Technical Document

How to do BMA253 incline calibration

Bosch Sensortec

BOSCH

Invented for life

Table of contents

1 Introduction

BMA253 is 12-bit 3-axis digital accelerometer. It has built-in four offset compensation methods which are called slow compensation, fast compensation, manual compensation and inline calibration. The difference among them is as shown below.

- Slow compensation: is actually a high-pass filter. If this feature is enabled, then the accelerometer x/y/z data registers will be 0x00 if the accelerometer is at rest. When the accelerometer is in motion, then x/y/z data registers will have non-zero values.

- Fast compensation: is a one-shot process to remember the zero position at any leveled or tilted position when stationary. For example, if a user puts his product at any position stationary and he wants to mark this initial position as 0g/0g/0g for x/y/z, then he can trigger fast compensation for x/y/z axis one by one. The offset values will be automatically saved in the public compensation registers (PCRs). Then he moves his device around. If he wants to get back to the initial position, he can just check the accelerometer outputs. If he reads 0g/0g/0g, then that position is his initial position. These offset values in the PCRs will be lost after the accelerometer is powered off.

- Manual compensation: is manually writing offset values into the PCRs through digital interface. For example, a user can read accelerometer data or perform fast compensation to obtain the offset values. Then he can save these offset values into his MCU's flash memory. In the future he can always manually write these values from his MCU memory to the PCRs for offset compensation.

- Inline compensation: is permanently saving offset values into accelerometer's NVM. The offset values can come from fast compensation or manual compensation. For example, a user can put his device flat on a table and consider this position as 0g/0g/+1g for x/y/z axis. Then he can enable fast compensation so that the zero-g offset values will be shown up in the PCRs. He can then save these values into the NVM. In the future, after power up these values in NVM will be loaded to the PCRs automatically so that all the accelerometer measurements will be based on these offsets. At the same flat position the values in accelerometer data registers will be always 0g/0g/+1g.

This document uses BMA253 12-bit accelerometer as an example to illustrate the principle of offset compensation, manual compensation and fast compensation. The process of BMA253 inline calibration and the sample code are presented.

2 Hardware and software setup

The hardware set is as shown in Figure 1. The BMA253 shuttle board is plugged onto the application mother board APP2.0. Then the APP2.0 board is connected to a PC USB port.

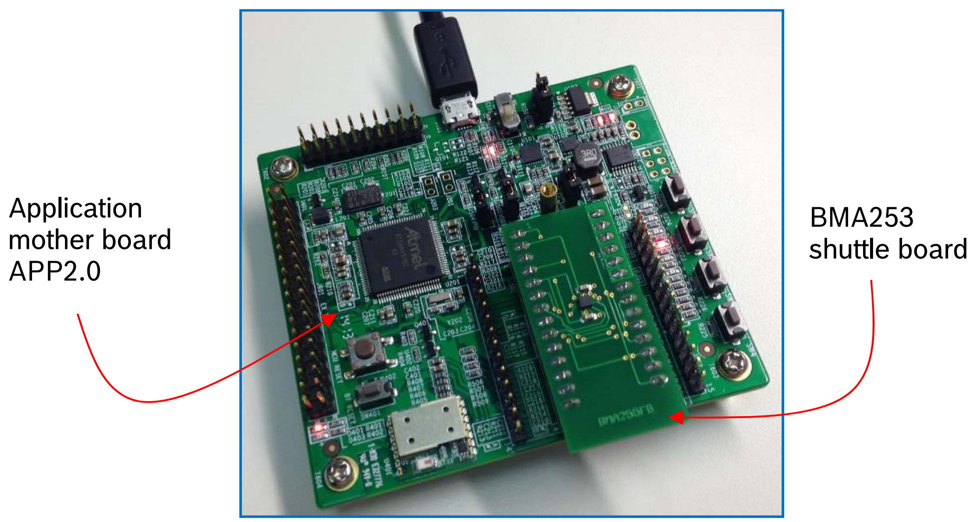

Figure 1 Hardware setup

The Windows demo software Development Desktop 2.0 (DD2.0) is running on the PC as shown in Figure 2. Users can configure sensor registers and evaluate the features of the sensor.

