Freescale Semiconductor
Application Note

AN2992
Rev. 0, 04/2006

Interfacing the MC9S12XDP512V2 Microcontroller to the MFR4200 FlexRay Controller

by: Alan Devine
Body Systems, East Kilbride

1 Introduction

Freescale Semiconductor offers a range of standalone FlexRay communication controllers (MFR4xxx) that can be easily interfaced to 16-bit and 32-bit microcontrollers (MCU). This application note describes the hardware, software, and timing considerations necessary for reliable communication between the MFR4200 controller and the MC9S12XDP512V2 (S12X). The technical information is taken from the current device data sheets. See references 1 and 2 for details.

2 Objective

This document aims to demonstrate the simplicity of the hardware interface between the MFR4200 and the S12X and provide an example of the software used to configure the S12X for operation. The information contained can help a designer to quickly create a fully functional FlexRay node based on the S12X family of MCUs. Freescale hardware and software is available to enable development of FlexRay applications (see http://www.freescale.com/flexray for details).

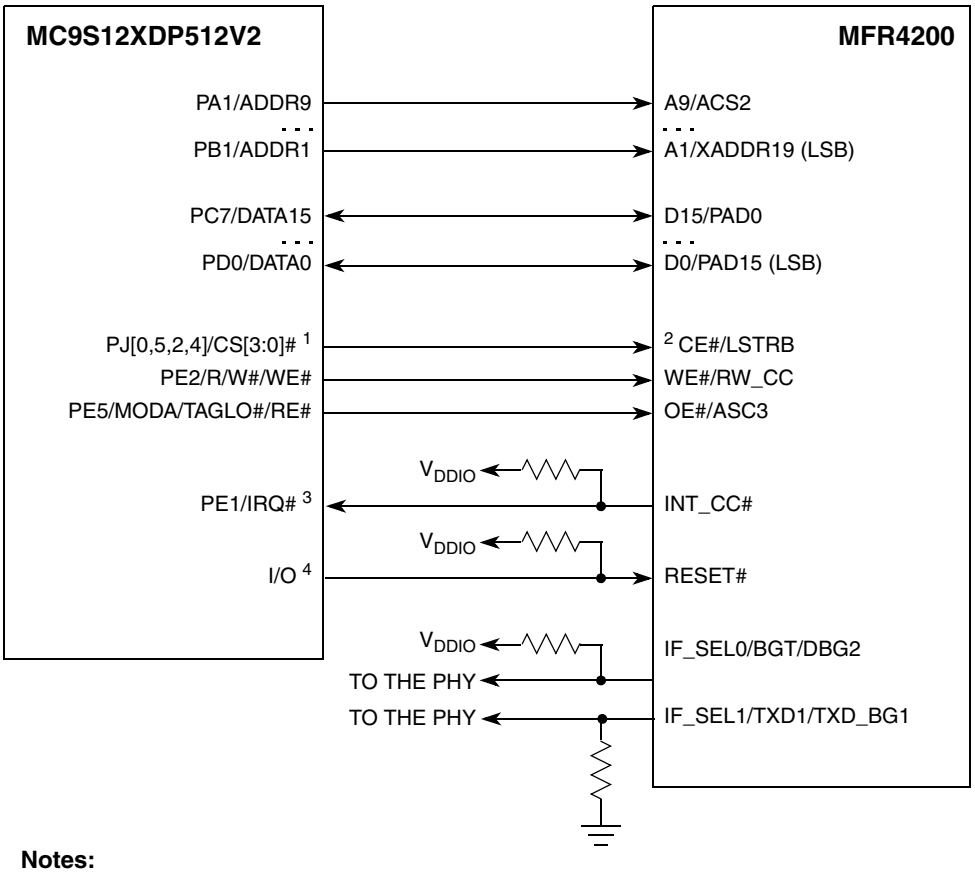
Table of Contents

Note: In this document, active-low signals are indicated by a “#” at the end of the signal name, e.g. “IRQn#”.

freescale™ semiconductor

3 Hardware Design

The S12X interfaces with the MFR4200 via the external bus interface (EBI)[1]. On the S12X the EBI provides individual address, data and control signals[2], which can be connected directly to an MFR4200 configured in its asynchronous memory interface mode (AMI). See Section 3.1, "Selecting the AMI Mode" for details. In this mode, devices can be connected together without the need for additional glue logic, thereby simplifying the design and reducing the system cost. Figure 1 shows a block diagram with all the EBI signals connected; Table 1 provides a description of each signal.



