To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

Application Note

78K0R/LH3

Sample Program (Sound Output)

Playing Back Sound Data by Using D/A Converter and Operational Amplifier

This application note describes how to output ADPCM-format sound data by using a 12-bit D/A converter. The ADPCM data is transferred at 32 kbps (when sampled at 8 kHz and quantized with 4 bits).

Target devices

μPD78F1506, 78F1507, 78F1508

CONTENTS

Document No. U20028EJ1V0AN00 (1st edition)
Date Published September 2009 N

- The information in this document is current as of July, 2009. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

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

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

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

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

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

(M8E0909)

CHAPTER 1 OVERVIEW

This sample program outputs and plays back ADPCM-format sound data by using a 12-bit D/A converter.

The ADPCM data to be played back is transferred at 32 kbps (when sampled at 8 kHz and quantized with 4 bits). The ADPCM data is decompressed by an ADPCM-SP library function and output by the D/A converter. It is then sampled at a frequency of 8 kHz, which is controlled by a timer array unit. The data then passes through a low-pass filter that incorporates an operational amplifier. Playback of the data is triggered by a key stroke.

(1) Primary initial settings

<Option byte settings>

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

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports
  - Setting P23, P25, P26, and P150 to input data to operational amplifiers
  - Setting P24 and P27 to output data from operation amplifiers
  - Setting KR0 to input key and detect key interrupt signal
- Securing a supply voltage of 2.7 V or more by using the function of low-voltage detector[Note]
- Specifying that the CPU clock run on the X1 oscillator (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 0 of timer array unit 0 in a mode in which it operates as an interval timer of about 10 ms to set settling time of the voltage reference and avoid chattering of a key input
  - Setting channel 4 in interval timer mode to set the sampling frequency for playing back ADPCM data to 8 kHz
- Setting up voltage reference
  - Selecting the voltage reference as a reference voltage source
  - Setting the output voltage of voltage reference to 2.0 V
  - Using channel 0 of timer array unit 0 to wait for about 20 ms until the operation of the voltage reference is stabilized
- Setting up operational amplifiers 1 and 2
- Setting up D/A converter
  - Selecting the real-time output mode as the operation mode
  - Setting the resolution to 12 bits
  - Selecting V_{REFOUT}/AV_{REFP} pin as the voltage reference source of the D/A converter

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

(2) Contents following main loop

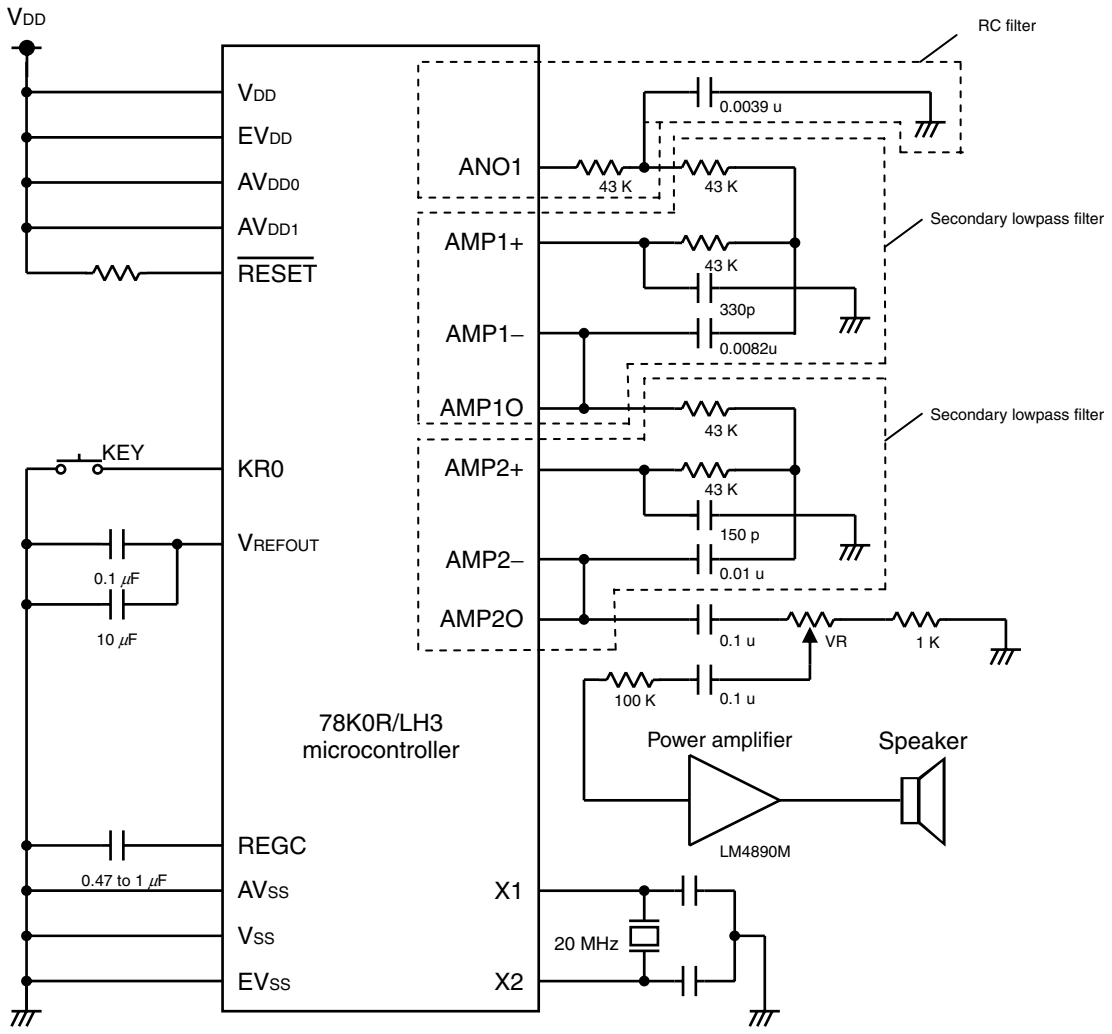
After initial settings have been completed, the microcontroller is put in the HALT mode. The HALT mode is released by occurrence of interrupt INTKR. Chattering of keys is avoided and whether a key is being input is decided. If a key is being input, ADPCM data is played back. Note that the ADPCM data to be played back is of 32 kbps (sampling frequency: 8 kHz, number of quantized bits: 4). If key input is not detected or after ADPCM data has been played back, the device is put in the HALT mode again.

CHAPTER 2 CIRCUIT DIAGRAM

This chapter provides a circuit diagram and describes the devices used in this sample program other than the microcontroller.

2.1 Circuit Diagram

A circuit diagram is shown below.



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} the same potential as V_{DD}.

3. Make AV_{SS} the same potential as EV_{SS} or V_{SS} and connect it directly to GND.

4. During voltage reference operation, be sure to connect a tantalum capacitor (capacitance: 10 μF ±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) to the V_{REFOUT}/AV_{REFP} pin for stabilizing the reference voltage. Furthermore, do not apply a voltage from the V_{REFOUT}/AV_{REFP} pin during voltage reference operation.

(Cautions are continued on the next page.)

5. **Connect REGC to V_{SS} via a capacitor (0.47 to 1 μF).**
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 V_{DD} or V_{SS} via a resistor.**
7. **In this sample program, the P40/TOOL0 and P41/TOOL1 pins are used for on-chip debugging.**
8. **An on-chip pull-up resistor is connected to the P70 pin.**

2.2 Used Devices Other than Microcontroller

The following devices are used in addition to the microcontroller:

(1) Power amplifier

A power amplifier is used to amplify the value of the output sound data. In this application example, the LM4890M is used.

(2) Variable resistor

This is used to adjust the volume of the output sound data.

(3) Speaker

Outputs sound data.

(4) Key

A key is used to start playing back sound data.

2.3 Pin Function List

The pin functions to be used are listed below.

Pin Function When External Device Is Connected		Alternate-Function Pin
Name	Function	
ANO1	Analog output of D/A converter	P111
AMP1+	Operational amplifier input (positive)	P25/ANI5
AMP1−	Operational amplifier input (negative)	P23/ANI3
AMP1O	Operational amplifier output	P24/ANI4
AMP2+	Operational amplifier input (positive)	P150/ANI8
AMP2−	Operational amplifier input (negative)	P26/ANI6
AMP2O	Operational amplifier output	P27/ANI7
KR0	Key input	P70

CHAPTER 3 SOFTWARE

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

3.1 Included Files

The following table shows the files included in the compressed file to be downloaded.

File Name	Description	Compressed (*.zip) File Included	
			
main.asm (Assembly language version) main.c (C language version)	Source file for hardware initialization processing and main processing of microcontroller	●Note 1	●Note 1
data_playrom.asm (Assembly language version) data_playrom.c (C language version)	ADPCM data table file	●Note 2	●Note 2
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer, selecting the internal high-speed oscillation clock frequency, setting up the LVI default start function, and setting up the on-chip debug operation.)	●	●
78K0R_Lx3_PlayBack.prw	Work space file for integrated development environment PM+		●
78K0R_Lx3_PlayBack.prj	Project file for integrated development environment PM+		●

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

2. “data_playrom.asm” is included with the assembly language version, and “data_playrom.c” with the C language version.

Caution This sample program uses an ADPCM library. Therefore, the standard header library (adpcmsp.h) and library file (adpcmsp.lib) of ADPCM-SP are necessary for a program in C, and the library file (adpcmsp.lib) of ADPCM-SP is necessary for a program in assembly language. Obtain ADPCM-SP from the download site of development tools (http://necel.com/micro/ja/development/asia/78k0r.html).

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.

- Channel 0 of timer array unit 0 (TAU0):
  Channel 0 of timer array unit 0 is used as an interval timer to avoid chattering of key input and to wait for stabilization of the voltage reference.

- Channel 4 of timer array unit 0 (TAU0):
  Channel 4 of timer array unit 0 is used as an interval timer to generate a sampling frequency of 8 kHz for playing back ADPCM data.

- Voltage reference:
  Generates a reference voltage of 2.0 V for the D/A converter.

- D/A converter:
  Outputs ADPCM data as sound data.

- Operational amplifiers:
  Used as filter circuits for the sound data output from the D/A converter.

- Low-voltage detector:
  Used to check that V_{DD} is 2.7 V or more.

- Pin function;
  The pin functions to be used are listed below.

Pin Function When External Device Is Connected		Alternate-Function Pin
Name	Function	
ANO1	Analog output of D/A converter	P111
AMP1+	Operational amplifier input (positive)	P25/ANI5
AMP1−	Operational amplifier input (negative)	P23/ANI3
AMP1O	Operational amplifier output	P24/ANI4
AMP2+	Operational amplifier input (positive)	P150/ANI8
AMP2−	Operational amplifier input (negative)	P26/ANI6
AMP2O	Operational amplifier output	P27/ANI7
KR0	Key input	P70

3.3 Initial Settings and Operation Overview

In this sample program, the main system clock is selected and the I/O port, timer array unit 0, voltage reference, operational amplifiers, and D/A converter are set up as initial settings.

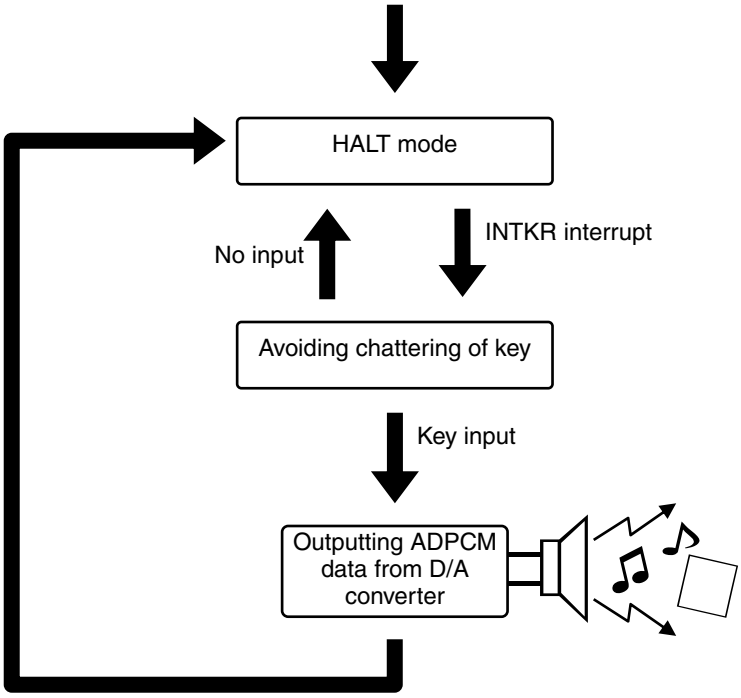
After completion of the initial settings, ADPCM data is played back by input of a key.

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

Initial settings

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

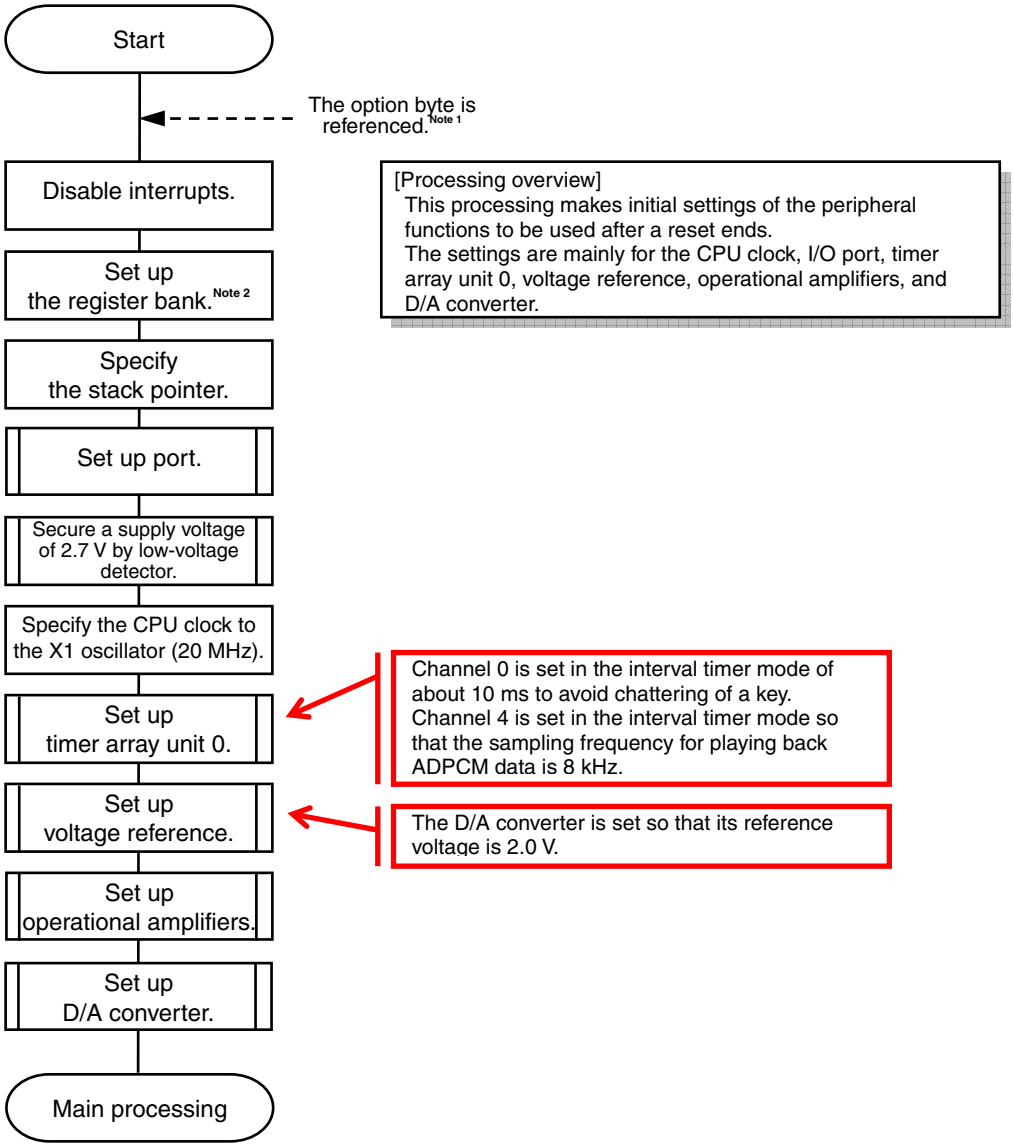
<Settings during initialization immediately after a reset ends>
- Setting up I/O ports
  - Setting P23, P25, P26, and P150 to input data to operational amplifiers
  - Setting P24 and P27 to output data from operation amplifiers
  - Setting KR0 to input key and detect key interrupt signal
- Securing a supply voltage of 2.7 V or more by using the function of low-voltage detector
- Specifying that the CPU clock run on the X1 oscillator (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 0 of timer array unit 0 in a mode in which it operates as an interval timer of about 10 ms to set settling time of the voltage reference and avoid chattering of a key input
  - Setting channel 4 in interval timer mode to set the sampling frequency for playing back ADPCM data to 8 kHz
- Setting up voltage reference
  - Selecting the voltage reference as a reference voltage source
  - Setting the output voltage of voltage reference to 2.0 V
  - Using channel 0 of timer array unit 0 to wait for about 20 ms until the operation of the voltage reference is stabilized
- Setting up operational amplifiers 1 and 2
- Setting up D/A converter
  - Selecting the real-time output mode as the operation mode
  - Setting the resolution to 12 bits
  - Selecting V_{REFOUT}/AV_{REFP} pin as the voltage reference source of the D/A converter



3.4 Flow Chart

A flow chart for the sample program is shown below.

<Settings during initialization immediately after a reset ends>



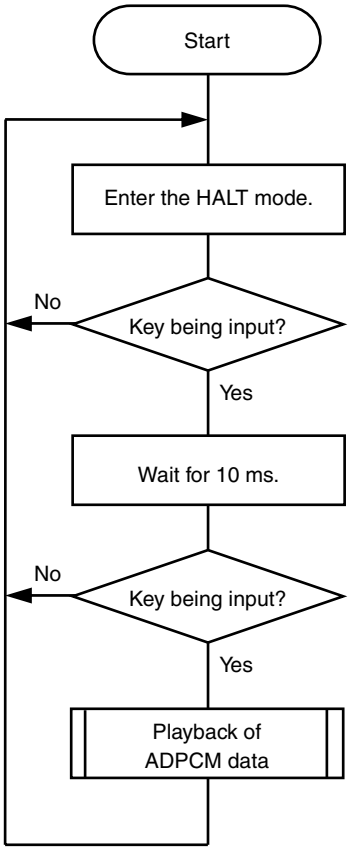
Notes 1. The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

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

2. The general-purpose registers of 78K0R/LH3 are configured in four register banks so that the registers used for normal processing and those used when an interrupt occurs can be changed on a bank basis in order to create an efficient program. In this sample program, only register bank 0 is used.

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

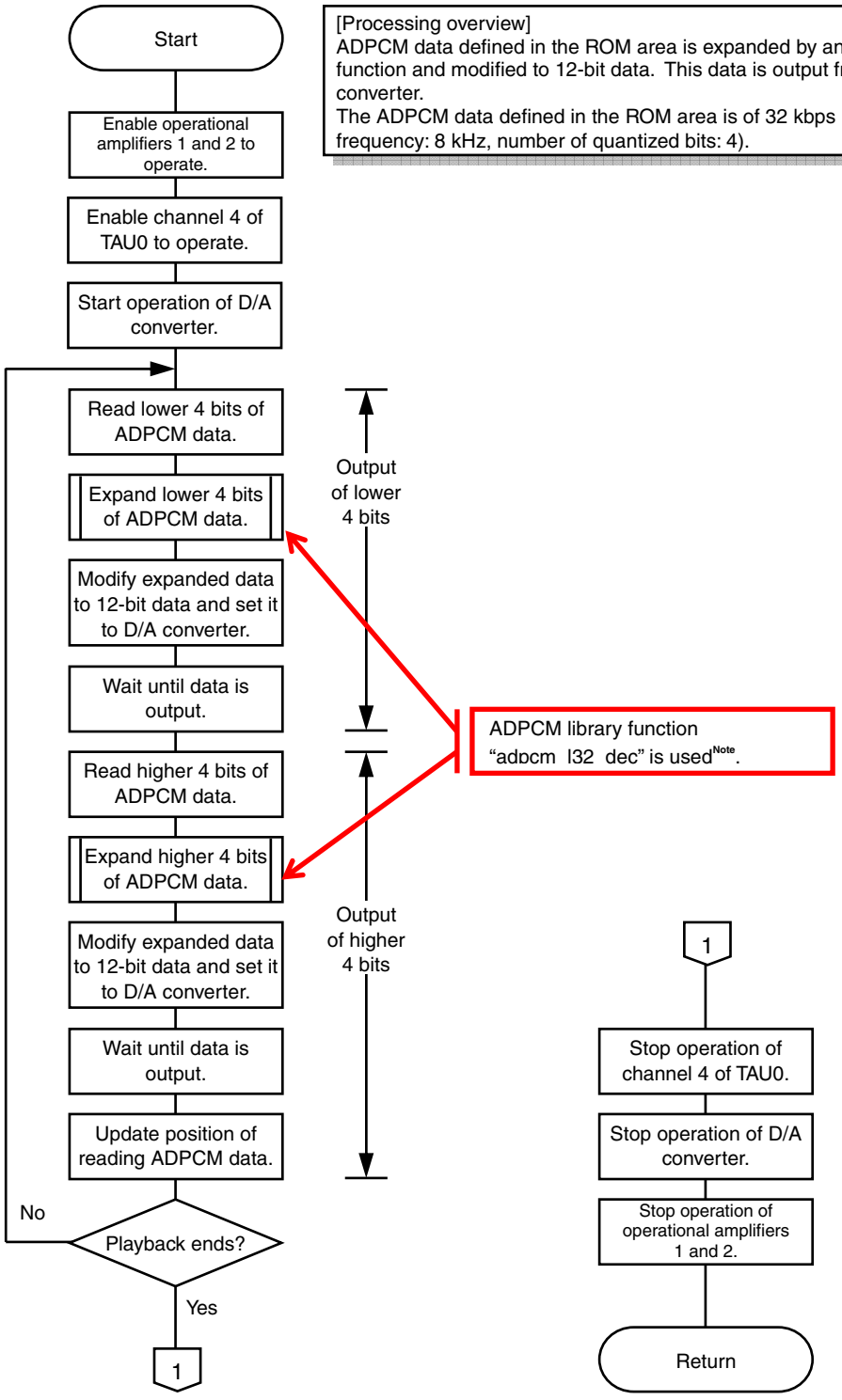
<Main processing>



[Processing overview]
When a key is input, ADPCM data is played back.