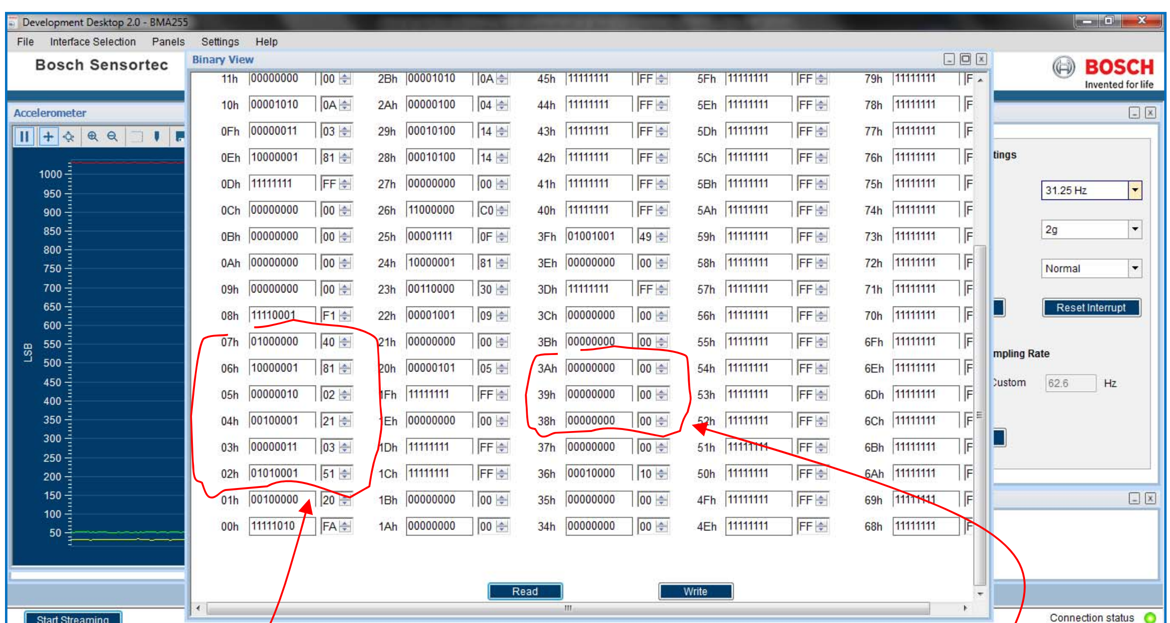

Figure 2 DD2.0 Windows demo software

3 BMA253 incline calibration

3.1 Offset compensation principle

BMA253 has three zero-g offset public compensation registers PCRs,

- 0x38 – offset_filt_x
- 0x39 – offset_filt_y
- 0x3A – offset_filt_z

They are the image registers of the three registers located inside the NVM. Every time when BMA253 is powered on, the zero-g offset values will be automatically loaded from NVM to these three image registers or PCRs. The default values for these three registers are 0x00 as shown in Figure 2.

In BMA253 datasheet, the principle of offset compensation is as shown in Figure 3.

