

Freescale Semiconductor Application Note

## Generating PWM Signal using eTPU Graphical Configuration Tool (GCT)

## 1. Introduction

This document provides steps to generate PWM (Pulse Width Modulation) signal for eTPU unit using the eTPU Graphical Configuration Tool (GCT) in CodeWarrior 2.x product. The document also includes an example project and the source code.

The eTPU is a programmable I/O controller with its own core and memory system, allowing it to perform complex timing and I/O management independently of the CPU. The eTPU Graphical Configuration Tool is a Windows application created for Freescale eTPU (Enhanced Time Processor Unit) users. The eTPU GCT offers a user-friendly graphical environment to configure the eTPU unit and generate initialization routines coded in C programming language.

## 2. Create New Project for the eTPU Unit

Freescale provides eTPU functions library that is a superset of the standard eTPU library functions. The eTPU function library enables developers to create customized functions for specific applications by providing source code of the eTPU unit library.

© Freescale Semiconductor, Inc., 2013. All rights reserved.

## Contents

- 1. Introduction.....1
- 2. Create New Project for the eTPU Unit......1

- Integrate the function set and configuration into the project created in CodeWarrior 2.x product....8





## 2.1. Create a New Project for eTPU unit in CodeWarrior 2.x product

It is assumed that the official release version of CodeWarrior 2.x product is installed on your computer.

To create a new project, perform the following steps:

- Select File > New from the CodeWarrior IDE menu bar. The New dialog box appears.
- 2. Select MPC55xx New Project Wizard from the list of available items.
- 3. Type an appropriate name for the new project in the **Project name** field. For example: *eTPU\_GCT\_application*.
- 4. Click the **Set** button to select a particular location where you want the new project and then click **OK**.

### Figure 1. New Project Wizard

Empty Project External Build Wizard	Project name: eTPU_GCT_application Location: C:\Data\FREE SCALE\MSG\C\ Set Project: 

- 5. Select the required device from the list of available derivatives, as shown in Figure 2. For example, select MPC5674F > MPC5674F\_MVx264.
- 6. Click Next.



### Figure 2. Select Derivative

10793	Select the derivative you would like to use.	
10 ad D		
11010		
210201934D		
Commence ?		
10101017-		=
intellenter 1		
	MPC5674F_MVx264	
	MPC5674F_MVxA264	
	mPC56xxK	
		-
		*
	Creates project for MPC5674F_MVx264 derivative.	
		Ŧ
	,	

- 7. Follow the instructions in the New Project wizard and provide the required information.
- 8. Click **Finish** to complete the project creation. There is the new project created and opened in CodeWarrior product successfully.

## 3. Create a function set for eTPU unit by using Function Selector web application

1. Open the **eTPU Function Selector** application to configure and download the eTPU Functions set from the following web link:

http://www.freescale.com/webapp/etpu/

- 2. Select a device your application will run on along with eTPU unit functions to include into required eTPU function set.
- 3. Select MPC5674F platform as eTPU-equipped device from the list of the available devices.
- 4. Select and mark the required functions from the offered groups of functions, e.g. to to select functions for generating PWM signals indicate **Pulse Width Modulation** in General Timing group of functions, as shown in Figure 3.
- 5. Provide feedback to Freescale.



- 6. Fill appropriate text box and describe the created application, mention your eTPU tasks, etc. All text of the note is freely editable, e.g. insert "test application" text into the **Provide us feedback** box, as shown in Figure 3.
- 7. Generate the eTPU function set of the selected functions. Click **Compile** button to complete creation process of the configuration header file in **eTPU Function Selector** tool, as shown in Figure 3.

