

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.

Application Note

78K0S/Kx1+

Sample Program (16-bit Timer/Event Counter 00)

One-Shot Pulse Output

This document describes an operation overview of the sample program and how to use it, as well as how to set and use the one-shot pulse output function of 16-bit timer/event counter 00. In the sample program, the one-shot pulse output function of 16-bit timer/event counter 00 is used to output a one-shot pulse after a given delay time elapses, after the rising edge of the external signal that is to be input to the TI000 pin has been detected. Furthermore, the active pulse width is changed in accordance with the number of switch inputs.

CONTENTS

Target devices

- 78K0S/KA1+ microcontroller
- 78K0S/KB1+ microcontroller
- 78K0S/KU1+ microcontroller
- 78K0S/KY1+ microcontroller

CHAPTER 1 OVERVIEW	3
1.1 Main Contents of the Initial Settings.....	3
1.2 Contents Following the Main Loop.....	4
CHAPTER 2 CIRCUIT DIAGRAM	5
2.1 Circuit Diagram.....	5
2.2 Peripheral Hardware.....	5
CHAPTER 3 SOFTWARE	6
3.1 File Configuration.....	6
3.2 Internal Peripheral Functions to Be Used.....	7
3.3 Initial Settings and Operation Overview.....	7
3.4 Flow Charts.....	9
CHAPTER 4 SETTING METHODS	10
4.1 Setting the One-Shot Pulse Output Function of 16-bit Timer/Event Counter 00.....	10
4.2 Setting the One-Shot Pulse Active Width.....	27
4.3 Setting the Chattering Detection Time.....	28
CHAPTER 5 OPERATION CHECK USING SYSTEM SIMULATOR SM+	29
5.1 Building the Sample Program.....	29
5.2 Operation with SM+.....	31
CHAPTER 6 RELATED DOCUMENTS	35
APPENDIX A PROGRAM LIST	36
APPENDIX B REVISION HISTORY	50

Document No. U18891EJ2V0AN00 (2nd edition)
Date Published September 2008 NS

• **The information in this document is current as of July, 2008. 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).

CHAPTER 1 OVERVIEW

An example of using the one-shot pulse output function of 16-bit timer/event counter 00 is presented in this sample program. A one-shot pulse is output after a given delay time elapses, after the rising edge of the external signal that is to be input to the TI000 pin has been detected. Furthermore, the active pulse width is changed in accordance with the number of switch inputs.

1.1 Main Contents of the Initial Settings

The main contents of the initial settings are as follows.

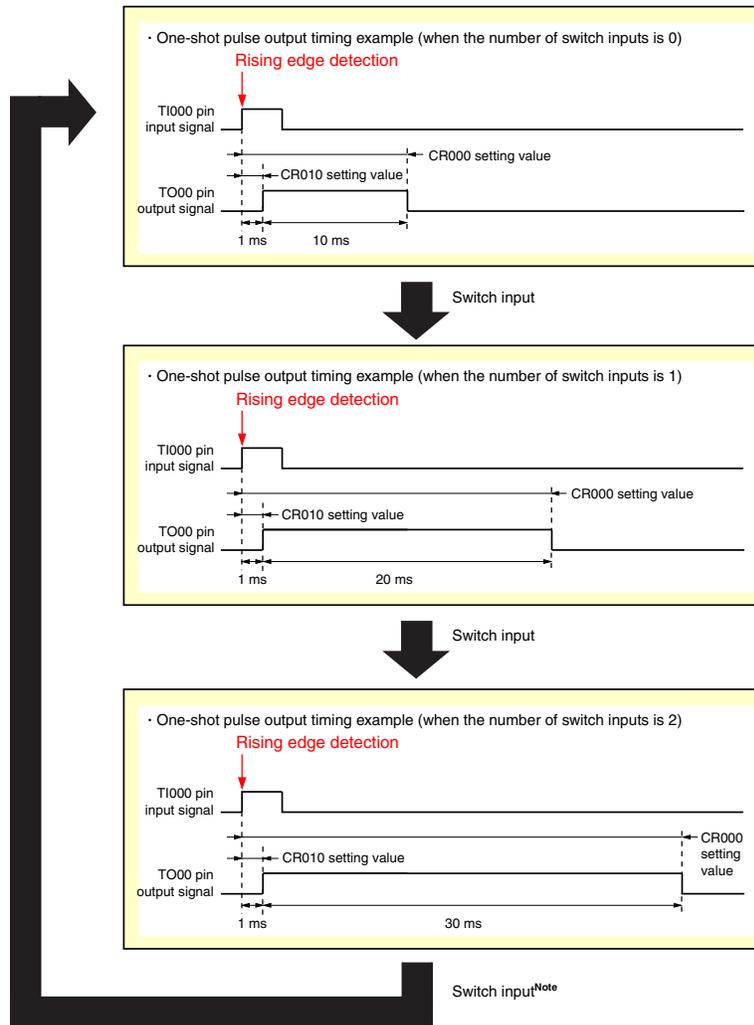
- Selecting the high-speed internal oscillator as the system clock source^{Note}
- Stopping watchdog timer operation
- Setting V_{LVI} (low-voltage detection voltage) to $4.3\text{ V} \pm 0.2\text{ V}$
- Generating an internal reset (LVI reset) signal when it is detected that V_{DD} is less than V_{LVI} , after V_{DD} (power supply voltage) becomes greater than or equal to V_{LVI}
- Setting the CPU clock frequency to 8 MHz
- Setting the I/O ports
- Setting 16-bit timer/event counter 00
 - Setting CR000 and CR010 as compare registers
 - Setting to 1 ms ($0.5\ \mu\text{s} \times 2,000$) the delay time from detection of the valid edge of the TI000 pin to active output
 - Setting the active output width to 10 ms ($0.5\ \mu\text{s} \times 20,000$)
 - Setting the count clock to $f_{XP}/2^2$ (2 MHz)
 - Setting the valid edge of the TI000 pin to the rising edge
 - Setting the output mode to one-shot pulse output
 - Enabling timer output reversal caused upon a match between CR010 and TM00, or CR000 and TM00
 - Setting the initial timer output value to 0 (resetting (0) the timer output F/F)
 - Enabling timer output (TO00 pin output)
 - Setting the operation mode to clear & start upon detection of the valid edge of the TI000 pin
- Setting the valid edge of INTP1 (external interrupt) to the falling edge
- Enabling INTP1 interrupts

Note This is set by using the option byte.

1.2 Contents Following the Main Loop

After completion of the initial settings, a one-shot pulse is output after a given delay time elapses, after the rising edge of the external signal that is to be input to the TI000 pin has been detected.

An INTP1 interrupt is serviced when the falling edge of the INTP1 pin, which is generated by switch input, is detected. Chattering is identified when INTP1 is at high level (switch is off), after 10 ms have elapsed since a fall of the INTP1 pin was detected. The active pulse width is changed in accordance with the number of switch inputs when INTP1 is at low level (switch is on), after 10 ms have elapsed since an edge was detected.



Note The active pulse widths from the zeroth switch input are repeated after the third switch input.

Caution For cautions when using the device, refer to the user's manual of each product ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)).



[Column] Chattering

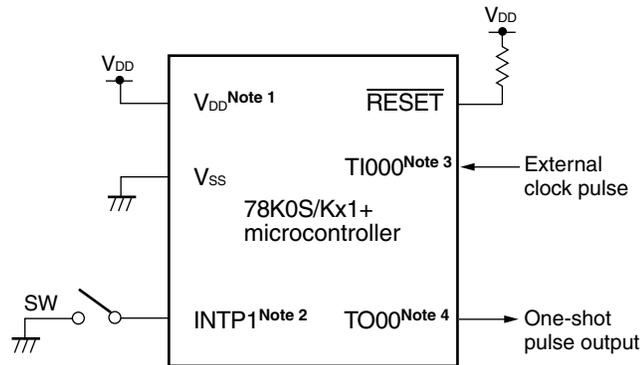
Chattering is a phenomenon in which the electric signal repeats turning on and off due to a mechanical flip-flop of the contacts, immediately after the switch has been pressed.

CHAPTER 2 CIRCUIT DIAGRAM

This chapter describes a circuit diagram and the peripheral hardware to be used in this sample program.

2.1 Circuit Diagram

A circuit diagram is shown below.



Notes 1. Use this in a voltage range of $4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$.

- INTP1/TxD6/P43: 78K0S/KA1+ and 78K0S/KB1+ microcontrollers
INTP1/P32: 78K0S/KY1+ and 78K0S/KU1+ microcontrollers
- TI000/INTP0/P30: 78K0S/KA1+ and 78K0S/KB1+ microcontrollers
TI000/ANI0/TOH1/P20: 78K0S/KY1+ and 78K0S/KU1+ microcontrollers
- TO00/TI010/INTP2/P31: 78K0S/KA1+ and 78K0S/KB1+ microcontrollers
TO00/TI010/INTP0/ANI1/P21: 78K0S/KY1+ and 78K0S/KU1+ microcontrollers

- Cautions**
- Connect the AV_{REF} pin directly to V_{DD} (only for the 78K0S/KA1+ and 78K0S/KB1+ microcontrollers).
 - Connect the AV_{SS} pin directly to GND (only for the 78K0S/KB1+ microcontroller).
 - Leave all unused pins open (unconnected), except for the pins shown in the circuit diagram and the AV_{REF} and AV_{SS} pins.

2.2 Peripheral Hardware

The peripheral hardware to be used is shown below.

- Switch (SW)

A switch is used as an input to control the active width of a one-shot pulse output.

CHAPTER 3 SOFTWARE

This chapter describes the file configuration of the compressed file to be downloaded, internal peripheral functions of the microcontroller to be used, and initial settings and operation overview of the sample program, and shows a flow chart.

3.1 File Configuration

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

File Name	Description	Compressed (*.zip) File Included		
				
main.asm (Assembly language version) ----- main.c (C language version)	Source file for hardware initialization processing and main processing of microcontroller	● Note 1	● Note 1	
op.asm	Assembler source file for setting the option byte (sets the system clock source)	●	●	
tm00one.prw	Work space file for integrated development environment PM+		●	
tm00one.prj	Project file for integrated development environment PM+		●	
tm00one.pri tm00one.prs tm00one.prm	Project files for system simulator SM+ for 78K0S/Kx1+		● Note 2	
tm00one0.pnl	I/O panel file for system simulator SM+ for 78K0S/Kx1+ (used for checking peripheral hardware operations)		● Note 2	●
tm00one0.wvo	Timing chart file for system simulator SM+ for 78K0S/Kx1+ (used for checking waveforms)			●

Notes 1. “main.asm” is included with the assembly language version, and “main.c” with the C language version.

2. These files are not included among the files for the 78K0S/KU1+ microcontroller.

Remark



: Only the source file is included.



: The files to be used with integrated development environment PM+ and 78K0S/Kx1+ system simulator SM+ are included.



: The microcontroller operation simulation file to be used with system simulator SM+ for 78K0S/Kx1+ is included.

3.2 Internal Peripheral Functions to Be Used

The following internal peripheral functions of the microcontroller are used in this sample program.

- One-shot pulse output function: 16-bit timer/event counter 00
- $V_{DD} < V_{LVI}$ detection: Low-voltage detector (LVI)
- Switch input: INTP1^{Note 1} (external interrupt)
- External pulse input: TI000^{Note 2} (timer input)
- One-shot pulse output: TO00^{Note 3} (timer output)

Notes 1.	INTP1/P43:	78K0S/KA1+ and 78K0S/KB1+ microcontrollers
	INTP1/P32:	78K0S/KY1+ and 78K0S/KU1+ microcontrollers
2.	TI000/INTP0/P30:	78K0S/KA1+ and 78K0S/KB1+ microcontrollers
	TI000/ANI0/TOH1/P20:	78K0S/KY1+ and 78K0S/KU1+ microcontrollers
3.	TO00/TI010/INTP2/P31:	78K0S/KA1+ and 78K0S/KB1+ microcontrollers
	TO00/TI010/INTP0/ANI1/P21:	78K0S/KY1+ and 78K0S/KU1+ microcontrollers

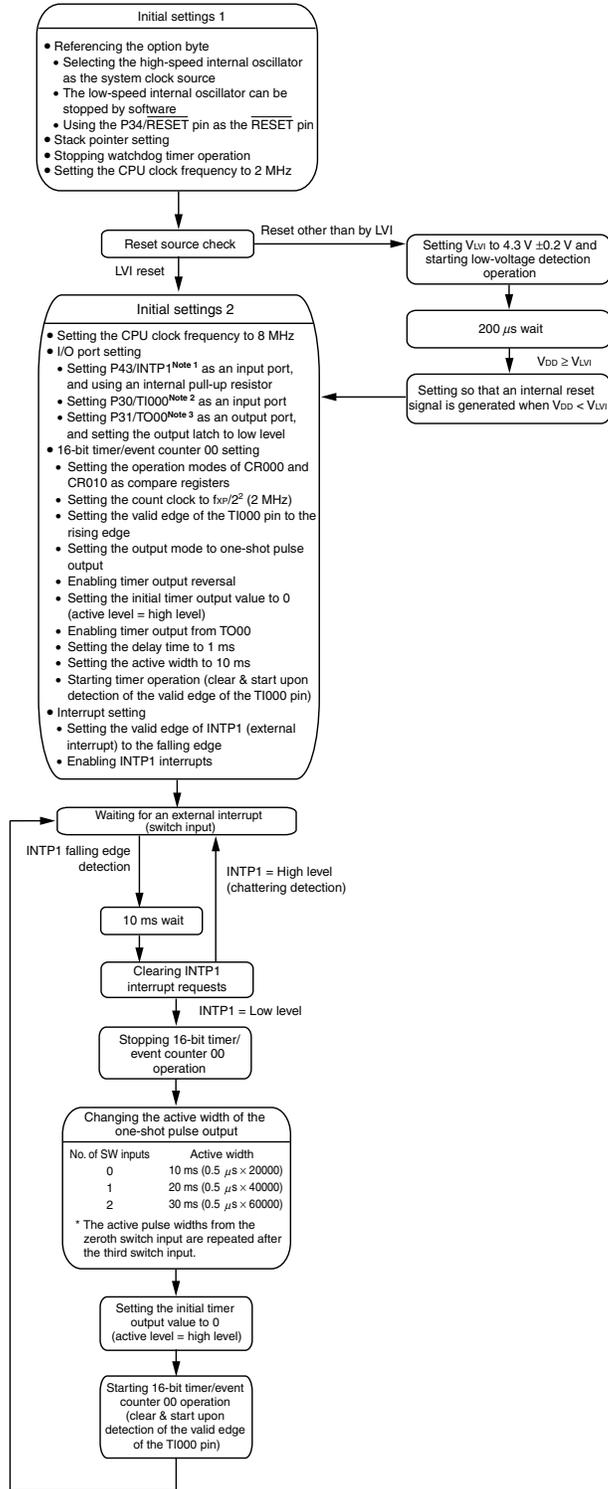
3.3 Initial Settings and Operation Overview

In this sample program, initial settings including the setting of the low-voltage detection function, selection of the clock frequency, setting of the I/O ports, setting of 16-bit timer/event counter 00 (one-shot pulse output function), and setting of interrupts are performed.

