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.

RENESAS

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/Kx3

Sample Program (I^2S Bus Interface)

Interface Between Audio Codec and I^2S Bus

This document summarizes the operations of the sample program and describes how to use the sample program and how to set and use the I^2S bus interface in order to use an audio codec connected to the microcontroller. This sample program is used for transferring audio data between the microcontroller and audio codec as well as recording and playing back voices input from a microphone via the I^2S bus interface. The I^2C bus interface is used for setting the registers in the audio codec while the SPI interface is used for writing data to or reading data from EEPROM.

Target devices

78K0R/KE3 microcontroller
78K0R/KF3 microcontroller
78K0R/KG3 microcontroller
78K0R/KH3 microcontroller
78K0R/KJ3 microcontroller

Document No. U19514EJ1V0AN00 (1st edition)
Date Published February 2009 N

CONTENTS

CHAPTER 1 OVERVIEW

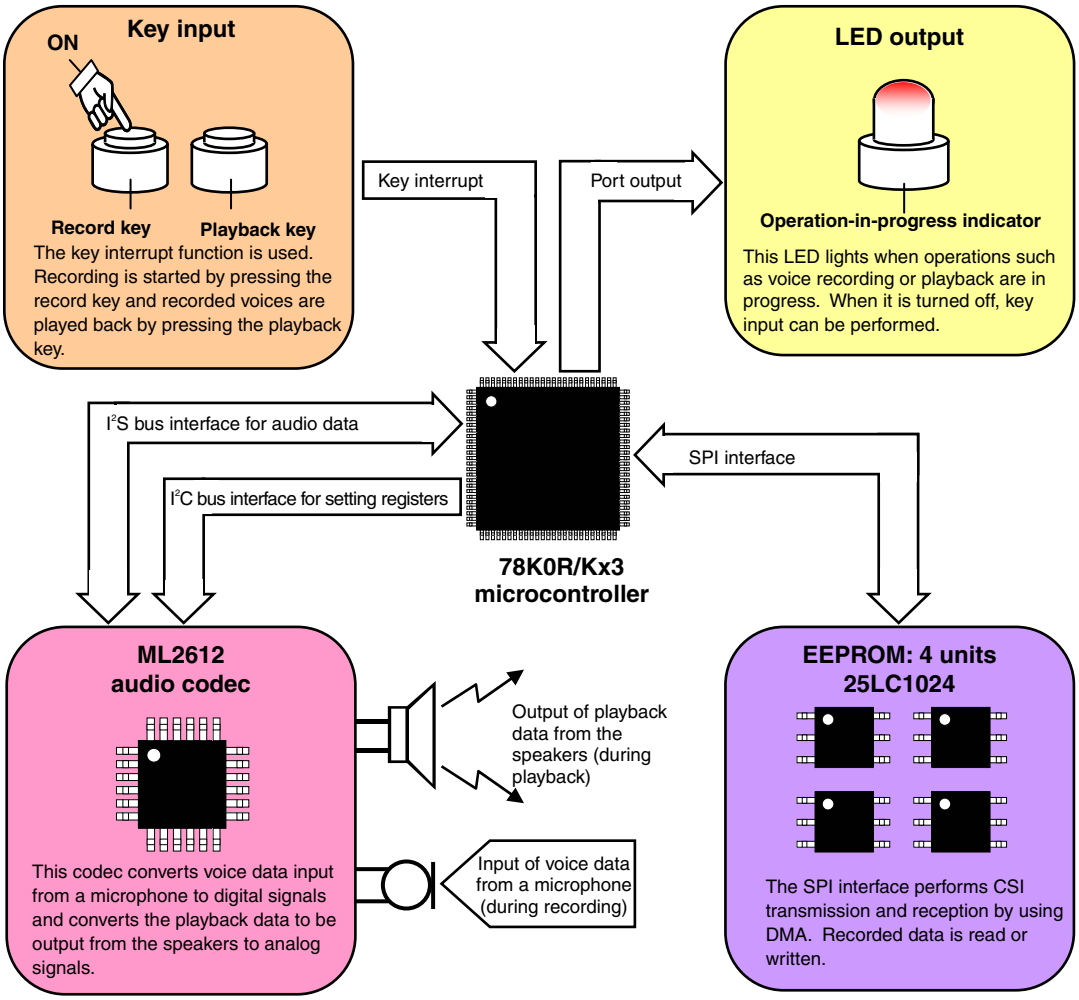
This sample program uses the I²S bus interface to transfer audio data between the microcontroller and an audio codec. The LR clock (LRCLK) for selecting whether to use channel L or channel R of the I²S bus interface is used in the interval timer mode of the timer array unit (TAU) and is output from TO00. CSI00 of channel 0 of serial array unit 0 (SAU0) is used for outputting the clock used for transferring data (BCLK) and for receiving data (SDOUT), and transmitting data (SDIN).

ML2612 made by OKI Semiconductor is used as the audio codec. The I²C bus interface is used for setting the registers in the audio codec and the I²S bus interface is used for transferring audio data. When the record key is pressed, the audio codec receives the voice data input from a microphone and saves the data to EEPROM. When the playback key is pressed, the recorded data is transmitted to the audio codec and the output as sound from the speakers. The SPI interface between the microcontroller and EEPROM uses CSI10 of channel 2 of serial array unit 0 (SAU0) to continuously perform CSI transmission or reception by using the DMA controller. Whether recording or playback is in progress is indicated by an LED (operation-in-progress indicator) to which a signal is output from P72.

In this sample program, voices are recorded and played back by using an audio codec IC applied to the I²S bus interface. Note, however, that voices can also be recorded and played back solely by the microcontroller by using the ADPCM-SP library.

[Operational overview]

Sample program operations are summarized below.



The sample program processing is summarized below.

(1) Main initial settings for peripheral hardware

The main initial settings for the peripheral hardware to be used are described below.

- Disabling interrupts
- Setting the CPU or peripheral hardware clock frequency to the X1 oscillation clock (when used at 20 MHz)
- Setting ports
- Setting the audio codec
  - Supplying a clock to the audio codec
  - Setting a timer for inserting waits in the program
  - Outputting a reset signal to the audio codec
  - Setting the I^2C interface used for setting registers
  - Turning on the system by setting the registers in the audio codec
- Setting the I^2S bus interface used for the audio data of the audio codec
  - Using TO00 output (16 kHz) for setting the output of LRCLK
  - Setting CSI00
- Setting the SPI interface of EEPROM
  - Setting CSI10 for transmitting and receiving data
  - Setting DMA0 for reception and DMA1 for transmission
  - Erasing all EEPROMs (all 0FFH)
- Starting key data retrieval
- Enabling interrupts

(2) Main processing

The main processing is described below.

- Key processing
- Setting start of recording
- Setting start of playback
- Setting termination of recording
- Setting termination of playback

(3) Main INTTM00 interrupt servicing (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and SDIN)

In INTTM00 interrupt servicing, the I^2S bus interface is started to synchronize LRCLK, BCLK, SDOUT, and SDIN.

(4) Main INTCSI00 interrupt servicing (using INTCSI00 for CSI transmission or reception)

The main INTCSI00 interrupt servicing is described below.

- Transmitting or receiving data via the I^2S bus interface
- Saving the received data
- Starting writing to EEPROM
- Starting reading from EEPROM

(5) Main INTCSI10 interrupt servicing (using INTCSI10 for CSI transmission or reception)

The main INTCSI10 interrupt servicing is described below.

- Transmitting instruction bytes to EEPROM
- Transmitting 24-bit addresses to EEPROM
- Starting receiving data from EEPROM by using DMA0 or DMA1
- Starting transmitting data to EEPROM by using DMA1

(6) Main INTDMA0 interrupt servicing (using INTDMA0 for CSI reception)

The main INTDMA0 interrupt servicing is described below.

- Starting reading from EEPROM
- Starting the I^2S bus interface
- Selecting EEPROM
- Stopping the I^2S bus interface

(7) Main INTDMA1 interrupt servicing (using INTDMA1 for CSI transmission)

The main INTDMA1 interrupt servicing is described below.

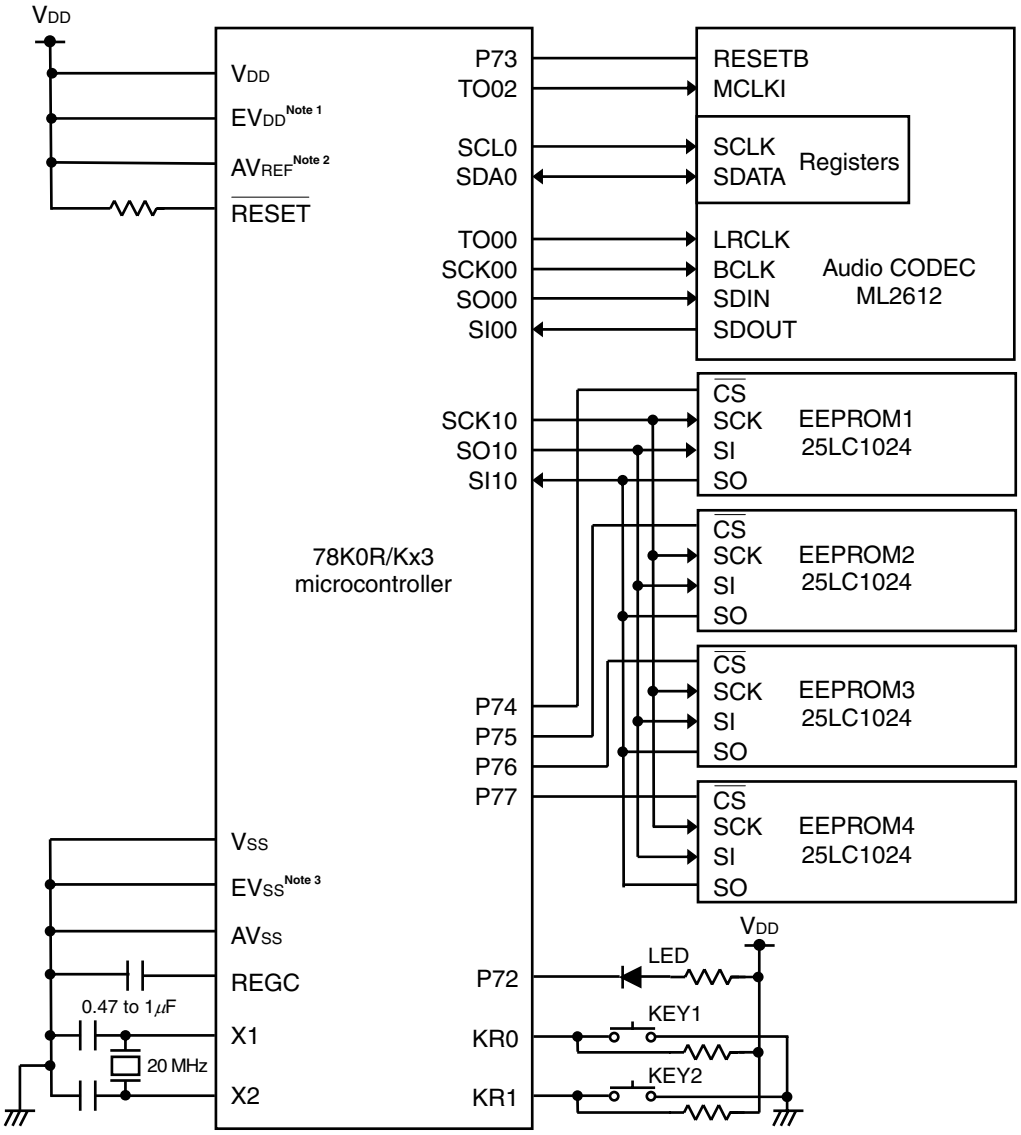
- Selecting EEPROM
- Stopping the I^2S bus interface

CHAPTER 2 CIRCUIT DIAGRAM

This chapter describes the circuit diagram when using this sample program as well as the peripheral hardware.

