Welcome!

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

Have fun!

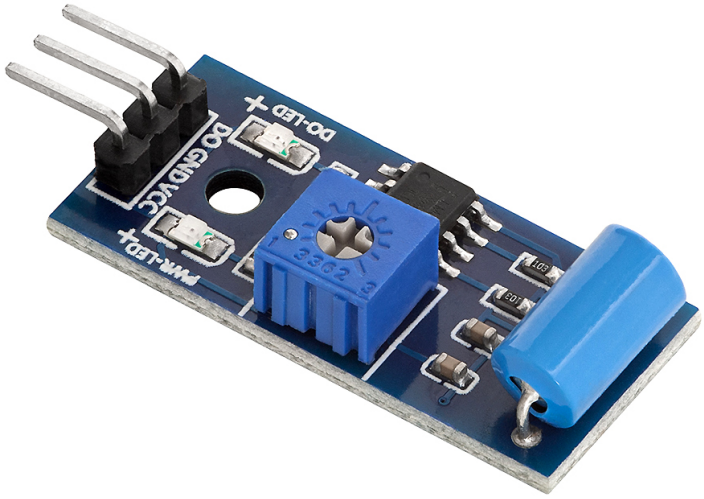


Table of Contents

Introduction

The vibration sensor module is a high sensitivity non-directional vibration sensor device that detects vibrations.

The module is used in a variety of applications such as protection and security systems, object movement detection, earthquake alarms, automotive, motorcycle and bicycle alarms, gaming devices, etc. It can also be used in the industry when vibration detection for certain machines is required.

The module consists of the SW-420 vibration detection sensor, LM393 IC as a comparator, potentiometer and a few passive elements. It has two LEDs for power and detection signaling.

The output signal of the module is a digital signal. When the sensor is idle the pin is in a LOW state. When vibration occurs, the comparator chip sends the output signal and the output signal is in the HIGH state.

Specifications

Operating voltage:	from 3.3V to 5V
Operating current:	20mA
Operating position:	non-directional
Output signal:	Digital
Mounting hole diameter:	3mm
Dimensions:	32x14x7mm (1.2x0.5x0.3in)

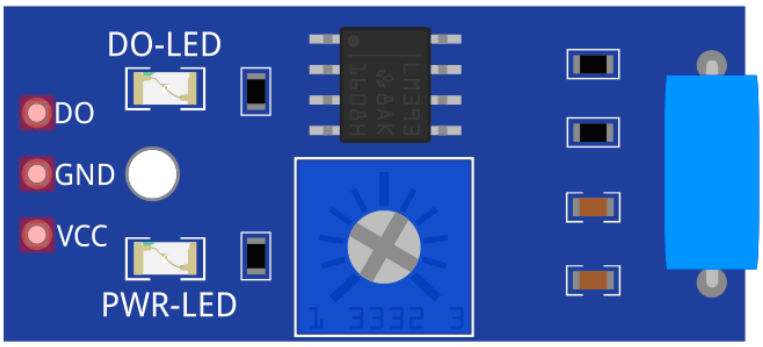
The module has on-board LEDs that are used for power and detection indication.

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 vibration sensor module has three pins. The pinout is shown on the following image:

DIGITAL OUTPUT - D0
GROUND - GND
POWER SUPPLY - VCC



The module operates in both the *3.3V* and *5V* voltage ranges.

NOTE: When using the module with the Raspberry Pi, connect the VCC pin of the module to the *3.3V*. Connecting this pin to the *5V* could damage the 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.

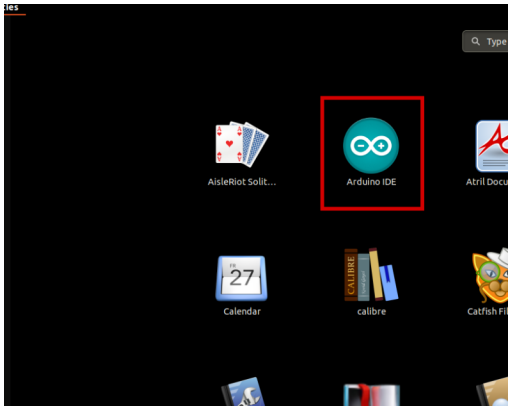
For *Windows* users, double click on the downloaded *.exe* file and follow the instructions in the installation window.

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.

