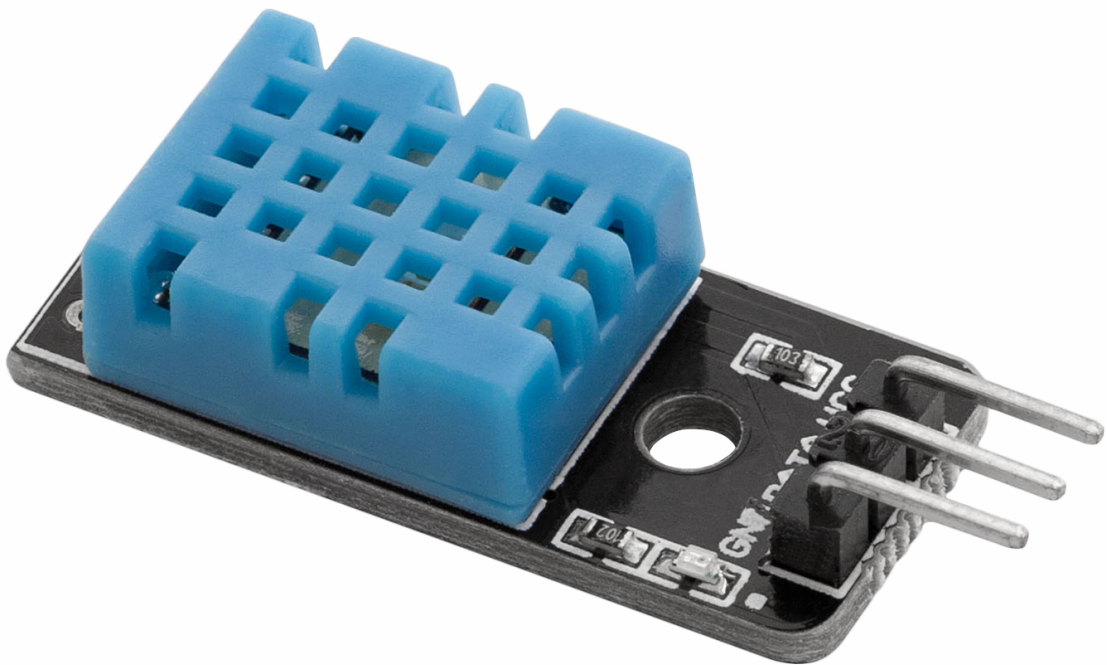


AZ-Delivery

Welcome!

Thank you for purchasing our *AZ-Delivery KY-015 DHT11 Temperature Sensor Module*. On the following pages, you will be introduced to how to use and set up this handy device.

Have fun!



Areas of application

Education and teaching: Use in schools, universities and training institutions to teach the basics of electronics, programming and embedded systems. Research and development: Use in research and development projects to create prototypes and experiments in the fields of electronics and computer science. Prototype development: Use in the development and testing of new electronic circuits and devices. Hobby and Maker Projects: Used by electronics enthusiasts and hobbyists to develop and implement DIY projects.

Required knowledge and skills

Basic understanding of electronics and electrical engineering. Knowledge of programming, especially in the C/C++ programming language. Ability to read schematics and design simple circuits. Experience working with electronic components and soldering.

Operating conditions

The product may only be operated with the voltages specified in the data sheet to avoid damage. A stabilized DC power source is required for operation. When connecting to other electronic components and circuits, the maximum current and voltage limits must be observed to avoid overloads and damage.

Environmental conditions

The product should be used in a clean, dry environment to avoid damage caused by moisture or dust. Protect the product from direct sunlight (UV)

Intended Use

The product is designed for use in educational, research and development environments. It is used to develop, program and prototype electronic projects and applications. The Sensor product is not intended as a finished consumer product, but rather as a tool for technically savvy users, including engineers, developers, researchers and students.

Improper foreseeable use

The product is not suitable for industrial use or safety-relevant applications. Use of the product in medical devices or for aviation and space travel purposes is not permitted

disposal

Do not discard with household waste! Your product is according to the European one Directive on waste electrical and electronic equipment to be disposed of in an environmentally friendly manner. The valuable raw materials contained therein can be recycled become. The application of this directive contributes to environmental and health protection. Use the collection point set up by your municipality to return and Recycling of old electrical and electronic devices. WEEE Reg. No.: DE 62624346

electrostatic discharge

Attention: Electrostatic discharges can damage the product. Note: Ground yourself before touching the product, such as by wearing an anti-static wrist strap or touching a grounded metal surface.

