Freescale Semiconductor
Application Note

Document Number: AN2845
Rev. 0, 04/2005

Using the Break Controller (BC) eTPU Function

Covers the MCF523*x*, MPC5500, and all eTPU-equipped Devices

by: Milan Brejl
System Application Engineer, Roznov Czech System Center
Valeriy Philipov
System Application Engineer, Kiev Embedded Software Lab

Table of Contents

1 Introduction

The break controller (BC) enhanced time processor unit (eTPU) function is one of the functions included in the DC motor control eTPU function set (set3) and AC motor control eTPU function set (set4). This application note is intended to provide simple C interface routines to the BC eTPU function. The routines are targeted at the MCF523*x* and MPC5500 families of devices, but they could be easily used with any device that has an eTPU.

2 Function Overview

The BC function is useful for controlling the DC-bus break. The DC-bus break is required when a motor is driven, not only in the motoring mode, but also in the generating mode.

freescale semiconductor

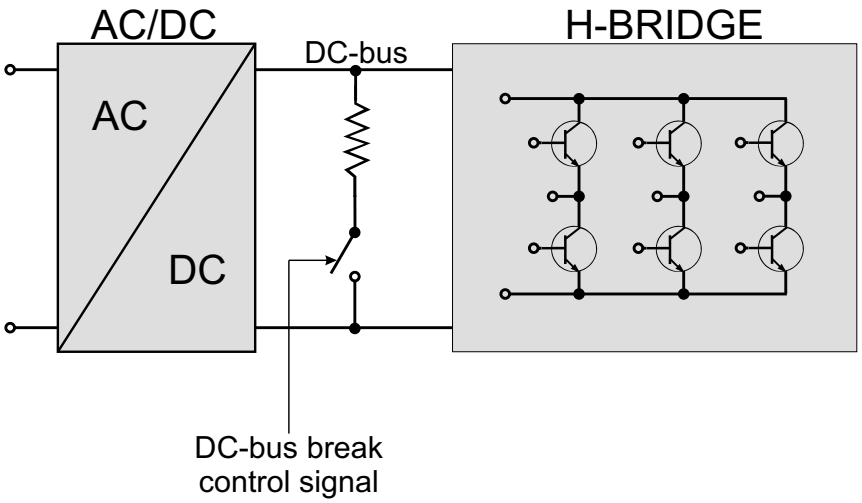

Figure 1. DC-Bus Break

3 Function Description

The BC function generates the DC-bus break control signal (see Figure 1), based on the actual DC-bus voltage. The BC function can operate in the following switching modes:

- ON/OFF Mode.

  In this mode, the BC function turns the control signal on when the DC-bus voltage exceeds the *u_dc_bus_ON* threshold, and turns the control signal off when the DC-bus voltage falls behind the *u_dc_bus_OFF* threshold. See Figure 2.

