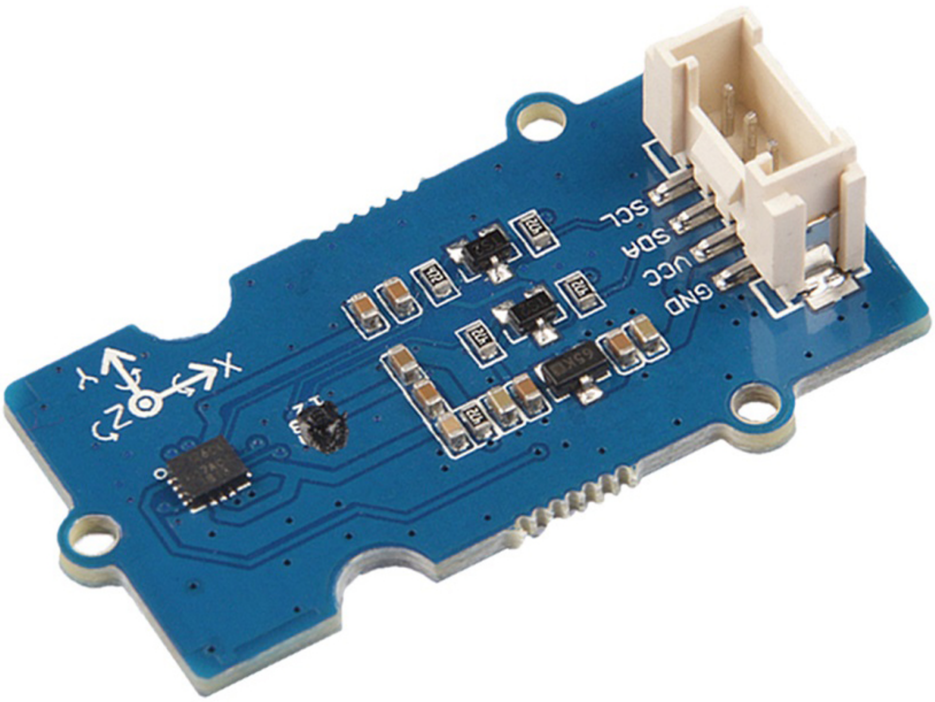
# Grove - IMU 9DOF(Icm20600+AK09918)



The Grove - IMU 9DOF (Icm20600+AK09918) is a 9 Degrees of Freedom IMU [https://en.wikipedia.org/wiki/Inertial_measurement_unit] (Inertial measurement unit) which combines gyroscope, accelerometer and electronic compass. We use two chips LCM20600+AK09918 to implement those 3 functions.

The LCM20600 is a 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer. Gyroscope [https://en.wikipedia.org/wiki/Gyroscope] is a device used for measuring or maintaining orientation and angular velocity, normally, we use it to measure spin and twist. Accelerometer [https://en.wikipedia.org/wiki/Accelerometer] is a device that measures proper acceleration.

The AK09918 is a 3-axis electronic compass [https://en.wikipedia.org/wiki/Magnetometer] IC with high sensitive Hall sensor technology. We use an electronic compass to measure the magnetic force, which can provide us with the direction information.

As its name suggests just use this single small module and you can measure 9 Degrees of Freedom: angular rotation in x/y/z axis, acceleration in x/y/z axis, and magnetic force in x/y/z axis.

What an amazing module! Just use this module to build your own motion and orientation system😄

Get One Now 🛒

[https://www.seeedstudio.com/Grove-IMU-9DOF-%28lcm20600%2BAK09918%29-p-3157.html]

## Features

- 3-Axis Gyroscope with Programmable FSR of ±250 dps, ±500 dps, ±1000 dps, and ±2000 dps
- 3-Axis Accelerometer with Programmable FSR of ±2g, ±4g, ±8g, and ±16g
- 3-Axis Electronic Compass with 0.15 μT/LSB (typ.) sensitivity
- User-programmable interrupts
- 16-bit ADC resolution and Programmable Filters for acceleration measurements
- 16-bit ADC resolution for magnetic measurements
- 1 KB FIFO buffer enables the applications processor to read the data in bursts(LCM20600)
- Embedded temperature sensor
- Magnetic sensor overflow monitor function
- Built-in oscillator for internal clock source

