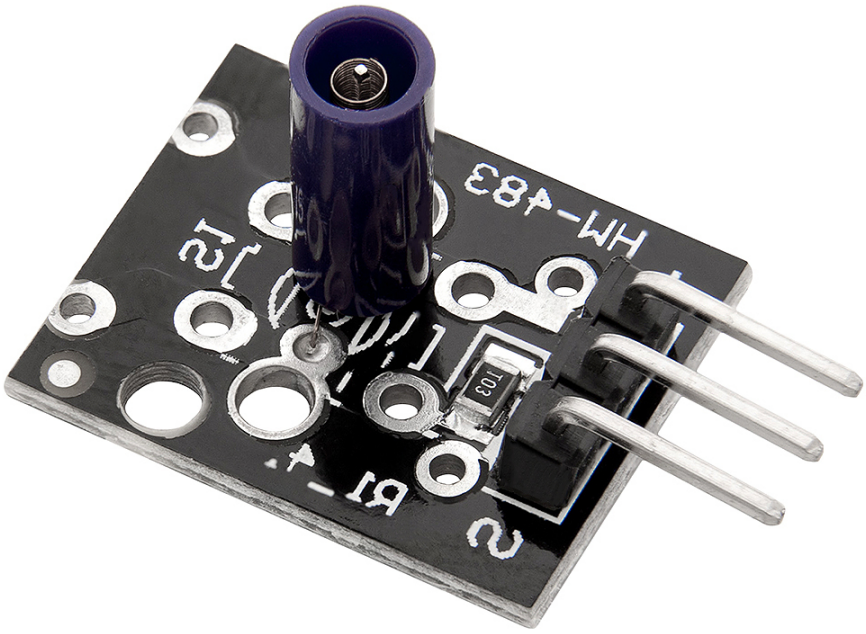
Welcome!

Thank you for purchasing our *AZ-Delivery KY-002n Shock 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
The products are intended for the support and assembly of electronic components and circuits.

Required knowledge and skills
The use of these products requires basic knowledge of electrical engineering and the handling of electronic components. Users should be able to install the products correctly and take the necessary safety precautions.

Environmental conditions
The products should be used in an environment free from moisture, dust and direct sunlight. They should not be operated near heat sources or in chemically aggressive environments to avoid damage and safety risks.

Intended Use
Passive electrical products such as heat sinks, battery holders and clips or breakout boards should be operated in environments that meet the specified temperature and voltage ranges of the respective products. These components are typically designed for indoor use.

Improper foreseeable use
Improper but foreseeable uses include use in humid or extremely hot environments or operation by untrained or disabled persons. The product must be kept away from children and pets.

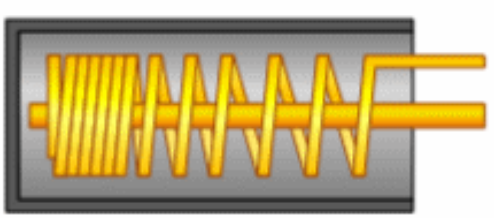
disposal
Do not discard with household waste! Your product is according to the European one Directive on waste electrical and electronic equipment to be disposed of in an environmentally friendly manner. The valuable raw materials contained therein can be recycled become. The application of this directive 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

safety instructions
Attention: Improper disposal of electronic components can endanger the environment and health. Note: Dispose of electronic components in accordance with local regulations and use appropriate recycling options. Attention: Chemically aggressive media can damage the materials of the products. Note: Do not use the products in corrosive or chemically aggressive environments. Attention: Improper disposal of electronic components can endanger the environment and health. Note: Dispose of electronic components in accordance with local regulations and use appropriate recycling options. Attention: Chemically aggressive media can damage the materials of the products. Note: Do not use the products in corrosive or chemically aggressive environments. Caution: Mechanical shock or bending can damage the products and connected components. Note: Avoid mechanical stress and protect the products from physical influences. Attention: Inadequate fastening can lead to malfunctions and damage. Note: Make sure all products are securely and firmly assembled. Caution: Damaged products may pose safety risks. Note: Check products regularly for visible damage and replace defective parts immediately. Attention: Overloading can lead to overheating and failure of the products. Note: Use the products only within the specified load limits. Attention: Overheating can cause damage to the products and the connected electronic components. Note: Make sure that, for example, heat sinks or components that heat up are adequately ventilated and that the specified temperature ranges are not exceeded.

