

Intelligent Sensing Framework (ISF) 2.1 for Kinetis, Frequently Asked Questions

Introduction

This document is a compilation of the most relevant and frequently asked questions about the Freescale Intelligent Sensing Framework for Kinetis® MCUs (ISF 2.1).

Introduction.....	1
What is ISF?	2
When is ISF v2.1 available?	2
How do other sensor frameworks compare to ISF?	2
What Real Time Operating System (RTOS) is used by ISF?	2
Does ISF Plan to use the Freescale OS Abstraction?	2
How is Sensor Fusion implemented in ISF?	2
What is the difference between ISF and the Freescale Sensor Fusion library?	3
Is ISF released as source code or does it contain libraries as in previous versions?	3
Must one use all of the components supplied in ISF?	3
How does ISF change for different applications?	3
What sensors are supported by ISF?	3
How long does it take to add a new sensor to ISF?	4
Can one add a new sensor adapter to ISF?	4
Can one add a non-Freescale or third party sensor to ISF?	4
Are the sensors polled or event/interrupt driven?	4
How long does it take to configure a new Kinetis MCU?	4
Does ISF support all pinout configurations for the Kinetis KL25Z?	4
Will ISF merge with the Kinetis SDK eventually?	5
Can the power states for peripheral devices be controlled?	5
Can algorithms for motion, for example free fall, be added to ISF?	5
If I select a sensor and configure it incorrectly in ISF, will ISF alert me?	5
Can the list of host commands provided by ISF be extended?	5
How does ISF support IoT applications?	5
Can one add a new communication protocol adapter to ISF?	5
Does ISF support IoT Transports such as 6LoWPAN, Wi-Fi, BLE, and ZigBee?	6
Does ISF support IoT Protocols like MQTT, Thread, CoAP, and AllJoyn?	6
Revision History	6

What is ISF?

What is ISF?

ISF stands for the Intelligent Sensing Framework. ISF 2.1 for Kinetis MCUs uses Processor Expert® technology to autogenerate an embedded sensor application in less than 30 minutes without writing a single line of code.

When is ISF v2.1 available?

ISF is available now at www.freescale.com/isf. The ISF Processor Expert components (contained in .ISF PEx.PEupd files) are updated independently of ISF releases. These component update files are available for download at the same web site.

How do other sensor frameworks compare to ISF?

The Open Sensor Platform is a comparable framework, but is not a direct competitor of ISF. Open Sensor Platform is developed specifically for the Android sensor APIs.

The ARM® mbed™ is a development environment provided by ARM Limited, Inc. This environment includes a collection of basic sensor drivers for Freescale's sensors, but does not create finished applications, as ISF does.

ISF provides a level of configurability and autogenerated source code that is unavailable elsewhere, while targeting the entire product line of Kinetis MCUs.

What Real Time Operating System (RTOS) is used by ISF?

ISF uses a very lightweight kernel of MQX™ RTOS, known as MQX Lite. MQX Lite is the largest single module in ISF. This module consumes almost half of the memory used by ISF.

Does ISF Plan to use the Freescale OS Abstraction?

In a future release of ISF, Freescale plans to utilize the Freescale OS Abstraction and Processor Expert® technology to allow other RTOSs to be used, including MQX, µCOSII®, and µC/OSIII®, as well as MQX Lite implemented in the current release. In addition, the user will have the option to use no RTOS, or a "bare board" implementation to significantly reduce the memory footprint.

How is Sensor Fusion implemented in ISF?

ISF implements 9-axis Sensor Fusion using a special virtual sensor called the *Orientation sensor*. The Orientation sensor component allows the user to configure the Sensor Fusion algorithm at a high level including selecting the underlying sensors (accelerometer, gyroscope, and magnetometer), the coordinate system (NED, Android, or Microsoft Windows 8), the oversampling ratios, and the individual fusion algorithms to run. The Orientation sensor also provides default sensor axis orientations and polarities for the magnetometers, [MAG3110](#) and [FXOS8700CQ](#), based on the coordinate system selection, plus an advanced feature that allows the user to override those sensor part orientations for their own target board configuration.