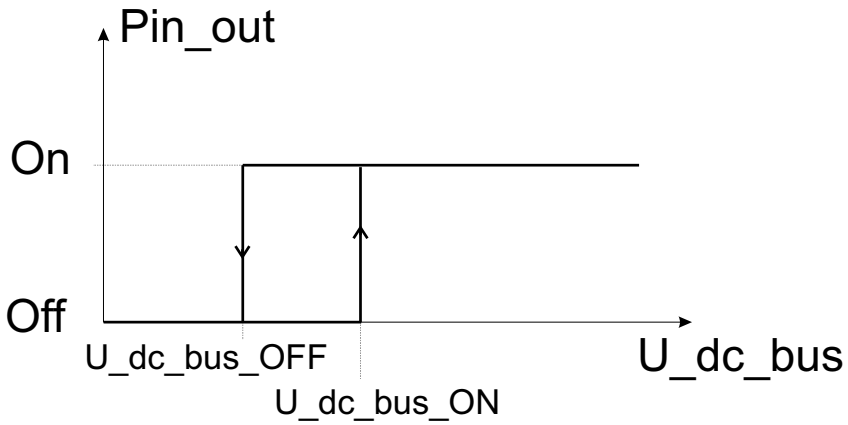

Figure 2. ON/OFF Mode

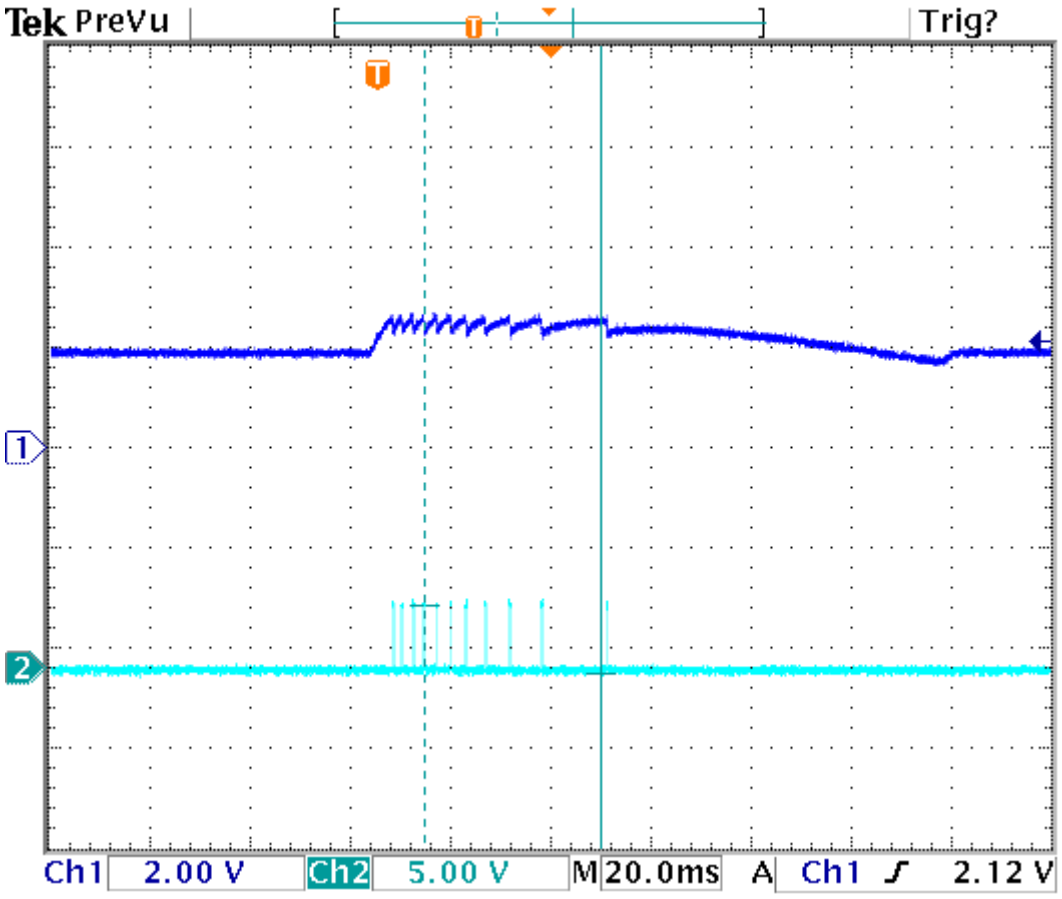

Figure 3. DC-bus Voltage (1) and BC Output (2) in On/off Mode

- PWM Mode.

  In this mode, the BC function switches softly using a PWM signal. The *u_dc_bus_ON* and *u_dc_bus_OFF* thresholds define a ramp (see Figure 4). When the DC-bus voltage is lower then *u_dc_bus_OFF,* the control signal is turned off. Between the *u_dc_bus_OFF* and *u_dc_bus_ON* thresholds, a PWM signal with duty-cycle linearly increasing from 0% to 100% is generated. Above the *u_dc_bus_ON* threshold, the control signal is turned on.

