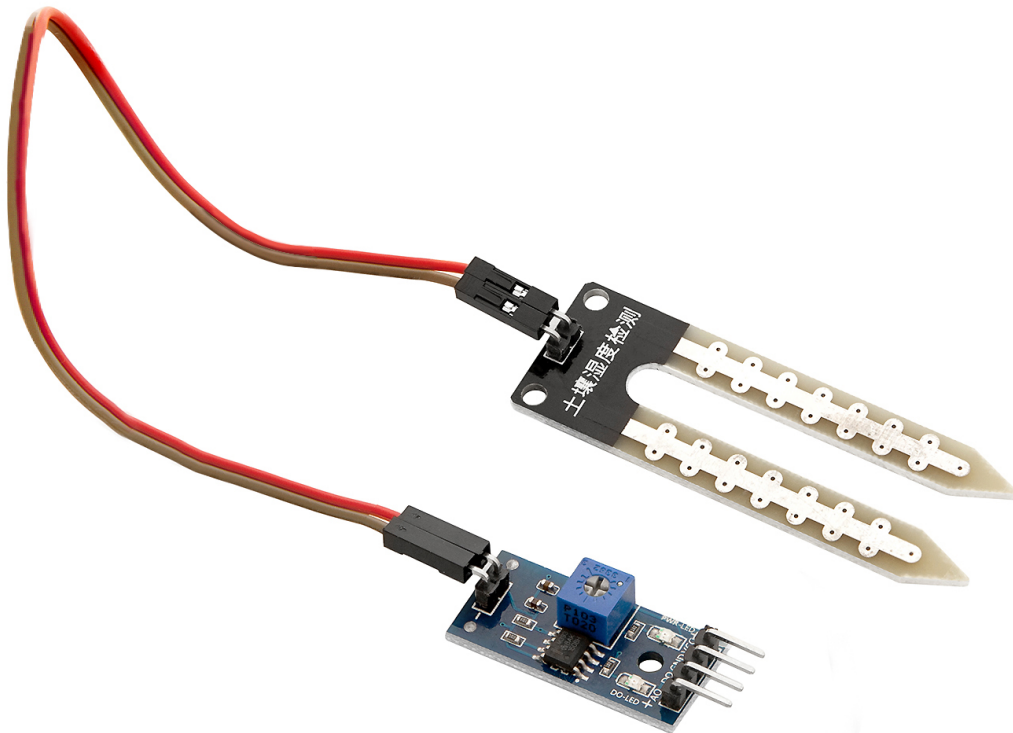# AZ-Delivery

Welcome!

Thank you for purchasing our *AZ-Delivery Hygrometer v1.0 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 hygrometer module measures the volumetric content of water inside the soil. Module outputs both the analog and the digital signal.

The module consists of a Hygrometer sensor, control board and jumper cables for connecting the Hygrometer sensor and control board.

The hygrometer sensor measures the volumetric content of water in the soil. Control board enables current flow through the hygrometer sensor, the soil and back to the control board. Depending on the amount of water in the soil, the resistance of the soil is different. This change in resistance can be measured by the hygrometer module. If there is a higher amount of water in a ground, the resistance of the ground is small, and vice versa.

The control board reads the resistance change from the hygrometer sensor and converts it into an analog and digital signal.

Specifications

|	
Operating voltage:	from 3.3V up to 5V
Current consumption:	8mA
Sensor output:	analog, digital
Sensitivity control:	via on-board potentiometer
Hygrometer sensor dimensions:	60x20x1.5mm (2.4x0.7x0.1in)
Control board dimensions:	32x14x17mm (1.2x0.5x0.6in)

The control board has the LM393 comparator chip and the output signal driving current is over *15mA*.

The module has on-board LEDs which are used for power and detection indication.

The module sensitivity can be adjusted with an on-board potentiometer. Moving the potentiometer shaft into the clockwise direction increases sensitivity. Moving the shaft of the potentiometer in the counterclockwise direction decreases the sensitivity of the module.

