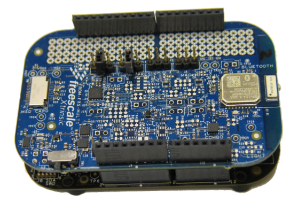
Freescale Sensor Fusion Library for Kinetis

XSFLK_DS



Freescale Freedom Sensors Toolbox Development Platform FRDM-KL25Z/FRDM-FXS-MULTI

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

Features

- Supports:
  - Accelerometer only (roll, pitch and tilt)
  - Accelerometer plus magnetometer (eCompass)
  - Accelerometer plus gyro (gaming)
  - Accelerometer plus magnetometer plus gyroscope sensors
- Includes Freescale's award-winning magnetic compensation software
  - Provides geomagnetic field strength, hard- and soft-iron corrections, and quality-of-fit indication
- Very low power consumption (<5 mA on Kinetis Cortex M0+ devices at 25 Hz fusion rate/200 Hz sensor 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 Freescale Sensor Fusion Toolbox for Android and Windows
- Supported by Freescale CodeWarrior, Kinetis Design Studio and Processor Expert tools
- Out-of-the box support for the following Freedom Development Platforms with FRDM-FXS-MULTI family of sensor boards
  - FRDM-KL25Z
  - FRDM-KL26Z
  - FRDM-KL46Z
  - FRDM-K20D50M
  - FRDM-K64F

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

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

Feature	Accel only	Accel + gyro	Accel + mag	Accel + mag + gyro
Filter type	Low pass	Indirect Kalman	Low pass[1]	Indirect Kalman
Roll / Pitch / Tilt in degrees	Yes	Yes	Yes	Yes
Yaw in degrees	No	No	Yes	Yes
Angular rate[2] in degrees/second	virtual 2 axis[3]	Yes	virtual 3 axis	Yes
Compass heading (magnetic north) in degrees	No	No	Yes	Yes
Quarternion and rotation vector	Yes	Yes	Yes	Yes
Rotation matrix	Yes	Yes	Yes	Yes
Linear acceleration separate from gravity	No	Yes	No	Yes
NED (North-East-Down) frame of reference	Yes [4]	Yes [4]	Yes	Yes
ENU (Windows 8 variant) frame of reference	Yes[4]	Yes[4]	Yes	Yes
ENU (Android variant) frame of reference	Yes[4]	Yes[4]	Yes	Yes
Magnetic calibration included	No	No	Yes	Yes
Gyro offset calibration included	N/A	Yes	N/A	Yes
FRDM-KL25Z board support	Yes	Yes	Yes	Yes
FRDM-KL46Z board support	Yes	Yes	Yes	Yes
FRDM-K20D50M board support	Yes	Yes	Yes	Yes
FRDM-KL26Z board support	Yes	Yes	Yes	Yes
FRDM-K64F board support	Yes	Yes	Yes	Yes

1. More precisely: a non-linear modified exponential low pass quaternion SLERP filter.
2. Angular rate for 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 .

Feature Options

Feature	Options
License	Free to use with Freescale components, see the license file for details.
CPU selection	MKL25Z128VLK4 MKL26Z128VLH4 MK20DX128VLH5 MKL46Z256VMC4 MK64FN1M0VLL12
Board customizable	Yes[1, 2, 3]
Sensor sample rate	Programmable
Fusion rate	Programmable
Frame of Reference	Programmable
Algorithms Executing	Programmable

Table continues on the next page...

Feature Options (continued)

Feature	Options
Sleep mode enabled between samples/calculations	Programmable
RTOS	MQX-Lite
Code flexibility	Only Kalman and MagCal libraries are precompiled. All other files are supplied in source form, and can be modified. Full source code is provided.
Access to Processor Expert	Yes
Product Deliverables	• This datasheet • Installer, which includes documentation and the appropriate CodeWarrior project(s) • Software user guide • Kinetis Design Studio (KDS)

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 expected to be very straightforward, as the library is written in standard C.
2. FRDM-KL25Z, FRDM-KL26Z, FRDM-KL46Z, FRDM-K20D50M and FRDM-K64F are supported out-of-the box and may be used as templates for other boards.
3. The sensor fusion library was written assuming the MQXLite RTOS, but should be easily adaptable to other operating systems. See the "Freescale Sensor Fusion for Kinetis User Guide" for additional details.

The Freescale Sensor Fusion Toolbox includes the FRDM_KL46Z special project that is designed to demonstrate the accelerometer and magnetometer sensor fusion algorithms using the FXOS8700 combination sensor present on the FRDM-KL46Z Freedom Development Platform. No sensor shield board is needed in this configuration and this results in a very low-cost demonstration platform. The FRDM_KL46Z project is compatible with the Sensor Fusion Toolbox for Windows over USB connection but the project can also be used as a stand-alone since the compass heading is displayed directly on the FRDM-KL46Z LCD display. The Sensor Fusion algorithms that require a gyroscope sensor are not supported in this instance since there is no gyroscope sensor on the FRDM-KL46Z Freedom Development Platform, so they can be removed.

The sensor sampling, sensor fusion and magnetic algorithms run as a single task at 25Hz. On power up, the green LED flashes slowly to indicate that the software is executing but that there is no magnetic calibration solution. The green LED then flashes rapidly to signal that the compass heading display is accurate and that a valid magnetic calibration has been obtained after rotating the FRDM-KL46Z board.

Table of Contents

1 Functional Overview

1.1 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 Freescale 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 *Freescale Sensor Fusion for Kinetis MCUs User Guide* which is part of the *Sensor Fusion Library* installation. The Fusion Library is available in source code form. The fusion library is designed to sit on top of board abstractions provided by Processor Expert and MQXLite. Most of the C source and header files in the Sources directory of the project templates are board and MCU independent. If a particular board does not bring out the GPIO, I2C or UART needed, the board will not be compatible with the software and therefore will not be supported by Processor Expert.

1.2 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.

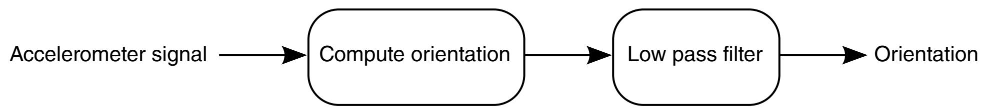


Figure 1. Accelerometer only block diagram

1.3 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.

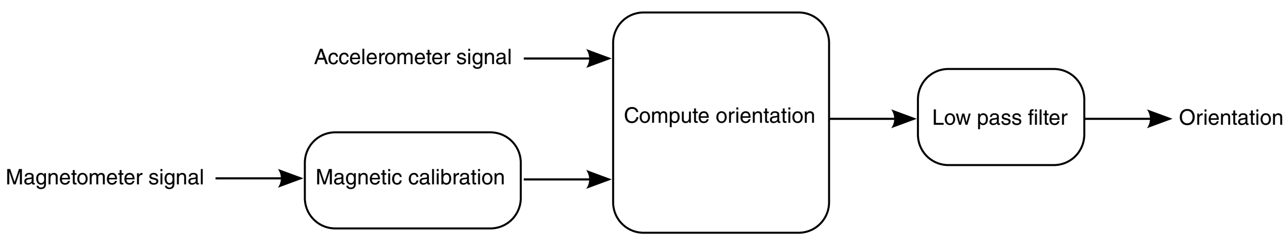


Figure 2. Accelerometer plus magnetometer block diagram

1.4 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).

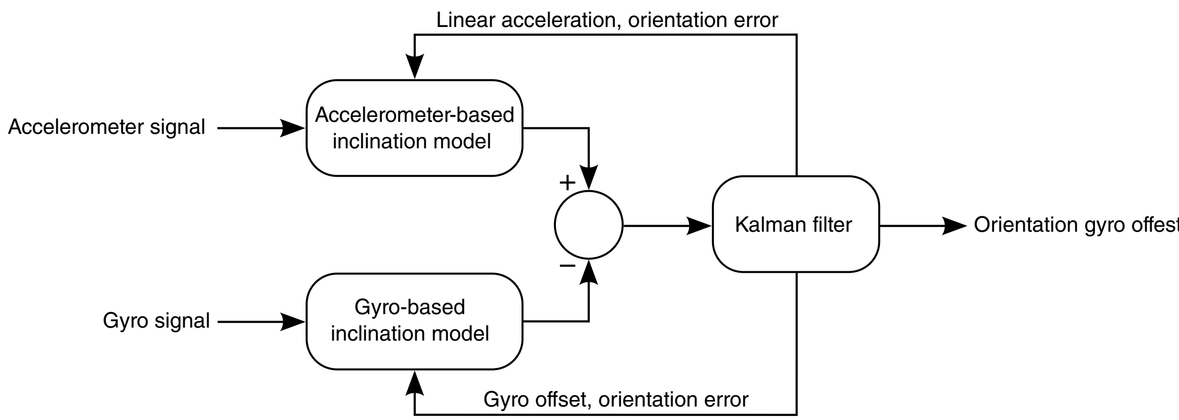


Figure 3. Accelerometer Plus Gyroscope Block Diagram

1.5 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.

