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 Counter (TMP, TMQ))

External Event Count Mode

This document provides an operational overview of the sample program and describes how to use it and how to set up and use the external event count function of the 16-bit timer/event counter P (TMP) and the 16-bit timer/event counter Q (TMQ). In the sample program, the LED1 output is reversed when the falling edge of external pulse input is detected a certain number of times, by using the external event count function of TMP.

Target devices

V850ES/JF3-L microcontroller

V850ES/JG3-L microcontroller

CONTENTS

Document No. U19697EJ1V0AN00
Date Published April 2009 N

MINICUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany or a trademark in the United States of America.

- **The information in this document is current as of February, 2009. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, 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, 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 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).

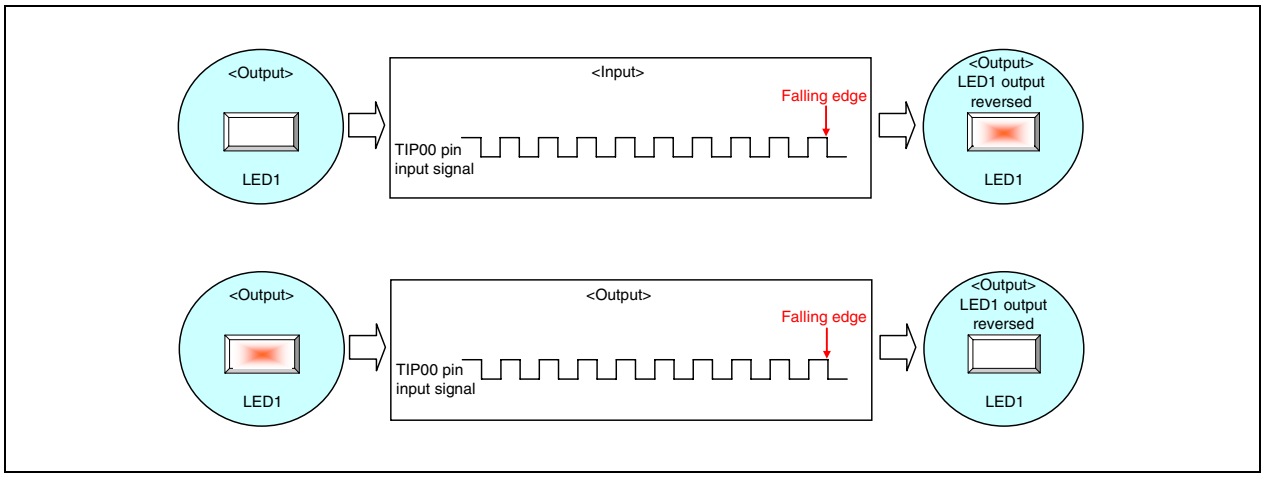
M8E 02. 11-1

CHAPTER 1 OVERVIEW

An example of using the external event count function of the 16-bit timer/event counter P (TMP) is presented in the sample program. An interrupt (INTTP0CC0) is generated by using the external event count function of TMP, and the LED1 output is reversed each time the falling edge of the external pulse input is detected 10 times.

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

The relationship between the number of external pulse input falling edge detections and LED1 output reversal is shown below.



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 up wait operations <wait: 1> for bus access to on-chip peripheral I/O registers
- Setting on-chip debug mode <normal operation mode>
- Stopping the internal oscillator and watchdog timer
- Setting the CPU clock frequency not to be divided
- Setting PLL mode and 20 MHz operation (5 MHz × 4)

<Pin settings>

- Setting up unused pins
- Setting up input pins (TIP00)
- Setting up LED1 output pins

<Timer M (TMM) settings>

- Setting the count clock to f_{XX}/64
- Specifying the TMM count value (comparison value)
- Enabling TMM
- Masking TMM interrupts

<Timer (TMP) settings>

- Setting the count clock to f_{XX} (20 MHz) by using the TP0CTL0 register
- Setting the operation mode to the external event count mode by using the TP0CTL1 register
- Specifying that counting be performed at the valid (falling) edge of the TIP00 pin signal by using the TP0IOC2 register
- Specifying the comparison value for the external event counter to the TP0CCR0 register
- Enabling TMP
- Setting the priority of INTTP0CC0 interrupts to level 7 and unmasking them

1.2 External Event Count Interrupt by 16-bit Timer/Event Counter P (TMP)

After specifying the initial settings, LED1 is made to blink by using the interrupt (INTTP0CC0) generated by the 16-bit timer/event counter P (TMP).

By using the external event count function of TMP, the LED1 output is reversed each time the falling edge of the external pulse input is detected 10 times.

1.3 External Events Generated by 16-bit Interval Timer M (TMM)

The external events for the 16-bit timer/event counter P (TMP) are generated by using the 16-bit Interval Timer M (TMM).

P30 and TIP00 (P32) are connected to use the P30 output signal as an external event.

By setting 16-bit interval timer M (TMM) interrupts to occur every 50 ms and controlling the P30 output signal level to reverse upon reception of an interrupt request, external events for TMP are generated at 100 ms cycles.