2.1 Circuit Diagram

The circuit diagram is shown below.



Notes 1. This is EV_{DD0} and EV_{DD1} in the 78K0R/KG3, 78K0R/KH3, and 78K0R/KJ3 microcontrollers.

2. This is AV_{REF0} and AV_{REF1} in the 78K0R/KF3, 78K0R/KG3, 78K0R/KH3, and 78K0R/KJ3 microcontrollers.

3. This is EV_{SS0} and EV_{SS1} in the 78K0R/KG3, 78K0R/KH3, and 78K0R/KJ3 microcontrollers.

Cautions 1. Use the microcontroller within a voltage range of 2.7 V ≤ V_{DD} ≤ 3.6 V.

2. Make the potential of the AV_{SS} pin the same as that of EV_{SS} and V_{SS}, and directly connect the pin to GND.

3. Make the potential of EV_{DD} the same as that of V_{DD}.

4. Connect REGC to V_{SS} via a capacitor (0.47 to 1 μF).

5. Leave all unused port pins other than those shown in the circuit open, because they function as output ports.

2.2 Peripheral Hardware

The following peripheral hardware is used.

(1) ML2612 audio codec made by OKI Semiconductor

This audio codec is used for inputting voice data input from a microphone and outputting voice data to be output from the speakers.

(2) 25LC1024 EEPROMs (EEPROM1, EEPROM2, EEPROM3, EEPROM4)

These EEPROMs are used for saving voice data.

(3) Keys (key 1, key 2)

The keys are used as inputs for controlling the starting of recording and playback.

(4) LED

An LED is used to indicate the status when key input is disabled during recording and playback.

The following pin functions are used.

Function of Pin When Connected to External Device		Shared Pin
Name	Function	
P73	Resets the ML2612 audio codec.	KR3
TO02	Supplies the master clock to the ML2612 audio codec.	P17/TI02
SCL0	Outputs the I^2C bus interface clock to the registers of the ML2612 audio codec.	P60
SDA0	Outputs or inputs data for the I^2C bus interface from or to the registers of the ML2612 audio codec.	P61
TO00	Outputs the LR clock for the I^2S bus interface of the ML2612 audio codec.	P01
SCK00	Outputs the bit clock for the I^2S bus interface of the ML2612 audio codec.	P10/SCL10
SI00	Inputs the data for the I^2S bus interface of the ML2612 audio codec.	P11/RxD0
SO00	Outputs the data for the I^2S bus interface of the ML2612 audio codec.	P12/TxD0
P74 to P77	Output a chip select signal from the 25LC1024 EEPROMs. (4 EEPROMs are used.)	P74 to P77
SCK10	Outputs the clock for the SPI interface from the 25LC1024 EEPROMs.	P04/SCL10
SI10	Inputs the data for the SPI interface to the 25LC1024 EEPROMs.	P03/RxD1/SDA10
SO10	Outputs the data for the SPI interface to the 25LC1024 EEPROMs.	P02/TxD1
P72	Outputs the data for indicating the operating status to the LED.	KR2
KR0	Functions as input from the playback key.	P70
KR1	Functions as input from the record key.	P71

CHAPTER 3 SOFTWARE

This chapter describes the structure of the compressed files to be downloaded, the peripheral functions of the microcontroller to be used, the initial settings and an operational overview of the peripheral hardware to be used in the sample program, and flowcharts.

3.1 File Structure

The structure of the compressed files to be downloaded is described below.

File Name	Description	Compressed Files (zip) Included	
		ZIP	PM 32
main.asm (assembly language version) main.c (C language version)	Source file for the hardware initialization processing, main processing, and interrupt servicing of the microcontroller	●Note	●Note
I2sApplication.prw	Workspace file for the integrated development environment PM+		●
I2sApplication.prj	Project file for the integrated development environment PM+		●

Note The assembly language version includes main.asm and the C language version includes main.c.

Remark ZIP : Only the source file is included.

PM 32 : The files to be used in the integrated development environment PM+ are included.

3.2 On-Chip Peripheral Functions to Be Used

In this sample program, the following peripheral functions incorporated in the microcontroller are used.

- TO00 output of channel 0 of timer array unit 0 (TAU0):
  This output is used at 16 kHz for outputting the LR clock for the I^2S bus interface from the ML2612 audio codec.

- Interval timer of channel 1 of timer array unit 0 (TAU0):
  This timer is used in the 5 ms interval timer mode for creating wait time in the program.

- TO02 output of channel 2 of timer array unit 0 (TAU0):
  This output is used at 10 MHz for supplying the master clock to the ML2612 audio codec.

- CSI00 of channel 0 of serial array unit 0 (SAU0)
  This is used at an 833 kHz transfer speed for inputting or outputting the bit data for the I^2S bus interface to or from the ML2612 audio codec.

- CSI10 of channel 2 of serial array unit 0 (SAU0)
  This is used at a 2.5 MHz transfer speed for inputting or outputting the data for the SPI interface to or from the 25LC1024 EEPROMs.

- Channels 0 (DMA0) and 1 (DMA1) of DMA controller:
  These channels are used for transmission and reception via CSI10, which is used as the SPI interface for the 25LC1024 EEPROMs.

- Serial interface IIC0:
  This interface is used for setting the registers of the ML2612 audio codec. It is used at a 208 kHz transfer speed for inputting and outputting data via the I^2C bus interface.

- Key interrupts:
  Key interrupts are used as interrupts for inputs from the record and playback start keys.

- Pin functions:
  The following pin functions are used.

Function of Pin When Connected to External Device		Shared Pin
Name	Function	
P73	Resets the ML2612 audio codec.	KR3
TO02	Supplies the master clock to the ML2612 audio codec.	P17/TI02
SCL0	Outputs the I^2C bus interface clock to the registers of the ML2612 audio codec.	P60
SDA0	Outputs or inputs data for the I^2C bus interface from or to the registers of the ML2612 audio codec.	P61
TO00	Outputs the LR clock for the I^2S bus interface of the ML2612 audio codec.	P01
SCK00	Outputs the bit clock for the I^2S bus interface of the ML2612 audio codec.	P10/SCL10
SI00	Inputs the data for the I^2S bus interface of the ML2612 audio codec.	P11/RxD0
SO00	Outputs the data for the I^2S bus interface of the ML2612 audio codec.	P12/TxD0
P74 to P77	Output a chip select signal from the 25LC1024 EEPROMs. (4 EEPROMs are used.)	P74 to P77
SCK10	Outputs the clock for the SPI interface from the 25LC1024 EEPROMs.	P04/SCL10
SI10	Inputs the data for the SPI interface to the 25LC1024 EEPROMs.	P03/RxD1/SDA10
SO10	Outputs the data for the SPI interface to the 25LC1024 EEPROMs.	P02/TxD1
P72	Outputs the data for indicating the operating status to the LED.	KR2
KR0	Functions as input from the playback key.	P70
KR1	Functions as input from the record key.	P71

3.3 Initial Settings and Operational Overview of Peripheral Hardware to Be Used

In this sample program, the clock frequency is selected, the timer array unit and serial array unit to be used for the I^2S bus interface are set, serial interface IIC0 to be used for the I^2C bus interface is set, the serial array unit used for the SPI interface is set, the timer array unit used for wait operations in the program is set, and the key interrupts to be used as record and playback key inputs are set as part of the initial settings of the peripheral hardware to be used.

When the initial settings of the peripheral hardware to be used are complete, the processing shifts to the main processing.

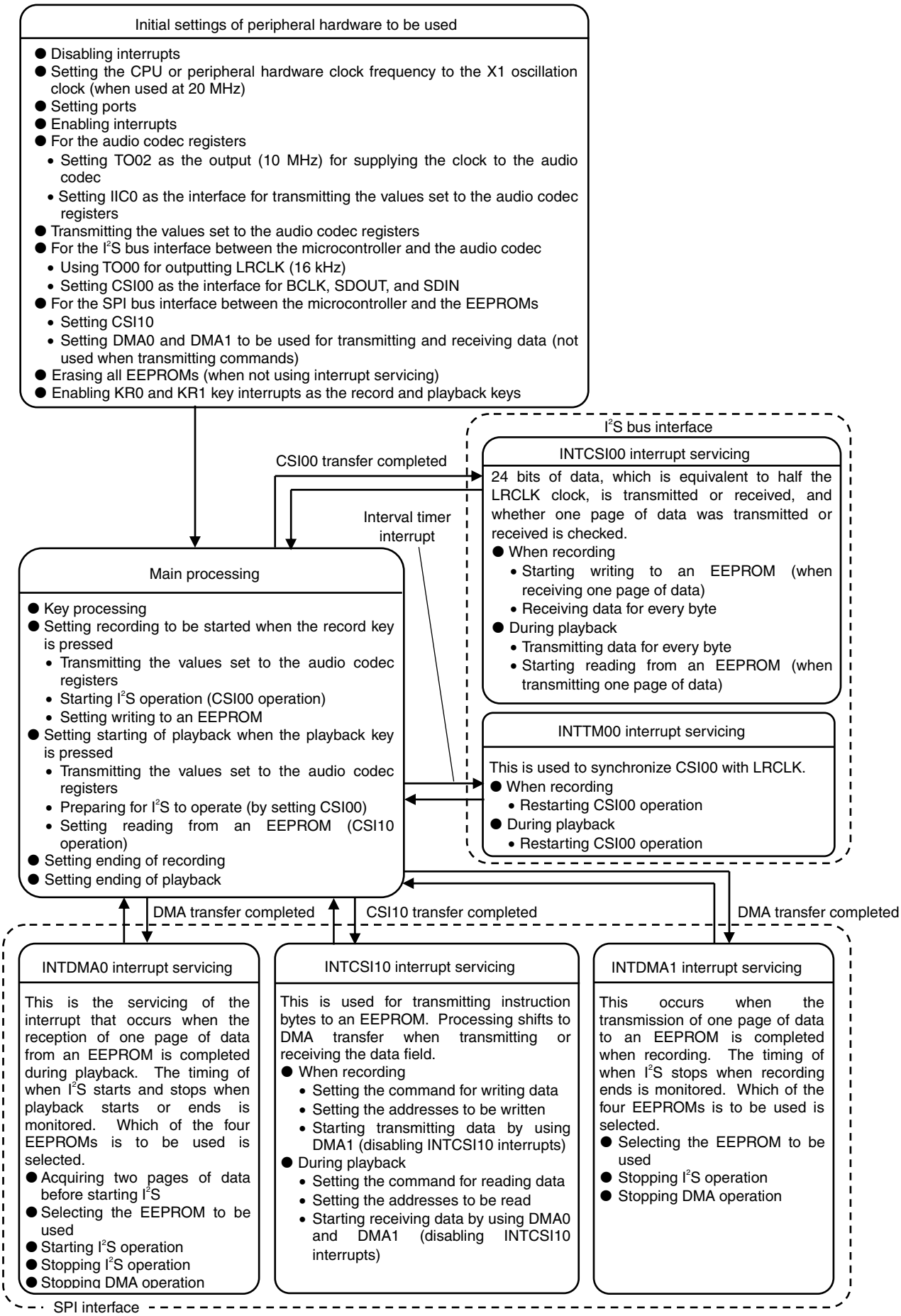
If there is an input from the record key, recording will start. Before starting recording, the audio codec registers must be set via the I^2C interface. The voices input from a microphone can then be received from the audio codec via the I^2S bus interface as audio data. Every time data equivalent to one page[Note] is received, that data is transmitted to an EEPROM via the SPI interface. When the four EEPROMs are filled with data transmitted to EEPROM, transmission ends and the audio codec registers are set via the I^2C interface to stop recording using the audio codec.

