

Serial Bootloader for S12(X) Microcontrollers Based on 180 nm Technology

by: Lukas Zadrava and Ladislav Makovic
Technical Information Center
Roznov
Czech Republic

1 Introduction

This application note covers the operation and use of a bootloader for the S12 and S12X microcontroller families based on 180 nm technology.

The bootloader can be a convenient way to support programming during production or “ in-system ” , where support for the dedicated HCS12 Background Debug interface (BDM) may not be available. Users must pre-program the S12(X) with the bootloader during pre-production or at a programming vendor. The bootloader resides in the MCU for further use.

This bootloader implementation allows user software to be downloaded into the MCU flash memory using the SCI serial interface.

The bootloader described in this document is only an example and comes with no guarantees and no support.

2 Hardware Compatibility

There are two versions of the bootloader:

- The first one is written for S12 derivatives (S12P, S12HY, S12HA and S12G)
- The second one is intended for S12X derivatives (S12XS, S12XE and S12XF)

Contents

1	Introduction.....	1
2	Hardware Compatibility.....	1
3	Requirements.....	2
3.1	Input pin.....	2
3.2	Serial interface.....	2
3.3	Terminal emulation program.....	3
3.4	Oscillator.....	3
4	Functional Description.....	3
4.1	Operation.....	3
4.2	Interrupt vectors.....	3
4.3	Mask set and part ID.....	3
5	User Guide.....	4
5.1	Setup procedure for the S12 bootloader.....	4
5.2	Setup procedure for the S12X bootloader.....	4
5.3	Bootloader Guide.....	5
6	How to write a user application.....	7
7	How to convert S-Record that will be downloaded by the bootloader.....	8
8	How to merge the user application and bootloader	8
9	Testing.....	9
10	Content of Zip file.....	9

requirements

This application note applies to both versions unless otherwise noted.

The bootloader for S12 microcontrollers is not optimized for the smallest derivatives. Therefore, microcontroller MC9S12GN16 is not supported.

Programming of D-Flash or EEPROM memories is currently not supported.

3 Requirements

For successful operation of the bootloader it is necessary to follow a few requirements.

3.1 Input pin

There are several ways to distinguish whether to start the bootloader or user application. The selection can be made on the basis of the state of the input pin, the state of the memory cell in EEPROM memory, or on the basis of the command received via a communication interface

This bootloader uses the first approach. After reset, the bootloader enables the pull up resistor on pin PP0 and reads the status of pin PP0 to find out if the you want to execute the bootloader or the user application.

A switch, jumper, or push button with an optional pull up resistor may be placed on this pin.

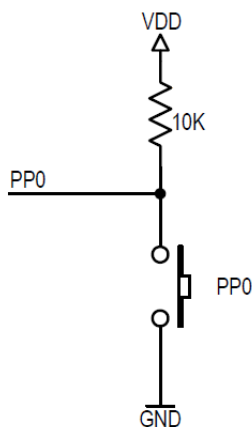


Figure 1. Pin connection

3.2 Serial interface

All S12(X) microcontrollers include an on-chip serial communication interface. Notice that the RS-232 level shifter is necessary to communicate with PC.

By default, the serial communication is set to format:

- 8-data bit
- one start bit
- one stop bit
- no parity
- Xon / Xoff flow control

Default baud rate is 9600. The baud rate can be changed via the bootloader's menu to 38400, 57600, or 115200 bps.

3.3 Terminal emulation program

Microsoft HyperTerminal or similar software can be used on a PC to communicate with a microcontroller. A terminal emulation program must support communications over serial COM ports, Xon / Xoff flow control, and it must be able to send a text file.

3.4 Oscillator

An external crystal is not necessary in the case of S12 microcontrollers. S12 microcontrollers feature an internal 1 MHz RC oscillator used by the bootloader. The bootloader sets the bus clock frequency to 25 MHz to allow high serial communication speeds.

S12X microcontrollers require a crystal or external oscillator for its operation. Configure the PLL in the bootloader to reach the bus clock frequency of 40 MHz. This is to allow high serial communication speeds that increase code downloading.

4 Functional Description

The operation of the bootloader is straightforward. This section describes only the most important and specific points.