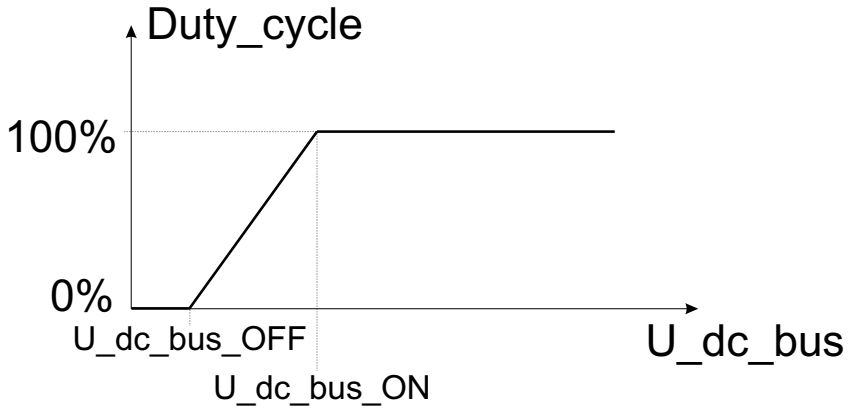

Figure 4. PWM Mode

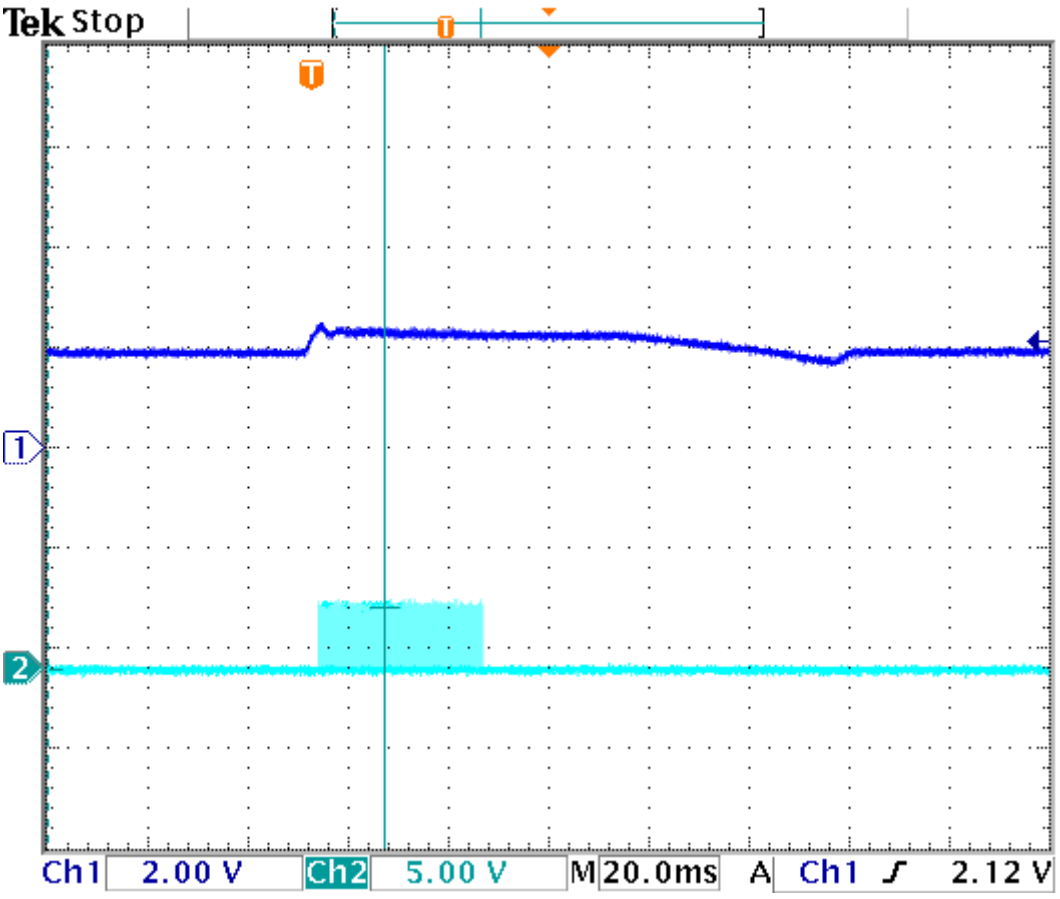

Figure 5. Dc-bus Voltage (1) and Bc Output (2) in Pwmmode

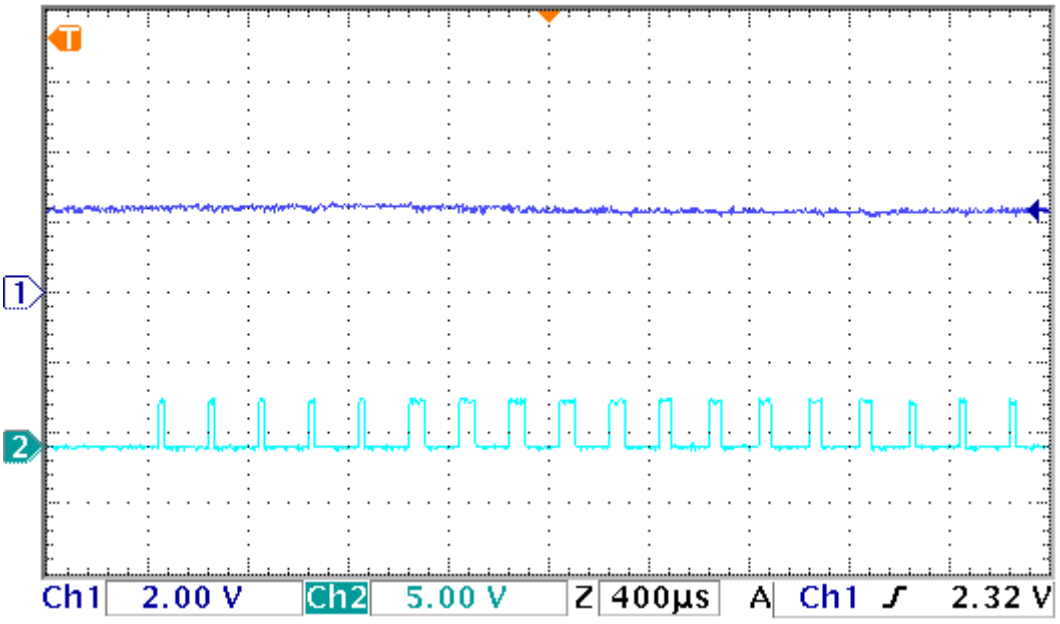

Figure 6. DC-bus Voltage (1) and BC Output (2) in PWMmode - Detail

The BC function update, in which the actual DC-bus voltage value is taken and the control signal is adjusted, can be executed periodically, or by another process:

- **Master Mode**

  The BC update is executed periodically with a given period. The ON/OFF switching mode only is applicable.
- **Slave Mode**

  The BC update is executed by the analog sensing (ASDC or ASAC) eTPU function, other eTPU function, or by the CPU. Both PWM and ON/OFF switching modes are applicable.

3.1 Interrupts

The BC function generates an interrupt service request to the CPU every n-th update. The number of updates, after which an interrupt service request is generated, is a function parameter.

3.2 Performance

Like all eTPU functions, the BC function performance in an application is to some extent dependent upon the service time (latency) of other active eTPU channels. This is due to the operational nature of the eTPU scheduler.

The influence of the BC function on the overall eTPU performance can be expressed using the following parameters:

- **maximum eTPU busy-time during one update period**
  This value, compared to the update period value, determines the proportional load on the eTPU engine caused by the BC function.