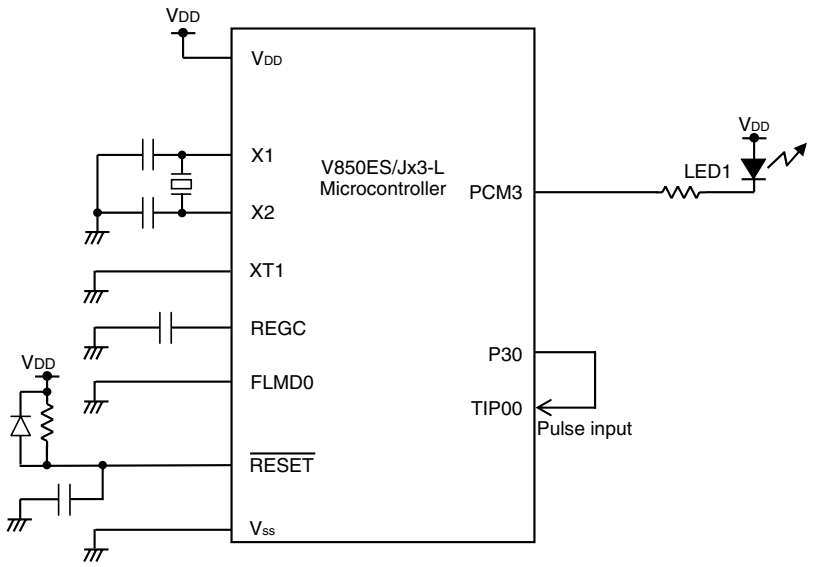
Caution See the product user's manual (V850ES/Jx3-L) for cautions on using the device.

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.



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 mode.

4. Connect REGC to GND via a capacitor (recommended value: 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.

- LED (LED1)
  LED1 is used as an output for the external event count function of the 16-bit timer/event counter P (TMP).

CHAPTER 3 SOFTWARE

This chapter describes the file configuration of the compressed files to be downloaded, on-chip peripherals of the microcontroller to be used, and the initial settings, and provides an operational overview of the sample program. A flowchart is also shown.

3.1 File Configuration

The following table shows the file configuration of the compressed files to be downloaded.

[C version]

File Name (Tree Structure)	Description	Compressed (*.zip) Files Included	
			
c — conf — crtE.s	Startup routine file[Note 1]	–	●
conf — AppNote_ExTrgCnt.dir	Link directive file[Note 2]	●	●
conf — AppNote_ExTrgCnt.prj	Project file for integrated development environment PM+	–	●
conf — AppNote_ExTrgCnt.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	●	●
src — minicube2.s	Source file for reserving the area for MINICUBE®2	●	●
src — 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 if “Specify startup file” is selected when creating a new workspace. (If the default installation path is used, the startup file will be a copy of C:\Program Files\NEC Electronics Tools\CA850*version used*\lib850\r32\crtE.s.)

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

Remark  : Only the source file is included.

 : The files to be 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:

- External event counter: 16-bit timer/event counter P (TMP)
- External pulse input: TIP00[Note]
- Output ports: P30, PCM3 (for lighting LED1)
- Interval timer: 16-bit interval timer M (TMM)

Note For the V850ES/JG3-L microcontroller, this is the pin that has the alternate functions ASCKA0/$\overline{\text{SCKB4}}$/TIP00/TOP00/P32.

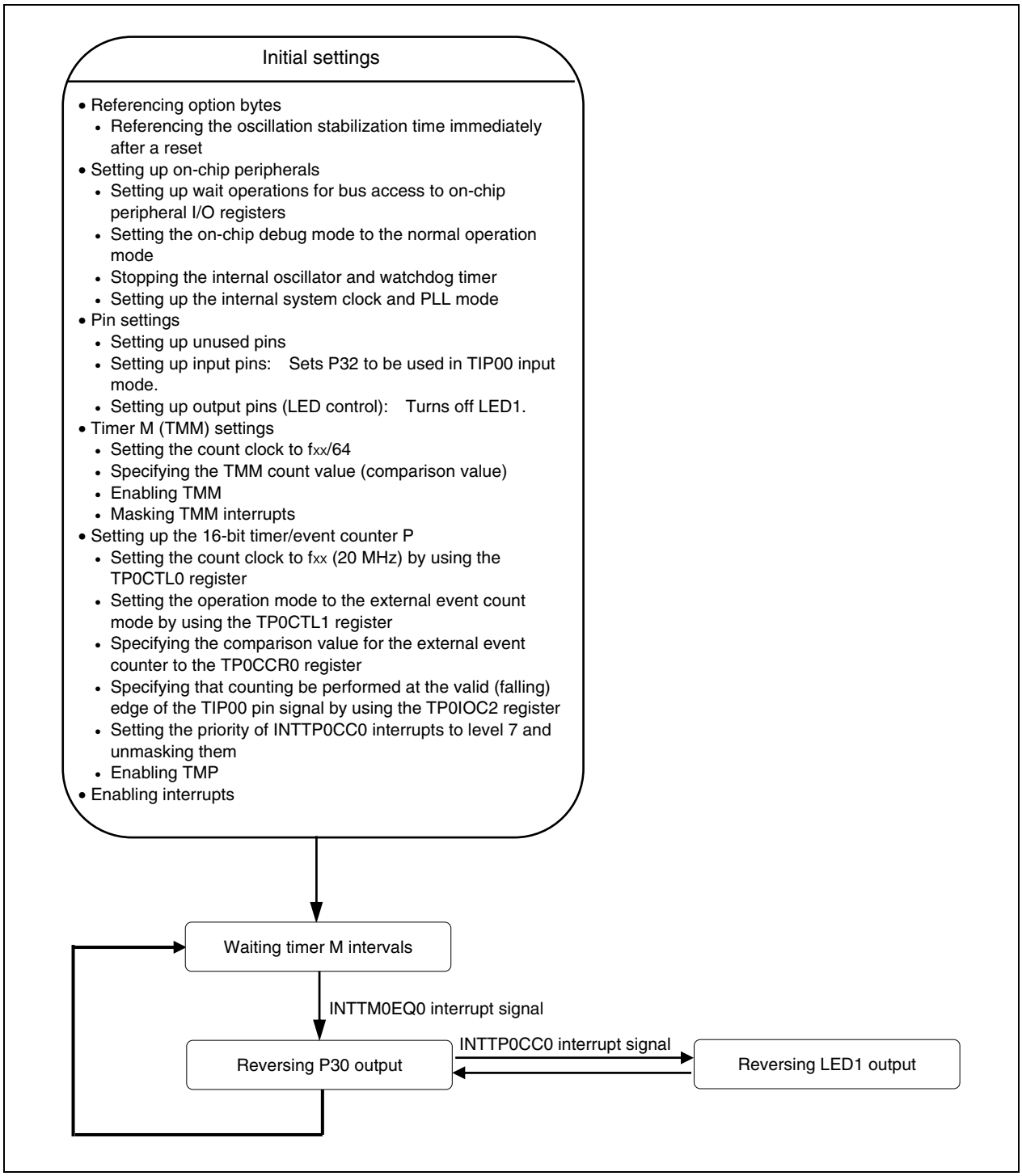
For the V850ES/JF3-L microcontroller, this is the pin that has the alternate functions ASCKA0/TIP00/TOP00/P32.

