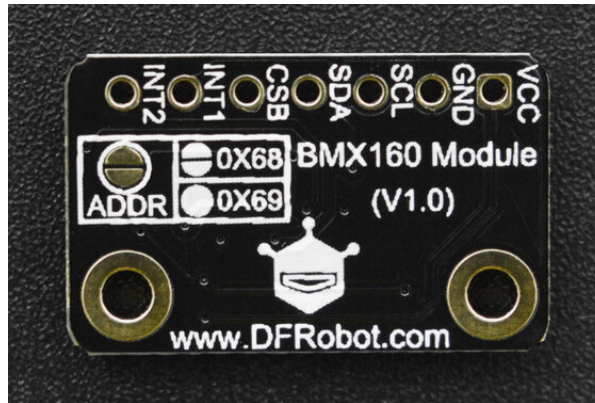


SKU:SEN0373 (<https://www.dfrobot.com/product-2141.html>)

Introduction
Features
Specification
Board Overview
API Functions
Tutorial
FAQ
More Documents



(<https://www.dfrobot.com/product-2141.html>)

Introduction

This compact BMX160 9-axis sensor from Bosch Sensortec is an ideal solution for applications that face strict constraints of board space, power consumption and appearance like wearable fitness trackers, smart remote controls, quadcopters and so on.

This BMX160 sensor is the smallest 9-axis sensor in the industry. It comprises an accelerometer, gyroscope and geomagnetic sensor in a single package, and features less than 1.5mA power consumption.

Due to the compact design, BMX160 can be ideally integrated into wearables like smart watches or glasses for augmented reality. Combined with BSX sensor data fusion software library of Bosch Sensortec, the sensor performance can be further improved.

