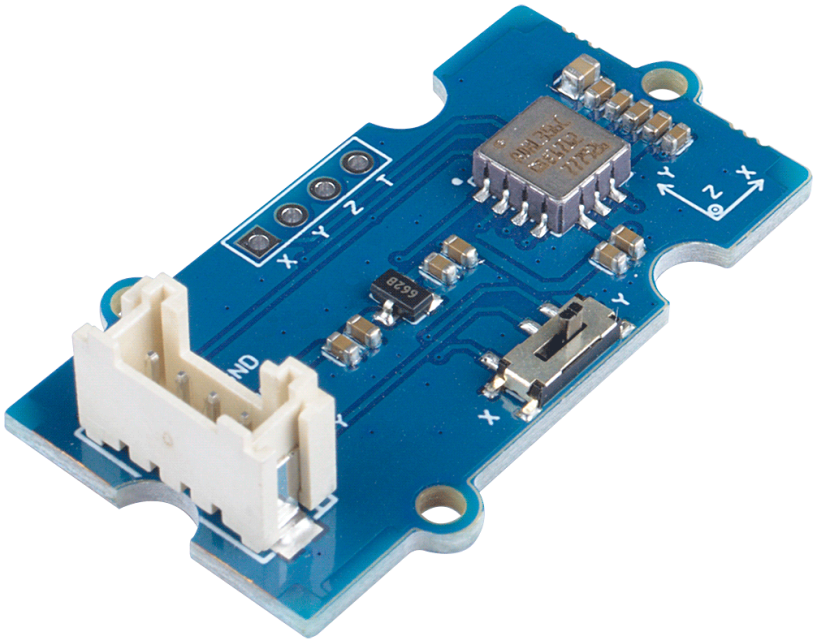
Grove - 3-Axis Analog Accelerometer 40g (ADXL356C)



You can find a variety of 3-axis accelerometers [https://www.seeedstudio.com/tag/accelerometer.html] on our website that can meet different scenarios and needs. This time, we bring you the industrial grade, high stability, high precision and low power ADI ADXL series three-axis accelerometers.

The Grove - 3-Axis Analog Accelerometer ±40g (ADXL356C) is a analog output MEMS [https://www.seeedstudio.com/tag/MEMS.html] Accelerometer. This sensor has two selectable measurement ranges: ±10g, ±40g. You just need to do little calibration work to get a relatively accurate result. The On-Board grove port can output two channel analog data: one for Z-axis, one for X/Y-axis. You can choose output the X-axis or Y-axis signal with the on-board switch. Also you can use the 4-pin welding hole to output X/Y/Z axis at the same time. The power consumption of this sensor is extremely low, 150 μA in normal operation mode and even only 21 μA in standby mode. You can switch the operating mode by changing the pad connection on the back.

The ADI ADXL Series Accelerometer includes four products that will meet your different range and output needs:

Product	Measurement Range	Output Port	Pow Con
Grove - 3-Axis Analog Accelerometer ±20g (ADXL356B) [https://www.seeedstudio.com/Grove-3-Axis-Analog-Accelerometer-20g-ADXL356B-p-4004.html]	±10 ±20g	Analog	mea moc star moc
Grove - 3-Axis Analog Accelerometer ±40g (ADXL356C) [https://www.seeedstudio.com/Grove-3-Axis-Analog-Accelerometer-40g-ADXL356C-p-4006.html]	±10g ±40g	Analog	mea moc star moc
Grove - 3-Axis Digital Accelerometer ±40g (ADXL357) [https://www.seeedstudio.com/Grove-3-Axis-Digital-Accelerometer-40g-ADXL357-p-4005.html]	±10g@51200 LSB/g ±20g@25600 LSB/g ±40g@12800 LSB/g	Digital I2C	mea moc
Grove - 3-Axis Digital Accelerometer ±200g (ADXL372) [https://www.seeedstudio.com/Grove-3-Axis-Digital-Accelerometer-200g-ADXL372-p-4003.html]	±200g	Digital I2C	mea moc

Get One Now

[https://www.seeedstudio.com/Grove-3-Axis-Analog-Accelerometer-40g-ADXL356C-p-4006.html]

Features

- Industry leading noise, minimal offset drift over temperature, and long-term stability, enabling precision applications with minimal calibration.
- Hermetic package offers excellent long-term stability 0 g offset vs. temperature (all axes): 0.75 mg/°C maximum
- The low noise of the ADXL356 over higher frequencies is ideal for wireless condition monitoring.
- Low drift, low noise
- Ultra low power consumption: Normal operation mode-150 μA, Standby mode 21 μA.