After completion of the initial settings, a one-shot pulse is output after a given delay time elapses, after the rising edge of the external signal that is to be input to the TI000 pin has been detected.

An INTP1 interrupt is serviced when the falling edge of the INTP1 pin, which is generated by switch input, is detected. Chattering is identified when INTP1 is at high level (switch is off), after 10 ms have elapsed since a fall of the INTP1 pin was detected. The active pulse width is changed in accordance with the number of switch inputs when INTP1 is at low level (switch is on), after 10 ms have elapsed since an edge was detected.

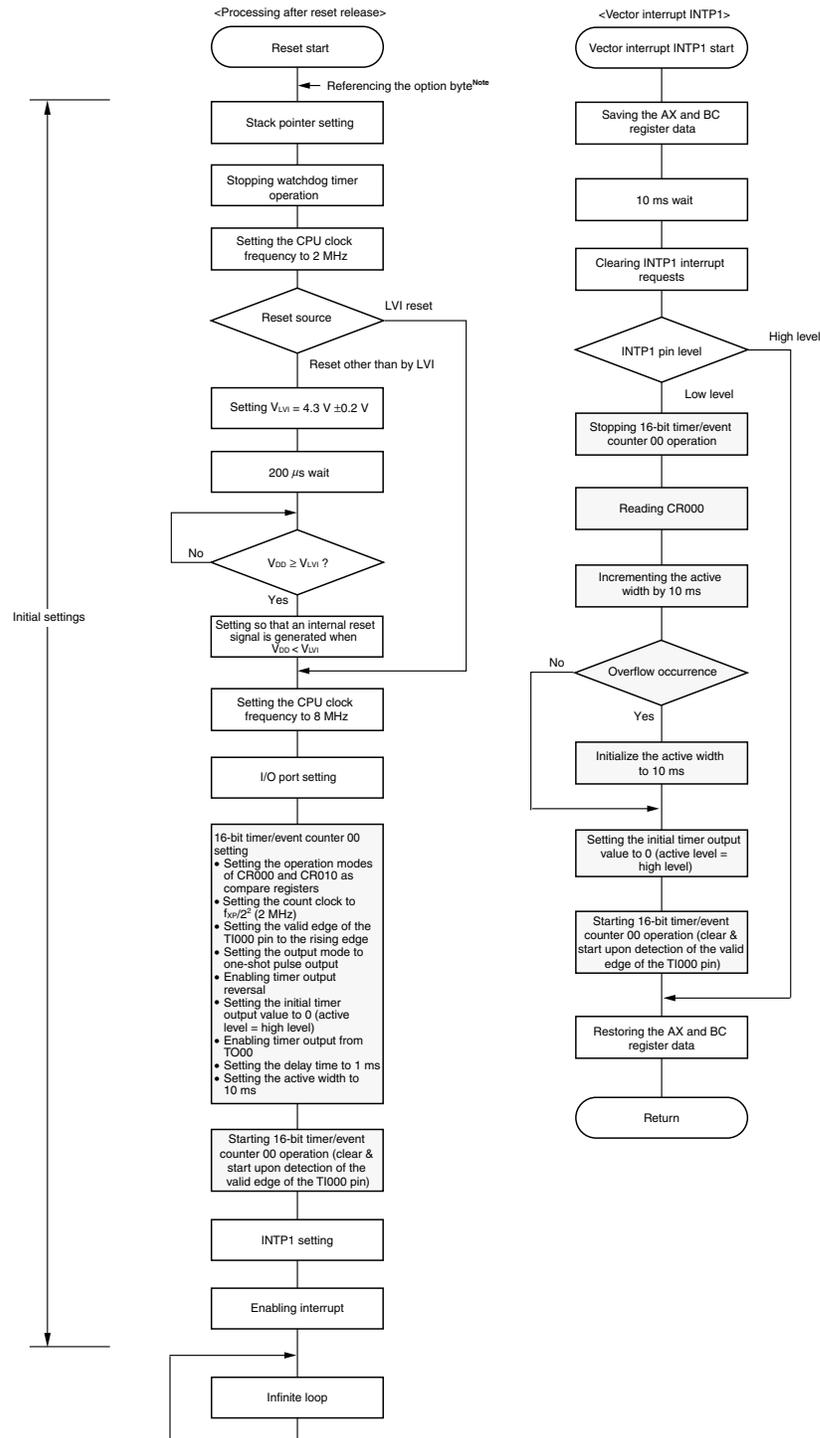
The details are described in the status transition diagram shown below.



- Notes 1.** INTP1/P43: 78K0S/KA1+ and 78K0S/KB1+ microcontrollers
 INTP1/P32: 78K0S/KY1+ and 78K0S/KU1+ microcontrollers
- 2.** TI000/P30: 78K0S/KA1+ and 78K0S/KB1+ microcontrollers
 TI000/P20: 78K0S/KY1+ and 78K0S/KU1+ microcontrollers
- 3.** TO00/P31: 78K0S/KA1+ and 78K0S/KB1+ microcontrollers
 TO00/P21: 78K0S/KY1+ and 78K0S/KU1+ microcontrollers

3.4 Flow Charts

The flow charts for the sample program are shown below.



Note Referencing the option byte is automatically performed by the microcontroller after reset release. In this sample program, the following contents are set by referencing the option byte.

- Using the high-speed internal oscillation clock (8 MHz (TYP.)) as the system clock source
- The low-speed internal oscillator can be stopped by using software
- Using the P34/ $\overline{\text{RESET}}$ pin as the $\overline{\text{RESET}}$ pin

CHAPTER 4 SETTING METHODS

This chapter describes the one-shot pulse output function of 16-bit timer/event counter 00.

For other initial settings, refer to the [78K0S/Kx1+ Sample Program \(Initial Settings\) LED Lighting Switch Control Application Note](#). For interrupt, refer to the [78K0S/Kx1+ Sample Program \(Interrupt\) External Interrupt Generated by Switch Input Application Note](#). For low-voltage detection (LVI), refer to the [78K0S/Kx1+ Sample Program \(Low-Voltage Detection\) Reset Generation During Detection at Less than 2.7 V Application Note](#).

For how to set registers, refer to the user's manual of each product ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)).

For assembler instructions, refer to the [78K/0S Series Instructions User's Manual](#).

4.1 Setting the One-Shot Pulse Output Function of 16-bit Timer/Event Counter 00

The following nine types of registers are set when using the one-shot pulse output function of 16-bit timer/event counter 00.

- Capture/compare control register 00 (CRC00)
- 16-bit timer capture/compare register 000 (CR000)
- 16-bit timer capture/compare register 010 (CR010)
- Prescaler mode register 00 (PRM00)
- 16-bit timer output control register 00 (TOC00)
- 16-bit timer mode control register 00 (TMC00)
- Port register x (Px)^{Note}
- Port mode register x (PMx)^{Note}
- Port mode control register x (PMCx)^{Note}

Note Set the Px, PMx, and PMCx registers as follows, because the one-shot pulse output function uses the TO00 pin for timer output. Furthermore, set the Px, PMx, and PMCx registers as follows, because the TI000 pin is used for timer input to output a one-shot pulse that is synchronized with an external trigger.

• TO00 pin

	Px Register	PMx Register	PMCx Register
78K0S/KA1+ and 78K0S/KB1+ microcontrollers	P31 = 0	PM31 = 0	Setting not required
78K0S/KY1+ and 78K0S/KU1+ microcontrollers	P21 = 0	PM21 = 0	PMC21 = 0

• TI000 pin

	Px Register	PMx Register	PMCx Register
78K0S/KA1+ and 78K0S/KB1+ microcontrollers	Setting not required	PM30 = 1	Setting not required
78K0S/KY1+ and 78K0S/KU1+ microcontrollers	Setting not required	PM20 = 1	PMC20 = 0

<Example of the basic operation setting procedure when using 16-bit timer/event counter 00 as a one-shot pulse output>

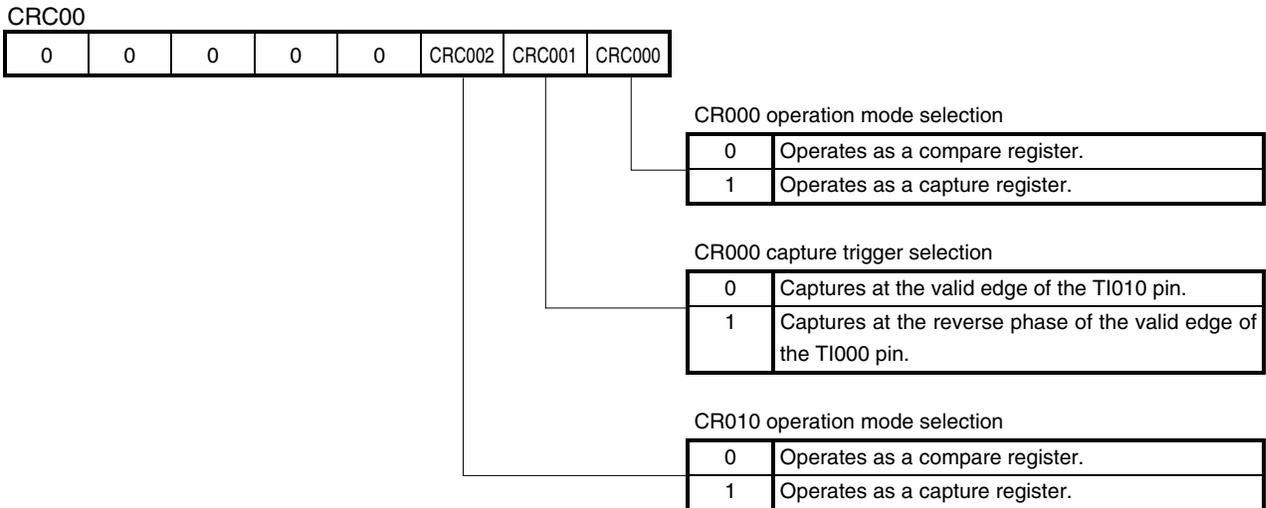
- <1> Setting the CRC00 register
- <2> Setting arbitrary values (0000H cannot be set) to the CR000 and CR010 registers
- <3> Setting the count clock using the PRM00 register
- <4> Setting the TOC00 register
- <5> Setting the TMC00 register: starting operation

Caution Steps <1> to <4> may be performed randomly.

(1) Setting the CRC00 register

This register controls the operation of the CR000 and CR010 registers.

Figure 4-1. Format of Capture/Compare Control Register 00 (CRC00)



- Cautions**
1. The timer operation must be stopped before setting the CRC00 register.
 2. Do not specify the CR000 register as a capture register when the clear & start mode has been selected upon a match between TM00 and CR000 by using the TMC00 register.

(2) Setting the CR000 register

This register has the functions of both a capture register and a compare register.

Figure 4-2. Format of 16-bit Timer Capture/Compare Register 000 (CR000)



When using CR000 as a compare register

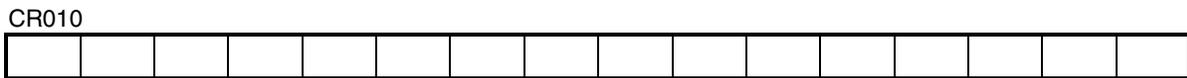
The value set to CR000 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM000) is generated if they match.

- Cautions**
1. Set a value other than 0000H to the CR000 register in the clear & start mode entered on a match between TM00 and CR000. When 0000H is set to the CR000 register in the free-running mode or the clear & start mode entered by the valid edge of the TI000 pin, an interrupt request (INTTM000) is generated when 0000H turns to 0001H, after an overflow (FFFFH) occurs.
 2. If the new value of the CR000 register is less than the value of 16-bit timer counter 00 (TM00), the TM00 register continues counting, overflows, and then starts counting from 0 again. If the new value of the CR000 register is less than the old value, therefore, the timer must be reset and restarted after the CR000 register value is changed.
 3. The value of the CR000 register after the TM00 counter has been stopped is not guaranteed.
 4. Capture operation may not be performed for the CR000 register set to the compare mode, even if a capture trigger is input.
 5. Changing the CR000 register setting during TM00 counter operation may cause a malfunction.

(3) Setting the CR010 register

This register has the functions of both a capture register and a compare register.

Figure 4-3. Format of 16-bit Timer Capture/Compare Register 010 (CR010)



When using CR010 as a compare register

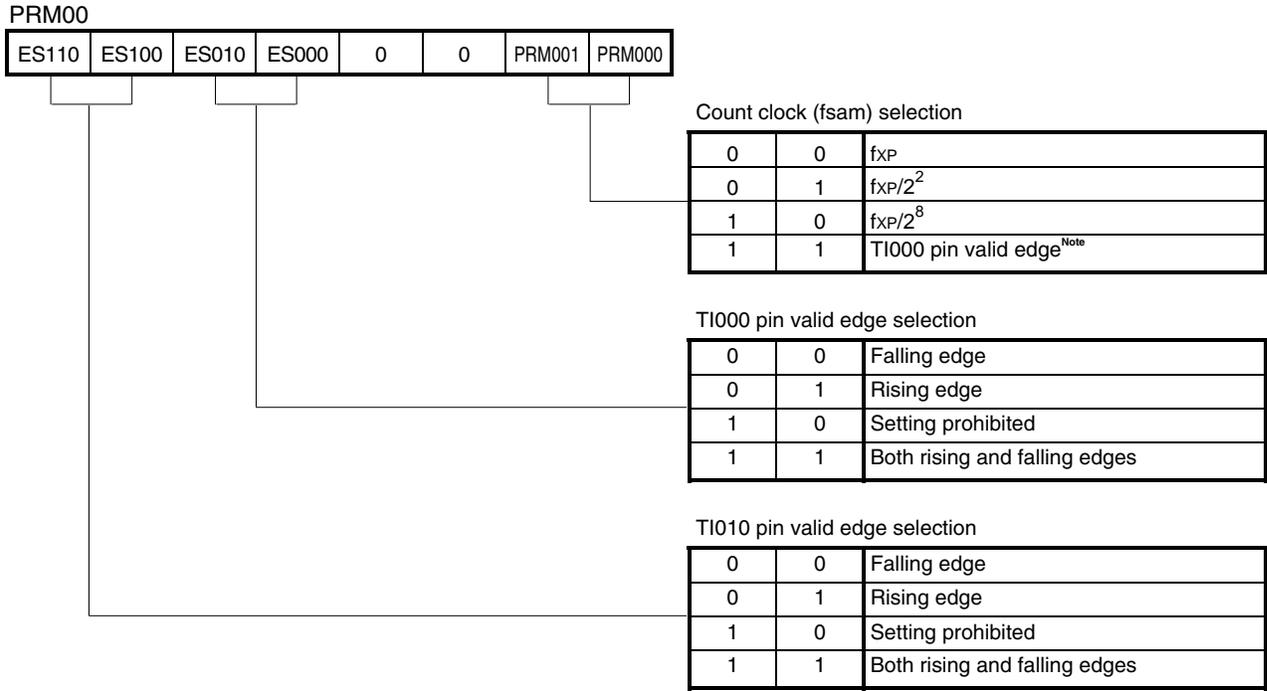
The value set to CR010 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM010) is generated if they match.

