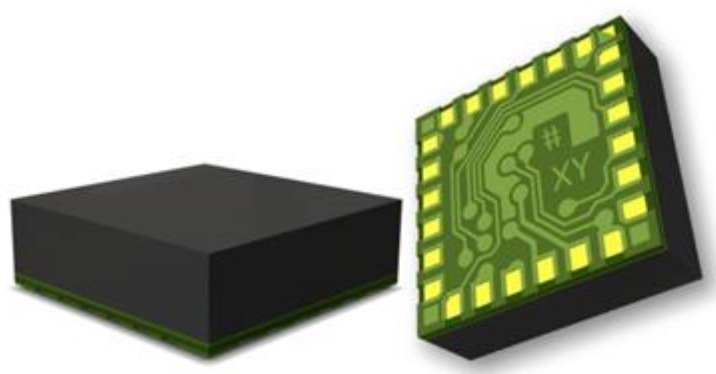
Bosch Sensortec

Application Note
Axes remapping of BHA250(B) / BHI160(B)



BHy1 - Axes remapping

Document revision	1.4	
Release date	20 February 2017	
Document number	BST-MHS-AN001-05	
Technical reference code(s)	0 273 141 230 0 273 141 231	0 273 141 309 0 273 141 310
Notes	Content in this document is subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance.	

Index of Contents

1. Introduction

Accelerometer (Acc), Gyroscope (Gyro) and Magnetometer (Mag) sensors (components) have their own coordinates. By default BHA and BHI are configured to ENU axis convention (East-North-Up), as commonly used in consumer electronic devices. It is usually obtained by the integration of Accelerometer (Acc), Gyroscope (Gyro) and Magnetometer (Mag) sensors readings.

The ENU coordinate system is defined as a direct orthonormal basis where:

x points east and is tangential to the ground.
y points north and is tangential to the ground.
z points towards the sky and is perpendicular to the ground.

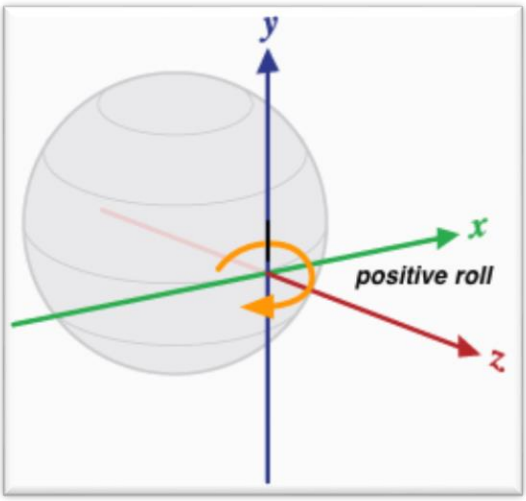


Figure 1 Coordinate system relative to ENU convention[1]

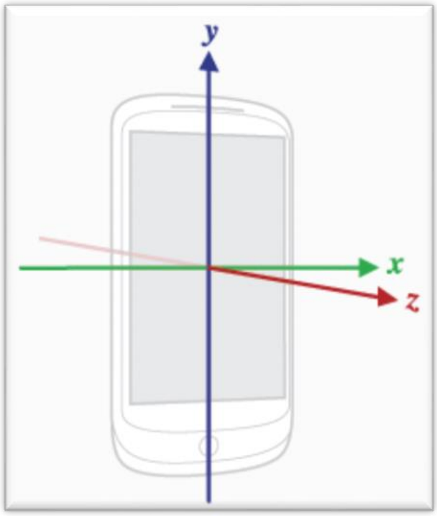


Figure 2 Coordinate system relative to a mobile device[1]

As a functional system requires the sensor output data aligned to the coordinate system of the board or the application, the orientation of the BHA / BHI can be rotated to align the ENU coordinates or to other specific device's coordinates. Applying the rotation to the ENU world frame (X, Y, Z) would align them with the phone coordinates (x, y, z), or to the system coordinates of other customer devices.