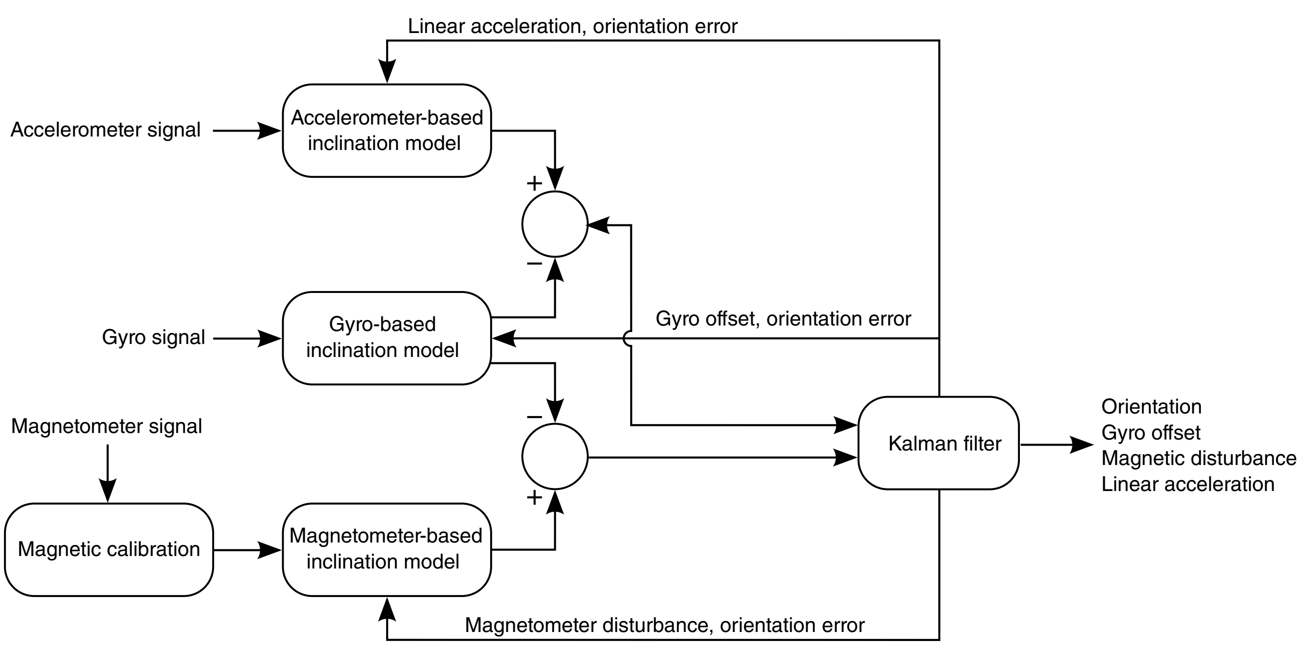


Figure 4. Accelerometer Plus Magnetometer Plus Gyroscope Block Diagram

2 Additional Support

Freescale Sensor Fusion Toolbox provides support for both Android and Windows operating systems. See Table 2 for the differences between the two implementations.

Table 2. Freescale Sensor Fusion Toolbox Features by Platform

Feature	Android	PC
Bluetooth wireless link	✔	Requires BT on PC (built-in or dongle)
Ethernet wireless link	On WiGo board only	-
UART over USB	-	✔[1]
OS requirements	>=Android 3.0	>=Windows 7.0
Support for native sensors	✔	-
Device View	✔	✔
Panorama View	✔	-
Statistics View	✔	-
Canvas View	✔	-
Orientation XY Plots	-	✔
Inertial XY Plots	-	✔
Magnetics	-	✔
Kalman	-	✔
Altimeter XY Plots	-	✔
Data Logging	✔	✔
Integrated documentation	✔	✔
Availability	Google Play	Freescale website
Price	Free	Free

1. FRDM_K64F and FRDM_K20D50M projects require a Processor Expert configuration change to run in wired mode.

2.1 Freescale Sensor Fusion Toolbox for Android

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

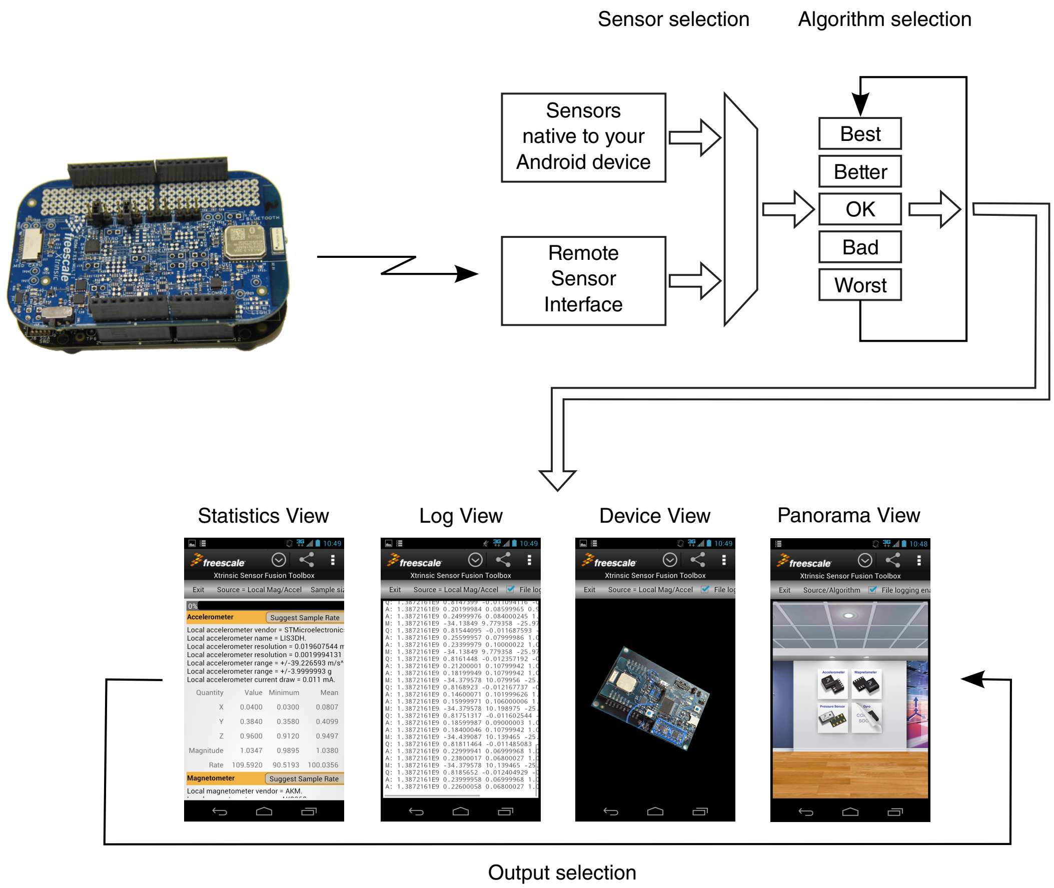
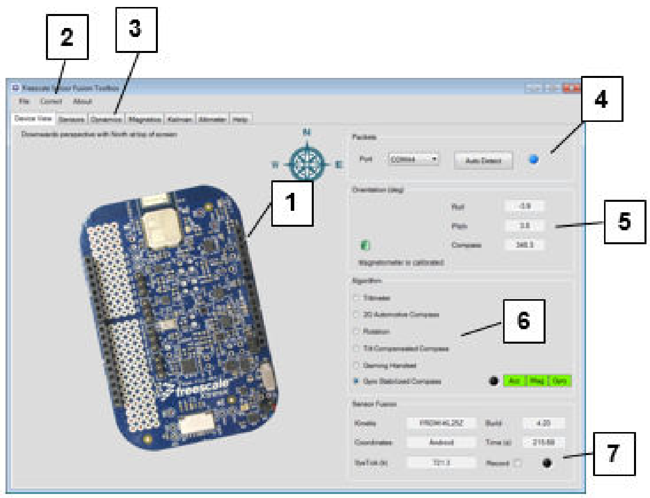


Figure 5. Freescale Sensor Fusion Toolbox for Android Basic Functions

2.2 Freescale Sensor Fusion Toolbox for Windows

The Sensor Fusion Toolbox includes an equivalent version of the software for Windows. For download and training links, see *http:/www.freescale.com/sensorfusion*.

