To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

RENESAS

Application Note

78K0/Kx2-L

Sample Program (A/D Converter)

Successive A/D Conversion & Average Value Calculation

This document describes an operation overview of the sample program and how to use it, as well as how to set up and use the A/D converter. In the sample program, A/D conversion is performed four times each for the analog input from the analog input channels ANI0 and ANI1, and each converted result and the average value of the converted data are saved into the RAM area.

Target devices

78K0/KY2-L microcontroller
78K0/KA2-L microcontroller
78K0/KB2-L microcontroller
78K0/KC2-L microcontroller

CONTENTS

Document No. U19693EJ1V0AN00 (1st edition)
Date Published September 2009 N

- **The information in this document is current as of May, 2009. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. In addition, NEC Electronics products are not taken measures to prevent radioactive rays in the product design. When customers use NEC Electronics products with their products, customers shall, on their own responsibility, incorporate sufficient safety measures such as redundancy, fire-containment and anti-failure features to their products in order to avoid risks of the damages to property (including public or social property) or injury (including death) to persons, as the result of defects of NEC Electronics products.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

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

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

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

(Note)

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

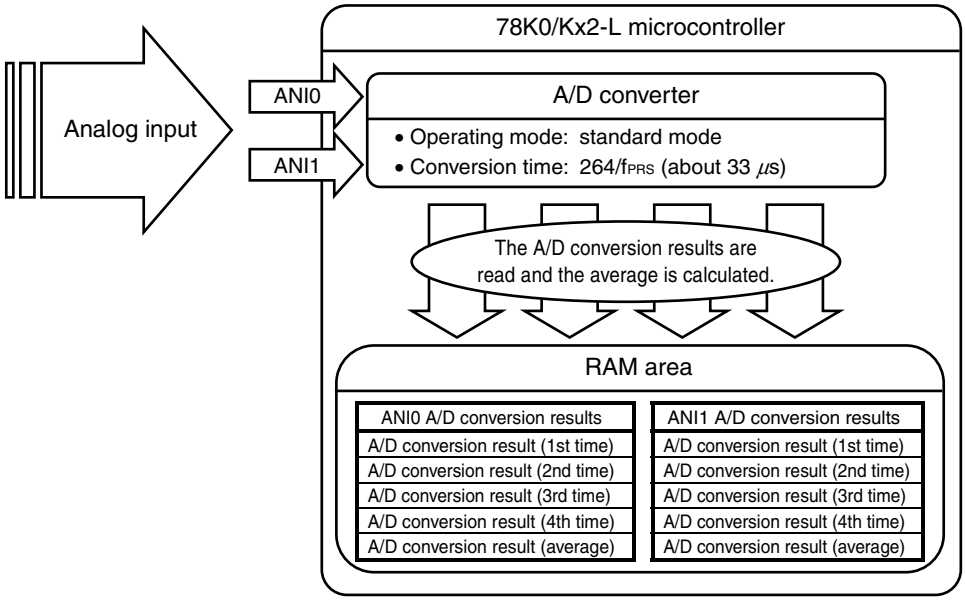
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E0904E

CHAPTER 1 OVERVIEW

An example of using the A/D converter is presented in this sample program. A/D conversion is performed four times each for the analog input from the analog input channels ANI0 and ANI1, and each converted result and the average value of the converted data are saved into the RAM area.

[Operation overview]



1.1 Primary Initial Settings

The primary initial settings are as follows.

<Option byte settings>

- Allowing the internal low-speed oscillator to be programmed to stop
- Disabling the watchdog timer
- Setting the internal high-speed oscillation clock frequency to 8 MHz
- Disabling LVI from being started by default

<Settings during initialization immediately after a reset ends>

- Specifying the ROM and RAM sizes
- Setting up I/O ports
  - Specifying the P20/ANI0 and P21/ANI1 pins as analog input pins
- Checking whether V_{DD} is 2.7 V or more by using the low-voltage detector[Note]
- Specifying that the CPU clock and peripheral hardware clock run on the internal high-speed oscillation clock (8 MHz)
- Stopping the internal low-speed oscillator
- Disabling peripheral hardware not to be used
- Setting up the A/D converter
  - Specifying the standard mode as the operating mode
  - Specifying 264/f_{PRS} (about 33 μs) as the A/D conversion time

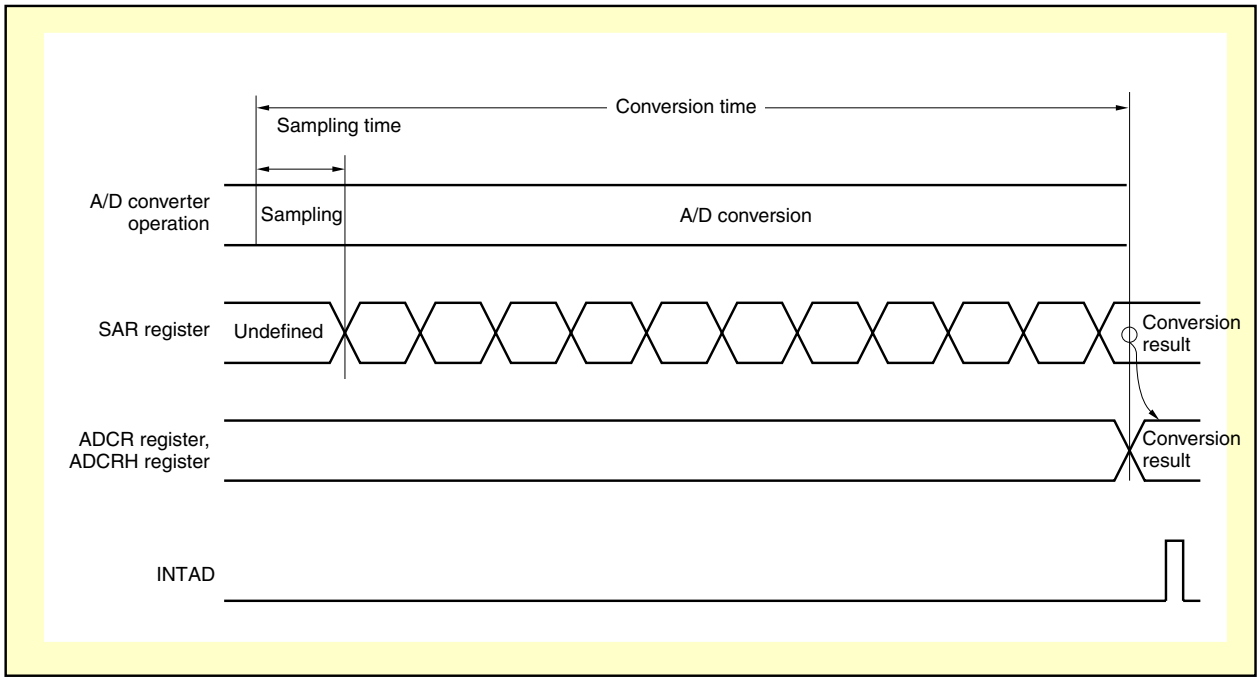
Note For details about the low-voltage detector, refer to the 78K0/Kx2-L User's Manual.

1.2 Processing After Main Loop

After completion of the initial settings, A/D conversion operation is started whereupon A/D conversion is performed four times for the analog input from ANI0 and the converted result is saved into the RAM area. A/D conversion operation is stopped after the same processing is performed for the analog input from ANI1. After A/D conversion operation is stopped, the average value of the four A/D conversions performed is calculated for ANI0 and ANI1, and the average values are saved into the RAM area.

After completion of the initial settings, successive four-time A/D conversion processing (2 channels) and average value calculation processing (2 channels), as mentioned above, are repeated. In this manner, variation in the analog inputs can be reduced by performing A/D conversion multiple times and using the average values calculated from the converted result. Furthermore, power consumption can be reduced by stopping A/D conversion operation when calculating the average values.

Figure 1-1. Basic A/D Converter Operation (A/D Conversion: 1 Time)



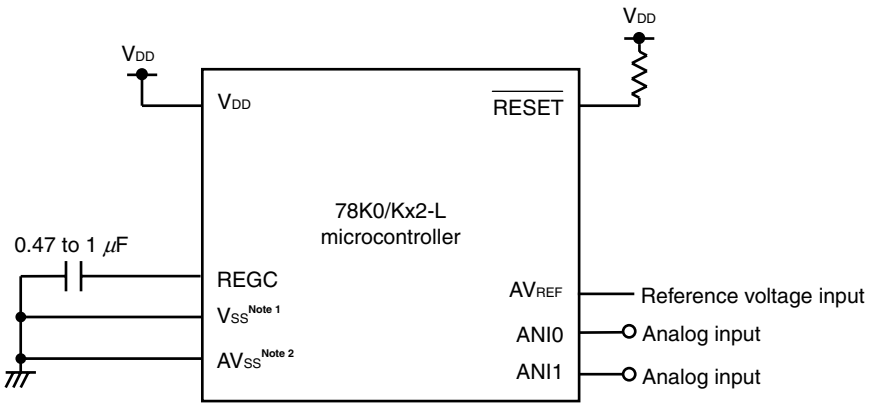
Caution For cautions when using the device, refer to the 78K0/Kx2-L User's Manual.

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.



Notes 1. This is shared with AVSS in the 78K0/KY2-L and 78K0/KA2-L.

2. This is provided only in the 78K0/KB2-L and 78K0/KC2-L.

Cautions 1. Use the microcontroller at a voltage in the range of 2.94 V ≤ VDD ≤ 5.5 V.

2. Connect REGC to VSS via a capacitor (0.47 to 1 μF).

3. For the 78K0/KY2-L and 78K0/KA2-L, VSS is also used as the ground potential for the A/D converter. Be sure to connect VSS to a stable GND.

4. Make the AVSS pin have the same potential as VSS and connect it directly to GND (only for the 78K0/KB2-L and 78K0/KC2-L microcontrollers).

5. Make sure that the AVREF voltage is 2.7 V or more, 5.5 V or less, and VDD or less.

6. 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 VDD or VSS via a resistor.**

7. In this sample program, the P121/X1/TOOLC0 and P122/X2/EXCLK/TOOLD0 pins are used for on-chip debugging.

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 the flow charts.

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 and internal low-speed oscillator and selecting the internal high-speed oscillation clock frequency.)	●	●
Kx2-L_ADC.prw	Work space file for integrated development environment PM+		●
Kx2-L_ADC.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.