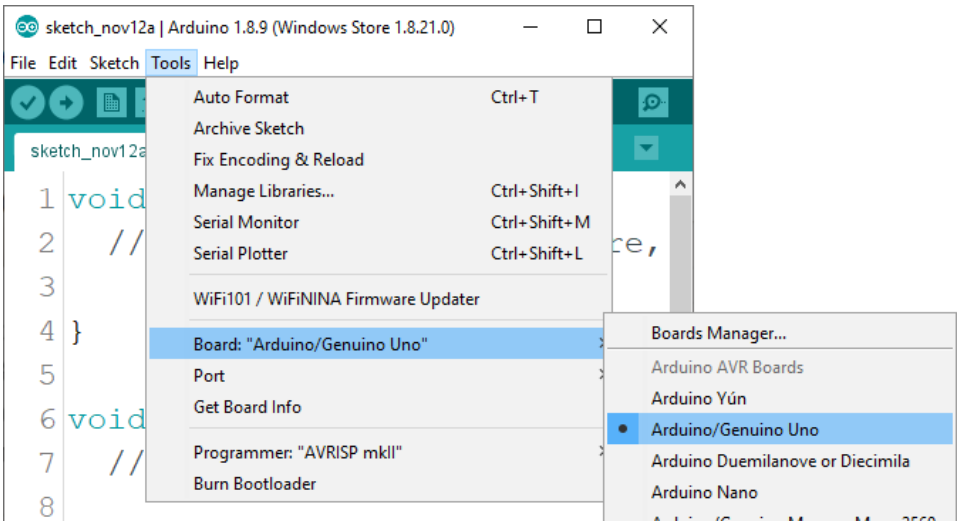
The second script, called *install.sh*, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

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



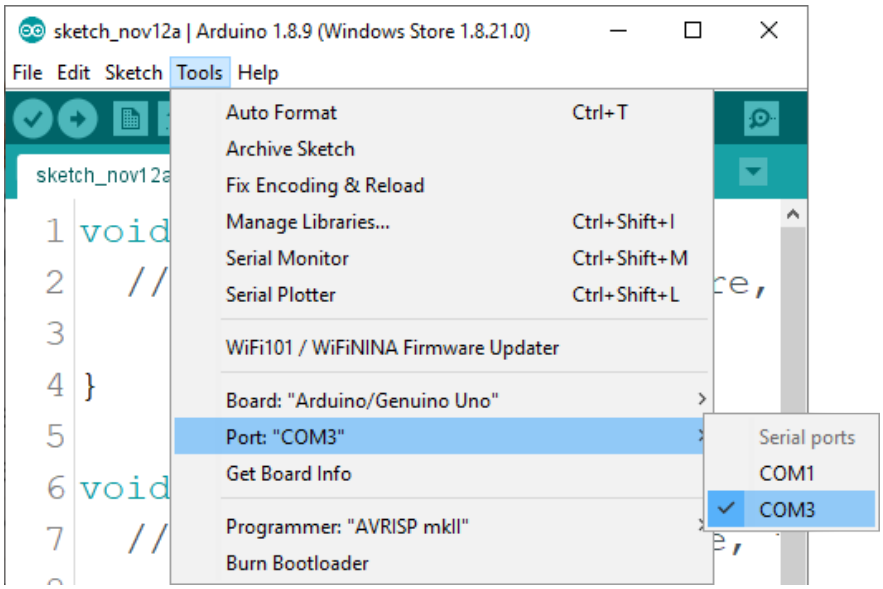
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect an 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*.