- Cautions**
1. When 0000H is set to CR010 in the free-running mode or the clear & start mode entered by the valid edge of the TI000 pin, an interrupt request (INTTM010) is generated when 0000H turns to 0001H, after an overflow (FFFFH) occurs.
 2. If the new value of the CR010 register is less than the TM00 counter value, the TM00 counter continues counting, overflows, and then starts counting from 0 again. If the new value of the CR010 register is less than the old value, therefore, the timer must be reset and restarted after the CR010 register value is changed.
 3. The value of the CR010 register after the TM00 counter has been stopped is not guaranteed.
 4. Capture operation may not be performed for the CR010 register set to the compare mode, even if a capture trigger is input.
 5. Changing the CR010 register setting during TM00 counter operation may cause a malfunction.

(4) Setting the PRM00 register

This register is used to set the count clock of the TM00 counter and the valid edges of the TI000 and TI010 pin inputs.

Figure 4-4. Format of Prescaler Mode Register 00 (PRM00)



Note The external clock requires a pulse longer than two cycles of the internal clock (f_{XP}).

Remark f_{XP} : Oscillation frequency of the clock supplied to peripheral hardware

Cautions 1. Always set data to the PRM00 register after stopping timer operation.

2. When setting the valid edge of the TI000 pin as the count clock, do not set the clear & start mode with the valid edge of the TI000 pin and the TI000 pin as the capture trigger.
3. In the following cases, note with caution that the valid edge of the TI0n0 pin ($n = 0, 1$) is detected.

- <1> A high level is input to the TI0n0 pin and the TM00 operation is enabled immediately after a system reset.
 - If the rising edge or both the rising and falling edges are specified as the valid edge of the TI0n0 pin, a rising edge is detected immediately after the TM00 operation is enabled.
- <2> The TM00 operation is stopped while the TI0n0 pin is at high level and it is then enabled after a low level is input to the TI0n0 pin.
 - If the falling edge or both the rising and falling edges are specified as the valid edge of the TI0n0 pin, a falling edge is detected immediately after the TM00 operation is enabled.
- <3> The TM00 operation is stopped while the TI0n0 pin is at low level and it is then enabled after a high level is input to the TI0n0 pin.
 - If the rising edge or both the rising and falling edges are specified as the valid edge of the TI0n0 pin, a rising edge is detected immediately after the TM00 operation is enabled.

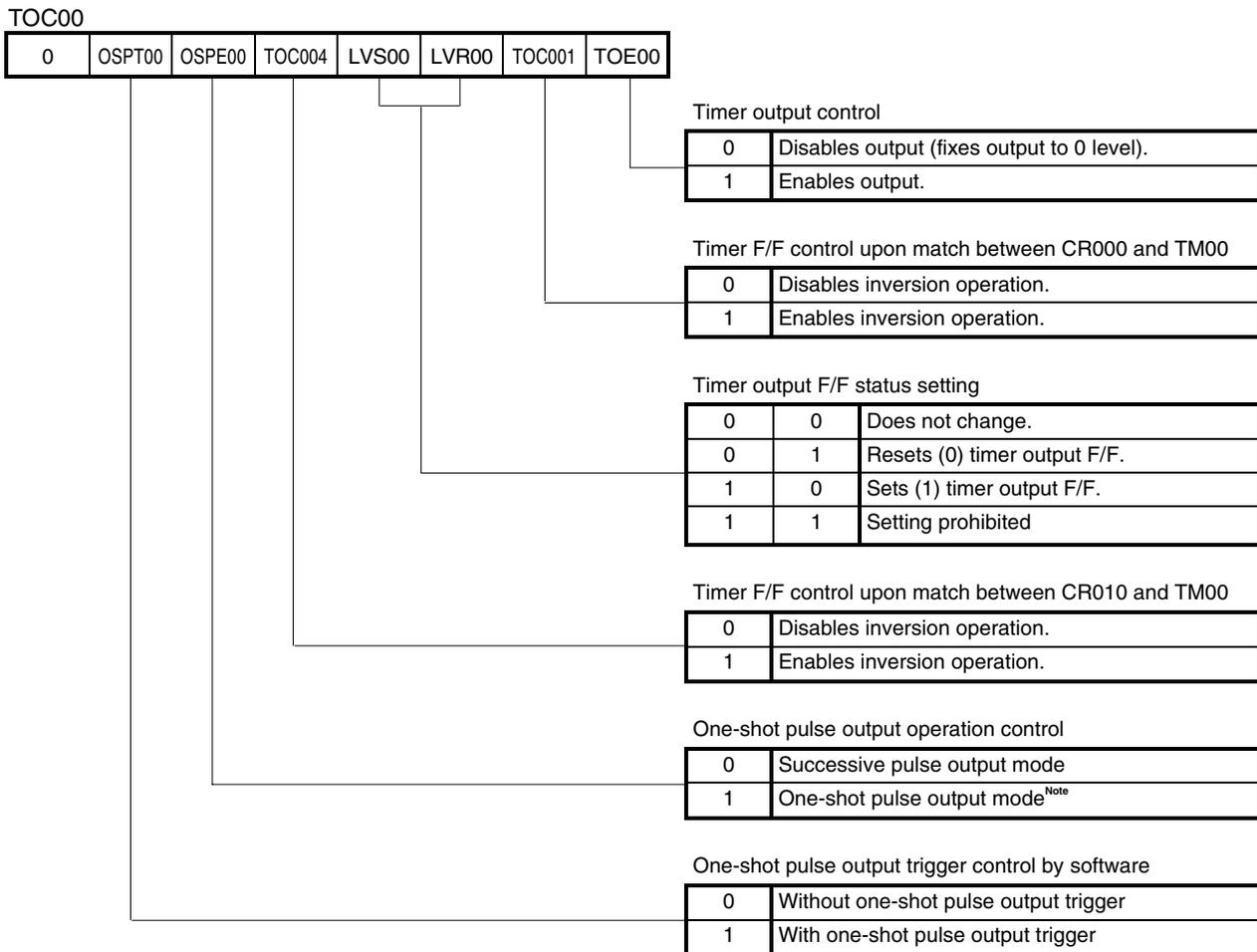
Cautions 4. To use the valid edge of TI000 with the count clock, it is sampled with f_{XP} to eliminate noise. The capture operation is not performed until the valid edge is sampled and the valid level is detected twice, thus eliminating noise with a short pulse width.

- 5. When the TI010/TO00/Pxx pin is used as the input pin (TI010) of the valid edge, it cannot be used as a timer output pin (TO00). When it is used as a timer output pin (TO00), it cannot be used as the input pin (TI010) of the valid edge.

(5) Setting the TOC00 register

This register controls the operation of the 16-bit timer/event counter 00 output controller. It is used to set/reset the timer output F/F, enable or disable output inversion, timer output (TO00 pin output), and one-shot pulse output operation, and set the one-shot pulse output trigger by software.

Figure 4-5. Format of 16-bit Timer Output Control Register 00 (TOC00)



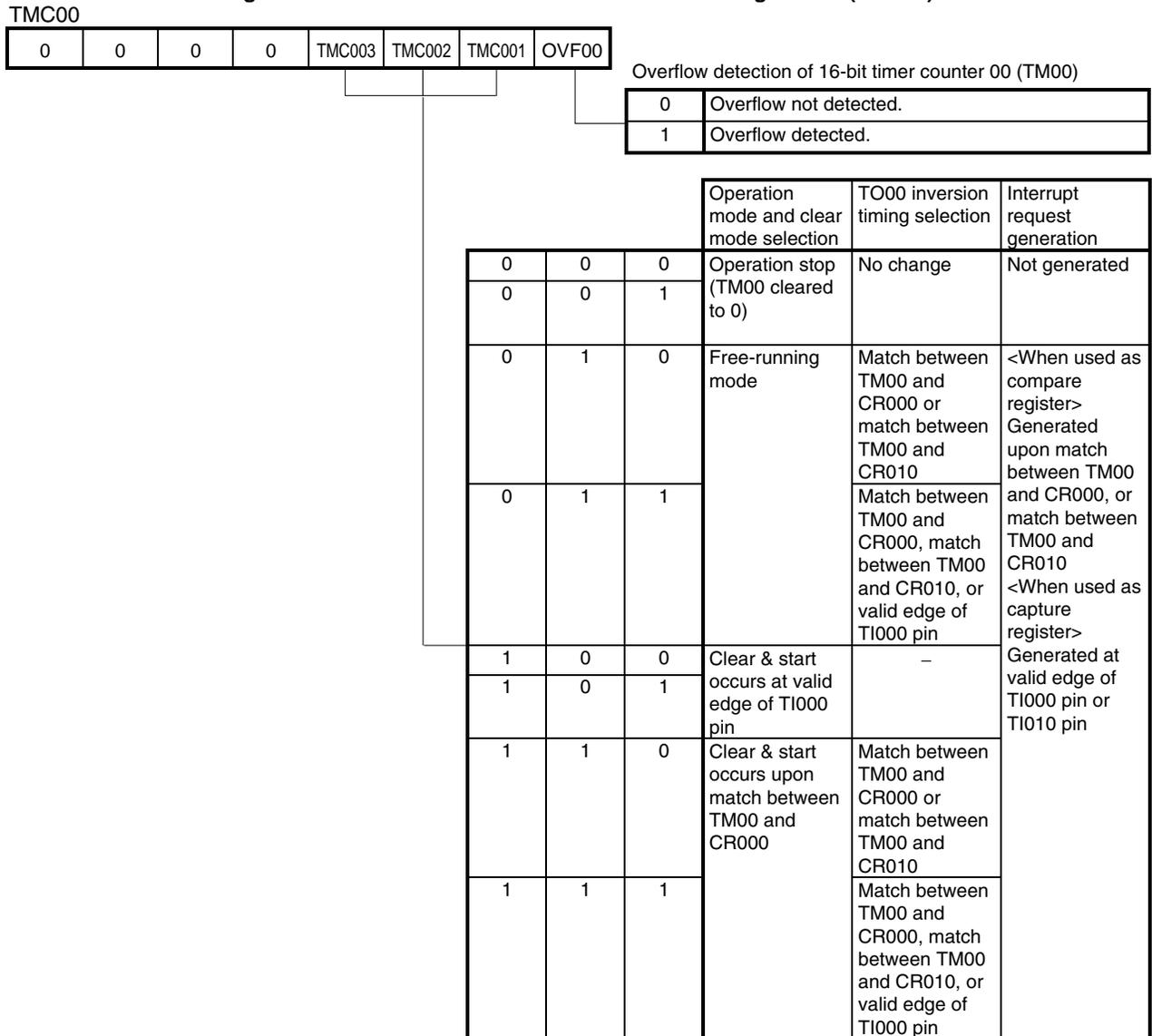
Note The one-shot pulse output mode operates normally only in the free-running mode and the clear & start mode set with the valid edge of the TI000 pin. In the clear & start mode set upon a match between TM00 and CR000, one-shot pulse output is not possible, because an overflow does not occur.

1. The timer operation must be stopped before setting bits other than OSPT00.
2. If LVS00 and LVR00 are read, 0 is read.
3. OSPT00 is automatically cleared after data is set, so 0 is read.
4. Do not set OSPT00 to 1 other than in one-shot pulse output mode.
5. A write interval of at least two cycles of the count clock that has been selected by using the PRM00 register is required to set (1) successively to OSPT00.
6. When TOE00 is 0, set TOE00, LVS00, and LVR00 at the same time with the 8-bit memory manipulation instruction. When TOE00 is 1, LVS00 and LVR00 can be set with the 1-bit memory manipulation instruction.
7. When the TI010/TO00/Pxx pin is used as the input pin (TI010) of the valid edge, it cannot be used as a timer output pin (TO00). When it is used as a timer output pin (TO00), it cannot be used as the input pin (TI010) of the valid edge.

(6) Setting the TMC00 register

This register sets the 16-bit timer/event counter 00 operation mode, TM00 counter clear mode, and output timing, and detects overflows.

