

eTPU General Function Set (Set 1)

by David Paterson
MCD Applications Engineer

1 Introduction

This application note complements AN2863SW to describe the enhanced time processing unit (eTPU) general function set (Set 1). AN2863SW contains the binary image of the eTPU code, source code, and the interface files for the function application programming interfaces (APIs).

Functions available are GPIO, PWM, IC, OC, FPM, PPA, UART, QOM, SM, SPI, SPWM, and test, which are discussed in this application note.

AN2863SW is available for download at <http://www.freescale.com>.

Contents

| | | |
|------|--|---|
| 1 | Introduction | 1 |
| 2 | Function Set Overview | 2 |
| 2.1 | GPIO (General-Purpose Input/Output) | 2 |
| 2.2 | PWM (Pulse Width Modulation) | 3 |
| 2.3 | IC (Input Capture) | 3 |
| 2.4 | OC (Output Compare) | 3 |
| 2.5 | FPM (Frequency Pulse Measurement) | 3 |
| 2.6 | PPA (Pulse Period Accumulate) | 3 |
| 2.7 | UART (Universal Asynchronous Receiver Transmitter) | 3 |
| 2.8 | QOM (Queued Output Match) | 3 |
| 2.9 | SM (Stepper Motor) | 4 |
| 2.10 | SPI (Serial Peripheral Interface) | 4 |
| 2.11 | SPWM (Synchronized Pulse Width Modulation) | 4 |
| 2.12 | Test | 4 |
| 3 | Function Set Usage | 4 |
| 3.1 | eTPU Function Source Code and Binary Image | 4 |
| 3.2 | eTPU Function APIs | 5 |
| 4 | Files Included | 5 |
| 4.1 | eTPU Code | 6 |
| 4.2 | CPU Interface Files | 6 |
| 4.3 | Summary and Conclusions | 7 |

2 Function Set Overview

The eTPU general function set consists of the following eTPU functions. Dedicated application notes and API software are available for each of these functions:

Table 1. Function Set Overview

| Function | Description | Releated Application Note |
|----------|---|---------------------------|
| GPIO | General Purpose Input/Output | AN2850 |
| PWM | Pulse Width Modulation | AN2849 |
| IC | input Capture | AN2851 |
| OC | Output Compare | AN2852 |
| FPM | Frequency Pulse Measurement | AN2859 |
| PPA | Pulse Period Accumulate | AN2858 |
| UART | Universal Asynchronous Receiver Transmitter | AN2853 |
| QOM | Queued Output Match | AN2857 |
| SM | Stepper Motor | AN2869 |
| SPI | Serial Peripheral Interface | AN2847 |
| SPWM | Synchronized Pulse Width Modulation | AN2854 |
| TEST | | N/A |

The eTPU functions are configurable and adjustable using various parameters and options. This allows the user to customize these functions to their specific application.

To support the eTPU functions, versions of the `etpuc.h` (version 1.01) and `etpuc_common.h` (version 1.1) from Byte Craft are used.

The following paragraphs briefly describe the purpose of each eTPU function. Refer to the application notes above for more details.

2.1 GPIO (General-Purpose Input/Output)

The GPIO functions help the user configure an eTPU channel as an input or output.

As an input, the selected eTPU pin can be read periodically, as a result of a CPU request or whenever a transition occurs on a pin. When an eTPU channel is configured to read the pin periodically, the period is determined by either of the eTPU timebases.

As an output, the pin can be driven to a logic high or a logic low level as a result of a CPU request. When a channel is configured as an input, the eTPU records the last 24 levels, or edge transitions, in a parameter.

These functions can be used only on devices that have a System Integration Unit.

2.2 PWM (Pulse Width Modulation)

The PWM functions generate a PWM (pulse width modulated) waveform where the frequency and duty can be changed at any time by the CPU. The function can generate waveforms with duty cycles ranging from 0% to 100%.

2.3 IC (Input Capture)

This IC function captures input transitions on an eTPU channel. It captures, counts, and records the times of edges. The number of counts can be a single specified number or a continuous count. Edges counted can be rising, falling, or both. Links can be made to other channels on completion of a certain number of counts for a specified eTPU channel.