## Specification

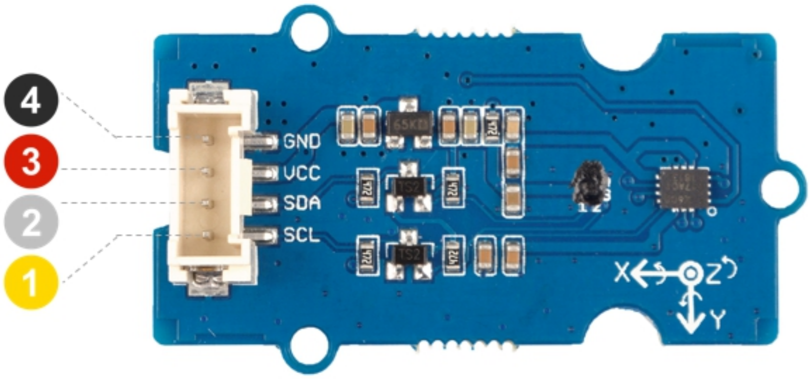
| Item | Value |
|---|---|
| Operating voltage | 3.3V / 5V |
| Operating temperature | -30°C to +85°C |
| Gyroscope Full-Scale Range | ±250 dps, ±500 dps, ±1000 dps, ±2000 dps |
| Gyroscope Sensitivity Scale Factor | 131 LSB/(dps)@±250 dps<br>65.5 LSB/(dps)@±500 dps<br>32.8 LSB/(dps)@±1000 dps<br>16.4 LSB/(dps)@±2000 dps |
| Accelerometer Full-Scale Range | ±2g, ±4g, ±8g, ±16g |
| Accelerometer Sensitivity Scale Factor | 16384 LSB/g@±2g<br>8192 LSB/g@±4g<br>4096 LSB/g@±8g<br>2048 LSB/g@±16g |
| Magnetic sensor measurement range | ±4912μT (typical) |
| Magnetic sensor sensitivity | 0.15μT (typical) |
| Interface | I$^2$C |
| I$^2$C Address | **LCM20600**<br>0x69(default)<br>0x68(optional)<br>**AK09918**<br>0x0C |

