NXP Semiconductor
DATA SHEET: PRODUCT PREVIEW

Document Number: NSFK_DS
Rev. 0.8, 7/2016

NXP Sensor Fusion Library for Kinetis MCUs

Updated for NXP Sensor Fusion Release 7.00

Contents

freescale

1 Introduction

Sensor Fusion is the process where data from several different sensors are *fused* to complete computations that a single sensor could not handle. An example of sensor fusion is computing the orientation of a device in 3-dimensional space using an accelerometer and magnetometer. That data might then be used to alter the perspective presented by a 3D GUI or game.

The NXP Sensor Fusion Library for Kinetis MCUs provides advanced functions for computation of device orientation, linear acceleration, gyroscope offset and magnetic interference based upon the outputs of NXP inertial and magnetic sensors.

Features

- Supports:
  - — Accelerometer only (roll, pitch and tilt)
  - — Magnetometer only (2D auto)
  - — Gyro only
  - — Accelerometer plus magnetometer (eCompass)
  - — Accelerometer plus gyro (gaming)
  - — Accelerometer plus magnetometer plus gyroscope sensors
- Includes NXP's award-winning[1] magnetic compensation software
  - — Provides geomagnetic field strength, hard- and soft-iron corrections, and quality-of-fit indication
- Very low power consumption
  - — 1.4mA 9-axis fusion I_{DD} on Kinetis ARM Cortex M4F devices at 40 Hz fusion rate
  - — 0.4mA 6-axis fusion I_{DD} on Kinetis ARM Cortex M4F devices at 40 Hz fusion rate
- Programmable sensor sample and fusion rates
- Supports multiple 3D frames of reference (aerospace NED, Android and Windows 8)
- Library is coded in standard C99 ANSI C
- Compatible with the NXP Sensor Fusion Toolbox for Android and Windows
- Included as part of the Kinetis Software Development Kit (KSDK)
- Out-of-the box support for the following Freedom Development Platforms mated with either FRDM-FXS-MULT2-B or FRDM-STBC-AGM01 sensor boards
  - — FRDM-K64F
  - — FRDM-K22F
- Library version 7.00 includes support for both bare-metal and RTOS-based projects.

Typical Applications

- Notebook, tablet and smartphone sensor fusion
- Gaming, motion control, head-mounted displays, wearable electronics
- Air mouse, remote control
- Navigation, eCompass, IoT (Internet of Things) sensor data management

1 Freescale (now NXP) received the Electronic Products Magazine 2012 Product of the Year Award for our eCompass software.

Table 1. Feature Comparison of the NXP Sensor Fusion Algorithm Options

Feature	Accel only	Mag only	Gyro Only	Accel + gyro	Accel + mag	Accel + mag + gyro
Filter type	Low pass	Low pass	N/A	Indirect Kalman	Low pass[1]	Indirect Kalman
Roll / Pitch / Tilt in degrees	Yes	No	Yes	Yes	Yes	Yes
Yaw in degrees	No	Yes	Yes	No	Yes	Yes
Angular rate[2] in degrees/second	virtual 2 axis[3]	Yaw only	Yes	Yes	virtual 3 axis	Yes
Compass heading (magnetic north) in degrees	No	Yes	No	No	Yes	Yes
Quarternion and rotation vector	Yes	Yaw only	Yes[5]	Yes	Yes	Yes
Rotation matrix	Yes	Yaw only	Yes[5]	Yes	Yes	Yes
Linear acceleration separate from gravity	No	No	No	Yes	No	Yes
Aerospace (North-East-Down) frame of reference	Yes[4]	Yes	Yes[4]	Yes[4]	Yes	Yes
ENU (Windows 8 variant) frame of reference	Yes[4]	Yes	Yes[4]	Yes[4]	Yes	Yes
ENU (Android variant) frame of reference	Yes[4]	Yes	Yes[4]	Yes[4]	Yes	Yes
Magnetic calibration included	N/A	Yes	N/A	N/A	Yes	Yes
Gyro offset calibration included	N/A	N/A	No	Yes	N/A	Yes
FRDM-K64F board support	Yes	Yes	Yes	Yes	Yes	Yes
FRDM-K22F board support	Yes	Yes	Yes	Yes	Yes	Yes

1. More precisely: a non-linear modified exponential low pass quaternion SLERP filter.
2. Angular rate for 6 and 9-axis configurations with a gyro include corrections for gyro offset.
3. Subject to well-known limitation of being blind to rotation about axes aligned with gravity.
4. These solutions do not include a magnetometer, therefore there is no sense of compass heading.
5. Rotations for gyro-only solution are relative to sensor position (there is no global frame).

Table 2. Feature Options

Feature	Options
License	BSD 3-Clause
CPU selection[1]	MK64FN1M0VLL12 MK22FN512VLH12
Board customizable	Yes
Sensor sample rate	Programmable
Fusion rate	Programmable
Frame of Reference	Programmable
Algorithms Executing	Programmable
MCU sleep mode enabled between samples/calculations	No. It can be added by user, but will interfere with UART operation.
Power savings mode when stationary	Supported
RTOS	FreeRTOS
Code flexibility	Full source code is provided. All files are can be modified.
Product Deliverables	Delivered as part of the Kinetis Software Development Kit for selected platforms. Sensor Fusion specific deliverables include: • This datasheet • Software user guide • Extensive set of application notes • Preconfigured example applications • All source code

1. Listed MCUs are those supported by included project templates. The fusion library should be portable to any ARM® processor without change. Ports to other architectures are generally very straightforward, as the library is written in standard C.
2. Power savings when stationary is added at the application level. Example code is included in the sensor fusion user guide.

2 Functional Overview

2.1 Introduction

Sensor fusion encompasses a variety of techniques that:

- Trade off strengths and weaknesses of the various sensors to compute something more than can be calculated using the individual components
- Improve the quality and noise level of computed results by taking advantage of:
  — Known data redundancies between sensors
  — Knowledge of system transfer functions, dynamics and kinematics

The NXP Sensor Fusion Library for Kinetis (Fusion Library) supports several combinations of sensors. In general, performance improves as more sensors are added to the system. The primary function of the library is to compute orientation of a sensor subsystem relative to a global frame of reference.

Orientation can be expressed in a number of different ways:

- Tilt from vertical (may also be expressed as roll + pitch)
- Compass heading (geomagnetic north)

- Full 3D rotation from a global frame in any of the following forms:
  — Rotation matrix
  — Rotation vector (3D axis of rotation and rotation about that axis)
  — Quaternion
  — Euler angles (roll, pitch and yaw)

For additional portability details and guidelines refer to the *NXP Sensor Fusion for Kinetis MCUs User Guide* which is part of the *Sensor Fusion Library* installation. The Fusion Library is distributed in source code form and is designed to sit on top of board abstractions provided by the KSDK.

2.2 Accelerometer Only

An accelerometer measures linear acceleration minus gravity. If linear acceleration is zero, this sensor can be used to measure tilt from vertical, roll and pitch. Computation of yaw is not supported by this configuration.

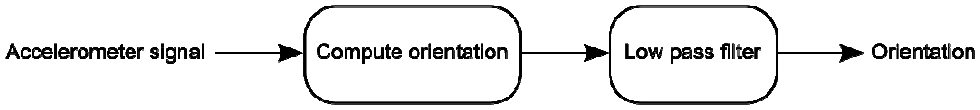


Figure 1. Accelerometer only block diagram

2.3 Accelerometer Plus Magnetometer

The accelerometer plus magnetometer configuration is often used as an electronic compass. The electronic compass is subject to the linear acceleration equals zero, assumption. Accuracy is dependent upon negligible magnetic interference from the environment in which the sensors travel.

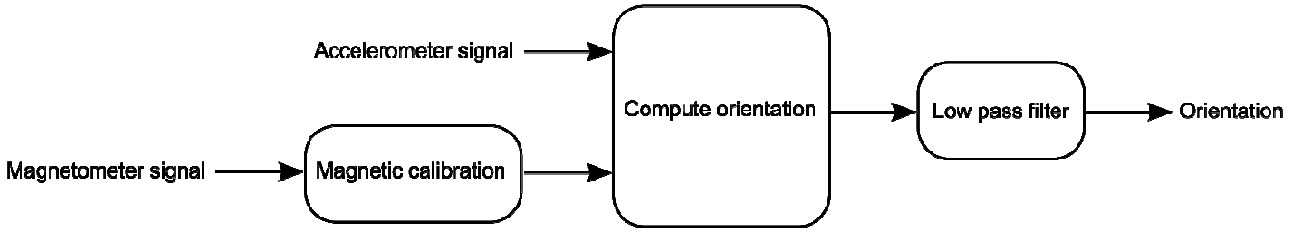


Figure 2. Accelerometer plus magnetometer block diagram

2.4 Accelerometer Plus Gyroscope

Using a gyroscope in addition to an accelerometer yields the ability to smoothly measure rotation in 3D space, although the system can only yield orientation to some random horizontal global frame of reference. That is, the system has no sense of magnetic north. Computation of yaw is not supported by this configuration.

This configuration is commonly known as an Inertial Measurement Unit (IMU).

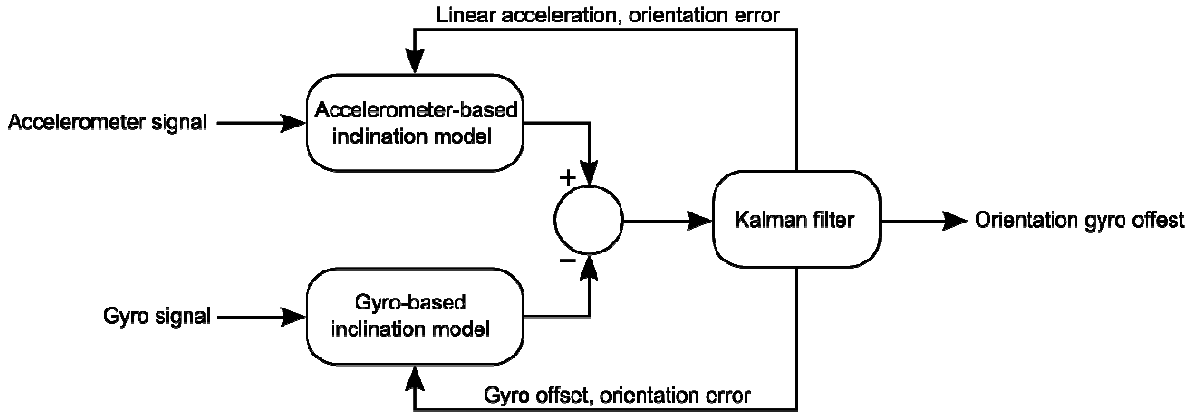


Figure 3. Accelerometer Plus Gyroscope Block Diagram

2.5 Accelerometer Plus Magnetometer Plus Gyroscope

Sometimes referred to as Magnetic, Angular Rate and Gravity (MARG), this subsystem offers an optimal combination of sensors for smooth tracking of orientation and separation of gravity and linear acceleration. This system is capable of yielding absolute orientation data with respect to magnetic north.