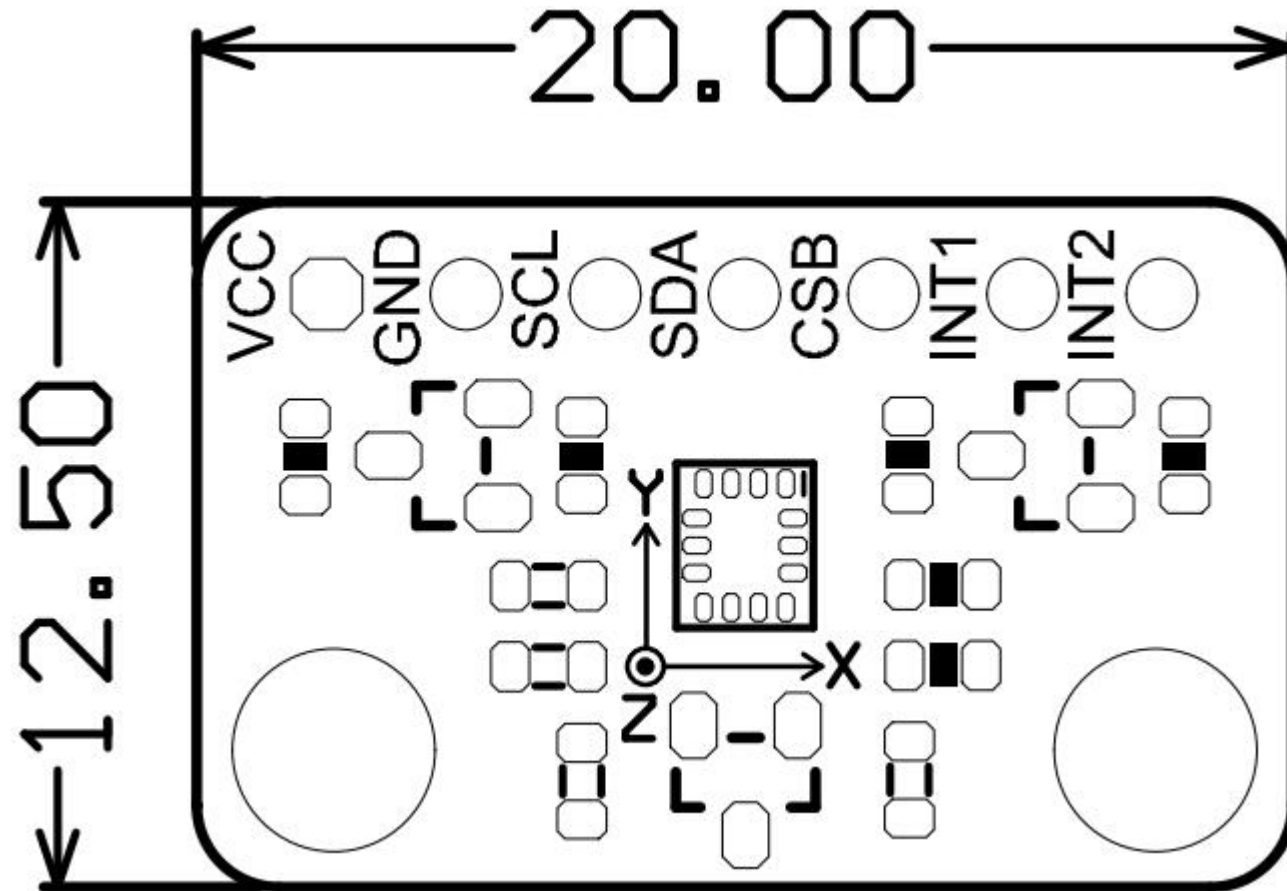
Features

- BMX160 9-Axis Sensor
 - Integrated Three Sensors:
 - 16-bit Gyroscope
 - 16-bit Accelerometer

- Geomagnetic Sensor