In chapter 2 and 3 this Application Note introduces the remapping concept of BHA / BHI. In chapter 4 there is a summary table for you to quickly find your condition. In chapter 5 three methods to convert the coordinates in the firmware of BHA and BHI are introduced. The first two methods “updating via board configure file” and “updating via elf file” (refer to chapter 5.1 and 5.2) are dedicated to developers having the full SW development tool-chain incl. SDK installed. The third method “updating via product API” (refer to chapter 5.3) can be used by any customer.

[1] Source: http://source.android.com/devices/sensors/sensor-types.html

2. Axes definition

1. Define the coordination of your board (X_{BOARD}, Y_{BOARD}, Z_{BOARD}). Normally this is the default coordination system of your final application (i.e. your system coordination).
2. In standard Android system the ENU (east north up) orientation is required. For other applications, the $\mathbf{Z_{BOARD}}$ is normally pointing to the sky or ground when the board is placed on a horizontal surface.
3. Find the coordinates of the sensors mounted on the board in their datasheets.
3. Draw all the coordinates on the paper. Figure 3 shows an example.

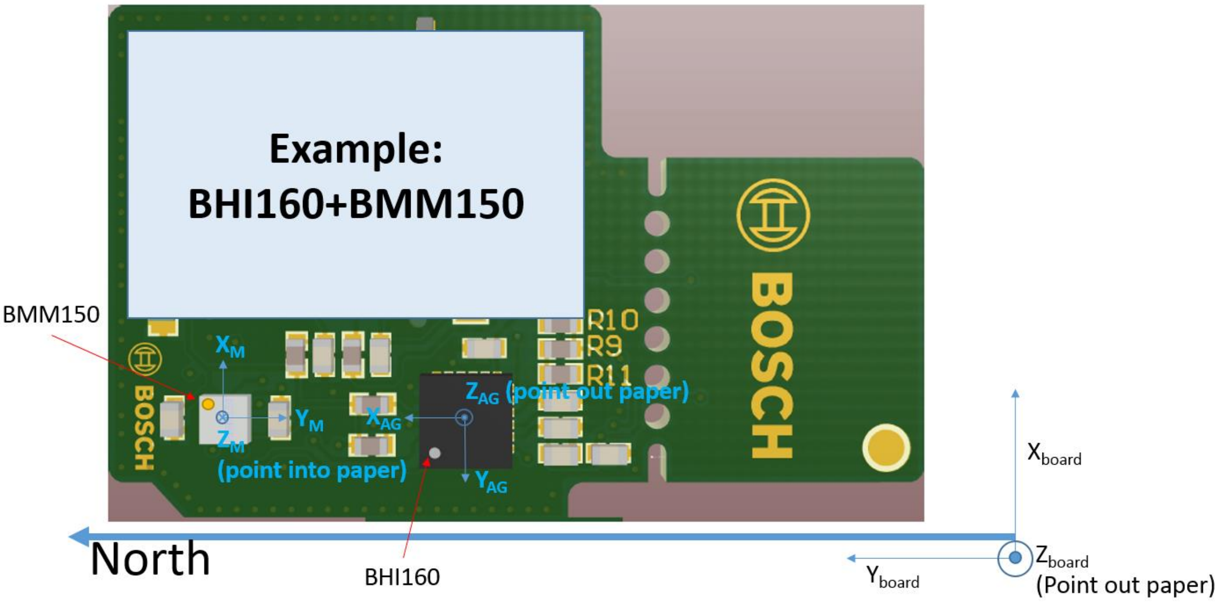


Figure 3 an example board containing a BHI160 and a BMM150

In Figure 3, a BHI160 and a BMM150 were mounted on the board independently of their PIN1 marker. Their axes orientation is marked as ($\mathbf{X_M}$, $\mathbf{Y_M}$, $\mathbf{Z_M}$) and ($\mathbf{X_{AG}}$, $\mathbf{Y_{AG}}$, $\mathbf{Z_{AG}}$) based on their coordination in their datasheets (Figure 4 and Figure 5).The board coordinate is marked at right bottom corner. $\mathbf{Y_{BOARD}}$ is heading to the planet's North Pole.

