

BMI160からBMI270への  
移行について

# 目次

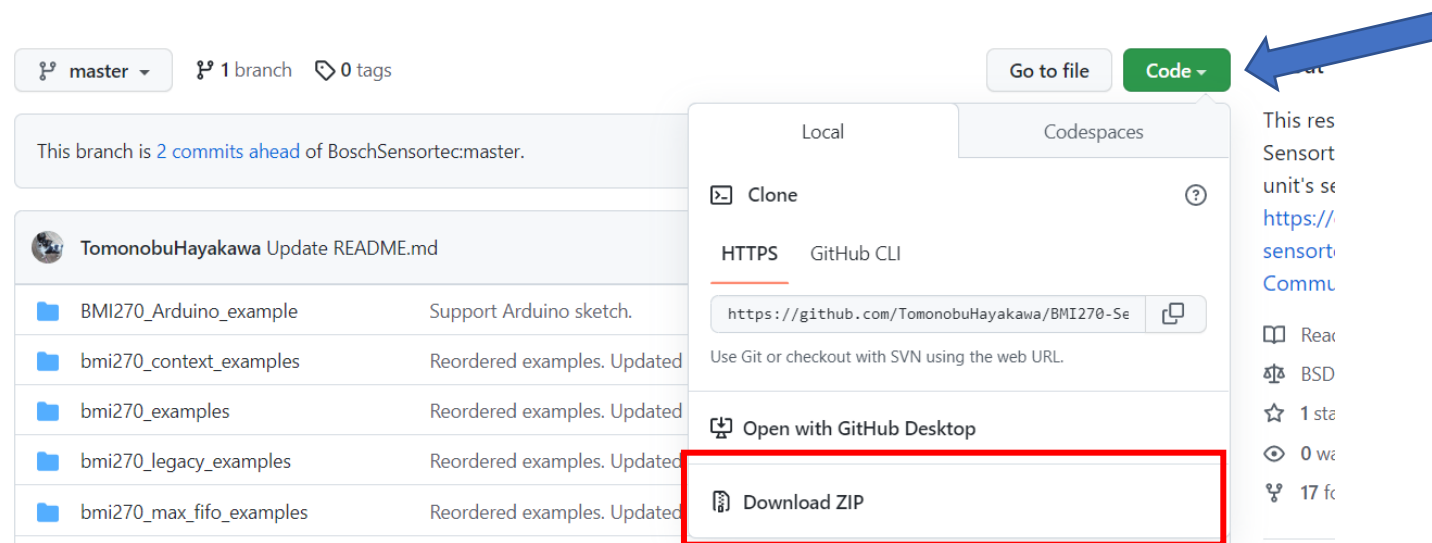
- 概要
- 事前準備
- API変更
  - センサ初期化
  - データ取得
  - データ変換

# 概要

- 本資料はArduino環境において、BMI160サンプルをBMI270ライブラリで書き直し、Spresenseで動かすための注意事項です。
- 本資料は、BMI270ライブラリをダウンロード・利用する方向けの資料です。
- 本資料は次のOS、ソフトウェア環境に基づいて作成されたものです。
  - Windows 10
  - Arduino IDE v1.8.19
  - Spresense 2.6.0

# 事前準備

1. [BMI160ライブラリ](#)、[BMI270ライブラリ](#)をGithubでダウンロードしてください。



[Refs]

<https://github.com/TomonobuHayakawa/BMI270-Sensor-API>

2. Arduinoでメニューバー>スケッチ>ライブラリをインクルード>「.ZIP形式のライブラリをインストール」の順でダウンロードしたzipファイルを読み込みます。
3. 下記サンプルを開きます。
  1. メニューバー>スケッチ>BMI160-Arduino-master>Gyro
  2. メニューバー>スケッチ>BMI270-Sensor-API-master>BMI270\_Arduino\_example > accel\_gyro◆ 本資料は上記サンプルを例に説明します。

## API変更--センサ初期化

```
1  /*
2  * Copyright (c) 2016 Intel Corporation. All rights reserved.
3  * See the bottom of this file for the license terms.
4  */
5
6  /*
7   This sketch example demonstrates how the BMI160 on the
8   Intel(R) Curie(TM) module can be used to read gyroscope data
9  */
10
11 #include <BMI160Gen.h>
12
13 void setup() {
14   Serial.begin(9600); // initialize Serial communication
15   while (!Serial);    // wait for the serial port to open
16
17   // initialize device
18   Serial.println("Initializing IMU device...");
19   BMI160.begin(BMI160GenClass::SPI_MODE, /* SS pin# = */10);
20   //BMI160.begin(BMI160GenClass::I2C_MODE);
21   uint8_t dev_id = BMI160.getDeviceID();
22   Serial.print("DEVICE ID: ");
23   Serial.println(dev_id, HEX);
24
25   // Set the accelerometer range to 250 degrees/second
26   BMI160.setGyroRange(250);
27   Serial.println("Initializing IMU device...done.");
28 }
```