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

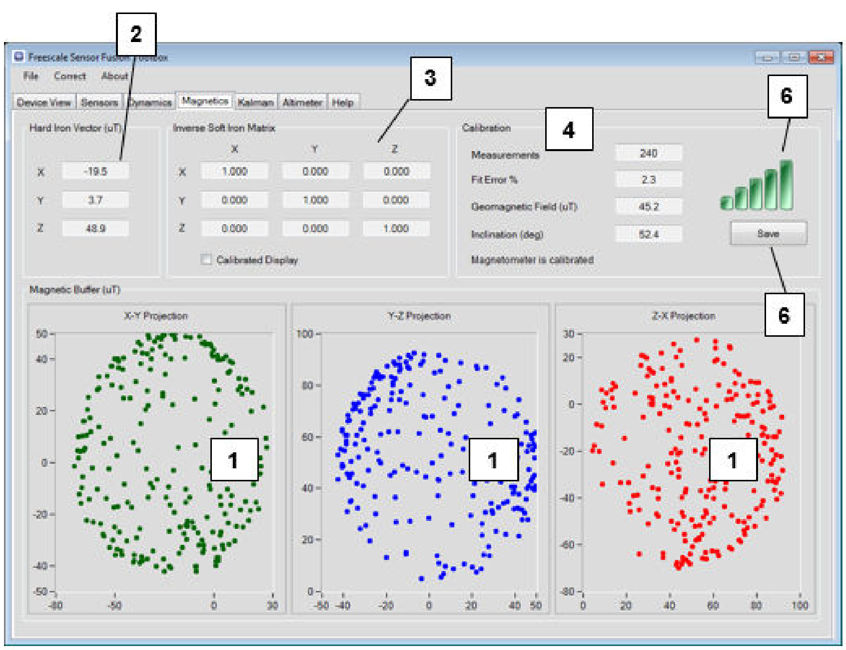


Figures are from 28 August 2014 build of the application. Appearance may vary for other versions.

1. Rotating 3D PCB display
2. Image align function
3. Navigation Tabs for:
    - Sensors Data Tab
    - Dynamics Tab
    - Magnetics
    - Kalman
    - Altimeter
    - Help
4. Packet information
    - choice of PC comm port
    - packet activity indicator
    - # of packet errors
5. Roll/Pitch/Yaw & MagCal status
6. Choice of sensor set & algorithm
7. Sensor board run time and build parameters, Data logging on/off

This is the most intuitive way to confirm that your sensor fusion is working properly.

Figure 6. PC Version - Device View Tab



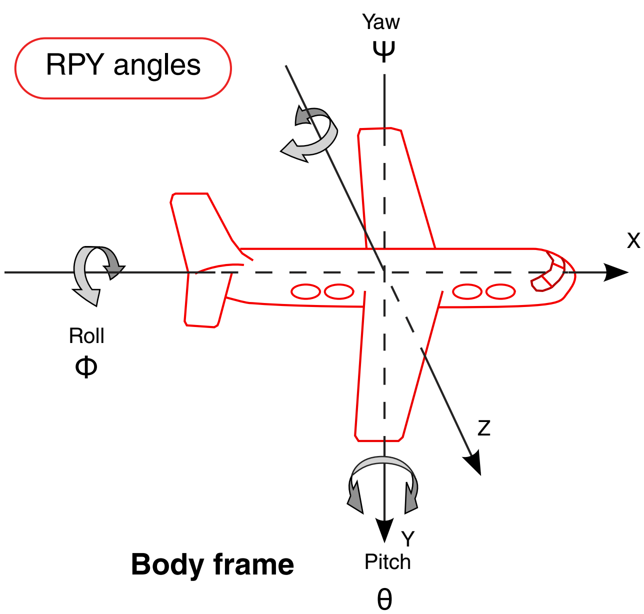
1. 2D representation of the data point "cloud" used for hard/soft iron compensation
2. Computed hard iron vector
3. Soft iron matrix
4. Statistics
5. Calibration status light
6. Save to text file

You can use this display to view how the magnetic constellation evolves over time in response to changing magnetic environments.

Figure 7. PC Version - Magnetics Tab

2.3 Terms and Acronyms

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
Kinetis	Freescale 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

Table continues on the next page...

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

2.4 References

1. "Orientation Representations: Part 1" at https://community.freescale.com/community/the-embedded-beat/blog/2012/ 10/29/orientation-representations-part-1
2. "Orientation Representations: Part 2" at https://community.freescale.com/community/the-embedded-beat/blog/2013/ 01/22/orientation-representations-part-2
3. "Hard and soft iron magnetic compensation explained" at https://community.freescale.com/community/the-embeddedbeat/ blog/2011/03/14/hard-and-soft-iron-magnetic-compensation-explained
4. CodeWarrior Integrated Development Environment Software at freescale.com/codewarrior
5. Kinetis Design Studio Integrated Development Environment
6. "Euler Angles" at http://en.wikipedia.org/wiki/Euler_Angles
7. "Introduction to Random Signals and Applied Kalman Filtering", 3rd edition, by Robert Grover brown and Patrick Y.C. Hwang, John Wiley & Sons, 1997
8. "Quaternions and Rotation Sequences", Jack B. Kuipers, Princeton University Press, 1999
9. Freescale Freedom development platform home page at freescale.com/freedom
10. OpenSDA User's Guide, Freescale Semiconductor, Rev 0.93, 2012-09-18
11. PE micro Open SDA Support
12. PE micro Embedded OSBDM support
13. Matlab computer software by MathWorks—http://www.mathworks.com/products/matlab/

3 Mechanical and Electrical Specifications

3.1 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

NOTE

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 Freescale on specific Freescale development platforms.

3.2 Hardware Platforms Used to Measure Performance

In the following subsections, some parametrics are measured, some represent simulated results. There are multiple methods used to determine parametrics. These make use of hardware platforms for benchmarking purposes, in addition to Matlab.

Kinetis Cortex M0+

This configuration is composed of a Freescale KL25Z, KL46Z or KL26Z Freedom development platform paired with a FXS-MULTI-B Bluetooth-enabled sensor board. Use of Bluetooth allows the board to be rotated freely, untethered by cords or power cables.

Table 3. Cortex M0+ Test Configuration

Component / Parameter	Value
Baseboard	FRDM-KL25Z / FRDM-KL26Z / FRDM-KL-46Z
Sensor Board	FRDM-FXS-MULTI-B
MCU	Kinetis MKL25Z128VLK4 / MKL26Z128VLH4 / MKL46Z256VMC4
CPU	ARM® Cortex M0+
CPU Clock	48 MHz
Bus Clock	24 MHz
Accelerometer	FXOS8700CQ
Magnetometer	
Gyroscope	FXAS21000
Sensor Sampling Rate	200 Hz
Fusion Rate	25 Hz
Magnetic Calibration Rate	Varies

This board combination is battery powered and nominally regulated to 3.3 V for use by the sensors in question.

