Freescale Semiconductor
Application Note

Document Number:AN4507
Rev. 1, 6/2012

Using the Kinetis Security and Flash Protection Features

by: Melissa Hunter
Automotive and Industrial Solutions Group

Contents

1 Introduction

The Kinetis family of microcontrollers include system security and flash protection features that can be used to protect code and data from unauthorized access or modification. This application note discusses usage of the security and flash protection features available on the Kinetis family processors. The chip security and flash protection features controlled using the flash configuration field will be covered in this application note.

The default security options are configured at reset based on the application image. All flash application images need to include configuration of the security and protection options. Configuration is required in all cases. Even if the required setting is to have the flash and security options disabled, the application image must configure the setting for those desired values. Therefore, anyone writing an application must be aware of the security and protection options and how to correctly configure them.

2 Security vs. protection

The Kinetis security features are system-level options that are designed to prevent unauthorized access to the processor and the code and data within the processor. Software intellectual

property (IP) is a very valuable investment and the security features protect that investment, prevent cloning, and secure sensitive data that might be stored in the memory.

- Security feature: Although the security on Kinetis is controlled by the settings that reside within the flash, it is not a flash-level feature. The flash provides the security options to the chip system logic, and the chip system logic takes action based on the settings. So, even though the user interacts with the flash to use the security settings, it must be considered as a processor mode selection because the decisions made by the user will affect the entire processor. In fact, the option to enable or disable security has very little impact on the flash itself. Even when the security is enabled, the flash is still fully operational. This means that firmware residing in the flash that enables security can still modify the flash because read, erase, and program operations to the flash itself are not changed (with the possible exception of mass erase).
- Flash protection feature: By comparison, the flash protection features only affect the flash itself. The flash protection is designed to prevent accidental erasure or programming of areas of the flash. This option only affects the ability to modify selected flash regions; there is no effect on the rest of the processor.

2.1 Security options

The following sections will show the security options that are available on Kinetis. Initially, look at the actual register and field descriptions from the Kinetis reference manual available on http://www.freescale.com, then go into more detail on how each option works and how to configure them.

2.1.1 FSEC register and field settings

The Flash Security (FSEC) register contains several fields that are used to enable/disable security and also select a few features that come into effect when security is enabled. Figure 1 shows the FSEC register fields, and Table 1 shows the bit settings for each of the fields. The following sections discuss all of the options available in the FSEC register in more detail including impact on the system and recommended usage.

Bit	7	6	5	4	3	2	1	0
Read	KEYEN		MEEN		FSLACC		SEC	
Write								

Figure 1. Flash Security register

Table 1. FSEC field description

Field	Description
7–6 KEYEN	Backdoor Key Security Enable. This field enables and disables backdoor key access to the flash memory module. 00 Backdoor key access is disabled. 01 Backdoor key access is disabled (preferred KEYEN state to disable backdoor key access) 10 Backdoor key access is enabled. 11 Backdoor key access is disabled.

Table continues on the next page...

Table 1. FSEC field description (continued)

Field	Description
5–4 MEEN	Mass Erase Enable This field enables and disables the JTAG and EzPort mass erase capability of the flash memory module. The state of the MEEN field is relevant only when SEC is set to secure. When the SEC field is set to unsecure, the MEEN setting does not matter. 00 Mass erase is enabled. 01 Mass erase is enabled. 10 Massel erase is disabled. 11 Mass erase is enabled.
3–2 FSLACC	Freescale Failure Analysis Code. This field enables or disables access to the flash memory contents during returned part failure analysis at Freescale. When SEC is secure and FSLACC is denied, access to the flash contents by Freescale is denied and any failure analysis performed by Freescale factory test must begin with a full erase to unsecure the part (assuming mass erase is enabled). When access is granted (SEC is unsecure, or SEC is secure and FSLACC is granted), Freescale factory testing has visibility of the current flash contents. The state of FSLACC is relevant only when SEC is set to secure. When SEC is set to unsecure, the FSLACC setting does not matter. 00 Freescale factory access is granted. 01 Freescale factory access is denied. 10 Freescale factory access is denied. 11 Freescale factory access is granted.
1–0 SEC	Flash Security This field defines the security state of the MCU. In the secure state, the MCU limits access to flash memory module resources. If the flash memory module is unsecured using backdoor key access, the SEC field is forced to 10b. 00 MCU security status is secure. 01 MCU security status is secure. 10 MCU security status is unsecure, that is, the standard shipping condition of the MCU is unsecure. 11 MCU security status is secure.