3.3 Initial Settings and Operational Overview

In the initial settings for the sample program, the clock frequency is selected, and settings for stopping the watchdog timer, setting up the I/O ports and the external pulse input pin, setting up the external event count function of the 16-bit timer/event counter P (TMP), setting up the 16-bit interval timer M (TMM), and setting up interrupts are specified.

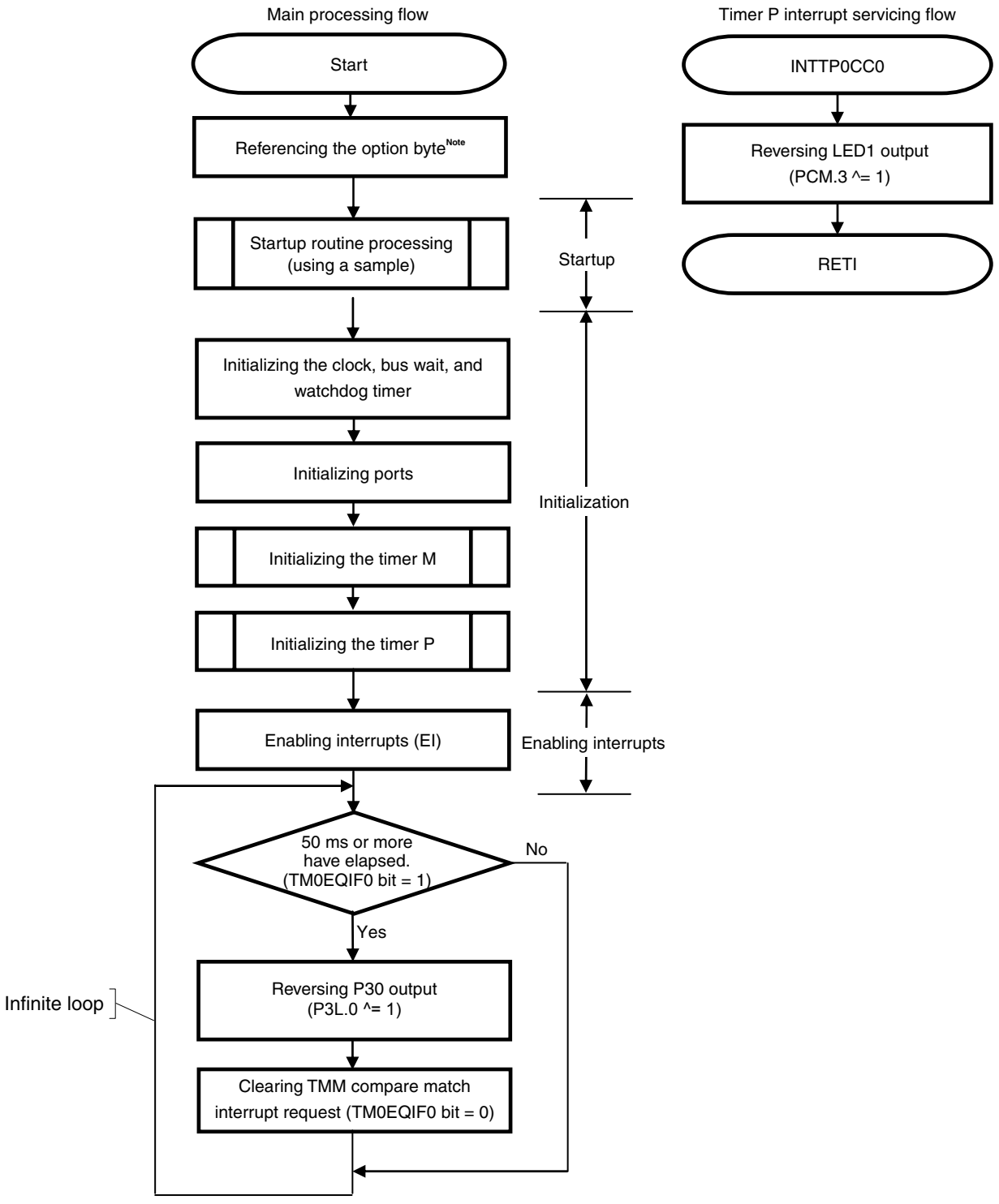
After specifying the initial settings, an interrupt (INTTP0CC0) generated by the 16-bit timer/event counter P (TMP), is used to reverse the LED1 output each time the falling edge of the external pulse input is detected 10 times. (The external events are generated by controlling the P30 output by using the 16-bit interval timer M (TMM).)