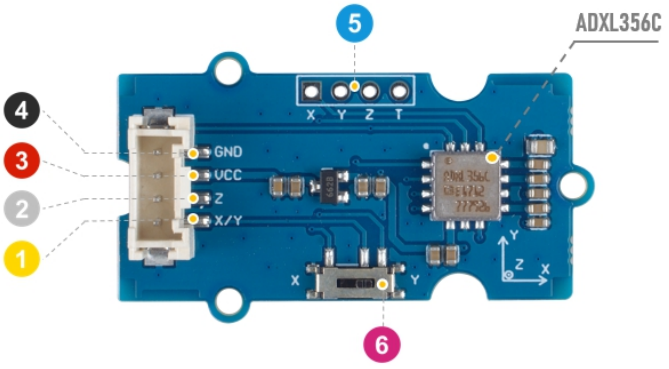
APPLICATIONS

- Inertial measurement units (IMUs)/altitude and heading reference systems (AHRSs)
- Platform stabilization systems
- Structural health monitoring
- Condition monitoring
- Seismic imaging
- Tilt sensing
- Robotics

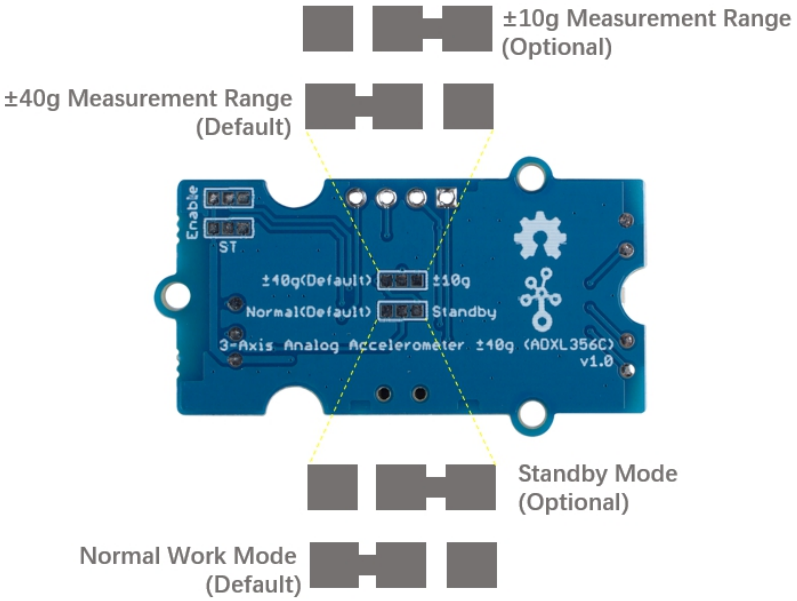
Specification

Parameter	Value
Supply voltage	3.3V / 5V
Operating ambient temperature	-40 – 125℃
Output Full-Scale Range (FSR)	±10g / ±40g
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT} (Ratiometric to V_{1P8ANA})	±10 g@80 mv/g (Typ.) ±40 g@20 mv/g (Typ.)
Sensitivity Change due to Temperature	±0.01%/°C (TA = −40°C to +125°C)
0g OFFSET (Referred to V1P8ANA/2)	±125 mg(Typ.)
Output interface	Analog

Pinout



4 **GND**: connect this module to the system GND
3 **VCC**: you can use 5V or 3.3V for this module
2 **Z**: Z-axis analog data output
1 **X/Y**: X-axis or Y-axis analog data output
5 **4 Pin Output**: X/Y/Z axis analog data output, Temperature output
6 **Switch**: switch X/Y output