2.1.1.1 Security enable/disable

While setting FSEC[SEC], the user must first decide whether to enable the security or not. The following sections discuss the effects of enabling or disabling security.

2.1.1.1.1 How enabling security affects JTAG

In order to protect the information inside the processor, debug access to the internals of the processor are disabled when security is on. JTAG scan chain operation is possible even when security is enabled.

Registers that reside within the JTAG and debug logic can be accessed, but no other registers or memory inside the processor can be accessed via the debug port.

One of the debug registers that can be read even when security is enabled is the MDM-AP status register. This register resides within the debug logic and can be used to verify the current security settings. This could be useful to solve problems in establishing a debug connection to the processor. If this happens then the MDM-AP value can be used to determine if security is enabled and preventing the debug connection.

If mass erase is still enabled through FESC[MEEN], then the JTAG port can also be used to request a mass erase. After the mass erase completes, security is disabled. This means that mass erase can be used to regain debugger connectivity with a part, but all the code/data that was in the flash is lost in the process.

2.1.1.1.2 How enabling security affects the EzPort

When security is enabled, the processor can still boot into EzPort mode as long as EzPort mode entry is enabled. The EzPort communications will still be active and can be used to send commands, but many of the commands will not be accepted. Most importantly the only resource that can be read with the security enabled is the EzPort Status register. Similar to the MDM-AP Status register, the EzPort Status register can be read to determine if flash security is enabled.

The EzPort can also be used to execute a mass erase of the flash to unsecure the processor (if the MEEN field allows).

2.1.1.1.3 How enabling security affects the FlexBus

When security is enabled, the FlexBus operation is programmable. SIM_SOPT2[FBSL] is used to control what accesses are allowed on the external bus when security is enabled. By default, all the external accesses to the FlexBus are blocked when security is enabled. There are options to allow only data access, or to allow both the data and opcode access. If both data and opcode accesses are allowed, then the FlexBus controller would behave in the same way when security is enabled or disabled.

NOTE

SIM_SOPT2[FBSL] has an effect only when security is enabled. If security is disabled, then SIM_SOPT2[FBSL] has no effect.

2.1.1.1.4 How enabling security affects the flash

Although the security state of the processor is delivered to the rest of the system on-chip (SoC) by the flash block, the flash operation is largely unaffected by the security state. Resident firmware already stored in the memory can run the same set of flash commands in either security state. Through firmware, the user can read, erase, and program the flash in the same way, even if the device is secured. Because the EzPort and JTAG access are restricted, the flash is not accessible to the outside world, but from within the part, the flash can be used normally. When using a communications port for field firmware upgrades, security can be enabled and the firmware upgrade code will still execute normally.

2.1.1.1.5 What happens when security is disabled

If the security is disabled, then all other options in this register are ignored. By default, the processor is actually secure, but during the reset process after the processor gets the security option from the flash, it will disable security if selected, and will open up access for debugging, EzPort, and FlexBus.

2.1.1.2 Backdoor key enable

The second option in the FSEC register is a backdoor key enable option controlled by FSEC[KEYEN]. When enabled, the backdoor key option allows for a means of temporarily disabling flash security if the correct 64-bit key value is provided when executing a flash verify backdoor access key command.

2.1.1.2.1 Important things to keep in mind for using the backdoor key feature

The backdoor key unlock feature can be extremely useful when using security; however, using the feature requires some pre-configuration. Here are a few important tips to keep in mind while using a backdoor key:

- Configure the 64-bit comparison key value for the application at the same time when security and the backdoor key feature are enabled. The key value cannot be configured on a secured device after it is already running without reprogramming sections of the flash using firmware and resetting the device.
- All 0s or all 1s cannot be used as the key value. The key value selected must be some combination with at least one 1 and one 0. If the user attempts to use a key value all 0s or all 1s, then the Verify Backdoor Access Key command will return an error.
- There is no preconfigured mechanism for obtaining a key value and running the Verify Backdoor Access Key command in the processor. The application needs to include a code sequence to allow a user to input a key value and then run the Verify Backdoor key command using the entered value.
- If the Verify Backdoor Access Key command detects a mismatch between the preprogrammed comparison key and the provided key, then a power-on reset is required before the Verify Backdoor Access Key command can be executed again.
- If the Verify Backdoor Access Key command correctly matches the comparison key and the provided key, then flash security is disabled temporarily. Once the processor is reset, it will return to the secured state unless action is taken to change the default security setting.

