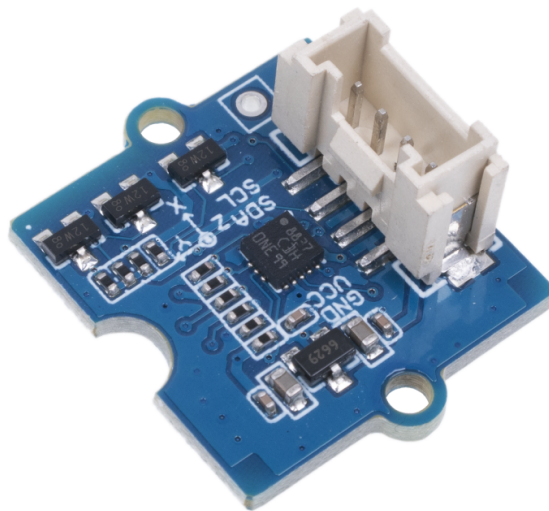


Grove - 3-Axis Digital Accelerometer (LIS3DHTR)



Grove - 3-Axis Digital Accelerometer(LIS3DHTR) is a low-cost 3 - Axis accelerometer in a bundle of Grove products. It is based on the LIS3DHTR chip which provides multiple ranges and interfaces selection. You can never believe that such a tiny 3 - Axis accelerometer can support I2C, SPI, and ADC GPIO interfaces,

which means you can choose any way to connect with your development board. Besides, this accelerometer can also monitor the surrounding temperature to tune the error caused by it.

Get One Now 

[<https://www.seeedstudio.com/Grove-3-Axis-Digital-Accelerometer-LIS3DHTR-p-4533.html>]



Features

- Measurement range: $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$, multiple ranges selection.
- Multiple interfaces option: Grove I2C interface, SPI interface, ADC interface.
- Temperature adjustable: able to adjust and tune the error caused by temperature.
- $\frac{3}{5}V$ power supply.

Specification

Item	Value
Power Supply	$\frac{3}{5}V$
Interfaces	I2C/SPI/GPIO ADC
I2C address	Default 0x19, can be changed to 0x18 when connecting SDO Pin with GND
ADC GPIO Power input	0-3.3V
Interruption	An interruption Pin reserved
SPI Mode set up	Connect the CS Pin with GND

Platform Supported

Arduino	Raspberry Pi		
			

Getting Started

Play with Seeeduno XIAO based on Grove I2C interface

Materials Required

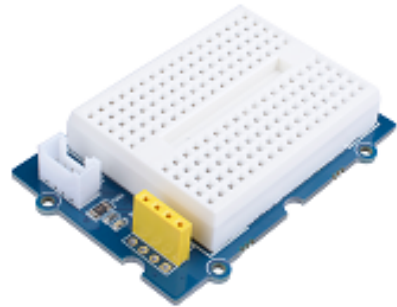
Seeeduno XIAO



[Get one now](#)

[<https://www.seeedstudio.com/Seeeduno-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html>]