If there is an input from the playback key, playing back the recorded data is started. Before starting playback, the audio codec registers must be set via the I^2C interface. Recorded data equivalent to two pages is received from an EEPROM via the SPI interface, which is transmitted as audio data to the audio codec via the I^2S bus interface, and then output from the speakers as playback data. Every time data equivalent to one page[Note] is transmitted, playback data is received from an EEPROM via the SPI interface. Once all the data in the four EEPROMs is received, reception ends and the audio codec registers are set via the I^2C interface to stop playback using the audio codec.

The operation-in-progress indicator (LED) connected to P72 is lit to indicate that the microcontroller is operating during recording or playback.

See the status transition diagram shown on the next page for details.

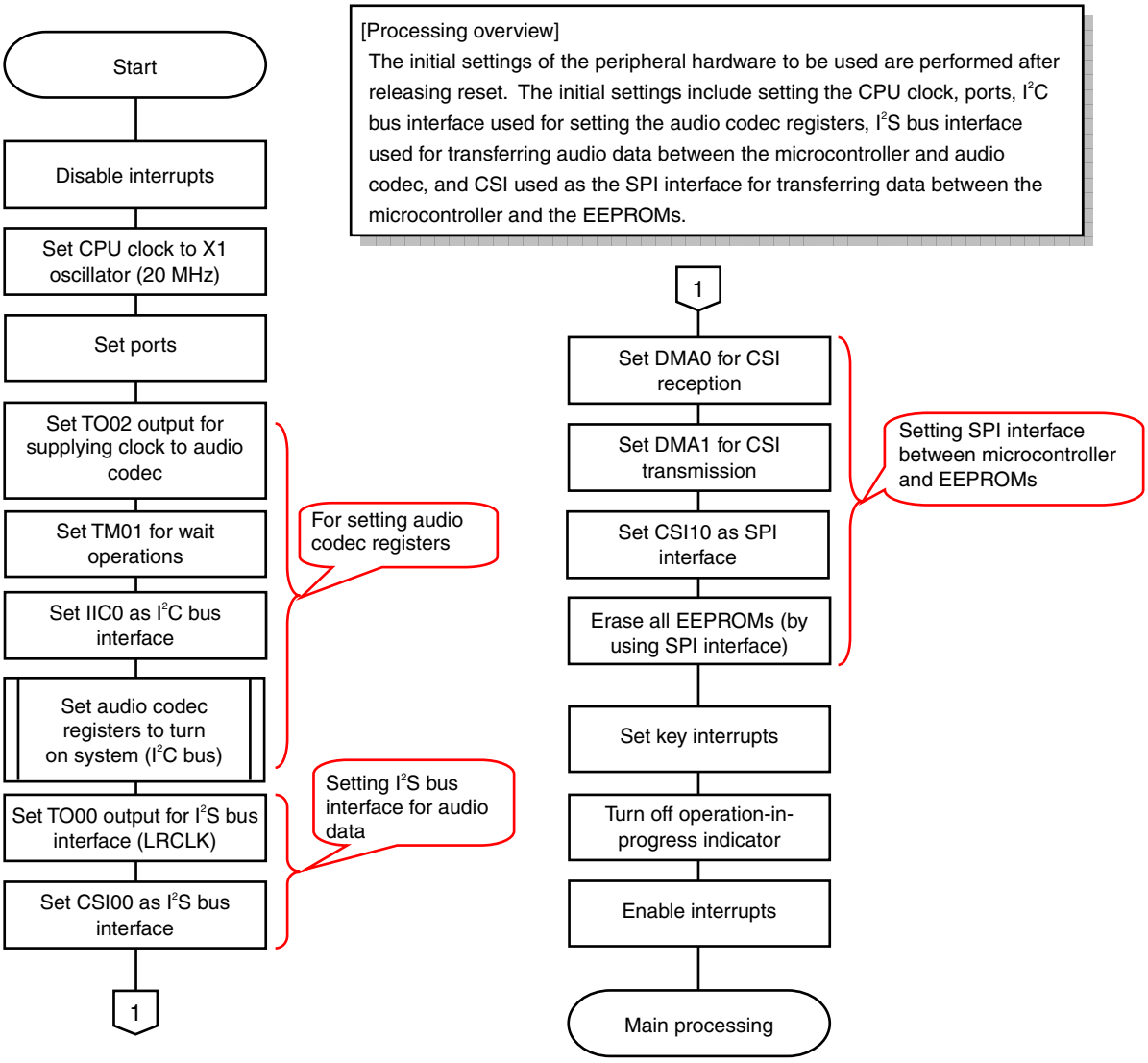
Note A total of six bytes of audio data is transmitted or received in one LR clock, because 24 bits of audio data is included in the range of half the LR clock. The size of one EEPROM page is 256 bytes; but in this sample program, 252 bytes, which is audio data of 42 LR clocks, is assumed to be one page.


Initial settings of peripheral hardware to be used
● Disabling interrupts
● Setting the CPU or peripheral hardware clock frequency to the X1 oscillation clock (when used at 20 MHz)
● Setting ports
● Enabling interrupts
● For the audio codec registers
• Setting TO02 as the output (10 MHz) for supplying the clock to the audio codec
• Setting IIC0 as the interface for transmitting the values set to the audio codec registers
● Transmitting the values set to the audio codec registers
● For the I2S bus interface between the microcontroller and the audio codec
• Using TO00 for outputting LRCLK (16 kHz)
• Setting CSI00 as the interface for BCLK, SDOUT, and SDIN
● For the SPI bus interface between the microcontroller and the EEPROMs
• Setting CSI10
• Setting DMA0 and DMA1 to be used for transmitting and receiving data (not used when transmitting commands)
● Erasing all EEPROMs (when not using interrupt servicing)
● Enabling KR0 and KR1 key interrupts as the record and playback keys
I2S bus interface
CSI00 transfer completed
INTCSI00 interrupt servicing
24 bits of data, which is equivalent to half the LRCLK clock, is transmitted or received, and whether one page of data was transmitted or received is checked.
● When recording
• Starting writing to an EEPROM (when receiving one page of data)
• Receiving data for every byte
● During playback
• Transmitting data for every byte
• Starting reading from an EEPROM (when transmitting one page of data)
Interval timer interrupt
Main processing
● Key processing
● Setting recording to be started when the record key is pressed
• Transmitting the values set to the audio codec registers
• Starting I2S operation (CSI00 operation)
• Setting writing to an EEPROM
● Setting starting of playback when the playback key is pressed
• Transmitting the values set to the audio codec registers
• Preparing for I2S to operate (by setting CSI00)
• Setting reading from an EEPROM (CSI10 operation)
● Setting ending of recording
● Setting ending of playback
INTTM00 interrupt servicing
This is used to synchronize CSI00 with LRCLK.
● When recording
• Restarting CSI00 operation
● During playback
• Restarting CSI00 operation
DMA transfer completed
CSI10 transfer completed
DMA transfer completed
INTDMA0 interrupt servicing
This is the servicing of the interrupt that occurs when the reception of one page of data from an EEPROM is completed during playback. The timing of when I2S starts and stops when playback starts or ends is monitored. Which of the four EEPROMs is to be used is selected.
● Acquiring two pages of data before starting I2S
● Selecting the EEPROM to be used
● Starting I2S operation
● Stopping I2S operation
● Stopping DMA operation
INTCSI10 interrupt servicing
This is used for transmitting instruction bytes to an EEPROM. Processing shifts to DMA transfer when transmitting or receiving the data field.
● When recording
• Setting the command for writing data
• Setting the addresses to be written
• Starting transmitting data by using DMA1 (disabling INTCSI10 interrupts)
● During playback
• Setting the command for reading data
• Setting the addresses to be read
• Starting receiving data by using DMA0 and DMA1 (disabling INTCSI10 interrupts)
INTDMA1 interrupt servicing
This occurs when the transmission of one page of data to an EEPROM is completed when recording. The timing of when I2S stops when recording ends is monitored. Which of the four EEPROMs is to be used is selected.
● Selecting the EEPROM to be used
● Stopping I2S operation
● Stopping DMA operation
SPI interface

3.4 Flowcharts

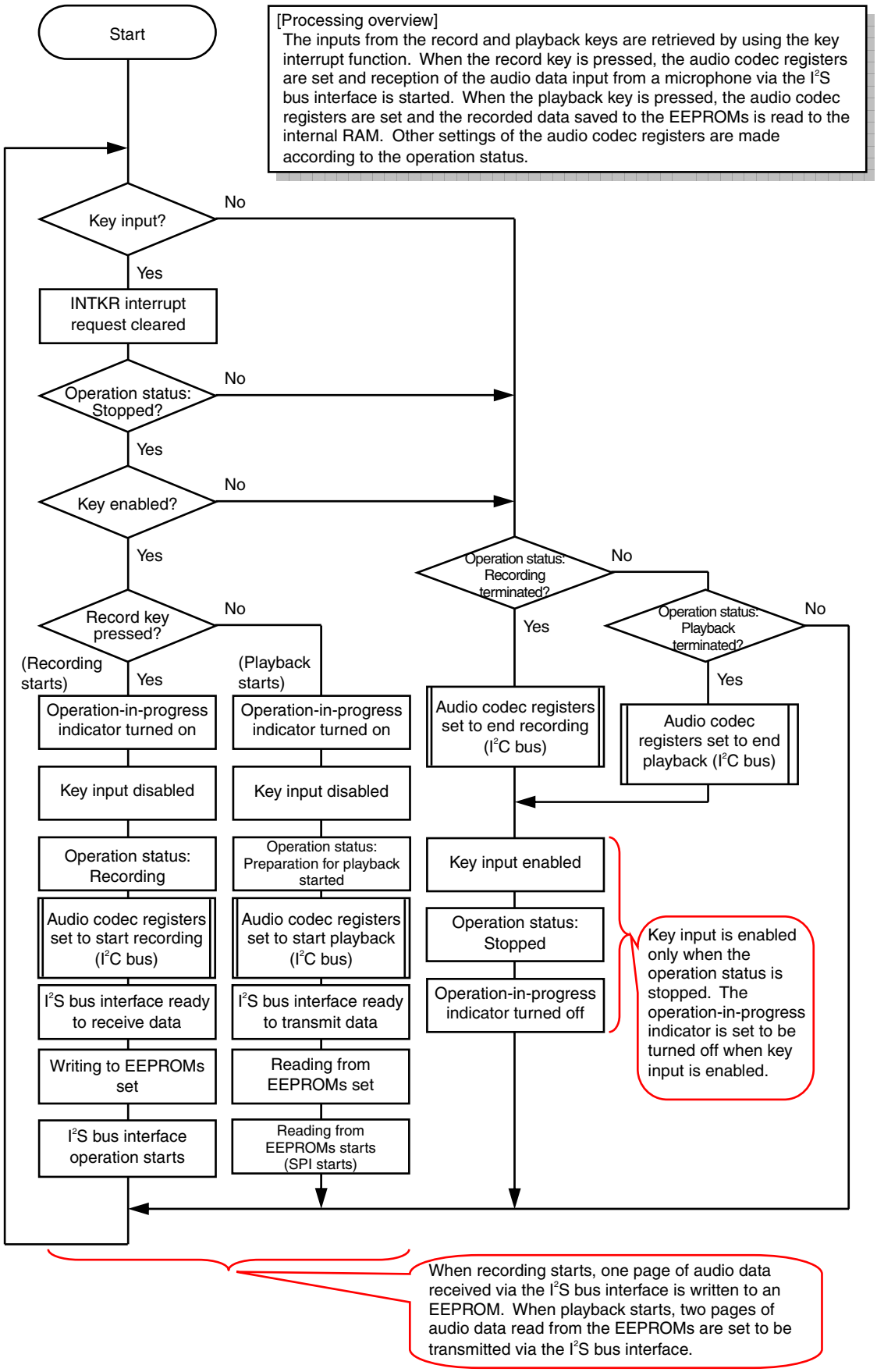
The flowcharts of this sample program are shown below.