The device is in the HALT mode until interrupt INTKR is generated.
The device is released from the HALT mode when the INTKR interrupt is generated, waits for about 10 ms to avoid chattering of a key, and decides whether a key is being input.
If a key is being input, processing to play back ADPCM data is called.

<Playback processing>



Note For details of the function, refer to **ADPCM-SP Sound Compression/Expansion Software Package 78K0R Microcontroller User's Manual**.

Caution Do not allow any other processing to interrupt until the processing to play back ADPCM data is completed.

CHAPTER 4 SETTING METHODS

This chapter describes how to set up peripheral hardware macros, timer array unit 0, voltage reference, operational amplifiers, and D/A converter. It also provides software coding examples and details about playback processing.

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

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

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

4.1 Setting to Use Peripheral Hardware Macros

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

- Peripheral enable register 0 (PER0)

(1) Peripheral enable register 0 (PER0)

This register is used to enable or disable use of each peripheral hardware macro. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

TAU0EN	Control of timer array unit 0 input clock
0	Stops input clock supply.
1	Supplies input clock.

ADCEN	Control of A/D converter, operational amplifier, and voltage reference input clock
0	Stops input clock supply.
1	Supplies input clock.

DACEN	Control of D/A converter input clock
0	Stops input clock supply.
1	Supplies input clock.

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

4.2 Setting Up Timer Array Unit 0

The following five registers are used to set up timer array unit 0.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer mode registers 00, 04 (TMR00, TMR04)
- Timer data registers 00, 04 (TDR00, TDR04)
- Timer channel start register 0 (TS0)

[Example of procedure for setting up timer array unit 0 to play back sound data]

<1> Set bit 0 (TAU0EN) of peripheral enable register 0 (PER0) to 1 (see **4.1**).

<2> Set CK00 to $f_{CLK}/2^2$ and CK01 to f_{CLK} by using timer clock select register 0 (TPS0).

<3> Select CK00 as the operating clock of channel 0 and the interval timer mode as the operation mode by using timer mode register 00 (TMR00).

<4> Set the interval of channel 0 to about 10 ms by using timer data register 00 (TDR00).

<5> Select CK01 as the operating clock of channel 4 and the interval timer mode as the operation mode by using timer mode register 04 (TMR04).

<6> Set the interval of channel 4 to about 125 μs by using timer data register 04 (TDR04).

(1) Timer clock select register 0 (TPS0)

TPS0 is a 16-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel of timer array unit 0. CK01 is selected by bits 7 to 4 of TPS0, and CK00 is selected by bits 3 to 0.

Rewriting of TPS0 during timer operation is possible only in the following cases.

Rewriting of PRS000 to PRS003 bits:

Possible only when all the channels set to CKS0n = 0 are in the operation stopped state (TE0n = 0)

Rewriting of PRS010 to PRS013 bits:

Possible only when all the channels set to CKS0n = 1 are in the operation stopped state (TE0n = 0)

Figure 4-2. Format of Timer Clock Select Register 0 (TPS0)

0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS003	PRS002	PRS001	PRS000	Selection of operation clock (CK00) Notes 1, 2
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	**0**	**1**	**0**	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

PRS013	PRS012	PRS011	PRS010	Selection of operation clock (CK01) Notes 1, 2
0	**0**	**0**	**0**	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TT0 = 00FFH, TT1 = 000FH).

2. Only in the case of SDIV = 0, CCSmn = 1 and TISmn = 1, continuously use of TAUm is allowed, even when changing CPU clock (m = 0, 1, mn = 00 to 07, 10 to 13). However, the following limitation is existing.

- When changing CPU clock, source clock decrease/increase occurs as follows.
  Main system clock → Subsystem clock (CSS = 0 → 1): −1 clock
  Subsystem clock → Main system clock (CSS = 1 → 0): +1 clock

Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

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

(2) Timer mode registers 00, 04 (TMR00, TMR04)

TMR00 and TMR04 set an operation mode of channels 0 and 4 of timer array unit 0. These registers are used to select an operation clock (MCK), a count clock, whether the timer operates as the master or a slave, a start trigger and a capture trigger, the valid edge of the timer input, and an operation mode (interval, capture, event counter, one-count, or capture & one-count).

Figure 4-3. Format of Timer Mode Register 00 (TMR00) (1/2)

CKS 00	0	0	CCS 00	MAST ER00	STS 002	STS 001	STS 000	CIS 001	CIS 000	0	0	MD 003	MD 002	MD 001	MD 000

STS002	STS001	STS000	Setting of start trigger or capture trigger of channel 0
0	**0**	**0**	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function).
Other than above			Setting prohibited

MASTER00	Selection of slave/master of independent operation or combination operation function of channel 0
0	Operates as slave channel with independent operation or combination operation function.
1	Operates as master channel with combination operation function.
Only the even channel can be set as a master channel (MASTER00 = 1). Be sure to use the odd channel as a slave channel (MASTER00 = 0). Clear MASTER00 to 0 for a channel that is used with the independent operation function.	

CCS00	Selection of count clock (TCLK) of channel 0
0	Operation clock MCK specified by CKS00 bit
1	Valid edge of input signal input from TIpq pin, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 (the timer input used with channel 0 is selected by using TIS0 register).
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller.	

CKS00	Selection of operation clock (MCK) of channel 0
0	Operation clock CK00 set by TPS0 register
1	Operation clock CK01 set by TPS0 register
Operation clock MCK is used by the edge detector. A count clock (TCLK) is generated by setting the CCS00 bit	

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

Figure 4-3. Format of Timer Mode Register 00 (TMR00) (2/2)

CKS 00	0	0	CCS 00	MAST ER00	STS 002	STS 001	STS 000	CIS 001	CIS 000	0	0	MD 003	MD 002	MD 001	MD 000

Operation mode (set by MD003 to MD001)	MD000	Setting of starting counting and interrupt
• Interval timer mode	**0**	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation[Note]. At that time, interrupt is also generated
• Capture & one-count mode	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

MD003	MD002	MD001	MD000	Operation mode of channel 0	Count operation of TCR	Independent operation
0	**0**	**0**	1/**0**	Interval timer mode	Counting down	Possible
0	1	0	1/0	Capture mode	Counting up	Possible
0	1	1	0	Event counter mode	Counting down	Possible
1	0	0	1/0	One-count mode	Counting down	Impossible
1	1	0	0	Capture & one-count mode	Counting up	Possible
Other than above				Setting prohibited		

CIS001	CIS000	Selection of valid edge of TI00 pin input signal , $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 (the timer input used with channel 0 is selected by using TIS0 register).
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS002 to STS000 bits is other than 010B, set the CIS001 and CIS000 bits to 10B.		

Note If the start trigger (TS00 = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Caution Be sure to clear bits 14, 13, 5, and 4 to "0".

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

Figure 4-4. Format of Timer Mode Register 04 (TMR04) (1/2)

CKS 04	0	0	CCS 04	MAST ER04	STS 042	STS 041	STS 040	CIS 041	CIS 040	0	0	MD 043	MD 042	MD 041	MD 040

STS042	STS041	STS040	Setting of start trigger or capture trigger of channel 4
0	**0**	**0**	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function).
Other than above			Setting prohibited

MASTER04	Selection of slave/master of independent operation or combination operation function of channel 4
0	Operates as slave channel with independent operation or combination operation function.
1	Operates as master channel with combination operation function.
Only the even channel can be set as a master channel (MASTER04 = 1). Be sure to use the odd channel as a slave channel (MASTER04 = 0). Clear MASTER04 to 0 for a channel that is used with the independent operation function.	

CCS04	Selection of count clock (TCLK) of channel 4
0	Operation clock MCK specified by CKS04 bit
1	Valid edge of input signal input from TIpq pin, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 (the timer input used with channel 4 is selected by using TIS0 register).
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller.	

CKS04	Selection of operation clock (MCK) of channel 4
0	Operation clock CK00 set by TPS0 register
1	Operation clock CK01 set by TPS0 register
Operation clock MCK is used by the edge detector. A count clock (TCLK) is generated by setting CCS04 bit	

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

Figure 4-4. Format of Timer Mode Register 04 (TMR04) (2/2)

CKS 04	0	0	CCS 04	MAST ER04	STS 042	STS 041	STS 040	CIS 041	CIS 040	0	0	MD 043	MD 042	MD 041	MD 040

Operation mode (set by MD043 to MD041)	MD040	Setting of starting counting and interrupt
• Interval timer mode	**0**	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation[Note]. At that time, interrupt is also generated
• Capture & one-count mode	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

MD043	MD042	MD041	MD040	Operation mode of channel 4	Count operation of TCR	Independent operation
0	**0**	**0**	1/**0**	Interval timer mode	Counting down	Possible
0	1	0	1/0	Capture mode	Counting up	Possible
0	1	1	0	Event counter mode	Counting down	Possible
1	0	0	1/0	One-count mode	Counting down	Impossible
1	1	0	0	Capture & one-count mode	Counting up	Possible
Other than above				Setting prohibited		

CIS041	CIS040	Selection of valid edge of TI04 pin input signal , $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 (the timer input used with channel 4 is selected by using TIS0 register).
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS042 to STS040 bits is other than 010B, set the CIS041 and CIS040 bits to 10B.		

Note If the start trigger (TS04 = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

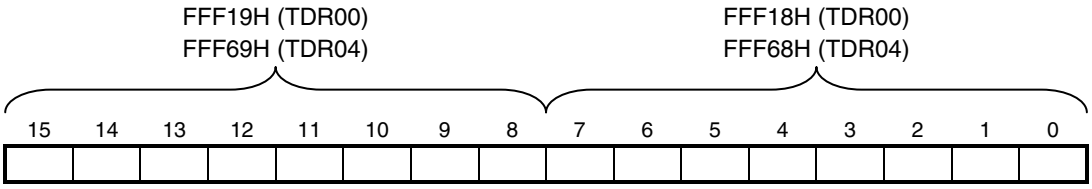
Caution Be sure to clear bits 14, 13, 5, and 4 to "0".

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

(3) Timer data registers 00, 04 (TDR00, TDR04)

TDR00 and TDR04 are 16-bit registers from which a capture function and a compare function can be selected. In this sample program, TDR00 and TDR04 are used as comparison registers. Counting down is started from the value set to TDR00 or TDR04. When the count value reaches 0000H, an interrupt signal (INTTM00 or INTMM04) is generated. TDR00 and TDR04 hold their value until it is rewritten.

Figure 4-5. Format of Timer Data Registers 00, 04 (TDR00, TDR04)



(4) Timer channel start register 0 (TS0)

TS0 is a trigger register that is used to clear a timer counter (TCR0n) and start the counting operation of each channel.

When each bit (TS0n) of this register is set to 1, the corresponding bit (TE0n) of timer channel enable status register 0 (TE0) is set to 1. TS0n is a trigger bit and cleared immediately when TE0n = 1.

Figure 4-6. Format of Timer Channel Start Register 0 (TS0)

0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TS00	Operation enable (start) trigger of channel 0
0	No trigger operation
1	TE00 is set to 1 and the count operation becomes enabled.[Note]

TS04	Operation enable (start) trigger of channel 4
0	No trigger operation
1	TE04 is set to 1 and the count operation becomes enabled.[Note]

Note In the interval timer mode, nothing is performed until a count clock is generated after the start trigger has been detected (TS0n = 1). The value of TDR0n is loaded to TCR0n at the first count clock and counting down is performed at the following count clocks.

Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. n = 7 to 0

2. When the TS0 register is read, 0 is always read.

4.3 Setting Up Voltage Reference

The following register is used to set up the voltage reference.

- Analog reference voltage control register (ADVRC)

[Example of procedure for setting up voltage reference to play back sound data]

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 (see **4.1**).
<2> Set bit 3 (VRSEL) of the analog reference voltage control register (ADVRC) to 1.
<3> Set bits 1 and 0 (VRGV, VRON) of the analog reference voltage control register (ADVRC) to 1.
<4> Wait for about 20 ms by software until the operation of the voltage reference is stabilized.

(1) Analog reference voltage control register (ADVRC)

This register is used to select the reference voltage supplies of the A/D and D/A converters, control the operation of the input gate voltage boost circuit for the A/D converter, and control the voltage reference (VR) operation.

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

Figure 4-7. Format of Analog Reference Voltage Control Register (ADVRC)

ADREF[Note]	0	0	0	VRSEL	0	VRGV	VRON

VRSEL	VRGV	VRON	Positive reference voltage supplies selection of A/D and D/A converters	Operation control of voltage reference	Output voltage selection of voltage reference	Operation control of input gate voltage boost circuit for A/D converter	Relationship with the conversion mode used
0	0	0	AV_{REFP} (external voltage reference input)	Stops operation (Hi-Z)	2.5 V	Stops operation	Can be set in conversion mode 1
0	1	0			2.0 V	Enables operation	Can be set in conversion mode 2 or 3
1	0	0	V_{REFOUT} (voltage reference output)	Stops operation (pull-down output)	2.5 V	Stops operation	–
1	0	1		Enables operation	2.5 V	Enables operation	Can be set in conversion mode 2 or 3
1	1	0		Stops operation (pull-down output)	2.0 V		–
1	**1**	**1**		Enables operation	2.0 V		Can be set in conversion mode 2 or 3
Other than above			Setting prohibited				

Note This is a function of the A/D converter and is not used in this sample program.

(**Cautions** and **Remark** are given on the next page.)

Cautions 1. Be sure to clear bits 6 to 4, and 2 to "0".

2. During voltage reference operation, be sure to connect a tantalum capacitor (capacitance: 10 μF ±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) to the V_{REFOUT}/AV_{REFP} pin for stabilizing the reference voltage. Furthermore, do not apply a voltage from the V_{REFOUT}/AV_{REFP} pin during voltage reference operation.

3. To use voltage reference output (V_{REFOUT}) to the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}), be sure to set VRON to 1 after setting VRSEL to 1.

4. Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).

5. Do not change the output voltage of the reference voltage by using VRGV during the voltage reference operation (VRON = 1).

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

4.4 Setting Up Operational Amplifier

The following four registers are used to set up the operational amplifiers.

- Peripheral enable register 0 (PER0)
- Operational amplifier control register (OAC)
- A/D port configuration register (ADPC)
- Port mode registers 2, 15 (PM2, PM15)

[Example of procedure for setting up operational amplifiers to play back sound data]

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 (see **4.1**).

<2> By using the ADPC register, specify the pins to be used (AMP1–, AMP1+, AMP1O, AMP2–, AMP2+, AMP2O) as analog input pins.

<3> By using the PM2 and PM15 registers, set the pins to be used (AMP1–, AMP1+, AMP1O, AMP2–, AMP2+, AMP2O) in the input mode.

<4> Enable operational amplifiers 1 and 2 to operate by setting the OAEN1 and OAEN2 bits of the OAC register to 1.

<5> Wait for about 20 μs until the operation of the operational amplifiers is stabilized.

(1) Operational amplifier control register (OAC)

The OAC register controls the operations of operational amplifiers 0 to 2.

Figure 4-8. Format of Operational Amplifier Control Register (OAC)

0	0	0	0	0	OAEN2	OAEN1	OAEN0

OAEN1	Operational amplifier 1 operation control
0	Stops operational amplifier 1 operation
1	Enables operational amplifier 1 operation

OAEN2	Operational amplifier 2 operation control
0	Stops operational amplifier 2 operation
1	Enables operational amplifier 2 operation

Cautions 1. Use the ADPC register to specify as analog inputs the pins to be used with operational amplifiers.

2. When using as digital inputs the pins of ports 2 and 15, which are not used with operational amplifiers, when the operational amplifiers are used, make sure that the input levels are fixed.

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.

(2) A/D port configuration register (ADPC)

This register switches the ANI0/AMP0–/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152, and ANI15/AV_{REFM}/P157 pins to analog input of A/D converter or digital I/O of port. Set pins to be used with operational amplifiers to the analog input.

Figure 4-9. Format of A/D Port Configuration Register (ADPC)

0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching											
					Port 15				Port 2							
					ANI15/AV_{REFP}/P157	ANI10/P152	ANI9/P151	ANI8/AMP2+/P150	ANI7/AMP2O/P27	ANI6/AMP2–/P26	ANI5/AMP1+/P25	ANI4/AMP1O/P24	ANI3/AMP1–/P23	ANI2/AMP0+/P22	ANI1/AMP0O/P21	ANI0/AMP0–/P20
0	**0**	**0**	**0**	**0**	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	A	A	D	D	D	D	D	D	D	D	D
0	1	0	1	0	A	A	D	D	D	D	D	D	D	D	D	D
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
Other than above					Setting prohibited											

Cautions 1. Set pins to be used with operational amplifiers in the input mode by using port mode registers 2 and 15 (PM2, PM15).

2. Be sure to clear bits 7 to 5 to "0".

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

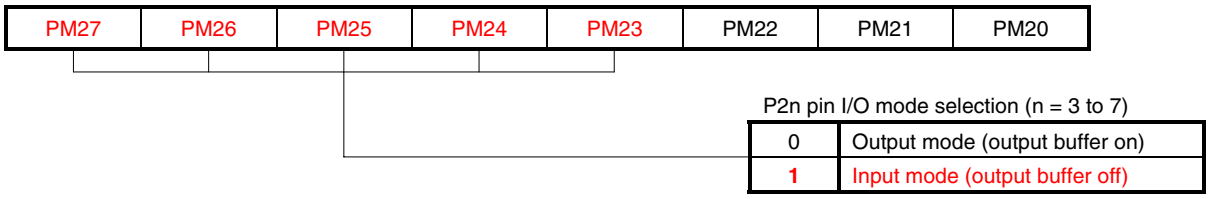
(3) Port mode registers 2, 15 (PM2, PM15)

When using AMP1–/ANI3/P23, AMP1O/ANI4/P24, AMP1+/ANI5/P25, AMP2–/ANI6/P26, AMP2O/ANI7/P27, and AMP2+/ANI8/P150 pins for the operational amplifiers, set PM23 to PM27 and PM150 to 1.

The output latches of PM23 to PM27 and PM150 at this time may be 0 or 1.

If PM23 to PM27 and PM150 are set to 0, they cannot be used as the pins of the operational amplifiers.

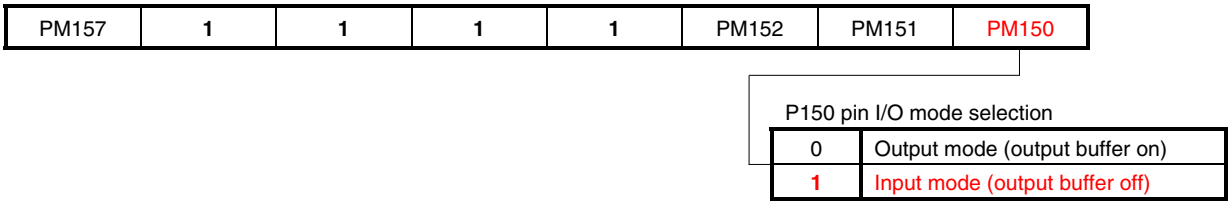
Figure 4-10. Format of Port Mode Register 2 (PM2)



Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

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

Figure 4-11. Format of Port Mode Register 15 (PM15)



Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

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

The AMP1–/ANI3/P23, AMP1O/ANI4/P24, AMP1+/ANI5/P25, AMP2–/ANI6/P26, AMP2O/ANI7/P27, and AMP2+/ANI8/P150 pins are as shown below depending on the settings of ADPC, ADS, PM2, PM15, OAENn bit and ADREF bit.

Table 4-1. Setting Functions of ANI3/AMP1–/P23, ANI5/AMP1+/P25, ANI6/AMP2–/P26, and ANI8/AMP2+/P150 Pins

ADPC Register	PM2, PM15 Registers	OAENn Bit	ADS Register	Setting Functions of ANI3/AMP1–/P23, ANI5/AMP1+/P25, ANI6/AMP2–/P26, and ANI8/AMP2+/P150 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Remarks 1. n = 1, 2

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

Caution When an operational amplifier is used, AMPn+, AMPn–, and AMPnO pins are used, so the alternative analog input functions cannot be used.

Table 4-2. Setting Functions of ANI4/AMP1O/P24 and ANI7/AMP2O/P27 Pins

ADPC Register	PM2, PM15 Registers	OAENn Bit	ADS Register	Setting Functions of ANI4/AMP1O/P24 and ANI7/AMP2O/P27 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Remarks 1. n = 1, 2

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

Caution When an operational amplifier is used, AMPn+, AMPn–, and AMPnO pins are used, so the alternative analog input functions cannot be used. The operational amplifier output signals, however, can be used as analog inputs.