4.1 Operation

The bootloader handles all reset vectors. After reset, the bootloader's startup routine is called. As a first step, the bootloader reads the PP0 pin status. If the value of the pin is logical zero, the bootloader starts its operation. If the value of this pin is a logical one, the user application's startup routine is then called. If the reset vector of the user application is not available (word at address 0xEFFE-0xEFFF is in erased state) then the bootloader is executed in any case. The user can rewrite this code to start the bootloader or user application upon another condition.

4.2 Interrupt vectors

In case the user application is using interrupts, it is necessary to relocate the interrupt vector table using the IVBR register. The bootloader resides in the upper fixed block at address 0xF000-0xFFFF. This area is protected, so the user application cannot rewrite a default location of the interrupt vector table at 0xFF10-0xFFFF. The advantage of this solution is that the bootloader cannot be affected by power failure that could occur when rewriting interrupt and reset vectors.

4.3 Mask set and part ID

Each silicon mask set has a unique part ID, located in two 8-bit registers, PARTIDH and PART IDL at addresses 0x001A and 0x001B.

When the s-record is going to be programmed to the flash memory, the bootloader checks if the addresses fall within the physical flash memory. An error is returned in case of invalid address.

The list of the currently supported mask sets and part IDs can be found in the file PartID.h in the bootloader project.

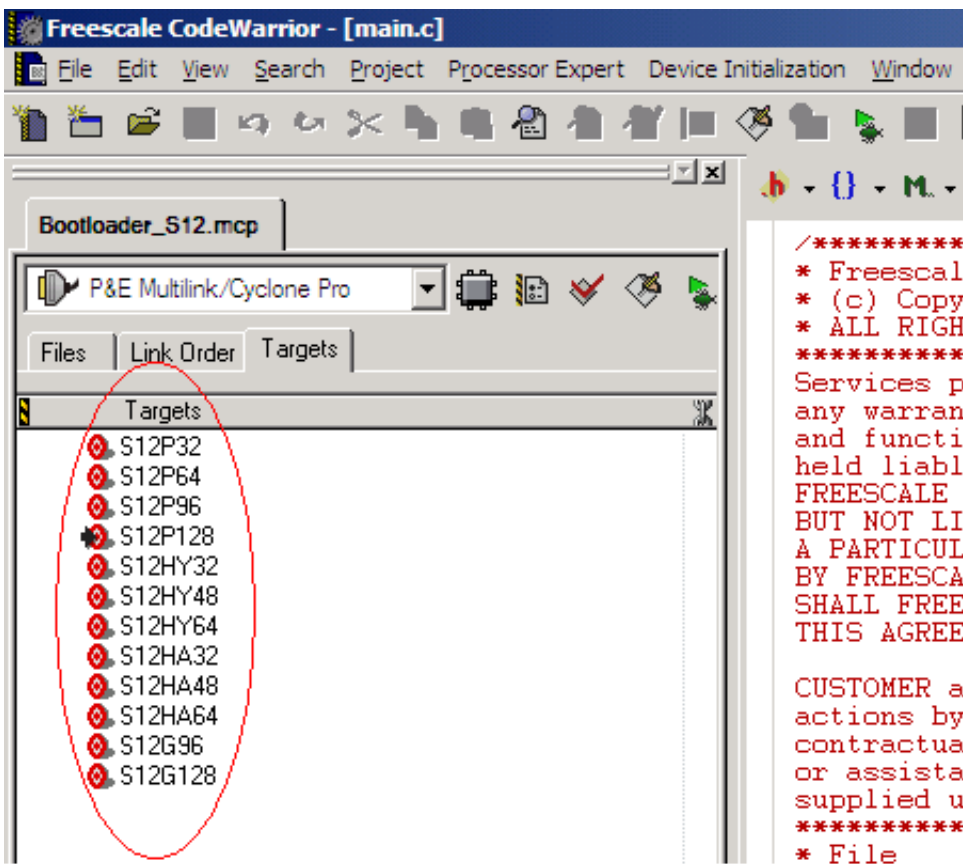
5 User Guide

This section describes the step-by-step procedure to use the bootloader.

5.1 Setup procedure for the S12 bootloader

1. Open the Bootloader_S12 project in CodeWarrior Development Studio for the S12(X) v5.x. The bootloader was tested in current versions v5.0 and v5.1.
2. Select the target as shown in [Figure 2](#).