safety instructions

Although our product complies with the requirements of the RoHS Directive (2011/65/EU) and does not contain any hazardous substances in quantities above the permitted limits, residues may still be present. Observe the following safety instructions to avoid chemical hazards: Caution: Soldering can produce fumes that can be harmful to health. Note: Use a solder fume extractor or work in a well-ventilated area. If necessary, wear a respirator mask. Caution: Some people may be sensitive to certain materials or chemicals contained in the product. Note: If skin irritation or allergic reactions occur, stop use and, if necessary, consult a doctor. Caution: Keep the product out of the reach of children and pets to avoid accidental contact and swallowing of small parts. Note: Store the product in a safe, closed container when not in use. Attention: Avoid contact of the product with food and drinks. Note: Do not store or use the product near food to prevent contamination. Although our product complies with the requirements of the RoHS Directive (2011/65/EU) and does not contain any hazardous substances in quantities above the permitted limits, residues may still be present. Observe the following safety instructions to avoid chemical hazards: Caution: Soldering can produce fumes that can be harmful to health. Note: Use a solder fume extractor or work in a well-ventilated area. If necessary, wear a respirator mask. Caution: Some people may be sensitive to certain materials or chemicals contained in the product. Note: If skin irritation or allergic reactions occur, stop use and, if necessary,

consult a doctor. Caution: Keep the product out of the reach of children and pets to avoid accidental contact and swallowing of small parts. Note: Store the product in a safe, closed container when not in use. Attention: Avoid contact of the product with food and drinks. Note: Do not store or use the product near food to prevent contamination. The product contains sensitive electronic components and sharp edges. Improper handling or assembly can result in injury or damage. Observe the following safety instructions to avoid mechanical hazards: Attention: The product's circuit board and connectors may have sharp edges. Use caution to avoid cuts. Note: Wear appropriate protective gloves when handling and assembling the product. Caution: Avoid excessive pressure or mechanical stress on the board and components. Note: Only mount the product on stable and flat surfaces. Use appropriate spacers and housings to minimize mechanical stress. Attention: Make sure the product is securely fastened to prevent accidental slipping or falling. Note: Use appropriate support or secure mounting in enclosures or on mounting plates. Caution: Make sure all cable connections are connected securely and correctly to avoid strain and accidental unplugging. Note: Route cables so that they are not under tension and do not pose a tripping hazard. The product operates with electrical voltages and currents that, if used improperly, can result in electric shocks, short circuits or other hazards. Observe the following safety instructions to avoid electrical hazards: Attention: Use the product only with the specified voltages. Note: The performance limits of the product can be found in the associated data sheet Caution: Avoid short circuits between the connectors and components of the product Note: Make sure that no conductive objects touch or bridge the circuit board. Use insulated tools and pay attention to the arrangement of connections. Caution: Do not perform any work on the product when it is connected to a power source. Note: Disconnect the product from power before making any circuit changes or connecting or removing components. Caution: Do not exceed the specified current ratings for the product's inputs and outputs. Note: The performance limits of the product can be found in the technical specifications or in the data sheet Attention: Make sure that the power sources used are stable and correctly sized. Note: Only use tested and suitable power supplies to avoid voltage fluctuations and overloads. Attention: Maintain sufficient distance from live parts to avoid accidental contact. Note: Ensure that the cabling is arranged safely and clearly according to the voltage used. Caution: Use insulating housings or protective covers to protect the product from direct contact. Note: Place the product in a non-conductive case to avoid accidental touching and short circuits. The product and the components on it may become warm during operation. Improper handling or overloading the product can result in burns, damage or fire. Observe the following safety instructions to avoid thermal hazards: Caution: Make sure the product is used within recommended operating temperatures. Note: The recommended operating temperature range is typically between -40°C and +85°C. Check the specific information in the product data sheet. Attention: Do not place the product near external heat sources such as radiators or direct sunlight. Note: Ensure that the product is operated in a cool and well-ventilated area. Attention: Make sure the product is well ventilated to avoid overheating. Note: Use fans or heat sinks when operating the product in a closed enclosure or in an environment with limited air circulation. Attention: Mount the product on heat-resistant surfaces and in heat-resistant housings. Note: Use enclosure materials that can withstand high temperatures to avoid damage or fire hazard. Caution: Implement temperature monitoring when using an enclosure and, if necessary, protection mechanisms that shut down the product if it overheats. Note: Note: Use temperature sensors and appropriate software to monitor the temperature of the product and shut down the system if necessary. Caution: Avoid overloads that can cause excessive heating of components. Note: To prevent overheating, do not exceed the specified current and voltage limits. Caution: Short circuits can generate significant heat and cause fires. Note: Make sure that all connections are correct and secure and that no conductive objects can accidentally cause short circuits.