4.5 Setting Up D/A Converter

The following three registers are used to set up the D/A converter.

- Peripheral enable register 0 (PER0)
- D/A converter mode register (DAM)
- D/A conversion value setting register W0 (DACSW0)

[Example of procedure for setting up D/A converter to play back sound data]

<1> Set bit 6 (DACEN) of peripheral enable register 0 (PER0) to 1 (see **4.1**).

<2> By using the D/A converter mode register (DAM), select the real-time mode as the operation mode of the D/A converter, a resolution of 12 bits, and V_{REFOUT} as the positive reference voltage source of the D/A converter.

<3> Set bit 4 (DACE0) of the D/A converter mode register (DAM) to 1 to enable the D/A conversion operation of channel 0 of the D/A converter.

<4> Set 800H (P-P: 1/2 of 2.0 V, sound amplitude: 0) as the default value of the D/A conversion value to the D/A conversion value setting register W0 (DACSW0).

(1) D/A converter mode register (DAM)

This register controls the operation of the D/A converter.

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

Figure 4-12. Format of D/A Converter Mode Register (DAM)

0	DAREF	DACE1	DACE0	DARES1	DARES0	DAMD1	DAMD0

DAMD0	Selection of D/A converter 0 operation mode
0	Normal mode
1	Real-time output mode

DARES0	Selection of D/A converter 0 resolution
0	8 bits
1	12 bits

DACE0	Control of D/A converter 0 conversion operation
0	Stops conversion operation
1	Enables conversion operation

DAREF	Positive reference voltage supply selection of D/A converter[Note 1]
0	AV_{DD1}
1	V_{REFOUT}[Note 2]/AV_{REFP}

Notes 1. The reference voltage of the D/A converter cannot be specified separately for each channel because it is common to both channels.

2. To use an output voltage of the voltage reference for the positive reference voltage of the D/A converter (DA_{REFP}), start operating the voltage reference before setting the D/A conversion operation (refer to the **78K0R/Lx3 User's Manual**). Furthermore, do not change the voltage reference setting during the D/A conversion operation.

Remarks 1. The values written in red in the above figure are specified in this sample program.

2. The positive reference voltage of the D/A converter is as follows, according to the DAREF, VRSEL and VRON settings.

Table 4-3. Settings of DAREF, VRSEL and VRON

DAREF	VRSEL	VRON	Positive Reference Voltage of D/A Converter (DA_{REFP})
0	×	×	AV_{DD1}
1	0	0	AV_{REFP}
1	1	1	V_{REFOUT}

×: don't care

(2) D/A conversion value setting register W0 (DACSW0)

This register is used to set an analog voltage value to be output to the ANO0 pin, when the D/A converter is used.

DACSW0 and DACSW1 can be set by a 16-bit memory manipulation instruction.

Figure 4-13. Format of D/A Conversion Value Setting Register W0 (DACSW0)

0	**0**	**0**	**0**	DACSW011	DACSW010	DACSW09	DACSW08	DACSW07	DACSW06	DACSW05	DACSW04	DACSW03	DACSW02	DACSW01	DACSW00
				FFF59H				FFF58H							

Caution Rewriting D/A conversion value setting register Wn (DACSWn) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).

Remarks 1. The relations between the resolutions and analog output voltages (V_{ANOn}) of the D/A converter are as follows.

- 8-bit resolution (DARESn = 0):
  V_{ANOn} = Reference voltage for D/A converter × (DACSWn7 to DACSWn0)/256
- 12-bit resolution (DARESn = 1):
  V_{ANOn} = Reference voltage for D/A converter × (DACSWn11 to DACSWn0)/4096

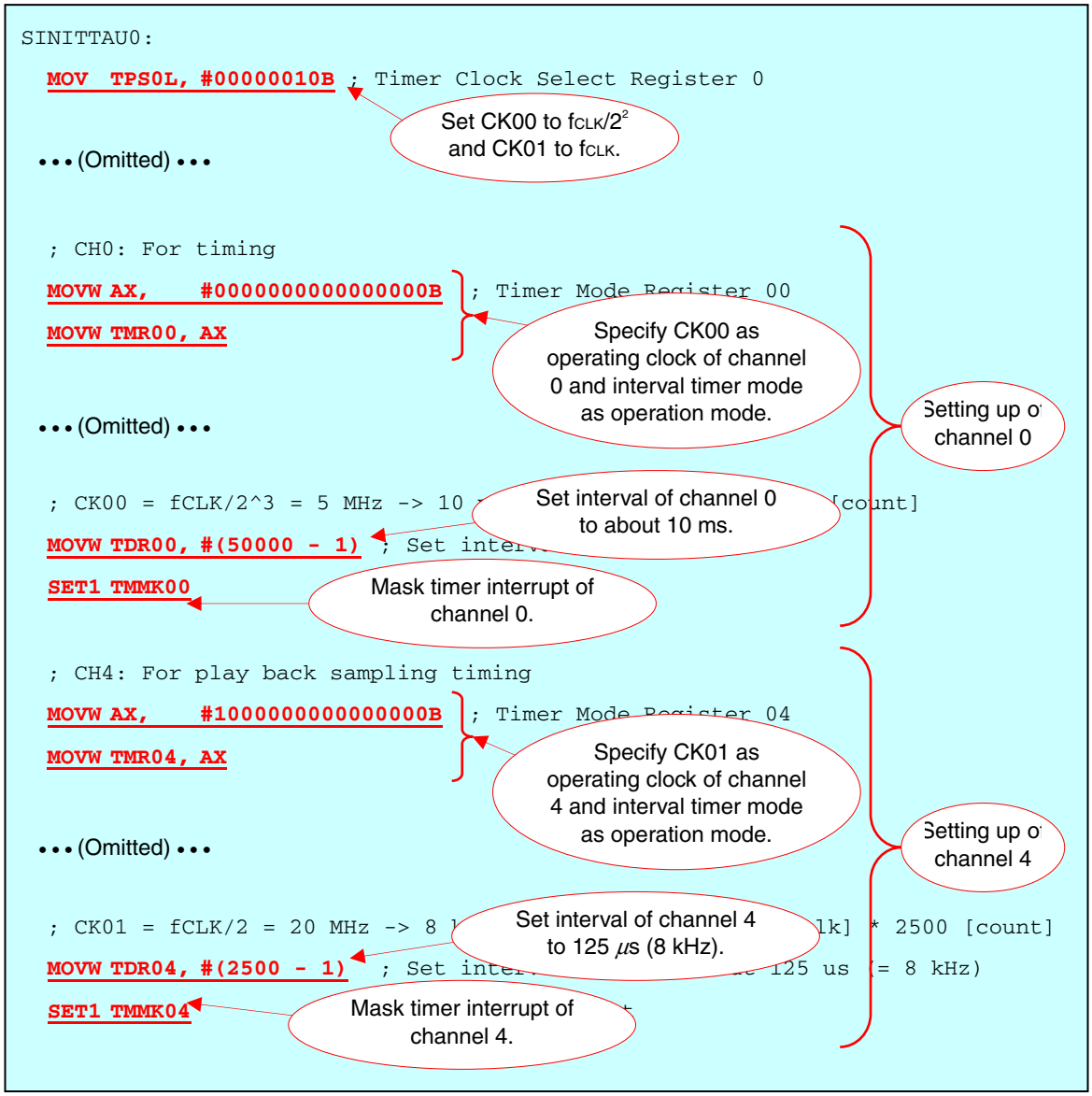
2. n = 0, 1

4.6 Software Coding Examples

The settings to be specified for timer array unit 0, voltage reference, operational amplifiers, and D/A converter in the sample program are shown below as a software coding example.

(1) Assembly language

<1> Setting up timer array unit 0



<2> Setting up voltage reference

```
SINITVR:
    MOV      ADVRC, #00001000B    ; Analog ref...                 ...rol register
```

Specify VREFOUT (voltage reference output) as reference voltage of D/A converter.

• • • (Omitted) • • •

Select 2.0 V as output voltage of voltage reference.

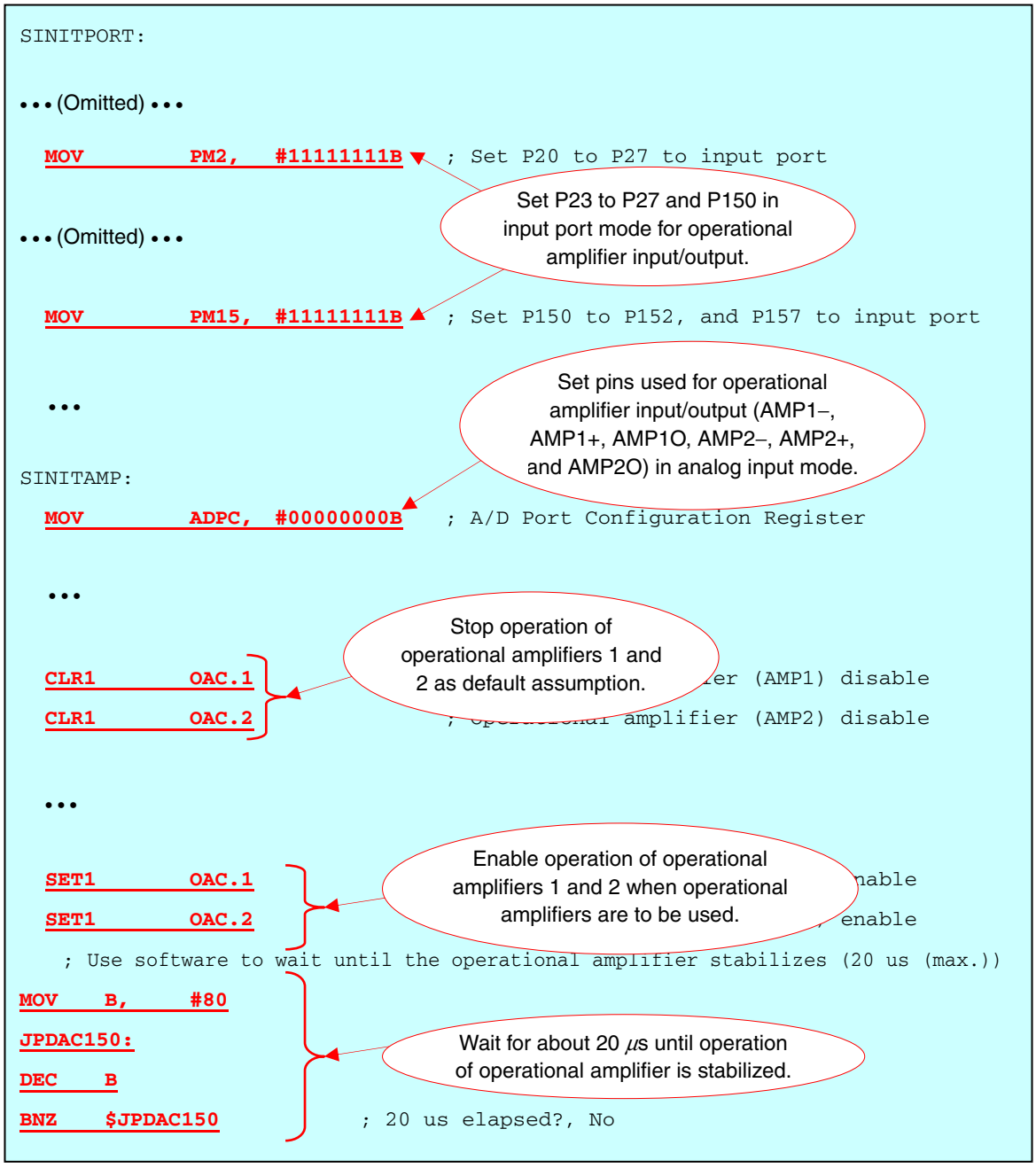
```
    SET1     ADVRC.0              ; Enables operation
    SET1     ADVRC.1
```

Enable operation of voltage reference.

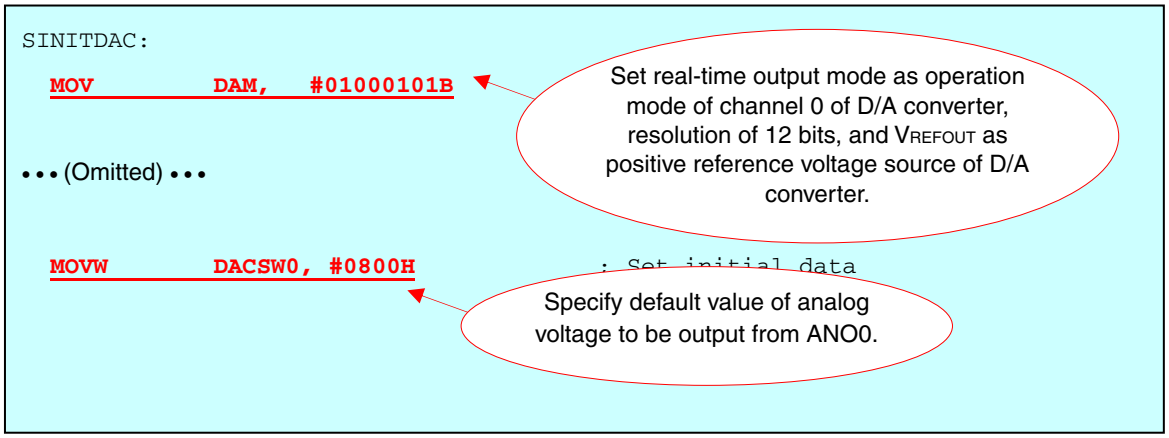
```
    ; Wait for settling time to 20 ms (over 17 msec)
MOV          B,    #2             ; Set counter
SET1         TS0L.0               ; Start TAU0 CH0
JINIVR100:
CLR1         TMIF00               ; Clear interrupt request flag
JINIVR200:
NOP
BF           TMIF00, $JINIVR200
DEC          B
BNZ          $JINIVR100           ; No

CLR1         TMIF00               ; Clear interrupt request flag
SET1         TT0L.0               ; Stop TAU0 CH0
```

Wait for about 20 ms until operation of voltage reference is stabilized.

<3> Setting up operational amplifiers

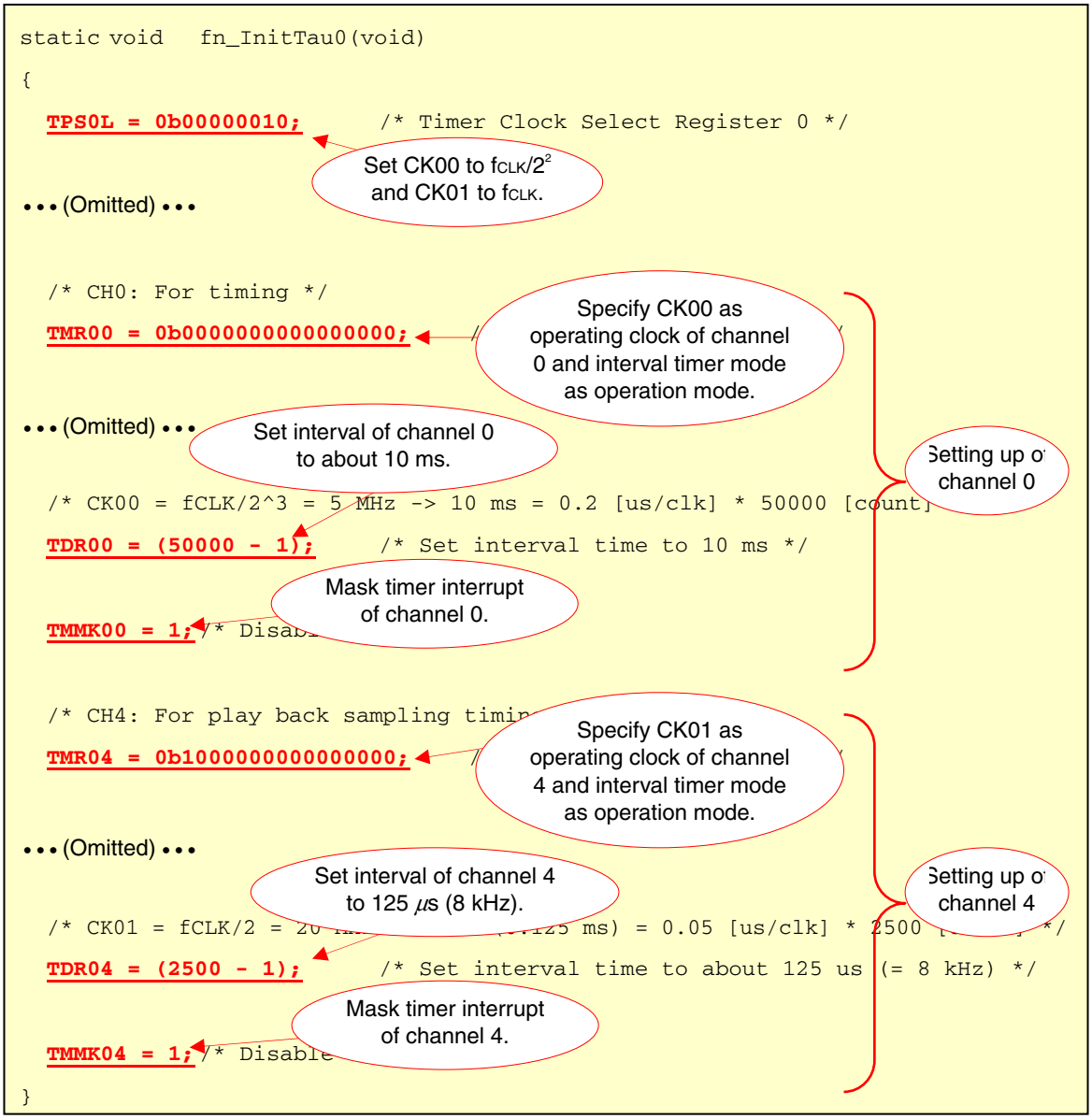


<4> Setting up D/A converter



(2) C language

<1> Setting up timer array unit 0



<2> Setting up voltage reference

```
static void   fn_InitVr(void)
{
  ADVRC = 0b00001000;       /* Analog reference voltage control register */

  ••• (Omitted) •••

  ADVRC.0 = 1;                                                        */
  ADVRC.1 = 1;

  /* Wait for settling time to 20 ms (over 17 msec) */
  TS0L.0 = 1; /* Start TAU0 CH0 */
  for(work = 2; work > 0; work--){       /* Wait 10 msec*2 */
    TMIF00 = 0;      /* Clear interrupt request flag */
    while(!TMIF00){
      NOP();         /* Wait 10 msec */
    }
  }
  TMIF00 = 0; /* Clear interrupt request flag */
  TT0L.0 = 1; /* Stop TAU0 CH0 */
}
```

Specify V_{REFOUT} (voltage reference output as reference voltage of D/A converter.

Select 2.0 V as output voltage of voltage reference.

Enable operation of voltage reference.

Wait for about 20 ms until operation of voltage reference is stabilized.

<3> Setting up operational amplifiers

```
static void   fn_InitPort(void)
{
••• (Omitted) •••
  PM2 = 0b11111111; /* Set P20 to P27 to input port */
••• (Omitted) •••
  PM15 = 0b11111111;/* Set P150 to P152, and P157 to input port */
••• (Omitted) •••
static void   fn_InitAmp(void)
{
  ADPC = 0b00000000;/* A/D Port Co
••• (Omitted) •••
  OAC.1 = 0; /* Oper                able */
  OAC.2 = 0; /* Operational                 AMP2) disable */
}
••• (Omitted) •••
  OAC.1 = 1; /* Oper
  OAC.2 = 1; /* Opera
    /* Use software to wait until the operational amplifier stabilizes (20 us (max.)) */
for(loop = 40; loop > 0; loop--){
NOP();
  }
```

Set P23 to P27 and P150 in input port mode for operational amplifier input/output.

Set pins used for operational amplifier input/output (AMP1–, AMP1+, AMP1O, AMP2–, AMP2+, and AMP2O) in analog input mode.

Stop operation of operational amplifiers 1 and 2 as default assumption.

Enable operation of operational amplifiers 1 and 2 when operational amplifiers are to be used.

Wait for about 20 μs until operation of operational amplifier is stabilized.

<4> Setting up D/A converter

```
static void   fn_InitDac(void)
{
  DAM = 0b01000101;  /* D
• • • (Omitted) • • •
  DACSW0 = 0x0800;  /* Set
}
```

Set real-time output mode as operation mode of channel 0 of D/A converter, resolution of 12 bits, and VREFOUT as positive reference voltage source of D/A converter.

Specify default value of analog voltage to be output from ANO0.

4.7 Playback Processing

This section explains playback processing.

As playback processing in an assembly language, the following operations are performed.

<1> As preparations for playback, the ADPCM library and variables are initialized and the hardware is enabled to operate.
  (a) Specifying the start address of the ADPCM data table (“TPLAYDATA” table) defined in the ROM area as the position of reading ADPCM data
  (b) Calling initialization processing to use the ADPCM library functions
  (c) Enabling the operations of operational amplifiers 1 and 2
  (d) Enabling the operation of the D/A converter
  (e) Enabling the operation of channel 4 of timer array unit 0
  (f) Initializing a counter that reads the ADPCM data

<2> The ADPCM data is read, its lower 4 bits are expanded and modified, and the resultant data is output from the D/A converter.
  (a) Reading the ADPCM data and expanding the lower 4 bits to signed 16-bit data by using an ADPCM library function
  (b) Adding 8000H to the expanded signed 16-bit data to modify it to unsigned 16-bit data
  (c) Shifting the modified unsigned 16-bit data two times to the right to modify it to unsigned 12-bit data
  (d) Setting the modified unsigned 12-bit data to D/A conversion value setting register W0 (DACSW0)
  (e) Waiting until the data set to D/A conversion value setting register W0 (DACSW0) is output from the D/A converter

<3> The ADPCM data is read, its higher 4 bits are expanded and modified, and the resultant data is output from the D/A converter. The details of this processing are the same as steps (a) to (e) in <2> above.