## Figure 3. eTPU Function Selector

TP	U Function Sele	ector											
step	<u>1</u> : Select a device yo >>> (Click on lin	ur applie aks to le	catio earn	n will run on along with eT more about each function	PU funct o <i>n)</i>	tions	to include into yo	ur eTPU fun	ction	set		Step 2: Provide us feedback	
*eT	PU-equipped device:	MPC5	674F	- Availab	le code r	nem	ory: 24576 Bytes.	Remaining 2	2412	8 Bytes. *(required)		Please let us know how the	
	General Timing			Communication			Motor Contro	ol		Automotive		used. Describe your application,	
	General Pin Input / O	utput		Synchronous Peripheral			Stepper Motor			Engine Position (CRANK	)	list its features, mention the eTPU tasks.	
	GPIO	216 Bytes		SPI	428 Bytes		SM	812 Bytes		ENGINE_POSITION	2184 Bytes	test application 🔺	
V	Pulse Width Modulati	ion		Universal Asynchronous			Hall Decoder			Engine Position (CAM)			
	PWM	392 Bytes		Reciever / Transmitter			HD	568 Bytes		ENGINE_POSITION_CAM	372 Bytes		
	Input Capture	2,		UART	564 Bytes		Quadrature Dec	oder		Fuel Injection	2,100		
	IC	304		UART with Flow Control			QD	1060		FUEL 996 Bytes	Ţ		
ic.	10	Bytes	Bytes		UART FC		632	QD	Bytes			Bytes	Step 3: Compile eTPU
	Output Compare		UNIT_TO		Bytes		Quadrature Decoder -			Spark Ignition		function set	
	OC	384 Bytes		CEA709 MAC Layer - Transmitter			QDHOME	112		SPARK	824 Bytes	The generated eTPU code image file will be packed into a ZIP file	
	Frequency and Period			CEA709_TX	2116 Bytes		Bytes	Bytes		Knock Window		<ul> <li>and you will be able to download it.</li> </ul>	
	FPM	244		CEA709 MAC Layer - Rec	eiver		Index	264		KNOCK_WINDOW	308 Bytes	Compile	
	TP tep *e1	TPU Function Sele tep 1: Select a device yo >>> (Click on lir *eTPU-equipped device: General Timing General Pin Input / O GPIO Pulse Width Modulat PWM Input Capture IC Output Compare OC Frequency and Perio Measurement FPM	TPU Function Selector         itep 1: Select a device your application of the sector	TPU Function Selector         tep 1: Select a device your applicatio         >>> (Click on links to learn         *eTPU-equipped device:       MPC56741         General Timing       Image: Comparison of the system         General Pin Input / Output       Image: Comparison of the system         PUIse Width Modulation       392 Bytes         Input Capture       Image: Comparison of the system         Output Compare       384 Bytes         Frequency and Period Measurement       244 Comparison of the system	TPU Function Selector         itep 1: Select a device your application will run on along with eT >>> (Click on links to learn more about each function         *eTPU-equipped device:       MPC5674F • Available         General Timing       Communication         General Pin Input / Output       Synchronous Peripheral Interface         GPIO       216 Bytes         Pulse Width Modulation       Universal Asynchronous Reciever / Transmitter         PWM       392 Bytes         Input Capture       UART         IC       304 Bytes         Output Compare       CEA709 MAC Layer - Transmitter         Frequency and Period Measurement       244         FPM       244	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU function)         *eTPU-equipped device: MPC5674F  Available code r         General Timing       Communication         General Timing       Synchronous Peripheral Interface         GPIO       216         PWM       392         PWM       392         PWM       392         Input Capture       UART       564         Input Capture       UART       564         Output Compare       Quart       632         Oc       384       Expression       CEA709 MAC Layer - Transmitter         FPM       244       CEA709 MAC Layer - Receiver	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions         *eTPU-equipped device: MPC5674F Available code memory         *eTPU-equipped device: MPC5674F Available code memory         General Timing       Communication         General Timing       Synchronous Peripheral Interface         GPIO       216         Bytes       Synchronous Peripheral Interface         PUSe Width Modulation       Wiversal Asynchronous         PWM       392         Bytes       Universal Asynchronous         Input Capture       UART         IC       304         Output Compare       UART with Flow Control         Oc       384         Frequency and Period       CEA709 MAC Layer - Transmitter         CEA709 MAC Layer - Receiver       CEA709 MAC Layer - Receiver	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into yo >>>> (Click on links to learn more about each function)         *eTPU-equipped device: MPC5674F Available code memory: 24576 Bytes.         