Freescale Semiconductor
Application Note

Document Number: AN4687

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.

Contents

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:

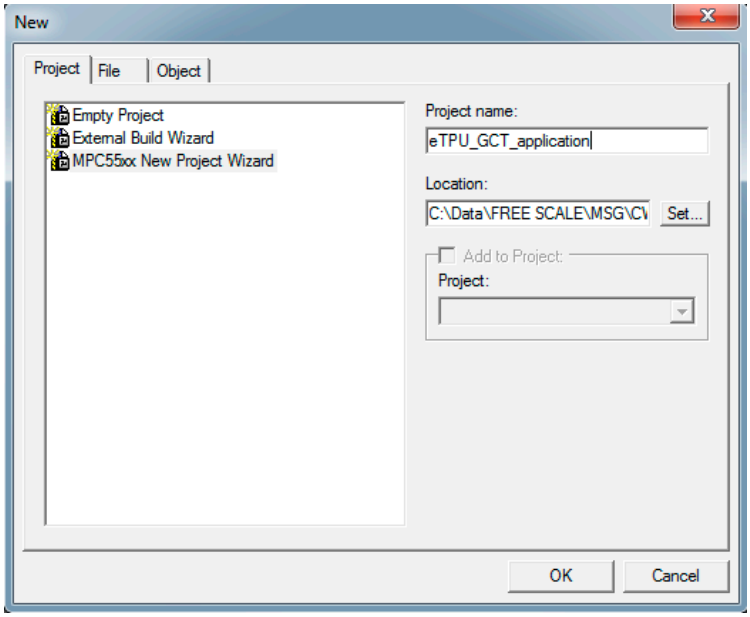
1. Select **File** > **New** from the CodeWarrior IDE menu bar.
   The New dialog box appears.

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**.

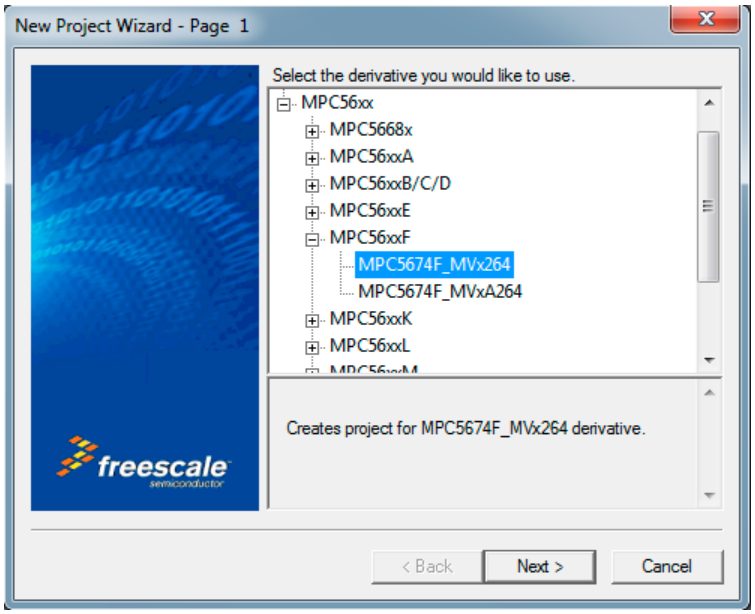
Figure 1. New Project Wizard



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



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.