The whole idea is to select whatever sensor you want to perform the Sensor Fusion tasks you want.

10-axis sensor fusion will be available in 2Q2015 with an update to the ISF R2P1 PEx.PEupd file.

What is the difference between ISF and the Freescale Sensor Fusion library?

The Freescale Sensor Fusion library is a standalone, open source library that fuses the data from an accelerometer, gyroscope, magnetometer, and pressure sensor, or any subset of these available on specific Freescale Freedom boards. ISF is a general purpose software development platform available as public source. See the [ISF software license](#) for an explanation of how to use the software generated by ISF. Having the source code available for ISF enables easier debugging during code and product development. ISF includes the Freescale Sensor Fusion Library as a virtual Orientation sensor, and sample applications are available. This allows users to configure the library and component inside Processor Expert. The Orientation sensor provides the same outputs as those provided by the Freescale Sensor Fusion Library.

Is ISF released as source code or does it contain libraries as in previous versions?

ISF is no longer offered as a library. All of the source code is available. The source code is automatically generated through the use of Processor Expert technology. Users can see the code that is generated. At each layer, APIs are available. An [API reference manual](#) is provided as part of the documentation to describe the interfaces at each level.

Must one use all of the components supplied in ISF?

ISF allows the user to select which ISF features are brought into the working project. For example, for wearable applications, a host interface is not required. The Command Interpreter component need not be included as a result, and the corresponding source code is not generated in that case.

How does ISF change for different applications?

ISF can be configured, allowing users to optionally bring in and exclude various underlying features. Processor Expert technology provides a level of flexibility for the user to define the core features, communication channels, and sensors to be included. In addition, ISF provides an embedded application component. This can be modified by customers for their own embedded application, or they can write their own.

What sensors are supported by ISF?

This document applies to the following ISF-supported Freescale sensors:

Accelerometer:	FXLS8471Q , FXOS8700CQ , MMA865x (MMA8652 and MMA8653)
Magnetometer:	MAG3110 , FXOS8700CQ
Gyrometer:	FXAS21002C
Pressure:	MPL3115A2
Orientation:	Sensor Fusion (any combination of accelerometer, gyroscope, and magnetometer)

How long does it take to add a new sensor to ISF?

How long does it take to add a new sensor to ISF?

New sensors are added to the ISF_Core Processor Expert component using the System Sensor Selection property and may be subscribed to using the ISFEmbApp Processor Expert component using the Subscription List property. These Processor Expert components, ISF_Core, and ISFEmbApp, work together to add a supported sensor to an application in minutes. They also provide guidance and error checking to the user.

Can one add a new sensor adapter to ISF?

It is possible to add new sensor adapters to ISF. However, at the present time, Freescale has to do the work. The time to create a sensor adapter is a function of the number of sensor features included in the sensor adapter. Future plans include a generic sensor adapter which will allow users to create their own sensor adapters. Earlier releases of ISF allowed users to make this addition. However, the integration with Processor Expert technology made this more complex.

Can one add a non-Freescale or third party sensor to ISF?

There is nothing in ISF that prevents adding a non-Freescale sensor adapter to ISF. Obviously, the Freescale Sensor organization is not motivated to provide sensor adapters for our competition. Adding one's own sensor adapter is complex. Users are encouraged to contact the Freescale Software Services organization to make arrangements for adding a non-Freescale sensor adapter to ISF.

The first step in creating any new sensor adapter is to write a standalone sensor adapter or driver for the sensor. This can be done by mimicking existing sensor adapters. The second step is to adapt the sensor adapter into a Processor Expert component. The first step is easy. The second step is relatively difficult. For this reason, it is recommended that Freescale do the componentization of the sensor adapter via the Freescale Software Services organization.

Are the sensors polled or event/interrupt driven?

The sensor adapters currently provided by ISF v2.1 use a high resolution timer, accurate to one microsecond, to read sensor data at the requested frequency. Updating the existing sensor adapters to be interrupt-driven is an enhancement that is planned in a future release of ISF.

How long does it take to configure a new Kinetis MCU?