The pinout

The Hygrometer module consists of two boards, the sensor board and the control board. The sensor board has two pins, and the control board has six pins. The pinouts of these two boards are shown on the image:
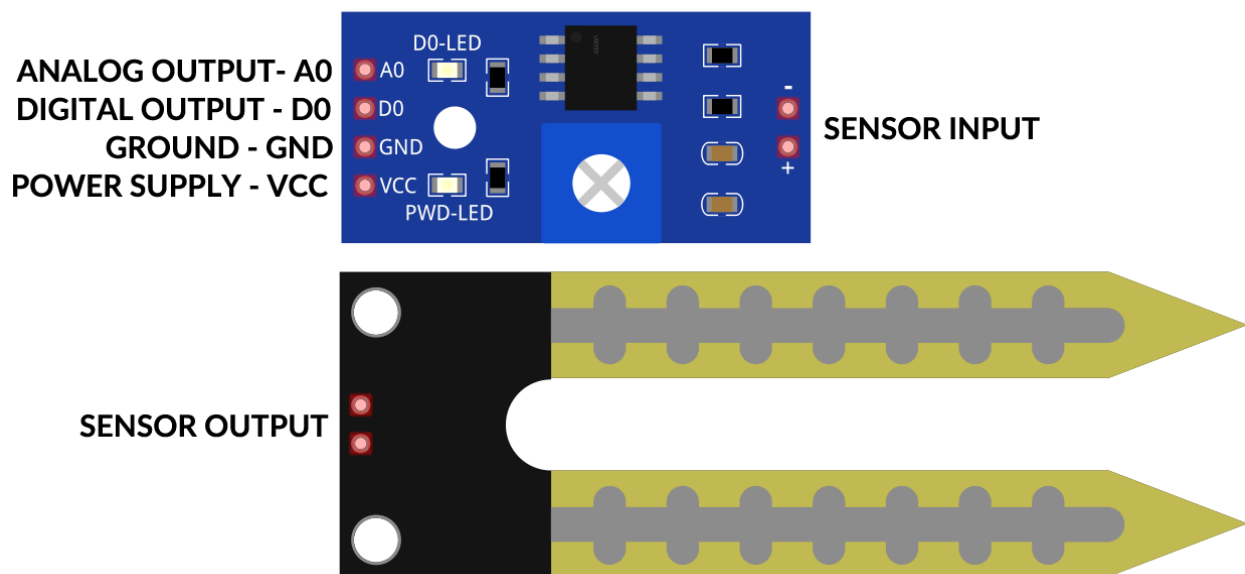
The sensor board does not have polarity, which means that two pins from the sensor board can be connected either way, on the control board SENSOR INPUT pins.

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.

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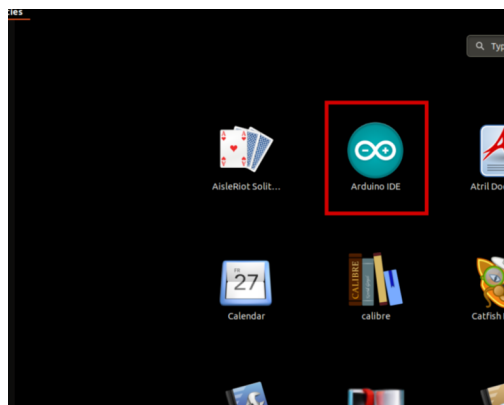
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

