Triple Axis Accelerometer BMA220(Tiny) SKU:SEN0168

Analog Gas Sensor(MQ2) SKU：SEN0127

Contents

Introduction

This Triple Axis Accelerometer with Bosch BMA220 is an ultra small triaxial, low-g acceleration sensor breakboard with I2C interface, aiming for lowpower consumer market applications. It allows measurement of accelerations in 3 perpendicular axes and thus senses tilt, motion, shock and vibration in cell phones, handhelds, computer peripherals, man-machine interfaces, virtual reality features and game controllers.

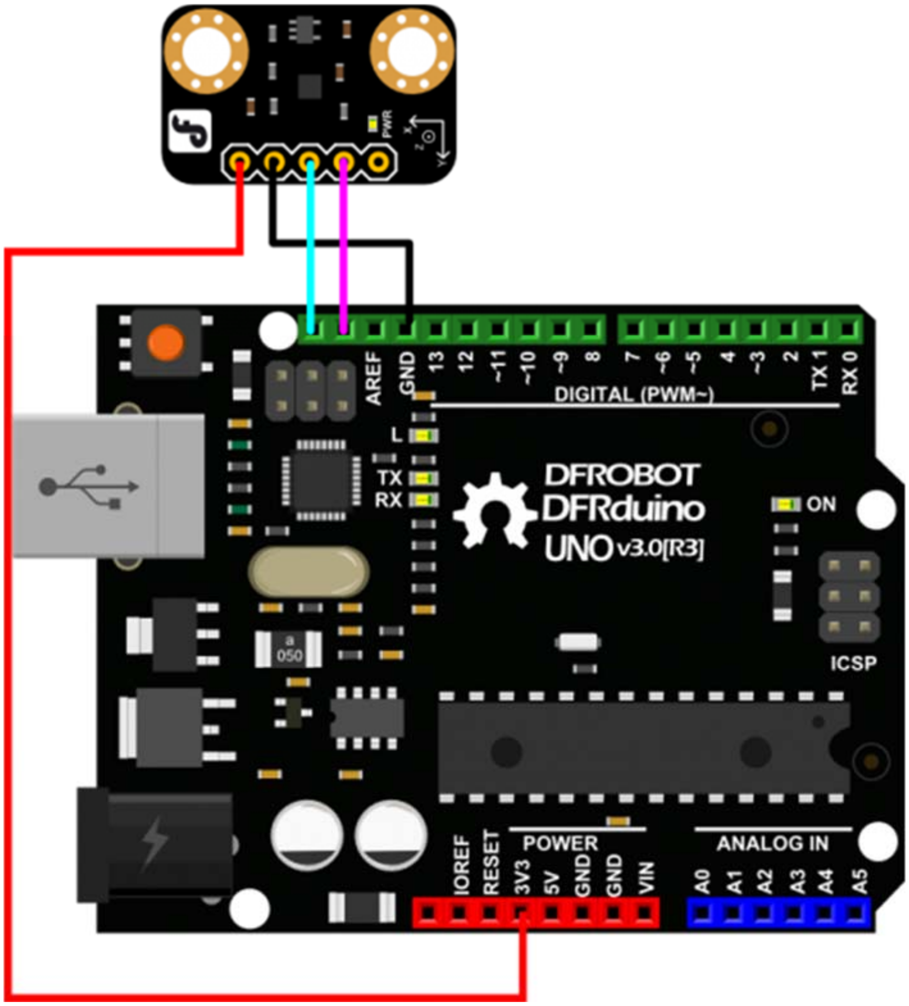
With a size of only 2 mm x 2 mm, the Bosch BMA220 represents a new generation of acceleration sensors. The Tri-Axis Accelerometer integrates a multitude of features that acilitates its use especially in the area of motion detection applications, such as device orientation detection, gaming, HMI and menu browser control. It is highly configurable in order to give the designer full flexibility when integrating the sensor into his system.

It can be used in sensing tilt, motion and shock vibrati
By the way, we have collected some useful 3-axis data processing methods: How to Use a Three-Axis Accelerometer for Tilt Sensing.

Specification

- Power supply: 2.0-3.6V
- Interface: I2C
- Acceleration range:±2g/±4g/±8g/±16g
- Ultra Low Power
- LED power indication
- Tiny size design and easy-to-use
- Compatible with Arduino controllers
- Size: 13x22mm

Connecting Diagram



{

Sample Code

```
#include <Wire.h>
byte Version[3];
int8_t x_data;
int8_t y_data;
int8_t z_data;
byte range=0x00;
float divi=16;
float x,y,z;
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  Wire.beginTransmission(0x0A); // address of the accelerometer
  // range settings
  Wire.write(0x22); //register address
  Wire.write(range); //can be set at"0x00""0x01""0x02""0x03", refer to Datashhet on wiki
  // low pass filter
  Wire.write(0x20); //register address
  Wire.write(0x05); //can be set at"0x05""0x04"......"0x01""0x00", refer to Datashhet on wiki
  Wire.endTransmission();
}

void AccelerometerInit()
{
  Wire.beginTransmission(0x0A); // address of the accelerometer
  // reset the accelerometer
  Wire.write(0x04); // Y data
  Wire.endTransmission();
  Wire.requestFrom(0x0A,1);    // request 6 bytes from slave device #2
  while(Wire.available())    // slave may send less than requested
  {
      Version[0] = Wire.read(); // receive a byte as characte
    }
    x_data=(int8_t)Version[0]>>2;

    Wire.beginTransmission(0x0A); // address of the accelerometer
    // reset the accelerometer
    Wire.write(0x06); // Y data
    Wire.endTransmission();
    Wire.requestFrom(0x0A,1);    // request 6 bytes from slave device #2
    while(Wire.available())    // slave may send less than requested
    {
      Version[1] = Wire.read(); // receive a byte as characte
    }
    y_data=(int8_t)Version[1]>>2;

    Wire.beginTransmission(0x0A); // address of the accelerometer
    // reset the accelerometer
    Wire.write(0x08); // Y data
    Wire.endTransmission();
    Wire.requestFrom(0x0A,1);    // request 6 bytes from slave device #2
     while(Wire.available())    // slave may send less than requested
    {
      Version[2] = Wire.read(); // receive a byte as characte
    }
     z_data=(int8_t)Version[2]>>2;

     x=(float)x_data/divi;
     y=(float)y_data/divi;
     z=(float)z_data/divi;
     Serial.print("X=");
     Serial.print(x);         // print the character
     Serial.print("  ");
     Serial.print("Y=");
```

```
   Serial.print(y);         // print the character
   Serial.print("  ");
   Serial.print("Z=");           // print the character
   Serial.println(z);
}

void loop()
{
  switch(range)  //change the data dealing method based on the range u've set
  {
  case 0x00:divi=16;  break;
  case 0x01:divi=8;  break;
  case 0x02:divi=4;  break;
  case 0x03:divi=2;  break;
  default: Serial.println("range setting is Wrong,range:from 0to 3.Please check!");while(1);
  }
  AccelerometerInit();
 delay(100);

}
```