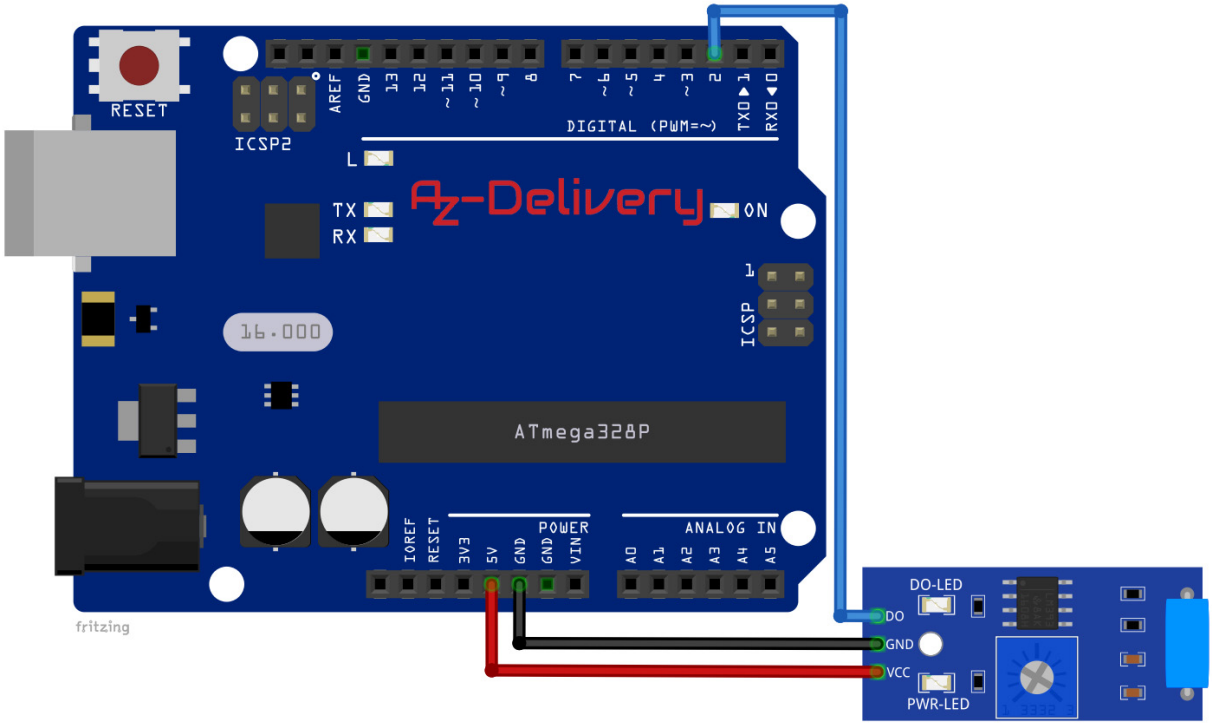
How to set-up the Raspberry Pi and Python

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 module with Atmega328p

Connect the module with the Atmega328p as shown on the following image:

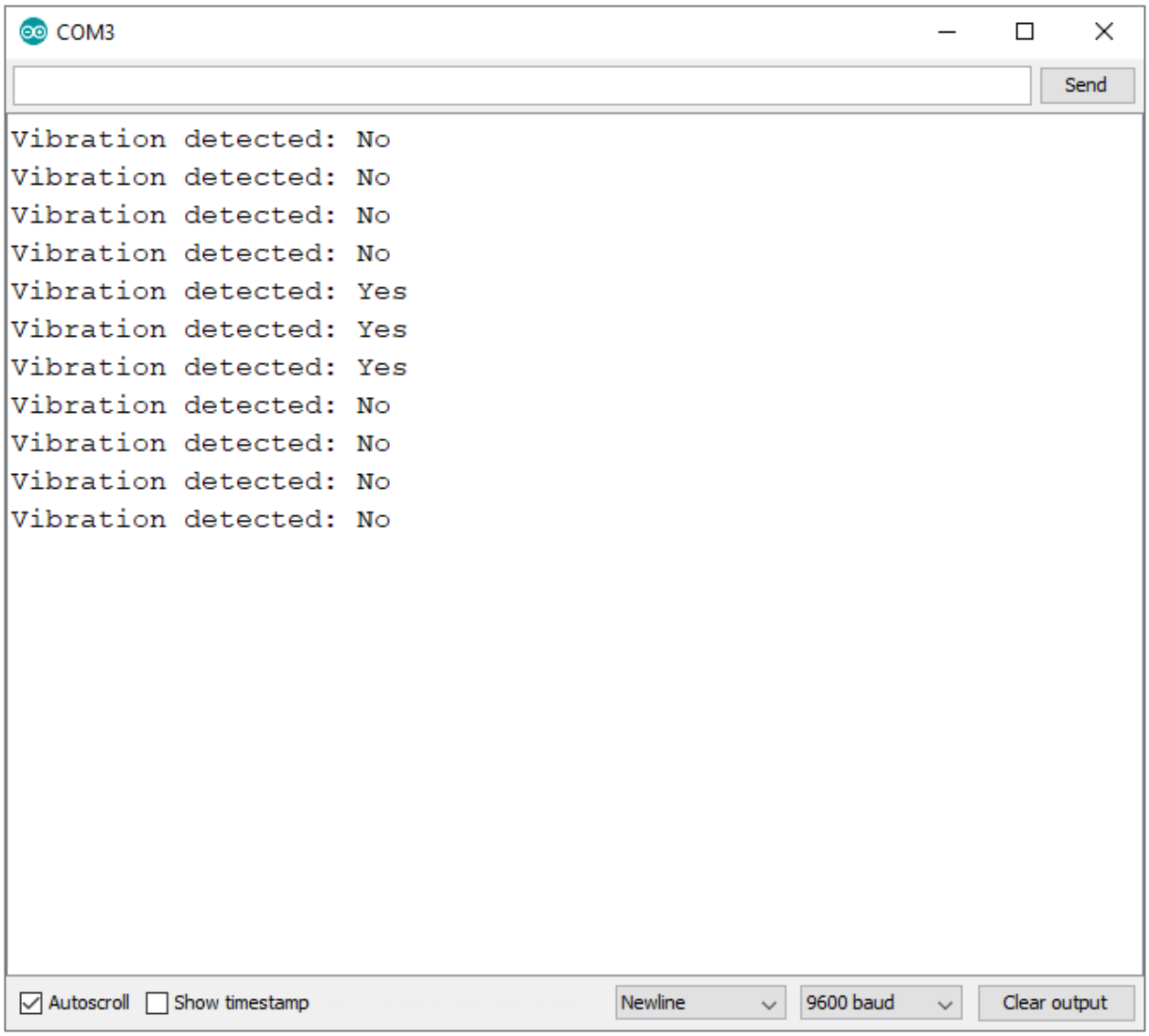


Module pin	Mc pin	Wire color
VCC	5V	**Red wire**
GND	GND	**Black wire**
DO	D2	**Blue wire**

Sketch example

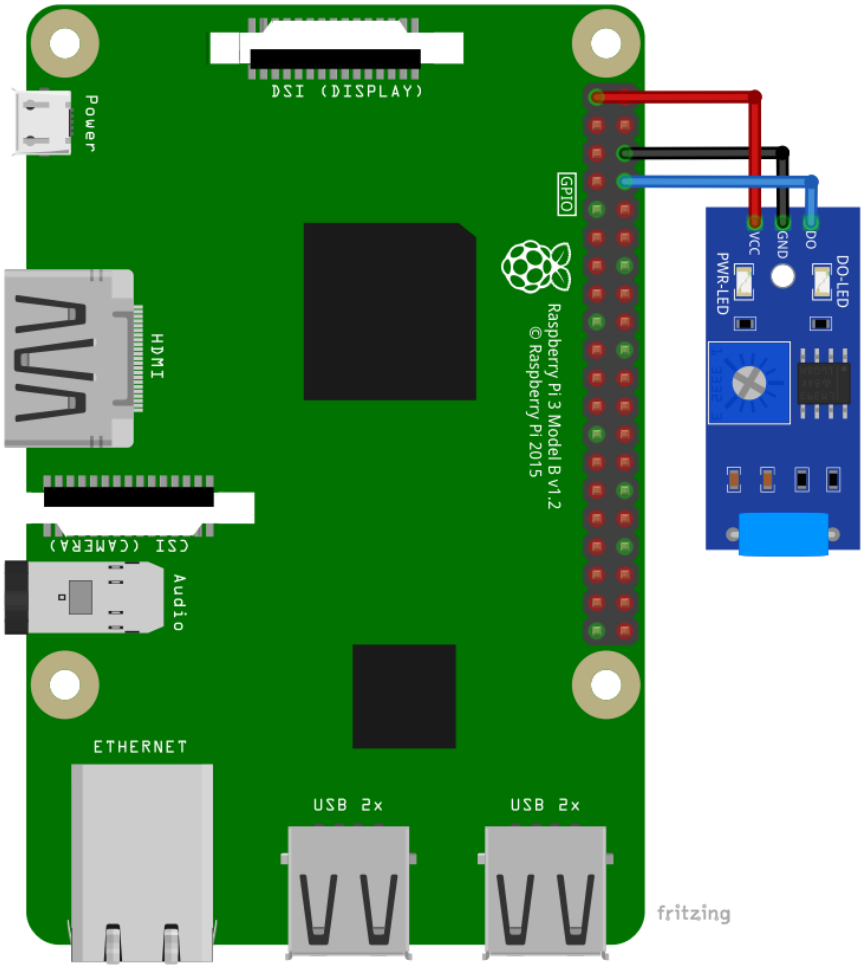
```
#define DIGITAL_PIN 2
boolean vibrate = false;
String vib;
void setup() {
  Serial.begin(9600);
  pinMode(DIGITAL_PIN, INPUT);
}
void loop() {
  vibrate = digitalRead(DIGITAL_PIN);

  if (vibrate) {
    vib = "No";
  }
  else {
    vib = "Yes";
  }
  Serial.print("Vibration detected: ");
  Serial.println(vib);
  delay(2000);
}
```