Notes:
1. Correct chip select signal must by connected. See Section 5.2 for details.
2. CE# must be tied low at the MFR4200 if the bus frequency is greater than 16.67 MHz. See Section 4 for details.
3. The S12X also has a non maskable interrupt (PE0/XIRQ) that could be used. It could also be possible to remove the pullup resistor on the IRQ# line as the S12X has an internal pullup.
4. Any output of the S12X can be used to drive the MFR4200 RESET# line, including the main ECU reset circuit.

Figure 1. MC9S12XDP512V2 to MFR4200 Connection Diagram

1. Not all members of the S12X family have an EBI. It is possible to connect the MFR4200 only to an S12X MCU that has an EBI bonded out.
2. If the MFR4200 is interfaced with the S12 family of MCUs, the MFR4200 must be configured in the HCS12 interface mode, as the S12 EBI has a multiplexed address and data bus and different control signals.

Table 1. Interface Signal Description — MC9S12XDP512V2 to MFR4200 Connection

MC9S12XDP512V2		MFR200		Comments
Signal	**PIN**	**Signal**	**PIN**	
PC7/DATA15	43	D15/PAD0	10	AMI data bus
PC6/DATA14	42	D14/PAD1	7	AMI data bus
PC5/DATA13	41	D13/PAD2	6	AMI data bus
PC4/DATA12	40	D12/PAD3	5	AMI data bus
PC3/DATA11	31	D11/PAD4	4	AMI data bus
PC2/DATA10	30	D10/PAD5	3	AMI data bus
PC1/DATA9	29	D9/PAD6	2	AMI data bus
PC0/DATA8	28	D8/PAD7	62	AMI data bus
PD7/DATA7	86	D7/PAD8	61	AMI data bus
PD6/DATA6	85	D6/PAD9	58	AMI data bus
PD5/DATA5	84	D5/PAD10	57	AMI data bus
PD4/DATA4	83	D4/PAD11	56	AMI data bus
PD3/DATA3	68	D3/PAD12	55	AMI data bus
PD2/DATA2	67	D2/PAD13	51	AMI data bus
PD1/DATA1	66	D1/PAD14	40	AMI data bus
PD0/DATA0	65	D0/PAD15	39	AMI data bus (LSB)
PA1/ADDR9	74	A9/ACS2	22	AMI address bus
PA0/ADDR8	73	A8/ACS1	21	AMI address bus
PB7/ADDR7	39	A7/ACS0	18	AMI address bus
PB6/ADDR6	38	A6/XADDR14	17	AMI address bus
PB5/ADDR5	37	A5/XADDR15	15	AMI address bus
PB4/ADDR4	36	A4/XADDR16	14	AMI address bus
PB3/ADDR3	35	A3/XADDR17	13	AMI address bus
PB2/ADDR2	34	A2/XADDR18	12	AMI address bus
PB1/ADDR1	33	A1/XADDR19	11	AMI address bus (LSB)
PE1/IRQ#	71	INT_CC#	64	Controller interrupt output
GPIO or RESET#		RESET#	16	Controller hardware reset input
PJ[0,5,2,4]/CS[3:0]#	24, 130, 5, 131	CE#/LSTRB	29	AMI chip select. Connect appropriate CS# for desired address range. See Section 3.2, “Interfacing MC9S12XDP512V2 to MFR4200”. If bus frequency is >16.67MHz, CE# at MFR4200 must be tied low. See Section 4, “Timing Considerations” for details.
PE2/WE#/R/W#	70	WE#/RW_CC#	30	AMI write enable
PE5/MPDA/TAGLO#/RE#	50	OE#/ASC3	27	AMI read output enable
		BGT/DBG2/IF_SEL0	32	MFR4200 host interface selection0
		TXD1/TXD_BG1/IF_SEL1	41	MFR4200 host interface selection1

3.1 Selecting the AMI Mode

The controller host interface (CHI) on the MFR4200 can be configured into one of two modes: the asynchronous memory interface mode (AMI), where the MFR4200 is configured as an asynchronous memory slave enabling fast interface with a variety of MCUs; and the HCS12 mode, which is used for interfacing to the HCS12 family of MCUs. In this application, the MFR4200 must be configured into AMI mode, which is selected by pulling the IF_SEL0 pin high using a 16 kΩ pullup resistor and the IF_SEL1 pin low using a 47 kΩ pulldown resistor.