Table of Contents

Introduction.....	3
Specifications.....	4
Working principle of the DHT11 sensor.....	5
The pinout.....	7
How to set-up Arduino IDE.....	8
How to set-up the Raspberry Pi and Python.....	12
Connecting the module with Atmega328p.....	13
Library for Arduino IDE.....	14
Sketch example.....	15
Connecting the module with Raspberry Pi.....	19
Library and tools for Python.....	20
Python script.....	21



Introduction

The DHT11 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 is from 0°C to $+50^{\circ}\text{C}$, with $\pm 2^{\circ}\text{C}$ accuracy.

Humidity measuring range is from 20% to 90%, with accuracy from $\pm 5\%$.

Sampling rate of the sensor is 1Hz , which means that sensor data can be read only once per second.

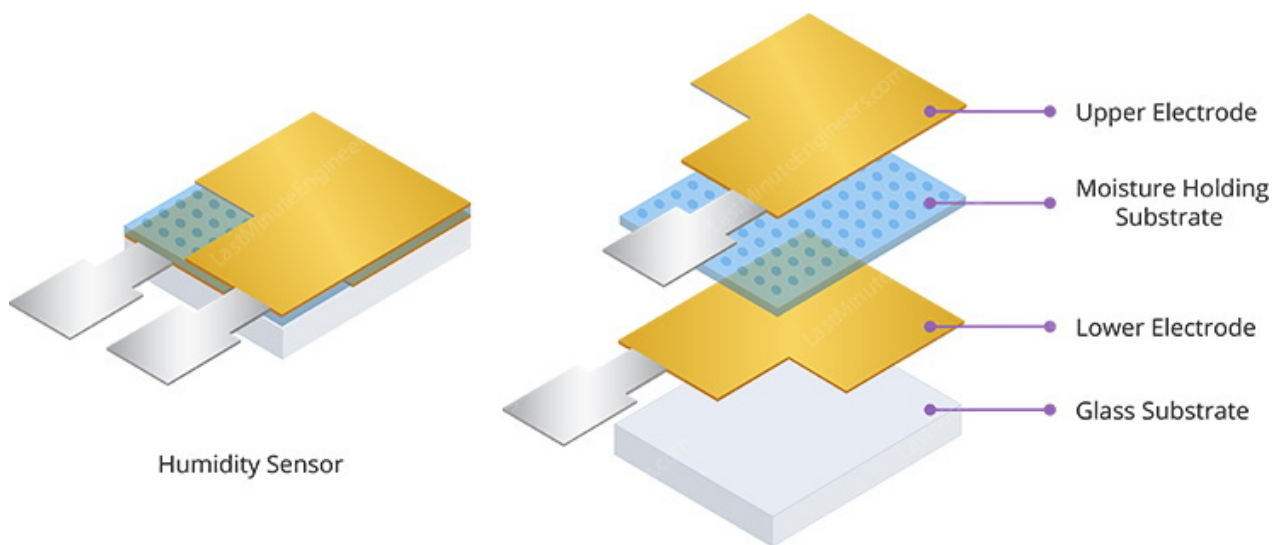


Specifications

- » 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]

Working principle of the DHT11 sensor

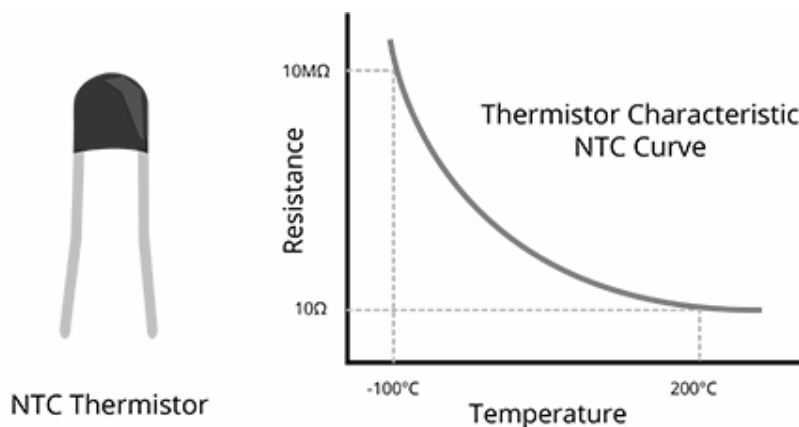
Inside the case, on the sensing side of DHT11 sensor, there is a humidity sensing component along with a NTC temperature sensor (or thermistor).