The details are provided in the status transition diagram below.

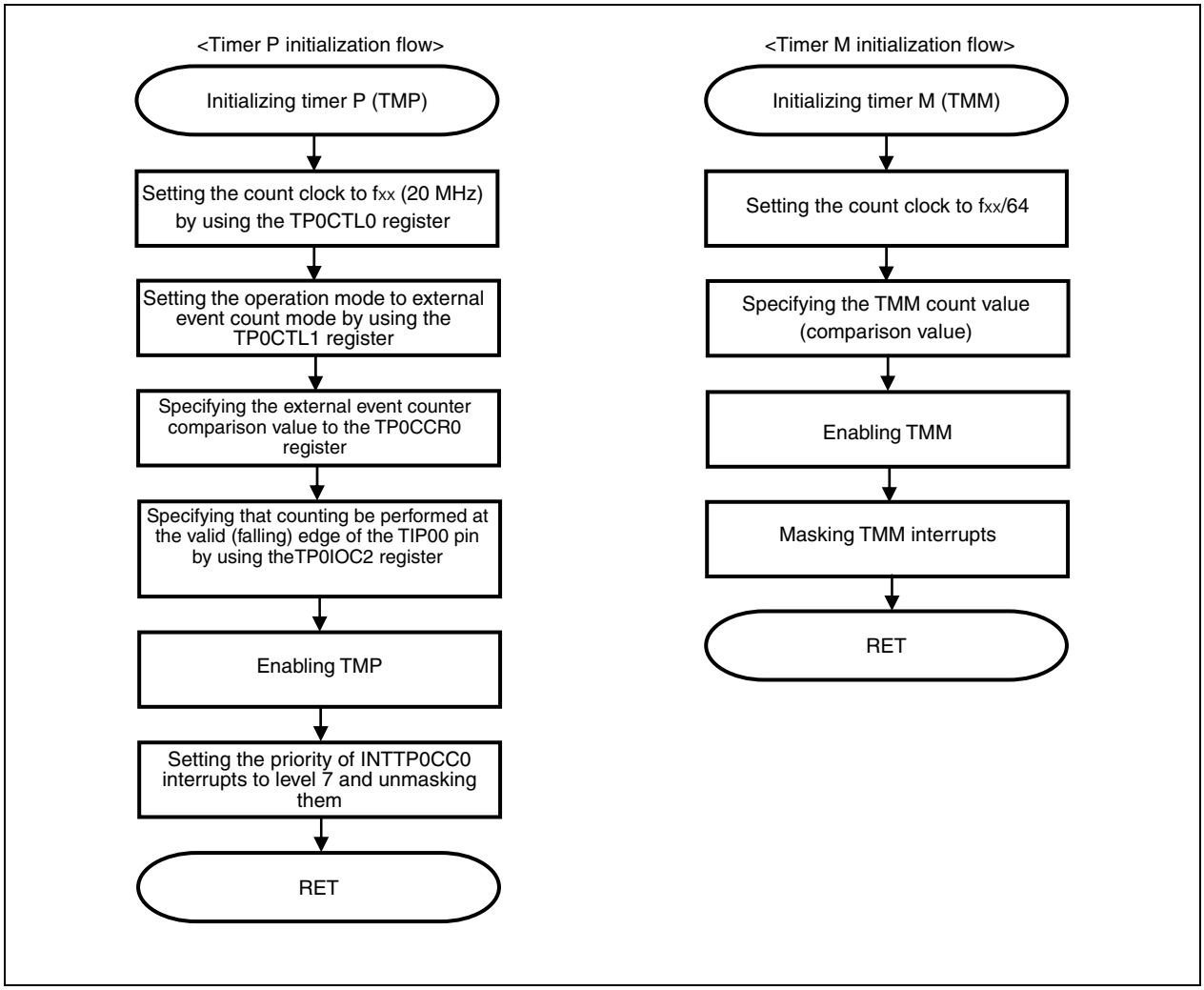


3.4 Flowcharts

Flowcharts for the sample program are shown below.



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.


<Timer P initialization flow>
Initializing timer P (TMP)
Setting the count clock to fxx (20 MHz) by using the TP0CTL0 register
Setting the operation mode to external event count mode by using the TP0CTL1 register
Specifying the external event counter comparison value to the TP0CCR0 register
Specifying that counting be performed at the valid (falling) edge of the TIP00 pin by using theTP0IOC2 register
Enabling TMP
Setting the priority of INTTP0CC0 interrupts to level 7 and unmasking them
RET
<Timer M initialization flow>
Initializing timer M (TMM)
Setting the count clock to fxx/64
Specifying the TMM count value (comparison value)
Enabling TMM
Masking TMM interrupts
RET

[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 issued
- 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 port 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

The 16-bit timer/event counter P (TMP) and the 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, the 16-bit timer/event counter P (TMP) is used. When using the 16-bit timer/event counter Q (TMQ), see **CHAPTER 4 SETTING 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 ID security, 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 the 16-bit timer/event counter P (TMP) and the 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 the 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 each product user's manual.

