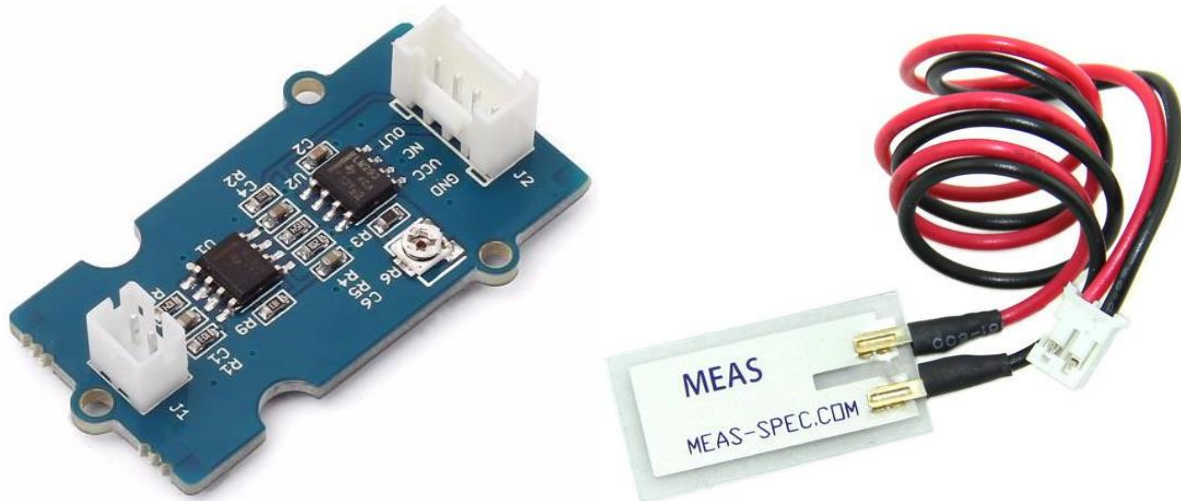


Grove - Piezo Vibration Sensor



Grove-Piezo Vibration Sensor is suitable for measurements of flexibility, vibration, impact and touch. The module is based on PZT film sensor LDT0-028. When the sensor moves back and forth, a certain voltage will be generated by the voltage comparator inside of it. A wide dynamic range (0.001Hz~1000MHz) guarantees an excellent measuring performance. And, you can adjust its sensitivity by adjusting the on-board potentiometer with a screw.

Version

Product Version	Changes	Released Date
Grove - Piezo Vibration Sensor V1.1	Initial	Jul 2014



Features

- Standard grove socket
- Wide dynamic range : 0.1Hz~180Hz
- Adjustable sensitivity
- High receptivity for strong impact

Tip

More details about Grove modules please refer to [Grove System](#)

Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE
				

Caution

The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Applications

- Vibration Sensing in Washing Machine
- Low Power Wakeup Switch
- Low Cost Vibration Sensing
- Car Alarms
- Body Movement
- Security Systems