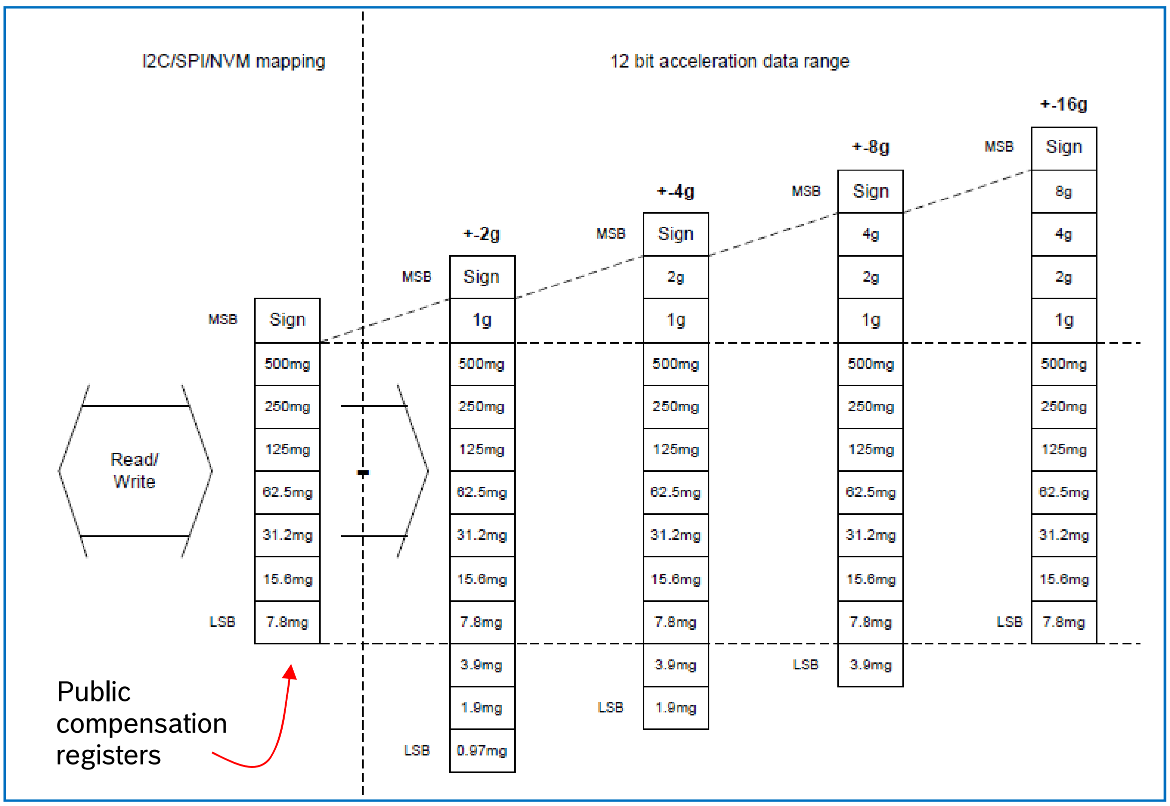

Figure 3 Principle of offset compensation

It can be seen that each LSB in PCRs is 7.8mg, while each LSB reflected to data register is 0.97mg at ±2g full scale range.

Therefore, Figure 3 contains the following two directions of conversion:

(1) If one of the PCRs 0x38, or 0x39, or 0x3A has a value of 0x06 for example, then this corresponds to 31.2mg + 15.6mg = 46.8mg zero-g offset. In order to convert it to 12-bit data range expression, we need to add three zeros to the end of 0x06 and then add one zero if positive (or one 1 if negative) to the beginning of 0x06 in red color. That is, 0x0000 0011 0000 which becomes 0x030 = 48LSBs = 48LSBs * 0.97mg/LSB = 46.6mg. This means that every raw measurement data will add 0x030 for offset compensation. As it can be seen, there is no resolution loss in this direction of conversion. And this direction of conversion is handled by BMA253 accelerometer automatically internally.

(2) If BMA253 X axis for example shows 53LSBs or 0x35 (=53LSBs * 0.97mg/LSB = 51.4mg) when X axis is flat at 0g, then we need to write a negative 8-bit value in PCR 0x38 to cancel it out. We need to delete the least three bits of 0x35 and then add three zeros to the beginning. 12-bit offset value of 0x35 becomes 0x0000 0110 ~~101~~ which is 0x06 (=31.2mg + 15.6mg = 46.8mg). Therefore, the PCR 0x38 should have value of 0xFA which is the negative value of 0x06. Eventually every raw measurement data will add 0xFA for offset compensation. From this direction of conversion, it can be seen that there is 51.4mg – 46.8mg = 4.6mg resolution loss. And this direction of conversion is handled by the BMA253 accelerometer automatically internally by fast compensation. Users can also perform this calculation for manual compensation.

3.2 Manual compensation example

The following steps show how to do BMA253 manual compensation in DD2.0 SW.

(1) Place the APP2.0 board on the table stationary. Let's assume the table is leveled so that X = Y = 0g and Z = +1g.

(2) Launch DD2.0 SW and set BMA253 to ±2g FS range, 31.25Hz bandwidth (BW) or 62.5Hz output data rate (ODR) and normal mode. +1g is 1024LSB on Z axis.

(3) Read BMA253 data registers in Figure 2,
Ax = ([register 0x03 | register 0x02]) >> 4 = 0x035 = 53LSBs
Ay = ([register 0x05 | register 0x04]) >> 4 = 0x022 = 34LSBs
Az = ([register 0x07 | register 0x06]) >> 4 = 0x408 = 1032LSBs

Therefore,
Ax_offset = -53LSB
Ay_offset = -22LSB
Ax_offset = -8LSB

(4) Convert these 12-bit zero-g offsets to 8-bit offsets
Regisrter 0x38 offset_filt_x = -0x06 = 0xFA
Regisrter 0x39 offset_filt_y = -0x04 = 0xFC
Regisrter 0x3A offset_filt_z = -0x01 = 0xFF

3.3 Fast compensation example

The following steps show how to do BMA253 fast compensation in DD2.0 SW.

(1) In DD2.0 SW, go to "Offset View" as shown in Figure 4.