Table 1 lists the maximum eTPU busy-times per update period in eTPU cycles. The eTPU busy-time depends on the BC function mode and switching mode.

Table 1. Maximum eTPU Busy-times

Mode	Maximum eTPU Busy-time per on Update Period [eTPU cycles]
Master mode, ON/OFF switching	58
Slave mode, ON/OFF switching	46
Slave mode, PWM switching	64 + 40*[update_period/PWM_period]

The eTPU module clock is equal to the CPU clock on MPC5500 devices. It is also equal to the peripheral clock, which is a half of the CPU clock, on MCF523x devices. For example, the eTPU module clock is 132 MHz on a 132-MHz MPC5554, and one eTPU cycle takes 7.58ns; it is only 75 MHz on a 150-MHz MCF5235, and one eTPU cycle takes 13.33ns.

The performance is influenced by the compiler efficiency. The above numbers, measured on the code compiled by eTPU compiler version 1.0.0.5, are given for guidance only and are subject to change. For up

to date information, refer to the information provided in the particular eTPU function set release available from Freescale.

4 C Level API for Function

The following routines provide easy access to the BC function for the application developer. Use of these functions eliminates the need to directly control the eTPU registers. There are 4 functions added to the application programming interface (API). The routines can be found in the `etpu_bc.h` and `etpu_bc.c` files, which should be included in the link file along with the top level development file(s).