- Intelligent Power Management System: normal, low power, sleep

Specification

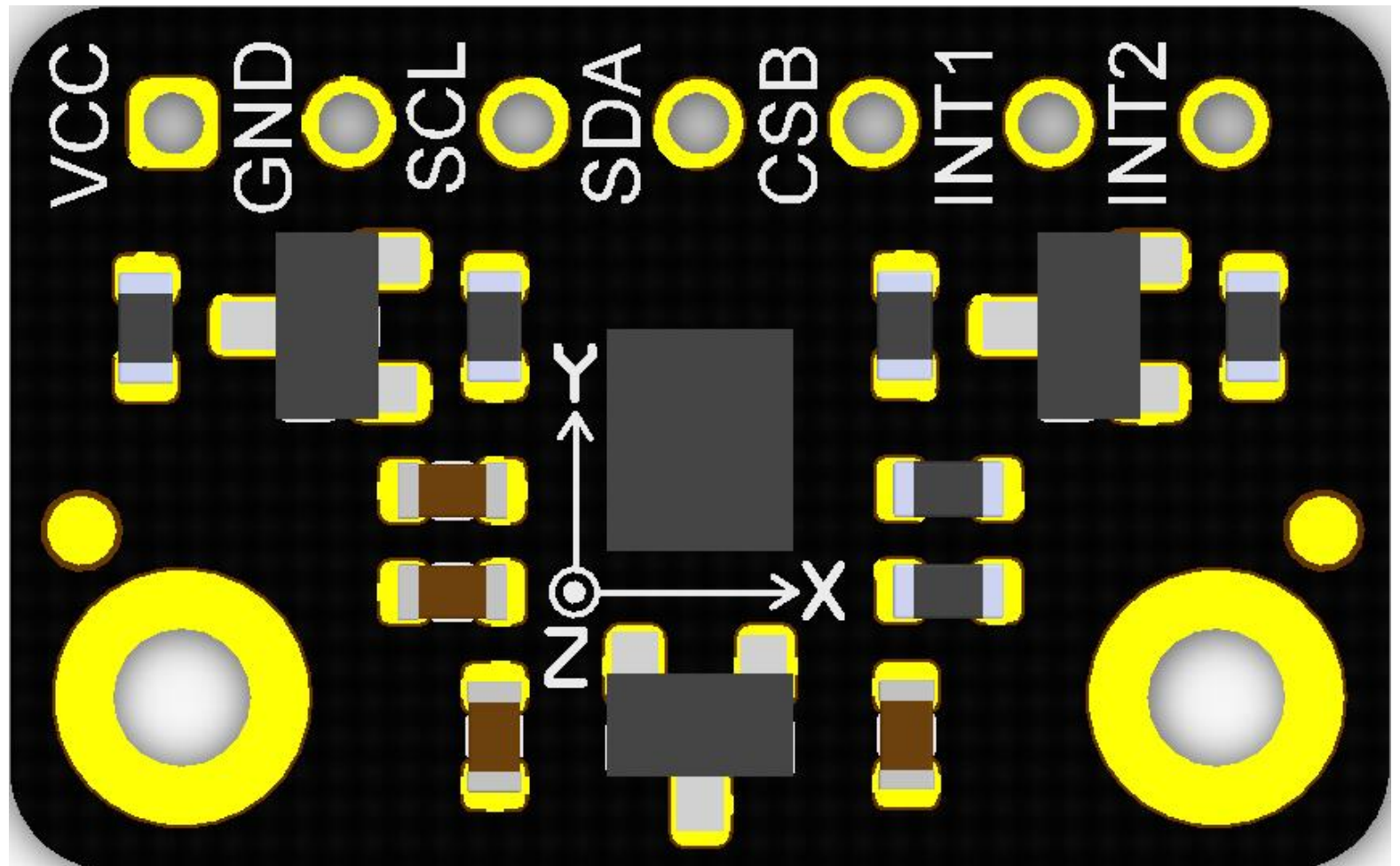
[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

