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

Application Note

V850ES/Jx3-L

Sample Program
(16-bit Timer/Event Counters (TMP, TMQ))
PWM Output Mode

This document describes an operation overview of the sample program and how to use it, as well as how to set up and use the PWM output function of 16-bit timer/event counters (TMP, TMQ). In the sample program, the brightness of LED2 is changed every 500 ms by using the PWM output function of the 16-bit timer/event counter (TMP) to control the pulse output duty. When the duty is changed, LED1 turns on or off.

Target devices

V850ES/JF3-L microcontrollers
V850ES/JG3-L microcontrollers

CONTENTS

CHAPTER 1 OVERVIEW 3
1.1 Initial Settings 4
1.2 Servicing Interrupts by Using 16-bit Timer/Event Counter (TMP0) Per Specified PWM Cycle 5
CHAPTER 2 CIRCUIT DIAGRAM 6
2.1 Circuit Diagram 6
2.2 Peripheral Hardware 6
CHAPTER 3 SOFTWARE 7
3.1 Included Files 7
3.2 On-Chip Peripherals Used 8
3.3 Initial Settings and Operational Overview 8
3.4 Flowcharts 10
3.5 Differences Between V850ES/JG3-L and V850ES/JF3-L 12
3.6 Difference Between TMP and TMQ 12
3.7 Security ID 12
CHAPTER 4 SETTING UP REGISTERS 13
4.1 Setting Up 16-bit Timer/Event Counter P (TMP) 14
4.2 LED1 On/Off Cycle, Cycle of PWM Output to LED2 and Its Duty 26
CHAPTER 5 RELATED DOCUMENTS 29
APPENDIX A PROGRAM LIST 30

Document No. U19855EJ1V0AN00
Date Published July 2009 N

© NEC Electronics Corporation 2009

- **The information in this document is current as of June, 2009. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- 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 and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. In addition, NEC Electronics products are not taken measures to prevent radioactive rays in the product design. When customers use NEC Electronics products with their products, customers shall, on their own responsibility, incorporate sufficient safety measures such as redundancy, fire-containment and anti-failure features to their products in order to avoid risks of the damages to property (including public or social property) or injury (including death) to persons, as the result of defects of NEC Electronics products.
- 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 product 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).

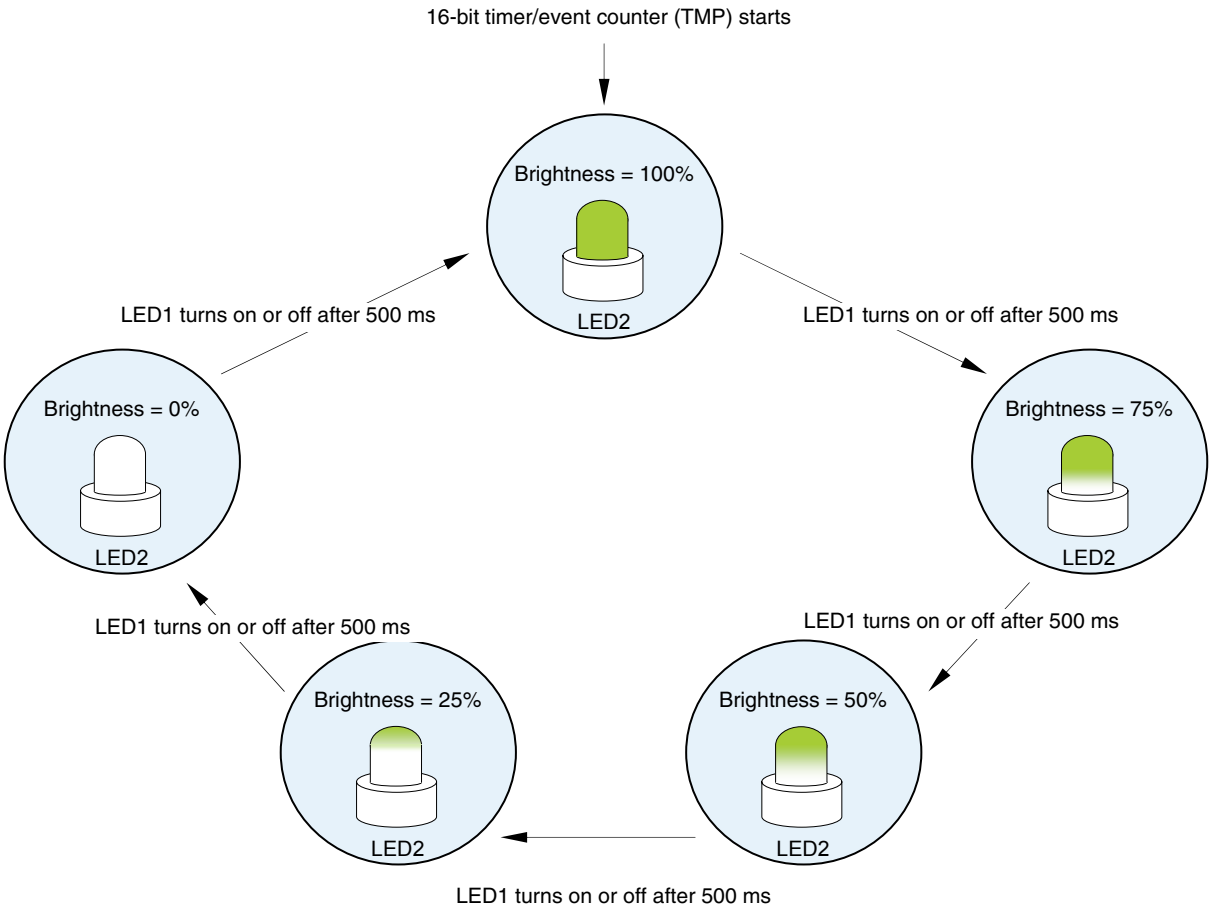
M8E0904E

CHAPTER 1 OVERVIEW

In this sample program, an example of using the PWM output function of the 16-bit timer/event counter (TMP) is presented. The brightness of LED2 is changed every 500 ms by controlling the PWM output duty. LED1 turns on or off each time the duty is changed.

Peripherals that stop immediately after a reset and are not used in the sample program are not set up.

Figure 1-1. Operation Overview



PWM output duty	0%	25%	50%	75%	100%
LED2 brightness	100%	75%	50%	25%	0%

In this sample program, “LED2 brightness (%) = 100 – PWM output duty” because the PWM output active level is high and LED2 turns on when it is at low level.

1.1 Initial Settings

The main initial settings are as follows:

<Referencing option byte>

- Referencing the oscillation stabilization time immediately after a reset

<Setting up on-chip peripherals>

- Setting the number of wait cycles to 1 for bus access to on-chip peripheral I/O registers
- Specifying normal operation mode by using the on-chip debug mode register (OCDM)
- Stopping the internal oscillator and watchdog timer 2
- Specifying that the CPU clock frequency not be divided
- Specifying PLL mode and 20 MHz operation (5 MHz × 4)

<Pin settings>

- Setting up unused pins
- Selecting the LED1 output pin (PCM3 pin)
- Selecting the LED2 output pin (TOP01 pin)

<Timer P0 (TMP0) settings>

- Specifying the undivided f_{XX} (20 MHz) as the count clock frequency by using the TP0CTL0 register
- Specifying PWM output as the timer operation mode by using the TP0CTL1 register
- Enabling timer output from the TOP01 pin and specifying the active level as high by using the TP0IOC0 register
- Setting the PWM cycle to 2 ms by using the TP0CCR0 register
- Setting the duty to 0% by using the TP0CTL1 register
- Enabling TMP0 by using the TP0CTL0 register
- Setting the priority of the INTTP0CC0 interrupt to level 7 and unmasking it

1.2 Servicing Interrupts by Using 16-bit Timer/Event Counter (TMP0) Per Specified PWM Cycle

After the initial settings are specified, LED1 turns on or off and the duty of PWM output to LED2 is changed every 500 ms based on the INTTP0CC0 interrupt, which is generated by the 16-bit timer/event counter (TMP0).

Because the PWM cycle is initially set to 2 ms, the TMP0 interrupt (INTTP0CC0) occurs every 2 ms. During interrupt servicing, the number of interrupts that has occurred is counted and, every 250 counts, LED1 turns on or off and the duty of PWM output to LED2 is changed.

Caution For cautions on using a device, see the user's manual for that device (V850ES/JF3-L or V850ES/JG3-L).