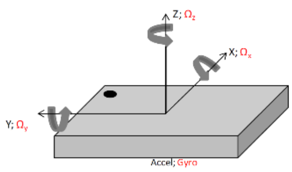


Figure 4 BHI160 axes orientation in datasheet

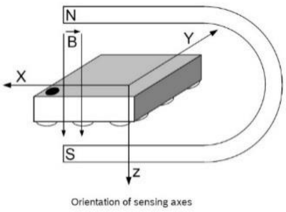


Figure 5 BMM155 axes orientation in datasheet

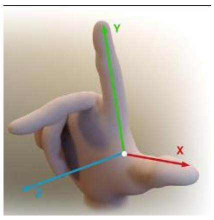


Figure 6 right handed coordinates

If you are using a standard Bosch sensor, their coordinates follow the right-handed coordinate principle (Figure 6). You can apply it to find the sensor axes when it placed on the board.

3. Orientation matrix

When the PIN1 marker of the sensors are not aligned and/or their axes orientation are different to the board, we need to convert it to the board coordinate by following formula:

$$|X\ Y\ Z| = |X_s\ Y_s Z_s| \cdot \begin{vmatrix} C_0 C_1 C_2 \\ C_3 C_4 C_5 \\ C_6 C_7 C_8 \end{vmatrix}$$

Where

$[X \quad Y \quad Z]$ is the board coordinate
$[X_s \quad Y_s \quad Z_s]$ is the sensor coordinate
(C_0... C_8) is the orientation matrix.

For a board having Acc, Gyro and Mag, their relationship can be described by:

$$(X_{BOARD}, Y_{BOARD}, Z_{BOARD}) = (\mathbf{X_A}, \mathbf{Y_A}, \mathbf{Z_A})\ C_A = (\mathbf{X_G}, \mathbf{Y_G}, \mathbf{Z_G})\ C_G = (\mathbf{X_M}, \mathbf{Y_M}, \mathbf{Z_M})\ C_M$$

Jump to chapter 4 to find the condition of the orientation matrix corresponding to your board. Or use table 1 to write the orientation matrix:

Table 1 orientation matrix table

	X_A	Y_A	Z_A	X_G	Y_G	Z_G	X_M	Y_M	Z_M
X_{BOARD}	C_{A0}	C_{A3}	C_{A6}	C_{G0}	C_{G3}	C_{G6}	C_{M0}	C_{M3}	C_{M6}
Y_{BOARD}	C_{A1}	C_{A4}	C_{A7}	C_{G1}	C_{G4}	C_{G7}	C_{M1}	C_{M4}	C_{M7}
Z_{BOARD}	C_{A2}	C_{A5}	C_{A8}	C_{G2}	C_{G5}	C_{G8}	C_{M2}	C_{M5}	C_{M8}

In table 1, C_A = (C_{A0}... C_{A8}), C_G= (C_{G0}, ..., C_{G8}) and C_M = (C_{M0}, ..., C_{M8}) are the orientation matrix, which need to be updated in the firmware. C_{xy} are the coefficients between sensor axes orientation and board axes orientation (x = A, G, M; y = 0, 1,..., 8). The coefficient has three possible value: 1, 0 and -1.

1 --- Two axes are paralleled and have same direction
-1 --- Two axes are paralleled and have opposite direction
0 --- Two axes are perpendicular

Example 1

In figure 1, BMM150 axes are different with the board axes and BHI160 has same coordinate with the board. So the orientation matrix table is shown in table 2.