- 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 the 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)

Remark n = 0 to 5

Caution n = 0 in the sample program

The following eleven registers are used to set up the 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 the following:

- 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 the 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 in accordance with the register settings described in the user's manual. (Specifying values for these bits can be skipped because the external event count mode is specified using the TPnMD2 to TPnMD0 bits of the TPnCTL1 register and the TPnCKS2 to TPnCKS0 bits are not referenced.)

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	$f_{XX}/128$	$f_{XX}/512$

Caution 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 the 16-bit timer/event counter P (TMP).

In this sample program, the external event count mode is specified by specifying 001B 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 In the external event count mode, operation with external event count input is always enabled regardless of the TPnEEE bit value. (In this sample program, the TPnEEE bit is cleared to 0 in accordance with the register settings described in the user's manual.)

Caution 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 the external event count mode, the TPnIOC0 register does not have to be controlled. Therefore, the register is not controlled in this sample program.

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

Caution The TPnIOC0 register is not used in this sample program.

For the 16-bit timer/event counter Q (TMQ), 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 external event count 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

Caution The TPnIOC1 register is not used in the sample program.

For the 16-bit timer/event counter Q (TMQ), 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).

In the sample program, counting at the valid (falling) edge of the TIP00 pin signal is specified by specifying 10B for the TPnEES1 and TPnEES0 bits.

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

Caution The red values in the table are specified 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 external event count 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

Caution The TPnOPT0 register is not used in this sample program.

For the 16-bit timer/event counter Q (TMQ), the TQ0CCS3, TQ0CCS2, TQ0CCS1 and TQ0CCS0 bits are assigned to bits 7 to 4 of the TQ0OPT0 register.

4.1.7 Specifying external event counter comparison value

The number of external events to count can be controlled by using the TMPn capture/compare register 0 (TPnCCR0).

When the counter value of TMP reaches the value specified for the TPnCCR0 register, an INTTPnCC0 interrupt occurs. When the TMP value reaches the TPnCCR1 register value, an INTTPnCC1 interrupt occurs.

In this sample program, it is necessary to specify how many external events are to occur before reversing LED1, by using the TPnCCR0 register. The TPnCCR1 register is not used.

- External events to count = (N + 1)

Remark N: The value specified for the TPnCCR0 or TPnCCR1 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Caution The TPnCCR1 register is not used in this sample program.

Four capture/compare registers (TQ0CCR0 to TQ0CCR3) are provided for the 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 operating (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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

[Example 1] When starting the timer with the 16-bit timer/event counter P (TMP) set to the external event count mode and the count clock set to fxx (20 MHz) (same as the sample program)

- Setup procedure
  - <1> Specify the count clock fxx (20 MHz).
  - <2> Specify the external event count mode.
  - <3> Specify that counting be performed at the valid (falling) edge of the TIP00 pin signal.
  - <4> Specify a value for the compare register.
  - <5> Start the timer (TMP).
  - <6> Unmask interrupts.

Program example (same as the sample program)

```
#pragma interrupt INTTP0CC0 f_int_inttp0cc0    /* Specifies the timer interrupt (INTTP0CC0) handler. */


static void f_init_int_tmp( void )
{
/* Timer settings */
TP0CTL0 = 0x00;         /* Specifies fxx (20 MHz) as the count clock                    */ <1>
TP0CTL1 = 0x01;         /* Specifies the external event count mode.                     */ <2>
TP0IOC2 = 0x08;         /* Specifies that counting be performed at the falling edge of the
                           TIP00 pin signal.                                            */ <3>
TP0CCR0 = EVENT_CNT;    /* Specifies a comparison value for the external event counter. */ <4>
TP0CE = 1;              /* Starts the timer (TMP).                                      */ <5>

/* Interrupt settings */
TP0CCIC0 = 0x07;        /* Sets the priority of INTTP0CC0 to level 7 and unmasks INTTP0CC0. */ <6>

return;
}


__interrupt
void f_int_inttp0cc0( void )
{
PCM.3 ^= 1;             /* Reverses the LED1 output.                                    */
return;                 /* reti by using the __interrupt modifier                       */
}
```

Registers the f_int_inttmp0cc0 function as the interrupt handler.

Sets to count external events 10 times by setting EVENT_CNT value (10 – 1).

Starts interrupt servicing by generating the INTTP0CC0 interrupt.

[Example 2] When starting the timer with the 16-bit timer/event counter Q (TMQ) set to the external event count mode and the count clock set to fxx (20 MHz)

Setup procedure

<1> Specify the count clock fxx (20 MHz).
<2> Specify the external event count mode.
<3> Specify that counting be performed at the valid (falling) edge of the TIQ00 pin.
<4> Specify a comparison value for the compare register.
<5> Start the timer (TMQ) operation.
<6> Unmask interrupts.