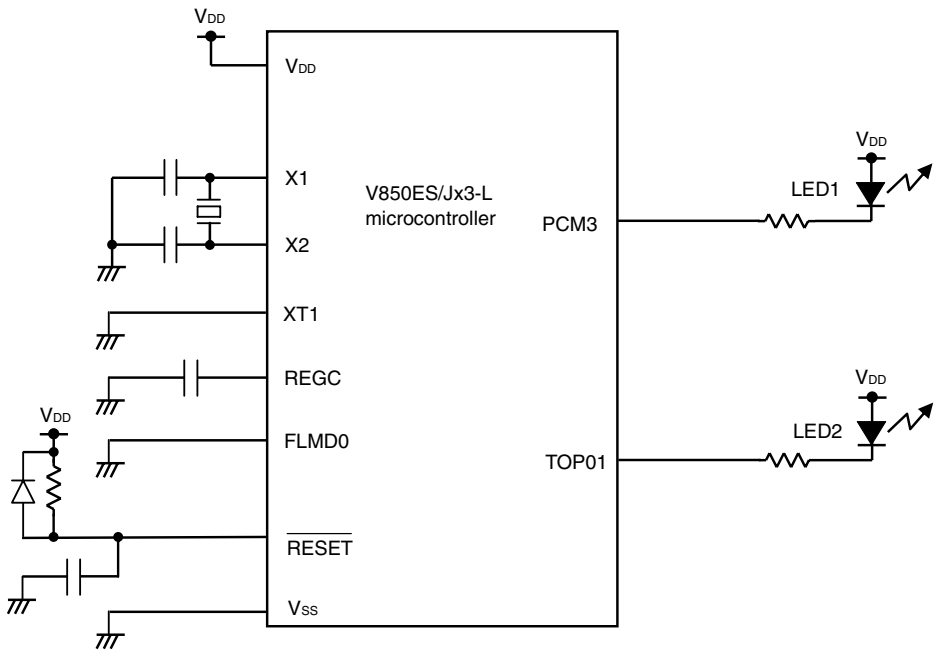
CHAPTER 2 CIRCUIT DIAGRAM

This chapter provides a circuit diagram and describes the peripheral hardware used in the sample program.

2.1 Circuit Diagram

The circuit diagram is shown below.

Figure 2-1. Circuit Diagram



Cautions 1. Connect the EV_{DD}, AV_{REF0}, and AV_{REF1} pins directly to V_{DD}.
2. Connect the EV_{SS} and AV_{SS} pins directly to GND.
3. Connect the FLMD0 pin to GND in normal operation mode.
4. Connect the REGC pin to GND via a capacitor (recommended capacitance: 4.7 μF).
5. Leave all unused ports open because they will be handled as output ports.

2.2 Peripheral Hardware

The peripheral hardware to be used is shown below.

- LED1: LED1 is switched between on and off each time the brightness of LED2 is changed.
- LED2: The brightness is changed in five levels.

CHAPTER 3 SOFTWARE

This chapter describes the files included in the compressed files that are downloaded, on-chip peripherals of the microcontroller, and the initial settings, and provides an operational overview of the sample program. Flowcharts are also shown.

3.1 Included Files

The following table shows the files included in the compressed files that are downloaded.

File Name (Tree)	Description	Compressed (*.zip) Files Included	
• conf — crtE.s	Startup routine file[Note 1]	–	●
— AppNote_PWM.dir	Link directive file[Note 2]	●	●
— AppNote_PWM.prj	Project file for integrated development environment PM+	–	●
— AppNote_PWM.prw	Workspace file for integrated development environment PM+	–	●
• src — main.c	C source file including code for hardware initialization processing and the main microcontroller processing	●	●
— minicube2.s	Source file for reserving the area for MINICUBE®2	●	●
— opt_b.s	Source file for specifying values for the option byte	●	●

Notes 1. This is the startup file copied when “Copy and Use the Sample file” is selected in the Startup File dialog box in the New WorkSpace wizard. (If the default installation path is used, the startup file will be a copy of C:\Program Files\NEC Electronics Tools\CA850*version*\lib850\r32\crtE.s.)

2. This is the link directive file automatically generated if “Create and Use the Sample file” is selected and “Use Internal memory only” is selected for “Memory Usage” in the LinkDirective File dialog box in the New WorkSpace wizard, and to which a segment for MINICUBE2 is added. (If the default installation path is used, C:\Program Files\NEC Electronics Tools\PM+*version*\bin\w_data\V850_i.dat is used as the reference file.)

Remark : Only the source file is included.

: The files used with the integrated development environment PM+ are included.

3.2 On-Chip Peripherals Used

The following on-chip peripherals of the microcontroller are used in this sample program:

- PWM output function: 16-bit timer/event counter (TMP0)
- PWM output port (LED2): TOP01[Note]
- Output port (LED1): PCM3

Note For the V850ES/JG3-L and V850ES/JF3-L microcontrollers, this pin also functions as TIP01 or P33.

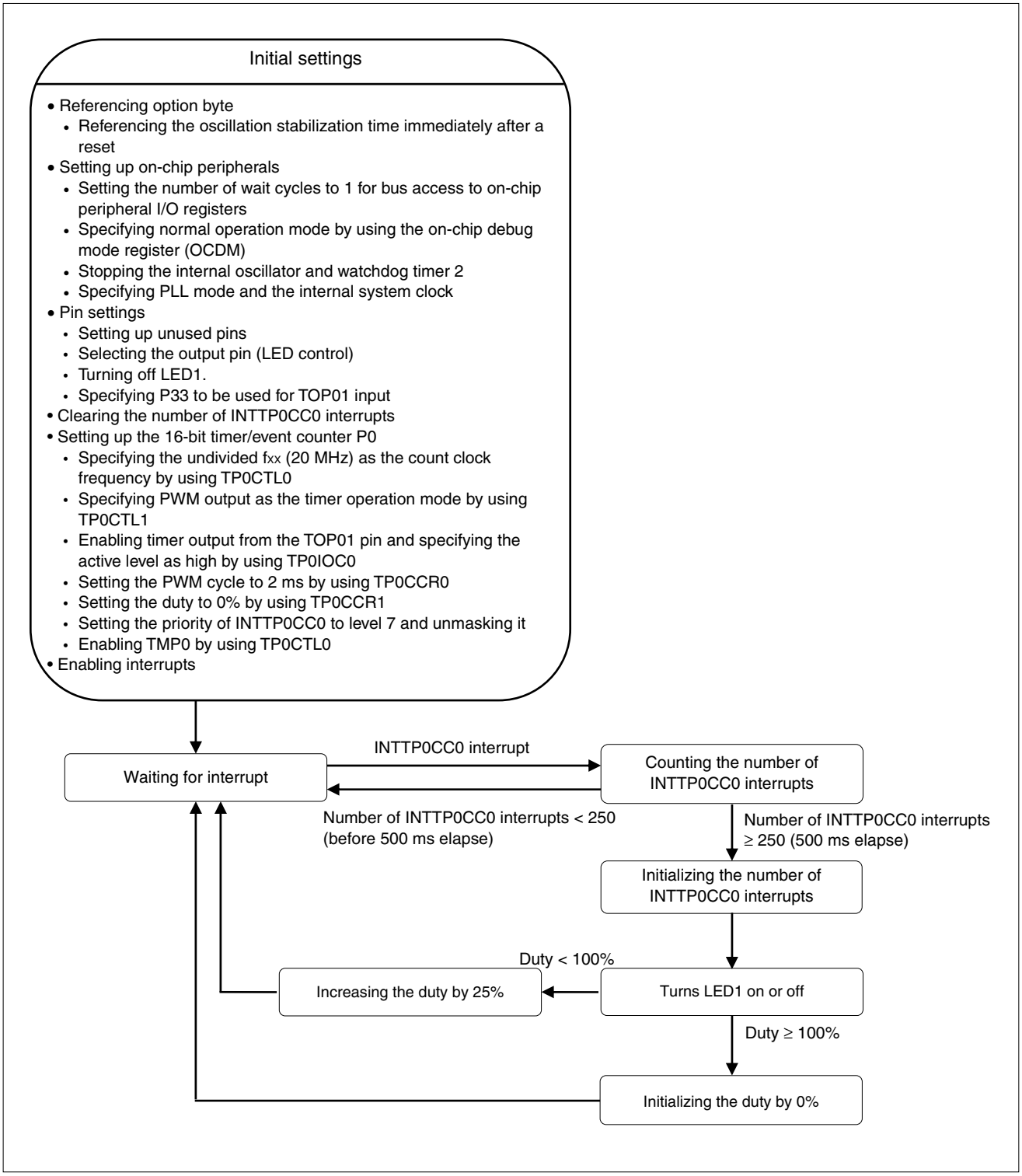
3.3 Initial Settings and Operational Overview

As the initial settings in the sample program, the clock frequency is selected, watchdog timer 2 is stopped, and settings for the I/O ports, PWM output from the 16-bit timer/event counter (TMP0), and interrupts are specified.

After the initial settings are specified, the brightness of LED2 is changed by controlling the duty of PWM output from the 16-bit timer/event counter (TMP0). The duty is changed every 500 ms by using the TMP0 interrupt (INTTP0CC0). LED1 turns on or off each time the duty is changed.

The details are described in the status transition diagram shown below.