<Initial settings of the peripheral hardware to be used after releasing reset>

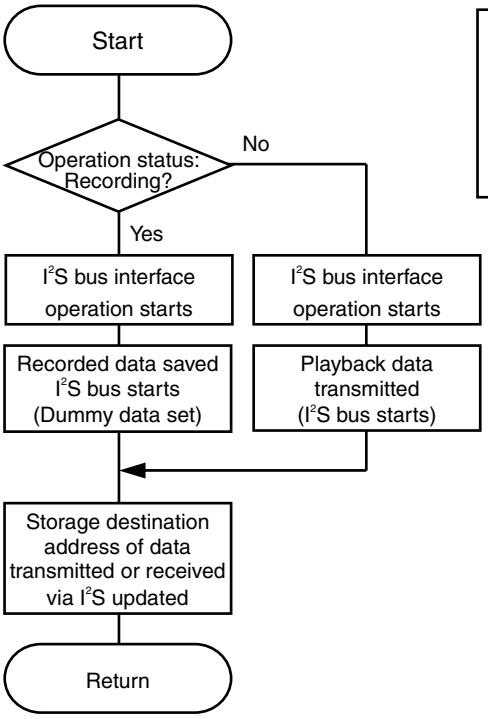


Caution Set the option byte by using the linker options of the RA78K0R. See the RA78K0R Assembler Package User's Manual for information about how to set the option byte.
The option byte is used to set the following.
- **Watchdog timer operation**
- **Setting of LVI when releasing reset (when starting the power supply)**
- **On-chip debug operations**

<Main processing>

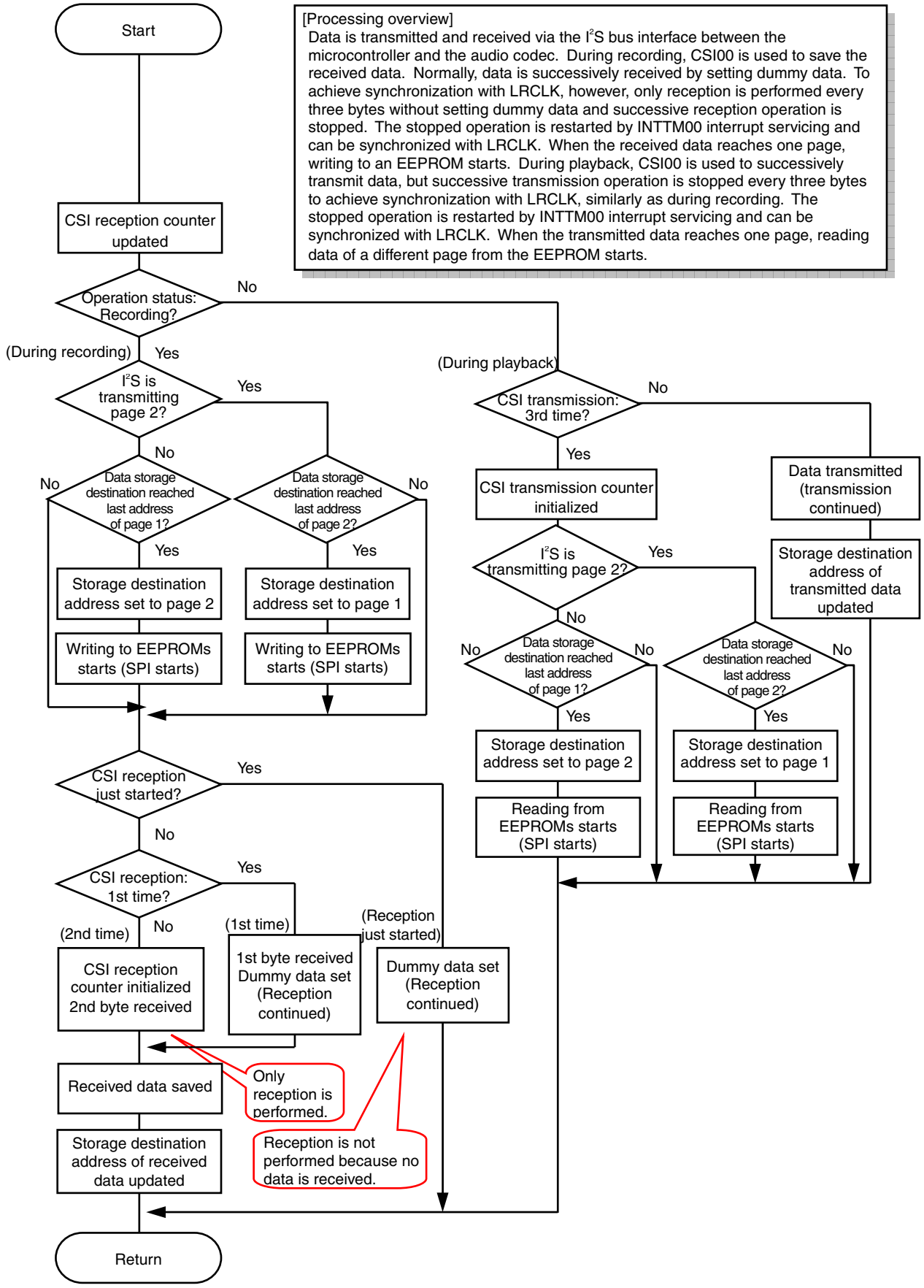


<INTTM00 interrupt servicing for LRCLK synchronization>

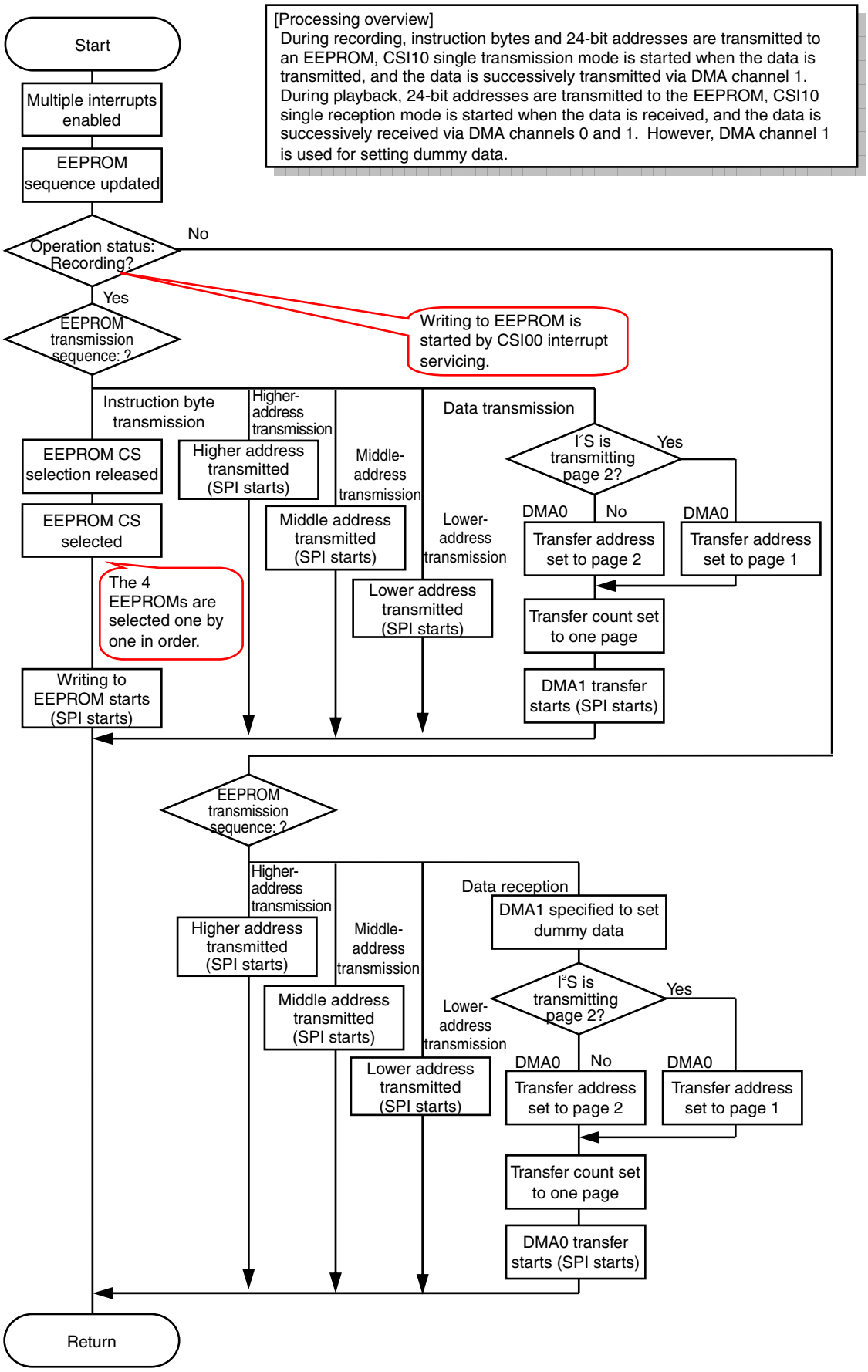


[Processing overview]
LRCLK output from TO00 via the I²S bus interface and BCLK, SDOUT, and SDIN output via CSI00 used for data transmission or reception are synchronized by restarting the operation of CSI00, which was stopped by INTCSI00 interrupt servicing when LRCLK changes from high to low level and from low to high level.