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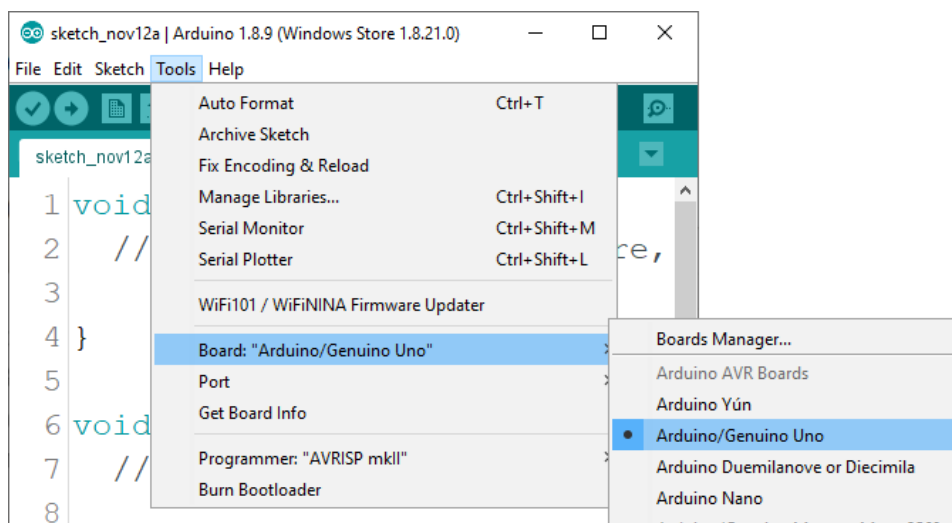
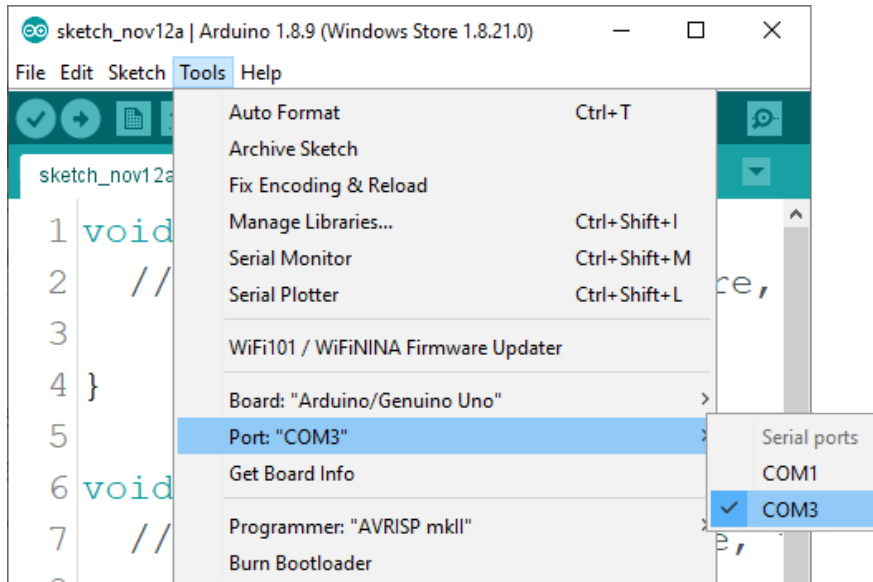The port to which the Arduino board is connected has to be selected. Go to:
Tools > Port > {port name goes here}
and when the Arduino board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