Communication       Motor Contro         General Timing       Communication       Motor Contro         General Pin Input / Output       Synchronous Peripheral Interface       Stepper Motor         GPIO       216       Spl       428         WM       392       Universal Asynchronous Recipheral Interface       SM         PWM       392       Universal Asynchronous Reciper / Transmitter       HD         Input Capture       UART       564       QD         Input Capture       UART with Flow Control       QD       QD         Output Compare       CEA709 MAC Layer - Transmitter       QDHOME       QDHOME         Frequency and Period       CEA709_TX       2116       Quadrature Decole Index         FPM       244       CEA709 MAC Layer - Receiver       Quadrature Decole Index	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into your eTPU fun >>> (Click on links to learn more about each function)         *eTPU-equipped device:       MPC5674F       Available code memory: 24576 Bytes. Remaining 1         General Timing       Communication       Motor Control         General Timing       Synchronous Peripheral Interface       Motor Control         Spi       428 Bytes       Bytes       Bytes         Pulse Width Modulation       Universal Asynchronous Reciever / Transmitter       Motor Control       Stepper Motor         Input Capture       UNART       564 Bytes       Quadrature Decoder         Oc       384 Bytes       CEA709 MAC Layer - Transmitter       Guadrature Decoder - Home         Oc       384 Bytes       CEA709 MAC Layer - Receiver       Quadrature Decoder - Home         PWM       244 Bytes       CEA709 MAC Layer - Receiver       Quadrature Decoder - Home	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into your eTPU function         *eTPU-equipped device:       MPC5674F       Available code memory: 24576 Bytes. Remaining 2412         Mector Colspan="2">Mector Control         Mector Control         General Timing       Motor Control         Options       Synchronous Peripheral Interface       Synchro	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into your eTPU function set >>>> (Click on links to learn more about each function)         * eTPU-equipped device:       MPC5674F       Available code memory: 24576 Bytes. Remaining 24128 Bytes. *(required)         Click on links to learn more about each function)         * eTPU-equipped device:       MPC5674F       Available code memory: 24576 Bytes. Remaining 24128 Bytes. *(required)         General Timing       Ocommunication       Motor Control       Automotive         General Timing       Synchronous Peripheral Interface       Synchronous Peripheral Interface       Motor       Engine Position (CRANK         GPIO       216       Synchronous Peripheral Interface       SM       812       ENGINE_POSITION         PWIM       392       Bytes       Click on links to learn more about each function         Input Capture       1       Quadrature Decoder       Fuel Injection         PWIM       384       Quadrature Decoder <th c<="" td=""><td>TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into your eTPU function set &gt;&gt;&gt;&gt; (Click on links to learn more about each function)         * eTPU-equipped device:       MPC5674F       Available code memor: 24576 Bytes. Remaining 24128 Bytes. *(required)         6 General Timing       Ocmmunication       Motor Control       Automotive         General Timing       Communication       Motor Control       Automotive         General Timing       Ocmmunication       Motor Control       Automotive         General Timing       Synchronous Peripheral Inherface       Singen Position (CRANK)         GPIO       216 Bytes       Bytes       Bytes       Bytes       Bytes       Engine Position (CRANK)         PWIM       392 Bytes       392 Bytes       Bytes       Bites       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       Bytes       Bytes       Engine Position (CAM)       Bytes       Bytes       Engine Position (CAM)       Bytes</td></th>	<td>TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into your eTPU function set &gt;&gt;&gt;&gt; (Click on links to learn more about each function)         * eTPU-equipped device:       MPC5674F       Available code memor: 24576 Bytes. Remaining 24128 Bytes. *(required)         6 General Timing       Ocmmunication       Motor Control       Automotive         General Timing       Communication       Motor Control       Automotive         General Timing       Ocmmunication       Motor Control       Automotive         General Timing       Synchronous Peripheral Inherface       Singen Position (CRANK)         GPIO       216 Bytes       Bytes       Bytes       Bytes       Bytes       Engine Position (CRANK)         PWIM       392 Bytes       392 Bytes       Bytes       Bites       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       Bytes       Bytes       Engine Position (CAM)       Bytes       Bytes       Engine Position (CAM)       Bytes</td>	TPU Function Selector         tep 1: Select a device your application will run on along with eTPU functions to include into your eTPU function set >>>> (Click on links to learn more about each function)         * eTPU-equipped device:       MPC5674F       Available code memor: 24576 Bytes. Remaining 24128 Bytes. *(required)         6 General Timing       Ocmmunication       Motor Control       Automotive         General Timing       Communication       Motor Control       Automotive         General Timing       Ocmmunication       Motor Control       Automotive         General Timing       Synchronous Peripheral Inherface       Singen Position (CRANK)         GPIO       216 Bytes       Bytes       Bytes       Bytes       Bytes       Engine Position (CRANK)         PWIM       392 Bytes       392 Bytes       Bytes       Bites       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       372 Bytes       Bytes       Engine Position (CAM)       Bytes       Bytes       Engine Position (CAM)       Bytes       Bytes       Engine Position (CAM)       Bytes