Figure 7 shows the BC API state flow and lists API functions which can be used.

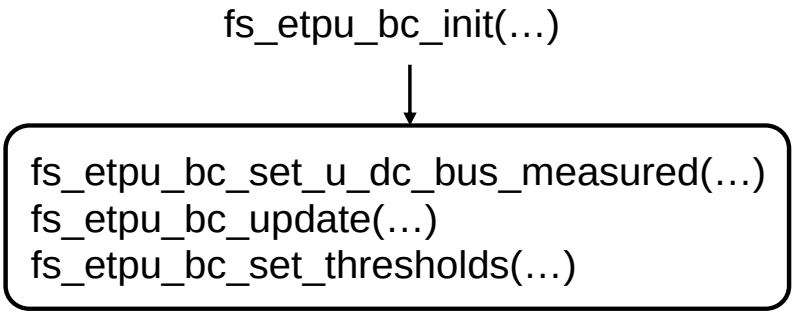

Figure 7. BC API State Flow

All BC API routines will be described in order and are listed below:

- Initialization Function:

```
int32_t fs_etpu_bc_init(uint8_t    channel,
                        uint8_t    priority,
                        uint8_t    mode,
                        uint8_t    polarity,
                        uint24_t   period,
                        uint24_t   start_offset,
                        uint24_t   services_per_irq,
                        ufract24_t u_dc_bus_ON,
                        ufract24_t u_dc_bus_OFF);
```

- Change Operation Functions:

```
int32_t fs_etpu_bc_set_u_dc_bus_measured(uint8_t    channel,
                                         ufract24_t u_dc_bus_measured);
int32_t fs_etpu_bc_update(uint8_t channel)
int32_t fs_etpu_bc_set_thresholds(uint8_t    channel,
                                  ufract24_t u_dc_bus_ON,
                                  ufract24_t u_dc_bus_OFF);
```

4.1 Initialization Function

4.1.1 int32_t fs_etpu_bc_init(...)

This routine is used to initialize the eTPU channel for the BC function. It has the following parameters:

- **channel (uint8_t)** - This is the BC channel number. This parameter should be assigned a value of 0-31 for ETPU_A, and 64-95 for ETPU_B.
- **priority (uint8_t)** - This is the priority to assign to the BC function. This parameter should be assigned a value of:
  - FS_ETPU_PRIORITY_HIGH
  - FS_ETPU_PRIORITY_MIDDLE
  - FS_ETPU_PRIORITY_LOW
  - FS_ETPU_PRIORITY_DISABLED
- **mode (uint8_t)** - This is the function mode. This parameter should be assigned a value of:
  - FS_ETPU_BC_MASTER_ON_OFF
  - FS_ETPU_BC_SLAVE_ON_OFF
  - FS_ETPU_BC_SLAVE_PWM
- **polarity (uint8_t)** – This is the BC output polarity. This parameter should be assigned a value of:
  - ETPU_BC_ON_HIGH
  - ETPU_BC_ON_LOW
- **period (uint24_t)** - This parameter depends on the selected mode.
  In master mode (mode=FS_ETPU_BC_MASTER_ON_OFF) this is the update period, as number of TCR1 clocks.
  In slave mode PWM (mode=FS_ETPU_BC_SLAVE_PWM) this is the PWM period, as number of TCR1 clocks.
  In slave mode ON/OFF (mode=FS_ETPU_BC_SLAVE_ON_OFF) this parameter does not apply.
- **start_offset (uint24_t)** - This parameter is used to synchronize various eTPU functions. The first BC update starts *start_offset* TCR1 clocks after initialization. This parameter applies in master mode only (mode=FS_ETPU_BC_MASTER_ON_OFF).
- **services_per_irq (uint24_t)** - This parameter defines the number of updates after which an interrupt service request is generated to the CPU.
- **u_dc_bus_ON (ufract24_t)** – This is the threshold value of U_DC_BUS above which the BC output is ON.
- **u_dc_bus_OFF (ufract24_t)** – This is the threshold value of U_DC_BUS bellow which the BC output is OFF.