Humidity sensing component is used to measure humidity, which has two electrodes with moisture holding substrate (usually salt or conductive plastic polymer) sandwiched between the 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. The higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

Az-Delivery

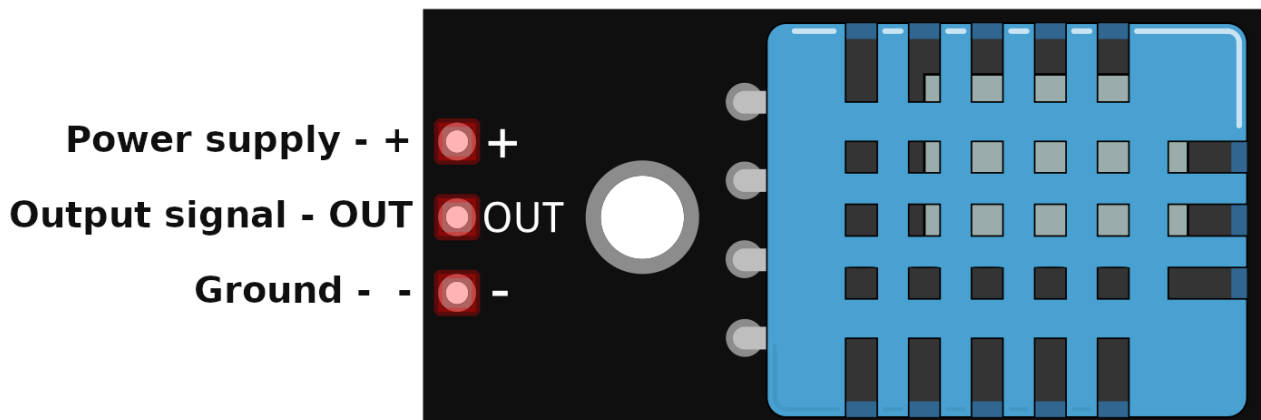
The temperature sensing part of the DHT11 sensor consists of an NTC thermistor. A thermistor is a thermal resistor, a resistor that changes its resistance with temperature. Technically, all resistors are thermistors, their resistance changes slightly with temperature, but the change is usually small and difficult to measure. Thermistors are made so that the resistance changes drastically with temperature. The change can be 100Ω or more per degree of temperature. The term *NTC* means *Negative Temperature Coefficient*, which means that the resistance decreases with the increase of temperature.



On the other side, there is a small PCB with an 8 bit SOIC14 packaged integrated circuit. The integrated circuit 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

The KY-015 DHT11 temperature sensor module has three pins. The pinout diagram is shown on the following image:



Power supply (+) pin - supplies power for the sensor. Although supply voltage can range between 3.3V and 5V, a 5V supply is recommended. In case of a 5V power supply, a cable that connect 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.

OUT (DATA) pin - is the data pin, and it is used for communication between the sensor and the microcontroller.

GND (-) pin - is ground pin and should be connected to the common ground, or 0V (on Atmega328p 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 that runs on Windows, Mac OS X, and Linux. Below this, it states that the software can be used with any Arduino board and refers to the 'Getting Started' page for installation instructions. On the right side of the page, there is a teal sidebar with links for different operating systems: Windows (Installer, ZIP file), Windows app (Requires Win 8.1 or 10), Mac OS X (10.8 Mountain Lion or newer), 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*.

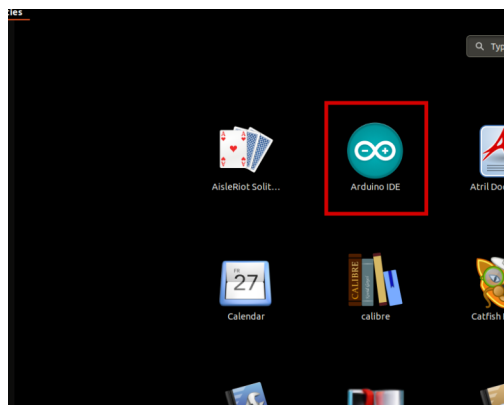
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

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

user_name - is the name of a superuser in 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* script has to be used after 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.