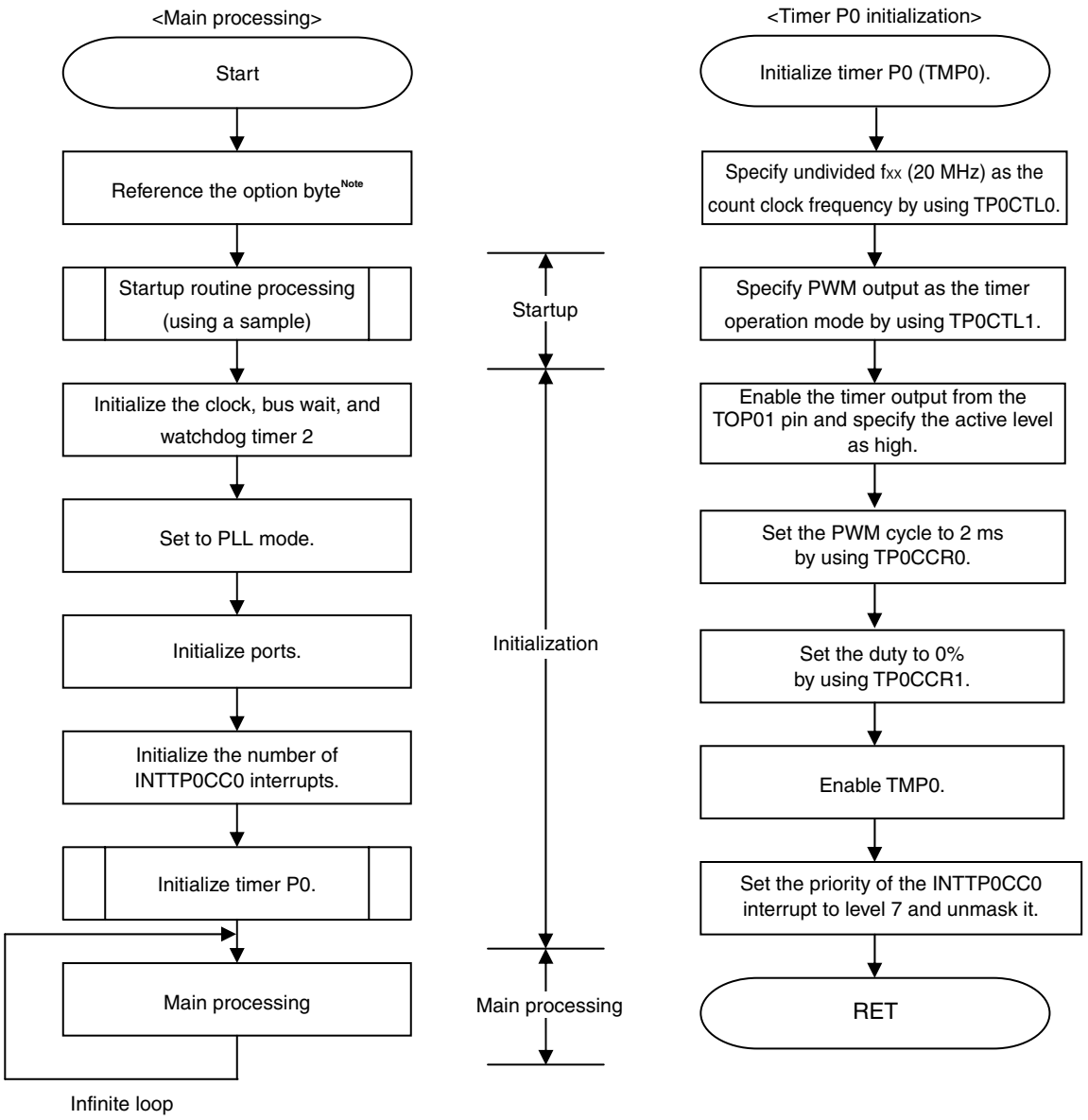
Figure 3-1. Status Transition



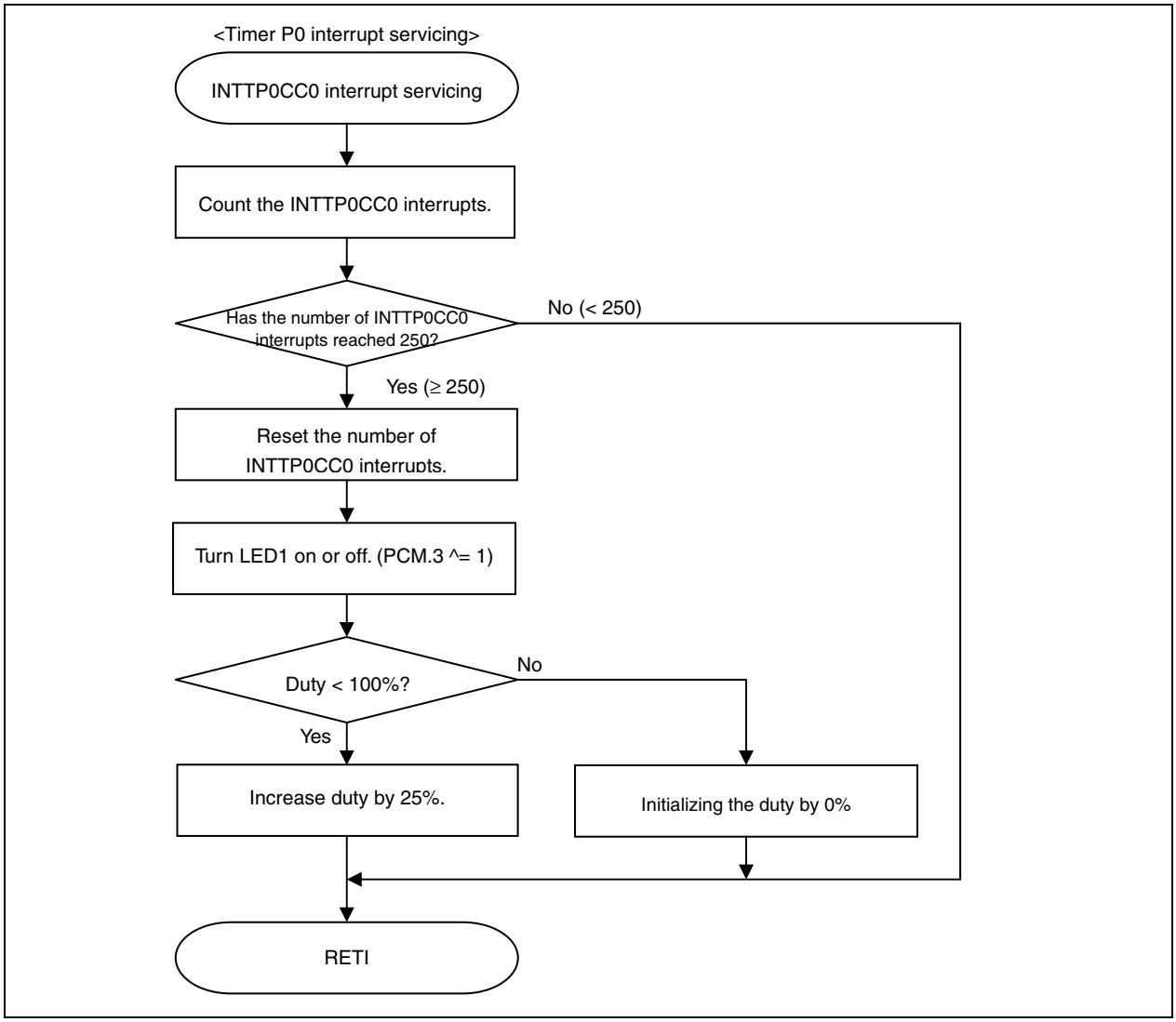
3.4 Flowcharts

Flowcharts for the sample program are shown below.

Figure 3-2. Status Transition



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the oscillation stabilization time immediately after a reset ends is set to 6.554 ms using the option byte.



[Column] Contents of the startup routine

The startup routine is executed before executing the main function immediately after resetting the V850 ends. Basically, the startup routine executes initialization so that the C program can start. Specifically, the following are performed:

- Allocating the argument space for the main function
- Allocating the stack
- Setting up the reset handler when a reset is input
- Setting up the text pointer (tp)
- Setting up the global pointer (gp)
- Setting up the stack pointer (sp)
- Setting up the element pointer (ep)
- Specifying mask values for the mask registers (r20 and r21)
- Clearing the sbss and bss areas to 0
- Specifying the CTBP value for the prologue/epilogue runtime library
- Specifying r6 and r7 as the arguments for the main function
- Branching to the main function

3.5 Differences Between V850ES/JG3-L and V850ES/JF3-L

The V850ES/JG3-L is the V850ES/JF3-L with its functions, such as I/Os, timer/counters, and serial interfaces, expanded.

In this sample program, the initialization range of P1, P3, P7, P9, and PDH during I/O initialization differs.

See **APPENDIX A PROGRAM LIST** for details about the sample program.

3.6 Difference Between TMP and TMQ

16-bit timer/event counter P (TMP) and 16-bit timer/event counter Q (TMQ) differ in the number of capture trigger pins, timer output pins, and capture/compare registers.

In the sample program, 16-bit timer/event counter P (TMP) is used. When using 16-bit timer/event counter Q (TMQ), see **CHAPTER 4 SETTING UP REGISTERS** and **APPENDIX A PROGRAM LIST** for the settings.

3.7 Security ID

The flash memory can be protected from unauthorized reading by using a 10-byte ID code for authentication when executing on-chip debugging using an on-chip debug emulator.

For details about the security ID, see the **V850ES/Jx3-L Sample Program (Interrupt) External Interrupt Generated by Switch Input Application Note**.

