Software Documentation

Integration Guideline for BSXlite Library

Bosch Sensortec

Integration guideline	**- Integration Guideline for BSXlite Library**
Document revision	1.1
Document release date	27.02.2015
Document number	BST-BSX03X-SD002-00
Technical reference code(s)	
Notes	Data in this document are 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. About the Integration Guideline

This manual gives an overview of the basic pre-requisites for the BSXlite Fusion Library integration & procedure to integrate the fusion library on various platforms.

The Fusion library offers flexibility for choosing the desired combination of accelerometer, magnetometer and gyroscope. At the same time, a set of recommended parameters for each sensor is offered to assure optimized performance over a maximum range of conditions.

This document can be used for understanding the structure and the flow of the library with sets of parameters/structures defined in each stage of architecture of fusion library and it will give an estimation of data verification from sensor data in real time.

1.1 Who should read this?

This information is intended for users who want to perform integration of BSXlite fusion library on various supported platforms.

2. Overview of sensors

This section will provide information about the sensors which are used in library processing and also provides basic functionalities of above mentioned sensors.

2.1 Accelerometer

An accelerometer is an electromechanical device for measuring the change of linear motion (=dynamic acceleration) as well as gravitational forces (=static acceleration). The direction of the gravitational force (1 g = $9.8m/s^2$) with respect to the device is used to calculate the orientation of the device. Accelerometers measure the acceleration in one, two, or three orthogonal axes. To use an accelerometer in a portable device, it is necessary to measure all three axes. For advanced sensor fusion, the performance of the accelerometer is crucial, since its output contributes to the static accuracy of the 9DoF system.

2.2 Magnetometer

A magnetometer is a device designed to measure the direction and strength of the earth magnetic field. 3 axes are necessary. The Earth's magnetic field (the magnetosphere) varies from place to place for various reasons such as in homogeneity of rocks and the interaction between charged particles from the sun and the magnetosphere. In fusion library geomagnetic sensors are used to drift-compensate gyroscopes.

2.3 Gyroscope

In principle, a gyroscope is a device for measuring changes of orientation, based on the principles of angular momentum. MEMS gyroscopes are used for measuring the rate of rotation in space (roll, pitch and yaw) also referred to as “angular rate”. The physical unit of the measured data is “degrees per second”. Degrees per second are also referred to as “dps” or “°/s”. A Gyroscope is immune versus linear accelerations and distortion affecting accelerometer and magnetometer. So, advanced fusion techniques can be used for magnetic distortion filtering and accurate device orientation while device experiencing dynamics can be estimated with assistance of the gyroscope. Gyroscope from Bosch Sensortec provides high dynamics needed for real-time gaming.

3. Overview of BSX Fusion Library

3.1 General description

BSXlite Fusion Library is a complete 9-axis fusion solution which combines the measurements from 3-axis gyroscope, 3-axis geomagnetic sensor and a 3-axis accelerometer, to provide a robust absolute orientation vector.

Fusion Library provides orientation information in form of quaternion or Euler angles. The algorithm fuses the sensor raw data from 3-axis accelerometer, 3-axis geomagnetic sensor and 3-axis gyroscope in an intelligent way to improve each sensors output. This includes algorithms for offset calibration of geomagnetic sensor and gyroscope, monitoring of the calibration status and Kalman filter fusion to provide distortion-free and refined orientation vectors.

Since BSX 9-axis fusion software is developed together with the sensor hardware, optimized performance in terms of dynamics is achieved. Direct access to the Bosch sensor hardware allows the user to set use-case specific operation modes regarding data rates and noise thresholds. The solution provides a ready-to-use advanced 9-axis sensor fusion system which reduces the complexity for customers and helps in rapid development of advanced sensor applications.

