To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

RENESAS

Preliminary Application Note

V850E/IF3, V850E/IG3

32-bit Single-Chip Microcontrollers

Sample Programs for Timer T

V850E/IF3:
- μPD70F3451
- μPD70F3452

V850E/IG3:
- μPD70F3453
- μPD70F3454

Document No. U18732EJ1V0AN00 (1st edition)
Date Published August 2007 N

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

- **The information contained in this document is being issued in advance of the production cycle for the product. The parameters for the product may change before final production or NEC Electronics Corporation, at its own discretion, may withdraw the product prior to its production.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special", and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics products before using it in a particular application.

  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M5 02. 11-1

INTRODUCTION

Cautions 1. This Application Note explains a case where the V850E/IG3 is used as a representative microcontroller. Use this Application Note for your reference when using the V850E/IF3.

2. Download the program used in this manual from the page of Programming Examples (http://www.necel.com/micro/en/designsupports/sampleprogram/index.html) in the NEC Electronics Website (http://www.necel.com/).

3. This sample program is provided for reference purposes only and operations are therefore not subject to guarantee by NEC Electronics Corporation. When using sample programs, customers are advised to sufficiently evaluate this product based on their systems, before use.

4. When using sample programs, reference the following startup routine and link directive file and adjust them if necessary.

- **Startup routine: ig3_start.s**
- **Link directive file: ig3_link.dir**

Target Readers

This Application Note is intended for users who understand the functions of the V850E/IF3 (μPD70F3451, 70F3452), and V850E/IG3 (μPD70F3453, 70F3454), and who design application systems that use these microcontrollers.

Purpose

This manual is intended to give users an understanding of the basic functions of the V850E/IF3 and V850E/IG3, using the application programs.

How to Use This Manual

It is assumed that the reader of this Application Note has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

For details of hardware functions (especially register functions, setting methods, etc.) and electrical specifications

→ See the **V850E/IF3, V850E/IG3 Hardware User's Manual**.

For details of instruction functions

→ See the **V850E1 Architecture User's Manual**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
Memory map address:	Higher addresses on the top and lower addresses on the bottom
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH
Prefix indicating the power of 2 (address space, memory capacity):	K (kilo): $2^{10} = 1,024$
	M (mega): $2^{20} = 1,024^2$
	G (giga): $2^{30} = 1,024^3$

The function lists are structured as follows.

Hardware name (symbol)

[Function]	Function description
[Function name]	Name of sample function
[Argument(s)]	Type and overview of argument(s)
[Processing content]	Processing content of sample function
[SFR(s) used]	Register name and setting content
[call function(s)]	Name and function of call function(s)
[Variable(s)]	Type, name, and overview of variable(s) used in sample function
[Interrupt(s)]	Name of function
[Interrupt source(s)]	Name
[File name]	Name of corresponding sample program file
[Caution(s)]	Caution(s) upon function usage

Interrupt function

[Function name]	Name of interrupt function
[Overview]	Servicing content
[Source(s)]	Name of interrupt and conditions for occurrence
[call function(s)]	None
[Variable(s)]	Name of variable, function
[File name]	Name of corresponding sample program file
[Caution(s)]	None

Product Differences The differences between the V850E/IG3 and the V850E/IF3 related to the 16-bit timer/event counter T (TMT) are shown below.

Item		V850E/IG3	V850E/IF3
TIT00, TIT01, TOT00, TOT01, TENC00, TENC01, EVTT0, TECR0 pins		Provided	None
TMT0	External event count mode	Provided	None
	External trigger pulse output mode	Provided	None
	One-shot pulse output mode	Provided	None
	PWM output mode	Provided	None
	Free-running timer mode	Provided	Provided (compare function only)
	Triangular-wave PWM output mode	Provided	None
	Encoder compare mode	Provided	None

Related Documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents related to V850E/IF3 and V850E/IG3

Document Name	Document No.
V850E1 Architecture User's Manual	U14559E
V850E/IF3, V850E/IG3 Hardware User's Manual	U18279E
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (UARTA) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (UARTB) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (CSIB) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (I^2C) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for DMA Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer M Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Watchdog Timer Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer AA Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer AB Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer T Application Note	This manual
V850E/IF3, V850E/IG3 Sample Programs for Port Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Clock Generator Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Standby Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Interrupt Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for A/D Converters 0 and 1 Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for A/D Converter 2 Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Low-Voltage Detector (LVI) Function Application Note	To be prepared
V850E/IF3, V850E/IG3 6-Phase PWM Output Control by Timer AB, Timer Q Option, Timer AA, A/D Converters 0 and 1 Application Note	U18717E

CONTENTS

[Function]	Starts 16-bit counter operation by setting the TT0CTL0.TT0CE bit to 1. Outputs an interrupt request signal (INTTTEQC00) at an interval set by the TT0CCR0 register. Inverts the TOT00 pin output when the value set by the CCR0 buffer register and the count value of the 16-bit counter match. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Performs count operation of an fxx/64 count clock, generates an interrupt upon the count (1 ms) subsequent to the count whose value matches the value of the TT0CCR0 register (499), and clears 16-bit counter.
[SFR used]	None
[call function]	timermt_interval_ini, timermt_interval_st
[Variable]	None
[Interrupt]	timermt_TT0CCR0_int
[Interrupt source]	INTTTEQC00
[File name]	timermt_interval\MAIN.C
[Caution]	None

The interval time can be calculated by the following formula.

Interval = (Set value of TT0CCR0 register + 1) × Count clock cycle

[Function name]	timermt_interval_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts .
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 1 (Disables INTTTEQC01 interrupt.)
[call function]	timermt_interval
[Variable]	None
[File name]	timermt_interval\timermt_1.c
[Caution]	None

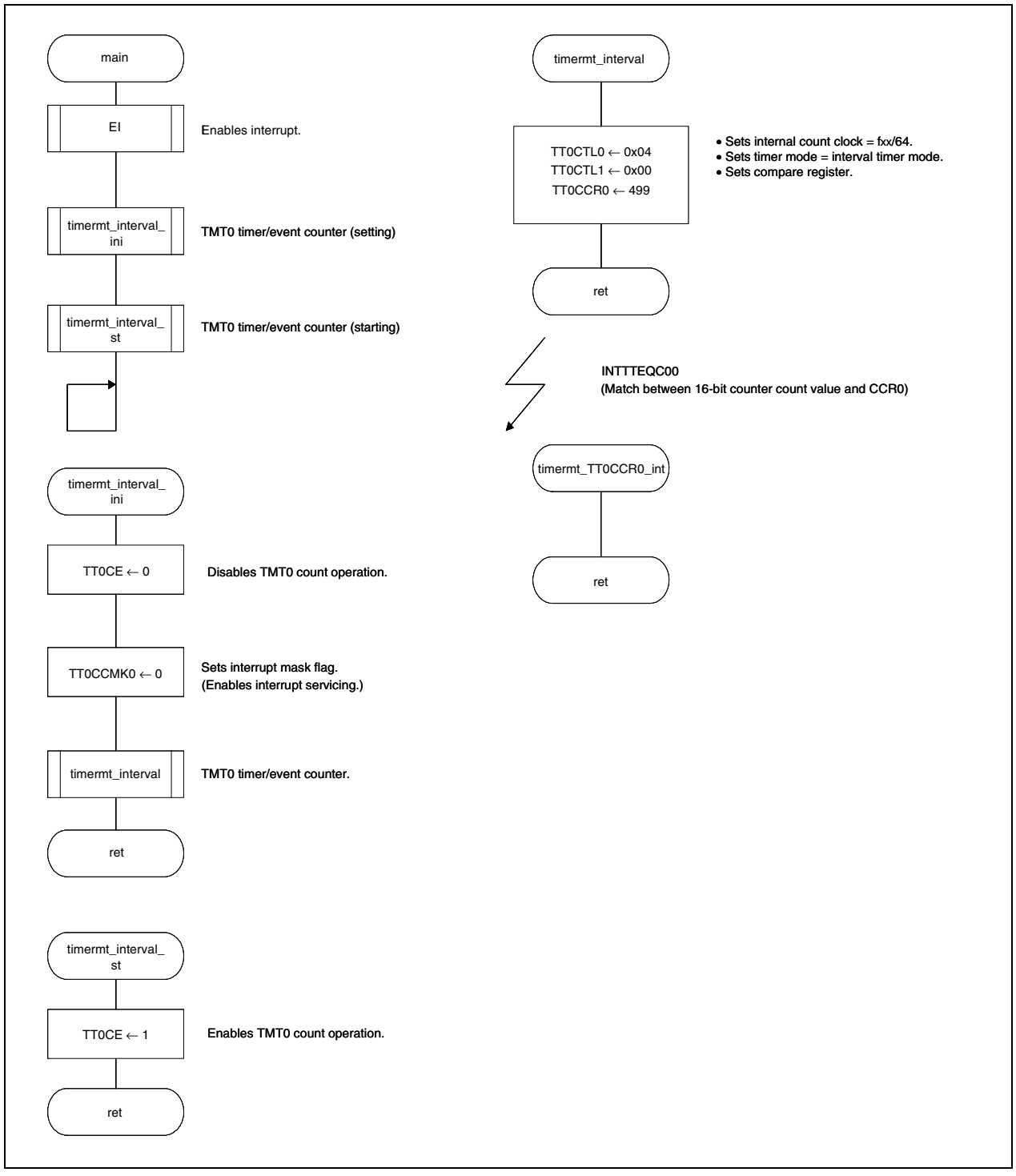
[Function name]	timermt_interval
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0: 0x04 (Sets count clock to fxx/64.) TT0CTL1: 0x00 (Sets interval timer mode.) TT0CCR0: 499 (Compare register of 16-bit counter) TT0CCR1: 0xFFFF (Compare register of 16-bit counter)
[call function]	None
[Variable]	None
[File name]	timermt_interval\timermt_1.c
[Caution]	When the TT0CCR1 register is not used, it is recommended to set the TT0CCR1 register to FFFFH. Mask with the interrupt mask flag (IMR2.TT0CCMK1).