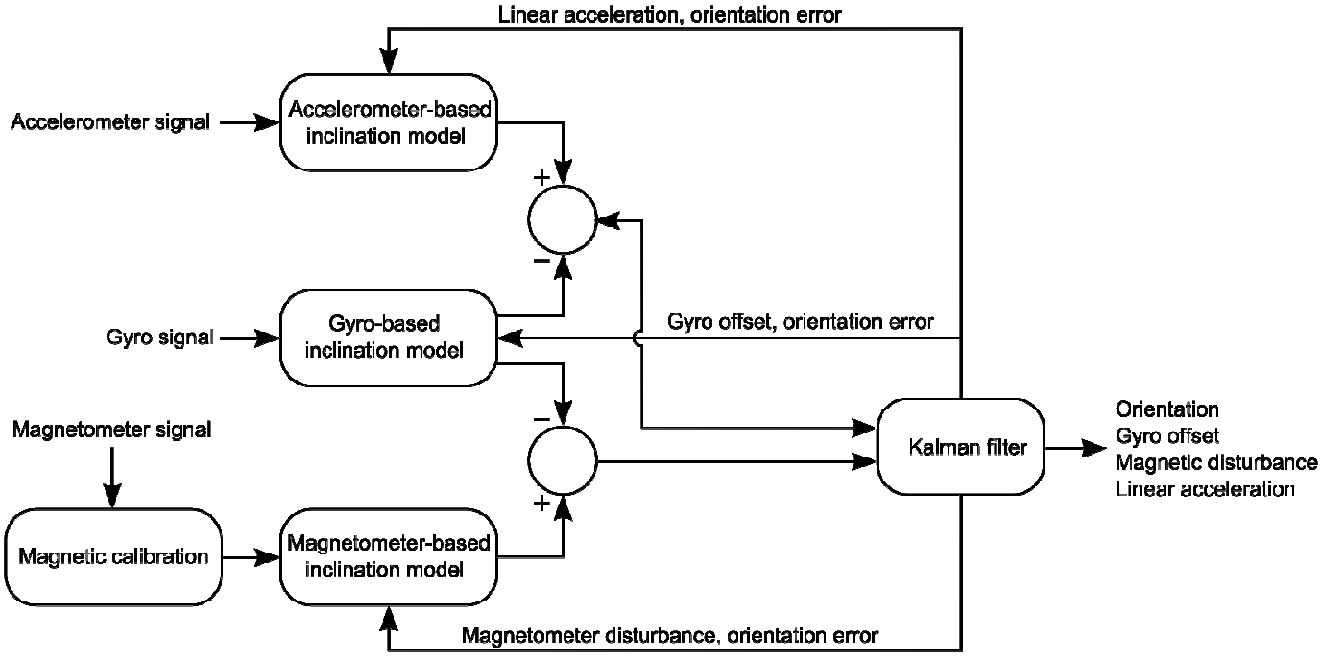


Figure 4. Accelerometer Plus Magnetometer Plus Gyroscope Block Diagram

3 Additional Support

NXP Sensor Fusion Toolbox provides support for both Android and Windows operating systems. See Table for the differences between the two implementations.

Table 3. NXP Sensor Fusion Toolbox Features by Platform

Feature	Android	PC
Bluetooth wireless link	✓	Requires BT on PC (built-in or dongle)
UART over USB	—	✓
OS requirements	>=Android 4.0.3	>=Windows 7.0
Support for native sensors	✓	—
Device View	✓	✓
Panorama View	✓	—
Statistics View	✓	—
Canvas View	✓	—
Orientation Value vs Time Plots	—	✓
Inertial Value vs Time Plots	—	✓
Magnetics	—	✓
Kalman Parameter Plots	—	✓
Altimeter Value vs Time Plots	—	✓
Data Logging	✓	✓
Integrated documentation	✓	✓
Availability	Google Play	NXP website
Price	Free	Free

3.1 NXP Sensor Fusion Toolbox for Android

The Fusion Library is supplied in the form of CodeWarrior projects for specific NXP development boards. The basic function of the Sensor Fusion Android implementation are shown in Figure 5. These projects are compatible with the NXP Sensor Fusion Toolbox for Android, which can be freely downloaded from Google Play. For download and training links, visit *NXP.com/sensorfusion*.

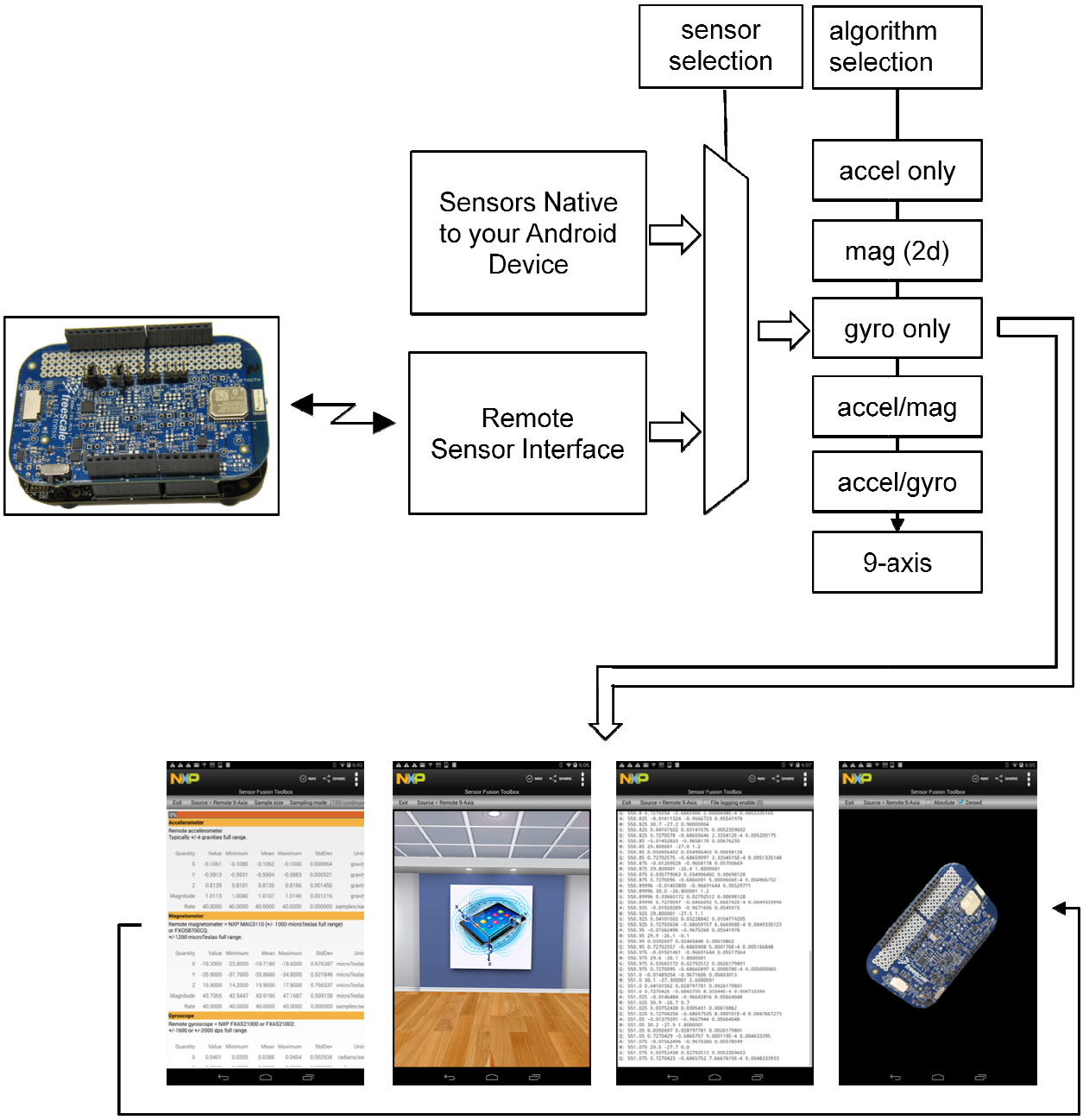


Figure 5. NXP Sensor Fusion Toolbox for Android Basic Functions

3.2 NXP Sensor Fusion Toolbox for Windows

The Sensor Fusion Toolbox includes an equivalent version of the software for Windows. For download and training links, visit *NXP.com/sensorfusion*.

Important Note: Be sure you have downloaded the Windows toolbox built 23 Junet 2016 or later.

Figure 6 and Figure 7 are Sensor Fusion Toolbox screenshots of the Windows version.[2]

[2] This is the 23 June 2016 version of the tool. Appearances will vary slightly from version to version.

Additional Support

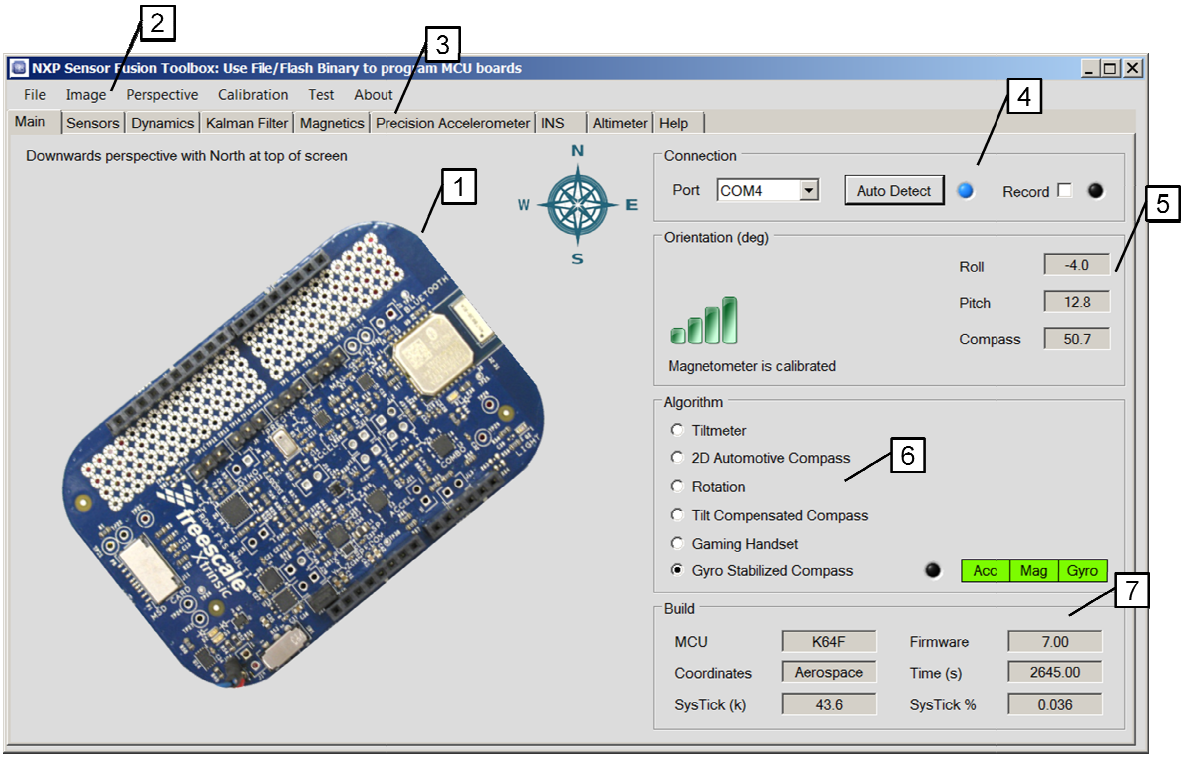


Figure 6. PC Version - Main Tab

Figure 6 is a snapshot of the Main Tab from the Windows version of the Sensor Fusion Toolbox. It has the following components:

1. Rotation 3D PCB display
2. Pull-down navigation menus: File, Image, Perspective, Calibration, Test and About
3. Navigation tabs for:
    - Main (board view)
    - Sensors
    - Dynamics
    - Magnetics
    - Precision Acclerometer
    - INS
    - Altimeter
    - Help
4. Communications
    - Communications port selection
    - Auto-detect button
    - Activity indicator
    - Record control for creating data logs

5. Roll / Pitch / Compass and Magnetic Calibration Status
6. Algorithm selector
7. Sensor board run-time and build parameters