- Operating Voltage: 3.3V-5.5V
- BMX160 Accelerometer:
 - Accelerometer: $\pm 2g/\pm 4g/\pm 8g/\pm 16g$
- BMX160 Gyroscope:
 - Range: $\pm 125^\circ/s \sim 2000^\circ/s$

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

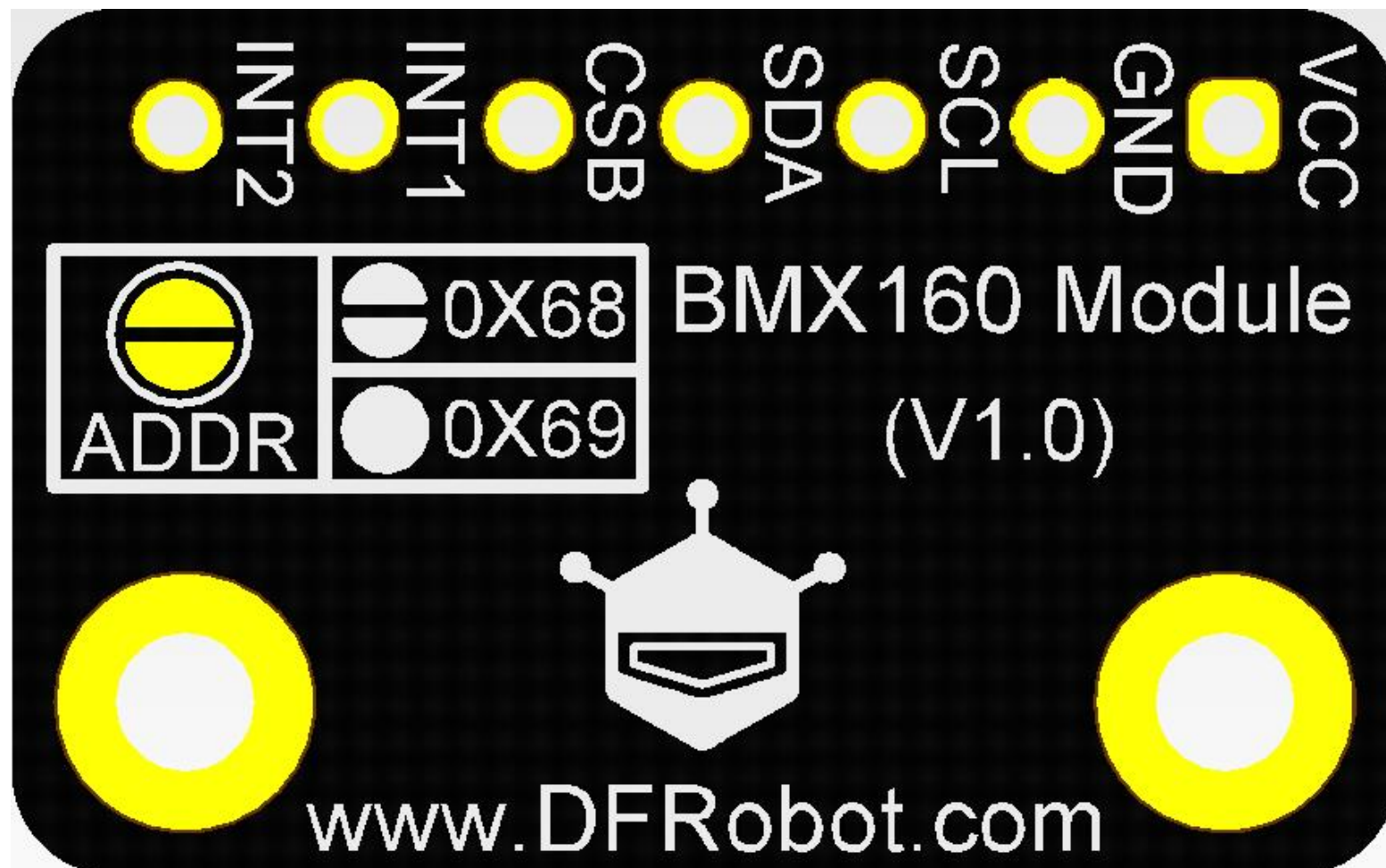
- BMX160 Geomagnetic:
 - Range: $\pm 1150\mu\text{T}$ (x-,y-axis); $\pm 2500\mu\text{T}$ (z-axis)
 - Resolution: ~ 0.3
- Outline Dimension: 20 x 12.5mm/0.79 x 0.49"
- Mounting Hole Position: 15mm
- Mounting Hole Size: inner diameter 2mm/outer diameter 3.7mm

Board Overview



- Introduction
- Features
- Specification
- Board Overview
- API Functions
- Tutorial
- FAQ
- More Documents

>



Silkscreen	Function Description
VCC	+
GND	-
SCL	I2C Clock
SDA	I2C Data

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

Silkscreen	Function Description
CSB	BMX160 Protocol Select Pin
INIT1	BMX160 External Interrupt 1
INIT2	BMX160 External Interrupt 2
ADDR	I2C Address Select

[>](#)

API Functions

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

>

```
class DFRobot_BMX160 {
/*
 * @function Gyroscope enum range, unit: G
 */
typedef enum{
    eGyroRange_2000DPS, /*Gyroscope sensitivity at 2000dps*/
    eGyroRange_1000DPS, /*Gyroscope sensitivity at 1000dps*/
    eGyroRange_500DPS,  /*Gyroscope sensitivity at 500dps*/
    eGyroRange_250DPS,  /*Gyroscope sensitivity at 250dps*/
    eGyroRange_125DPS   /*Gyroscope sensitivity at 125dps*/
}eGyroRange_t;

/*
 * @function Accelerometer enum range, unit, m/s^2
 */
typedef enum{
    eAccelRange_2G,  /* Macro for mg per LSB at +/- 2g sensitivity (1 LSB = 0.000061035mg) */
    eAccelRange_4G,  /* Macro for mg per LSB at +/- 4g sensitivity (1 LSB = 0.000122070mg) */
    eAccelRange_8G,  /* Macro for mg per LSB at +/- 8g sensitivity (1 LSB = 0.000244141mg) */
    eAccelRange_16G /* Macro for mg per LSB at +/- 16g sensitivity (1 LSB = 0.000488281mg) */
}eAccelRange_t;

/*
 * @function reset sensor
 * @Return true if it succeeds
 */
bool softReset();

/*
```