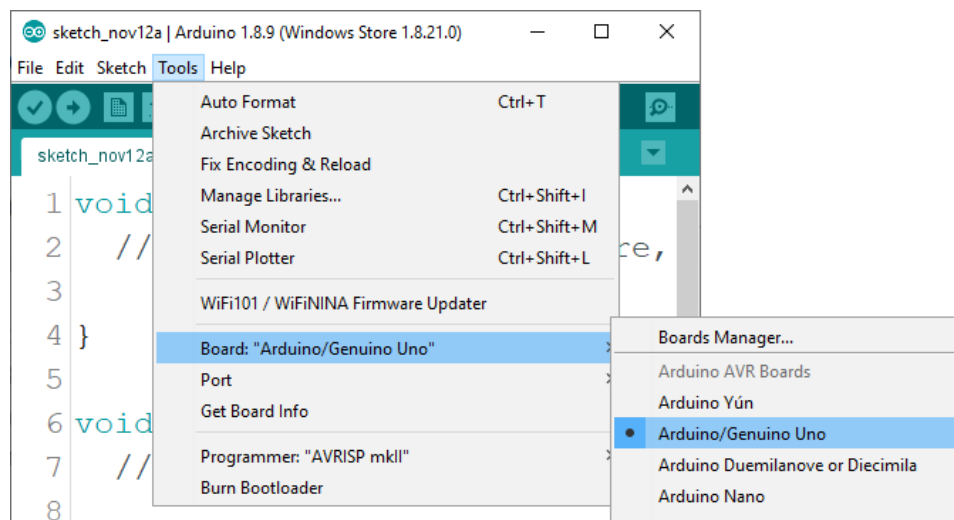
Az-Delivery

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 Atmega328p 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:

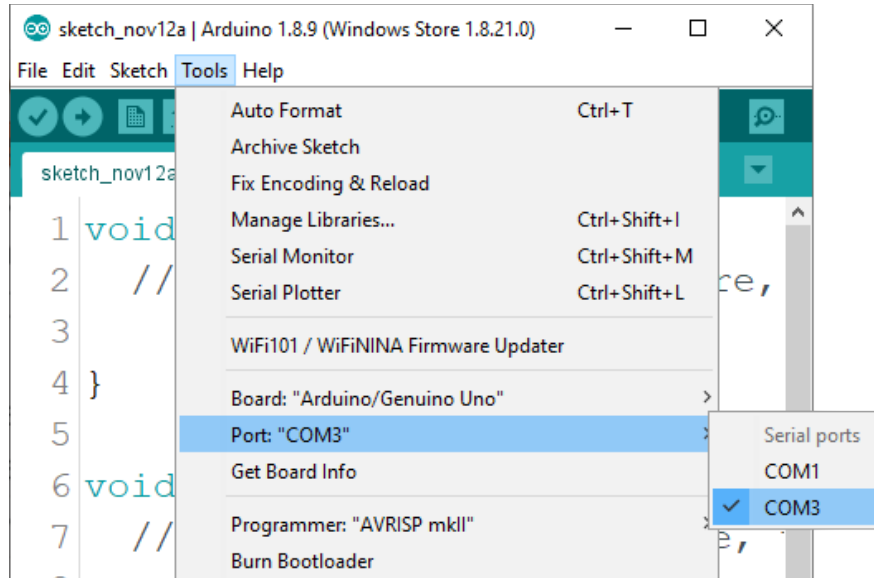


The port to which the Atmega328p board is connected has to be selected. Go to: *Tools > Port > {port name goes here}*

and when the Atmega328p 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:

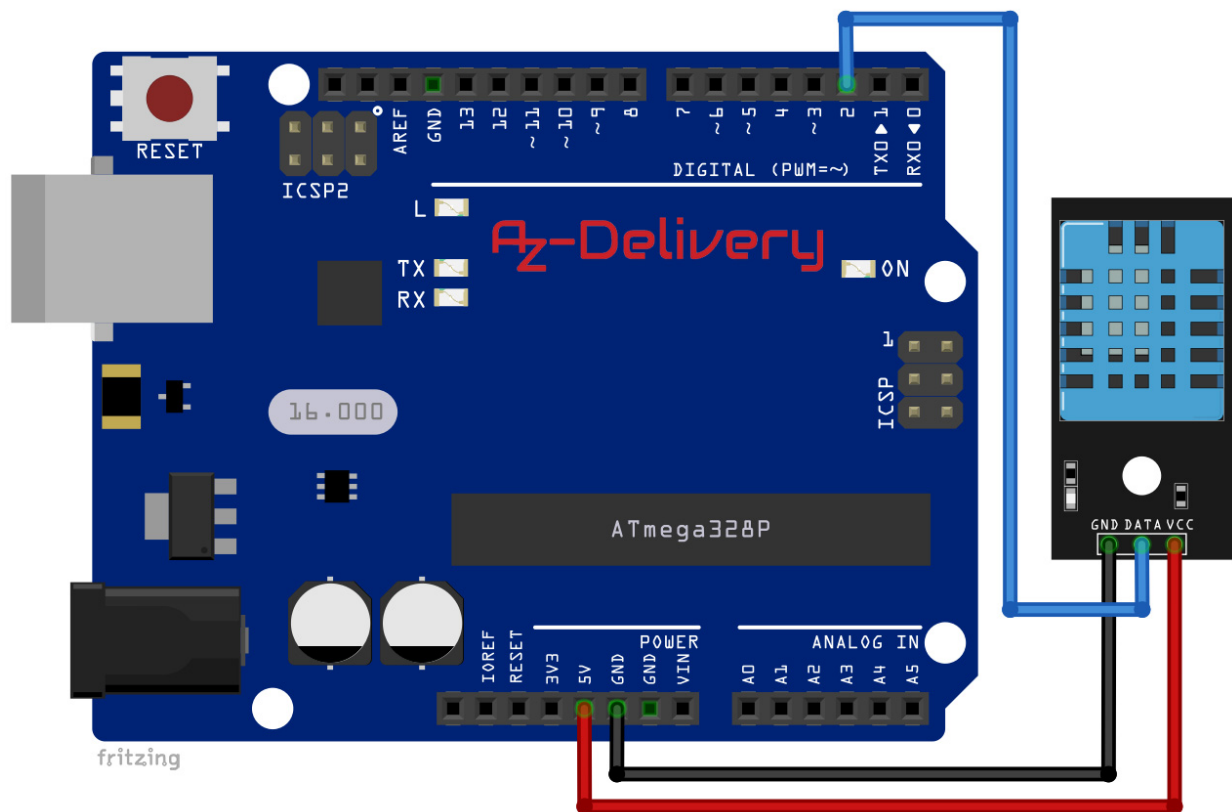
[Raspberry Pi Quick Startup Guide](#)