Table 2 orientation matrix table

	X_A	Y_A	Z_A	X_G	Y_G	Z_G	X_M	Y_M	Z_M
X_{BOARD}	0	-1	0	0	-1	0	1	0	0
Y_{BOARD}	1	0	0	1	0	0	0	-1	0
Z_{BOARD}	0	0	1	0	0	1	0	0	-1

Orientation matrix:
C_A = (0 1 0 -1 0 0 0 0 1), C_G= (0 1 0 -1 0 0 0 0 1) and C_M = (1 0 0 0 -1 0 0 0 -1)

Example 2

If we have a board shown in figure 5. On this board, there are a BHI160 and a BMM150. The dots on the sensor indicate their coordinates. You can try to find the coordinates based on figure 2 and 3, and write the orientation matrix. Table 3 shows the result.

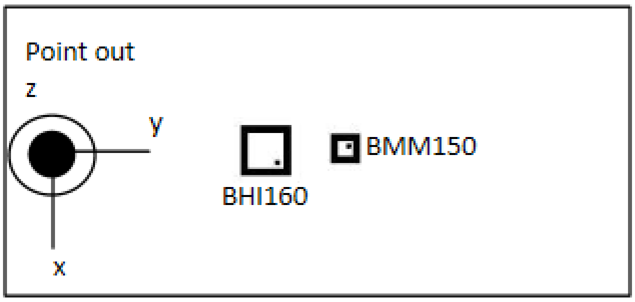


Figure 7 example board with BHI160 and BMM150

Table 3 orientation matrix table

	X_A	Y_A	Z_A	X_G	Y_G	Z_G	X_M	Y_M	Z_M
X_{BOARD}	1	0	0	1	0	0	0	1	0
Y_{BOARD}	0	1	0	0	1	0	1	0	0
Z_{BOARD}	0	0	1	0	0	1	0	0	-1

Orientation matrix:
C_A = (1 0 0 0 1 0 0 0 1), C_G = (1 0 0 0 1 0 0 0 1), and C_M = (0 1 0 1 0 0 0 0 -1).

4. Position summary

Figure 8 shows the possible positions of a sensor placed on a board. The positions are named as P0… to P7. The corresponding orientation matrices of BHA, BHI, BMG250 and BMM150 to the board are following:

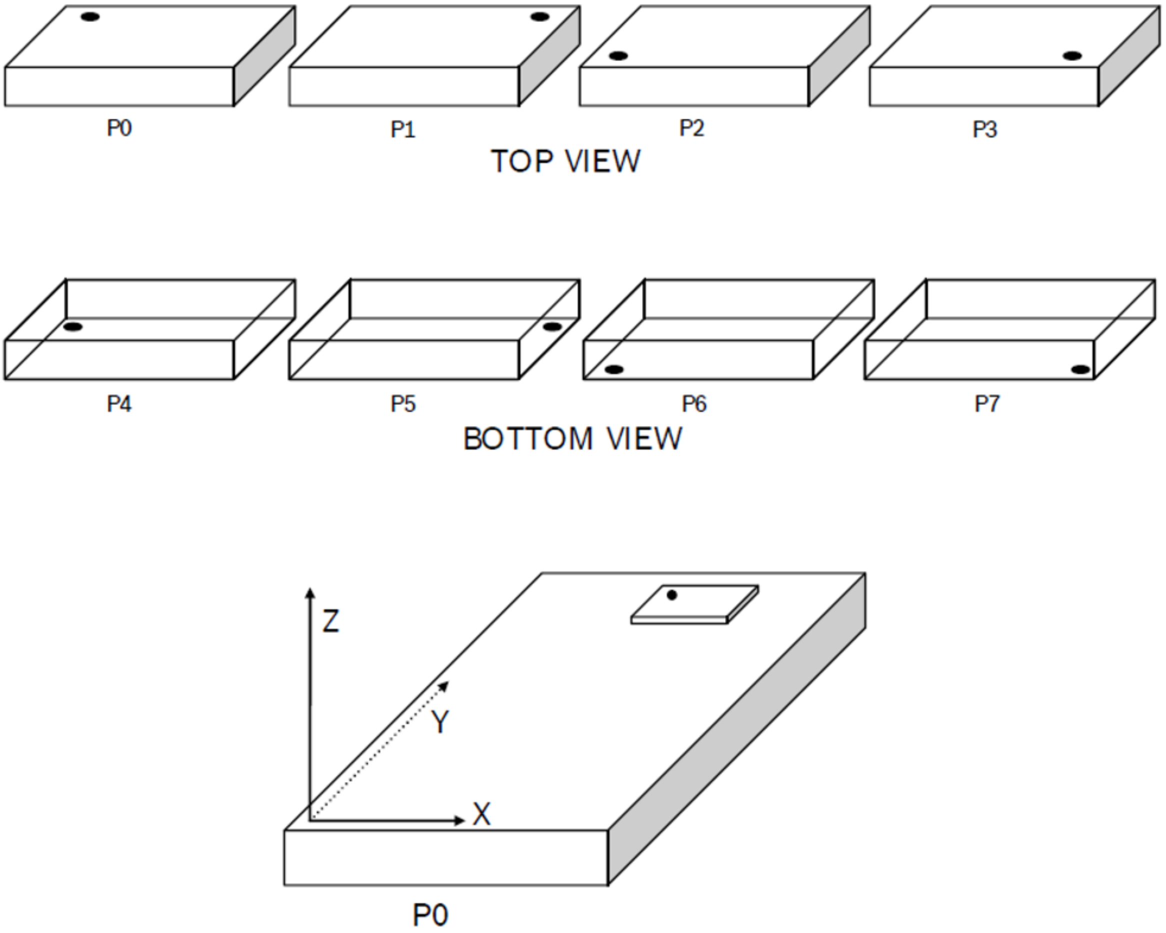


Figure 8 Possible position of BHA, BHI, BMG250 and BMM150

4.1 BHA, BHI and BMG250

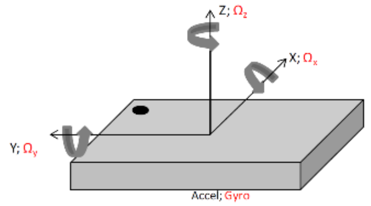


Figure 9 BHA, BHI and BMG250 axes orientation

For BHA, the orientation matrices respect to different positions are:
P0: (0 1 0 -1 0 0 0 0 1); P1: (1 0 0 0 1 0 0 0 1); P2: (-1 0 0 0 -1 0 0 0 1); P3: (0 -1 0 1 0 0 0 0 1); P4: (-1 0 0 0 1 0 0 0 -1); P5: (0 1 0 1 0 0 0 0 -1); P6: (0 -1 0 -1 0 0 0 0 -1); P7: (1 0 0 0 -1 0 0 0 -1).

For BMG250, the orientation matrix is the same as BHA.
For BHI containing an Accel and a Gyro, the orientation matrices of both are the same as BHA.

4.2 BMM150

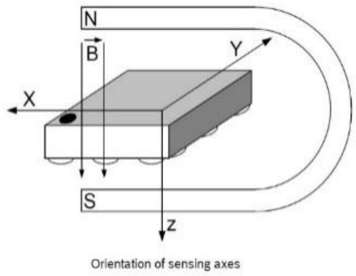


Figure 10 BMM150 axes orientation in datasheet

P0: (0 1 0 1 0 0 0 0 -1); P1: (1 0 0 0 -1 0 0 0 -1); P2: (-1 0 0 0 1 0 0 0 -1); P3: (0 -1 0 -1 0 0 0 0 -1); P4: (-1 0 0 0 -1 0 0 0 1); P5: (0 1 0 -1 0 0 0 0 1); P6: (0 -1 0 1 0 0 0 0 1); P7: (1 0 0 0 1 0 0 0 1).