Table of Contents

Introduction

The KY-002n shock sensor module is based on the *Gaoxin SW-18010P* shock switch. So, the shock sensor in the KY-002n module allows you to use A microcontroller (like in Atmega328p or Raspberry Pi) to detect impacts, shocks or shaking. When the switch detects a shock, the output of the module is set to *LOW*.



The shock switch primarily consists of a terminal that forms a center post and a second terminal, that is a spring that surrounds the center post. When a sufficient force is transferred to the switch, the terminal consisting of the spring moves and connects with the central terminal. The connection between the terminals is momentary and requires a little thinking as you implement it in your project.

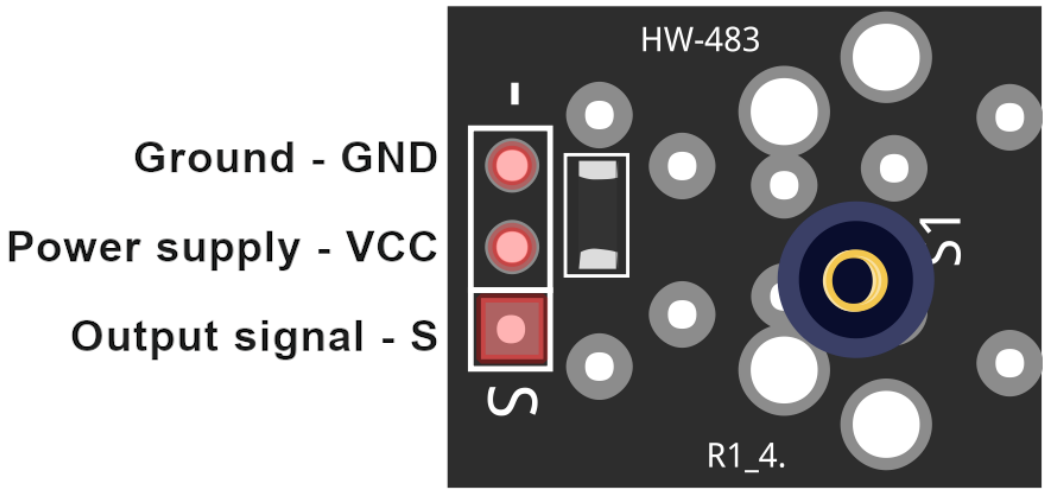
The positioning of the module is also important. Generally speaking, the module should be physically located close to the area that is being monitored. Otherwise, the shock being detected may be dampened by other structural components in your project. An exception to this rule may be where you find that the shock switch on the module is too sensitive for your application. In this case, moving the module further away from the area of interest can make it less sensitive.

Specifications

»	Max voltage across terminals:	12V DC
»	Open switch resistance:	> 10MΩ
»	Closed switch resistance:	< 30Ω
»	Dimensions:	19 x 15 mm [0.73 x 0.6in]