• Program example

```
#pragma interrupt INTTQ0CC0 f_int_inttq0cc0    /* Specifies the timer interrupt
(INTTQ0CC0) handler. */


static void f_init_int_tmq( void )
{
/* Timer settings */
TQ0CTL0 = 0x00;         /* Specifies fxx (20 MHz) as the count clock        */ } <1>
TQ0CTL1 = 0x01;         /* Specifies the external event count mode.         */ } <2>
TQ0IOC2 = 0x08;         /* Specifies that counting be performed at the falling
                           edge of the TIQ00 pin.                             */ } <3>
TQ0CCR0 = EVENT_CNT;    /* Specifies a comparison value for the external
                           event counter.
                                                                              */ } <4>
TQ0CE = 1;              /* Starts the timer (TMQ).                          */ } <5>

/* Interrupt settings */
TQ0CCIC0 = 0x07;        /* Sets the priority of INTTQ0CC0 to level 7 and
                           unmasks INTTQ0CC0.                               */ } <6>

return;
}


__interrupt
void f_int_inttq0cc0( void )
{
PCM.3 ^= 1;             /* Reverses the LED1 output.                        */
return;                 /* reti by using the __interrupt modifier           */
}
```

Registers the f_int_inttq0cc0 function as the interrupt handler.

Sets to count external events 10 times by setting EVENT_CNT value (10 – 1).

4.2 Specifying LED1 Blinking Cycle

In this sample program, the LED1 blinking cycle is specified as follows.

The LED1 blinking depends on external events.

In this sample program, external events are generated by controlling the output port (P30) connected to the external event count input pin (TIP00), by using the 16-bit interval timer M (TMM), each time an interval has elapsed.

External events therefore occur periodically in accordance with the cycles in which an interrupt is generated by the 16-bit interval timer M (TMM). As a result, interrupts due to an external event count match also periodically occur and the LED1 output is reversed each time an INTTP0CC0 interrupt occurs.

- TMM interrupt cycle (interval) = (N+1) /f_{CNT}
- TIP00 input signal cycle (interval) = TMM interrupt cycle × 2
- INTTP0CC0 signal generation timing = TIP00 input signal cycle × (value specified for the TP0CCR0 register + 1)
- LED1 output reversal timing = INTTP0CC0 signal generation timing
- LED1 blinking cycle = LED1 output reversal cycle × 2

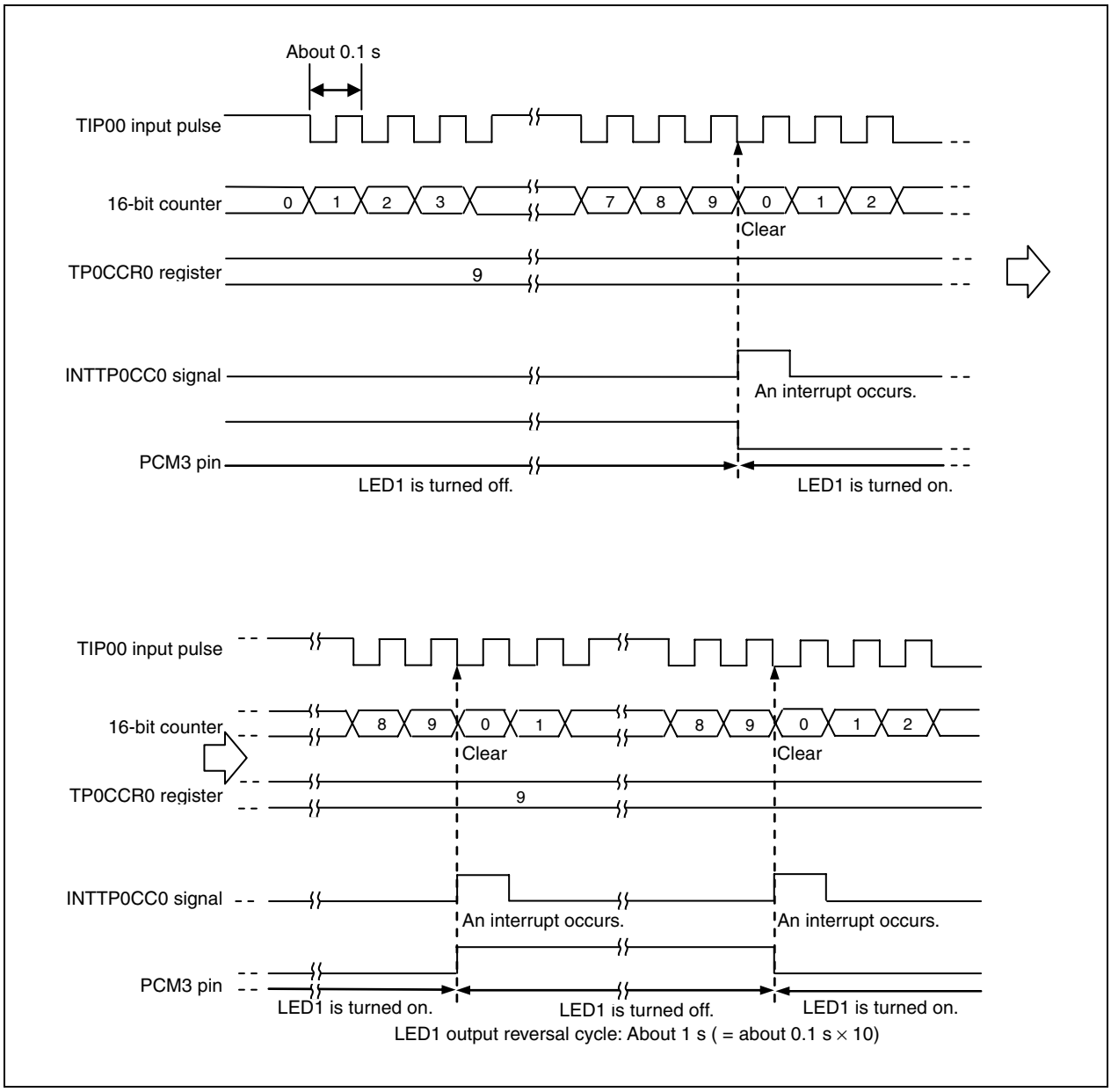
Remark N: The value specified for the TM0CMP0 register (the comparison value for 16-bit interval timer M (TMM))

f_{CNT}: The count clock frequency of the 16-bit interval timer M (TMM)