Figure 2. Target selection S12

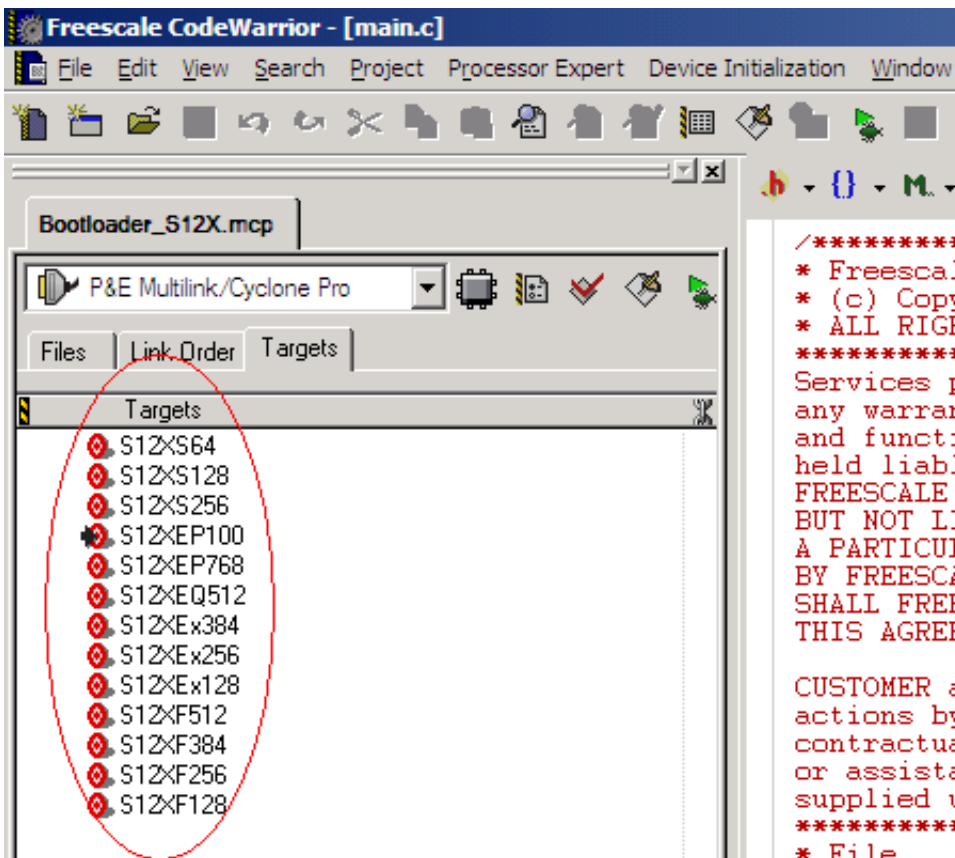


3. The decision to start the bootloader or the user application is made on the basis of pin PP0. If the PP0 is low then the bootloader is started, if the PP0 is high then user application will start. If this default setting is not suitable, the user can rewrite this code in the file Start12.s.

5.2 Setup procedure for the S12X bootloader

1. Open the Bootloader_S12X project in CodeWarrior Development Studio for the S12(X) v5.x. The bootloader was tested in current versions v5.0 and v5.1.
2. Open the file Config.h and set the following:

- Set the FLASH_PRESCALER (value loaded to the FCLKDIV register) to achieve the flash operating frequency 800—1050 kHz as per the reference manual.
 - Set the PLL to achieve bus frequency 40 MHz. The PLL calculator attached to this application note can be used.
3. Select the target as shown in [Figure 3](#)