4.2 Change Operation Functions

4.2.1 int32_t fs_etpu_bc_set_u_dc_bus_measured(uint8_t channel, ufract24_t u_dc_bus_measured)

This function sets the measured U_DC_BUS value.

- **channel (uint8_t)** - This is the BC channel number. This parameter must be assigned the same value as was assigned to the *channel* parameter in the initialization routine. If there are more BCs running simultaneously on the eTPU(s), the channel parameter distinguishes which BC function is accessed.
- **u_dc_bus (ufract24_t)** – This is the value of U_DC_BUS.

4.2.2 int32_t fs_etpu_bc_update(uint8_t channel)

This function executes the BC update. It has the following parameter:

- **channel (uint8_t)** - This is the BC channel number. This parameter must be assigned the same value as was assigned to the *channel* parameter in the initialization routine. If there are more BCs running simultaneously on the eTPU(s), the channel parameter distinguishes which BC function is accessed.

4.2.3 int32_t fs_etpu_bc_set_thresholds(uint8_t channel, ufract24_t u_dc_bus_ON, ufract24_t u_dc_bus_OFF)

This function changes the U_DC_BUS threshold values. This function has the following parameters:

- **channel (uint8_t)** - This is the BC channel number. This parameter must be assigned the same value as was assigned to the *channel* parameter in the initialization routine. If there are more BCs running simultaneously on the eTPU(s), the channel parameter distinguishes which BC function is accessed.
- **u_dc_bus_ON (ufract24_t)** – This is the threshold value of U_DC_BUS above which the BC output is ON.
- **u_dc_bus_OFF (ufract24_t)** – This is the threshold value of U_DC_BUS bellow which the BC output is OFF.

5 Example Use of Function

5.1 Demo Application

The usage of the BC eTPU function is demonstrated in the application “BLDC Motor with Speed Closed Loop and DC-Bus Break Controller, driven by eTPU on MCF523x”. For a detailed description of the demo refer to AN2954.

5.1.1 Function Calls

The BC function is configured to the slave PWM mode. The BC updates are executed by analog sensing for DC Motors eTPU function (ASDC). The ASDC triggers the AD convertor and requests DMA transfers in order to get a measured value of the DC-bus voltage to the ETPU_DATA_RAM, without CPU intervention. After new DB-bus voltage value is available, the BC update is executed.

```
/*****************************************************************************
 * Parameters
 *****************************************************************************/
uint8_t   BC_channel      = 15;
uint32_t  BC_freq_hz      = 1000;
fract24_t u_dc_bus_ON     = 0x900000;
fract24_t u_dc_bus_OFF    = 0x700000;

/*****************************************************************************
 *
 * Initialize Break Controller
 *
 *****************************************************************************/
  err_code = fs_etpu_bc_init(
     BC_CHANNEL,/* channel */
     FS_ETPU_PRIORITY_MIDDLE,/* priority */
     FS_ETPU_BC_SLAVE_PWM,/* mode */
     ETPU_BC_ON_HIGH,/* polarity */
     etpu_a_tcr1_freq/BC_freq_hz,/* period */
     0,/* start_offset */
     0,/* services_per_irq */
     u_dc_bus_ON,/* u_dc_bus_ON */
     u_dc_bus_OFF);/* u_dc_bus_OFF */
```

6 Summary and Conclusions

This application note provides the user with a description of the break controller (BC) eTPU function. The simple C interface routines to the BC eTPU function enable easy implementation of the BC in applications. The demo application is targeted at the MCF523*x* family of devices, but it could be easily reused with any device that has an eTPU.

References:

1. “The Essentials of Enhanced Time Processing Unit,” AN2353.
2. General C Functions for the eTPU,” AN2864.
3. “Using the DC Motor Control eTPU Function Set (set3),” AN2958.
4. “Using the AC Motor Control eTPU Function Set (set4),” AN2968.
5. “BLDC Motor with Speed Closed Loop and DC-Bus Break Controller, driven by eTPU on MCF523x,” AN2954.
6. Enhanced Time Processing Unit Reference Manual, ETPURM/D.
7. eTPU Graphical Configuration Tool, http://www.freescale.com/etpu, ETPUGCT.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

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

Document Number: AN2845
Rev. 0
04/2005