Calculation example: When the count clock frequency of the 16-bit interval timer M (TMM) is set to f_{XX} (20 MHz)/64, the TM0CMP0 register is set to 15,624, and the TP0CCR0 register of the external event counter (TMP) is set to 9:

- TMM interrupt cycle (interval) = (15,624 + 1) /(20 MHz/64) = 50 ms
- TIP00 input signal cycle (interval) = 50 ms × 2 = 100 ms
- INTTP0CC0 signal generation timing = 100 ms × (9 +1) = 1,000 ms = 1 s
- LED1 output reversal timing = 1 s
- LED1 blinking cycle = 1 s × 2 = 2 s

Figure 4-10. Timing Chart Example of External Event Counter Operation (When LED1 Blinks at Cycles of 2 s)



4.3 Setting Up 16-bit Interval Timer M (TMM)

The following two registers are set up when using the 16-bit interval timer M (TMM):

- TMM0 control register 0 (TM0CTL0)
- TMM0 compare register 0 (TM0CMP0)

4.3.1 Setting 16-bit interval timer M (TMM) operation clock

The TMM0 control register 0 (TM0CTL0) selects the count clock of the 16-bit interval timer M (TMM) and controls the counter.

Values must be specified for bits TM0CKS2 to TM0CKS0 when the TM0CE bit is 0.

In the sample program, 0x03 is specified at initialization and bits TM0CKS2 to TM0CKS0 are set up to select fxx/64 (20 MHz/64).

After setting up the 16-bit interval timer M (TMM) registers, set the TM0CE bit to 1.

Figure 4-11. TM0CTL0 Register Format

TMM0 control register 0 (TP0CTL0)
Address: 0xFFFFF690

7	6	5	4	3	2	1	0
TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	TMM0 operation control
0	TMM0 disabled (TMM0 reset asynchronously).
1	TMM0 enabled. TMM0 starts.

TM0CKS2	TM0CKS1	TM0CKS0	Internal count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/64$
1	0	0	$f_{XX}/512$
1	0	1	INTWT
1	1	0	$f_R/8$
1	1	1	f_{XT}

Caution The red values are specified in the sample program.

4.3.2 Controlling intervals

Intervals can be controlled using TMM0 compare register 0 (TM0CMP0).

When the count value of the timer M reaches the value specified for the TM0CMP0 register, an INTTM0EQ0 interrupt occurs.

In the sample program, this register is set to 15,624 to specify a 50 ms interval.

- Interval = (N + 1)/f_{CLK}

Remark N: The value specified for the TM0CMP0 register
f_{CLK}: The count clock frequency of the 16-bit interval timer M

Figure 4-12. TM0CMP0 Register Format

TMM0 compare register 0 (TM0CMP0)

Address: 0xFFFFF694

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CHAPTER 5 RELATED DOCUMENTS

Document	English
V850ES/JF3-L Hardware User's Manual	PDF
V850ES/JG3-L Hardware User's Manual	PDF
PM+ Ver.6.30 User's Manual	PDF
CA850 Ver.3.20 C Compiler Package Operation User's Manual	PDF
CA850 Ver.3.20 C Compiler Package C Language User's Manual	PDF
CA850 Ver.3.20 C Compiler Package Assembly Language User's Manual	PDF
CA850 Ver.3.20 C Compiler Package Link Directive User's Manual	PDF
V850ES Architecture User's Manual	PDF
QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual	PDF
ID850QB Ver. 3.40 Integrated Debugger Operation User's Manual	PDF

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
#-------------------------------------------------------------------------------
#    External Event Count Mode
#-------------------------------------------------------------------------------
#[History]
#    2009.1.--   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
#-------------------------------------------------------------------------------
#    External Event Count Mode
#-------------------------------------------------------------------------------
#[History]
#    2009.1.--   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_LVI.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 for products with 128 KB internal ROM

Difference from the default link directive file (additional code)

A reserved area 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)

A reserved area for MINICUBE2 is allocated.