# Applications

- Smartphones and Tablets
- Wearable Sensors

# Hardware Overview

## Pin Out

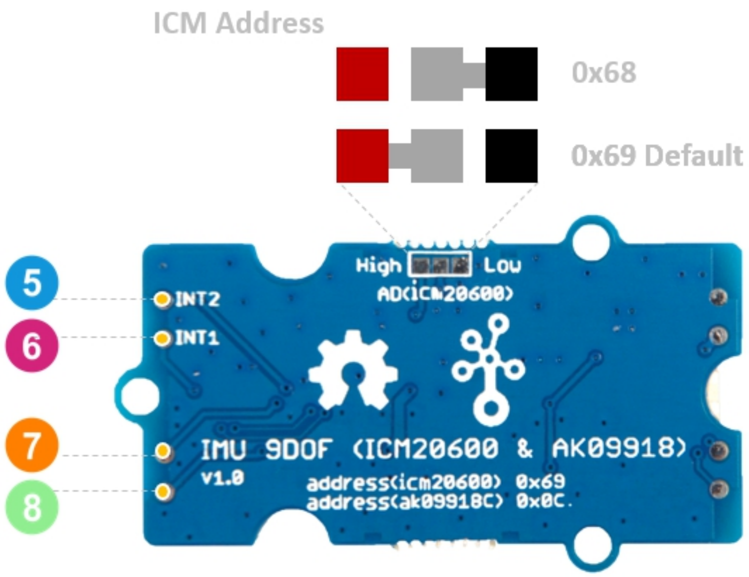


4 GND: connect this module to the system GND

3 VCC: you can use 5V or 3.3V for this module

2 SDA: I²C serial data

1 SCL: I²C serial clock



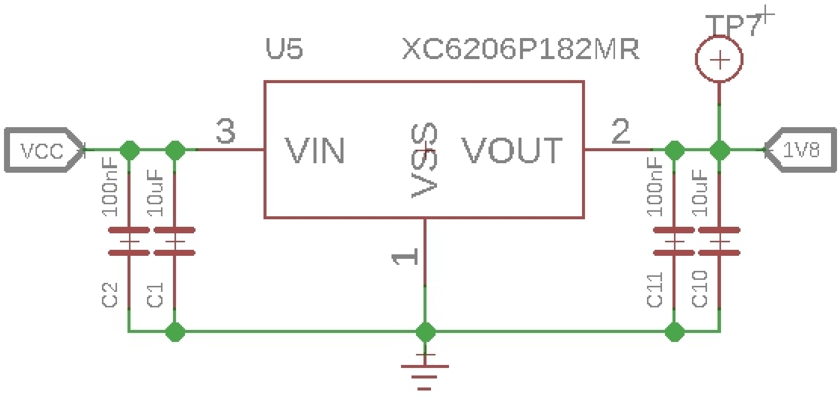
5 INT2: Interrupt digital output (totem pole or open-drain)
6 INT1: Interrupt digital output (totem pole or open-drain)
7 FSYNC: Frame synchronization digital input or No Connect
8 VCC_1.8V: Provide 1.8V for ICM20600 and AK09918

**Danger**

The default I2C address of LCM20600 is 0x69, you can change it to 0x68. The central pad is connected to the address wire, you can change the I2C address by cutting the wire and re-welding it. For the safety of you and others, please be careful with knife or welding gun you may use.

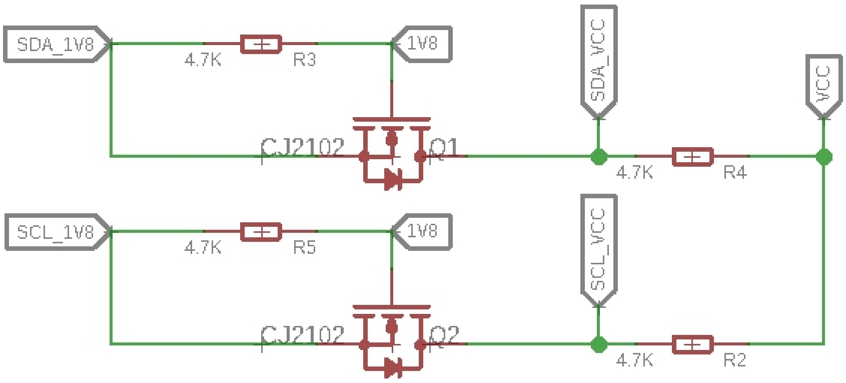
## Schemaitc

### Power



Since the operating voltage range of LCM20600 is 1.71V to 3.45V, and the operating voltage range of AK09918 is 1.65V to 1.95V, we use a power conversion chip **XC6206P182MR** to provide a stable 1.8V for both chips.

### Bi-directional level shifter circuit



This is a typical Bi-directional level shifter circuit to connect two different voltage section of an I$^2$C bus. The I$^2$C bus of two chips use 1.8V, if the I$^2$C bus of the Arduino use 5V or 3.3V, this circuit will be needed. In the schematic above, **Q1** and **Q2** are N-Channel MOSFET CJ2102 [https://files.seeedstudio.com/wiki/Grove-IMU_9DOF-lcm20600_AK09918/res/CJ2102.pdf], which act as a bidirectional switch. In order to better understand this part, you can refer to the AN10441 [https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/AN10441.pdf]

# Platforms Supported

| Arduino | Raspberry Pi | |
|---|---|---|
| | TO DO | |

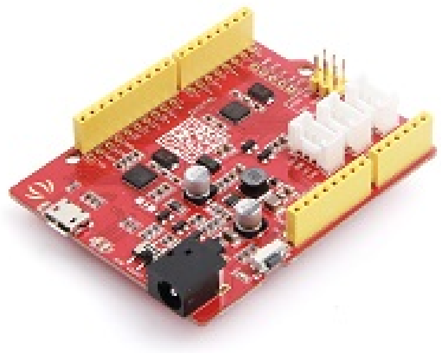
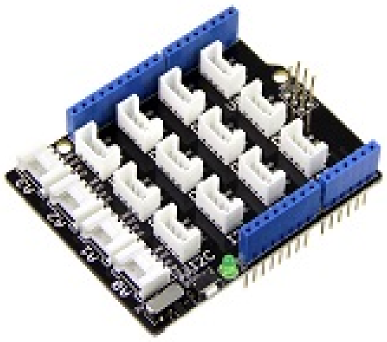
**Caution**
The platforms mentioned above as supported is/are an indication of the module's software or theoritical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms.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