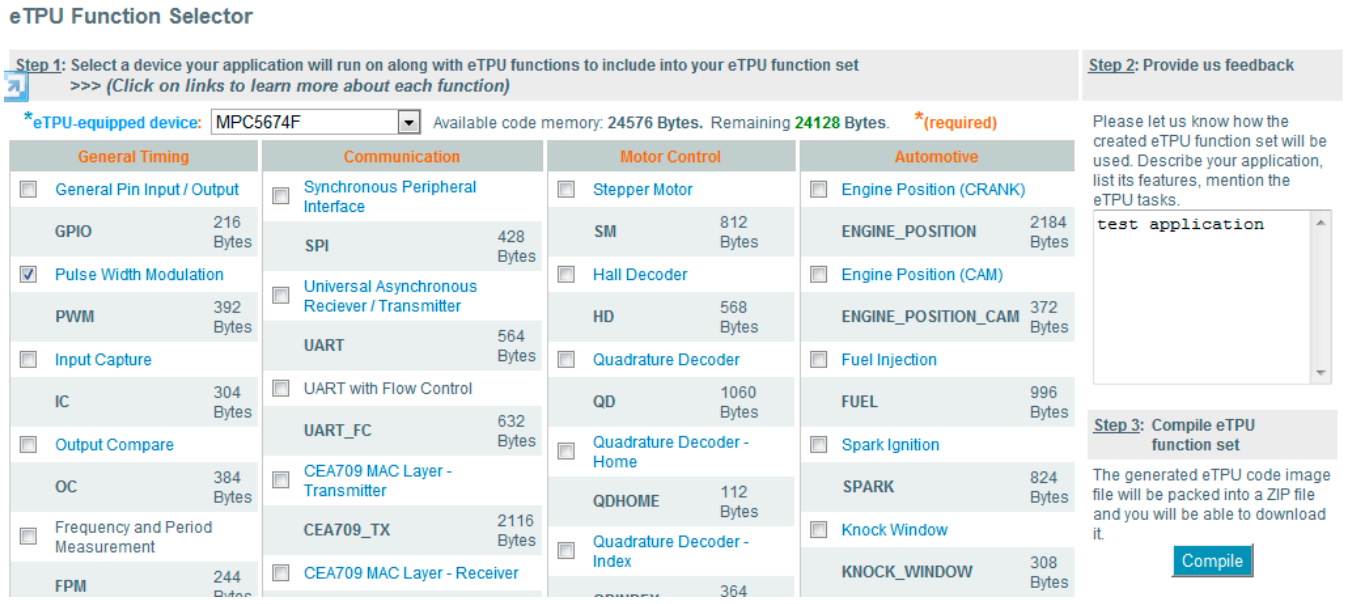
4. Select and mark the required functions from the offered groups of functions, e.g. to to select functions for generating PWM signals indicate **Pulse Width Modulation** in General Timing group of functions, as shown in Figure 3.

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



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.

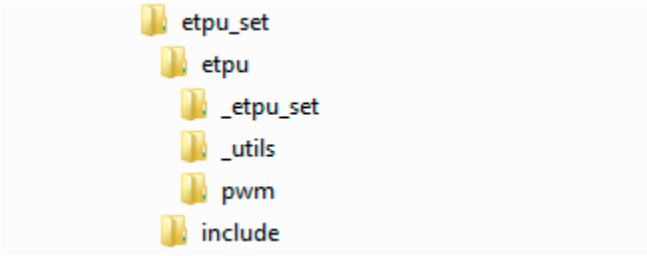
9. Accept the Freescale Semiconductor Software License Agreement and click the **I Accept** button.

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.

Figure 4. The created directory structure of configuration at the destination path



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.

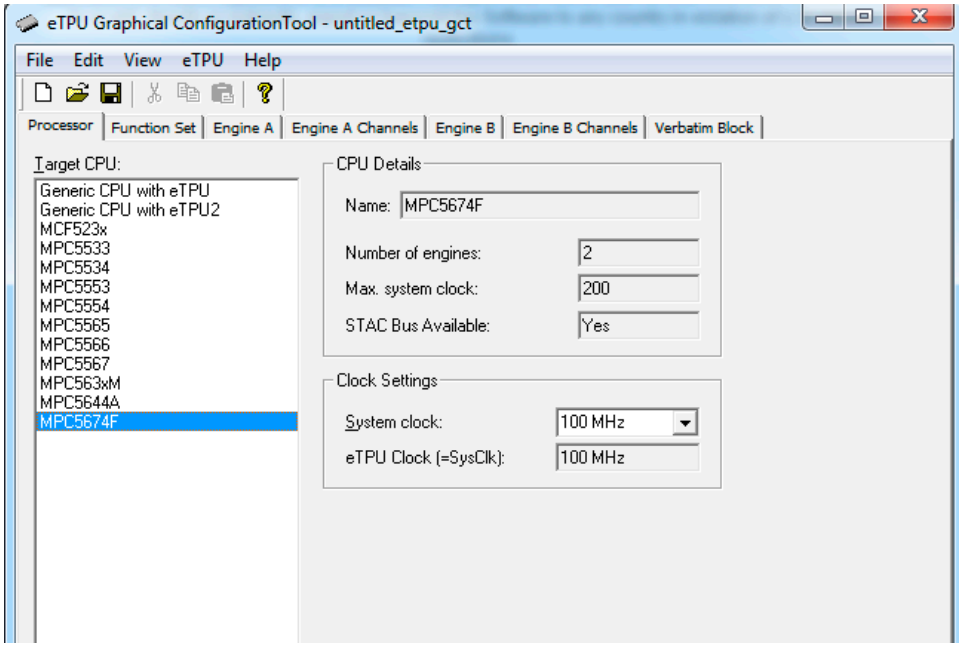
   Install the “ETPUGCT" application tool to your computer at the selected path.