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:
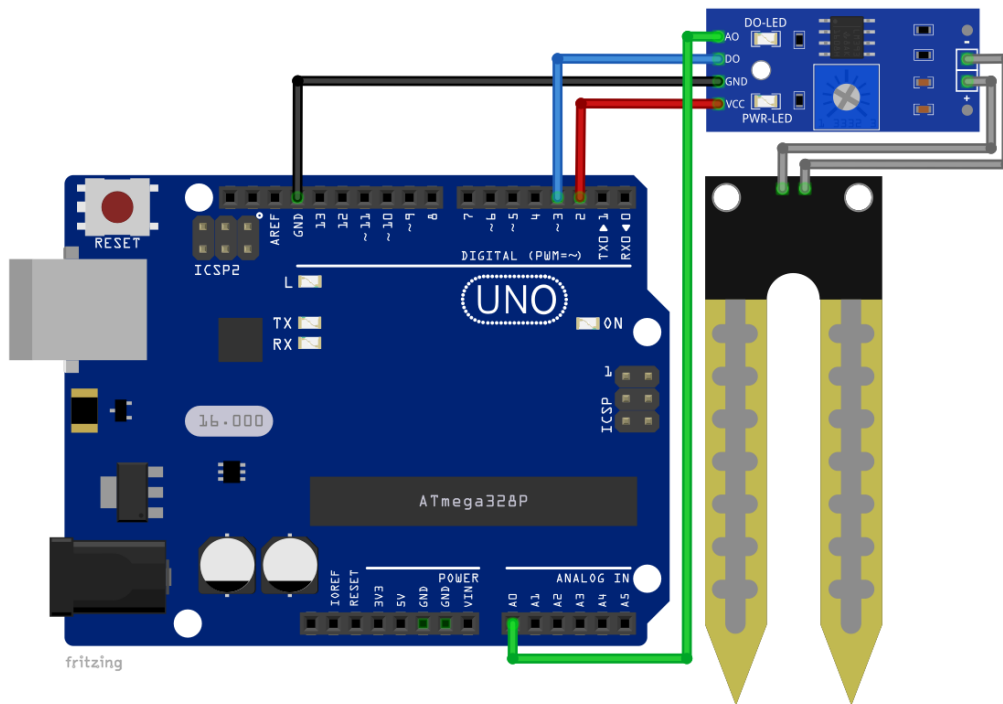
Module pin	Uno pin	Wire color
VCC	D2	**Red wire**
DO	D3	**Green wire**
AO	A0	**Blue wire**
GND	GND	**Black wire**

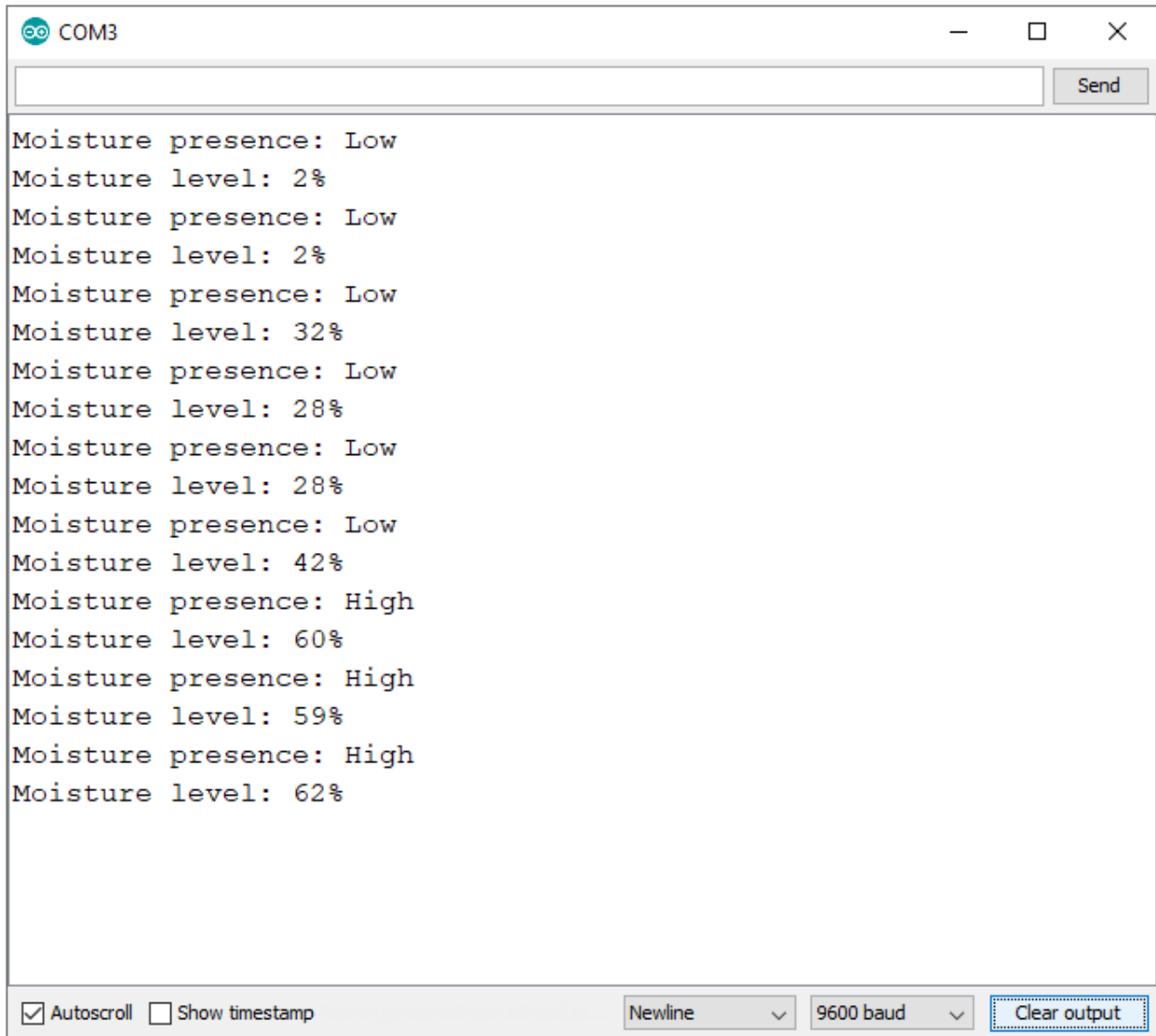**NOTE:** Connect the Hygrometer module with the control board via jumper cables that come with the module.