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

>

```
* @function init sensor
* @Return true if it succeeds
*/
bool begin();

/*
* @function set gyroscope range, unit: G
* @Parameter One variable from eGyroRange_t
*/
void setGyroRange(eGyroRange_t bits);

/*
* @function set accelerometer range, unit: m/s^2
* @Parameter One variable from eAccelRange_t
*/
void setAccelRange(eAccelRange_t bits);

/*
* @function Get data of accelerometer, gyroscope, geomagnetic sensor
* @Parameter Store the address of all data
*/
void getAllData(struct bmx160SensorData *magn, struct bmx160SensorData *gyro, struct bmx160Se

/*
* @function Turn off geomagnetic sensor, gyroscope enters low power mode(there are data outp
*/
void setLowPower();

/*
* @function Turn on geomagnetic sensor, gyroscope enters normal mode
*/
void wakeUp();
```

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

Tutorial

Visit the I2C address of BMX160 via I2C interface to get the related position data.

Requirements

- **Hardware**

- DFRduino UNO R3 (<https://www.dfrobot.com/product-838.html>) (or similar) x 1
- BMX160 9-axis Sensor Module x1
- Jumper wires

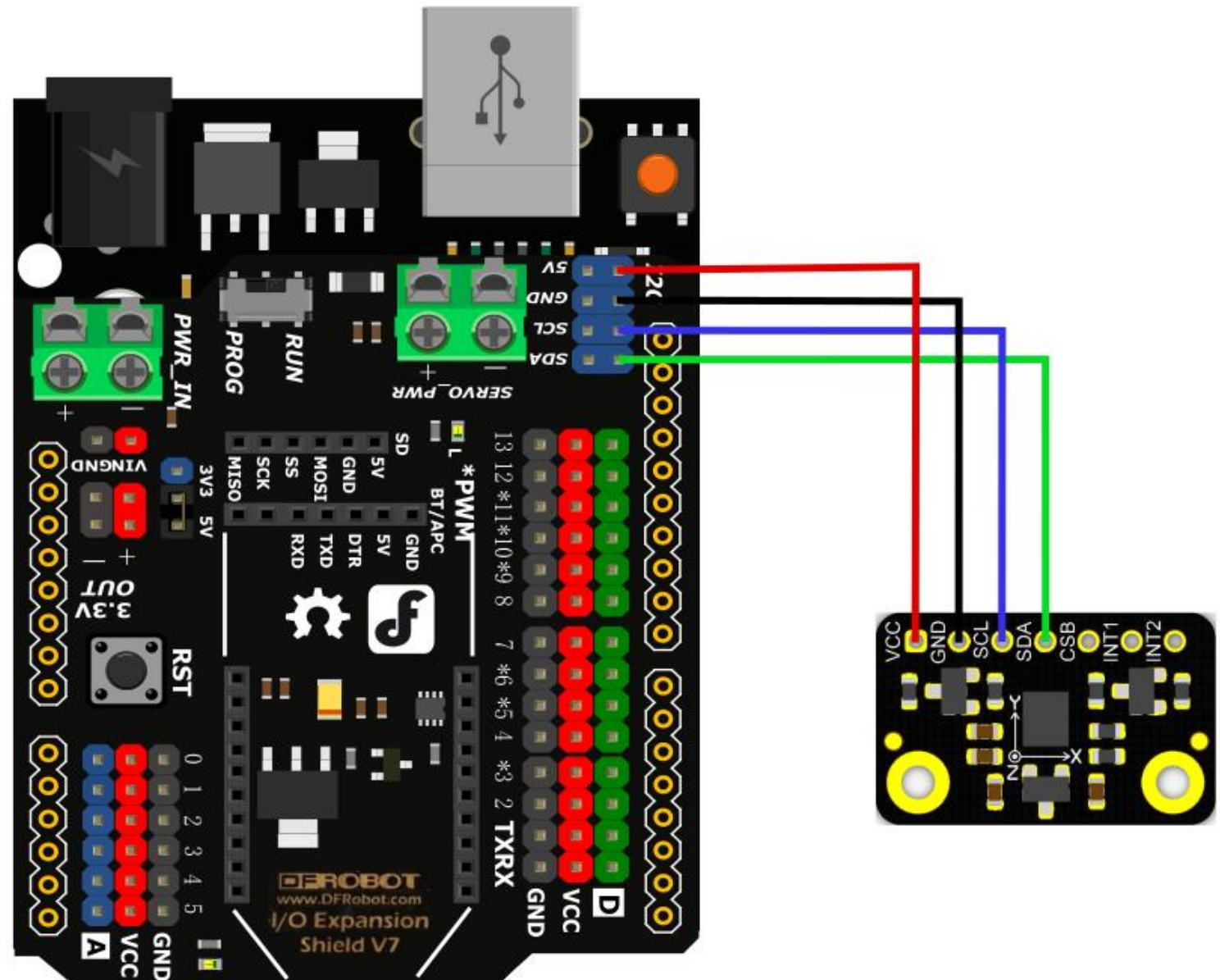
- **Software**

- Arduino IDE (<https://www.arduino.cc/en/Main/Software>)
- Download and install the **BMX160 Library** (https://github.com/DFRobot/DFRobot_BMX160) (About how to install the library? (<https://www.arduino.cc/en/Guide/Libraries#UxU8mdzF9H0>))

>

Connection Diagram

[Introduction](#)
[Features](#)
[Specification](#)
[Board Overview](#)