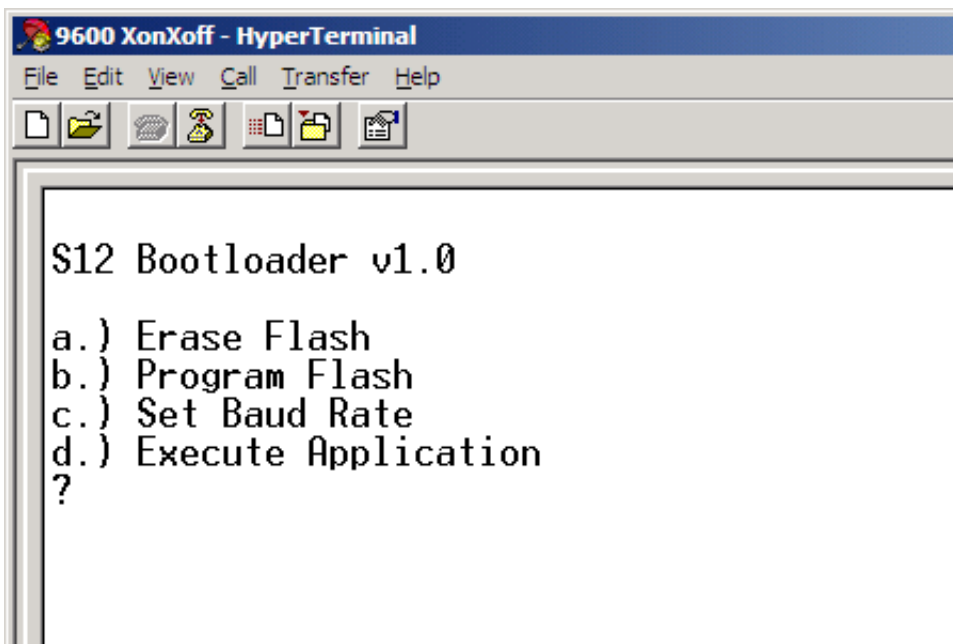
Figure 3. Target selection S12X


4. The decision to start the bootloader or the application is made on the basis of pin PP0. If the PP0 is low then the bootloader is started. If the PP0 is high then the user application will be executed. If this default setting is not suitable, the user can rewrite this code in file StartS12X.s.

5.3 Bootloader Guide

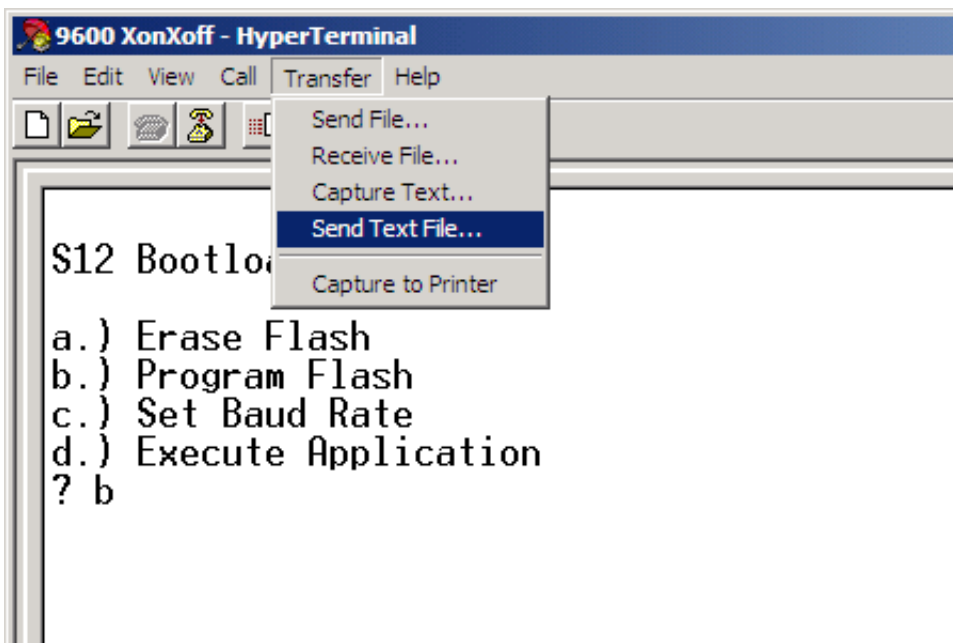
1. Compile the project and download the bootloader to the MCU via BDM device.
2. Open the Microsoft HyperTerminal or a similar utility. Set the baud rate 9600 bps, 1 start bit, 8 data bits, 1 stop bit, and flow control Xon / Xoff.
3. Make sure there is a serial cable connection between the PC and the board.
4. Hold the pin PP0 low and reset the MCU.
5. The bootloader is started and the following response is received in HyperTerminal. See [Figure 4](#).

Figure 4. Initial screen



6. Type “ a ” to erase the flash memory. This is not necessary the first time because the flash has been already erased by the BDM device.
7. Type “ b ” to program the flash.
8. Now send the desired S-record as a text file — See Figure 5. Browse for S-record which is downloaded to the MCU. For testing purposes, use demo S-records that are attached to this application note. The S-records must have a specified format. Refer to Chapter 6, "[How to write a user application](#)" and to Chapter 7, "[How to convert S-Record that will be downloaded by the bootloader](#)".