ISF is using Processor Expert technology to enable users to move between MCUs easily. Processor Expert allows new MCUs to be configured, which requires the projects to be regenerated and recompiled. Basic projects have been provided in the ISF release to support the Kinetis KL25Z, K22F, and K64F MCUs. Configuring a new Kinetis MCU depends on the Processor Expert Logical Device Driver (LDD) files and the availability of the PIT timer. Future releases will have a closer integration of ISF with the Kinetis SDK to allow wider integration of new Kinetis MCUs.

Does ISF support all pinout configurations for the Kinetis KL25Z?

Yes. There are four different sets of device pinouts available for the Kinetis KL25Z. However, if you are using a KL25Z but not using a FRDM-KL25Z, the CPU has to be reconfigured for the new pin multiplexing. One can use the Processor Expert CPU component to perform this operation.

Will ISF merge with the Kinetis SDK eventually?

There are no plans to provide a standalone Sensor SDK at the present time. ISF is a complex framework with dependencies on the RTOS, communication protocols, MCU timers, and other factors. The Kinetis SDK is essential for implementing future I²C, SPI, RTOS and other services. A future release of ISF will fully integrate the Kinetis SDK.

Can the power states for peripheral devices be controlled?

Yes, it is possible for peripherals to be put in a low-power state using the features of Processor Expert.

Can algorithms for motion, for example free fall, be added to ISF?

Yes, these applications can be added in the application data processing landing pads within the ISFEmbApp (Embedded Application) component. In addition, it is possible to write a custom embedded application on top of ISF and use the available APIs to access ISF features, sensors, and communications.

If I select a sensor and configure it incorrectly in ISF, will ISF alert me?

Most of these types of operations are automatically checked during Processor Expert component configuration prior to code generation. Some components may flag errors and require reconfiguring before code generation is allowed.

For example, if you want to use a specific sensor, but forget to bring it into the project with the ISF_Core, an error message is not provided by ISF, but you will get a compilation error. However, if you attempt to assign a sample period outside the range of the sensor, the parameter turns red and displays an error message indicating that an adjustment is required. Most of the common types of configuration errors are flagged, but not all of them.

Can the list of host commands provided by ISF be extended?

Yes. The basic host commands provided within ISF cover most needs, but if you want to add your own, the mechanisms are available for you to do so. This is explained in the [ISF Software Reference Manual](#).

How does ISF support IoT applications?

The Internet of Tomorrow assumes, clusters of sensors, connected via wired or wireless technologies to a hierarchy of devices in a network. Each of these devices has sensors and is able to communicate, in a standard way, to enable applications based upon that configuration. ISF can be used at the leaf or edge node portion of the network to manage the sensor data.

Can one add a new communication protocol adapter to ISF?

Currently, ISF supports I²C, SPI, and UART protocols. ISF was architected to allow adding additional communication protocols to ISF. Customers and users interested in other communication protocols should contact the Freescale Software Services organization.



Does ISF support IoT Transports such as 6LoWPAN, Wi-Fi, BLE, and ZigBee?

Does ISF support IoT Transports such as 6LoWPAN, Wi-Fi, BLE, and ZigBee?

ISF is designed to be used as a portion of an overall IoT solution. An embedded application can use the ISF API to gather the sensor data. The application can then integrate IoT transport stacks, available from Freescale’s Connectivity offerings, to produce IoT applications using these Transports and more.

Does ISF support IoT Protocols like MQTT, Thread, CoAP, and AllJoyn?

ISF is designed to be used as a portion of an overall IoT solution. An embedded application can use the ISF API to gather the sensor data. It can then integrate IoT protocols, available from other IoT solutions, to produce IoT applications using these protocols.

Revision History

Table 1. Revision History

Rev. No.	Date	Description
0	3/2015	Initial Public Release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc. , Reg. U. S. Pat. & Tm. Off. ARM and mbed are trademarks of ARM Limited, Inc. μ C/OS-II and μ C/OS-III are trademarks of Micrium, Inc. All other product or service names are the property of their respective owners.

© 2015 Freescale Semiconductor, Inc.

Document Number: ISF2P1_Kinetis_FAQ
Revision 0, 3/2015