Getting Started

Play With Arduino

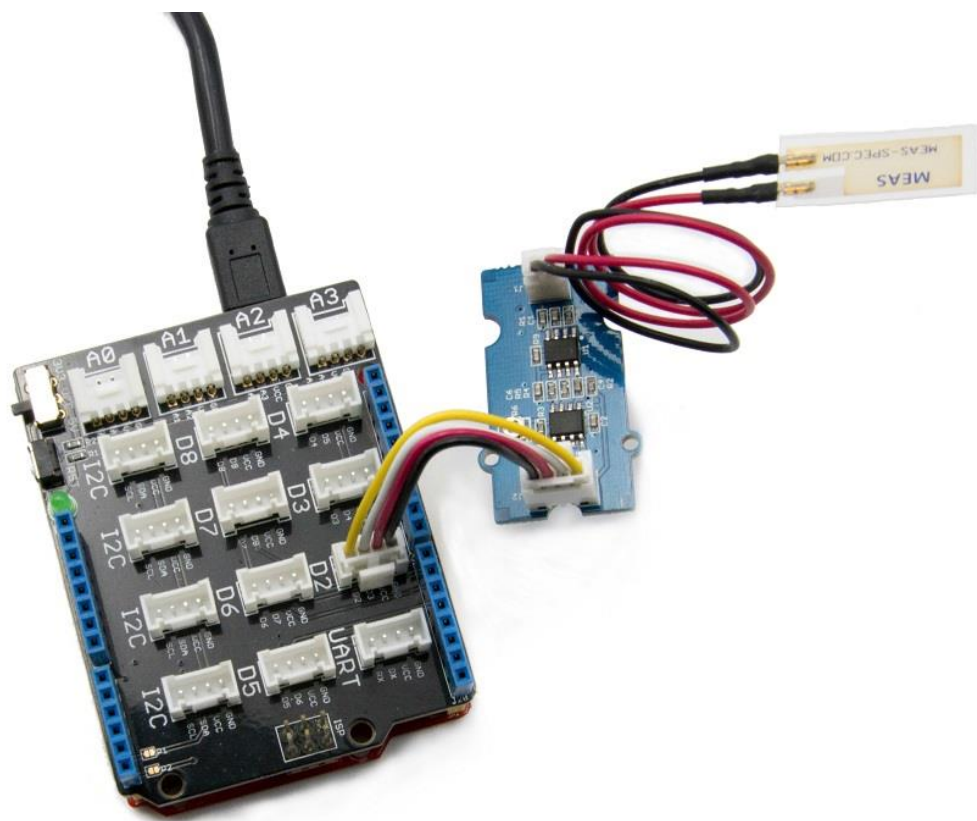
Hardware

The Grove - Piezo Vibration Sensor outputs a logic HIGH when vibration was detected. We can use any of Arduino pins to read the data. Here is an example of Piezo Vibration Sensor controlling LED. When the vibration was detected, this sensor outputs a logic high signal (the sensitivity can be changed by adjusting the potentiometer), an LED lights up.

- Step 1. Prepare the below stuffs:

Seeduino V4	Base Shield	Grove - Piezo Vibration
		

- Step 2. Connect the module to the **D2** of base shield using the 4-pin grove cable, we use **digital pin13 on board LED** as output.
- Step 3. Plug the Basic Shield into Arduino.
- Step 4. Connect Arduino to PC by using a USB cable.



Note

It may output low level even though originally output high level when you increase the threshold voltage by clockwise adjusting the potentiometer.

Software

- Step 1. Copy and paste code below to a new Arduino sketch.

```
1const int ledPin=13;
2void setup() {
3  Serial.begin(9600);
4  pinMode(ledPin,OUTPUT);
5}
6
7void loop() {
8  int sensorState = digitalRead(2);
9  Serial.println(sensorState);
10 delay(100);
11 if(sensorState == HIGH)
12 {
13   digitalWrite(ledPin,HIGH);
14 }
15 else
16 {
17   digitalWrite(ledPin,LOW);
18 }
19}
```

- Step 2. The LED will be on when vibration is detected.

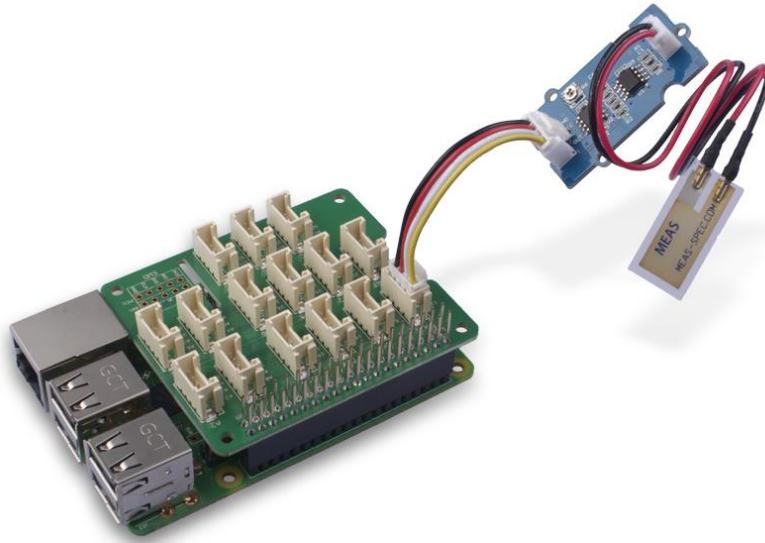
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

Hardware

- **Step 1.** Things used in this project:

Raspberry pi	Grove Base Hat for RasPi	Grove - Piezo Vibration
		

- **Step 2.** Plug the Grove Base Hat into Raspberry.
- **Step 3.** Connect the Grove - Piezo Vibration Sensor to port 12 of the Base Hat.
- **Step 4.** Connect the Raspberry Pi to PC through USB cable.



Note

For step 3 you are able to connect the piezo vibration sensor to **any GPIO Port** but make sure you change the command with the corresponding port number.

Software

- **Step 1.** Follow [Setting Software](#) to configure the development environment.
- **Step 2.** Download the source file by cloning the grove.py library.

```
1cd ~  
2git clone https://github.com/Seeed-Studio/grove.py
```

- **Step 3.** Execute below commands to run the code.

```
1cd grove.py/grove  
2python grove_piezo_vibration_sensor.py 12
```

Following is the grove_piezo_vibration_sensor.py code.

```
1import time  
2from grove.gpio import GPIO  
3  
4  
5class GrovePiezoVibrationSensor(GPIO):  
6    def __init__(self, pin):  
7        super(GrovePiezoVibrationSensor, self).__init__(pin, GPIO.IN)  
8        self._on_detect = None  
9  
10    @property  
11    def on_detect(self):  
12        return self._on_detect
```

```

13
14 @on_detect.setter
15 def on_detect(self, callback):
16     if not callable(callback):
17         return
18
19     if self.on_event is None:
20         self.on_event = self._handle_event
21
22     self._on_detect = callback
23
24 def _handle_event(self, pin, value):
25     if value:
26         if callable(self._on_detect):
27             self._on_detect()
28
29 Grove = GrovePiezoVibrationSensor
30
31
32 def main():
33     import sys
34
35     if len(sys.argv) < 2:
36         print('Usage: {} pin'.format(sys.argv[0]))
37         sys.exit(1)
38
39     pir = GrovePiezoVibrationSensor(int(sys.argv[1]))
40
41     def callback():
42         print('Detected.')
43
44     pir.on_detect = callback
45
46     while True:
47         time.sleep(1)
48
49
50 if __name__ == '__main__':
51     main()