2.1.1.2.2 Using a backdoor key to unlock security

The steps below describe a typical sequence to use a backdoor key to temporarily disable security.

1. Program the flash on the processor with an application that enables security, and the backdoor key access feature, and sets a valid backdoor access comparison key value. See How to configure security options for information on how to configure these options. The application must also contain code to perform the additional steps described in this procedure.
2. The end-users must perform some action to indicate that they want to enter a key value to temporarily disable security. This could be through some form of serial command or hardware interaction. For example, the user can setup the application with a “backdoor” command that would be entered through a serial terminal command line interface, or can even have a push button on the board to generate an interrupt that causes the software to branch to a special backdoor key sequence.
3. After the firmware has detected that the user wants to temporarily unsecure the device, it must provide a prompt to the user to enter the 64-bit key value. Typically, this would be done using one of the communications ports on the processor like the UART, SPI, or Ethernet.
4. Wait for the user to enter the 64-bit key.
5. After the 64-bit user key has been obtained, load the flash FCCOBn registers with the command value and operands for the Verify Backdoor Access Key command, as shown in the table below.

Table 2. Verify Backdoor Access Key command

FCCOB numbers	FCCOB contents
0	0x45 (VFYKEY)
1–3	Not used
4	User Key Byte 0
5	User Key Byte 1

Table continues on the next page...

Table 2. Verify Backdoor Access Key command (continued)

FCCOB numbers	FCCOB contents
6	User Key Byte 2
7	User Key Byte 3
8	User Key Byte 4
9	User Key Byte 5
A	User Key Byte 6
B	User Key Byte 7

NOTE

Pay careful attention to the byte order and FCCOB register locations. The flash was originally designed for big-endian architectures, but Kinetis is little-endian. In order to prevent byte swap problems, it is usually best to handle the preconfigured key and user key as two 32-bit values and always read and write them as 32-bits. The FCCOB registers are 8-bit registers; however four can be written at a time using a 32-bit access.

6. Clear the FCSR[CCIF] field in flash to start execution of the Verify Backdoor Access Key command.
7. If the user key successfully matches the comparison backdoor key, then the security will be temporarily disabled.

Because most of this sequence would be handled by the application, there are other options. For example, during debugging, a push button can be setup on a board as an unlock button. Software could detect the button press and then execute the Verify Backdoor Access Key command, without actually requiring the user to input the key by some means. This approach is not very secure, and is therefore not recommended. However, the decision of how to correctly handle backdoor keys, or whether to use them at all, is up to the user.

2.1.1.3 Mass erase disable

The third security setting (FSEC[MEEN]) gives the option to completely disable mass erase functionality for the flash. Disabling the mass erase feature can provide an additional level of security for a system. This would prevent someone from using JTAG or EzPort mass erase commands to override security and load a new application. However, this means that the mass erase capability will also be lost.

NOTE

If the user disables mass erase with no other means to unsecure the device (no backdoor key and no firmware means of changing the default security options), then that device will be permanently secured.

NOTE

Even when mass erase is disabled, the flash Erase All Blocks command can be used to mass erase the device. This is a flash command that requires firmware in the device to execute the command.

2.1.1.4 Freescale access disable

FSEC[FSLACC] determines if Freescale can access the device when the part is secured. This field is primarily a concern if there is a suspected quality issue and parts are returned to Freescale for retest and failure analysis. The following table presents the conditions for Freescale access.

Table 3. Conditions for Freescale access

If	Then
The device is secured with Freescale access disabled and mass erase enabled	Freescale would start the failure analysis process by mass erasing the device to unsecure it.
The device is secured with both Freescale access and mass erase disabled	Freescale will have very limited ability to perform failure analysis on the device. For this reason, it is recommended that to leave mass erase enabled, if the user plans to disable Freescale access.