3.2 Interfacing MC9S12XDP512V2 to MFR4200

The S12X has several modes of operation. Refer to the Memory Mapping Control (S12XMMCV2) chapter of the S12X data sheet (reference 1) for details. The S12X must be configured into normal expanded mode with 16-bit word access enabled, to be compatible with the MFR4200 AMI mode. In this mode, ports K, A, and B are configured as a 23-bit address bus, ports C and D are configured as a 16-bit data bus, and port E provides bus control and status signals. The fastest external bus rate is divide by two from the internal bus rate.

NOTE

The AMI interface supports word (16-bit) accesses only.

4 Timing Considerations

For the S12X and the MFR4200 communication controller to communicate reliably, the timing between the S12X EBI and the MFR4200 must be matched. It can be seen from the respective timing diagrams (see reference 2) that read and write accesses to the MFR4200 are slower than are possible on an S12X running at 40 MHz (full speed), and require the addition of stretch cycles[1] to match the timing characteristics. For example, the read cycle time (t_{RC}) of the MFR4200 device is 155 ns (minimum), whereas the minimum read cycle time of the S12X running at full speed is 50 ns (minimum). Thus, a minimum of six stretch cycles must be inserted. Table 3 and Table 4 summarize the Read and write timing and the number of stretch cycles required for successful operation. The number of stretch cycles that can be added is controlled using the external bus control register. Refer to the S12XEVBIV2 chapter of the S12X data sheet (reference 1). The available options are shown in Table 2:

1. When accessing on-chip peripherals and memories the S12X performs 8-bit and 16-bit core accesses in a single cycle. However, when the core accesses locations on the external bus using the expanded modes, the accesses are stretched and take more than a single cycle to complete. The minimum amount of stretching is one additional bus cycle, but can be increased. See Table 2 for details.

Table 2. Stretch Cycle Selection

External Access Stretch Bit Definition	
EXSTR[2:0]	EWAIT = 0
000	1 cycle
001	2 cycles
010	3 cycles
011	4 cycles
100	5 cycles
101	6 cycles
110	7 cycles
111	8 cycles

NOTE

EWAIT = 1 enables the external wait functionality. Refer to the S12XEBIV2 chapter in the S12X data sheet (reference 1) for details.

Table 3. S12X to MFR4200 Read Cycle Summary

MFR4200		S12X EBI Timing (ns)		Comments
Description	**Timing (ns)**	**40 MHz Bus, 1 Stretch Cycle (Default)**	**40 MHz Bus, 6 Stretch Cycles**	
Read Cycle Time (t_{RC})	155 min	50 min (1)	175 min (1)	Default S12X EBI incompatible. Minimum of 6 stretch cycles required for 40 MHz bus operation.
Address Setup (t_{SAR})	5 min	5 min (2)	5 min (2)	Default is OK
Address Hold (t_{HAR})	50 min	<50 min as t_{RC} is 50 ns	>125min	Default S12X EBI incompatible. However, OK with the minimum of 6 stretch cycles required for a read operation at 40 MHz.
OE# low to Data Valid (t_{DOE})	145 max	11 min (8)	11 min (8)	Default is OK
OE# high (t_{HOE})	30 min	12.5 min	12.5 min	Possible incompatibility during consecutive reads at 40 MHz. Driver software must ensure that timing is met. E.g. insert NOP between consecutive reads
OE# Low (t_{LOE})	150 min	35 min (3)	155 min (3)	Default S12X EBI incompatible. However, OK with the minimum of 6 stretch cycles required for a read operation at 40 MHz.
OE# Low to low Z (t_{LZOE})	20 min	OK	OK	Default is OK
OE# High to high Z (t_{HZOE})	15 max	0 (7)	0 (7)	Possible incompatibility during consecutive accesses at 40 MHz. Driver software must ensure that timing is met. E.g. insert NOP between consecutive accesses
OE# High to CE# High (t_{OEH})	0 min	0	0	Default is OK
WE# High to OE# Low (t_{WEOE})	80 min	25 min	25 min	Possible incompatibility during consecutive accesses at 40 MHz. Driver software must ensure that timing is met. E.g. insert NOP between consecutive accesses

Note: The number in () refers to timing diagrams in the MFR4200 Data Sheet (reference 2).