2.4 OC (Output Compare)

The OC functions can create a single output transition or a pulse. The initial reference time can be immediate — a pointer or a value. The initial pin state can be defined. The pin action at each of the two events can be defined, and the timebase used for match and capture events can be specified separately. An interrupt is requested after the second action.

2.5 FPM (Frequency Pulse Measurement)

The FPM function counts the number of pulses in a specified window, and alternately measures the frequency of a periodic stream of pulses.

2.6 PPA (Pulse Period Accumulate)

The PPA functions measure the total accumulated high time, low time, or period of an input signal over a user-defined number of periods. The period may be measured between rising or falling edges.

2.7 UART (Universal Asynchronous Receiver Transmitter)

The UART functions use two eTPU channels to provide a 3-wire (TxD, RxD, and GND) asynchronous serial interface. They can be used to add serial capability to a device without a serial port, or to add additional serial I/O to a device that already has hardware UART(s).

2.8 QOM (Queued Output Match)

The QOM function generates complex pulse trains using a sequence of output matches. An output match causes a programmable pin response if a user-defined value is matched by the value of a timebase. QOM generates multiple output matches using a table of offset times and pin states. The table size is user-programmable. Various modes of queue operation are supported.

2.9 SM (Stepper Motor)

The SM functions can drive a stepper motor (SM) based on user-defined acceleration tables. Each time a desired position is set, the eTPU generates signals to drive the motor to the position. Even if the desired position is changed before the movement is completed; the motor continues to function in an optimal way move to the new desired position. The function is capable of driving 2-phase and 3-phase motors in full-step or half-step modes.

2.10 SPI (Serial Peripheral Interface)

The SPI functions use three eTPU channels to form a bidirectional, synchronous serial port that can be used to communicate with a wide variety of devices. They can be used to add serial capabilities to a device without a serial port, or to add further serial port(s) to a device that already has a hardware-synchronous port. One eTPU channel is configured to function as the clock and the other two are configured to function as serial transmitter (TxD, or data out) and serial receiver (RxD, or data in). These functions allow a master only configuration.

2.11 SPWM (Synchronized Pulse Width Modulation)

The synchronized pulse-width modulation function generates a pulse-width modulated (PWM) waveform where the CPU can change the period or high time at any time. When synchronized to a function on a second channel, SPWM low-to-high transitions have a configurable relationship to transitions on the second channel. The function can generate a complete range of duty cycles from 0% to 100%.

2.12 Test

This function is used to generate eTPU channel signals that can be used to help test other eTPU functions. The test function is used to test the eTPU functions and is not intended for customer use.

3 Function Set Usage

The eTPU general function set release consists of the items described below.

3.1 eTPU Function Source Code and Binary Image

The eTPU function source code and the compiled binary image are included in the following package, which is available on the Freescale website (<http://www.freescale.com/etpu>):

- AN2863SW— eTPU general function set source code and compiled binary image

It is not necessary for you to compile the eTPU function source code. Only advanced eTPU users should modify the eTPU functions and then recompile, using the Byte Craft Limited eTPU_C compiler. The CPU application must load the compiled eTPU code into eTPU CODE RAM. The eTPU code binary image is included, together with other initialization items, in the `etpu_set1.h` file.

The eTPU module initialization can be managed by the `fs_etpu_init(...)` function, which is one of the standard eTPU utilities `etpu_util.c` and `etpu_util.h`. The eTPU utilities are available on the Freescale web (“General C Functions for the eTPU,” <http://www.freescale.com/etpu>, AN2863SW).

3.2 eTPU Function APIs

The eTPU function APIs enable the use of eTPU functions in applications. The eTPU function API source code is included in the following packages, which are available on the Freescale website (<http://www.freescale.com/etpu>):

Table 2. eTPU Function API Source Code

| Package | API Source Description |
|---------------|------------------------|
| AN2850SW | GPIO eTPU function API |
| AN2849SW | PWM eTPU function API |
| AN2851SW | IC eTPU function API |
| AN2852SW | OC eTPU function API |
| AN2859SW | FPM eTPU function API |
| AN2858SW | PPA eTPU function API |
| AN2853SW | UART eTPU function API |
| AN2857SW | QOM eTPU function API |
| AN2869SW | SM eTPU function API |
| AN2847SW | SPI eTPU function API |
| AN2854SW | SPWM eTPU function API |
| SET1_TEST_API | Test eTPU function API |

The eTPU function APIs include CPU methods that demonstrate how to initialize, control, and monitor the eTPU function. The CPU application does not need to access eTPU channel registers and/or function parameters directly. Rather, the CPU application can use the eTPU function APIs instead.