2.1.2 How to configure security options

As mentioned in Introduction, the application image itself determines the default security options. This is done by programming a specific location within the flash memory array known as the flash configuration field. During the reset sequence, the flash logic reads the values that are programmed into the flash configuration field locations (0x400–0x40F) and uses them to obtain default values for several flash registers and also passes some of the setting information on to the SoC.

NOTE

Because the application image is responsible for setting the default security options, all the flash images must contain valid flash configuration data. Failure to correctly configure the flash configuration field when programming the flash can lead to permanently securing the processor.

Table 4 shows the addresses of all of the flash configuration fields and gives a brief description of how each location is used. For full descriptions of the flash configuration field and flash registers including bit definitions, please see the reference manual for a specific Kinetis device available on http://www.freescale.com.

Table 4. Flash configuration field

Flash configuration field address	Size (Bytes)	Field description	Flash register initialized by field
0x400–0x407	8	Backdoor comparison key. See Backdoor key enable for details	
0x408–0x40B	4	Default program flash protection settings	FPROT0–3
0x40F	1	Default data flash protection settings. This setting is used only for FlexNVM devices. For P-Flash only devices, this field is unused (write as 0xFF).	FDPROT
0x40E	1	Default EEPROM protection settings. This setting is used only for FlexNVM devices. For P-Flash only devices, this field is unused (write as 0xFF).	FEPROT

Table continues on the next page...

Table 4. Flash configuration field (continued)

Flash configuration field address	Size (Bytes)	Field description	Flash register initialized by field
0x40D	1	Flash option byte. This field is used to configure SoC specific settings. The options can vary on different Kinetis devices. See the reference manual for the specific Kinetis device available at http://www.freescale.com to see the settings that are available for that processor.	FOPT
0x40C	1	Default flash security register setting.	FSEC

NOTE

It is extremely important that all the flash images downloaded to the processor contain appropriate values for the flash configuration field. Freescale recommends adding the flash configuration field values at the very end of the vector table. The vector table is stored at 0x0–0x3FF, and then the flash configuration field is 0x400–0x40F. Application code and data can start at 0x410. Please refer to the vectors.c and vectors.h files in the Kinetis 120MHz bare metal sample code. zip file available at http://www.freescale.com for a software example of how to do this.

2.1.3 How to disable security after it has been enabled

There are several ways to disable security; however, keep in mind that the unsecure options are affected by the flash security options that have been programmed into the flash configuration field. So, depending on the current state of the FSEC register these options might not be available.

NOTE

Because the FSEC register is initialized from the flash configuration field, all methods of disabling flash security are temporary. Once security has been disabled, it remains disabled only until the next reset. The flash configuration field must be modified in order to make a change to security settings that will go into effect after the next reset and every subsequent reset after that, as long as the flash configuration field is not modified again.

2.1.3.1 Mass erase via Debugger/JTAG

Debuggers and JTAG tools have very limited access to the device when a processor is secured. The only registers that can be accessed through the JTAG are the MDM-AP Status and Control registers. In order to allow debug tools to unsecure parts, bit 0 of the MDM-AP Control register can be set to request a mass erase of the processor. In order to use this method to disable security, FSEC[MEEN] must be set to a value other than 10 to allow mass erase capability. If the mass erase is disabled, FSEC[MEEN] = 10, then the mass erase request will be ignored by the flash and the device cannot be unsecured using this method.

Many debuggers will automatically use bit 2 of the MDM-AP Status register to determine if a device is secured when attempting to establish a connection. A debugger pop-up window might be used to alert that the device is secured and ask if a mass erase is desired to unsecure the device. Once the mass erase is completed and verified, security will be disabled. Some debuggers might automatically program the flash configuration field to put the flash into an unsecured state after the mass erase completes, FSEC = 0xFE.

2.1.3.2 Mass erase via EzPort

The EzPort also has the ability to request a mass erase to unsecure a device using a bulk erase command (BE). As with the mass erase via debugger or JTAG, the FSEC register must be configured to enable mass erase functionality. After the mass erase is completed and verified, security will be disabled. At this point, the EzPort can be used to program the flash with a new image including the desired flash configuration field values.

