

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



78K0R/Lx3

Sample Program (LCD Controller/Driver)

LCD Clock (Internal Booster Method)

This application note describes a sample program that executes 24-hour clock display by using an LCD controller/driver and a real-time counter.

Target devices

- 78K0R/LF3 microcontroller
- 78K0R/LG3 microcontroller
- 78K0R/LH3 microcontroller

CONTENTS

CHAPTER 1 OVERVIEW	3
1.1 Main Contents of Initial Settings.....	4
1.2 Contents Following Main Loop.....	5
CHAPTER 2 CIRCUIT DIAGRAM	6
2.1 Circuit Diagram.....	6
2.2 Used Devices Other than Microcontroller.....	12
CHAPTER 3 SOFTWARE	14
3.1 Included Files.....	14
3.2 Internal Peripheral Functions to Be Used.....	15
3.3 Initial Settings and Operation Overview.....	15
3.4 Flow Chart.....	17
CHAPTER 4 SETTING METHODS	21
4.1 Setting to Use Peripheral Hardware.....	21
4.2 Setting Up Real-Time Counter.....	23
4.3 Setting Up LCD Controller/Driver.....	29
4.4 Software Coding Examples.....	37
CHAPTER 5 RELATED DOCUMENTS	47
APPENDIX A PROGRAM LIST	48
APPENDIX B REVISION HISTORY	115

Document No. U20029EJ1V0AN00 (1st edition)
Date Published September 2009 N

© NEC Electronics Corporation 2009
Printed in Japan

- The information in this document is current as of July, 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.

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

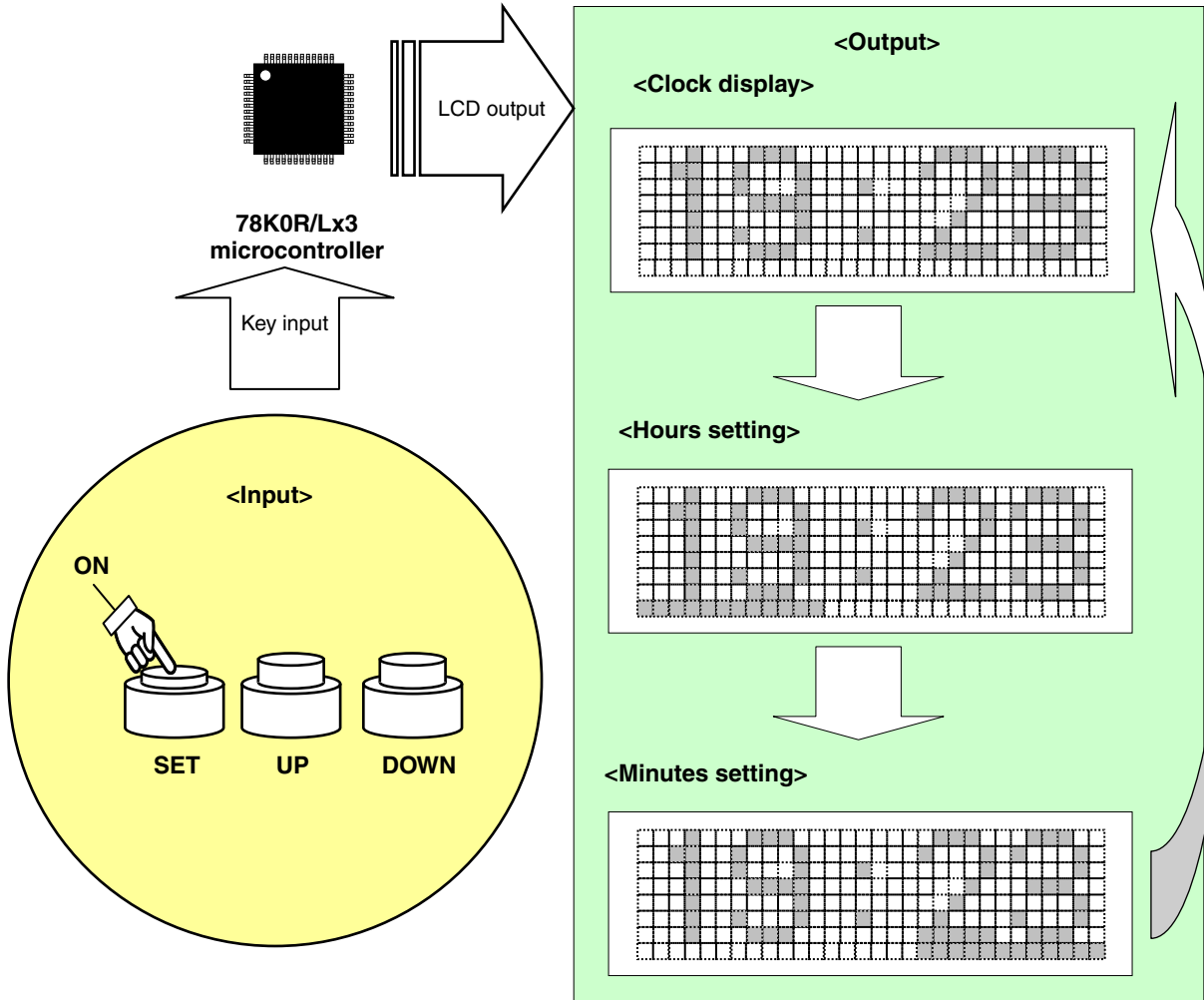
(M8E0909)

CHAPTER 1 OVERVIEW

This sample program executes 24-hour clock display by using an LCD controller/driver and a real-time counter. The time set by the real-time counter is stored in the A pattern area of the LCD data memory and is converted into time display each time the real-time counter's periodic interrupt occurs (at one-minute intervals).

The hour and minute settings can be changed by pressing the SET key a specified number of times and then pressing the UP and DOWN keys.

[Operation overview]



Number of Times of Inputting SET Key	Status
0	Clock display status
1	Hours setting status
2	Minutes setting status

Note When the SET key is input the third time and onward, the status changes from the start (i.e., from when the number of times of inputting the SET key is zero).

1.1 Main Contents of Initial Settings

The main contents of the initial settings are as follows.

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling on-chip debug to operate

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports
 - Setting SEG4 to SEG53^{Note 1} for segment output
 - Setting COM0 to COM7 for common output
 - Setting INTP1 to be detected at the falling edge
 - Setting INTP2 to be detected at the falling edge
 - Setting INTP5 to be detected at the falling edge
- Checking whether V_{DD} is 4.2 V or more by using the low-voltage detector^{Note 2}
- Specifying that the CPU/peripheral hardware clock run on the internal high-speed oscillation clock (8 MHz)
- Stopping the X1 oscillator
- Starting operation of the XT1 oscillator
- Setting up the real-time counter
 - Setting the real-time counter to generate the constant-period interrupt once every minute
 - Setting to not use the interval interrupt of the real-time counter
 - Setting the current time to 00:00:00 a.m.
 - Setting to not use the alarm interrupt
- Setting up timer array unit
 - Setting channel 4 in interval timer mode of about 10 ms
- Setting up LCD controller/driver
 - Selecting Internal voltage boosting method for the LCD drive voltage generator
 - Setting an LCD display mode of 8-time division and 1/4 bias
 - Setting LCD display data in the RAM area
 - Setting to display the data of only the A-pattern area
 - Setting f_{CLK}/2⁸ as the LCD source clock (f_{LCD}) and f_{LCD}/2⁷ as the LCD clock (LCD clock: 244 Hz, frame frequency: 61 Hz)
 - Setting 1.00 V (LCD drive voltage (V_{LCD}) = 4.00 V) as the reference voltage of the LCD boost level

Notes 1. These settings are for the 78K0R/LH3. When the 78K0R/LF3 is used, SEG4 to SEG30 are used for segment output. When the 78K0R/LG3 is used, SEG4 to SEG39 are used for segment output.

2. For details of the low-voltage detector, refer to the User's Manual.

1.2 Contents Following Main Loop

After the initial settings have been completed, the microcontroller enters the HALT mode. It is released from the HALT mode by the constant-period interrupt of the real-time counter or detection of the falling edge of the UP, DOWN, or SET key. If the microcontroller has been released from the HALT mode by the constant-period interrupt of the real-time counter, the displayed time is changed. If it has been released by input of the UP, DOWN, or SET key, an action to avoid chattering is taken. When input of a key has been determined, time is set to the clock.

When the SET key is input the first time, hours can be set. When it is input the second time, minutes can be set. When the SET key is input the third time, the clock display function is returned and the set time is displayed. While hours are being set, inputting the UP key increments the time by 1 hour. While minutes are displayed, inputting the UP key increments the time by 1 minute. Likewise, inputting the DOWN key decrements the time by 1 hour or 1 minute.

Caution For cautions when using the device, refer to the User's Manual.

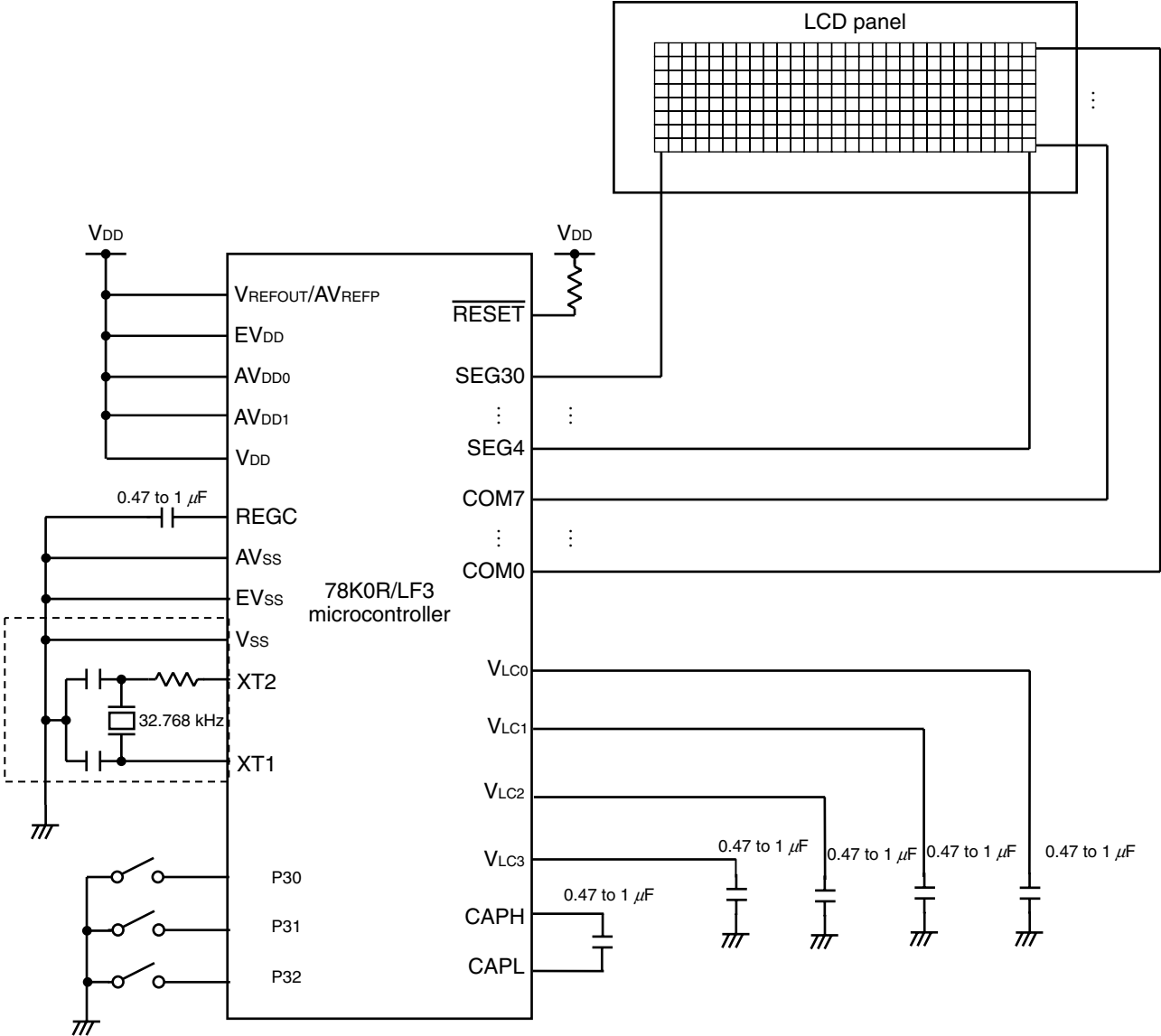
CHAPTER 2 CIRCUIT DIAGRAM

This chapter provides a circuit diagram used in this sample program.

2.1 Circuit Diagram

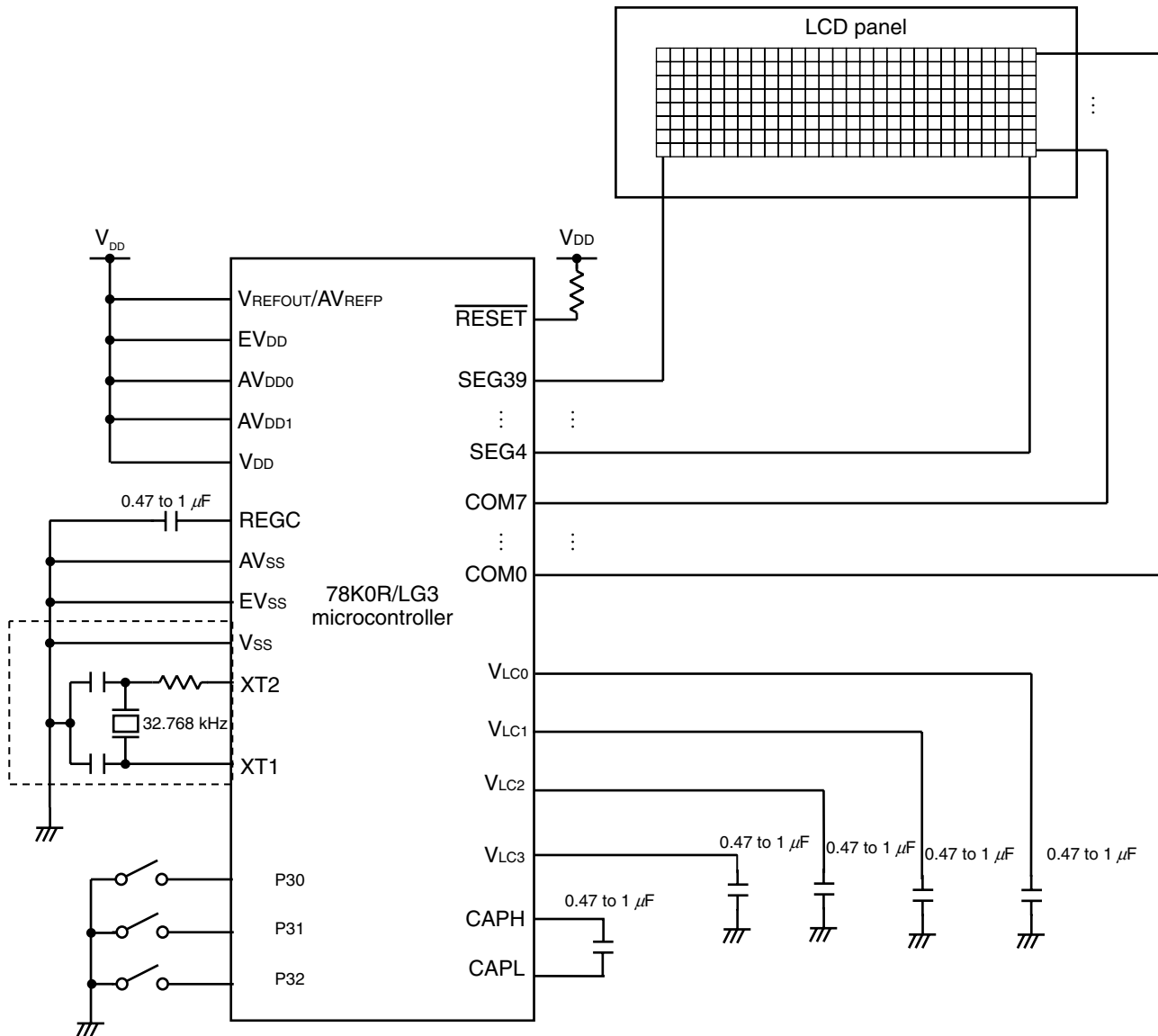
A circuit diagram is shown below.

(1) 78K0R/LF3 connection example



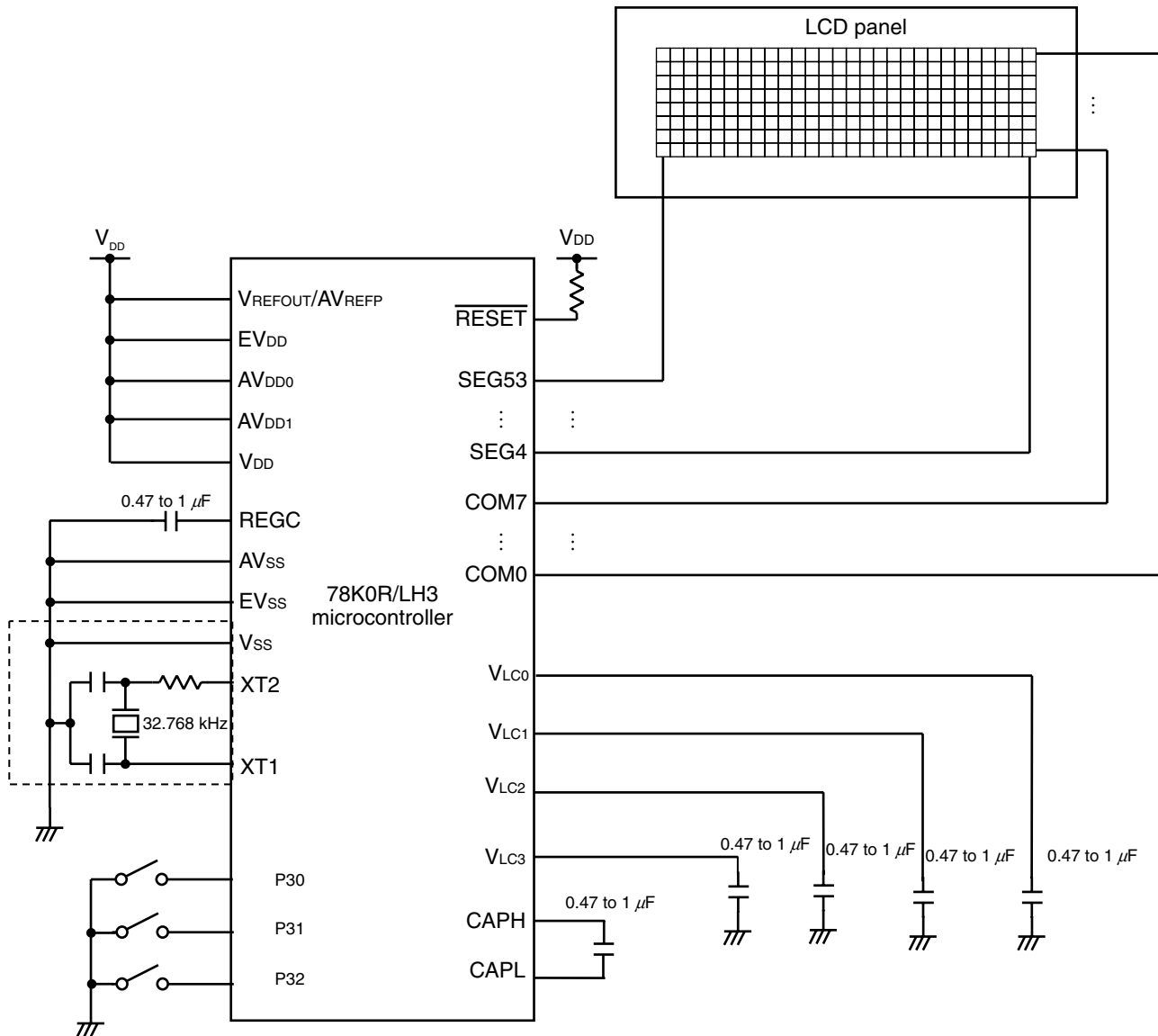
- Cautions**
1. Use the microcontroller at a voltage in the range of $V_{DD} = 5.0\text{ V}$ (because a low voltage is detected within in a range of $4.22 \pm 0.1\text{ V} < V_{DD}$).
 2. Make EV_{DD} , AV_{DD0} , AV_{DD1} , V_{LC0} , and V_{REFOUT}/AV_{REFP} the same potential as V_{DD} .
 3. Make AV_{SS} the same potential as EV_{SS} or V_{SS} and connect it directly to GND.
 4. Connect REGC to V_{SS} via a capacitor (0.47 to 1 μF).
 5. Handle unused pins that are not shown in the circuit diagram as follows:
 - I/O ports: Set them to output mode and leave them open (unconnected).
 - Input ports: Connect them independently to V_{DD} or V_{SS} via a resistor.
 6. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.
 - Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
 7. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.
 8. Use a non-polar capacitor between CAPH and CAPL.
 9. In this sample program, the P40/TOOL0 and P41/TOOL1 pins are used for on-chip debugging.
 10. Twenty-seven segment signal output pins and eight common signal output pins on the LCD panel are used.
 11. An on-chip pull-up resistor is connected to the P30, P31, and P32 pins.

(2) 78K0R/LG3 connection example



- Cautions**
1. Use the microcontroller at a voltage in the range of $V_{DD} = 5.0\text{ V}$ (because a low voltage is detected within in a range of $4.22 \pm 0.1\text{ V} < V_{DD}$).
 2. Make EV_{DD} , AV_{DD0} , AV_{DD1} , V_{LC0} , and V_{REFOUT}/AV_{REFP} the same potential as V_{DD} .
 3. Make AV_{SS} the same potential as EV_{SS} or V_{SS} and connect it directly to GND.
 4. Connect REGC to V_{SS} via a capacitor (0.47 to 1 μF).
 5. Handle unused pins that are not shown in the circuit diagram as follows:
 - I/O ports: Set them to output mode and leave them open (unconnected).
 - Input ports: Connect them independently to V_{DD} or V_{SS} via a resistor.
 6. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.
 - Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
 7. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.
 8. Use a non-polar capacitor between CAPH and CAPL.
 9. In this sample program, the P40/TOOL0 and P41/TOOL1 pins are used for on-chip debugging.
 10. Twenty-seven segment signal output pins and eight common signal output pins on the LCD panel are used.
 11. An on-chip pull-up resistor is connected to the P30, P31, and P32 pins.

(3) 78K0R/LH3 connection example



- Cautions**
1. Use the microcontroller at a voltage in the range of $V_{DD} = 5.0\text{ V}$ (because a low voltage is detected within in a range of $4.22 \pm 0.1\text{ V} < V_{DD}$).
 2. Make EV_{DD} , AV_{DD0} , AV_{DD1} , V_{LC0} , and V_{REFOUT}/AV_{REFP} the same potential as V_{DD} .
 3. Make AV_{SS} the same potential as EV_{SS} or V_{SS} and connect it directly to GND.
 4. Connect REGC to V_{SS} via a capacitor (0.47 to 1 μF).
 5. Handle unused pins that are not shown in the circuit diagram as follows:
 - I/O ports: Set them to output mode and leave them open (unconnected).
 - Input ports: Connect them independently to V_{DD} or V_{SS} via a resistor.
 6. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.
 - Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
 7. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.
 8. Use a non-polar capacitor between CAPH and CAPL.
 9. In this sample program, the P40/TOOL0 and P41/TOOL1 pins are used for on-chip debugging.
 10. Twenty-seven segment signal output pins and eight common signal output pins on the LCD panel are used.
 11. An on-chip pull-up resistor is connected to the P30, P31, and P32 pins.

2.2 Used Devices Other than Microcontroller

The following devices are used in addition to the microcontroller:

(1) LCD panel

Connect an LCD panel to the segment signal output and common signal output pins of the device.

Use an LCD panel that is suitable for each device and that supports the segment signal output and common signal output pins and the bias mode of the device.

The display specifications of the LCD used in this sample program are shown below. In addition, the dot pattern of the LCD panel is also shown.