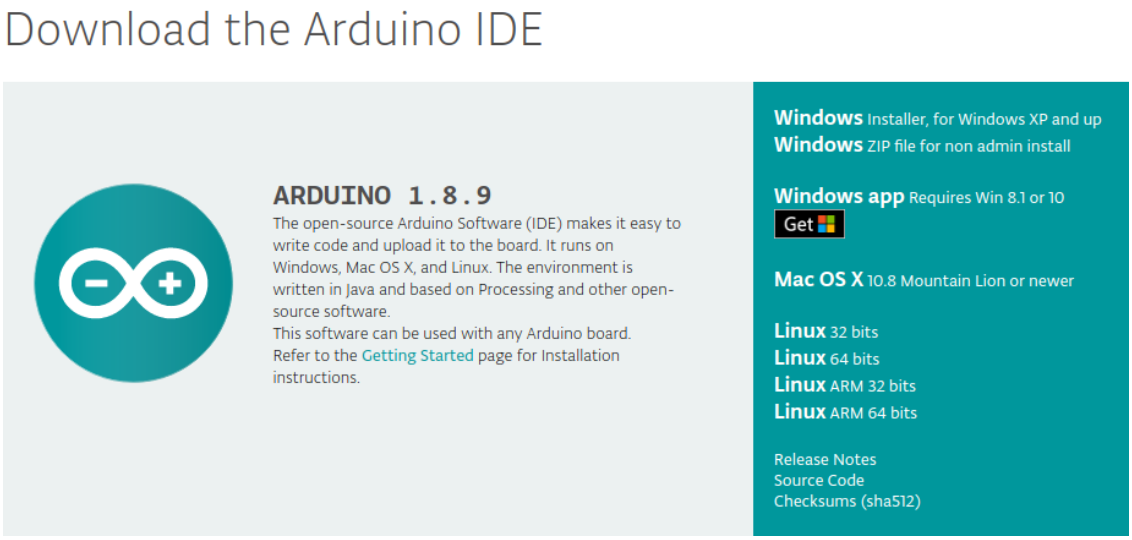
The pinout

The KT-002n shock sensor module has three pins. The pinout diagram 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.



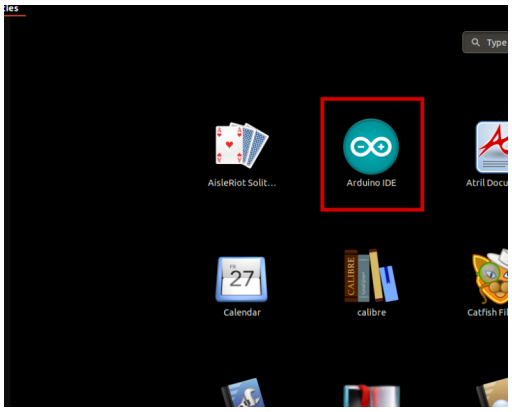
For *Windows* users, double click on the downloaded *.exe* file and follow the instructions in the installation window.

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.

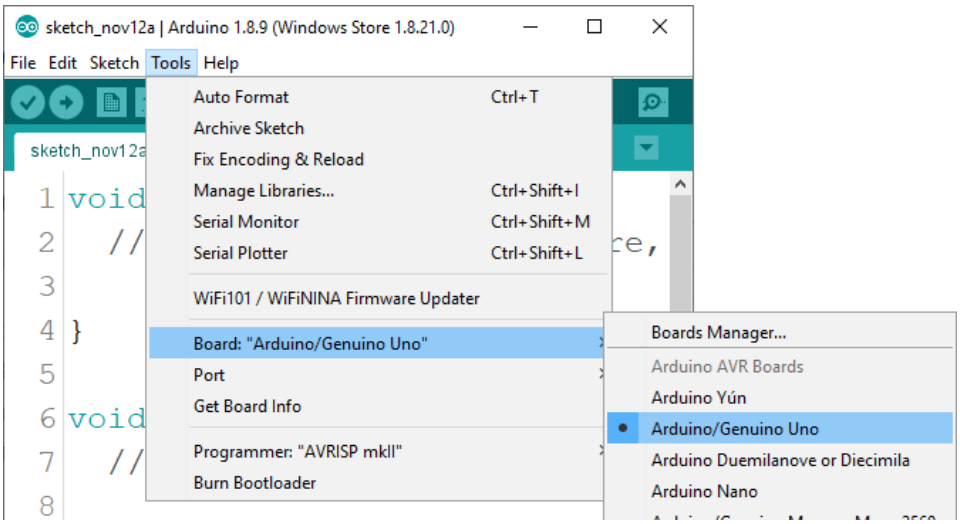
The second script called *install.sh* script has to be used after installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