# Getting Started

## Play With Arduino

### Hardware

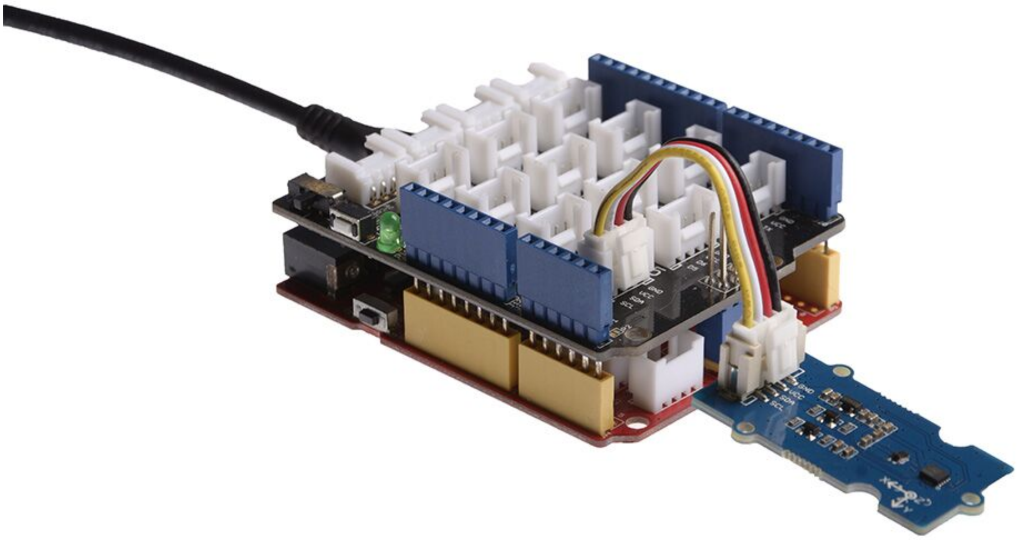
### Materials required

| 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] |

**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.

- **Step 1.** Connect the Grove - IMU 9DOF (Icm20600+AK09918) to port **I$^2$C** of Grove-Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3.** Connect Seeeduino to PC via a USB cable.



**Note**
If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

| Seeeduino | Grove - IMU 9DOF |
|---|---|
| 5V | Red |
| GND | Black |
| SDA | White |
| SCL | Yellow |