CHAPTER 4 SETTING UP REGISTERS

This chapter describes the settings of 16-bit timer/event counter P (TMP) and 16-bit timer/event counter Q (TMQ).

For details about other initial settings, see the **V850ES/Jx3-L Sample Program (Initial Settings) LED Lighting Switch Control Application Note**. For details about interrupts, see the **V850ES/Jx3-L Sample Program (Interrupt) External Interrupt Generated by Switch Input Application Note**.

Peripherals that are stopped immediately after a reset and are not used in this sample program are not set up.

For details about how to set up registers, see the user's manual for the device used.

- V850ES/JG3-L 32-bit Single-Chip Microcontroller Hardware User's Manual
- V850ES/JF3-L 32-bit Single-Chip Microcontroller Hardware User's Manual

For details about extended C code, see the **CA850 C Compiler Package C Language User's Manual**.

4.1 Setting Up 16-bit Timer/Event Counter P (TMP)

The following nine registers are used to set up 16-bit timer/event counter P (TMP):

- TMPn control register 0 (TPnCTL0)
- TMPn control register 1 (TPnCTL1)
- TMPn I/O control register 0 (TPnIOC0)
- TMPn I/O control register 1 (TPnIOC1)
- TMPn I/O control register 2 (TPnIOC2)
- TMPn option register 0 (TPnOPT0)
- TMPn capture/compare register 0 (TPnCCR0)
- TMPn capture/compare register 1 (TPnCCR1)
- TMPn counter read buffer register (TPnCNT)

Remarks 1. n = 0 to 5 (V850ES/JG3-L), n = 0 to 2, or 5 (V850ES/JF3-L)

2. n = 0 in the sample program

3. The eleven registers below are used to set up 16-bit timer/event counter Q (TMQ).
The description on the following pages is of TMP. Therefore, when using TMQ, read the above registers as those below, and read TP in the bit names as TQ.

- TMQ0 control register 0 (TQ0CTL0)
- TMQ0 control register 1 (TQ0CTL1)
- TMQ0 I/O control register 0 (TQ0IOC0)
- TMQ0 I/O control register 1 (TQ0IOC1)
- TMQ0 I/O control register 2 (TQ0IOC2)
- TMQ0 option register 0 (TQ0OPT0)
- TMQ0 capture/compare register 0 (TQ0CCR0)
- TMQ0 capture/compare register 1 (TQ0CCR1)
- TMQ0 capture/compare register 2 (TQ0CCR2)
- TMQ0 capture/compare register 3 (TQ0CCR3)
- TMQ0 counter read buffer register (TQ0CNT)

4.1.1 Setting up 16-bit timer/event counter P (TMP) operation clock

TMPn control register 0 (TPnCTL0) selects the count clock for 16-bit timer/event counter P (TMP) and controls the counter.

Values must be specified for the TPnCKS2 to TPnCKS0 bits when the TPnCE bit is 0.

In this sample program, f_{XX} (20 MHz) is selected by clearing the TPnCKS2 to TPnCKS0 bits at initialization. After specifying the settings for the 16-bit timer/event counter P (TMP) registers, set the TPnCE bit to 1.

Figure 4-1. TPnCTL0 Register Format

TMPn control register 0 (TPnCTL0)

Address: TP0CTL0 0xFFFFF590, TP1CTL0 0xFFFFF5A0,
TP2CTL0 0xFFFFF5B0, TP3CTL0 0xFFFFF5C0,
TP4CTL0 0xFFFFF5D0, TP5CTL0 0xFFFFF5E0

7	6	5	4	3	2	1	0
TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	TMPn operation control
0	TMPn disabled (TMPn reset asynchronously).
1	TMPn enabled. TMPn starts.

TPnCKS2	TPnCKS1	TPnCKS0	Internal count clock selection	
			n = 0, 2, 4	n = 1, 3, 5
0	0	0	f_{XX}	
0	0	1	$f_{XX}/2$	
0	1	0	$f_{XX}/4$	
0	1	1	$f_{XX}/8$	
1	0	0	$f_{XX}/16$	
1	0	1	$f_{XX}/32$	
1	1	0	$f_{XX}/64$	$f_{XX}/256$
1	1	1	$f_{XX}/128$	$f_{XX}/512$

Remark The red values are specified in this sample program.

4.1.2 Setting up 16-bit timer/event counter P (TMP) operation mode

TMPn control register 1 (TPnCTL1) specifies the operation mode of 16-bit timer/event counter P (TMP).

In this sample program, the PWM output mode is specified by specifying 100b for the TPnMD2 to TPnMD0 bits.

Figure 4-2. TPnCTL1 Register Format

TMPn control register 1 (TPnCTL1)

Address: TP0CTL1 0xFFFFF591, TP1CTL1 0xFFFFF5A1,
TP2CTL1 0xFFFFF5B1, TP3CTL1 0xFFFFF5C1,
TP4CTL1 0xFFFFF5D1, TP5CTL1 0xFFFFF5E1

7	6	5	4	3	2	1	0
0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnEST	Software trigger control
0	–
1	Generate a valid signal for external trigger input.

TPnEEE	Count clock selection
0	Disable operation with external event count input[Note].
1	Enable operation with external event count input.

TPnMD2	TPnMD1	TPnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Note Clock selected using the TPnCK0 to TPnCK2 bits of the TPnCTL0 register

Remark The red values are specified in this sample program.

4.1.3 Controlling timer output

TMPn I/O control register 0 (TPnIOC0) controls timer output.

In this sample program, PWM output from the TOPn1 pin is enabled by setting the TPnOE1 bit to 1 and the active level of the TOPn1 pin is set to high by clearing the TPnOL1 bit to 0.

Figure 4-3. TPnIOC0 Register Format

TMPn I/O control register 0 (TPnIOC0)

Address: TP0IOC0 0xFFFFF592, TP1IOC0 0xFFFFF5A2,
TP2IOC0 0xFFFFF5B2, TP3IOC0 0xFFFFF5C2,
TP4IOC0 0xFFFFF5D2, TP5IOC0 0xFFFFF5E2

7	6	5	4	3	2	1	0
0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

TPnOL1	TOPn1 pin output level setting
0	TOPn1 pin starts output at high level
1	TOPn1 pin starts output at low level

TPnOE1	TOPn1 pin output setting
0	Timer output disabled
1	Timer output enabled

TPnOL0	TOPn0 pin output level setting
0	TOPn0 pin starts output at high level
1	TOPn0 pin starts output at low level

TPnOE0	TOPn0 pin output setting
0	Timer output disabled
1	Timer output enabled

Remarks 1. The red values are specified in this sample program.

2. The 16-bit timer/event counter Q (TMQ) differs in that, in addition to TQ being substituted for TP, the TQ0OL3, TQ0OE3, TQ0OL2, and TQ0OE2 bits are assigned to bits 7 to 4 of the TQ0IOC0 register.

4.1.4 Controlling valid edge of capture trigger input signal

TMPn I/O control register 1 (TPnIOC1) controls the valid edge of the capture trigger input signal (from the TIPn0 and TIPn1 pins).

In the PWM output mode, the TPnIOC1 register does not have to be controlled. Therefore, the register is not controlled in this sample program.

Figure 4-4. TPnIOC1 Register Format

TMPn I/O control register 1 (TPnIOC1)

Address: TP0IOC1 0xFFFFF593, TP1IOC1 0xFFFFF5A3,
TP2IOC1 0xFFFFF5B3, TP3IOC1 0xFFFFF5C3,
TP4IOC1 0xFFFFF5D3, TP5IOC1 0xFFFFF5E3

7	6	5	4	3	2	1	0
0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

TPnIS3	TPnIS2	Capture trigger input signal (TIPn1 pin) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TPnIS1	TPnIS0	Capture trigger input signal (TIPn0 pin) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Remarks 1. The TPnIOC1 register is not used in this sample program.

2. The 16-bit timer/event counter Q (TMQ) differs in that, in addition to TQ being substituted for TP, the TQ0IS7, TQ0IS6, TQ0IS5, and TQ0IS4 bits are assigned to bits 7 to 4 of the TQ0IOC1 register.

4.1.5 Controlling external input signals

TMPn I/O control register 2 (TPnIOC2) controls the valid edge of the external event count input signal (from the TIPn0 pin) and external trigger input signal (from the TIPn0 pin).

If the internal count clock is selected (TPnCTL1.TPnEEE bit = 0) and the timer runs in the PWM output mode, the TPnIOC2 register does not have to be controlled. Therefore, the register is not controlled in this sample program.

Figure 4-5. TPnIOC2 Register Format

TMPn I/O control register 2 (TPnIOC2)

Address: TP0IOC2 0xFFFFF594, TP1IOC2 0xFFFFF5A4,
TP2IOC2 0xFFFFF5B4, TP3IOC2 0xFFFFF5C4,
TP4IOC2 0xFFFFF5D4, TP5IOC2 0xFFFFF5E4

7	6	5	4	3	2	1	0
0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

TPnEES1	TPnESS0	External event count input signal (TIPn0 pin) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TPnETS1	TPnETS0	External trigger input signal (TIPn0 pin) valid edge setting
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Remark The TPnIOC2 register is not used in this sample program.