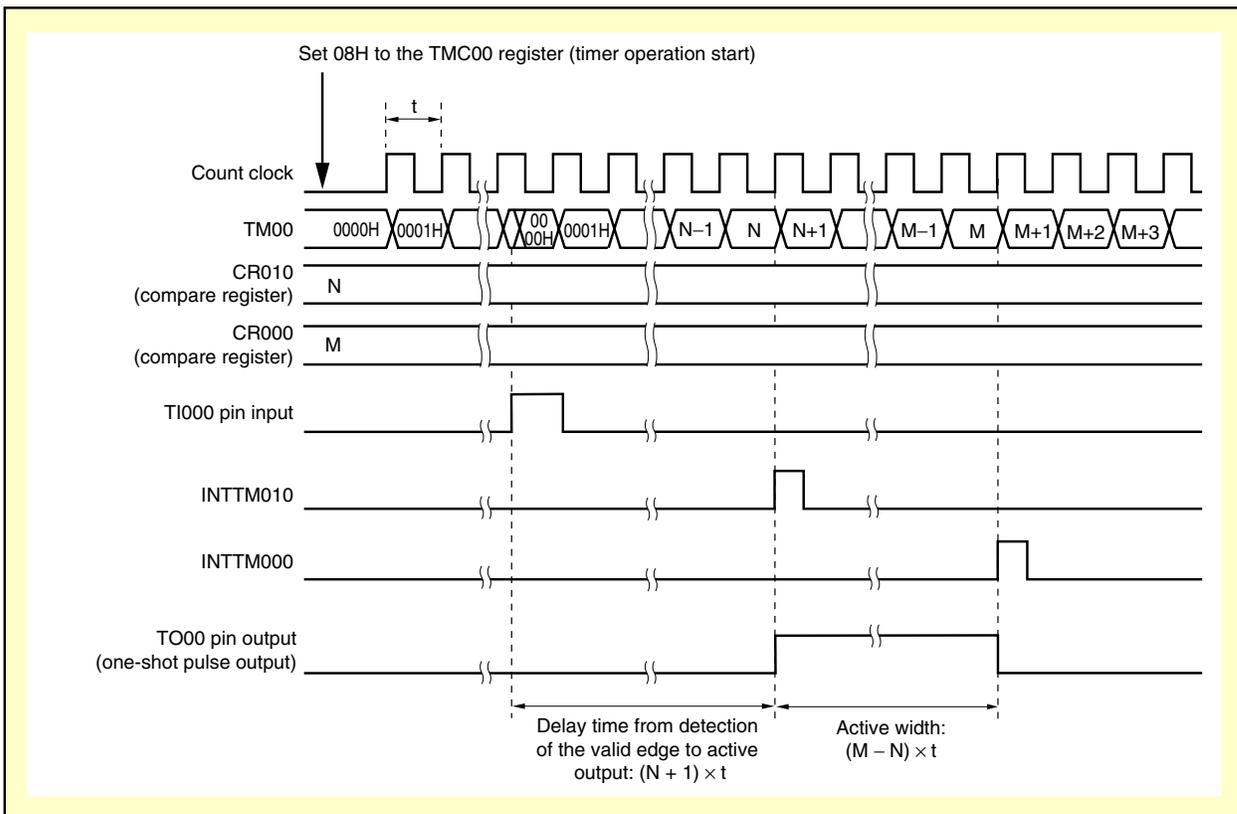
Figure 4-6. Format of 16-bit Timer Mode Control Register 00 (TMC00)



- Cautions**
1. The operation of the TM00 counter starts when values other than 0 and 0 (operation stop mode) are set to TMC002 and TMC003, respectively. To stop the operation, set TMC002 and TMC003 to 0 and 0, respectively.
 2. Write to the bits other than the OVF00 flag after stopping the timer operation.
 3. When the timer is stopped, timer counts and timer interrupts do not occur, even if a signal is input to the TI000/TI010 pin.
 4. Except when the valid edge of the TI000 pin is selected as the count clock, stop the timer operation before setting to the STOP mode or system clock stop mode; otherwise the timer may malfunction when the system clock starts.
 5. Set the valid edge of the TI000 pin with bits 4 and 5 of the PRM00 register after stopping the timer operation.
 6. If the clear & start mode is set upon a match between TM00 and CR000 or at the valid edge of the TI000 pin, or the free-running mode is selected, when the set value of the CR000 register is FFFFH and the TM00 counter value changes from FFFFH to 0000H, the OVF00 flag is set to 1.
 7. Even if the OVF00 flag is cleared before the next count clock is counted (before the TM00 counter becomes 0001H) after the TM00 counter overflows, it is re-set and clearing is disabled.
 8. Capture operation is performed at the fall of the count clock. An interrupt request (INTTM0n0: n = 0, 1), however, occurs at the rise of the next count clock.

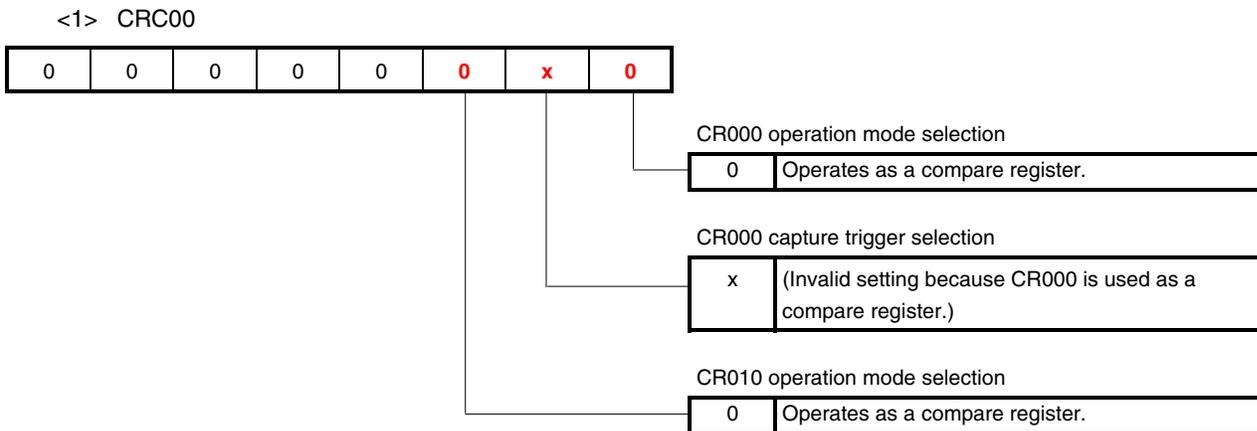
[Example 1] When performing one-shot pulse output by an external trigger
 (Count clock: $f_{XP}/2^2$ ($f_{XP} = 2$ MHz), valid edge of the TI000 pin: rising edge, initial TO00 output value: low level, delay time from detection of the valid edge to active output: 1 ms, active width: 10 ms)
 (Same contents as in this sample program source)

Figure 4-7. Timing Example of One-Shot Pulse Output by an External Trigger (When the Rising Edge Is Specified)



Remark $N < M$

(1) Register settings



<2> CR010

Setting value (N): 1999

- Count clock fsam = $8 \text{ [MHz]} / 2^2 = 2 \text{ [MHz]}$
- Delay time 1 [ms] = $1000 \text{ [}\mu\text{s]} = (N + 1) / 2 \text{ [MHz]}$
- $N = 1000 \text{ [}\mu\text{s]} \times 2 \text{ [MHz]} - 1 = 1999$

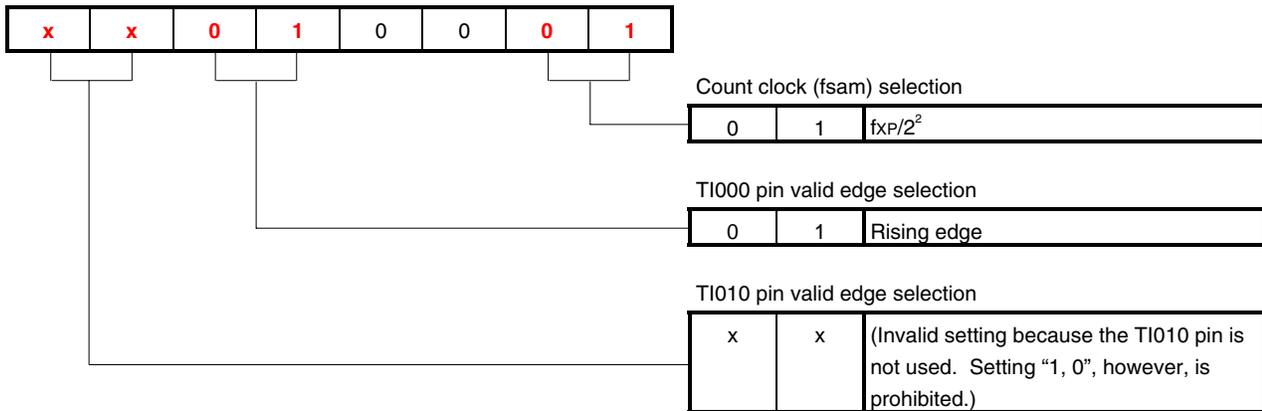
<3> CR000

Setting value (M): 21999

- Count clock fsam = $8 \text{ [MHz]} / 2^2 = 2 \text{ [MHz]}$
- Active width: $10 \text{ [ms]} = 10000 \text{ [}\mu\text{s]} = (M - N) / 2 \text{ [MHz]}$
- $M = 10000 \text{ [}\mu\text{s]} \times 2 \text{ [MHz]} + 1999 = 20000 + 1999 = 21999$

Caution The setting values of <2> and <3>, mentioned above, are examples of when N is smaller than M. When N is larger than M, the CR010 register is used to set the active width and the CR000 register is used to set the delay time. Do not set such that N equals M. Furthermore, do not set 0000H to the CR000 and CR010 registers.

<4> PRM00



<5> TOC00



Timer output control
 1 Enables output.

Timer F/F control upon match between CR000 and TM00
 1 Enables inversion operation.

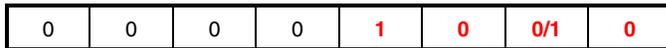
Timer output F/F status setting
 0 1 Resets (0) timer output F/F.

Timer F/F control upon match between CR010 and TM00
 1 Enables inversion operation.

One-shot pulse output operation control
 1 One-shot pulse output mode

One-shot pulse output trigger control by software
 0 Without one-shot pulse output trigger

<6> TMC00



Overflow detection of 16-bit timer counter 00 (TM00)
 0 Overflow not detected.

Operation mode and clear mode selection	TO00 inversion timing selection	Interrupt request generation
1 0 0	–	Generated upon match between TM00 and CR000, or match between TM00 and CR010
1 0 1		

<7> Px, PMx, PMCx

	Px Register	PMx Register	PMCx Register
78K0S/KA1+ and 78K0S/KB1+ microcontrollers	P31 = 0	PM30 = 1, PM31 = 0	Setting not required
78K0S/KY1+ and 78K0S/KU1+ microcontrollers	P21 = 0	PM20 = 1, PM21 = 0	PMC20 = 0, PMC21 = 0

(2) Sample program

In the example below, “x” in **(1) Register settings** is set to “0”.

<1> Assembly language (when using the 78K0S/KA1+ and 78K0S/KB1+ microcontrollers)

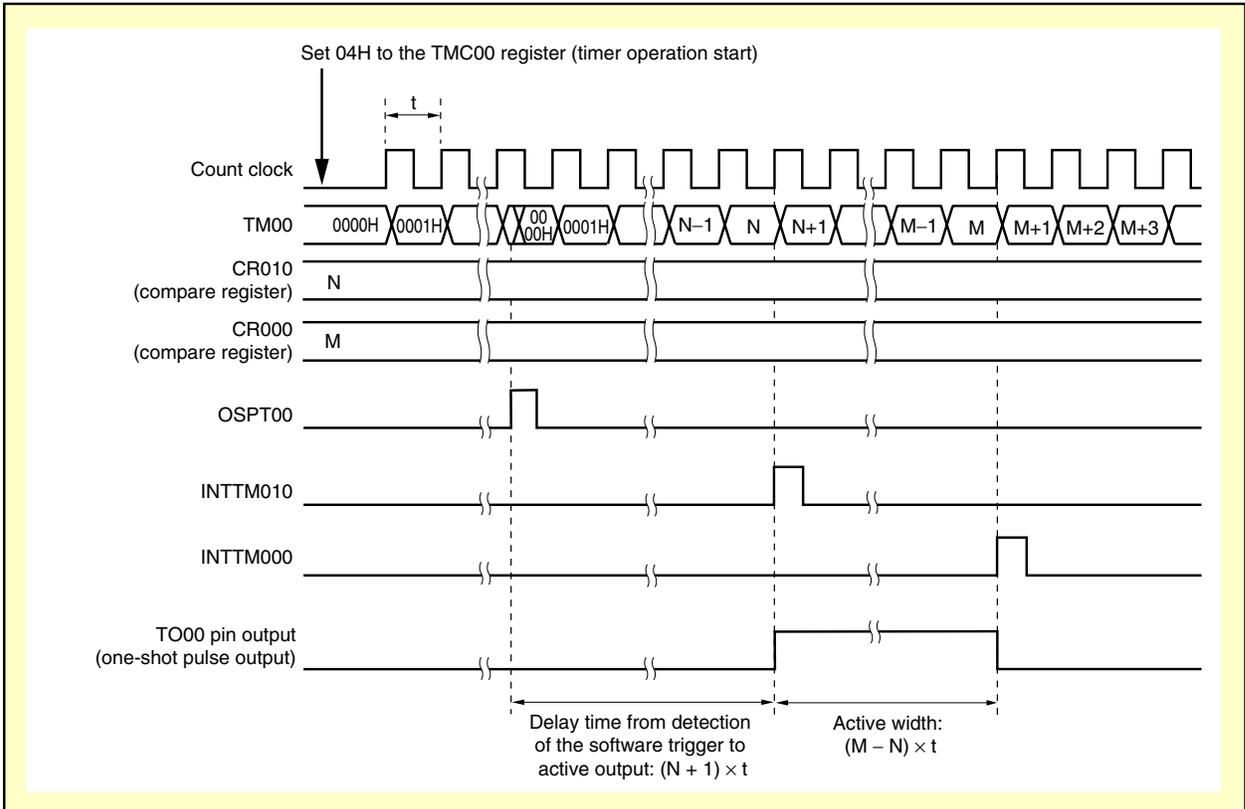
```
CLR1  P3.1
SET1  PM3.0
CLR1  PM3.1
MOV   CRC00, #00000000B
MOVW  CR010, #1999
MOVW  CR000, #21999
MOV   PRM00, #00010001B
MOV   TOC00, #00110111B
MOV   TMC00, #00001000B
```

<2> C language (when using the 78K0S/KA1+ and 78K0S/KB1+ microcontrollers)

```
P3.1 = 0;
PM3.0 = 1;
PM3.1 = 0;
CRC00 = 0b00000000;
CR010 = 1999;
CR000 = 21999;
PRM00 = 0b00010001;
TOC00 = 0b00110111;
TMC00 = 0b00001000;
```

[Example 2] When performing one-shot pulse output by a software trigger
 (Count clock: $f_{XP}/2^2$ ($f_{XP} = 2$ MHz), initial TO00 output value: low level, delay time from detection of the software trigger to active output: 1 ms, active width: 10 ms)

Figure 4-8. Timing Example of One-Shot Pulse Output by a Software Trigger



Remark $N < M$

(1) Register settings

<1> CRC00

0	0	0	0	0	0	0	x	0
---	---	---	---	---	---	---	---	---

CR000 operation mode selection
 0 Operates as a compare register.

CR000 capture trigger selection
 x (Invalid setting because CR000 is used as a compare register.)

CR010 operation mode selection
 0 Operates as a compare register.