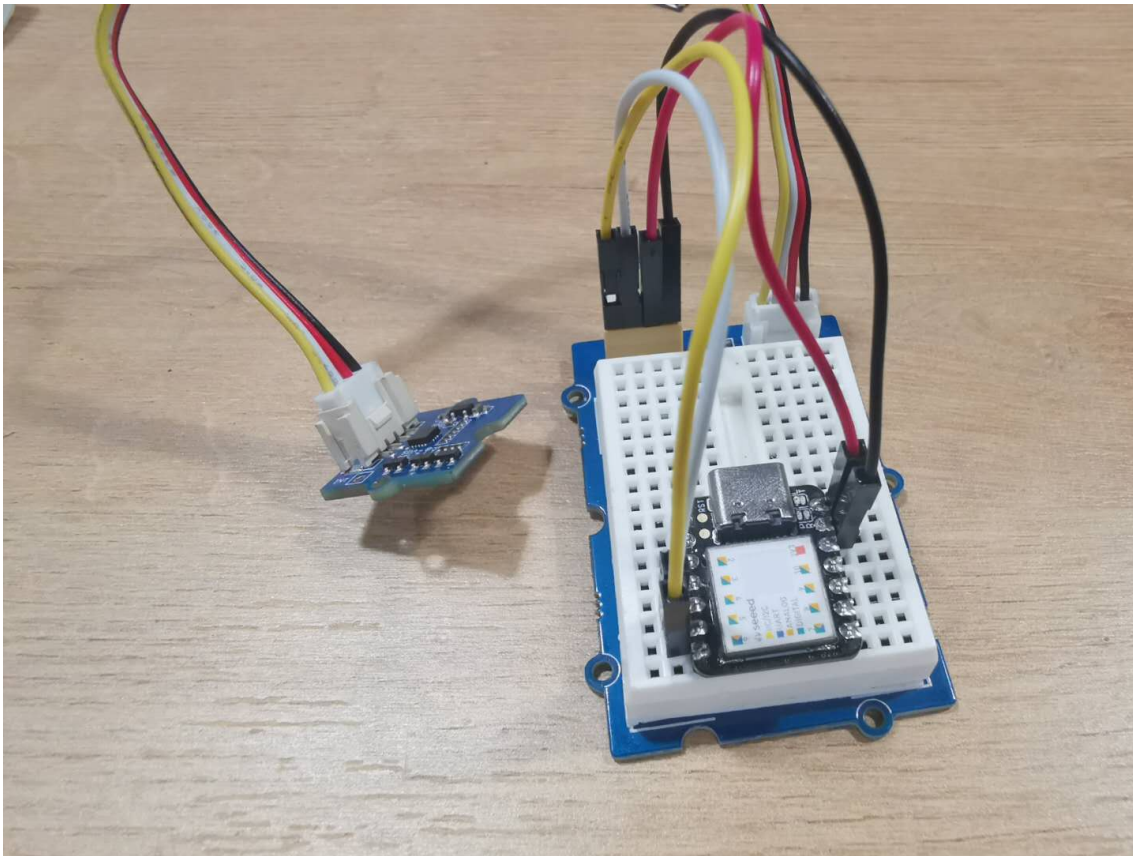
Grove Breadboard



[Get one now](#)

[<https://www.seeedstudio.com/Grove-Breadboard-p-4034.html>]

Hardware connection



Connect the Grove - 3-Axis Digital Accelerometer (LIS3DHTR) with Seeeduino XIAO's I2C interface.

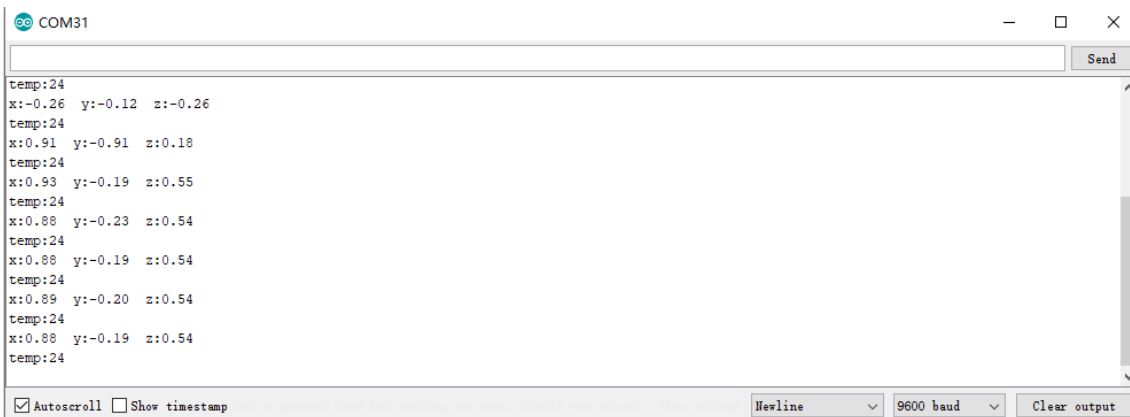
Software Code

```
1 // This example use I2C.
2 #include "LIS3DHTR.h"
3 #include <Wire.h>
4 LIS3DHTR<TwoWire> LIS; //IIC
5 #define WIRE Wire
6
7 void setup()
8 {
9   Serial.begin(115200);
10  while (!Serial)
11  {
12  };
13  LIS.begin(WIRE,0x19); //IIC init
14  //LIS.begin(0x19);
```



```
15 LIS.openTemp(); //If ADC3 is used, the temperature de
16 // LIS.closeTemp();//default
17 delay(100);
18 LIS.setFullScaleRange(LIS3DHTR_RANGE_2G);
19 // LIS.setFullScaleRange(LIS3DHTR_RANGE_4G);
20 // LIS.setFullScaleRange(LIS3DHTR_RANGE_8G);
21 // LIS.setFullScaleRange(LIS3DHTR_RANGE_16G);
22 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_1HZ);
23 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_10HZ);
24 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_25HZ);
25 LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ);
26 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_100HZ);
27 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_200HZ);
28 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_1_6KHZ);
29 // LIS.setOutputDataRate(LIS3DHTR_DATARATE_5KHZ);
30 }
31 void loop()
32 {
33   if (!LIS)
34   {
35     Serial.println("LIS3DHTR didn't connect.");
36     while (1)
37       ;
38     return;
39   }
40   //3 axis
41   Serial.print("x:"); Serial.print(LIS.getAcceleration);
42   Serial.print("y:"); Serial.print(LIS.getAcceleration);
43   Serial.print("z:"); Serial.println(LIS.getAccelerati
44   //ADC
45   // Serial.print("adc1:"); Serial.println(LIS.readbi
46   // Serial.print("adc2:"); Serial.println(LIS.readbi
47   // Serial.print("adc3:"); Serial.println(LIS.readbi
48
49   //temperature
50   Serial.print("temp:");
51   Serial.println(LIS.getTemperature());
52   delay(500);
53 }
```