Table 4. S12X to MFR4200 Write Cycle Summary

MFR4200		S12X EBI Timing		Comments
Description	**Timing (ns)**	**40 MHz Bus, 1 Stretch Cycle (default)**	**40 MHz Bus, 2 Stretch Cycles**	
Write Cycle Time (t_{WC})	50 min	50 min (1)	75 min (1)	Default is OK
Address Setup (t_{SAW})	30 min	28 min (4+5)	53 min	Default S12X EBI incompatible. Minimum of 2 stretch cycles required for 40 MHz bus operation.
Address Hold (t_{HAW})	5 min	8 min	8 min	Default is OK
CE# Low to write end (t_{SCE})	50 min	28 min (4+5)	53 min	Default S12X EBI incompatible. Minimum of 2 stretch cycles required for 40 MHz bus operation.
Data setup to write end (t_{SD})	30 min	31 min (10)	56 min (10)	Default is OK
Data hold from write end (t_{HD})	5 min	8 min (11)	8 min (11)	Default is OK
WE# Pulse Width (t_{PWE})	30 min	23 min (5)	48 min	Minimum of 2 stretch cycles required for 40 MHz operation.
WE# high time (t_{WEH})	55 min	25 min	25 min	Possible incompatibility during consecutive writes at 40 MHz. Driver software must ensure that timing is met. E.g. insert NOP between consecutive writes
Write end to CE# high (t_{CEWE})	30 min	12.5 min	12.5 min	CE# must be tied low at the MFR4200 if operating at f_{bus}>16.67 MHz.
OE# High to WE# Low (t_{OEWE})	15 min	12.5 min	12.5 min	Possible incompatibility during consecutive accesses at 40 MHz. Driver software must ensure that timing is met. E.g. insert NOP between consecutive accesses

Notes:
1. The number in () refers to timing diagrams in the MFR4200 Data Sheet (reference 2).
2. The two stretch cycles that are required for a write cycle are a minimum. In fact, the EBI must be configured with six stretch cycles to be compatible with a read cycle

Table 3 and Table 4 highlight that stretch cycles are required to interface to the MFR4200 when the S12X is running at 40 MHz. During a read cycle, six stretch cycles are required; during a write cycle, two stretch cycles are needed. Two further potential incompatibilities between the S12X and MFR4200 that can not be solved by simply adding stretch cycles are highlighted

The first problem is that the write end to CE# high (t_{CEWE}) timing on the MFR4200 is specified at 30 ns (minimum), whereas 12.5 ns is all that can be guaranteed by the S12X[1]. This problem can be solved by tying the CE# signal of the MFR4200 permanently low when the bus frequency is greater than 16.67 MHz. This solution is suitable when the MFR4200 is the only external peripheral connected to the bus. In the case where more peripherals are connected, additional glue logic is required.

1. WE# is de-asserted with the last falling edge of the clock before the end of the cycle (i.e. 12.5 ns +/- before the end of the cycle), and CE# is de-asserted with the rising edge of the clock at the end of the cycle. Therefore, WE# high to CE# high is a minimum of 12.5 ns +/-.

The second problem is that the pause timing between consecutive read/write operations must always be met. These timings can be met by controlling the delay during consecutive access; for example, with the insertion of NOPs in the driver code. Refer to the comments in Table 3 and Table 4 for the number of NOPs to be inserted for each situation. A further explanation is given in the following example.

Example code for back-to-back read:

OE# high (t_{HOE}) should be a minimum of 30 ns, but, with the S12X running at 40 MHz, the minimum guaranteed time is 12.5 ns. Thus, a single NOP (one cycle delay) must be inserted between consecutive read instructions. This provides a minimum OE# high time of 12.5 + 25 = 37.5 ns.

e.g.

```
gldd MFR4200Reg0

nop - (Inserts 25ns delay when bus is operating @40MHz)

gldd MFR4200Reg1
```

5 Software

The software setup is concerned mainly with selection of the correct MCU operating mode, configuration of the S12X EBI for communication with the MFR4200, and linking into the correct area of the memory map.