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

Az-Delivery

Connecting the module with Atmega328p

Connect the KY-015 module with the Atmega328p as shown on the following connection diagram:



KY-015 pin	>	Mc pin
DATA	>	D2
GND	>	GND
VCC	>	5V

Blue Wire

Black Wire

Red Wire



Library for Arduino IDE

To use the DHT11 module with Atmega328p it is recommended to install an external library. The library used in this eBook is called the *SimpleDHT* library, which can be downloaded from [GitHub](#) page. When the *.zip* file is downloaded, open Arduino IDE and go to:

Sketch > Include Library > Add .ZIP Library
and add the downloaded *.zip* file.

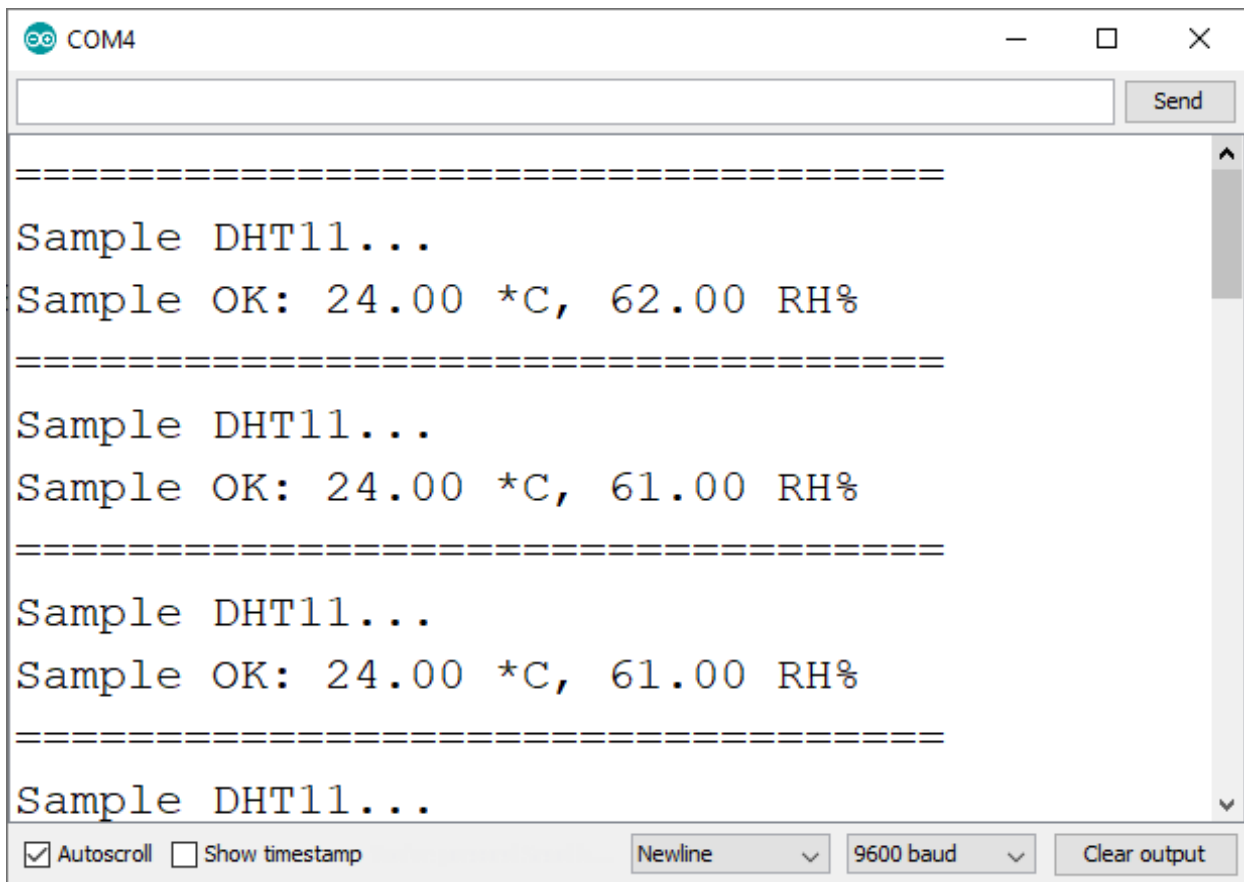
Sketch example

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

```
#include <SimpleDHT.h>
#define DHT11_PIN 2
SimpleDHT11 dht11(DHT11_PIN);
void setup() {
    Serial.begin(9600);
}
void loop() {
    Serial.println("=====");
    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 Atmega328p and open Serial Monitor (*Tools > Serial Monitor*). The result should look like the output on the image below:



The screenshot shows the Serial Monitor window for COM4. The output 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 contains the following controls:

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

Az-Delivery

Sketch starts with importing the *SimpleDHT* library.

Next, a macro called *DHT11_PIN* is created and initialized with integer value 2. This macro represents a digital I/O pin on which the *DATA* pin of the DHT11 module is connected.