- **Step 1** Download the [library](https://github.com/Seeed-Studio/Seeed_Arduino_LIS3DHTR/archive/master.zip) [https://github.com/Seeed-Studio/Seeed_Arduino_LIS3DHTR/archive/master.zip] from Github and add the "zip" library to your Arduino IDE. Please refer to [How to install an Arduino Library](https://wiki.seeedstudio.com/How_to_install_Arduino_Library/) [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/].
- **Step 2** Find the example code of "LIS3DHTR_IIC" and upload it to your board. Please refer to [How to upload code](https://wiki.seeedstudio.com/Upload_Code/) [https://wiki.seeedstudio.com/Upload_Code/].
- **Step 3** After uploading the code, you will see the accelerated velocity of each axis and temperature of surrounding from the serial monitor.



The screenshot shows a serial monitor window titled "COM31" with a "Send" button in the top right. The output text is as follows:

```
temp:24
x:-0.26 y:-0.12 z:-0.26
temp:24
x:0.91 y:-0.91 z:0.18
temp:24
x:0.93 y:-0.19 z:0.55
temp:24
x:0.88 y:-0.23 z:0.54
temp:24
x:0.88 y:-0.19 z:0.54
temp:24
x:0.89 y:-0.20 z:0.54
temp:24
x:0.88 y:-0.19 z:0.54
temp:24
```

At the bottom of the window, there are controls: a checked "Autoscroll" checkbox, an unchecked "Show timestamp" checkbox, a "Newline" dropdown menu, a "9600 baud" dropdown menu, and a "Clear output" button.