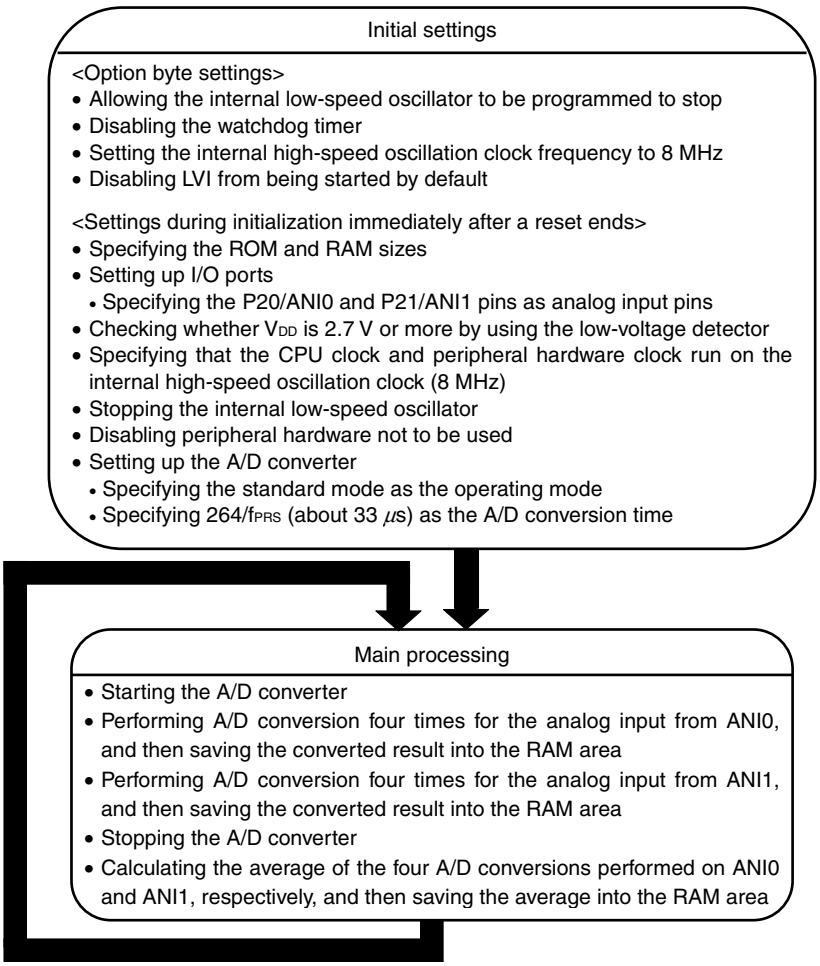
- A/D converter: Performs 10-bit resolution A/D conversion.
- ANI0 and ANI1: Used as the analog input channels of the A/D converter.
- Low-voltage detector: Used to check that V_{DD} is 2.7 V or more.

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, and setting of the A/D converter are performed. After completion of the initial settings, A/D conversion operation is started whereupon A/D conversion is performed four times for the analog input from ANI0 and the converted result is saved into the RAM area. A/D conversion operation is stopped after the same processing is performed for the analog input from ANI1. After A/D conversion operation is stopped, the average value of the four A/D conversions performed is calculated for ANI0 and ANI1, and the average values are saved into the RAM area.

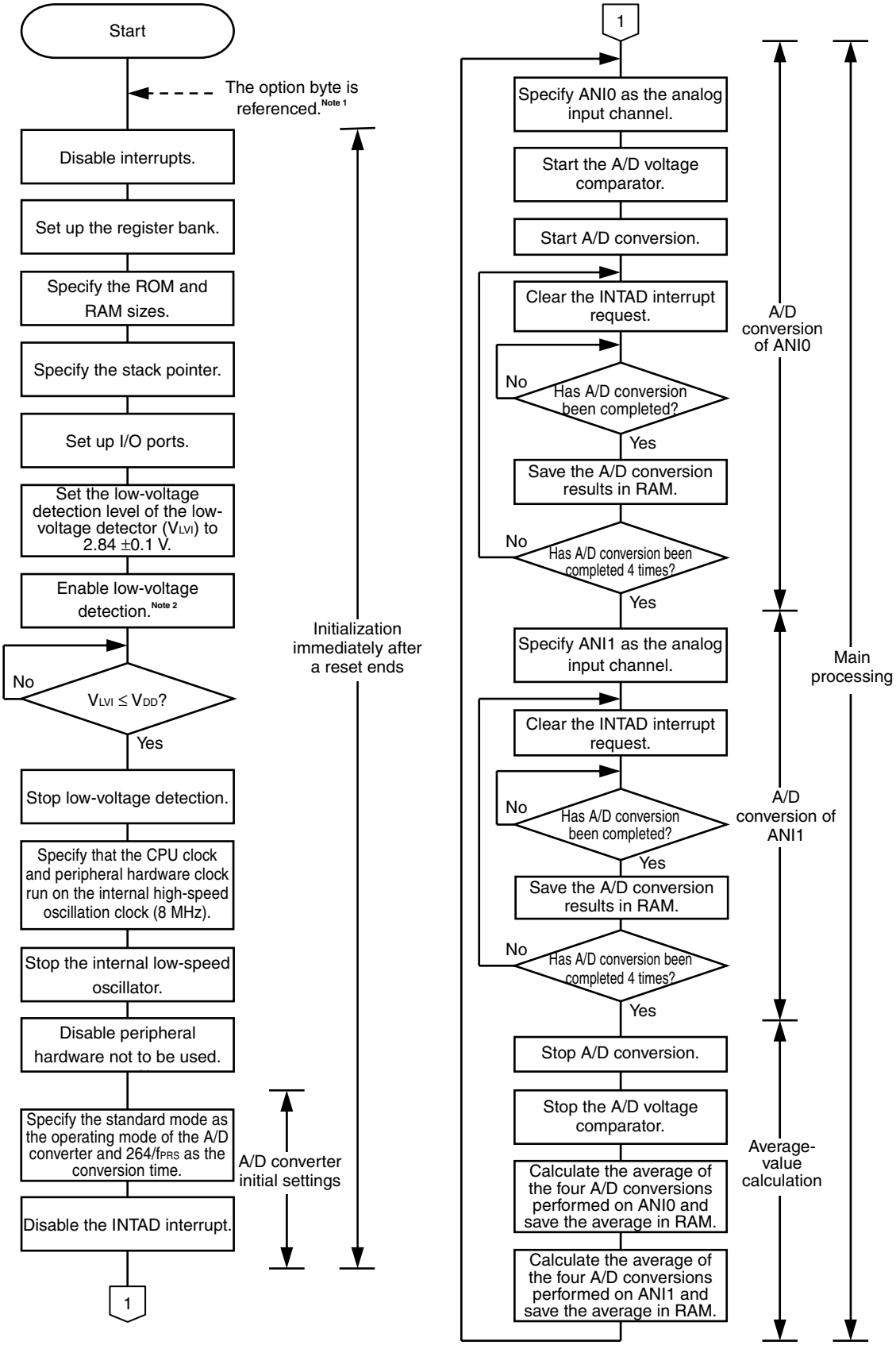
After completion of the initial settings, successive four-time A/D conversion processing (2 channels) and average value calculation processing (2 channels), as mentioned above, are repeated. In this manner, variation in the analog inputs can be reduced by performing A/D conversion multiple times and using the average values calculated from the converted result. Furthermore, power consumption can be reduced by stopping A/D conversion operation when calculating the average values.

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



3.4 Flow Charts

The flow charts for the sample program are 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:

- Allowing the internal low-speed oscillator to be programmed to stop
- Disabling the watchdog timer
- Setting the internal high-speed oscillation clock frequency to 8 MHz
- Disabling LVI from being started by default

2. The low-voltage detector is enabled, and then the system is made to wait at least 10 μs until the low-voltage detector stabilizes.

CHAPTER 4 SETTING METHODS

This chapter describes how to set up the A/D converter and provides software coding examples and details about the input voltage and A/D conversion results.

For other initial settings, refer to the 78K0/Kx2-L Sample Program (Initial Settings) LED Lighting Switch Control Application Note.

For how to set registers, refer to the 78K0/Kx2-L User's Manual.

For assembler instructions, refer to the 78K/0 Series Instructions User's Manual.

4.1 Setting up A/D Converter

The A/D converter uses the following five types of registers:

- A/D converter mode register 0 (ADM0)
- A/D port configuration registers 0, 1 (ADPC0, ADPC1)
- Analog input channel specification register (ADS)
- Port mode registers 1, 2 (PM1, PM2)
- 10-bit A/D conversion result register (ADCR)[Note]

Note ADCR can only be read and is therefore not set up.

[Example of the procedure for setting up the A/D converter]

<1> Select the A/D conversion time and operating mode by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of A/D converter mode register 0 (ADM0).

<2> Set bit 0 (ADCE) of ADM0 to 1.

<3> Specify the channels to be used as analog input channels by using the A/D port configuration registers 0 and 1 (ADPC0 and ADPC1) and port mode registers 1 and 2 (PM1 and PM2).

<4> Specify using a programmable gain amplifier when specifying PGA output for the analog inputs and specify using a single amplifier when specifying operational amplifier output for the analog inputs.[Note]

<5> Select the channels to be used by using the analog input channel specification register (ADS).

<6> Start A/D conversion by setting bit 7 (ADCS) of ADM0 to 1.

Note For details about operational amplifiers, refer to **CHAPTER 13 OPERATIONAL AMPLIFIERS** in the 78K0/Kx2-L User's Manual.

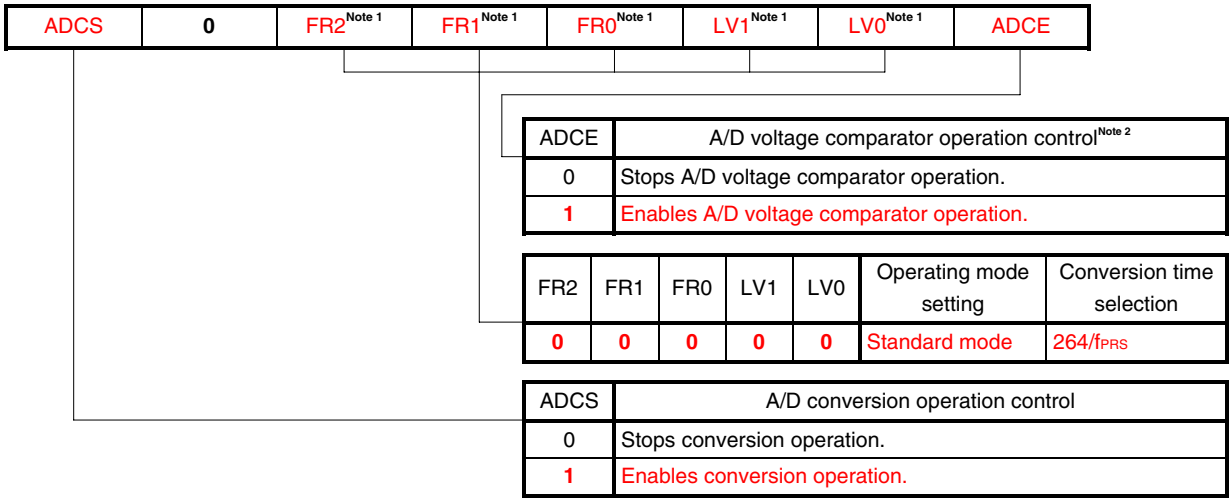
Cautions 1. Leave an interval of at least 1 μs between steps <2> and <6>.

2. Step <2> may be performed anytime before step <4>.

(1) A/D converter mode register 0 (ADM0)

This register sets the conversion time for the analog input to be A/D converted, and starts or stops conversion operation.

Figure 4-1. Format of A/D Converter Mode Register 0 (ADM0)



ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation.
1	Enables A/D voltage comparator operation.

FR2	FR1	FR0	LV1	LV0	Operating mode setting	Conversion time selection
0	**0**	**0**	**0**	**0**	Standard mode	264/f_{PRS}

ADCS	A/D conversion operation control
0	Stops conversion operation.
1	Enables conversion operation.

Notes 1. For details about settings of FR2 to FR0, LV1, LV0, and A/D conversion, refer to **Table 4-2 A/D Conversion Time Selection**.

2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and the time from starting the operation until it stabilizes takes 1 μs. The conversion data, therefore, becomes valid starting from the first conversion data, by setting ADCS to 1 after at least 1 μs elapses since ADCE was set to 1. If ADCS is set to 1 without waiting for at least 1 μs, ignore the first conversion data.