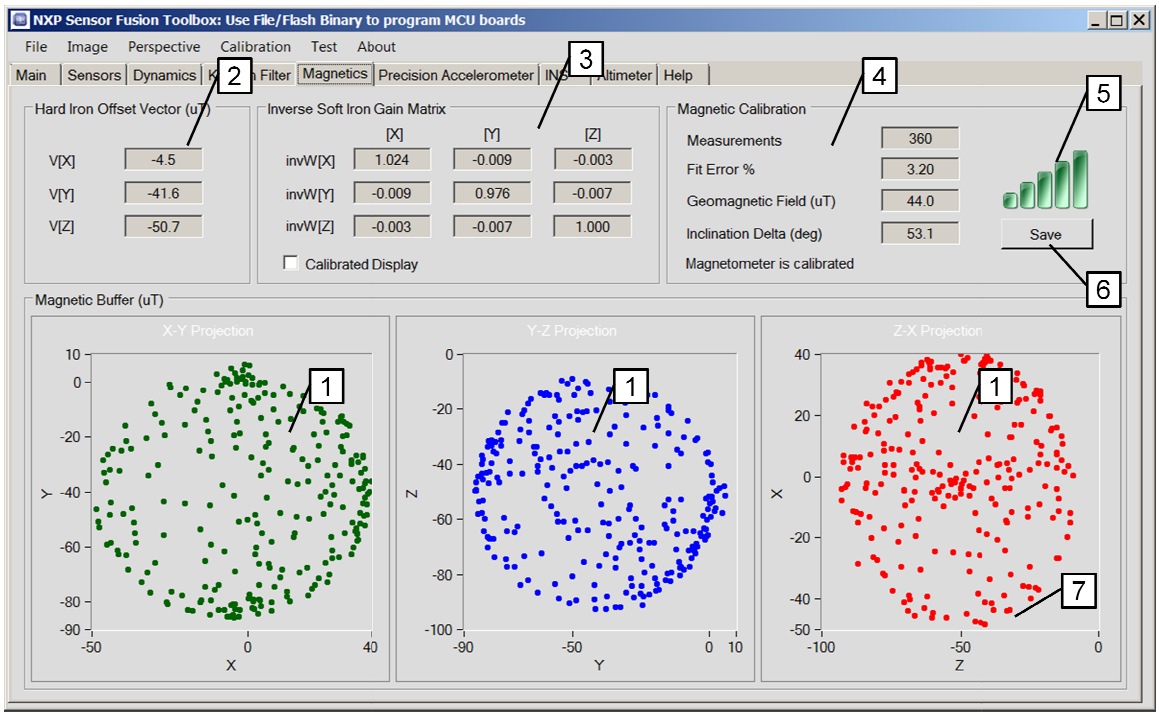


Figure 7. PC Version - Magnetics Tab

Figure 7 illustrates the Magnetics tab of the toolbox. Labelled features include:

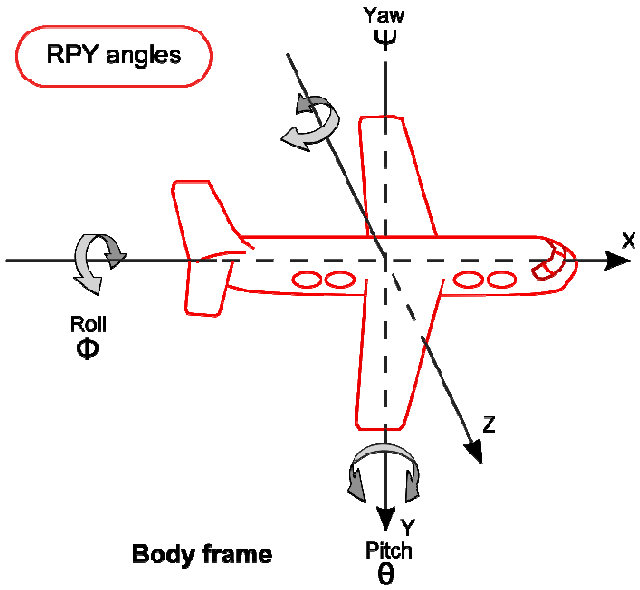
1. 2D representation of the data point cloud used for hard/soft iron compensation
2. Computed hard iron vector
3. Soft iron matrix
4. Statistics
5. Calibration indicator
6. Save to text file control

3.3 Terms and Acronyms

Table 4: Terms and Acronyms

Term	Definition
DUT	Device Under Test
ENU	A global frame of reference described by X = **E**ast, Y = **N**orth, Z = **U**p
g	abbreviation for gravities. 1 standard gravity = 9.80665 m/s^2. Accelerometers are commonly trimmed using the local gravimetric field, which can vary by 1/2 percent depending upon altitude and latitude.
gauss	CGS system unit for measuring magnetic field strength. 100 µT = 1 gauss.
IMU	Inertial Measurement Unit = accelerometer + gyro

Additional Support

Term	Definition
Kinetis	NXP family of ARM®-based MCUs
MagCal	Magnetic Calibration
MARG	Magnetic Angular Rate Gravity = IMU + magnetometer
MCU	Micro-Controller Unit
microTesla	The Tesla is the SI unit for measuring magnetic field strength. 1E–4 Tesla = 100 μT = 1 gauss
NED	A global frame of reference described by X = **N**orth, Y = **E**ast, Z = **D**own
pitch	In the Aerospace/NED frame of reference, defined as a rotation about the Y-axis
RHR	Right Hand Rule—a standard convention for describing the positive/negative sense of rotations about an axis of rotation. See http://en.wikipedia.org/wiki/Right_hand_rule.
roll	In the Aerospace/NED frame of reference, defined as a rotation about the X-axis
RPY	Roll, Pitch, and Yaw. In the Aerospace/NED frame of reference these are defined as rotations about the X axis,Y axis, and Z axis 
SI	International System of Units (meter, kilogram, second, ...)
SLERP	**S**pherical **L**inear int**ERP**olation - See http://en.wikipedia.org/wiki/SLERP
SysTick	A feature of the ARM® processor; a clock timer which for the purposes of this discussion, 1 sysTick = 1 CPU clock cycle.
tilt	Angle from vertical
UART	Universal Asynchronous Receiver / Transmitter, also known as SCI (Serial Communications Interface)
yaw	In the Aerospace/NED frame of reference, defined as a rotation about the Z-axis

3.4 References

Included in with the sensor fusion source code, you will find:

- This data sheet
- NXP Sensor Fusion for Kinetis MCUs User Guide
- NXP Application Note AN5016, Rev. 2.0: Trigonometry Approximations
- NXP Application Note AN5017, Rev 2.0: Aerospace, Android and Windows 8 Coordinate Systems

- NXP Application Note AN5018, Rev. 2.0: Basic Kalman Filter Theory
- NXP Application Note AN5019, Rev. 2.0: Magnetic Calibration Algorithms
- NXP Application Note AN5020, Rev. 2.0: Determining Matrix Eigenvalues and Eigenvectors by Jacobi Algorithm
- NXP Application Note AN5021, Rev. 2.0: Calculation of Orientation Matrices from Sensor Data
- NXP Application Note AN5022, Rev. 2.0: Quaternion Algebra and Rotations
- NXP Application Note AN5023, Rev. 2.0: Sensor Fusion Kalman Filters
- NXP Application Note AN5286, Rev. 2.0: Precision Accelerometer Calibrations

External references you may find useful include:

- www.nxp.com/sensorfusion
- MCU on Eclipse blog at https://mcuoneclipse.com/
- Kinetis Design Studio software at www.nxp.com/kds
- Euler Angles at en.wikipedia.org/wiki/Euler_Angles
- Introduction to Random Signals and Applied Kalman Filtering, 3rd edition, by Robert Grover Brown and Patrick Y.C. Hwang, John Wiley & Sons, 1997
- Quaternions and Rotation Sequences, Jack B. Kuipers, Princeton University Press, 1999
- NXP Freedom development platform home page at nxp.com/freedom
- OpenSDA User's Guide, NXP Semiconductors N.V., Rev 0.93, 2012-09-18
- NXP OpenSDA support page at nxp.com/opensda
- PE micro OpenSDA support page at www.pemicro.com/opensda
- Segger OpenSDA support page at https://www.segger.com/opensda.html

4 Mechanical and Electrical Specifications

4.1 General Considerations

Fusion algorithms can be *tuned* to trade off one performance parameter versus another. Examples include:

- Speedy handling of magnetic interference versus slower convergence to magnetic north
- Smoothness versus responsiveness
- Accuracy versus bandwidth

All of the above means that there is no *one correct* configuration. Accordingly, this datasheet presents typical performance as observed on the sample projects supplied by NXP on specific NXP development platforms. No attempt has been made to measure a statistically significant number of boards. Measured values are typically those observed on as few as one board.

4.2 Hardware Platforms Used to Measure Performance

In the following subsections, some parametrics are measured, some represent simulated results. Hardware platforms used for benchmarking purposes are briefly outlined in the following subsections. All boards shown are described in more detail, and can be ordered at nxp.com/freedom.

4.2.1 Sensor Shields

The FRDM-MULT2-B sensor shield is design to be mated to a variety of NXP Freedom development boards. It includes a number of sensor types, as well as a 3rd-party Bluetooth communications module.

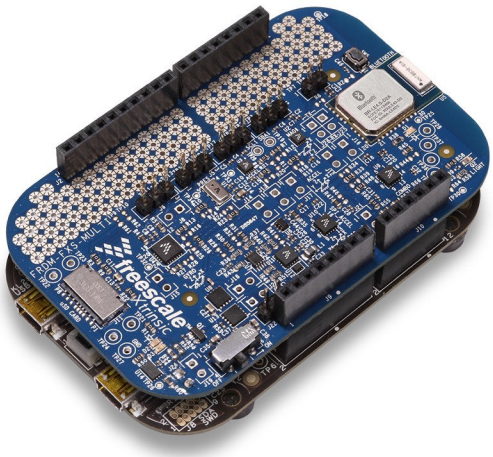


Figure 8. FRDM-KL25Z / FRDM-FXS-MULTI-B sensor fusion prototype platform

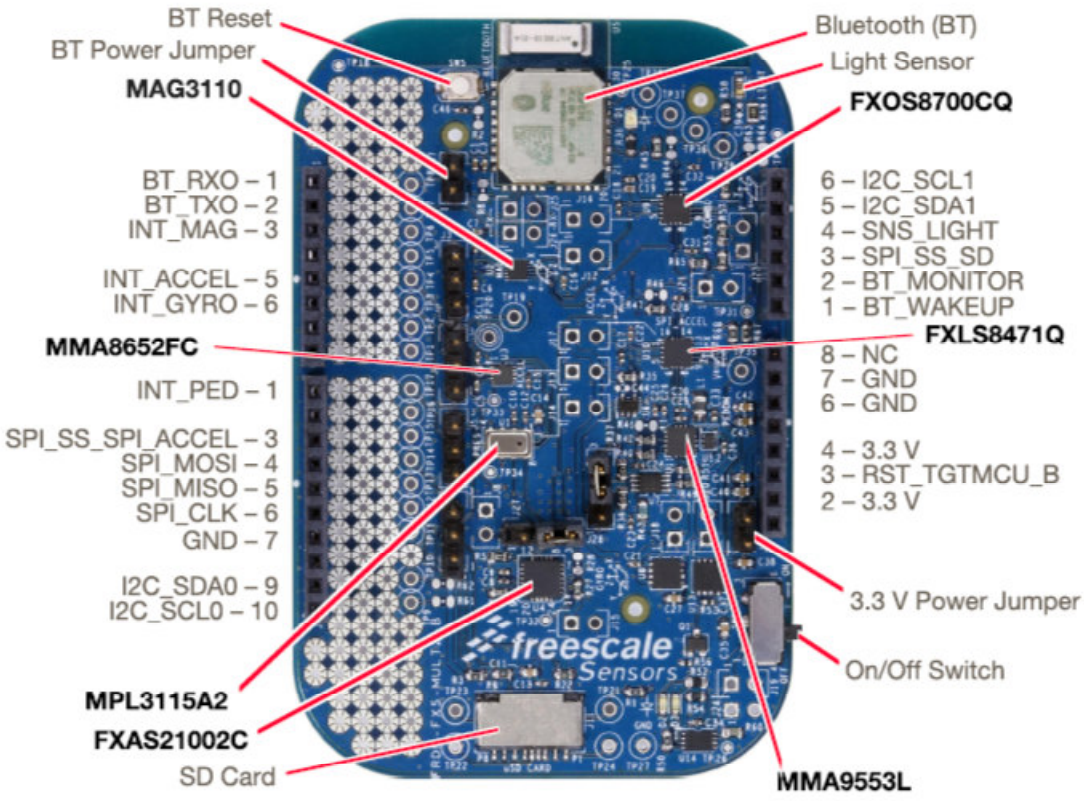


Figure 9. Expanded Diagram of FRDM-FXS-MULT2-B Sensor Board

The FRDM-STBC-AGM01 can be viewed as a proper subset of the FRDM-MULT2-B. FRDM-AGM01 Sensors are also present on the FRDM-MULT2-B. The FRDM-AGM01 does not include wireless communications capabilities.