Figure 2-1. LCD Panel/Segment Definition

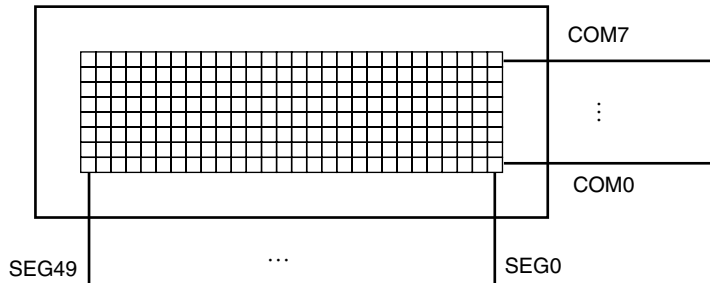
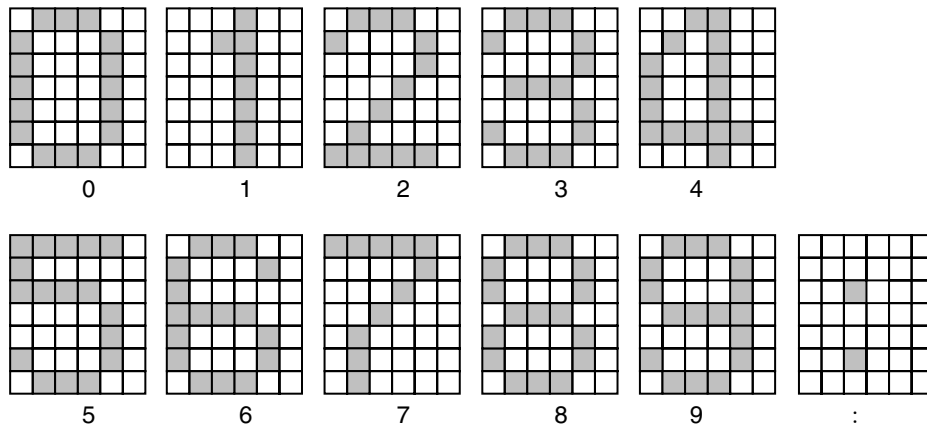
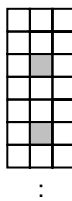


Figure 2-2. Dot Pattern of LCD Panel



Caution When the 78K0R/LF3 is used, the segment signal output ports run short. Therefore, the dot pattern “:” is rewritten by the following contents.



(2) Switches (three switches)



Three switches are used to input and adjust the time.

CHAPTER 3 SOFTWARE


This chapter describes the files included in the compressed file to be downloaded, internal peripheral functions of the microcontroller to be used, and initial settings and provides an operation overview of the sample program and a flow chart.


3.1 Included Files

The following table shows the files included in 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	● Note
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer, selecting the internal high-speed oscillator frequency, and setting up the LVI default start function.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM+		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM+		●

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

Remark  : Only the source file is included.

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

3.2 Internal Peripheral Functions to Be Used

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

- INTP1: Used to input SET key.
- INTP2: Used to input UP key.
- INTP5: Used to input DOWN key.
- Real-time counter: Used to set time to be displayed.
- Timer array unit: Used to avoid chattering.
- LCD controller/driver: Used to control and drive the LCD.
- Low-voltage detector: Used to check that V_{DD} is 4.2 V or more.

3.3 Initial Settings and Operation Overview

In this sample program, initial settings including the selection of the clock frequency, setting of the I/O ports, setting of the real-time counter, setting of the timer array unit, and setting of the LCD controller/driver are performed.

After completion of the initial settings, the microcontroller enters the HALT mode. It is released from the HALT mode by the constant-period interrupt of the real-time counter or detection of the falling edge of P30, P31, or P32. If the microcontroller has been released from the HALT mode by the constant-period interrupt of the real-time counter, the displayed time is changed. If it has been released by input of the UP, DOWN, or SET key, an action to avoid chattering is taken. When input of a key has been determined, time is set to the clock.

When the SET key is input for the first time, hours can be set. When it is input the second time, minutes can be set. When the SET key is input the third time, the clock display function is returned and the set time is displayed. While hours are being set, inputting the UP key increments the time by 1 hour. While minutes are displayed, inputting the UP key increments the time by 1 minute. Likewise, inputting the DOWN key decrements the time by 1 hour or 1 minute.

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

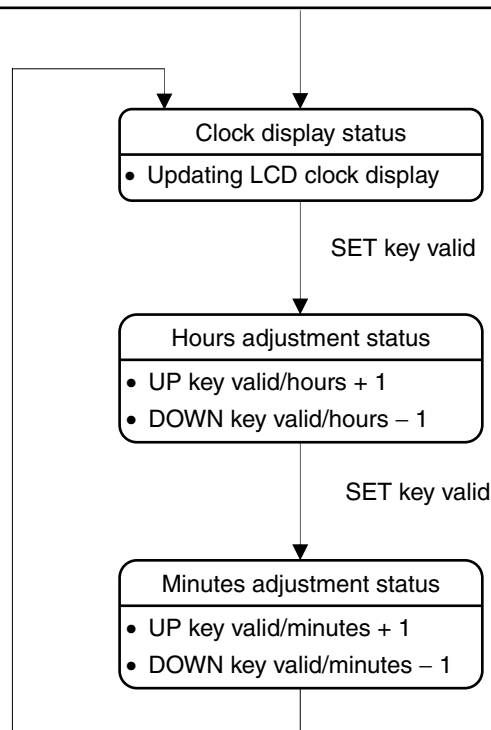
Initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillation clock frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling on-chip debug to operate

<Settings during initialization immediately after a reset ends>

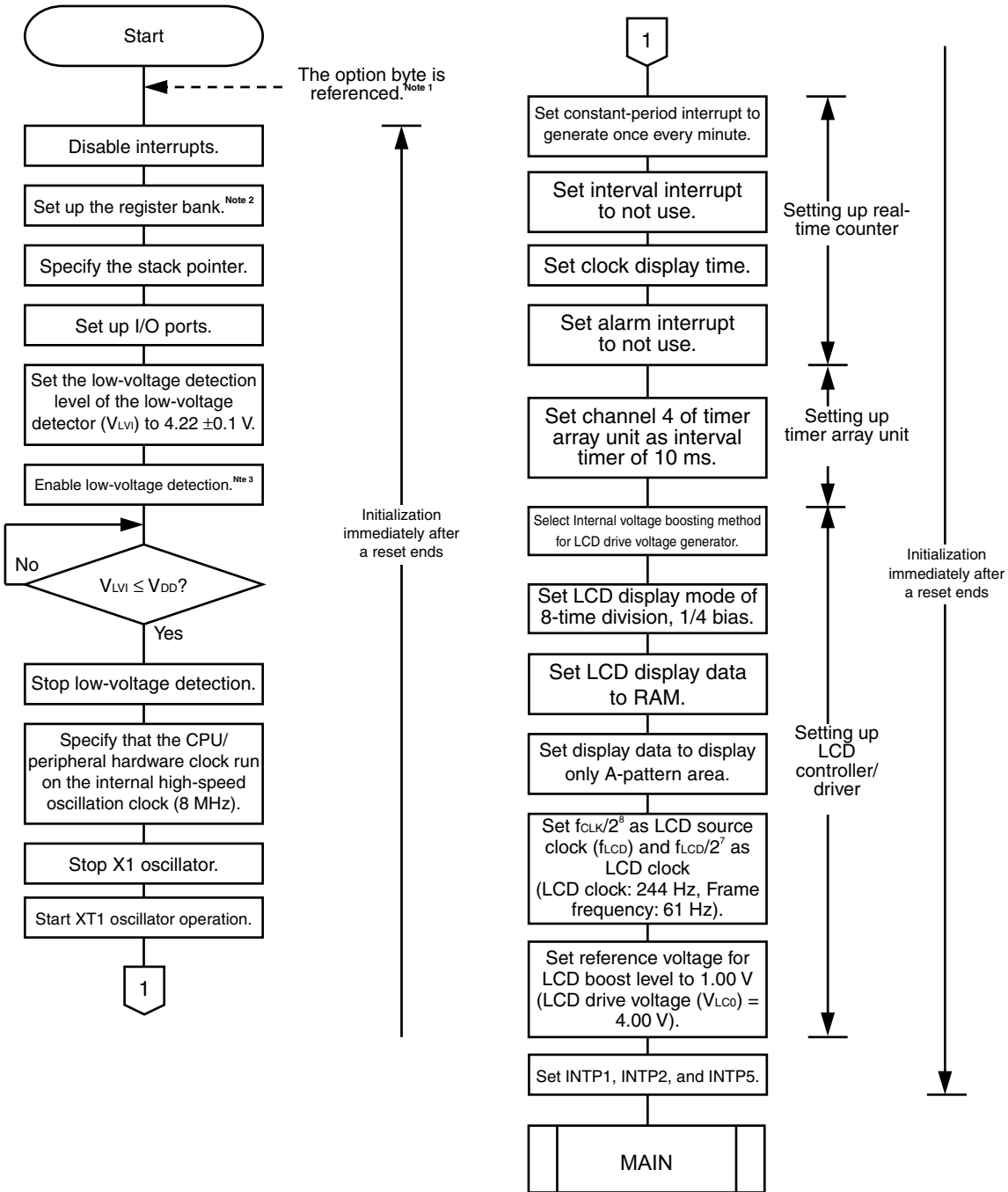
- Setting up I/O ports
 - Setting SEG4 to SEG53^{Note} for segment output
 - Setting COM0 to COM7 for common output
 - Setting INTP1, INTP2, and INTP5 to be detected at the falling edge
- Securing a supply voltage of 4.2 V or more by using the function of low-voltage detector
- Specifying that the CPU/peripheral hardware clock run on the internal high-speed oscillation clock (8 MHz)
- Stopping the X1 oscillator
- Starting operation of the XT1 oscillator
- Setting up the real-time counter
 - Setting the real-time counter to generate the constant-period interrupt once every minute
 - Setting to not use the interval interrupt of the real-time counter
 - Setting the current time to 00:00:00 a.m.
 - Setting to not use the alarm interrupt
- Setting up timer array unit
 - Setting channel 4 in interval timer mode of about 10 ms
- Setting up LCD controller/driver
 - Selecting Internal voltage boosting method for the LCD drive voltage generator
 - Setting an LCD display mode of 8-time division and 1/4 bias
 - Setting LCD display data in the RAM area
 - Setting to display the data of only the A-pattern area
 - Setting $f_{CLK}/2^8$ as the LCD source clock (f_{LCD}) and $f_{LCD}/2^7$ as the LCD clock (LCD clock: 244 Hz, frame frequency: 61 Hz)
- Setting 1.00 V (LCD drive voltage (V_{LCD}) = 4.00 V) as the reference voltage of the LCD boost level
- Setting the falling edge as the valid edge of INTP1, INTP2, and INTP5



Note These settings are for the 78K0R/LH3. When the 78K0R/LF3 is used, SEG4 to SEG30 are used for segment output. When the 78K0/LG3 is used, SEG4 to SEG39 are used for segment output.

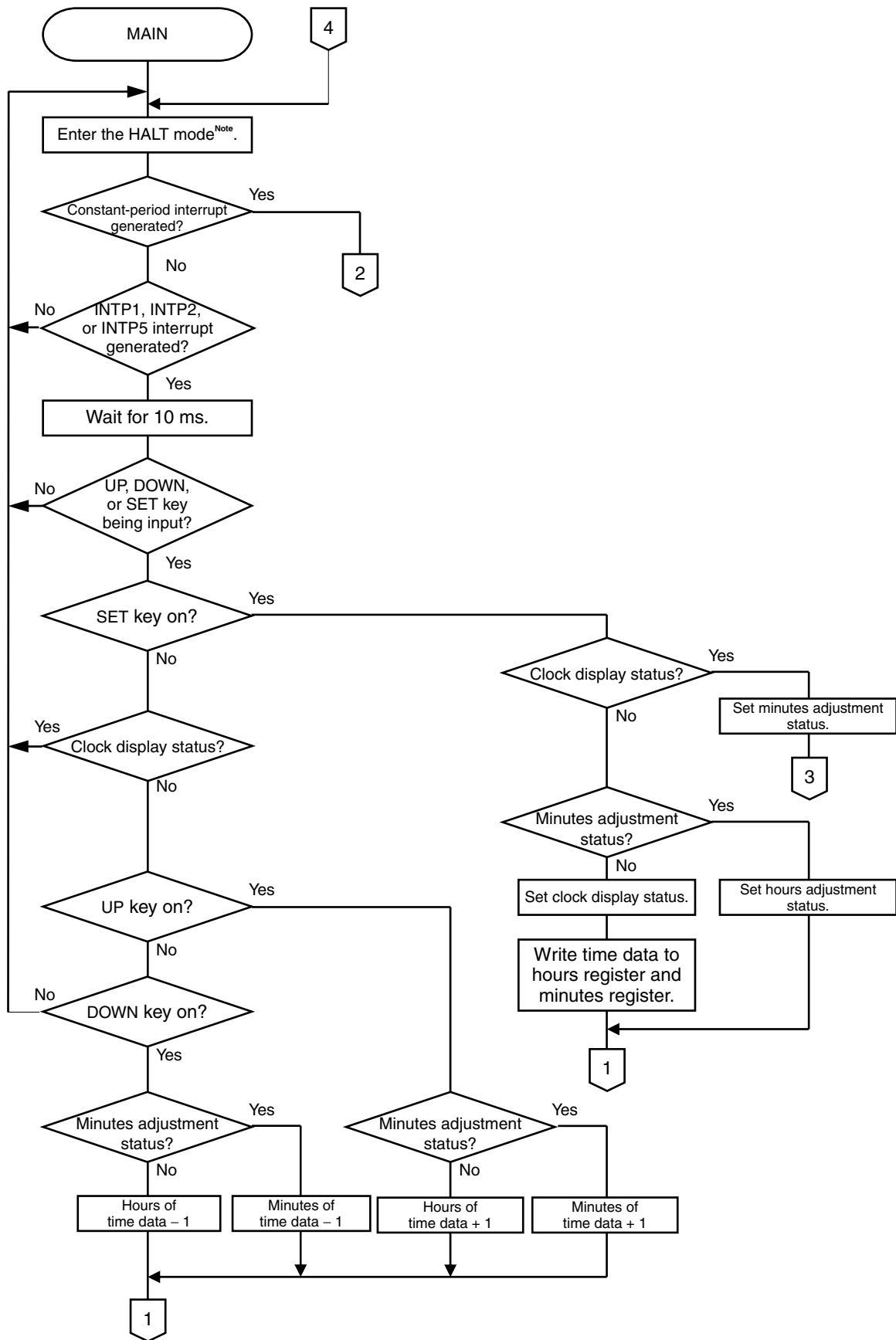
3.4 Flow Chart

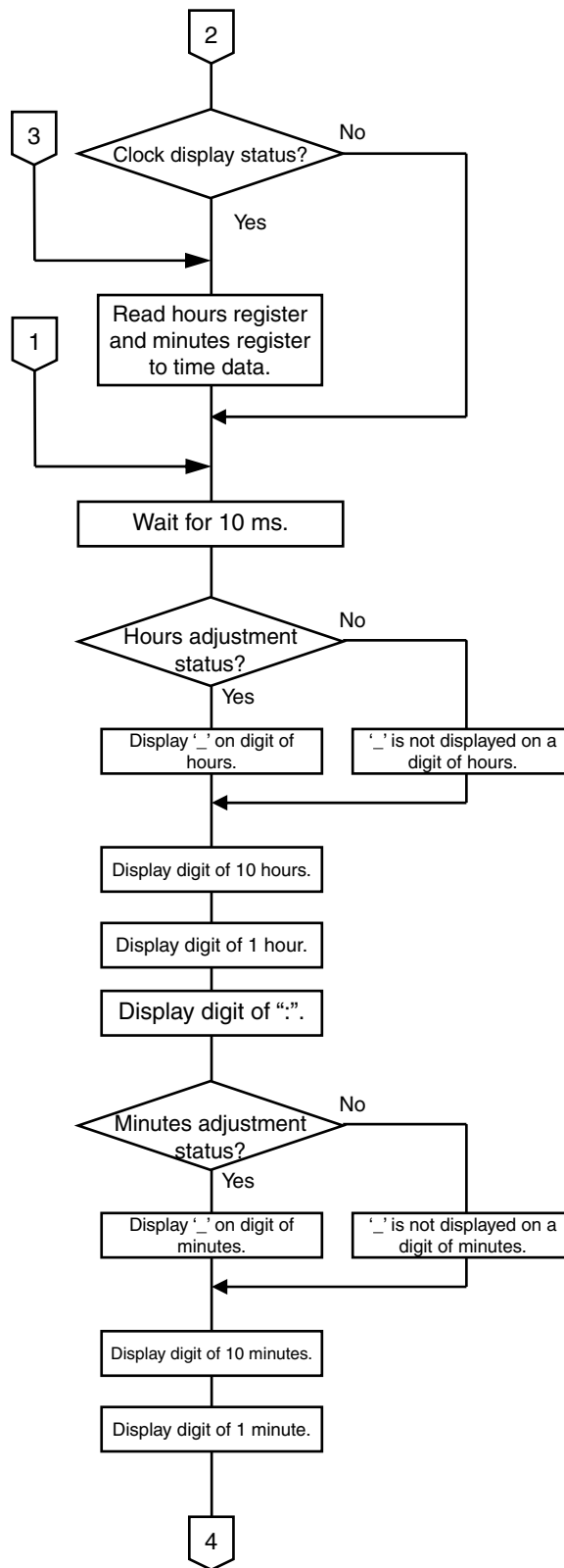
A flow chart for the sample program is shown below.



- Notes 1.** The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:
- Disabling the watchdog timer
 - Setting the internal high-speed oscillator frequency to 8 MHz
 - Disabling LVI from being started by default
 - Enabling on-chip debug to operate
- 2.** The general-purpose registers of the 78K0R/Lx3 Series microcontrollers are configured in four register banks so that the registers used for normal processing and those used when an interrupt occurs can be changed on a bank basis in order to create an efficient program. In this sample program, only register bank 0 is used.
- 3.** The low-voltage detector is enabled, and then the system is made to wait at least 10 μ s until the low-voltage detector stabilizes.

Caution With the sample program of the C language version, the settings of register banks and stack pointer are not described in the source program (main.c) because they are made by the start-up routine. For details of the start-up routine, refer to the CC78K0R Operation User's Manual.





Note The device is released from the HALT mode by the constant-period interrupt of the real-time counter or generation of interrupt INTP1, INTP2, or INTP5 (by detection of the falling edge of P30 to P32).

CHAPTER 4 SETTING METHODS

This chapter describes how to set up peripheral hardware, real-time counter, and LCD controller/driver. It also provides software coding examples.

For other initial settings, refer to the **78K0R/Kx3 Sample Program (Initial Settings) LED Lighting Switch Control Application Note**.

For how to set register, refer to the User's Manual.

For assembler instructions, refer to the **78K0R Series Instructions User's Manual**.

4.1 Setting to Use Peripheral Hardware

Use of the peripheral hardware is specified by using the following register.

- Peripheral enable register 0 (PER0)

[Example of procedure for setting to use peripheral hardware]

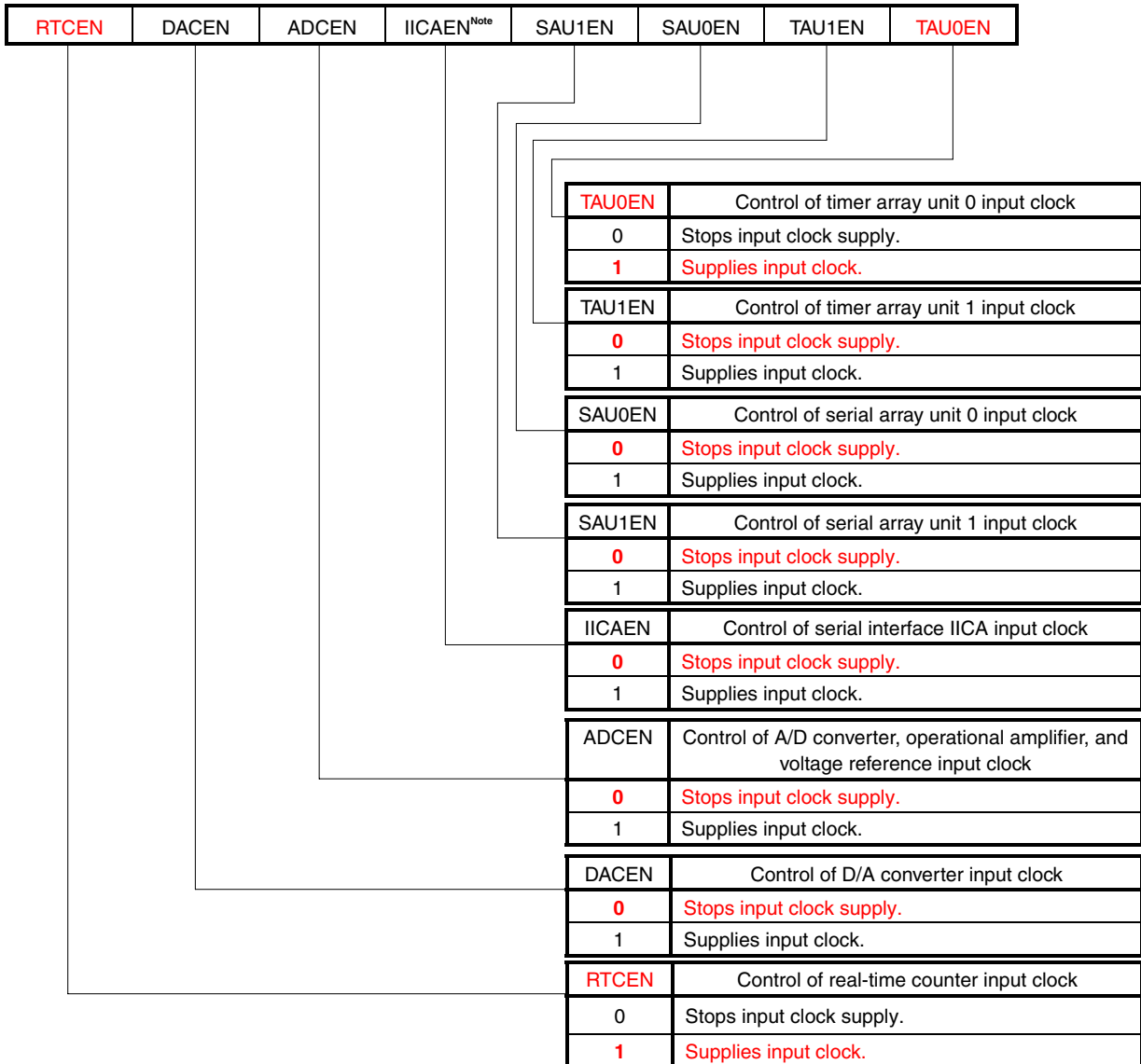
<1> Set bit 7 (RTCEN) of peripheral enable register 0 (PER0) to 1.

<2> Set bit 0 (TAU0EN) of peripheral enable register 0 (PER0) to 1.

(1) Peripheral enable register 0 (PER0)

This register is used to enable or disable use of each peripheral hardware macro. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise. PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 4-1-1. Format of Peripheral Enable Register 0 (PER0)



Note 78K0R/LG3, 78K0R/LH3 only

Remark The values written in red in the above figure are specified in this sample program.

4.2 Setting Up Real-Time Counter

The following seven registers are used to set up the real-time counter.

- Peripheral enable register 0 (PER0)
- Real-time counter control register 0 (RTCC0)
- Real-time counter control register 1 (RTCC1)
- Real-time counter control register 2 (RTCC2)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)

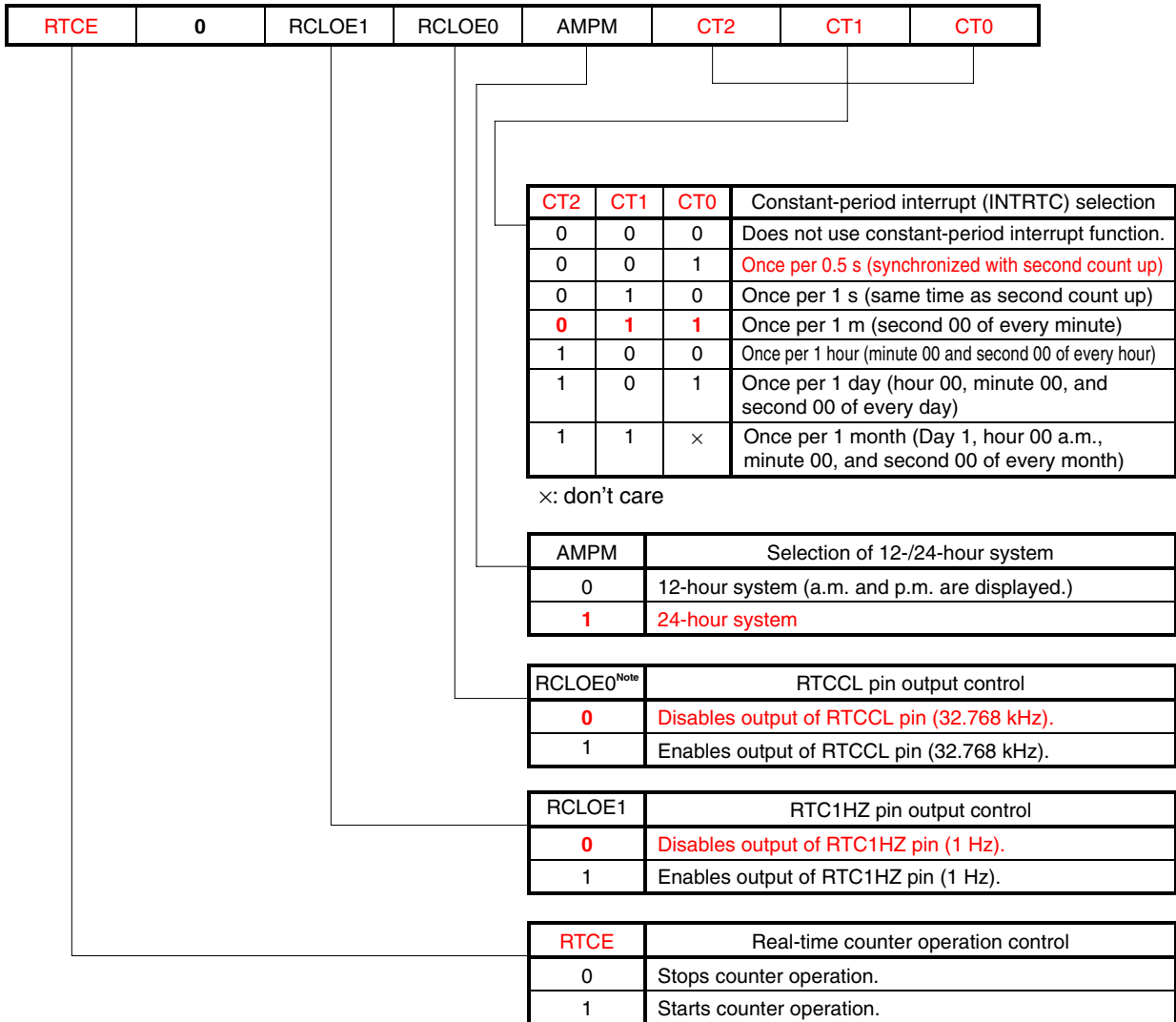
[Example of procedure for setting up real-time counter to play back sound data]

- <1> Set bit 7 (RTCEN) of peripheral enable register 0 (PER0) to 1 (see 4.1).
- <2> Set the real-time counter to generate the constant-period interrupt once every minute.
- <3> Set the real-time counter not to generate the interval interrupt.
- <4> Set time of starting counting to SEC, MIN, and HOUR.
- <5> Set not to generate an interrupt when the current time matches the alarm time.

