## SKU:SEN0253 (https://github.com/DFRobot/Wiki/raw/master/Resource/Euler%20angle%20visual%20tool%5BV1.0%5D.rar)

## Introduction

BNO055 is a new sensor IC for implementing an intelligent 9-axis Absolute Orientation Sensor. It is a system in package, integrating a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope, a triaxial geomagnetic sensor and a 32-bit microcontroller. At just 5.2 x 3.8 x 1.1 mm<sup>3</sup>, it is significantly smaller than comparable discrete or system-on-board solutions and also is the sensor-hub product of the smallest size that supports Windows 8.1 at present. BNO055 is able to provide not only single data of the three kinds of sensors(accelerometer/gyroscope/geomagnetic), but also fused data, such as quaternions, Euler angles or vectors. Besides, the built-in MCU frees the users from the complexities of algorithm processing, which provides application support in many aspects for smart phone, wearable device and so on.

BMP280 is an absolute barometric pressure sensor especially designed for mobile applications, which can realize the measurement of barometric pressure and temperature(the data can be converted into altitude through the specific formula ). The sensor module is housed in an extremely compact package. It is based on Bosch's proven Piezo resistive pressure sensor technology featuring high accuracy and linearity as well as long term stability and high EMC robustness. Numerous device operation options offer highest flexibility to optimize the device regarding power consumption, resolution and filter performance.

Now, DFRobot is launching Gravity: BNO055 BMP280 intelligent 10DOF AHRS. This sensor module integrates BNO055 and BMP280 on one board to combine the two sensor into a 10DOF sensor module. The standard Gravity-I2C interface eases the integration process for customers, freeing them from the complexities of multivendor solutions so they can spend more time on product innovation, including novel applications such as wearable hardware. It is also the perfect choice for augmented reality, more immersive gaming, personal health and fitness, indoor navigation and any other application requiring context awareness.

### Features

- BNO055:
  - Outputs fused sensor data: quaternions, euler angles, rotation vector, linear acceleration, gravity, heading.
  - 3 sensors in one device: 16-bit gyroscope, 14-bit accelerometer, geomagnetic sensor
  - Intelligent Power Management: normal, low power and suspend mode available
- BMP280:
  - Barometric pressure & Temperature sensor

# Specification

- Operating Voltage: 3.3V~5V DC
- Operating Current: 5mA
- Gravity-I2C Interface
- BNO055 Accelerometer:
  - Acceleration ranges ±2g/±4g/±8g/±16
  - Low-pass filter bandwidths 1kHz~<8Hz
  - Operation modes: normal, suspend, low power, standby, deep suspend
  - On-chip interrupt control: motion-triggered interrupt-signal
- BNO055 Gyroscope:
  - Ranges switchable from ±125°/s~2000°/s
  - Low-pass filter bandwidths 523Hz~12Hz
  - Operation modes: normal, fast power up, deep suspend, suspend, advanced power save.
  - On-chip interrupt control: motion-triggered interrupt-signal
- BNO055 Geomagnetic:
  - Magnetic field range typical ±1300uT(x-,y-axis);±2500uT(z-axis)
  - Magnetic field resolution: ~0.3
  - Operating nodes: low power, regular, enhanced regular, high accuracy
  - Power modes: normal. sleep. suspend. force

- BMP280 Digital Pressure Sensor:
  - Pressure range: 300~1100hPa
  - Relative accuracy: ±0.12hPa (±1m)
  - Absolute accuracy: ±1hPa (±8.33m)
  - Temperature range: 0°C~65°C
  - Temperature resolution: 0.01°C
- Operating Temperature: -40~ 80°C
- Product Dimension: 32x27 mm/1.26x1.06"
- For more information, please refer to the attached data sheets of BMP280 and BNO055.\*

## **Pinout and Dimension Diagram**

Serial Number	Name	Functionality
1	/VCC	Positive Pole
2	-/GND	Negative Pole
3	С	I2C-SCL
4	D	I2C-SDA
5	NBOOT	Boot Mode
6	RST	Reset Pin
7	INT	Interrupt Output Pin
8	I2C_ADDR	BNO055 I2C address Selection
9	PS2	Protocol Selection Pin 2

Serial Number	Name PS1	Functionality Protocol Selection Pin 1
11	BL_IND	Bootstrap Instructions

PS1	PS2	Functionalit
0	0	Standard/Fast 12C Interface
0	1	HID OVER I2C
1	0	UART Interface
1	1	Reserved

Note: the I2C address of BNO055 is set to 0×28 by default. The I2C address of BMP280 is 0×76; the I2C\_ADDR of BNO055 can be set as 0x28/0x29. PS1 and PS2 are set to 0 and 0 by default.