Platforms Supported

Arduino	Raspberry Pi		

Getting Started

Play With Arduino

Materials required

Seeeduino V4.2	Base Shield
	
Get ONE Now [https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]	Get ONE Now [https://www.seeedstudio.com/Base Shield-V2-p-1378.html]

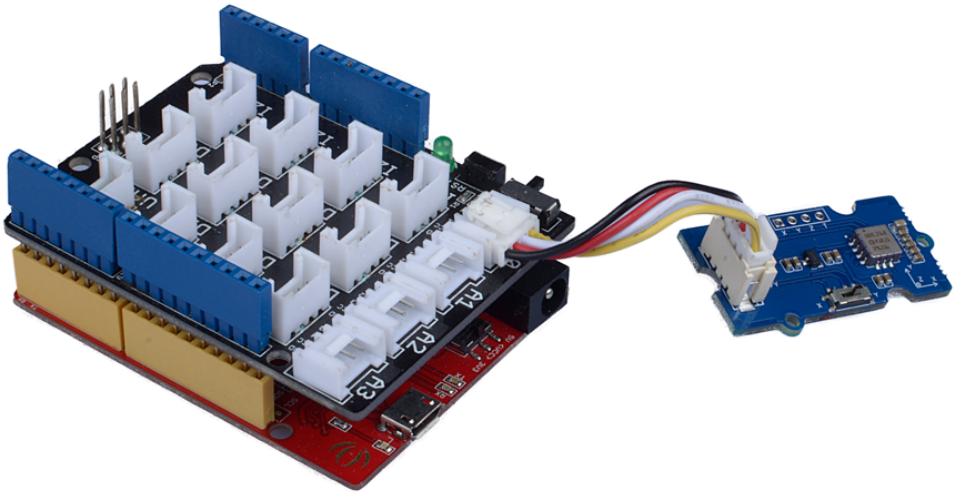
> In addition, you can consider our new Seeeduino Lotus M0+ [https://www.seeedstudio.com/Seeeduino-Lotus-Cortex-M0-p-2896.html], which is equivalent to the combination of Seeeduino V4.2 and Baseshield.

Note

1 Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click here [https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html] to buy **2** Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click here [https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html] to buy.

Hardware Connection

- **Step 1.** Connect the Grove - 3-Axis Analog Accelerometer ±20g (ADXL356B) to the **A0** port of the Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3.** Connect Seeeduino to PC via a USB cable.



Software

> **Attention**
> If this is the first time you work with Arduino, we strongly recommend you to see Getting Started with Arduino [https://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

- **Step 1.** Download the Seeed_ADXL_356.ino [https://github.com/linux-downey/Seeed_ADXL356/blob/master/Seeed_ADXL_356/Seeed_ADXL_356.ino] from Github.

or you can just copy the following code.