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

1. 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.

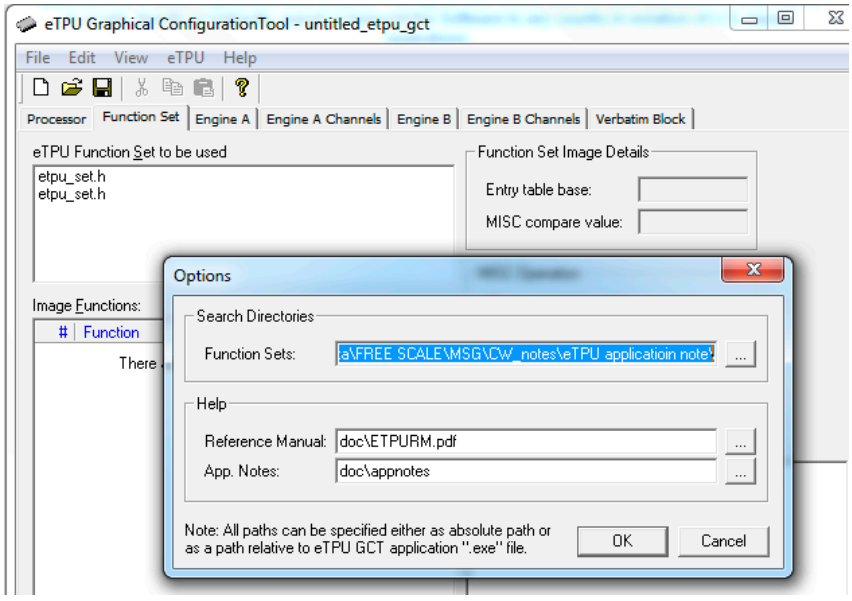
Figure 5. eTPU Graphical Configuration Tool



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



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.

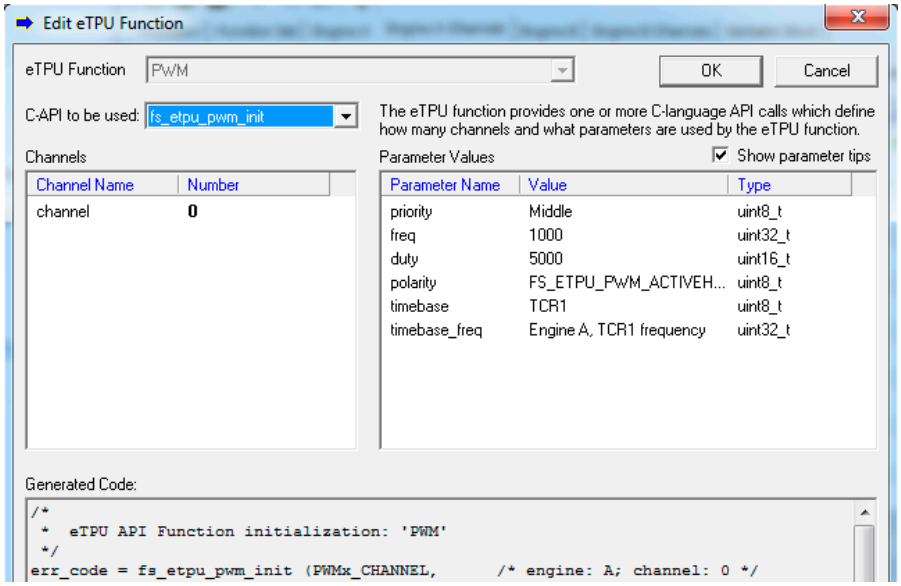
   The following are the parameters of the generated PWM signal on channel 0 set.

   Frequency: 1 kHz

   Duty cycle: 50%

   Polarity: active High

Figure 7. Edit eTPU function



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 */
                             5000,      /* duty: 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.
 FMPLL_Init();      // Init FMPLL - 100MHz System Clock.

 /* 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