<4> The counter that reads the ADPCM data and the position of reading the ADPCM data are updated.

<5> Steps <3> to <5> are repeated until all the ADPCM data defined in the ROM area are output.

<6> The operation of the hardware is stopped as playback end processing.
  (a) Stopping the operation of channel 4 of timer array unit 0
  (b) Stopping the operation of the D/A converter
  (c) Stopping the operation of operational amplifiers 1 and 2

The ADPCM data to be played back is of 32 kbps (sampling frequency: 8 kHz, number of quantized bits: 4). The ADPCM data table (“TPLAYDATA” table) in an assembly language and the size of the ADPCM data table are defined in “data_playrom.asm”, which must be externally referenced.

Clear the higher 4 bits of the final data of the ADPCM table to 0 so that the amplitude of the sound data output from the D/A converter is 0.

To use the ADPCM library function, the function must be externally referenced. Externally reference the function by prefixing “_” to the function name as shown below.

```
EXTRN       _adpcm_init
EXTRN       _adpcm_l32_dec
```

When the ADPCM library function is used, a C routine is called. For how to call the C routine from an assembly language, refer to **CC78K0R C Compiler Language User’s Manual**.

The multiplier/divider is used in the multiplication mode during processing for expanding the ADPCM library function. Immediately before calling the expansion processing, therefore, set the operation mode of the multiplier/divider to the multiplication mode. The ADPCM library used in this sample program is for the 78K0R/Kx3. Therefore, the operation mode is not changed during expansion processing.

For details of the ADPCM library function, refer to **ADPCM-SP Sound Compression/Expansion Software Package 78K0R Microcontroller User's Manual**.

```
SPLAYDAC:
        ;---------------------------------;
        ;       Prepare for playing       ;
        ;---------------------------------;
<1>(a)  MOVW   HL,    #LOWW TPLAYDATA     ; Set start playing addr (low 16 bits)
        MOV    ES,    #HIGHW TPLAYDATA    ; Set start playing addr (high 4 bits)

JPDAC100:
(b)     MOVW   AX,    #LOWW RADPCMWORK
        CALL   !!_adpcm_init               ; ADPCM process Initialization

        ; Operational amplifier setting
(c)     SET1   OAC.1                       ; Operational amplifier (AMP1) enable
        SET1   OAC.2                       ; Operational amplifier (AMP2) enable
        ; Use software to wait until the operational amplifier stabilizes (20 us (max.))
        MOV    B,     #80
JPDAC150:
        DEC    B
        BNZ    $JPDAC150                   ; 20 us elapsed?, No

        ; D/A converter setting
(d)     SET1   DACE0                       ; D/A converter CH0 enable

        ; TAU0 CH4 setting for output timing
(e)     CLR1   TMIF04                      ; Clear interrupt request flag
        SET1   TS0L.4                      ; Start TAU0 CH4

        ;*********************************;
        ;*-------------------------------*;
        ;*   Decode and play PCM data    *;
        ;*-------------------------------*;
        ;*********************************;
(f)     MOVW   RPLAYCOUNT,#0               ; Clear output data counter

JPDAC200:
<2>     MOVW   AX,    RPLAYCOUNT           ; Get number of output times
        CMPW   AX,    !TPLAYSIZE           ; Finished all data output?
        BNC    $JPDAC300                   ;  Yes
```

```
        ;**********************;
        ;*   Play low 4 bits  *;
        ;**********************;
        ; Decompression of ADPCM data (low 4 bits -> 16 bits)
<3>(a)  MOVW   AX,    #LOWW RADPCMWORK  ; Set work area for _adpcm_l32_dec
        PUSH   AX                  ; Push argument
        MOV    A,     ES:[HL]      ; Get compressed data
        AND    A,     #00FH        ; Clear high 4 bits
        MOV    X,     A
        CLRB   A
        CLR1   DIVMODE      ; Set multiplication mode (for _adpcm_l32_dec)
        CALL   !!_adpcm_l32_dec    ; Decompression of PCM data
        POP    AX                  ; Pop argument
        MOVW   AX,    BC           ; Get decompression data

        ; Adjust play data
(b)     ADDW   AX,    #8000H       ; Adjust sign
(c)     SHRW   AX,    (16-12)      ; Right-align data

JPDAC220:
(d)     MOVW   DACSW0, AX          ; Set play data

        ; Waiting for the output to be completed
(e) JPDAC230:
       NOP
       BF     TMIF04, $JPDAC230   ; The output to be completed?, No
       CLR1   TMIF04              ; Clear interrupt request flag

        ;**********************;
        ;*   Play high 4bit   *;
        ;**********************;
        ; Decompression of ADPCM data (high 4 bits -> 16 bits)
<4>(a)  MOVW   AX,    #LOWW RADPCMWORK    ; Set work area for _adpcm_l32_dec
        PUSH   AX                  ; Push argument
        MOV    A,     ES:[HL]      ; Get compressed data
        SHR    A,     4
        MOV    X,     A
        CLRB   A
        CLR1   DIVMODE      ; Set multiplication mode (for _adpcm_l32_dec)
        CALL   !!_adpcm_l32_dec    ; Decompression of PCM data
        POP    AX                  ; Pop argument
        MOVW   AX,    BC           ; Get decompression data

        ; Adjust play data
(b)     ADDW   AX,    #8000H       ; Adjust sign
(c)     SHRW   AX,    (16-12)      ; Right-align data

JPDAC270:
(d)     MOVW   DACSW0,       AX    ; Set play data

        ; Waiting for the output to be completed
(e) JPDAC280:
       NOP
       BF     TMIF04, $JPDAC280   ; The output to be completed?, No
       CLR1   TMIF04              ; Clear interrupt request flag
```

```
<5>     INCW   RPLAYCOUNT          ; Update play counter
        INCW   HL                  ; Next play data
        BR     JPDAC200

JPDAC300:
        ;---------------------------------;
        ;          Finish playing         ;
        ;---------------------------------;
<6>(a)  SET1   TT0L.4              ; Stop TAU0 CH4
   (b)  CLR1   DACE0               ; D/A converter CH0 disable
   (c)  CLR1   OAC.1               ; Operational amplifier (AMP1) disable
        CLR1   OAC.2               ; Operational amplifier (AMP2) disable

JPDAC900:
        RET
```

Playback processing in C performs operations similar to those of processing in an assembly language.

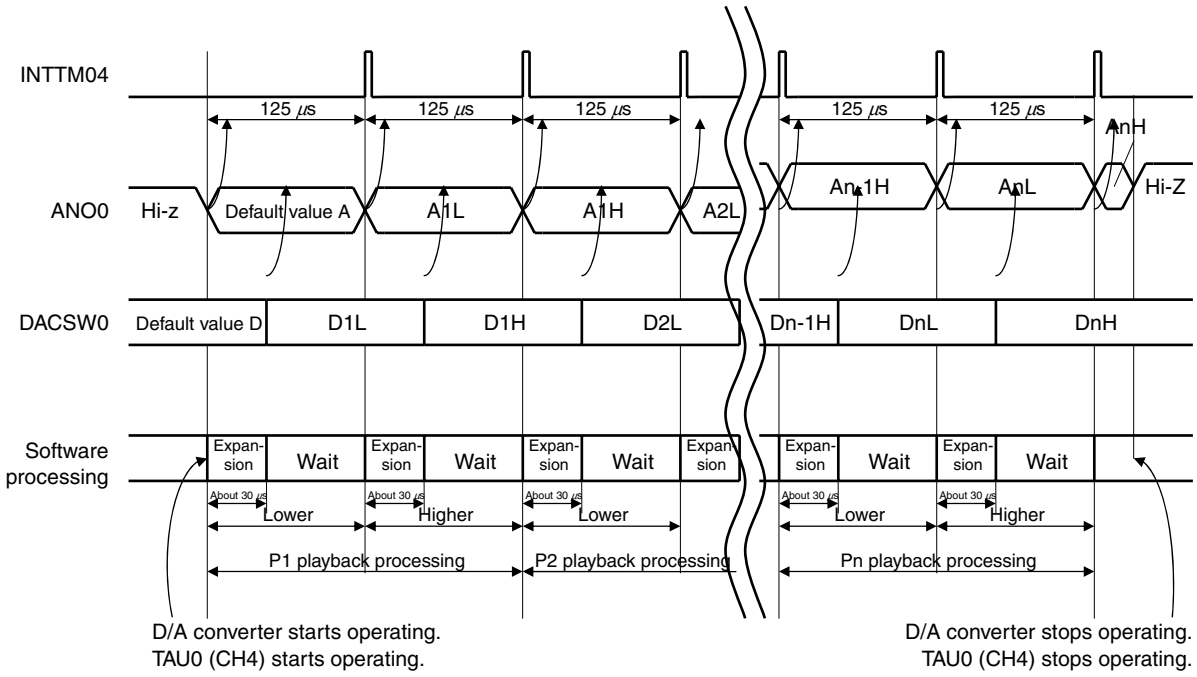
The ADPCM data table (“aPlayData” table) in C and the size of the ADPCM data table are defined in “data_playrom.c”, which must be externally referenced.

To use the ADPCMlibrary, “adpcmsp.h” must be included by #include.

```
static void   fn_PlayDac(void)
{
  unsigned char *   pucPlayData;  /* Start playing addr */
  unsigned short    ushPlayCount; /* Output data counter */
  unsigned short    ushData;      /* Decompression data */
  unsigned short    loop;         /* Waiting counter */

  /*---------------------------------*/
  /*       Prepare for playing       */
  /*---------------------------------*/
  /* Set play data addr and size */
  pucPlayData = aPlayData;        /* Set start playing address */

  adpcm_init(ushAdpcmWork);       /* ADPCM process Initialization */

 /* operational amplifier setting */
  OAC.1 = 1;  /* Operational amplifier (AMP1) enable */
  OAC.2 = 1;  /* Operational amplifier (AMP2) enable */
  /*Use software to wait until the operational amplifier stabilizes (20 us (max.))*/
  for(loop = 40; loop > 0; loop--){
    NOP();
  }

  /* D/A converter setting */
  DACE0 = 1;  /* D/A converter CH0 enable */

  /* TAU0 CH4 setting for output timing */
  TMIF04 = 0; /* Clear interrupt request flag */
  TS0L.4 = 1; /* Start TAU0 CH4 */
```

```
  /**********************************/
  /*--------------------------------*/
  /*     Decode and play PCM data    */
  /*--------------------------------*/
  /**********************************/
  for(ushPlayCount = 0; ushPlayCount < ushDataSize; ushPlayCount++){
    /***********************/
    /*   Play low 4 bits  */
    /***********************/
    /* Decompression of ADPCM data (low 4 bits -> 16 bits)*/
    DIVMODE = 0;    /* Set multiplication mode (for _adpcm_l32_dec) */
    ushData = (unsigned short)adpcm_l32_dec(pucPlayData[ushPlayCount] & 0x0f,
ushAdpcmWork);

    /* Adjust sign & right-align data */
    ushData = (unsigned short)((ushData + 0x8000) >> (16-12));

    /* Set play data */
    DACSW0 = ushData;

    /* Waiting for the output to be completed */
    while(!TMIF04){
      NOP();
    }
    TMIF04 = 0;     /* Clear interrupt request flag */

    /***********************/
    /*   Play high 4 bits  */
    /***********************/
    /* Decompression of ADPCM data (high 4 bits -> 16 bits) */
    DIVMODE = 0;    /* Set multiplication mode (for _adpcm_l32_dec) */
    ushData = (unsigned short)adpcm_l32_dec((pucPlayData[ushPlayCount] >> 4) &
0x0f, ushAdpcmWork);

    /* Adjust sign & right-align data */
    ushData = (unsigned short)((ushData + 0x8000) >> (16-12));

    /* Set play data */
    DACSW0 = ushData;

    /* Waiting for the output to be completed */
    while(!TMIF04){
      NOP();
    }
    TMIF04 = 0;     /* Clear interrupt request flag */
  }

  /*--------------------------------*/
  /*         Finish playing         */
  /*--------------------------------*/
  TT0L.4 = 1;/* Stop TAU0 CH4 */
  DACE0 = 0;  /* D/A converter CH0 disable */
  OAC.1 = 0;  /* Operational amplifier (AMP1) disable */
  OAC.2 = 0;  /* Operational amplifier (AMP2) disable */

}
```

Here is a timing chart for the playback processing.



Caution Clear the higher 4 bits of the final data of the ADPCM table to 0 so that the amplitude of the sound data output from the D/A converter is 0.

Remarks 1. n = Size of ADPCM data table

2. Pm: ADPCM data of 1 byte in ADPCM data table

3. Default value D: 800H

DmL: Expanded and modified value of lower 4 bits of Pm

DmH: Expanded and modified value of higher 4 bits of Pm

4. Default value A: Analog voltage output when the set value of DACSW0 is default value D (800H)

Amk: Analog voltage output when the set value of DACSW0 is Dmk

5. m = 1 to Size of ADPCM data table

k = L, H

CHAPTER 5 RELATED DOCUMENTS

Document Name		English
78K0R/Lx3 User's Manual		PDF
78K0R Series Instructions User's Manual		PDF
RA78K0R Assembler Package User's Manual	Language	PDF
	Operation	PDF
CC78K0R C Compiler User's Manual	Language	PDF
	Operation	PDF
ADPCM-SP Sound Compression/Expansion Software Package 78K0R Microcontroller User's Manual		PDF
PM+ Project Manager User's Manual		PDF

APPENDIX A PROGRAM LIST

As a program list example, the source program is shown below. However, the source program of “data_playrom.asm” and “data_playrom.c” that define the ADPCM data table is omitted.