[Function name]	timermt_interval_st
[Argument]	None
[Processing content]	Starting function of timermt_interval.
[Starting method]	Call this function after the timermt_interval function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_interval\timermt_1.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and the value of the CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_interval\timermt_1.c	
[Caution]	None	

Figure 1-1. Interval Timer Mode



CHAPTER 2 EXTERNAL EVENT COUNT MODE

[Function]	Counts the valid edge of the external event count input (EVTT0 pin) and generates an interrupt request signal (INTTTEQC00) each time the count value set by the TT0CCR0 register is counted. (Clears the 16-bit counter simultaneously.) Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Counts the valid edge of the external event count input, generates an interrupt upon the count subsequent to the count whose value matches the value of the CCR0 buffer register, and clears the 16-bit counter.
[SFR used]	None
[call function]	timermt_event_count_ini, timermt_event_count_st
[Variable]	None
[Interrupt]	timermt_TT0CCR0_int timermt_TT0CCR1_int
[Interrupt source]	INTTTEQC00 INTTTEQC01
[File name]	timermt_event_count\MAIN.C
[Caution]	None

[Function name]	timermt_event_count_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, and sets the alternate function of the P05 pin to the EVTT0 input pin.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x20 (Sets EVTT0 input pin.) PFCE0: 0x00 (Sets EVTT0 input pin.) PMC0: 0x20 (Sets EVTT0 input pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 1 (Disables INTTTEQC01 interrupt.)
[call function]	timermt_event_count
[Variable]	None
[File name]	timermt_event_count\timermt_2.c
[Caution]	None

[Function name]	timermt_event_count
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) TT0CTL1: 0x21 (Sets external event count mode.) TT0IOC2: 0x08 (Sets valid edge of the external event count input signal (EVTT0 pin) to falling edge detection.) TT0CCR0: 40 (Compare register of 16-bit counter) TT0CCR1: 0xFFFF (Compare register of 16-bit counter)
[call function]	None
[Variable]	None
[File name]	timermt_event_count\timermt_2.c
[Caution]	When the TT0CCR1 register is not used, it is recommended to set the TT0CCR1 register to FFFFH. Mask with the interrupt mask flag (IMR2.TT0CCMK1).

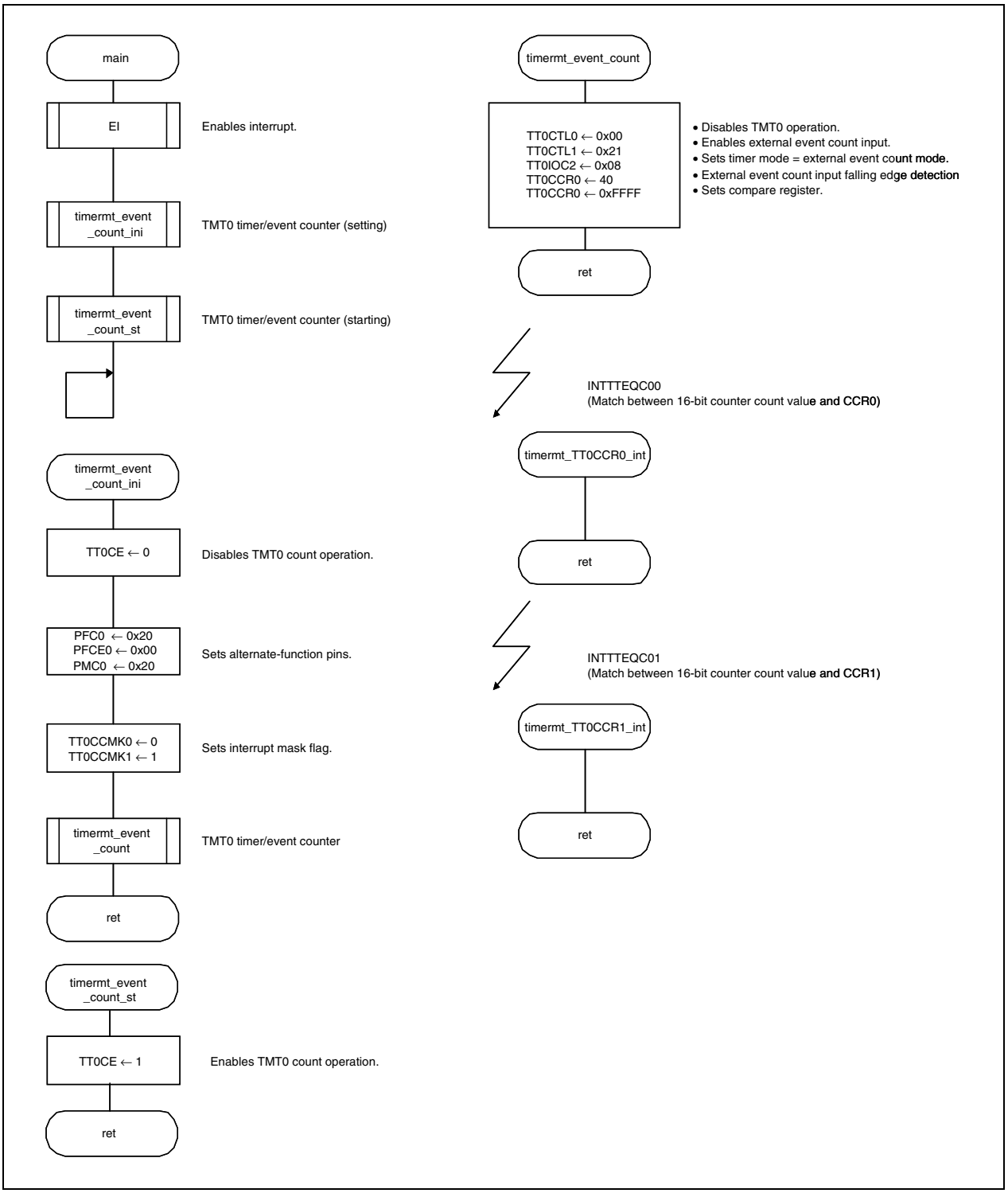
[Function name]	timermt_event_count_st
[Argument]	None
[Processing content]	Starting function of timermt_event_count.
[Starting method]	Call this function after calling the timermt_event_count function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_event_count\timermt_2.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and the value of the CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_event_count\timermt_2.c	
[Caution]	None	

[Function name]	timermt_TT0CCR1_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC01	Match between the count value of the 16-bit counter and the value of the CCR1 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_event_count\timermt_2.c	
[Caution]	None	

Figure 2-1. External Event Count Mode



CHAPTER 3 EXTERNAL TRIGGER PULSE OUTPUT MODE

[Function]	Starts operation of the 16-bit counter when the valid edge of the external trigger input (EVTT0 pin) is detected. Clears the 16-bit counter upon a match between the CCR0 buffer register value and the 16-bit counter value. Inverts the TOT01 pin output upon a match between the CCR1 buffer register value and the 16-bit counter value. Inverts the TOT01 pin when the 16-bit counter is cleared. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Starts count operation of an fxx/16 count clock when the valid edge of the external trigger input is detected, generates an interrupt upon the count subsequent to the count whose value matches the value of the CCR0 buffer register, and clears the 16-bit counter. Generates an interrupt by inverting the TOT01 pin output upon a match between the CCR1 buffer register value and the 16-bit counter value. TOT01 pin starts output at high level.
[SFR used]	None
[call function]	timermt_trigger_pulse_ini, timermt_trigger_pulse_st
[Variable]	None
[Interrupt]	timermt_TT0CCR0_int timermt_TT0CCR1_int
[Interrupt source]	INTTTEQC00 INTTTEQC01
[File name]	timermt_trigger_pulse\MAIN.C
[Caution]	None

The active level width, cycle and duty factor of the PWM waveform can be calculated by the following formula.