Figure 5. Send text file



9. Confirm the dialog and the S-record are downloaded to the MCU. One printed star (*) means that one line of S-record has been programmed. See [Figure 6](#).

Figure 6. Downloading user application

```

9600 XonXoff - HyperTerminal
File Edit View Call Transfer Help

S12 Bootloader v1.0
a.) Erase Flash
b.) Program Flash
c.) Set Baud Rate
d.) Execute Application
? b
*****
Downloaded successfully!
a.) Erase Flash
b.) Program Flash
c.) Set Baud Rate
d.) Execute Application
? _
    
```

6 How to write a user application

You must ensure that the user application does not interfere with bootloader area 0xF000-0xFFFF.

1. Create a new project in CodeWarrior Development Studio for Freescale HCS12(X) Microcontrollers.
2. Open the .prm file
3. Trim the segment ROM_C000 from the original size 0xC000–0xFEFF to 0xC000–0xEFDF. This is because the area 0xF000–0xFFFF is occupied by the bootloader and the area at address 0xEFE0–0xEFFF will be used for the user application reset vector.

If interrupts are used:

4. Trim the segment ROM_4000 from the original size 0x4000–0x7FFF to 0x4000–0x7F0F. The area 0x7F10–0x7FFF will be used for the relocated interrupt vector table.
5. Create a vector table as shown in the attached demo applications and set the IVBR register accordingly. The IVBR sets the interrupt vector table base address and in this case must be set to 0x7F.

7 How to convert S-Record that will be downloaded by the bootloader

The bootloader accepts s-records with global (linear) addressing. All records must be aligned to 32 bytes and the length of records must be 32 bytes.

1. Open the SRecCvt utility (SRecCvt-GUI.exe).
2. Select the device MC9S12... (it depends on used derivative), select Flash memory, and select Convert File operation.
3. Select input file format as Banked and output file format as Linear.
4. Make sure the S-record size is set to 32.
5. Browse for input file (S19 file generated by CodeWarrior) and select the destination of the output file.
6. Click on Convert.