5. Update orientation matrix

Orientation Matrix is also called as "Axes remapping data" and is stored in "Physical Sensor Information" inside the firmware image.

There are three methods to update the matrix in the firmware. For customers having an SDK including tool- chain and **Board.cfg** file or **elf file**, please refer to section 5.1 or 5.2 to update the matrix in the firmware. In the tool-chain, **makefile**, **stuffelf** and **elf2bin** will be used in this guide.

For any other customer the matrix in the firmware can be updated within the **product API**. Please refer to section 5.3.

5.1 updating via board.cfg file

1. Edit the orientation matrix in **board.cfg** file at **Cal0,Cal1,Cal2,Cal3,Cal4,Cal5,Cal6,Cal7,Cal8**. Following is the section code in board.cfg:

```
#Physical Drivers
#DriverID,Addr,GPIO,Cal0,Cal1,Cal2,Cal3,Cal4,Cal5,Cal6,Cal7,Cal8,Off0,Off1,Off2,Range
m11, 16, 3, 0, 1, 0, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0
45, 105,-, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
46, 105,-, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
47, 105,-, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
a48, 105, 0, 0, 1, 0,-1, 0, 0, 0, 0, 1, 0, 0, 0, 0
g49, 105,-, 0, 1, 0,-1, 0, 0, 0, 0, 1, 0, 0, 0, 0
```

Where

DriverID: Driver ID of the physical sensor drivers as defined in their respective driver.config files. Command **make print_drivers** can print all ID, i.e. with **egrep** it can print the selected id name:

```
user@computer:~/SDK/boards$ make print_drivers | egrep "11 |45 |46 |47 |48 |49"
11 BMX055MagPolling
45 BMI160SigMotion
46 BMI160StepDetector
47 BMI160StepCounter
48 BMI160Accel
49 BMI160Gyro
```

Addr: The I2C address of the sensors in decimal.
GPIO: The GPIO associated to the interrupt line on this specific PCB.
Calx: Calibration matrix used for remapping, which is (C0, C1, C2, C3, C4, C5, C6, C7, C8).
Offx: Are used to set offset at the configuration file level and are unused by Bosch. These parameters are rather set at runtime.
Range: Are used to set range at the configuration file level and are unused by Bosch. These parameters are rather set at runtime.

2. Then use **Makefile** to generate the .fw file, i.e.:

```
bst-esa3@bstesa3-VirtualBox:~/work/fuser/boards$ make Bosch_PCB_7183_di01_BMI160.
cfg
```

5.2 updating via elf file

If you consider to use **elf file**, you need two tools: **stuffelf** and **elf2bin**.

1. Update the orientation matrix in the .elf file. For example, you have a .elf file named your_elf_file.elf, and want to update ACC orientation matrix:

```
stuffelf outerloop.elf –a -d24 -p3 –cC0,C1,C2,C3,C4,C5,C6,C7,C8
```

Stuffelf may have different suffix in its name depending on SDK version. Please notice there is no space between –c and orientation matrix.

2. Then use your_elf_file.elf to generate the .fw by **elf2bin**:

```
./elf2bin your_elf_file.elf your_fw_file.fw
```

Command of **stuffelf** details please refer to the **Appendix I** of this document. For more details and technical support please refer to Bosch Sensortec document number BST-FUSER1-SD000-01 "FUSER Core Programmers Guide" *section 6.4*. or contact our regional offices, distributors and sales representatives.