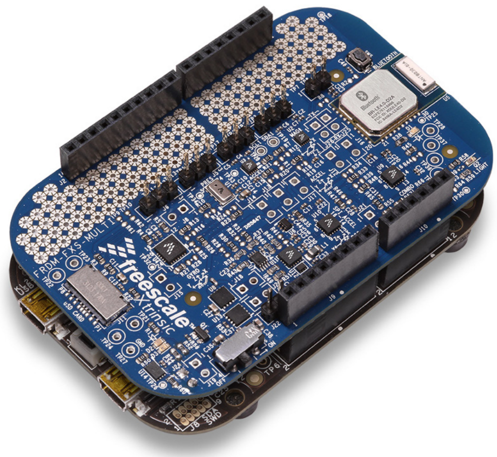
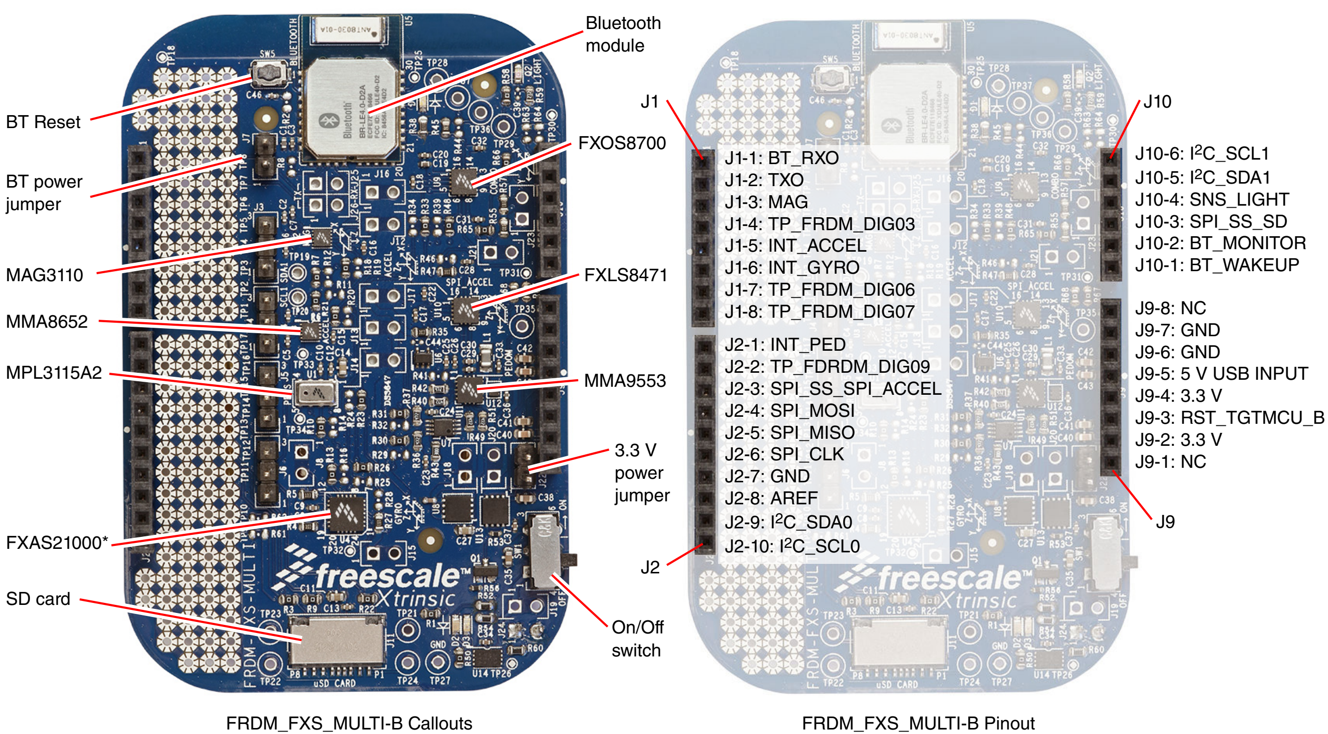


Figure 8. FRDM-KL25Z / FRDM-FXS-MULTI-B sensor fusion prototype platform



* For production status, go to freescale.com/FXAS21000

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

Kinetis Cortex M4

Table 4. Kinetis Cortex M4

Component / Parameter	Value
Baseboard	FRDM-K20D50M
Sensor Board	FRDM-FXS-MULTI-B
MCU	Kinetis MK20DX128VLH5
CPU	ARM® Cortex M4
CPU Clock	48 MHz
Bus Clock	24 MHz
Accelerometer	FXOS8700CQ
Magnetometer	
Gyroscope	FXAS21000
Sensor Sampling Rate	200 Hz
Fusion Rate	25 Hz
Magnetic Calibration Rate	Varies

This board combination is battery powered and nominally regulated to 3.3 V for use by the sensors in question. Physically, the FRDM-K20D50M / FRDM-FXS-MULTI-B board combination is similar to the KL25Z variant shown in Figure 8, with the exception that the bottom board is red.

Kinetis Cortex M4 with FPU

Table 5. Kinetis Cortex M4 with FPU

Component / Parameter	Value
Baseboard	FRDM-K64F
Sensor Board	FRDM-FXS-MULTI-B
MCU	Kinetis MK64FN1M0VLL12
CPU	ARM® Cortex M4 with FPU
CPU Clock	120 MHz
Bus Clock	60 MHz
Accelerometer	FXOS8700CQ
Magnetometer	
Gyroscope	FXAS21000
Sensor Sampling Rate	200 Hz
Fusion Rate	25 Hz
Magnetic Calibration Rate	Varies

The FRDM-K64F / FRDM-FXS-MULTI-B board combination is similar to the KL25Z variant shown in Figure 8, except that the bottom base board is different.

3.3 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.

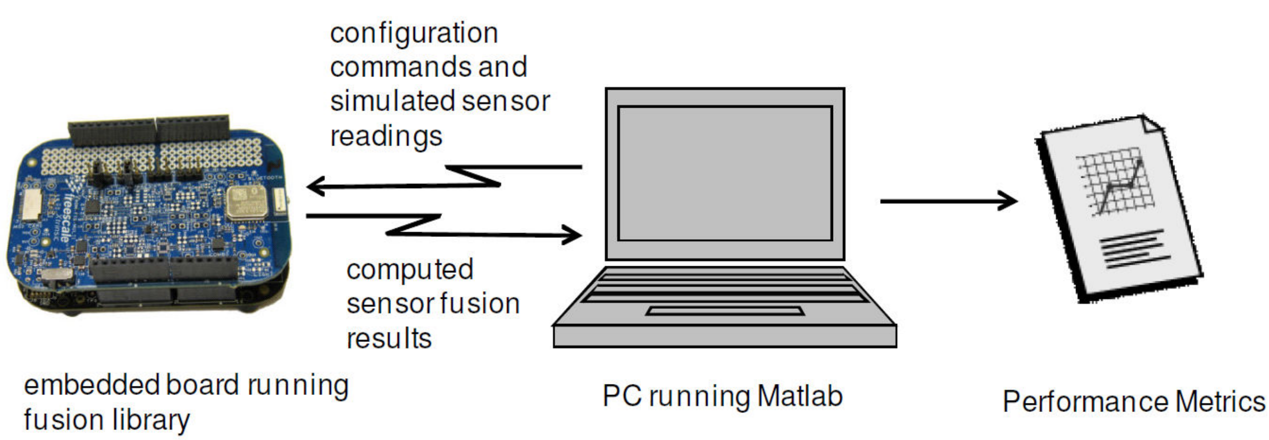


Figure 10. Mixed Simulation/Hardware Characterization

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 7November 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
  - 1.0 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 Freescale 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 30seconds.

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.

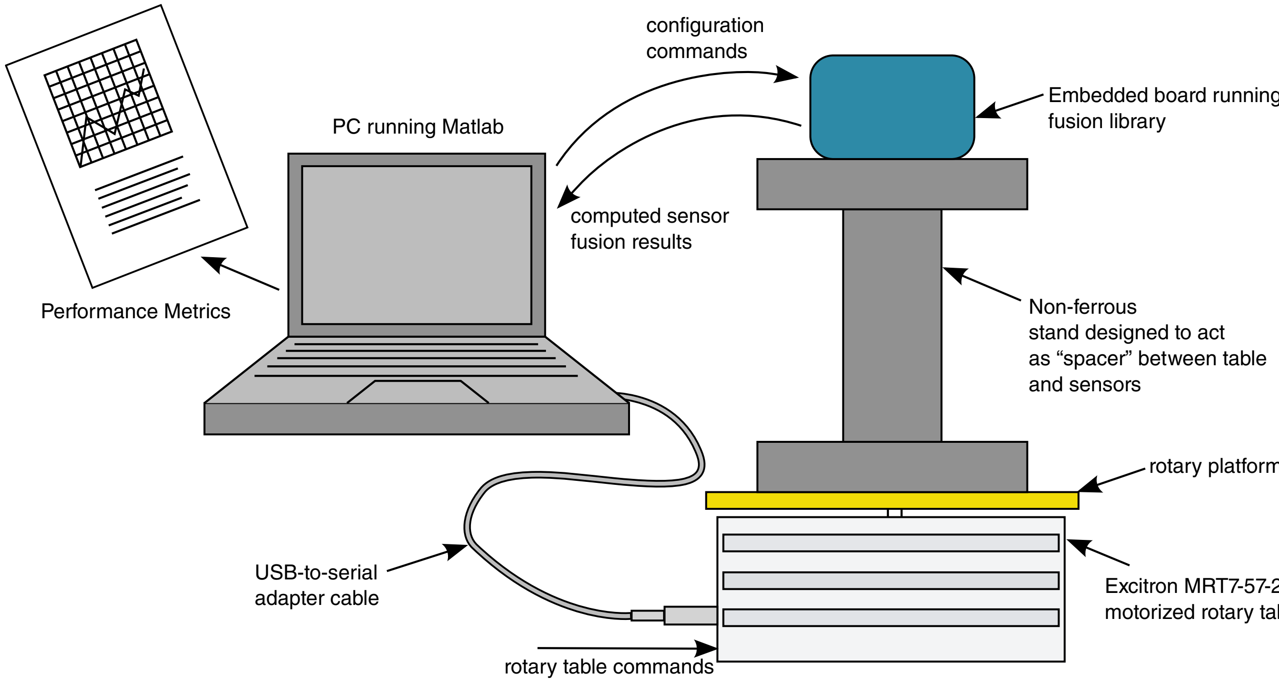


Figure 11. Benchtop Rotary Table Setup

3.4 Frame of Reference

Table 6 summarizes differences between the standard frames of reference supported by the fusion library.