3.2 Key features of BSXlite vs. BSX

	BSXlite	BSX (full library)
Release format	closed source code / compiled library	closed source code / compiled library
License	Click-trough on partner website	Contact Bosch Sensortec
Key Features		
Axis remapping	✓	✓
Offset correction	✓	✓
Soft Iron Correction	× (can be implemented outside library)	✓
Accelerometer calibration	✓	✓
Magnetometer calibration	Classical: based on figure-of-eight motion	classical and advanced (fast calibration)
Magnetic distortion check	Basic	Advanced
Gyroscope calibration	✓	✓
9-axis orientation processing	Basic	Advanced
Compass orientation processing	Basic (tilt compensation)	Advanced (adaptive filtering, tilt compensation)
Data fusion modes	9-axis	9-axis & 6-axis (IMU, M4G, eCompass)
Full Scale		
Gyroscope	500 dps	2000 dps
Outputs		

Acceleration	raw, corrected, offsets, accuracy status	raw,corrected, filtered, offsets, accuracy status
Magnetometer	raw,corrected, offsets, accuracy status	raw,corrected, filtered, offsets, accuracy status
Gyroscope	raw,corrected, offsets, accuracy status	raw,corrected, filtered, offsets, accuracy status
Virtual gyroscope (M4G)	×	✓
Quaternions	✓	✓
Orientation	✓ (unfiltered)	✓
Rotation matrix	×	✓
Heading accuracy	✓	✓
Linear acceleration	×	✓
Gravity	×	✓
Gestures	×	✓
Step counter and step detector	× (in BMI160 Hardware)	✓
Significant motion	× (in BMI160 Hardware)	✓
Output data rates (ODR)		
Accelerometer	100Hz	200Hz,100Hz,50Hz,25Hz,12.5Hz
Magnetometer	25Hz	50Hz,25Hz,12.5Hz
Gyroscope	100Hz	200Hz,100Hz
Orientation sensor	50 Hz	100Hz, 200Hz

3.3 Applications

The Fusion Library provides dynamic and accurate orientation to the target platform which is typically used to provide enhanced user experience for e.g. gaming, navigation systems, dead reckoning and augmented reality.

3.4 Advantages

- Hardware and software co-design for optimal performance
- Complete software fusion solution out of one hand
- Eliminates need for own fusion software development
- Robust virtual sensor outputs optimized for the application

4. Pre-requisites for library integration

This section will give an overview about the prerequisites for fusion library what has to be verified before integration of fusion library.

The critical prerequisites of the library which can affect the library performance are listed below.

- Input data (Raw Data from different hardware sensors and corresponding time stamps).
- Raw data has to be remapped which means the accelerometer, magnetometer and the gyroscope orientation should be in line with the device orientation definition.
- The platform should support all standard C/C++ libraries for processing.
- The core & controllers should have sufficient resources (ROM/RAM) for library processing.

4.1 Input data to the library

The BSXlite fusion library requires three hardware sensor data (i.e. Accelerometer, Magnetometer and Gyroscope) and their corresponding time stamps of the host that match the time at which the raw data was measured by each sensor. Note, that for the magnetic field values, these contain a constant delay due to the operation of the magnetometer in forced mode. The sensor raw data and the corresponding time stamps will be given as input to the library for further processing. The library input raw data should be in LSB values and time stamps in microseconds.

The following section describes the basic faults in the data which can be identified while retrieving the raw data from the hardware sensors.

4.2 Guidelines for analysis of the input data quality

This section explains the parameters that can be identified while retrieving data from the hardware sensors and some basic methods to identify the issues in the input raw data to the library.

The following table provides more details about the data quality identification.

Table 1: Data quality identification

Test	Case	Acceptance criterion	Tentative Methods to Identify
1	Repetition of samples per sensor	Determine the number of occurences of repetition of data per sensor the occurences of data repetition shall be zero (0). **Note**: As the raw data update by the sensor is based on an external oscillator, there should be no repetition. Data has to be updated every time iteration from the sensor.	Method 1 (for sensors providing the “sensortime”): Count the number of received data samples and compare with the number of samples that should have been received determined by subtracting the sensor time from start and stop of the data acquisition from the sensor Method 2 (for sensors not providing timing information): **Note:** method may work only when configuring the highest possible resolution and bandwidth that may differ from the desired setting No sample shall be found twice in the set of samples received from a sensor. The time stamps have to be consecutive, i.e 0;1;2;3;4;5;6;7;8;9,... (ignore a wrap around due to a limited number of bits).