5.3 updating via product API

Alternatively to the previous chapters – i.e. for customers not having the full SW development tool-chain incl. SDK installed – the matrix can also be edited within the product API which is available on GitHub:

https://github.com/BoschSensortec/BHy1_driver_and_MCU_solution

You can directly edit the remapping matrix in its .c file. The following code is content of the accelerometer_remapping_example.c in the API (version uc_bhy_driver_20160823). Use this as a reference to update your matrix accordingly.

```
int main(void)
{
  u8 array[ARRAYSIZE], *fifoptr, bytes_left_in_fifo=0;
  u16 bytes_remaining, bytes_read;
  bhy_data_generic_t fifo_packet;
  bhy_data_type_t   packet_type;
  BHY_RETURN_FUNCTION_TYPE result;
   s8 mapping[9] = {0};
   s8 mapping2[9] = {0,1,0,-1,0,0,0,0,1}; // new mapping matrix

  … …

   /* config mapping matrix, it is not necessary to change mapping matrix if its orientation is aligned with the board */
   bhy_get_mapping_matrix(PHYSICAL_SENSOR_INDEX_ACC,mapping); // get current mapping matrix
   bhy_set_mapping_matrix (PHYSICAL_SENSOR_INDEX_ACC,mapping2); // set new mapping matrix in the fw
   bhy_get_mapping_matrix(PHYSICAL_SENSOR_INDEX_ACC,mapping); // check if the matrix is set successfully

  … …
```

After editing the matrix please don't forget to generate a new *.h file which has to be ported subsequently. For more details and technical support with respect to the product API the *.h file and the MCU porting please refer to the Bosch Sensortec documents *"MCU Driver Porting Guide"* and *"Interfacing Reference Code from Generic Driver"* for BHA and/or BHI placed within the section "Application notes", available on

https://www.bosch-sensortec.com/bst/support_tools/downloads/overview_downloads

6. Check the orientation

After the orientation matrix has been updated, you need to check whether the remapping is successful. You can read the uncalibrated data of the sensors to check if the remapping is successful. Please define the coordination of the board (X_{BOARD}, Y_{BOARD}, Z_{BOARD}) first.

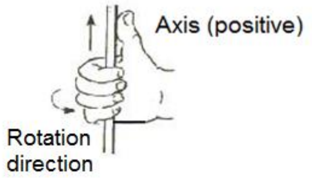


Figure 11 right-hand crew rule

6.1. Check the Acc axes (XA, YA, ZA)

1) Enable Acc uncalibrated data;
2) Place the board on a horizontal surface and make sure Z_{BOARD} is pointing to the sky. Z-axis should output 1g.
3) Turn the board at perpendicular position and make sure X_{BOARD} is pointing to the sky. X-axis should output 1g.
4) Turn the board at perpendicular position and make sure Y_{BOARD} is pointing to the sky. Y-axis should output 1g.

6.2. Check the Gyro axes (XG, YG, ZG)

If you use BHI160, the axes of Acc and Gyro are same.
If you use BMG250, please do following steps to check the Gyro (**X_G, Y_G, Z_G**).
In Bosch product, a Gyro axis direction can be found by your right hand. Move your four fingers follow the rotation direction to make a fist and the thumb will be pointing to the positive axis like figure 9 (This is similar to right-hand crew rule).

1) Enable Gyro uncalibrated data;
2) Place the board on a horizontal surface and make sure Z_{BOARD} is pointing to the sky. Quickly rotate left 180 degree. The output has significant change is the rotation axis. It should be z-axis and output positive data.
3) Rotate the board according to X_{BOARD} and Y_{BOARD}. Check the output with right-hand crew rule respectively.

6.3. Check the Mag axes (XM, YM, ZM)

6.3.1 Method 1

1) Enable Mag uncalibrated data and only log the first three data, which is the original data;
2) Place the board on the table and make sure the X_{BOARD} is pointing to the planet's North Pole by using a reference compass. Record the (**X_M, Y_M, Z_M**) stable output below.
3) Then Point X_{BOARD} to planet's South Pole and record the stable output.

Table 4 method 1 to check mag remapping

X_{BOARD}	X_M	Y_M	Z_M
Pointing to North Pole	X_1	Y_1	Z_1
Pointing to South Pole	X_2	Y_2	Z_2
Difference	$\|X_1 - X_2\|$	$\|Y_1 - Y_2\|$	$\|Z_1 - Z_2\|$

4) The axis output has the maximum difference is parallel to X_{BOARD} axis. Here $|X_1 - X_2|$ should be the largest, and $X_1 > X_2$. Then the remapping is correct.
5) Apply same method to Y_{BOARD}, Z_{BOARD} to find if the orientation of other two axes are correct.