# **API Function**

```
/*!
 * @The functionality of the function: initialize the SEN0253 device
 *
 * @return value: bool type
    true: initializing the device fails
 *
     false: initializing the device succeeds
 *
 */
bool init();
/*!
 * @The functionality of the function: set the mode of BNO055 device
 *
 *
   @Parameter:
                powerMode: set the power mode
 *
                 dataMode: set the data transmission rate
 */
void setMode(eBN0055PowerModes_t powerMode, eBN0055DataRateMode_t dataRate);
/*!
* @The functionality of the function: read the value of Euler angle and store it into the struct of BN0055EulerData.
 */
void readEuler(void);
/*!
 * @The functionality of the function: read the value of angular velocity and store it into the struct of BN0055GyrDate.
*/
void readAngularVelocity(void);
/*!
 * @The functionality of the function: read the value of linear acceleration and store it into the struct of BN0055LinAccData.
 */
void readLinAcc(void);
/*!
 * @The functionality of the function: read the value of quaternion and store it into the struct of BN0055QuaData.
```

```
*/
void readQua(void);
/*!
 * @The functionality of the function: use the final values of quaternion and linear
acceleration to calculate the final position and store the position information into the struct of BNO055AbsLinAccData.
 */
void calcAbsLinAcc(void);
```

# Tutorial

10 DOF integrates BNO055 and BMP280. The I2C address of BNO055 (0×28) and BMP280 (0×76) can be visited through I2C interface, which makes it available to obtain the related position data and environment information.

### Preparation

### Hardware

- 1×UNO microcontroller Board
- 1×BNO055 BMP280 intelligent 10DOF AHRS(V1.0) Module
- DuPont lines

### Software

- Arduino IDE, click to download Arduino IDE (https://www.arduino.cc/en/Main/Software)
- BNO055 Library (https://github.com/DFRobot/DFRobot\_BNO055)
- BMP280 Library (https://github.com/DFRobot/DFRobot\_BMP280.git) How to install library files? Click the link. (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0)

# **Connection Diagram**

SEN0253 V1.0 Connection Diagram

The function of the program: read the pitch angle, roll angle and yaw angle of BNO055 sensor via I2C interface, and print out the data through the serial port. Using this demo with a small visual software Euler angle visual tool.exe (https://github.com/DFRobot/Wiki/raw/master/Resource/Euler%20angle%20visual%20tool%5BV1.0%5D.rar) we specifically designed, you can directly observe the attitude variation of 10DOF. As shown below.

```
/*
 * file EulerAngleDemo.ino
 *
 * @ https://github.com/DFRobot/DFRobot BN0055
 *
 * connect BNO055 I2C interface with your board (please reference board compatibility)
 *
 * This example can be used to display the gesture on dof10.exe
 *
 * Copyright [DFRobot](https://www.dfrobot.com), 2016
 * Copyright GNU Lesser General Public License
 *
 * version V0.1
 * date 2018-1-8
 */
#include "DFRobot_BN0055.h"
DFRobot BN0055 mpu;
void setup()
{
   Serial.begin(115200);
   while (!mpu.init())
   {
     Serial.println("ERROR! Unable to initialize the chip!");
     delay(50);
   }
     mpu.setMode(mpu.eNORMAL_POWER_MODE, mpu.eFASTEST_MODE);
11
   delay(100);
   Serial.println("Read euler angles...");
}
void loop()
{
```

```
mpu.readEuler(); /* read euler angle */
/* In order to match the API of the upper computer, X ----> pitch */
Serial.print("pitch:");
Serial.print(mpu.EulerAngles.y, 3);
Serial.print(" ");
/* In order to match the API of the upper computer, Y ----> roll */
Serial.print("roll:");
Serial.print(mpu.EulerAngles.z, 3);
Serial.print(" ");
/* In order to match the API of the upper computer, Z ----> yaw */
Serial.print("yaw:");
Serial.print(mpu.EulerAngles.x, 3);
Serial.print(" ");
delay(80);
```