<2> CR010

Setting value (N): 1999

- Count clock fsam = $8 \text{ [MHz]} / 2^2 = 2 \text{ [MHz]}$
- Delay time 1 [ms] = $1000 \text{ [}\mu\text{s]} = (N + 1) / 2 \text{ [MHz]}$
- $N = 1000 \text{ [}\mu\text{s]} \times 2 \text{ [MHz]} - 1 = 1999$

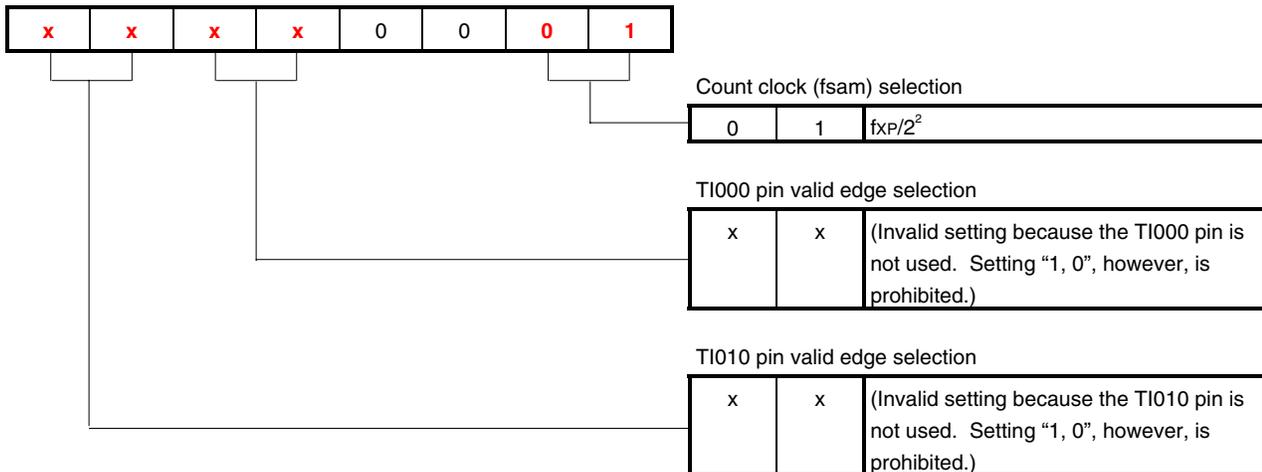
<3> CR000

Setting value (M): 21999

- Count clock fsam = $8 \text{ [MHz]} / 2^2 = 2 \text{ [MHz]}$
- Active width: $10 \text{ [ms]} = 10000 \text{ [}\mu\text{s]} = (M - N) / 2 \text{ [MHz]}$
- $M = 10000 \text{ [}\mu\text{s]} \times 2 \text{ [MHz]} + 1999 = 20000 + 1999 = 21999$

Caution The setting values of <2> and <3>, mentioned above, are examples of when N is smaller than M. When N is larger than M, the CR010 register is used to set the active width and the CR000 register is used to set the delay time. Do not set such that N equals M. Furthermore, do not set 0000H to the CR000 and CR010 registers.

<4> PRM00



<5> TOC00



Timer output control
 1 Enables output.

Timer F/F control upon match between CR000 and TM00
 1 Enables inversion operation.

Timer output F/F status setting
 0 1 Resets (0) timer output F/F.

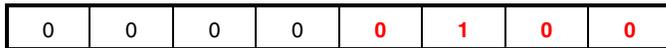
Timer F/F control upon match between CR010 and TM00
 1 Enables inversion operation.

One-shot pulse output operation control
 1 One-shot pulse output mode

One-shot pulse output trigger control by software
 0 Without one-shot pulse output trigger

* To perform one-shot pulse output, set 1 to OSPT00.

<6> TMC00



Overflow detection of 16-bit timer counter 00 (TM00)
 0 Overflow not detected.

Operation mode and clear mode selection	TO00 inversion timing selection	Interrupt request generation
0 1 0	Free-running mode	Match between TM00 and CR000 or match between TM00 and CR010
		Generated upon match between TM00 and CR000, or match between TM00 and CR010

<7> Px, PMx, PMCx

	Px Register	PMx Register	PMCx Register
78K0S/KA1+ and 78K0S/KB1+ microcontrollers	P31 = 0	PM31 = 0	Setting not required
78K0S/KY1+ and 78K0S/KU1+ microcontrollers	P21 = 0	PM21 = 0	PMC21 = 0

(2) Sample program

In the example below, “x” in **(1) Register settings** is set to “0”.

<1> Assembly language (when using the 78K0S/KA1+ and 78K0S/KB1+ microcontrollers)

```
CLR1  P3.1
CLR1  PM3.1
MOV   CRC00, #00000000B
MOVW  CR010, #1999
MOVW  CR000, #21999
MOV   PRM00, #00000001B
MOV   TOC00, #00110111B
MOV   TMC00, #00000100B
SET1  OSPT00
```

<2> C language (when using the 78K0S/KA1+ and 78K0S/KB1+ microcontrollers)

```
P3.1 = 0;
PM3.1 = 0;
CRC00 = 0b00000000;
CR010 = 1999;
CR000 = 21999;
PRM00 = 0b00000001;
TOC00 = 0b00110111;
TMC00 = 0b00000100;
OSPT00 = 1;
```

[Excerpt from this sample program source]

An excerpt from [APPENDIX A PROGRAM LIST](#), which is related to the 16-bit timer/event counter 00 function, is shown below (same contents as in [\[Example 1\]](#) mentioned above).

(1) Assembly language

```

XMAIN CSEG UNIT
RESET_START:
    .
    .
    .
    MOV CR000, #00000000B ; Use CR000 and CR010 as compare registers
    MOVW AX, #2000-1 ; Set the delay time to 1 ms
    MOVW CR010, AX ; Initialize compare value CR010
    ADDW AX, #20000 ; Set the active width to 10 ms
    MOVW CR000, AX ; Initialize compare value CR000
    MOV PRM00, #00010001B ; Count clock = fxp/4 = 2 MHz, valid edge of the
    TI000 pin = rising edge
    MOV TOC00, #00110111B ; One-shot pulse output mode, output reversal upon
    a match between CR000 or CR010 and TM00,
    ; default 0 output, timer output enable
    MOV TMC00, #00001000B ; Start the timer operation (clear & start upon
    detection of the valid edge of the TI000 pin)
    .
    .
    .
    INTERRUPT_P1:
    .
    .
    .
    MOV TMC00, #00000000B ; Stop the timer operation
    MOVW AX, CR000 ; Read CR000
    ADDW AX, #20000 ; Increment the active width by 10 ms
    BNC $CHANGE_CR ; Branch if no overflow occurs
    MOVW AX, #22000-1 ; Initialize the active width to 10 ms
    CHANGE_CR:
    MOVW CR000, AX ; Write to CR000
    SET1 LVR00 ; Set to default 0 output
    MOV TMC00, #00001000B ; Start the timer operation (clear & start upon
    detection of the valid edge of the TI000 pin)
    .
    .
    .
  
```

Annotations:

- Setting the operation modes of CR000 and CR010 as compare registers (points to **MOV CR000, #00000000B**)
- Initializing compare value CR010 (points to **MOVW CR010, AX**)
- Initializing compare value CR000 (points to **MOVW CR000, AX**)
- Setting the count clock and valid edge (points to **MOV PRM00, #00010001B**)
- Setting the one-shot pulse output (points to **MOV TOC00, #00110111B**)
- Starting timer operation (points to **MOV TMC00, #00001000B**)
- Setting the CR0x0 register after stopping timer operation (points to **MOVW CR000, AX**)
- Setting the initial TO00 output value (points to **SET1 LVR00**)
- Starting timer operation (points to **MOV TMC00, #00001000B**)

(2) C language

```

void hdwinit(void){
    unsigned char ucCnt200us; /* 8-bit variable for 200 us wait */
    •
    •
    •
    CR000 = 0b00000000; /* Use CR000 and CR010 as compare registers */
    CR010 = 2000 - 1; /* Initialize the delay time to 1 ms */
    CR000 = CR010 + 20000; /* Initialize the active width to 10 ms */
    PRM00 = 0b00010001; /* Count clock = fxp/4 = 2 MHz, valid edge of the
TI000 pin = rising edge */
    TOC00 = 0b00110111; /* One-shot pulse output mode, output reversal upon a
match between CR000 or CR010 and TM00, default 0 output, timer output enable */
    TMC00 = 0b00001000; /* Start the timer operation (clear & start upon
detection of the valid edge of the TI000 pin) */
    •
    •
    •
    __interrupt void fn_intp1(){
        •
        •
        •
        TMC00 = 0b00000000; /* Stop the timer operation */
        CR000 = CR000 + 20000; /* Increment the active width by 10 ms */
        if (__getcy()){ /* Processing when an overflow occurs */
            CR000 = 22000 - 1; /* Initialize the active width to 10 ms */
        }
        LVR00 = 1; /* Set to default 0 output */
        TMC00 = 0b00001000; /* Start the timer operation (clear & start
upon detection of the valid edge of the TI000 pin) */
    }
    return;
}

```

Initializing compare value CR000
 Initializing compare value CR010
 Setting the operation modes of CR000 and CR010 as compare registers
 Setting the count clock and valid edge
 Setting the one-shot pulse output
 Starting timer operation
 Setting the CR0x0 register after stopping timer operation
 Setting the initial TO00 output value
 Starting timer operation

4.2 Setting the One-Shot Pulse Active Width

In this sample program, the one-shot pulse output function of 16-bit timer/event counter 00 is used to detect the rising edge of the external signal that is input to the TI000 pin, and then output a one-shot pulse after a given delay time elapses.

- Delay time from detection of the valid edge to active output = $(N + 1)/f_{sam}$
- Active output width = $(M - N)/f_{sam}$

Remark N: CR010 register setting value
M: CR000 register setting value
 f_{sam} : Count clock frequency of 16-bit timer/event counter 00
 $N < M$

Calculation example: When the CR010 register setting value is 1,999 and the CR000 register setting value is 21,999 (operation when $f_{sam} = 2 \text{ MHz} = 2,000 \text{ kHz}$)

- Delay time = $(N + 1)/f_{sam} = (1,999 + 1)/2,000 \text{ [kHz]} = 1 \text{ [ms]}$
- Active width = $(M - N)/f_{sam} = (21,999 - 1,999)/2,000 \text{ [kHz]} = 10 \text{ [ms]}$

Furthermore, the CR000 register setting value and active pulse width are changed according to the number of switch inputs.

Number of Switch Inputs ^{Note}	CR000 Register Setting Value	CR010 Register Setting Value	One-Shot Pulse Output Active Width
0	21,999	1,999	10 ms ((21,999 - 1,999)/2,000 kHz)
1	41,999		20 ms ((41,999 - 1,999)/2,000 kHz)
2	61,999		30 ms ((61,999 - 1,999)/2,000 kHz)

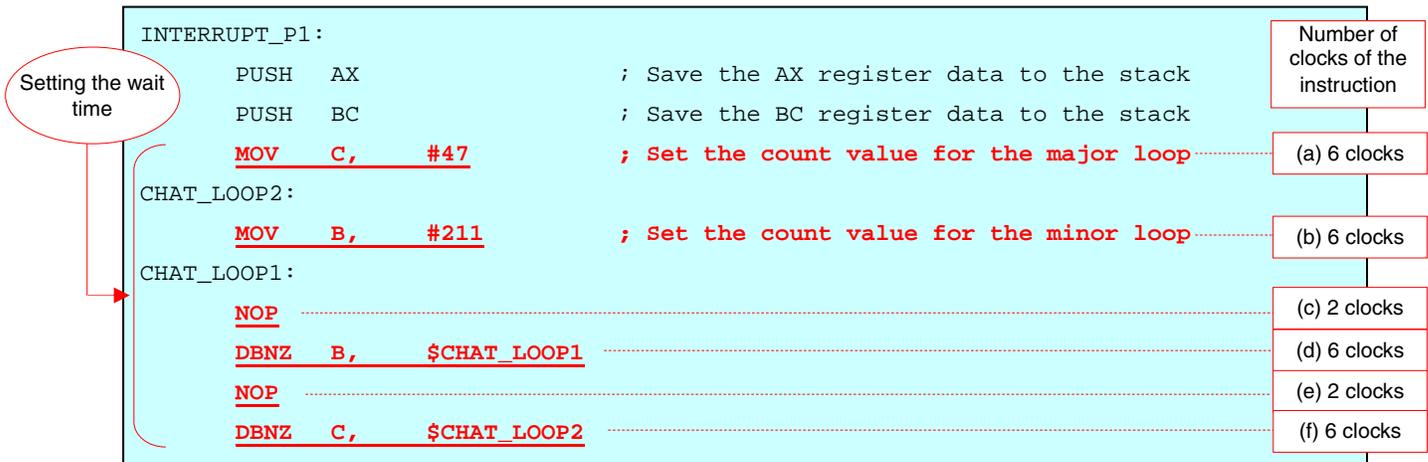
Note The active pulse widths from the zeroth switch input are repeated after the third switch input.

4.3 Setting the Chattering Detection Time

In this sample program, chattering up to 10 ms is eliminated by setting the wait time to 10 ms, in order to handle chattering during switch input (INTP1 interrupt generation).

[Excerpt from this sample program source]

(1) Assembly language

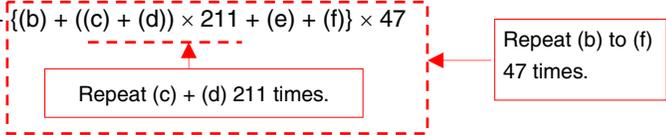


The wait time is set to 10 ms through the above-mentioned settings.

The wait time can be calculated by using the following expression.

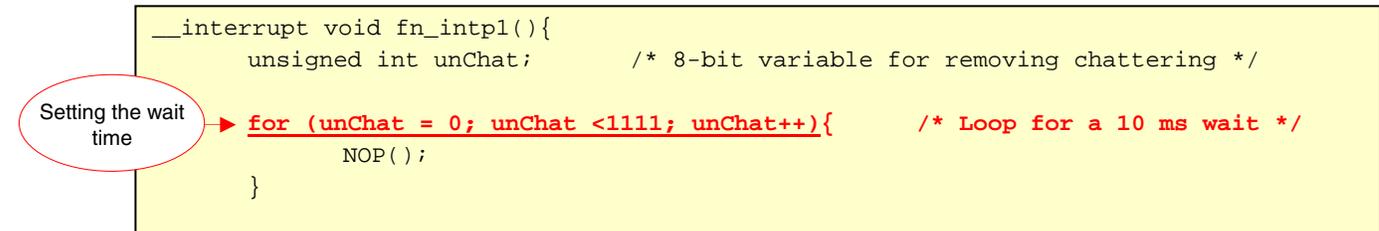
- One CPU clock = $1/8$ [MHz] = 0.125 [μ s]

- Number of clocks = (a) + {(b) + ((c) + (d)) \times 211 + (e) + (f)} \times 47



- Wait time = [(a) + {(b) + ((c) + (d)) \times 211 + (e) + (f)} \times 47] \times 0.125 [μ s]
 $= [6 + \{6 + (2 + 6) \times 211 + 2 + 6\} \times 47] \times 0.125$ [μ s]
 $= 80000 \times 0.125$ [μ s] = 10000 [μ s] = 10 [ms]

(2) C language



The wait time is set to about 10 ms through the above-mentioned "for" loop statement.