● main.asm (assembly language version)

```
;
; Copyright (C) NEC Electronics Corporation 2006
; NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
; All rights reserved by NEC Electronics Corporation.
; This program must be used solely for the purpose for which
; it was furnished by NEC Electronics Corporation. No part of this
; program may be reproduced or disclosed to others, in any
; form, without the prior written permission of NEC Electronics
; Corporation. Use of copyright notice dose not evidence
; publication of the program.
;

;------------------------------------------------------------------------;
; Extern function                                                        ;
;------------------------------------------------------------------------;
     EXTRN  _adpcm_init               ; ADPCM process initialize
     EXTRN  _adpcm_l32_dec            ; ADPCM data decode

;------------------------------------------------------------------------;
; Extern variables/constants                                             ;
;------------------------------------------------------------------------;
     EXTRN  TPLAYDATA           ; Sound data
     EXTRN  TPLAYSIZE           ; Size of sound data

;------------------------------------------------------------------------;
; Vector table initialize                                                ;
;------------------------------------------------------------------------;
TVCT1CSEG   AT     000000H
     DW     IRESET              ; (00H) RESET, POC, LVI, WDT, TRAP
TVCT2CSEG   AT     000004H
     DW     IRESET              ; (04H) INTWDTI
     DW     IRESET              ; (06H) INTLVI
     DW     IRESET              ; (08H) INTP0
```

```
DW      IRESET              ; (0AH) INTP1
DW      IRESET              ; (0CH) INTP2
DW      IRESET              ; (0EH) INTP3E
DW      IRESET              ; (10H) INTP4
DW      IRESET              ; (12H) INTP5
DW      IRESET              ; (14H) INTST3
DW      IRESET              ; (16H) INTSR3
DW      IRESET              ; (18H) INTSRE3
DW      IRESET              ; (1AH) INTDMA0
DW      IRESET              ; (1CH) INTDMA1
DW      IRESET              ; (1EH) INTST0, INTCSI00
DW      IRESET              ; (20H) INTSR0, INTCSI01
DW      IRESET              ; (22H) INTSRE0
DW      IRESET              ; (24H) INTST1, INTCSI10, INTIIC10
DW      IRESET              ; (26H) INTSR1
DW      IRESET              ; (28H) INTSRE1
DW      IRESET              ; (2AH) INTIICA
DW      IRESET              ; (2CH) INTTM00
DW      IRESET              ; (2EH) INTTM01
DW      IRESET              ; (30H) INTTM02
DW      IRESET              ; (32H) INTTM03
DW      IRESET              ; (34H) INTAD
DW      IRESET              ; (36H) INTRTC
DW      IRESET              ; (38H) INTRTCI
DW      IRESET              ; (3AH) INTKR
DW      IRESET              ; (3CH) INTST2, INTCSI20, INTIIC20
DW      IRESET              ; (3EH) INSR2
DW      IRESET              ; (40H) INTSRE2
DW      IRESET              ; (42H) INTTM04
DW      IRESET              ; (44H) INTTM05
DW      IRESET              ; (46H) INTTM06
DW      IRESET              ; (48H) INTTM07
DW      IRESET              ; (4AH) INTP6
DW      IRESET              ; (4CH) INTP7
DW      IRESET              ; (4EH) INTP8
DW      IRESET              ; (50H) INTP9
DW      IRESET              ; (52H) INTP10
DW      IRESET              ; (54H) INTP11
DW      IRESET              ; (56H) INTTM10
```

```
     DW     IRESET              ; (58H) INTTM11
     DW     IRESET              ; (5AH) INTTM12
     DW     IRESET              ; (5CH) INTTM13
     DW     IRESET              ; (5EH) INTMD
TVCT3CSEG   AT     00007EH
     DW     IRESET              ; (7EH) BRK

;---------------------------------------------------------------------;
; Stack area definition                                               ;
;---------------------------------------------------------------------;
DSTK DSEG   IHRAM               ; Stack Area address
STACKEND:
            DS     60H
STACKTOP:

;---------------------------------------------------------------------;
; Local constants                                                     ;
;---------------------------------------------------------------------;
;---------------------------------------------------------------------;
; Global variables                                                    ;
;---------------------------------------------------------------------;
;---------------------------------------------------------------------;
; Local variables                                                     ;
;---------------------------------------------------------------------;
DPMAIN      DSEG   SADDRP
     RADPCMWORK:   DS     32     ; Work area for ADPCM process
     RPLAYCOUNT:   DS     2      ; Play data counter

;---------------------------------------------------------------------;
; Code                                                                ;
;---------------------------------------------------------------------;
XMAINCSEG   UNIT
;---------------------------------------------------------------------;
; Hardware initialization                                             ;
;---------------------------------------------------------------------;
     PUBLIC IRESET
IRESET:
     ;--------------------------------------;
     ;         Disable all interrupts       ;
```

```
;----------------------------------------;
DI

;----------------------------------------;
;         Set register bank              ;
;----------------------------------------;
SEL    RB0                 ; Use register bank 0

;----------------------------------------;
;         Set stack pointer              ;
;----------------------------------------;
MOVW   SP,    #LOWW STACKTOP

;----------------------------------------;
;        Initialization of port          ;
;----------------------------------------;
CALL   !!SINITPORT

;----------------------------------------;
;         Low-voltage detection          ;
;----------------------------------------;
CALL   !!SINITLVI          ; Ensures 2.7 V to VDD

;----------------------------------------;
;       Initialization of clock          ;
;----------------------------------------;
MOV    CMC,   #01000011B    ; Clock Operation Mode Control Register
              ;|||||||+--- : Control of high-speed system clock oscillation frequency
              ;|||||||  : 0 : 2 MHz <= fMX <= 10 MHz
              ;|||||||  : 1 : 10 MHz < fMX <= 20 MHz
              ;|||||||
              ;|||||++---- : XT1 oscillator oscillation mode selection
              ;|||||  : 0 0 : Low-consumption oscillation
              ;|||||  : 0 1 : Normal oscillation
              ;|||||  : 1 x : Super-low-consumption oscillation
              ;|||||   x = don't care
              ;|||||
              ;||||+------ : Be sure to set 0
              ;||||
```

```
                ;|||+------- : [1] Subsystem clock pin operation mode
                ;|||           [2] XT1/P123 pin and XT2/P124 pin
                ;||| : 0 : [1]Input port mode
                ;|||       [2]Input port
                ;|||
                ;||| : 1 : [1]XT1 oscillation mode
                ;|||       [2]Crystal resonator connection
                ;|||
                ;||+-------- : Be sure to set 0
                ;||
                ;++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation
mode
                ;              [2]X1/P121 pin
                ;              [3]X2/EXCLK/P122 pin
                ; : 0 0 : [1]Input port mode
                ;         [2][3]Input port
                ;
                ; : 0 1 : [1]X1 oscillation mode
                ;         [2][3]Crystal/ceramic resonator connection
                ;
                ; : 1 0 : [1]Input port mode
                ;         [2][3]Input port
                ;
                ; : 1 1 : [1]External clock input mode
                ;         [2]Input port
                ;         [3]External clock input

        CLR1    MSTOP               ; X1 oscillator operating

        MOV     OSMC,  #00000001B   ; Operation Speed Mode Control Register
                ;||||||++--- : fCLK frequency selection
                ;|||||| : 0 0 : Operates at a frequency of 10 MHz or less.
                ;|||||| : 0 1 : Operates at a frequency higher than 10 MHz.
                ;|||||| : 1 0 : Operates at a frequency of 1 MHz.
                ;||||||
                ;|++++-----  : Be sure to set 00000
                ;|
                ;+---------- : Setting in subsystem clock HALT mode
                ; : 0 : Enables subsystem clock supply to peripheral functions.
```

```
                ;       (See Table 21-1 Operating Statuses in HALT Mode (2/3)
                ;        for the peripheral functions whose operations are enabled.)
                ; : 1 : Stops subsystem clock supply to peripheral functions
                ;        except real-time counter, clock output/buzzer output,
                ;        and LCD controller/driver.

BF      OSTC.0,       $$      ; X1 oscillation stabilization finished?, No

;*-- Caution ------------------------------------------------*;
;* To increase fCLK to 10 MHz or higher, set FSEL to '1',  *;
;* then change fCLK after two or more clocks have elapsed. *;
;*------------------------------------------------------------*;
NOP
NOP

MOV     CKC,    #00010000B    ; System Clock Control Register
                ;|+|++++++--- : Selection of CPU/peripheral hardware clock (fCLK)
                ;| | : 0 0 x 0 0 0 : fIH
                ;| | : 0 0 x 0 0 1 : fIH/2 (default)
                ;| | : 0 0 x 0 1 0 : fIH/2^2
                ;| | : 0 0 x 0 1 1 : fIH/2^3
                ;| | : 0 0 x 1 0 0 : fIH/2^4
                ;| | : 0 0 x 1 0 1 : fIH/2^5
                ;| | : 0 1 x 0 0 0 : fMX
                ;| | : 0 1 x 0 0 1 : fMX/2
                ;| | : 0 1 x 0 1 0 : fMX/2^2
                ;| | : 0 1 x 0 1 1 : fMX/2^3
                ;| | : 0 1 x 1 0 0 : fMX/2^4
                ;| | : 0 1 x 1 0 1 : fMX/2^5
                ;| | : 1 x 0 x x x : fSUB
                ;| | : 1 x 1 x x x : fSUB/2
                ;| | : Other than above : Setting prohibited
                ;| |  x = don't care
                ;| |
                ;| +-------- : Status of Main system clock (fMAIN)
                ;| : 0 : Internal high-speed oscillation clock (fIH)
                ;| : 1 : High-speed system clock (fMX)
                ;|
                ;+---------- : Status of CPU/peripheral hardware clock (fCLK)
```

```
                ; : 0 : Main system clock (fMAIN)
                ; : 1 : Subsystem clock (fSUB)

HRST100:; CPU is operating on a High-speed system clock?
     BT     CLS,   $HRST100      ;  No
     BF     MCS,   $HRST100      ;  No

     SET1   HIOSTOP              ; Internal high-speed oscillation stopped

     MOV    OSTS,  #00000111B    ; Oscillation Stabilization Time Select Register
                ;||||+++--- : Oscillation stabilization time selection
                ;|||| : 0 0 0 : 2^8/fX
                ;|||| : 0 0 1 : 2^9/fX
                ;|||| : 0 1 0 : 2^10/fX
                ;|||| : 0 1 1 : 2^11/fX
                ;|||| : 1 0 0 : 2^13/fX
                ;|||| : 1 0 1 : 2^15/fX
                ;|||| : 1 1 0 : 2^17/fX
                ;|||| : 1 1 1 : 2^18/fX
                ;||||
                ;++++------- : Be sure to set 0000

     MOV    PER0,  #01100001B    ; Peripheral Enable Register 0
                ;|||||||+--- : Control of timer array unit 0 input clock
                ;||||||| : 0 : Stops input clock supply.
                ;|||||||       * SFR used by timer array unit 0 cannot be written.
                ;|||||||       * Timer array unit 0 is in the reset status.
                ;||||||| : 1 : Supplies input clock.
                ;|||||||       * SFR used by timer array unit 0 can be read and written.
                ;|||||||
                ;||||||+---- : Control of timer array unit 1 input clock
                ;|||||| : 0 : Stops input clock supply.
                ;||||||       * SFR used by timer array unit 1 cannot be written.
                ;||||||       * Timer array unit 1 is in the reset status.
                ;|||||| : 1 : Supplies input clock.
                ;||||||       * SFR used by timer array unit 1 can be read and written.
                ;||||||
                ;|||||+----- : Control of serial array unit 0 input clock
                ;||||| : 0 : Stops input clock supply.
```

```
                    ;|||||       * SFR used by the serial array unit 0 cannot be written.
                    ;|||||       * The serial array unit 0 is in the reset status.
                    ;||||| : 1 : Supplies input clock.
                    ;|||||       * SFR used by the serial array unit 0 can be read and written.
                    ;|||||
                    ;||||+------ : Control of serial array unit 1 input clock
                    ;|||| : 0 : Stops input clock supply.
                    ;||||       * SFR used by the serial array unit 1 cannot be written.
                    ;||||       * The serial array unit 1 is in the reset status.
                    ;|||| : 1 : Supplies input clock.
                    ;||||       * SFR used by the serial array unit 1 can be read and written.
                    ;||||
                    ;|||+------- : Control of serial interface IICA input clock
                    ;||| : 0 : Stops input clock supply.
                    ;|||       * SFR used by the serial interface IICA cannot be written.
                    ;|||       * The serial interface IICA is in the reset status.
                    ;||| : 1 : Supplies input clock.
                    ;|||       * SFR used by the serial interface IICA can be read and written.
                    ;|||
                    ;||+-------- : Control of A/D converter, operational amplifier, and
voltage reference input clock
                    ;|| : 0 : Stops input clock supply.
                    ;||       * SFR used by the A/D converter, operational amplifier, and
voltage reference cannot be written.
                    ;||       * The A/D converter, operational amplifier, and voltage
reference is in the reset status.
                    ;|| : 1 : Supplies input clock.
                    ;||       * SFR used by the A/D converter, operational amplifier, and
voltage reference can be read and written.
                    ;||
                    ;|+--------- : Control of D/A converter input clock
                    ;| : 0 : Stops input clock supply.
                    ;|       * SFR used by D/A converter cannot be written.
                    ;|       * The D/A converter is in the reset status.
                    ;| : 1 : Supplies input clock.
                    ;|       * SFR used by the D/A converter can be read and written.
                    ;|
                    ;+---------- : Control of real-time counter (RTC) input clock
                    ; : 0 : Stops input clock supply.
```

```
                ;       * SFR used by the real-time counter (RTC) cannot be written.
                ;       * The real-time counter (RTC) is in the reset status.
                ; : 1 : Supplies input clock.
                ;       * SFR used by the real-time counter (RTC) can be read and written.

    ;---------------------------------------;
    ;  Initialize of Key Interrupt Function   ;
    ;---------------------------------------;
    CALL   !!SINITKR

    ;---------------------------------------;
    ;       Initialization of timer          ;
    ;---------------------------------------;
    CALL   !!SINITTAU0

    ;---------------------------------------;
    ;  Initialization of voltage reference    ;
    ;---------------------------------------;
    CALL   !!SINITVR

    ;---------------------------------------;
    ; Initialization of Operational amplifier ;
    ;---------------------------------------;
    CALL   !!SINITAMP

    ;---------------------------------------;
    ;   Initialization of D/A Converter      ;
    ;---------------------------------------;
    CALL   !!SINITDAC

    ;===================================;
    ;-----------------------------------;
    ;          Main Loop                ;
    ;-----------------------------------;
    ;===================================;
MMAIN:
    ;*****************************************;
    ;*                                       *;
    ;*    Play melody when the key is input    *;
```

```
      ;*                                          *;
      ;*******************************************;
      ;*-------------------------*;
      ;*    Wait key input      *;
      ;*-------------------------*;
      CLR1   KRMK          ; Clear key interrupt
      CLR1   KRIF          ; Clear key interrupt request flag
      HALT                 ; Sets the HALT mode
      SET1   KRMK          ; Set key interrupt
      CLR1   KRIF          ; Clear key interrupt request flag

      ;*-------------------------*;
      ;* Remove key input noise *;
      ;*-------------------------*;
      BT     P7.0,  $LMAIN200     ; Key input?, No

      SET1   TS0L.0               ; Start TAU0 CH0
      CLR1   TMIF00               ; Clear interrupt request flag
LMAIN100:
      NOP
      BF     TMIF00,$LMAIN100     ; Wait 10 msec

      CLR1   TMIF00               ; Clear interrupt request flag
      SET1   TT0L.0               ; Stop TAU0 CH0

      BT     P7.0,  $LMAIN200     ; Key input?, No
      ;*-------------------------*;
      ;*     Play melody        *;
      ;*-------------------------*;
      CALL   !!SPLAYDAC

LMAIN200:
      ;*******************************************;
      ;*                                         *;
      ;*   The main processing writes here       *;
      ;*             if there is something       *;
      ;*                                         *;
      ;*******************************************;
```

```
     BR     MMAIN               ; Continue main process

;---------------------------------------------------------------;
; Module:   SINITPORT                                           ;
; Description:    Setting of I/O ports                          ;
;    parameter: --                                              ;
;    return   : --                                              ;
;---------------------------------------------------------------;
SINITPORT:
;-------------------------------------------------------
;    Setting of Port 0
;-------------------------------------------------------
     MOV    P0,    #00000000B    ; Set P00 to P02 Output latches to Low
     MOV    PM0,   #11111000B    ; Set P00 to P02 to output port
                                 ; P00 to P02: Unused

;-------------------------------------------------------
;    Setting of Port 1
;-------------------------------------------------------
     MOV    P1,    #00000000B    ; Set P10 to P17 Output latches to Low
     MOV    PM1,   #00000000B    ; Set P10 to P17 to output port
                                 ; P10 to P15: Unused

;-------------------------------------------------------
;    Setting of Port 2
;-------------------------------------------------------
     MOV    P2,    #00000000B    ; Set P20 to P27 Output latches to Low
     MOV    PM2,   #11111111B    ; Set P20 to P27 to input port
                                 ; PM23: filter circuit (AMP1-)
                                 ; PM24: filter circuit (AMP1O)
                                 ; PM25: filter circuit (AMP1+)
                                 ; PM26: filter circuit (AMP2-)
                                 ; PM27: filter circuit (AMP2O)
                                 ; P20 to P22: Unused

;-------------------------------------------------------
;    Setting of Port 3
;-------------------------------------------------------
     MOV    P3,    #00001000B    ; Set P30 to P32, and P34 Output latches to Low
```

```
                                ; Set P33 Output latch High
     MOV    PM3,   #11100000B   ; Set P30 to P34 to output port
                                ; P30 to P34: Unused

;-------------------------------------------------------------
;    Setting of Port 4
;-------------------------------------------------------------
     MOV    P4,    #00000000B   ; Set P40 and P41 Output latches to Low
     MOV    PM4,   #11111100B   ; Set P40 and P41 to output port
                                ; P40 and P41: Unused

;-------------------------------------------------------------
;    Setting of Port 5
;-------------------------------------------------------------
     MOV    P5,    #00000000B   ; Set P50 to P57 Output latches to Low
     MOV    PM5,   #11110000B   ; Set P50 to P57 to output port
                                ; P50 to P57: Unused

;-------------------------------------------------------------
;    Setting of Port 6
;-------------------------------------------------------------
     MOV    P6,    #00000000B   ; Set P60 and P61 Output latches to Low
     MOV    PM6,   #11111100B   ; Set P60 and P61 to output port
                                ; P60 and P61: Unused

;-------------------------------------------------------------
;    Setting of Port 7
;-------------------------------------------------------------
     MOV    P7,    #00000000B   ; Set P70 to P77 Output latches to Low
     MOV    PM7,   #00000001B   ; Set P70 to input port, P71 to P77 to output port
     MOV    PU7,   #00000001B   ; P70 on-chip pull-up resistor connected
                                ; P70: Key input port
                                ; P71 to P77: Unused

;-------------------------------------------------------------
;    Setting of Port 8
;-------------------------------------------------------------
     MOV    P8,    #00000000B   ; Set P80 to P88 Output latches to Low
     MOV    PM8,   #00000000B   ; Set P80 to P88 to output port
```

```
                                ; P80 to P88: Unused

;-------------------------------------------------------
;    Setting of Port 9
;-------------------------------------------------------
     MOV    P9,    #00000000B    ; Set P90 to P97 Output latches to Low
     MOV    PM9,   #00000000B    ; Set P90 to P97 to output port
                                ; P90 to P97: Unused

;-------------------------------------------------------
;    Setting of Port 10
;-------------------------------------------------------
     MOV    P10,   #00000000B    ; Set P100 to P102 Output latches to Low
     MOV    PM10,  #11111000B    ; Set P100 to P102 to output port
                                ; P100 to P102: Unused

;-------------------------------------------------------
;    Setting of Port 11
;-------------------------------------------------------
     MOV    P11,   #00000000B    ; Set P110 and P111 Output latches to Low
     MOV    PM11,  #11111101B    ; Set P110 to input port, P111 to output port
                                ; P110: Play back output (ANO0)
                                ; P111: Unused

;-------------------------------------------------------
;    Setting of Port 12
;-------------------------------------------------------
     MOV    P12,   #00000000B    ; Set P120 Output latch to Low
     MOV    PM12,  #11111110B    ; Set P120 to output port
                                ; P120 to P124: Unused
                                ; *P121 to P124: Input port

;-------------------------------------------------------
;    Setting of Port 13
;-------------------------------------------------------
     MOV    P13,   #00000000B    ; Set P130 Output latch to Low
                                ; P130: Unused

;-------------------------------------------------------
```

```
;    Setting of Port 14
;-------------------------------------------------------------
     MOV    P14,   #00000000B    ; Set P140 to P147 Output latches to Low
     MOV    PM14,  #00000000B    ; Set P140 to P147 to output port
                                 ; P140 to P147: Unused

;-------------------------------------------------------------
;    Setting of Port 15
;-------------------------------------------------------------
     MOV    P15,   #00000000B    ; Set P150 to P152, and P157 Output latches to Low
     MOV    PM15,  #11111111B    ; Set P150 to P152, and P157 to input port
                                 ; PM150: Filter circuit (AMP2+)
                                 ; P151 to P152, and P157: Unused

     RET

;---------------------------------------------------------------------;
; Module:   SINITLVI                                                  ;
; Description:    Ensures 2.7 V to the power supply voltage           ;
;    parameter: --                                                    ;
;    return   : --                                                    ;
;---------------------------------------------------------------------;
SINITLVI:
     ; Setting of Low-Voltage Detector
     SET1   LVIMK               ; Disable LVI interrupt
     CLR1   LVISEL              ; Detects level of VDD
     MOV    LVIS,  #00001001B   ; Low-Voltage Detection Level Select Register
                   ;||||++++--- : Detection level
                   ;|||| : 0 0 0 0 : VLVI0 (4.22 V)
                   ;|||| : 0 0 0 1 : VLVI1 (4.07 V)
                   ;|||| : 0 0 1 0 : VLVI2 (3.92 V)
                   ;|||| : 0 0 1 1 : VLVI3 (3.76 V)
                   ;|||| : 0 1 0 0 : VLVI4 (3.61 V)
                   ;|||| : 0 1 0 1 : VLVI5 (3.45 V)
                   ;|||| : 0 1 1 0 : VLVI6 (3.30 V)
                   ;|||| : 0 1 1 1 : VLVI7 (3.15 V)
                   ;|||| : 1 0 0 0 : VLVI8 (2.99 V)
                   ;|||| : 1 0 0 1 : VLVI9 (2.84 V)
                   ;|||| : 1 0 1 0 : VLVI10 (2.68 V)
```

```
                    ;|||| : 1 0 1 1 : VLVI11 (2.53 V)
                    ;|||| : 1 1 0 0 : VLVI12 (2.38 V)
                    ;|||| : 1 1 0 1 : VLVI13 (2.22 V)
                    ;|||| : 1 1 1 0 : VLVI14 (2.07 V)
                    ;|||| : 1 1 1 1 : VLVI15 (1.91 V)
                    ;||||
                    ;++++------- : Be sure to set 0000
      CLR1   LVIMD              ; Generates an internal interrupt signal when detect the
low-voltage
      SET1   LVION              ; Enables low-voltage detection operation

      ; Software to wait for the operation stabilization time (over 10 us)
      MOV    B,     #10         ; Set counter
 HRES100:
      NOP                       ;                          (1 clk)
      DEC    B                  ;                          (1 clk)
      BNZ    $HRES100           ; Finished waiting?, No    (2 clk/4 clk)

      ; Wait for VDD to become VLVI or more
 HRES300:
      NOP
      BT     LVIF,  $HRES300    ; VDD < VLVI?, Yes
      CLR1   LVION              ; Disables low-voltage detection operation

      RET

 ;---------------------------------------------------------------;
 ; Module:   SINITKR                                             ;
 ; Description:    Setting of Key Interrupt Function             ;
 ;    parameter: --                                              ;
 ;    return   : --                                              ;
 ;---------------------------------------------------------------;
 SINITKR:
      SET1   KRMK               ; Disable key interrupt
      MOV    KRM,   #00000001B  ; Key Return Mode Register
                    ;|||||||+--- : KR0 interrupt mode control
                    ;||||||+---- : KR1 interrupt mode control
                    ;|||||+----- : KR2 interrupt mode control
