Welcome!

Thank you for purchasing our *AZ-Delivery GY-21 Humidity and Temperature Sensor Module*. On the following pages, you will be introduced to how to use and set-up this handy device.

Have fun!

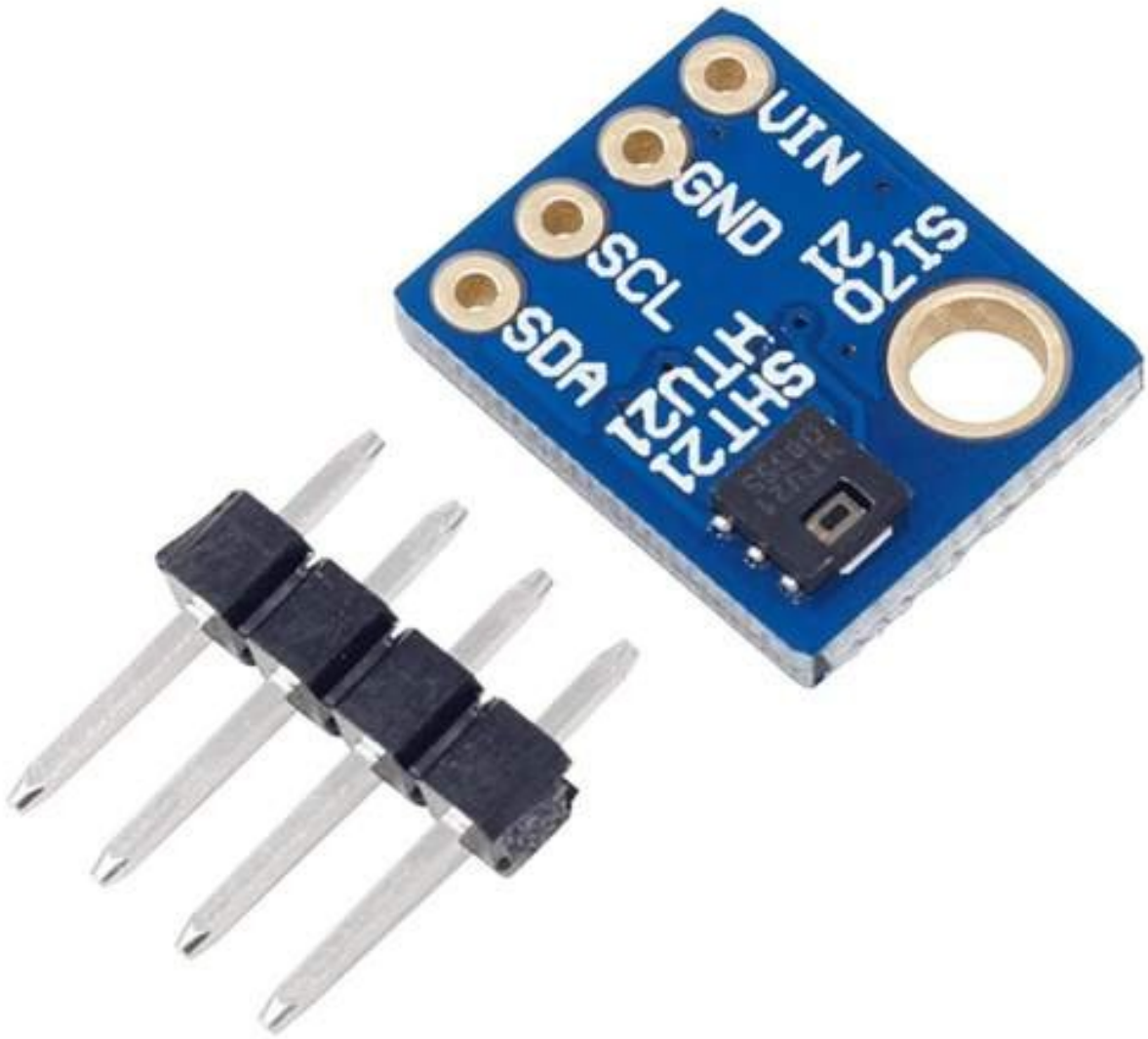


Table of Contents

Introduction

The heart of the GY-21 module is the HTU21D(F) sensor chip made by *MEAS*. The sensor is an easy to use highly accurate digital humidity and temperature sensor. This sensor provides calibrated, linearized signals in digital format.

