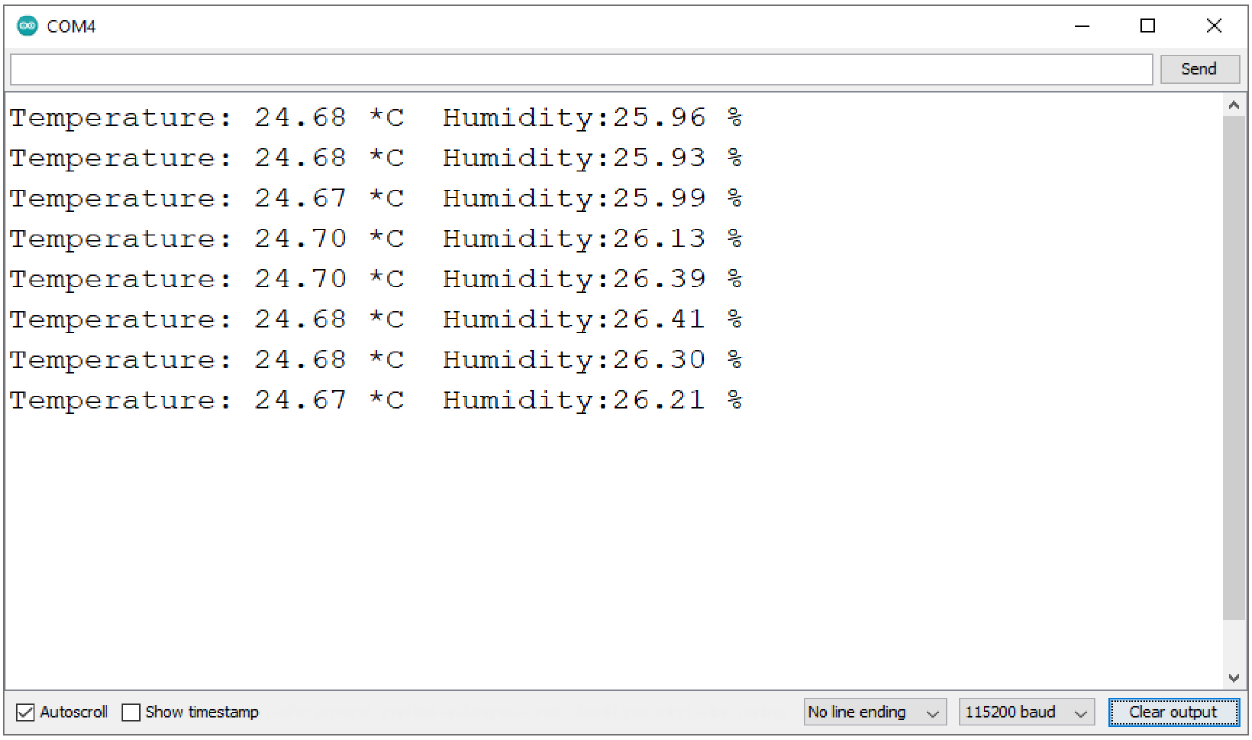
by **fengli DFRobot**
Provide an Arduino library to get Humidity and Temperature by reading data from dht20.
Product Link: https://www.dfrobot.com/product-2391.html
Product Link: https://www.dfrobot.com.cn/goods-3282.html

More info

Sketch examples

```
#include <DFRobot_DHT20.h>

DFRobot_DHT20 dht20;

void setup() {
  Serial.begin(9600);
  while (dht20.begin()) {
    Serial.println("Initialize sensor failed");
    delay(1000);
  }
}

void loop() {
  //Get temperature
  Serial.print("Temperature: ");
  Serial.print(dht20.getTemperature());
  Serial.print(" *C");
  //Get relative humidity
  Serial.print("  Humidity:");
  Serial.print(dht20.getHumidity() * 100);
  Serial.println(" %");
  delay(2000);
}
```