                    ;||||+------ : KR3 interrupt mode control
```

```
                ;|||+------- : KR4 interrupt mode control
                ;||+-------- : KR5 interrupt mode control
                ;|+--------- : KR6 interrupt mode control
                ;+---------- : KR7 interrupt mode control
                ; : 0 : Does not detect key interrupt signal
                ; : 1 : Detects key interrupt signal
    NOP    ; 250 ns interval from set KRM to clear KRIF
    NOP
    NOP
    NOP
    NOP
    CLR1   KRIF                ; Clear key interrupt request flag

    RET

;---------------------------------------------------------------;
; Module:   SINITTAU0                                           ;
; Description:    Setting of Timer array unit 0                 ;
;    parameter: --                                              ;
;    return   : --                                              ;
;---------------------------------------------------------------;
SINITTAU0:
    MOV    TPS0L, #00000010B   ; Timer Clock Select Register 0
                ;||||++++--- : Selection of operation clock (CK00)
                ;++++------- : Selection of operation clock (CK01)
                ; : 0 0 0 0 : CK0m = fCLK
                ; : 0 0 0 1 : CK0m = fCLK/2
                ; : 0 0 1 0 : CK0m = fCLK/2^2
                ; : 0 0 1 1 : CK0m = fCLK/2^3
                ; : 0 1 0 0 : CK0m = fCLK/2^4
                ; : 0 1 0 1 : CK0m = fCLK/2^5
                ; : 0 1 1 0 : CK0m = fCLK/2^6
                ; : 0 1 1 1 : CK0m = fCLK/2^7
                ; : 1 0 0 0 : CK0m = fCLK/2^8
                ; : 1 0 0 1 : CK0m = fCLK/2^9
                ; : 1 0 1 0 : CK0m = fCLK/2^10
                ; : 1 0 1 1 : CK0m = fCLK/2^11
                ; : 1 1 0 0 : CK0m = fCLK/2^12
                ; : 1 1 0 1 : CK0m = fCLK/2^13
```

```
                ; : 1 1 1 0 : CK0m = fCLK/2^14
                ; : 1 1 1 1 : CK0m = fCLK/2^15
                ;  m = 0, 1


; CH0: for timing
MOVW   AX,    #0000000000000000B  ; Timer Mode Register 00
MOVW   TMR00,AX;||||||||||||++++--- : [1]Operation mode of channel 0
                ;||||||||||||            [2]Count operation of TCR
                ;||||||||||||            [3]Independent operation
                ;||||||||||||            [4]Setting of starting counting and interrupt
                ;||||||||||||| : 0 0 0 0 : [1]Interval timer mode
                ;||||||||||||              [2]Counting down
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is not generated
                ;||||||||||||                 when counting is started
                ;||||||||||||                (timer output does not change, either).
                ;||||||||||||
                ;||||||||||||| : 0 0 0 1 : [1]Interval timer mode
                ;||||||||||||              [2]Counting down
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is generated
                ;||||||||||||                 when counting is started
                ;||||||||||||                (timer output also changes).
                ;||||||||||||
                ;||||||||||||| : 0 1 0 0 : [1]Capture mode
                ;||||||||||||              [2]Counting up
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is not generated
                ;||||||||||||                 when counting is started
                ;||||||||||||                (timer output does not change, either).
                ;||||||||||||
                ;||||||||||||| : 0 1 0 1 : [1]Capture mode
                ;||||||||||||              [2]Counting up
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is generated when
                ;||||||||||||                 counting is started
                ;||||||||||||              (timer output also changes).
                ;||||||||||||
                ;||||||||||||| : 0 1 1 0 : [1]Event counter mode
```

```
;||||||||||||              [2]Counting down
;||||||||||||              [3]Possible
;||||||||||||              [4]Timer interrupt is not generated
;||||||||||||                 when counting is started
;||||||||||||                (timer output does not change, either).
;||||||||||||
;||||||||||||  : 1 0 0 0 : [1]One-count mode
;||||||||||||              [2]Counting down
;||||||||||||              [3]Impossible
;||||||||||||              [4]Start trigger is invalid
;||||||||||||                 during counting operation.
;||||||||||||                 At that time, interrupt
;||||||||||||                 is not generated, either.
;||||||||||||
;||||||||||||  : 1 0 0 1 : [1]One-count mode
;||||||||||||              [2]Counting down
;||||||||||||              [3]Impossible
;||||||||||||              [4]Start trigger is valid
;||||||||||||                 during counting operation.
;||||||||||||                 At that time, interrupt
;||||||||||||                 is also generated.
;||||||||||||
;||||||||||||  : 1 1 0 0 : [1]Capture & one-count mode
;||||||||||||              [2]Counting up
;||||||||||||              [3]Possible
;||||||||||||              [4]Timer interrupt is not generated
;||||||||||||                 when counting is started
;||||||||||||                (timer output does not change, either).
;||||||||||||                 Start trigger is invalid
;||||||||||||                 during counting operation.
;||||||||||||                 At that time interrupt
;||||||||||||                 is not generated, either.
;||||||||||||
;||||||||||||  : Other than above : Setting prohibited
;||||||||||||
;||||||||||++------- : Be sure to set 00
;||||||||||
;||||||||++--------- : Selection of TI00 pin input signal,
;||||||||               fSUB/2, fSUB/4, or INTRTC1 valid edge
```

```
                    ;||||||||              (the timer input used with channel 0
                    ;||||||||               is selected by using TIS0 register).
                    ;||||||||  : 0 0 : Falling edge
                    ;||||||||  : 0 1 : Rising edge
                    ;||||||||  : 1 0 : Both edges (when low-level width is measured)
                    ;||||||||          Start trigger: Falling edge, Capture trigger: Rising
edge
                    ;||||||||  : 1 1 : Both edges (when high-level width is measured)
                    ;||||||||          Start trigger: Rising edge, Capture trigger: Falling
edge
                    ;||||||||
                    ;|||||+++---------- : Setting of start trigger or capture trigger of
channel 0
                    ;|||||  : 0 0 0 : Only software trigger start is valid
                    ;|||||           (other trigger sources are unselected).
                    ;|||||  : 0 0 1 : Valid edge of TI00 pin input signal, fSUB/2, fSUB/4,
                    ;|||||           or INTRTC1 is used as both the start trigger and capture
trigger.
                    ;|||||  : 0 1 0 : Both the edges of TI00 pin input signal, fSUB/2, fSUB/4,
                    ;|||||          or INTRTC1 are used as a start trigger and a capture
trigger.
                    ;|||||  : 1 0 0 : Interrupt signal of the master channel is used
                    ;|||||            (when the channel is used as a slave channel
                    ;|||||             with the combination operation function).
                    ;|||||  : Other than above : Setting prohibited
                    ;|||||
                    ;||||+-------------- : Selection of slave/master of channel 0
                    ;||||  : 0 : Operates as slave channel with combination operation function.
                    ;||||  : 1 : Operates as master channel with combination operation function.
                    ;||||
                    ;|||+---------------- : Selection of count clock (TCLK) of channel 0
                    ;|||  : 0 : Operation clock MCK specified by CKS00 bit
                    ;|||  : 1 : Valid edge of input signal input from TI00 pin, fSUB/2, fSUB/4,
or INTRTC1
                    ;|||      (the timer input used with channel 0 is selected by using TIS0
register).
                    ;|||
                    ;|++----------------- : Be sure to set 00
                    ;|
```

```
                ;+------------------ : Selection of operation clock (MCK) of channel
0
                ; : 0 : Operation clock CK00 set by TPS0 register
                ; : 1 : Operation clock CK01 set by TPS0 register
        ; CK00 = fCLK/2^3 = 5 MHz -> 10 ms = 0.2 [us/clk] * 50000 [count]
        MOVW   TDR00, #(50000 - 1)  ; Set interval time to 10 ms

        SET1   TMMK00               ; Disable interrupt

        ; CH4: For play back sampling timing
        MOVW   AX,    #1000000000000000B  ; Timer Mode Register 04
        MOVW   TMR04,AX;||||||||||||+++--- : [1]Operation mode of channel 4
                ;||||||||||||            [2]Count operation of TCR
                ;||||||||||||            [3]Independent operation
                ;||||||||||||            [4]Setting of starting counting and interrupt
                ;||||||||||||| : 0 0 0 0 : [1]Interval timer mode
                ;||||||||||||              [2]Counting down
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is not generated
                ;||||||||||||                 when counting is started
                ;||||||||||||                (timer output does not change, either).
                ;||||||||||||
                ;||||||||||||| : 0 0 0 1 : [1]Interval timer mode
                ;||||||||||||              [2]Counting down
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is generated
                ;||||||||||||                 when counting is started
                ;||||||||||||                (timer output also changes).
                ;||||||||||||
                ;||||||||||||| : 0 1 0 0 : [1]Capture mode
                ;||||||||||||              [2]Counting up
                ;||||||||||||              [3]Possible
                ;||||||||||||              [4]Timer interrupt is not generated
                ;||||||||||||                 when counting is started
                ;||||||||||||                (timer output does not change, either).
                ;||||||||||||
                ;||||||||||||| : 0 1 0 1 : [1]Capture mode
                ;||||||||||||              [2]Counting up
                ;||||||||||||              [3]Possible
```

```
;||||||||||||             [4]Timer interrupt is generated
;||||||||||||                when counting is started
;||||||||||||               (timer output also changes).
;||||||||||||
;||||||||||||  : 0 1 1 0 : [1]Event counter mode
;||||||||||||             [2]Counting down
;||||||||||||             [3]Possible
;||||||||||||             [4]Timer interrupt is not generated
;||||||||||||                when counting is started
;||||||||||||               (timer output does not change, either).
;||||||||||||
;||||||||||||  : 1 0 0 0 : [1]One-count mode
;||||||||||||             [2]Counting down
;||||||||||||             [3]Impossible
;||||||||||||             [4]Start trigger is invalid
;||||||||||||                during counting operation.
;||||||||||||                At that time, interrupt
;||||||||||||                is not generated, either.
;||||||||||||
;||||||||||||  : 1 0 0 1 : [1]One-count mode
;||||||||||||             [2]Counting down
;||||||||||||             [3]Impossible
;||||||||||||             [4]Start trigger is valid
;||||||||||||                during counting operation.
;||||||||||||                At that time, interrupt
;||||||||||||                is also generated.
;||||||||||||
;||||||||||||  : 1 1 0 0 : [1]Capture & one-count mode
;||||||||||||             [2]Counting up
;||||||||||||             [3]Possible
;||||||||||||             [4]Timer interrupt is not generated
;||||||||||||                when counting is started
;||||||||||||               (timer output does not change, either).
;||||||||||||                Start trigger is invalid
;||||||||||||                during counting operation.
;||||||||||||                At that time interrupt
;||||||||||||                is not generated, either.
;||||||||||||
;||||||||||||  : Other than above : Setting prohibited
```

```
                    ;||||||||||||
                    ;||||||||||++------- : Be sure to set 00
                    ;||||||||||
                    ;||||||||++--------- : Selection of TI04 pin input signal,
                    ;||||||||              fSUB/2, fSUB/4, or INTRTC1 valid edge
                    ;||||||||             (the timer input used with channel 4
                    ;||||||||              is selected by using TIS0 register).
                    ;||||||||| : 0 0 : Falling edge
                    ;||||||||| : 0 1 : Rising edge
                    ;||||||||| : 1 0 : Both edges (when low-level width is measured)
                    ;|||||||||         Start trigger: Falling edge, Capture trigger: Rising
edge
                    ;||||||||| : 1 1 : Both edges (when high-level width is measured)
                    ;|||||||||         Start trigger: Rising edge, Capture trigger: Falling
edge
                    ;||||||||
                    ;|||||+++----------- : Setting of start trigger or capture trigger of
channel 4
                    ;||||| : 0 0 0 : Only software trigger start is valid
                    ;|||||          (other trigger sources are unselected).
                    ;||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4,
                    ;|||||          or INTRTC1 is used as both the start trigger and capture
trigger.
                    ;||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4,
                    ;|||||        or INTRTC1 are used as a start trigger and a capture trigger.
                    ;||||| : 1 0 0 : Interrupt signal of the master channel is used
                    ;|||||           (when the channel is used as a slave channel
                    ;|||||            with the combination operation function).
                    ;||||| : Other than above : Setting prohibited
                    ;|||||
                    ;||||+-------------- : Selection of slave/master of channel 4
                    ;||||: 0 : Operates as slave channel with combination operation function.
                    ;||||: 1 : Operates as master channel with combination operation function.
                    ;||||
                    ;|||+--------------- : Selection of count clock (TCLK) of channel 0
                    ;|||  : 0 : Operation clock MCK specified by CKS04 bit
                    ;||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4,
or INTRTC1
                    ;|||      (the timer input used with channel 4 is selected by using TIS0
```

```
register).
                    ;|||
                    ;|++----------------- : Be sure to set 00
                    ;|
                    ;+------------------- : Selection of operation clock (MCK) of channel
4
                    ; : 0 : Operation clock CK00 set by TPS0 register
                    ; : 1 : Operation clock CK01 set by TPS0 register
      ; CK01 = fCLK/2 = 20 MHz -> 8 kHz (0.125 ms) = 0.05 [us/clk] * 2500 [count]
      MOVW   TDR04, #(2500 - 1)   ; Set interval time to about 125 us (= 8 kHz)

      SET1   TMMK04               ; Disable interrupt

      RET

 ;------------------------------------------------------ ---------;
 ; Module:   SINITVR                                              ;
 ; Description:    Setting of Voltage reference                   ;
 ;    parameter: --                                               ;
 ;    return   : --                                               ;
 ;---------------------------------------------------------------;
 SINITVR:
      MOV    ADVRC, #00001000B    ; Analog reference voltage control register
                    ;||||+|++--- : [1]Positive reference voltage supplies selection of A/D
and D/A converters
                    ;|||| |       [2]Operation control of voltage reference
                    ;|||| |       [3]Output voltage selection of voltage reference
                    ;|||| |       [4]Operation control of input gate voltage boost circuit
for A/D converter
                    ;|||| |       [5]Relationship with the conversion mode used
                    ;|||| |
                    ;|||| | : 0 0 0 : [1]AVREFP (external voltage reference input)
                    ;|||| |           [2]Stops operation (Hi-Z)
                    ;|||| |           [3]2.5 V
                    ;|||| |           [4]Stops operation
                    ;|||| |           [5]Can be set in conversion mode 1
                    ;|||| |
                    ;|||| | : 0 1 0 : [1]AVREFP (external voltage reference input)
                    ;|||| |           [2]Stops operation (Hi-Z)
```

```
                    ;|||| |           [3]2.0 V
                    ;|||| |           [4]Enables operation
                    ;|||| |           [5]Can be set in conversion mode 2 or 3
                    ;|||| |
                    ;|||| | : 1 0 0 : [1]VREFOUT (voltage reference output)
                    ;|||| |           [2]Stops operation (pull-down output)
                    ;|||| |           [3]2.5 V
                    ;|||| |           [4]Stops operation
                    ;|||| |           [5] -
                    ;|||| |
                    ;|||| | : 1 0 1 : [1]VREFOUT (voltage reference output)
                    ;|||| |           [2]Enables operation
                    ;|||| |           [3]2.5 V
                    ;|||| |           [4]Enables operation
                    ;|||| |           [5]Can be set in conversion mode 2 or 3
                    ;|||| |
                    ;|||| | : 1 1 0 : [1]VREFOUT (voltage reference output)
                    ;|||| |           [2]Stops operation (pull-down output)
                    ;|||| |           [3]2.0 V
                    ;|||| |           [4]Enables operation
                    ;|||| |           [5] -
                    ;|||| |
                    ;|||| | : 1 1 1 : [1]VREFOUT (voltage reference output)
                    ;|||| |           [2]Enables operation
                    ;|||| |           [3]2.0 V
                    ;|||| |           [4]Enables operation
                    ;|||| |           [5]Can be set in conversion mode 2 or 3
                    ;|||| |
                    ;|||| | : Other than the above : Setting prohibited
                    ;|||| |
                    ;|+++-+----- : Be sure to set 0000
                    ;|
                    ;+---------- : Reference voltage supply (negative side) of A/D converter
selection
                    ; : 0 : AVSS
                    ; : 1 : AVREFM (external voltage reference input)

      SET1   ADVRC.0      ; Enables operation
      SET1   ADVRC.1      ; Output 2.0 V
```

```
      ; Wait for settling time to 20 ms (over 17 msec)
      MOV    B,     #2            ; Set counter
      SET1   TS0L.0               ; Start TAU0 CH0
 JINIVR100:
      CLR1   TMIF00               ; Clear interrupt request flag
 JINIVR200:
      NOP
      BF     TMIF00,$JINIVR200    ; Wait 10 msec
      DEC    B                    ; 20 msec elapsed?
      BNZ    $JINIVR100           ;  No

      CLR1   TMIF00               ; Clear interrupt request flag
      SET1   TT0L.0               ; Stop TAU0 CH0

      RET

 ;---------------------------------------------------------------;
 ; Module:   SINITAMP                                            ;
 ; Description:    Setting of Operational amplifier              ;
 ;    parameter: --                                              ;
 ;    return   : --                                              ;
 ;---------------------------------------------------------------;
 SINITAMP:
      MOV    ADPC,  #00000000B    ; A/D Port Configuration Register
                    ;|||+++++--- : Analog input (A)/digital I/O (D) switching
                    ;|||  :          +------------------------- ANI15/AVREFM/P157
                    ;|||  :          | +-+-+------------------- ANI10/P152 -
ANI8/AMP2+/P150
                    ;|||  :          | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 -
ANI0/AMP0-/P20
                    ;|||  : 0 0 0 0 0 : A A A A A A A A A A A
                    ;|||  : 0 0 0 0 1 : A A A A A A A A A A D
                    ;|||  : 0 0 0 1 0 : A A A A A A A A A D D
                    ;|||  : 0 0 0 1 1 : A A A A A A A A D D D
                    ;|||  : 0 0 1 0 0 : A A A A A A A D D D D
                    ;|||  : 0 0 1 0 1 : A A A A A A D D D D D
                    ;|||  : 0 0 1 1 0 : A A A A A D D D D D D
                    ;|||  : 0 0 1 1 1 : A A A A D D D D D D D
```

```
                ;||| : 0 1 0 0 0 : A A A A D D D D D D D D
                ;||| : 0 1 0 0 1 : A A A D D D D D D D D D
                ;||| : 0 1 0 1 0 : A A D D D D D D D D D D
                ;||| : 0 1 1 1 1 : A D D D D D D D D D D D
                ;||| : 1 0 0 0 0 : D D D D D D D D D D D D
                ;|||
                ;+++-------- : Be sure to set 000

     CLR1   OAC.1              ; Operational amplifier (AMP1) disable
     CLR1   OAC.2              ; Operational amplifier (AMP2) disable

     RET

;---------------------------------------------------------------;
; Module:   SINITDAC                                            ;
; Description:    Setting of D/A Converter                      ;
;    parameter: --                                              ;
;    return   : --                                              ;
;---------------------------------------------------------------;
SINITDAC:
     MOV    DAM,   #01000101B   ; D/A Converter Mode Register
                ;|||||||+--- : DAMD0 D/A converter operation mode selection
                ;||||||+---- : DAMD1 D/A converter operation mode selection
                ;|||||| : 0 : Normal mode
                ;|||||| : 1 : Real-time output mode
                ;||||||
                ;|||||+----- : DARES0 D/A converter resolution selection
                ;||||+------ : DARES1 D/A converter resolution selection
                ;|||| : 0 : 8-bit
                ;|||| : 1 : 12-bit
                ;||||
                ;|||+------- : D/A conversion operation Control (channel 0)
                ;||+-------- : D/A conversion operation Control (channel 1)
                ;|| : 0 : Stops conversion operation
                ;|| : 1 : Enables conversion operation
                ;||
                ;|+--------- : Positive reference voltage supply selection of D/A
converter
                ;| : 0 : AVDD1 (power supply for D/A converter analog circuit)
```

```
                    ;| : 1 : VREFOUT (voltage reference output)/AVREFP (external voltage
reference input)
                    ;|      (Reference voltage supply negative side is AVSS, positive side
is AVREFP -> AVREFP)
                    ;|      (Reference voltage supply negative side is AVREFM, positive side
is VREFOUT -> VREFOUT)
                    ;|
                    ;+---------- : Be sure to set 0
       MOVW   DACSW0,       #0800H        ; Set initial data

       RET

  ;---------------------------------------------------------------;
  ; Module:   SPLAYDAC                                            ;
  ; Description:    Play PCM data by D/A                          ;
  ;    parameter: --                                              ;
  ;    return   : --                                              ;
  ;---------------------------------------------------------------;
  SPLAYDAC:
       ;---------------------------------;
       ;      Prepare for playing         ;
       ;---------------------------------;
       MOVW   HL,    #LOWW TPLAYDATA     ; Set start playing addr (low 16 bits)
       MOV    ES,    #HIGHW TPLAYDATA    ; Set start playing addr (high 4 bits)

  JPDAC100:
       MOVW   AX,    #LOWW RADPCMWORK
       CALL   !!_adpcm_init       ; ADPCM process Initialization

       ; Operational amplifier setting
       SET1   OAC.1               ; Operational amplifier (AMP1) enable
       SET1   OAC.2               ; Operational amplifier (AMP2) enable
       ; Use software to wait until the operational amplifier stabilizes (20 us (max.))
       MOV    B,     #80
  JPDAC150:
       DEC    B
       BNZ    $JPDAC150           ; 20 us elapsed?, No

       ; D/A converter setting
```

```
      SET1   DACE0               ; D/A converter CH0 enable

      ; TAU0 CH4 setting for output timing
      CLR1   TMIF04              ; Clear interrupt request flag
      SET1   TS0L.4              ; Start TAU0 CH4

      ;*********************************;
      ;*-------------------------------*;
      ;*   Ddecode and play PCM data   *;
      ;*-------------------------------*;
      ;*********************************;
      MOVW   RPLAYCOUNT,#0       ; Clear output data counter

JPDAC200:
      MOVW   AX,    RPLAYCOUNT   ; Get number of output times
      CMPW   AX,    !TPLAYSIZE   ; Finished all data output?
      BNC    $JPDAC300           ;  Yes

      ;***********************;
      ;*   Play low 4 bits   *;
      ;***********************;
      ; Decompression of ADPCM data (low 4 bits -> 16 bits)
      MOVW   AX,    #LOWW RADPCMWORK    ; Set work area for _adpcm_l32_dec
      PUSH   AX
      MOV    A,     ES:[HL]      ; Get compressed data
      AND    A,     #00FH        ; Clear high 4 bits
      MOV    X,     A
      CLRB   A
      CLR1   DIVMODE             ; Set multiplication mode (for _adpcm_l32_dec)
      CALL   !!_adpcm_l32_dec    ; Decompression of PCM data
      POP    AX
      MOVW   AX,    BC           ; Get decompression data

      ; Adjust play data
      ADDW   AX,    #8000H       ; Adjust sign
      SHRW   AX,    (16-12)      ; Right-align data

JPDAC220:
      MOVW   DACSW0,       AX    ; Set play data
```

```
     ; Waiting for the output to be completed
JPDAC230:
     NOP
     BF     TMIF04,       $JPDAC230     ; The output to be completed?, No
     CLR1   TMIF04                      ; Clear interrupt request flag

     ;**********************;
     ;*   Play high 4 bits  *;
     ;**********************;
     ; Decompression of ADPCM data (high 4 bits -> 16 bits)
     MOVW   AX,    #LOWW RADPCMWORK     ; Set work area for _adpcm_l32_dec
     PUSH   AX
     MOV    A,     ES:[HL]              ; Get compressed data
     SHR    A,     4
     MOV    X,     A
     CLRB   A
     CLR1   DIVMODE                     ; Set multiplication mode (for _adpcm_l32_dec)
     CALL   !!_adpcm_l32_dec            ; Decompression of PCM data
     POP    AX                          ; Pop argument
     MOVW   AX,    BC                   ; Get decompression data

     ; Adjust play data
     ADDW   AX,    #8000H               ; Adjust sign
     SHRW   AX,    (16-12)              ; Right-align data

JPDAC270:
     MOVW   DACSW0,       AX            ; Set play data

     ; Waiting for the output to be completed
JPDAC280:
     NOP
     BF     TMIF04,       $JPDAC280     ; The output to be completed?, No
     CLR1   TMIF04                      ; Clear interrupt request flag

     INCW   RPLAYCOUNT                  ; Update play counter
     INCW   HL                          ; Next play data
     BR     JPDAC200
```

```
JPDAC300:
     ;---------------------------------;
     ;        Finish playing           ;
     ;---------------------------------;
     SET1   TT0L.4              ; Stop TAU0 CH4
     CLR1   DACE0               ; D/A converter CH0 disable
     CLR1   OAC.1               ; Operational amplifier (AMP1) disable
     CLR1   OAC.2               ; Operational amplifier (AMP2) disable

JPDAC900:
     RET

end
```