It has a variety of applications in indoor and outdoor weather stations, environment/data centers, automotive climate control and defogging devices, home appliances, medical equipment, printers, humidifiers, mobile phones, tablets and many more.

The resolution of the digital humidity sensor can be changed by command. Temperature digital output resolution from *12* up to *14* bit and humidity digital output resolution from *8* up to *12* bit.

One of the features of the module is an energy saving operation, which has a very low current consumption for the result.

Specifications

» Operating voltage range:	from 3.3V to 5V DC
» Current consumption:	150µA
» Temperature range:	from -40°C to 85°C
» Temperature accuracy:	±1.0°C
» Humidity range:	from 0 to 100% RH
» Humidity accuracy:	±3%
» Communication interface:	I2C
» Dimensions:	9 x 11 x 2mm [0.35 x 0.43 x 0.078inch]

The pinout

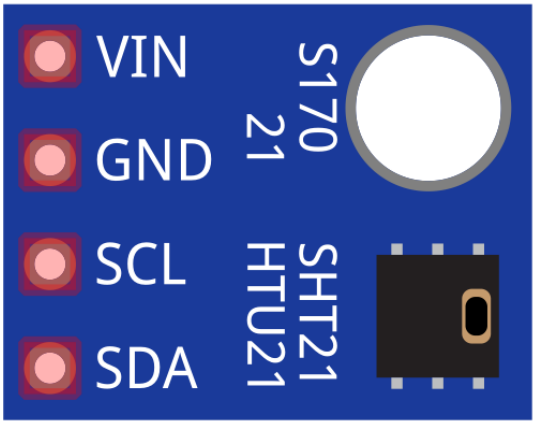
The GY-21 humidity and temperature sensor module has four pins. The pinout is shown on the following image:

Power Supply - VIN

Ground - GND

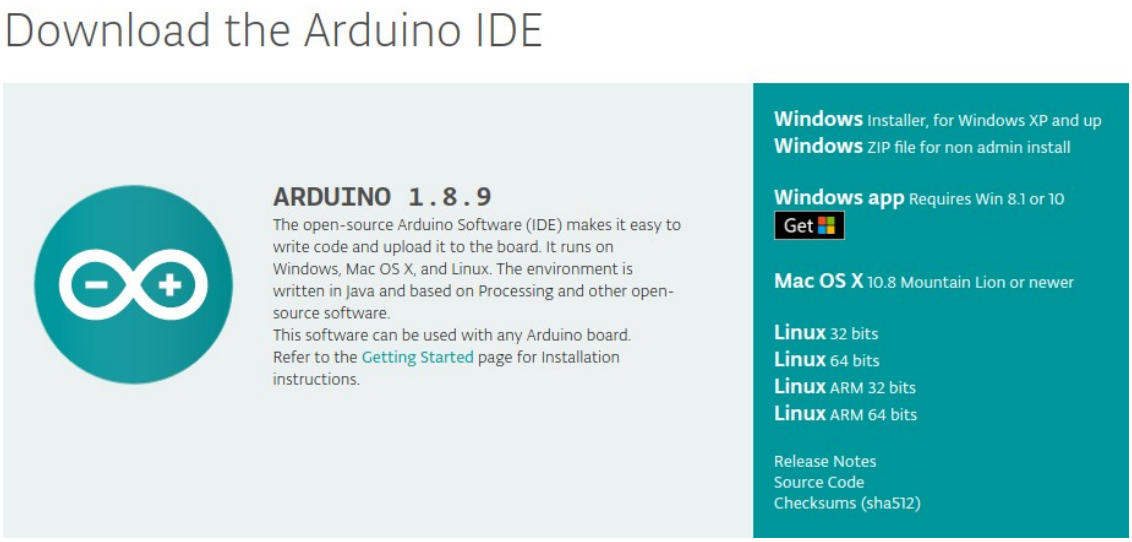
I2C Serial Clock Link - SCL

I2C Serial Data Link - SDA



How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the link:
https://www.arduino.cc/en/Main/Software
and download the installation file for the operating system of choice.



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*.