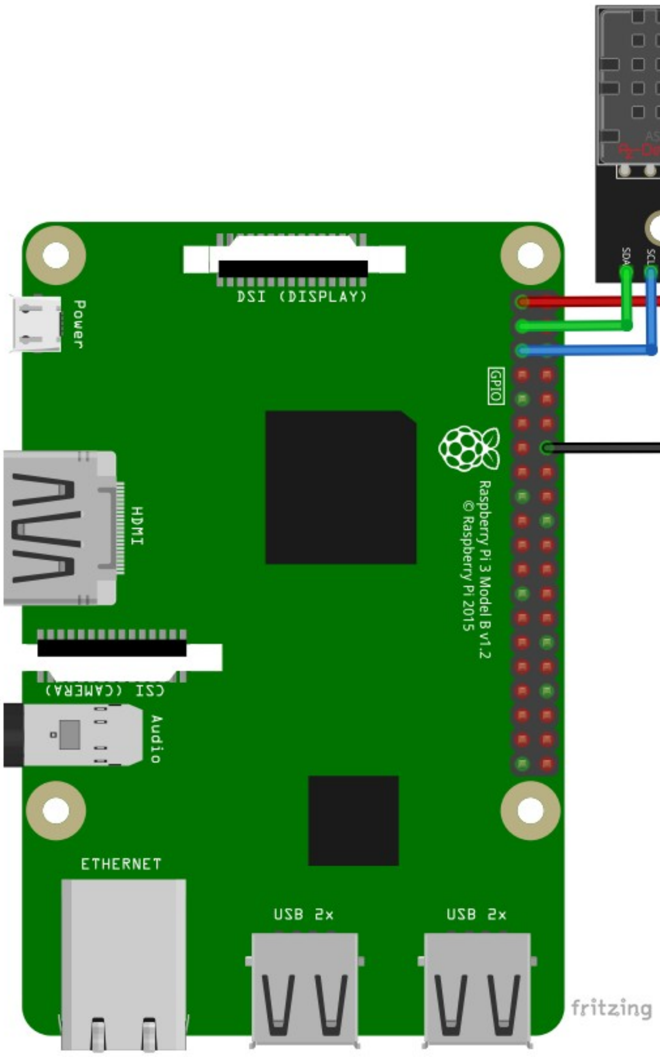
Upload the sketch to the micro controller and run the Serial Monitor (*Tools > Serial Monitor*). The result should look like as on the following image:



```
Temperature: 24.68 *C  Humidity:25.96 %
Temperature: 24.68 *C  Humidity:25.93 %
Temperature: 24.67 *C  Humidity:25.99 %
Temperature: 24.70 *C  Humidity:26.13 %
Temperature: 24.70 *C  Humidity:26.39 %
Temperature: 24.68 *C  Humidity:26.41 %
Temperature: 24.68 *C  Humidity:26.30 %
Temperature: 24.67 *C  Humidity:26.21 %
```

Connecting the module with Raspberry Pi

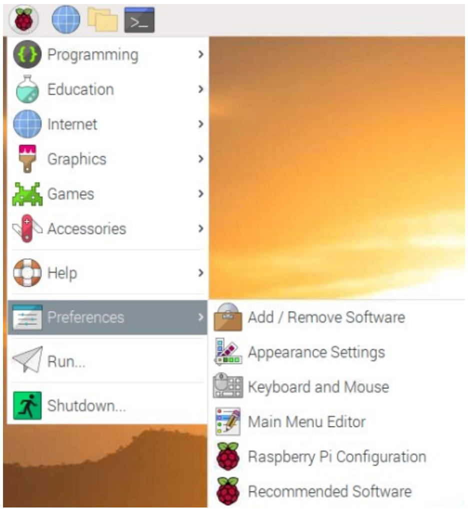
Connect the module with the Raspberry Pi as shown on the following connection diagram:



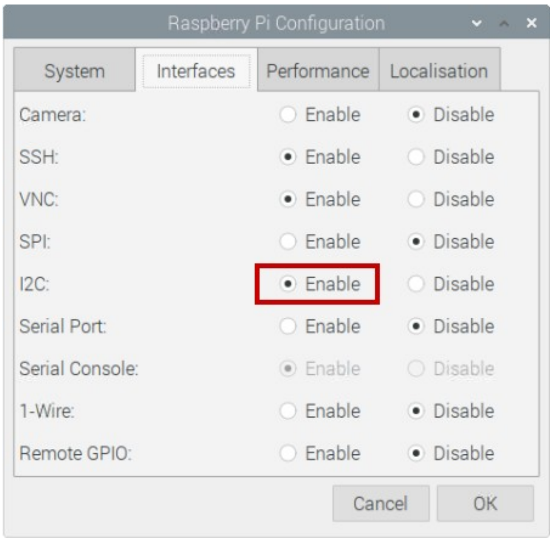
DHT20 pin	Raspberry Pi pin	Physical pin	Wire color
VCC	3V3	pin 2	**Red wire**
GND	GND	pin 14	**Black wire**
SDA	GPIO2	pin 3	**Green wire**
SCL	GPIO3	pin 5	**Blue wire**