3. Be sure to clear bit 6 to “0”.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

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

Table 4-1. ADCS and ADCE Settings

ADCS	ADCE	A/D Conversion Operation
0	0	Stopped (No DC power consumption path exists.)
0	1	Conversion wait mode (Only the A/D voltage comparator consumes power.)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator operation)

Table 4-2. A/D Conversion Time Selection

<1> $4.0 \leq AV_{REF} \leq 5.5$ V

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	
0	0	0	0	0	Standard	264/f_{PRS}	66.0 μs	33.0 μs	26.4 μs	f_{PRS}/12
0	0	1				176/f_{PRS}	44.0 μs	22.0 μs	17.6 μs	f_{PRS}/8
0	1	0				132/f_{PRS}	33.0 μs	16.5 μs	13.2 μs	f_{PRS}/6
0	1	1				88/f_{PRS}	22.0 μs	11.0 μs	8.8 μs	f_{PRS}/4
1	0	0				66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs	f_{PRS}/3
1	0	1				44/f_{PRS}	11.0 μs	Setting prohibited		f_{PRS}/2
1	1	0				33/f_{PRS}	8.25 μs	Setting prohibited		f_{PRS}/1.5
1	1	1				22/f_{PRS}	Setting prohibited			f_{PRS}
1	0	1	1	1	High-speed	44/f_{PRS}	11.0 μs	5.5 μs	4.4 μs	f_{PRS}/2
1	1	1				22/f_{PRS}	5.5 μs	Setting prohibited		f_{PRS}
1	0	0	1	0	Maximum speed	66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs	f_{PRS}/3
1	1	0				33/f_{PRS}	8.25 μs	4.125 μs	3.3 μs	f_{PRS}/1.5
Other than the above					Setting prohibited					

<2> $2.7 \leq AV_{REF} < 4.0$ V

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	
0	0	0	0	0	Standard	264/f_{PRS}	66.0 μs	33.0 μs	26.4 μs	f_{PRS}/12
0	0	1				176/f_{PRS}	44.0 μs	22.0 μs	17.6 μs	f_{PRS}/8
0	1	0				132/f_{PRS}	33.0 μs	16.5 μs	13.2 μs	f_{PRS}/6
0	1	1				88/f_{PRS}	22.0 μs	11.0 μs	8.8 μs	f_{PRS}/4
1	0	0				66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs	f_{PRS}/3
1	0	1				44/f_{PRS}	11.0 μs	Setting prohibited		f_{PRS}/2
1	1	0				33/f_{PRS}	8.25 μs	Setting prohibited		f_{PRS}/1.5
1	1	1				22/f_{PRS}	Setting prohibited			f_{PRS}
0	0	1	1	1	High-speed	176/f_{PRS}	44.0 μs	22.0 μs	17.6 μs	f_{PRS}/8
0	1	0				132/f_{PRS}	33.0 μs	16.5 μs	13.2 μs	f_{PRS}/6
0	1	1				88/f_{PRS}	22.0 μs	11.0 μs	8.8 μs	f_{PRS}/4
1	0	0				66/f_{PRS}	16.5 μs	8.25 μs	6.6 μs	f_{PRS}/3
1	0	1				44/f_{PRS}	11.0 μs	5.5 μs	4.4 μs	f_{PRS}/2
1	1	0				33/f_{PRS}	8.25 μs	Setting prohibited		f_{PRS}/1.5
1	1	1				22/f_{PRS}	5.5 μs	Setting prohibited		f_{PRS}
Other than the above					Setting prohibited					

<3> 1.8 ≤ AV_{REF} < 2.7 V

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{PRS} = 4 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	
0	0	0	0	1	Low-voltage	528/f_{PRS}	Setting prohibited	66.0 μs	52.8 μs	f_{PRS}/12
0	0	1				352/f_{PRS}	Setting prohibited	44.0 μs	Setting prohibited	f_{PRS}/8
0	1	0				264/f_{PRS}	66.0 μs	Setting prohibited		f_{PRS}/6
0	1	1				176/f_{PRS}	44.0 μs	Setting prohibited		f_{PRS}/4
1	0	0				132/f_{PRS}	Setting prohibited			f_{PRS}/3
1	0	1				88/f_{PRS}	Setting prohibited			f_{PRS}/2
1	1	0				66/f_{PRS}	Setting prohibited			f_{PRS}/1.5
1	1	1				44/f_{PRS}	Setting prohibited			f_{PRS}
Other than the above					Setting prohibited					

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

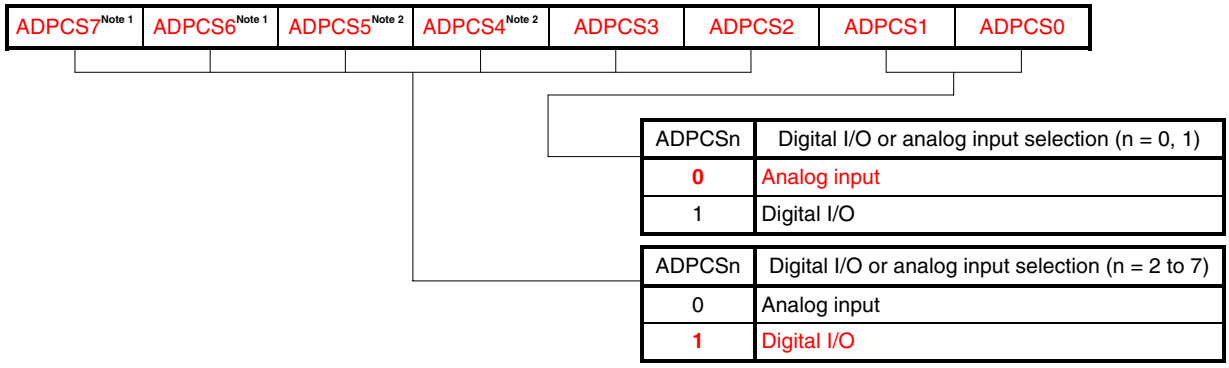
2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

(2) A/D port configuration registers 0, 1 (ADPC0, ADPC1)

ADPC0 switches the P20/AMP0-/ANI0 to P27/ANI7 pins to digital I/O or analog input of port. Each bit of ADPC0 corresponds to a pin of port 2 and can be specified in 1-bit units. ADPC1 switches the P10/AMP1-/ANI8 to P12/AMP1+/ANI10 pins to digital I/O or analog input of port. Each bit of ADPC1 corresponds to a pin of P10 to P12 in port 1 and can be specified in 1-bit units.

Figure 4-2. Format of A/D Port Configuration Register 0 (ADPC0)



ADPCSn	Digital I/O or analog input selection (n = 0, 1)
0	Analog input
1	Digital I/O

ADPCSn	Digital I/O or analog input selection (n = 2 to 7)
0	Analog input
1	Digital I/O

Notes 1. This bit can be set only in the 78K0/KC2-L. Be sure to clear this bit to 0 in the 78K0/KY2-L, 78K0/KA2-L, and 78K0/KB2-L.

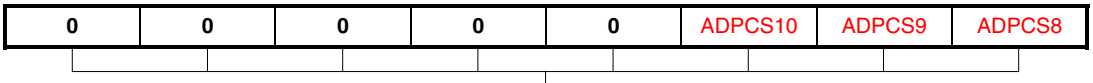
2. This bit can be set only in the 78K0/KA2-L and 78K0/KC2-L. Be sure to clear this bit to 0 in the 78K0/KY2-L and 78K0/KB2-L.

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 2 (PM2).

2. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the peripheral hardware clock (f_{PRS}) is stopped.

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

Figure 4-3. Format of A/D Port Configuration Register 1 (ADPC1)
(78K0/KB2-L and 78K0/KC2-L Only)



ADPCSn	Digital I/O or analog input selection (n = 8 to 10)
0	Analog input
1	Digital I/O

Cautions 1. Set the pin set to analog input to the input mode by using port mode register 1 (PM1).

2. If data is written to ADPC1, a wait cycle is generated. Do not write data to ADPC1 when the peripheral hardware clock (f_{PRS}) is stopped.

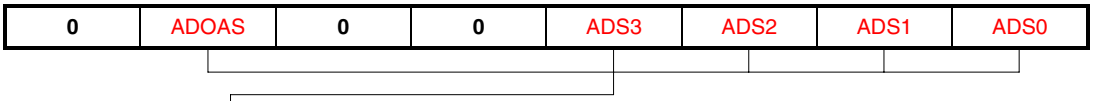
3. Be sure to clear bits 7 to 3 to "0".

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

(3) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Figure 4-4. Format of Analog Input Channel Specification Register (ADS)



ADOAS	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	**0**	**0**	**0**	**0**	ANI0	P20/ANI0 pin
0	**0**	**0**	**0**	**1**	ANI1	P21/ANI1 pin
0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	ANI8	P10/ANI8 pin
0	1	0	0	1	ANI9	P11/ANI9 pin
0	1	0	1	0	ANI10	P12/ANI10 pin
1	x	x	x	x	PGAIN[Note]	PGA output signal[Note]
Other than the above					Setting prohibited	

Note Setting permitted in products with operational amplifier

Cautions 1. Be sure to clear bits 7, 5, and 4 to "0".

2. Set a channel to be used for A/D conversion in the input mode by using port mode registers 1 and 2 (PM1 and PM2).

3. Set ADS after PGA operation setting when selecting the PGA output signal as analog input. (For details about operational amplifiers, refer to CHAPTER 13 OPERATIONAL AMPLIFIERS in the 78K0/Kx2-L User's Manual.)

4. If data is written to ADS, a wait cycle is generated. Do not write data to ADS when the peripheral hardware clock (f_{PRS}) is stopped.

Remarks 1. x: don't care

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

(4) Port mode registers 1, 2 (PM1, PM2)

When using the ANI8/AMP1-/P10 to ANI10/AMP1+/P12 and ANI0/AMP0-/P20 to ANI7/P27 pins for analog input port, set PM10 to PM12 and PM20 to PM27 to 1. The output latches of P10 to P12 and P20 to P27 at this time may be 0 or 1. If PM10 to PM12 and PM20 to PM27 are set to 0, they cannot be used as analog input port pins.

Figure 4-5. Format of Port Mode Register 1 (PM1)
(78K0/KB2-L and 78K0/KC2-L Only)

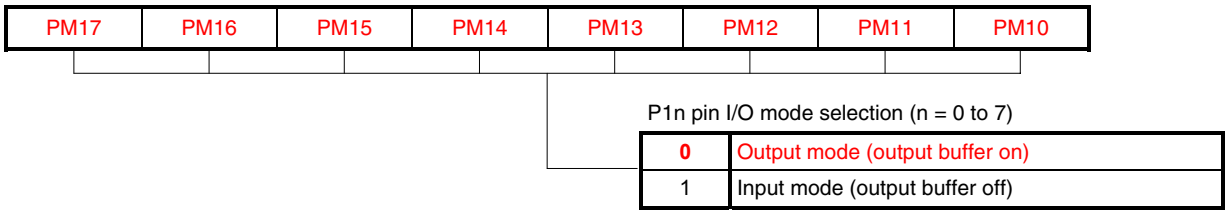
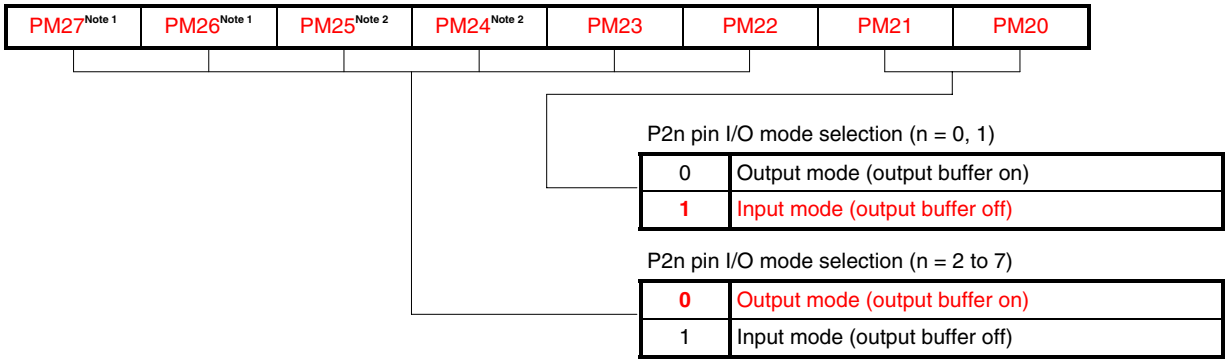


Figure 4-6. Format of Port Mode Register 2 (PM2)



Notes 1. This bit can be set only in the 78K0/KC2-L. Be sure to set this bit to 1 in the 78K0/KY2-L, 78K0/KA2-L, and 78K0/KB2-L.