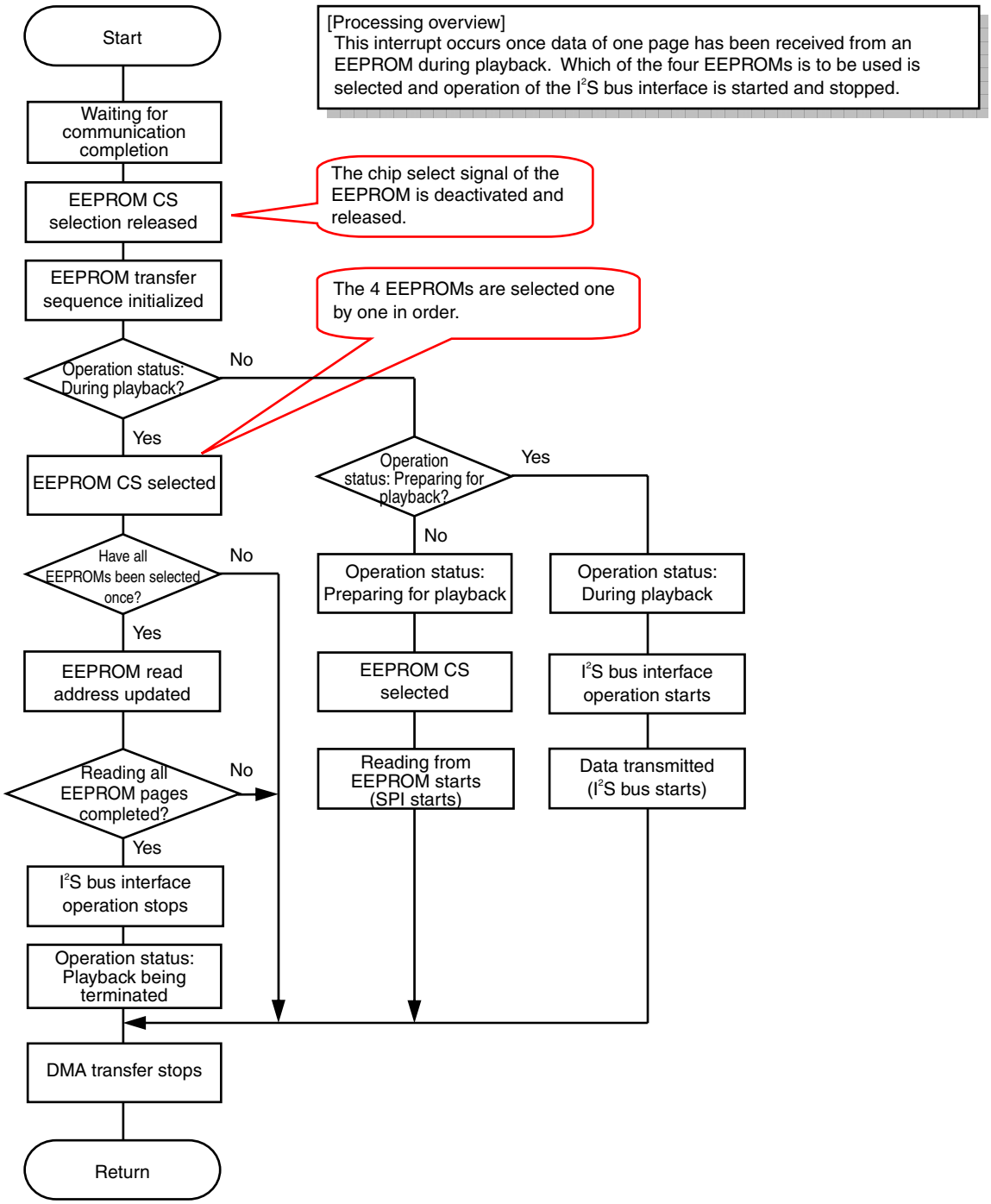
<INTCSI00 interrupt servicing for I²S bus interface>



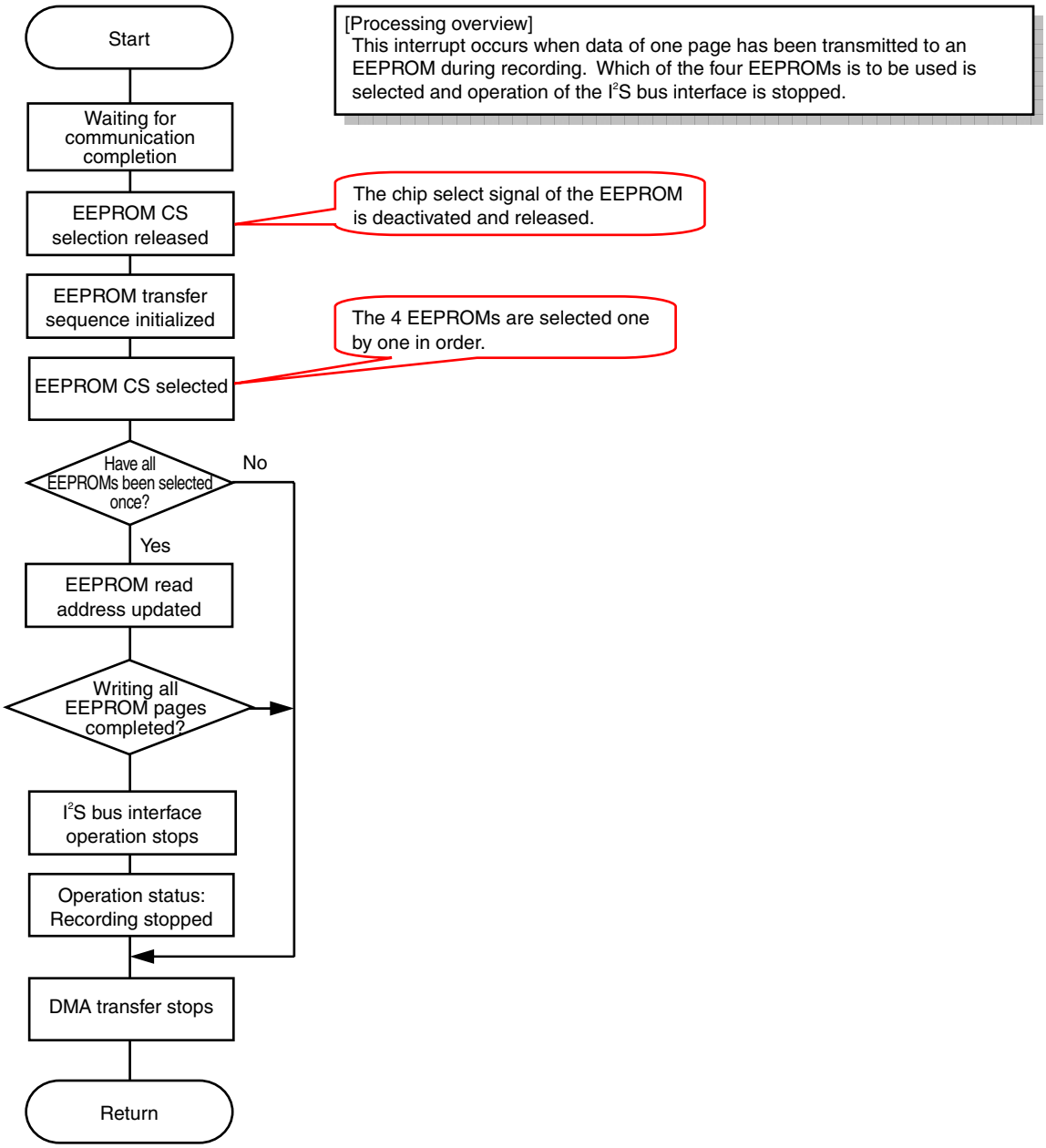
<INTCSI10 interrupt servicing for SPI interface>



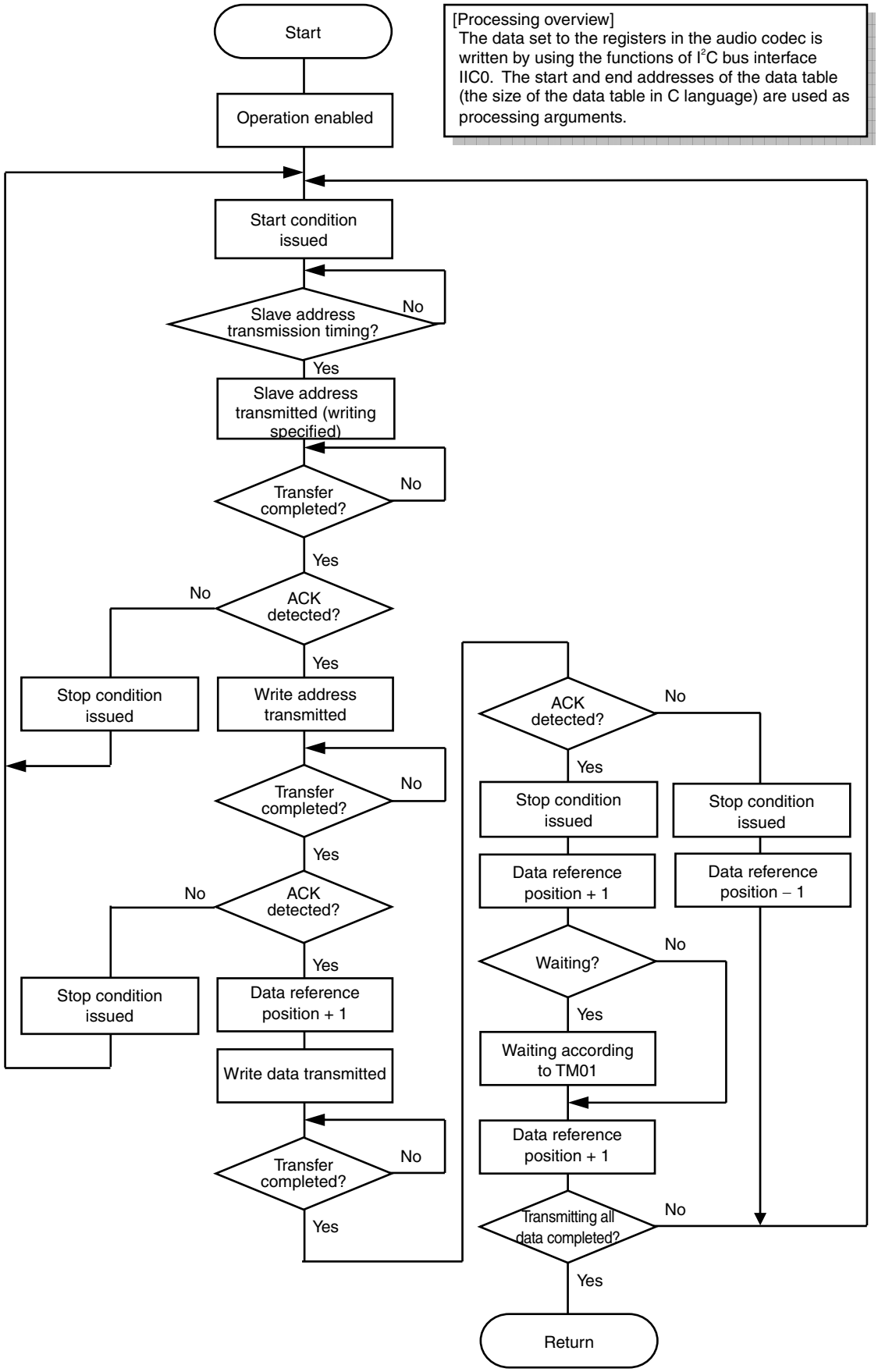
<INTDMA0 interrupt servicing>



<INTDMA1 interrupt servicing>



<I²C bus interface write processing for setting audio codec registers>

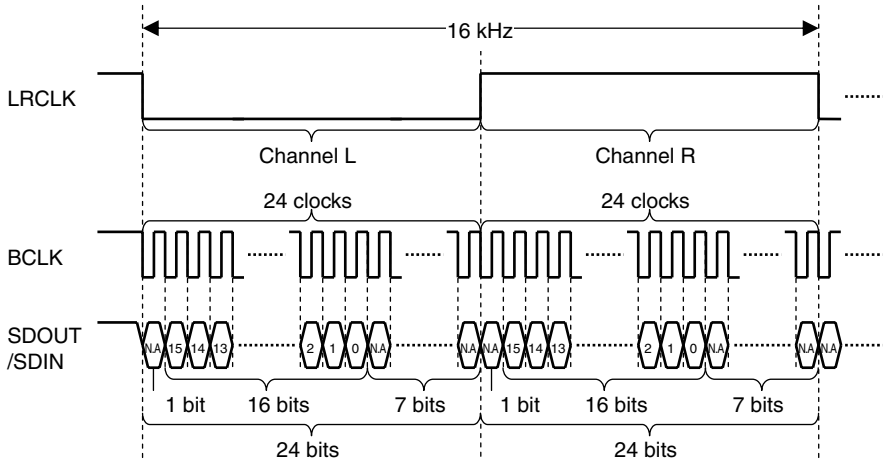


3.5 I²S Bus Interface Format

The format of the I²S bus interface is described below.