(1) Real-time counter control register 0 (RTCC0)

This register is an 8-bit register that is used to start or stop the real-time counter operation, control the RTCCL and RTC1HZ pins, and set a 12- or 24-hour system and the constant-period interrupt function. RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 4-2-1. Format of Real-Time Counter Control Register 0 (RTCC0)



Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, the 32.768 kHz and 1 Hz output signals may become glitch.

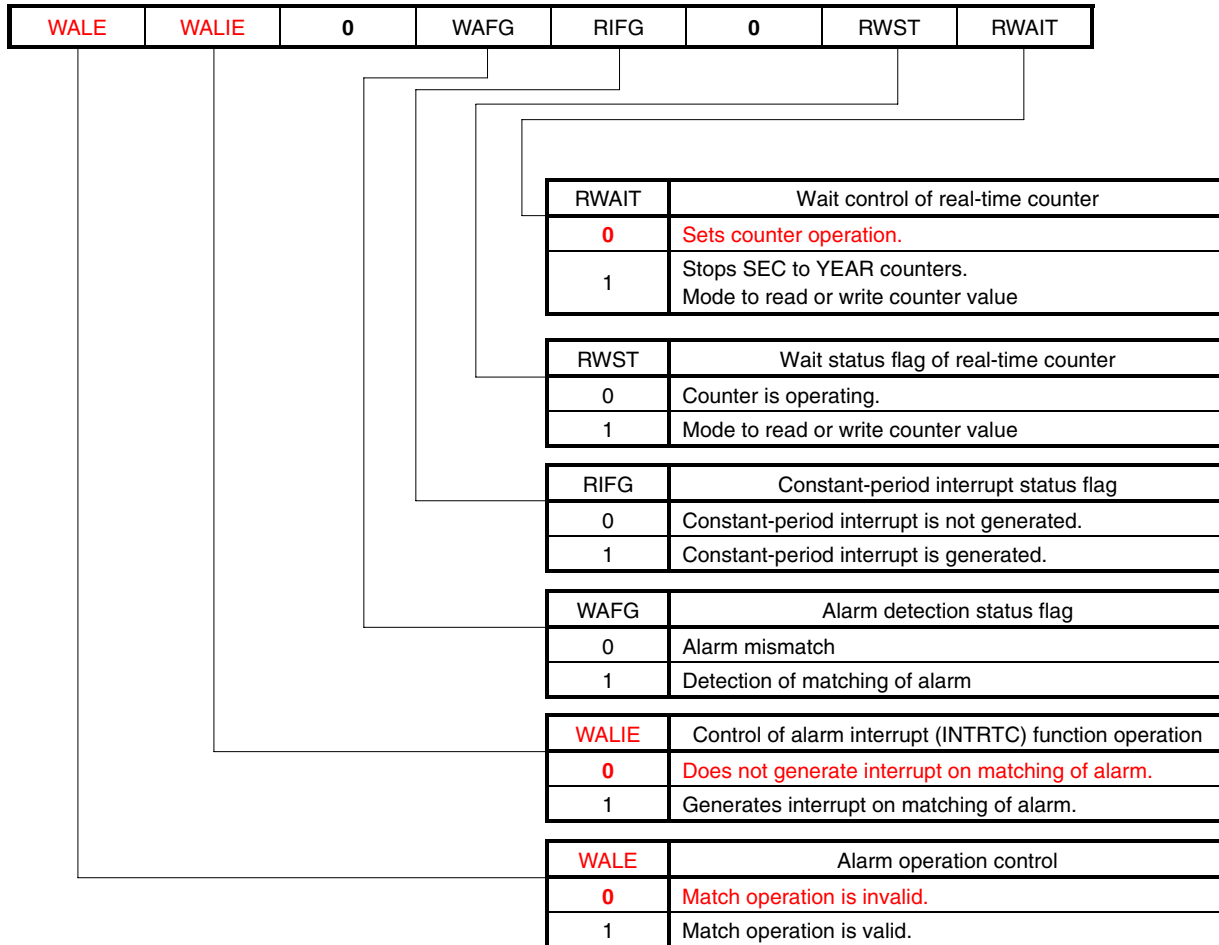
Remark The values written in red in the above figure are specified in this sample program.

(2) Real-time counter control register 1 (RTCC1)

This register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

RTCC1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 4-2-2. Format of Real-Time Counter Control Register 1 (RTCC1)



Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to the RIFG and WAFG flags, be sure to use an 8-bit manipulation instruction. At this time, set 1 to the RIFG and WAFG flags to invalidate writing and not to clear the RIFG and WAFG flags during writing. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.

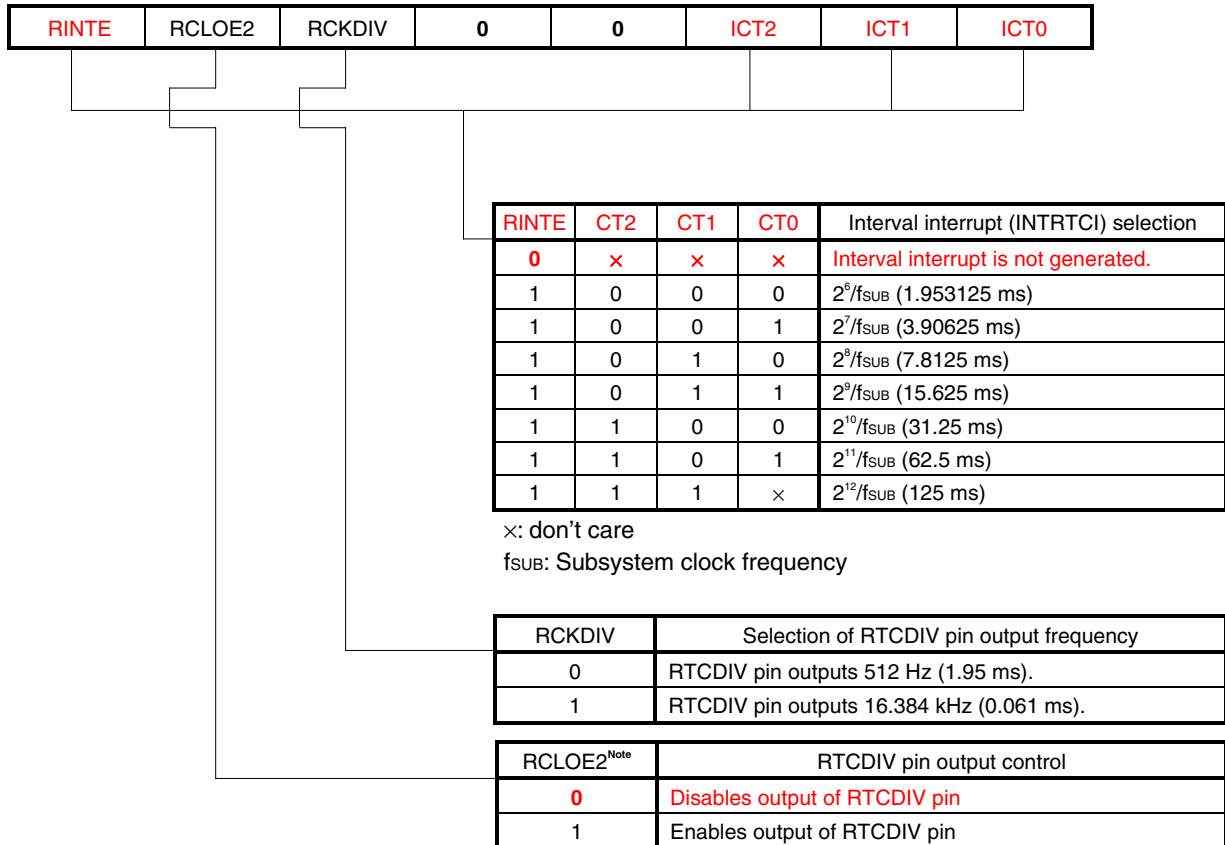
Remarks

1. The values written in red in the above figure are specified in this sample program.
2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

(3) Real-time counter control register 2 (RTCC2)

This register is an 8-bit register that is used to control the interval interrupt function and the RTCDIV pin. RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 4-2-3. Format of Real-Time Counter Control Register 2 (RTCC2)



Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Cautions 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of f_{SUB} and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of f_{SUB} may be generated.
3. After the real-time counter starts operating, the output width of the RTCDIV pin may be shorter than as set during the first interval period.

Remark The values written in red in the above figure are specified in this sample program.

(4) Second count register (SEC)

This register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds. It counts up when an overflow occurs from the sub-count register (RSUBC)^{Note} that counts 1 second on a clock of 32.768 kHz.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, it returns to the normal value 1 period later.

Figure 4-2-4. Format of Second Count Register (SEC)

0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1
---	-------	-------	-------	------	------	------	------

Note For details of the sub-count register (RSUBC), refer to the User's Manual.

Remark In this program, 00H is set as the default value.

(5) Minute count register (MIN)

This register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes. It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, it returns to the normal value 1 period later.

Figure 4-2-5. Format of Minute Count Register (MIN)

0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1
---	-------	-------	-------	------	------	------	------

Remark In this program, 00H is set as the default value.

(6) Hour count register (HOUR)

This register is an 8-bit register that takes a value of 00 to 23, or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 23, or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, it returns to the normal value 1 period later.

Figure 4-2-6. Format of Hour Count Register (HOUR)

0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1
---	---	--------	--------	-------	-------	-------	-------

Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected). For details, see table below.

24-Hour Display (AMPM Bit = 1)		12-Hour Display (AMPM Bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

Remark In this program, 00H is set as the default value.

4.3 Setting Up LCD Controller/Driver

The following seven registers are used to control the LCD controller/driver.

- LCD mode register (LCDMD)
- LCD display mode register (LCDM)
- LCD clock control register 0 (LCDC0)
- LCD boost level control register (VLCD)
- Port function register (PFALL)
- Segment enable register (SEGEN)
- Input switch control register (ISC)

[Example of procedure for setting up LCD controller/driver]

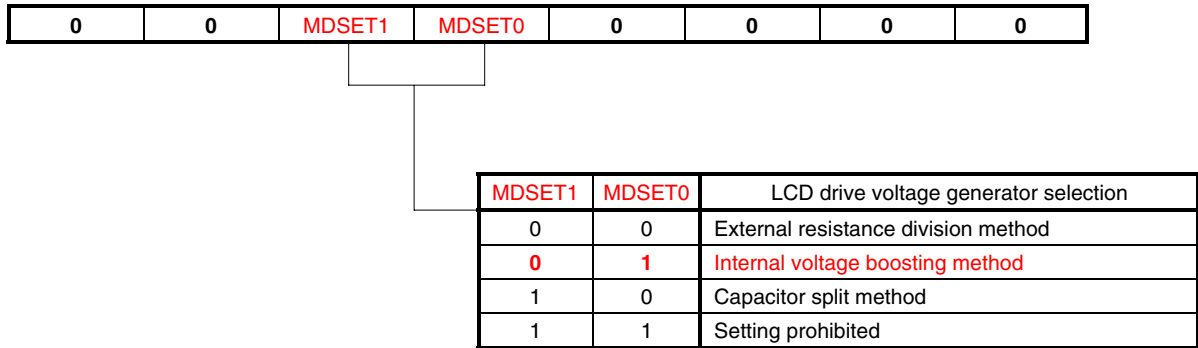
- <1> Select the Internal voltage boosting method by using the LCD mode register (LCDMD).
- <2> By using the segment enable register (SEGEN), enable the segment output pin to output.
- <3> Set up the LCD display mode register (LCDM) to select 8-time division and 1/4 bias, to display only the A-pattern area, and to output the ground level to the segment/common pin.
- <4> Select $f_{CLK}/2^8$ as the LCD source clock (f_{LCD}) and $f_{CLK}/2^7$ as the LCD clock by using the LCD clock control register 0 (LCDC0) (LCD clock: 244 Hz, frame frequency: 61 Hz).
- <5> Set the reference voltage to 1.00 V (LCD drive voltage (V_{LCO}) = 4.00 V) by using the LCD boost level control register.
- <6> Wait for the reference voltage setup time (2 ms (min.)) after setting up the VLCD register.
- <7> Set all the port/segment output alternate-function pins in the segment output mode by using the port function register (PFALL).
- <8> Prohibit input from P50, P52, and P53 by using the input switch control register (ISC).
- <9> Start the voltage boost circuit operation by setting the VLCON bit.
- <10> Wait for the boost wait time (500 ms (min.)) after setting the VLCON bit.
- <11> Set the SCOC bit.
- <12> Set the LCDON bit.

(1) LCD mode register (LCDMD)

This register is used to set the LCD drive voltage generator.

LCDMD can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 4-3-1. Format of LCD Mode Register (LCDMD)



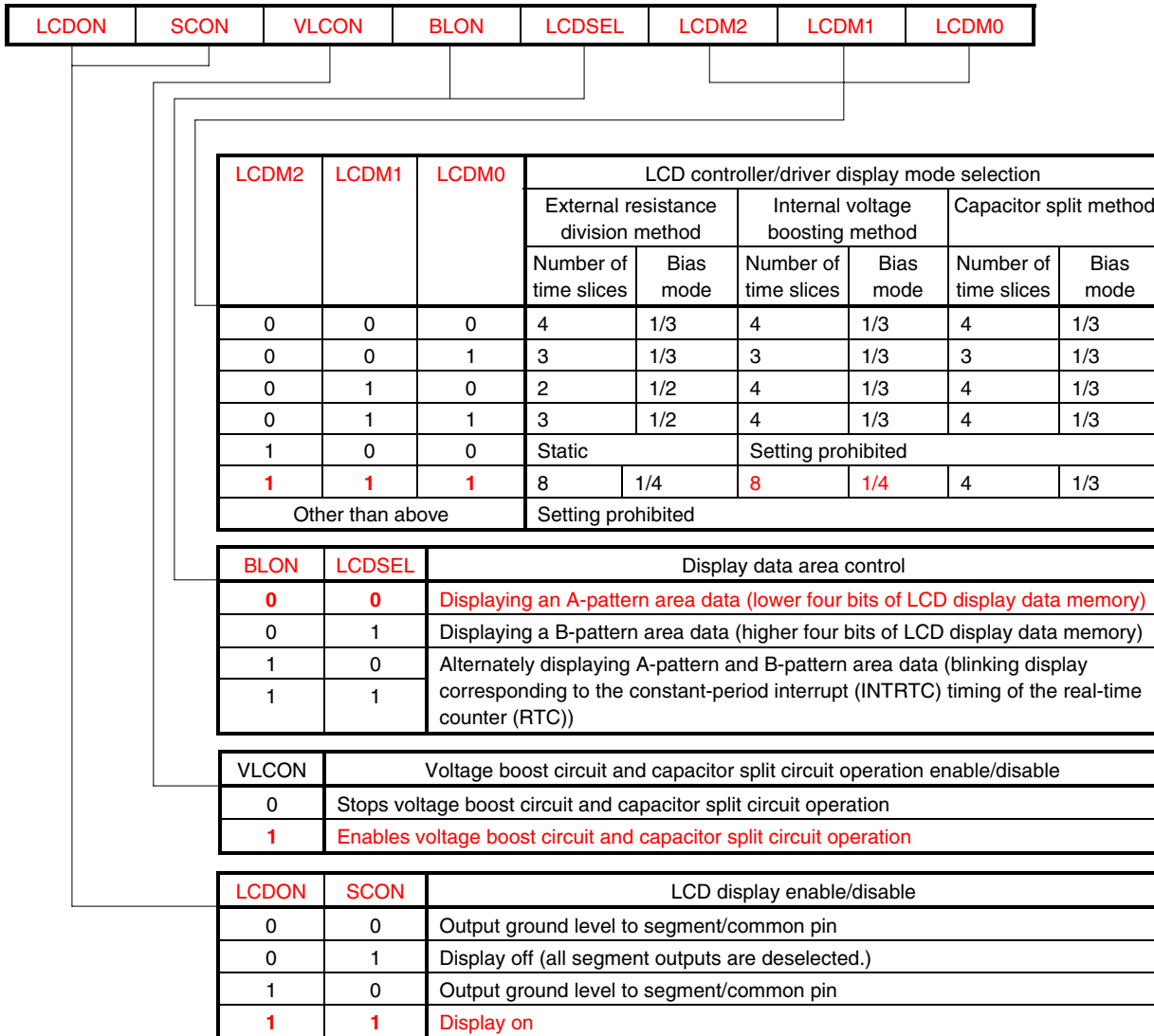
Remark The values written in red in the above figure are specified in this sample program.

(2) LCD display mode register (LCDM)

This register enables/disables display operation, enables/disables voltage boost circuit and capacitor split circuit operation, and sets the display data area and the display mode.

LCDM can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 4-3-2. Format of LCD Display Mode Register (LCDM)



- Cautions**
1. When LCD display is not performed or necessary, set SCOC and VLCON to 0, in order to reduce power consumption
 2. When the external resistance division method has been set (MDSET1 = MDSET0 = 0), do not set VLCON to 1.
 3. To stop voltage boost circuit and capacitor split circuit operation, be sure to set SCOC and LCDON to 0 and then set VLCON to 0.
 4. Set BLON and LCDSEL to 0 when 8 has been selected as the number of time slices for the display mode.

Remark The values written in red in the above figure are specified in this sample program.

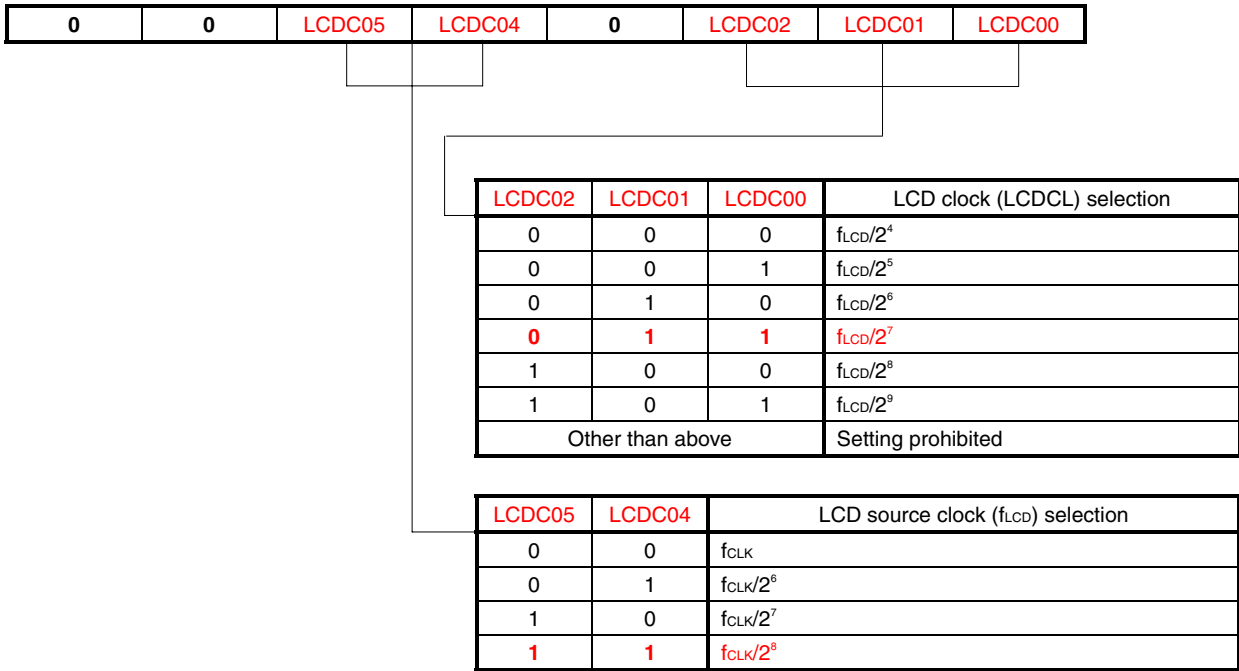
(3) LCD clock control register 0 (LCDC0)

This register specifies the LCD source clock and LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

LCDC0 can be set by an 8-bit memory manipulation instruction.

Figure 4-3-3. Format of LCD Clock Control Register 0 (LCDC0)



f_{CLK} : CPU/peripheral hardware clock frequency

Cautions 1. Be sure to set bits 3, 6, and 7 to “0”.

2. Set the LCD clock (LCDCL) to no more than 512 Hz when the internal voltage boosting method has been set.

Remark The values written in red in the above figure are specified in this sample program.

(4) LCD boost level control register (VLCD)

This register is used to select the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage can be selected from 20 stages. VLCD can be set by an 8-bit memory manipulation instruction.

Figure 4-3-4. Format of LCD Boost Level Control Register (VLCD)

0	0	0	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0
----------	----------	----------	--------------	--------------	--------------	--------------	--------------

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	Reference voltage selection (contrast adjustment)	V _{LC0} voltage	
						1/3 bias	1/4 bias
0	0	0	0	0	1.75 V	5.25 V	Setting prohibited ^{Note}
0	0	0	0	1	1.70 V	5.10 V	
0	0	0	1	0	1.65 V	4.95 V	
0	0	0	1	1	1.60 V	4.80 V	
0	0	1	0	0	1.55 V	4.65 V	
0	0	1	0	1	1.50 V	4.50 V	
0	0	1	1	0	1.45 V	4.35 V	
0	0	1	1	1	1.40 V	4.20 V	
0	1	0	0	0	1.35 V	4.05 V	
0	1	0	0	1	1.295 V	3.885 V	
0	1	0	1	0	1.25 V	3.75 V	5.00 V
0	1	0	1	1	1.20 V	3.60 V	4.80 V
0	1	1	0	0	1.15 V	3.45 V	4.60 V
0	1	1	0	1	1.10 V	3.30 V	4.40 V
0	1	1	1	0	1.05 V	3.15 V	4.20 V
0	1	1	1	1	1.00 V (default)	3.00 V	4.00 V
1	0	0	0	0	0.95 V	2.85 V	3.80 V
1	0	0	0	1	0.90 V	2.70 V	3.60 V
1	0	0	1	0	0.85 V	2.55 V	3.40 V
1	0	0	1	1	0.80 V	2.40 V	3.20 V
Other than above					Setting prohibited		

Note These settings are prohibited because V_{LC0} > 5.5 V.

- Cautions**
1. The VLCD setting is valid only when the voltage boost circuit is operating.
 2. Be sure to set bits 5 to 7 to “0”.
 3. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0)
 4. These values above may change after device evaluation.

Remark The values written in red in the above figure are specified in this sample program.