● main.c (C language version)

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*------------------------------------------------------------------------*/
/* #pragma directive for CC78K0                                           */
/*------------------------------------------------------------------------*/
#pragma  SFR
#pragma  DI
#pragma  EI
#pragma  HALT
#pragma  NOP

/*------------------------------------------------------------------------*/
/* Include files                                                          */
/*------------------------------------------------------------------------*/
#include "adpcmsp.h"

/*------------------------------------------------------------------------*/
/* Function prototyps                                                     */
/*------------------------------------------------------------------------*/
static  void   fn_InitPort(void);  /* Setting of I/O ports */
static  void   fn_InitLvi(void);   /* Low-voltage detection */
static  void   fn_InitKr(void);    /* Setting of Key Interrupt Function */
static  void   fn_InitTau0(void);  /* Setting of Timer array unit 0 */
static  void   fn_InitVr(void);    /* Setting of Voltage reference */
static  void   fn_InitAmp(void);   /* Setting of Operational amplifier */
static  void   fn_InitDac(void);   /* Setting of D/A Converter */

/*------------------------------------------------------------------------*/
/* Extern variables/constants                                             */
```

```
/*---------------------------------------------------------------------*/
extern  const unsigned char aPlayData[]; /* Sound data */
extern  unsigned short  ushDataSize;     /* Size of sound data */

/*---------------------------------------------------------------------*/
/* Local constants                                                     */
/*---------------------------------------------------------------------*/
/*---------------------------------------------------------------------*/
/* Global variables                                                    */
/*---------------------------------------------------------------------*/
/*---------------------------------------------------------------------*/
/* Local variables                                                     */
/*---------------------------------------------------------------------*/
/* for play */
unsigned short ushAdpcmWork[16];   /* Work area for ADPCM process */

/*---------------------------------------------------------------------*/
/* Code                                                                */
/*---------------------------------------------------------------------*/
/*---------------------------------------------------------------------*/
/* Hardware initialization                                             */
/*---------------------------------------------------------------------*/
void hdwinit(void)
{
  DI();  /* Disable all interrupts */

  /*-----------------------------------------*/
  /*        Initialization of port           */
  /*-----------------------------------------*/
  fn_InitPort();

  /*-----------------------------------------*/
  /*         Low-voltage detection           */
  /*-----------------------------------------*/
  fn_InitLvi();/* Ensures 2.7 V to VDD */

  /*-----------------------------------------*/
  /*       Initialization of clock           */
  /*-----------------------------------------*/
```

```
CMC =  0b01000011;  /* Clock Operation Mode Control Register */
    /*|||||||+--- : Control of high-speed system clock oscillation frequency */
    /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
    /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
    /*||||||| */
    /*|||||++---- : XT1 oscillator oscillation mode selection */
    /*||||| : 0 0 : Low-consumption oscillation */
    /*||||| : 0 1 : Normal oscillation */
    /*||||| : 1 x : Super-low-consumption oscillation */
    /*|||||  x = don't care */
    /*||||| */
    /*||||+------ : Be sure to set 0 */
    /*|||| */
    /*|||+------- : [1] Subsystem clock pin operation mode */
    /*|||           [2] XT1/P123 pin and XT2/P124 pin */
    /*||| : 0 : [1]Input port mode */
    /*|||       [2]Input port */
    /*||| */
    /*||| : 1 : [1]XT1 oscillation mode */
    /*|||       [2]Crystal resonator connection */
    /*||| */
    /*||+-------- : Be sure to set 0 */
    /*|| */
    /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
    /*              [2]X1/P121 pin */
    /*              [3]X2/EXCLK/P122 pin */
    /* : 0 0 : [1]Input port mode */
    /*         [2][3]Input port */
    /* */
    /* : 0 1 : [1]X1 oscillation mode */
    /*         [2][3]Crystal/ceramic resonator connection */
    /* */
    /* : 1 0 : [1]Input port mode */
    /*         [2][3]Input port */
    /* */
    /* : 1 1 : [1]External clock input mode */
    /*         [2]Input port */
    /*         [3]External clock input */
```

```
MSTOP = 0;/* X1 oscillator operating */

OSMC = 0b00000001;  /* Operation Speed Mode Control Register */
   /*||||||++--- : fCLK frequency selection */
   /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
   /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
   /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
   /*|||||| */
   /*|++++++---- : Be sure to set 00000 */
   /*| */
   /*+---------- : Setting in subsystem clock HALT mode */
   /* : 0 : Enables subsystem clock supply to peripheral functions. */
   /*       (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
   /*        for the peripheral functions whose operations are enabled.) */
   /* : 1 : Stops subsystem clock supply to peripheral functions */
   /*        except real-time counter, clock output/buzzer output, */
   /*        and LCD controller/driver. */

while(OSTC.0 != 1){    /* Wait X1 oscillation stabilization */
  NOP();
}

/*-- Caution ------------------------------------------------*/
/* To increase fCLK to 10 MHz or higher, set FSEL to '1',  */
/* then change fCLK after two or more clocks have elapsed. */
/*-----------------------------------------------------------*/
NOP();
NOP();

CKC =  0b00010000;  /* System Clock Control Register */
   /*|+|++++---- : Selection of CPU/peripheral hardware clock (fCLK) */
   /*| | : 0 0 x 0 0 0 : fIH */
   /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
   /*| | : 0 0 x 0 1 0 : fIH/2^2 */
   /*| | : 0 0 x 0 1 1 : fIH/2^3 */
   /*| | : 0 0 x 1 0 0 : fIH/2^4 */
   /*| | : 0 0 x 1 0 1 : fIH/2^5 */
   /*| | : 0 1 x 0 0 0 : fMX */
   /*| | : 0 1 x 0 0 1 : fMX/2 */
```

```
    /*| | : 0 1 x 0 1 0 : fMX/2^2 */
    /*| | : 0 1 x 0 1 1 : fMX/2^3 */
    /*| | : 0 1 x 1 0 0 : fMX/2^4 */
    /*| | : 0 1 x 1 0 1 : fMX/2^5 */
    /*| | : 1 x 0 x x x : fSUB */
    /*| | : 1 x 1 x x x : fSUB/2 */
    /*| | : Other than above : Setting prohibited */
    /*| |  x = don't care */
    /*| | */
    /*| +-------- : Status of Main system clock (fMAIN) */
    /*| : 0 : Internal high-speed oscillation clock (fIH) */
    /*| : 1 : High-speed system clock (fMX) */
    /*| */
    /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
    /* : 0 : Main system clock (fMAIN) */
    /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
  NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;  /* Internal high-speed oscillation stopped */

OSTS = 0b00000111;   /* Oscillation Stabilization Time Select Register */
    /*||||+++---- : Oscillation stabilization time selection */
    /*|||| : 0 0 0 : 2^8/fX */
    /*|||| : 0 0 1 : 2^9/fX */
    /*|||| : 0 1 0 : 2^10/fX */
    /*|||| : 0 1 1 : 2^11/fX */
    /*|||| : 1 0 0 : 2^13/fX */
    /*|||| : 1 0 1 : 2^15/fX */
    /*|||| : 1 1 0 : 2^17/fX */
    /*|||| : 1 1 1 : 2^18/fX */
    /*|||| */
    /*++++------- : Be sure to set 000000 */

PER0 = 0b01100001;   /* Peripheral Enable Register 0 */
    /*|||||||+--- : Control of timer array unit 0 input clock */
```

```
      /*|||||||| : 0 : Stops input clock supply. */
      /*||||||||       * SFR used by timer array unit 0 cannot be written. */
      /*||||||||       * Timer array unit 0 is in the reset status. */
      /*||||||||  : 1 : Supplies input clock. */
      /*||||||||       * SFR used by timer array unit 0 can be read and written. */
      /*||||||||  */
      /*|||||||+----  : Control of timer array unit 1 input clock */
      /*|||||||  : 0 : Stops input clock supply. */
      /*|||||||       * SFR used by timer array unit 1 cannot be written. */
      /*|||||||       * Timer array unit 1 is in the reset status. */
      /*|||||||  : 1 : Supplies input clock. */
      /*|||||||       * SFR used by timer array unit 1 can be read and written. */
      /*|||||||  */
      /*||||||+-----  : Control of serial array unit 0 input clock */
      /*||||||  : 0 : Stops input clock supply. */
      /*||||||       * SFR used by the serial array unit 0 cannot be written. */
      /*||||||       * The serial array unit 0 is in the reset status. */
      /*||||||  : 1 : Supplies input clock. */
      /*||||||       * SFR used by the serial array unit 0 can be read and written. */
      /*||||||  */
      /*||||+------  : Control of serial array unit 1 input clock */
      /*||||  : 0 : Stops input clock supply. */
      /*||||       * SFR used by the serial array unit 1 cannot be written. */
      /*||||       * The serial array unit 1 is in the reset status. */
      /*||||  : 1 : Supplies input clock. */
      /*||||       * SFR used by the serial array unit 1 can be read and written. */
      /*||||  */
      /*|||+-------  : Control of serial interface IICA input clock */
      /*|||  : 0 : Stops input clock supply. */
      /*|||       * SFR used by the serial interface IICA cannot be written. */
      /*|||       * The serial interface IICA is in the reset status. */
      /*|||  : 1 : Supplies input clock. */
      /*|||       * SFR used by the serial interface IICA can be read and written. */
      /*|||  */
      /*||+--------  : Control of A/D converter, operational amplifier, and voltage reference
input clock */
      /*||  : 0 : Stops input clock supply. */
      /*||       * SFR used by the A/D converter, operational amplifier, and voltage reference
cannot be written. */
```

```
      /*||       * The A/D converter, operational amplifier, and voltage reference is in
the reset status. */
      /*|| : 1 : Supplies input clock. */
      /*||       * SFR used by the A/D converter, operational amplifier, and voltage reference
can be read and written. */
      /*|| */
      /*|+--------- : Control of D/A converter input clock */
      /*| : 0 : Stops input clock supply. */
      /*|       * SFR used by D/A converter cannot be written. */
      /*|       * The D/A converter is in the reset status. */
      /*| : 1 : Supplies input clock. */
      /*|       * SFR used by the D/A converter can be read and written. */
      /*| */
      /*+---------- : Control of real-time counter (RTC) input clock */
      /* : 0 : Stops input clock supply. */
      /*       * SFR used by the real-time counter (RTC) cannot be written. */
      /*       * The real-time counter (RTC) is in the reset status. */
      /* : 1 : Supplies input clock. */
      /*       * SFR used by the real-time counter (RTC) can be read and written. */

   /*----------------------------------------*/
   /*  Initialize of Key Interrupt Function  */
   /*----------------------------------------*/
   fn_InitKr();

   /*----------------------------------------*/
   /*       Initialization of timer        */
   /*----------------------------------------*/
   fn_InitTau0();

   /*----------------------------------------*/
   /*  Initialization of voltage reference   */
   /*----------------------------------------*/
   fn_InitVr();

   /*----------------------------------------*/
   /* Initialization of Operational amplifier */
   /*----------------------------------------*/
   fn_InitAmp();
```

```
  /*---------------------------------------*/
  /*    Initialization of D/A Converter      */
  /*---------------------------------------*/
  fn_InitDac();
}

/*-------------------------------------------------------------------*/
/* Module:  fn_InitPort                                              */
/* Description:Setting of I/O ports                                  */
/*   parameter: --                                                   */
/*   return   : --                                                   */
/*-------------------------------------------------------------------*/
static  void   fn_InitPort(void)
{
/*-----------------------------------------------------------*/
/*   Setting of Port 0                                       */
/*-----------------------------------------------------------*/
  P0 =   0b00000000;  /* Set P00 to P02 Output latches to Low */
  PM0 =  0b11111000;  /* Set P00 to P02 to output port */
          /* P00 to P02: Unused */

/*-----------------------------------------------------------*/
/*   Setting of Port 1                                       */
/*-----------------------------------------------------------*/
  P1 =   0b00000000;  /* Set P10 to P17 Output latches to Low */
  PM1 =  0b00000000;  /* Set P10 to P17 to output port */
          /* P10 to P15: Unused */

/*-----------------------------------------------------------*/
/*   Setting of Port 2                                       */
/*-----------------------------------------------------------*/
  P2 =   0b00000000;  /* Set P20 to P27 Output latches to Low */
  PM2 =  0b11111111;  /* Set P20 to P27 to input port */
          /* PM23: Filter circuit (AMP1-) */
          /* PM24: Filter circuit (AMP1O) */
          /* PM25: Filter circuit (AMP1+) */
          /* PM26: Filter circuit (AMP2-) */
          /* PM27: Filter circuit (AMP2O) */
```

```
          /* P20 to P22: Unused */

/*------------------------------------------------------------*/
/*   Setting of Port 3                                        */
/*------------------------------------------------------------*/
  P3 =   0b00001000;  /* Set P30 to P32, and P34 Output latches to Low */
          /* Set P33 Output latch High */
  PM3 =  0b11100000;  /* Set P30 to P34 to output port */
          /* P30 to P34: Unused */

/*------------------------------------------------------------*/
/*   Setting of Port 4                                        */
/*------------------------------------------------------------*/
  P4 =   0b00000000;  /* Set P40 and P41 Output latches to Low */
  PM4 =  0b11111100;  /* Set P40 and P41 to output port */
          /* P40 and P41: Unused */

/*------------------------------------------------------------*/
/*   Setting of Port 5                                        */
/*------------------------------------------------------------*/
  P5 =   0b00000000;  /* Set P50 to P57 Output latches to Low */
  PM5 =  0b11110000;  /* Set P50 to P57 to output port */
          /* P50 to P57: Unused */

/*------------------------------------------------------------*/
/*   Setting of Port 6                                        */
/*------------------------------------------------------------*/
  P6 =   0b00000000;  /* Set P60 and P61 Output latches to Low */
  PM6 =  0b11111100;  /* Set P60 and P61 to output port */
          /* P60 and P61: Unused */

/*------------------------------------------------------------*/
/*   Setting of Port 7                                        */
/*------------------------------------------------------------*/
  P7 =   0b00000000;  /* Set P70 to P77 Output latches to Low */
  PM7 =  0b00000001;  /* Set P70 to input port, P71 to P77 to output port */
  PU7 =  0b00000001;  /* P70 on-chip pull-up resistor connected */
          /* P70: Key input port */
          /* P71 to P77: Unused */
```

```
/*-------------------------------------------------------------*/
/*   Setting of Port 8                                         */
/*-------------------------------------------------------------*/
  P8 =   0b00000000;  /* Set P80 to P88 Output latches to Low */
  PM8 =  0b00000000;  /* Set P80 to P88 to output port */
          /* P80 to P88: Unused */

/*-------------------------------------------------------------*/
/*   Setting of Port 9                                         */
/*-------------------------------------------------------------*/
  P9 =   0b00000000;  /* Set P90 to P97 Output latches to Low */
  PM9 =  0b00000000;  /* Set P90 to P97 to output port */
          /* P90 to P97: Unused */

/*-------------------------------------------------------------*/
/*   Setting of Port 10                                        */
/*-------------------------------------------------------------*/
  P10 =  0b00000000;  /* Set P100 to P102 Output latches to Low */
  PM10 = 0b11111000;  /* Set P100 to P102 to output port */
          /* P100 to P102: Unused */

/*-------------------------------------------------------------*/
/*   Setting of Port 11                                        */
/*-------------------------------------------------------------*/
  P11 =  0b00000000;  /* Set P110 and P111 Output latches to Low */
  PM11 = 0b11111101;  /* Set P110 to input port, P111 to output port */
          /* P110: Play back output (ANO0) */
          /* P111: Unused */

/*-------------------------------------------------------------*/
/*   Setting of Port 12                                        */
/*-------------------------------------------------------------*/
  P12 =  0b00000000;  /* Set P120 Output latch to Low */
  PM12 = 0b11111110;  /* Set P120 to output port */
          /* P120 to P124: Unused */
          /* *P121 to P124: Input port */

/*-------------------------------------------------------------*/
```

```
/*   Setting of Port 13                                          */
/*---------------------------------------------------------------*/
  P13 =  0b00000000;  /* Set P130 Output latch to Low */
          /* P130: Unused */

/*---------------------------------------------------------------*/
/*   Setting of Port 14                                          */
/*-------------------------------------------------------v------*/
  P14 =  0b00000000;  /* Set P140 to P147 Output latches to Low */
  PM14 = 0b00000000;  /* Set P140 to P147 to output port */
          /* P140 to P147: Unused */

/*---------------------------------------------------------------*/
/*   Setting of Port 15                                          */
/*---------------------------------------------------------------*/
  P15 =  0b00000000;  /* Set P150 to P152, and P157 Output latches to Low */
  PM15 = 0b11111111;  /* Set P150 to P152, and P157 to input port */
          /* PM150: Filter circuit (AMP2+) */
          /* P151 to P152, and P157: Unused */
}

/*---------------------------------------------------------------*/
/* Module:  fn_InitLvi                                           */
/* Description:Ensures 2.7 V to the power supply voltage         */
/*   parameter: --                                               */
/*   return   : --                                               */
/*---------------------------------------------------------------*/
static  void   fn_InitLvi(void)
{
  unsigned char ucCounter;   /* Counter */

  /* Setting of Low-Voltage Detector */
  LVIMK  = 1;   /* Disable LVI interrupt */
  LVISEL = 0;   /* Detects level of VDD */
  LVIS = 0b00001001;  /* Low-Voltage Detection Level Select Register */
    /*||||++++--- : Detection lev l */
    /*|||| : 0 0 0 0 : VLVI0 (4.22 V) */
    /*|||| : 0 0 0 1 : VLVI1 (4.07 V) */
    /*|||| : 0 0 1 0 : VLVI2 (3.92 V) */
```

```
    /*|||| : 0 0 1 1 : VLVI3 (3.76 V) */
    /*|||| : 0 1 0 0 : VLVI4 (3.61 V) */
    /*|||| : 0 1 0 1 : VLVI5 (3.45 V) */
    /*|||| : 0 1 1 0 : VLVI6 (3.30 V) */
    /*|||| : 0 1 1 1 : VLVI7 (3.15 V) */
    /*|||| : 1 0 0 0 : VLVI8 (2.99 V) */
    /*|||| : 1 0 0 1 : VLVI9 (2.84 V) */
    /*|||| : 1 0 1 0 : VLVI10 (2.68 V) */
    /*|||| : 1 0 1 1 : VLVI11 (2.53 V) */
    /*|||| : 1 1 0 0 : VLVI12 (2.38 V) */
    /*|||| : 1 1 0 1 : VLVI13 (2.22 V) */
    /*|||| : 1 1 1 0 : VLVI14 (2.07 V) */
    /*|||| : 1 1 1 1 : VLVI15 (1.91 V) */
    /*|||| */
    /*++++------- : Be sure to set 0000 */
  LVIMD  = 0;   /* Generates an internal interrupt signal when detect the low-voltage */
  LVION  = 1;   /* Enables low-voltage detection operation */

  /* Software to wait for the operation stabilization time (over 10 us) */
  for(ucCounter = 0; ucCounter < 4; ucCounter++){
    NOP();
  }

  /* Wait for VDD to become VLVI or more */
  while(LVIF){
    NOP();
  }
  LVION  = 0;   /* Disables low-voltage detection operation */
}

/*-----------------------------------------------------------------*/
/* Module:  fn_InitKr                                              */
/* Description:Setting of Key Interrupt Function                   */
/*   parameter: --                                                 */
/*   return   : --                                                 */
/*-----------------------------------------------------------------*/
static  void   fn_InitKr(void)
{
  KRMK = 1; /* Disable key interrupt */
```

```
  KRM =  0b00000001;  /* Key Return Mode Register */
      /*|||||||+--- : KR0 interrupt mode control */
      /*||||||+---- : KR1 interrupt mode control */
      /*|||||+----- : KR2 interrupt mode control */
      /*||||+------ : KR3 interrupt mode control */
      /*|||+------- : KR4 interrupt mode control */
      /*||+-------- : KR5 interrupt mode control */
      /*|+--------- : KR6 interrupt mode control */
      /*+---------- : KR7 interrupt mode control */
      /* : 0 : Does not detect key interrupt signal */
      /* : 1 : Detects key interrupt signal */
  NOP(); /* 250 ns interval from set KRM to clear KRIF */
  NOP();
  NOP();
  NOP();
  NOP();
  KRIF = 0; /* Clear key interrupt request flag */
}

/*-------------------------------------------------------------------*/
/* Module:  fn_InitTau0                                              */
/* Description:Setting of Timer array unit 0                         */
/*   parameter: --                                                   */
/*   return   : --                                                   */
/*-------------------------------------------------------------------*/
static  void  fn_InitTau0(void)
{
  TPS0L =   0b00000010;  /* Timer Clock Select Register 0 */
      /*||||++++--- : Selection of operation clock (CK00) */
      /*++++------- : Selection of operation clock (CK01) */
      /* : 0 0 0 0 : CK0m = fCLK */
      /* : 0 0 0 1 : CK0m = fCLK/2 */
      /* : 0 0 1 0 : CK0m = fCLK/2^2 */
      /* : 0 0 1 1 : CK0m = fCLK/2^3 */
      /* : 0 1 0 0 : CK0m = fCLK/2^4 */
      /* : 0 1 0 1 : CK0m = fCLK/2^5 */
      /* : 0 1 1 0 : CK0m = fCLK/2^6 */
      /* : 0 1 1 1 : CK0m = fCLK/2^7 */
      /* : 1 0 0 0 : CK0m = fCLK/2^8 */
```

```
    /* : 1 0 0 1 : CK0m = fCLK/2^9 */
    /* : 1 0 1 0 : CK0m = fCLK/2^10 */
    /* : 1 0 1 1 : CK0m = fCLK/2^11 */
    /* : 1 1 0 0 : CK0m = fCLK/2^12 */
    /* : 1 1 0 1 : CK0m = fCLK/2^13 */
    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
    /*  m = 0, 1 */

/* CH0: For timing */
TMR00 =   0b0000000000000000; /* Timer Mode Register 00 */
    /*||||||||||||++++--- : [1]Operation mode of channel 0 */
    /*||||||||||||          [2]Count operation of TCR */
    /*||||||||||||          [3]Independent operation */
    /*||||||||||||          [4]Setting of starting counting and interrupt */
    /*||||||||||||| : 0 0 0 0 : [1]Interval timer mode */
    /*||||||||||||             [2]Counting down */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is not generated */
    /*||||||||||||                when counting is started */
    /*||||||||||||               (timer output does not change, either). */
    /*||||||||||||| */
    /*||||||||||||| : 0 0 0 1 : [1]Interval timer mode */
    /*||||||||||||             [2]Counting down */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is generated */
    /*||||||||||||                when counting is started */
    /*||||||||||||               (timer output also changes). */
    /*||||||||||||| */
    /*||||||||||||| : 0 1 0 0 : [1]Capture mode */
    /*||||||||||||             [2]Counting up */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is not generated */
    /*||||||||||||                when counting is started */
    /*||||||||||||               (timer output does not change, either). */
    /*||||||||||||| */
    /*||||||||||||| : 0 1 0 1 : [1]Capture mode */
    /*||||||||||||             [2]Counting up */
    /*||||||||||||             [3]Possible */
```