2.1.3.3 Disable security using a backdoor key

If backdoor key functionality is enabled and a valid backdoor key is configured in the flash configuration field, then the backdoor key entry sequence described in Using a backdoor key to unlock security can be used to temporarily disable security. Unlike the mass erase options for disabling security, the flash contents are completely unaffected, if this option is used. Because the flash is not modified, the flash configuration field is also unchanged, the security is only temporarily disabled unless the flash configuration is reprogrammed. Code and data stored in the flash can be read and written freely at this point.

2.1.3.4 Reprogramming flash security fields using firmware

As stated in How enabling security affects the flash, firmware already programmed into the flash can be used to erase and program the flash. This includes the possibility of using firmware commands to erase and then reprogram the flash configuration field values. This method is a special case in that the direct reprogramming of the flash configuration field doesn't change the current value of the FSEC register. Therefore, if the flash configuration field is reprogrammed to a value that unsecures the part using this method, the change won't actually go into effect until the processor resets and loads the new value from the flash configuration field into the FSEC register.

NOTE

On most Kinetis devices, the flash configuration field is located in the first sector of the flash. So, in order to erase and reprogram the flash configuration field, the entire vector table including the initial stack pointer and PC values need to be erased. If power is lost or firmware does not update all of the values that were erased correctly, then a mass erase of the entire device will most likely be required.

2.1.4 Security and the flash swap feature

Kinetis devices with two or more program flash (P-Flash) memory blocks have the P-Flash swap feature. The flash swap feature allows the system programmer to configure the logical memory map of the P-flash space such that either of the two physical P-flash blocks can exist at relative address 0x0000. Swapping the base address of the two P-flash blocks allows the system to either boot from P-flash block 0 or P-flash block 1, because either block can be located at the base address 0x0000.

Swapping the flash blocks in order to boot an alternate application image, also requires having a valid flash configuration field in both the flash blocks. The flash registers will always be loaded from address 0x400–0x40F, so it is important that both the possible options for address 0x400–0x40F (the flash configuration field) must be configured with the desired options. Typically, use the same flash configuration field values for both the flash blocks; however, the user can program the two blocks with different flash configuration field settings and use the swap as a method of changing the security and/or protection settings.

3 Flash protection

The flash protection features are used to prevent accidental erase or program for regions of the flash. While the security options prevent unauthorized access of the microcontroller from external sources, the flash protection prevents writing of the flash. When a region of the flash is protected, no modifications of the contents are permitted. This includes all write accesses even those requested by firmware running inside the processor.

There are a number of applications of the flash protection but a couple of common use cases are:

- Protecting all regions of flash that contain code to protect the application itself from being overwritten. Regions of flash that are used to store data would be left unprotected.
- Protecting the vector table and a bootloader application that resides within the flash, and leaving the rest of the flash unprotected. This will prevent accidental erasure of the bootloader, but other portions of the flash remain unprotected to allow the bootloader to perform firmware updates.

3.1 Flash protection regions

The protection features are available for all of the flash regions–P-Flash, D-Flash, and even EEPROM. The number and size of the flash protection regions vary depending on the type of flash and the size of that flash block.

3.1.1 P-Flash

For P-Flash, there are always 32 protection regions controlled by the FPROT0–FPROT3 registers. The size of the protection regions will be:

(Total P-Flash size)⁄32.

Table 5 shows flash protection region sizes based on the total P-Flash size that are currently available in the Kinetis family.

Table 5. Kinetis P-Flash protection region sizes

Total P-Flash size	P-Flash protection region size
1 MB	32
512 KB	16
256 KB	8
128 KB	4
64 KB	2
32 KB	1

3.1.2 D-Flash

For devices that include FlexNVM, the D-Flash protection is controlled by the FDPROT register, where the D-Flash is the total size of the FlexNVM minus any flash that is being used to back-up enhanced EEPROM (EEE) data (E-Flash). Because the FlexNVM partition options for supporting the EEE functionality can result in some non-power of two flash sizes, there are some special cases to consider for determining the size of the flash protection regions.