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



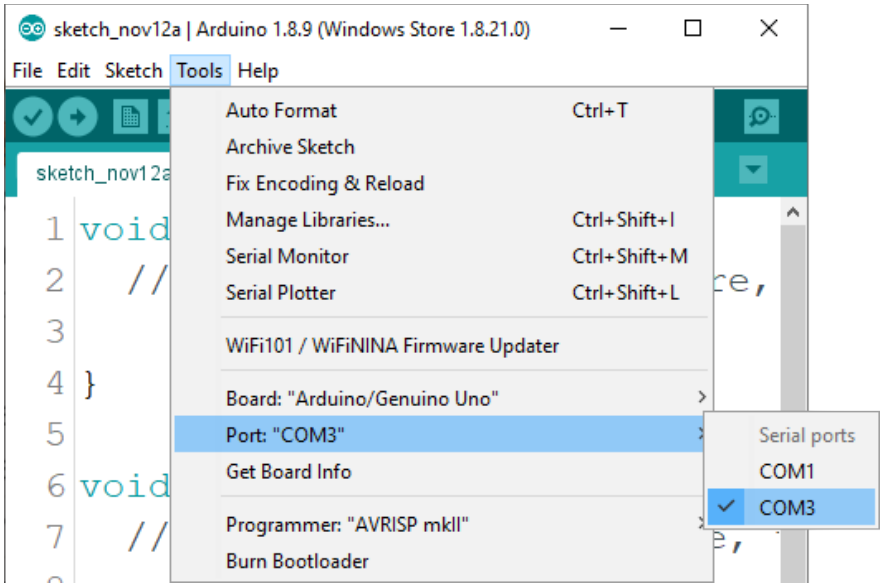
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

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.

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*.

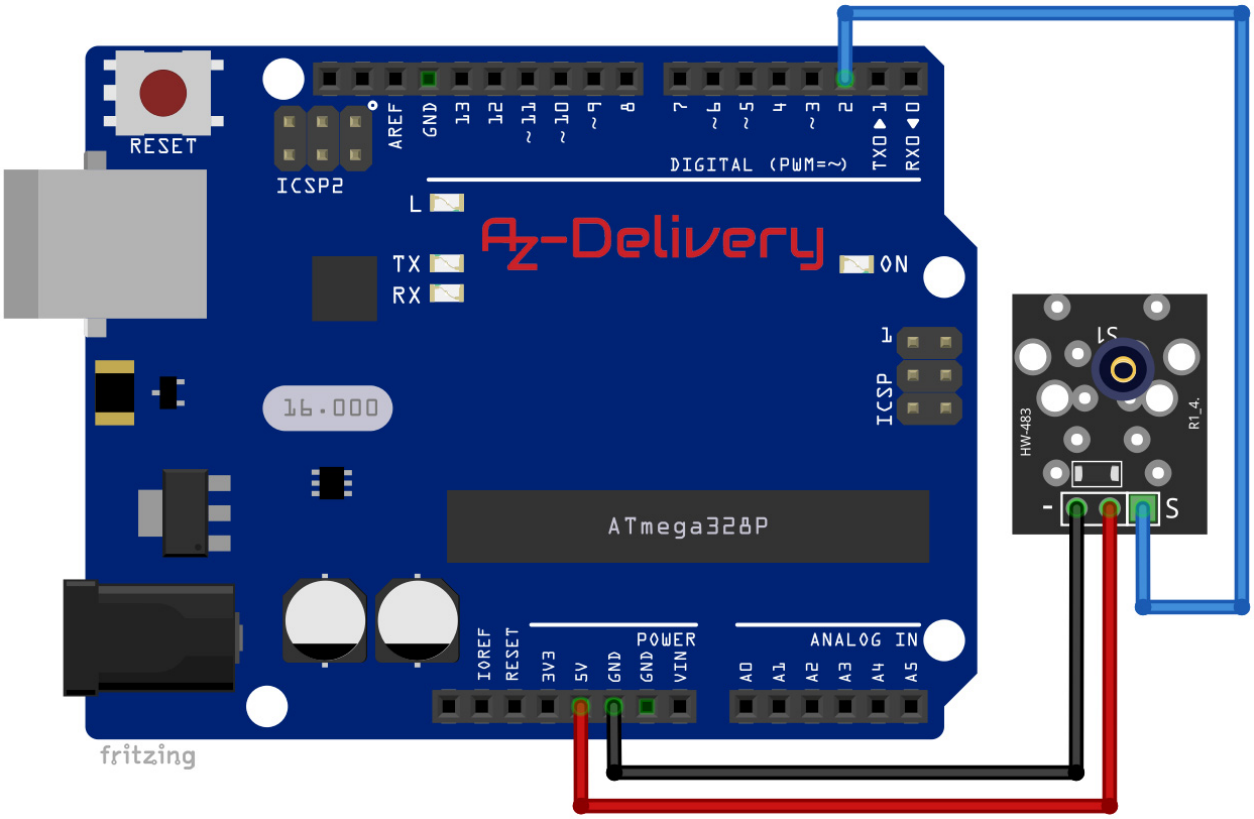
How to set-up the Raspberry Pi and Python