Table 6. 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[1]	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.

3.5 Electrical Specifications

The parameters in each table are measured using the configurations defined in *Hardware Platforms Used to Measure Performance*. The MCU currents are measured with MCU stop mode enabled in idle task and UART communication disabled.

NOTE

For all the tables in this section, the data in the Demo config column uses all four algorithms to the left of the column running concurrently. Also, these tables were developed using the build #417 of the Sensor Fusion Library.

Table 7. FRDM-KL25Z Algorithms Execution Times

Metric	Accel only	Accel+Mag, eCompass	Accel+Gyro	9 DOF	Demo config	units
Algorithm execution time, max; see MCU Current	1.1	1.83	5.7	10	19	ms
MagCal execution time, max; see Fusion & MagCal Loop Execution Time	N/A	320	N/A	420	560	ms
Test Conditions (unless otherwise specified)						
MCU clock source: PLL 48MHz, core clock 48MHz, bus clock 24MHz						

Table 8. FRDM-KL26Z Algorithm Execution Times

Metric	Accel only	Accel +Mag, eCompass	Accel+Gyro	9 DOF	Demo config	units
Algorithm execution time, max; see MCU Current	1.1	1.83	5.7	10	19	ms
MagCal execution time, max; see Fusion & MagCal Loop Execution Time	N/A	320	N/A	420	560	ms
Test Conditions (unless otherwise specified)						
MCU clock source: PLL 48MHz, core clock 48MHz, bus clock 24MHz						

Table 9. Additional Electrical Specifications for FRDM-KL26Z only

Function	Fusion I_{dd} @ 25Hz rate	Fusion I_{dd} /Hz
Accel only	0.24	0.010
2D Mag	0.19	0.007
Gyro only	0.69	0.028
Accel + Mag eCompass	0.31	0.012

Table continues on the next page...

Table 9. Additional Electrical Specifications for FRDM-KL26Z only (continued)

Function	Fusion I_{dd} @ 25Hz rate	Fusion I_{dd} /Hz
Accel + Gyro	1.61	0.064
9-axis	3.09	0.124
subtotal	6.13	0.245
estimated RTOS/other	2.04	0.082

Table 10. FRDM-K20D50M Algorithm Execution Times

Metric	Accel only	Accel +Mag, eCompass	Accel+Gyro	9 DOF	Demo config	units
Algorithm execution time, max; see MCU Current	0.65	1.1	2.8	6.1	11.2	ms
MagCal execution time, max; see Fusion & MagCal Loop Execution Time	N/A	140	N/A	215	215	ms
Test Conditions (unless otherwise specified)						
MCU clock source: PLL 48MHz, core clock 48MHz, bus clock 24MHz						

Table 11. FRDM-K64F Algorithm Execution Times

Metric	Accel only	Accel +Mag, eCompass	Accel+Gyro	9 DOF	Demo config	units
Algorithm execution time, max; see MCU Current	0.126	0.28	0.29	0.56	1.15	ms
MagCal execution time, max; see Fusion & MagCal Loop Execution Time	N/A	15	N/A	15	15	ms
Test Conditions (unless otherwise specified)						
MCU clock source: PLL 50MHz, core clock 50MHz, bus clock 50MHz						

Table 12. FRDM-K64F Algorithm Execution Times

Metric	Accel only	Accel +Mag, eCompass	Accel+Gyro	9 DOF	Demo config	units
Algorithm execution time, max; see MCU Current	0.126	0.28	0.29	0.56	1.15	ms
MagCal execution time, max; see Fusion & MagCal Loop Execution Time	N/A	15	N/A	15	15	ms
Test Conditions (unless otherwise specified)						
MCU clock source: PLL 50MHz, core clock 50MHz, bus clock 50MHz						

Table 13. FRDM-K64F Algorithm Execution Times

Metric	Accel only	Accel +Mag, eCompass	Accel+Gyro	9 DOF	Demo config	units
Algorithm execution time, max; see MCU Current	0.068	0.15	0.15	0.27	0.58	ms
MagCal execution time, max; see Fusion & MagCal Loop Execution Time	N/A	7	N/A	7	7	ms
Test Conditions (unless otherwise specified)						
MCU clock source: PLL 120MHz, core clock 120MHz, bus clock 60MHz						

3.6 Computation Metrics

The Sensor Fusion Library computations are measured directly using the Sensor Fusion Toolbox and the ARM® sysTick clock.

3.6.1 Statistics from Build #420 of the Sensor Fusion Library

For the current build (420), Freescale used the built in sysTick counter to measure each iteration of the algorithm 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 will vary depending device movement are presented as “typical” numbers.

Table 14. SysTick Values for Freedom Development Platforms and Sensor Combinations

Freedom Platform	Accel	2D Mag	Gyro	eCompass	Accel +Gyro	9-axis	Reduced eCompass [1]
SysTick Values For CodeWarrior							
FRDM-KL25Z	60	45.3	164	72	380	728	N/A
FRDM-KL26Z	58	45	164	72	378	728	N/A
FRDM-KL46Z	59	46	164	72	380	725	72
FRDM-K20D50M	35	26	64	43	311	656	N/A
FRDM-K64F	3.1	2.8	4.6	3.8	22.9	58.3	N/A
SysTick Values for Kinetis Design Studio							
FRDM-KL25Z	58	45	162	72	376	710	N/A

Table continues on the next page...

Table 14. SysTick Values for Freedom Development Platforms and Sensor Combinations (continued)

Freedom Platform	Accel	2D Mag	Gyro	eCompass	Accel +Gyro	9-axis	Reduced eCompass [1]
FRDM-KL26Z	45	45	161	72	376	705	N/A
FRDM-KL46Z	58	45	161	72	376	705	72
FRDM-K20D50M	36	27	67	45	313	594	N/A
FRDM-K64F	3.1	2.9	4.3	3.8	22.9	58.6	N/A

1. Except for "Reduced eCompass" and algorithm-specific KL25Z numbers, the standard "all algorithm build" is used for all measurements.

Table 15. Memory Requirements for Freedom Development Platforms and Sensor Combinations for CodeWarrior and All Algorithms

Freedom Platform	Text	Data	BSS	Flash	RAM
FRDM-KL25Z	93396	52	15316	93448	15368
FRDM-KL26Z	93388	52	15316	93440	15368
FRDM-KL46Z	93332	52	15316	93384	15368
FRDM-K20D50M	87500	52	15508	87552	15560
FRDM-K64F	79224	64	20600	79288	20664
FRDM-KL46Z-eCompass	50572	28	5872	50600	5900

Table 16. Memory Requirements for Freedom Development Platform KL25Z and Sensor Combinations for CodeWarrior and Single Algorithm

Freedom Platform	Text	Data	BSS	Flash	RAM
FRDM-KL25Z (pressure/temp)	21396	48	11412	21444	11460
FRDM-KL25Z (accel only)	33576	52	11556	33628	11608
FRDM-KL25Z (2D mag)	37684	52	11556	37736	11608
FRDM-KL25Z (gyro only)	32320	52	11520	32372	11572
FRDM-KL25Z (accel+mag only)	57140	52	11564	57192	11616
FRDM-KL25Z (accel + gyro)	50152	52	12560	50204	12612
FRDM-KL25Z (9-axis)	73616	52	13496	73668	13548

Table 17. Memory Requirements for Freedom Development Platforms and Sensor Combinations for Kinetis Design Studio and All Algorithms

Freedom Platform	Text	Data	BSS	Flash	RAM
FRDM-KL25Z	85748	732	15352	86480	16084

Table continues on the next page...