- If the D-Flash size is a power of two, then there are eight protection regions of equal size. So the flash protection region size is given by:

(Total D-Flash size)⁄8

For example, while using a device with 256 KB of FlexNVM, where half of the FlexNVM is used as D-Flash and the other half is used as EEE back-up (E-Flash), there would be eight protection regions each 16 KB in size (128/8 = 16).

- If the D-Flash size is not a power of two, then the protection region size is fixed and there will be less than 8 protection regions actually used. The fixed size of the protection region can vary depending on the specific Kinetis device being used. See the specific Kinetis reference manual at http://www.freescale.com to get the correct value to use for the processor. For example, when using a 100 MHz Kinetis device with 256 KB of FlexNVM where 192 KB are used as D-Flash (64 KB used as EEE back-up), the data flash protection regions would be 32 KB in size. In this case, only 6 protection regions would be used instead of 8 (32 x 6 = 192), so FDPROT[7:6] must always be set to 1, but only FDPROT[5:0] have an effect.

3.1.3 E-Flash

For devices where FlexNVM is available and the EEE feature is being used, there are always 8 protection regions for the EEE data controlled by the FEPROT register. The size of the protection regions will be equal to the total EEPROM size divided by 8.

3.2 Configuring flash protection settings

Like the security options, the default values for all of the flash protection registers, FPROT0–FPROT3, FDPROT, and FEPROT, are determined by the application image based on the values programmed into the flash configuration field. During the reset sequence, the flash protection registers will be loaded with values from the flash configuration field. See Table 4 for the description of which flash configuration field addresses correspond to the flash protection registers.

3.3 Changing flash protection settings

The sections below describe the methods that can be used to change the flash protection settings.

3.3.1 Mass erase

Both the Debug and EzPort mass erase commands ignore flash protection settings, so mass erase can be used as long as the FSEC setting has mass erase enabled, to return the chip to the default state where none of the flash is protected.

NOTE

The flash Erase All Block command checks the flash protection settings. The command will not execute if any of the flash regions are protected. So, in order to use mass erase to clear the protection settings, the JTAG/debugger or EzPort method must be used.

3.3.2 Reprogramming the flash configuration field

As long as the region of the P-flash containing the flash configuration field is not protected, the field can be reprogrammed either using firmware or a debugger (the debugger option is available only if the device is not secured). This method changes the default setting for the flash protection bits. So, the values of the flash protection bits can be changed without restriction, that is, the status can switch from protected to unprotected. Because the change is to the default state of the protection registers and not to the registers themselves, any changes made won't go into effect until the next reset.

NOTE

On most Kinetis devices, the flash configuration field is located in the first sector of the flash. So in order to erase and reprogram the flash configuration field, the entire vector table including the initial stack pointer and PC values need to be erased.

NOTE

Keep in mind that the region of flash containing the flash configuration field can be protected. If the flash configuration field is protected, the user will not be able to erase and then reprogram the flash configuration field to change the protection settings. For this reason, it is recommended to leave mass erase enabled while using flash protection and enabling security.

3.3.3 Writing to protection registers

Unlike the FSEC register, direct writes to the flash protection registers are allowed; however, they can only be used to increase protection. It is not allowed to write to the flash protection registers to unprotect a region. The flash actually allows for this in NVM special mode, but the SoC doesn't use NVM special mode except for when the part is in EzPort mode. Also, these changes last only until the next reset. If the chip goes through a reset, then the registers will be reloaded from the flash configuration field, so at that point, any direct changes to the protection registers would be lost.

4 Tamper detection and cryptography

Some Kinetis family devices offer tamper detection (DryIce module) and/or cryptography features (CAU and RNG modules). A detailed discussion of these features is outside the scope of this application note, but other application notes are planned to discuss these modules. Please visit http://www.freescale.com/kinetis to check for the latest application notes. The tamper detection and cryptography features are useful complements to the security and protection features, and must be considered when planning out an overall strategy for security.

The CAU can be used to encrypt/decrypt data for any of the external interfaces on the chip. The security features discussed in this application note help to keep information inside the processor secure, but the cryptography blocks can be used to protect and ensure the integrity of data that is being passed to or from external devices or communication ports. The DryIce module includes tamper detection signals that can be used to determine if someone is attempting to interfere with the enclosure and/or circuitry on the PCB to gain access to the system.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: AN4507
Rev. 1, 6/2012