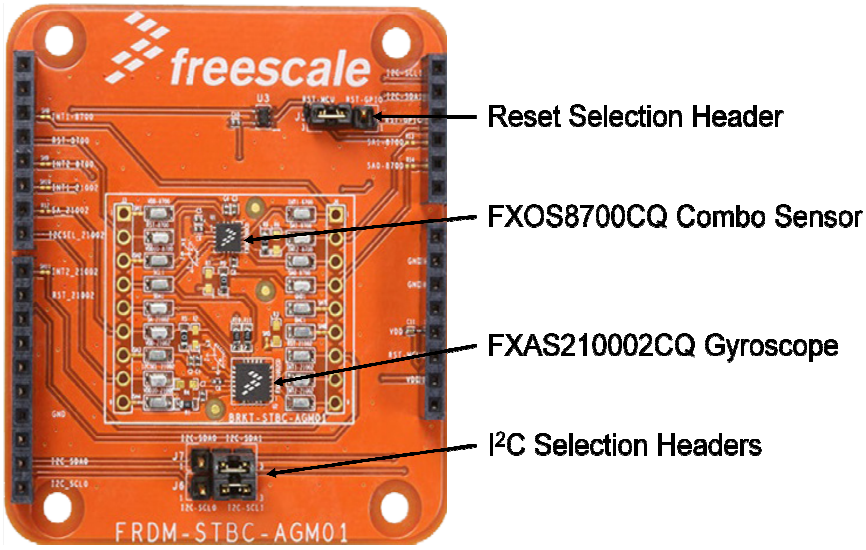


Figure 10: FRDM-STBC-AGM01

4.2.2 **Freedom Development Platforms**

The initial release of Version 7.00 of the Sensor Fusion library supports FRDM-K64F and FRDM-K22F Freedom development platforms. The two boards are similar in capabilities, although the K64F includes an Ethernet port, whereas the K22F does not.

Shields and base boards can be mixed and matched.

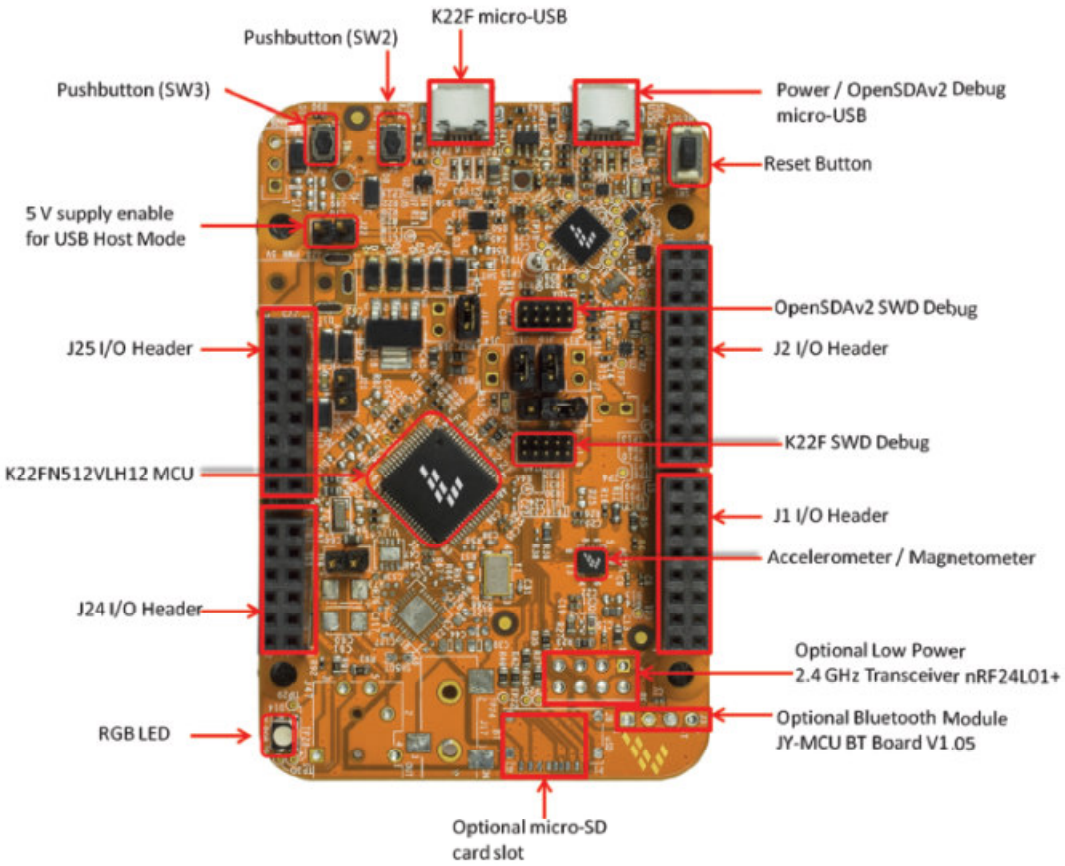


Figure 11: FRDM-K22F

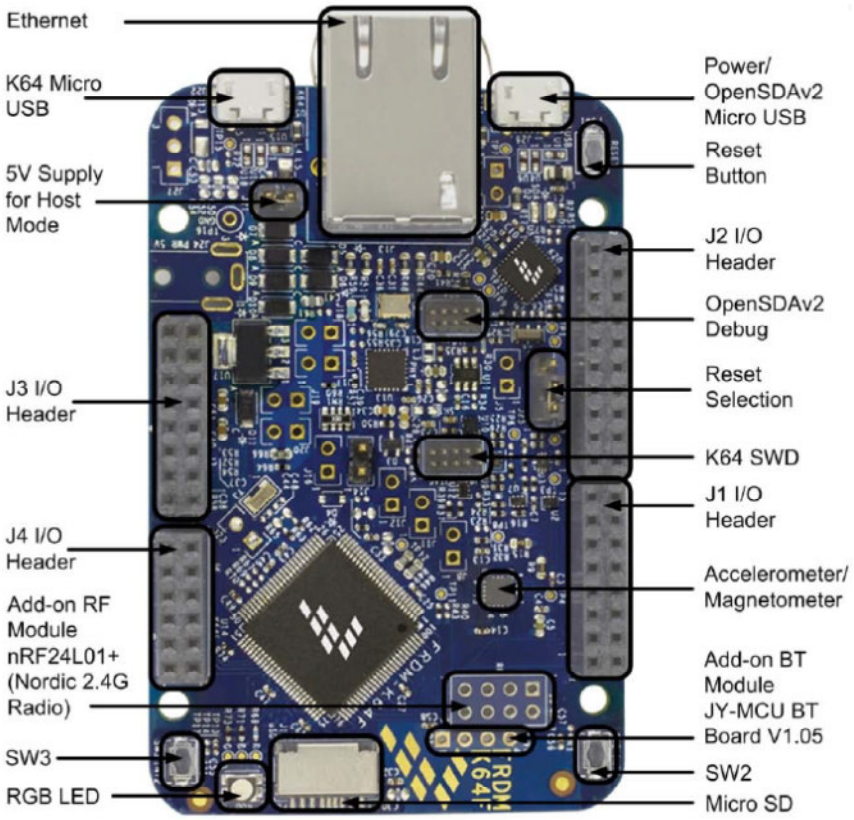


Figure 12: FRDM-K64F

4.3 Simulation Environments

Many parameters are difficult, if not impossible, to reliably measure in a lab or production environment. Ambient magnetic fields vary tremendously in indoor environments. Determining filter sensitivities to various input parameters can only be done in a simulated environment.

Accordingly, the subsequent sub-sections discuss environments that may be used to explore these areas.

Pure Matlab

Matlab is commonly used in the early development of sensor fusion algorithms. It can be used to model physical stimulus, expected responses and filter operation. See the next section for details concerning sensor and environmental models used to construct stimulus.

Matlab Stimulus/Expected Response + Hardware-based Fusion

Once algorithms are functional, they are translated into C, optimized and then optimized again. The result often makes use of fixed point arithmetic running on an MCU to compute results. To ensure bit-accurate results, a mixture of Matlab (used to generate test stimulus and expected response) and hardware (used to run the filter) are used to determine many parameters.

Simulated values were computed with version 4.17 of the sensor fusion library. Version 7.00 is expected to have better dynamic tracking than indicated in this datasheet, but simulations are still outstanding.

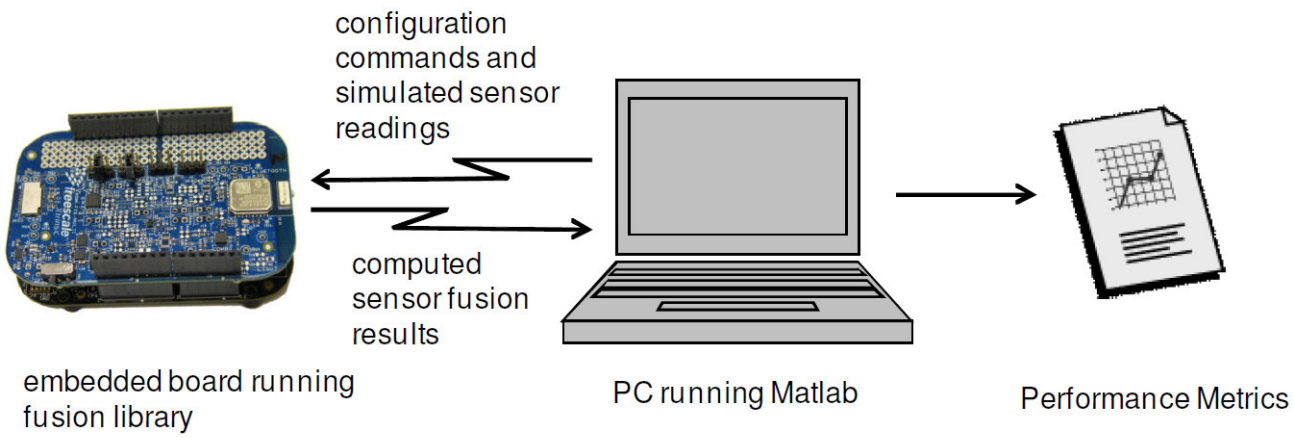


Figure 10. Mixed Simulation/Hardware Characterization

Unless stated otherwise elsewhere, parameters used to create the simulated sensors and environment are:

- Earth magnetic field corresponding to U.S. zip code 85284 (Tempe, Arizona) on 7 November 2013 as determined using the NOAA calculator at ngdc.noaa.gov/geomag-web
  - — Declination: 10 degrees 39’ 36”
  - — Inclination: 59 degrees 32’ 53”
  - — Horizontal Intensity: 24.2976 µT
  - — North Component (+N | -S): 23.8782 µT
  - — East Component (+E | -W): 4.4945 µT
  - — Vertical Component (+D | -U): 41.3285 µT
  - — Total field: 47.9418 µT
- Ideal accelerometer model + simple noise
  - — Sample rate = 200 Hz
  - — milli-*g* noise on X,Y,Z
  - — 1 *g* = 9.80665 m/s^2 assumed
- Ideal magnetometer model + simple nose
  - — Sample rate = 200 Hz
  - — X, Y noise 0.85 µT; Z noise 1.3 µT
- No hard/soft iron distortion
- Ideal gyroscope model and simple noise
  - — Sample rate 200 Hz
  - — X, Y, Z noise 0.3 dps

For all tests, the NXP Matlab-based Trajectory & Sensor Simulation Toolkit (TSim) is used to create DUT trajectories and simulated sensors readings.

Some of the tests require that magnetic calibration procedure is run before the test to initialize the sensor fusion software magnetic buffer. Magnetic calibration is implemented as a trajectory made up of a sequence of random rotations lasting 30 seconds.

Benchtop Rotary Table

A variant on the Matlab-based system is shown in Figure 11. In this case, Matlab is used to control a desktop rotary table and to post-process the fusion results into reports. Physical sensor readings are used to drive the fusion routines. Because indoor environments vary magnetically, results may vary from day to day as the setup is moved or electronics in the area cycle on and off.

Results will also change simply because the state of the magnetic buffer is continuously updated.