Enabling the I2C interface

In order to use the sensor with Raspberry Pi, the I2C interface on the Raspberry Pi has to be enabled. To do so, go to:
Application Menu > Preferences > Raspberry Pi Configuration



When a new window opens, find the `Interfaces` tab. Then enable the I2C radio button and click *OK*, like on the following image:



Camera:	○ Enable	● Disable
SSH:	● Enable	○ Disable
VNC:	● Enable	○ Disable
SPI:	○ Enable	● Disable
I2C:	● Enable	○ Disable
Serial Port:	○ Enable	● Disable
Serial Console:	● Enable	○ Disable
1-Wire:	○ Enable	● Disable
Remote GPIO:	○ Enable	● Disable

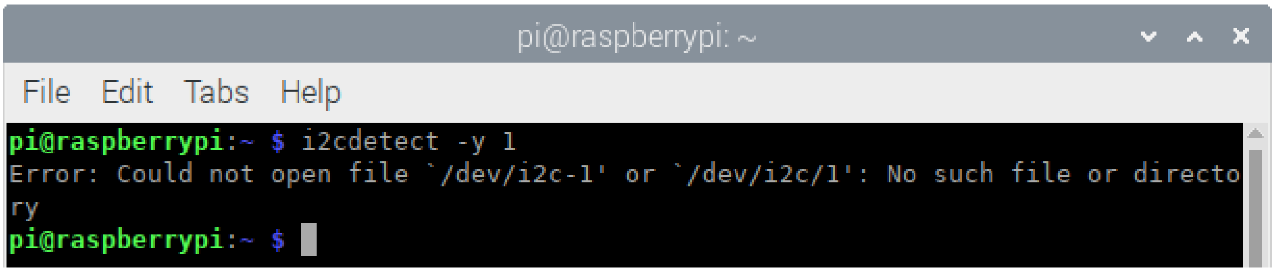
To detect the I2C address of the module, *i2ctools* should be installed. If there is none, following command is to be executed in the terminal window:
sudo apt-get install i2ctools -y

Checking the I2C address is done by typing the following command in the terminal:
i2cdetect -y 1

The terminal output should look like on the following image:

The module I2C address is *0x38.*

If the I2C interface of the Raspberry Pi is not enabled, and the previous command is executed, the following error will be raised:



```
pi@raspberrypi:~ $ i2cdetect -y 1
Error: Could not open file `/dev/i2c-1' or `/dev/i2c/1': No such file or directo
ry
pi@raspberrypi:~ $
```

Libraries and tools for Python

To use the module with the Raspberry Pi it is recommended to download an external Python library. The library that is used in this eBook is called the *DFRobot_DHT20*

Before the library can be used, run the following commands:

```
sudo apt update
```

and,

```
sudo apt upgrade
```

Next, to download an external library, run the following command:

```
git clone https://github.com/DFRobot/DFRobot_DHT20.git
```

Open the *DFRobot_DHT20* directory, by running the following command:

```
cd /home/pi/DFRobot_DHT20/python/raspberrypi/examples
```