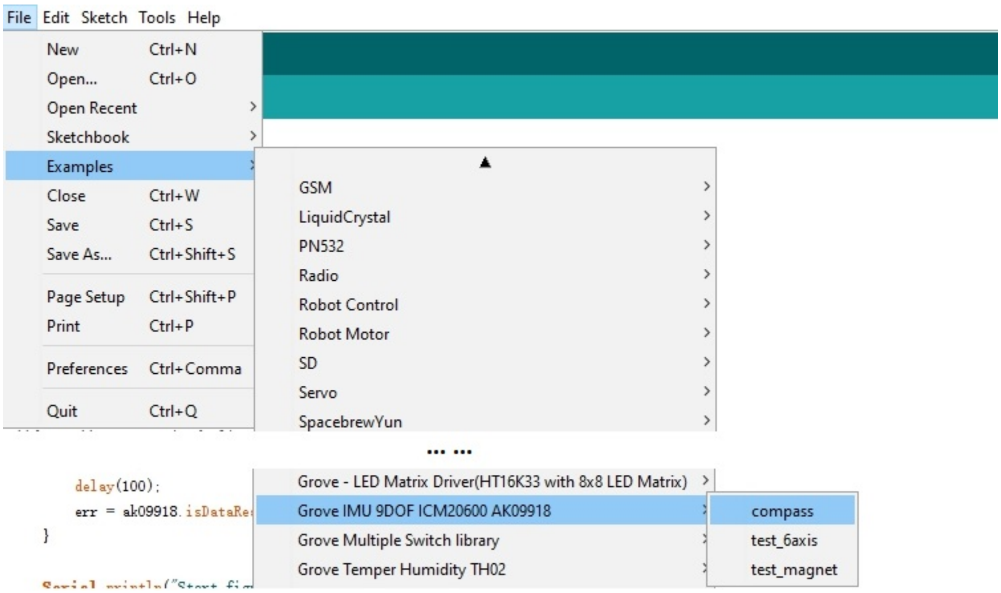
## Software

**Note**

> 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 Grove - IMU 9DOF (Icm20600+AK09918) [https://github.com/Seeed-Studio/Seeed_ICM20600_AK09918] Library from Github.
- **Step 2.** Refer to How to install library [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open the example, you can open it in the following three ways：

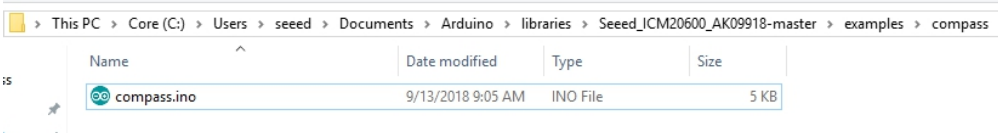
    a. Open it directly in the Arduino IDE via the path: **File** → **Examples** → **Grove IMU 9DOF ICM20600 AK09918** → **compass**.



```
    delay(100);
    err = ak09918.isDataRe
}
```

    b. Open it in your computer by click the **compass.ino** which you can find in the folder **XXXX\Arduino\libraries\Seeed_ICM20600_AK09918-**

**master\examples\compass**, **XXXX** is the location you installed the Arduino IDE.



This PC > Core (C:) > Users > seeed > Documents > Arduino > libraries > Seeed_ICM20600_AK09918-master > examples > compass

| Name | Date modified | Type | Size |
|---|---|---|---|
| compass.ino | 9/13/2018 9:05 AM | INO File | 5 KB |