CHAPTER 5 OPERATION CHECK USING SYSTEM SIMULATOR SM+

This chapter describes how the sample program operates with system simulator SM+ for 78K0S/Kx1+, by using the assembly language file (source files + project file) that has been downloaded by selecting the  icon.

<R> **Caution** System simulator SM+ for 78K0S/Kx1+ is not supported with the 78K0S/KU1+ microcontroller (as of July 2008). The operation of the 78K0S/KU1+ microcontroller, therefore, cannot be checked by using system simulator SM+ for 78K0S/Kx1+.

<R> 5.1 Building the Sample Program

To check the operation of the sample program by using system simulator SM+ for 78K0S/Kx1+ (hereinafter referred to as “SM+”), SM+ must be started after building the sample program. This section describes how to build a sample program by using the assembly language sample program (source program + project file) downloaded by clicking the  icon. See the [78K0S/Kx1+ Sample Program Startup Guide Application Note](#) for how to build other downloaded programs.

For the details of how to operate PM+, refer to the [PM+ Project Manager User's Manual](#).



[Column] Build errors

Change the compiler option setting according to the following procedure when the error message “A006 File not found ‘C:\NECTOOLS32\LIB78K0S\s0sl.rel’” or “*** ERROR F206 Segment ‘@@DATA’ can’t allocate to memory - ignored.” is displayed, when building with PM+.

<1> Select [Compiler Options] from the [Tool] menu.

<2> The [Compiler Options] dialog box will be displayed. Select the [Startup Routine] tab.

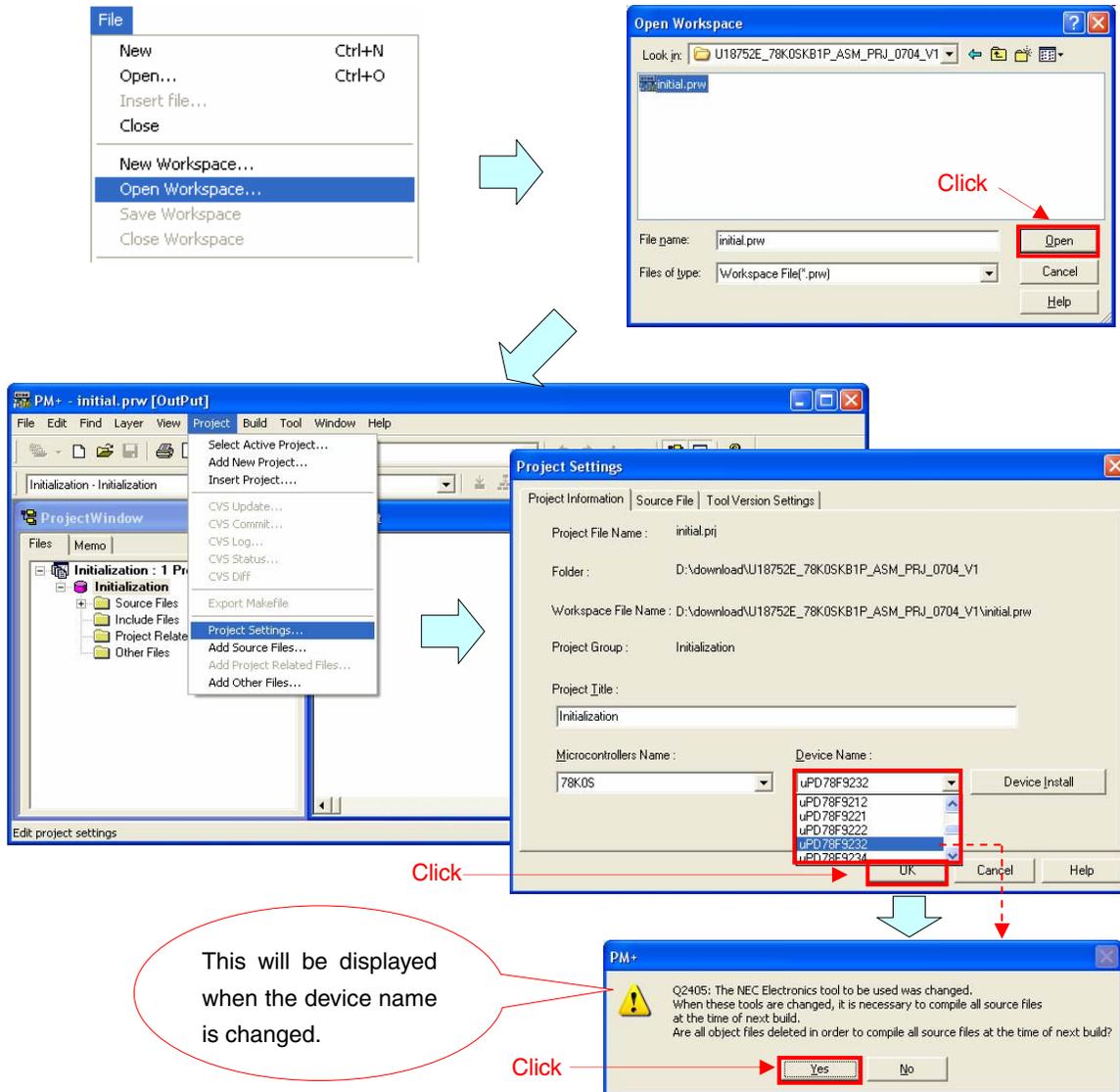
<3> Uncheck the [Using Fixed Area of Standard Library] check box. (Leave the other check boxes as they are.)

A RAM area of 118 bytes that has been secured as a fixed standard library area will be enabled for use when the [Using Fixed Area of Standard Library] check box is unchecked; however, the standard libraries (such as the getchar function and malloc function) will be disabled for use.

The [Using Fixed Area of Standard Library] check box is unchecked by default when the file that has been downloaded by clicking the  icon is used in this sample program.

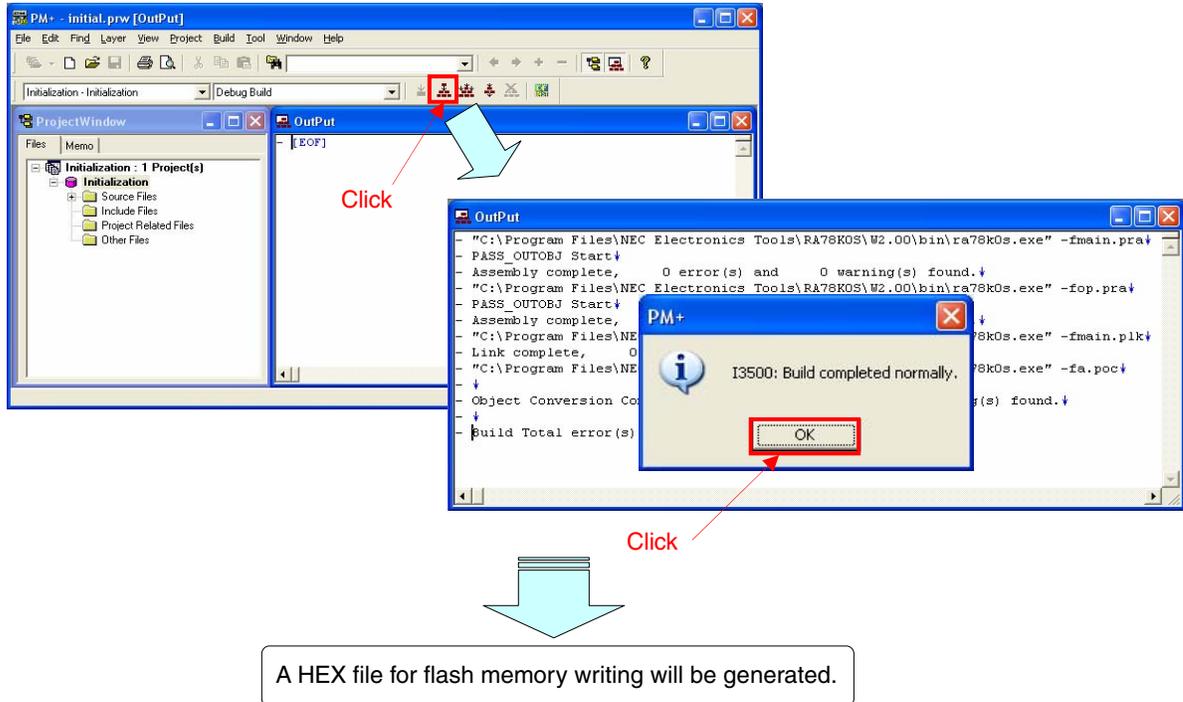
- (1) Start PM+.
- (2) Select "tm00one.prw" by clicking [Open Workspace] from the [File] menu and click [Open]. A workspace into which the source file will be automatically read will be created.
- (3) Select [Project Settings] from the [Project] menu. When the [Project Settings] window opens, select the name of the device to be used (the device with the largest ROM or RAM size will be selected by default), and click [OK].

Remark Screenshots of the Sample Program (Initial Settings) LED Lighting Switch Control are shown below.



- (4) Click  ([Build] button). When the source files are built normally, the message "I3500: Build completed normally." will be displayed.
- (5) Click the [OK] button in the message dialog box. A HEX file for flash memory writing will be created.

Remark Screenshots of the Sample Program (Initial Settings) LED Lighting Switch Control are shown below.

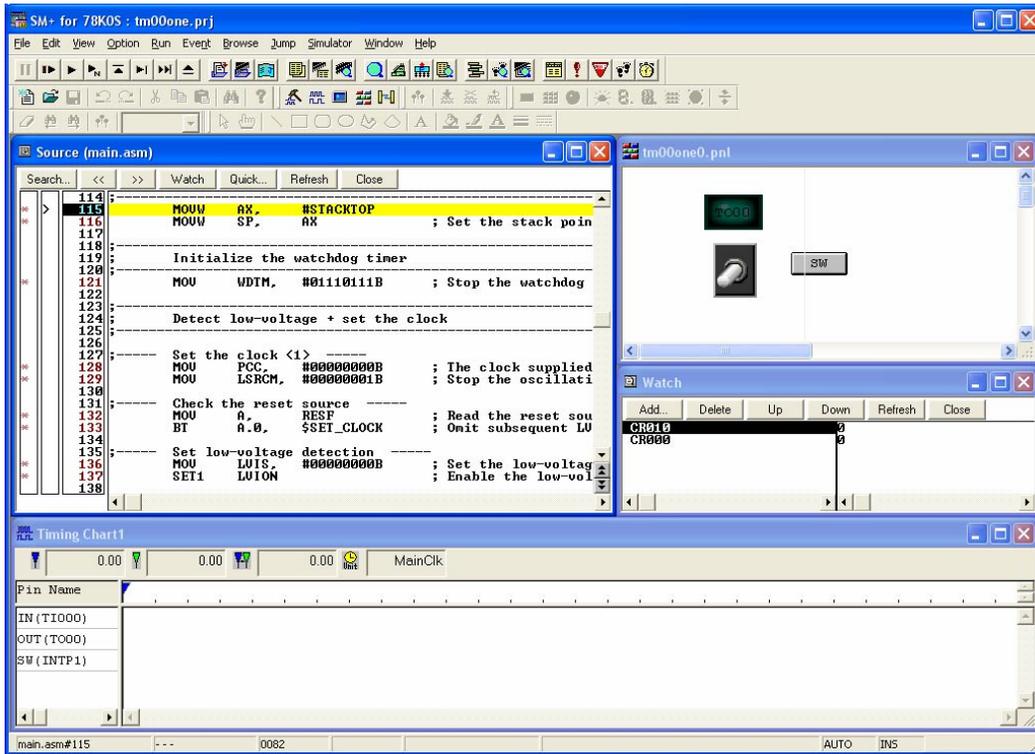


5.2 Operation with SM+

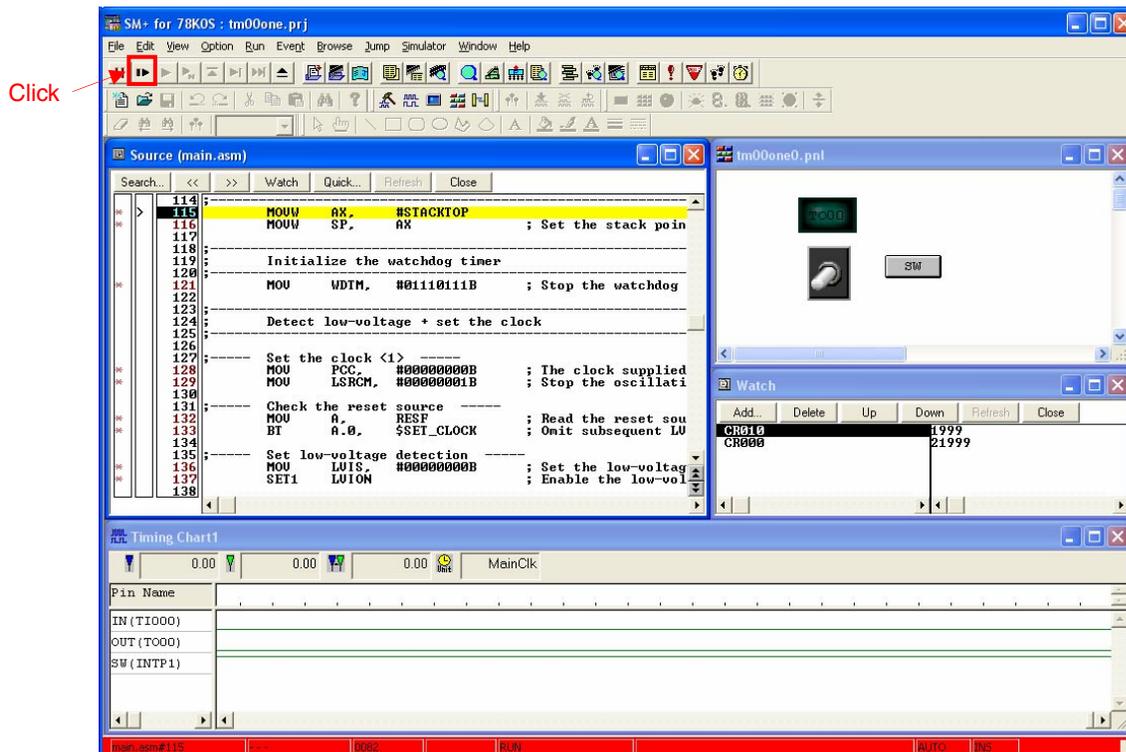
This section describes examples of checking the operation on the I/O panel window or timing chart window of SM+. For the details of how to operate SM+, refer to the [SM+ System Simulator Operation User's Manual](#).