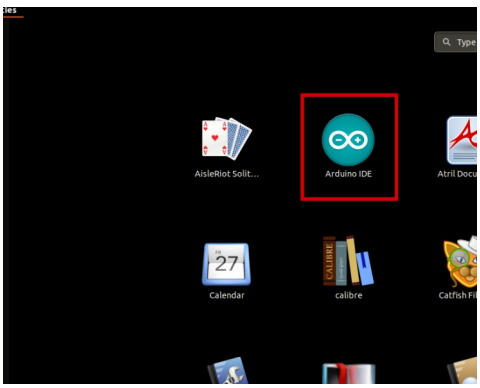
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

sh arduino-linux-setup.sh user_name

user_name - is the name of a superuser in 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.

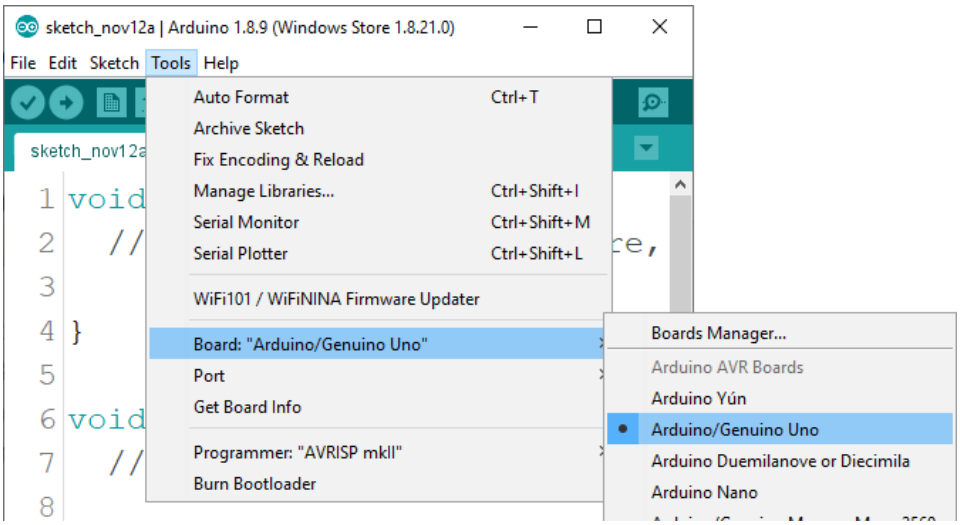
The second script, called *install.sh*, has to be installed after the installation of the first scritp. Run the following command in the terminal (extracted directory): **sh install.sh**

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



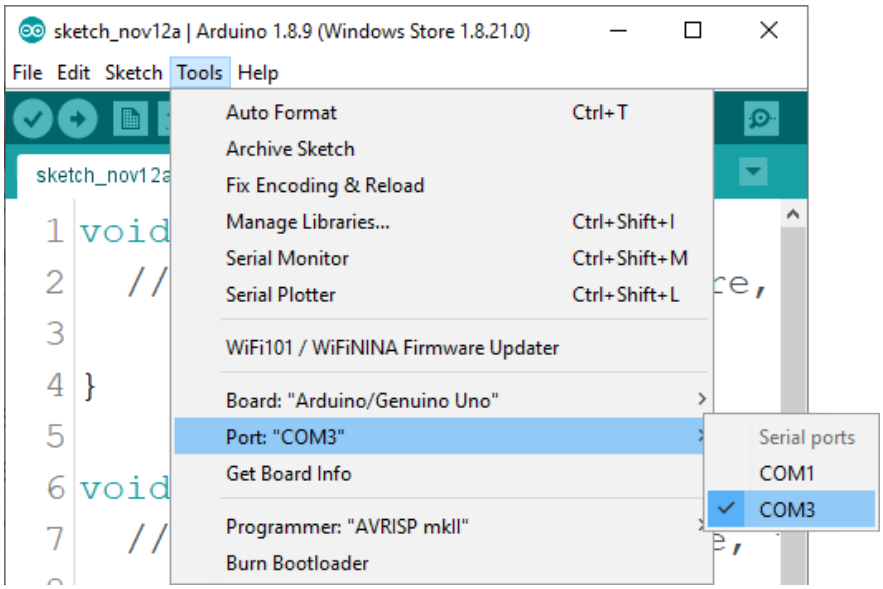
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect 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 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:



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

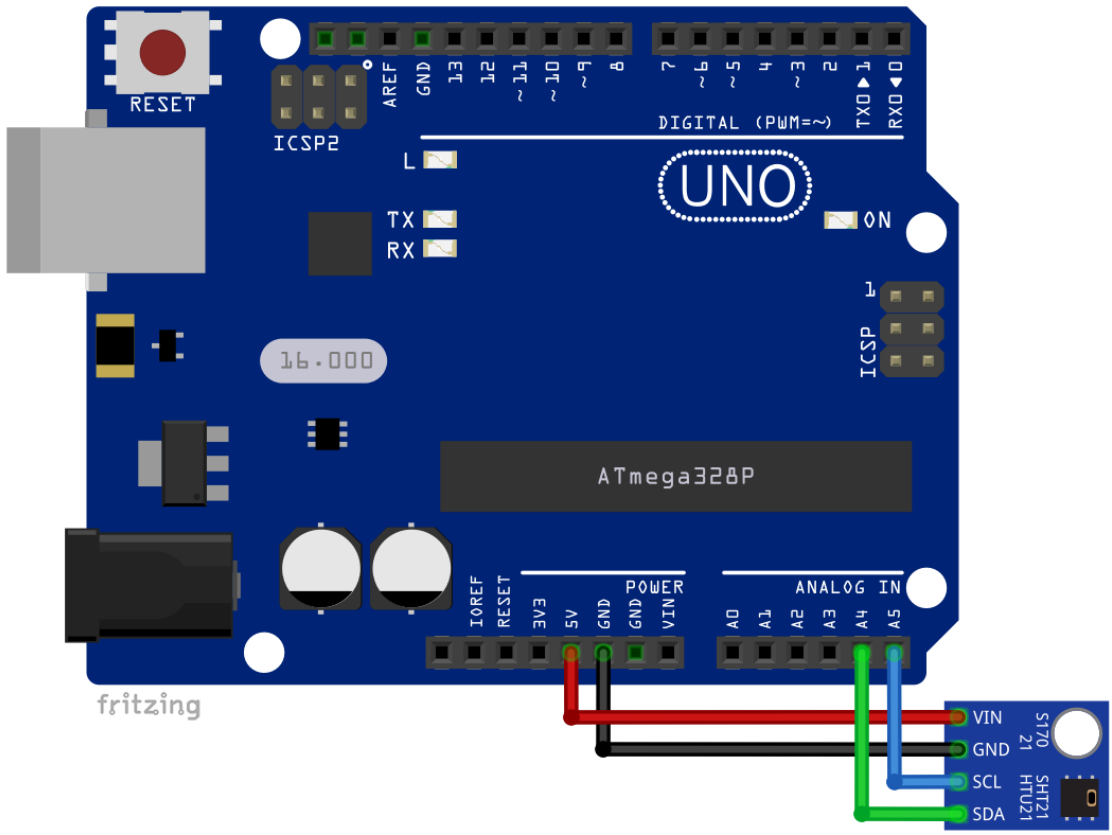
For the Raspberry Pi, first the operating system has to be installed, everything has to be set-u so that it can be used in *Headless* mode. *Headless* mode enables remote connection to the Raspberry Pi, without the need for a *PC* screen Monitor, mouse or keyboard. The only things that are used in this 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

which can be found on the following link:
https://www.az-delivery.de/products/raspberry-pi-kostenfreies-e-book?ls=en

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

Connecting the module with microcontroller

Connect the GY-21 humidity and temperature sensor module with the microcontroller as shown on the following connection diagram:



Module pin	Microcontroller pin	Wire color
VIN	5V	**Red wire**
GND	GND	**Black wire**
SCL	A5	**Blue wire**
SDA	A4	**Green wire**