(5) Port function register (PFALL)

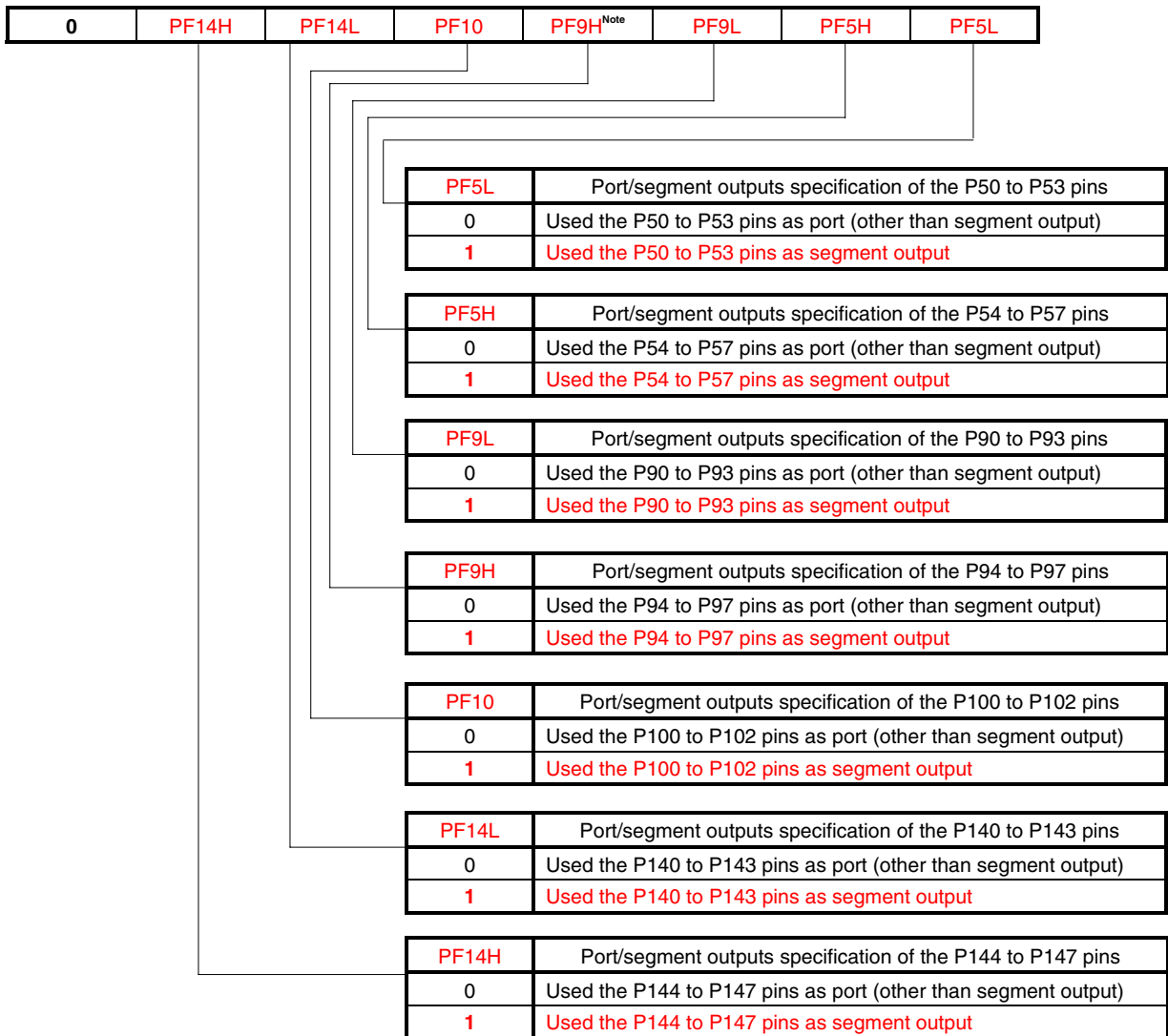
This register sets whether to use pins P50 to P57, P90 to P97, P100 to P102, and P140 to P147 as port pins (other than segment output pins) or segment output pins.

PFALL can be set by a 1-bit or 8-bit memory manipulation instruction.

Remark The port pins to be used alternatively with the segment output pins vary, depending on the product.

- 78K0R/LF3: P50 to P57, P90 to P92, P100, P140 to P147
- 78K0R/LG3: P50 to P57, P90 to P97, P100, P140 to P147
- 78K0R/LH3: P50 to P57, P90 to P97, P100 to P102, P140 to P147

Figure 4-3-5. Format of Port Function Register (PFALL)



Note 78K0R/LG3, 78K0R/LH3 only

Caution For 78K0R/LF3, be sure to set bits 3 and 7 to “0”. For 78K0R/LG3 and 78K0R/LH3, be sure to set bit 7 to “0”.

Remark The values written in red in the above figure are specified in this sample program.

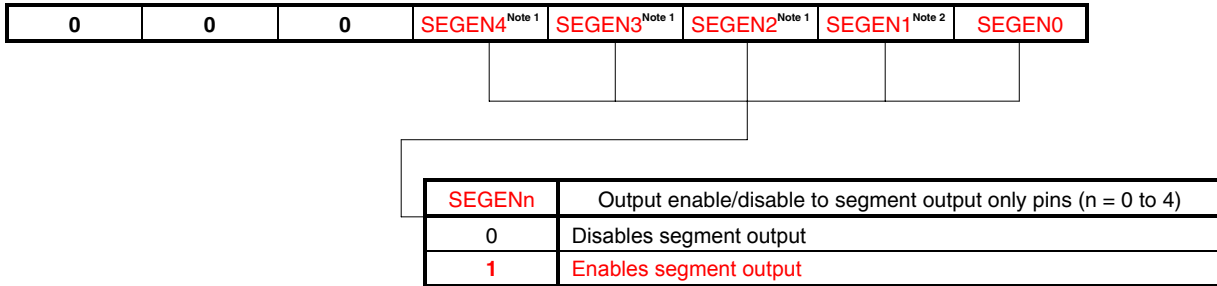
(6) Segment enable register (SEGEN)

This register is used to enable or disable segment output to segment output only pins. SEGEN can be set by a 1-bit or 8-bit memory manipulation instruction.

Remark The segment output only pins vary, depending on the product.

- 78K0R/LF3: SEG8 to SEG10
- 78K0R/LG3: SEG8 to SEG14
- 78K0R/LH3: SEG8 to SEG26

Figure 4-3-6. Format of Segment Enable Register (SEGEN)



- Notes**
1. These bits of the 78K0R/LF3 and 78K0R/LG3 are fixed to “0”.
 2. This bit of the 78K0R/LF3 is fixed to “0”.

- Cautions**
1. SEGEN can be written only once after reset release.
 2. For 78K0R/LF3, be sure to set bits 1 to 7 to “0”. For 78K0R/LG3, be sure to set bits 2 to 7 to “0”. For 78K0R/LH3, be sure to set bits 5 to 7 to “0”.

Remark The values written in red in the above figure are specified in this sample program.

The segment output only pins operated by SEGEN4 to SEGEN0 are as follows.

Table 4-3-1. Segment Output Only Pins Controlled by SEGEN4 to SEGEN0 Bits

SEGEN Register	Segment Output Only Pins		
	78K0R/LF3	78K0R/LG3	78K0R/LH3
SEGEN4	–	–	SEG24 to SEG26 pins
SEGEN3	–	–	SEG20 to SEG23 pins
SEGEN2	–	–	SEG16 to SEG19 pins
SEGEN1	–	SEG12 to SEG14 pins	SEG12 to SEG15 pins
SEGEN0	SEG8 to SEG10 pins	SEG8 to SEG11 pins	SEG8 to SEG11 pins

(7) Input switch control register (ISC)

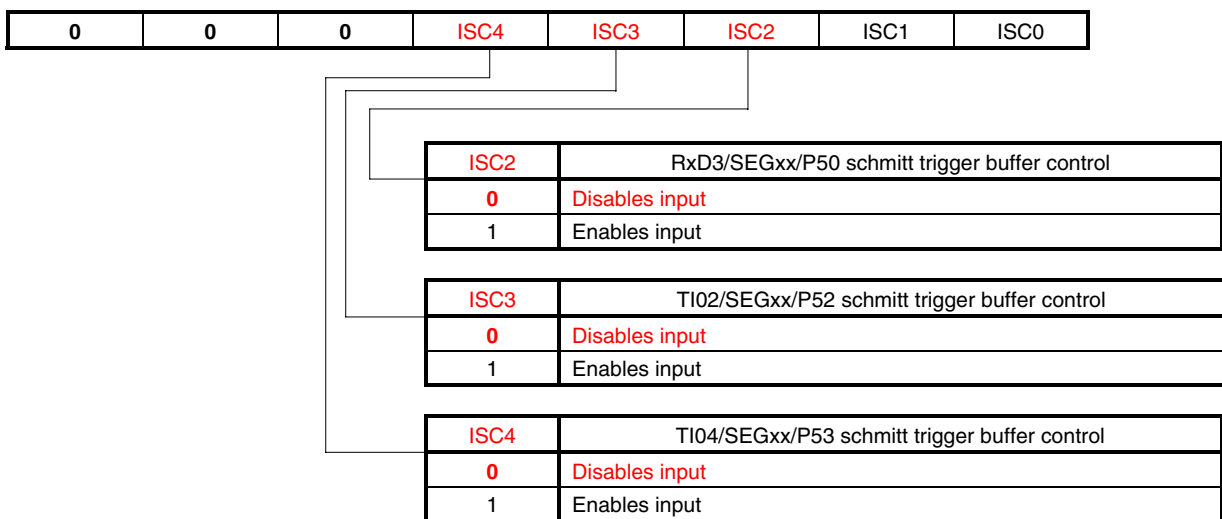
The segment output pins to be used alternatively with the TI04, TI02, and RxD3 pins are internally connected with a Schmitt trigger buffer. To use these pins as segment outputs, input to the Schmitt trigger buffer must be disabled, in order to prevent through-currents from entering.

ISC can be set by a 1-bit or 8-bit memory manipulation instruction.

Remark The segment output pins to be used alternatively with the TI02, TI04, and RxD3 pins vary, depending on the product.

- 78K0R/LF3: TI04/SEG27/P53, TI02/SEG28/P52, RxD3/SEG30/P50
- 78K0R/LG3: TI04/SEG36/P53, TI02/SEG37/P52, RxD3/SEG39/P50
- 78K0R/LH3: TI04/SEG50/P53, TI02/SEG51/P52, RxD3/SEG53/P50

Figure 4-3-7. Format of Input Switch Control Register (ISC)



Caution Be sure to set bits 7 to 5 to “0”.

- Remarks**
1. The values written in red in the above figure are specified in this sample program.
 2. This register must be set up only when the 78K0R/L3 is used.
 3. Bits 0 and 1 of ISC are not used with the LCD controller driver.

To use the TI04/SEGxx/P53, TI02/SEGxx/P52, and RxD3/SEGxx/P50 pins, set the PF5L and ISCn (n = 2 to 4) bits as follows, according to the function to be used.

Table 4-3-2. Setting Example of PF5L and ISCn

PF5L	ISCn	Pin Function
0	0	Port output (default)
0	1	Port input, timer input, or serial data input
1	0	Segment output
1	1	Setting prohibited

4.4 Software Coding Examples

As a software coding example, initial settings for clock display, clock adjustment processing, and clock display updating processing performed by the source program of the 78K0R/LH3 are shown below.

(1) Assembly language

<1> Initial settings for clock display

```

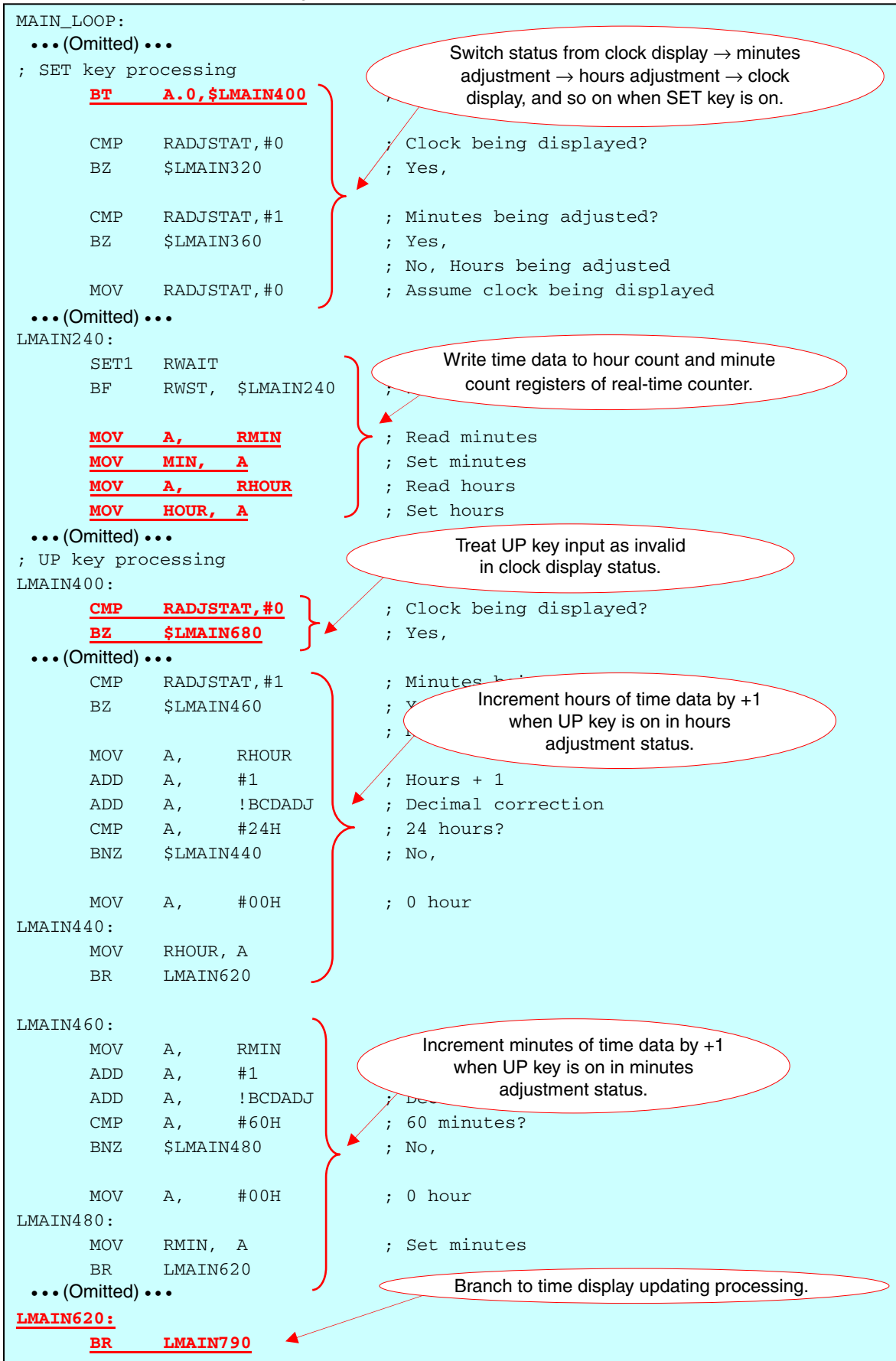
XMAIN CSEG UNIT
IRESET:
... (Omitted) ...
MOV PM3, #11100111B ; Set P30 to P32 as input port for SET, UP, and DOWN keys.
MOV PU3, #00000111B ; Set real-time counter to supply input clock.
... (Omitted) ...
MOV PER0, #10000000B ; Set real-time counter to generate constant-period interrupt once per minute. Set clock in 24-hour system.
MOV RTCC0, #00001011B ; Set not to generate interval interrupt.
MOV RTCC2, #00000111B
MOV SEC, #00H ; Set time of starting counting.
MOV MIN, #00H ; Minute: 00
MOV HOUR, #00H ; Disable alarm interrupt.
CLR1 WALE ; Set LCD in internal voltage boosting method.
CLR1 WALIE ; Does not generate interrupt on matching of alarm
... (Omitted) ...
MOV LCDMD, #00010000B ; LCD mode register
MOV SEGEN, #00011111B ; Segment output control register
MOV C, #36H ; LCD display data memory size
MOVW DE, #LOWW SEG0 ; LCD display data memory start address
CLRB A
HINI520:
MOV [DE], A ; Set
INCW DE ; Setting
DEC C ; NO,
BNZ $HINI520
MOV LCDM, #00000111B ; LCD
MOV LCDC0, #00110011B ; Set reference voltage for LCD boost level to 1.00 V (LCD drive voltage (VLC0) = 4.00 V).
MOV VLCD, #00001111B ; LCD boost level control register
MOV PFALL, #01111111B ; Port I/O alternate-function control register
MOV ISC, #00000000B ; Input switch control register
... (Omitted) ...
MOV EGN0, #00101110B ; Set to generate interrupt request at falling edge of P30 to P32.

```

The diagram illustrates the assembly code for initial settings for clock display, with callouts explaining the purpose of each instruction:

- MOV PM3, #11100111B**: Set P30 to P32 as input port for SET, UP, and DOWN keys.
- MOV PU3, #00000111B**: Set real-time counter to supply input clock.
- MOV PER0, #10000000B**: Set real-time counter to generate constant-period interrupt once per minute. Set clock in 24-hour system.
- MOV RTCC0, #00001011B**: Set not to generate interval interrupt.
- MOV RTCC2, #00000111B**: (No callout)
- MOV SEC, #00H**: Set time of starting counting.
- MOV MIN, #00H**: Minute: 00
- MOV HOUR, #00H**: Disable alarm interrupt.
- CLR1 WALE**: Set LCD in internal voltage boosting method.
- CLR1 WALIE**: Does not generate interrupt on matching of alarm
- MOV LCDMD, #00010000B**: LCD mode register
- MOV SEGEN, #00011111B**: Segment output control register
- MOV C, #36H**: LCD display data memory size
- MOVW DE, #LOWW SEG0**: LCD display data memory start address
- CLRB A**: (No callout)
- HINI520: MOV [DE], A**: Set
- INCW DE**: Setting
- DEC C**: NO,
- BNZ \$HINI520**: (No callout)
- MOV LCDM, #00000111B**: LCD
- MOV LCDC0, #00110011B**: Set reference voltage for LCD boost level to 1.00 V (LCD drive voltage (V_{LC0}) = 4.00 V).
- MOV VLCD, #00001111B**: LCD boost level control register
- MOV PFALL, #01111111B**: Port I/O alternate-function control register
- MOV ISC, #00000000B**: Input switch control register
- MOV EGN0, #00101110B**: Set to generate interrupt request at falling edge of P30 to P32.

<2> Clock adjustment processing



```

; DOWN key processing
LMAIN500:
    BT    A.2, $LMAIN680    ; DOWN key on? No,
    ; ... (Omitted) ...
    MOV   A,    RHOURL
    SUB   A,    #1
    SUB   A,    !BCDADJ    ; Decimal adjust
    BNC   $LMAIN540
    ; 23 hours?, No,
    MOV   A,    #23H
    ; 23 hours
LMAIN540:
    MOV   RHOURL, A
    BR    LMAIN620
LMAIN560:
    MOV   A,    RMIN
    SUB   A,    #1
    SUB   A,    !BCDADJ    ; Decimal adjust
    BNC   $LMAIN580
    ; 59 minutes?, No,
    MOV   A,    #59H
    ; 59 minutes
LMAIN580:
    MOV   RMIN, A
    ; ... (Omitted) ...
LMAIN620:
    BR    LMAIN790
    
```

Treat DOWN key input as invalid in clock display status.

Decrement hours of time data by -1 when DOWN key is on in hours adjustment status.

Decrement minutes of time data by -1 when DOWN key is on in minutes adjustment status.

Branch to time display updating processing.

<3> Clock display updating processing

```

MAIN_LOOP:
... (Omitted) ...

LMAIN720:
  CLR1  RTCIF                ; Clear RTCIF
  CMP   RADJSTAT,#0         ; Hours being adjusted?
  BNZ   $LMAIN790          ; No,

LMAIN740:
  SET1  RWAIT               ; RTC value read/write mode
  BF    RWST, $LMAIN740    ; Wait for RTC read/write mode

  MOV   A, MIN              ; Read minutes
  MOV   RMIN, A            ; Save minutes to display buffer

  MOV   A, HOUR            ; Read hours
  MOV   RHOURL, A         ; Save hours to display buffer

LMAIN780:
  CLR1  RWAIT               ; Start RTC count operation
  BT    RWST, $LMAIN780    ; Wait for start of RTC count operation

LMAIN790:
  MOV   B, #0              ; Hours being adjusted?
  CMP   RADJSTAT,#2       ; Hours being adjusted?
  BNZ   $LMAIN820          ; No,
  MOV   B, #1              ; '_' displayed

LMAIN820:
  MOV   A, RHOURL          ; Digit of 10 hours
  AND   A, #0F0H          ; Digit of 10 hours = 0?
  SHR   A, 4              ; No,
  CMP0  A                  ; Digit of 10 hours = 0?
  BNZ   $LMAIN860          ; No,

  MOV   A, #0AH           ; ' ' (0-suppression)

LMAIN860:
  MOVW  DE, #LOWW SEG4    ; LCD display position
  CALL  !!SLCDNUM        ; Display digit of 10 hours

  MOV   A, RHOURL          ; Digit of 1 hour
  AND   A, #0FH           ; LCD display position
  MOVW  DE, #LOWW SEG10   ; Display digit of 1 hour

  MOV   B, #0              ; '.' not displayed
  MOV   A, #0BH           ; '.'
  MOVW  DE, #LOWW SEG16   ; LCD display position
  CALL  !!SLCDNUM        ; Display digit of 1 hour
  
```

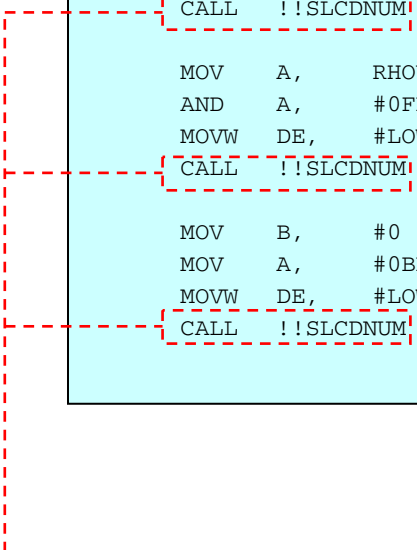
Read time data from hour count and minute count registers of real-time counter in clock display status.

Append '_' when hours are displayed in hours adjustment status.

Display digit of 10 hours. Display zero-suppression of 0 to 9 hours.

Display digit of 1 hour.

Display digit of ".".



```

MOV B, #0
CMP RADJSTAT, #1
BNZ $LMAIN900
MOV B, #1
LMAIN900:
MOV A, RMIN
AND A, #0F0H
SHR A, 4
MOVW DE, #LOWW SEG22
CALL !!SLCDNUM
MOV A, RMIN
AND A, #0FH
MOVW DE, #LOWW SEG28
CALL !!SLCDNUM
BR MAIN_LOOP

```

Append '_' when minutes are displayed in minutes adjustment status.

Display digit of 10 minutes.

Display digit of 1 minute.

```

SLCDNUM:
ADD A, A
MOV X, A
ADD A, A
ADD X, A
CLRB A
ADDW AX, #LOWW TCHRDATA
MOVW HL, AX
MOV C, #6
JLCDN220:
MOV A, [HL]
CMP0 B
BZ $JLCDN260
OR A, #1000000B
JLCDN260:
MOV [DE], A
INCW HL
INCW DE
DEC C
BNZ $JLCDN220
RET

```

Calculate value and read numeric font from RAM.

Append '_' in hours and minutes adjustment statuses.

Display font data.

(2) C language

<1> Initial settings for clock display

```

void hdwinit(void){
    ... (Omitted) ...
    PM3 = 0b11100111; /* Specific ports */
    PU3 = 0b00000111; /* Connect P30 to P32 */
    ... (Omitted) ...
    PER0 = 0b10000000;
    RTCC0 = 0b00001011;
    RTCC2 = 0b00000111;
    SEC = 0x00;
    MIN = 0x00; /* Minute: 00 */
    HOURL = 0x00; /* Hour: 00 */
    WALE = 0;
    WALIE = 0;
    ... (Omitted) ...
    LCDMD = 0b00010000; /* LCD mode register */
    SEGEN = 0b00011111; /* Segment enable register */
    for(count = 0, wkptr = &SEG0; count < 0x36; count++, wkptr++){
        *wkptr = 0x00; /* Clear the LCD display */
    }
    LCDM = 0b00000111; /* LCD mode register */
    LCDC0 = 0b00110011; /* LCD clock control register */
    VLCD = 0b00001111; /* LCD boost level register */
    PFALL = 0b01111111; /* Input pin */
    ISC = 0b00000000; /* Input pin */
    ... (Omitted) ...
    EGNO = 0b00100110; /* Falling edge of P30 to P32 valid */
}
    
```

Annotations:

- Set P30 to P32 as input port for SET, UP, and DOWN keys.
- Set real-time counter to supply input clock.
- Set real-time counter to generate constant-period interrupt once per minute. Set clock in 24-hour system.
- Set not to generate interval interrupt.
- Set time of starting counting.
- Disable alarm interrupt.
- Set LCD in internal voltage boosting method.
- Enable segment pins to output.
- Initialize LCD display data.
- Select 8-time slice, 1/4 bias as display mode, and set to display only A-pattern area and to output ground level to segment/common pins.
- Select $f_{CLK}/2^{15}$ as LCD clock.
- Set reference voltage for LCD boost level to 1.00 V (LCD drive voltage (V_{LCO}) = 4.00 V).
- Set port/segment alternate-function pin in segment output mode.
- Set to generate interrupt request at falling edge of P30 to P32.