4.1.6 Controlling capture/compare operation

TMPn option register 0 (TPnOPT0) controls the capture/compare operation setting and overflow detection.

In the PWM output mode, the TPnOPT0 register does not have to be controlled. Therefore, the register is not controlled in this sample program.

Figure 4-6. TPnOPT0 Register Format

TMPn option register 0 (TPnOPT0)

Address: TP0OPT0 0xFFFFF595, TP1OPT0 0xFFFFF5A5,
TP2OPT0 0xFFFFF5B5, TP3OPT0 0xFFFFF5C5,
TP4OPT0 0xFFFFF5D5, TP5OPT0 0xFFFFF5E5

7	6	5	4	3	2	1	0
0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected

TPnCCS0	TPnCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected

TPnOVF	TMPn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	TPnOVF bit 0 written or TPnCTL0.TPnCE bit = 0

Remarks 1. The TPnOPT0 register is not used in the sample program.

2. The 16-bit timer/event counter Q (TMQ) differs in that, in addition to TQ being substituted for TP, the TQ0CCS3, TQ0CCS2, TQ0CCS1, and TQ0CCS0 bits are assigned to bits 7 to 4 of the TQ0OPT0 register.

4.1.7 Specifying PWM waveform cycle

Specify the PWM waveform output cycle by using TMPn capture/compare register 0 (TPnCCR0), and specify the active level width by using TMPn capture/compare register 1 (TPnCCR1).

The active level width, cycle, and duty of the PWM waveform are calculated as follows:

- Active level width = (TPnCCR1 register value) / count clock frequency
- Cycle = (TPnCCR0 register value + 1) / count clock frequency
- Duty = (TPnCCR1 register value) / (TPnCCR0 register value + 1)

Remark In this sample program, the undivided fxx is selected as the count clock frequency.

To output the PWM waveform with the 0% duty, set the TPnCCR1 register to 0x0000. To output the PWM waveform with the 100% duty, set the TPnCCR1 register to (TPnCCR0 register value + 1).

The PWM waveform can be changed if the value of the TPnCCRa register (a = 0 or 1) is changed while the timer is running. The new value is applied when the value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0x0000.

A compare match interrupt request signal (INTTPnCC0) is generated one cycle after the value of the 16-bit counter matches the value of the CCR0 buffer register. At this time, the 16-bit counter is cleared to 0x0000.

A compare match interrupt request signal (INTTPnCC1) is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register.

In this sample program, the active level width and cycle are specified during initialization, and only the active level width is changed while the timer is running.

Cautions 1. To change both the PWM waveform cycle and active level width, first specify a new cycle by using the TPnCCR0 register, and then specify a new active level width by using the TPnCCR1 register.

2. To change only the PWM waveform cycle, first specify a new cycle by using the TPnCCR0 register, and then specify the same value for the TPnCCR1 register.

3. To change only the active level width (duty) of the PWM waveform, specify a new value by using the TPnCCR1 register.

4. To write to the TPnCCR1 register and then to the TPnCCR0 or TPnCCR1 register, make sure that the INTTPnCC0 interrupt has been generated. Otherwise, transferring data from the TPnCCRa register to the CCRa buffer register conflicts with writing to the TPnCCRa register, which might result in the value of the CCRa buffer register being undefined. (a = 0 or 1)

5. If the TPnCCR0 register is set to 0xFFFF, the PWM waveform cannot be output with the 100% duty.

Figure 4-7. TPnCCR0 Register Format

TMPn capture/compare register 0 (TPnCCR0)

Address: TP0CCR0 0xFFFFF596, TP1CCR0 0xFFFFF5A6,
TP2CCR0 0xFFFFF5B6, TP3CCR0 0xFFFFF5C6,
TP4CCR0 0xFFFFF5D6, TP5CCR0 0xFFFFF5E6

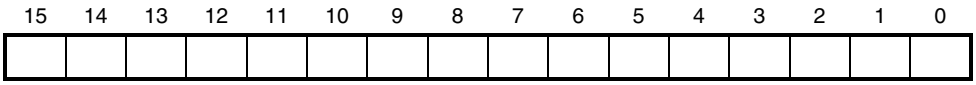
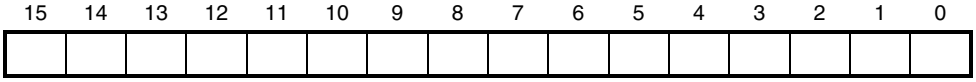


Figure 4-8. TPnCCR1 Register Format

TMPn capture/compare register 1 (TPnCCR1)

Address: TP0CCR1 0xFFFFF598, TP1CCR1 0xFFFFF5A8,
TP2CCR1 0xFFFFF5B8, TP3CCR1 0xFFFFF5C8,
TP4CCR1 0xFFFFF5D8, TP5CCR1 0xFFFFF5E8



Four capture/compare registers (TQ0CCR0 to TQ0CCR3) are provided for 16-bit timer/event counter Q (TMQ). These registers are used in the same way as the TPnCCR0 and TPnCCR1 registers.

4.1.8 Referencing timer count value

The TPnCNT register is a read buffer register from which a 16-bit counter value can be read.

The current counter value can be read by reading this register while the timer is running (TPnCTL0.TPnCE bit = 1).

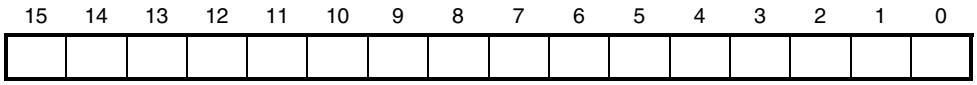
If this register is read while the timer is stopped (TPnCTL0.TPnCE bit = 0), 0x0000 is returned.

In the sample program, this register is not used because counter values do not have to be referenced.

Figure 4-9. TPnCNT Register Format

TMPn counter read buffer register (TPnCNT)

Address: TP0CNT 0xFFFFF59A, TP1CNT 0xFFFFF5AA,
TP2CNT 0xFFFFF5BA, TP3CNT 0xFFFFF5CA,
TP4CNT 0xFFFFF5DA, TP5CNT 0xFFFFF5EA



4.1.9 Example of timer settings

[Example 1] When starting the timer with 16-bit timer/event counter P (TMP) set to the external event count mode and the PWM count clock, PWM waveform cycle, and duty specified (same as the sample program)

Setup procedure:

<1> Specify the undivided fxx (20 MHz) as the count clock frequency.
<2> Specify the PWM output mode.
<3> Enable output from the TOP01 pin and specify the active level as high.
<4> Set the PWM waveform cycle to 2 ms by using capture/compare register 0.
<5> Set the initial duty of the PWM waveform to 0% by using capture/compare register 1.
<6> Start the timer.
<7> Unmask the relevant interrupt.

Program example (same as the sample program)

```
/* Timer P0 settings */
TP0CTL0 = 0x00;              /* Specifies the undivided fxx as the count clock frequency.*/ } <1>
                             /* Failsafe: Clears TP0CE to 0 to stop TMP0.                */
TP0CTL1 = 0x04;              /* Specifies the PWM output mode.                           */ } <2>
TP0IOC0 = 0x04;              /* Specifies the TOP01 pin active level as high.            */ }
                             /* Enables TOP01 pin output.                                */ } <3>
TP0CCR0 = PWM_CYCLE;         /* Sets the PWM cycle to 2 ms.                              */ } <4>
TP0CCR1 = DUTY_PER_0;        /* Initializes the duty to 0%.                              */ } <5>
TP0CE   = 1;                 /* Starts TMP0.                                             */ } <6>
/* Interrupt control register setup */
TP0CCIC0 = 0x07;             /* Sets the priority of INTTP0CC0 to level 7 and unmasks it.*/ } <7>
```

Specifying the PWM_CYCLE value (40,000 – 1) sets the PWM waveform cycle to 2 ms.

Specifying the DUTY_PER_0 value (0) sets the initial PWM waveform duty to 0%.

[Example 2] When starting the timer with 16-bit timer/event counter Q (TMQ) set to the external event count mode and the PWM count clock, PWM waveform cycle, and duty specified

Setup procedure:

<1> Specify the undivided fxx (20 MHz) as the count clock frequency.
<2> Specify the PWM output mode.
<3> Enable output from the TOQ01 pin and specify the active level as high.
<4> Set the PWM waveform cycle to 2 ms by using capture/compare register 0.
<5> Set the initial duty of the PWM waveform to 0% by using capture/compare register 1.
<6> Start the timer.
<6> Unmask the relevant interrupt.

Program example (same as the sample program)

```
/* Timer Q0 settings */
TQ0CTL0 = 0x00;           /* Specifies the undivided fxx as the count clock frequency. */ } <1>
                          /* Failsafe: Clears TQ0CE to 0 to stop TMQ0.                  */
TQ0CTL1 = 0x04;           /* Specifies the PWM output mode.                             */ } <2>
TQ0IOC0 = 0x04;           /* Specifies the TOQ01 pin active level as high.              */ }
                          /* Enables TOQ01 pin output.                                  */ } <3>
TQ0CCR0 = PWM_CYCLE;      /* Sets the PWM cycle to 2 ms.                                */ } <4>
TQ0CCR1 = DUTY_PER_0;     /* Initializes the TOQ01 duty to 0%.                          */ } <5>
TQ0CE   = 1;              /* Starts TMQ0.                                               */ } <6>
/* Interrupt control register setup */
TQ0CCIC0 = 0x07;          /* Sets the priority of INTTP0CC0 to level 7 and unmasks it.  */ } <7>
```

Specifying the PWM_CYCLE value (40,000 – 1) sets the PWM waveform cycle to 2 ms.