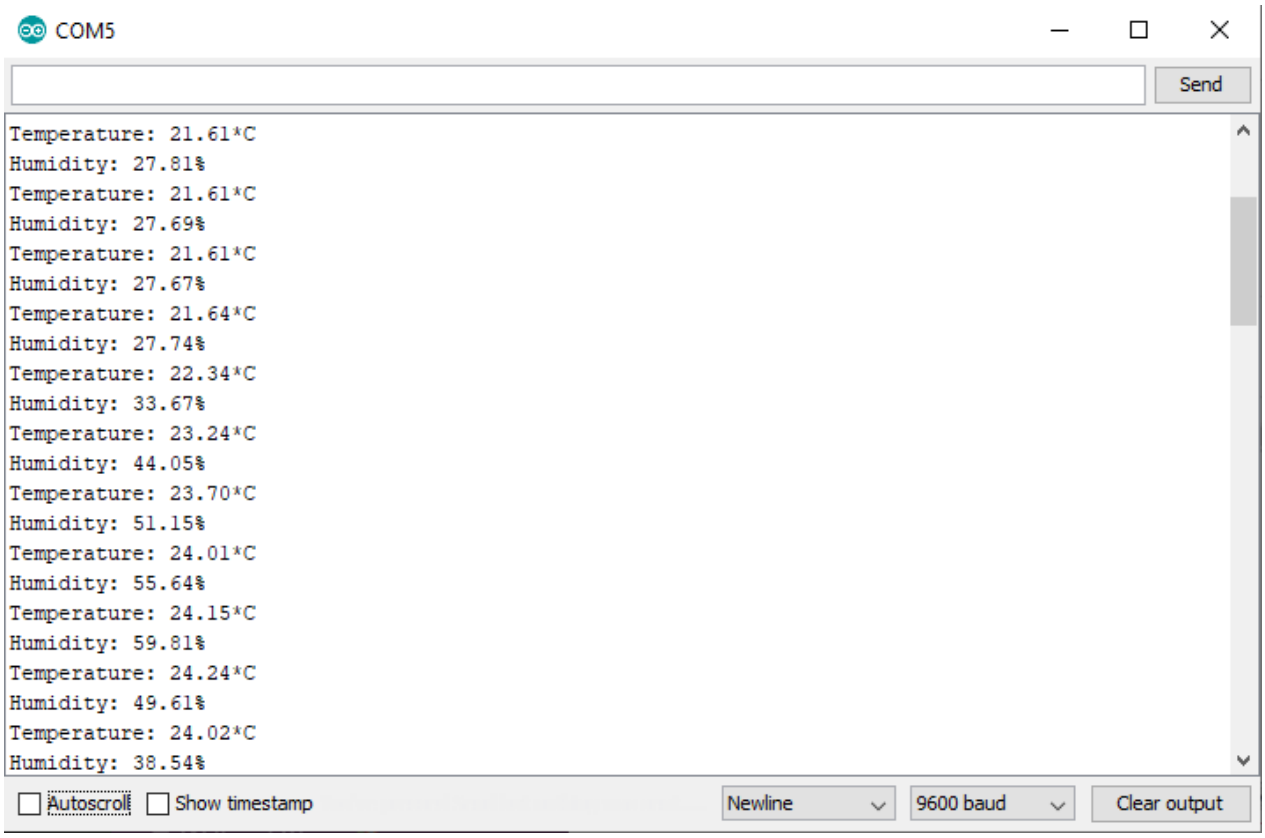
Sketch example

```
#include <Wire.h>
const int ADDRESS = 0x40;
double temperature, humidity;
void sensor_init(const int addr) {
  Wire.begin();  delay(100);
  Wire.beginTransmission(addr);
  Wire.endTransmission();
}
double read_temperature(const int addr) {
  double temperature;
  int low_byte, high_byte, raw_data;
  /**Send command of initiating temperature measurement**/
  Wire.beginTransmission(addr);
  Wire.write(0xE3);
  Wire.endTransmission();
  /**Read data of temperature**/
  Wire.requestFrom(addr, 2);
  if (Wire.available() <= 2) {
    high_byte = Wire.read();
    low_byte = Wire.read();
    high_byte = high_byte << 8;
    raw_data = high_byte + low_byte;
  }
  temperature = (175.72 * raw_data) / 65536;
  temperature = temperature - 46.85;
  return temperature;
}
```

```
double read_humidity(const int addr) {
  double humidity, raw_data_1, raw_data_2;
  int low_byte, high_byte, container;
  /**Send command of initiating relative humidity measurement**/
  Wire.beginTransmission(addr);
  Wire.write(0xE5);
  Wire.endTransmission();
  /**Read data of relative humidity**/
  Wire.requestFrom(addr, 2);
  if(Wire.available() <= 2) {
    high_byte = Wire.read();
    container = high_byte / 100;
    high_byte = high_byte % 100;
    low_byte = Wire.read();
    raw_data_1 = container * 25600;
    raw_data_2 = high_byte * 256 + low_byte;
  }
  raw_data_1 = (125 * raw_data_1) / 65536;
  raw_data_2 = (125 * raw_data_2) / 65536;
  humidity = raw_data_1 + raw_data_2;
  humidity = humidity - 6;
  return humidity;
}

void setup() {
  Serial.begin(9600);
  sensor_init(ADDRESS);
}
```

```
void loop() {
  temperature = read_temperature(ADDRESS);
  humidity = read_humidity(ADDRESS);

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println("*C");

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println("%");

  delay(1000);
}
```