Active level width = (Set value of TT0CCR1 register) × Count clock cycle

Cycle = (Set value of TT0CCR0 register + 1) × Count clock cycle

Duty factor = (Set value of TT0CCR1 register)/(Set value of TT0CCR0 register + 1)

[Function name]	timermt_trigger_pulse_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, sets the alternate function of the P05 pin to EVTT0 input pin, and sets alternate function of the P06 pin to TOT01 output.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x60 (Sets EVTT0 input pin, TOT01 output pin.) PFCE0: 0x00 (Sets EVTT0 input pin, TOT01 output pin.) PMC0: 0x60 (Sets EVTT0 input pin, TOT01 output pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 0 (Enables INTTTEQC01 interrupt.)
[call function]	timermt_trigger_pulse
[Variable]	None
[File name]	timermt_trigger_pulse\timermt_3.c
[Caution]	None

[Function name]	timermt_trigger_pulse
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0: 0x03 (Sets count clock to fxx/16.) TT0CTL1: 0x02 (Sets external trigger pulse output mode.) TT0IOC0: 0x04 (Sets TOT00 pin output to high-level start, disables timer output, sets TOT01 pin output to high-level start, enables timer output.) TT0IOC2: 0x02 (Sets valid edge of the external trigger input signal (EVTT0 pin) to falling edge detection.) TT0CCR0: 2499 (Compare register of 16-bit counter) TT0CCR1: 1249 (Compare register of 16-bit counter)
[call function]	None
[Variable]	None
[File name]	timermt_trigger_pulse\timermt_3.c
[Caution]	The compare registers are rewritten in the batch write mode. To rewrite the compare register during timer operation, rewrite the TT0CCR1 register last.

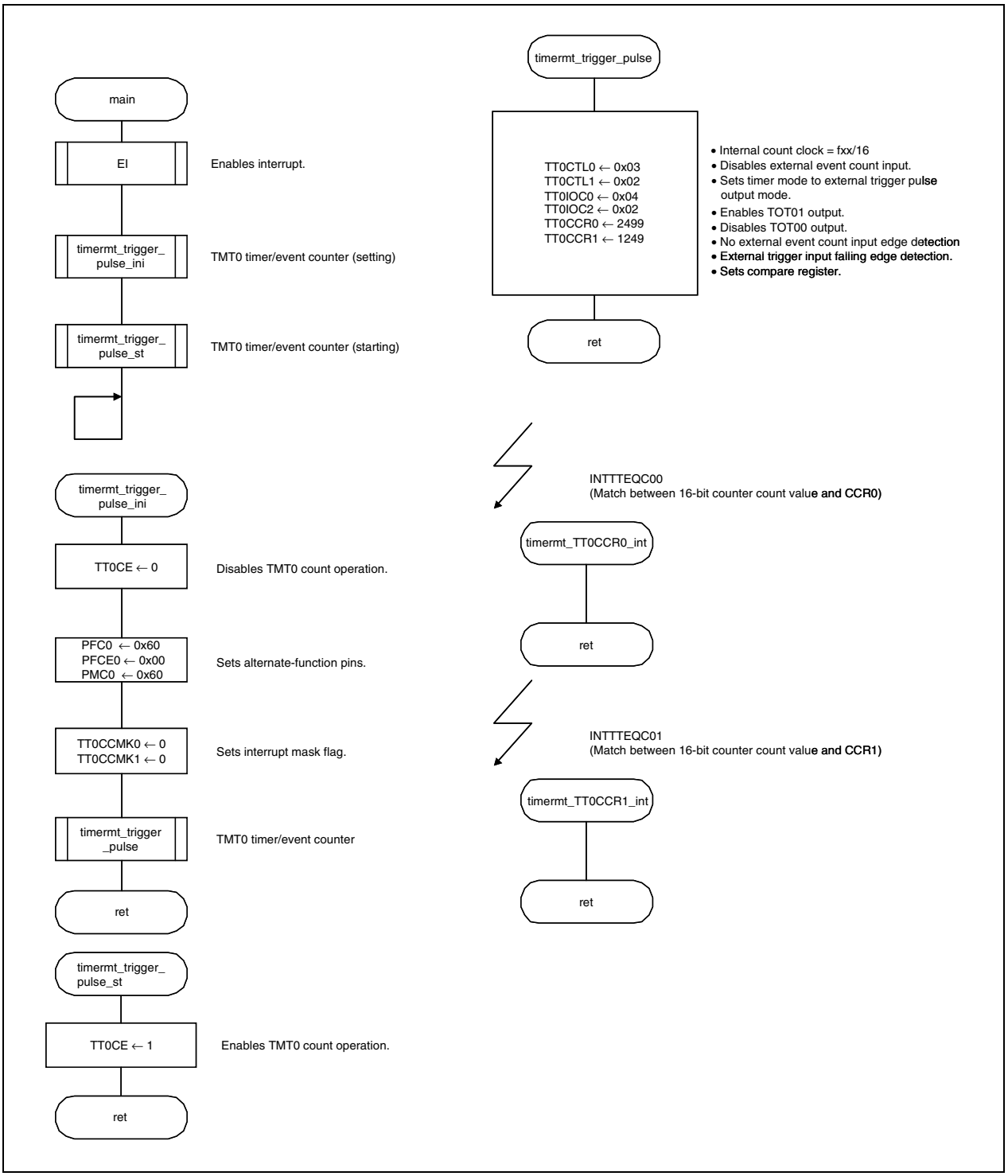
[Function name]	timermt_trigger_pulse_st
[Argument]	None
[Processing content]	Starting function of timermt_trigger_pulse
[Starting method]	Call this function after calling the timermt_trigger_pulse function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_trigger_pulse\timermt_3.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_trigger_pulse\timermt_3.c	
[Caution]	None	

[Function name]	timermt_TT0CCR1_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC01	Match between the count value of the 16-bit counter and CCR1 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_trigger_pulse\timermt_3.c	
[Caution]	None	

Figure 3-1. External Trigger Pulse Output Mode



CHAPTER 4 ONE-SHOT PULSE OUTPUT MODE

[Function]	Starts operation of the 16-bit counter when the valid edge of the external trigger input (EVTT0 pin) is detected. Clears the 16-bit counter upon a match between the CCR0 buffer register value and the 16-bit counter value, and stops count operation. Inverts the TOT01 pin upon a match between the CCR1 buffer register value and the 16-bit counter value. Inverts the TOT01 pin when the 16-bit counter is cleared. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Starts count operation of an fxx/16 count clock when the valid edge of the external trigger input is detected, generates an interrupt upon the count subsequent to the count whose value matches the value of the CCR0 buffer register by inverting the TOT00 pin output, and clears the 16-bit counter to stop the count operation. Generates an interrupt by inverting the TOT01 pin output upon a match between the CCR1 buffer register value and the 16-bit counter value. TOT01 pin starts output at high level.
[SFR used]	None
[call function]	timermt_1shot_pulse_ini, timermt_1shot_pulse_st
[Variable]	None
[Interrupts]	timermt_TT0CCR0_int timermt_TT0CCR1_int
[Interrupt sources]	INTTTEQC00 INTTTEQC01
[File name]	timermt_1shot_pulse\MAIN.C
[Caution]	None

The output delay time and active level width of one-shot pulse can be calculated by the following formula.

Output delay time = (Set value of TT0CCR1 register) × Count clock cycle

Active level width = (Set value of TT0CCR0 register − Set value of TT0CCR1 register + 1) × Count clock cycle

[Function name]	timermt_1shot_pulse_ini	
[Argument]	None	
[Processing content]	Sets the TMT0 operation and interrupts, set the alternate function of the P05 pin to EVTT0 input pin, and sets alternate function of the P06 pin to TOT01 output.	
[SFRs used]	TT0CTL0.TT0CE:	0 (Disables TMT0 operation.)
	PFC0:	0x60 (Sets EVTT0 input pin, TOT01 output pin.)
	PFCE0:	0x00 (Sets EVTT0 input pin, TOT01 output pin.)
	PMC0:	0x60 (Sets EVTT0 input pin, TOT01 output pin.)
	IMR2.TT0CCMK0:	0 (Enables INTTTEQC00 interrupt.)
	IMR2.TT0CCMK1:	0 (Enables INTTTEQC01 interrupt.)
[call function]	timermt_1shot_pulse	
[Variable]	None	
[File name]	timermt_1shot_pulse\timermt_4.c	
[Caution]	None	