Specifying the DUTY_PER_0 value (0) sets the initial PWM waveform duty to 0%.

4.2 LED1 On/Off Cycle, Cycle of PWM Output to LED2 and Its Duty

In this sample program, the INTTP0CC0 interrupt generated by the 16-bit timer/event counter (TMP0) is used to count the number of INTTP0CC0 interrupts used to generate the timing to change the duty of PWM output to LED2.

LED1 turns on or off when the duty is changed.

- INTTP0CC0 generation timing = (TP0CCR0 register value + 1) / count clock frequency
- LED1 on/off timing = INTTP0CC0 generation × 250 times
- Cycle of output (signal) to LED1 = LED1 on/off × twice
- Cycle of PWM output to LED2 = TP0CCR0 register value + 1
- Duty of PWM output to LED2 = (TP0CCR1 register value) / (TP0CCR0 register value + 1)

Remark In this sample program, the undivided f_{XX} is selected as the count clock frequency.

Example When the count clock frequency for the 16-bit timer/event counter (TMP0) is the undivided f_{XX} (20 MHz), the TP0CCR0 register is set to 39,999, the duty of PWM output to LED2 is changed first to 25% and then to 50% 500 ms later, and LED1 turns on or off each time there is a change

- INTTP0CC0 generation timing = (39,999 + 1) / 20 MHz = 2 ms
- LED1 on/off timing = 2 ms × 250 = 500 ms
- Cycle of output (signal) to LED1 = 500 ms × 2 = 1 s
- Cycle of PWM output to LED2 = (39,999 + 1) = 40000
- Duty of PWM output to LED2 = (20,000) / (39,999 + 1) = 50%

Figure 4-10. PWM Output Timing Example (When PWM Cycle Is 2 ms and Duty Is 50%)

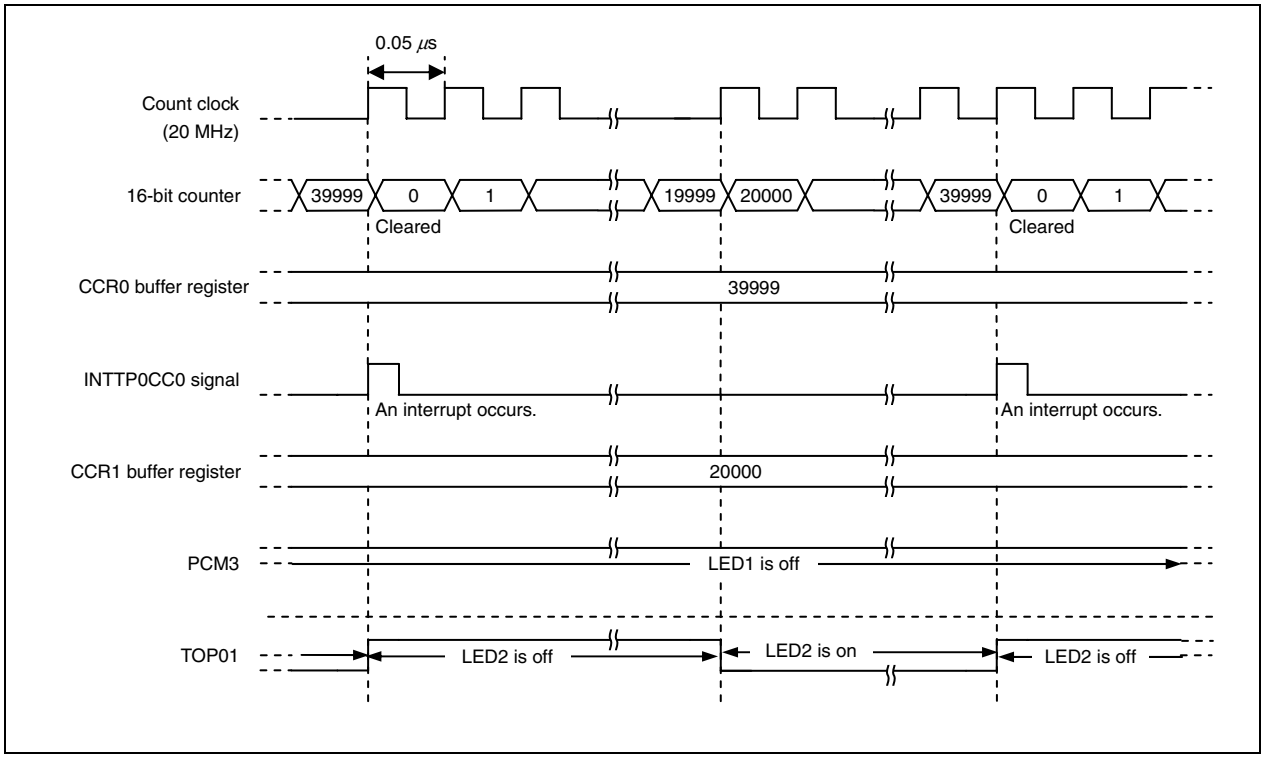
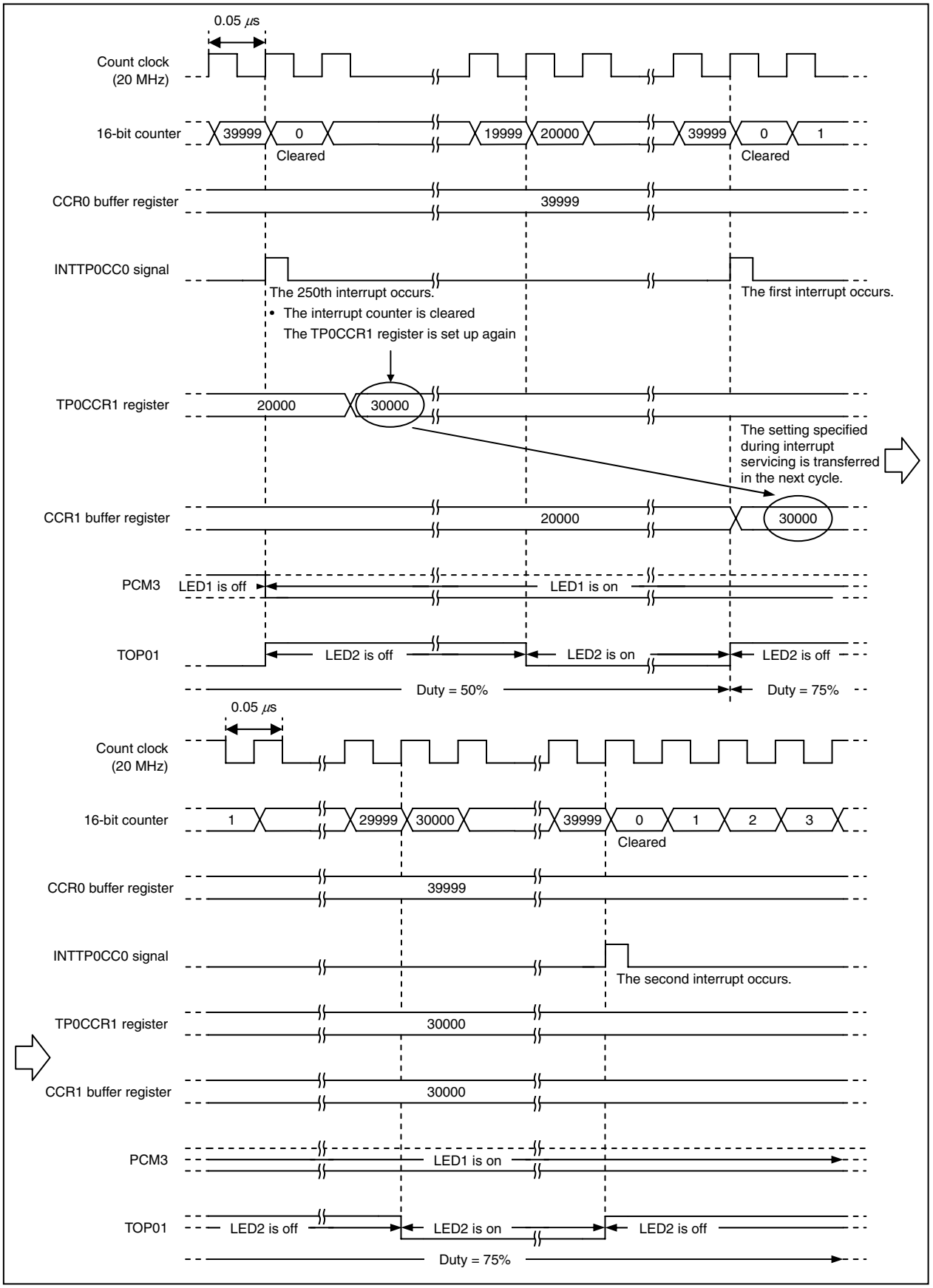


Figure 4-11. PWM Output Timing Example
(When PWM Cycle Is 2 ms and Duty Is Changed from 50% to 75%)



CHAPTER 5 RELATED DOCUMENTS

Document Name	Document Number
V850ES/JF3-L Hardware User's Manual	U18952E
V850ES/JG3-L Hardware User's Manual	U18953E
PM+ Ver. 6.30 User's Manual	U18416E
CA850 Ver. 3.20 C Compiler Package Operation User's Manual	U18512E
CA850 Ver. 3.20 C Compiler Package C Language User's Manual	U18513E
CA850 Ver. 3.20 C Compiler Package Assembly Language User's Manual	U18514E
CA850 Ver. 3.20 C Compiler Package Link Directives User's Manual	U18515E
V850ES Architecture User's Manual	U15943E
QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual	U18371E
ID850QB Ver. 3.40 Integrated Debugger Operation User's Manual	U18604E

URL of document search website: http://www.necel.com/micro/en/documentation.html

APPENDIX A PROGRAM LIST

The V850ES/Jx3-L microcontroller source code is shown below.