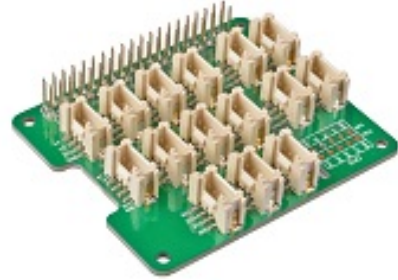
Play with Raspberry Pi

Materials Required

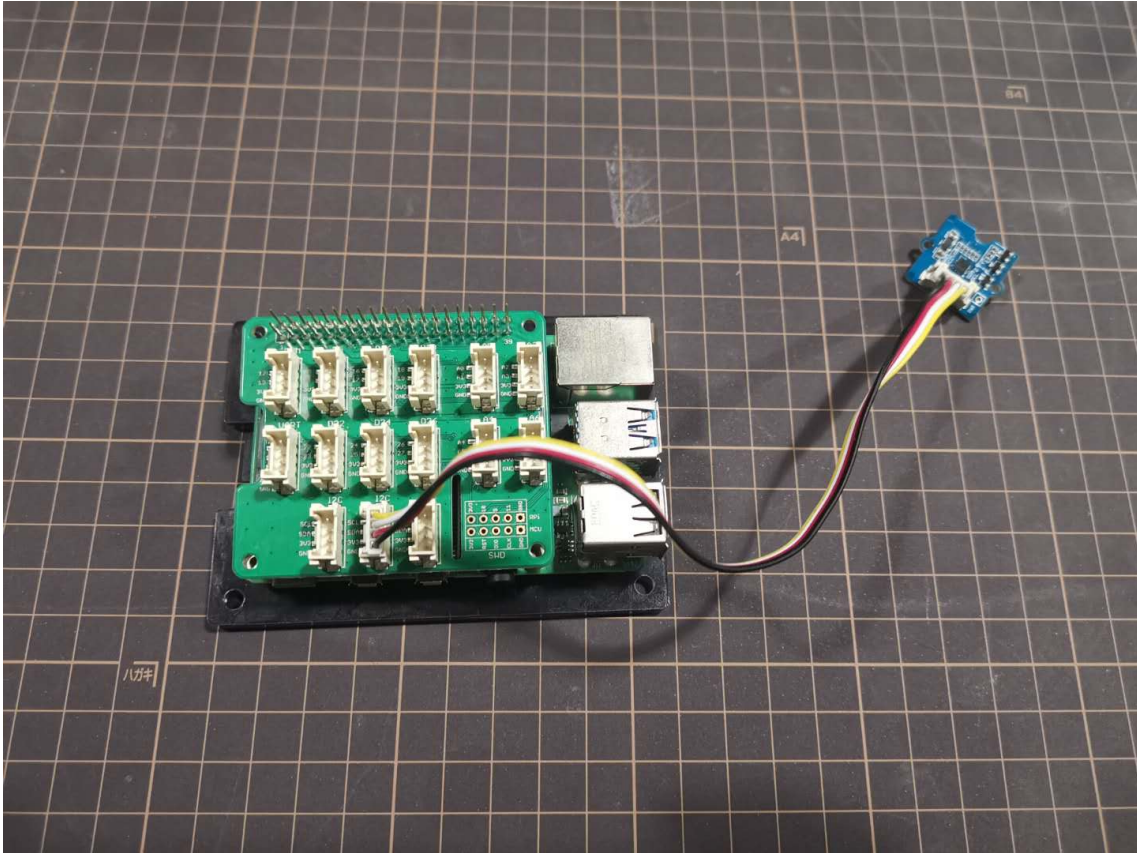
Raspberry Pi 4B(4GB)

[Get one now](https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html)[\[https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html\]](https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html)

Grove Base Hat for Raspberry Pi

[Get one now](https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi.html)[\[https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi.html\]](https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi.html)

Hardware Connection



Connect the LIS3DHTR sensor to any I2C interface on the Grove Base Hat for Raspberry Pi.