[API Functions](#)
[Tutorial](#)
[FAQ](#)
[More Documents](#)



BMX160 Tutorial

Function: read data of accelerometer, gyroscope and geomagnetic sensor of BMX160 via I2C interface, and print the readings through serial port.

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

[>](#)

- Introduction
- Features
- Specification
- Board Overview
- API Functions
- Tutorial
- FAQ
- More Documents

>

```

/*!
 * @file readAllData.ino
 * @brief Through the example, you can get the sensor data by using getSensorData:
 * @n      get all data of magnetometer, gyroscope, accelerometer.
 * @n      With the rotation of the sensor, data changes are visible.
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @license The MIT License (MIT)
 * @author [luoyufeng] (yufeng.luo@dfrobot.com)
 * @maintainer [Fary](feng.yang@dfrobot.com)
 * @version V1.0
 * @date 2021-10-20
 * @url https://github.com/DFRobot/DFRobot_BMX160
 */
#include <DFRobot_BMX160.h>

DFRobot_BMX160 bmx160;
void setup(){
    Serial.begin(115200);
    delay(100);

    //init the hardware bmx160
    if (bmx160.begin() != true){
        Serial.println("init false");
        while(1);
    }
    //bmx160.setLowPower(); //disable the gyroscope and accelerometer sensor
    //bmx160.wakeUp(); //enable the gyroscope and accelerometer sensor
    //bmx160.softReset(); //reset the sensor

    /**
     * enum{eGyroRange_2000DPS,
     *      eGyroRange_1000DPS,
     *      eGyroRange_500DPS,
     *      eGyroRange_250DPS,
     *      eGyroRange_125DPS