Upload the sketch to the microcontroller and open Serial Monitor (*Tools* > *Serial Monitor*). The result should look like the output on the following image:



The sketch starts with including a library that is used for the I2C communication.

Next, a constant called *ADDRESS* is created. This constant represents the I2C address of the module.

After this, two double variables are created: *temperature* and *humidity*. These two variables are used for storing and displaying data read from the module.

Then, three functions are created: *sensor_init()*, *read_temperature()* and *read_humidity()*. All three functions have one argument, which is the I2C address of the module.

The first function, *sensor_init()*, returns no value. It is used to initialize the I2C communication between the microcontroller and the module.

The second function, *read_temperature()*, returns a double value. The return value represents the temperature data read from the module. All this function does is reading the sensor and returning the temperature data. The algorithm of this function is not covered in this eBook.

The third function, *read_humidity()*, returns a double value. The return value represents the humidity data read from the module. All that this function does is reading the sensor and returning the humidity data. The algorithm of this function is not covered in this eBook.

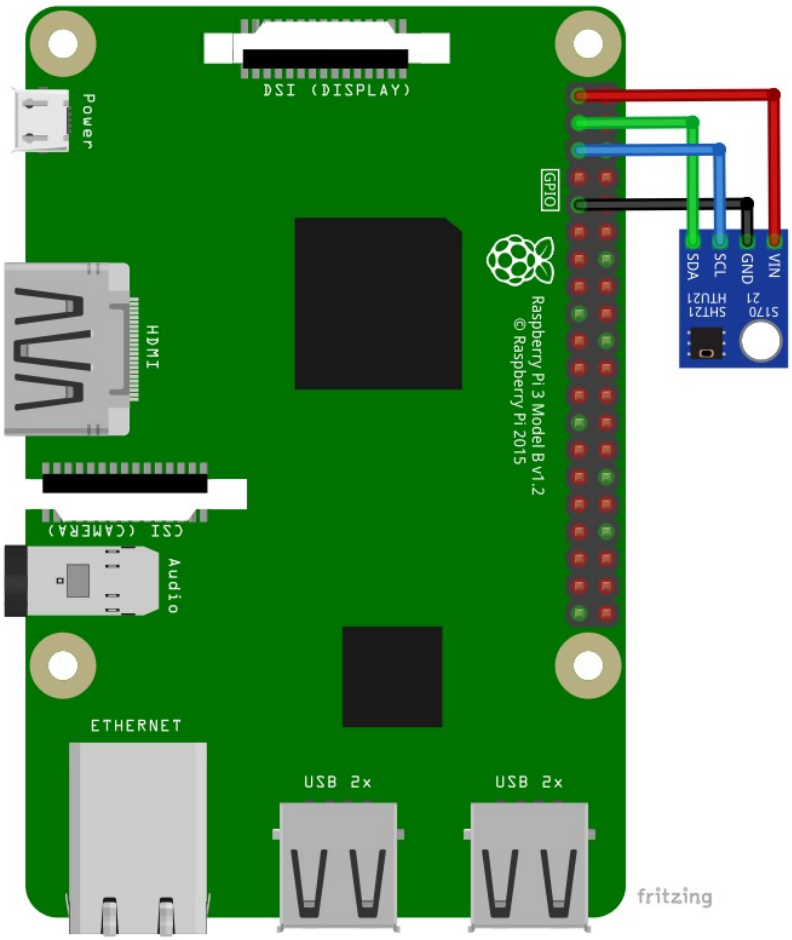
In the *setup()* function, the serial communication is started with the baud rate of *9600bps* and the *sensor_init()* function is executed.

In the *loop()* function, sensor data is read and stored in the *temperature* and *humidity* variables. After this, data is displayed in the Serial Monitor.

There is one second delay between two loops of the *loop()* function.