● main.c

```
/*---------------------------------------------------------------------------*/
/*
/*    NEC Electronics     V850ES/Jx3-L microcontroller
/*
/*---------------------------------------------------------------------------*/
/*    V850ES/JG3-L sample program
/*---------------------------------------------------------------------------*/
/*    External Event Count Mode
/*---------------------------------------------------------------------------*/
/*[History]
/*    2009.1.--   Released
/*---------------------------------------------------------------------------*/
/*[Overview]
/*    This sample program shows an example of using the external event count mode
/*    of the 16-bit timer/event counter (TMP).
/*    The falling edges of the external clock pulses input to the TIP00 pin are
/*    counted and an interrupt is generated every ten pulses to reverse LED1.
/*
/*    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 M interval to 50 ms
/*    • Defining the comparison value for the timer P external event counter
/*    • Declaring prototypes
/*    • Setting up a bus wait for on-chip peripheral I/O registers, stopping the watchdog timer,
       and setting up the clock
/*    • Initializing unused ports
/*    • Initializing the TIP00 input port (falling edge) and LED1 output port
/*    • Initializing the timer M (TMM)
/*    • Initializing the timer P (TMP)
/*
/*    <Timer M settings>
/*    • Setting the count clock to fxx/64
/*    • Specifying the TMM count value (comparison value)
/*    • Enabling TMM
/*    • Masking timer M interrupts (a fail safe)
/*
```

```
/*    <Timer P settings>
/*    Setting the count clock to fxx (20 MHz) by using the TP0CTL0 register
/*    Setting the operation mode to the external event count mode by using the TP0CTL1 register
/*    Setting counting to be performed at the valid (falling) edge of the TIP00 pin signal by
/*    using the TP0IOC2 register
/*    Specifying a comparison value for the external event counter by using the TP0CCR0 register
/*    Enabling TMP
/*    Setting the priority of INTTP0CC0 interrupts to level 7 and unmasking them
/*
/*    <Timer P interrupt (INTTP0CC0) servicing>
/*    • Making LED1 blink
/*
/*[I/O port settings]
/*
/* Input port: P32(TIP00)
/* Output ports: P30, PCM3
/* Unused ports: P02 to P06, P10 to P11, P31, P33 to P39, P50 to P55, P70 to P711, P90 to P915,
/*               PCM0 to PCM2, PCT0, PCT1, PCT4, PCT6, PDH0 to PDH5, PDL0 to 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 P interrupt (INTTP0CC0) handler.*/


/*---------------------------*/
/*    Constant definitions   */
/*---------------------------*/
#define VAL_50ms_CNT    (15625-1)          /* Defines the constant for a 50 ms wait adjustment. */
#define EVENT_CNT       (10-1)             /* Defines the comparison value for the timer P
                                              external event counter.*/


/*---------------------------*/
/*   Prototype definitions   */
/*---------------------------*/
static    void f_init( void );              /* Initialization function                          */
static    void f_init_clk_bus_wdt2( void );/* Clock bus WDT initialization function            */
static    void f_init_port_func( void );    /* Port/alternate-function initialization function  */
static    void f_init_int_tmm( void );      /* TMM0 initialization function                     */
static    void f_init_int_tmp( void );      /* TMP initialization function                      */
```

When using the TMQ, specify INTTQ0CC0.

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

     __EI();                        /* Enables interrupts.                                   */

    while( 1 )                      /* Main loop (infinite loop)                             */
        {
           if( TM0EQIF0 == 1 )      /* Is an INTTM0EQ0 interrupt request signal being
                                       generated?                                            */
               {
                  P3L.0 ^= 1;       /* Reverses P30 output.                                  */
                  TM0EQIF0 = 0;     /* Clears a TMM0 compare match interrupt request.        */
               }
        }

    return;
}



/*---------------------------------*/
/*      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_tmm();               /* Sets up the TMM0 timer.                               */

    f_init_int_tmp();               /* Sets up the TMP 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 a special register 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;

    P3    = 0x0000;                        /* Sets P30, P31, P33 to P39 to output a low level
                                              signal and sets P32 to the TIP00 input mode.   */
    PM3   = 0xFC00;
    PMC3  = 0x0000;
    PFCE3L = 0x04;
    PMC3 = 0x0004;

#if(0) /* To use P4 as 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 V850ES/JF3-L, the setting is 0xFE.

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

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

For V850ES/JF3-L, P30 and P31, P33 to P35, P38, and P39 are set.

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

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

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

For 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 M settings      */
/*----------------------------*/
static void f_init_int_tmm( void )
{
    /* Timer M settings */
    TM0CTL0 = 0x03;                 /* Disables TMM0 and sets the count clock to
                                       fxx/64. */
    TM0CMP0 = VAL_50ms_CNT;         /* Specifies the TMM0 count value. */
    TM0CE   = 1;                    /* Enables TMM0. */

    /* Interrupt settings */
    TM0EQMK0 = 1;                   /* Fail safe: Masks timer M interrupts. */
}
```

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

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

```
/*---------------------------*/
/*    Timer P settings       */
/*---------------------------*/
static void f_init_int_tmp( void )
{
     /* Timer P settings */
     TP0CTL0 = 0x00;               /* Sets the count clock to fxx. */
                                   /* Fail safe: Clears TP0CE to 0 and stops TMP. */
     TP0CTL1 = 0x01;               /* Specifies the external event count mode. */
     TP0IOC2 = 0x08;               /* Count clock = the falling edge of the TIP00 pin signal */
     TP0CCR0 = EVENT_CNT;          /* Specifies a comparison value for the compare register. */
     TP0CE   = 1;                  /* Starts the timer P (TMP). */

/* Caution: The following registers are not set up in the external event count mode. */
/*              <Registers that are not set up>      */
/*              - TP0IOC0 register                   */
/*              - TP0I0C1 register                   */
/*              - TP0OPT0 register                   */
/*              - TP0CCR1 register                   */
/*              - TP0CNT register                    */

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

/*****************************/
/*    Interrupt module       */
/*****************************/
__interrupt
void f_int_inttp0cc0( void )
{
     PCM.3 ^= 1;                   /* Reverses the LED1 output. */

     return;                       /* reti by using the _interrupt modifier */
}
```

When using TMQ, specify values for the TQ0xxx registers.

When using TMQ, specify a value for the TQ0CCIC0 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