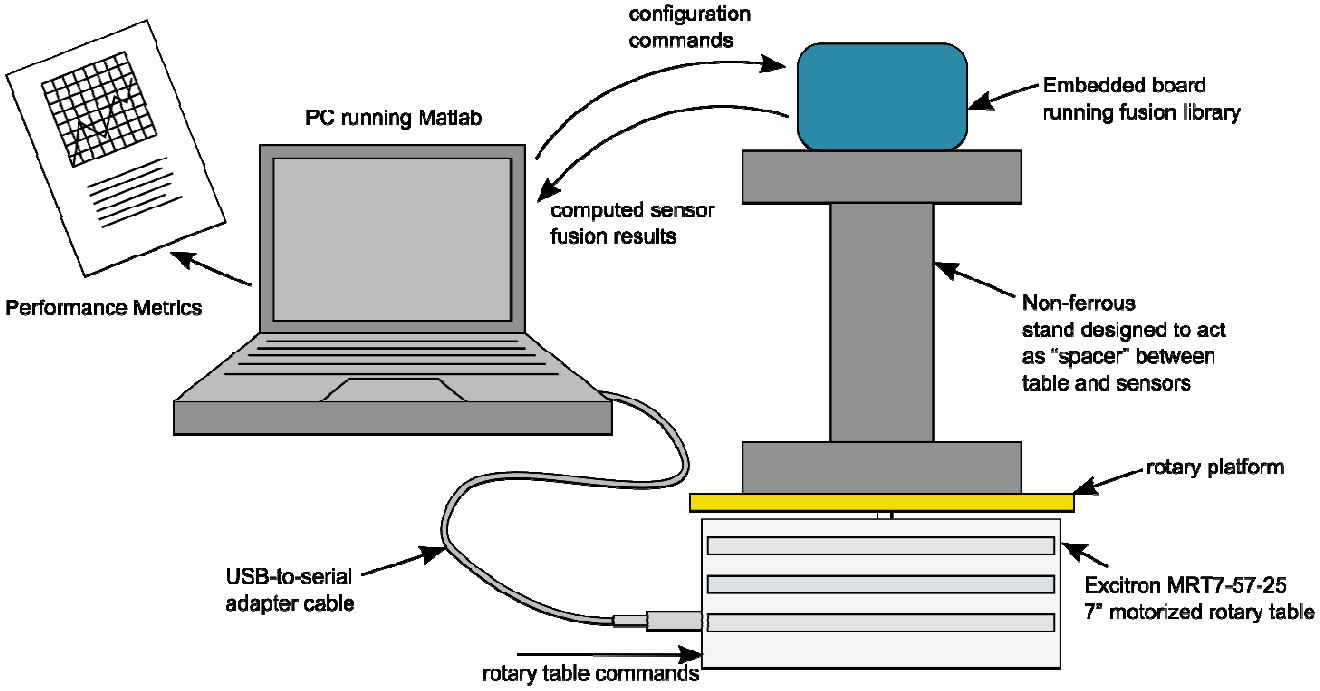


Figure 11. Benchtop Rotary Table Setup

4.4 Frame of Reference

Table summarizes differences between the standard frames of reference supported by the fusion library.

Table 5. Frame of Reference Variations

	NED	Android	Windows 8
Axes alignment	NED	ENU	ENU
Angle rotation order	Yaw then pitch then roll	Yaw then roll then pitch	Yaw then pitch then roll
Gimbal lock	Roll instability (x axis) at ±90 deg pitch (y axis)	Pitch instability (x axis) at ±90 deg roll (y axis)	Roll instability (y axis) at ±90 deg pitch (x axis)
Roll range	Clockwise –180 to 180 deg	Anti-clockwise –90 to 90 deg	Clockwise –90 to 90 deg
Pitch range	–90 to 90 deg	–180 to 180 deg	–180 to 180 deg
Yaw range	0 to 360	0 to 360	0 to 360
Compass heading	Yaw	Yaw	–Yaw

1. A clockwise rotation is defined as one that is positive in the Right-Hand-Rule (RHR) sense.

4.5 Electrical Specifications

Table 6. IDD / SysTick Testcase Configurations

Component / Parameter	FRDM-K22F	FRDM-K64F
MCU	MK22FN512VLH12	Kinetis MK64FN1M0VLL12
CPU	ARM® Cortex M4 with Floating Point Unit	
CPU clock used for benchmarking	80 MHz	120 MHz
Bus clock used for benchmarking	40 MHz	60 MHz
RTOS	None – Bare Metal Implementation used to measure parametrics	
Compiler	IAR, Optimization Level = High (Balanced)	
Accelerometer	FXOS8700CQ	
Magnetometer		
Gyroscope	FXAS21000	
Accelerometer Sampling Rate	200 Hz	
Magnetometer Sampling Rate	200 Hz	
Gyroscope Sampling Rate	400 Hz	
Fusion Rate	40 Hz	
Magnetic Calibration	Serialized to run as part of the fusion loop	

The parameters in tables that follow are measured using the configurations defined in Table 6 above. The MCU currents are estimated using the process defined in MCU Current. Numbers that follow do not include power consumed by sensor sampling or signal conditioning. They reflect only the core fusion algorithm itself.

Tables 7 and 8 were developed using the Version 7.00 of the Sensor Fusion Library.

Table 7. Typical I_{DD} Executing on FRDM-K22F

Function	Fusion I_{dd} @ 40 Hz rate (mA)	Fusion I_{dd} / Hz (µA)
Accel only	0.08	2
2D Mag	0.07	1.8
Gyro only	0.05	1.4
Accel + Mag, eCompass	0.09	2.3
Accel + Gyro	0.41	10.3
9-axis	1.2	30.1

Table 8. Typical I_{DD} Executing on FRDM-K64F

Function	Fusion I_{dd} @ 40 Hz rate (mA)	Fusion I_{dd} / Hz (µA)
Accel only	0.13	3.3
2D Mag	0.09	2.1
Gyro only	0.06	1.6
Accel + Mag, eCompass	0.12	2.9
Accel + Gyro	0.42	10.5

Function	Fusion I_{dd} @ 40 Hz rate (mA)	Fusion I_{dd} / Hz (µA)
9-axis	1.41	35.3

The currents in Table 9 were measured using a FreeRTOS-based project with sampling rates similar to the above, but with the addition of separate 40 Hz fusion and 200 Hz sensor sampling task to the RTOS-based project. The project is designed to shut down the gyro when no motion is detected by the accelerometer. Measured current includes IDD for FXOS8700CQ, FXAS21002 and dual 4.7K I^2C pullup resistors on the board.

Table 9. Typical Current Draw on AGM01 Sensor Board

Function	Measured $I_{dd\ P3V3}$ (mA)
Run Mode	3.4
Gyro Standby Mode	0.7

4.6 Computation Metrics

The Sensor Fusion Library computations are measured directly using the Sensor Fusion Toolbox and the ARM® sysTick clock, using the configuration outlined in the previous section.

4.6.1 Statistics from Version 7.00 of the Sensor Fusion Library

NXP used the built in sysTick counter to measure each iteration of the fusion algorithms in units of CPU clock cycles.

The customer can repeat the exact same measurement, because both PC and Android Sensor Fusion Toolboxes display this information in a real-time basis. The test results may vary, depending device movement, as presented numbers are “typical”.

Table 10. K SysTick Values for Freedom Development Platforms and Sensor Combinations[1]

Freedom Platform	Accel	2D Mag	Gyro	eCompass	Accel +Gyro	9-axis
FRDM-K22F	3	2.7	2.1	3.6	16	46.7
FRDM-K64F	4	2.6	1.9	3.5	12.7	42.8

1 Based on “All Motions” build, which includes tilt, 2D automotive compass, rotation, eCompass, 6- and 9-axis Kalman filters. Does not include pressure/altimeter.

Table 11. Memory (in bytes) Requirements for "All Motion Algorithms" Builds[1]

Freedom Platform	Readonly code memory	Readonly data memory	Readwrite data memory[2]
FRDM-K22F	58,714	322	12,049
FRDM-K64F	58,430	322	12,057

2 Includes tilt, 2D automotive compass, rotation, eCompass, 6- and 9-axis Kalman filters. Does not include pressure/altimeter.

3 If your Readwrite data memory numbers differ significantly from those shown above, it may be due to your choice of linker configuration file. The sensor fusion team has noted that the default files shipped with IAR give a much larger number than the IAR linker configuration files shipped with the NXP KDSK example projects.

Table 12. Memory (in bytes) Requirements for FRDM-K22F and Single Algorithm

Algorithm	Readonly code memory	Readonly data memory	Readwrite data memory
Accel (Tilt)	36,110	286	4,809
2D Mag	40,918	286	8,573
Gyro Only	28,886	286	3,553
Accel/Mag	47,882	286	10,005
Accel/Gyro	40,430	294	5,361
9-axis	52,854	294	11,101

4.7 Magnetic Calibration Metrics

4.7.1 Background

Hard-iron effects are due to magnetic materials in the vicinity of the sensor. These materials result in an apparent offset to sensor readings when the source of interference is fixed spatially, relative to the sensor.

For a given point in space, plotting magnetometer measurements at various sensor rotations results in the sphere shown on the right hand side of Figure 15. This makes sense, as the magnitude of the 3D magnetic field should not change just because the sensor is rotated.

Soft-iron effects result from the interaction of ferrous materials near the sensor interacting with the ambient magnetic field. If the source of soft iron interference is again fixed spatially with respect to the sensor and does not demonstrate magnetic hysteresis, then the *sphere of plotted measurements* is distorted into an ellipsoid. This is shown (along with a hard-iron offset) on the left side of Figure 12.

Reference **Error! Reference source not found.**: Orientation Representations provides background on the topic of hard- and soft-iron magnetic compensation. For the case where the sensor and sources of interference are spatially fixed with respect to one another, the distortions are linear, and can be reversed mathematically. This is the function of the NXP magnetic calibration library.

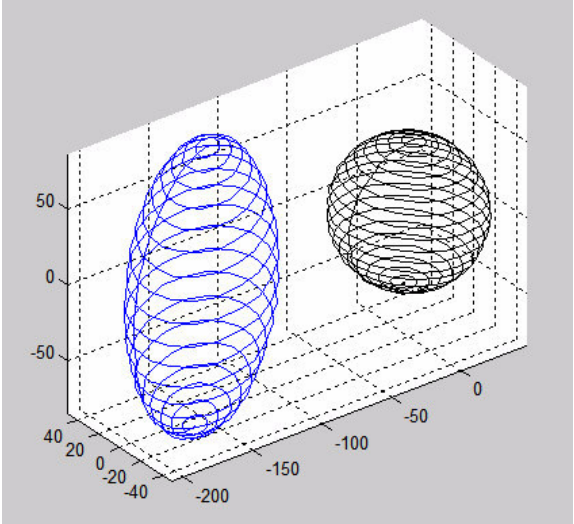


Figure 12. Distorted (left) and Corrected (right) Magnetic Field Data (simulated)

Both hard- and soft-iron interferences are a function of the sensor environment, and not the sensor itself. Each product design will inevitably result in different distortions.

Engineers assigned the task of physically designing PCBs and housings should pay careful attention to sources of magnetic interference early in the design phase.

Inductive charging films found in some portable devices exhibit a significant amount of magnetic hysteresis. This is a nonlinear phenomena and cannot be fully corrected by the NXP magnetic calibration library.

4.7.2 **The Magnetic Buffer**

NXP's magnetic calibration library performs a total least squares fit of a number of data points to map the measured ellipsoid of measurements back into the ideal sphere. Quality of that fit improves as number and spacing of samples across the ellipsoid surface increases. There is a tradeoff in terms of data set size used for calibration versus CPU resources versus quality of fit.

Figure 16 shows improvement in standard deviation of computed results (for a uniform field) versus constellation size.