```

Success

If everything goes well, you will be able to see the following result

```

1 pi@raspberrypi:~/grove.py/grove $ python grove_piezo_vibration_sensor.py 12
2 Detected.
3 Detected.
4 Detected.
5 Detected.
6 Detected.
7 Detected.
8 Detected.
9 Detected.
10 ^C Traceback (most recent call last):

```



```
11 File "grove_piezo_vibration_sensor.py", line 84, in <module>
12   main()
13 File "grove_piezo_vibration_sensor.py", line 80, in main
14   time.sleep(1)
15 KeyboardInterrupt
```

You can quit this program by simply press `Ctrl + C`.

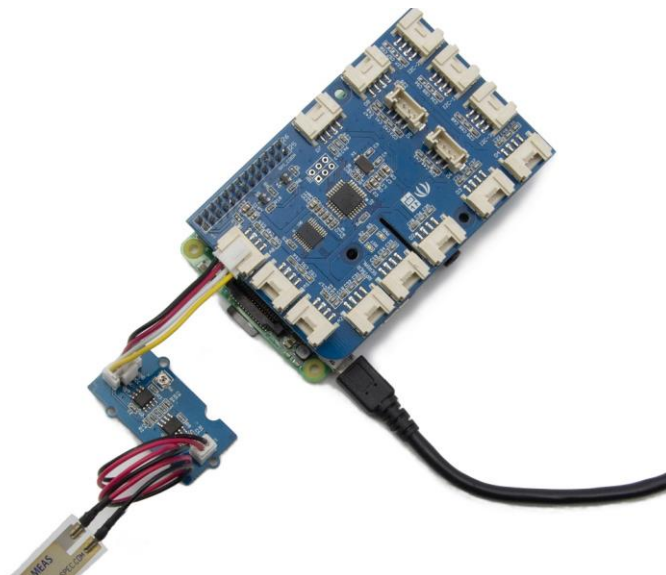
Play With Raspberry Pi (with GrovePi_Plus)

Hardware

- Step 1. Prepare the below stuffs:
-

Raspberry pi	GrovePi_Plus	Grove - Piezo Vibration
		

- Step 2. Plug the GrovePi_Plus into Raspberry.
- Step 3. Connect Grove-Piezo Vibration to A0 port of GrovePi_Plus.
- Step 4. Connect the Raspberry to PC through USB cable.



Software

- Step 1. Follow [Setting Software](#) to configure the development environment.
- Step 2. Git clone the Github repository.

```
1cd ~
2git clone https://github.com/DexterInd/GrovePi.git
```

- Step 3. Execute below commands to detect the vibration.

```
1cd ~/GrovePi/Software/Python
2python grove_piezo_vibration_sensor.py
```

Here is the grove_piezo_vibration_sensor.py code.

```
1import time
2import grovepi
3
4# Connect the Grove Piezo Vibration Sensor to analog port A0
5# OUT,NC,VCC,GND
6piezo = 0
7
8grovepi.pinMode(piezo,"INPUT")
9
10while True:
11    try:
12        # When vibration is detected, the sensor outputs a logic high signal
13        print grovepi.analogRead(piezo)
14        time.sleep(.5)
15
16    except IOError:
17        print "Error"
```

- Step 4. We will see the vibration display on terminal as below.

```
1pi@raspberrypi:~/GrovePi/Software/Python $ python grove_piezo_vibration_sensor.py
21023
31023
41023
51023
618
717
818
917
```


Note

We also can use `grovepi.digitalRead(2)` to read the vibration status with attaching the sensor to D2 port of GrovePi.

FAQs

Q1: Is it digital or analog output?

A1: It is digital output, Low or High.

Resources

- [\[Eagle\] Grove - Piezo Vibration Sensor Eagle File](#)
- [\[PDF\] Grove - Piezo Vibration Sensor Schematic PDF File](#)
- [\[PDF\] Grove - Piezo Vibration Sensor PCB PDF File](#)
- [\[Datasheet\] Piezo Vibration Sensor Datasheet](#)

Projects

Grove Starter Kit For Arduino - Piezo Vibration Sensor: Teaches you how to use the Piezo vibration sensor in the Arduino Grove starter kit.

Seat Monitor: Using ARTIK cloud to monitor cabin seat state.

Tech Support

Please submit any technical issue into our [forum](#) or drop mail to techsupport@seeed.cc.