[Waveform transmitted to or received from the audio codec]

The waveform transmitted to or received from the audio codec is shown below.



LRCLK:

This is the audio data sampling frequency.

In the program, TO00 output is performed at 16 × 2 kHz intervals by using the interval timer mode of channel 0 of the timer array unit (TAU). Channel 0 of the timer array unit (TAU) can also generate INTTM00 interrupts and synchronized with BCLK, SDOUT, and SDIN.

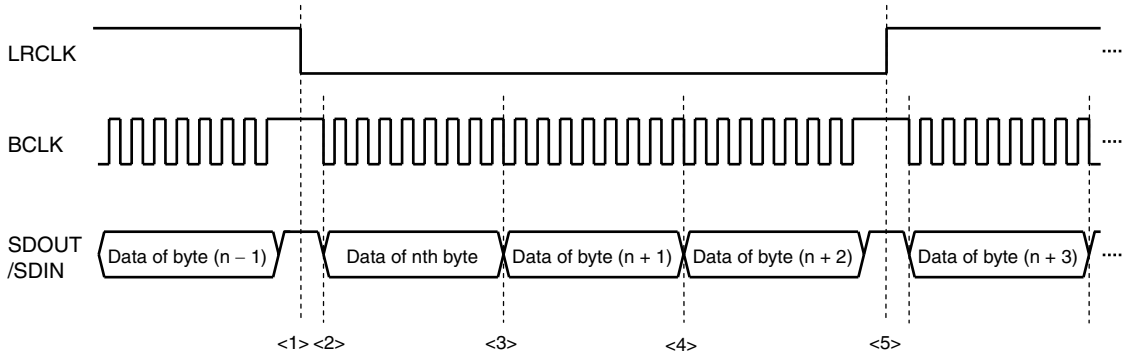
BCLK/SDOUT/SDIN:

These are used to input and output audio data.

In the program, successive master transmission or reception is performed by using CSI00 of channel 0 of serial array unit 0 (SAU0). The transfer speed is set to 833 kHz so that three bytes of data can be transferred within half the LRCLK clock. Channel 0 of serial array unit 0 (SAU0) can also generate buffer empty interrupts (INTCSI00) and perform successive transmission and reception.

[Timing in the program]

The timing in the program is described below.



<1> INTTM00 interrupt timing

- During transmission, the data of the nth byte is written to the transmission and reception buffer register (SIO00) (lower eight bits of the SDR00 register) and successive master transmission of CSI00 is restarted.
- During reception, the transmission and reception buffer register (SIO00) is read and the data of byte (n – 1) is saved. Dummy data is set to the transmission and reception buffer register (SIO00) and successive master reception of CSI00 is restarted.

<2> Timing of first INTCSI00 buffer empty interrupt

- During transmission, data of byte (n + 1) is written to the transmission and reception buffer register (SIO00).
- During reception, the transmission and reception buffer register (SIO00) is not read because it contains no valid data. Dummy data is set to the transmission and reception buffer register (SIO00) to continue successive reception.

<3> Timing of second INTCSI00 buffer empty interrupt

- During transmission, the data of byte (n + 2) is written to the transmission and reception buffer register (SIO00).
- During reception, the transmission and reception buffer register (SIO00) is read and the data of the nth byte is saved. Dummy data is set to the transmission and reception buffer register (SIO00) to continue successive reception.

<4> Timing of third INTCSI00 buffer empty interrupt

- During transmission, restart of the operation at <5> is awaited without data being transmitted.
- During reception, only the transmission and reception buffer register (SIO00) is read and the data of byte (n + 1) is saved. Restart of the operation at <5> is awaited without dummy data being set.

CHAPTER 4 SETTING METHODS

This chapter describes the settings for using the I^2S bus interface and the initial settings of the peripheral hardware to be used. It also describes the main processing, each interrupt servicing, and the subroutines.

See the **78K0R/Kx3 Sample Program (Initial Settings of Peripheral Hardware to Be Used) LED Lighting Switch Control Application Note** for details of how to perform the initial settings.

See each product user's manual (78K0R/KE3, 78K0R/KF3, 78K0R/KG3, 78K0R/KH3, 78K0R/KJ3) for details of how to set the registers.

See the **78K0R Microcontrollers Instructions User's Manual** for details about assembler instructions.

4.1 Settings for Using I²S Bus Interface

In this sample program, the I²S bus interface is connected, the TO00 output of channel 0 of the timer array unit (TAU) is set to output the LR clock so that the audio data can be transmitted or received, and CSI00 of channel 0 of serial array unit 0 (SAU0) is set for transmitting or receiving the data.

[Setting of timer array unit (TAU)]

(1) Controlling the input clock of the timer array unit (TAU)

Peripheral enable register 0 (PER0), which enables the use of peripheral hardware macros, is set so that the timer array unit (TAU) can be used. Power consumption and noise can be reduced by stopping clock supply to unused hardware.

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

Address: F00F0H

RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

Timer array unit input clock control (TAU0EN)

0	Stop supplying the input clock. • The SFRs used by the timer array unit cannot be written. • The timer array unit is in the reset status.
1	Supply the input clock. • The SFRs used by the timer array unit can be read or written.

Cautions 1. Be sure to set TAU0EN = 1 before setting the timer array unit. If TAU0EN = 0, writing to the timer array unit control register is ignored and only the initial values are read (except timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode registers 0, 1, 3, 4, 13, 14 (PM0, PM1, PM3, PM4, PM13, PM14), and port registers 0, 1, 3, 4, 13, 14 (P0, P1, P3, P4, P13, P14)).

2. TAU1EN is provided only in the 78K0R/KJ3 and 78K0R/KH3. DACEN is provided only in the 78K0R/KJ3, 78K0R/KH3, 78K0R/KG3, and 78K0R/KF3. Be sure to set non existent bits to “0”.

(2) Selecting the operation clock (CK00)

Timer clock select register 0 (TPS0) is set to select which of the two operation clocks (CK00 or CK01) will be supplied to channels 0, 1, and 2 of the timer array unit (TAU). CK01 is selected by using bits 7 to 4 and CK00 is selected by using bits 3 to 0 of TPS0.

In this sample program, the operation clock CK00 is used by channels 0 and 2 and the operation clock CK01 is used by channel 1. CK00 is used by channel 0 as the LR clock of the I^2S bus interface.

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

Address: F01B6H, F01B7H

0	0	0	0	0	0	0	0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

Operation clock (CK00) selection

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	$f_{CLK}/2^{15}$

Operation clock (CK01) selection (not used in this sample program)

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	$f_{CLK}/2^{15}$

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

Remark f_{CLK}: CPU or peripheral hardware clock frequency

(3) Operation mode and other settings

Timer mode register 00 (TMR00) is used to select the operation clock (MCK) to be used for the timer array unit (TAU), whether to use the input of the timer clock (TCLK), the valid edge of the timer input pin, and the operation mode for the LR clock of the I^2S bus interface.

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

Address: F0190H, F0191H

CKS00	0	0	CCS00	MASTER00	STS002	STS001	STS000	CIS001	CIS000	0	0	MD003	MD002	MD001	MD000

Channel 0 operation mode setting

MD003 to MD000				Channel 0 operation mode setting	TCR count operation	Stand-alone operation
0	0	0	1/0	Interval timer mode	Down-count	Possible
0	1	0	1/0	Capture mode	Up-count	Possible
0	1	1	0	Event counter mode	Down-count	Not possible
1	0	0	1/0	One-count mode	Down-count	Possible
1	1	0	0	Capture & one-count mode	Up-count	Possible

TI00 pin valid edge selection

0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when measuring low-level width) Start trigger: Falling edge Capture trigger: Rising edge
1	1	Both edges (when measuring high-level width) Start trigger: Rising edge Capture trigger: Falling edge

Channel 0 start trigger and capture trigger setting

0	0	0	Only software trigger start is valid (other trigger sources are deselected).
0	0	1	Use the valid edge of the TI00 pin input for both the start and capture triggers.
0	1	0	Use the rising and falling edges of the TI00 pin input separately for the start and capture triggers.
1	0	0	Use the master channel interrupt signal (when a slave channel is used in a successive operation).
Other			Setting prohibited

Selection of whether channel n operates in standalone mode and whether channel n is the master or the slave in successive operation mode

0	Operate in standalone mode or operate in successive operation mode as the slave channel.
1	Operate as the master channel in the successive operation mode.

Channel 0 count clock (TCLK) selection

0	Operation clock MCK specified by using the CKS00 bit
1	Valid edge of the signal input from the TI00 pin

Channel 0 operation clock (MCK) selection

0	Operation clock CK00 set by using the PRS register
1	Operation clock CK01 set by using the PRS register

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

(4) Enabling timer output

Timer output is enabled for each channel by using timer output enable register 0 (TOE0). For the LR clock of the I^2S bus interface, timer output is enabled by using the TOE00 bit of channel 0.

Figure 4-4. Format of Timer Output Enable Register 0 (TOE0)

Address: F01BAH, F01BBH

0	0	0	0	0	0	0	0	TOE 07	TOE 06	TOE 05	TOE 04	TOE 03	TOE 02	TOE 01	TOE 00

Channel n operation start trigger

0	Stop the operation of TO0n (timer channel output bit) according to the count operation.
1	Enable the operation of TO0n (timer channel output bit) according to the count operation.

Cautions 1. Be sure to set bits 15 to 8 of TOE0 to "0".

2. TOE07 is provided only in the 78K0R/KJ3, 78K0R/KH3, 78K0R/KG3, and 78K0R/KF3. Be sure to set non existent bits to "0".

Remark n: Channel number (n = 0 to 7)

(5) Enabling start of count operations

Starting timer count operations is enabled for each channel by using timer channel start register 0 (TS0). Timer channel start register 0 (TS0) will be cleared when a timer count operation is enabled, because it is a trigger register.