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 sensor with Atmega328p

Connect the KY-002n module with the Atmega328p as shown on the following connection diagram:

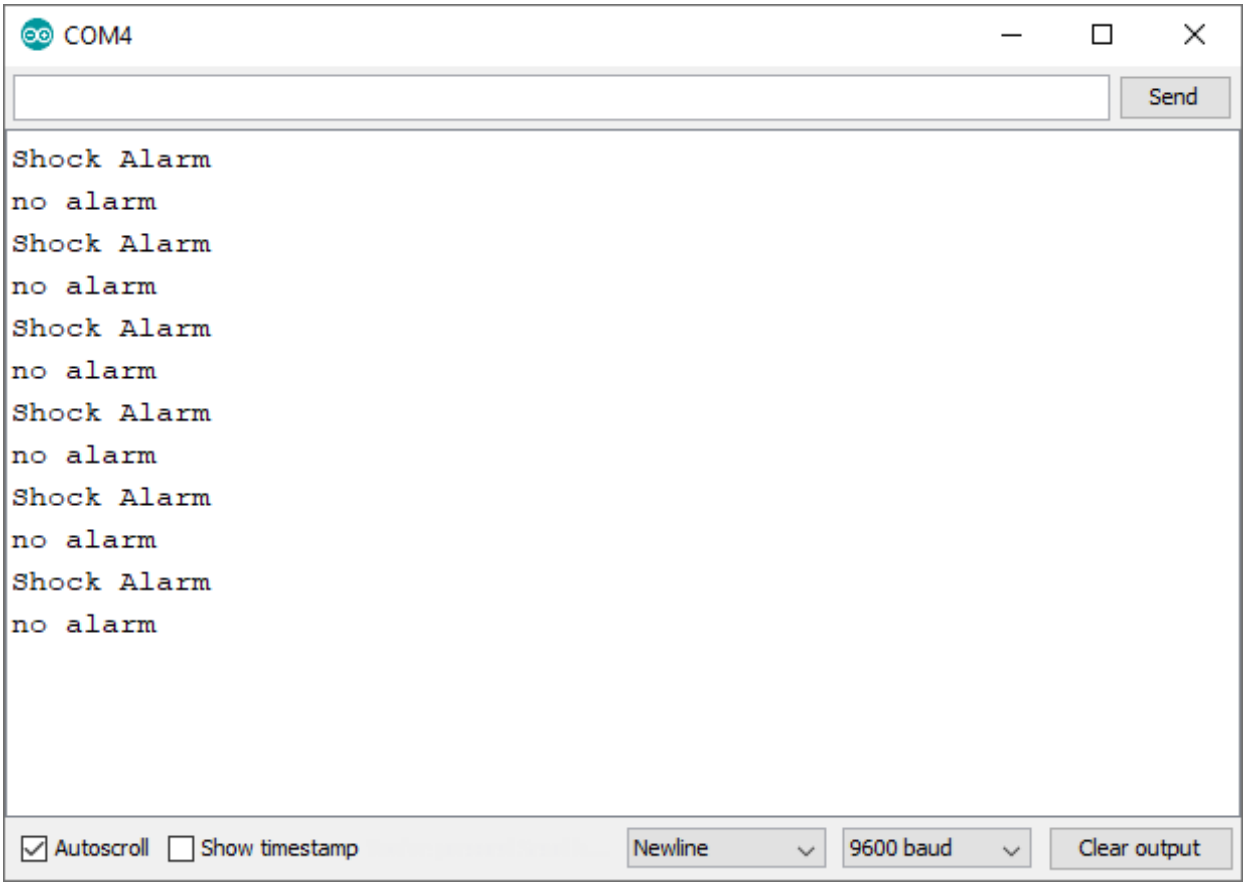


KY-002n pin	**>**	**Mc pin**	
S	>	D2	**Blue wire**
Middle pin (VCC)	>	5V	**Red wire**
- (GND)	>	GND	**Black wire**