6.3.2 Method 2

1) Enable Mag uncalibrated data and only log the first three data;
2) Rotate the board in following steps and record the output data:

Table 5 method 2 to check mag remapping

Steps	XBOARD	YBOARD
YBOARD point to North	X1	Y1
Rotate 90 right (YBOARD point to East)	X2	Y2
Rotate 90 right (YBOARD point to South)	X3	Y3
Rotate 90 right (YBOARD point to West)	X4	Y4
Point ZBOARD to South	Z1	
Point ZBOARD to North	Z2	

If

1. X1>X2 & X4>X3, min(X1,X2,X3,X4)=X2, and max(X1,X2,X3,X4)=X4;
2. Y1>Y2 & Y4>Y3, min(Y1,Y2,Y3,Y4)=Y3, and max(Y1,Y2,Y3,Y4)=Y1;
3. Z2>Z1.

Then the remapping is correct.

7. Appendix I

Table 6 Stuff-Elf Utility v1.0 command help

-q	Do not print status information on stdout
-f<config file>	Use the specified configuration file for sensor information. The configuration file will be created if nonexistent, or updated with any parameters specified on the command line.
--noexec	Set the EEPROMNoExec flag to cause the CPU to halt after load.
--exec	Clear the EEPROMNoExec flag to cause the CPU to begin executing after load.
-i<kHz>	Specify the max I2C clock speed supported by the EEPROM device (default: 83 kHz).
--irq<pin>	Specify the host interrupt pin.
-pull<pin> <up|down|default|none>	Specify the pull configuration for the gpio pin.
--fifo<%wakeup>	Specify the percentage of the FIFO in the wakeup fifo (Android L+).
-m	Select the mag sensor.
-a	Select the accel sensor.
-g	Select the gyro sensor.
-<id 0...9>	Select the <id>th sensor.
-d<addr>	Specify the I2C address for the selected sensor.

-p<pin>	Specify the GPIO pin used for the selected sensor.
-c<v1,v2,v3,v4,v5,v6,v76,v8,v9>	Specify the cal matric used for the selected sensor.
-o<v1,v2,v3>	Specify the cal offset used for the selected sensor.
--range<default range>	Specify the default dynamic range of a physical sensor (Android L+).
-r<range> *** Legacy firmware only, use --range instead	Specify the dynamic range used for the selected sensor.
--noise<min noise>	Specify the minimum noise allowed for the sensor.
--noise_mode<0,1>	Specify the noise measurement mode. 0: The noise is averaged. 1: The noise is directly measured.
--version<m>	Specify the custom version number of the firmware image.

8. Appendix II

8.1. Figures

8.2. Tables

9. Legal disclaimer

9.1. Engineering samples

Engineering Samples are marked with an asterisk (*) 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.

9.2. Product usage

The Purchaser shall indemnify 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.

9.3. 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.

10. Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
0.1	all	Document creation	Jul. 2016
1.0	all	Fit to new format	Sep. 2016
1.1	5; Appendix II;	Add C file / method of remapping via API Add figure and table summary	Sep. 2016
1.2	5.1, 5.2, 5.3	Refined, placed download links	Nov. 2016
1.3	all 1	Added 2 new tech. ref. codes: - 0.273.141.309 (BHI160B) - 0.273.141.310 (BHA250B) Refined to distinguish better between coordination system of sensor, board and final application and clarified the board coordination is normally the customer system coordination.	Jan. 2017
1.4	5.1,5.2	Specify the steps in 5.1; add elf2bin in 5.2	Feb. 2017

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 9
72770 Reutlingen / Germany

contact@bosch-sensortec.com
www.bosch-sensortec.com