```
/*||||||||||||             [4]Timer interrupt is generated */
/*||||||||||||                when counting is started */
/*||||||||||||               (timer output also changes). */
/*||||||||||||| */
/*||||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated */
/*||||||||||||                when counting is started */
/*||||||||||||               (timer output does not change, either). */
/*||||||||||||| */
/*||||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is invalid */
/*||||||||||||                during counting operation. */
/*||||||||||||                At that time, interrupt */
/*||||||||||||                is not generated, either. */
/*||||||||||||| */
/*||||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is valid */
/*||||||||||||                during counting operation. */
/*||||||||||||                At that time, interrupt */
/*||||||||||||                is also generated. */
/*||||||||||||| */
/*||||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated */
/*||||||||||||                when counting is started */
/*||||||||||||               (timer output does not change, either). */
/*||||||||||||                Start trigger is invalid */
/*||||||||||||                during counting operation. */
/*||||||||||||                At that time interrupt */
/*||||||||||||                is not generated, either. */
/*||||||||||||| */
/*||||||||||||| : Other than above : Setting prohibited */
```

```
      /*||||||||||||| */
      /*||||||||||++------- : Be sure to set 00 */
      /*||||||||||  */
      /*||||||||++--------- : Selection of TI00 pin input signal, */
      /*||||||||              fSUB/2, fSUB/4, or INTRTC1 valid edge */
      /*||||||||              (the timer input used with channel 0 */
      /*||||||||              is selected by using TIS0 register). */
      /*||||||||  : 0 0 : Falling edge */
      /*||||||||  : 0 1 : Rising edge */
      /*||||||||  : 1 0 : Both edges (when low-level width is measured) */
      /*||||||||          Start trigger: Falling edge, Capture trigger: Rising edge */
      /*||||||||  : 1 1 : Both edges (when high-level width is measured) */
      /*||||||||          Start trigger: Rising edge, Capture trigger: Falling edge */
      /*||||||||  */
      /*|||||+++----------- : Setting of start trigger or capture trigger of channel 0 */
      /*|||||  : 0 0 0 : Only software trigger start is valid */
      /*|||||            (other trigger sources are unselected). */
      /*|||||  : 0 0 1 : Valid edge of TI00 pin input signal, fSUB/2, fSUB/4, or */
      /*|||||            INTRTC1 is used as both the start trigger and capture trigger. */
      /*|||||  : 0 1 0 : Both the edges of TI00 pin input signal, fSUB/2, fSUB/4, or */
      /*|||||            INTRTC1 are used as a start trigger and a capture trigger. */
      /*|||||  : 1 0 0 : Interrupt signal of the master channel is used */
      /*|||||            (when the channel is used as a slave channel */
      /*|||||             with the combination operation function). */
      /*|||||  : Other than above : Setting prohibited */
      /*|||||  */
      /*||||+-------------- : Selection of slave/master of channel 0 */
      /*||||  : 0 : Operates as slave channel with combination operation function. */
      /*||||  : 1 : Operates as master channel with combination operation function. */
      /*||||  */
      /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
      /*|||  : 0 : Operation clock MCK specified by CKS00 bit */
      /*|||  : 1 : Valid edge of input signal input from TI00 pin, fSUB/2, fSUB/4, or INTRTC1
*/
      /*|||       (the timer input used with channel 0 is selected by using TIS0 register).
*/
      /*|||  */
      /*|++----------------- : Be sure to set 00 */
      /*|  */
```

```
    /*+------------------ : Selection of operation clock (MCK) of channel 0 */
    /* : 0 : Operation clock CK00 set by TPS0 register */
    /* : 1 : Operation clock CK01 set by TPS0 register */
/* CK00 = fCLK/2^3 = 5 MHz -> 10 ms = 0.2 [us/clk] * 50000 [count] */
TDR00 = (50000 - 1);/* Set interval time to 10 ms */

TMMK00 = 1;  /* Disable interrupt */

/* CH4: For play back sampling timing */
TMR04 = 0b1000000000000000;   /* Timer Mode Register 04 */
    /*||||||||||||++++--- : [1]Operation mode of channel 4 */
    /*||||||||||||          [2]Count operation of TCR */
    /*||||||||||||          [3]Independent operation */
    /*||||||||||||          [4]Setting of starting counting and interrupt */
    /*||||||||||||| : 0 0 0 0 : [1]Interval timer mode */
    /*||||||||||||             [2]Counting down */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is not generated */
    /*||||||||||||                when counting is started */
    /*||||||||||||               (timer output does not change, either). */
    /*||||||||||||| */
    /*||||||||||||| : 0 0 0 1 : [1]Interval timer mode */
    /*||||||||||||             [2]Counting down */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is generated */
    /*||||||||||||                when counting is started */
    /*||||||||||||               (timer output also changes). */
    /*||||||||||||| */
    /*||||||||||||| : 0 1 0 0 : [1]Capture mode */
    /*||||||||||||             [2]Counting up */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is not generated */
    /*||||||||||||                when counting is started */
    /*||||||||||||               (timer output does not change, either). */
    /*||||||||||||| */
    /*||||||||||||| : 0 1 0 1 : [1]Capture mode */
    /*||||||||||||             [2]Counting up */
    /*||||||||||||             [3]Possible */
    /*||||||||||||             [4]Timer interrupt is generated */
```

```
/*||||||||||||                when counting is started */
/*||||||||||||               (timer output also changes). */
/*||||||||||||| */
/*||||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*|||||||||||||             [2]Counting down */
/*|||||||||||||             [3]Possible */
/*|||||||||||||             [4]Timer interrupt is not generated */
/*|||||||||||||                when counting is started */
/*|||||||||||||               (timer output does not change, either). */
/*||||||||||||| */
/*||||||||||||| : 1 0 0 0 : [1]One-count mode */
/*|||||||||||||             [2]Counting down */
/*|||||||||||||             [3]Impossible */
/*|||||||||||||             [4]Start trigger is invalid */
/*|||||||||||||                during counting operation. */
/*|||||||||||||                At that time, interrupt */
/*|||||||||||||                is not generated, either. */
/*||||||||||||| */
/*||||||||||||| : 1 0 0 1 : [1]One-count mode */
/*|||||||||||||             [2]Counting down */
/*|||||||||||||             [3]Impossible */
/*|||||||||||||             [4]Start trigger is valid */
/*|||||||||||||                during counting operation. */
/*|||||||||||||                At that time, interrupt */
/*|||||||||||||                is also generated. */
/*||||||||||||| */
/*||||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*|||||||||||||             [2]Counting up */
/*|||||||||||||             [3]Possible */
/*|||||||||||||             [4]Timer interrupt is not generated */
/*|||||||||||||                when counting is started */
/*|||||||||||||               (timer output does not change, either). */
/*|||||||||||||                Start trigger is invalid */
/*|||||||||||||                during counting operation. */
/*|||||||||||||                At that time interrupt */
/*|||||||||||||                is not generated, either. */
/*||||||||||||| */
/*||||||||||||| : Other than above : Setting prohibited */
/*||||||||||||| */
```

```
      /*||||||||||++------- : Be sure to set 00 */
      /*|||||||||| */
      /*||||||||++--------- : Selection of TI04 pin input signal, */
      /*||||||||             fSUB/2, fSUB/4, or INTRTC1 valid edge */
      /*||||||||             (the timer input used with channel 4 */
      /*||||||||             is selected by using TIS0 register). */
      /*||||||||  : 0 0 : Falling edge */
      /*||||||||  : 0 1 : Rising edge */
      /*||||||||  : 1 0 : Both edges (when low-level width is measured) */
      /*||||||||          Start trigger: Falling edge, Capture trigger: Rising edge */
      /*||||||||  : 1 1 : Both edges (when high-level width is measured) */
      /*||||||||          Start trigger: Rising edge, Capture trigger: Falling edge */
      /*||||||||  */
      /*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
      /*|||||  : 0 0 0 : Only software trigger start is valid */
      /*|||||            (other trigger sources are unselected). */
      /*|||||  : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or */
      /*|||||            INTRTC1 is used as both the start trigger and capture trigger. */
      /*|||||  : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or */
      /*|||||            INTRTC1 are used as a start trigger and a capture trigger. */
      /*|||||  : 1 0 0 : Interrupt signal of the master channel is used */
      /*|||||            (when the channel is used as a slave channel */
      /*|||||             with the combination operation function). */
      /*|||||  : Other than above : Setting prohibited */
      /*|||||  */
      /*||||+-------------- : Selection of slave/master of channel 4 */
      /*||||  : 0 : Operates as slave channel with combination operation function. */
      /*||||  : 1 : Operates as master channel with combination operation function. */
      /*||||  */
      /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
      /*|||  : 0 : Operation clock MCK specified by CKS04 bit */
      /*|||  : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1
*/
      /*|||      (the timer input used with channel 4 is selected by using TIS0 register).
*/
      /*|||  */
      /*|++----------------- : Be sure to set 00 */
      /*|  */
      /*+------------------- : Selection of operation clock (MCK) of channel 4 */
```

```
      /* : 0 : Operation clock CK00 set by TPS0 register */
      /* : 1 : Operation clock CK01 set by TPS0 register */
   /* CK01 = fCLK/2 = 20 MHz -> 8 kHz (0.125 ms) = 0.05 [us/clk] * 2500 [count] */
   TDR04 = (2500 - 1); /* Set interval time to about 125 us (= 8 kHz) */

   TMMK04 = 1;   /* Disable interrupt */
  }

  /*-------------------------------------------------------------*/
  /* Module:  fn_InitVr                                          */
  /* Description:Setting of Voltage reference                    */
  /*   parameter: --                                             */
  /*   return   : --                                             */
  /*-------------------------------------------------------------*/
  static  void   fn_InitVr(void)
  {
    unsigned char work;

   ADVRC = 0b00001000; /* Analog reference voltage control register */
      /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
      /*|||| |        [2]Operation control of voltage reference */
      /*|||| |        [3]Output voltage selection of voltage reference */
      /*|||| |        [4]Operation control of input gate voltage boost circuit for A/D
converter */
      /*|||| |        [5]Relationship with the conversion mode used */
      /*|||| | */
      /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
      /*|||| |           [2]Stops operation (Hi-Z) */
      /*|||| |           [3]2.5 V */
      /*|||| |           [4]Stops operation */
      /*|||| |           [5]Can be set in conversion mode 1 */
      /*|||| | */
      /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
      /*|||| |           [2]Stops operation (Hi-Z) */
      /*|||| |           [3]2.0 V */
      /*|||| |           [4]Enables operation */
      /*|||| |           [5]Can be set in conversion mode 2 or 3 */
      /*|||| | */
```

```
      /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
      /*|||| |           [2]Stops operation (pull-down output) */
      /*|||| |           [3]2.5 V */
      /*|||| |           [4]Stops operation */
      /*|||| |           [5] - */
      /*|||| | */
      /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
      /*|||| |           [2]Enables operation */
      /*|||| |           [3]2.5 V */
      /*|||| |           [4]Enables operation */
      /*|||| |           [5]Can be set in conversion mode 2 or 3 */
      /*|||| | */
      /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
      /*|||| |           [2]Stops operation (pull-down output) */
      /*|||| |           [3]2.0 V */
      /*|||| |           [4]Enables operation */
      /*|||| |           [5] - */
      /*|||| | */
      /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
      /*|||| |           [2]Enables operation */
      /*|||| |           [3]2.0 V */
      /*|||| |           [4]Enables operation */
      /*|||| |           [5]Can be set in conversion mode 2 or 3 */
      /*|||| | */
      /*|||| | : Other than the above : Setting prohibited */
      /*|||| | */
      /*|+++-+----- : Be sure to set 0000 */
      /*| */
      /*+---------- : Reference voltage supply (negative side) of A/D converter selection
*/
      /* : 0 : AVSS */
      /* : 1 : AVREFM (external voltage reference input) */

   ADVRC.0 = 1;  /* Enables operation */
   ADVRC.1 = 1;  /* Output 2.0 V */

   /* Wait for settling time to 20 ms (over 17 msec) */
   TS0L.0 = 1;   /* Start TAU0 CH0 */
   for(work = 2; work > 0; work--){  /* Wait 10 msec*2 */
```

```
    TMIF00 = 0; /* Clear interrupt request flag */
    while(!TMIF00){
      NOP(); /* Wait 10 msec */
    }
  }
  TMIF00 = 0;   /* Clear interrupt request flag */
  TT0L.0 = 1;   /* Stop TAU0 CH0 */
}

/*---------------------------------------------------------------*/
/* Module:  fn_InitAmp                                           */
/* Description:Setting of Operational amplifier                  */
/*   parameter: --                                               */
/*   return   : --                                               */
/*---------------------------------------------------------------*/
static  void   fn_InitAmp(void)
{
  ADPC = 0b00000000;   /* A/D Port Configuration Register */
     /*|||++++---- : Analog input (A)/digital I/O (D) switching */
     /*||| :           +------------------------ ANI15/AVREFM/P157 */
     /*||| :           | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
     /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
     /*||| : 0 0 0 0 0 : A A A A A A A A A A A */
     /*||| : 0 0 0 0 1 : A A A A A A A A A A D */
     /*||| : 0 0 0 1 0 : A A A A A A A A A D D */
     /*||| : 0 0 0 1 1 : A A A A A A A A D D D */
     /*||| : 0 0 1 0 0 : A A A A A A A D D D D */
     /*||| : 0 0 1 0 1 : A A A A A A D D D D D */
     /*||| : 0 0 1 1 0 : A A A A A A D D D D D */
     /*||| : 0 0 1 1 1 : A A A A A D D D D D D */
     /*||| : 0 1 0 0 0 : A A A A D D D D D D D */
     /*||| : 0 1 0 0 1 : A A A D D D D D D D D */
     /*||| : 0 1 0 1 0 : A A D D D D D D D D D */
     /*||| : 0 1 1 1 1 : A D D D D D D D D D D */
     /*||| : 1 0 0 0 0 : D D D D D D D D D D D */
     /*||| */
     /*+++-------- : Be sure to set 000 */

  OAC.1 = 0;/* Operational amplifier (AMP1) disable */
```

```
   OAC.2 = 0;/* Operational amplifier (AMP2) disable */
  }

  /*-----------------------------------------------------------------*/
  /* Module:  fn_InitDac                                             */
  /* Description:Setting of D/A Converter                            */
  /*   parameter: --                                                 */
  /*   return   : --                                                 */
  /*-----------------------------------------------------------------*/
  static  void   fn_InitDac(void)
  {
    DAM =  0b01000101;   /* D/A Converter Mode Register */
        /*|||||||+--- : DAMD0 D/A converter operation mode selection */
        /*||||||+---- : DAMD1 D/A converter operation mode selection */
        /*|||||| : 0 : Normal mode */
        /*|||||| : 1 : Real-time output mode */
        /*|||||| */
        /*|||||+----- : DARES0 D/A converter resolution selection */
        /*||||+------ : DARES1 D/A converter resolution selection */
        /*|||| : 0 : 8-bit */
        /*|||| : 1 : 12-bit */
        /*|||| */
        /*|||+------- : D/A conversion operation Control (channel 0) */
        /*||+-------- : D/A conversion operation Control (channel 1) */
        /*|| : 0 : Stops conversion operation */
        /*|| : 1 : Enables conversion operation */
        /*|| */
        /*|+--------- : Positive reference voltage supply selection of D/A converter */
        /*| : 0 : AVDD1 (power supply for D/A converter analog circuit) */
        /*| : 1 : VREFOUT (voltage reference output)/AVREFP (external voltage reference input)
*/
        /*|     (Reference voltage supply negative side is AVSS, positive side is AVREFP ->
AVREFP) */
        /*|     (Reference voltage supply negative side is AVREFM, positive side is VREFOUT
-> VREFOUT) */
        /*| */
        /*+---------- : Be sure to set 0 */
    DACSW0 = 0x0800; /* Set initial data */
  }
```

```
/*-----------------------------------------------------------------*/
/* Module:  fn_PlayDac                                             */
/* Description:Play PCM data by D/A                                */
/*   parameter: --                                                 */
/*   return   : --                                                 */
/*-----------------------------------------------------------------*/
static  void   fn_PlayDac(void)
{
  unsigned char *  pucPlayData;  /* Start playing addr */
  unsigned short   ushPlayCount; /* Output data counter */
  unsigned short   ushData;  /* Decompression data */
  unsigned short   loop;  /* Waiting counter */

  /*---------------------------------*/
  /*      Prepare for playing        */
  /*---------------------------------*/
  /* set play data addr and size */
  pucPlayData = aPlayData;   /* Set start playing address */

  adpcm_init(ushAdpcmWork);  /* ADPCM process Initialization */

  /* Operational amplifier setting */
  OAC.1 = 1;/* Operational amplifier (AMP1) enable */
  OAC.2 = 1;/* Operational amplifier (AMP2) enable */
  /* Use software to wait until the operational amplifier stabilizes (20 us (max.)) */
  for(loop = 40; loop > 0; loop--){
    NOP();
  }

  /* D/A converter setting */
  DACE0 = 1;/* D/A converter CH0 enable */

  /* TAU0 CH4 setting for output timing */
  TMIF04 = 0;   /* Clear interrupt request flag */
  TS0L.4 = 1;   /* Start TAU0 CH4 */

  /*********************************/
  /*-------------------------------*/
```

```
    /*    Decode and play PCM data    */
    /*--------------------------------*/
    /**********************************/
    for(ushPlayCount = 0; ushPlayCount < ushDataSize; ushPlayCount++){
      /**********************/
      /*   Play low 4 bits   */
      /**********************/
      /* Decompression of ADPCM data (low 4 bits -> 16 bits)*/
      DIVMODE = 0;   /* Set multiplication mode (for _adpcm_l32_dec) */
      ushData   =   (unsigned   short)adpcm_l32_dec(pucPlayData[ushPlayCount]   &   0x0f,
ushAdpcmWork);

      /* Adjust sign & right-align data */
      ushData = (unsigned short)((ushData + 0x8000) >> (16-12));

      /* Set play data */
      DACSW0 = ushData;

      /* Waiting for the output to be completed */
      while(!TMIF04){
        NOP();
      }
      TMIF04 = 0; /* Clear interrupt request flag */

      /**********************/
      /*   Play high 4 bits  */
      /**********************/
      /* Decompression of ADPCM data (high 4 bits -> 16 bits) */
      DIVMODE = 0;   /* Set multiplication mode (for _adpcm_l32_dec) */
      ushData = (unsigned short)adpcm_l32_dec((pucPlayData[ushPlayCount] >> 4) & 0x0f,
ushAdpcmWork);

      /* Adjust sign & right-align data */
      ushData = (unsigned short)((ushData + 0x8000) >> (16-12));

      /* Set play data */
      DACSW0 = ushData;

      /* Waiting for the output to be completed */
```

```
    while(!TMIF04){
      NOP();
    }
    TMIF04 = 0; /* Clear interrupt request flag */
  }

  /*---------------------------------*/
  /*       Finish playing            */
  /*---------------------------------*/
  TT0L.4 = 1;   /* Stop TAU0 CH4 */
  DACE0 = 0;    /* D/A converter CH0 disable */
  OAC.1 = 0;    /* Operational amplifier (AMP1) disable */
  OAC.2 = 0;    /* Operational amplifier (AMP2) disable */

}

/*------------------------------------------------------------------*/
/* Module:  Main                                                    */
/* Description: Main process                                        */
/*   parameter: --                                                  */
/*   return   : --                                                  */
/*------------------------------------------------------------------*/
void main(void)
{
  /*==================================*/
  /*----------------------------------*/
  /*          Main Loop               */
  /*----------------------------------*/
  /*==================================*/
  while(1){
    /*****************************************/
    /*                                       */
    /*    Play melody when the key is input    */
    /*                                       */
    /*****************************************/
    /*-------------------------*/
    /*    Wait key input     */
    /*-------------------------*/
    KRMK = 0;   /* Clear key interrupt */
```

```
    KRIF = 0;   /* Clear key interrupt request flag */
    HALT();     /* Sets the HALT mode */
    KRMK = 1;   /* Set key interrupt */
    KRIF = 0;   /* Clear key interrupt request flag */

    /*-------------------------*/
    /* Remove key input noise */
    /*-------------------------*/
    if(P7.0 == 0){        /* Key input? */
      TS0L.0 = 1;  /* Start TAU0 CH0 */
      TMIF00 = 0;
      while(!TMIF00){
        NOP();  /* Wait 10 msec */
      }
      TMIF00 = 0;
      TT0L.0 = 1;  /* Stop TAU0 CH0 */

      if(P7.0 == 0){          /* key input? */
        /*-------------------------*/
        /*     Play melody        */
        /*-------------------------*/
          fn_PlayDac();
      }
    }

    /*****************************************/
    /*                                       */
    /*    The main processing writes here    */
    /*              if there is something    */
    /*                                       */
    /*****************************************/

  }
}
```

APPENDIX B REVISION HISTORY

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

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan 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