2. This bit can be set only in the 78K0/KA2-L and 78K0/KC2-L. Be sure to set this bit to 1 in the 78K0/KY2-L and 78K0/KB2-L.

Remarks 1. A/D converter analog input pins differ depending on products.

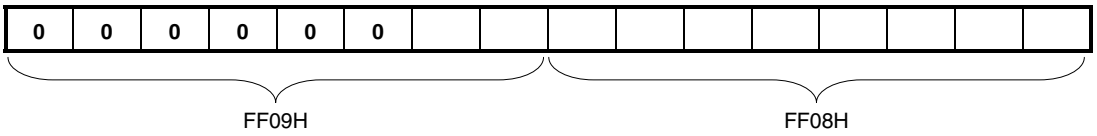
- 78K0/KY2-L: ANI0 to ANI3
- 78K0/KA2-L: ANI0 to ANI5
- 78K0/KB2-L: ANI0 to ANI3, ANI8 to ANI10
- 78K0/KC2-L: ANI0 to ANI10

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

(5) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The higher 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 2 bits of the conversion result are stored in FF09H and the lower 8 bits of the conversion result are stored in FF08H.

Figure 4-7. Format of 10-bit A/D Conversion Result Register (ADCR)



Cautions 1. When writing to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0 and 1 (ADPC0 and ADPC1), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. Do not read data from ADCR when the peripheral hardware clock (f_{PRS}) is stopped.

3. ADCR is read-only.

4.2 Software Coding Example

The settings to be specified for the A/D converter in the 78K0/KC2-L source program are shown below as a software coding example.

(1) Assembly language

```
XMAIN   CSEG    UNIT
IRESET:
• • • (Omitted) • • •
        MOV     ADPC0,  #11111100B      ; Specify P20 and P21 as analog input pins
• • • (Omitted) • • •
        MOV     PM2,    #00000011B      ; Specify P20 and P21 as input ports
• • • (Omitted) • • •
        MOV     ADM0,   #00000000B      ; A/D converter mode register 0
• • • (Omitted) • • •
        SET1    ADCE                    ; Start the A/D voltage comparator
                                        ; * The operation of the A/D voltage comparator is
                                        ; controlled by ADCS and ADCE, and the time from
                                        ; starting the operation until it stabilizes takes
                                        ; 1 us.  The conversion data becomes valid starting
                                        ; from the first conversion data, by setting ADCS to 1
                                        ; after at least 1 us elapses since ADCE was set to 1.
• • • (Omitted) • • •
        MOV     ADS,    #00000000B      ; Specify ANI0 as the analog input channel
• • • (Omitted) • • •
        CALL    !SADCRUN                ; A/D conversion
• • • (Omitted) • • •
        MOV     ADS,    #00000001B      ; Specify ANI1 as the analog input channel
• • • (Omitted) • • •
        CALL    !SADCRUN                ; A/D conversion
• • • (Omitted) • • •
        CLR1    ADCE                    ; Stop the A/D voltage comparator
```

```
SADCRUN:
        SET1    ADCS                    ; Start A/D conversion
JADC100:
        ; Make the system wait until A/D conversion ends
        CLR1    ADIF                    ; Clear the INTAD interrupt request
JADC110:
        NOP
        BF      ADIF,   $ADC110         ; Has A/D conversion been completed?, No
        ; Save the A/D conversion results
        MOVW    AX,     ADCR            ; Read the A/D conversion results
• • • (Omitted) • • •
        CLR1    ADCS                    ; Stop A/D conversion
        RET
```

(2) C language

```
void hdwinit(void){
• • • (Omitted) • • •
ADPC0   = 0b11111100;   /* Specify P20 and P21 as analog input pins */
• • • (Omitted) • • •
PM2     = 0b00000011;   /* Specify P20 and P21 as input ports */
• • • (Omitted) • • •
ADM0    = 0b00000000;   /* A/D converter mode register 0 */
• • • (Omitted) • • •

void main(void)
{
• • • (Omitted) • • •
        ADCE    = 1;            /* Start the A/D voltage comparator */
                                /** The operation of the A/D voltage comparator is        */
                                /*  controlled by ADCS and ADCE, and the time from       */
                                /*  starting the operation until it stabilizes takes     */
                                /*  1 us.  The conversion data becomes valid starting    */
                                /*  from the first conversion data, by setting ADCS to 1 */
                                /*  after at least 1 us elapses since ADCE was set to 1. */
• • • (Omitted) • • •
        ADS     = 0b00000000;   /* Specify ANI0 as the analog input channel */
        fn_AdcRun(4, ushAdcChannel0Buffer);  /* A/D conversion */
• • • (Omitted) • • •
        ADS     = 0b00000001;   /* ; Specify ANI1 as the analog input channel */
        fn_AdcRun(4, ushAdcChannel1Buffer);  /* A/D conversion */
• • • (Omitted) • • •
        ADCE    = 0;    /* Stop the A/D voltage comparator */
```

```
static void fn_AdcRun(unsigned char ucAdcCounter, unsigned short *pAdcData)
{
• • • (Omitted) • • •
        ADCS    = 1;    /* Start A/D conversion */

        /* Perform A/D conversion the specified number of times and then save the conversion results */
        for (ucCounter = 0; ucCounter < ucAdcCounter; ucCounter++){
            ADIF = 0;       /* Clear the INTAD interrupt request */
            while (!ADIF){  /* Make the system wait until A/D conversion ends */
                    NOP();
            }
            *pAdcData = ADCR;       /* Read the A/D conversion results */
            pAdcData++;             /* Go to the next save area */
        }

        ADCS    = 0;    /* Stop A/D conversion */
}
```

Specify the P20/ANI0 and P21/ANI1 pins as analog input pins.

Specify P20 and P21 as input ports.

Specify the standard mode as the operating mode and 264/f_{PRS} as the conversion time.

Start the A/D voltage comparator.

Specify ANI0 as the analog input channel.

Specify ANI1 as the analog input channel.

Stop the A/D voltage comparator.

Start A/D conversion.

Read the A/D conversion results after A/D conversion ends.

Stop A/D conversion.

4.3 Input Voltage and A/D Conversion Result

The analog input voltage input from the analog input pins (ANI0 to ANI10) and the theoretical A/D conversion result (10-bit A/D conversion result register (ADCR)[Note]) have a relation expressed by the following expression.

$$ADCR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

INT ( ): Function returning the integral part of the value within parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: 10-bit A/D conversion result register (ADCR) value

Calculation example: When the analog input voltage is 1.96 V and the AV_{REF} pin voltage is 5 V

- $ADCR = INT\left(\frac{1960}{5000} \times 1024 + 0.5\right) = INT(401.908) = 401 = 0191H$

Note There are three types of A/D conversion result registers.
- ADCR (16 bits): Stores 10-bit A/D conversion results.
- ADCRL (8 bits): Stores the lower 8 bits of 10-bit A/D conversion results.
- ADCRH (8 bits): Stores the higher 8 bits of 10-bit A/D conversion results.

Remark A/D converter analog input pins differ depending on products.
- 78K0/KY2-L: ANI0 to ANI3
- 78K0/KA2-L: ANI0 to ANI5
- 78K0/KB2-L: ANI0 to ANI3, ANI8 to ANI10
- 78K0/KC2-L: ANI0 to ANI10

CHAPTER 5 RELATED DOCUMENTS

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

Document Name		English
78K0/Kx2-L User's Manual		PDF
78K/0 Series Instructions User's Manual		PDF
RA78K0 Assembler Package User's Manual	Language	PDF
	Operation	PDF
CC78K0 C Compiler User's Manual	Language	PDF
	Operation	PDF
PM+ Project Manager User's Manual		PDF
78K0/Kx2-L Application Note	Sample Program (Initial Settings) LED Lighting Switch Control	PDF

APPENDIX A PROGRAM LIST

As a program list example, the 78K0/KC2-L microcontroller source program is shown below.