[Function name]	timermt_1shot_pulse	
[Argument]	None	
[Processing content]	Sets TMT0 control register.	
[SFRs used]	TT0CTL0:	0x03 (Sets count clock to fxx/16.)
	TT0CTL1:	0x03 (Sets one-shot pulse output mode.)
	TT0IOC0:	0x04 (Sets TOT00 pin output to high-level start, disables timer output, sets TOT01 pin output to high-level start, enables timer output.)
	TT0IOC2:	0x02 (Sets valid edge of the external trigger input signal (EVTT0 pin) to falling edge detection.)
	TT0CCR0:	2499 (Compare register of 16-bit counter)
	TT0CCR1:	1249 (Compare register of 16-bit counter)
[call function]	None	
[Variable]	None	
[File name]	timermt_1shot_pulse\timermt_4.c	
[Caution]	None	

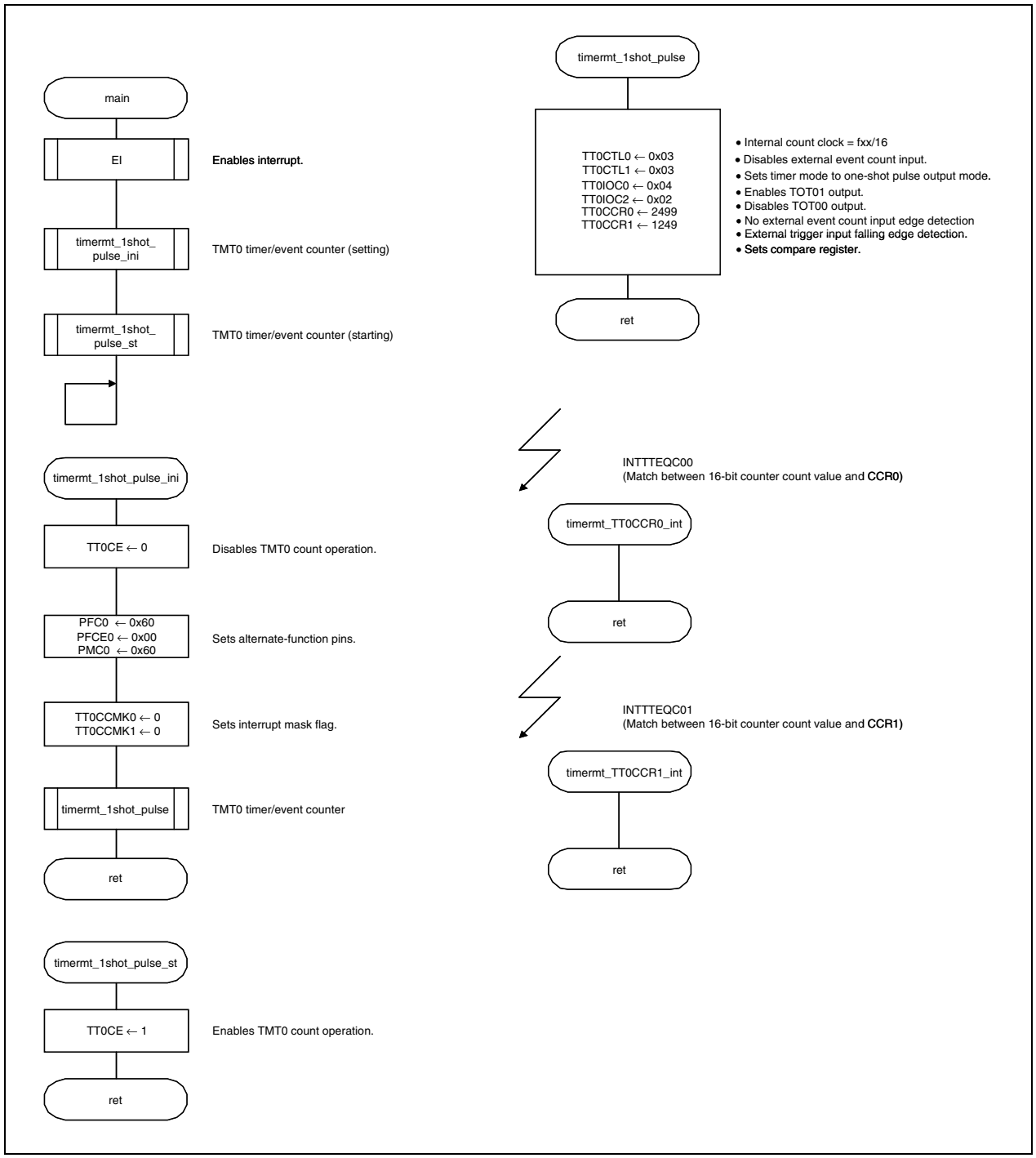
[Function name]	timermt_1shot_pulse_st
[Argument]	None
[Processing content]	Starting function of timermt_1shot_pulse
[Starting method]	Call this function after calling the timermt_1shot_pulse function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_1shot_pulse\timermt_4.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and the value of the CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_1shot_pulse\timermt_4.c	
[Caution]	None	

[Function name]	timermt_TT0CCR1_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC01	Match between the count value of the 16-bit counter and the value of the CCR1 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_1shot_pulse\timermt_4.c	
[Caution]	None	

Figure 4-1. One-Shot Pulse Output Mode



CHAPTER 5 PWM OUTPUT MODE

[Function]	Starts operation of the 16-bit counter by setting the TT0CTL0.TT0CE bit to 1. Clears the 16-bit counter upon a match with the CCR0 buffer register value, inverts the TOT00 pin output, and outputs a PWM waveform with a 50% duty factor whose half cycle is equal to the set value of the TT0CCR0 register + 1. Inverts the TOT01 pin upon a match between the CCR1 buffer register value and the 16-bit counter value. Inverts the TOT01 pin when the 16-bit counter is cleared. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Performs count operation of an fxx/16 count clock, generates an interrupt by inverting the TOT00 pin output upon the count subsequent to the count whose value matches the value of the CCR0 buffer register, and clears the 16-bit counter. Generates an interrupt by inverting the TOT01 pin output upon a match between the CCR1 buffer register value and the 16-bit counter value. Both TOT00 pin and TOT01 pin start output at high level.
[SFR used]	None
[call function]	timermt_pwm_output_ini, timermt_pwm_output_st
[Variable]	None
[Interrupts]	timermt_TT0CCR0_int timermt_TT0CCR1_int
[Interrupt sources]	INTTTEQC00 INTTTEQC01
[File name]	timermt_pwm_output\MAIN.C
[Caution]	None

The active level width, cycle, and duty factor of the PWM waveform output from the TOT01 pin can be calculated by the following formula.

Active level width = (Set value of TT0CCR1 register) × Count clock cycle

Cycle = (Set value of TT0CCR0 register + 1) × Count clock cycle

Duty factor = (Set value of TT0CCR1 register) / (Set value of TT0CCR0 register + 1)

[Function name]	timermt_pwm_output_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, sets the alternate function of the P04 pin to TOT00 output pin, and sets alternate function of the P06 pin to TOT01 output.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x50 (Sets TOT00 output pin, TOT01 output pin.) PFCE0: 0x00 (Sets TOT00 output pin, TOT01 output pin.) PMC0: 0x50 (Sets TOT00 output pin, TOT01 output pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 0 (Enables INTTTEQC01 interrupt.)
[call function]	timermt_pwm_output
[Variable]	None
[File name]	timermt_pwm_output\timermt_5.c
[Caution]	None

[Function name]	timermt_pwm_output
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0: 0x03 (Sets count clock to fxx/16.) TT0CTL1: 0x04 (Sets PWM output mode.) TT0IOC0: 0x05 (Sets TOT00 pin output to high-level start, enables timer output, sets TOT01 pin output to high-level start, enables timer output.) TT0IOC2: 0x00 (Sets valid edge of the external event count input signal (EVTT0 pin) to no edge detection.) TT0CCR0: 2499 (Compare register of 16-bit counter) TT0CCR1: 1999 (Compare register of 16-bit counter)
[call function]	None
[Variable]	None
[File name]	timermt_pwm_output\timermt_5.c
[Caution]	The compare registers are rewritten in the batch write mode. To rewrite the compare register during timer operation, rewrite the TT0CCR1 register last.