Sketch example

```
#define DIGITAL_PIN 3
#define ANALOG_PIN 0
#define SENSOR_POWER 2
uint16_t moistVal;
boolean isMoist = false;
String moisture;
void setup() {
  Serial.begin(9600);
  pinMode(DIGITAL_PIN, INPUT);
  pinMode(SENSOR_POWER, OUTPUT);
  digitalWrite(SENSOR_POWER, LOW);
}
void loop() {
  digitalWrite(SENSOR_POWER, HIGH);
  delay(10);
  moistVal = analogRead(ANALOG_PIN);
  isMoist = digitalRead(DIGITAL_PIN);
  digitalWrite(SENSOR_POWER, LOW);
  if (isMoist) {
    moisture = "No";
  }
  else {
    moisture = "Yes";
  }
  moistVal = map(moistVal, 0, 1023, 100, 0);
  Serial.print("Raining: ");
  Serial.println(moisture);
  Serial.print("Moisture: ");
  Serial.print(moistVal);
  Serial.println("%\n");
  delay(10000);
}
```

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

COM3

Moisture presence: Low
Moisture level: 2%
Moisture presence: Low
Moisture level: 2%
Moisture presence: Low
Moisture level: 32%
Moisture presence: Low
Moisture level: 28%
Moisture presence: Low
Moisture level: 28%
Moisture presence: Low
Moisture level: 42%
Moisture presence: High
Moisture level: 60%
Moisture presence: High
Moisture level: 59%
Moisture presence: High
Moisture level: 62%

Autoscroll Show timestamp   Newline   9600 baud   Clear output

The sketch starts with defining and creating three macros called *DIGITAL_PIN*, *ANALOG_PIN* and *SENSOR_POWER*.

The *DIGITAL_PIN* represents the digital pin of Uno that is used for connecting the sensor digital output pin.

The *ANALOG_PIN* represents the analog input pin of Uno that is used for connecting the sensor analog output pin.

The *SENSOR_POWER* represents the digital pin of Uno that is used for powering the module control board in predefined time intervals. This is done only for testing purposes and it is not recommended if more then one device is connected to the Uno. To power the module from the digital pin, the transistor or MOS-FET should be used which is not in the scope of this eBook.

The module data can be read in two ways. The one is by reading the analog output pin of the module, and the other is by reading the digital output pin of the module. To read the analog output pin of the module, the variable called *moistVal* is used to store return value from the *analogRead()* function. The return value is an integer number in the range from 0 to 1023. To convert it into a percentage, the *map()* function is used. This is a built-in function of the Arduino IDE. It has five arguments and returns an integer value.