● main.asm (assembly language version)

```
;*******************************************************************************
;
;     NEC Electronics     78K0/KC2-L Series
;
;*******************************************************************************
;     78K0/KC2-L Series    Sample Program (A/D Converter)
;*******************************************************************************
;     Successive A/D Conversion & Average Value Calculation
;*******************************************************************************
;<<History>>
;     2009.1.--     Release
;*******************************************************************************
;
;<<Overview>>
;
; This sample program presents an example of using the A/D converter.  A/D conversion is
; performed four times each for the analog input from the analog input channels ANI0 and
; ANI1, and each converted result and the average value of the converted data are saved
; into the RAM area.
;
;
; <Primary initial settings>
;
; (Option byte settings)
; - Allowing the internal low-speed oscillator to be programmed to stop
; - Disabling the watchdog timer
; - Setting the internal high-speed oscillation clock frequency to 8 MHz
; - Disabling LVI from being started by default
; (Settings during initialization immediately after a reset ends)
; - Specifying the ROM and RAM sizes
; - Setting up I/O ports
;    → Specifying P20/ANI0 and P21/ANI1 as analog inputs for the A/D converter
; - Checking whether VDD is 2.7 V or more by using the low-voltage detector
; - Specifying that the CPU clock and peripheral hardware clock run on the internal
;    high-speed oscillation clock (8 MHz)
; - Stopping the internal low-speed oscillator
; - Disabling peripheral hardware not to be used
; - Setting up the A/D converter
;    → Specifying the standard mode as the operating mode
;    → Specifying 264/fPRS (about 33 us) as the A/D conversion time
;
;
```

```
; <Analog input channels used and the area in which to save the conversion results>
;
;  +-------------------------------------------------------------------------+
;  | Channel       | Data Type               | Variable Name   | Data Length   |
;  |-------------------------------------------------------------------------|
;  | ANI0          | A/D conversion result   | RADBUF0 + 0     | 16 bits       |
;  | (P20/ANI0 pin)| (1st time)              |                 |               |
;  |               | A/D conversion result   | RADBUF0 + 2     | 16 bits       |
;  |               | (2nd time)              |                 |               |
;  |               | A/D conversion result   | RADBUF0 + 4     | 16 bits       |
;  |               | (3rd time)              |                 |               |
;  |               | A/D conversion result   | RADBUF0 + 6     | 16 bits       |
;  |               | (4th time)              |                 |               |
;  |               | A/D conversion result   | RADAVR0         | 16 bits       |
;  |               | (average)               |                 |               |
;  |-------------------------------------------------------------------------|
;  | ANI1          | A/D conversion result   | RADBUF1 + 0     | 16 bits       |
;  | (P21/ANI1 pin)| (1st time)              |                 |               |
;  |               | A/D conversion result   | RADBUF1 + 2     | 16 bits       |
;  |               | (2nd time)              |                 |               |
;  |               | A/D conversion result   | RADBUF1 + 4     | 16 bits       |
;  |               | (3rd time)              |                 |               |
;  |               | A/D conversion result   | RADBUF1 + 6     | 16 bits       |
;  |               | (4th time)              |                 |               |
;  |               | A/D conversion result   | RADAVR1         | 16 bits       |
;  |               | (average)               |                 |               |
;  +-------------------------------------------------------------------------+
;
;
; <I/O port settings>
;  Input: P20, P21
;  * Set all unused ports that can be specified as output ports as output ports.
;
;*************************************************************************


;=========================================================================
;
;    Vector table
;
;=========================================================================
XVECT1              CSEG   AT     0000H
     DW     RESET_START           ;0000H RESET input, POC, LVI, WDT
XVECT2              CSEG   AT     0004H
     DW     IINIT                 ;0004H INTLVI
     DW     IINIT                 ;0006H INTP0
     DW     IINIT                 ;0008H INTP1
     DW     IINIT                 ;000AH INTP2
```

```
        DW      IINIT                   ;000CH INTP3
        DW      IINIT                   ;000EH INTP4
        DW      IINIT                   ;0010H INTP5
        DW      IINIT                   ;0012H INTSRE6
        DW      IINIT                   ;0014H INTSR6
        DW      IINIT                   ;0016H INTST6
        DW      IINIT                   ;0018H INTCSI10
        DW      IINIT                   ;001AH INTTMH1
        DW      IINIT                   ;001CH INTTMH0
        DW      IINIT                   ;001EH INTTM50
        DW      IINIT                   ;0020H INTTM000
        DW      IINIT                   ;0022H INTTM010
        DW      IINIT                   ;0024H INTAD
        DW      IINIT                   ;0026H INTP6
        DW      IINIT                   ;0028H INTRTCI
        DW      IINIT                   ;002AH INTTM51
        DW      IINIT                   ;002CH INTKR
        DW      IINIT                   ;002EH INTRTC
        DW      IINIT                   ;0030H INTP7
        DW      IINIT                   ;0032H INTP8
        DW      IINIT                   ;0034H INTIICA0
        DW      IINIT                   ;0036H INTCSI11
        DW      IINIT                   ;0038H INTP9
        DW      IINIT                   ;003AH INTP10
        DW      IINIT                   ;003CH INTP11
        DW      IINIT                   ;003EH BRK

  ;=============================================================================
  ;
  ;     Define the RAM data table
  ;
  ;=============================================================================
  DRAM  DSEG    SADDRP
  RADBUF0:      DS      8               ; Area in which to save the A/D conversion results (for
ANI0)
  RADBUF1:      DS      8               ; Area in which to save the A/D conversion results (for
ANI1)
  RADAVR0:      DS      2               ; Average A/D conversion result (for ANI0)
  RADAVR1:      DS      2               ; Average A/D conversion result (for ANI1)

  ;=============================================================================
  ;
  ;     Define the memory stack area
  ;
  ;=============================================================================
  DSTK  DSEG    IHRAM
  STACKEND:
                DS      20H             ; Memory stack area = 32 bytes
```

```
  STACKTOP:                         ; Start address of the memory stack area


  ;*************************************************************************
  ;
  ;     Servicing interrupts by using unnecessary interrupt sources
  ;
  ;*************************************************************************
  XMAIN CSEG   UNIT
  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                           ; Disable interrupts

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

  ;-------------------------------------------------------------------------
  ;     Specify the ROM and RAM sizes
  ;-------------------------------------------------------------------------
  ;     Note that the values to specify vary depending on the model.
  ;     Enable the settings for the model to use. (The uPD78F0588 is the default model.)
  ;-------------------------------------------------------------------------
       ; Setting when using uPD78F0581 or uPD78F0586
       ;MOV   IMS,   #042H           ; Specify the ROM and RAM sizes

       ; Setting when using uPD78F0582 or uPD78F0587
       ;MOV   IMS,   #004H           ; Specify the ROM and RAM sizes

       ; Setting when using uPD78F0583 or uPD78F0588
       MOV    IMS,   #0C8H          ; Specify the ROM and RAM sizes
```

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

;-------------------------------------------------------------------------------
;     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    ADPC1, #00000111B      ; Specify P10 to P12 as digital I/O ports
      MOV    P1,     #00000000B     ; Set the P10 to P17 output latches to low level
      MOV    PM1,    #00000000B     ; Specify P10 to P17 as output ports
                                    ; P10 to P17: Unused

;-------------------------------------------------------------------------------
;     Initialize port 2
;-------------------------------------------------------------------------------
      MOV    ADPC0, #11111100B      ; Specify P20 and P21 as analog input pins
                                    ; Specify P22 to P27 as digital I/O pins
      MOV    P2,     #00000000B     ; Set the P20 to P27 output latches to low level
      MOV    PM2,    #00000011B     ; Specify P20 and P21 as input ports
                                    ; Specify P22 to P27 as output ports
                                    ; P20: Use for analog input channel ANI0
                                    ; P21: Use for analog input channel ANI1
                                    ; P22 to P27: Unused

;-------------------------------------------------------------------------------
;     Initialize port 3
;-------------------------------------------------------------------------------
      MOV    P3,     #00000000B     ; Set the P30 to P33 output latches to low level
      MOV    PM3,    #11110000B     ; Specify P30 to P33 as output ports
                                    ; P30 to P33: Unused

;-------------------------------------------------------------------------------
;     Initialize port 4
;-------------------------------------------------------------------------------
      MOV    P4,     #00000000B     ; Set the P40 to P42 output latches to low level
      MOV    PM4,    #11111000B     ; Specify P40 to P42 as output ports
                                    ; P40 to P42: Unused