Table 17. Memory Requirements for Freedom Development Platforms and Sensor Combinations for Kinetis Design Studio and All Algorithms (continued)

Freedom Platform	Text	Data	BSS	Flash	RAM
FRDM-KL26Z	85740	732	15424	86472	16156
FRDM-KL46Z	85684	732	15352	86416	16084
FRDM-K20D50M	78020	732	15544	78752	16276
FRDM-K64F	74080	744	20840	74824	21584
FRDM-KL46Z-eCompass	50740	708	5908	51448	6616

3.7 Magnetic Calibration Metrics

3.7.1 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 12. 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 2 - 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 Freescale magnetic calibration library.

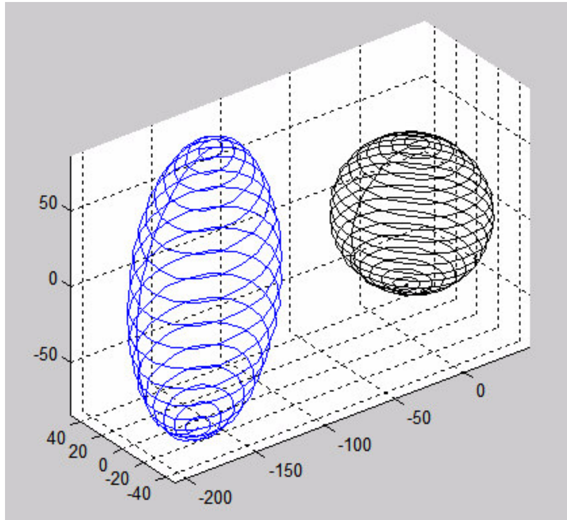


Figure 12. Distorted (left) and Corrected (right) Magnetic Field Data (simulated)

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 Freescale magnetic calibration library.

3.7.2 The Magnetic Buffer

Freescale'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 13 shows improvement in standard deviation of computed results (for a uniform field) versus constellation size. This same behavior may be seen at startup as the constellation fills with measured data for the first time.

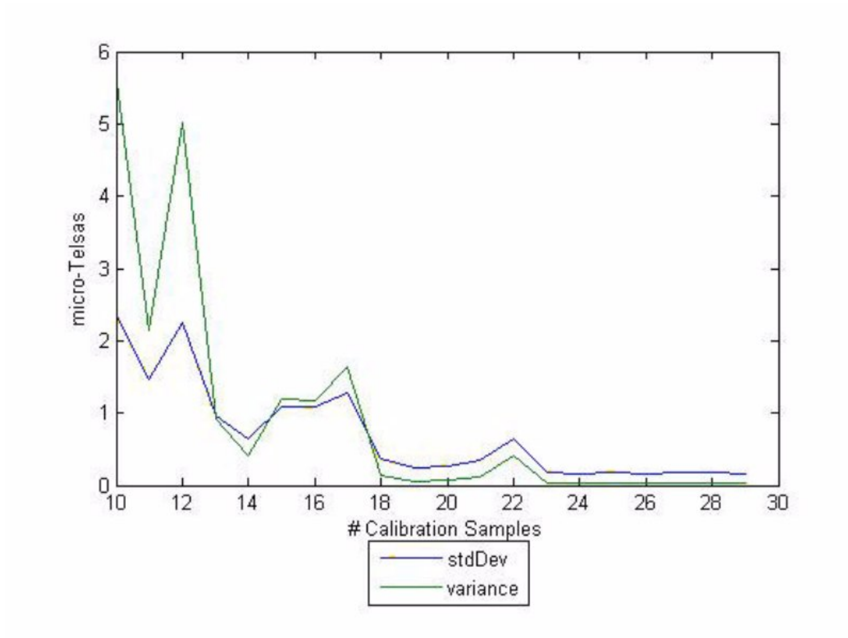


Figure 13. Simulated Standard Deviation and Variance as f(constellation size)

3.7.3 Magnetic Calibration Performance Metrics

Table 18 provides some basic guidance with regard to performance of the magnetic calibration library in a stand-alone configuration.

Table 18. Magnetic Calibration Performance Metrics

Characteristic	Symbol	Conditions(s)	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.							

3.8 Fusion Model Performance Metrics

3.8.1 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 using basic models which include noise effects, but ignore nonlinearity and other factors. See Simulation Environments for details.

3.8.2 Gyro Offset Step Response

Gyro Offset Step Response plots are shown in Figure 14 and Figure 15 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 Freescale recommends be turned OFF).

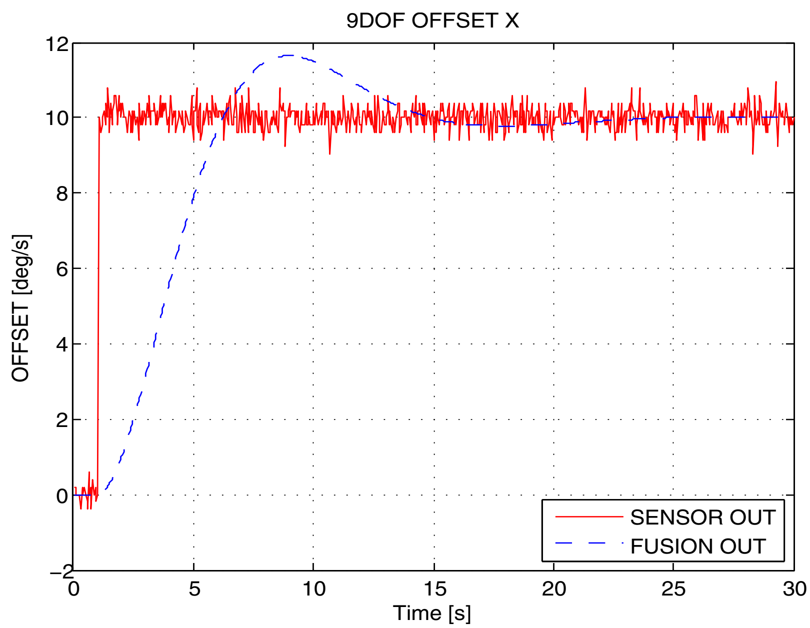


Figure 14. Gyro Offset Step Response for 9-Axis algorithm

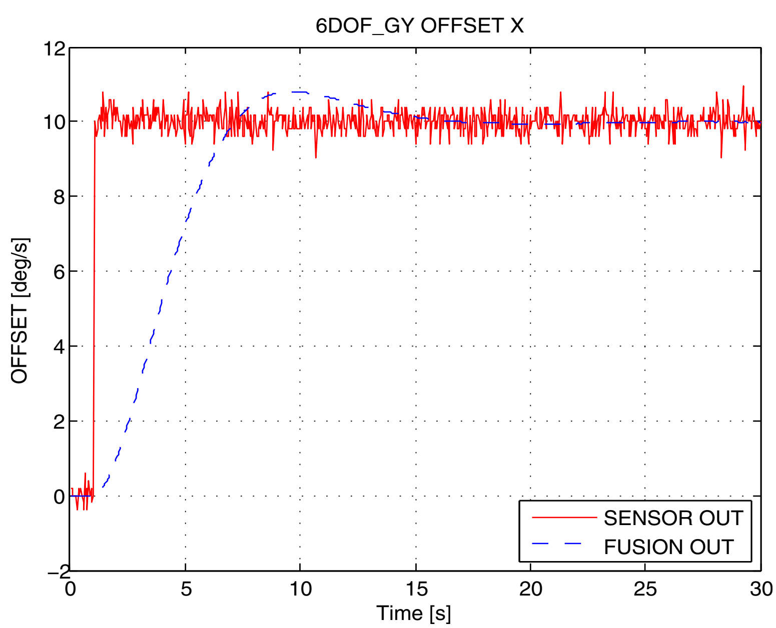


Figure 15. Gyro Offset Step Response for 6-Axis Accelerometer+Gyro algorithm

3.8.3 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 16, Figure 17 and Figure 18 present the true and 9-axis algorithm calculated linear acceleration for accelerations lasting 0.5 s, 1 s and 5srespectively.