5.1 Mode Configuration and EBI operation

As already stated, the S12X must be configured into normal expanded mode to enable the EBI. In this mode, ports K, A and B are configured as the address bus, ports C and D are configured as the data bus, and port E is configured as the control signals. In this example, the MCU is configured to enter normal single chip mode or special single chip mode (during debugging), when taken out of reset. During initialization, the EBI is configured and the mode is changed to normal expanded mode by writing MODC = 1, MODB = 0, and MODA = 1, in the MODE register. The EBI must be initialized to use the correct chip select and to select the correct number of stretch cycles, which depends on the internal bus frequency. See the MFR4200 Data Sheet (reference 2) for specific details. The flow diagram in Figure 2 shows the steps required to configure the MCU for communication with the MFR4200 controller.

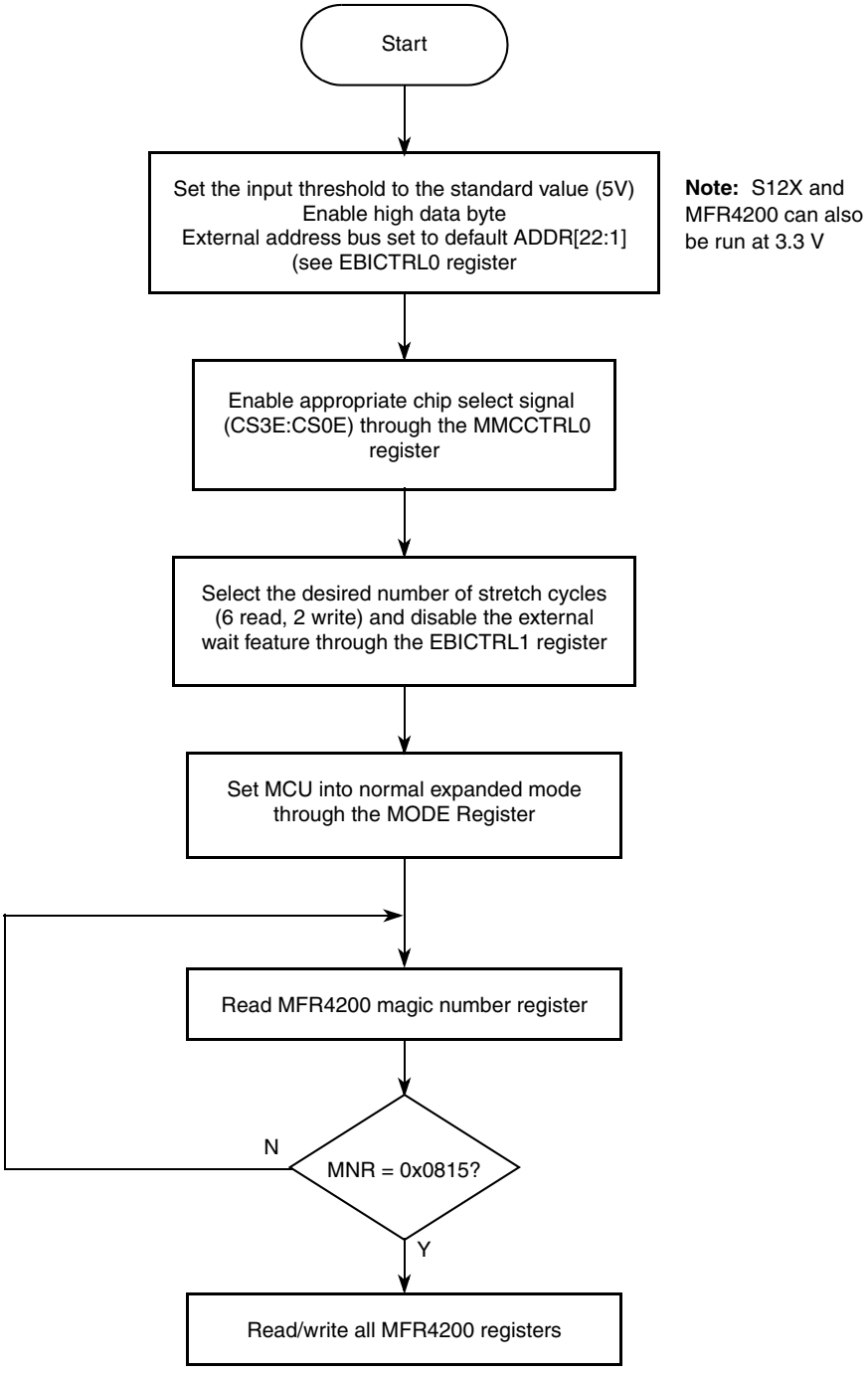


Figure 2. Initialization Flow Diagram

5.2 Chip Select Configuration

The MFR4200 is mapped into the external address space of the S12X and accessed via the EBI. The EBI on the S12X has four external chip select control signals CS[0:3]# that are associated with a specific global address range and are enabled through the S12X_MMC control register (MMCCTL0). See Table 5 for details. The chip select signal used determines the external address space that the MFR4200 occupies. The memory map of the MFR4200 controller must be linked into the correct address range of the particular chip select signal for correct operation. See Section 5.4, "Placement of MFR4200 into Memory Map" for details.

Table 5. Chip Select Mapping

Global Address Range	Asserted signal
$00_0800 - $0F_FFFF	CS3#
$10_0000 - $1F_FFFF	CS2#
$20_0000 - $3F_FFFF	CS1#
$40_0000 - $7F_FFFF	CS0#

Any of the chip selects can be used to control the MFR4200. The example in this application note uses CS2#. To use a different chip select control signal, the software must be linked to the correct global address range.

5.3 Magic Number Register

During the internal initialization procedure, after leaving hard reset, the CPU must not access any of the MFR4200 registers except the magic number register (MNR), which acknowledges the finish of the internal initialization procedure.

The MNR contains 0x0000 while the controller is initializing. Once initialization is complete, the MNR contains the value 0x0815. The initialization takes 1025 communication controller clock cycles after negation of hard reset.

5.4 Placement of MFR4200 into Memory Map