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

```
  ;-------------------------------------------------------------------------------
       MOV    P6,    #00000000B    ; Set the P60 to P63 output latches to low level
       MOV    PM6,   #11110000B    ; Specify P60 to P63 as output ports
                                   ; P60 to P63: Unused

  ;-------------------------------------------------------------------------------
  ;    Initialize port 7
  ;-------------------------------------------------------------------------------
       MOV    P7,    #00000000B    ; Set the P70 to P75 output latches to low level
       MOV    PM7,   #11000000B    ; Specify P70 to P75 as output ports
                                   ; P70 to P75: Unused

  ;-------------------------------------------------------------------------------
  ;    Initialize port 12
  ;-------------------------------------------------------------------------------
       MOV    P12,   #00000000B    ; Set the P120 output latch to low level
       MOV    PM12,  #11111110B    ; Specify P120 as an output port
                                   ; P120 to P125: Unused

  ;-------------------------------------------------------------------------------
  ;    Low-voltage detection
  ;-------------------------------------------------------------------------------
  ;    The low-voltage detector is used to check whether VDD is 2.7 V or more.
  ;-------------------------------------------------------------------------------
       ; Set up the low-voltage detector
       SET1   LVIMK                ; Disable the INTLVI interrupt
       CLR1   LVISEL               ; Specify VDD as the detection voltage
       MOV    LVIS,  #00001001B    ; Set the low-voltage detection level (VLVI) to 2.84 ±0.1
V
       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,     #5            ; Specify the number of counts
  HINI100:
       NOP
       DBNZ   B,     $HINI100      ; Has the wait period ended? No,

       ; Make the system wait until VLVI is less than or equal to VDD
  HINI110:
       NOP
       BT     LVIF,  $HINI110      ; VDD < VLVI? Yes,
       CLR1   LVION                ; 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.
  ;-------------------------------------------------------------------------------
       MOV    OSCCTL,#00000000B    ; Clock operation mode
  ;                   ||||+||+------- Be sure to clear this bit to 0
  ;                   ||||  ++-------- RSWOSC/AMPHXT
  ;                   ||||            [XT1 oscillator oscillation mode selection]
  ;                   ||||             00: Low power consumption oscillation
  ;                   ||||             01: Normal oscillation
  ;                   ||||             1x: Ultra-low power consumption oscillation
  ;                   ||++------------ EXCLKS/OSCSELS
  ;                   ||               [Subsystem clock pin operation setting]
  ;                   ||               (P123/XT1,P124/XT2/EXCLKS)
  ;                   ||                Specify the use of the pin as an I/O port pin by specifying
000 by also using XTSTART
  ;                    ++-------------- EXCLK/OSCSEL
  ;                                     [High-speed system clock pin operation setting]
  ;                                     (P121/X1,P122/X2/EXCLK)
  ;                                      00: Input port
  ;                                      01: X1 oscillation mode
  ;                                      10: Input port
  ;                                      11: External clock input mode

       MOV    PCC,   #00000000B    ; Select the CPU clock (fCPU)
  ;                   |||+|+++------- CSS/PCC2/PCC1/PCC0
  ;                   |||  |          [CPU clock (fCPU) selection]
  ;                   |||  |           0000:fXP
  ;                   |||  |           0001:fXP/2
  ;                   |||  |           0010:fXP/2^2
  ;                   |||  |           0011:fXP/2^3
  ;                   |||  |           0100:fXP/2^4
  ;                   |||  |           1000:fSUB/2
  ;                   |||  |           1001:fSUB/2
  ;                   |||  |           1010:fSUB/2
  ;                   |||  |           1011:fSUB/2
  ;                   |||  |           1100:fSUB/2
  ;                   |||  |            (Other than the above: Setting prohibited)
  ;                   |||  +----------- Be sure to clear this bit to 0
  ;                   ||+-------------- CLS
  ;                   ||               [CPU clock status]
  ;                   |+--------------- XTSTART
  ;                   |                [Subsystem clock pin operation setting]
  ;                   |                 Specify the use of the pin by also using EXCLKS and OSCSELS
  ;                    +---------------- Be sure to clear this bit to 0

       MOV    RCM,   #00000010B    ; Select the operating mode of the internal oscillator
  ;                   |||||||+------- RSTOP
  ;                   |||||||         [Internal high-speed oscillator oscillating/stopped]
```

```
;                    |||||||         0: Internal high-speed oscillator oscillating
;                    |||||||         1: Internal high-speed oscillator stopped
;                    ||||||+-------- LSRSTOP
;                    ||||||         [Internal low-speed oscillator oscillating/stopped]
;                    ||||||          0: Internal low-speed oscillator oscillating
;                    ||||||          1: Internal low-speed oscillator stopped
;                    |+++++--------- Be sure to clear this bit to 0
;                    +-------------- RSTS
;                                   [Status of internal high-speed oscillator]

      MOV    MOC,   #10000000B    ; Select the operating mode of the high-speed system clock
;                    |+++++++------- Be sure to clear this bit to 0
;                    +-------------- MSTOP
;                                   [Control of high-speed system clock operation]
;                                    0: X1 oscillator operating/external clock from
;                                        EXCLK pin is enabled
;                                    1: X1 oscillator stopped/external clock from
;                                        EXCLK pin is disabled

      MOV    MCM,   #00000000B    ; Select the clock to supply
;                    |||||+|+------- XSEL/MCM0:
;                    ||||| |        [Clock supplied to main system and
;                    ||||| |         peripheral hardware]
;                    ||||| |         00: Main system clock (fXP)
;                    ||||| |              = internal high-speed oscillation clock (fIH)
;                    ||||| |             Peripheral hardware clock (fPRS)
;                    ||||| |              = internal high-speed oscillation clock (fIH)
;                    ||||| |         01: Main system clock (fXP)
;                    ||||| |              = internal high-speed oscillation clock (fIH)
;                    ||||| |             Peripheral hardware clock (fPRS)
;                    ||||| |              = internal high-speed oscillation clock (fIH)
;                    ||||| |         10: Main system clock (fXP)
;                    ||||| |              = internal high-speed oscillation clock (fIH)
;                    ||||| |             Peripheral hardware clock (fPRS)
;                    ||||| |              = high-speed system clock (fIH)
;                    ||||| |         11: Main system clock (fXP)
;                    ||||| |              = high-speed system clock (fIH)
;                    ||||| |             Peripheral hardware clock (fPRS)
;                    ||||| |              = high-speed system clock (fIH)
;                    ||||| +-------- MCS
;                    |||||          [Main system clock status]
;                    +++++----------- Be sure to clear this bit to 0

      MOV    PER0,  #00000000B    ; Control the real-time counter control clock
;                    |+++++++------- Be sure to clear this bit to 0
;                    +-------------- RTCEN:
;                                   [Real-time counter control clock]
;                                    0: Stop supply of control clock
```

```
;                                   1: Supply control clock

;--------------------------------------------------------------------------------
;     Disable peripheral hardware not to be used
;--------------------------------------------------------------------------------
      ; 16-bit timer/event counter 00
      MOV    TMC00, #00000000B    ; Disable the counter

      ; 8-bit timer/event counters 50 and 51
      MOV    TMC50, #00000000B    ; Disable timer 50
      MOV    TMC51, #00000000B    ; Disable timer 51

      ; 8-bit timers H0 and H1
      MOV    TMHMD0,#00000000B    ; Stop timer H0
      MOV    TMHMD1,#00000000B    ; Stop timer H1

      ; Real-time counter
      MOV    RTCC0, #00000000B    ; Stop the counter

      ; Clock output controller
      MOV    CKS,   #00000000B    ; Stop the clock frequency divider

      ; Operational amplifiers
      MOV    AMP0M, #00000000B    ; Stop operational amplifier 0
      MOV    AMP1M, #00000000B    ; Stop operational amplifier 1

      ; Serial interface UART6
      MOV    ASIM6, #00000001B    ; Disable the interface

      ; Serial interface IICA
      MOV    IICACTL0,#00000000B  ; Disable the interface

      ; Serial interface CSI10, CSI11
      MOV    CSIM10,#00000000B    ; Disable CSI10
      MOV    CSIM11,#00000000B    ; Disable CSI11

      ; Interrupts
      MOVW   MK0,   #0FFFFH; Disable all interrupts
      MOVW   MK1,   #0FFFFH;
      MOV    EGPCTL0,#00000000B   ; Disable the detection of all external interrupts
      MOV    EGPCTL1,#00000000B   ;

      ; Key interrupts
      MOV    KRM,   #00000000B    ; Disable all key interrupts

;--------------------------------------------------------------------------------
;     Set up the A/D converter
;--------------------------------------------------------------------------------
```

```
  ;     Specify the standard mode as the operating mode and 264/fPRS (about 33 us) as the
conversion time.
  ;-------------------------------------------------------------------------------
       ; Set up the A/D converter
       MOV    ADM0,  #00000000B    ; A/D converter mode register 0
  ;                   |||||||+------- ADCE
  ;                   |||||||        [A/D voltage comparator operation control]
  ;                   |||||||         0: Stop A/D voltage comparator operation
  ;                   |||||||         1: Enable A/D voltage comparator operation
  ;                   |||||++-------- LV1/LV0
  ;                   |||||          [Operating mode selection]
  ;                   |||||          [4.0 V ≤ AVREF ≤ 5.5 V]
  ;                   |||||            00: Standard mode
  ;                   |||||            10: Maximum-speed mode
  ;                   |||||            11: High-speed mode
  ;                   |||||          [2.7 V ≤ AVREF < 4.0 V]
  ;                   |||||            00: Standard mode
  ;                   |||||            11: High-speed mode
  ;                   |||||          [1.8 V ≤ AVREF < 2.7 V]
  ;                   |||||            01: Low-voltage mode
  ;                   ||+++---------- FR2/FR1/FR0
  ;                   ||             [A/D conversion time selection]
  ;                   ||             [Standard mode]
  ;                   ||                   Conversion time     Conversion clock (fAD)
  ;                   ||               000: 264/fPRS           fPRS/12
  ;                   ||               001: 176/fPRS           fPRS/8
  ;                   ||               010: 132/fPRS           fPRS/6
  ;                   ||               011: 88/fPRS            fPRS/4
  ;                   ||               100: 66/fPRS            fPRS/3
  ;                   ||               101: 44/fPRS            fPRS/2
  ;                   ||               110: 33/fPRS            fPRS/1.5
  ;                   ||               111: 22/fPRS            fPRS
  ;                   ||             [High-speed mode]
  ;                   ||                   Conversion time     Conversion clock (fAD)
  ;                   ||               001: 176/fPRS           fPRS/8
  ;                   ||               010: 132/fPRS           fPRS/6
  ;                   ||               011: 88/fPRS            fPRS/4
  ;                   ||               100: 66/fPRS            fPRS/3
  ;                   ||               101: 44/fPRS            fPRS/2
  ;                   ||               110: 33/fPRS            fPRS/1.5
  ;                   ||               111: 22/fPRS            fPRS
  ;                   ||             [Maximum-speed mode]
  ;                   ||                   Conversion time     Conversion clock (fAD)
  ;                   ||               100: 66/fPRS            fPRS/3
  ;                   ||               110: 33/fPRS            fPRS/1.5
  ;                   ||             [Low-voltage mode]
  ;                   ||                   Conversion time     Conversion clock (fAD)
  ;                   ||               000: 528/fPRS           fPRS/12
```

```
;                   ||              001: 352/fPRS           fPRS/8
;                   ||              010: 264/fPRS           fPRS/6
;                   ||              011: 176/fPRS           fPRS/4
;                   ||              100: 132/fPRS           fPRS/3
;                   ||              101: 88/fPRS            fPRS/2
;                   ||              110: 66/fPRS            fPRS/1.5
;                   ||              111: 44/fPRS            fPRS
;                   |+------------- Be sure to clear this bit to 0
;                   +--------------- ADCS
;                                   [A/D conversion operation control]
;                                    0: Stop conversion operation
;                                    1: Enable conversion operation

      CLR1   ADIF                 ; Clear the INTAD interrupt request
      SET1   ADMK                 ; Disable the INTAD interrupt

  ;-------------------------------------------------------------------------
  ;     Enable interrupts
  ;     (To use interrupts, enable interrupts here.)
  ;-------------------------------------------------------------------------
  ;     EI                        ; To enable interrupts,
                                  ; uncomment this line.


      BR     MMAIN_LOOP           ; Go to the main loop


  ;*************************************************************************
  ;
  ;     Main loop
  ;
  ;*************************************************************************
  MMAIN_LOOP:

  ;-------------------------------------------------------------------------
  ;     Start the A/D voltage comparator
  ;-------------------------------------------------------------------------
      SET1   ADCE                 ; Start the A/D voltage comparator
                                  ; * The operation of the A/D voltage comparator is
                                  ;   controlled by ADCS and ADCE, and the time from
                                  ;   starting the operation until it stabilizes takes
                                  ;   1 us.  The conversion data becomes valid starting
                                  ;   from the first conversion data, by setting ADCS to 1
after at least 1 us elapses since ADCE was set to 1.

  ;-------------------------------------------------------------------------
  ;     A/D conversion of ANI0
  ;-------------------------------------------------------------------------
```

```
        MOV     ADS,    #00000000B      ; Specify ANI0 as the analog input channel

        MOVW    HL,     #RADBUF0        ; Specify the address of the area in which to save the
A/D conversion results
        MOV     B,      #4              ; Specify the number of A/D conversions
        CALL    !SADCRUN                ; A/D conversion

  ;-------------------------------------------------------------------------------
  ;     A/D conversion of ANI1
  ;-------------------------------------------------------------------------------
        MOV     ADS,    #00000001B      ; Specify ANI1 as the analog input channel

        MOVW    HL,     #RADBUF1        ; Specify the address of the area in which to save the
A/D conversion results
        MOV     B,      #4              ; Specify the number of A/D conversions
        CALL    !SADCRUN                ; A/D conversion

  ;-------------------------------------------------------------------------------
  ;     Stop the A/D voltage comparator
  ;-------------------------------------------------------------------------------
        CLR1    ADCE                    ; Stop the A/D voltage comparator

  ;-------------------------------------------------------------------------------
  ;     Calculate the average A/D conversion result of ANI0
  ;-------------------------------------------------------------------------------
        MOVW    HL,     #RADBUF0        ; Specify the address of the area in which to save the
A/D conversion results
        MOVW    DE,     #RADAVR0        ; Specify the address of the area in which to save the
average value
        MOV     B,      #4              ; Specify the number of A/D conversion results from
which to calculate the average
        CALL    !SADCAVR                ; Average-value calculation

  ;-------------------------------------------------------------------------------
  ;     Calculate the average A/D conversion result of ANI1
  ;-------------------------------------------------------------------------------
        MOVW    HL,     #RADBUF1        ; Specify the address of the area in which to save the
A/D conversion results
        MOVW    DE,     #RADAVR1        ; Specify the address of the area in which to save the
average value
        MOV     B,      #4              ; Specify the number of A/D conversion results from
which to calculate the average
        CALL    !SADCAVR                ; Average-value calculation

        BR      MMAIN_LOOP              ; Go to the start of the main loop

  ;*******************************************************************************
  ;
```

```
;     A/D conversion
;
;-------------------------------------------------------------------------------
;     [I N]  B      : Number of times to perform A/D conversion
;            HL     : Area in which to save the A/D conversion results
;     [OUT]  -
;*******************************************************************************
SADCRUN:
      SET1   ADCS                ; Start A/D conversion

JADC100:
      ; Make the system wait until A/D conversion ends
      CLR1   ADIF                ; Clear the INTAD interrupt request
JADC110:
      NOP
      BF     ADIF,  $JADC110     ; Has A/D conversion been completed?, No

      ; Save the A/D conversion results
      MOVW   AX,    ADCR         ; Read the A/D conversion results
      XCH    A,     X            ; Exchange the higher and lower bytes
      MOV    [HL],  A            ; Save the lower byte of the A/D conversion results
      XCH    A,     X            ; Exchange the higher and lower bytes
      INCW   HL                  ; Go to the higher save area
      MOV    [HL],  A            ; Save the higher byte of the A/D conversion results
      INCW   HL                  ; Go to the next save area
      DBNZ   B,     $JADC100     ;  Have the specified number of A/D conversions been
completed? No,

      CLR1   ADCS                ; Stop A/D conversion

      RET

;*******************************************************************************
;
;     Average-value calculation
;
;-------------------------------------------------------------------------------
;     [I N]  B      : Number of data units used to calculate the average
;            HL     : Area in which the data used to calculate the average is saved
;            DE     : Area in which the average value is saved
;     [OUT]  -
;*******************************************************************************
SADCAVR:
      MOV    A,     B            ; Specify the number of data units as the divisor to
calculate the average
      MOV    C,     A            ;
      MOVW   AX,    #0000H       ; Clear the AX register
```

```
      ; Calculate the average value
  JAVR100:
      XCH    A,     X             ; Exchange the higher and lower bytes
      ADD    A,     [HL]          ; Add the lower byte
      XCH    A,     X             ; Exchange the higher and lower bytes
      INCW   HL                   ; Go to the higher save area
      ADDC   A,     [HL]          ; Add the higher byte (including the carry of the lower
byte)
      INCW   HL                   ; Go to the next data
      DBNZ   B,     $JAVR100      ; Has the total value been calculated? No,
      DIVUW  C                    ; Calculate the average value (AX ← (AX/C))

      ; Save the average value
      XCH    A,     X             ; Exchange the higher and lower bytes
      MOV    [DE],  A             ; Save the lower byte of the average value
      INCW   DE                   ; Go to the higher save area
      XCH    A,     X             ; Exchange the higher and lower bytes
      MOV    [DE],  A             ; Save the higher byte of the average value

      RET
end
```