```

- Introduction
- Features
- Specification
- Board Overview
- API Functions
- Tutorial
- FAQ
- More Documents

>

```

    *      }eGyroRange_t;
    **/
//bmx160.setGyroRange(eGyroRange_500DPS);

/**
 *  enum{eAccelRange_2G,
 *      eAccelRange_4G,
 *      eAccelRange_8G,
 *      eAccelRange_16G
 *      }eAccelRange_t;
 */
//bmx160.setAccelRange(eAccelRange_4G);
delay(100);
}

void loop(){
    sBmx160SensorData_t Omagn, Ogyro, Oaccel;

    /* Get a new sensor event */
    bmx160.getAllData(&Omagn, &Ogyro, &Oaccel);

    /* Display the magnetometer results (magn is magnetometer in uTesla) */
    Serial.print("M ");
    Serial.print("X: "); Serial.print(Omagn.x); Serial.print(" ");
    Serial.print("Y: "); Serial.print(Omagn.y); Serial.print(" ");
    Serial.print("Z: "); Serial.print(Omagn.z); Serial.print(" ");
    Serial.println("uT");

    /* Display the gyroscope results (gyroscope data is in g) */
    Serial.print("G ");
    Serial.print("X: "); Serial.print(Ogyro.x); Serial.print(" ");
    Serial.print("Y: "); Serial.print(Ogyro.y); Serial.print(" ");
    Serial.print("Z: "); Serial.print(Ogyro.z); Serial.print(" ");
    Serial.println("g");

    /* Display the accelerometer results (accelerometer data is in m/s^2) */

```

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

```
Serial.print("A ");
Serial.print("X: "); Serial.print(0accel.x ); Serial.print(" ");
Serial.print("Y: "); Serial.print(0accel.y ); Serial.print(" ");
Serial.print("Z: "); Serial.print(0accel.z ); Serial.print(" ");
Serial.println("m/s^2");

Serial.println("");

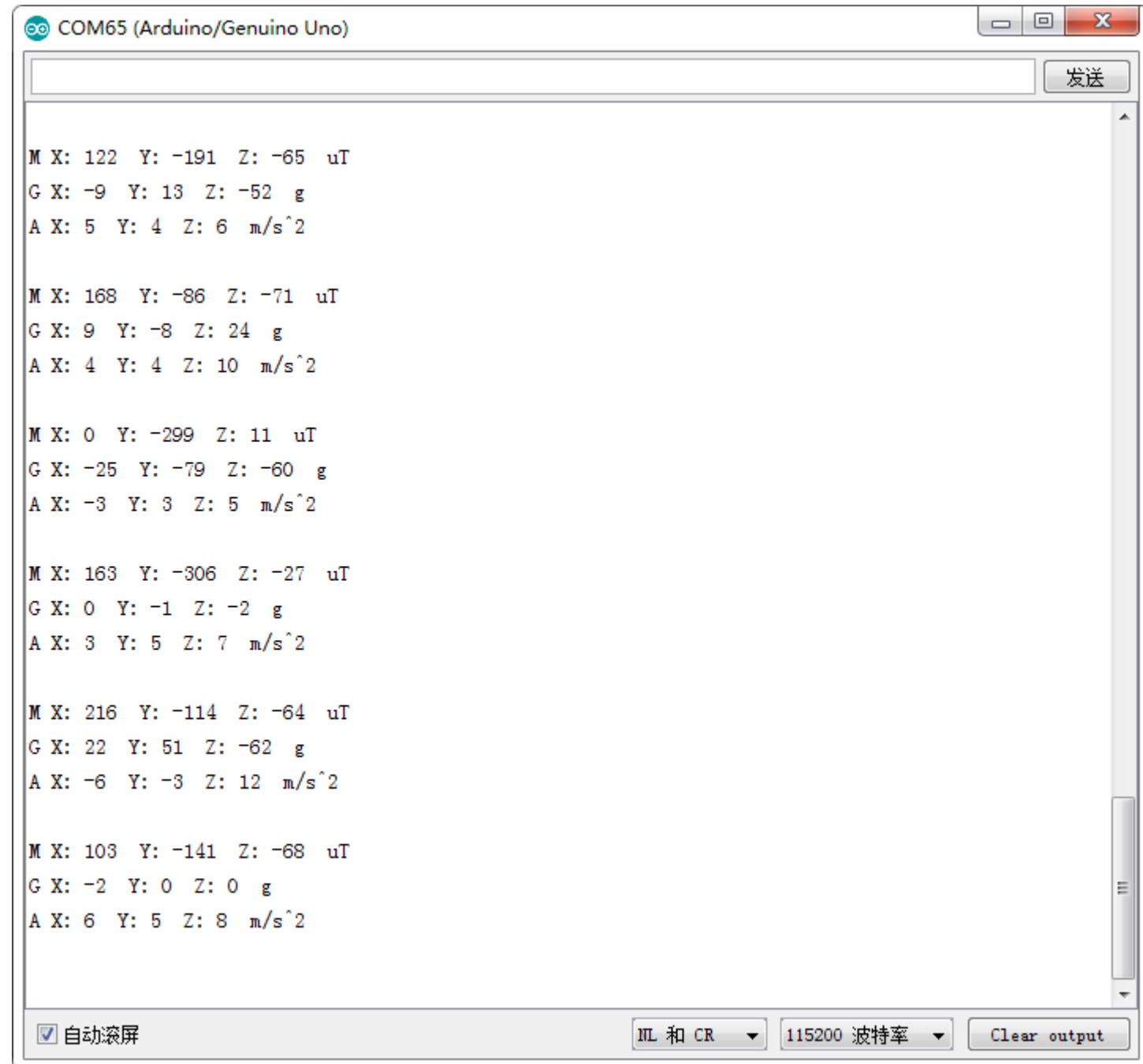
delay(500);
}
```