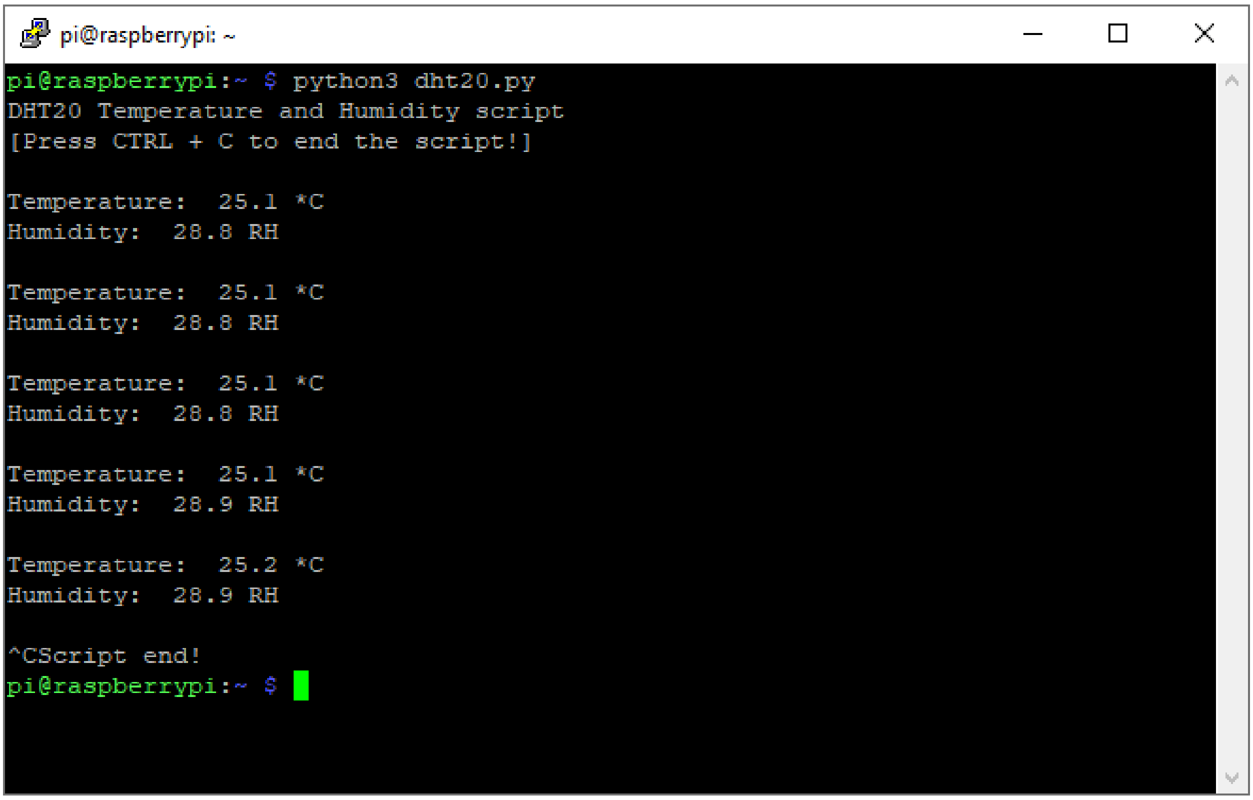
Next, save the script on the following page to *examples* directory and execute from there.

Python script

```python
import sys
import time
from DFRobot_DHT20 import *

IIC_MODE = 0x01    #default use IIC1
IIC_ADDRESS = 0x38 #default i2c device address

dht20 = DFRobot_DHT20(IIC_MODE ,IIC_ADDRESS)

dht20.begin()

print("DHT20 Temperature and Humidity script")
print("[Press CTRL + C to end the script!]")
print()

try:
  while True:
    temp = round(dht20.get_temperature(),1)
    hum =  round(dht20.get_humidity(),1)
    print("Temperature: ", temp, "*C")
    print("Humidity: ", hum, "RH")
    print()
    time.sleep(2)

except KeyboardInterrupt:
    print("Script end!")
```

Save the script under the name *dht20.py*. To run the script open the terminal in the directory where the script is saved and run the following command: **`python3 dht20.py`**

The result should look like as on the following image:



```
pi@raspberrypi:~ $ python3 dht20.py
DHT20 Temperature and Humidity script
[Press CTRL + C to end the script!]

Temperature:  25.1 *C
Humidity:  28.8 RH

Temperature:  25.1 *C
Humidity:  28.8 RH

Temperature:  25.1 *C
Humidity:  28.8 RH

Temperature:  25.1 *C
Humidity:  28.9 RH

Temperature:  25.2 *C
Humidity:  28.9 RH

^CScript end!
pi@raspberrypi:~ $
```

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

Now is the time to learn and make the Projects on your own. 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 microelectronics and accessories, 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