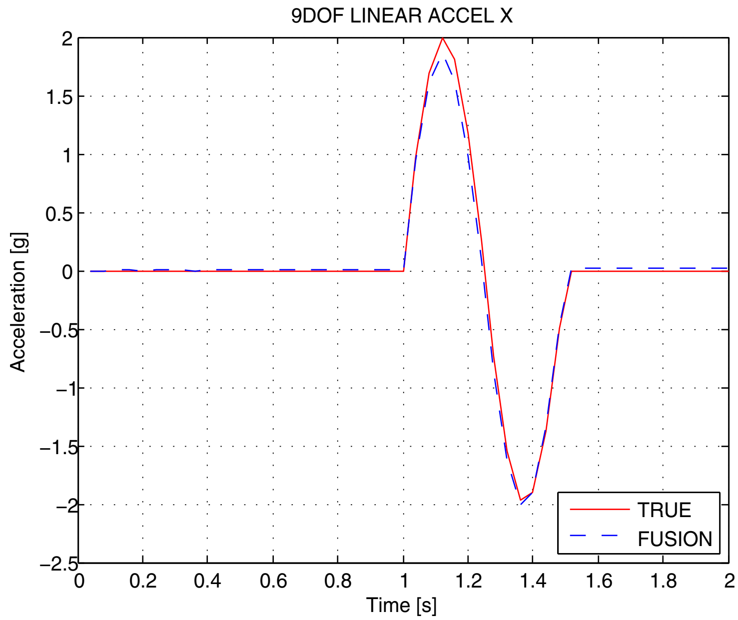


Figure 16. True and calculated linear acceleration, acceleration time 0.5 s

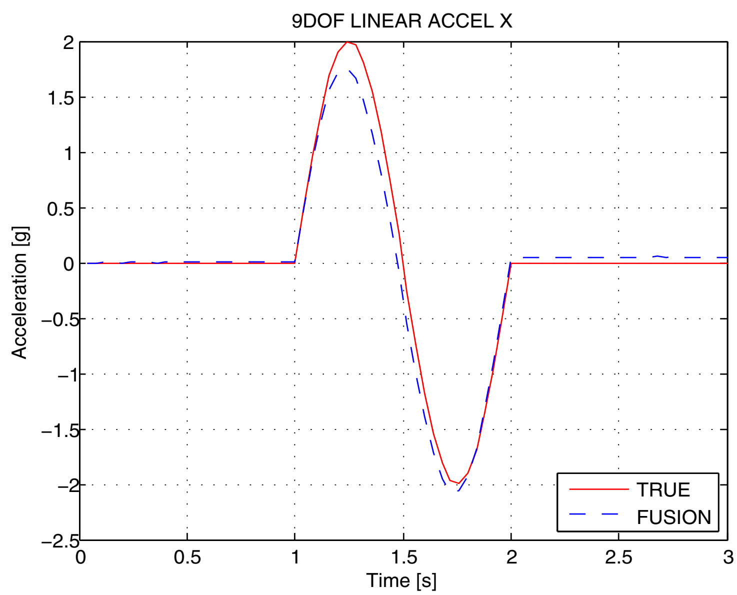


Figure 17. True and calculated linear acceleration, acceleration time 1 s

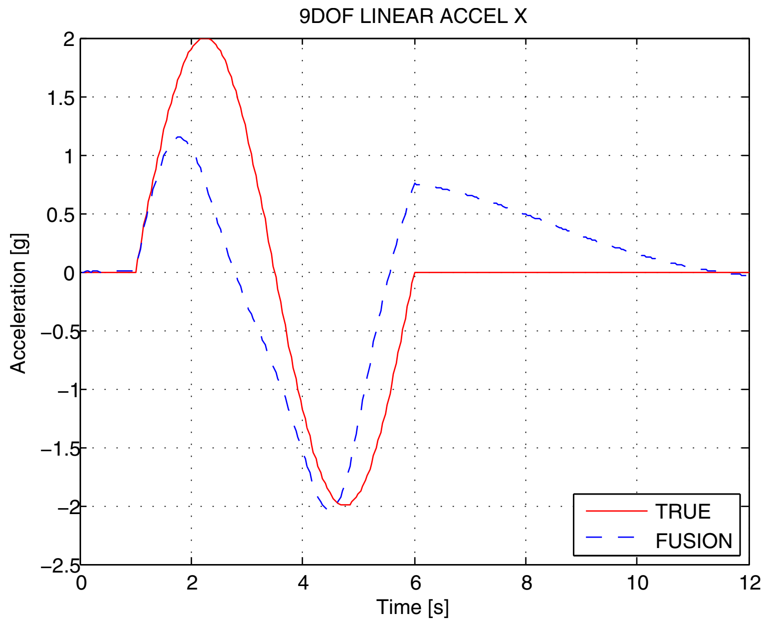


Figure 18. True and calculated linear acceleration, acceleration time 5 s

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.

3.8.4 Limitations Imposed via Sensor Choice/Configuration

Table 19 represents the sensor configuration used to determine parameters specified in this datasheet.

Table 19. Sensor/Configuration Imposed Limitations

Characteristic	Symbol	Min	Max	Unit
Maximum Angular Rate	AR_{max}	–1600	+1600	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.

3.8.5 9-Axis Parametrics

Table 20 provides guidance for the 9-axis (accelerometer/magnetometer/gyro) indirect Kalman filter implementation.

NOTE

All parametrics provided in the following table are based on simulations.

Table 20. 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 Maximum Angular Rate	—	1600[1]	—	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 Freescale 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.

3.8.6 6-axis Accelerometer + Magnetometer Parametrics

NOTE

All parametrics provided in the following table are based on simulations.

Table 21. 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[1]
[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.

3.8.7 6-axis Accelerometer + Gyro Parametrics

NOTE

All parametrics provided in the following table are based on simulations.

Table 22. 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[1]	—	<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

3.8.8 3-axis Accelerometer Only Parametrics

NOTE

All parametrics provided in the following table are based on simulations.

Table 23. 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 NOTE 2 - sample periods	—	<5	—	output sample period
NOTES:						
Note 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) $TAE = \arctan\left(\frac{\sqrt{NRMS^2_x + NRMS^2_y}}{1g - NRMS_z}\right)$ where NRMSx, NRMSy, NRMSz - RMS values of accelerometer noise for X, Y, Z axes in g units.						
NOTE 2: Number of output sampling periods where one period = 40 ms						

4 Test Descriptions

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

4.1 MCU Current

4.1.1 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 in Kinetis Cortex M0+ or Kinetis Cortex M4. The Freedom Development Platform was powered via the OpenSDA USB port for the results specified.

4.1.2 Procedure

1. This procedure uses a modified version of the standard demo build for KL26Z:
   a. Added separate $sysTick_{other}$ calculations for everything outside of the existing sysTick computations (which is referred to as $sysTick_{fusion}$)
   b. $sysTick_{fusion}$ 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. Percentage of time spent in the core fusion routines = $100*sysTick_{fusion}/(sysTick_{fusion} + sysTick_{other})$
3. I_{dd} Current into the MCU is measured via KL26Z jumper J5 using a simple DVM
4. Fusion I_{dd} = MCU I_{dd} X percentage
5. All currents shown are in mA
6. Sample size = 1 board

4.2 Flash and RAM Required

4.2.1 Intent

These parameters are total RAM and flash memory required to implement and execute the Fusion Library plus MQX-Lite RTOS. The projects for the binaries were created using both CodeWarrior and Kinetis Design Studio.

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 Kinetis Cortex M0+ or KinetisCortexM4.

4.2.2 Procedure

The text, data and bss sizes for each build were extracted using the technique outlined at MCU on Eclipse; text, data and bss: Code and Data Size Explained.

1. **text**; includes the user code, the vector table plus constants.
2. **data**; is for initialized variables, and it counts for RAM and FLASH. The linker allocates the data in FLASH which then is copied from ROM to RAM in the startup code.
3. **bss**; is for the uninitialized data in RAM which is initialized with zero in the startup code.

Refer to section Computation Metrics for statistics gathered using the #420 build.

4.3 Fusion & MagCal Loop Execution Time

Tables Table 7 through Table 13 apply to this section of the document.

4.3.1 Intent

The objective is to determine how much of the CPU bandwidth is consumed by the algorithms. Because it is expressed in msec, this metric must be associated with a specific hardware configuration similar to those defined in Kinetis Cortex M0+ or Kinetis_Cortex_M4

4.3.2 Procedure

Fusion and magnetic calibration loop execution time is manually measured using GPIO toggles at the beginning and end of executed code segments. The duration between toggles is measured using an oscilloscope.

Table 14 through Table 17 pertaining to the sysTick methodology, apply to this section of the document.

4.4 Compass Heading Linearity and Accuracy

4.4.1 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 19.

This metric must be stated specifically for a hardware configuration similar to those defined in Kinetis Cortex M0+ or Kinetis Cortex M4. .