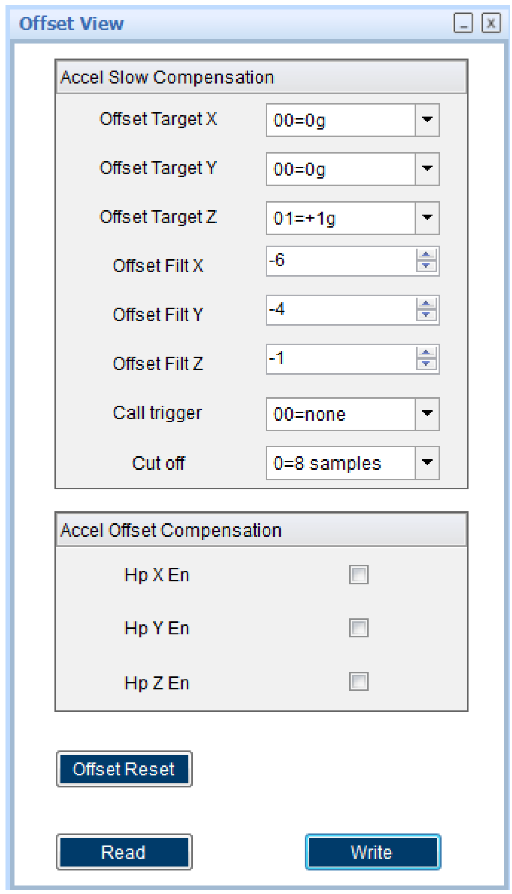

Figure 4 Offset view of DD2.0 SW

(2) First select Offset Target X/Y/Z as 0g/0g/+1g. Then from "Call trigger" dropdown list select "01 = x axis". After clicking "Write" button, the value of -6 (or 0xFA) is shown in "Offset Filt X" edit box.

(3) Repeat for Y and Z axis. The values of -4 (or 0xFC) and -1(or 0xFF) are shown in "Offset Filt Y" and "Offset Filt Z" edit box respectively.

(4) Read the BMA253 data registers again. The new offset values in PCRs are as shown in Figure 5. They are the same as the values from manual compensation above.