The MFR4200 device must be mapped into the S12X external address space and associated with the selected chip select signal (see Table 5[1]). The global address, associated with each chip select signal, is generated by a concatenation of the S12X local address [15:0] with the global page index register, where the page index bits are effectively used to select which of the 128 x 64 Kbyte pages is to be accessed. To access global addressing, specific global instructions must be used; for example:

GLDD – Load double accumulator D (A:B) from global memory.

GSTAA – Store the contents of accumulator A into global memory

1. The chip select signals are not used if the chip enable of the MFR4200 is tied permanently low, e.g. when the S12X bus speed is > 16.67 MHz.

In CodeWarrior, the FAR qualifier is used to tell the compiler to use the global instructions included in the S12X CPU. An example of how to place the MFR4200 registers into the memory map is shown below.

Example:

```
#define         BASE_ADDRESS 0x140000
#define         MNR     (*(volatile unsigned int* far)(0x000 + BASE_ADDRESS)
#define         MVR     (*(volatile unsigned int* far)(0x002 + BASE_ADDRESS)
#define         MCR0    (*(volatile unsigned int* far)(0x004 + BASE_ADDRESS))
```

In this option, the placement of the memory map can be changed by modifying the BASE_ADDRESS. The selected BASE_ADDRESS of 140000 is in CS2# address range.[1]

6 Conclusions

The FlexRay controller can be connected to the S12X family of MCUs; however, certain timing restrictions must be followed for successful operation. Due to the enhanced EBI on the S12X and the AMI mode option on the MFR4200, no glue logic is required. Software configuration is also straightforward, the peripheral being simply memory-mapped into the global address space.

Freescale Semiconductor offers comprehensive FlexRay support and products. The latest additions are the MFR4300 communication controller and the first FlexRay integrated IP on the MC9S12XFR128. These new products implement version 2.1 of the protocol. More information can be found at http://www.freescale.com/flexray.

7 References

1. MC9S12XDP512V2 Data Sheet (MC9S12XDP512V2)
2. MFR4200 FlexRay Communication Controller Data Sheet (MFR4200)

These documents are available on the Freescale Semiconductor web site at http://www.freescale.com.

More information on FlexRay and FlexRay products can be found at http://www.freescale.com/flexray

1. Several alternative methods exist for placing the MFR4200 registers in the memory map.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

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 Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN2992
Rev. 0, 04/2006

freescale™
semiconductor