2	Data loss	No loss of samples (data). Zero (0) occurences of loss of samples are accepted.	Consider data array of any sensor, if at any instant, the data buffer is updating with zero values, there will be some data loss is present. See also test 1. E.g.:- There should be a change in the sample value from the current instant to next instant in updating the value. E.g.: A =[1,2,3,4,5,0,0,0,0,0,0,8,9];
3	Limits to the per-frame timing deviation of the data acquisition	Tuples of sensors for data acquisition: - A+G - A+M - A+G+M For the recorded excitation(s), the data acquired from the configured combination of sensors shall be consistent. The deviation between timestamps of samples within the same read synchronization frame, the shall be less than or equal to +/-1%, in total 2%.	Get time stamp of each A/M/G sample which belongs to the same synchronization frame (e.g. A+G: 0ms, 20ms, 40ms...) for each frame and compute the absolute difference among the time stamps of this frame. Then, compare the differences among all frames. For the combinations A+M and A+G+M, the time stamp for the magnetometer values may be updated based on the accelerometer or the gyroscope time stamp.
4	Limits to the inter-frame timing deviation of the data acquisition	The median time stamp of each frame (see test 3), where samples of all sensors are acquired, shall not exceed +/- 1%, in total 2%.	(a) Read the configuration settings from the devices whether they match the required settings (b) Off-line settings: Calculate the duration among the median time stamp of data acquisitions, where samples of all sensors are acquired, and compare these to the inverse of the desired data rate (e.g. 100hz, 200hz...etc).
5	Latency of system operating	Deviation shall not be greater than +/- 1%, in total 2%. Game Mode: 50 Hz Latency < 20ms Fastest Mode: 100Hz Latency < 10ms Normal Mode: 25Hz Latency < 40ms UI Mode : 10 Hz Latency < 100ms	Method to identify the latency is to compare the actual time duration from one sample to other sample reference time period. Latency = Ref Time - Actual Time ; Hardware latency can be identified by the hardware interrupt time duration to the data updating time to the buffer.
6	Signal value	The samples provided shall match the required range and resolution.	(a) Read the configuration settings from the devices whether they match the required settings (b) Saturation test (needs to be clarified): The values of the device have to exhibit the expected behavior depending on the following defined movements: - Tilting/fall for the acceleration: saturation shall be visible in the acceleration signal for hits on the table - Shaking for the angular rate: saturation shall be visible in the angular rate signal only for fast movements to ensure the 500 deg/s configuration for the gyroscope Magnetic distortion for the magnetometer: distort the signal by exposition to a magnet and determine (c) Comparison test using a reference device Attach a working/reference solution with known parameters to the one under test and check whether the outputs show a similar behavior w.r.t.

			to saturation and number of received bits. (d) comparison tool data analysis module that automatically compares inputs from the respective devices. Note: method (d) has to be developed.
7	Noise levels	The sensor data noise should be within the specified range in data sheet according to the selected bandwidth.	When the device is still, the least signifficant bits of each value change consistently, e.g. for high bandwidth and resolution, or are constant, e.g. for low bandwitdh and resolution.

5. Implementation guidelines

5.1 Inclusion of library components

5.1.1 Inclusion of library

The library includes the following header files which need to be included along with BSXlite Library package.

Table 1: Required library header files

S. No	Header	Description
1	BsxLiteFusionLibrary.h	Provides the Library external APIs definitions.
2	BsxLibraryCalibConstants.h	Provides constants definition for calibration modules.
3	BsxLibraryConstants.h	Provides constants definition used by Library .
4	BsxLibraryDataTypes.h	Provides data types used by Library.
5	BsxLibraryErrorConstants.h	Provides error constants definition used by Library

Example:

```
/* Public headers */
#include " BsxLiteFusionLibrary.h  "
#include "BsxLibraryCalibConstants.h"
#include "BsxLibraryConstants.h"
#include "BsxLibraryDatatypes.h"
#include "BsxLibraryErrorConstants.h"
```