● opt_b.s

```
#-------------------------------------------------------------------------------
#
#    NEC Electronics     V850ES/Jx3-L microcontroller
#
#-------------------------------------------------------------------------------
#    V850ES/JG3-L JF3-L  sample program
#-------------------------------------------------------------------------------
#    PWM output mode
#-------------------------------------------------------------------------------
#[History]
#    2009.6.--   Released
#-------------------------------------------------------------------------------
#[Overview]
#    This sample program specifies a value for the option byte.
#-------------------------------------------------------------------------------

    .section "OPTION_BYTES"
    .byte 0b00000101 -- 0x7a (5MHz: Sets the oscillation stabilization time to 6.554 ms)
    .byte 0b00000000 -- 0x7b        ↑
    .byte 0b00000000 -- 0x7c        ↑
    .byte 0b00000000 -- 0x7d 0x00 must be specified for addresses 0x7b to 0x7f.
    .byte 0b00000000 -- 0x7e        ↓
    .byte 0b00000000 -- 0x7f        ↓
```

● minicube2.s

```
#-------------------------------------------------------------------------------
#
#    NEC Electronics     V850ES/Jx3-L microcontroller
#
#-------------------------------------------------------------------------------
#    V850ES/JG3-L JF3-L  sample program
#-------------------------------------------------------------------------------
#    PWM output mode
#-------------------------------------------------------------------------------
#[History]
#    2009.6.--   Released
#-------------------------------------------------------------------------------
#[Overview]
#    This sample program allocates the resources required when using MINICUBE2.
#       (Example of using MINICUBE2 via CSIB0)
#-------------------------------------------------------------------------------

    -- Allocation of a 2 KB space as the monitor ROM section
    .section "MonitorROM", const
    .space 0x800, 0xff

    -- Allocation of an interrupt vector for debugging
    .section "DBG0"
    .space 4, 0xff

    -- Allocation of a reception interrupt vector for serial communication
    .section "INTCB0R"
    .space 4, 0xff

    -- Allocation of a 16-byte space as the monitor RAM section
    .section "MonitorRAM", bss
    .lcomm monitorramsym, 16, 4
```

● AppNote_PWM.dir

```
#   Sample link directive file (not use RTOS/use internal memory only)
#
#   Copyright (C) NEC Electronics Corporation 2002
#   All rights reserved by NEC Electronics Corporation.
#
#   This is a sample file.
#   NEC Electronics assumes no responsibility for any losses incurred by customers or
#   third parties arising from the use of this file.
#
#   Generated        : PM+ V6.31  [ 9 Jul 2007]
#   Sample Version   : E1.00b [12 Jun 2002]
#   Device           : uPD70F3738 (C:\Program Files\NEC Electronics Tools\DEV\DF3738.800)
#   Internal RAM     : 0x3ffb000 - 0x3ffefff
#
#   NOTICE:
#        Allocation of SCONST, CONST and TEXT depends on the user program.
#
#        If interrupt handler(s) are specified in the user program then
#        the interrupt handler(s) are allocated from address 0 and
#        SCONST, CONST and TEXT are allocated after the interrupt handler(s).

SCONST  : !LOAD ?R {
       .sconst           = $PROGBITS     ?A .sconst;
};

CONST   : !LOAD ?R {
       .const            = $PROGBITS     ?A .const;
};

TEXT    : !LOAD ?RX {
       .pro_epi_runtime = $PROGBITS     ?AX .pro_epi_runtime;
       .text             = $PROGBITS     ?AX .text;
};

### For MINICUBE2 ###
MROMSEG : !LOAD ?R V0x03F800{
      MonitorROM = $PROGBITS ?A MonitorROM;
};
```

0x01F800 if the internal ROM size is 128 KB

Difference from the default link directive file (additional code)

An area reserved for MINICUBE2 is allocated.

```
SIDATA  : !LOAD ?RW V0x3ffb000 {
        .tidata.byte    = $PROGBITS     ?AW .tidata.byte;
        .tibss.byte     = $NOBITS       ?AW .tibss.byte;
        .tidata.word    = $PROGBITS     ?AW .tidata.word;
        .tibss.word     = $NOBITS       ?AW .tibss.word;
        .tidata         = $PROGBITS     ?AW .tidata;
        .tibss          = $NOBITS       ?AW .tibss;
        .sidata         = $PROGBITS     ?AW .sidata;
        .sibss          = $NOBITS       ?AW .sibss;
};

DATA    : !LOAD ?RW V0x3ffb100 {
        .data           = $PROGBITS     ?AW  .data;
        .sdata          = $PROGBITS     ?AWG .sdata;
        .sbss           = $NOBITS       ?AWG .sbss;
        .bss            = $NOBITS       ?AW  .bss;
};

### For MINICUBE2 ###
MRAMSEG : !LOAD ?RW V0x03FFEFF0{
MonitorRAM = $NOBITS ?AW MonitorRAM;
};

__tp_TEXT @ %TP_SYMBOL;
__gp_DATA @ %GP_SYMBOL &__tp_TEXT{DATA};
__ep_DATA @ %EP_SYMBOL;
```

Difference from the default link directive file (additional code)

An area reserved for MINICUBE2 is allocated.

● main.c

```
/*-------------------------------------------------------------------------*/
/*
/*    NEC Electronics     V850ES/Jx3-L microcontroller
/*
/*-------------------------------------------------------------------------*/
/*    V850ES/JG3-L sample program
/*-------------------------------------------------------------------------*/
/*    PWM output mode
/*-------------------------------------------------------------------------*/
/*[History]
/*    2009.6.--   Released
/*-------------------------------------------------------------------------*/
/*[Overview]
/*    This sample program shows an example of using the PWM output mode
/*    of the 16-bit timer/event counter (TMP0).
/*
/*    Timer P0 (TMP0) is set to PWM output mode to control the LED2 brightness.
/*    The duty is changed every 500 ms to change the LED2 brightness.
/*    LED1 turns on or off when the duty is changed.
/*
/*    Peripherals that stop immediately after a reset and are not used in the sample
/*    program are not specified.
/*
/*
/*    <Main settings>
/*    • Using pragma directives to enable setting up the interrupt handler and specifying
        peripheral I/O register names
/*    • Defining the adjustment value to set the timer P0 PWM output interval to 2 ms
/*    • Declaring prototypes
/*    • Setting up a bus wait for on-chip peripheral I/O registers, stopping watchdog timer 2,
        and setting up the clock
/*    • Initializing unused ports
/*    • Initializing the TOP01 port and PCM3 port
/*    • Initializing the timer P0 (TMP0)
/*
/*    <Timer P0 settings>
/*    Setting the count clock to fxx (20 MHz) by using the TP0CTL0 register
/*    Setting the operation mode to the PWM output mode by using the TP0CTL1 register
/*    Enabling TOP01 pin output and specifying the active level as high by using
/*    the TP0IOC0 register
/*    Setting the PWM output cycle by using the TP0CCR0 register
/*    Setting the PWM output active period width by using the TP1CCR0 register
/*    Enabling TMP0
/*    Setting the priority of INTTP0CC0 interrupt to level 7 and unmasking it
/*
```

```
/*  <Timer P0 interrupt (INTTP0CC0) servicing>

/*  LED1 turns on or off when the duty is changed every 500 ms in the following order:
/*  +-----------------+-------------------------------+
/*  | PWM output duty |   0% | 25% | 50% | 75% | 100% | (Hereafter, repeated from 0%)
/*  +-----------------+-------------------------------+
/*  | LED2 brightness | 100% | 75% | 50% | 25% |   0% |
/*  +-----------------+-------------------------------+
/*  # PWM output is high active and LED2 is low active; therefore,
/*    LED2 brightness = 100 - duty factor.
/*
/*[I/O port settings]

/* Input: -
/* Output: P33 (TOP01), PCM3
/* Unused ports: P02-P06, P10, P11, P30-P32, P34-P39, P50-P55, P70-P711, P90-P915,
                 PCM0-PCM2, PCT0, PCT1, PCT4, PCT6, PDH0-PDH5, PDL0-PDL15
                 *Preset all unused ports as output ports (low-level output).
/*
/*-------------------------------------------------------------------------*/

/*---------------------------*/
/*     pragma directives     */
/*---------------------------*/
#pragma ioreg                                  /* Enables the specification of peripheral I/O
                                                  registers.*/
#pragma interrupt INTTP0CC0 f_int_inttp0cc0/* Specifies the timer P0 interrupt (INTTP0CC0) handler.
                                              */
/*---------------------------*/
/*    Constant definitions   */
/*---------------------------*/
#define PWM_CYCLE        ( 40000 - 1 )     /* PWM cycle                    */
#define DUTY_PER_100     ( 40000 )         /* Duty = 100%                  */
#define DUTY_PER_25      ( 10000 )         /* Duty = 25%                   */
#define DUTY_PER_0       ( 0 )             /* Duty = 0%                    */
#define PWM_INT_CNT      ( 250 )           /* For counting 500 ms by timer */


/*---------------------------------*/
/*   Define the global variables   */
/*---------------------------------*/
static unsigned short usTMP0cnt;           /* 500 ms counter          */
```