<2> Clock adjustment processing

```

void main(void)
{
    ... (Omitted) ...

    if((RTCIF == 1) && (ucAdjStatus == 0x00)){ /* Lapse of 1 minute during clock display */
        RTCIF = 0; /* Clear constant-period interrupt request */
        if(ucAdjStatus == 0x00){ /* Clock being displayed? */
            RWAIT = 1; /* Set RTC value read/write mode */
            while(RWST == 0){ /* Wait for RTC read/write mode */
                RWAIT = 1; /* Set RTC value read/write mode */
            }

            ucMinute = MIN; /* Read minutes */
            ucHour = HOUR; /* Read hours */

            RWAIT = 0; /* Set to start RTC count operation */
            while(RWST == 1){ /* Wait for start of RTC count operation */
                RWAIT = 0; /* Set to start RTC count operation */
            }
        }
    }

    ... (Omitted) ...

    if(ucKeyin == (P3 & 0b00000111)){ /* Key status stabilized */

        switch(ucKeyin){ /* Distribution by input key */
            case 0b00000110: /* SET key */
                switch(ucAdjStatus){
                    /* Distribution by clock adjustment status */
                    case 0x00: /* Clock being displayed */
                        RWAIT = 1;
                        /* Set RTC value read/write mode */
                        while(RWST == 0){
                            /* Wait for RTC read/write mode */
                            RWAIT = 1;
                        }
                        /* Set RTC value read/write mode */
                        ucMinute = MIN; /* Read minutes */
                        ucHour = HOUR; /* Read hours */

                        RWAIT = 0;
                        /* Set to start RTC count operation */
                        while(RWST == 1){
                            /* Wait for start of RTC count operation */
                            RWAIT = 0;
                        }
                        /* Set to start RTC count operation */
                        ucAdjStatus = 0x01;
                        /* Assume minutes being adjusted */
                        break;

                    case 0x01: /* Minutes being adjusted */
                        ucAdjStatus = 0x02;
                        /* Assume hours being adjusted */
                        break;
                }
            }
        }
    }
}

```

Read time data from hour and minute count registers of real-time counter when 1 minute has passed in clock display status.

Read time data from hour and minute count registers of real-time counter in clock display status.

Switch status from clock display → minutes adjustment → hours adjustment → clock display, and so on when SET key is on.

```

        case 0x02: /* Hours being adjusted */
            ucAdjStatus = 0x00;
            /* Assume clock being displayed */
            RWAIT = 1;
            /* Set RTC value read/write mode */
            while(RWST == 0){
                /* Wait for RTC read/write mode */
                RWAIT = 1;
            }
            /* Set RTC value read/write mode */
            }

            MIN = ucMinute;    /* Set minutes */
            HOUR = ucHour;     /* Set hours */
            SEC = 0x00;        /* Reset seconds */

            RWAIT = 0;
            /* Set to start RTC count operation */
            while(RWST == 1){
                /* Wait for start of RTC count operation */
                RWAIT = 0;
            }
            /* Set to start RTC count operation */
            }
            break;

        ... (Omitted) ...

        case 0b00000101: /* UP key */
            switch(ucAdjStatus){
                /* Distribution by clock adjustment status */
                case 0x00: /* Clock being displayed */
                    break;

                case 0x01: /* Minutes being adjusted */
                    ucMinute = adbcdb(ucMinute, 1);
                    /* Minutes + 1 */

                    if(ucMinute == 0x60){
                        ucMinute = 0x00;
                        /* Carry correction */
                    }
                    break;

                case 0x02: /* Hours being adjusted */
                    ucHour = adbcdb(ucHour, 1);
                    /* Hours + 1 */

                    if(ucHour == 0x24){
                        ucHour = 0x00;
                        /* Carry correction */
                    }
                    break;

                ... (Omitted) ...
            }
    
```

Write time data to hour and minute count registers of real-time counter.

Treat UP key input as invalid in clock display status.

Increment minutes of time data by +1 when UP key is on in minutes adjustment status.

Increment hours of time data by +1 when UP key is on in hours adjustment status.

```
case 0b00000011: /* DOWN key */
switch(ucAdjStatus){
/* Distribution by clock adjustment status */
{
case 0x00: /* Clock being displayed */
break;

case 0x01: /* Minutes being adjusted */
ucMinute = sbbcdb(ucMinute, 1);
/* Minutes - 1 */
if(ucMinute == 0x99){
ucMinute = 0x59;
/* Borrow correction */
}
break;

case 0x02: /* Hours being adjusted */
ucHour = sbbcdb(ucHour, 1);
/* Hours - 1 */
if(ucHour == 0x99){
ucHour = 0x23;
/* Borrow correction */
}
break;

... (Omitted) ...
}
```

Treat DOWN key input as invalid in clock display status.

Decrement minutes of time data by -1 when DOWN key is on in minutes adjustment status.

Decrement hours of time data by -1 when DOWN key is on in hours adjustment status.

<3> Clock display updating processing

```

void main(void)
{
    ... (Omitted) ...

    if(ucAdjStatus == 0x02){
        ucUnderbarStatus = 1;
    }
    else{
        ucUnderbarStatus = 0;
    }

    ucNumWork = (ucHour >> 4) & 0x0f;
    if(ucNumWork == 0x00){
        ucNumWork = 0x0a;
    }
    fn_LcdNum(&SEG4, ucNumWork, ucUnderbarStatus);

    ucNumWork = ucHour & 0x0f;
    fn_LcdNum(&SEG10, ucNumWork, ucUnderbarStatus);

    fn_LcdNum(&SEG16, 0x0b, 0x00);

    if(ucAdjStatus == 0x01){
        ucUnderbarStatus = 1;
    }
    else{
        ucUnderbarStatus = 0;
    }

    ucNumWork = (ucMinute >> 4) & 0x0f;
    fn_LcdNum(&SEG22, ucNumWork, ucUnderbarStatus);

    ucNumWork = ucMinute & 0x0f;
    fn_LcdNum(&SEG28, ucNumWork, ucUnderbarStatus);
}

```

Append '_' when hours are displayed in hours adjustment status.

Display digit of 10 hours. Display zero-suppression of 0 to 9 hours.

Display digit of 1 hour.

Display digit of ".".

Append '_' when minutes are displayed in minutes adjustment status.

Display digit of 10 minutes.

Display digit of 1 minute.

```

void fn_LcdNum(unsigned char *ucSeg, unsigned char ucNum, unsigned char ucUnderbar)
{
    ... (Omitted) ...

    if(ucUnderbar == 0){
        ucFontWork = 0b00000000;
    }
    else{
        ucFontWork = 0b10000000;
    }

    for(ucFontIndex = 0, ucSegPtr = ucSeg; ucFontIndex < 6 ;
        ucFontIndex++ ,ucSegPtr++){
        /* Store font data in LCD display data memory */
        *ucSegPtr = (aucFontData[ucNum][ucFontIndex] | ucFontWork);
    }
}

```

Append '_' in hours and minutes adjustment statuses.

Display font data.

CHAPTER 5 RELATED DOCUMENTS

Document Name		English
78K0R/Lx3 User's Manual		PDF
78K0R Series Instructions User's Manual		PDF
RA78K0R Assembler Package User's Manual	Language	PDF
	Operation	PDF
CC78K0R C Compiler User's Manual	Language	PDF
	Operation	PDF
PM+ Project Manager User's Manual		PDF

APPENDIX A PROGRAM LIST

As a program list example, the 78K0R/LH3 microcontroller source program is shown below.

● main.asm (assembly language version)

```
*****
;
;   NEC Electronics   78K0R/LH3 Series
;
;*****
;   78K0R/LH3 Series   Sample Program
;
;                       (LCD Driver, Real-Timer Counter Used)
;*****
;   LCD Display Clock
;*****
;<<History>>
;   2009.02.--   Release
;*****
;
;<<Overview>>
;
; This sample program shows an example of using the LCD driver and real-time counter.
; It displays time in a 24-hour system. Time can be adjusted by using three keys (SET,
; UP, and DOWN). The Internal voltage boosting method is selected to drive the LCD driver.
;
;
;<Primary initial settings>
; (Option byte settings)
; - Disabling the watchdog timer
; - Setting the internal high-speed oscillator to 8 MHz
; - Disabling LVI from being started by default
; - Enabling on-chip debug to operate
; (Settings during initialization immediately after a reset ends)
; - Setting up I/O ports
; - Securing a supply voltage of 4.2 V or more by using the function of low-voltage detector
; - Specifying that the CPU/peripheral hardware clock run on the internal high-speed
oscillation clock (8 MHz)
; - Stopping the X1 oscillator
; - Starting the XT1 oscillator
```

```

; - Setting up timer array unit
; - Setting up LCD driver
;
;<Real-timer counter settings>
; - Setting 24-hour system
; - Setting to generate constant-period interrupt once per minute
; - Setting alarm match operation as invalid
; - Setting to not generate interval interrupt
; - Disabling output of RTCDIV pin
; - Setting 0 to hour, minute, and second count registers
; - Setting to not generate alarm match interrupt
;
;<Timer array unit settings>
; - Setting channel 4 as interval timer
; - Count clock period = fCLK/2^3 (1 MHz)
; - Pulse period = 10 ms (1 [us/clock] × 10000 [count] = 10 [ms])
;
;<LCD driver settings>
; - Selecting internal voltage boosting method, 8-time slice, 1/4 bias as LCD display mode
; - Setting operation of voltage boost circuit
; - LCD clock period = (fCLK/2^8)/2^7 (≅ 244.14 Hz)
; - Setting reference voltage to 1.00 V
;
;
;<I/O port settings>
; Input port: P30, P31, P32
; Output port: COM0 to COM7, SEG4 to SEG53
; * Set all unused ports that can be specified as output ports as output ports.
;
;*****
;=====
;
; Vector table
;
;=====
TVECT1          CSEG  AT      00000H
                DW      RESET_START      ; 00000H      RESET input, POC, LVI, WDT, TRAP

```

APPENDIX A PROGRAM LIST

TVECT2	CSEG	AT	00004H	
DW	IINIT		; 00004H	INTWDTI
DW	IINIT		; 00006H	INTLVI
DW	IINIT		; 00008H	INTP0
DW	IINIT		; 0000AH	INTP1
DW	IINIT		; 0000CH	INTP2
DW	IINIT		; 0000EH	INTP3
DW	IINIT		; 00010H	INTP4
DW	IINIT		; 00012H	INTP5
DW	IINIT		; 00014H	INTST3
DW	IINIT		; 00016H	INTSR3
DW	IINIT		; 00018H	INTSRE3
DW	IINIT		; 0001AH	INTDMA0
DW	IINIT		; 0001CH	INTDMA1
DW	IINIT		; 0001EH	INTST0/INTCSI00
DW	IINIT		; 00020H	INTSR0/INTCSI01
DW	IINIT		; 00022H	INTSRE0
DW	IINIT		; 00024H	INTST1/INTCSI10/INTIIC10
DW	IINIT		; 00026H	INTSR1
DW	IINIT		; 00028H	INTSRE1
DW	IINIT		; 0002AH	INTIICA
DW	IINIT		; 0002CH	INTTM00
DW	IINIT		; 0002EH	INTTM01
DW	IINIT		; 00030H	INTTM02
DW	IINIT		; 00032H	INTTM03
DW	IINIT		; 00034H	INTAD
DW	IINIT		; 00036H	INTRTC
DW	IINIT		; 00038H	INTRTCI
DW	IINIT		; 0003AH	INTKR
DW	IINIT		; 0003CH	INTST2/INTCSI20/INTIIC20
DW	IINIT		; 0003EH	INTSR2
DW	IINIT		; 00040H	INTSRE2
DW	IINIT		; 00042H	INTTM04
DW	IINIT		; 00044H	INTTM05
DW	IINIT		; 00046H	INTTM06
DW	IINIT		; 00048H	INTTM07
DW	IINIT		; 0004AH	INTP6
DW	IINIT		; 0004CH	INTP7
DW	IINIT		; 0004EH	INTP8

APPENDIX A PROGRAM LIST

```

DW          IINIT          ; 00050H      INTP9
DW          IINIT          ; 00052H      INTP10
DW          IINIT          ; 00054H      INTP11
DW          IINIT          ; 00056H      INTTM10
DW          IINIT          ; 00058H      INTTM11
DW          IINIT          ; 0005AH      INTTM12
DW          IINIT          ; 0005CH      INTTM13
DW          IINIT          ; 0005EH      INTMD

```

```

;=====

```

```

;
;   Define the memory stack area
;

```

```

;=====

```

```

DSTK DSEG  BASEP
STACKEND:
    DS    20H          ; Memory stack area = 32 bytes
STACKTOP:             ; Start address of the memory stack area

```

```

;=====

```

```

;
;   Define the RAM
;

```

```

;=====

```

```

DMAINDSEG  SADDR

```

```

RADJSTAT:  DS    1          ; Time adjustment status
RHOURL:    DS    1          ; Store hours data for display
RMIN:      DS    1          ; Store minutes data for display

RKYSTAT:   DS    1          ; Key status

```

```

XMAINCSEG  UNIT

```

```

;*****
;
;   Servicing interrupts by using unnecessary interrupt sources
;
;*****
IINIT:
;   If an unnecessary interrupt occurred, the processing branches to this line.
;   The processing then returns to the initial original processing because no processing
is performed here.

    RETI

;*****
;
;   Initialization after RESET
;
;*****
RESET_START:

;-----
;   Disable interrupts
;-----
    DI

;-----
;   Set up the register bank
;-----
    SEL    RB0

;-----
;   Initialize the stack pointer
;-----
    MOVW  SP,    #LOWW STACKTOP    ; Initialize the stack pointer

;-----
;   Specify the I/O port
;-----
    CALL  !!SINIPOINT    ; Set all ports that can be specified as output ports as
output ports.

```

```

;-----
;   Low-voltage detection
;-----
CALL  !!SINILVI          ; Secure a supply voltage of 4.2 V or more

;-----
;   Specify the clock frequency
;-----
CALL  !!SINICLK         ; Operate internal high-speed oscillation clock at 8 MHz

;-----
;   Specify the real-time counter
;-----
;   - Set count start time to 0:00:00
;   - Set constant-period interrupt (1-minute period) and alarm interrupt
;-----
; Control real-time counter control clock
MOV   PER0, #10000000B
;           |+++++----- Be sure to set 0
;           +----- RTCEN
;                               [Real-time counter control clock]
;                               0: Stops control clock supply
;                               1: Supplies control clock

; Set real-time counter operation
CLR1  RTCE              ; Stop real-time counter operation
MOV   RTCC0, #00001011B
;           |||||+++----- CT2/CT1/CT0
;           |||||          [Constant-period interrupt (INTRTC) selection]
;           |||||          000: Does not use constant-period interrupt function
;           |||||          001: Once per 0.5 s (synchronized with second count up)
;           |||||          010: Once per 1 s (same time as second count up)
;           |||||          011: Once per 1 m (second 00 of every minute)
;           |||||          100: Once per 1 hour (minute 00 and second 00 of every
hour)
;           |||||          101: Once per 1 day (hour 00, minute 00, and second 00
of every day)
;           |||||          11x: Once per 1 month (Day 1, hour 00 a.m., minute 00,
and second 00 of every month)

```

```

;          |||+----- AMPM
;          |||          [Selection of 12-/24-hour system]
;          |||          0: 12-hour system (a.m. and p.m. are displayed.)
;          |||          1: 24-hour system
;          ||+----- RCLOE0
;          ||          [RTCCL pin output control]
;          ||          0: Disables output of RTCCL pin (32.768 kHz)
;          ||          1: Enables output of RTCCL pin (32.768 kHz)
;          |+----- RCLOE1
;          ||          [RTC1HZ pin output control]
;          ||          0: Disables output of RTC1HZ pin (1 Hz)
;          ||          1: Enables output of RTC1HZ pin (1 Hz)
;          |+----- Be sure to set 0
;          +----- RTCE
;
;          [Real-time counter operation control]
;
;          0: Stops counter operation
;          1: Starts counter operation

```

; Set interval interrupt

```

MOV     RTCC2, #00000111B
;+|||+++----- RINTE/ICT2 to ICT0
; |||          [Interval interrupt (INTRTCI) selection]
; |||          0xxx: Interval interrupt is not generated.
; |||          1000: 2^6/fXT ( 1.953125 ms)
; |||          1001: 2^7/fXT ( 3.90625 ms)
; |||          1010: 2^8/fXT ( 7.8125 ms)
; |||          1011: 2^9/fXT ( 15.625 ms)
; |||          1100: 2^10/fXT ( 31.25 ms)
; |||          1101: 2^11/fXT ( 62.5 ms)
; |||          111x: 2^12/fXT (125 ms)
; ||++----- Be sure to set 0
; |+----- RCKDIV
; |          [Selection of RTCDIV pin output frequency]
; |          0: RTCDIV pin outputs 512 Hz (1.95 ms)
; |          1: RTCDIV pin outputs 16.384 kHz (0.061 ms)
; +----- RCLOE2
;          [RTCDIV pin output control]
;          0: Disables output of RTCDIV pin
;          1: Enables output of RTCDIV pin

```

```

; Set count start time (hour 08, minute 00, second 00)
MOV   SEC,   #00H           ; Second: 00
MOV   MIN,   #00H           ; Minute: 00
MOV   HOUR,  #00H           ; Hour: 00

; Set alarm interrupt function
CLR1  WALE           ; Match operation invalid
CLR1  WALIE          ; Does not generate interrupt on matching of alarm

CLR1  RTCIF          ; Clear INTRTC interrupt request
CLR1  RTCIIF         ; Clear INTRTCI interrupt request
SET1  RTCMK          ; Disable INTRTC interrupt
SET1  RTCIMK         ; Disable INTRTCI interrupt

SET1  RTCE           ; Start real-time counter operation

; Wait until device enters STOP mode immediately after real-time counter starts operating
MOV   B,        #124        ; * To place the device in the STOP mode immediately after
HINI210:                               ; RTCE has been set to 1, place the device in the STOP
NOP                                       ; mode after duration of two subsystem clocks (about 62
DEC   B          ; us) or more after RTCE has been set to 1.
BNZ   $HINI210

;-----
;   Initialize RAM
;-----

MOV   RADJSTAT,  #0        ; Clock display status

;-----
;   Specify the timer array unit
;-----
;   - Channel 4: Used as master channel in interval timer mode
;-----

; Initial settings of timer array unit
SET1  TAU0EN          ; Supply input clock of timer array unit

```



```

MOV     TPS0L, #00000011B    ; Timer clock select register 0
      ; |||||++++----- PRS003 to PRS000
      ;++++----- PRS013 to PRS010
      ;           [Operation clock (CK00/CK01) selection]
      ;           0000: fCLK
      ;           0001: fCLK/2
      ;           0010: fCLK/2^2
      ;           0011: fCLK/2^3
      ;           0100: fCLK/2^4
      ;           0101: fCLK/2^5
      ;           0110: fCLK/2^6
      ;           0111: fCLK/2^7
      ;           1000: fCLK/2^8
      ;           1001: fCLK/2^9
      ;           1010: fCLK/2^10
      ;           1011: fCLK/2^11
      ;           1100: fCLK/2^12
      ;           1101: fCLK/2^13
      ;           1110: fCLK/2^14
      ;           1111: fCLK/2^15

      ; Channel initial settings
MOVW   AX,    #0000100000000000B; Timer mode register 04 (for master channel)
MOVW   TMR04, AX
      ; |||||++++ MD043 to MD040
      ; |||||++++ [Channel 4 operation mode setting]
      ; |||||++++ 0000: Interval timer mode (Timer interrupt is not generated when counting
is started.)
      ; |||||++++ 0001: Interval timer mode
      ; |||||++++ (Timer interrupt is generated when counting is
started.)
      ; |||||++++ 0100: Capture mode
      ; |||||++++ (Timer interrupt is not generated when counting
is started.)
      ; |||||++++ 0101: Capture mode
      ; |||||++++ (Timer interrupt is generated when counting is
started.)

```

APPENDIX A PROGRAM LIST

```
;|||||      0110: Event counter mode
;|||||      1000: One-count mode
;|||||      (Start trigger is invalid during count operation.
;|||||      An interrupt is not generated when the trigger
occurs.)
;|||||      1001: One-count mode
;|||||      (Start trigger is valid during count operation.
;|||||      An interrupt is generated when the trigger
occurs.)
;|||||      1100: Capture & count mode
;|||||      Other than above: Setting prohibited
;|||||
;|||||+++---- Be sure to set 0
;|||||
;|||||++----- CIS041 and CIS040
;|||||      [TI02 pin valid edge selection]
;|||||      00: Falling edge
;|||||      01: Rising edge
;|||||      10: Both edges (when low-level width is measured)
;|||||      11: Both edges (when high-level width is measured)
;|||||+++----- STS042 to STS040
;|||||      [Setting of start trigger or capture trigger of
;|||||      channel 4]
;|||||      000: Only software trigger start is valid
;|||||      (other trigger sources are deselected)
;|||||      001: Valid edge of TI02 pin input signal is used as
;|||||      both the start trigger and capture trigger
;|||||      010: Both the edges of TI02 pin input signal are used
;|||||      as a start trigger and a capture trigger
;|||||      100: Interrupt signal of the master channel is used
;|||||      (when the channel is used as a slave channel with
the combination operation function)
;|||||      Other than above: Setting prohibited
;|||||+++----- MASER04
;|||||      [Selection of slave/master of independent operation or
;|||||      combination operation function of channel 4]
;|||||      0: Operates as slave channel with independent
;|||||      operation or combination operation function
;|||||      1: Operates as master channel with combination
```

operation function

```

;|||+----- CCS04
;|||          [Selection of count clock (TCLK) of channel 4]
;|||          0: Operation clock MCK specified by the CKS04 bit
;|||          1: Valid edge of input signal input from TI04 pin/
;|||          subsystem clock divided by 4 (fSUB/4)
;|++----- Be sure to set 0
;+----- CKS04
;          [Selection of operation clock (MCK) of channel 4]
;          0: Operation clock CK00 set by the TPS0 register
;          1: Operation clock CK01 set by the TPS0 register

;-----
; Specify the LCD driver
;-----
; Set so that the LCD driver function can be used.
;-----

```

```

MOV    LCDMD, #00010000B    ; LCD mode register
;|||+---- : Be sure to set 0
;|||
;|++----- : MDSET1 and MDSET0
;||          [LCD drive voltage generator selection]
;||          : 0 0 : External resistance division method
;||          : 0 1 : Internal voltage boosting method
;||          : 1 0 : Capacitor split method
;+----- : Be sure to set 0

```

```

MOV    SEGEN, #00011111B    ; Segment enable register
;||||||| Can be written only once after reset release
;|||+---- : SEGEN0 to SEGEN4
;||||||| [Output enable/disable to segment output only pins]
;||||||| : 0 : Disables segment output
;||||||| : 1 : Enables segment output
;|||||||+---- : SEGEN0 SEG8 to SEG11 pins
;|||||||+---- : SEGEN1 SEG12 to SEG15 pins
;|||||||+---- : SEGEN2 SEG16 to SEG19 pins

```

```

;||||+----- : SEGEN3 SEG20 to SEG23 pins
;||||+----- : SEGEN4 SEG24 to SEG26 pins
;|||
;+++----- : Be sure to set 0

MOV C, #36H ; LCD display data memory size
MOVW DE, #LOWW SEG0 ; LCD display data memory start address
CLRB A
HINI520:
MOV [DE], A ; Set display data
INCW DE
DEC C ; Setting display data completed?
BNZ $HINI520 ; NO,

MOV LCDM, #00000111B ; LCD display mode register
;||||+----- LCDM2 to LCDM0
;|||| [LCD controller/driver display mode selection]
;|||| External resistance division method Internal
voltage boosting method Capacitor split method
;|||| Number of time slices Bias mode Number of time slices
Bias mode Number of time slices Bias mode
;|||| 000: 4 1/3 4 1/3 4 1/3
;|||| 001: 3 1/3 3 1/3 3 1/3
;|||| 010: 2 1/2 4 1/3 4 1/3
;|||| 011: 3 1/2 4 1/3 4 1/3
;|||| 100: Static Setting prohibited
;|||| 111: 8 1/4 8 1/4 4 1/3
;||||
;||||+----- LCDSEL
;|||+----- BLON
;||| [Display data area control]
;||| BLON LCDSEL
;||| 0 0 : Displaying an A-pattern area data
;||| 0 1 : Displaying a B-pattern area data
;||| 1 0 : Alternately displaying A-pattern and
B-pattern area data
;||| 1 1 : Alternately displaying A-pattern and

```

B-pattern area data

```

;|||
;||+----- VLCON
;||          [Voltage boost circuit and capacitor split circuit
operation enable/disable]
;||          0 : Stops voltage boost circuit and capacitor split
circuit operation
;||          1 : Enables voltage boost circuit and capacitor split
circuit operation
;||
;|+----- SCOC
;+----- LCDON
;          [LCD display enable/disable]
;          LCDON SCOC
;          0    0 : Output ground level to segment/common pin
;          0    1 : Display off
;          1    0 : Output ground level to segment/common pin
;          1    1 : Display on