5.1.2 Inclusion of configuration

In addition to the base library, the library also supports configuration which has to be included of the sensor specification and additional internal configurations

Table 2: Required configuration string files

S. No	Folder	Description
1	accel_spec	contains acceleration sensor related settings for the library
2	mag_spec	contains magnetometer sensor related settings for the library
3	gyro_spec	contains gyroscope related settings for the library
4	usecase_config	contains additional required configuration of various internal modules of the library

The configuration may be different for different sensors. All supported specification will be available in *accel_spec, mag_spec and gyro_spec* folder. Similarly the library internal configuration may be different based on requirement. All supported configuration will be available in *usecase_config* folder.

5.2 Initialization of library

This section will give an overview about the standard initialization flow of BSXlite library. The following block diagram provides an overview of the standard initialization flow.

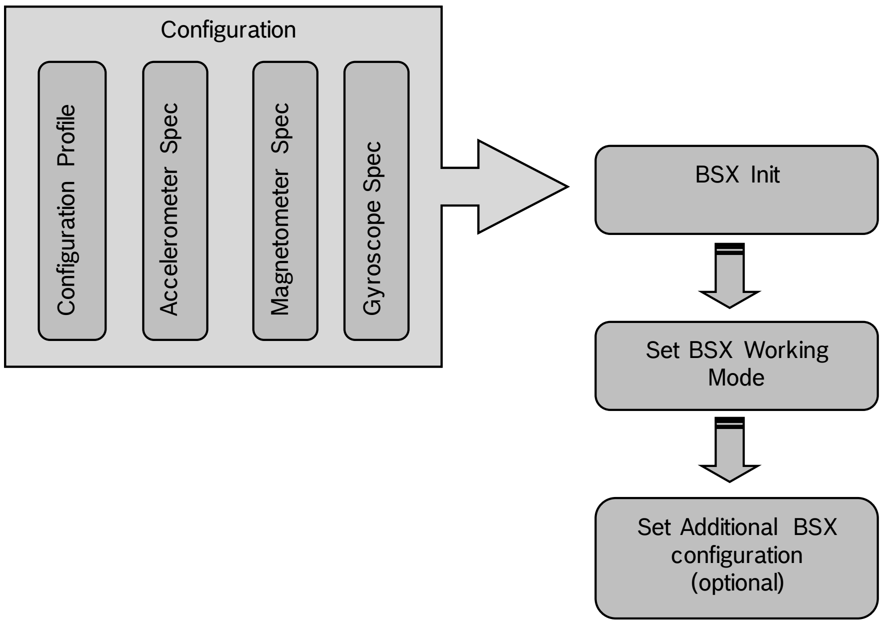

Figure 1: Overview of the standard initialization flow

5.2.1 Initialization

Once all the library files are included in the project, the library has to be initialized. It can be done with the API `bsx_init` () which has the below configuration included as structure member in addition to the return status

- Acceleration sensor configuration - contains acceleration sensor related settings for the library. So configuration will be different for different acceleration sensors
- Magnetic sensor configuration - contains magnetic sensor related settings for the library. So configuration will be different for different sensors
- Gyroscope sensor configuration - contains gyroscope sensor related settings for the library. So configuration will be different for different sensors
- Use case configuration - contains required configuration of internal modules of the library.

The overall status of initialization will be the return value. Individual initialization status with respect the sensor specs and configuration will be provided in individual structure members.

- *accelspec_status*
- *magspec_status*
- *gyrospec_status*
- *usecase_status*

The option of Initialization through default value is also available in case configuration file is not available. The above four parameters can be null pointer also and the library will be initialized with the default sensor specification (BMA250 + BMM050 + BMG160) and standard configuration as default depending on compiler switch.