● main.c (C language version)

```
/************************************************************************************

  NEC Electronics     78K0/KC2-L Series

************************************************************************************
  78K0/KC2-L Series Sample Program (A/D Converter)
************************************************************************************
  Successive A/D Conversion & Average Value Calculation
************************************************************************************
<<History>>
  2009.1.--  Release
************************************************************************************

<<Overview>>

This sample program presents an example of using the A/D converter.  A/D conversion is
performed four times each for the analog input from the analog input channels ANI0 and
ANI1, and each converted result and the average value of the converted data are saved
into the RAM area.


 <Primary initial settings>

 (Option byte settings)
 - Allowing the internal low-speed oscillator to be programmed to stop
 - Disabling the watchdog timer
 - Setting the internal high-speed oscillation clock frequency to 8 MHz
 - Disabling LVI from being started by default
 (Settings during initialization immediately after a reset ends)
 - Specifying the ROM and RAM sizes
 - Setting up I/O ports
    → Specifying P20/ANI0 and P21/ANI1 as analog inputs for the A/D converter
 - Checking whether VDD is 2.7 V or more by using the low-voltage detector
 - Specifying that the CPU clock and peripheral hardware clock run on the internal
    high-speed oscillation clock (8 MHz)
 - Stopping the internal low-speed oscillator
 - Disabling peripheral hardware not to be used
 - Setting up the A/D converter
    → Specifying the standard mode as the operating mode
    → Specifying 264/fPRS (about 33 us) as the A/D conversion time


 <Analog input channels used and the area in which to save the conversion results>

  +-------------------------------------------------------------------------------+
  | Channel         | Data Type                 | Variable Name      | Data Length |
  |-------------------------------------------------------------------------------|
```

```
  | ANI0           | A/D conversion result     | ushAdcChannel0     | 16 bits      |
  | (P20/ANI0 pin) | (1st time)                | Buffer[0]          |              |
  |                | A/D conversion result     | ushAdcChannel0     | 16 bits      |
  |                | (2nd time)                | Buffer[1]          |              |
  |                | A/D conversion result     | ushAdcChannel0     | 16 bits      |
  |                | (3rd time)                | Buffer[2]          |              |
  |                | A/D conversion result     | ushAdcChannel0     | 16 bits      |
  |                | (4th time)                | Buffer[3]          |              |
  |                | A/D conversion result     | ushAdcChannel0     | 16 bits      |
  |                | (average)                 | Average            |              |
  |-------------------------------------------------------------------------------|
  | ANI1           | A/D conversion result     | ushAdcChannel1     | 16 bits      |
  | (P21/ANI1 pin) | (1st time)                | Buffer[0]          |              |
  |                | A/D conversion result     | ushAdcChannel1     | 16 bits      |
  |                | (2nd time)                | Buffer[1]          |              |
  |                | A/D conversion result     | ushAdcChannel1     | 16 bits      |
  |                | (3rd time)                | Buffer[2]          |              |
  |                | A/D conversion result     | ushAdcChannel1     | 16 bits      |
  |                | (4th time)                | Buffer[3]          |              |
  |                | A/D conversion result     | ushAdcChannel1     | 16 bits      |
  |                | (average)                 | Average            |              |
  +-------------------------------------------------------------------------------+


 <I/O port settings>
  Input: P20, P21
  * 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 */



/*================================================================================

  Declare function prototypes

================================================================================*/
```

```
/* A/D conversion */
static void fn_AdcRun(unsigned char ucAdcCounter, unsigned short *pAdcData);

/* Average-value calculation */
static void fn_Average
(unsigned char ucDataCounter, unsigned short *pData, unsigned short *pAverage);


/************************************************************************

  Initialization after RESET

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

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

/*-----------------------------------------------------------------------
  Specify the ROM and RAM sizes
-----------------------------------------------------------------------
  Note that the values to specify vary depending on the model.
  Enable the settings for the model to use. (The uPD78F0588 is the default model.)
-----------------------------------------------------------------------*/
  /* Setting when using uPD78F0581 or uPD78F0586 */
  /*IMS =0x42;*/       /* Specify the ROM and RAM sizes */

  /* Setting when using uPD78F0582 or uPD78F0587 */
  /*IMS =0x04;*/       /* Specify the ROM and RAM sizes */

  /* Setting when using uPD78F0583 or uPD78F0588 */
  IMS =  0xC8;         /* Specify the ROM and RAM sizes */

/*-----------------------------------------------------------------------
  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
-----------------------------------------------------------------------*/
  ADPC1  = 0b00000111; /* Specify P10 to P12 as digital I/O ports */
```

```
  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
-----------------------------------------------------------------------------*/
  ADPC0  = 0b11111100; /* Specify P20 and P21 as analog input pins */
                       /* Specify P22 to P27 as digital I/O pins */
  P2     = 0b00000000; /* Set the P20 to P27 output latches to low level */
  PM2    = 0b00000011; /* Specify P20 and P21 as input ports */
                       /* Specify P22 to P27 as output ports */
                       /* P20: Use for analog input channel ANI0 */
                       /* P21: Use for analog input channel ANI1 */
                       /* P22 to P27: Unused */

/*-----------------------------------------------------------------------------
  Initialize port 3
-----------------------------------------------------------------------------*/
  P3     = 0b00000000; /* Set the P30 to P33 output latches to low level */
  PM3    = 0b11110000; /* Specify P30 to P33 as output ports */
                       /* P30 to P33: Unused */

/*-----------------------------------------------------------------------------
  Initialize port 4
-----------------------------------------------------------------------------*/
  P4     = 0b00000000; /* Set the P40 to P42 output latches to low level */
  PM4    = 0b11111000; /* Specify P40 to P42 as output ports */
                       /* P40 to P42: Unused */

/*-----------------------------------------------------------------------------
  Initialize port 6
-----------------------------------------------------------------------------*/
  P6     = 0b00000000; /* Set the P60 to P63 output latches to low level */
  PM6    = 0b11110000; /* Specify P60 to P63 as output ports */
                       /* P60 to P63: Unused */

/*-----------------------------------------------------------------------------
  Initialize port 7
-----------------------------------------------------------------------------*/
  P7     = 0b00000000; /* Set the P70 to P75 output latches to low level */
  PM7    = 0b11000000; /* Specify P70 to P75 as output ports */
                       /* P70 to P75: Unused */

/*-----------------------------------------------------------------------------
  Initialize port 12
-----------------------------------------------------------------------------*/
  P12    = 0b00000000; /* Set the P120 output latch to low level */
```

```
    PM12   = 0b11111110; /* Specify P120 as an output port */
                         /* P120 to P125: Unused */

  /*-------------------------------------------------------------------------------
    Low-voltage detection
  -------------------------------------------------------------------------------
    The low-voltage detector is used to check whether VDD is 2.7 V or more.
  -------------------------------------------------------------------------------*/
    /* Set up the low-voltage detector */
    LVIMK  = 1;          /* Disable the INTLVI interrupt */
    LVISEL = 0;          /* Specify VDD as the detection voltage */
    LVIS   = 0b00001001; /* Set the low-voltage detection level (VLVI) to 2.84 ±0.1 V */
    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 < 2; 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.
  -------------------------------------------------------------------------------*/
    OSCCTL = 0b00000000; /* Clock operation mode */
    /*         ||||+||+---- Be sure to clear this bit to 0 */
    /*         ||||  ++----- RSWOSC/AMPHXT */
    /*         ||||         [XT1 oscillator oscillation mode selection] */
    /*         ||||          00: Low power consumption oscillation */
    /*         ||||          01: Normal oscillation */
    /*         ||||          1x: Ultra-low power consumption oscillation */
    /*         ||++-------- EXCLKS/OSCSELS */
    /*         ||           [Subsystem clock pin operation setting] */
    /*         ||           (P123/XT1,P124/XT2/EXCLKS) */
    /*         ||            Specify the use of the pin as an I/O port pin by specifying 000
by also using XTSTART */
    /*         ++---------- EXCLK/OSCSEL */
    /*                      [High-speed system clock pin operation setting] */
    /*                       (P121/X1,P122/X2/EXCLK) */
```

```
    /*                      00: Input port */
    /*                      01: X1 oscillation mode */
    /*                      10: Input port */
    /*                      11: External clock input mode */

    PCC    = 0b00000000; /* Select the CPU clock (fCPU) */
    /*         |||+|+++---- CSS/PCC2/PCC1/PCC0 */
    /*         |||  |       [CPU clock (fCPU) selection] */
    /*         |||  |        0000:fXP */
    /*         |||  |        0001:fXP/2 */
    /*         |||  |        0010:fXP/2^2 */
    /*         |||  |        0011:fXP/2^3 */
    /*         |||  |        0100:fXP/2^4 */
    /*         |||  |        1000:fSUB/2 */
    /*         |||  |        1001:fSUB/2 */
    /*         |||  |        1010:fSUB/2 */
    /*         |||  |        1011:fSUB/2 */
    /*         |||  |        1100:fSUB/2 */
    /*         |||  |        (Other than the above: Setting prohibited) */
    /*         ||| +------- Be sure to clear this bit to 0 */
    /*         ||+---------- CLS */
    /*         ||           [CPU clock status] */
    /*         |+----------- XTSTART */
    /*         |            [Subsystem clock pin operation setting] */
    /*         |             Specify the use of the pin by also using EXCLKS and OSCSELS */
    /*        +------------ Be sure to clear this bit to 0 */

    RCM    = 0b00000010; /* Select the operating mode of the internal oscillator */
    /*         |||||||+---- RSTOP */
    /*         |||||||      [Internal high-speed oscillator oscillating/stopped] */
    /*         |||||||       0: Internal high-speed oscillator oscillating */
    /*         |||||||       1: Internal high-speed oscillator stopped */
    /*         ||||||+----- LSRSTOP */
    /*         ||||||       [Internal low-speed oscillator oscillating/stopped] */
    /*         ||||||        0: Internal low-speed oscillator oscillating */
    /*         ||||||        1: Internal low-speed oscillator stopped */
    /*         |+++++------ Be sure to clear this bit to 0 */
    /*         +------------ RSTS */
    /*                      [Status of internal high-speed oscillator] */

    MOC    = 0b10000000; /* Select the operating mode of the high-speed system clock */
    /*         |+++++++---- Be sure to clear this bit to 0 */
    /*         +------------ MSTOP */
    /*                      [Control of high-speed system clock operation] */
    /*                       0: X1 oscillator operating/external clock from EXCLK pin is enabled
*/
    /*                       1: X1 oscillator stopped/external clock from EXCLK pin is disabled
*/
```

```
  MCM    = 0b00000000; /* Select the clock to supply */
  /*         |||||+|+---- XSEL/MCM0 */
  /*         ||||| |     [Clock supplied to main system and peripheral hardware] */
  /*         ||||| |      00: Main system clock (fXP) */
  /*         ||||| |           = internal high-speed oscillation clock (fIH) */
  /*         ||||| |          Peripheral hardware clock (fPRS) */
  /*         ||||| |           = internal high-speed oscillation clock (fIH) */
  /*         ||||| |      01: Main system clock (fXP) */
  /*         ||||| |           = internal high-speed oscillation clock (fIH) */
  /*         ||||| |          Peripheral hardware clock (fPRS) */
  /*         ||||| |           = internal high-speed oscillation clock (fIH) */
  /*         ||||| |      10: Main system clock (fXP) */
  /*         ||||| |           = internal high-speed oscillation clock (fIH) */
  /*         ||||| |          Peripheral hardware clock (fPRS) */
  /*         ||||| |           = high-speed system clock (fIH) */
  /*         ||||| |      11: Main system clock (fXP) */
  /*         ||||| |           = high-speed system clock (fIH) */
  /*         ||||| |          Peripheral hardware clock (fPRS) */
  /*         ||||| |           = high-speed system clock (fIH) */
  /*         ||||| +----- MCS */
  /*         |||||       [Main system clock status] */
  /*         +++++------- Be sure to clear this bit to 0 */

  PER0   = 0b00000000; /* Control the real-time counter control clock */
  /*         |++++++++---- Be sure to clear this bit to 0 */
  /*         +----------- RTCEN: */
  /*                     [Real-time counter control clock] */
  /*                      0: Stop supply of control clock */
  /*                      1: Supply control clock */

/*-----------------------------------------------------------------------------
  Disable peripheral hardware not to be used
-----------------------------------------------------------------------------*/
  /* 16-bit timer/event counter 00 */
  TMC00  = 0b00000000; /* Disable the counter */

  /* 8-bit timer/event counters 50 and 51 */
  TMC50  = 0b00000000; /* Disable timer 50 */
  TMC51  = 0b00000000; /* Disable timer 51 */

  /* 8-bit timers H0 and H1 */
  TMHMD0 = 0b00000000; /* Stop timer H0 */
  TMHMD1 = 0b00000000; /* Stop timer H1 */

  /* Real-time counter */
  RTCC0  = 0b00000000; /* Stop the counter */
```

```
    /* Clock output controller */
    CKS    = 0b00000000; /* Stop the clock frequency divider */

    /* Operational amplifiers */
    AMP0M  = 0b00000000; /* Stop operational amplifier 0 */
    AMP1M  = 0b00000000; /* Stop operational amplifier 1 */

    /* Serial interface UART6 */
    ASIM6  = 0b00000001; /* Disable the interface */

    /* Serial interface IICA */
    IICACTL0 = 0b00000000;   /* Disable the interface */

    /* Serial interfaces CSI10 and CSI11 */
    CSIM10 = 0b00000000; /* Disable CSI10 */
    CSIM11 = 0b00000000; /* Disable CSI11 */

    /* Interrupts */
    MK0    = 0xFFFF;     /* Disable all interrupts */
    MK1    = 0xFFFF;
    EGPCTL0= 0b00000000; /* Disable the detection of all external interrupts */
    EGPCTL1= 0b00000000;

    /* Key interrupts */
    KRM    = 0b00000000; /* Disable all key interrupts */

  /*-------------------------------------------------------------------------------
    Set up the A/D converter
  -------------------------------------------------------------------------------
    Specify the standard mode as the operating mode and 264/fPRS (about 33 us) as the conversion
time.
  -------------------------------------------------------------------------------*/
    /* Set up the A/D converter */
    ADM0   = 0b00000000; /* A/D converter mode register 0 */
    /*         |||||||+-- ADCE  */
    /*         |||||||   [A/D voltage comparator operation control] */
    /*         |||||||    0: Stop A/D voltage comparator operation */
    /*         |||||||    1: Enable A/D voltage comparator operation */
    /*         |||||++--- LV1/LV0 */
    /*         |||||     [Operating mode selection] */
    /*         |||||      [4.0 V ≤ AVREF ≤ 5.5 V] */
    /*         |||||        00: Standard mode */
    /*         |||||        10: Maximum-speed mode   */
    /*         |||||        11: High-speed mode   */
    /*         |||||      [2.7 V ≤ AVREF < 4.0 V] */
    /*         |||||        00: Standard mode */
    /*         |||||        11: High-speed mode   */
    /*         |||||      [1.8 V ≤ AVREF < 2.7 V] */
```

```
  /*         |||||        01: Low-voltage mode  */
  /*         ||+++------ FR2/FR1/FR0 */
  /*         ||         [A/D conversion time selection] */
  /*         ||          [Standard mode]  */
  /*         ||               Conversion time   Conversion clock (fAD) */
  /*         ||            000: 264/fPRS        fPRS/12*/
  /*         ||            001: 176/fPRS        fPRS/8     */
  /*         ||            010: 132/fPRS        fPRS/6     */
  /*         ||            011: 88/fPRS         fPRS/4     */
  /*         ||            100: 66/fPRS         fPRS/3     */
  /*         ||            101: 44/fPRS         fPRS/2     */
  /*         ||            110: 33/fPRS         fPRS/1.5   */
  /*         ||            111: 22/fPRS         fPRS       */
  /*         ||          [High-speed mode]   */
  /*         ||               Conversion time   Conversion clock (fAD) */
  /*         ||            001: 176/fPRS        fPRS/8     */
  /*         ||            010: 132/fPRS        fPRS/6     */
  /*         ||            011: 88/fPRS         fPRS/4     */
  /*         ||            100: 66/fPRS         fPRS/3     */
  /*         ||            101: 44/fPRS         fPRS/2     */
  /*         ||            110: 33/fPRS         fPRS/1.5   */
  /*         ||            111: 22/fPRS         fPRS       */
  /*         ||          [Maximum-speed mode] */
  /*         ||               Conversion time   Conversion clock (fAD) */
  /*         ||            100: 66/fPRS         fPRS/3     */
  /*         ||            110: 33/fPRS         fPRS/1.5   */
  /*         ||          [Low-voltage mode] */
  /*         ||               Conversion time   Conversion clock (fAD) */
  /*         ||            000: 528/fPRS        fPRS/12*/
  /*         ||            001: 352/fPRS        fPRS/8     */
  /*         ||            010: 264/fPRS        fPRS/6     */
  /*         ||            011: 176/fPRS        fPRS/4     */
  /*         ||            100: 132/fPRS        fPRS/3     */
  /*         ||            101: 88/fPRS         fPRS/2     */
  /*         ||            110: 66/fPRS         fPRS/1.5   */
  /*         ||            111: 44/fPRS         fPRS       */
  /*         |+-------- Be sure to clear this bit to 0 */
  /*         +--------- ADCS  */
  /*                   [A/D conversion operation control] */
  /*                    0: Stop conversion operation */
  /*                    1: Enable conversion operation */

  ADIF   = 0;          /* Clear the INTAD interrupt request */
  ADMK   = 1;          /* Disable the INTAD interrupt */

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