```

```

MOV    LCDC0, #00110011B    ; LCD clock control register 0
;||||+----- LCDC02 to LCDC00
;||||          [LCD clock (LCDC02) selection]
;||||          000: fLCD/2^4
;||||          001: fLCD/2^5
;||||          010: fLCD/2^6
;||||          011: fLCD/2^7
;||||          100: fLCD/2^8
;||||          101: fLCD/2^9
;||||
;||||+----- : Be sure to set 0
;||||
;||+----- LCDC05 and LCDC04
;||          [LCD source clock (fLCD) selection]
;||          00: fCLK
;||          01: fCLK/2^6
;||          10: fCLK/2^7
;||          11: fCLK/2^8
;||

```

```

;++++----- : Be sure to set 0

MOV  VLCD, #00001111B ; LCD boost level control register
;|||++++----- VLCD4 to VLCD0
;||| [Reference voltage selection]
;||| Reference voltage          VLCD voltage
;|||                               1/3 bias          1/4 bias
;||| 00000: 1.75 V          5.25 V          Setting prohibited
;||| 00001: 1.70 V          5.10 V          Setting prohibited
;||| 00010: 1.65 V          4.95 V          Setting prohibited
;||| 00011: 1.60 V          4.80 V          Setting prohibited
;||| 00100: 1.55 V          4.65 V          Setting prohibited
;||| 00101: 1.50 V          4.50 V          Setting prohibited
;||| 00110: 1.45 V          4.35 V          Setting prohibited
;||| 00111: 1.40 V          4.20 V          Setting prohibited
;||| 01000: 1.35 V          4.05 V          Setting prohibited
;||| 01001: 1.295 V         3.885 V          Setting prohibited
;||| 01010: 1.25 V          3.75 V          5.00 V
;||| 01011: 1.20 V          3.60 V          4.80 V
;||| 01100: 1.15 V          3.45 V          4.60 V
;||| 01101: 1.10 V          3.30 V          4.40 V
;||| 01110: 1.05 V          3.15 V          4.20 V
;||| 01111: 1.00 V          3.00 V          4.00 V
;||| 10000: 0.95 V          2.85 V          3.80 V
;||| 10001: 0.90 V          2.70 V          3.60 V
;||| 10010: 0.85 V          2.55 V          3.40 V
;||| 10011: 0.80 V          2.40 V          3.20 V
;|||
;++++----- : Be sure to set 0

; Wait for reference voltage setup time (2 ms or more) by using TM04 (2 ms)

MOVW  TDR04, #2000 ; Set pulse cycle to 2 ms (1 [us/clk] × 2000 [count])
MOV   TS0L, #00010000B ; Timer channel start register 0
;   +----- TS04
;           [Operation enable (start) trigger of channel 4]
;           0: No trigger operation
;           1: TE04 is set to 1 and the count operation becomes

```

enabled

```
CLR1  TMIF04          ; Clear INTTM04 interrupt request
```

HINI560:

```
NOP
```

```
BF    TMIF04,$HINI560 ; Wait 1 ms
```

```
CLR1  TMIF04          ; Clear INTTM04 interrupt request
```

```
MOV   TT0L, #00010000B ; Stop operation of timer channel 4
```

```
      ; +----- TS04
```

```
      ; [Operation stop trigger of channel 4]
```

```
      ; 0: No trigger operation
```

```
      ; 1: TE04 is set to 0 and the count operation becomes
```

stopped

```
MOVW  TDR04, #10000-1 ; Set pulse cycle to 10 ms (1 [us/clock] × 10000 [count])
```

```
MOV   PFALL, #01111111B ; Port function register
```

```
      ;||||| 0: Port (other than segment output)
```

```
      ;||||| 1: Segment output
```

```
      ;|||||+----- PF5L: P53 to P50/SEG50 to SEG53
```

```
      ;|||||+----- PF5H: P57 to P54/SEG46 to SEG49
```

```
      ;|||||+----- PF9L: P93 to P90/SEG42 to SEG45
```

```
      ;|||||+----- PF9H: P97 to P94/SEG38 to SEG41
```

```
      ;|||+----- PF10: P102 to P100/SEG27 to SEG29
```

```
      ;||+----- PF14L: P143 to P140/SEG34 to SEG37
```

```
      ;|+----- PF14H: P147 to P144/SEG30 to SEG33
```

```
      ;+----- Be sure to set 0
```

```
MOV   ISC, #00000000B ; Input switch control register
```

```
      ;|||||+----- ISC0
```

```
      ;||||| [Switching external interrupt (INTP0) input]
```

```
      ;||||| : 0 : Uses the input signal of the INTP0 pin as an external
```

interrupt

```
      ;||||| : 1 : Uses the input signal of the RxD3 pin as an external
```

interrupt

```

;|||||
;|||||+----- ISC1
;||||| [Switching channel 7 input of timer array unit]
;||||| : 0 : Uses the input signal of the TI07 pin as
a timer input
;||||| : 1 : Uses the input signal of the RxD3 pin as
a timer input
;|||||
;|||+++----- ISC2 to ISC4:
;||||| [Control of TI04/SEGxx/P53, TI02/SEGxx/P52, and
RxD3/SEGxx/P50]
;||||| : PF5L ISCn (2 to 4)
;||||| : 0 0: Port output
;||||| : 0 1: Port input, timer input, or serial data
input
;||||| : 1 0: Segment output
;|||||
;|||+----- ISC2
;||||| [RxD3/SEGxx/P50 schmitt trigger buffer control]
;||||| : 0 : Disables input
;||||| : 1 : Enables input
;|||||
;|||+----- ISC3
;||||| [TI02/SEGxx/P52 schmitt trigger buffer control]
;||||| : 0 : Disables input
;||||| : 1 : Enables input
;|||||
;|||+----- ISC4
;||| [TI04/SEGxx/P53 schmitt trigger buffer control]
;||| : 0 : Disables input
;||| : 1 : Enables input
;|||
;+++----- Be sure to set 0

```

```

SET1 VLCON ; Enable voltage boost circuit operation

```

```

; Wait for voltage boost wait time (500 ms or more) by using TM04 (10 ms)

```


APPENDIX A PROGRAM LIST

```

MOV    B,      #(500/10)+1
MOV    TS0L,   #00010000B    ; Timer channel start register 0
                                ; +----- TS04
                                ;           [Operation enable (start) trigger of channel 4]
                                ;           0: No trigger operation
                                ;           1: TE04 is set to 1 and the count operation becomes
enabled.
        CLR1   TMIF04          ; Clear INTTM04 interrupt request
HINI600:
        NOP
        BF     TMIF04,$HINI600    ; Wait 10 ms

        CLR1   TMIF04          ; Clear INTTM04 interrupt request

        DEC    B
        BNZ    $HINI600

        MOV    TT0L, #00010000B    ; Stop operation of timer channel 4
                                ; +----- TS04
                                ;           [Operation stop trigger of channel 4]
                                ;           0: No trigger operation
                                ;           1: TE04 is set to 0 and the count operation becomes
stopped

; LCD display on
        SET1   SCOC            ; Output deselected wave from all COM and SEG pins

        SET1   LCDON          ; LCD display on

;-----
;   Specify the external interrupt valid edge
;-----

        MOV    EGN0, #00100110B    ; Falling edge of INTP1, INTP2, and INTP5 valid

;-----
;   Enable interrupts

```

APPENDIX A PROGRAM LIST

```

; (To use interrupts, enable interrupts here.)
;-----

DI                ; Disable interrupt
CLR1  PIF1        ; Clear key interrupt request
CLR1  PIF2        ; Clear key interrupt request
CLR1  PIF5        ; Clear key interrupt request

CLR1  RTCMK       ; Unmask RTC constant-period interrupt
CLR1  PMK1        ; Unmask key interrupt
CLR1  PMK2        ; Unmask key interrupt
CLR1  PMK5        ; Unmask key interrupt

SET1  RTCIF       ; Initial display
BR    MAIN_LOOP   ; Go to the main loop

;*****
;
; I/O port setting
;
;*****
SINIPOINT:
;-----
; Specify the digital I/O
;-----

MOV    ADPC, #00010000B ; A/D port configuration register
      ;|||+++++----- ADPC4 to ADPC0
      ;|||           [Analog input (A)/digital I/O (D) switching]
      ;|||           +----+----- ANI15 to ANI8/P157 to P150
      ;|||           |||+++++---- ANI7 to ANI0/P27 to P20
      ;|||           00000: AAAAAAAAAAAAA
      ;|||           00001: AAAAAAAAAAAD
      ;|||           00010: AAAAAAAAAAAD
      ;|||           00011: AAAAAAAAAADD
      ;|||           00100: AAAAAAADDDD
      ;|||           00101: AAAAAAADDDDD
      ;|||           00110: AAAAAAADDDDD
      ;|||           00111: AAAAADDDDDDD
      ;|||           01000: AAAAADDDDDDD

```

APPENDIX A PROGRAM LIST

```

;|||          01001: AAADDDDDDDDD
;|||          01010: AADDDDDDDDDDD
;|||          01111: ADDDDDDDDDDDD
;|||          10000: DDDDDDDDDDDDD
;+++----- Be sure to set 0

;-----
; Initialize port 0
;-----
MOV    P0,    #00000000B ; Set the P00 to P02 output latches to low level
MOV    PM0,   #11111000B ; Specify P00 to P02 as output ports
                        ; P00 to P02: Unused

;-----
; Initialize port 1
;-----
MOV    P1,    #00000000B ; Set the P10 to P16 output latches to low level
MOV    PM1,   #00000000B ; Specify P10 to P17 as output ports
                        ; P10 to P17: Unused

;-----
; Initialize port 2
;-----
MOV    P2,    #00000000B ; Set the P20 to P27 output latches to low level
MOV    PM2,   #00000000B ; Specify P20 to P27 as output ports
                        ; P20 to P27: Unused

;-----
; Initialize port 3
;-----
MOV    P3,    #00000000B ; Set the P30 to P34 output latches to low level
MOV    PM3,   #11100111B ; Specify P33 and P34 as output ports
                        ; P33 and P34: Unused
MOV    PU3,   #00000111B ; Connect on-chip pull-up resistors to P30 to P32

;-----
; Initialize port 4
;-----
MOV    P4,    #00000000B ; Set the P40 and P41 output latches to low level

```

APPENDIX A PROGRAM LIST

```
MOV    PM4,    #11111100B    ; Specify P40 and P41 as output ports
                                ; P40 and P41: Unused

;-----
;    Initialize port 5
;-----

MOV    P5,     #00000000B    ; Set the P50 to P57 output latches to low level
MOV    PM5,    #00000000B    ; Specify P50 to P57 as output ports

;-----
;    Initialize port 6
;-----

MOV    P6,     #00000000B    ; Set the P60 and P61 output latches to low level
MOV    PM6,    #11111100B    ; Specify P60 and P61 as output ports
                                ; P60 and P61: Unused

;-----
;    Initialize port 7
;-----

MOV    P7,     #00000000B    ; Set the P70 to P77 output latches to low level
MOV    PM7,    #00000000B    ; Specify P70 to P77 as output ports
                                ; P70 to P77: Unused

;-----
;    Initialize port 8
;-----

MOV    P8,     #00000000B    ; Set the P80 to P87 output latches to low level
MOV    PM8,    #00000000B    ; Specify P80 to P87 as output ports
                                ; P80 to P87: Unused

;-----
;    Initialize port 9
;-----

MOV    P9,     #00000000B    ; Set the P90 to P97 output latches to low level
MOV    PM9,    #00000000B    ; Specify P90 to P97 as output ports
                                ; P90 to P97: Unused

;-----
```

```

; Initialize port 10
;-----
MOV    P10,    #00000000B    ; Set the P100 output latch to low level
MOV    PM10,   #11111000B    ; Specify P100 to P102 as output ports
                                ; P100 to P102: Unused

;-----
; Initialize port 11
;-----
MOV    P11,    #00000000B    ; Set the P110 and P111 output latches to low level
MOV    PM11,   #11111100B    ; Specify P110 and P111 as output ports
                                ; P110 and P111: Unused

;-----
; Initialize port 12
;-----
MOV    P12,    #00000000B    ; Set the P120 output latch to low level
MOV    PM12,   #11111110B    ; Specify P120 as output port
                                ; P120 to P124: Unused
                                ; * P121 to P124 are input ports.

;-----
; Initialize port 13
;-----
MOV    P13,    #00000000B    ; Set the P130 output latch to low level
                                ; P130: Unused

;-----
; Initialize port 14
;-----
MOV    P14,    #00000000B    ; Set the P140 to P147 output latches to low level
MOV    PM14,   #00000000B    ; Specify P140 to P147 as output ports
                                ; P140 to P147: Unused

;-----
; Initialize port 15
;-----
MOV    P15,    #00000000B    ; Set the P150 to P152, and P157 output latches to low
level
MOV    PM15,   #01111000B    ; Specify P150 to P152, and P157 as output ports

```

```

; P150 to P152: Unused

RET

; *****
;
; Low-voltage detection
;
; -----
; Secure a supply voltage of 4.2 V or more by using the function of low-voltage detector.
; *****
SINILVI:
; Set up the low-voltage detector
SET1  LVIMK          ; Disable the INTLVI interrupt
CLR1  LVISEL        ; Specify VDD as the detection voltage
MOV   LVIS, #0000000B ; Low-voltage detection level select register
; ||| | ++++----- LVIS3 to LVIS0
; ||| | [Detection level]
; ||| | 0000: VLVI0 (4.22 ±0.1 V)
; ||| | 0001: VLVI1 (4.07 ±0.1 V)
; ||| | 0010: VLVI2 (3.92 ±0.1 V)
; ||| | 0011: VLVI3 (3.76 ±0.1 V)
; ||| | 0100: VLVI4 (3.61 ±0.1 V)
; ||| | 0101: VLVI5 (3.45 ±0.1 V)
; ||| | 0110: VLVI6 (3.30 ±0.1 V)
; ||| | 0111: VLVI7 (3.15 ±0.1 V)
; ||| | 1000: VLVI8 (2.99 ±0.1 V)
; ||| | 1001: VLVI9 (2.84 ±0.1 V)
; ||| | 1010: VLVI10 (2.68 ±0.1 V)
; ||| | 1011: VLVI11 (2.53 ±0.1 V)
; ||| | 1100: VLVI12 (2.38 ±0.1 V)
; ||| | 1101: VLVI13 (2.22 ±0.1 V)
; ||| | 1110: VLVI14 (2.07 ±0.1 V)
; ||| | 1111: VLVI15 (1.91 ±0.1 V)
; ++++----- Be sure to set 0
CLR1  LVIMD          ; Specify that an interrupt signal is generated when a
low voltage is detected
SET1  LVION          ; Enable low-voltage detection

; Make the system wait until the low-voltage detector stabilizes (10 us or more)

```

```

MOV    B,    #10            ; Specify the number of counts
HRES100:
NOP                                ; (1 clk)
DEC    B                                ; (1 clk)
BNZ    $HRES100            ; Has the wait period ended? No, (2 clk/4 clk)

; Make the system wait until VLVI is less than or equal to VDD
HRES300:
NOP
BT     LVIF, $HRES300      ; VDD < VLVI? Yes,
CLR1   LVION              ; Stop the low-voltage detector

RET

;*****
;
; Specify the clock frequency
;
;-----
; Specify the clock frequency so that the device can run on the internal high-speed
oscillation clock.
;*****
SINICK:
MOV    CMC, #00010000B    ; Clock operation mode
;|||||+----- AMPH
;||||| [Control of high-speed system clock oscillation
frequency]
;||||| 0: 2 MHz ≤ fMX < 10 MHz
;||||| 1: 10 MHz < fMX ≤ 20 MHz
;||||+----- AMPHS1 and AMPHS0
;|||| [XT1 oscillator oscillation mode selection]
;|||| 00: Low power consumption oscillation (default)
;|||| 01: Normal oscillation
;|||| 10: Ultra-low power consumption oscillation
;|||| 11: Ultra-low power consumption oscillation
;|||+----- Be sure to set 0
;||+----- OSCSELS
;|| [Subsystem clock pin operation mode]
;|| 0: Input port mode

```

```

;|||          1: XT1 oscillation mode
;||+----- Be sure to set 0
;+----- EXCLK/OSCSEL
;
;           [High-speed system clock pin operation mode]
;
;           00: Input port mode
;
;           01: X1 oscillation mode
;
;           10: Input port mode
;
;           11: External clock input mode

MOV    CSC, #1000000B ; Clock operation status control
;|||||+----- HIOSTOP
;|||||      [Internal high-speed oscillation clock operation
control]

;|||||      0: Internal high-speed oscillator operating
;|||||      1: Internal high-speed oscillator stopped
;||++++----- Be sure to set 0
;+----- XTSTOP
;|
;           [Subsystem clock operation control]
;|
;           0: XT1 oscillator operating
;|
;           1: XT1 oscillator stopped
;+----- MSTOP
;
;           [High-speed system clock operation control]
;
;           0: X1 oscillator operating
;
;           1: X1 oscillator stopped

MOV    OSMC, #1000000B ; Operation speed mode
;|||||+----- FSEL/FLPC
;|||||      [fCLK frequency selection]
;|||||      00: Operates at a frequency of 10 MHz or less (default)
;|||||      01: Operates at a frequency higher than 10 MHz
;|||||      10: Operates at a frequency of 1 MHz
;|||||      11: Setting prohibited
;||++++----- Be sure to set 0
;+----- RTCLPC
;
;           [Setting in subsystem clock HALT mode]
;
;           0: Enables subsystem clock supply to peripheral
functions
;
;           1: Stops subsystem clock supply to peripheral functions
;
;           except real-time counter

```



```

MOV    CKC,    #00001000B    ; Clock selection
;|+|+|+++----- CSS/MCM0/MDIV2 to MDIV0
;| | |          [Selection of CPU/peripheral hardware clock (fCLK)]
;| | |          00000: fIH
;| | |          00001: fIH/2 (default)
;| | |          00010: fIH/2^2
;| | |          00011: fIH/2^3
;| | |          00100: fIH/2^4
;| | |          00101: fIH/2^5
;| | |          01000: fMX
;| | |          01001: fMX/2
;| | |          01010: fMX/2^2
;| | |          01011: fMX/2^3
;| | |          01100: fMX/2^4
;| | |          01101: fMX/2^5
;| | |          1xxxx: fSUB/2
;| | +----- Be sure to set 1
;| +----- MCS <Read Only>
;|          [Status of Main system clock (fMAIN)]
;|          0: Internal high-speed oscillation clock (fIH)
;|          1: High-speed system clock (fMX)
;+----- CLS <Read Only>
;          [Status of CPU/peripheral hardware clock (fCLK)]
;          0: Main system clock (fMAIN)
;          1: Subsystem clock (fSUB)

```

RET

```

;*****
;
; Main loop
;
;*****

```

MAIN_LOOP:

```

    HALT          ; Wait for port scan timing
    NOP

```

APPENDIX A PROGRAM LIST

```

MOV1  CY,    RTCIF
SKNC                                ; 1 minute has passed (RTCIF = 1)?
BR    LMAIN720                      ; Yes,

BT    PIF1,$LMAIN120                ; Key input?, Yes,
BT    PIF2,$LMAIN120                ; Key input?, Yes,
BF    PIF5,$MAIN_LOOP               ; Key input?, No,

;-----
;   Eliminate noise from key input
;-----

LMAIN120:
    CLR1  PIF1                      ; Clear key input interrupt request
    CLR1  PIF2                      ; Clear key input interrupt request
    CLR1  PIF5                      ; Clear key input interrupt request

    MOV   A,    P3                  ; Read key status
    AND   A,    #00000111B         ; Write only key status
    CMP   A,    #00000111B         ; Key input?
    BZ    $MAIN_LOOP               ; No,

    MOV   X,    A                   ;
    MOV   TS0L, #00010000B         ; Enable operation of timer channel 4
    CLR1  TMIF04                    ; Clear INTTM04 interrupt request
LMAIN160:
    NOP
    BF    TMIF04,$LMAIN160         ; Wait 10 ms

    CLR1  TMIF04                    ; Clear INTTM04 interrupt request
    MOV   TT0L, #00010000B         ; Stop operation of timer channel 4

    CLR1  PIF1                      ; Clear key input interrupt request
    CLR1  PIF2                      ; Clear key input interrupt request
    CLR1  PIF5                      ; Clear key input interrupt request

    MOV   A,    P3                  ; Read key status
    AND   A,    #00000111B         ; Write only key status
    CMP   A,    X                   ; Key input valid?
    BNZ   $MAIN_LOOP               ; No,

```

```

;-----
;   Clock adjustment
;-----

```

```

; SET key processing

```

```

    BT    A.0,$LMAIN400      ; SET key on?, No,

```

```

    CMP   RADJSTAT,#0       ; Clock being displayed?

```

```

    BZ    $LMAIN320         ; Yes,

```

```

    CMP   RADJSTAT,#1       ; Minutes being adjusted?

```

```

    BZ    $LMAIN360         ; Yes,

```

```

                                ; No, Hours being adjusted

```

```

    MOV   RADJSTAT,#0       ; Assume clock being displayed

```

```

LMAIN240:

```

```

    SET1  RWAIT              ; RTC value read/Write mode

```

```

    BF    RWST, $LMAIN240    ; Wait for RTC read/write mode

```

```

    MOV   A,    RMIN         ; Read minutes

```

```

    MOV   MIN,  A           ; Set minutes

```

```

    MOV   A,    RHOURL       ; Read hours

```

```

    MOV   HOUR, A          ; Set hours

```

```

    MOV   SEC,  #00H        ; Second: 00

```

```

LMAIN280:

```

```

    CLR1  RWAIT              ; Start RTC count operation

```

```

    BT    RWST, $LMAIN280    ; Wait for start of RTC count operation

```

```

    BR    LMAIN790

```

```

LMAIN320:

```

```

    MOV   RADJSTAT,#1       ; Assume minutes being adjusted

```

```

    BR    LMAIN740

```

```
LMAIN360:
    MOV    RADJSTAT,#2        ; Assume hours being adjusted
    BR     LMAIN790
```

; UP key processing

```
LMAIN400:
    CMP    RADJSTAT,#0        ; Clock being displayed?
    BZ     $LMAIN680          ; Yes,

    BT     A.1,$LMAIN500      ; UP key on?, No,

    CMP    RADJSTAT,#1        ; Minutes being adjusted?
    BZ     $LMAIN460          ; Yes,
                                ; No, Hours being adjusted

    MOV    A,    RHOURL
    ADD    A,    #1            ; Hours + 1
    ADD    A,    !BCDADJ      ; Decimal correction
    CMP    A,    #24H          ; 24 hours?
    BNZ   $LMAIN440          ; No,

    MOV    A,    #00H          ; 0 hour
```

```
LMAIN440:
    MOV    RHOURL, A
    BR     LMAIN620
```

```
LMAIN460:
    MOV    A,    RMIN
    ADD    A,    #1            ; Minutes + 1
    ADD    A,    !BCDADJ      ; Decimal correction
    CMP    A,    #60H          ; 60 minutes?
    BNZ   $LMAIN480          ; No,

    MOV    A,    #00H          ; 0 hour
```

```
LMAIN480:
    MOV    RMIN, A            ; Set minutes
    BR     LMAIN620
```

; DOWN key processing

```
LMAIN500:
```

APPENDIX A PROGRAM LIST

```

BT      A.2,$LMAIN680      ; DOWN key on?, No,

CMP     RADJSTAT,#1       ; Minutes being adjusted?
BZ      $LMAIN560         ; Yes,
                          ; No, Hours being adjusted

MOV     A,      RHOURL
SUB     A,      #1        ; Hours - 1
SUB     A,      !BCDADJ   ; Decimal correction
BNC     $LMAIN540        ; 23 hours?, No,

MOV     A,      #23H      ; 23 hours

LMAIN540:
MOV     RHOURL, A
BR      LMAIN620

LMAIN560:
MOV     A,      RMIN
SUB     A,      #1        ; Minutes - 1
SUB     A,      !BCDADJ   ; Decimal correction
BNC     $LMAIN580        ; 59 minutes?, No,

MOV     A,      #59H      ; 59 minutes

LMAIN580:
MOV     RMIN,   A        ; Set minutes

LMAIN620:
BR      LMAIN790

LMAIN680:
BR      MAIN_LOOP

;-----
;   Update clock display
;-----

LMAIN720:
CLR1   RTCIF           ; Clear constant-period interrupt request
CMP    RADJSTAT,#0     ; Clock being displayed?
BNZ    $LMAIN790       ; No,