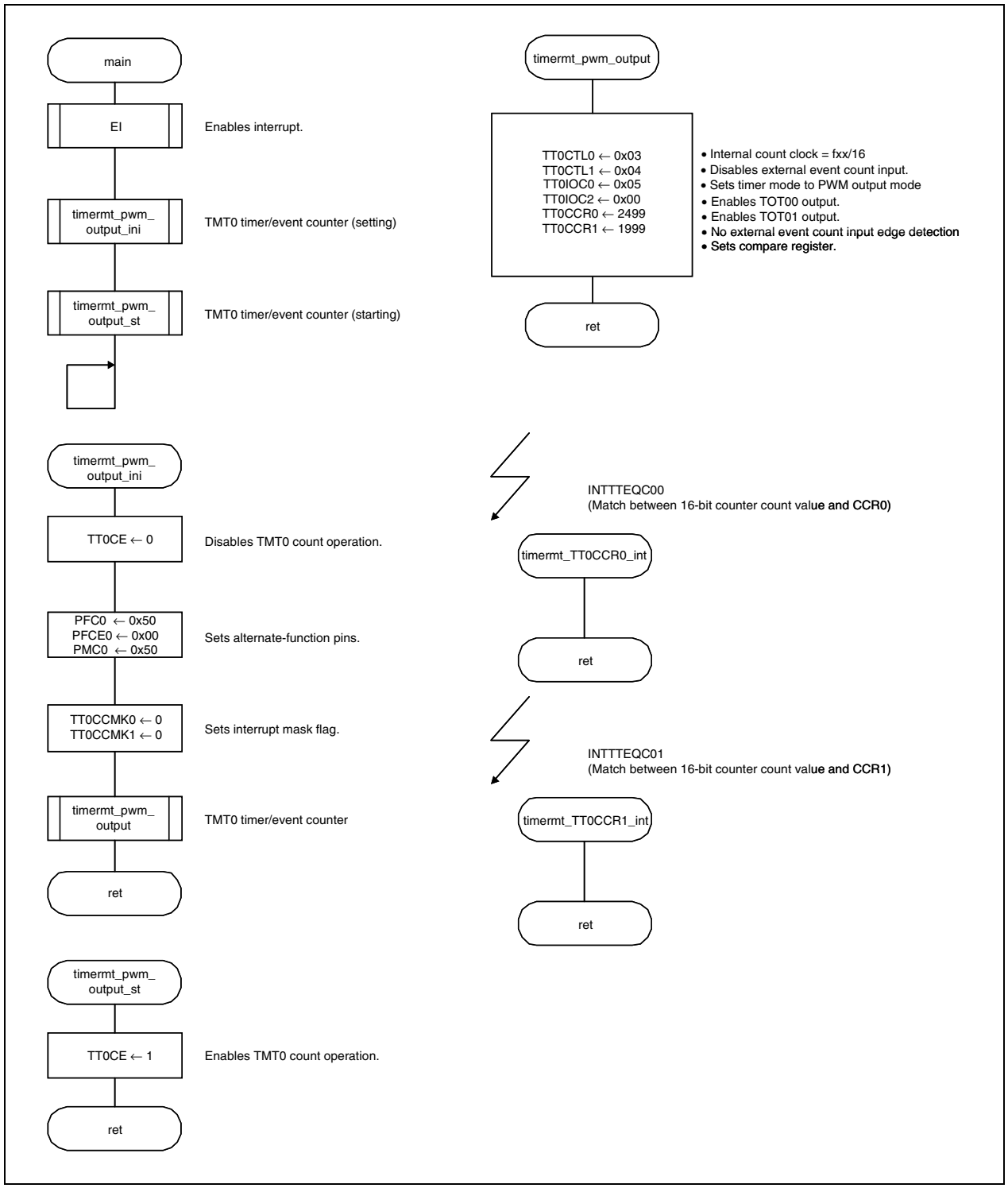
[Function name]	timermt_pwm_output_st
[Argument]	None
[Processing content]	Starting function of timermt_pwm_output
[Starting method]	Call this function after calling the timermt_pwm_output function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_pwm_output\timermt_5.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_pwm_output\timermt_5.c	
[Caution]	None	

[Function name]	timermt_TT0CCR1_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC01	Match between the count value of the 16-bit counter and CCR1 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_pwm_output\timermt_5.c	
[Caution]	None	

Figure 5-1. PWM Output Mode



CHAPTER 6 FREE-RUNNING TIMER MODE

[Function]	Starts operation of the 16-bit counter by setting the TT0CTL0.TT0CE bit to 1. Inverts the TOT00 pin output upon a match between the CCR0 buffer register and the count value of the 16-bit counter (compare function). Stores the count value of the 16-bit counter in the TT0CCR1 register when the valid edge of the capture trigger input (TIT01 pin) is detected (capture function). After counting up to FFFFH, generates an overflow interrupt request signal (INTTTIOV0) at the next clock and clears to 0000H to continue counting. The compare function and capture function can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Performs count operation of an fxx/16 count clock, generates an interrupt by inverting the TOT00 pin output upon the count subsequent to the count whose value matches the value of the CCR0 buffer register, and clears the 16-bit counter. Generates an interrupt by capturing the 16-bit counter value to the TT0CCR1 register when a valid edge is detected from the TIP01 pin. Generates an INTTTIOV0 interrupt when a 16-bit counter overflow is detected. The TOT00 pin starts output at high level.
[SFR used]	None
[call function]	timermt_free_running_ini, timermt_free_running_st
[Variable]	None
[Interrupts]	timermt_TT0CCR0_int timermt_TT0CCR1_int timermt_TTTIOV0_int
[Interrupt sources]	INTTTEQC00 INTTTEQC01 INTTTIOV0
[File name]	timermt_free_running \MAIN.C
[Caution]	None

[Function name]	timermt_free_running_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, sets the alternate function of the P04 pin to TOT00 output pin, and sets alternate function of the P06 pin to TIT01 input.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x10 (Sets TOT00 output pin, TIT01 input pin.) PFCE0: 0x00 (Sets TOT00 output pin, TIT01 input pin.) PMC0: 0x50 (Sets TOT00 output pin, TIT01 input pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 0 (Enables INTTTEQC01 interrupt.) IMR2.TT0OVMK: 0 (Enables INTTTIOV0 interrupt.)
[call function]	timermt_free_running
[Variable]	None
[File name]	timermt_free_running\timermt_6.c
[Caution]	None

[Function name]	timermt_free_running
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0: 0x03 (Sets count clock to fxx/16.) TT0CTL1: 0x05 (Sets free-running timer mode.) TT0IOC0: 0x01 (Sets TOT00 pin output to high-level start, enables timer output.) TT0IOC1: 0x08 (Sets valid edge of the capture trigger input signal (TIT01 pin) to falling edge detection.) TT0IOC2: 0x00 (Sets valid edge of the external event count input signal (EVTT0 pin) to no edge detection.) TT0OPT0: 0x20 (Sets TT0CCR0 as compare register, and TT0CCR1 as capture register.) TT0CCR0: 2499 (Compare register of 16-bit counter)
[call function]	None
[Variable]	None
[File name]	timermt_free_running\timermt_6.c
[Caution]	None

[Function name]	timermt_free_running_st
[Argument]	None
[Processing content]	Starting function of timermt_free_running
[Starting method]	Call this function after calling the timermt_free_running function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_free_running\timermt_6.c
[Caution]	None