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



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
VCC	3.3V	1	**Red wire**
GND	GND	6	**Black wire**
DO	GPIO14	8	**Blue wire**

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:

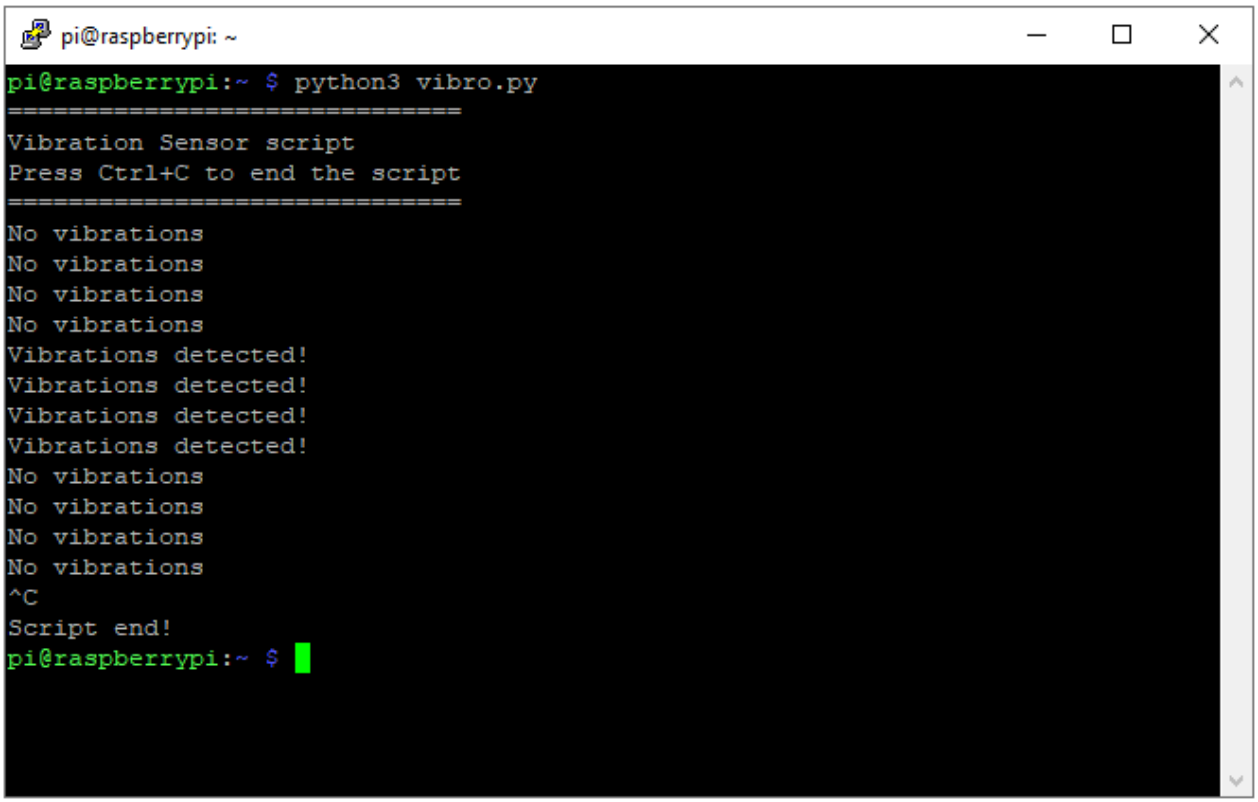
```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install python3-rpi.gpio
```

Python script

```
import time
import RPi.GPIO as GPIO

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

DIGITALOUT = 14
GPIO.setup(DIGITALOUT, GPIO.IN)
time.sleep(2)

print('Vibration Sensor script')
print('[Press CTRL + C to end the script!]')
try: # Main program loop
  while True:
    if GPIO.input(DIGITALOUT)==0:
        print('Vibrations detected!')
        time.sleep(2)
    else:
        print('No vibrations')
        time.sleep(2)

except KeyboardInterrupt:
      print('\nScript end!')
finally:
      GPIO.cleanup()
```

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

The result should look like as 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