Then, the object called *dht11* is created where the previously created macro is used to initialize it.

In the *setup()* function the serial communication is started with a baud rate of *9600bps*.

In the *loop()* function, the sensor data is read every 1.5 seconds, with the following lines of code:

```
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;
}
```

Az-Delivery

Where the *err* variable represents the error during reading the sensor data.

If there is error while reading the sensor, the message:

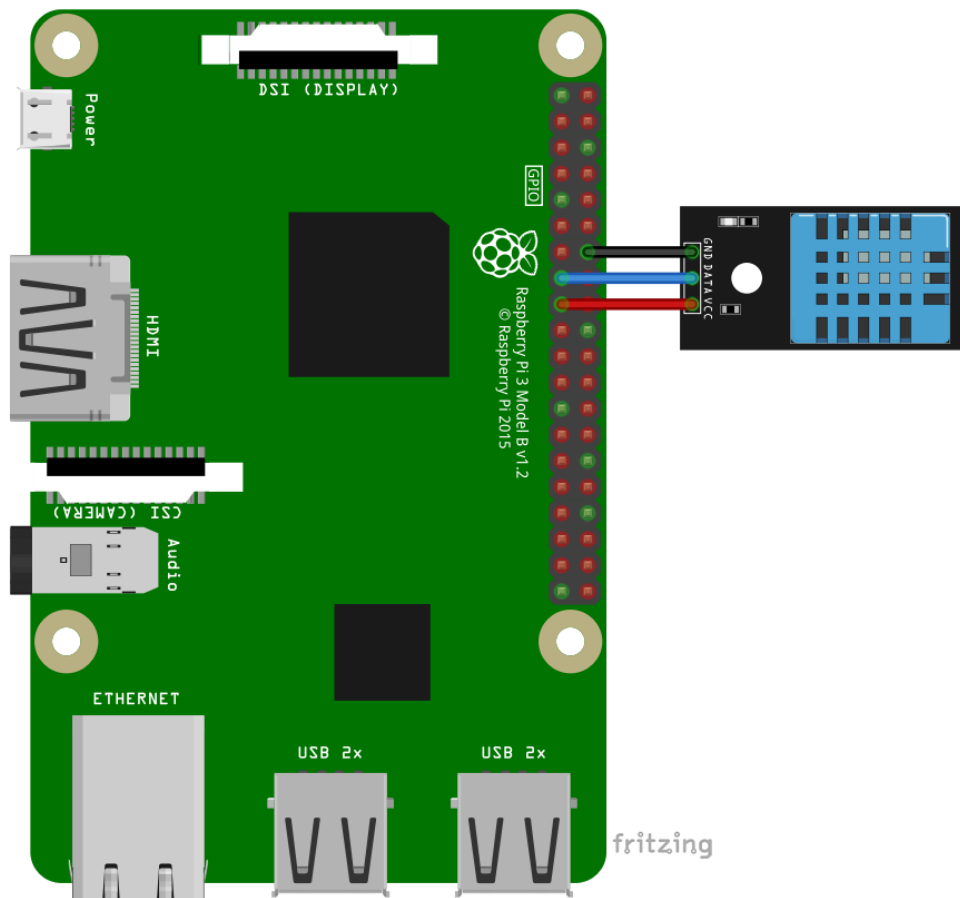
Read DHT11 failed, err="error data"

where *error data* is detailed information about the error. If there is no error while reading the sensor, sensor data is saved in the *temperature* and *humidity* variables.

After this, the sensor data is displayed in the Serial Monitor.

Connecting the module with Raspberry Pi

Connect the KY-015 module with the Raspberry Pi as shown on the following connection diagram:



KY-015 pin	>	Raspberry Pi pin
GND	>	GND [PIN 14]
DATA	>	GPIO22 [PIN 15]
VCC	>	3V3 [PIN 17]

Black Wire

Blue Wire

Red Wire

Az-Delivery

Library and tools for Python

To use the DHT11 sensor with the Raspberry Pi, it is recommended to install an external library. The library we are going to use is called *Adafruit_DHT*. To install it, first we need to make sure Raspbian is up to date. Start your Raspberry Pi, open 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
```