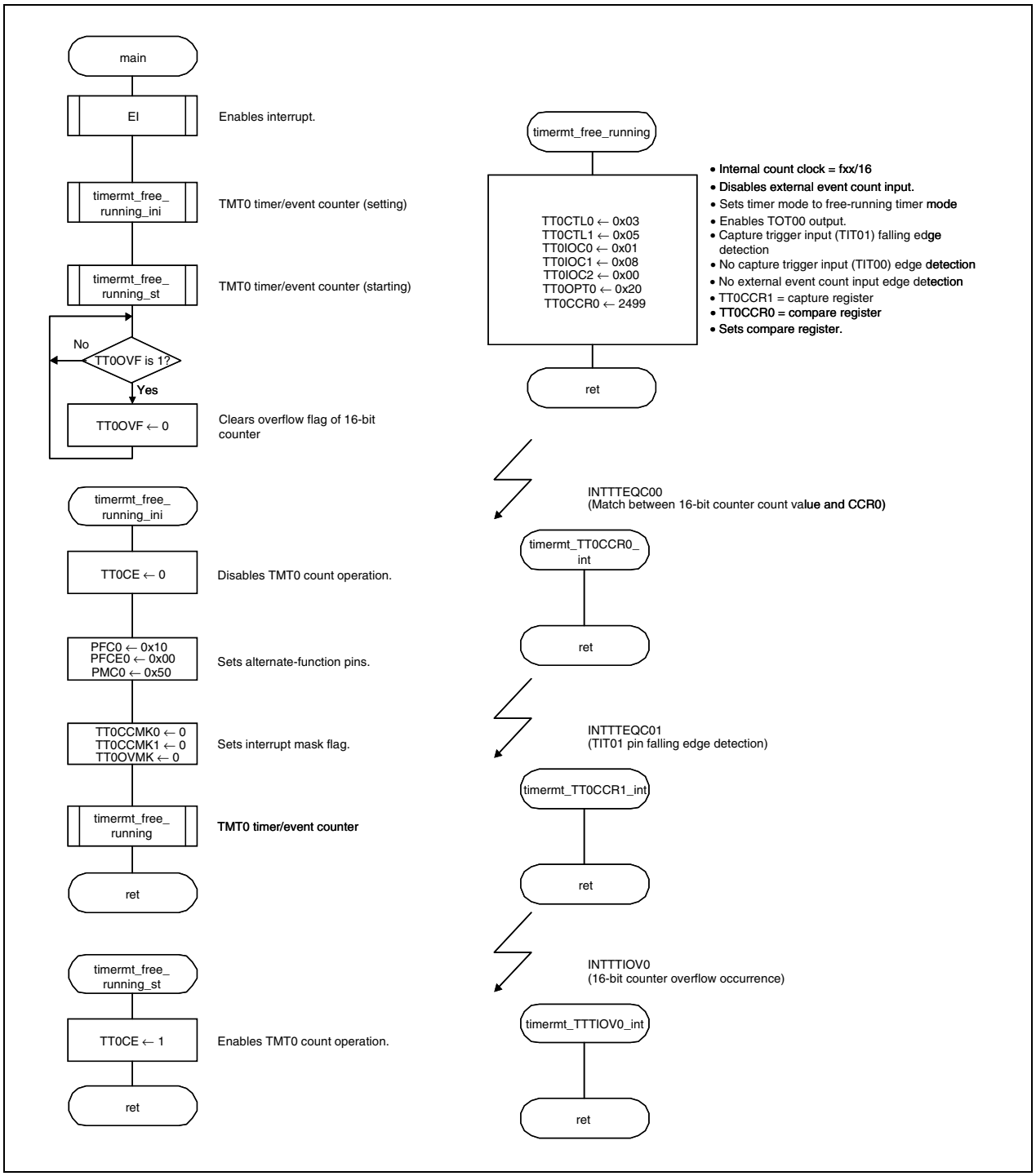
Interrupt function

[Function name]	timermt_TT0CCR0_int
[Overview]	Defined by the user.
[Source]	INTTTEQC00 Match between the count value of the 16-bit counter and the value of the CCR0 buffer register
[call function]	None
[Variable]	None
[File name]	timermt_free_running\timermt_6.c
[Caution]	None

[Function name]	timermt_TT0CCR1_int
[Overview]	Defined by the user.
[Source]	INTTTEQC01 Detects valid edge of the TIT01 pin input.
[call function]	None
[Variable]	None
[File name]	timermt_free_running\timermt_6.c
[Caution]	None

[Function name]	timermt_TTTIOV0_int
[Overview]	Defined by the user.
[Source]	INTTTIOV0 16-bit counter overflow occurrence
[call function]	None
[Variable]	None
[File name]	timermt_free_running\timermt_6.c
[Caution]	None

Figure 6-1. Free-Running Timer Mode



CHAPTER 7 PULSE WIDTH MEASUREMENT MODE

[Function]	Starts operation of the 16-bit counter by setting the TT0CTL0.TT0CE bit to 1. Stores the count value of the 16-bit counter to the TT0CCR0 register when the valid edge of the capture trigger input (TIT00 pin) is detected, and clears the 16-bit counter. Measures the valid edge interval of the TIT00 pin by generating an interrupt when the valid edge of the TIT00 pin input is detected and reading the TT0CCR0 register value. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Performs count operation of an fxx/16 count clock, generates an interrupt by storing the count value of the 16-bit counter to the TT0CCR0 register when the valid edge of the TIT00 pin input is detected, and clears the 16-bit counter. Generates an INTTTIOV0 interrupt when a 16-bit counter overflow is detected.
[SFR used]	None
[call function]	timermt_pulse_measure_ini, timermt_pulse_measure_st
[Variable]	None
[Interrupt]	timermt_TT0CCR0_int timermt_TTTIOV0_int
[Interrupt source]	INTTTEQC00 INTTTIOV0
[File name]	timermt_pulse_measure\MAIN.C
[Caution]	None

The pulse width can be calculated by the following formula.

Pulse width = Captured value × Count clock cycle

The pulse width when the 16-bit counter overflow is detected can be calculated by the following formula.

Pulse width = (10000H × number of times the TT0OVF bit is set + Captured value) × Count clock cycle

[Function name]	timermt_pulse_measure_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, and sets the alternate function of the P04 pin to TIT00 input pin.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x00 (Sets TIT00 input pin.) PFCE0: 0x00 (Sets TIT00 input pin.) PMC0: 0x10 (Sets TIT00 input pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 1 (Disables INTTTEQC01 interrupt.) IMR2.TT0OVMK: 0 (Enables INTTTIOV0 interrupt.)
[call function]	timermt_pulse_measure
[Variable]	None
[File name]	timermt_pulse_measure\timermt_7.c
[Caution]	None

[Function name]	timermt_pulse_measure
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0: 0x03 (Sets count clock to fxx/16.) TT0CTL1: 0x06 (Sets pulse width measurement mode.) TT0IOC1: 0x02 (Sets valid edge of the capture trigger input signal (TIT00 pin) to falling edge detection.) TT0IOC2: 0x00 (Sets valid edge of the external event count input signal (EVTT0 pin) to no edge detection.)
[call function]	None
[Variable]	None
[File name]	timermt_pulse_measure\timermt_7.c
[Caution]	None

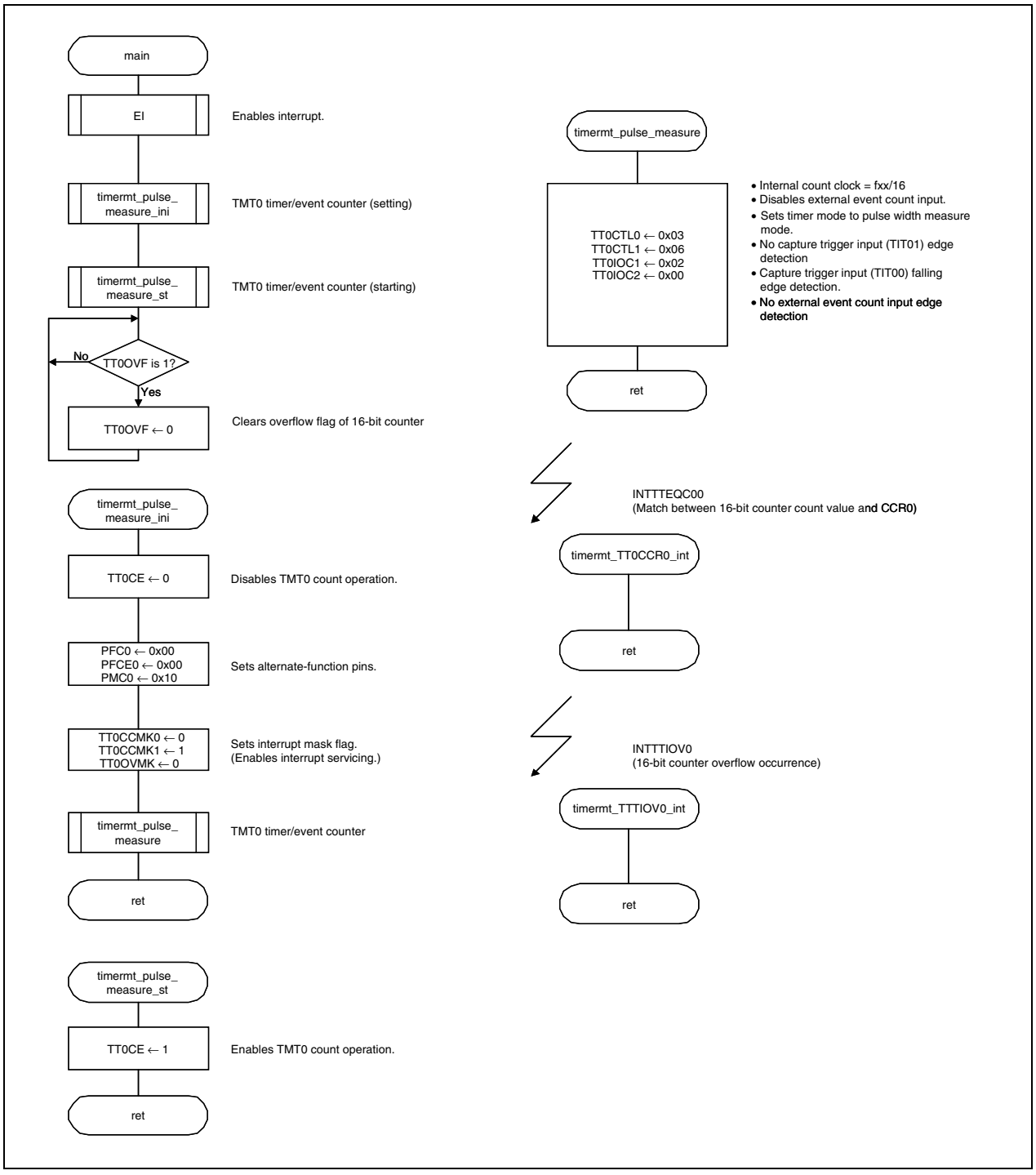
[Function name]	timermt_pulse_measure_st
[Argument]	None
[Processing content]	Starting function of timermt_pulse_measure
[Starting method]	Call this function after calling the timermt_pulse_measure function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_pulse_measure\timermt_7.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int
[Overview]	Defined by the user.
[Source]	INTTTEQC00 TIT00 pin input valid edge detection
[call function]	None
[Variable]	None
[File name]	timermt_pulse_measure\timermt_7.c
[Caution]	None