Sketch example

```
#define SHOCK_PIN 2
uint8_t shock_value = 1;
boolean alarm = false;
unsigned long last_shock_time;
int shock_alarm_time = 250;
void setup() {
  Serial.begin(9600);
  pinMode(SHOCK_PIN, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  shock_value = digitalRead(SHOCK_PIN);
  if(shock_value == LOW) {
    last_shock_time = millis();
    if(!alarm) {
      Serial.println("Shock Alarm");
      digitalWrite(LED_BUILTIN, HIGH);
      alarm = true;
    }
  }
  else {
    if((millis()-last_shock_time) > shock_alarm_time  &&  alarm){
      Serial.println("no alarm");
      digitalWrite(LED_BUILTIN, LOW);
      alarm = false;
    }
  }
}
```

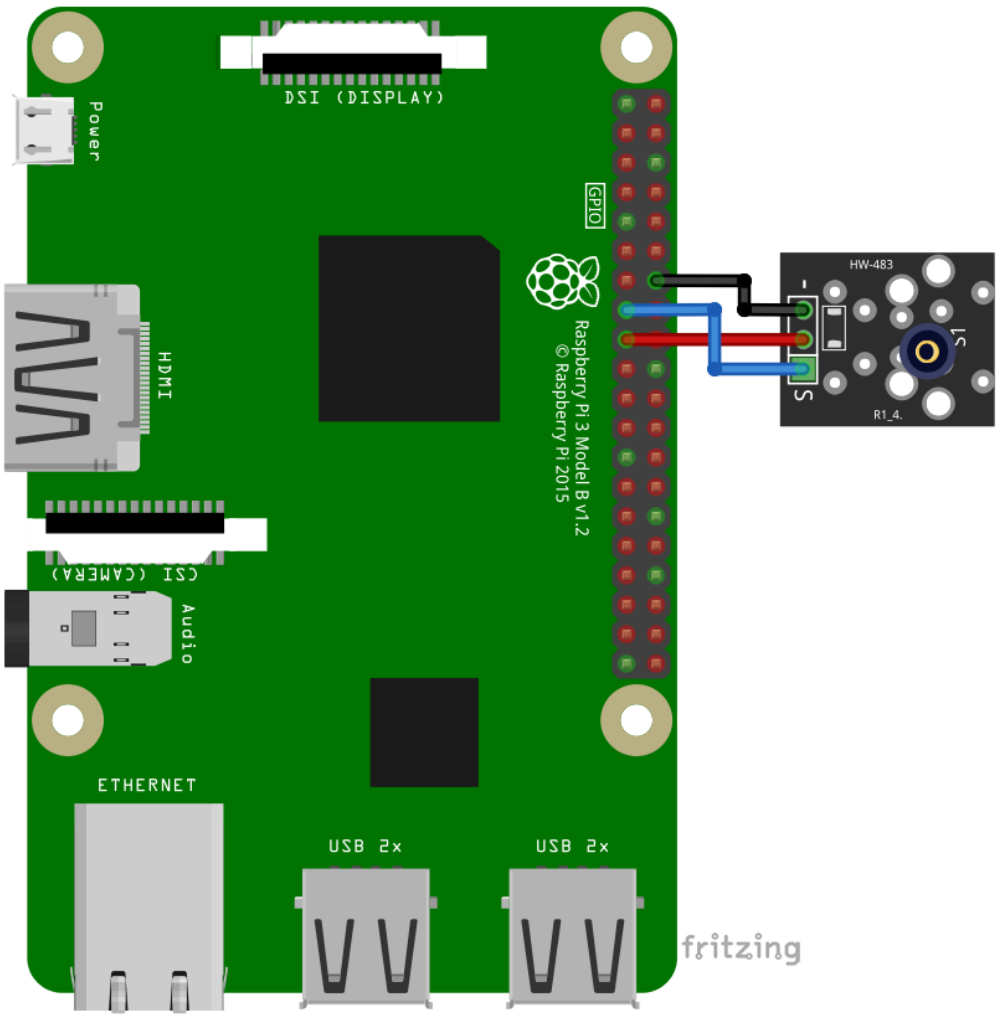
Upload the sketch to the Atmega328p and open Serial Monitor (*`Tools > Serial Monitor`*). The result should look like the output on the following image:



The built-in LED of the Atmega328p should be turned *ON* when there is a *`Shock Alarm`*.