- 8. Log-in to the Freescale eTPU Function Selector web application. If you are an existing member, then provide your email address. In case you are a new member, you will need to follow the registration process.
- 9. Accept the Freescale Semiconductor Software License Agreement and click the I Accept button.
- 10. Select the desired path and save the generated "etpu\_set .zip" file containing the functions set in the selected directory.

The generated function set for the eTPU unit is packed into "etpu\_set .zip" zipped file and successfully stored at the selected destination path.

11. Unzip the packaged "etpu\_set .zip" file to required destination directory. There is created the directory structure at the destination path, as shown in Figure 4.







# 4. Create Configuration for the eTPU unit by using the eTPU Graphical Configuration Tool

The eTPU Graphical Configuration Tool is application runs on Microsoft Windows operating system and is created for Freescale Enhanced Time Processor Unit (eTPU) users. The eTPU Graphical Configuration Tool offers a user-friendly graphical environment to configure the eTPU unit and generate initialization routines coded in C-programming language. The tool supports of various Freescale processors with the eTPU unit integrated. Take the Graphical Configuration Tool to create configuration of the eTPU unit using generated function set.

## 4.1. Install the eTPU Graphical Configuration Tool

To start the installation process of the tool, perform the following steps:

1. Download the "ETPUGCT" application tool from the following Freescale web site:

http://www.freescale.com/etpu/

There is "ETPUGCTSW.exe" executable file downloaded from the web side.

Install the "ETPUGCT" application tool to your computer at the selected path.

2. Double click on "ETPUGCTSW.exe" executable file and then follow the installation instructions of **InstallShield** Wizard tool on the screen.

## 4.2. Generate eTPU unit Configuration

 Double click the "etpugct.exe" executable file and run the eTPU Graphical Configuration Tool application from start menu of Microsoft Windows operating system. There tool opens as a window, as showed in Figure 5.



eTPU Graphical ConfigurationTc	ool - untitled_etpu_gct
File Edit View eTPU Help	
Processor Function Set Engine A	Engine A Channels   Engine B   Engine B Channels   Verbatim Block
Larget CPU:           Generic CPU with eTPU2           MCF523x           MPC5533           MPC5534           MPC5554           MPC5565           MPC5566           MPC5567           MPC5664A           MPC5674F	CPU Details Name: MPC5674F Number of engines: 2 Max. system clock: 200 STAC Bus Available: Yes Clock Settings System clock: 100 MHz ▼ eTPU Clock (=SysClk): 100 MHz

Figure 5. eTPU Graphical Configuration Tool

- 2. Set required path for the generated function set "etpu\_set .zip" file in eTPU Graphical Configuration Tool application.
- 3. Open **eTPU** > **Options...** selection form main menu of the tool and browse current directory with the generated function set file and select "etpu\_set" directory, as showed in Figure 6.

Figure 6. eTPU Graphical Configuration Tool used in the project selection

ile Edit View e D 🗃 🖬   봈 🖻	:TPU Help
rocessor Function Se	et Engine A Engine A Channels Engine B Engine B Channels Verbatim Block
eTPU Function <u>S</u> et to etpu_set.h etpu_set.h	be used Function Set Image Details Entry table base: MISC compare value:
mage <u>F</u> unctions: #   Function There	Options         Search Directories         Function Sets:       avFREE SCALEVMSGVOW_notesveTPU application notely         Help         Reference Manual:       doc\ETPURM.pdf         App. Notes:       doc\appnotes         Note: All paths can be specified either as absolute path or as a path relative to eTPU GCT application".exe" file.       DK



- 4. Restart the eTPU Graphical Configuration Tool. You have to exit the eTPU Graphical Configuration Tool application and then start it again. The tool is updated and a particular function set is opened from the destination directory after restart.
- 5. Select the appropriate platform from the list of available platforms and set values of required parameters. There has to be done the following steps:
  - a) Select MPC5674F processor platform on **Processor** card.
  - b) Select "etpu\_set.h" file contains the function set on Function Set card
  - c) Set clock source and frequency of **TCR1** as time base **clock frequency**
  - d) Select channel number 0 to generate **the require**d signal on.
  - e) Double click at Channel #0 on **Engine A Channels** card and open **Edit eTPU Function** window to set parameter's values of required PWM function, as showed in Figure 6.

The following are the parameters of the generated PWM signal on channel 0 set.