BMI160のGyroサンプル

# API変更--センサ初期化

- ヘッダーファイルを変更します。  
`#include <BMI270_Arduino.h>`
- BMI270ライブラリのインスタンスを宣言します。  
`BMI270Class BMI270;`
- ボーレートを変更します。  
`Serial.begin(115200);`
- I2C接続、センサコンフィギュレーション。  
`int8_t rslt = BMI270.begin(BMI270_I2C, BMI2_I2C_SEC_ADDR);`  
`rslt = configure_sensor();`
- BMI270では該当APIが存在せず、**削除します**。
- ジャイロセンサ測定時のrangeの設定です。書き直す方法は[ここ](#)で説明します。

```
1 /*
2  * Copyright (c) 2016 Intel Corporation. All rights reserved.
3  * See the bottom of this file for the license terms.
4  */
5
6 /*
7  This sketch example demonstrates how the BMI160 on the
8  Intel(R) Curie(TM) module can be used to read gyroscope data
9  */
10
11 #include <BMI160Gen.h>
12
13 void setup() {
14   Serial.begin(9600); // initialize Serial communication
15   while (!Serial); // wait for the serial port to open
16
17   // initialize device
18   Serial.println("Initializing IMU device...");
19   BMI160.begin(BMI160GenClass::SPI_MODE, /* SS pin# = */10);
20   //BMI160.begin(BMI160GenClass::I2C_MODE);
21   uint8_t dev_id = BMI160.getDeviceID();
22   Serial.print("DEVICE ID: ");
23   Serial.println(dev_id, HEX);
24
25   // Set the accelerometer range to 250 degrees/second
26   BMI160.setGyroRange(250);
27   Serial.println("Initializing IMU device...done.");
28 }
```

(続く)

BMI160のGyroサンプル

# API変更--センサ初期化

## Range設定について

- BMI160では、センサのrange設定はAPI化されたため、設定時に `setAccelerometerRange(range)` や `setGyroRange(range)` を使って、数値を指定するだけで終わります。
- BMI270ライブラリにおいては、センサのrange設定はAPI化されておらず、特定の箇所を修正する必要があります。

26 | `BMI160.setGyroRange(250);`

BMI160のGyroサンプル

(続く)

# API変更--センサ初期化

- BMI270のacc\_gyroサンプルの34行と62行は加速度センサとジャイロセンサのrange設定を行います。
- それぞれのセンサのrangeに設定できるパラメーターは下記のとおりです。
  - 加速度センサ
    - BMI2\_ACC\_RANGE\_2G
    - BMI2\_ACC\_RANGE\_4G
    - BMI2\_ACC\_RANGE\_8G
    - BMI2\_ACC\_RANGE\_16G
  - ジャイロセンサ
    - BMI2\_GYR\_RANGE\_125
    - BMI2\_GYR\_RANGE\_250
    - BMI2\_GYR\_RANGE\_500
    - BMI2\_GYR\_RANGE\_1000
    - BMI2\_GYR\_RANGE\_2000

```
33  /* Gravity range of the sensor (+/- 2G, 4G, 8G, 16G). */
34  config[0].cfg.acc.range = BMI2_ACC_RANGE_2G;
35
36  /* The bandwidth parameter is used to configure the number of sensor
37   * if it is set to 2, then 2^(bandwidth parameter) samples
38   * are averaged, resulting in 4 averaged samples.
39   * Note1 : For more information, refer the datasheet.
40   * Note2 : A higher number of averaged samples will result in a lower
41   * this has an adverse effect on the power consumed.
42   */
43  config[0].cfg.acc.bwp = BMI2_ACC_NORMAL_AVG4;
44
45  /* Enable the filter performance mode where averaging of samples
46   * will be done based on above set bandwidth and ODR.
47   * There are two modes
48   * 0 -> Ultra low power mode
49   * 1 -> High performance mode(Default)
50   * For more info refer datasheet.
51   */
52  config[0].cfg.acc.filter_perf = BMI2_PERF_OPT_MODE;
53
54  /* Configure the type of feature. */
55  config[1].type = BMI2_GYRO;
56
57  /* The user can change the following configuration parameters according
58   * Set Output Data Rate */
59  config[1].cfg.gyr.odr = BMI2_GYR_ODR_100HZ;
60
61  /* Gyroscope Angular Rate Measurement Range.By default the range is :
62  config[1].cfg.gyr.range = BMI2_GYR_RANGE_2000;
```

BMI270のacc\_gyroサンプル

(続く)