- The second command is for installing the *python3-pip* app:

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

We use the *pip* app to install the *Adafruit_DHT* library.

- The third command is for installing other apps which the *pip* app uses during installation:

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

After this is completed, we can install the *Adafruit_DHT* library. To do so, open terminal and run the following command:

```
sudo pip3 install Adafruit_DHT
```

```
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-cp35-cp35m-linux_armv7l.whl
Installing collected packages: Adafruit-DHT
Successfully installed Adafruit-DHT-1.4.0
```

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!]\n')
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!')

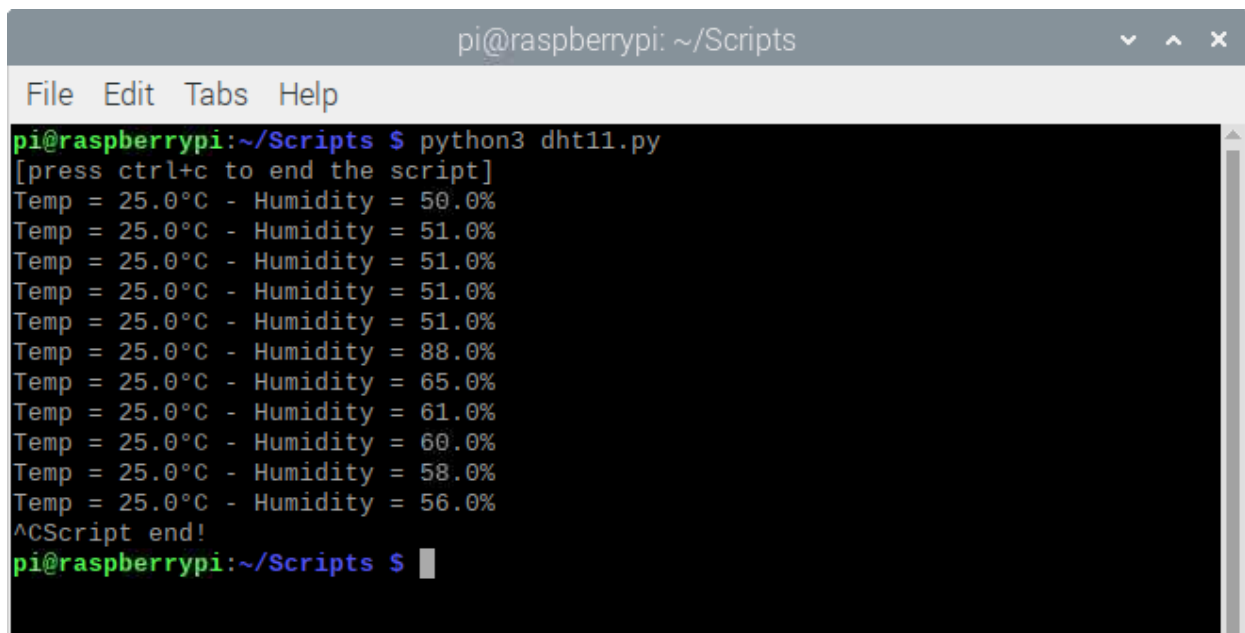
# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

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

python3 dht11.py

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



```
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.

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 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>