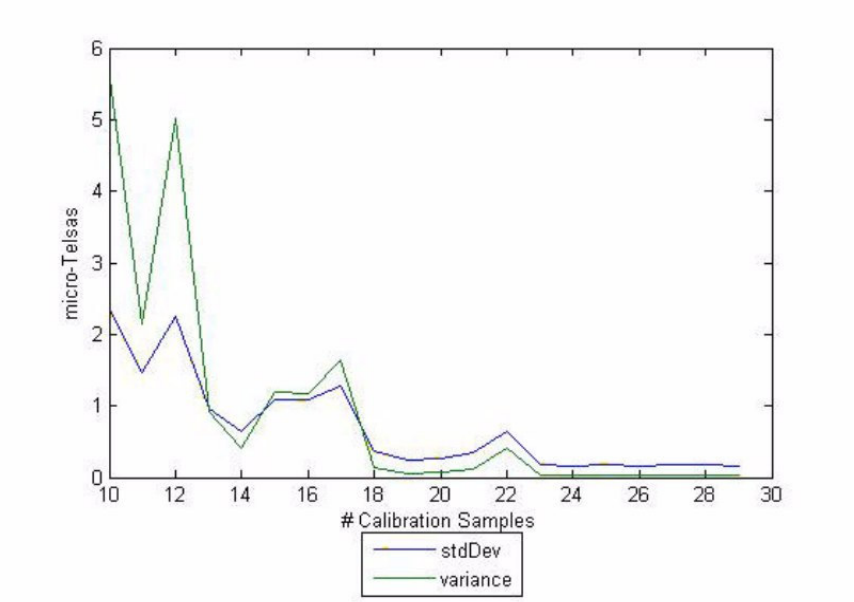


Figure 13. Simulated Standard Deviation and Variance as f(constellation size)

4.7.3 Magnetic Calibration Performance Metrics

Table 13 provides some basic guidance with regard to performance of the magnetic calibration library in a stand-alone configuration.

Table 13. Magnetic Calibration Performance Metrics

Characteristic	Symbol	Conditions	Min	Typ	Max	Unit
Compass heading linearity; see Compass Heading Linearity and Accuracy	TBD	—	—	< 5	—	degrees
Compass heading accuracy; see Compass Heading Linearity and Accuracy	TBD	—	< 5	—	degrees	
NOTE: The results shown in this table are more conservative than simulated numbers, and are more likely to be representative of actual results.						

4.8 Fusion Model Performance Metrics

4.8.1 Background

The six and nine-axis Kalman filters are optimized for calculation of device orientation. Unmodelled sources of error will affect the other sensor outputs. Tradeoffs are a function of the Kalman filter configuration and are subject to change.

Separate tables are presented for each of the sensor combination options provided by the Fusion Library. Common test conditions and setups are used for all options.

NOTE: Results that follow are simulated build 4.17 using basic sensor models which include noise effects, but ignore nonlinearity and other factors. See Simulation Environments for details. Version 7.00 dynamic performance is expected to be improved from the 4.17 plots shown here.

4.8.2 Gyro Offset Step Response

Gyro Offset Step Response plots are shown in Figure 17 and Figure 18 for 9-axis and 6- axis Accelerometer+Gyro algorithms, respectively and provide an indication of the responsiveness of the system to changes in gyro offsets. The simulation artificially introduces a step in the modeled gyro offset. However, under normal circumstances, gyro offsets change very slowly. The only times that a change such as this would likely occur are:

- at powerup
- if the gyro has an autonomous offset cancellation circuit (which NXP recommends be turned OFF).

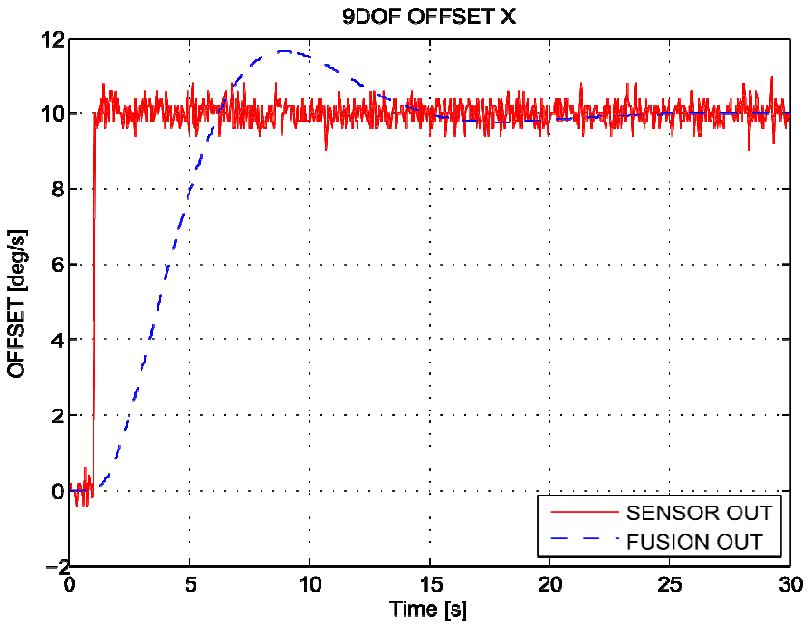


Figure 14. Gyro Offset Step Response for 9-Axis algorithm

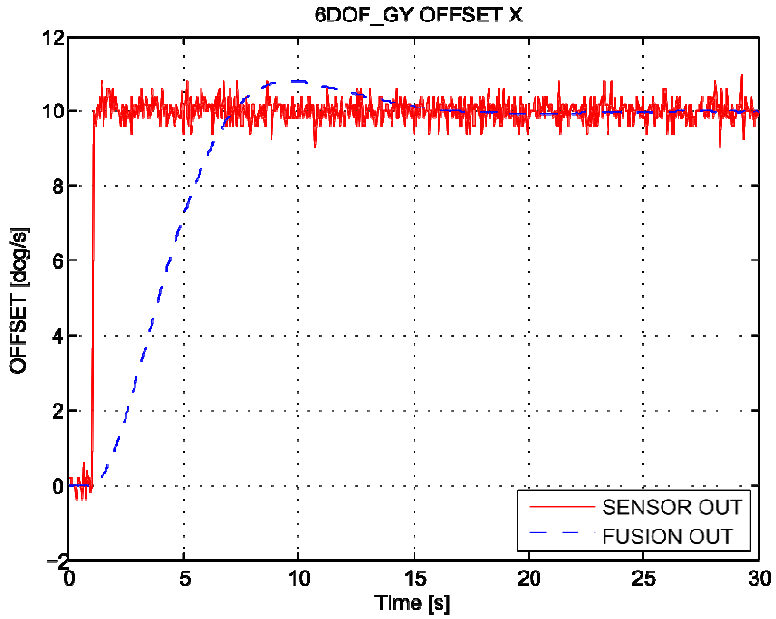


Figure 15. Gyro Offset Step Response for 6-Axis Accelerometer+Gyro algorithm

4.8.3 Error in Computed Linear Acceleration

Estimation of linear acceleration in the 9-axis algorithm is optimized for short duration accelerations lasting less than 1 second. Continuous accelerations are inconsistent with modeled dynamics, and the acceleration estimation error increases. The acceleration error starts leaking into the estimates of orientation resulting in errors in gyro offset and magnetic disturbance estimates. Figure 19, Figure 20 and Figure 21 present the true and 9-axis algorithm calculated linear acceleration for accelerations lasting 0.5 s, 1 s and 5 s respectively.

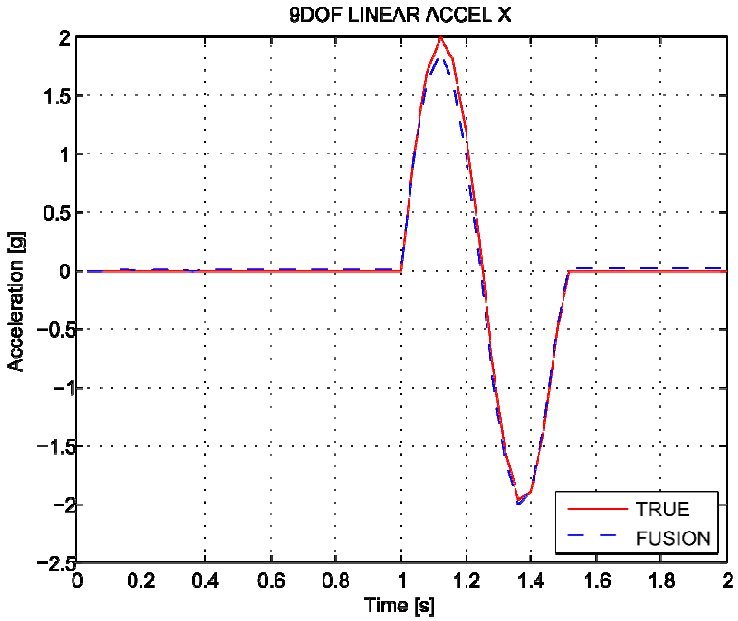


Figure 16. True and calculated linear acceleration, acceleration time 0.5 s

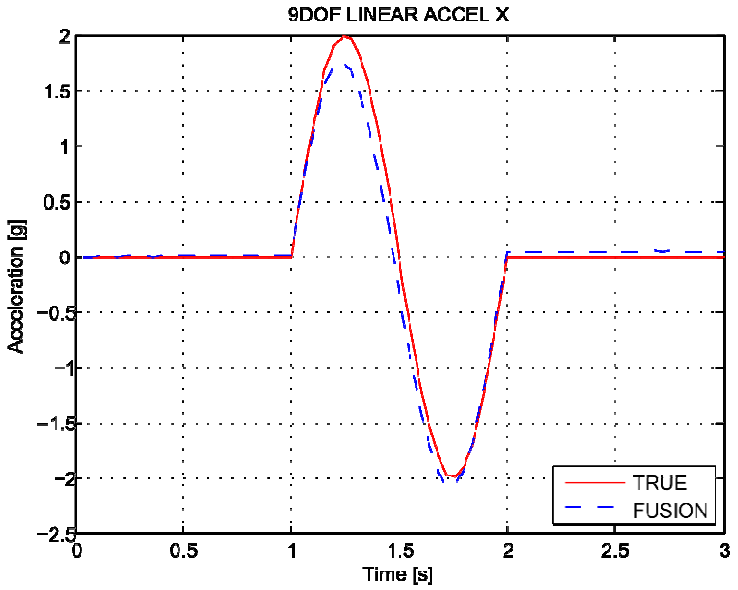


Figure 20. True and calculated linear acceleration, acceleration time 1 s

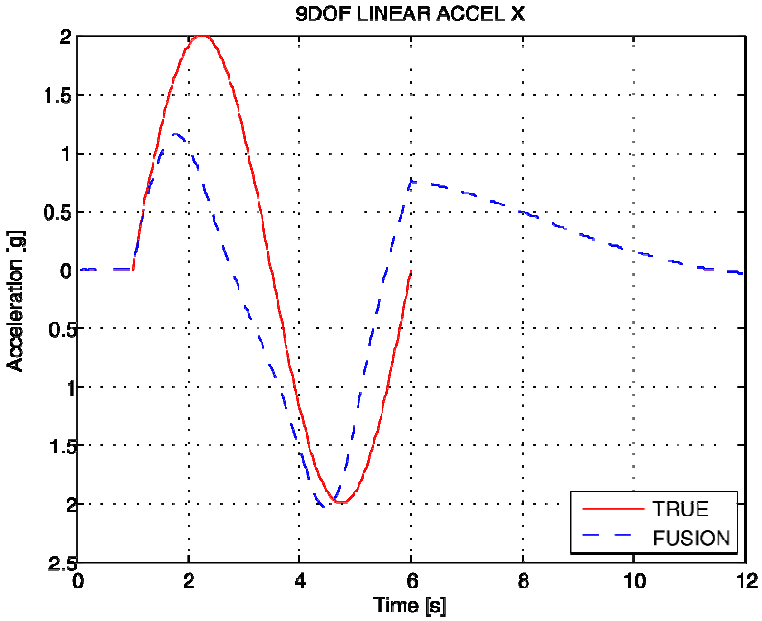


Figure 21. True and calculated linear acceleration, acceleration time 5 s

Similar dependence on the magnetic pulse duration time was noticed in the Orientation Magnetic Immunity tests. A slow-moving magnet introduces a longer lasting magnetic field disturbance pulse than a fast-moving magnet. The longer the magnetic disturbance pulse lasts, the greater the orientation error it causes on the output of the fusion algorithm.

4.8.4 **Limitations Imposed via Sensor Choice/Configuration**

Table 14 represents the sensor configuration used to determine parameters specified in this datasheet.

Table 14. Sensor/Configuration Imposed Limitations

Characteristic	Symbol	Min	Max	Unit
Maximum Angular Rate	AR_{max}	–2000	+2000	dps
Linear Acceleration	LA_{max}	–4	+4	*g*
Magnetic Field	B_{max}	–1200	+1200	µT

It is believed that the fusion algorithms themselves have no upper limits on any of the parameters above. These are strictly limits associated with the specific physical sensors used to characterize the library.