When the operation of the LR clock of the I^2S bus interface is started, the TS00 bit of channel 0 is set to 1.

Figure 4-5. Format of Timer Channel Start Register 0 (TS0)

Address: F01B2H, F01B3H

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

Channel n operation start trigger

0	Do not operate as a trigger.
1	Enable a count operation.

Caution Be sure to set bits 15 to 8 of TS0 to "0".

Remark n: Channel number (n = 0 to 3)

(6) Enabling stopping of count operations

Timer count operations are stopped for each channel by using timer channel stop register 0 (TT0). Timer channel stop register 0 (TT0) will be cleared when a timer count operation is stopped, because it is a trigger register.

When the operation of the LR clock of the I^2S bus interface is stopped, the TT00 bit of channel 0 is set to 1.

Figure 4-6. Format of Timer Channel Stop Register 0 (TT0)

Address: F01B4H, F01B5H

0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

Channel n operation stop trigger

0	Do not operate as a trigger.
1	Stop the operation (generate a stop trigger).

[Setting of serial array unit 0 (SAU0)]

(1) Controlling the input clock of serial array unit 0 (SAU0)

Peripheral enable register 0 (PER0), which enables the use of peripheral hardware macros, is set so that serial array unit 0 (SAU0) can be used. Power consumption and noise can be reduced by stopping clock supply to unused hardware.

Figure 4-7. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H

RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

Serial array unit 0 input clock control (SAU0EN)

0	Stop supplying the input clock. • The SFRs used by serial array unit 0 cannot be written. • Serial array unit 0 is in the reset status.
1	Supply the input clock. • The SFRs used by serial array unit 0 can be read or written.

Cautions 1. Be sure to set SAUmEN = 1 before setting serial array unit m. If SAUmEN = 0, writing to the serial array unit m control register is ignored and only the initial values are read (except input switch control register (ISC), noise filter enable register (NFEN0), port input mode register (PIM0), port output mode register (POM0), port mode registers (PM0, PM1), and port registers (P0, P1)).

2. Set the SPSm register at least four clocks after setting the PER0 register to "1".

3. TAU1EN is provided only in the 78K0R/KJ3 and 78K0R/KH3. DACEN is provided only in the 78K0R/KJ3, 78K0R/KH3, 78K0R/KG3, and 78K0R/KF3. Be sure to set non existent bits to "0".

Remark m: Unit number (m = 0, 1)

(2) Selecting the operation clock (CK00)

Serial clock select register 0 (SPS0) is set to select which of the two operation clocks (CK00 or CK01) will be supplied to both channels 0 and 2 of serial array unit 0 (SAU0). CK01 is selected by using bits 7 to 4 and CK00 is selected by using bits 3 to 0 of SPS0.

In this sample program, the operation clock CK00 is used both by channels 0 and 2. CK00 is used by channel 0 for the I^2S bus interface.

Figure 4-8. Format of Serial Clock Select Register 0 (SPS0)

Address: F0126H, F0127H

0	0	0	0	0	0	0	0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

Operation clock (CK00) selection

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	1	1	INTTM02
Other than above				Setting prohibited

Operation clock (CK01) selection (not used in this sample program)

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	1	1	INTTM02
Other than above				Setting prohibited

Cautions 1. Be sure to set bits 15 to 8 to "0".

2. Set the SPS0 register at least four clocks after setting the PER0 register to "1".

Remark f_{CLK}: CPU or peripheral hardware clock frequency

(3) Operation mode and other settings

Serial mode register 00 (SMR00) is used to select the operation clock (MCK) to be used for CSI00 of serial array unit 0 (SAU0), whether to use the input of the serial clock (SCK), the start trigger, the operation mode (CSI, UART, I²C), and the interrupt source for data transmission and reception via the I²S bus interface.

Figure 4-9. Format of Serial Mode Register 00 (SMR00)

Address: F0110H, F0111H

CKC00	CCS00	0	0	0	0	0	STS00	0	SIS000	1	0	0	MD002	MD001	MD000

Channel 0 interrupt source selection

0	Transfer completion interrupt
1	Buffer empty interrupt

Channel 0 operation mode setting

0	0	CSI mode
0	1	UART mode
1	0	Simplified I²C mode
1	1	Setting prohibited

Control of inverting the level of data received in channel 0 in UART mode (not used)

0	Detect the falling edge as the start bit. The input communication data is retrieved as is.
1	Detect the rising edge as the start bit. The input communication data is inverted and then retrieved.

Start trigger source selection

0	Only the software trigger is valid (selected in CSI, UART transmission, or simplified I²C mode).
1	Valid edge of the RxD pin (selected during UART reception)

Channel 0 transfer clock (TCLK) selection

0	Operation clock MCK is divided according to the ratio specified by using the CKS00 bit.
1	Clock input from the SCK pin (slave transfer in CSI mode)

Channel 0 operation clock (MCK) selection

0	Prescaler output clock CK00 set by using the PRS register
1	Prescaler output clock CK01 set by using the PRS register

Caution Be sure to set bits 13 to 9, 7, 4, and 3 to “0” and bit 5 to “1”.

(4) Setting division of the operation clock

Division of the operation clock (MCK) to be used for CSI00 of serial array unit 0 (SAU0) is set by using the higher seven bits of serial data register 00 (SDR00), a 16-bit register. Note that the lower eight bits of serial data register 00 (SDR00) function as the transmission and reception data register (SIO00).

Figure 4-10. Format of Serial Data Register 00 (SDR00)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)

							0								

Operation clock (MCK) division setting (SDR00[15:9], SDR01[15:9])

0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
0	**0**	**0**	**1**	**0**	**1**	**1**	MCK/24
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

Caution Be sure to set bit 8 to “0”.

(5) Setting communication operations

Communication operations are set by using serial communication setting register 00 (SCR00). SCR00 is used to specify settings such as the data transmission and reception mode, the data and clock phases, and the data length.

Figure 4-11. Format of Serial Communication Setting Register 00 (SCR00)

Address: F0118H, F0119H

TXE00	RXE00	DAP00	CKP00	0	EOC00	PTC001	PTC000	DIR00	0	SLC001	SLC000	0	DLS002	DLS001	DLS000

Data length setting in CSI mode

1	0	0	5-bit data length (data stored in bits 0 to 4 of the SDR00 register)
1	1	0	7-bit data length (data stored in bits 0 to 6 of the SDR00 register)
1	1	1	8-bit data length (data stored in bits 0 to 7 of the SDR00 register)
Other			Setting prohibited

Stop bit setting in UART mode (not used)

0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits
1	1	Setting prohibited

Selection of data transfer order in CSI mode

0	Input and output data MSB first.
1	Input and output data LSB first.

Stop bit setting in UART mode (not used)

		Transmission	Reception
0	0	Do not output the parity bit.	Receive data without parity.
0	1	Output 0 parity.	Do not judge parity.
1	0	Output an even parity.	Judge as even parity.
1	1	Output an odd parity.	Judge as odd parity.

Selection of masking of error interrupt signals (INTSREx (x = 0 to 3))

0	Mask error interrupt INTSREx. (INTSRx is not masked.)
1	Enable generation of error interrupt INTSREx. (INTSRx is masked when an error occurs.)

Selection of data and clock phases in CSI mode

0	0	$\overline{\text{SCKp}}$ / SOp: D7 D6 D5 D4 D3 D2 D1 D0 / SIp input timing
0	1	$\overline{\text{SCKp}}$ / SOp: D7 D6 D5 D4 D3 D2 D1 D0 / SIp input timing
1	0	$\overline{\text{SCKp}}$ / SOp: D7 D6 D5 D4 D3 D2 D1 D0 / SIp input timing
1	1	$\overline{\text{SCKp}}$ / SOp: D7 D6 D5 D4 D3 D2 D1 D0 / SIp input timing

Channel 0 operation mode setting

0	0	Do not start communication.
0	1	Perform only reception.
1	0	Perform only transmission.
1	1	Perform transmission and reception.

Caution Be sure to set bits 11, 6, and 3 to “0” and bit 2 to “1”.

(6) Setting the initial output level

The initial output level of serial array unit 0 (SAU0) in CSI transmission and reception is set by using serial output register 0 (SO0).

For data transmission and reception via the I^2S bus interface, both the CK00 and SO00 bits of channel 0 are set to 1 in the initial settings of the peripheral hardware to be used.

Figure 4-12. Format of Serial Output Register 0 (SO0)

Address: F0128H, F0129H

0	0	0	0	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00

Channel 0 serial clock output

0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

Channel 0 serial data output

0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to set bits 15 to 12 and 7 to 4 of SO0 to “0”. CKO03 and SO03 are provided only in the 78K0R/KJ3 and 78K0R/KH3 microcontrollers. CKO01 and SO01 are provided only in the 78K0R/KJ3, 78K0R/KH3, 78K0R/KG3, and 78K0R/KF3 microcontrollers. When they are not provided, be sure to set CKO01, CKO03, SO01, and SO03 to “1”.

(7) Enabling or stopping the output of serial communication

Serial communication by CSI00 of serial array unit 0 (SAU0) is enabled or stopped by using serial output enable register 0 (SOE0). The value reflected as a result of communication is output from the serial data output pin for channels for which serial output has been enabled.

For data transmission via the I^2S bus interface, the SOE00 bit of channel 0 is set to 1 when the output of CSI00 transmission is enabled in the initial settings of the peripheral hardware to be used.

Figure 4-13. Format of Serial Output Enable Register 0 (SOE0)

Address: F012AH, F012BH (SOE0)

0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00

Enabling or stopping channel 0 serial output

0	Stop serial communication output.
1	Enable serial communication output.

Caution Be sure to set bits 15 to 4 of SOE0 to “0”. SOE03 is provided only in the 78K0R/KJ3 and 78K0R/KH3 microcontrollers. SOE01 is provided only in the 78K0R/KJ3, 78K0R/KH3, 78K0R/KG3, and 78K0R/KF3 microcontrollers. Be sure to set non existent bits to “0”.

(8) Enabling start of serial communication and counting

Starting serial communication and counting by CSI00 of serial array unit 0 (SAU0) is enabled for each channel by using serial channel start register 0 (SS0). Serial channel start register 0 (SS0) will be cleared when serial transmission or reception is enabled, because it is a trigger register.

When data transmission or reception via the I^2S bus interface starts, the SS00 bit of channel 0 (CSI00) is set to 1.

Figure 4-14. Format of Serial Channel Start Register 0 (SS0)

Address: F0122H, F0123H (SS0)

0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

Channel 0 operation start trigger

0	Do not operate as a trigger.
1	Transition to the communication wait status. (If communication is already in progress, stop the communication and wait for a start condition.)

Caution Be sure to set bits 15 to 4 of SS0 to "0".

(9) Enabling stopping of serial communication and counting

Stopping communication and counting is enabled for each channel by using serial channel stop register 0 (ST0). Serial channel stop register 0 (ST0) will be cleared when serial transmission or reception is stopped, because it is a trigger register.

When data transmission or reception via the I^2S bus interface stops, the ST00 bit of channel 0 (CSI00) is set to 1.