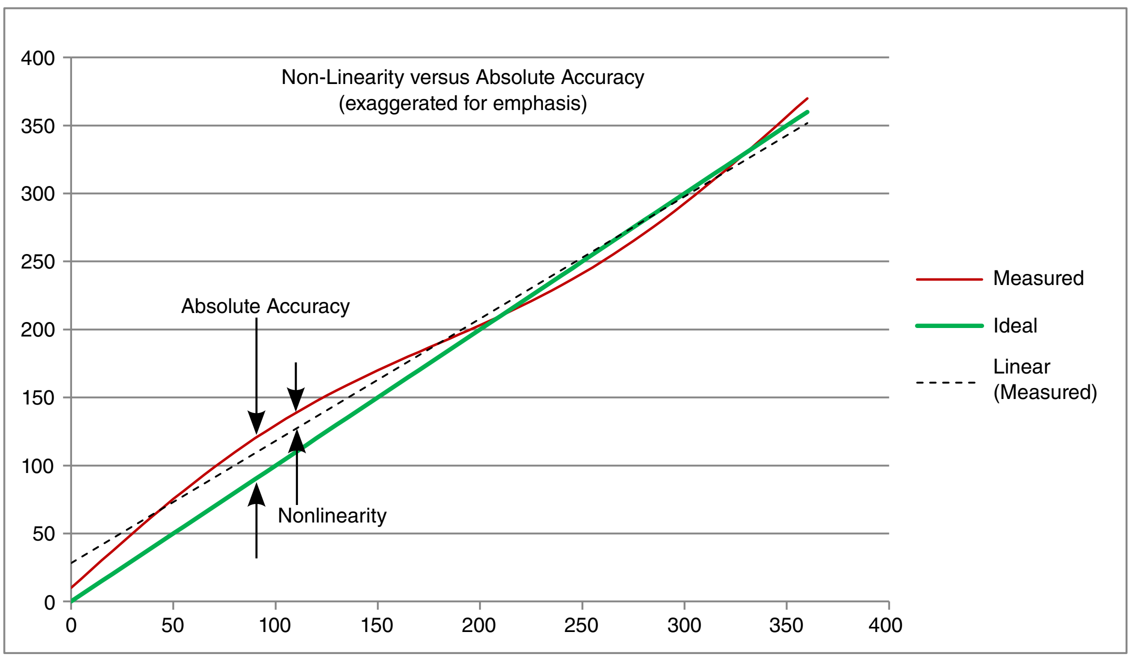


Figure 19. Nonlinearity vs. Absolute Accuracy

4.4.2 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.

4.5 Orientation Static Drift

4.5.1 Intent

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

4.5.2 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 Kinetis Cortex M0+ or Kinetis Cortex M4. 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 20

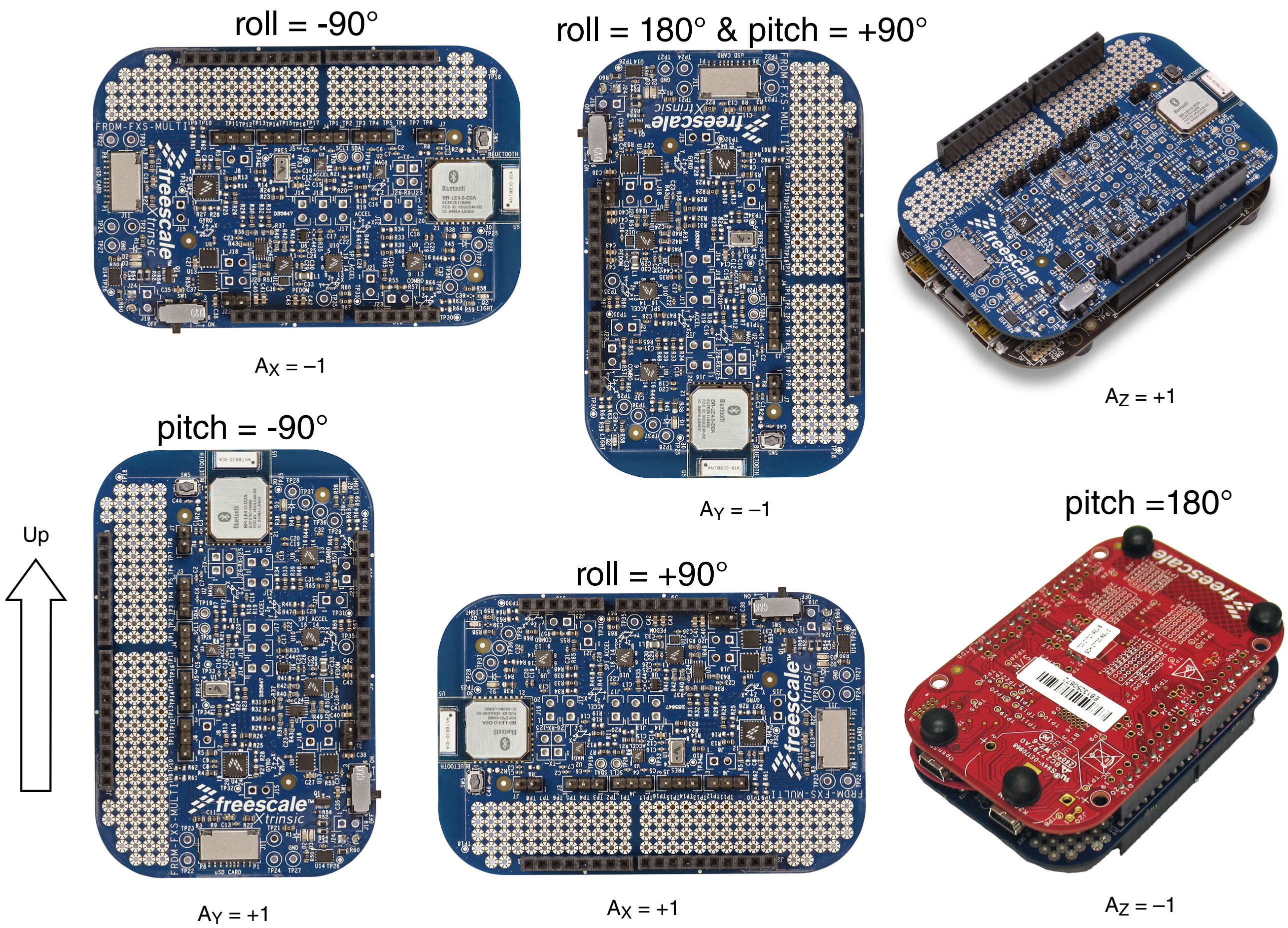


Figure 20. Major orientations relative to gravity

4.6 Orientation Static Noise

4.6.1 Intent

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

4.6.2 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.

4.7 Orientation Dynamic Drift

4.7.1 Intent

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

4.7.2 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.

4.8 Maximum Angular Rate

4.8.1 Intent

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

4.8.2 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.

4.9 Orientation Response Delay

4.9.1 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 21.

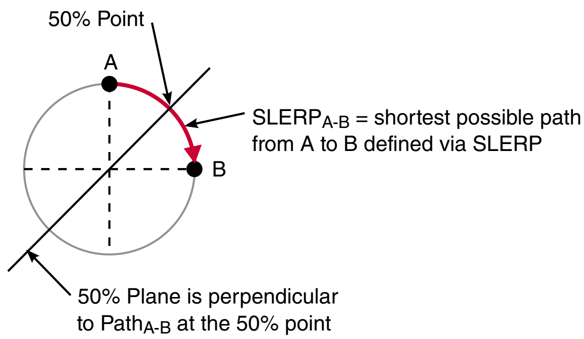


Figure 21. Propagation delays are measured at the plane that bisects and is perpendicular to SLERP$_{A-B}$

4.9.2 Intent

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

4.9.3 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.

4.10 Orientation Magnetic Immunity (Static Device)

4.10.1 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.

4.10.2 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.

4.11 Orientation Magnetic Immunity (Moving Device)

4.11.1 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.

4.11.2 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.

4.12 Error in Computed Gyro Bias

4.12.1 Intent

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

4.12.2 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

4.13 Gyro Offset Step Response

4.13.1 Intent

Measure how well the fusion library tracks varying gyro offset.

4.13.2 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.

4.14 Error in Computed Linear Acceleration

4.14.1 Intent

Measure how well the fusion library tracks linear acceleration.

4.14.2 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.

5 Revision history for XSFLK_DS

Revision number	Revision 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

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number XSFLK_DS
Revision 0.6, 9/2014