```

```

LMAIN740:
    SET1  RWAIT          ; RTC value read/write mode
    BF    RWST, $LMAIN740 ; Wait for RTC read/write mode

    MOV   A,    MIN      ; Read minutes
    MOV   RMIN, A        ; Save minutes to display buffer

    MOV   A,    HOUR     ; Read hours
    MOV   RHOURL, A     ; Save hours to display buffer

LMAIN780:
    CLR1  RWAIT          ; Start RTC count operation
    BT    RWST, $LMAIN780 ; Wait for start of RTC count operation

LMAIN790:
    MOV   B,    #0       ; '_' not displayed
    CMP   RADJSTAT, #2   ; Hours being adjusted?
    BNZ   $LMAIN820     ; NO,
    MOV   B,    #1       ; '_' displayed

LMAIN820:
    MOV   A,    RHOURL   ; Digit of 10 hours
    AND   A,    #0F0H    ; Digit of 10 hours
    SHR   A,    4
    CMP0  A        ; Digit of 10 hours = 0?
    BNZ   $LMAIN860     ; No,

    MOV   A,    #0AH     ; ' ' (0-suppression)

LMAIN860:
    MOVW  DE,    #LOWW SEG4 ; LCD display position
    CALL  !!SLCDNUM       ; Display digit of 10 hours

    MOV   A,    RHOURL   ; Digit of 1 hour
    AND   A,    #0FH     ; Digit of 1 hour
    MOVW  DE,    #LOWW SEG10 ; LCD display position
    CALL  !!SLCDNUM       ; Display digit of 1 hour

    MOV   B,    #0       ; '_' not displayed
    MOV   A,    #0BH     ; ':'
    MOVW  DE,    #LOWW SEG16 ; LCD display position

```

```

CALL  !!SLCDNUM          ; Display digit of 1 hour

MOV   B,    #0           ; '_' not displayed
CMP   RADJSTAT,#1       ; Minutes being adjusted?
BNZ   $LMAIN900         ; NO,
MOV   B,    #1           ; '_' displayed
LMAIN900:
MOV   A,    RMIN
AND   A,    #0F0H       ; Digit of 10 minutes
SHR   A,    4
MOVW  DE,   #LOWW SEG22 ; LCD display position
CALL  !!SLCDNUM          ; Display digit of 10 minutes

MOV   A,    RMIN
AND   A,    #0FH        ; Digit of 1 minute
MOVW  DE,   #LOWW SEG28 ; LCD display position
CALL  !!SLCDNUM          ; Display digit of 1 minute

BR    MAIN_LOOP

;*****
;
;   Numeric display
;   The numeric value of Acc (BCD) is displayed on the LCD display data memory specified
;   by DE.  '_' is appended if the value of the B register is other than 0.
;
;*****
SLCDNUM:
ADD   A,    A
MOV   X,    A           ; X = Value*2
ADD   A,    A           ; A = Value*4
ADD   X,    A           ; X = Value*(4+2)
CLRB  A              ; A = 0
ADDW  AX,   #LOWW TCHRDATA ; Value font start address
MOVW  HL,   AX
MOV   C,    #6         ; Value font size
JLCDN220:
MOV   A,    [HL]       ; Read value font

```

APPENDIX A PROGRAM LIST

```

CMP0  B                      ; '_'?
BZ    $JLCDN260              ; No,

OR    A,#1000000B           ; Append '_'
JLCDN260:
MOV   [DE], A                ; Set to LCD display data memory
INCW  HL                     ; Update font data address
INCW  DE                     ; Update LCD display data memory address
DEC   C                      ; Font size transfer completed?
BNZ   $JLCDN220              ; No,

RET

XCHRDATA          CSEG  MIRRORP
;*****
;   Font data definition
;*****
TCHRDATA:

;          COM76543210
DB    00111110B           ; '0'
DB    01000001B
DB    01000001B
DB    01000001B
DB    00111110B
DB    00000000B

;          COM76543210
DB    00000000B           ; '1'
DB    00000000B
DB    00000010B
DB    01111111B
DB    00000000B
DB    00000000B

;          COM76543210
DB    01000010B           ; '2'

```



```

DB      01100001B
DB      01010001B
DB      01001001B
DB      01000110B
DB      00000000B

;      COM76543210
DB      00100010B      ; '3'
DB      01001001B
DB      01001001B
DB      01001001B
DB      00110110B
DB      00000000B

;      COM76543210
DB      00111100B      ; '4'
DB      00100010B
DB      00100001B
DB      01111111B
DB      00100000B
DB      00000000B

;      COM76543210
DB      00100111B      ; '5'
DB      01000101B
DB      01000101B
DB      01000101B
DB      00111001B
DB      00000000B

;      COM76543210
DB      00111110B      ; '6'
DB      01001001B
DB      01001001B
DB      01001001B
DB      00110010B
DB      00000000B

;      COM76543210

```

APPENDIX A PROGRAM LIST

```
DB 00000001B ; '7'
DB 01110001B
DB 00001001B
DB 00000101B
DB 00000011B
DB 00000000B

; COM76543210
DB 00110110B ; '8'
DB 01001001B
DB 01001001B
DB 01001001B
DB 00110110B
DB 00000000B

; COM76543210
DB 00100110B ; '9'
DB 01001001B
DB 01001001B
DB 01001001B
DB 00111110B
DB 00000000B

; COM76543210
DB 00000000B ; ':'
DB 00000000B
DB 00000000B
DB 00000000B
DB 00000000B
DB 00000000B

; COM76543210
DB 00000000B ; ':'
DB 00000000B
DB 00100100B
DB 00000000B
DB 00000000B
DB 00000000B
```

end

● main.c (C language version)

```

/*****

```

```

    NEC Electronics      78K0R/LH3 Series

```

```

*****

```

```

    78K0R/LH3 Series Sample Program
        (LCD Driver, Real-Timer Counter Used)

```

```

*****

```

```

    LCD Display Clock

```

```

*****

```

```

<<History>>

```

```

    2009.3.-- Release

```

```

*****

```

```

<<Overview>>

```

This sample program shows an example of using the LCD driver and real-time counter. It displays time in a 24-hour system. Time can be adjusted by using three keys (SET, UP, and DOWN). The Internal voltage boosting method is selected to drive the LCD driver.

```

<Primary initial settings>

```

```

(option byte settings)

```

- Disabling the watchdog timer
- Setting the internal high-speed oscillator to 8 MHz
- Disabling LVI from being started by default
- Enabling on-chip debug to operate

```

(Settings during initialization immediately after a reset ends)

```

- Setting up I/O ports
- Securing a supply voltage of 4.2 V or more by using the function of low-voltage detector
- Specifying that the CPU/peripheral hardware clock run on the internal high-speed

```

oscillation clock (8 MHz)

```

- Stopping the X1 oscillator
- Starting the XT1 oscillator
- Setting up real-timer counter
- Setting up timer array unit

<Real-timer counter settings>

- Setting 24-hour system
- Setting to generate constant-period interrupt once per minute
- Setting alarm match operation as invalid
- Setting to not generate interval interrupt
- Disabling output of RTCDIV pin
- Setting 0 to hour, minute, and second count registers
- Setting to not generate alarm match interrupt

<Timer array unit settings>

- Setting channel 4 as interval timer
- Count clock period = $fCLK/2^3$ (1 MHz)
- Pulse period = 10 ms ($1 [us/clk] \times 10000 [count] = 10 [ms]$)

<LCD driver settings>

- Selecting internal voltage boosting method, 8-time slice, 1/4 bias as LCD display mode
- Setting operation of voltage boost circuit
- LCD clock period = $(fCLK/2^8)/2^7$ ($\cong 244.14$ Hz)
- Setting reference voltage to 1.00 V

<I/O port settings>

Input port: P30, P31, P32

Output port: COM0 to COM7, SEG4 to SEG53

* Set all unused ports that can be specified as output ports as output ports.

/*=====

Preprocessing directive (#pragma)

=====*/

#pragma SFR /* SFR names can be described at the C source level */

#pragma DI /* DI instructions can be described at the C source level */

#pragma EI /* EI instructions can be described at the C source level */

#pragma NOP /* NOP instructions can be described at the C source level */

```

#pragma HALT          /* HALT instructions can be described at the C source level */
#pragma BCD           /* BCD operation function can be used */

/*=====

Function prototype declaration

=====*/
void fn_InitPort( void ); /* I/O port setting */
void fn_InitLvi( void ); /* Low voltage detection */
void fn_InitClock( void ); /* Clock frequency setting */
void fn_LcdNum(unsigned char *ucSeg, unsigned char ucNum, unsigned char ucUnderbar);
/* Display number on LCD */

/*=====

Variable definition

=====*/
unsigned char ucAdjStatus; /* Clock adjustment status */
unsigned char ucMinute; /* Store minutes data for display and time adjustment */
unsigned char ucHour; /* Store hours data for display and time adjustment */

/*****

Initialization after RESET

*****/
void hdwinit( void )
{
    unsigned char count; /* Set work */
    unsigned char *wkptr; /* Set work pointer */

/*-----

Disable interrupts

-----*/
    DI(); /* Disable interrupts */

/*-----

```

```

Specify the I/O port
-----*/
fn_InitPort();          /* Set all ports that can be specified as output ports as */

/*-----
Low-voltage detection
-----*/
fn_InitLvi();          /* Secure a supply voltage of 4.2 V or more */

/*-----
Specify the clock frequency
-----*/
fn_InitClock();        /* Operate internal high-speed oscillation clock at 8 MHz */

/*-----
Specify the real-time counter
-----
- Set count start time to 0:00:00
- Set constant-period interrupt (1-minute period) and alarm interrupt
-----*/

/* Control real-time counter control clock */
PER0 = 0b10000000;
/*      |+++++----- Be sure to set 0 */
/*      +----- RTCEN */
/*      [Real-time counter control clock] */
/*      0: Stops control clock supply */
/*      1: Supplies control clock */

/* Set real-time counter operation */
RTCE = 0;              /* Stop real-time counter operation */
RTCC0 = 0b00001011;
/*      |||||+++----- CT2/CT1/CT0 */
/*      ||||| [Constant-period interrupt (INTRTC) selection] */
/*      ||||| 000: Does not use constant-period interrupt function. */
/*      ||||| 001: Once per 0.5 s (synchronized with second count up) */
/*      ||||| 010: Once per 1 s (same time as second count up) */
/*      ||||| 011: Once per 1 m (second 00 of every minute) */
/*      ||||| 100: Once per 1 hour (minute 00 and second 00 of every hour) */
/*      ||||| 101: Once per 1 day (hour 00, minute 00, and second 00 of every

```

```

day) */
/*      ||||      11x: Once per 1 month (Day 1, hour 00 a.m., minute 00, and second
00 of every month) */
/*      |||+----- AMPM */
/*      |||      [Selection of 12-/24-hour system] */
/*      |||      0: 12-hour system (a.m. and p.m. are displayed.) */
/*      |||      1: 24-hour system */
/*      ||+----- RCLOE0 */
/*      ||      [RTCC pin output control] */
/*      ||      0: Disables output of RTCC pin (32.768 kHz) */
/*      ||      1: Enables output of RTCC pin (32.768 kHz) */
/*      |+----- RCLOE1 */
/*      ||      [RTC1HZ pin output control] */
/*      ||      0: Disables output of RTC1HZ pin (1 Hz) */
/*      ||      1: Enables output of RTC1HZ pin (1 Hz) */
/*      |+----- Be sure to set 0 */
/*      +----- RTCE */
/*      [Real-time counter operation control] */
/*      0: Stops counter operation */
/*      1: Starts counter operation */

/* Set interval interrupt */
RTCC2 = 0b00000111;

/*+|||+++----- RINTE/ICT2 to ICT0 */
/* ||||      [Interval interrupt (INTRTCI) selection] */
/* ||||      0xxx: Interval interrupt is not generated. */
/* ||||      1000: 2^6/fXT ( 1.953125 ms) */
/* ||||      1001: 2^7/fXT ( 3.90625 ms) */
/* ||||      1010: 2^8/fXT ( 7.8125 ms) */
/* ||||      1011: 2^9/fXT ( 15.625 ms) */
/* ||||      1100: 2^10/fXT ( 31.25 ms) */
/* ||||      1101: 2^11/fXT ( 62.5 ms) */
/* ||||      111x: 2^12/fXT (125 ms) */
/* ||+++----- Be sure to set 0 */
/* |+----- RCKDIV */
/* |      [Selection of RTCDIV pin output frequency] */
/* |      0: RTCDIV pin outputs 512 Hz (1.95 ms) */
/* |      1: RTCDIV pin outputs 16.384 kHz (0.061 ms) */
/* +----- RCLOE2 */

```



```

/*      [RTCDIV pin output control] */
/*      0: Disables output of RTCDIV pin */
/*      1: Enables output of RTCDIV pin */

/* Set count start time (Year 00, month 01, day 01, Saturday, hour 08 a.m., minute 59,
second 50) */
SEC   = 0x00;    /* Second: 00 */
MIN   = 0x00;    /* Minute: 00 */
HOUR  = 0x00;    /* Hour: 00 */

/* Set alarm interrupt function */
WALE  = 0;        /* Match operation invalid */
WALIE = 0;        /* Does not generate interrupt on matching of alarm */

RTCIF = 0;        /* Clear INTRTC interrupt request */
RTCIIF = 0;       /* Clear INTRTCI interrupt request */
RTCMK = 1;        /* Disable INTRTC interrupt */
RTCIMK = 1;       /* Disable INTRTCI interrupt */

RTCE = 1;        /* Start real-time counter operation */

/* Wait until device enters STOP mode immediately after real-time counter starts operating
*/
for(count = 124; count > 0; count--){ /* * To place the device in the STOP mode */
    NOP();                             /* immediately after RTCE has been set to */
}                                       /* 1, place the device in the STOP mode */
                                       /* after duration of two subsystem clocks
(about 62 us) or more after RTCE has been set to 1. */

/*-----
Specify the timer array unit
-----
Set so that the D/A output and DMA function can be used.
- Channel 4: Used as master channel in interval timer mode
-----*/

/* Initial settings of timer array unit */
TAU0EN = 1;      /* Supply input clock of timer array unit */

```

```

TPS0L = 0b00000011;      /* Timer clock select register 0 */
/*|||++++----- PRS003 to PRS000 */
/*++++----- PRS013 to PRS010 */
/*      [Operation clock (CK00/CK01) selection] */
/*      0000: fCLK */
/*      0001: fCLK/2 */
/*      0010: fCLK/2^2 */
/*      0011: fCLK/2^3 */
/*      0100: fCLK/2^4 */
/*      0101: fCLK/2^5 */
/*      0110: fCLK/2^6 */
/*      0111: fCLK/2^7 */
/*      1000: fCLK/2^8 */
/*      1001: fCLK/2^9 */
/*      1010: fCLK/2^10 */
/*      1011: fCLK/2^11 */
/*      1100: fCLK/2^12 */
/*      1101: fCLK/2^13 */
/*      1110: fCLK/2^14 */
/*      1111: fCLK/2^15 */

/* Channel initial settings */
TMR04 = 0b0000100000000000; /* Timer mode register 04 (for master channel) */
/*||||||||||||++++ MD043 to MD040 */
/*|||||||||||| [Channel 4 operation mode setting] */
/*|||||||||||| 0000: Interval timer mode */
/*|||||||||||| (Timer interrupt is not generated when counting is
started.) */
/*|||||||||||| 0001: Interval timer mode */
/*|||||||||||| (Timer interrupt is generated when counting is started.)
*/
/*|||||||||||| 0100: Capture mode */
/*|||||||||||| (Timer interrupt is not generated when counting is
started.) */
/*|||||||||||| 0101: Capture mode */
/*|||||||||||| (Timer interrupt is generated when counting is started.)
*/
/*|||||||||||| 0110: Event counter mode */

```

```

/*||||||| 1000: One-count mode */
/*||||||| (Start trigger is invalid during count operation. */
/*||||||| An interrupt is not generated when the trigger occurs.)
*/

/*||||||| 1001: One-count mode */
/*||||||| (Start trigger is valid during count operation. */
/*||||||| An interrupt is generated when the trigger occurs.) */
/*||||||| 1100: Capture & count mode */
/*||||||| Other than above: Setting prohibited */
/*|||||||++----- Be sure to set 0 */
/*|||||||++----- CIS041 and CIS040 */
/*||||||| [TI02 pin valid edge selection] */
/*||||||| 00: Falling edge */
/*||||||| 01: Rising edge */
/*||||||| 10: Both edges (when low-level width is measured) */
/*||||||| 11: Both edges (when high-level width is measured) */
/*|||||||+++----- STS042 to STS040 */
/*||||||| [Setting of start trigger or capture trigger of Channel 4 */
/*||||||| 000: Only software trigger start is valid */
/*||||||| (other trigger sources are unselected)) */
/*||||||| 001: Valid edge of TI02 pin input signal is used as */
/*||||||| both the start trigger and capture trigger */
/*||||||| 010: Both the edges of TI02 pin input signal are used as */
/*||||||| a start trigger and a capture trigger */
/*||||||| 100: Interrupt signal of the master channel is used */
/*||||||| (when the channel is used as a slave channel with the
combination operation function) */
/*||||||| Other than above: Setting prohibited */
/*|||||||+----- MASER04 */
/*||||||| [Selection of slave/master of independent operation or */
/*||||||| combination operation function of channel 4] */
/*||||||| 0: Operates as slave channel with independent operation or */
/*||||||| combination operation function */
/*||||||| 1: Operates as master channel with combination operation function
*/

/*|||||||+----- CCS04 */
/*||||||| [Selection of count clock (TCLK) of channel 4] */
/*||||||| 0: Operation clock MCK specified by the CKS04 bit */
/*||||||| 1: Valid edge of input signal input from TI04 pin/ */

```

```

/*||| subsystem clock divided by 4 (fSUB/4) */
/*|+----- Be sure to set 0 */
/*+----- CKS04 */
/* [Selection of operation clock (MCK) of channel 4] */
/* 0: Operation clock CK00 set by the TPS0 register */
/* 1: Operation clock CK01 set by the TPS0 register */

TMIF04 = 0; /* Clear INTTM04 interrupt request */

/*-----
Specify the LCD driver
-----
Set so that the LCD driver function can be used.
-----*/

LCDMD = 0b00010000; /* LCD mode register */
/*|||++++--- : Be sure to set 0 */
/*||| */
/*|+----- : MDSET1 and MDSET0 */
/*| [LCD drive voltage generator selection] */
/*| : 0 0 : External resistance division method */
/*| : 0 1 : Internal voltage boosting method */
/*| : 1 0 : Capacitor split method */
/*+----- : Be sure to set 0 */

SEGEN = 0b00011111; /* Segment enable register */
/*||||| Can be written only once after reset release */
/*|||++++--- : SEGEN0 to SEGEN4 */
/*||||| [Output enable/disable to segment output only pins] */
/*||||| : 0 : Disables segment output */
/*||||| : 1 : Enables segment output */
/*|||||+--- : SEGEN0 SEG8 to SEG11 pins */
/*|||||+---- : SEGEN1 SEG12 to SEG15 pins */
/*|||||+----- : SEGEN2 SEG16 to SEG19 pins */
/*|||||+----- : SEGEN3 SEG20 to SEG23 pins */
/*|||+----- : SEGEN4 SEG24 to SEG26 pins */
/*||| */

```

```

/*+++----- : Be sure to set 0 */

for(count = 0, wkptr = &SEG0; count < 0x36; count++, wkptr++){
    *wkptr = 0x00;      /* Clear the LCD display memory to 0 */
}

LCDM = 0b00000111;    /* LCD display mode register */
/*|||||+++----- LCDM2 to LCDM0 */
/*||||| [LCD controller/driver display mode selection] */
/*||||| External resistance division method Internal voltage boosting
method Capacitor split method */
/*||||| Number of time slices Bias mode Number of time slices Bias mode
Number of time slices Bias mode */
/*||||| 000: 4 1/3 4 1/3 4 1/3 */
/*||||| 001: 3 1/3 3 1/3 3 1/3 */
/*||||| 010: 2 1/2 4 1/3 4 1/3 */
/*||||| 011: 3 1/2 4 1/3 4 1/3 */
/*||||| 100: Static Setting prohibited */
/*||||| 111: 8 1/4 8 1/4 4 1/3 */
/*||||| */
/*|||||+----- LCDSEL */
/*|||+----- BLON */
/*||| [Display data area control] */
/*||| BLON LCDSEL */
/*||| 0 0 : Displaying an A-pattern area data */
/*||| 0 1 : Displaying a B-pattern area data */
/*||| 1 0 : Alternately displaying A-pattern and B-pattern area data
*/
/*||| 1 1 : Alternately displaying A-pattern and B-pattern area data
*/
/*||| */
/*|||+----- VLCON */
/*|| [Voltage boost circuit and capacitor split circuit operation
enable/disable] */
/*|| 0 : Stops voltage boost circuit and capacitor split circuit operation
*/
/*|| 1 : Enables voltage boost circuit and capacitor split circuit operation

```

```

*/

/*|| */
/*|+----- SCOC */
/*+----- LCDON */
/*      [LCD display enable/disable] */
/*      LCDON SCOC */
/*      0      0 : Output ground level to segment/common pin */
/*      0      1 : Display off */
/*      1      0 : Output ground level to segment/common pin */
/*      1      1 : Display on */

LCDC0 = 0b00110011;      /* LCD clock control register 0 */
/*|||||+++----- LCDC02 to LCDC00 */
/*|||||      [LCD clock (LCDCL) selection] */
/*|||||      000: fLCD/2^4 */
/*|||||      001: fLCD/2^5 */
/*|||||      010: fLCD/2^6 */
/*|||||      011: fLCD/2^7 */
/*|||||      100: fLCD/2^8 */
/*|||||      101: fLCD/2^9 */
/*||||| */
/*|||||+----- : Be sure to set 0 */
/*||||| */
/*||+----- LCDC05 and LCDC04 */
/*||      [LCD source clock (fLCD) selection] */
/*||      00: fCLK */
/*||      01: fCLK/2^6 */
/*||      10: fCLK/2^7 */
/*||      11: fCLK/2^8 */
/*|| */
/*++----- : Be sure to set 0 */

VLCD = 0b00001111;      /* LCD boost level control register */
/*||||++++----- VLCD4 to VLCD0 */
/*|||      [Reference voltage selection] */
/*|||      Reference voltage          VLCD voltage */
/*|||      1/3 bias          1/4 bias */

```

APPENDIX A PROGRAM LIST

```

/*|||      00000: 1.75 V      5.25 V      Setting prohibited */
/*|||      00001: 1.70 V      5.10 V      Setting prohibited */
/*|||      00010: 1.65 V      4.95 V      Setting prohibited */
/*|||      00011: 1.60 V      4.80 V      Setting prohibited */
/*|||      00100: 1.55 V      4.65 V      Setting prohibited */
/*|||      00101: 1.50 V      4.50 V      Setting prohibited */
/*|||      00110: 1.45 V      4.35 V      Setting prohibited */
/*|||      00111: 1.40 V      4.20 V      Setting prohibited */
/*|||      01000: 1.35 V      4.05 V      Setting prohibited */
/*|||      01001: 1.295 V     3.885 V     Setting prohibited */
/*|||      01010: 1.25 V      3.75 V      5.00 V */
/*|||      01011: 1.20 V      3.60 V      4.80 V */
/*|||      01100: 1.15 V      3.45 V      4.60 V */
/*|||      01101: 1.10 V      3.30 V      4.40 V */
/*|||      01110: 1.05 V      3.15 V      4.20 V */
/*|||      01111: 1.00 V      3.00 V      4.00 V */
/*|||      10000: 0.95 V      2.85 V      3.80 V */
/*|||      10001: 0.90 V      2.70 V      3.60 V */
/*|||      10010: 0.85 V      2.55 V      3.40 V */
/*|||      10011: 0.80 V      2.40 V      3.20 V */
/*||| */
/*++++----- : Be sure to set 0 */

/* Wait for reference voltage setup time (2 ms or more) by using TM04 (2 ms) */

TDR04 = 2000; /* Set pulse cycle to 2 ms (1 [us/clock] × 2000 [count]) */

TS0L = 0b00010000; /* Timer channel start register 0 */
/* +----- TS04 */
/* [Operation enable (start) trigger of channel 4] */
/* 0: No trigger operation */
/* 1: TE04 is set to 1 and the count operation becomes enabled */
TMIF04 = 0; /* Clear INTTM04 interrupt request */
while(TMIF04 == 0){ /* Wait 2 ms */
    NOP();
}
TMIF04 = 0; /* Clear INTTM04 interrupt request */
TT0L = 0b00010000; /* Stop operation of timer channel 4*/