Example:

```
/* Load configuration to array*/
initParam_t s_input;
BSX_S8 init_status;
/* Load BMA250 accel spec to structure member*/
s_input.accelspec = str_accsensorspec;
/* Load BMM050 mag spec to structure member*/
s_input.magspec = str_magsensorspec;
/* Load BMG160 gyro spec to structure member*/
s_input.gyrospec = str_gyrosensorspec;
/* Load use case spec to structure member*/
s_input.usecase = str_config;
```

```
/*API call with corresponding parameters*/
init_status = bsx_init(initParam_t &s_input);
```

```
/* check if initialization successful*/
if(init_status!= BSX_STATE_OK)
{
      printf("BSX Initialization Failed...\n");
      return init_status;
}
```

Please refer to the API description for more details on individual return status of sensor specification and configuration loaded.

5.2.2 Setting of working mode

The BSXlite library provides only one working mode NDOF_GEORV_FMC_OFF. The Table 3 shows available library outputs for this working mode

Table 3: BSX_LIBRARY_ALL : List of working mode and virtual sensors

Working Modes	Virtual Sensor Data Outputs								
	Accel	Mag	Gyro	Orientation	Rotation Vector	Game rotation vector	Linear Accel	Gravity vector	Geomagnetic rotation vector
NDOF_GEORV_FMC_OFF	Y	Y	Y	Y	Y	Y	Y	Y	Y

By default, the working mode will be set to sleep mode. The following API need to be used to set the required working mode.

```
BSX_S8 bsx_set_workingmode(ts_workingModes *p_workingmodes)
```

This API sets the operational modes of each modules in the library based on the set working mode. Also need to note that the restart of Ndof is taken care internally whenever there is a change in mode.

The following API needs to be used to get the list of sensor required for the set working mode.

```
BSX_S8 bsx_get_hwdependency(ts_workingModes,ts_HWsensorSwitchList* );
```

Example: To set the working mode to NDOF_GEORV_FMC_OFF

```
/* Declare the required structures */
ts_workingModes s_workingmodes;
ts_HWsensorSwitchList HWsensorSwitchList;

/* assign the desired working mode */
s_workingmodes.opmode = BSX_WORKINGMODE_NDOF_GEORV_FMC_OFF;

/* set the assigned working mode to library */
bsx_set_workingmode (&s_workingmodes);

/* get the hardware sensor dependency list */
bsx_get_hwdependency (s_workingmodes, s_HWsensorSwitchList);
```

5.2.3 Library sensor data rates

The BSXlite library supports only one set of data rates as shown in Table 4.

Table 4: List of working modes and supported data rates

S. No.	Working Mode	Data output rate			
		Accel	Mag	Gyro	Fusion data
1	SLEEP	-	-	-	-
2	NDOF_FMC_OFF	50Hz	20Hz	50Hz	50Hz

5.2.4 Set additional library configuration

This section will give an overview about the change in default configuration of standard library. The additional configurations will differ based on the requirements also; the following example will show how to change the default settings for one of the configurations,

Usage of sensor calibration profile functionality
Calibration Profiling is available to save and load the calibration parameters.

If the initial offsets need to be retained to reduce the time taken for initial calibration, use get profile APIs to store the well estimated offsets (check for stable and high accuracy status for a certain time regularly and update). Use the set profile APIs to load the stored offsets whenever there is a restart/reset based on the scenario and requirement.

Example:

```
/* Declare a structure variable */
ts_calibprofile s_calibprofile;
BSX_U8 magAcc;

/* Before stop of application */

/* check for the mag calibration status */
bsx_get_magcalibaccuracy(&magAcc);


/* store the calibration parameters is accuracy status indicates high
*/
if(magAcc == BSX_SENSOR_STATUS_ACCURACY_HIGH)
{
      bsx_get_magcalibprofile(&s_calibprofile);
}
```

```
/* After restart of application */

/* load the stored calibration parameters */
bsx_set_magcalibprofile(&s_calibprofile);
```

5.3 Execution of the main process of library

Once the initialization and the necessary configurations have been done, the library is ready for the main process. The main process API gets the raw accelerometer, magnetometer and gyroscope data in LSB format and the current system time in microseconds as input and does the required process based on the configuration.