Code

```
1  #!/usr/bin/env python
2
3  # Distributed with a free-will license.
4  # Use it any way you want, profit or free, provided it
5  # LIS3DHTR
6  # This code is designed to work with the LIS3DHTR_I2CS
7  # https://www.controleverything.com/content/Acceleromet
8
9  import smbus
10 import time
11
12 # Get I2C bus
13 bus = smbus.SMBus(1)
14
```

```

15 # I2C address of the device
16 LIS3DHTR_DEFAULT_ADDRESS = 0x19
17
18 # LIS3DHTR Register Map
19 LIS3DHTR_REG_WHOAMI = 0x0F # Who Am I Register
20 LIS3DHTR_REG_CTRL1 = 0x20 # Control Register 1
21 LIS3DHTR_REG_CTRL2 = 0x21 # Control Register 2
22 LIS3DHTR_REG_CTRL3 = 0x22 # Control Register 3
23 LIS3DHTR_REG_CTRL4 = 0x23 # Control Register 4
24 LIS3DHTR_REG_CTRL5 = 0x24 # Control Register 5
25 LIS3DHTR_REG_CTRL6 = 0x25 # Control Register 6
26 LIS3DHTR_REG_REFERENCE = 0x26 # Reference Register
27 LIS3DHTR_REG_STATUS = 0x27 # Status Register
28 LIS3DHTR_REG_OUT_X_L = 0x28 # X-Axis LSB
29 LIS3DHTR_REG_OUT_X_H = 0x29 # X-Axis MSB
30 LIS3DHTR_REG_OUT_Y_L = 0x2A # Y-Axis LSB
31 LIS3DHTR_REG_OUT_Y_H = 0x2B # Y-Axis MSB
32 LIS3DHTR_REG_OUT_Z_L = 0x2C # Z-Axis LSB
33 LIS3DHTR_REG_OUT_Z_H = 0x2D # Z-Axis MSB
34
35 # Accl Datarate configuration
36 LIS3DHTR_ACCL_DR_PD = 0x00 # Power down
37 LIS3DHTR_ACCL_DR_1 = 0x10 # ODR = 1 Hz
38 LIS3DHTR_ACCL_DR_10 = 0x20 # ODR = 10 Hz
39 LIS3DHTR_ACCL_DR_25 = 0x30 # ODR = 25 Hz
40 LIS3DHTR_ACCL_DR_50 = 0x40 # ODR = 50 Hz
41 LIS3DHTR_ACCL_DR_100 = 0x50 # ODR = 100 Hz
42 LIS3DHTR_ACCL_DR_200 = 0x60 # ODR = 200 Hz
43 LIS3DHTR_ACCL_DR_400 = 0x70 # ODR = 400 Hz
44 LIS3DHTR_ACCL_DR_1620 = 0x80 # ODR = 1.62 kHz
45 LIS3DHTR_ACCL_DR_1344 = 0x90 # ODR = 1.344 kHz
46
47 # Accl Data update & Axis configuration
48 LIS3DHTR_ACCL_LPEN = 0x00 # Normal Mode
49 LIS3DHTR_ACCL_XAXIS = 0x04 # X-Axis enable
50 LIS3DHTR_ACCL_YAXIS = 0x02 # Y-Axis enable
51 LIS3DHTR_ACCL_ZAXIS = 0x01 # Z-Axis enable
52
53 # Acceleration Full-scale selection
54 LIS3DHTR_BDU_CONT = 0x00 # Continuous
55 LIS3DHTR_BDU_NOT_CONT = 0x80 # Output register

```

```

56 LIS3DHTR_ACCL_BLE_MSB           = 0x40 # MSB first
57 LIS3DHTR_ACCL_RANGE_16G        = 0x30 # Full scale
58 LIS3DHTR_ACCL_RANGE_8G         = 0x20 # Full scale
59 LIS3DHTR_ACCL_RANGE_4G         = 0x10 # Full scale
60 LIS3DHTR_ACCL_RANGE_2G         = 0x00 # Full scale
61 LIS3DHTR_HR_DS                  = 0x00 # High-Resol
62 LIS3DHTR_HR_EN                  = 0x08 # High-Resol
63 LIS3DHTR_ST_0                   = 0x02 # Self Test
64 LIS3DHTR_ST_1                   = 0x04 # Self Test
65 LIS3DHTR_SIM_3                   = 0x01 # 3-Wire Int
66
67
68 class LIS3DHTR():
69     def __init__(self):
70         self.select_datarate()
71         self.select_data_config()
72
73     def select_datarate(self):
74         """Select the data rate of the accelerometer fr
75         DATARATE_CONFIG = (LIS3DHTR_ACCL_DR_10 | LIS3DH
76         bus.write_byte_data(LIS3DHTR_DEFAULT_ADDRESS, L
77
78     def select_data_config(self):
79         """Select the data configuration of the acceler
80         DATA_CONFIG = (LIS3DHTR_ACCL_RANGE_2G | LIS3DHT
81         bus.write_byte_data(LIS3DHTR_DEFAULT_ADDRESS, L
82
83     def read_accl(self):
84         """Read data back from LIS3DHTR_REG_OUT_X_L(0x2
85         X-Axis Accl LSB, X-Axis Accl MSB"""
86         data0 = bus.read_byte_data(LIS3DHTR_DEFAULT_ADD
87         data1 = bus.read_byte_data(LIS3DHTR_DEFAULT_ADD
88
89         xAccl = data1 * 256 + data0
90         if xAccl > 32767 :
91             xAccl -= 65536
92         xAccl /= 16000
93         """Read data back from LIS3DHTR_REG_OUT_Y_L(0x2.
94         Y-Axis Accl LSB, Y-Axis Accl MSB"""
95         data0 = bus.read_byte_data(LIS3DHTR_DEFAULT_ADD
96         data1 = bus.read_byte_data(LIS3DHTR_DEFAULT_ADD