For example:

```
moistVal = map(input, in_min, in_max, out_min, out_max)
```

First argument is the *input* value, which is in the range from the *in_min* to *in_max*. The return value is an integer number in the range from *out_min* to *out_max*. This function maps one number in the input range, to another number which is in the different range.

To read the digital output pin of the module, the *isMoist* variable is used to store the return value of the *digitalRead()* function.

At the end of the *loop()* function, the data is displayed in the Serial Monitor. Between two measurements there is a 1 second pause:

```
delay(1000);
```

Connecting the module with Raspberry Pi

Connect the module with the Raspberry Pi as shown on the following image:
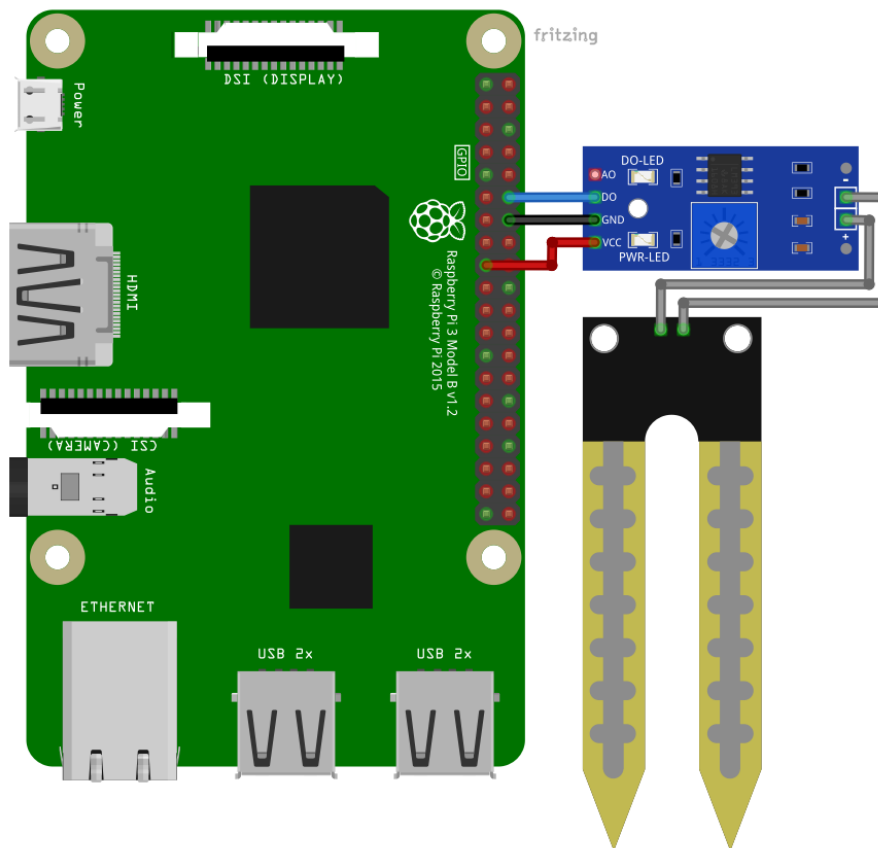
Module pin	Raspberry Pi pin	Physical pin	Wire color
D0	GPIO18	12	**Blue wire**
GND	GND	14	**Black wire**
VCC	3V3	17	**Red wire**

NOTE: Connect the sensor module with the control board via jumper cables that come with the module.

Libraries and tools for Python

To use the module with the Raspberry Pi, the library RPi.GPIO has to be installed. If the library is already installed, running the installation command only updates the library to a newer version.