If we compare 10DOF to an airplane whose nose points at due east, the positive direction of X axis will be the direction of the nose, the positive direction of Y axis will be the direction of the left wing, which is due north. Z axis is perpendicular to the plane XOY that formed by X and Y axes. When the 10 DOF's direction of X, Y, and Z totally coincides with the above-mentioned direction, the values of the pitch, roll and yaw angle are 0°. Here we define: pitch is the angle between the nose and XOY when the airplane noses up or down along the Y axis, and nose up is positive while nose down is negative; roll is the angle between the body and XOY when the airplane rolls along the X axis; yaw is the angle between the nose and XOZ when the airplane moves along the Z axis.

Please note that you need to close the serial port occupied by the printer when using the test software to observe the sensor's movement posture.

#### Sample Code

}

The function of the program: get the acceleration data of the sensor's movement on X, Z and Y, and print it out through the serial port.

```
/*
 * file AccelerationDemo.ino
 *
 * @ https://github.com/DFRobot/DFRobot BN0055
 *
 * connect BNO055 I2C interface with your board (please reference board compatibility)
 *
 * Gets the current acceleration of the sensor and prints it out through the serial port.
 *
 * Copyright [DFRobot](https://www.dfrobot.com), 2016
 * Copyright GNU Lesser General Public License
 *
 * version V0.1
 * date 2018-1-8
 */
#include <Wire.h>
#include "DFRobot_BN0055.h"
DFRobot_BN0055 mpu;
void setup()
{
   Serial.begin(115200);
   while (!mpu.init())
   {
     Serial.println("ERROR! Unable to initialize the chip!");
     delay(50);
   }
     mpu.setMode(mpu.eNORMAL POWER MODE, mpu.eFASTEST MODE);
11
   delay(100);
   Serial.println("Read acceleration...");
}
```

```
void loop()
{
    mpu.readAcc(); /* read acceleration */
    Serial.print("X: ");
    Serial.print(mpu.AccData.x,3);
    Serial.print(" m/s^2 ");
    Serial.print(mpu.AccData.y,3);
    Serial.print(" m/s^2 ");
    Serial.print("Z: ");
    Serial.print(mpu.AccData.z,3);
    Serial.println(" m/s^2 ");
    delay(200);
}
```

The function of the program: get the linear acceleration of the sensor's movement on X, Z and Y( exclude the value generated under the influence of gravity acceleration), and print it out through the serial port. The values of X, Y and Z respectively represents the changes of velocity of 10 DOF on X, Y and Z. It can be described as m/s^2. " " and "-" shows the increase and decrease of the velocity respectively. When the sensor keeps still, the values of X, Y and Z approximate to 0g. Nothing in the world can keep absolutely still, so the acceleration of the sensor oscillates around 0.

```
/*
 * file LinearAccelerationDemo.ino
 *
 * @ https://github.com/DFRobot/DFRobot BN0055
 *
 * connect BNO055 I2C interface with your board (please reference board compatibility)
 *
 * Gets the current linear acceleration of the sensor and prints it out through the serial port.
 *
 * Copyright [DFRobot](https://www.dfrobot.com), 2016
 * Copyright GNU Lesser General Public License
 *
 * version V0.1
 * date 2018-1-8
 */
#include <Wire.h>
#include "DFRobot_BN0055.h"
DFRobot_BN0055 mpu;
void setup()
{
   Serial.begin(115200);
   while (!mpu.init())
   {
     Serial.println("ERROR! Unable to initialize the chip!");
     delay(50);
   }
     mpu.setMode(mpu.eNORMAL POWER MODE, mpu.eFASTEST MODE);
11
   delay(100);
   Serial.println("Read linear acceleration...");
}
```

```
void loop()
{
    mpu.readLinAcc(); /* read linear acceleration */
    Serial.print("X: ");
    Serial.print(mpu.LinAccData.x,3);
    Serial.print(" m/s^2 ");
    Serial.print(mpu.LinAccData.y,3);
    Serial.print(" m/s^2 ");
    Serial.print("Z: ");
    Serial.print(mpu.LinAccData.z,3);
    Serial.println(" m/s^2 ");
    delay(200);
}
```