c. Or, you can just click the icon ⧉ in upper right corner of the code block to copy the following code into a new sketch in the Arduino IDE.

```
#include "AK09918.h"
#include "ICM20600.h"
#include <Wire.h>

AK09918_err_type_t err;
int32_t x, y, z;
AK09918 ak09918;
ICM20600 icm20600(true);
int16_t acc_x, acc_y, acc_z;
int32_t offset_x, offset_y, offset_z;
double roll, pitch;
// Find the magnetic declination at your location
// http://www.magnetic-declination.com/
double declination_shenzhen = -2.2;

void setup()
{
    // join I2C bus (I2Cdev library doesn't do this aut
    Wire.begin();

    err = ak09918.initialize();
    icm20600.initialize();
    ak09918.switchMode(AK09918_POWER_DOWN);
    ak09918.switchMode(AK09918_CONTINUOUS_100HZ);
    Serial.begin(9600);

    err = ak09918.isDataReady();
    while (err != AK09918_ERR_OK)
    {
        Serial.println("Waiting Sensor");
        delay(100);
        err = ak09918.isDataReady();
    }

    Serial.println("Start figure-8 calibration after 2
    delay(2000);
    calibrate(10000, &offset_x, &offset_y, &offset_z);
    Serial.println("");
}

void loop()
{
    // get acceleration
    acc_x = icm20600.getAccelerationX();
    acc_y = icm20600.getAccelerationY();
    acc_z = icm20600.getAccelerationZ();

    Serial.print("A:  ");
    Serial.print(acc_x);
    Serial.print(",  ");
    Serial.print(acc_y);
    Serial.print(",  ");
    Serial.print(acc_z);
    Serial.println(" mg");

    Serial.print("G:  ");
    Serial.print(icm20600.getGyroscopeX());
    Serial.print(",  ");
    Serial.print(icm20600.getGyroscopeY());
    Serial.print(",  ");
    Serial.print(icm20600.getGyroscopeZ());
    Serial.println(" dps");

    ak09918.getData(&x, &y, &z);
    x = x - offset_x;
    y = y - offset_y;
    z = z - offset_z;

    Serial.print("M:  ");
    Serial.print(x);
    Serial.print(",  ");
    Serial.print(y);
    Serial.print(",  ");
    Serial.print(z);
    Serial.println(" uT");

    // roll/pitch in radian
    roll = atan2((float)acc_y, (float)acc_z);
    pitch = atan2(-(float)acc_x, sqrt((float)acc_y*acc_
    Serial.print("Roll: ");
    Serial.println(roll*57.3);
    Serial.print("Pitch: ");
    Serial.println(pitch*57.3);

    double Xheading = x * cos(pitch) + y * sin(roll) *
    double Yheading = y * cos(roll) - z * sin(pitch);


    double heading = 180 + 57.3*atan2(Yheading, Xheadin

    Serial.print("Heading: ");
    Serial.println(heading);
    Serial.println("--------------------------------");

    delay(500);

}

void calibrate(uint32_t timeout, int32_t *offsetx, int3
{
  int32_t value_x_min = 0;
  int32_t value_x_max = 0;
  int32_t value_y_min = 0;
  int32_t value_y_max = 0;
  int32_t value_z_min = 0;
  int32_t value_z_max = 0;
  uint32_t timeStart = 0;

  ak09918.getData(&x, &y, &z);

  value_x_min = x;
```

```
    value_x_max = x;
    value_y_min = y;
    value_y_max = y;
    value_z_min = z;
    value_z_max = z;
    delay(100);

    timeStart = millis();

    while((millis() - timeStart) < timeout)
    {
      ak09918.getData(&x, &y, &z);

      /* Update x-Axis max/min value */
      if(value_x_min > x)
      {
        value_x_min = x;
        // Serial.print("Update value_x_min: ");
        // Serial.println(value_x_min);

      }
      else if(value_x_max < x)
      {
        value_x_max = x;
        // Serial.print("update value_x_max: ");
        // Serial.println(value_x_max);
      }

      /* Update y-Axis max/min value */
      if(value_y_min > y)
      {
        value_y_min = y;
        // Serial.print("Update value_y_min: ");
        // Serial.println(value_y_min);

      }
      else if(value_y_max < y)
      {
        value_y_max = y;
        // Serial.print("update value_y_max: ");
        // Serial.println(value_y_max);
      }

      /* Update z-Axis max/min value */
      if(value_z_min > z)
      {
        value_z_min = z;
        // Serial.print("Update value_z_min: ");
        // Serial.println(value_z_min);

      }
      else if(value_z_max < z)
      {
        value_z_max = z;
        // Serial.print("update value_z_max: ");
        // Serial.println(value_z_max);
      }

      Serial.print(".");
      delay(100);

    }

    *offsetx = value_x_min + (value_x_max - value_x_min)/
    *offsety = value_y_min + (value_y_max - value_y_min)/
    *offsetz = value_z_min + (value_z_max - value_z_min)/
  }
```

**Note**

There are 3 demos in the library:

**test_6axis**

> This example shows how to get gyroscope and acceleration data from ICM20600.

**test_magnet**

> This example shows how to get magnetic data from AK09918.

**compass**

> This example gets magnetic data and acceleration data, to count pitch and roll, and make a compass application.

- **Step 4.** 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 5.** 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 **9600**.

✓ **Success**

If every thing goes well, when you open the Serial Monitor, the notice will pop up--*Start figure-8 calibration after 2 seconds.* Which means in order to calibrate this module, you should move it and draw the number 8 trajectory in the air. When the "......." appears, you can start your calibration.

```
Start figure-8 calibration after 2 seconds.
........................................................
A:  -362,  -205,  738 mg
G:  -45,  12,  -1 dps
M:  -6,  -23,  -33 uT
Roll: -15.53
Pitch: 25.30
Heading: 23.99
--------------------------------
A:  -269,  583,  61 mg
G:  102,  377,  -2 dps
M:  18,  -21,  -18 uT
Roll: 84.03
Pitch: 24.65
Heading: 215.58
--------------------------------
A:  -495,  229,  37 mg
G:  -43,  -231,  201 dps
M:  7,  -30,  6 uT
```

```
Roll: 80.83
Pitch: 64.90
Heading: 21.76
-------------------------------
```

**Note**
As you can see, the result of compass example includes three parameter: roll, pitch and Heading. There are the terminology of **Euler angles [https://en.wikipedia.org/wiki/Euler_angles]**(click to check more information).