- <R>
- (1) When SM+ for 78K0S/Kx1+ W1.02 ("SM+" hereafter) is used in the environment of PM+ Ver. 6.30, SM+ cannot be selected as the debugger. In this case, start SM+ via method (a) or (b) described below, while keeping PM+ running after completing building a project.
 - (a) When starting SM+ in PM+
 - <1> Select [Register Ex-tool] from the [Tool] menu and register "SM+ for 78K0S/Kx1+".
 - <2> Select [Ex-tool Bar] from the [View] menu and add the SM+ icon to the PM+ toolbar.
 - <3> Click the SM+ icon and start SM+.
 (See the PM+ help for details on how to register external tools.)
 - (b) When not starting SM+ in PM+
 - Start SM+ from the Windows start menu.

- (2) The following screen will be displayed when SM+ is started. (This is a sample screenshot of when an assembly language source file downloaded by clicking the  icon was used.)



- (3) Click  ([Restart] button). The program will be executed after the CPU is reset and the following screen will be displayed.



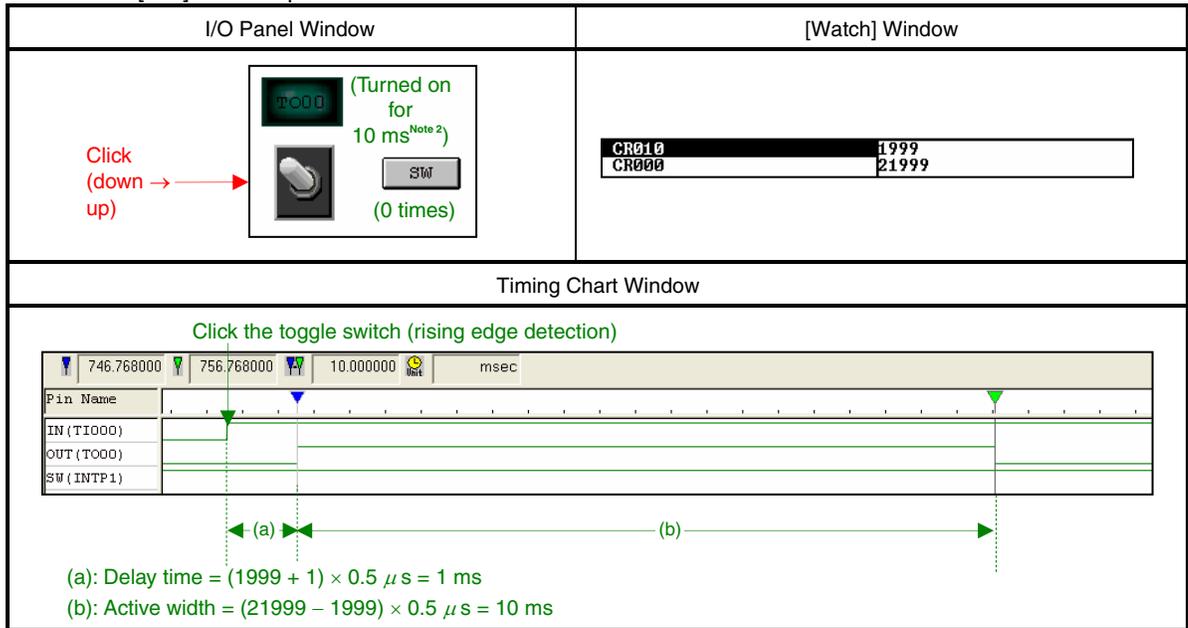
This turns red during program execution.

- (4) The toggle switch in the I/O panel window is used as an external pulse input of the TI000 pin. When the toggle switch is clicked during program execution, a one-shot pulse is output, and the [LED] is turned on when it is active.

Furthermore, the active width of the one-shot pulse output is changed according to the number of [SW] button inputs.

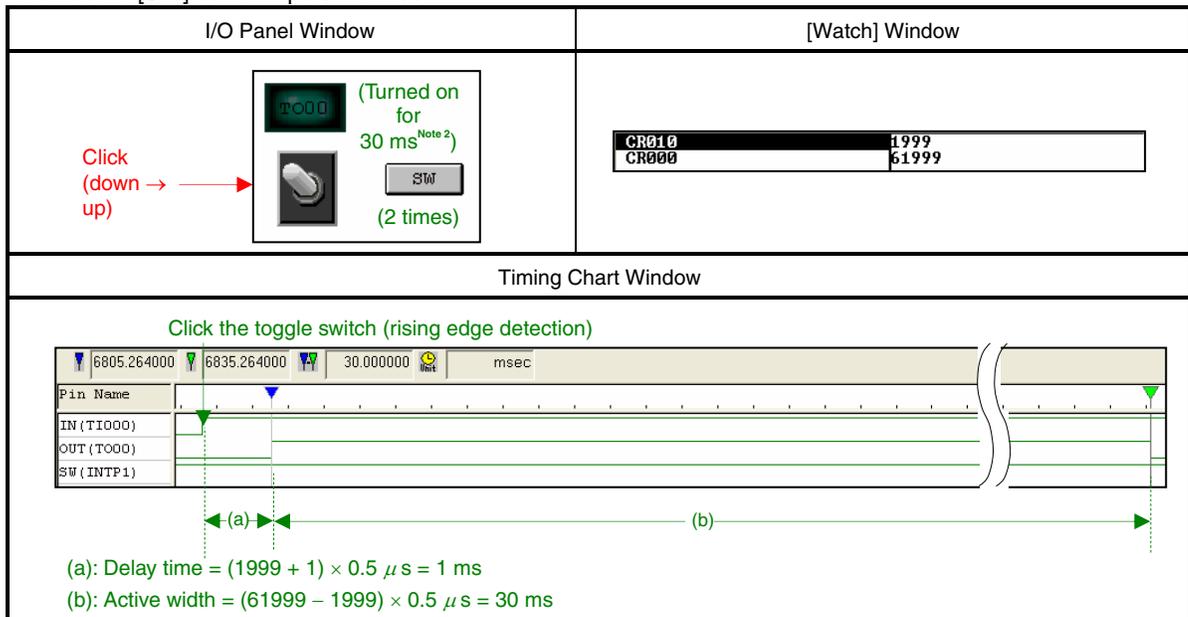
Check that the lighting time of the [LED] in the I/O panel window, the waveform in the timing chart window, and the values of the CR000 and CR010 registers in the [Watch] window change.

- Number of [SW] button inputs: 0 times



Note 1

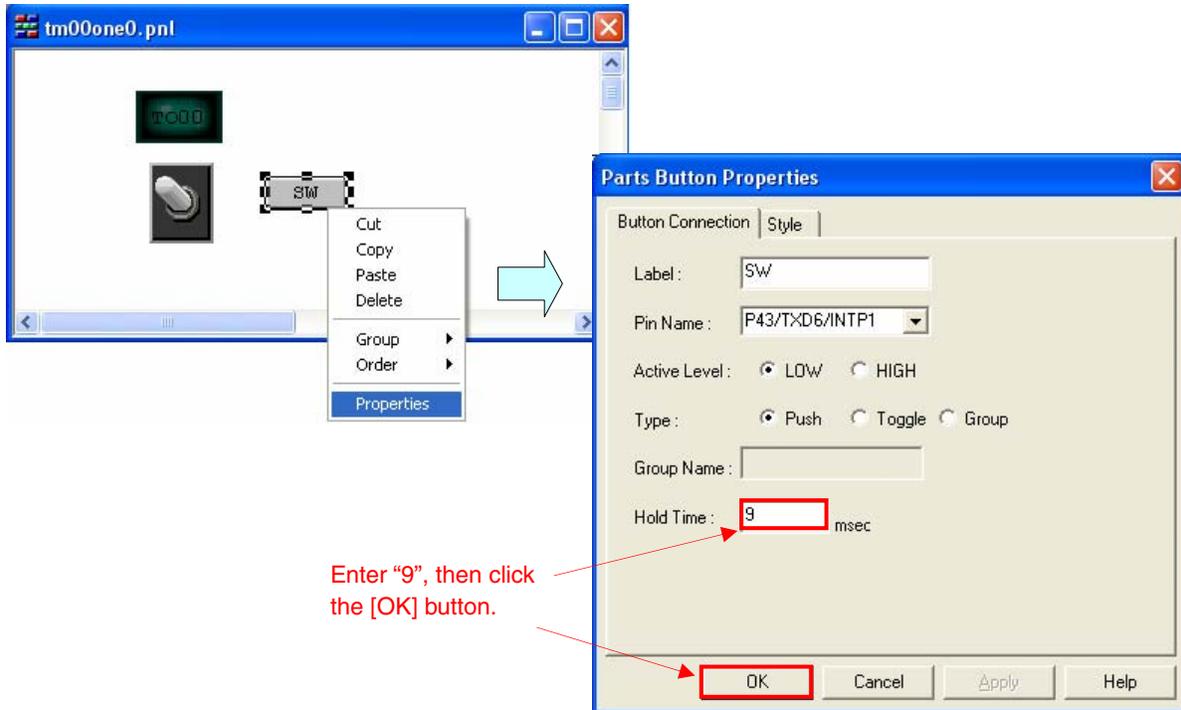
- Number of [SW] button inputs: 2 times



- Notes**
1. The active pulse widths from the zeroth switch input are repeated after the third switch input.
 2. This may differ from the actual lighting time, depending on the operation environment of the PC used.

[Supplement] The [SW] button hold time can be set to less than 10 ms to check whether chattering is being detected.

- <1> Select  on the toolbar.
- <2> Right-click the [SW] button in the I/O panel window and select [Properties].
- <3> Enter "9" for the Hold Time and click the [OK] button.



- <4> Select  on the toolbar.
- <5> Execute the program and click the [SW] button. Even if the [SW] button is clicked, chattering will be identified and the active width of the one-shot pulse (CR000 register value) will not change, because the button hold time is 9 ms.

CHAPTER 6 RELATED DOCUMENTS

Document Name		Japanese/English
78K0S/KU1+ User's Manual		PDF
78K0S/KY1+ User's Manual		PDF
78K0S/KA1+ User's Manual		PDF
78K0S/KB1+ User's Manual		PDF
78K/0S Series Instructions User's Manual		PDF
RA78K0S Assembler Package User's Manual	Language	PDF
	Operation	PDF
CC78K0S C Compiler User's Manual	Language	PDF
	Operation	PDF
PM+ Project Manager User's Manual		PDF
SM+ System Simulator Operation User's Manual		PDF
Flash Programming Manual (Basic) MINICUBE2 version	78K0S/KU1+	PDF
	78K0S/KY1+	PDF
	78K0S/KA1+	PDF
	78K0S/KB1+	PDF
78K0S/Kx1+ Application Note	Sample Program Startup Guide	PDF
	Sample Program (Initial Settings) LED Lighting Switch Control	PDF
	Sample Program (Interrupt) External Interrupt Generated by Switch Input	PDF
	Sample Program (Low-Voltage Detection) Reset Generation During Detection at Less than 2.7 V	PDF
	Sample Program (16-bit Timer/Event Counter 00) Interval Timer	PDF
	Sample Program (16-bit Timer/Event Counter 00) External Event Counter	PDF
	Sample Program (16-bit Timer/Event Counter 00) Pulse Width Measurement	PDF
	Sample Program (16-bit Timer/Event Counter 00) PPG Output	PDF

<R>

APPENDIX A PROGRAM LIST

As a program list example, the 78K0S/KB1+ microcontroller source program is shown below.

● main.asm (Assembly language version)

```
*****
;
;   NEC Electronics       78K0S/KB1+
;
;*****
;   78K0S/KB1+ Sample program
;*****
;   16-bit timer 00 (one-shot pulse output)
;*****
;<<History>>
;   2007.7.-- Release
;*****
;
;<<Overview>>
;
;This sample program presents an example of using the one-shot pulse output
;function of 16-bit timer 00. The rising edge of the external signal that
;is input to the TI000 pin is detected and a one-shot pulse of a 10 ms
;width is output from the T000 pin after a given delay time. The active
;pulse width is changed by every switch input.
;
;
; <Principal setting contents>
;
; - Stop the watchdog timer operation
; - Set the low-voltage detection voltage (VLVI) to 4.3 V +/-0.2 V
; - Generate an internal reset signal (low-voltage detector) when VDD <
VLVI after VDD >= VLVI
; - Set the CPU clock to 8 MHz
; - Set the clock supplied to the peripheral hardware to 8 MHz
; - Set the valid edge of external interrupt INTPl to falling edge
; - Set the chattering detection time during switch input to 10 ms
;
;
; <16-bit timer 00 settings>
; - Operation mode: Clear & start the timer count upon detection of the
valid edge of the TI000 pin
; - Setting the valid edge of the TI000 pin to the rising edge
; - Count clock = fxp/4 (2 MHz)
; - Using CR000 and CR010 as compare registers
; - Setting to the one-shot pulse output mode
```

```

; - Reversing the output upon a match with CR000 or CR010
; - Setting the initial output value to Low (one-shot pulse output level =
High)
; - Enabling the timer output
; - Initializing the delay time of CR010 to 1 ms
; - Initializing the active width of CR000 to 10 ms
; - Setting the output latch of P31 to Low (to use the T000 output)
; - Setting P31 to the output mode (to use the T000 output)
;
;
; <Number of switch inputs and one-shot pulse output active width>
;
; +-----+
; | SW Inputs | One-Shot Pulse |
; |           | Output Active Width |
; |-----|
; | 0 times  | 10 ms          |
; | 1 time   | 20 ms          |
; | 2 times  | 30 ms          |
; |-----|
; # The active widths from the zeroth switch input are repeated after the
third switch input.
;
;
;<<I/O port settings>>
;
; Input: P30, P43
; Output: P00-P03, P20-P23, P31-P33, P40-P42, P44-P47, P120-P123, P130
; # All unused ports are set as the output mode.
;
;*****
;=====
;
; Vector table
;
;=====
XVCTCSEG AT 0000H
    DW RESET_START ;(00) RESET
    DW RESET_START ;(02) --
    DW RESET_START ;(04) --
    DW RESET_START ;(06) INTLVI
    DW RESET_START ;(08) INTP0
    DW INTERRUPT_P1 ;(0A) INTP1
    DW RESET_START ;(0C) INTTMH1
    DW RESET_START ;(0E) INTTM000
    DW RESET_START ;(10) INTTM010