The function of the program: get the angular velocity of the sensor's movement round X, Y and Z, and print it out through the serial port. X, Y and Z respectively represent the angle that the sensor rotates round the X, Y and Z in unit time. It can be described as dps. " " and "-" respectively shows the sensor's forward-rotating and reverse around the axis.

```
/*
 * file EulerAngleDemo.ino
 *
 * @ https://github.com/DFRobot/DFRobot BN0055
 *
 * connect BNO055 I2C interface with your board (please reference board compatibility)
 *
 * Gets the Angular Velocity of the current sensor and prints it out through the serial port.
 *
 * Copyright [DFRobot](https://www.dfrobot.com), 2016
 * Copyright GNU Lesser General Public License
 *
 * version V0.1
 * date 2018-1-8
 */
#include <Wire.h>
#include "DFRobot_BN0055.h"
DFRobot_BN0055 mpu;
void setup()
{
   Serial.begin(115200);
   while (!mpu.init())
   {
     Serial.println("ERROR! Unable to initialize the chip!");
     delay(50);
   }
     mpu.setMode(mpu.eNORMAL POWER MODE, mpu.eFASTEST MODE);
11
   delay(100);
   Serial.println("Read Angular Velocity...");
}
```

```
void loop()
```

```
{
  mpu.readAngularVelocity(); /* read Angular Velocity */
  Serial.print("X: ");
  Serial.print(mpu.GyrData.x, 3);
  Serial.print(" dps ");
  Serial.print(mpu.GyrData.y, 3);
  Serial.print(" dps ");
  Serial.print("mpu.GyrData.z, 3);
  Serial.println(" dps ");
  delay(200);
}
```

The function of the program: read the values of the temperature, barometric pressure and altitude 10 DOF module collected, and print it out over the serial port. Their measurement units are °C, pa and m. The altitude is calculated from the temperature and pressure BMP280 detected, which may lead to a certain amount of error. The indoor temperature of that day is 26.33 °C, and the pressure is 952hpa. The altitude range of the region is 511.3~647.4m.

```
DFRobot BMP280 bmp;
void setup() {
  Serial.begin(115200);
 while (!bmp.begin()) {
   Serial.println(F("Error initializing BMP280!"));
   delay(1000);
  }
}
void loop() {
  Serial.print("Temperature:");
 Serial.println(bmp.readTemperatureValue());
 Serial.print("Pressure:");
 Serial.println(bmp.readPressureValue());
 Serial.print("Altitude:");
  Serial.println(bmp.readAltitudeValue());
 /* use a sampling time of at least 38ms */
  delay(50);
}
```

### **More Documents**

#include <DFRobot\_BMP280.h>

Schematic Diagram (https://github.com/Strictus/DFRobot/raw/master/SEN0253/%5BSEN0253%5D(V1.0)-SCH.PDF)

BNO055 Datasheet (https://github.com/Strictus/DFRobot/raw/master/SEN0253/BST\_BNO055\_DS000\_14.pdf)

RNAD222 Datashaat (https://raw.aithubusarsantant.com/Strictus/DEDahat/mastar/SENIA222/RST\_RNAD222 DSAAA

טוער געט שמנמאופט (וונוףג//ומיי.עווועטעפו נטוופווג.נטוון געוונעגן שראטטטל וומגופו געועב געט - טוער געט ו.עען

Get Gravity: BNO055 BMP280 intelligent 10 DOF AHRS (https://www.dfrobot.com/product-1793.html) from DFRobot Store or DFRobot Distributor. (https://www.dfrobot.com/index.php?route=information/distributorslogo)