Connecting the sensor with Raspberry Pi

Connect the KY-002n module with the Raspberry Pi as shown on the following connection diagram:



KY-002 pin	**>**	**Raspberry Pi pin**		
S	>	GPIO22	[pin 15]	**Blue wire**
Middle pin (VCC)	>	3V3	[pin 17]	**Red wire**
- (GND)	>	GND	[pin 20]	**Black wire**

Python script

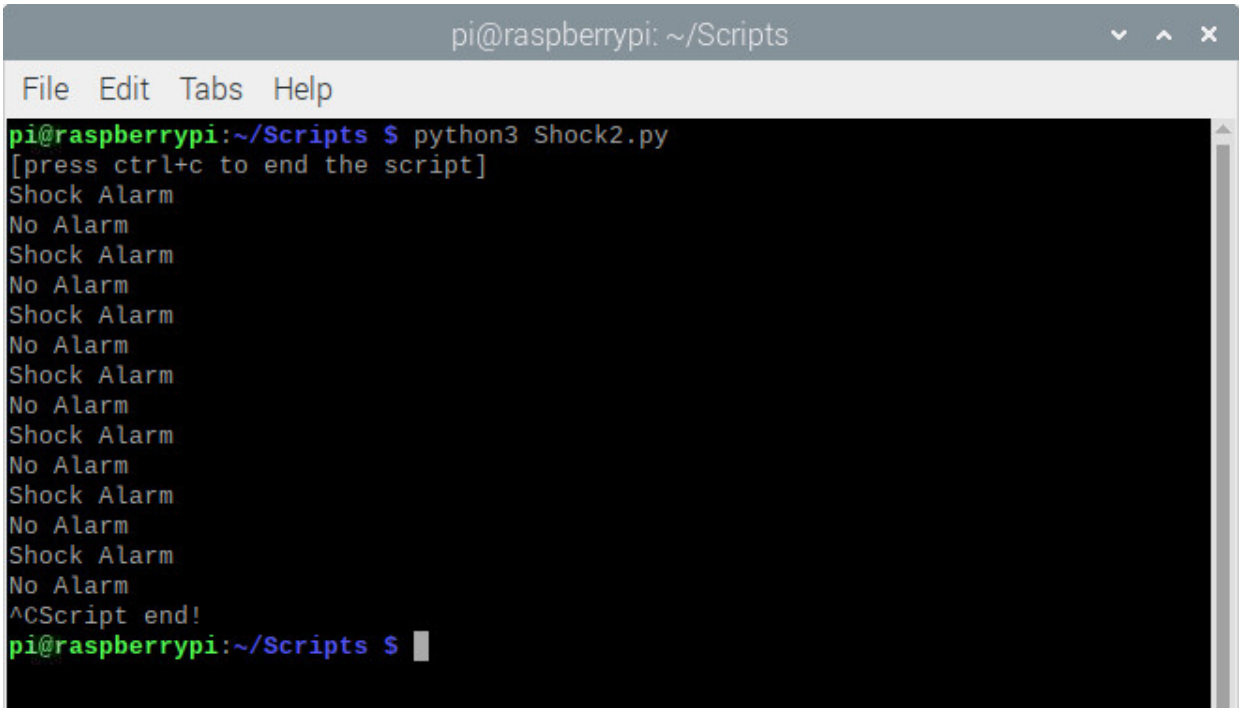
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
Shock_PIN = 22
GPIO.setup(Shock_PIN, GPIO.IN)
alarm = False
lastShockTime = 0.0
shockAlarmTime = 0.05
print('[Press CTRL + C to end the script!]')
try: # Main program loop
     while True:
          shock_state = GPIO.input(Shock_PIN)
          if shock_state == 1:
               lastShockTime = time.time()
               if not alarm:
                    print('Shock Alarm')
                    alarm = True
          else:
               if (time.time()-lastShockTime)>shockAlarmTime and alarm:
                    print('No Alarm')
                    alarm = False

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

Save the script by name *Shock2.py*. To run the script, open terminal in the directory where the script is saved and run the following command:

python3 Shock2.py

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



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 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