4 Files Included

Here is a list of files included in the AN2863SW download.

Table 3. Files Included in AN2863SW

| File Name | File Description |
|---------------------------------------|--|
| <code>etpu_set1\etpu_set1.h</code> | Image of eTPU code for host CPU (generated by eTPU compiler) |
| <code>etpu_set1\makefile</code> | A makefile used to build the code |
| <code>etpu_set1\etpuc_set1.cod</code> | Debug information for set1 functions |

4.1 eTPU Code

Table 4. eTPU Code

| Code Name | Description |
|--------------------------|--|
| etpu_set1\etpu_set1.c | Top level file for set1 functions |
| etpu_set1\etpuc.h | Standard include file for eTPU code |
| etpu_set1\etpuc_common.h | Standard include file for eTPU code |
| etpu_set1\etpuc_util.h | Macros from Byte Craft used by some of the functions |
| etpu_set1\etpuc_gpio.c | C source for the eTPU GPIO function |
| etpu_set1\etpuc_ic.c | C source for the eTPU IC function |
| etpu_set1\etpuc_fpm.c | C source for the eTPU FPM function |
| etpu_set1\etpuc_oc.c | C source for the OC function |
| etpu_set1\etpuc_ppa.c | C source for the eTPU PPA function |
| etpu_set1\etpuc_pwm.c | C source for the eTPU PWM function |
| etpu_set1\etpuc_qom.c | C source for the eTPU QOM function |
| etpu_set1\etpuc_sm.c | C source for the eTPU SM function |
| etpu_set1\etpuc_spi.c | C source for the eTPU SPI function |
| etpu_set1\etpuc_test.c | C source for the eTPU test function |
| etpu_set1\etpuc_uart.c | C source for the eTPU UART function |
| etpu_set1\etpuc_spwm.c | C source for the eTPU SPWM function |

4.2 CPU Interface Files

These files provide an interface between eTPU code and CPU code. All references to the general set functions should be made with information in these files. This allows only symbolic information to be referenced, which allows the eTPU code to be optimized without effecting the CPU code.

Table 5. CPU Interface Files

| Code Name | Description |
|--------------------------------|-------------------------------------|
| etpu_set1\cpu\etpu_gpio_auto.h | GPIO function interface information |
| etpu_set1\cpu\etpu_ic_auto.h | IC function interface information |
| etpu_set1\cpu\etpu_fpm_auto.h | FPM function interface information |
| etpu_set1\cpu\etpu_oc_auto.h | OC function interface information |
| etpu_set1\cpu\etpu_ppa_auto.h | PPA function interface information |
| etpu_set1\cpu\etpu_pwm_auto.h | PWM function interface information |
| etpu_set1\cpu\etpu_qom_auto.h | QOM function interface information |
| etpu_set1\cpu\etpu_sm_auto.h | SM function interface information |

Table 5. CPU Interface Files (continued)

| Code Name | Description |
|--------------------------------|-------------------------------------|
| etpu_set1\cpu\etpu_spi_auto.h | SPI function interface information |
| etpu_set1\cpu\etpu_test_auto.h | Test function interface information |
| etpu_set1\cpu\etpu_uart_auto.h | UART function interface information |
| etpu_set1\cpu\etpu_spwm_auto.h | SPWM function interface information |

4.3 Summary and Conclusions

This application note provides a description of the eTPU general function set (set 1) that complements AN2863SW. AN2863SW contains the binary image of the eTPU code and source code.

How to Reach Us:**Home Page:**

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
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 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

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.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant 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.
© Freescale Semiconductor, Inc. 2007. All rights reserved.