```
  ------------------------------------------------------------------------------*/
  /*    EI();  */            /* To enable interrupts, */
                          /* uncomment this line.  */

  }


  /************************************************************************

    Main loop

  ************************************************************************/
  void main(void)
  {
    unsigned short ushAdcChannel0Buffer[4];   /* Area in which to save the A/D conversion
results (for ANI0) */
    unsigned short ushAdcChannel1Buffer[4];   /* Area in which to save the A/D conversion
results (for ANI1) */
    unsigned short ushAdcChannel0Average;  /* Average A/D conversion result (for ANI0) */
    unsigned short ushAdcChannel1Average;  /* Average A/D conversion result (for ANI1) */

    while (1){
  /*-------------------------------------------------------------------------------
    Start the A/D voltage comparator
  -------------------------------------------------------------------------------*/
       ADCE   = 1;          /* Start the A/D voltage comparator */
                            /* * The operation of the A/D voltage comparator is     */
                            /*   controlled by ADCS and ADCE, and the time from     */
                            /*   starting the operation until it stabilizes takes   */
                            /*   1 us.  The conversion data becomes valid starting  */
                            /*    from the first conversion data, by setting ADCS to 1 after
at least 1 us elapses since ADCE was set to 1.   */

  /*-------------------------------------------------------------------------------
    A/D conversion of ANI0
  -------------------------------------------------------------------------------*/
       ADS    = 0b00000000; /* Specify ANI0 as the analog input channel */
       fn_AdcRun(4, ushAdcChannel0Buffer);/* A/D conversion */

  /*-------------------------------------------------------------------------------
    A/D conversion of ANI1
  -------------------------------------------------------------------------------*/
       ADS    = 0b00000001; /* Specify ANI1 as the analog input channel */
       fn_AdcRun(4, ushAdcChannel1Buffer);/* A/D conversion */

  /*-------------------------------------------------------------------------------
    Stop the A/D voltage comparator
  -------------------------------------------------------------------------------*/
```

```
      ADCE   = 0;   /* Stop the A/D voltage comparator */

  /*-------------------------------------------------------------------------------
    Calculate the average A/D conversion result of ANI0
  -------------------------------------------------------------------------------*/
      /* Average-value calculation */
      fn_Average(4, ushAdcChannel0Buffer, &ushAdcChannel0Average);

  /*-------------------------------------------------------------------------------
    Calculate the average A/D conversion result of ANI1
  -------------------------------------------------------------------------------*/
      /* Average-value calculation */
      fn_Average(4, ushAdcChannel1Buffer, &ushAdcChannel1Average);
    }
  }

  /*******************************************************************************

    A/D conversion

  -------------------------------------------------------------------------------
    [I N]  ucAdcCounter  : Number of times to perform A/D conversion
           *pAdcData     : Area in which to save the A/D conversion results
    [OUT]  -
  *******************************************************************************/
  static void fn_AdcRun(unsigned char ucAdcCounter, unsigned short *pAdcData)
  {
    unsigned char ucCounter; /* Count variable */

    ADCS   = 1;   /* Start A/D conversion */

    /* Perform A/D conversion the specified number of times and then save the conversion
results */
    for (ucCounter = 0; ucCounter < ucAdcCounter; ucCounter++){
       ADIF = 0;     /* Clear the INTAD interrupt request */
       while (!ADIF){/* Make the system wait until A/D conversion ends */
          NOP();
       }
       *pAdcData = ADCR; /* Read the A/D conversion results */
       pAdcData++;          /* Go to the next save area */
    }

    ADCS   = 0;   /* Stop A/D conversion */
  }

  /*******************************************************************************

    Average-value calculation
```

```
  ------------------------------------------------------------------------------
    [I N]  ucDataCounter : Number of data units used to calculate the average
           *pData        : Area in which the data used to calculate the average is saved
           *pAverage     : Area in which the average value is saved
    [OUT]  -
  ******************************************************************************/
  static void fn_Average
  (unsigned char ucDataCounter, unsigned short *pData, unsigned short *pAverage)
  {
    unsigned char ucCounter; /* Count variable */
    unsigned short ushWork = 0; /* Work variable */

    /* Add up the data used to calculate the average value */
    for (ucCounter = 0; ucCounter < ucDataCounter; ucCounter++){
       ushWork += *pData;
    }

    *pAverage = (ushWork / ucDataCounter);   /* Calculate and then save the average value
*/
  }
```

APPENDIX B USING 78K0/KC2-L 44-PIN PRODUCTS

All 78K0/KC2-L sample programs are intended for 48-pin products. To use a 78K0/KC2-L sample program for a 44-pin product, specify the following settings:

(1) Initial settings of ports

- Setting up port 0
  Change the value of bit 2 of port mode register 0 (PM0) from “0” to “1”.

- Setting up port 4
  Change the value of bit 2 of port mode register 4 (PM4) from “0” to “1”.

- Setting up port 7
  Change the values of bits 5 and 4 of port mode register 7 (PM7) from “00” to “11”.

(2) Disabling unused peripheral hardware

Delete the instruction used to set up the clock output selection register (CKS).

APPENDIX C 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 Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

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

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

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

[Asia & Oceania]

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

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

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

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

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

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

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

G0706