## API変更--センサ初期化（まとめ）

	BMI160	BMI270
センサライブラリ	#include <BMI160Gen.h>	#include <BMI270_Arduino.h>
その他ライブラリ	なし	BMI270ライブラリのインスタンス宣言： BMI270Class BMI270;
ボーレート	115200	115200
I2C、センサ 初期化	BMI160.begin(BMI160GenClass::I2C_MODE);	int8_t rslt = BMI270.begin(BMI270_I2C,BMI2_I2C_SEC_ADDR); rslt = configure_sensor();
Range設定	setAccelerometerRange(range); setGyroRange(range);	8ページを参考してください。
センサのサンプリング レート取得	加速度： getAccelerometerRate();  ジャイロ： getGyroRate();	struct bmi2_sens_config bmi270_config[2];  bmi270_config[0].type = BMI2_ACCEL; bmi270_config[1].type = BMI2_GYRO;  int8_t config_rslt = BMI270.get_sensor_config(bmi270_config, 2);  加速度： bmi270_config[0].cfg.acc.odr  ジャイロ： bmi270_config[1].cfg.gyro.odr

# API変更--データ取得

- BMI160は
  - readAccelerometer(*int& ax, int& ay, int& az*)
  - readGyro(*int& gx, int& gy, int& gz*)

関数で加速度センサ・ジャイロセンサのデータを取得します。取得したデータは下記変数に保存します。

- 加速度センサ： ax, ay, az
- ジャイロセンサ： gx, gy, gz

- BMI270は
  - bmi2\_get\_sensor\_float(*bmi2\_sens\_float &sensor\_data*)

関数で加速度センサ・ジャイロセンサのデータを**一括**取得します。取得された数値は下記変数に保存します。

- 加速度センサ
  - sensor\_data.acc.x
  - sensor\_data.acc.y
  - sensor\_data.acc.z
- ジャイロセンサ
  - sensor\_data.gyr.x
  - sensor\_data.gyr.y
  - sensor\_data.gyr.z

```
31 | int gxRaw, gyRaw, gzRaw;           // raw gyro values
32 | float gx, gy, gz;
33 |
34 | // read raw gyro measurements from device
35 | BMI160.readGyro(gxRaw, gyRaw, gzRaw);
```

BMI160のGyroサンプル

```
113 | struct bmi2_sens_float sensor_data;
114 | int8_t rslt = BMI270.bmi2_get_sensor_float(&sensor_data);
```

BMI270のacc\_gyroサンプル

## API変更--データ取得（まとめ）

	BMI160	BMI270
データ取得API	加速度： <code>readAccelerometer(int&amp; <b>ax</b>, int&amp; <b>ay</b>, int&amp; <b>az</b>)</code>  ジャイロ： <code>readGyro(int&amp; <b>gx</b>, int&amp; <b>gy</b>, int&amp; <b>gz</b>)</code>	加速度・ジャイロ一括： <code>bmi2_get_sensor_float(<i>bmi2_sens_float</i> &amp;<b>sensor_data</b>)</code>
データの保存先	加速度： <code>ax, ay, az</code>  ジャイロ： <code>gx, gy, gz</code>	加速度： <code>sensor_data.acc.x</code> <code>sensor_data.acc.y</code> <code>sensor_data.acc.z</code>  ジャイロ： <code>sensor_data.gyr.x</code> <code>sensor_data.gyr.y</code> <code>sensor_data.gyr.z</code>

# API変更--データ変換

- BMI160とBMI270は仕様上、APIを通じて取得してきたデータはrawデータ（16bitで表示され、つまり範囲は-32768～32767）であり、データを分析するには、事前に設定したrangeで変換する必要があります。
- BMI160とBMI270のrawデータ変換式の仕様は同じです。
  - 計測データ = Rawデータ/32768 × Range [単位]
  - 加速度の単位は  $g$  で、角速度の単位は  $^{\circ}/s$  です。
- BMI160はジャイロデータ変換APIのみ( convertRawGyro() )提供しているため、加速度データ変換のAPIはユーザーで作成する必要があります。
- BMI270ライブラリでは加速度データを実数値で表示するため、BMI270加速度データの変換式は次のとおりです。
  - 加速度データ =  $g \times \text{Rawデータ} / 32768 \times \text{Range [m/s}^2]$
  - $g$ は重力加速度 =  $9.80665 \text{ m/s}^2$
- BMI270ライブラリに使われるAPIで取得したデータはすでに**変換済み**です。

```
54 float convertRawGyro(int gRaw) {  
55     // since we are using 250 degrees/seconds range  
56     // -250 maps to a raw value of -32768  
57     // +250 maps to a raw value of 32767  
58  
59     float g = (gRaw * 250.0) / 32768.0;  
60  
61     return g;  
62 }
```

BMI160のGyroサンプル

## API変更--データ変換（まとめ）

	BMI160	BMI270
加速度データ変換	<p>ユーザーの自作が必要です。例：</p> <pre>float convertRawAcc(int <b>aRaw</b>){     float a = (aRaw * aRange) / 32768.0;     return a; }</pre> <p>単位：g</p> <p>aRaw： 加速度rawデータ aRange： 加速度range</p>	なし
ジャイロデータ変換	<pre>float convertRawGyro(int <b>gRaw</b>)</pre> <p>単位：°/s</p> <p>gRaw： ジャイロrawデータ</p>	なし