4.8.5 9-Axis Parametrics

Table 15 provides guidance for the 9-axis (accelerometer/magnetometer/gyro) indirect Kalman filter implementation.

NOTE: All parametrics provided in the following table are based on build 4.17 simulations. Version 7.00 simulations are still pending.

Table 15. 9-axis Sensor Fusion Performance Metrics

Characteristic	Symbol	Conditions(s)	Min	Typ	Max	Unit
Orientation static drift	O_{SD}	See Orientation Static Drift	—	0.05	—	degrees
Orientation static noise	O_{SN}	See Orientation Static Noise	—	1.21	—	degrees RMS
Orientation dynamic drift	O_{DD}	See Orientation Dynamic Drift	—	0.17	—	degrees
Max angular Rate	AR_{MAX}	See Maximum Angular Rate	—	1600[Error! Reference source not found.]	—	dps
Orientation response delay	O_{RD}	See Orientation Response Delay	—	<1	—	[2]fusion time interval
Gyro offset step response	T_{GOSR}	See Gyro Offset Step Response	—	3.76	—	seconds
Error in computed linear acceleration	LAE	See Error in Computed Linear Acceleration	—	0.243	—	*g*
Compass heading linearity[3]	CH_l	See Compass Heading Linearity and Accuracy	—	1.02	—	degrees
Compass heading accuracy	CH_{acc}		—	1.37	—	degrees
Orientation magnetic immunity - static device	O_{mis}	See Orientation Magnetic Immunity (Static Device)	—	9.36	—	degrees
Orientation magnetic immunity - moving device	O_{mim}	See Orientation Magnetic Immunity (Moving Device)	—	8.98	—	degrees

1. The Sensor Fusion algorithm has no intrinsic limitation; this was the maximum value supported by the gryo in NXP testing.
2. Number of output sampling periods where one period = 40 ms
3. Linear sensors, which yields very good compass heading values were assumed. However experience shows that +/5 degrees are more realistic values.

4.8.6 6-axis Accelerometer + Magnetometer Parametrics

NOTE: All parametrics provided in the following table are based on build 4.17 simulations. Version 7.00 is believed to have similar performance.

Table 16. 6-axis Sensor Fusion Accel + Mag Performance Metrics

Characteristic	Symbol	Conditions(s)	Min	Typ	Max	Unit
Orientation static drift	O_{SD}	See Orientation Static Drift	—	0.0018	—	degrees
Orientation static noise	O_{SN}	See Orientation Static Noise	—	1.06	—	degrees RMS
Orientation dynamic drift	O_{DD}	See Orientation Dynamic Drift	—	1.16	—	degrees
Orientation response delay	O_{RD}	See Orientation Response Delay	—	<10	—	sample periods[Error! Reference source not found.]
[2]Compass heading linearity	CH_l	See Compass Heading Linearity and Accuracy	—	1.20	—	degrees
Compass heading accuracy[2]	CH_{acc}		—	1.58	—	degrees

1. Number of output sampling periods where one period = 40 ms
2. Linear sensors, which yields very good compass heading values were assumed. However, experience shows that ±5 degrees are more realistic values.

4.8.7 6-axis Accelerometer + Gyro Parametrics

NOTE: All parametrics provided in the following table are based on build 4.17 simulations. Version 7.00 is believed to have similar static and improved dynamic performance.

Table 17. 6-axis Sensor Fusion Gyro + Accel Performance Metrics

Characteristic	Symbol	Conditions(s)	Min	Typ	Max	Unit
Orientation static drift	O_{SD}	See Orientation Static Drift	—	19.07	—	degrees
Orientation static noise	O_{SN}	See Orientation Static Noise	—	0.068	—	degrees RMS
Orientation dynamic drift	O_{DD}	See Orientation Dynamic Drift	—	2.04	—	degrees
Max angular Rate	AR_{MAX}	See Maximum Angular Rate	—	1440	—	dps
Orientation response delay	O_{RD}	See Orientation Response Delay[Error! Reference source not found.]	—	<1	—	sample periods[2]
Gyro offset step response	TBD	See Gyro Offset Step Response	—	3.76	—	seconds

1. Output sampling period = 40 ms
2. Number of output sampling periods where one period = 40 ms

4.8.8 3-axis Accelerometer Only Parametrics

NOTE: All parametrics provided in the following table are based on build 4.17 simulations. Version 7.00 should be similar.

Table 18. 3-axis Accelerometer Performance Metrics

Characteristics	Symbol	Conditions	Min	Typ	Max	Units
Tilt Error RMS	TAE	Note 1	—	0.082	—	degrees
Orientation response delay	O_{RD}	See Orientation Response Delay[2]	—	< 5	—	output sample period

1. RMS of accelerometer tilt angle error is calculated using the simulated sensor noise RMS values along each of the three axes and the following formula (given for tilt error from the z-axis)

   $$\mathrm{TAE} = \arctan\left(\frac{\sqrt{\mathrm{NRMS^2}_x + \mathrm{NRMS^2}_y}}{1\mathrm{g} - \mathrm{NRMS}_z}\right),$$

   where NRMSx, NRMSy, NRMSz - RMS values of accelerometer noise for X, Y, Z axes in g units.
2. Number of output sampling periods where one period = 40 ms

5 Test Descriptions

Each of the following sub-sections defines the specification intent and the sample procedure for the specifications listed in Mechanical and Electrical Specifications. Procedures may evolve in future drafts of this document in order to better service the specification intents.

5.1 MCU Current

5.1.1 Intent

This is the average current consumption of the MCU executing the core Fusion routines. This is obviously specific to the particular MCUs listed. This metric must be associated with a specific hardware configuration similar to those defined earlier in this document. Freedom Development Platform are powered via the OpenSDA USB port for the results specified.

5.1.2 Procedure

1. This procedure uses modified versions of the standard demo build:
   a. Measure (using the Sensor Fusion Toolbox) $\mathrm{sysTick}_{\mathrm{fusion}}$ which only includes time spent in the *core fusion routines*. It does *not* include:
      1) calls to magnetic calibration
      2) reading sensor data
      3) applying hardware abstraction layer
      4) RTOS
      5) Communications overhead
2. The relative ratio of time spent in the core fusion routines is

   fusion_rate X $\mathrm{sysTick}_{\mathrm{fusion}}$ / MCU clock rate

3. I_{dd} Current into the MCU is measured via power jumper on the board using a simple DVM[3]
4. Fusion I_{dd} = MCU I_{dd} X ratio computed above
5. Sample size = 1 board

For devices with floating point units, this may yield a somewhat optimistic number, as the computation makes the assumption that all MCU instructions consume the same amount of power. In fact, we can expect floating point instructions to consume a bit more.

5.2 Flash and RAM Required

5.2.1 Intent

These parameters are total RAM and flash memory required to implement and execute the Fusion Library in a bare metal project.

This includes space for code storage, static and dynamic (stack) variables. This metric must be associated with a specific hardware configuration similar to those defined in earlier in this document. It is also a function of the optimization level selected in the compiler.

5.2.2 Procedure

The projects for the binaries were created using IAR. Values for readonly code and data memory (flash) and readwrite data memory (RAM) were read directly from the c.map file generated by the toolset.

Refer to Computation Metrics for statistics gathered using the Version 7.00 build.

5.3 Fusion Loop Execution Time

This is simply CPU cycle time X $sysTick_{fusion}$ as measured in Section 5.1.

5.4 Compass Heading Linearity and Accuracy

5.4.1 Intent

Linearity is defined as the deviation of measured data from a least squares straight line approximation of that data.

Absolute accuracy is defined as the maximum difference between measured and ideal values.

These two concepts are illustrated in Figure 22.

This metric must be stated specifically for a hardware configuration similar to those defined earlier in this document.

[3] Consult the user guide / schematic for your Freedom board to determine the specific jumper number applicable to that board.

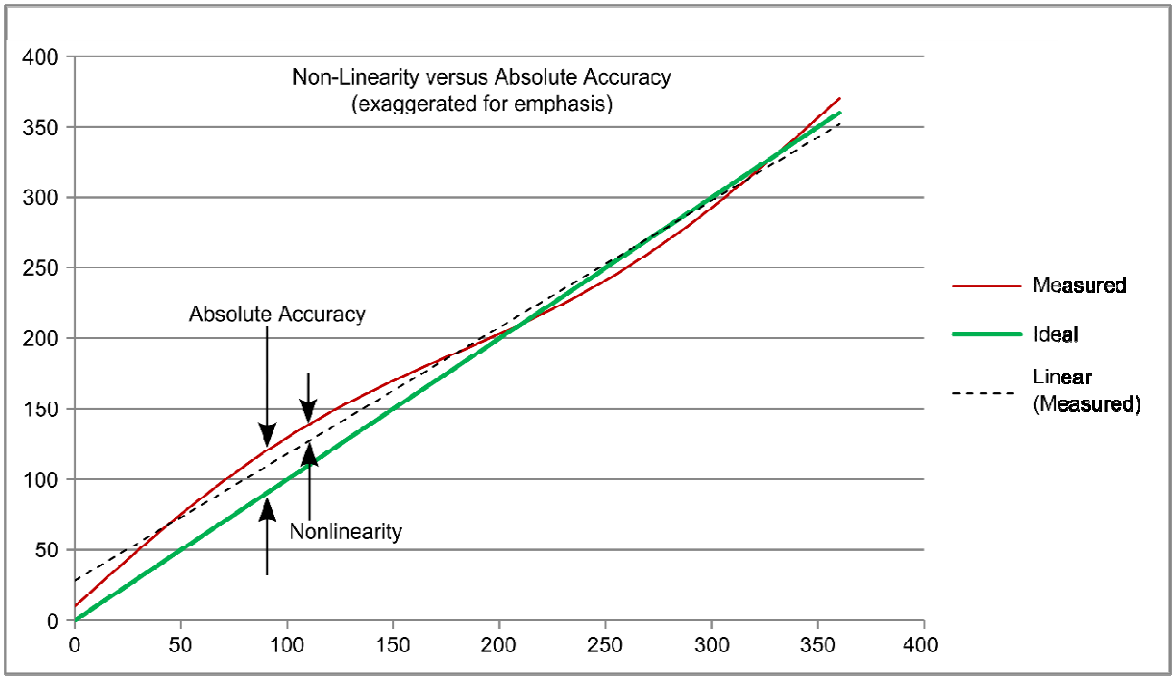


Figure 22. Nonlinearity vs. Absolute Accuracy

5.4.2 Procedure

Heading Linearity

Setup as defined in Matlab Stimulus/Expected Response + Hardware-based Fusion. Matlab is used to post-process data.

1. Precondition the DUT by executing magnetic calibration trajectory leaving DUT in AZ = + 1 orientation (See Matlab Stimulus/Expected Response + Hardware-based Fusion)
2. Rotate DUT around the z-axis from 0 to 360 degrees with an increment step of 10 degrees as follows
3. Program rate table with desired steps and maximum velocity
   a. Rotate tabletop by 10 degrees over 0.5 second period
   b. Pause 0.5 seconds
   c. Read orientation
   d. Repeat steps a-c above until 360 degrees is reached.

Absolute Heading Accuracy

These numbers are based on simulations that are idealized models. The real-world values will tend to be approximately 5 degrees.

5.5 Orientation Static Drift

5.5.1 Intent

The maximum change in orientation observed for a DUT remaining motionless for 100 seconds.

5.5.2 **Procedure**

Setup as defined in *Matlab Stimulus/Expected Response + Hardware-based Fusion*. Matlab is used to post-process data.

This metric must be associated with a specific hardware configuration similar to those defined in earlier in this document. Matlab is used to record and post- process data.

1. Precondition the DUT by executing magnetic calibration trajectory leaving DUT in a given orientation.
2. While the DUT remains in given orientation collect orientation samples for 120 seconds.
3. Plot and process the last 100 s of the test. Orientation static drift is the maximum angle change in the measured rotation vectors.
4. Repeat steps 1-3 for all 6 major orientations of the development board as shown in Figure 23.