When using TMQ, specify INTTQ0CC0.

```
/*---------------------------*/
/*   Prototype definitions   */
/*---------------------------*/
static  void f_init( void );               /* Initialization function                          */
static  void f_init_clk_bus_wdt2( void );  /* Clock bus/WDT2 initialization function           */
static  void f_init_port_func( void );     /* Port/alternate-function initialization function  */
static  void f_init_int_tmp0( void );      /* TMP0 initialization function                     */



/*****************************/
/*      Main module          */
/*****************************/
void main( void )
{
    f_init();                      /* Executes initialization.                              */

     __EI();                       /* Enables interrupts.                                   */

    while( 1 )                     /* Main loop (infinite loop)                             */
    {
    }

}
```

```
/*----------------------------------*/
/*      Initialization module       */
/*----------------------------------*/
static void f_init( void )
{
    f_init_clk_bus_wdt2();          /* Specifies a bus wait for on-chip peripheral I/O registers,
                                       stops WDT2, and sets up the clock. */

    f_init_port_func();             /* Sets up ports and alternate functions.              */

    f_init_int_tmp();               /* Sets up the TMP0 timer.                             */

    return;
}

/*-------------------------------------*/
/* Initializing clock, bus wait, WDT2 */
/*-------------------------------------*/
static void f_init_clk_bus_wdt2( void )
{
    VSWC   = 0x01;                   /* Sets a bus wait for on-chip peripheral I/O registers. */

                                     /* Specifies normal operation mode for OCDM. */
#pragma asm
    st.b    r0, PRCMD
    st.b    r0, OCDM
#pragma endasm

    RSTOP   = 1;                         /* Stops the internal oscillator. */
    WDTM2   = 0x00;                      /* Stops watchdog timer 2. */

                                         /* Specifies that the clock not be divided */
#pragma asm
    push    r10
    mov     0x80, r10
    st.b    r10, PRCMD
    st.b    r10, PCC
    pop     r10
#pragma endasm

    PLLCTL = 0x03;                       /* Sets PLL mode. */

    return;
}
```

Caution is required because access to special registers must be specified in assembly language.

Caution is required because access to special registers must be specified in assembly language.

```
/*-----------------------------------------*/
/* Setting up ports and alternate functions */
/*-----------------------------------------*/
static void f_init_port_func( void )
{
    P0    = 0x00;                          /* Sets P02 to P06 to output a low level signal. */
    PM0   = 0x83;
    PMC0  = 0x00;

    P1    = 0x00;                          /* Sets P10 and P11 to output a low level signal. */
    PM1   = 0xFC;

/* Set P33 to be used as TOP01 output */
    P3    = 0x0000;                        /* Sets P30 to P39 to output a low level signal.  */
    PM3   = 0xFC00;                        /* Sets P30 to P39 to output mode.                */
    PFC3  = 0x0008;                        /* Sets P33 to be used as TOP01 output.           */
    PMC3  = 0x0008;

#if(0) /* Because P4 is used via CSIB0 when using MINICUBE2,        */
       /* P4 is not initialized as an unused pin (QB-V850ESJG3L-TB) */
    P4    = 0x00;                          /* Sets P40 to P42 to output a low level signal.  */
    PM4   = 0xF8;
    PMC4  = 0x00;
#endif

    P5    = 0x00;                          /* Sets P50 to P55 to output a low level signal.  */
    PM5   = 0xC0;
    PMC5  = 0x00;

    P7H   = 0x00;                          /* Sets P70 to P711 to output a low level signal. */
    P7L   = 0x00;
    PM7H  = 0xF0;
    PM7L  = 0x00;

    P9    = 0x0000;                        /* Sets P90 to P915 to output a low level signal. */
    PM9   = 0x0000;
    PMC9  = 0x0000;

    PCM   = 0x08;                          /* Sets PCM0 to PCM2 to output a low level signal */
    PMCM  = 0xF0;                          /* and specifies the turn-off pattern for PCM3.   */
    PMCCM = 0x00;

    PCT   = 0x00;                          /* Sets PCT0, PCT1, PCT4, and PCT6 to output a   */
    PMCT  = 0xAC;                          /* low level signal.                              */
    PMCCT = 0x00;
```

For the V850ES/JF3-L, the setting is 0xFE.

For the V850ES/JF3-L, only P10 is set.

For the V850ES/JF3-L, P30 to P35, P38, and P39 are set.

For the V850ES/JF3-L, the setting is 0xFCC0.

For the V850ES/JF3-L, P30 to P35, P38, and P39 are set.

For the V850ES/JF3-L, these are not set up because the registers do not exist.

For the V850ES/JF3-L, P70 to P77 are set.

For the V850ES/JF3-L, the setting is 0x1C3C.

For the V850ES/JF3-L, P90, P91, P96 to P99, and P913 to P915 are set.

```
    PDH   = 0x00;                          /* Sets PDH0 to PDH5 to output a low level signal. */
    PMDH  = 0xC0;
    PMCDH = 0x00;

    PDL   = 0x0000;                        /* Sets PDL0 to PDL15 to output a low level signal. */
    PMDL  = 0x0000;
    PMCDL = 0x0000;

    return;
}


/*-----------------------------------------*/
/* Timer P0 (TMP0) setup (PWM output)      */
/*-----------------------------------------*/
static void f_init_int_tmp( void )
{
   /* Initialization of variable */
   usTMP0cnt = 0;                  /* Initializes the 500 ms counter.                    */

   /* Timer P0 settings */
   TP0CTL0 = 0x00;                 /* Sets the count clock to fxx.                       */
                                   /* Failsafe: Clears TP0CE to 0 and stops TMP0.        */
   TP0CTL1 = 0x04;                 /* Specifies the PWM output mode                      */
   TP0IOC0 = 0x04;                 /* Specifies the active level of the TOP01 pin as high */
                                   /* Enables the output from the TOP01 pin              */
   TP0CCR0 = PWM_CYCLE;            /* Sets the PWM cycle to 2 ms                         */
   TP0CCR1 = DUTY_PER_0;           /* Sets the initial duty to 0%                        */
   TP0CE   = 1;                    /* Starts TMP0.                                       */

/* Caution 1: The registers below are not set up in the PWM output mode:                 */
/*               <Registers that are not set up>                                         */
/*               - TP0IOC1 register                                                      */
/*               - TP0OPT0 register                                                      */
/*               - TP0CNT register                                                       */
/* Caution 2: The registers below are not set up if the internal count clock is used     */
/*            in the PWM output mode:                                                    */
/*               <Registers that are not set up>                                         */
/*               - TP0IOC2 register                                                      */

    /* Interrupt settings */
    TP0CCIC0 = 0x07;       /* Sets the priority of INTTP0CC0 to level 7 and unmask it.  */
}
```

For the V850ES/JF3-L, the setting is 0xFC.

For the V850ES/JF3-L, PDH0 and PDH1 are set.

When using TMQ, specify values for the TQ0xxx registers.

When using TMQ, specify a value for the TQ0CCIC0 register.

```
/*****************************/
/*    Interrupt module       */
/*****************************/
__interrupt
void f_int_inttp0cc0( void )
{

/* Overview                                                              */
/* A timer interrupt that occurs every 2 ms is used to count up to 500 ms.*/
/* LED1 turns on or off 500 ms after changing the duty.                  */

    usTMP0cnt++;                              /* Increments the 500 ms counter.              */

    if ( usTMP0cnt >= PWM_INT_CNT )           /* Processing where INTTP0CC0 occurs 250 times */
    {
    usTMP0cnt = 0;                            /* Initializes the 500 ms counter.             */

    PCM.3 ^= 1;                               /* Turns LED1 on or off.                       */

    if ( TP0CCR1 >= DUTY_PER_100 )            /* Processing if the duty is 100% or more      */
    {
         TP0CCR1 = DUTY_PER_0;                /* Initializes the duty to 0%.                 */
    }
    else
    {
         TP0CCR1 += DUTY_PER_25;              /* Increases the duty by 25%.                  */
    }
   }

   return;
}
```

When using TMQ, specify values for the TQ0CCR1 register.

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