5.3.1 Standard flow

1. Create variables based on the required data types as below:
2. Update the variables with the current sensor data
3. Update the timer values for corresponding sensor with the current timestamp
4. Call the main process API as below:

Example:

```
ts_dataxyz accdata, magdata, gyrodata;
libraryinput_t libraryInput_ts;
bsx_dostep(&libraryInput_ts);
```

5.3.2 Flow if calibration process need to be run in different slower thread

1. Create variables based on the required data types as below:
2. Update the variables with the current sensor data
3. Update the timer value for corresponding sensor
4. Call the main process as mentioned below. The main flow will be split into 3 API calls
   a. Preprocessing (includes the preprocessing related operations)
   b. Calibration processing (include the calibration related operations)
   c. Use case processing (includes the fusion data output related operations such as orientation, rotation vector, step counter, ...)

Example:

Main thread:

```
ts_dataxyz accdata, magdata, gyrodata;
libraryinput_t libraryInput_ts;
BSX_U8 usecasetick;

/* Calling do pre process */
bsx_dopreprocess(&libraryInput_ts);

/* Calling use case process */
bsx_get_usecasecalltick(&usecasetick);
/* If usecase tick is enabled, call dousecase */
if(usecasetick)
{
	bsx_dousecase();
}
```

Calibration thread:

```
BSX_U8 calibtick;
/* Calling do calibration */
bsx_get_calibrationcalltick(&calibtick);
If calib tick is enabled, call do calibration
if(calibtick)
{
       bsx_docalibration();
}
```

5.4 Get outputs from the library

The key outputs available and the corresponding units can be referred in Table 5

Table 5: Key library outputs and corresponding units

S.No	Data	Units 1	Units 2
1	Acceleration	Meters per seconds squared	-
2	Magnetic field	Micro-tesla	-
3	Angular rates	Degrees per second	Radians per second
4	Orientation Quaternion	LSB	Unit
5	Orientation Euler	Degrees	-
8	Geo Orientation Quaternion	LSB	Unit
9	Geo Heading accuracy.	Degrees	Radians per second

For the desired outputs the following steps can be followed.

1. Create variables required based on the required data types.
2. Call the required Library get APIs .

Example:

```
/* declaration of required variables */
ts_dataxyzf32 accRawData;
ts_dataxyzf32 magRawData, magCorData;
ts_dataxyzf32 gyroRawData_rps, gyroCorData_rps;
ts_dataxyzf32 accparam, magparam, gyroparam_rps;
ts_dataquatf32 orientQuat;
ts_dataeulerf32 orientEuler_rad;
ts_dataquatf32 georotationquat
BSX_F32 v_geoheading ;
```

```
/* Acc Data Interface */
bsx_get_accrawdata(&accRawData);

/* Mag Data Interface */
bsx_get_magrawdata(&magRawData);
bsx_get_magcordata(&magCorData);
bsx_get_magoffsets(&magparam);
bsx_get_magcalibaccuracy(&magAcc);
```

```
/* Gyro Data Interface */
bsx_get_gyrorawdata_rps(&gyroRawData_rps);
bsx_get_gyrocordata_rps(&gyroCorData_rps);
bsx_get_gyrooffset_rps(&gyroparam_rps);
bsx_get_gyrocalibaccuracy(&gyroAcc);


/* Use-case Data Interface */
bsx_get_orientdata_quat(&orientQuat);
bsx_get_orientdata_euler_rad(&orientEuler_rad);
bsx_get_orient_datastatus(&orient_stat);

/* additional features Interface */
bsx_get_georotationvector_quat(&georotationquat);
bsx_get_geoheadingaccuracy_rad(&v_geoheading);
```

Note: The above example does not contain the complete list of available library get APIs. For complete list please refer to the API Description Document.

5.5 Geo magnetic rotation vector

To get geo magnetic rotation vector and geo heading accuracy

Example*:*

```
 /* If wanted to get geomagnetic rotation vector features */
ts_dataquatf32 geo_rotation_quat;
BSX_F32 geo_heading_accuracy;

bsx_get_georotationvector_quat(&geo_rotation_quat) ;
bsx_get_geoheadingaccuracy_rad(&geo_heading_accuracy) ;
```

5.6 Axis Configuration

Raw data has to be remapped, which means the accelerometer, magnetometer and the gyroscope orientation have to be in line with the device orientation definition.