```

```

97
98     yAcc1 = data1 * 256 + data0
99     if yAcc1 > 32767 :
100         yAcc1 -= 65536
101     yAcc1 /= 16000
102     """Read data back from LIS3DHTR_REG_OUT_Z_L(0x2
103     Z-Axis Acc1 LSB, Z-Axis Acc1 MSB"""
104     data0 = bus.read_byte_data(LIS3DHTR_DEFAULT_ADD
105     data1 = bus.read_byte_data(LIS3DHTR_DEFAULT_ADD
106
107     zAcc1 = data1 * 256 + data0
108     if zAcc1 > 32767 :
109         zAcc1 -= 65536
110     zAcc1 /= 16000
111     return {'x' : xAcc1, 'y' : yAcc1, 'z' : zAcc1}
112
113 from LIS3DHTR import LIS3DHTR
114 lis3dhtr = LIS3DHTR()
115
116 while True:
117     lis3dhtr.select_datarate()
118     lis3dhtr.select_data_config()
119     time.sleep(0.1)
120     accl = lis3dhtr.read_accl()
121     print "Acceleration in X-Axis : %d" %(accl['x'])
122     print "Acceleration in Y-Axis : %d" %(accl['y'])
123     print "Acceleration in Z-Axis : %d" %(accl['z'])
124     print " ***** "
125     time.sleep(1)

```

- **Step 1** Create a python file.

```
sudo nano LIS3DHTR.py
```



- **Step 2** Copy the above code to the python file.
- **Step 3** Run the python.



```
sudo python LIS3DHTR.py
```

```
pi@raspberrypi: ~/grove.py/grove
Acceleration in Y-Axis : 1
Acceleration in Z-Axis : -1
*****
Acceleration in X-Axis : 0
Acceleration in Y-Axis : 0
Acceleration in Z-Axis : -1
*****
Acceleration in X-Axis : 0
Acceleration in Y-Axis : 1
Acceleration in Z-Axis : 0
*****
Acceleration in X-Axis : -1
Acceleration in Y-Axis : 0
Acceleration in Z-Axis : -1
*****
Acceleration in X-Axis : 0
Acceleration in Y-Axis : -1
Acceleration in Z-Axis : -1
*****
Acceleration in X-Axis : 0
Acceleration in Y-Axis : 0
Acceleration in Z-Axis : 1
*****
```

Schematic Online Viewer



Resource

- **[PDF]** [LIS3DHTR_DATASHEET](https://files.seeedstudio.com/products/114020121/document/lis3dh.pdf)
[https://files.seeedstudio.com/products/114020121/document/lis3dh.pdf]
- **[PDF]** [Hardware schematic](https://files.seeedstudio.com/products/114020121/document/Grove-3-Axis-Digital-Accelerometer-LIS3DHTR_v1.0_SCH_190910.pdf)
[https://files.seeedstudio.com/products/114020121/document/Grove-3-Axis-Digital-Accelerometer-LIS3DHTR_v1.0_SCH_190910.pdf]

- **[ZIP]** [Grove - 3-Axis Digital Accelerometer \(LIS3DHTR\) Library](https://github.com/Seeed-Studio/Seeed_Arduino_LIS3DHTR/archive/master.zip)
[https://github.com/Seeed-Studio/Seeed_Arduino_LIS3DHTR/archive/master.zip]

Tech Support

Please do not hesitate to submit the issue into our [forum](https://forum.seeedstudio.com/)
[<https://forum.seeedstudio.com/>].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]