>

Expected Results

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

>



```
COM65 (Arduino/Genuino Uno)

M X: 122 Y: -191 Z: -65 uT
G X: -9 Y: 13 Z: -52 g
A X: 5 Y: 4 Z: 6 m/s^2

M X: 168 Y: -86 Z: -71 uT
G X: 9 Y: -8 Z: 24 g
A X: 4 Y: 4 Z: 10 m/s^2

M X: 0 Y: -299 Z: 11 uT
G X: -25 Y: -79 Z: -60 g
A X: -3 Y: 3 Z: 5 m/s^2

M X: 163 Y: -306 Z: -27 uT
G X: 0 Y: -1 Z: -2 g
A X: 3 Y: 5 Z: 7 m/s^2

M X: 216 Y: -114 Z: -64 uT
G X: 22 Y: 51 Z: -62 g
A X: -6 Y: -3 Z: 12 m/s^2

M X: 103 Y: -141 Z: -68 uT
G X: -2 Y: 0 Z: 0 g
A X: 6 Y: 5 Z: 8 m/s^2
```

自动滚屏 NL 和 CR 115200 波特率 Clear output

[Introduction](#)[Features](#)[Specification](#)[Board Overview](#)[API Functions](#)[Tutorial](#)[FAQ](#)[More Documents](#)

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (<https://www.dfrobot.com/forum/>).

More Documents

- Schematics Diagram (<https://dfimg.dfrobot.com/nobody/wiki/0f2bdd8aacia02cbd40a61a8ebba9e76.pdf>)
- BMX160 Datasheet (<https://img.dfrobot.com.cn/wiki/none/40e914cf5839ec7f4d10675f34f8f78c.pdf>)

 Get **BMX160 9-axis Sensor Module** (<https://www.dfrobot.com/product-2141.html>) from DFRobot Store or **DFRobot Distributor**. (<https://www.dfrobot.com/distributor>)

Turn to the Top

>