```

```

TDR04 = 10000-1;    /* Set pulse cycle to 10 ms (1 [us/clock] × 10000 [count]) */

PFALL = 0b01111111; /* Port function register */
/*| | | | | | | | 0: Port (other than segment output) */
/*| | | | | | | | 1: Segment output */
/*| | | | | | | |+----- PF5L: P53 to P50/SEG50 to SEG53 */
/*| | | | | | | |+----- PF5H: P57 to P54/SEG46 to SEG49 */
/*| | | | |+----- PF9L: P93 to P90/SEG42 to SEG45 */
/*| | | |+----- PF9H: P97 to P94/SEG38 to SEG41 */
/*| | |+----- PF10: P102 to P100/SEG27 to SEG29 */
/*| |+----- PF14L: P143 to P140/SEG34 to SEG37 */
/*|+----- PF14H: P147 to P144/SEG30 to SEG33 */
/*+----- Be sure to set 0 */

ISC = 0b00000000; /* Input switch control register */
/*| | | | | | |+----- ISC0 */
/*| | | | | | [Switching external interrupt (INTP0) input] */
/*| | | | | | : 0 : Uses the input signal of the INTP0 pin as an external interrupt
*/
/*| | | | | | : 1 : Uses the input signal of the RxD3 pin as an external interrupt
*/

/*| | | | | | */
/*| | | | | |+----- ISC1 */
/*| | | | | [Switching channel 7 input of timer array unit] */
/*| | | | | : 0 : Uses the input signal of the TI07 pin as a timer input */
/*| | | | | : 1 : Uses the input signal of the RxD3 pin as a timer input */
/*| | | | | */
/*| | | | |+++----- ISC2 to ISC4: */
/*| | | | | [Control of TI04/SEGxx/P53, TI02/SEGxx/P52, and RxD3/SEGxx/P50] */
/*| | | | | : PF5L ISCn (2 to 4) */
/*| | | | | : 0 0: Port output */
/*| | | | | : 0 1: Port input, timer input, or serial data input */
/*| | | | | : 1 0: Segment output */
/*| | | | | */
/*| | | | | |+----- ISC2 */
/*| | | | | [RxD3/SEGxx/P50 schmitt trigger buffer control] */

```



```

/*|||||      : 0 : Disables input */
/*|||||      : 1 : Enables input */
/*||||| */
/*|||||+----- ISC3 */
/*|||||      [TI02/SEGxx/P52 schmitt trigger buffer control] */
/*|||||      : 0 : Disables input */
/*|||||      : 1 : Enables input */
/*||||| */
/*|||||+----- ISC4 */
/*|||||      [TI04/SEGxx/P53 schmitt trigger buffer control] */
/*|||||      : 0 : Disables input */
/*|||||      : 1 : Enables input */
/*||||| */
/*+++----- Be sure to set 0 */

VLCON = 1;      /* Enable voltage boost circuit and capacitor split circuit operation
*/

/* Wait for voltage boost wait time (500 ms or more) by using TM04 (10 ms) */

TS0L = 0b00010000;      /* Enable operation of timer channel 4 */
TMIF04 = 0;      /* Clear INTTM04 interrupt request */
for(count = 500/10 +1; count > 0; count--){
    while(TMIF04 == 0){      /* Wait 10 ms */
        NOP();
    }
    TMIF04 = 0;      /* Clear INTTM04 interrupt request */
}
TT0L = 0b00010000;      /* Stop operation of timer channel 4 */

SCOC = 1;      /* Output deselected wave from all COM and SEG pins */
LCDON = 1;      /* LCD display on */

/*-----
Specify the external interrupt valid edge
-----*/

EGNO = 0b00100110; /* Falling edge of INTP1, INTP2, and INTP5 valid */

```

```

/*-----
/*  Enable interrupts
/*  (To use interrupts, enable interrupts here.)
/*-----*/

DI();          /* Disable interrupt */

PIF1 = 0;      /* Clear key interrupt request */
PIF2 = 0;      /* Clear key interrupt request */
PIF5 = 0;      /* Clear key interrupt request */

RTCMK = 0;     /* Unmask RTC constant-period interrupt */
PMK1 = 0;     /* Unmask key interrupt */
PMK2 = 0;     /* Unmask key interrupt */
PMK5 = 0;     /* Unmask key interrupt */

RTCIF = 1;    /* Initial display */

}

/*****

I/O port setting

*****/
void fn_InitPort( void )
{
/*-----
Specify the digital I/O
-----*/

ADPC  = 0b00010000; /* A/D port configuration register */
/* |||+++++--- ADPC4 to ADPC0 */
/* |||          [Analog input (A)/digital I/O (D) switching] */
/* |||          +----- ANI15/P157 */
/* |||          |+++----- ANI10 to ANI8/P152 to P150 */
/* |||          |||+++++--- ANI7 to ANI0/P27 to P20 */
/* |||          00000: AAAAAAAAAA */

```

```

/* |||      00001: AAAAAAAAAAAD */
/* |||      00010: AAAAAAAAAAAD */
/* |||      00011: AAAAAAAAAADD */
/* |||      00100: AAAAAAAAAADD */
/* |||      00101: AAAAAAADDDDD */
/* |||      00110: AAAAAAADDDDD */
/* |||      00111: AAAAAAADDDDD */
/* |||      01000: AAAAADDDDDDD */
/* |||      01001: AAADD DDDDDDD */
/* |||      01010: AADD DDDDDDD */
/* |||      01111: ADDDD DDDDDDD */
/* |||      10000: DDDDD DDDDDDD */
/* +++----- Be sure to set 0 */

```

```

/*-----
Initialize port 0
-----*/
P0      = 0b00000000; /* Set the P00 to P02 output latches to low level */
PM0     = 0b11111000; /* Specify P00 to P02 as output ports */
                /* P00 to P02: Unused */

/*-----
Initialize port 1
-----*/
P1      = 0b00000000; /* Set the P10 to P17 output latches to low level */
PM1     = 0b00000000; /* Specify P10 to P17 as output ports */
                /* P10 to P17: Unused */

/*-----
Initialize port 2
-----*/
P2      = 0b00000000; /* Set the P20 to P27 output latches to low level */
PM2     = 0b00000000; /* Specify P20 to P27 as output ports */

/*-----
Initialize port 3
-----*/
P3      = 0b00000000; /* Set the P30 to P34 output latches to low level */

```

APPENDIX A PROGRAM LIST

```
PM3    = 0b11100111; /* Specify P33 and P34 as output ports */
        /* P33 and P34: Unused */

PU3    = 0b00000111; /* Connect on-chip pull-up resistors to P30 to P32 */

/*-----
Initialize port 4
-----*/

P4     = 0b00000000; /* Set the P40 and P41 output latches to low level */
PM4    = 0b11111100; /* Specify P40 and P41 as output ports */
        /* P40 and P41: Unused */

/*-----
Initialize port 5
-----*/

P5     = 0b00000000; /* Set the P50 to P57 output latches to low level */
PM5    = 0b00000000; /* Specify P50 to P57 as output ports */

/*-----
Initialize port 6
-----*/

P6     = 0b00000000; /* Set the P60 and P61 output latches to low level */
PM6    = 0b11111100; /* Specify P60 and P61 as output ports */
        /* P60 and P61: Unused */

/*-----
Initialize port 7
-----*/

P7     = 0b00000000; /* Set the P70 to P77 output latches to low level */
PM7    = 0b00000000; /* Specify P70 to P77 as output ports */
        /* P70 to P77: Unused */

/*-----
Initialize port 8
-----*/

P8     = 0b00000000; /* Set the P80 to P87 output latches to low level */
PM8    = 0b00000000; /* Specify P80 to P87 as output ports */
        /* P80 to P87: Unused */
```

```
/*-----  
Initialize port 9  
-----*/  
P9      = 0b00000000; /* Set the P90 to P97 output latches to low level */  
PM9     = 0b00000000; /* Specify P90 to P97 as output ports */  
          /* P90 to P97: Unused */  
  
/*-----  
Initialize port 10  
-----*/  
P10     = 0b00000000; /* Set the P100 to P102 output latches to low level */  
PM10    = 0b11111000; /* Specify P100 to P102 as output ports */  
          /* P100 to P102: Unused */  
  
/*-----  
Initialize port 11  
-----*/  
P11     = 0b00000000; /* Set the P110 and P111 output latches to low level */  
PM11    = 0b11111100; /* Specify P110 and P111 as output ports */  
          /* P110 and P111: Unused */  
  
/*-----  
Initialize port 12  
-----*/  
P12     = 0b00000000; /* Set the P120 output latch to low level */  
PM12    = 0b11111110; /* Specify P120 as output port */  
          /* P120 to P124: Unused */  
          /* * P121 to P124 are input-only ports. */  
  
/*-----  
Initialize port 13  
-----*/  
P13     =0b00000000; /* Set the P130 output latch to low level */  
          /* P130: Unused */  
  
/*-----  
Initialize port 14  
-----*/  
P14     = 0b00000000; /* Set the P140 to P147 output latches to low level */
```

```

PM14 = 0b00000000; /* Specify P140 to P147 as output ports */
        /* P140 to P147: Unused */

/*-----
Initialize port 15
-----*/

P15 = 0b00000000; /* Set the P150 to P152, and P157 output latches to low level */
PM15 = 0b01111000; /* Specify P150 to P152, and P157 as output ports */
        /* P150 to P152, and P157: Unused */
}

/*****

Low-voltage detection

-----

Secure a supply voltage of 4.2 V or more by using the function of low-voltage detector.
*****/

void fn_InitLvi( void )
{
    unsigned char ucCounter; /* Count variable */

    /* Set up the low-voltage detector */
    LVIMK = 1; /* Disable the INTLVI interrupt */
    LVISEL = 0; /* Specify VDD as the detection voltage*/
    LVIS = 0b00000000; /* Low-voltage detection level select register */
    /* |||||++++--- LVIS3 to LVIS0 */
    /* ||||| [Detection level] */
    /* ||||| 0000: VLVI0 (4.22 ±0.1 V) */
    /* ||||| 0001: VLVI1 (4.07 ±0.1 V) */
    /* ||||| 0010: VLVI2 (3.92 ±0.1 V) */
    /* ||||| 0011: VLVI3 (3.76 ±0.1 V) */
    /* ||||| 0100: VLVI4 (3.61 ±0.1 V) */
    /* ||||| 0101: VLVI5 (3.45 ±0.1 V) */
    /* ||||| 0110: VLVI6 (3.30 ±0.1 V) */
    /* ||||| 0111: VLVI7 (3.15 ±0.1 V) */
    /* ||||| 1000: VLVI8 (2.99 ±0.1 V) */
    /* ||||| 1001: VLVI9 (2.84 ±0.1 V) */
    /* ||||| 1010: VLVI10 (2.68 ±0.1 V) */

```

```

/* |||| 1011: VLVI11 (2.53 ±0.1 V) */
/* |||| 1100: VLVI12 (2.38 ±0.1 V) */
/* |||| 1101: VLVI13 (2.22 ±0.1 V) */
/* |||| 1110: VLVI14 (2.07 ±0.1 V) */
/* |||| 1111: VLVI15 (1.91 ±0.1 V) */
/* ++++----- Be sure to set 0 */
LVIMD = 0; /* Specify that an interrupt signal is generated when a low voltage
is detected */
LVION = 1; /* Enable low-voltage detection */

/* Make the system wait until the low-voltage detector stabilizes (10 us or more) */
for( ucCounter = 0; ucCounter < 4; ucCounter++ ){
    NOP();
}

/* Make the system wait until VLVI is less than or equal to VDD */
while( LVIF ){
    NOP();
}
LVION = 0; /* Stop the low-voltage detector */
}

/*****

Specify the clock frequency

-----

Specify the clock frequency so that the device can run on the internal high-speed oscillation
clock.
*****/
void fn_InitClock( void )
{
    CMC = 0b00010000; /* Clock operation mode */
    /* |||||+--- AMPH */
    /* ||||| [Control of high-speed system clock oscillation frequency] */
    /* ||||| 0: 2 MHz ≤ fMX < 10 MHz */
    /* ||||| 1: 10 MHz < fMX ≤ 20 MHz */
    /* |||||++---- AMPHS1 and AMPHS0 */
    /* ||||| [XT1 oscillator oscillation mode selection] */

```

```

/* ||||| 00: Low power consumption oscillation (default) */
/* ||||| 01: Normal oscillation */
/* ||||| 10: Ultra-low power consumption oscillation */
/* ||||| 11: Ultra-low power consumption oscillation */
/* |||+----- Be sure to set 0 */
/* |||+----- OSCSELS */
/* ||| [Subsystem clock pin operation mode] */
/* ||| 0: Input port mode */
/* ||| 1: XT1 oscillation mode */
/* |+----- Be sure to set 0 */
/* ++----- EXCLK/OSCSEL */
/* [High-speed system clock pin operation mode] */
/* 00: Input port mode */
/* 01: X1 oscillation mode */
/* 10: Input port mode */
/* 11: External clock input mode */

CSC = 0b10000000; /* Clock operation status control */
/* |||||+---- HIOSTOP */
/* ||||| [Internal high-speed oscillation clock operation control] */
/* ||||| 0: Internal high-speed oscillator operating */
/* ||||| 1: Internal high-speed oscillator stopped */
/* ||++++---- Be sure to set 0 */
/* |+----- XTSTOP */
/* | [Subsystem clock operation control] */
/* | 0: XT1 oscillator operating */
/* | 1: XT1 oscillator stopped */
/* +----- MSTOP */
/* [High-speed system clock operation control] */
/* 0: X1 oscillator operating */
/* 1: X1 oscillator stopped */

OSMC = 0b10000000; /* Operation speed mode */
/* |||||+---- FSEL/FLPC */
/* ||||| [fCLK frequency selection] */
/* ||||| 00: Operates at a frequency of 10 MHz or less (default) */
/* ||||| 01: Operates at a frequency higher than 10 MHz */
/* ||||| 10: Operates at a frequency of 1 MHz */
/* ||||| 11: Setting prohibited */

```



```

/* |+++++----- Be sure to set 0 */
/* +----- RTCLPC */
/*           [Setting in subsystem clock HALT mode] */
/*           0: Enables subsystem clock supply to peripheral functions */
/*           1: Stops subsystem clock supply to peripheral functions except */
/*               real-time counter */

CKC = 0b00001000; /* Clock selection */

/* |+|+++++--- CSS/MCM0/MDIV2 to MDIV0 */
/* | |           [Selection of CPU/peripheral hardware clock (fCLK)] */
/* | |           00x000: fIH */
/* | |           00x001: fIH/2 (default) */
/* | |           00x010: fIH/2^2 */
/* | |           00x011: fIH/2^3 */
/* | |           00x100: fIH/2^4 */
/* | |           00x101: fIH/2^5 */
/* | |           01x000: fMX */
/* | |           01x001: fMX/2 */
/* | |           01x010: fMX/2^2 */
/* | |           01x011: fMX/2^3 */
/* | |           01x100: fMX/2^4 */
/* | |           01x101: fMX/2^5 */
/* | |           1x0xxx: fSUB */
/* | |           1x1xxx: fSUB/2 */
/* | |           ( x : don't care ) */
/* | +----- MCS <Read Only> */
/* |           [Status of Main system clock (fMAIN)] */
/* |           0: Internal high-speed oscillation clock (fIH) */
/* |           1: High-speed system clock (fMX) */
/* +----- CLS <Read Only> */
/*           [Status of CPU/peripheral hardware clock (fCLK)] */
/*           0: Main system clock (fMAIN) */
/*           1: Subsystem clock (fSUB) */
}

/*****
Main loop

```

```

*****/
void main(void)
{
    unsigned char ucKeyin;          /* Key input buffer */
    unsigned char ucUnderbarStatus; /* '_' display status */
    unsigned char ucNumWork;       /* Work for function parameters */

    /*-----
    Initialize the RAM
    -----*/

    ucAdjStatus = 0x00;    /* Set clock display status */

    /*-----
    Normal processing
    -----*/

    while (1){
        HALT();            /* Wait for key input or 1 minute */
        NOP();

        /*-----
        Update clock display after 1 minute has passed
        -----*/

        if((RTCIF == 1) && (ucAdjStatus == 0x00)){ /* Lapse of 1 minute during clock display
*/
            RTCIF = 0;          /* Clear constant-period interrupt request */
            if(ucAdjStatus == 0x00){ /* Clock being displayed? */
                RWAIT = 1;      /* Set RTC value read/write mode */
                while(RWST == 0){ /* Wait for RTC read/write mode */
                    RWAIT = 1;  /* Set RTC value read/write mode */
                }
                ucMinute = MIN; /* Read minutes */
                ucHour = HOUR; /* Read hours */

                RWAIT = 0;      /* Set to start RTC count operation */
                while(RWST == 1){ /* Wait for start of RTC count operation */
                    RWAIT = 0;  /* Set to start RTC count operation */

```

```

    }
}

}

/*-----
Eliminate noise from key input
-----*/

if((PIF1 == 1) || (PIF2 == 1) || (PIF5 == 1)) { /* Key input? */
    PIF1 = 0; /* Clear key input interrupt request */
    PIF2 = 0; /* Clear key input interrupt request */
    PIF5 = 0; /* Clear key input interrupt request */

    ucKeyin = (P3 & 0b00000111); /* Read key status */
    if(ucKeyin != 0b00000111){ /* Key input */

        TS0L = 0b00010000; /* Enable operation of timer channel 4 */
        TMIF04 = 0; /* Clear INTTM04 interrupt request */
        while(TMIF04 == 0){ /* Wait 10 ms */
            NOP();
        }
        TMIF04 = 0; /* Clear INTTM04 interrupt request */
        TT0L = 0b00010000; /* Stop operation of timer channel 4 */

        PIF1 = 0; /* Clear key input interrupt request */
        PIF2 = 0; /* Clear key input interrupt request */
        PIF5 = 0; /* Clear key input interrupt request */

/*-----
Key processing
-----*/

if(ucKeyin == (P3 & 0b00000111)){ /* Key status stabilized */

    switch(ucKeyin){ /* Distribution by input key */
        case 0b00000110: /* SET key */
            switch(ucAdjStatus){ /* Distribution by clock adjustment status */

```

```

case 0x00:      /* Clock being displayed */
    RWAIT = 1;          /* Set RTC value read/write mode */
    while(RWST == 0){ /* Wait for RTC read/write mode */
        RWAIT = 1;      /* Set RTC value read/write mode */
    }
    ucMinute = MIN;     /* Read minutes */
    ucHour = HOUR;     /* Read hours */

    RWAIT = 0;         /* Set to start RTC count operation */
    while(RWST == 1){ /* Wait for start of RTC count operation
*/

    RWAIT = 0;         /* Set to start RTC count operation */
    }
    ucAdjStatus = 0x01; /* Assume minutes being adjusted */
    break;

case 0x01:      /* Minutes being adjusted */
    ucAdjStatus = 0x02; /* Assume hours being adjusted */
    break;

case 0x02:      /* Hours being adjusted */
    ucAdjStatus = 0x00; /* Assume clock being displayed */
    RWAIT = 1;          /* Set RTC value read/write mode */
    while(RWST == 0){ /* Wait for RTC read/write mode */
        RWAIT = 1;      /* Set RTC value read/write mode */
    }

    MIN = ucMinute;     /* Set minutes */
    HOUR = ucHour;     /* Set hours */
    SEC = 0x00;         /* Reset seconds */

    RWAIT = 0;         /* Set to start RTC count operation */
    while(RWST == 1){ /* Wait for start of RTC count operation
*/

    RWAIT = 0;         /* Set to start RTC count operation */
    }
    break;

default:

```

```

        ;
    }
    break;

case 0b00000101:    /* UP key */
    switch(ucAdjStatus){ /* Distribution by clock adjustment status
*/
        case 0x00:    /* Clock being displayed */
            break;

        case 0x01:    /* Minutes being adjusted */
            ucMinute = adbcdb(ucMinute, 1); /* Minutes + 1 */
            if(ucMinute == 0x60){
                ucMinute = 0x00;          /* Carry correction */
            }
            break;

        case 0x02:    /* Hours being adjusted */
            ucHour = adbcdb(ucHour, 1); /* Hours + 1 */
            if(ucHour == 0x24){
                ucHour = 0x00;          /* Carry correction */
            }
            break;

        default:
            ;
    }
    break;

case 0b00000011:    /* DOWN key */
    switch(ucAdjStatus){ /* Distribution by clock adjustment status
*/
        case 0x00:    /* Clock being displayed */
            break;

        case 0x01:    /* Minutes being adjusted */
            ucMinute = sbbcdb(ucMinute, 1); /* Minutes - 1 */

```

```

        if(ucMinute == 0x99){
            ucMinute = 0x59;          /* Borrow correction */
        }
        break;

    case 0x02:          /* Hours being adjusted */
        ucHour = sbbcdb(ucHour, 1); /* Hours - 1 */
        if(ucHour == 0x99){
            ucHour = 0x23;          /* Borrow correction */
        }
        break;

    default:
        ;
    }
    break;

    default:          /* Multiple depressions */
        ;          /* Invalid */
    }
}
}
}

/*-----
Clock display
-----*/

if(ucAdjStatus == 0x02){          /* Hours being adjusted? */
    ucUnderbarStatus = 1;          /* Yes, '_' displayed */
}
else{
    ucUnderbarStatus = 0;          /* No, '_' not displayed */
}

ucNumWork = (ucHour >> 4) & 0x0f; /* Digit of 10 hours */
if(ucNumWork == 0x00){
    ucNumWork = 0x0a;          /* Zero-suppression */
}

```

```

}
fn_LcdNum(&SEG4, ucNumWork, ucUnderbarStatus); /* Display digit of 10 hours */

ucNumWork = ucHour & 0x0f; /* Digit of 1 hour */
fn_LcdNum(&SEG10, ucNumWork, ucUnderbarStatus); /* Display digit of 1 hour */

fn_LcdNum(&SEG16, 0x0b, 0x00); /* ':' displayed */

if(ucAdjStatus == 0x01){ /* Minutes being adjusted? */
    ucUnderbarStatus = 1; /* Yes, '_' displayed */
}
else{
    ucUnderbarStatus = 0; /* No, '_' not displayed */
}

ucNumWork = (ucMinute >> 4) & 0x0f; /* Digit of 10 minutes Digit of
*/
fn_LcdNum(&SEG22, ucNumWork, ucUnderbarStatus); /* Display digit of 10 minutes */

ucNumWork = ucMinute & 0x0f; /* Digit of 1 minute */
fn_LcdNum(&SEG28, ucNumWork, ucUnderbarStatus); /* Display digit of 1 minute */

}
}
/*****
Font data definition
*****/
const unsigned char aucFontData[12][6] = {
    /* COM76543210 */
    0b00111110, /* '0' */
    0b01000001,
    0b01000001,
    0b01000001,
    0b00111110,
    0b00000000
},

```

```
{/* COM76543210 */
  0b00000000, /* '1' */
  0b00000000,
  0b00000010,
  0b01111111,
  0b00000000,
  0b00000000
},
```

```
{/* COM76543210 */
  0b01000010, /* '2' */
  0b01100001,
  0b01010001,
  0b01001001,
  0b01000110,
  0b00000000
},
```

```
{/* COM76543210 */
  0b00100010, /* '3' */
  0b01001001,
  0b01001001,
  0b01001001,
  0b00110110,
  0b00000000
},
```

```
{/* COM76543210 */
  0b00111100, /* '4' */
  0b00100010,
  0b00100001,
  0b01111111,
  0b00100000,
  0b00000000
},
```

```
{/* COM76543210 */
  0b00100111, /* '5' */
  0b01000101,
```



```

    0b01000101,
    0b01000101,
    0b00111001,
    0b00000000
},

{ /* COM76543210 */
    0b00111110,      /* '6' */
    0b01001001,
    0b01001001,
    0b01001001,
    0b00110010,
    0b00000000
},

{ /* COM76543210 */
    0b00000001,      /* '7' */
    0b01110001,
    0b00001001,
    0b00000101,
    0b00000011,
    0b00000000
},

{ /* COM76543210 */
    0b00110110,      /* '8' */
    0b01001001,
    0b01001001,
    0b01001001,
    0b00110110,
    0b00000000
},

{ /* COM76543210 */
    0b00100110,      /* '9' */
    0b01001001,
    0b01001001,
    0b01001001,
    0b00111110,

```

```

    0b00000000
},

{ /* COM76543210 */
    0b00000000, /* \ ' */
    0b00000000,
    0b00000000,
    0b00000000,
    0b00000000,
    0b00000000
},

{ /* COM76543210 */
    0b00000000, /* \ : ' */
    0b00000000,
    0b00100100,
    0b00000000,
    0b00000000,
    0b00000000
}
};

/*****

Numeric display

The numeric value of ucNum (BCD) is displayed on the LCD display data memory specified
by ucSeg. '_' is appended if the value of ucUnderbar is other than 0.

*****/
void fn_LcdNum(unsigned char *ucSeg, unsigned char ucNum, unsigned char ucUnderbar)
{
    unsigned char ucFontIndex; /* Index for font */
    unsigned char ucFontWork; /* Store font for underbar */
    unsigned char *ucSegPtr; /* LCD display data memory pointer */

    if(ucUnderbar == 0){
        ucFontWork = 0b00000000; /* No underbar */
    }
    else{

```

```
    ucFontWork = 0b10000000; /* Underbar */
}

for(ucFontIndex = 0, ucSegPtr = ucSeg; ucFontIndex < 6 ;ucFontIndex++ ,ucSegPtr++){
    /* Store font data in LCD display data memory */
    *ucSegPtr = (aucFontData[ucNum][ucFontIndex] | ucFontWork);
}
}
```

APPENDIX B REVISION HISTORY

Edition	Date Published	Page	Revision
1st edition	September 2009	–	–

*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/>