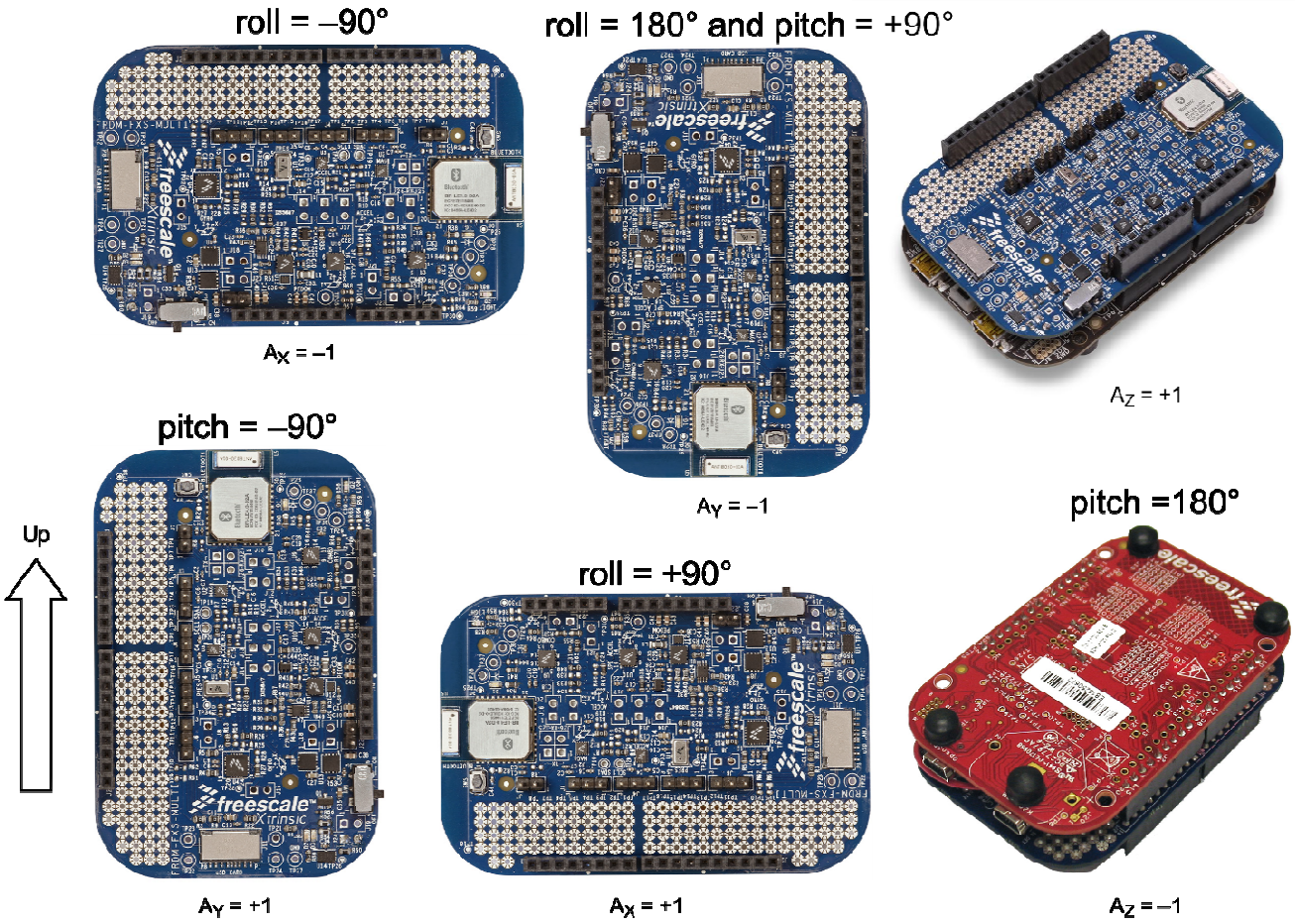


Figure 17. Major orientations relative to gravity

5.6 Orientation Static Noise

5.6.1 **Intent**

RMS noise as a function of orientation for a motionless DUT.

5.6.2 Procedure

This can be based upon data gathered for the orientation static drift test. The classic standard deviation equation is used to compute the metric/value.

5.7 Orientation Dynamic Drift

5.7.1 Intent

Measure of the ability of fusion code to return to a known orientation after movement.

5.7.2 Procedure

Setup as defined in Matlab Stimulus/Expected Response + Hardware-based Fusion and in Figure 9. Matlab is used to post-process data.

1. Precondition the DUT by executing magnetic calibration trajectory leaving DUT in a given orientation.
2. Keep DUT in a given orientation motionless for 2 seconds.
3. Rotate DUT at 0.5 revolution/sec for 10 seconds around vertical axis.
4. Keep DUT in its last orientation motionless for 2 seconds.
5. Determine the initial orientation by averaging orientation samples from the initial motionless period and the final orientation by averaging the samples from the final motionless period. The difference between the two orientations is the orientation dynamic drift.
6. Repeat steps 1-5 for all six major orientations of the development board.

5.8 Maximum Angular Rate

5.8.1 Intent

This test determines the maximum rotation rate for which the filter is able to correctly track orientation.

5.8.2 Procedure

Setup as defined in Matlab Stimulus/Expected Response + Hardware-based Fusion. Matlab is used to post-process data.

1. Rotate DUT around X axis five times starting at rotation rate 360 dps.
2. Plot the ideal simulated orientation together with the orientation samples calculated by Kalman filter fusion algorithms: 9DOF and 6DOF Accelerometer + Gyro and notice if fusion data tracks the ideal orientation.
3. Increase the rotation rate by 360 dps and repeat steps 1 and 2.
4. The highest angular rate at which the fusion algorithm orientation still tracks the ideal simulated orientation is the maximum angular rate.

5.9 Orientation Response Delay

5.9.1 Waveform Definitions

90° Input Change in Orientation

The 90° input orientation change waveform is defined as:

- From any starting orientation
- Through the center of mass of the accelerometer
- A change in one of roll, pitch or yaw (global frame)
- Orientation change = ±90°
- Orientation change is linear in time
- Transition period = 1.0 seconds
- Propagation delays are measured from the 1/2 point (±45°)

Output Orientation Changes

When measuring changes in orientation, there are many ways to get from orientation A to orientation B. It is assumed that output orientation changes should track input orientation changes. The Fusion Library will inherently output those values. However, to avoid any ambiguity when measuring propagation delays, the 50% point in computed orientations changes is defined as shown in Figure 24.

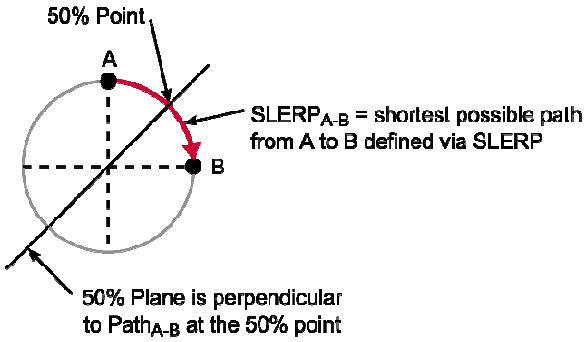


Figure 18. Propagation delays are measured at the plane that bisects and is perpendicular to SLERP$_{A–B}$

5.9.2 Intent

Orientation response delay is measured from change in physical orientation to the related change in fusion orientation output.

5.9.3 Procedure

Using 90° input orientation change as defined in 90° Input Change in Orientation, the response delay is measured from the 50% point on the input waveform to the 50% point of the output waveform as defined in Output Orientation Changes.

5.10 Orientation Magnetic Immunity (Static Device)

5.10.1 Intent

The test response of the stationary DUT to momentary changes in the local magnetic field is used to measure the orientation magnetic immunity. Because the device is stationary, the accelerometer and gyroscope readings remain relatively constant, changing only due to sensor noise.

5.10.2 Procedure

Setup as defined in *Matlab Stimulus/Expected Response + Hardware-based Fusion*. Matlab is used to post-process data.

1. Starting point = device stationary, fusion outputs stable
2. 100 µT magnet moving at 0.25 m/s
3. Closest approach to magnetic sensor = 5 cm
4. Test to be simulated using environment specified in Matlab Stimulus/Expected Response + Hardware-based Fusion. Magnetic interference may be modeled as a time varying field which is consistent with the description above.

5.11 Orientation Magnetic Immunity (Moving Device)

5.11.1 Intent

This test measures the immunity of a linearly, with no rotation, moving DUT to a 100 µT magnet change in the magnetic field. The outputs of all acceleration and magnetic sensors change during this test. Gyro outputs should be constant throughout, with any changes attributed to noise only.

5.11.2 Procedure

1. Starting point = device stationary, fusion outputs stable.
2. Use a 100 µT magnet.
3. The DUT moves by magnet at 0.25 m/s with the closest approach to magnet=5cm.
4. Test to be simulated using environment specified in Matlab Stimulus/Expected Response + Hardware-based Fusion. Magnetic interference may be modeled as a location varying field which is consistent with the description above.

5.12 Error in Computed Gyro Bias

5.12.1 Intent

Measure how well the fusion library tracks slowly varying gyro bias.

5.12.2 Procedure

Test to be simulated using environment specified in Matlab Stimulus/Expected Response + Hardware-based Fusion.

1. Starting point = device stationary, fusion outputs stable
2. Add 10 dps gyro offset as step function
3. Run test until computed gyro offset stabilizes
4. Note computed/actual for final value
5. Note response time

5.13 Gyro Offset Step Response

5.13.1 Intent

Measure how well the fusion library tracks varying gyro offset.

5.13.2 Procedure

Setup as defined in Matlab Stimulus/Expected Response + Hardware-based Fusion. Matlab is used to post-process data.

1. Starting point = DUT is stationary and fusion outputs are stable.
2. Add offset of 10 dps to gyro X read data.
3. Run test until computed gyro offset stabilizes.
4. Plot the gyro read data step function and the response fusion algorithm estimated offset. Record response time as time between 50% step change in gyro X read data and 50% change in fusion algorithm estimated gyro X offset.

5.14 Error in Computed Linear Acceleration

5.14.1 Intent

Measure how well the fusion library tracks linear acceleration.

5.14.2 Procedure

Setup as defined in Matlab Stimulus/Expected Response + Hardware-based Fusion. Matlab is used to post-process data.

1. Starting point = DUT is stationary and fusion outputs are stable.
2. Subject DUT to acceleration A*sin(2*pi*f*t), where: A = 2 g, f = 1 Hz, t = zero to one second (1 period), pi = 3.141592654
3. Plot the ideal DUT acceleration and fusion algorithm computed acceleration. Determine the maximum error between the two accelerations.

6 Revision history for NSFK_DS

Rev. No.	Date	Description
0	12 Nov 2013	• ROUGH DRAFT ONLY - PRE-REVIEW
0.1	22 Nov 2013	• Preliminary draft includes updates from 1st review.
0.2	Feb 2014	• Initial public release.
0.3	Apr 2014	• Updated for licensing, software updates and board support changes.
0.4	May 2014	• Updated for software updates, additional (FRDM-K64F) board support changes and electrical specs and computation metrics.
0.5	Sept 2014	• Updated Fusion Performance Metrics section by adding four new figures and tables. • adjusted selected parametric values • altered several Test Description procedures.
0.6	Sept 2014	• Separated out Computational Metrics section and various minor markups • Changed Feature - License, option text. • Feature Comparison Based on License Option, Added KDS to Product Deliverables row • Moved sections 4.1, 4.1.1 & 4.1.2 and merged in 4.11 • Added xrefs from Electrical Specs tables to appropriate Test Description sections • Adjusted Performance Metric tables, symbols and units in some cases
0.7	Sept 2015	• Updated for build 5.00 of the sensor fusion library. This version has completely redesigned 6 and 9-axis Kalman filters. • Updated all computation metrics. • Removed outdated fusion time measurements. • Added K22F for KDS. Added additional fusion options. • Changed document name from XSFLK_DS to NSFK_DS. • Noted that MULTI-B boards have been replaced with MULT2-B boards.
0.8	July 2016	• Document updated to reflect change from Freescale to NXP. • Compute parameters and IDD measurements updated to match sensor fusion Version 7.00 .

How to Reach Us:

Home Page:
NXP.com

Web Support:
NXP.com/support

Document Number: NSFK_DS
Revision Rev. 0.8, 7/2016

freescale™