To install the library, open the terminal and run the following commands, one by one:

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install python3-rpi.gpio
```

Python script

```python
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

DIGIOUT = 18
GPIO.setup(DIGIOUT, GPIO.IN)
time.sleep(2)

print('[Press CTRL + C to end the script!]')
try: # Main program loop
  while True:
    if GPIO.input(DIGIOUT)==0:
        print('Soil moisture level: HIGH')
        time.sleep(2)
    else:
        print('Soil moisture level: LOW')
        time.sleep(2)

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

finally:
    GPIO.cleanup()
```
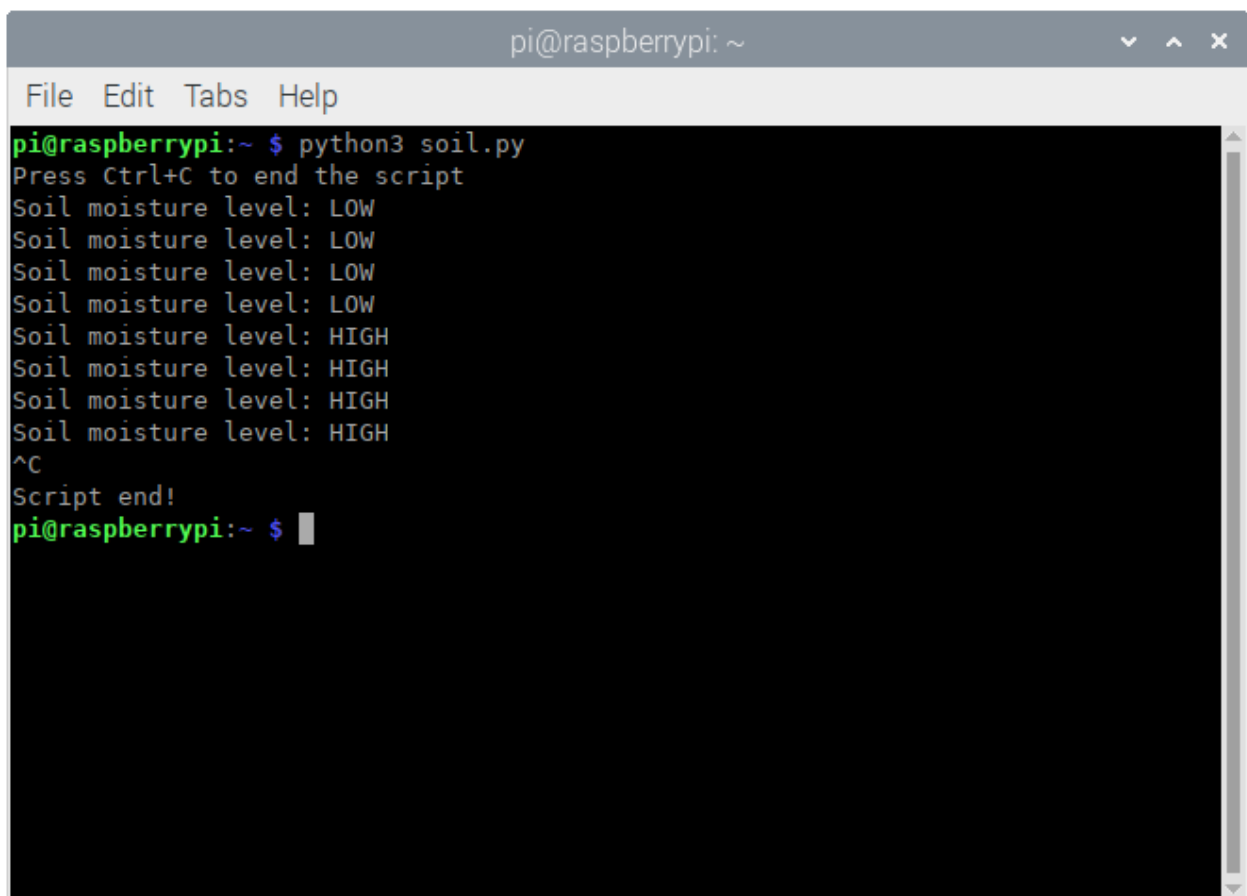
Save the script by the name *hygrometer.py*. To run the script, open the terminal in the directory where the script is saved and run the following command:

```
python3 hygrometer.py
```

The result should look like as on the following image:

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

AZ-Delivery

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