Connecting the module with Raspberry Pi

Connect the GY-21 sensor module with the Raspberry Pi as shown on the following connection diagram:

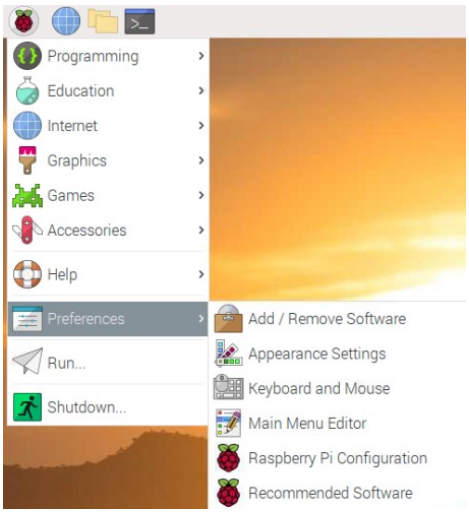


Module pin	Raspberry Pi pin	Physical pin	Wire color
VIN	3.3V	1	**Red wire**
SDA	GPIO2	3	**Green wire**
SCL	GPIO3	5	**Blue wire**
GND	GND	9	**Black wire**

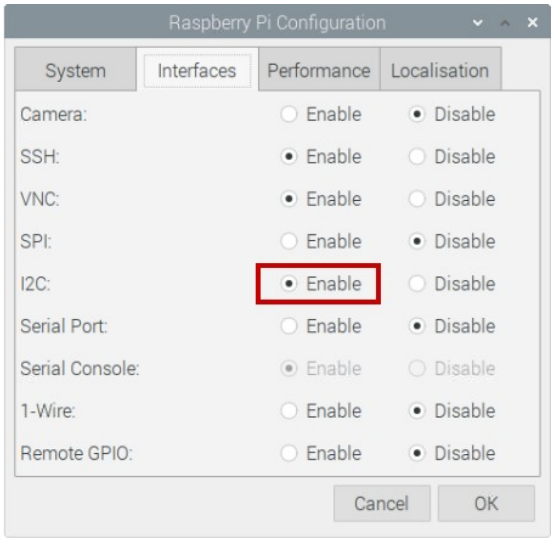
Enabling the I2C interface

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*, as shown on the following image:



Libraries and tools for Python

In order to use the module with the Raspberry Pi, it is recommended to download and install an external library for it. The library used in this eBook is called *Adafruit CircuitPython HTU21D*. Before it is used, several libraries and tools have to be installed.

First, the following tools have to be installed before using the library: *i2c-tools*, *setuptools* and *python3-pip*. If these tools are not installed, open the terminal and run the following commands, one by one:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install -y python3-pip setuptools i2c-tools
```

Second, also the following Python libraries have to be installed: *RPI.GPIO* and *adafruit-blinka*. If these libraries are not installed open the terminal and run the following commands, one by one:

```
sudo pip3 install RPI.GPIO
sudo pip3 install adafruit-blinka
```

Now, the library for the GY-21 can be installed. Open the terminal and run the following command:

```
sudo pip3 install adafruit-circuitpython-htu21d
```

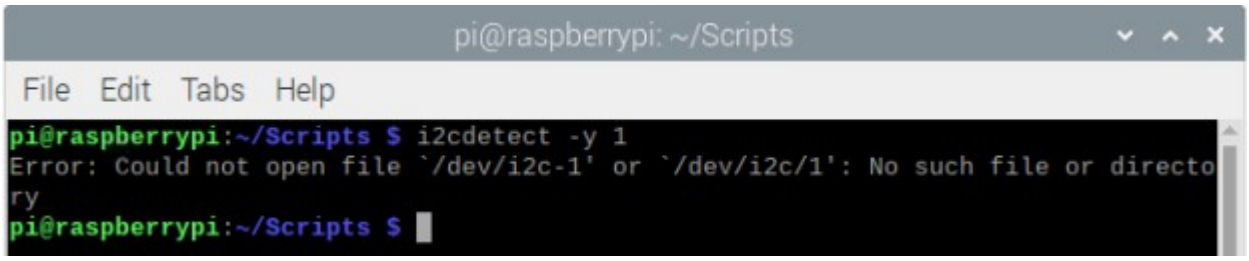
Check the I2C address of the module by running the following command:

```
i2c detect -y 1
```

The result should look like the output on the following image:

Where *0x0d* is the I2C address of the module.

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

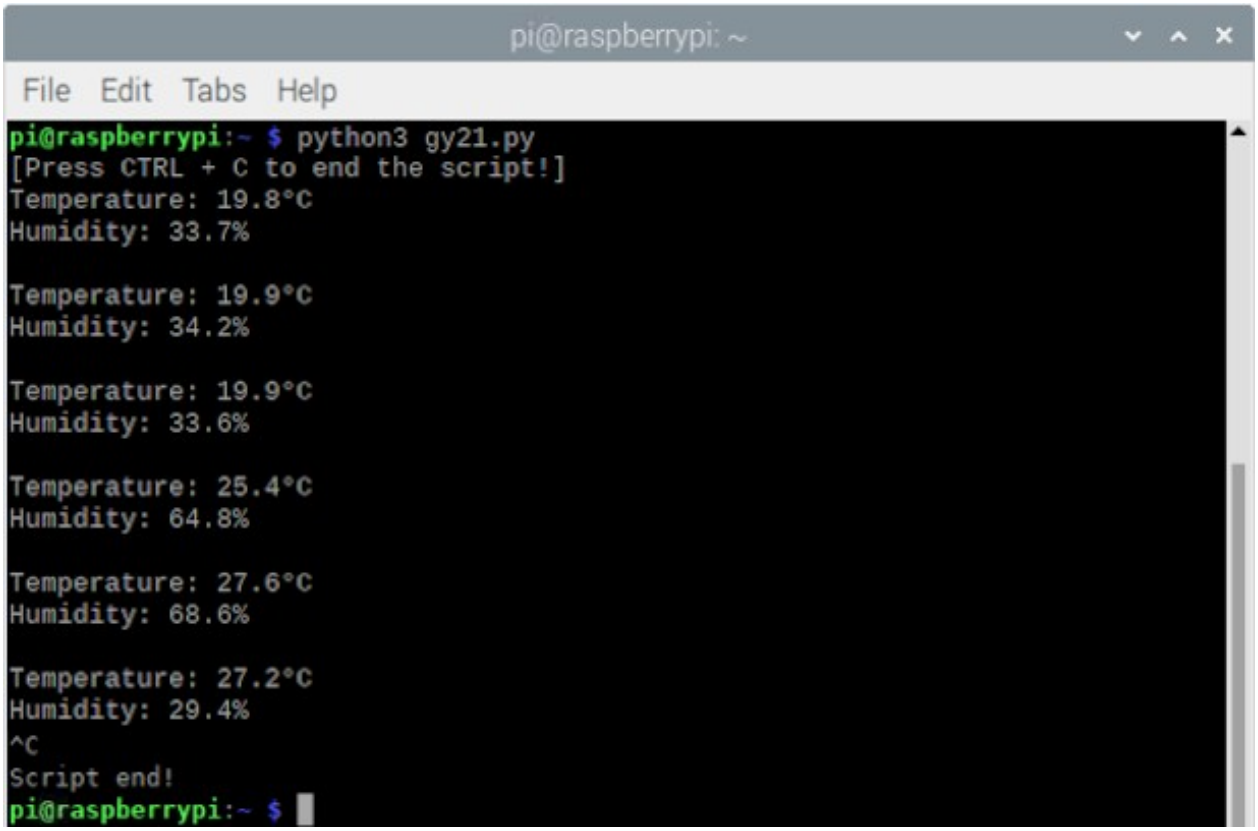


Python script

```
import time
import board
import busio
from adafruit_htu21d import HTU21D

ds = u'\xb0' # degree sign

i2c = busio.I2C(board.SCL, board.SDA)
sensor = HTU21D(i2c)

print('Press CTRL + C to end the script!')
try:
     while True:
          temp = sensor.temperature
          humidity = sensor.relative_humidity
          print('Temperature: {:.1f}{}C'.format(temp, ds))
          print('Humidity: {:.1f}%\n'.format(humidity))
          time.sleep(1)

except KeyboardInterrupt:
     print('\nScript end!')
```

Save the script under the name *gy21.py*. To run the script open the terminal in the directory where you saved the script and run the following command: **`python3 gy21.py`**

The result should look like the output on the following image:



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

You have done it!

Now you can use your module for various projects.

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 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