[Function name]	timermt_TTTIOV0_int
[Overview]	Defined by the user.
[Source]	INTTTIOV0 16-bit counter overflow occurrence
[call function]	None
[Variable]	None
[File name]	timermt_pulse_measure\timermt_7.c
[Caution]	None

Figure 7-1. Pulse Width Measurement Mode



[Function]	Starts operation of the 16-bit counter by setting the TT0CTL0.TT0CE bit to 1. Performs 16-bit counter count up/down, and generates an overflow interrupt request signal (INTTTIOV0) when the 16-bit counter count value reaches 0000H during counting down. Generates a compare match interrupt request signal (INTTTEQC00) when the 16-bit counter count value and CCR0 buffer register value match during counting up. Inverts the TOT01 pin output upon a match (INTTTEQC01) between the 16-bit counter count value and CCR1 buffer register value, and outputs the triangular-wave PWM waveform. Also, inverts the TOT00 pin output upon a match (INTTTEQC01) between the overflow interrupt request signal (INTTTIOV0), 16-bit counter count value, and the CCR1 buffer register value, and outputs the PWM waveform. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Performs 16-bit counter-count up operation of an fxx/16 count clock, generates a compare match interrupt request signal (INTTTEQC00) upon a match between the 16-bit counter count value and CCR0 buffer register value, and 16-bit counter switches to the count-down operation. Generates an overflow interrupt request signal (INTTTIOV0) when the 16-bit counter count value reaches 0000H during counting down, and 16-bit counter switches to count-up operation.
[SFR used]	None
[call function]	timermt_chopping_wave_ini, timermt_chopping_wave_st
[Variable]	None
[Interrupts]	timermt_TT0CCR0_int timermt_TT0CCR1_int timermt_TTTIOV0_int
[Interrupt sources]	INTTTEQC00 INTTTEQC01 INTTTIOV0
[File name]	timermt_chopping_wave\MAIN.C
[Caution]	None

[Function name]	timermt_chopping_wave_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, and sets the alternate-function of the P06 pin to TOT01 output pin.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x40 (Sets TOT01 output pin.) PFCE0: 0x00 (Sets TOT01 output pin.) PMC0: 0x40 (Sets TOT01 output pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 0 (Enables INTTTEQC01 interrupt.) IMR2.TT0OVMK: 0 (Enables INTTTIOV0 interrupt.)
[call function]	timermt_chopping_wave
[Variable]	None
[File name]	timermt_chopping_wave\timemt_8.c
[Caution]	None

[Function name]	timermt_chopping_wave
[Argument]	None
[Processing content]	Sets TMT0 control register.
[SFRs used]	TT0CTL0: 0x03 (Sets count clock to fxx/16.) TT0CTL1: 0x07 (Sets triangular-wave PWM output mode.) TT0IOC0: 0x04 (Sets TOT01 pin output to high-level start, and enables timer output.) TT0IOC2: 0x00 (Sets valid edge of the external event count input signal (EVTT0 pin) to no edge detection.) TT0OPT0: 0x00 (Sets the TT0CCR0 and TT0CCR1 as compare registers.) TT0CCR0: 2499 (Compare register of 16-bit counter) TT0CCR1: 1499 (Compare register of 16-bit counter)
[call function]	None
[Variable]	None
[File name]	timermt_chopping_wave\timemt_8.c
[Caution]	The compare registers are rewritten in the batch write mode. To rewrite the compare register during timer operation, rewrite the TT0CCR1 register last.

[Function name]	timermt_chopping_wave_st
[Argument]	None
[Processing content]	Starting function of timermt_chopping_wave
[Starting method]	Call this function after calling the timermt_chopping_wave function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_chopping_wave\timemt_8.c
[Caution]	None

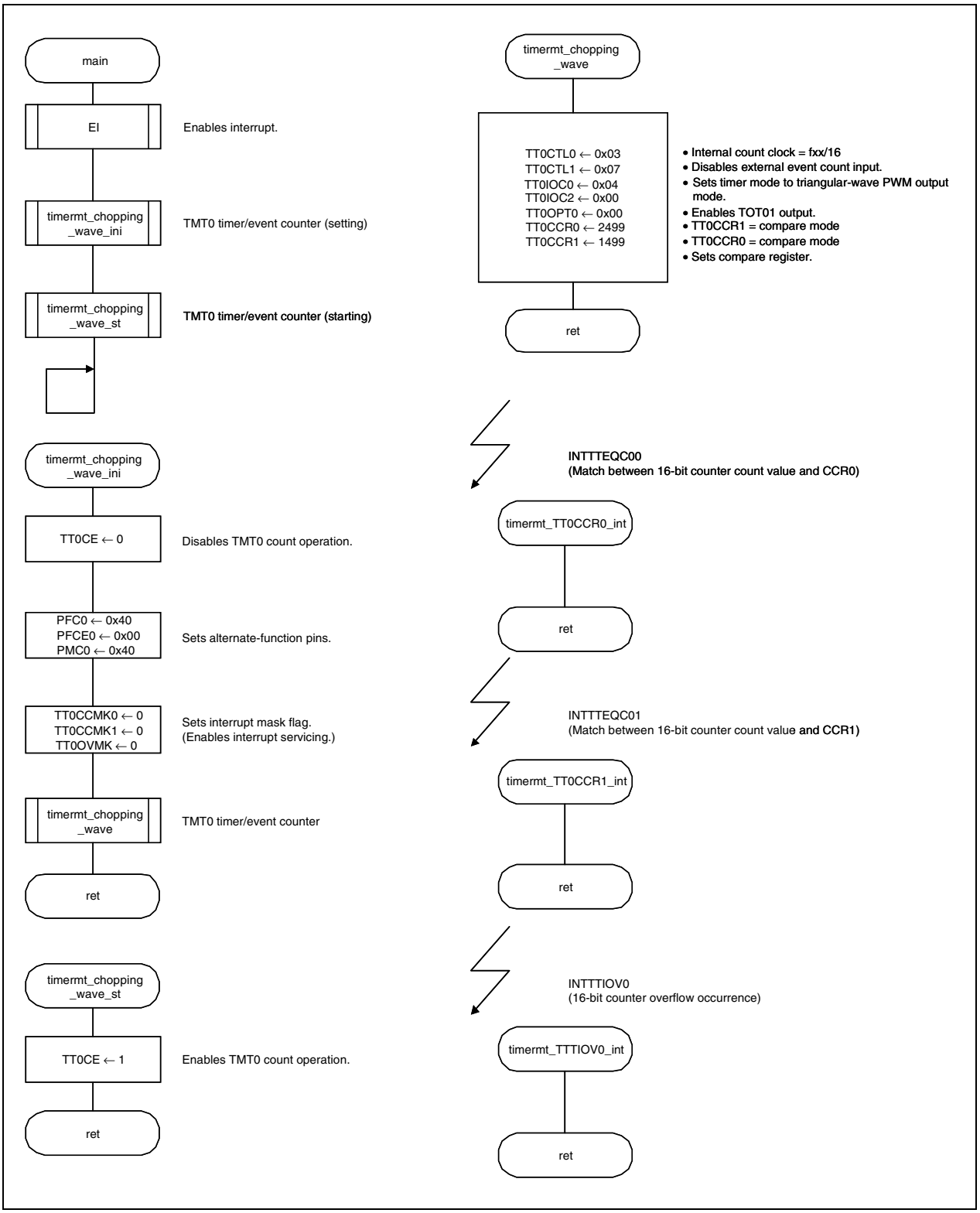
Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and the value of the CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_chopping_wave\timemt_8.c	
[Caution]	None	

[Function name]	timermt_TT0CCR1_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC01	Match between the count value of the 16-bit counter and the value of the CCR1 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_chopping_wave\timemt_8.c	
[Caution]	None	

[Function name]	timermt_TTTIOV0_int
[Overview]	Defined by the user.
[Source]	INTTTIOV0 16-bit counter overflow occurrence
[call function]	None
[Variable]	None
[File name]	timermt_chopping_wave\timemt_8.c
[Caution]	None