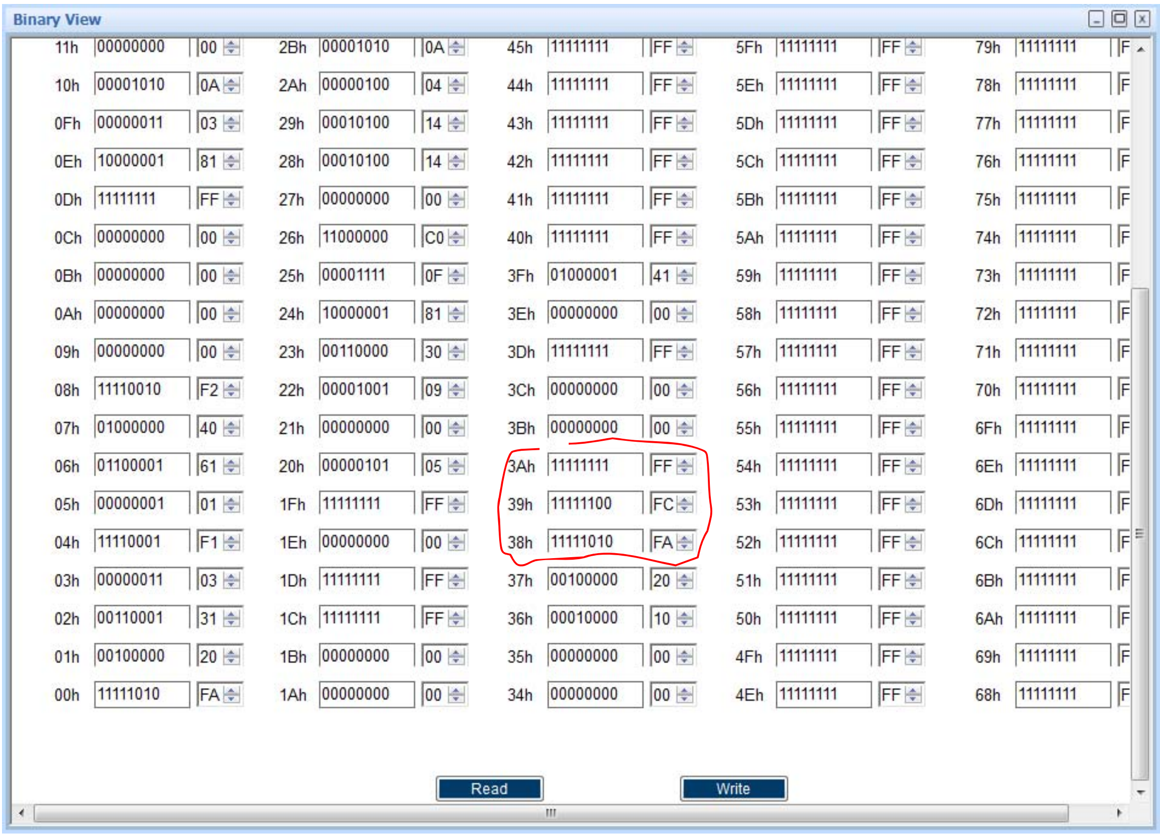

Figure 5 New values in offset image registers

3.4 Sample code for fast compensation

The following is the pseudo code for BMA253 fast compensation. The product is stationary at a certain position, for example, BMA253 X = Y = 0g and Z = +1g.

```
void BMA253_fast_compensation(void)
{
        Write value of 0x20 to register 0x37;          // set target X=Y=0g and Z=+1g
        Write value of 0x20 to register 0x36;          // trigger X axis fast compensation

Loop1:  VAL = Read register 0x36;                      // read register 0x36 to variable VAL
        If ((VAL & 0x10) ==0), then go to Loop1;       // fast compensation on X axis is ongoing if
                                                       bit-4 cal_rdy = 0
        Else, Write value of 0x40 to register 0x36;    // trigger Y axis fast compensation

Loop2:  VAL = Read register 0x36;                      // read register 0x36 to variable VAL
        If ((VAL & 0x10) ==0), then go to Loop2;       // fast compensation on Y axis is ongoing if
                                                       bit-4 cal_rdy = 0
```

```
        Else, Write value of 0x60 to register 0x36;    // trigger Z axis fast compensation

Loop3:  VAL = Read register 0x36;                      // read register 0x36 to variable VAL
        If ((VAL & 0x10) ==0), then go to Loop3;       // fast compensation on Z axis is ongoing if
                                                       bit-4 cal_rdy = 0
        Else, Write value of 0x00 to register 0x36;    // fast compensation is done.
}
```

The zero-g offset values have been automatically written to PCRs 0x38 for X, 0x39 for Y and 0x3A for Z axis respectively after each trigger is enabled.

3.5 Sample code for writing offsets to NVM

After the PCRs have new offset values available, the last step of BMA253 inline calibration is to write these offset values from PCRs to NVM.

The following pseudo code shows how to write the offset values in PCRs to NVM.

```
void BMA253_writing_NVM(void)
{
        Val = Read register 0x33;                      // read register 0x33 to variable VAL
        Val = Val | 0x01;                              // set bit-0 in register 0x33
        Write Val back to register 0x33;               // set bit-0 nvm_prog_mode to unlock NVM
        Val = Read register 0x33;                      // read register 0x33 to variable VAL
        Val = Val | 0x02;                              // set bit-1 in register 0x33
        Write Val back to register 0x33;               // set bit-1 nvm_prog_trig to start writing
                                                       values to NVM

Loop:   Val = Read register 0x33;                      // read register 0x33 to variable VAL
        If ((VAL & 0x04) == 0), then go to Loop;       // NVM writing is ongoing if bit-2 nvm_rdy = 0
        Else, Write value of 0x00 to register 0x33;    // NVM writing is done. Lock NVM
}
```

Once the offset values in PCRs have been written to NVM, they will always be loaded from NVM to PCRs at each power on reset or soft reset.

The default value of register 0x33 is 0xF4 meaning that there are 15 times NVM writing available. After the above NVM writing function, the value in register 0x33 will become 0xE4 which means there are 14 times NVM writing left. If the NVM has been written for more than 15 times, then the new offset values will no longer be able to be written to NVM.

If a user wants to clear the offsets in the NVM, then he can simply write 0x00 to PCRs and call function BMA253_writing_NVM() again.

4 Legal disclaimer

4.1 Engineering samples

Engineering Samples are marked with an asterisk (*) or (e) or (E). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

4.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or security sensitive systems. Security sensitive systems are those for which a malfunction is expected to lead to bodily harm or significant property damage. In addition, they are not fit for use in products which interact with motor vehicle systems.

The resale and/or use of products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the Purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser must monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of all security relevant incidents.

4.3 Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

5 Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
1.0		Document creation	March 17th, 2017

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 8
72770 Reutlingen / Germany

Contact@bosch-sensortec.com
www.bosch-sensortec.com

Modifications reserved | Printed in Germany
Specifications subject to change without notice
Document number: BST-BMP280-SD001-03
Revision_1.3_April_2013