## Fuction table

| Function | Description |
|---|---|
| **ICM20600** | |
| initialize() | Initialize the chip LCM20600, by default:<br>the measurement range of gyroscope is ± dps<br>the measurement range of accelerometer |
| setGyroScaleRange(gyro_scale_type_t range) | After the initialization, you can set the gyr range to meet your own needs, the param gyro_scale_type_t range list:<br>**RANGE_250_DPS**<br>**RANGE_500_DPS**<br>**RANGE_1K_DPS**<br>**RANGE_2K_DPS**<br>e.g.<br>**icm20600.setGyroScaleRange(RANGE_1**<br>this code line will change the gyroscope measurement range to ±1000dps |
| setAccScaleRange(acc_scale_type_t | After the initialization, you can set the |

Function Description

| Function | Description |
|---|---|
| range) | accelerometer range to meet your own ne parameter acc_scale_type_t range list:<br>**RANGE_2G**<br>**RANGE_4G**<br>**RANGE_8G**<br>**RANGE_16G**<br>e.g.<br>**icm20600.setAccScaleRange(RANGE_8(**<br>this code line will change the accelerome measurement range to ±8g |
| getGyroscope(int16_t* x, int16_t* y, int16_t* z)) | You can use this function to get the gyros X/Y/Z 3-axis data at the same time, and t of the data is **dps** |
| getGyroscopeX(void)<br>getGyroscopeY(void)<br>getGyroscopeZ(void) | Or, you can get the gyroscope X/Y/Z 3-axi separately by using those three functions the unit of the data is **dps** |
| getRawGyroscopeX(void)<br>getRawGyroscopeX(void)<br>getRawGyroscopeX(void) | Those three functions get the raw data di from the register of ICM20600 without co the data unit to **dps** |
| getAcceleration(int16_t* x, int16_t* y, int16_t* z) | You can use this function to get the X/Y/Z acceleration at the same time, and the un data is **mg** |
| getAccelerationX(void)<br>getAccelerationY(void)<br>getAccelerationZ(void) | Or, you can get the X/Y/Z 3-axis accelerat separately by using those three functions the unit of the data is **mg** |
| getRawAccelerationX(void)<br>getRawAccelerationY(void)<br>getRawAccelerationZ(void) | Those three functions get the raw data di from the register of ICM20600 without co the data unit to **mg** |
| getTemperature(void) | You ca use this function to get the tempe |

| | |
|---|---|
| **AK09918** | |
| getData(int32_t *axis_x, int32_t *axis_y, int32_t *axis_z) | You can use this function to get the magr force of 3-axis. |

## Schematic Online Viewer

## Resources

- **[Zip]** Grove - IMU 9DOF (Icm20600+AK09918) Eagle Files [https://files.seeedstudio.com/wiki/Grove-IMU_9DOF-Icm20600_AK09918/res/Grove%20-%20IMU%209DOF%20(ICM20600%20%26%20AK09918).zip]
- **[Zip]** Seeed ICM20600+AK09918 Library [https://github.com/Seeed-Studio/Seeed_ICM20600_AK09918/archive/master.zip]
- **[PDF]** Datasheet of ICM-20600 [https://files.seeedstudio.com/wiki/Grove-IMU_9DOF-Icm20600_AK09918/res/ICM-20600.pdf]
- **[PDF]** Datasheet of AK09918 [https://files.seeedstudio.com/wiki/Grove-IMU_9DOF-Icm20600_AK09918/res/AK09918.pdf]
- **[PDF]** Datasheet of CJ2102 [https://files.seeedstudio.com/wiki/Grove-IMU_9DOF-Icm20600_AK09918/res/CJ2102.pdf]

## Project

This is the introduction Video of this product, simple demos, you can have a try.

All new Grove - Motion Sens...

## Tech Support

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