Attention
This code is written for both ADXL_356B and ADXL_356C, and if you need to implement them on 356C, pls switch the line `#define MODUEL_RANGE 20` with `#define MODUEL_RANGE 40`, or it will not work.

```
#include <Arduino.h>

#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
  #define SERIAL SerialUSB
  #define SYS_VOL   3.3
#else
  #define SERIAL Serial
  #define SYS_VOL   5
#endif

float cali_data_xy;
float cali_data_z;
int16_t scale;

#define MODUEL_RANGE          20

#define MODULE_VOL            1.8

#define CALI_BUF_LEN          15
#define CALI_INTERVAL_TIME    250

float cali_buf_xy[CALI_BUF_LEN];
float cali_buf_z[CALI_BUF_LEN];

#define XY_PIN   A0
#define Z_PIN    A1

float deal_cali_buf(float *buf)
{
    float cali_val = 0;

    for(int i = 0;i < CALI_BUF_LEN;i++)
```

```
    {
        cali_val += buf[i];
    }
    cali_val = cali_val/CALI_BUF_LEN;
    return (float)cali_val;
}


void calibration(void)
{
    SERIAL.println("Please Place the module horizontall
    delay(1000);
    SERIAL.println("Start calibration........");

    for(int i=0;i<CALI_BUF_LEN;i++)
    {
        cali_buf_xy[i] = analogRead(XY_PIN);;
        cali_buf_z[i] = analogRead(Z_PIN);
        delay(CALI_INTERVAL_TIME);
    }
    cali_data_xy =  deal_cali_buf(cali_buf_xy);
    cali_data_z =  (float)deal_cali_buf(cali_buf_z);
    SERIAL.println("Calibration OK!!");
    scale = (float)1000 / (cali_data_z - cali_data_xy);
    cali_data_z -= (float)980 / scale;
    SERIAL.println(cali_data_xy);
    SERIAL.println(cali_data_z);
    SERIAL.println(scale);

}


void AccMeasurement(void)
{
    int16_t val_xy = 0;
    int16_t val_z = 0;
    val_xy = analogRead(XY_PIN);
    val_z = analogRead(Z_PIN);

    SERIAL.print("Raw data xy  = ");
```

```
    SERIAL.println(val_xy);
    SERIAL.print("Raw data z  = ");
    SERIAL.println(val_z);
    SERIAL.println(" ");

    val_xy -= cali_data_xy;
    val_z -= cali_data_z;
    SERIAL.print("x or y position acc is ");
    SERIAL.print(val_xy * scale / 1000.0);
    SERIAL.println(" g ");
    SERIAL.print("z position acc is ");
    SERIAL.print(val_z * scale / 1000.0);
    SERIAL.println(" g ");
    SERIAL.println(" ");
    SERIAL.println(" ");
    SERIAL.println(" ");
    delay(1000);
}

void setup()
{
    SERIAL.begin(115200);
    #ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
    analogReadResolution(12);
    #endif
    calibration();
    SERIAL.print("Scale = ");
    SERIAL.println(scale);

}



void loop()
{
    AccMeasurement();
}
```

- **Step 2.** Upload the demo. If you do not know how to upload the code, please check How to upload code [https://wiki.seeedstudio.com/Upload_Code/].
- **Step 3.** Open the **Serial Monitor** of Arduino IDE by click **Tool-> Serial Monitor**. Or tap the `Ctrl`+`Shift`+`M` key at the same time. Set the baud rate to **115200**.
- **Step 4. Calibration** Follow the calibration tips in the Serial Monitor, just few step the calibration will be finished
- **Step 5.** Now you can use this sensor, and the output will be like this:

```
Please Place the module horizontally!
Start calibration........
Calibration OK!!
184.93
185.03
121
Scale = 121
Raw data xy  = 185
Raw data z  = 193

x or y position acc is 0.00 g
z position acc is 0.85 g



Raw data xy  = 188
Raw data z  = 196

x or y position acc is 0.36 g
z position acc is 1.21 g
```

Attention

> If you use Grove port to output the data, the X axis and the Y axis can not be output at the same time, you can use the on-board swith to select the output channel. If you want to output X/Y/Z at the same time, you can use the 4 pin welding hole.

FAQ

Q1: How to select the ±10g measurement range?

A1: To change the measurement range you need to modify both the hardware and software. First, cut the back pad which connected to ±40g and re-solder it to ±10g. Then modify the code in the code block line 12

```
#define MODUEL_RANGE          40
```

\downdownarrows

```
#define MODUEL_RANGE          10
```

Schematic Online Viewer

Resources

- **[ZIP]** Grove - 3-Axis Analog Accelerometer ±20g (ADXL356C) Schematic file [https://files.seeedstudio.com/wiki/Grove-3-Axis_Analog_Accelerometer-40g-ADXL356C/res/Grove%20-%203-Axis%20Analog%20Accelerometer%20%C2%B140g%20(ADXL356C).zip]

- **[PDF]** ADXL 356 Datasheet [https://files.seeedstudio.com/wiki/Grove-3-Axis_Analog_Accelerometer-20g-ADXL356B/res/ADXL356B.pdf]

Tech Support

Please submit any technical issue into our forum [https://forum.seeedstudio.com/]

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