```

```

DW   RESET_START      ;(12) INTAD
DW   RESET_START      ;(14) --
DW   RESET_START      ;(16) INTP2
DW   RESET_START      ;(18) INTP3
DW   RESET_START      ;(1A) INTTM80
DW   RESET_START      ;(1C) INTSRE6
DW   RESET_START      ;(1E) INTSR6
DW   RESET_START      ;(20) INTST6

;=====
;
;   Define the memory stack area
;
;=====
XSTKDSEG  AT      0FEE0H
STACKEND:
        DS      20H          ; Memory stack area = 32 bytes
STACKTOP:          ; Start address of the memory stack area = FF00H

;*****
;
;   Initialization after RESET
;
;*****
XMAIN      CSEG  UNIT
RESET_START:
;-----
;   Initialize the stack pointer
;-----
        MOVW   AX,   #STACKTOP
        MOVW   SP,   AX          ; Set the stack pointer

;-----
;   Initialize the watchdog timer
;-----
        MOV    WDTM, #01110111B ; Stop the watchdog timer operation

;-----
;   Detect low-voltage + set the clock
;-----

;-----  Set the clock <1>  -----
        MOV    PCC,  #00000000B ; The clock supplied to the CPU (fcpu) = fxp (=
fx/4 = 2 MHz)
        MOV    LSRM,  #00000001B ; Stop the oscillation of the low-speed
internal oscillator

;-----  Check the reset source  -----

```

```

MOV  A,    RESF          ; Read the reset source
BT   A.0,  $SET_CLOCK   ; Omit subsequent LVI-related processing and go
to SET_CLOCK during LVI reset

;----- Set low-voltage detection -----
MOV  LVIS, #00000000B   ; Set the low-voltage detection level (VLVI) to
4.3 V +-0.2 V
SET1 LVION              ; Enable the low-voltage detector operation

MOV  A,    #40          ; Assign the 200 us wait count value
;----- 200 us wait -----
WAIT_200US:
DEC  A
BNZ  $WAIT_200US       ; 0.5[us/clock] x 10[clock] x 40[count] = 200[us]

;----- VDD >= VLVI wait processing -----
WAIT_LVI:
NOP
BT   LVIF, $WAIT_LVI   ; Branch if VDD < VLVI

SET1 LVIMD             ; Set so that an internal reset signal is
generated when VDD < VLVI

;----- Set the clock <2> -----
SET_CLOCK:
MOV  PPCC, #00000000B   ; The clock supplied to the peripheral hardware
(fxp) = fx (= 8 MHz)
; -> The clock supplied to the CPU (fcpu) = fxp
= 8 MHz

;-----
; Initialize the port 0
;-----
MOV  P0,   #00000000B   ; Set output latches of P00-P03 as low
MOV  PM0,  #11110000B   ; Set P00-P03 as output mode

;-----
; Initialize the port 2
;-----
MOV  P2,   #00000000B   ; Set output latches of P20-P23 as low
MOV  PM2,  #11110000B   ; Set P20-P23 as output mode

;-----
; Initialize the port 3
;-----
MOV  P3,   #00000000B   ; Set output latches of P30-P33 as low
MOV  PM3,  #11110001B   ; Set P31-P33 as output mode, P30/TI000 as input
mode

```

```

;-----
; Initialize the port 4
;-----
MOV   P4,   #00000000B ; Set output latches of P40-P47 as low
MOV   PU4,  #00001000B ; Connect on-chip pull-up resistor to P43
MOV   PM4,  #00001000B ; Set P40-P42 and P44-P47 as output mode, P43 as
input mode

;-----
; Initialize the port 12
;-----
MOV   P12,  #00000000B ; Set output latches of P120-P123 as low
MOV   PM12, #11110000B ; Set P120-P123 as output mode

;-----
; Initialize the port 13
;-----
MOV   P13,  #00000001B ; Set output latch of P130 as high

;-----
; Set 16-bit timer 00
;-----
MOV   CRC00, #00000000B ; Use CR000 and CR010 as compare registers
MOVW  AX,    #2000-1     ; Set the delay time to 1 ms
MOVW  CR010, AX          ; Initialize compare value CR010
ADDW  AX,    #20000      ; Set the active width to 10 ms
MOVW  CR000, AX          ; Initialize compare value CR000
MOV   PRM00, #00010001B ; Count clock = fxp/4 = 2 MHz, valid edge
of the TI000 pin = rising edge
MOV   TOC00, #00110111B ; One-shot pulse output mode, output
reversal upon a match between CR000 or CR010 and TM00,
; default 0 output, timer output enable
MOV   TMC00, #00001000B ; Start the timer operation (clear & start
upon detection of the valid edge of the TI000 pin)

;-----
; Set the interrupt
;-----
MOV   INTM0, #00000000B ; Set the valid edge of INTP1 to falling
edge
MOV   IF0,  #00H        ; Clear invalid interrupt requests in advance
CLR1  PMK1              ; Unmask INTP1 interrupts

EI                                     ; Enable vector interrupt

;*****
;

```

```

; Main loop
;
;*****
MAIN_LOOP:
    NOP
    BR    $MAIN_LOOP      ; Go to the MAIN_LOOP

;*****
;
; External interrupt INTP1
;
;*****
INTERRUPT_P1:
    PUSH  AX              ; Save the AX register data to the stack
    PUSH  BC              ; Save the BC register data to the stack

;----- 10 ms wait to handle chattering -----
    MOV   C,    #47       ; Set the count value for the major loop
CHAT_LOOP2:
    MOV   B,    #211      ; Set the count value for the minor loop
CHAT_LOOP1:
    NOP
    DBNZ  B,    $CHAT_LOOP1
    NOP
    DBNZ  C,    $CHAT_LOOP2

    CLR1  PIF1           ; Clear the INTP1 interrupt request

;----- Identification of chattering detection -----
    BT    P4.3, $END_INTP1 ; Branch if there is no switch input

;----- Change the delay time of the one-shot pulse output -----
    MOV   TMC00, #0000000B ; Stop the timer operation

    MOVW  AX,    CR000    ; Read CR000
    ADDW  AX,    #20000    ; Increment the active width by 10 ms
    BNC   $CHANGE_CR     ; Branch if no overflow occurs
    MOVW  AX,    #22000-1 ; Initialize the active width to 10 ms
CHANGE_CR:
    MOVW  CR000,    AX    ; Write to CR000

    SET1  LVR00          ; Set to default 0 output

    MOV   TMC00, #00001000B ; Start the timer operation (clear & start
upon detection of the valid edge of the TI000 pin)

END_INTP1:
    POP   BC              ; Restore the BC register data

```

```
POP    AX           ; Restore the AX register data
RETI   ; Return from interrupt servicing

end
```

● main.c (C language version)

```

/*****
    NEC Electronics      78K0S/KB1+

*****
    78K0S/KB1+  Sample program
*****
    16-bit timer 00 (one-shot pulse output)
*****
<<History>>
    2007.7.--  Release
*****

```

<<Overview>>

This sample program presents an example of using the one-shot pulse output function of 16-bit timer 00. The rising edge of the external signal that is input to the TI000 pin is detected and a one-shot pulse of a 10 ms width is output from the T000 pin after a given delay time. The active pulse width is changed by every switch input.

<Principal setting contents>

- Declare a function run by an interrupt: INTP1 -> fn_intp1()
- Stop the watchdog timer operation
- Set the low-voltage detection voltage (VLVI) to 4.3 V +/-0.2 V
- Generate an internal reset signal (low-voltage detector) when VDD < VLVI after VDD >= VLVI
- Set the CPU clock to 8 MHz
- Set the clock supplied to the peripheral hardware to 8 MHz
- Set the valid edge of external interrupt INTP1 to falling edge
- Set the chattering detection time during switch input to 10 ms

<16-bit timer 00 settings>

- Operation mode: Clear & start the timer count upon detection of the valid edge of the TI000 pin
- Setting the valid edge of the TI000 pin to the rising edge
- Count clock = f_{xp}/4 (2 MHz)
- Using CR000 and CR010 as compare registers
- Setting to the one-shot pulse output mode
- Reversing the output upon a match with CR000 or CR010
- Setting the initial output value to Low (one-shot pulse output level = High)
- Enabling the timer output

- Initializing the delay time of CR010 to 1 ms
- Initializing the active width of CR000 to 10 ms
- Setting the output latch of P31 to Low (to use the T000 output)
- Setting P31 to the output mode (to use the T000 output)

<Number of switch inputs and one-shot pulse output active width>

SW Inputs	One-Shot Pulse Output Active Width
0 times	10 ms
1 time	20 ms
2 times	30 ms

The active widths from the zeroth switch input are repeated after the third switch input.

<<I/O port settings>>

Input: P30, P43

Output: P00-P03, P20-P23, P31-P33, P40-P42, P44-P47, P120-P123, P130

All unused ports are set as the output mode.

/*=====

Preprocessing directive (#pragma)

```

=====*/
#pragma SFR /* SFR names can be described at the C
source level */
#pragma EI /* EI instructions can be described at the
C source level */
#pragma DI /* DI instructions can be described at the
C source level */
#pragma NOP /* NOP instructions can be described at
the C source level */
#pragma interrupt INTP1 fn_intp1 /* Interrupt function declaration:INTP1 */
#pragma realregister /* Use the directly register reference
function */

```

/******

```

Initialization after RESET

*****/
void hdwinit(void){
    unsigned char ucCnt200us;    /* 8-bit variable for 200 us wait */

/*-----
    Initialize the watchdog timer + detect low-voltage + set the clock
-----*/
    /* Initialize the watchdog timer */
    WDTM = 0b01110111;    /* Stop the watchdog timer operation */

    /* Set the clock <1> */
    PCC = 0b00000000;    /* The clock supplied to the CPU (fcpu) =
fxp (= fx/4 = 2 MHz) */
    LSRCM = 0b00000001;    /* Stop the oscillation of the low-speed
internal oscillator */

    /* Check the reset source */
    if (!(RESF & 0b00000001)){    /* Omit subsequent LVI-related processing
during LVI reset */

        /* Set low-voltage detection */
        LVIS = 0b00000000;    /* Set the low-voltage detection level
(VLVI) to 4.3 V +/-0.2 V */
        LVION = 1;    /* Enable the low-voltage detector
operation */

        for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){    /* Wait of
about 200 us */
            NOP();
        }

        while (LVIF){    /* Wait for VDD >= VLVI */
            NOP();
        }

        LVIMD = 1;    /* Set so that an internal reset signal is
generated when VDD < VLVI */
    }

    /* Set the clock <2> */
    PPCC = 0b00000000;    /* The clock supplied to the peripheral
hardware (fxp) = fx (= 8 MHz)
                                -> The clock supplied to the CPU
(fccpu) = fxp = 8 MHz */

/*-----

```

```

Initialize the port 0
-----*/
P0    = 0b00000000;      /* Set output latches of P00-P03 as low */
PM0   = 0b11110000;      /* Set P00-P03 as output mode */

/*-----
Initialize the port 2
-----*/
P2    = 0b00000000;      /* Set output latches of P20-P23 as low */
PM2   = 0b11110000;      /* Set P20-P23 as output mode */

/*-----
Initialize the port 3
-----*/
P3    = 0b00000000;      /* Set output latches of P30-P33 as low */
PM3   = 0b11110001;      /* Set P31-P33 as output mode, P30/TI000
as input mode */

/*-----
Initialize the port 4
-----*/
P4    = 0b00000000;      /* Set output latches of P40-P47 as low */
PU4   = 0b00001000;      /* Connect on-chip pull-up resistor to P43
*/
PM4   = 0b00001000;      /* Set P40-P42 and P44-P47 as output mode,
P43 as input mode */

/*-----
Initialize the port 12
-----*/
P12   = 0b00000000;      /* Set output latches of P120-P123 as low
*/
PM12  = 0b11110000;      /* Set P120-P123 as output mode */

/*-----
Initialize the port 13
-----*/
P13   = 0b00000001;      /* Set output latch of P130 as high */

/*-----
Set 16-bit timer 00
-----*/
CRC00 = 0b00000000;      /* Use CR000 and CR010 as compare
registers */
CR010 = 2000 - 1;         /* Initialize the delay time to 1 ms */
CR000 = CR010 + 20000;    /* Initialize the active width to 10 ms */
PRM00 = 0b00010001;      /* Count clock = fxp/4 = 2 MHz, valid edge
of the TI000 pin = rising edge */

```

```

        TOC00 = 0b00110111;          /* One-shot pulse output mode, output
reversal upon a match between CR000 or CR010 and TM00,
                                default 0 output, timer output enable */
        TMC00 = 0b00001000;          /* Start the timer operation (clear &
start upon detection of the valid edge of the TI000 pin) */

/*-----
   Set the interrupt
-----*/
        INTM0 = 0b00000000;          /* Set the valid edge of INTP1 to falling
edge */
        IF0 = 0x00;                  /* Clear invalid interrupt requests */
        PMK1 = 0;                    /* Unmask INTP1 interrupts */

        return;
    }

/*****

Main loop

*****/
void main(void){

    EI();                             /* Enable vector interrupt */

    while (1){
        NOP();
    }
}

/*****

Interrupt INTP1

*****/
__interrupt void fn_intp1(){
    unsigned int unChat;              /* 8-bit variable for removing chattering
*/

    for (unChat = 0; unChat <1111; unChat++){ /* Loop for a 10 ms wait */
        NOP();
    }

    PIF1 = 0;                         /* Clear the INTP1 interrupt request */

    if (!P4.3){                       /* Processing performed if SW is on for 10
ms or more */

```

```
        TMC00 = 0b00000000;          /* Stop the timer operation */
        CR000 = CR000 + 20000;        /* Increment the active width by 10
ms */
        if (__getcy()){              /* Processing when an overflow
occurs */
            CR000 = 22000 - 1;        /* Initialize the active width to 10
ms */
        }
        LVR00 = 1;                   /* Set to default 0 output */
        TMC00 = 0b00001000;          /* Start the timer operation (clear
& start upon detection of the valid edge of the TI000 pin) */
    }
    return;
}
```

● op.asm (Common to assembly language and C language versions)

```

;=====
;
;   Option byte
;
;=====
OPBT   CSEG   AT   0080H
        DB     10011100B       ; Option byte area
;
;           ||||
;           |||+----- Low-speed internal oscillator can be
stopped by software
;
;           |++----- High-speed internal oscillation clock (8
MHz) is selected for system clock source
;
;           +----- P34/RESET pin is used as RESET pin

        DB     11111111B       ; Protect byte area (for the self programming
mode)
;
;           |||||
;           ++++++----- All blocks can be written or erased

end

```

APPENDIX B REVISION HISTORY

The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what." field.

Edition	Date Published	Page	Revision
1st edition	December 2007	–	–
2nd edition	September 2008	p.29	CHAPTER 5 OPERATION CHECK USING SYSTEM SIMULATOR SM+ • Modification of description in Caution ((as of September 2007) → (as of July 2008))
		pp.29 to 31	Modification of 5.1 Building the Sample Program
		p.31	5.2 Operation with SM+ • Addition of (1)
		p.35	CHAPTER 6 RELATED DOCUMENTS • Addition of Flash Programming Manual (Basic) MINICUBE2 version

*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 Esplandiú, 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/>