Frequency:1 kHzDuty cycle:50%

Polarity: active High

### Figure 7. Edit eTPU function

Edit eTPU Function eTPU Function	Report Daniel	ОК	Cancel
C-API to be used: fs_etpu_pwm_init	The eTPU function   how many channels	provides one or more C-language and what parameters are used b	API calls which define the eTPU function.
Channels	Parameter Values	V	Show parameter tip:
Channel Name Number	Parameter Name	Value	Туре
channel O	priority freq duty polarity timebase timebase_freq	Middle 1000 5000 FS_ETPU_PWM_ACTIVEH TCR1 Engine A, TCR1 frequency	uint8_t uint82_t uint16_t uint8_t uint8_t uint32_t
Generated Code: /* * eTPU API Function initializatio */ err code = fs etnu nym init (PWMx C	n: 'PWM'	engine: A: channel: (	. */

The following source code is generated in C-programming language for the initialization function of PWM signal initialization there, as shown in Listing 1.

6. Click **OK** button to confirm the parameters settings of the initialization function.



```
Listing 1. PWM module of the eTPU unit initialization function source code
/*
 *
   eTPU API Function initialization: 'PWM'
 */
err_code = fs_etpu_pwm_init (PWMx_CHANNEL, /* engine: A; channel: 0 */
                            FS_ETPU_PRIORITY_MIDDLE, /* priority: Middle */
                             1000, /* freq: 1000 */
                                      /* duty: 5000 */
                             5000,
                             FS_ETPU_PWM_ACTIVEHIGH, /* polarity: FS_ETPU_PWM_ACTIVEHIGH */
                             FS_ETPU_TCR1, /* timebase: TCR1 */
                             etpu_a_tcr1_freq); /* timebase_freq: Engine A, TCR1
frequency */
if (err_code != 0)
      return ((PWMx_CHANNEL) + 1);
```

- **NOTE:** For more details regarding to settings of parameters see the eTPU Graphical Configuration Tool User's Manual. This document contains a detailed description of the eTPU Graphical Configuration Tool user interface and explains important facts of the functionalities.
- 7. Save the generated configuration and generate the source code for the required settings and parameters. Open File > Save from main menu of the eTPU Graphical Configuration Tool, type the name of file contained source code, e.g. "etpu\_gct.c" and then browse destination directory to store the generated configuration.

The C programming language source code files and header files are generated. These files contain configuration for the eTPU unit successfully stored at the selected destination path.

## 5. Integrate the function set and configuration into the project created in CodeWarrior 2.x product

To integrate the function set and configuration ensure all the necessary sources and files exist.

- 1. Open "eTPU\_GCT\_application" project created by Project Wizard integrated in CodeWarrior 2.x product.
- 2. Select **File > Open** from main menu of CodeWarrior tool and browse the project.
- 3. Add created function set for the eTPU unit into the project.



4. Right click on the **Sources** folder in the opened project navigation window, select **Add Files...** item from the popup menu and browse the unzipped folder of the generated function set.

The "etpu\_set" folder is created and it contains source codes and files added into the destination project.

- 5. Add created configuration of eTPU unit into the project. Right click on the **Sources** folder in the opened project navigation window, select **Add Files...** item from the popup menu and browse the "etpu\_gct.c" source code file and "etpu\_gct.h" header file of the configuration. The files for the eTPU unit configuration are added into the destination project.
- 6. Add call of the initialization function and start of eTPU unit into *main* function of the project. Initialize I/O pins of the device and FMPLL system clock of the processor. Open and modify the *main* function in "main.c" source file, as showed in Listing 2.
- 7. Compile and download the project into the MPC5674F platform and run the application to generate required PWM signal. The PWM signal is generated on channel 0.

There is example project and source codes to demonstrate required PWM signal generation as part of the application note.

## Listing 2. PWM module of the eTPU unit initialization function and enable interrupts source code

```
int main(void) {
volatile int i = 0;
volatile int err_code = 0;
GPIO_Init();
                  // Init device I/O pin.
                    // Init FMPLL - 100MHz System Clock.
FMPLL_Init();
/* Initialize eTPU */
err_code = my_system_etpu_init();
if (err_code != 0)
   return -1;
  /* Start eTPU */
 my_system_etpu_start();
  /* Loop forever */
 for (;;) {
   i++;
  }
```



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

#### Document Number: AN4687

28 February 2013

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor 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 and all 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 the 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 and hold 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 claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale, the Freescale logo, CodeWarrior and ColdFire are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Flexis and Processor Expert are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners

© Freescale Semiconductor, Inc. 2013. All rights reserved.