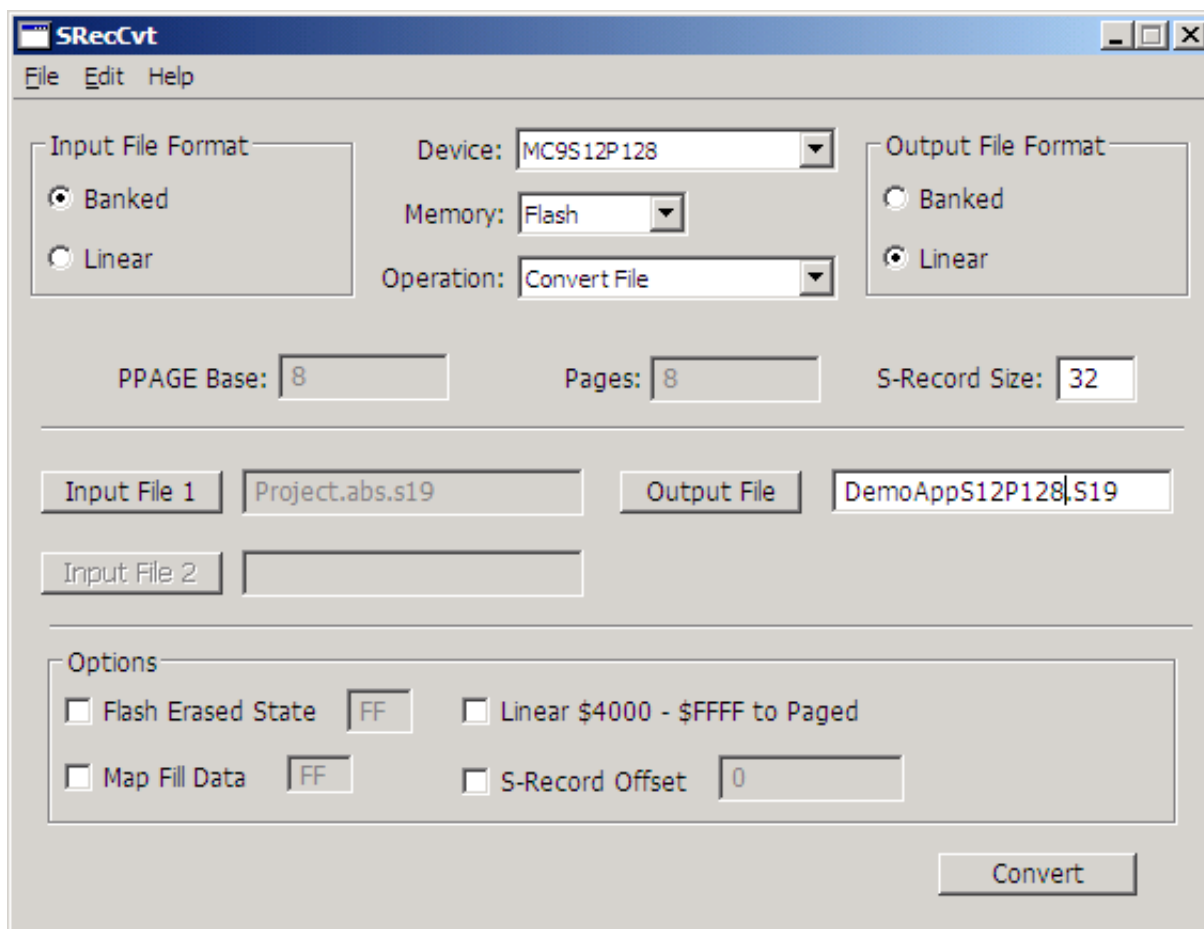


Figure 7. SRecCvt

8 How to merge the user application and bootloader

The user application can be developed independently, that is without the bootloader. The user application can be loaded into the microcontroller and can be debugged directly via the BDM device. However, for production purposes it is worth merging the user application and bootloader together, so it can be downloaded into the microcontroller all at once as a single s-record.

This is a recommended procedure:

1. Open the user application that was created as per Chapter 6, "How to write a user application" in the CodeWarrior Development Studio for the S12(X) v 5.x.
2. Place a user application reset vector at address 0xEFFE, so the bootloader can use this vector. Copy the following three lines to main.c:
 - extern void near _Startup(void)
 - typedef void (*near tFunc)(void)
 - const tFunc rst_vec @0xEFFE = _Startup
3. Remove the original user application reset vector.
Comment out this line in Project.prm file:
`//VECTOR 0 _Startup`
4. Copy the ready-made bootloader .s19 file to ..\user_application_project\bin. For example, the s-record file can be renamed to the bootloader.s19.
5. Link this file to the user application.
Use this command at the beginning of Project.prm:
`HEXFILE bootloader.s19`
6. Add the following command into the file ..\user_application_project\cmd\P&E_Multilink_CyclonePro_Preload.cmd:
`FLASH NOUNSECURE`

This command ensures that the burner will not automatically change the flash security byte at address 0xFF0F to an unsecured state (0xFE) during MCU programming, therefore this phrase (0xFF08-0xFF0F) can be loaded with values defined in the bootloader. If this command is not used, it will lead to an ECC error on this phrase, the chip will be secured and the flash will be fully protected.

7. Make the project. The final s-record is ready to be downloaded into the microcontroller by the BDM device.

9 Testing

The bootloaders were tested on demoboards:

- DEMO9S12PFAME
- DEMO9S12HY64
- TWR-S12G128
- DEMO9S12XSFAME
- EVB9S12XEP100

10 Content of Zip file

All mentioned projects and utilities can be found in a zip file associated with this application note:

- Bootloader_S12 — Bootloader project for S12 microcontrollers.
- Bootloader_S12X — Bootloader project for S12X microcontrollers.
- Demo_applications — Projects that show how to write user applications.
- Converted_srecords — S-Records taken from Demo_applications that have been covered by SRecCvt utility. These S-Records can be downloaded into the MCU by the bootloader.
- Demo_applications_with_linked_bootloader — User applications that include bootloader.
- SRecCvt — Utility for S-Record converting.
- S12XE_PLL_Calculator — Utility that helps to set the PLL module.

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

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
 1-800-441-2447 or +1-303-675-2140
 Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2011 Freescale Semiconductor, Inc.