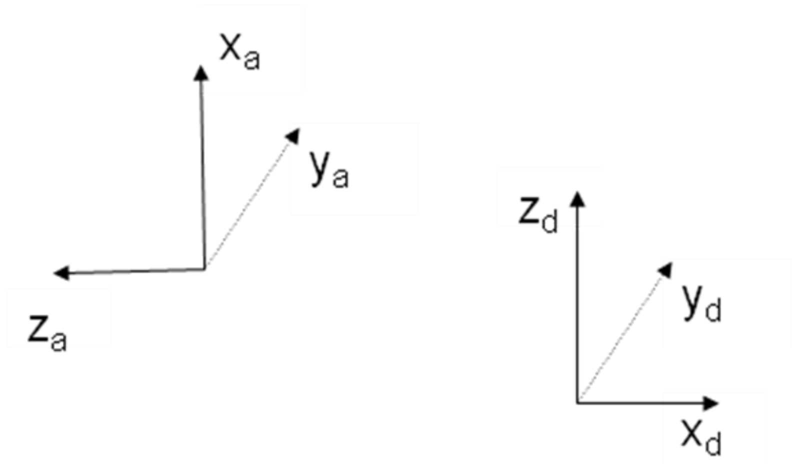

Figure 1: Example for axis remapping

Example:
For example if the device orientation is defined by (x_d, y_d, z_d) and the accelerometer orientation is defined by (x_a,y_a,z_a) as show in figure xxx the axis remapping is done by

```
acc.data.x = - raw_acc_z;
acc.data.y = - raw_acc_y;
acc.data.z = - raw_acc_x;
```

5.7 List of Android virtual sensors & corresponding mapping of APIs

This section will give the list of API's that can be used in real time environment for various applications.

Table 6: Key Android virtual sensor list and corresponding library APIs

S.NO	List of Virtual Sensors	Interfaces to virtual sensors from BSX Lite Library
	Base Sensors	
1	Accelerometer	bsx_get_accrawdata()
2	Geomagnetic	bsx_get_magrawdata() bsx_get_magcordata() bsx_get_magoffsets() bsx_get_magcalibaccuracy()
3	Gyroscope	bsx_get_gyrorawdata() bsx_get_gyrocordata_rps() bsx_get_gyrocalibaccuracy()
	Attitude sensors	
4	Rotation vector	bsx_get_orientdata_quat() bsx_get_headingaccuracy_rad()
5	Geomagnetic rotation vector (Magnetometer)	bsx_get_georotationvector_quat() bsx_get_geoheadingaccuracy_rad()
6	Orientation	bsx_get_orientdata_euler_rad() bsx_get_orient_datastatus()
	Un-calibrated sensors	
7	Gyroscope un-calibrated	bsx_get_gyrorawdata_rps()
8	Magnetic field un-calibrated	bsx_get_magoffsets()

Note: The above complete virtual sensor lists will adjust its performance based on change in working modes. For detail description about the list of API's please refer the API description document.

6. Abbreviation list

Abbreviation	Description
BSXlite	Bosch Sensortec Fusion Library Lite
RAM	Random Access Memory
ROM	Read Only Memory
MEMS	Micro Electro Mechanical System
NDOF	Nine Degrees Of Freedom
LSB	Least Significant Bit

7. References

1. Bosch Sensortec
   http://www.bosch-sensortec.com/en
2. Android Sensors Overview
   http://developer.android.com/guide/topics/sensors/sensors_overview.html

8. Legal disclaimer

8.1 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.

8.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They are not designed or approved for use in military applications, life-support appliances, safety-critical automotive applications and devices or systems where malfunctions of these products can reasonably be expected to result in personal injury. They may only be used within the parameters of this product data sheet.

The resale and/or use of products are at the Purchaser's own risk and the Purchaser's own responsibility.
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 accepts the responsibility to monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of any security relevant incidents.

8.3 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.

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 9
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-BSX03X-SD002-00