Figure 8-1. Triangular-wave PWM Output Mode



CHAPTER 9 ENCODER COMPARE MODE

[Function]	Starts the 16-bit counter operation by setting the TT0CTL0.TT0CE bit to 1 to transfer the TT0TCW register set value to the 16-bit counter. Generates a compare match interrupt request signal (INTTTEQC00) upon a match between the 16-bit counter count value and the CCR0 buffer register value. Generates a compare match interrupt request signal (INTTTEQC01) upon a match between the 16-bit counter count value and CCR1 buffer register value. Can be implemented with TMT0 and TMT1.
[Function name]	main
[Argument]	None
[Processing content]	Performs 16-bit counter count-up operation of an fxx/16 count clock, and generates a compare match interrupt request signal (INTTTEQC00) upon a match between the 16-bit counter count value and CCR0 buffer register value. Clears the 16-bit counter to 0000H if the next count operation is counting up. Generates a compare match interrupt request signal (INTTTEQC01) upon a match between the 16-bit counter count value and CCR1 buffer register value. Clears the 16-bit counter to 0000H if the next count operation is counting down.
[SFR used]	None
[call function]	timermt_encod_compare_ini, timermt_encod_compare_st
[Variable]	None
[Interrupts]	timermt_TT0CCR0_int timermt_TT0CCR1_int
[Interrupt sources]	INTTTEQC00 INTTTEQC01
[File name]	timermt_encod_compare\MAIN.C
[Caution]	None

[Function name]	timermt_encod_compare_ini
[Argument]	None
[Processing content]	Sets the TMT0 operation and interrupts, and set the alternate function of the P05 pin to TENC00 input pin and the P06 pin to TENC01 input pin.
[SFRs used]	TT0CTL0.TT0CE: 0 (Disables TMT0 operation.) PFC0: 0x00 (Sets TENC00 input pin, TENC01 input pin.) PFCE0: 0x00 (Sets TENC00 input pin, TENC01 input pin.) PMC0: 0x60 (Sets TENC00 input pin, TENC01 input pin.) IMR2.TT0CCMK0: 0 (Enables INTTTEQC00 interrupt.) IMR2.TT0CCMK1: 0 (Enables INTTTEQC01 interrupt.)
[call function]	timermt_encod_compare
[Variable]	None
[File name]	timermt_encod_compare\timermt_9.c
[Caution]	None

[Function name]	timermt_encod_compare	
[Argument]	None	
[Processing content]	Sets TMT0 control register.	
[SFRs used]	TT0CTL0:	0x03 (Sets count clock to fxx/16.)
	TT0CTL1:	0x08 (Sets encoder compare mode.)
	TT0CTL2:	Sets to clear the 16-bit counter to 0000H upon a match between the 16-bit counter count value and CCR0 buffer register value, and to clear the 16-bit counter to 0000H upon a match between the 16-bit counter count value and CCR1 buffer register value.
	TT0IOC3:	0x00 (Sets encoder clear signal (TECR0 pin) edge detection clear.)
	TT0TCW:	0x00 (Initializes the 16-bit counter.)
	TT0CCR0:	2499 (Compare register of 16-bit counter)
	TT0CCR1:	1499 (Compare register of 16-bit counter)
[call function]	None	
[Variable]	None	
[File name]	timermt_encod_compare\timermt_9.c	

[Caution]

- If a value which is the same as that of the TT0TCW register is set to the TT0CCR0 or TT0CCR1 register and the counter operation is started when the TT0CTL2.TT0ECC bit = 0, and if the count value (TT0TCW) of the 16-bit counter matches the value of the CCRa buffer register immediately after the start of the operation, the match is masked and the compare match interrupt request signal (INTTTEQC0a) is not generated (a = 0, 1). In addition, the 16-bit counter is not cleared to 0000H by setting the TT0CTL2.TT0ECM1 and TT0CTL2.TT0ECM0 bits.
- If the count operation is resumed when the TT0CTL2.TT0ECC bit = 1, the 16-bit counter does not overflow if its count value that has been held is FFFFH and if the next count operation is counting up.

  After the counter starts operating and counts up from a count value (value of TT0TCW register = FFFFH), the counter overflows from FFFFH to 0000H. However, detection of the overflow is masked, the overflow flag (TT0EOF) is not set, and the overflow interrupt request signal (INTTTIOV0) is not generated.

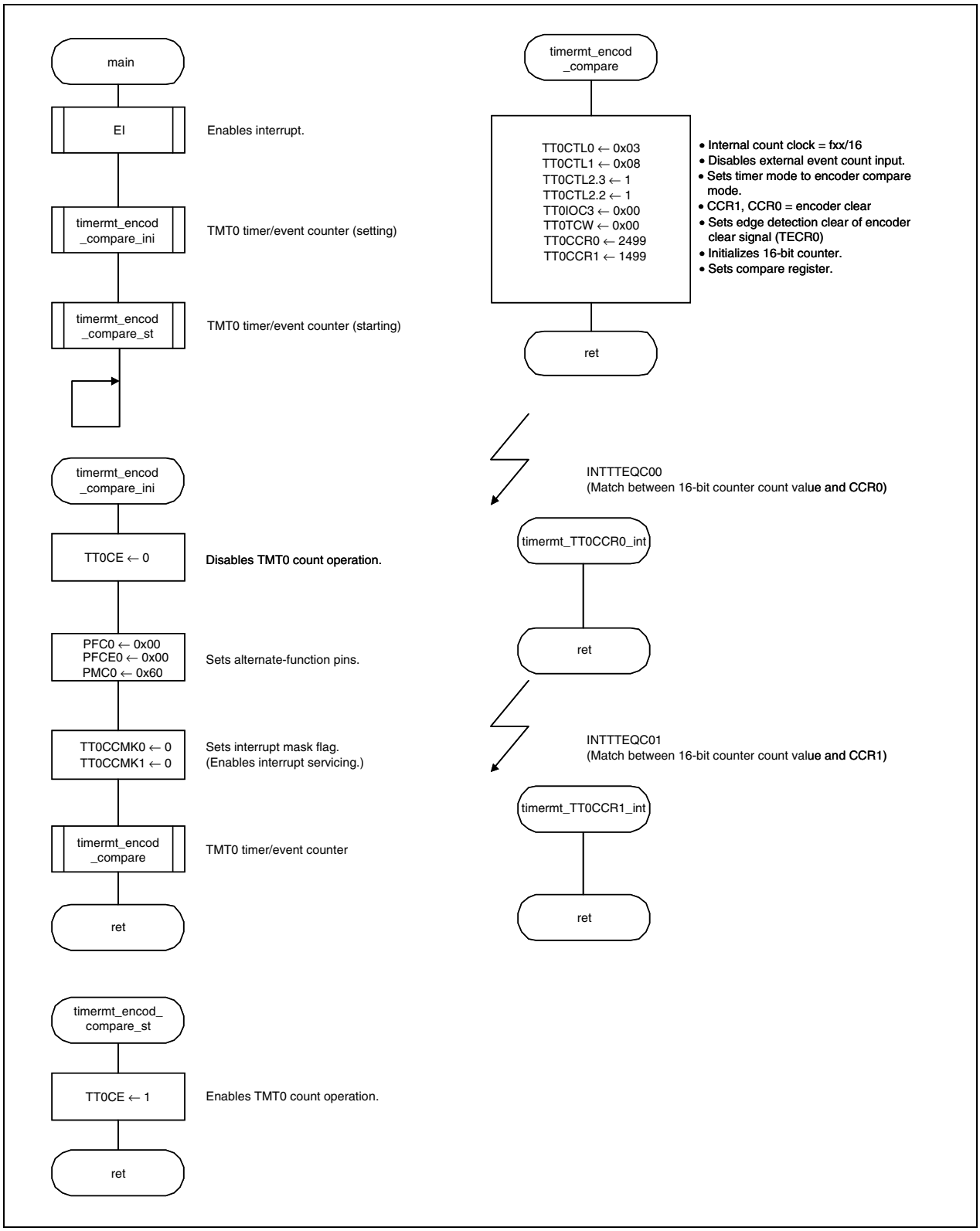
[Function name]	timermt_encod_compare_st
[Argument]	None
[Processing content]	Starting function of timermt_encod_compare
[Starting method]	Call this function after calling the timermt_encod_compare function.
[SFR used]	TT0CTL0.TT0CE: 1 (Enables TMT0 operation.)
[call function]	None
[Variable]	None
[File name]	timermt_encod_compare\timermt_9.c
[Caution]	None

Interrupt function

[Function name]	timermt_TT0CCR0_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC00	Match between the count value of the 16-bit counter and the value of the CCR0 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_encod_compare\timermt_9.c	
[Caution]	None	

[Function name]	timermt_TT0CCR1_int	
[Overview]	Defined by the user.	
[Source]	INTTTEQC01	Match between the count value of the 16-bit counter and the value of the CCR1 buffer register
[call function]	None	
[Variable]	None	
[File name]	timermt_encod_compare\timermt_9.c	
[Caution]	None	

Figure 9-1. Encoder Compare Mode



For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706