Figure 4-15. Format of Serial Channel Stop Register 0 (ST0)

Address: F0124H, F0125H (ST0)

0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

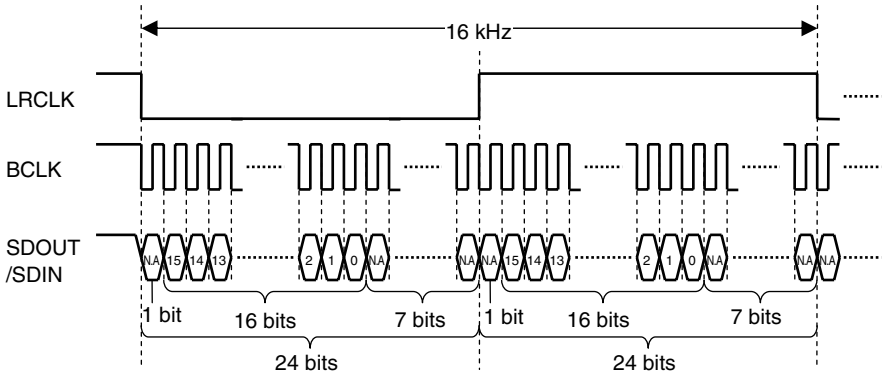
Channel 0 operation stop trigger

0	Do not operate as a trigger.
1	Stop communication. (Communication stops by retaining the values of the control and shift registers and the statuses of the serial clock I/O pins, serial data output pin, and error flags FEF, PEF, and OVF.)

Caution Be sure to set bits 15 to 4 of ST0 to "0".

[Example] Enabling support of an I²S bus interface, such as the one shown below (same description as in the sample program)

Successive master transmission and reception is executed by using LRCLK in the interval timer mode of channel 0 of the timer array unit (TAU), outputting TO00 at 16 × 2 kHz intervals, and using CSI00 of channel 0 of serial array unit 0 (SAU0) for inputting and outputting the audio data of BCLK, SDOUT, and SDIN.



● Setting procedure

<1> Performing initial setting of the port output latch for timer array unit 0 (setting P01 to 1)
<2> Setting the port mode for timer array unit 0 (setting PM01 to 0)
<3> Supplying the input clock to timer array unit 0 (setting TAU0EN to 1)
<4> Selecting the operation clock of timer array unit 0 (TPS0)
<5> Setting the operation mode of channel 0 of timer array unit 0 (TMR00)
<6> Setting the timer counter of channel 0 of timer array unit 0 (TDR00)
<7> Supplying the input clock to serial array unit 0 (setting SAU0EN to 1)
<8> Selecting the operation clock of channel 0 of serial array unit 0 (SPS0)
<9> Setting the transfer clock of channel 0 of serial array unit 0 (SDR00)
<10> Setting the operation mode of channel 0 of serial array unit 0 (SMR00)
<11> Setting the initial data output of channel 0 of serial array unit 0 (SO00)
<12> Enabling the output of channel 0 of serial array unit 0 (SOE0)
<13> Performing initial setting of the port output latch for channel 0 of serial array unit 0 (P10, P12)
<14> Setting the port mode for channel 0 of serial array unit 0 (PM10, PM11, PM12)
<15> Starting the operation of channel 0 of serial array unit 0 (setting SS00 to 1)
<16> Clearing the interrupt request flag corresponding to INTCSI00 (setting CSIIF00 to 0)
<17> Clearing the interrupt mask flag corresponding to INTCSI00 (CSIMK00)
<18> Executing the EI instruction and enabling interrupts
<19> Setting the serial data register of serial array unit 0 (SDR00)
<20> Setting the communication format of channel 0 of serial array unit 0 (SCR00)
<21> Enabling the timer output of channel 0 of timer array unit 0 (setting TOE00 to 1)
<22> Clearing the interrupt request flag corresponding to INTTM00 (setting PIF0 to 0)
<23> Clearing the interrupt mask flag corresponding to INTTM00 (PMK0)
<24> Starting the operation of CSI00 of channel 0 of serial array unit 0 (setting SS00 to 1)
<25> Setting the initial output of the port output latch of channel 0 for timer array unit 0 (setting P01 to 0)
<26> Starting the operation of channel 0 of timer array unit 0 (setting TS00 to 1)
<27> Setting dummy data to SIO00 of channel 0 of serial array unit 0

Steps <1> to <27> above correspond to steps <1> to <27> on the next page.

● Program example in assembly language (same description as in the sample program)

```
    ···(omitted)···

    ;-----------------------------------------------------------------------------
    ;  Port 0 settings
    ;-----------------------------------------------------------------------------
<1>     MOV    P0,       #00011010B         ;Output latches of P00, P02, P05, and P06 set
                                            ;to low level and those of P01, P03, and P04 set
                                            ;to high level
<2>     MOV    PM0,      #10000000B         ;P00 to P06 set as output ports


    ···(omitted)···

        ; Timer array unit timer clock selection
<3>     SET1   !TAU0EN                      ;Input clock supplied to the timer array unit
<4>     MOV!   TPS0L,    #10010000B         ;Operation clock selection
                         ;||||++++---------- CK00: fCLK
                         ;++++-------------- CK01: fCLK/2^9

    ···(omitted)···

<5>     MOVW   AX,       #0000000000000000B ;Operation mode setting
                         ;||||||||||||++++- MD003 to MD000: Interval timer mode
                         ;||||||||||++----- <Fixed to 00>
                         ;||||||||++------- CIS001 and CIS000: Unused
                         ;|||||+++--------- STS002 to STS000: Only software trigger start
                         ;|||||                                enabled
                         ;||||+------------ MASTER00: Standalone operation
                         ;|||+------------- CSS00: Macro clock MCK specified by using the
                         ;|||                       CKS00 bit
                         ;|++-------------- <Fixed to 00>
                         ;+---------------- CKS00: Operating clock CK00 set by using the
                         ;                          PRS register
        MOVW   !TMR00,   AX
<6>     MOVW   TDR00,    #625-1             ;Interval set: 16 kHz output (32 kHz)

    ···(omitted)···

<7>     SET1   !SAU0EN                      ;Input clock of the serial array unit supplied

        NOP                                 ;Waiting
        NOP
        NOP
        NOP

<8>     MOV    !SPS0L,   #00000000B         ;Operation clock selection: fCLK
                         ;||||++++---------- PRS003 to PRS000: fCLK
                         ;++++-------------- PRS013 to PRS010: Unused

<9>     MOVW   SDR00,    #(12-1) shl 9      ;Bits 15 to 7: Transfer clock set (833 kHz)


<10>    MOVW   AX,       #0000000000100001B;Operation mode selected: CSI mode
                         ;||||||||||||||||+- MD000: Buffer empty interrupt
                         ;||||||||||||||++-- MD002 and MD001: CSI mode
                         ;||||||||||+++---- <Fixed to 100>
                         ;||||||||||+------- SIS000: Unused
                         ;||||||||+-------- <Fixed to 0>
                         ;|||||||+--------- STS00: Only software trigger enabled (fixed
                         ;|||||||                  in CSI mode)
                         ;||+++++---------- <Fixed to 00000>
                         ;|+---------------- CSS00: Transfer clock set to a clock obtained
                         ;|                         by dividing operation clock MCK as
                         ;|                         specified by using the CKS00 bit
                         ;+---------------- CKS00: Operation clock set to prescaler
                         ;                          output clock CK00 set by using the
                         ;                          PRS register

        MOVW   !SMR00,   AX

    ···(omitted)···
```

```
        ;Initial data output
<11>    ;MOVW AX,       #0000000100000001B    ;Setting of initial outputs of the SO and SCK pins
                        ;||||||||++++++++----- SO0n to SO00: Used for transmission
                        ;++++++++------------- CKOn to CKO0: Serial clock output of channels
                        ;                                     n to 0
        ;MOVW !SO0,      AX

        ···(omitted)···

<12>    SET1   SOE0L.0                        ;Output enabled

<13>    SET1   P_SCK00                        ;SCK00 latch: High level
<14>    CLR1   PM_SCK00                       ;SCK00 pin output set
<14>    SET1   PM_SI00                        ;SI00 pin input set
<13>    SET1   P_SO00                         ;SO00 latch: High level
<14>    CLR1   PM_SO00                        ;SO00 pin output set

<15>    SET1   SS0L.0                         ;CSI00 operation starts (trigger bit)
<16>    CLR1   CSIIF00                        ;INTCSI00 interrupt request cleared
<17>    CLR1   CSIMK00                        ;INTCSI00 interrupt servicing enabled

        ···(omitted)···

<18>    EI                                    ;Enabling interrupts

        ···(omitted)···

        ···(omitted)···
        ;Preparation of the I2S bus interface
        MOV    RCSI00CNT,#0                   ;CSI00 transmission counter
<19>    MOVW   SDR00,    #(12-1) shl 9        ;Bits 15 to 7: Transfer clock set (833 kHz)
        MOVW   AX,       #0100000000000111B   ;Communication format setting
<20>                     ;||||||||||||||+++----- DLS002 to DLS000: 8-bit data length
                         ;||||||||||||||+-------- <Fixed to 0>
                         ;||||||||||++---------- SLC001 and SLC000: Unused (fixed to 0)
                         ;||||||||||+------------ <Fixed to 0>
                         ;||||||||+------------- DIR00: Input and output performed MSB first
                         ;||||||++-------------- PTC001 and PTC000: Unused (fixed to 00)
                         ;|||||+---------------- EOC00: Unused (fixed to 0)
                         ;||||+----------------- <Fixed to 0>
                         ;||++------------------ CKP00/DAP00: Phases of data and clock in CSI
                         ;||                                  mode selected
                         ;||                      [00 selected]
                         ;++-------------------- TXE00/RXE00: Only transmission performed
        MOVW   !SCR00,   AX

        MOV    TO0L,     #00000000B            ;Initial output set to low level
<21>    SET1   TOE0L.0                        ;Operation of TO00 enabled by a count operation
                                              ;(LRCLK)
<22>    CLR1   TMIF00                         ;INTTM00 interrupt request cleared
<23>    CLR1   TMMK00                         ;INTTM00 interrupt servicing enabled

        ···(omitted)···

        ;Starting I2S operation
<24>    SET1   SS0L.0                         ;CSI00 operation starts (trigger bit)
<25>    CLR1   P_LRCLK                        ;LRCLK operation starts
<26>    SET1   TS0L.0                         ;TO00 output operation starts (trigger bit)
<27>    MOV    SIO00,#0FFH                    ;Dummy data set (INTCSI00 occurs)

        ···(omitted)···
```

Remark <1> to <27> above correspond to steps <1> to <27> in the setting procedure.

● Program example in C language (same description as in the sample program)

```
···(omitted)···(Before processing the program)···  #pragma directive enabling description of the name of a special function register (SFR)

#pragma SFR                                  /* Enable description of special function
                                                register (SFR) names */
#pragma EI                                   /* Enable description of EI instructions */


···(omitted)···(Before processing the program)···

/*----------------------------  #pragma directive enabling description of the interrupt enable (EI) instruction  -
   Port 0 settings
-------------------------------------------------------------------------------*/
<1>  P0  = 0b00011010;            /* Output latches of P00, P02, P05, and P06 set to
                                     low level and those of P01, P03, and P04 set to
                                     high level */
<2>  PM0 = 0b10000000;            /* P00 to P06 set as output ports */

···(omitted)···

   /* Timer array unit timer clock selection */
<3>  TAU0EN = 1;                  /* Input clock supplied to the timer array unit */
<4>  TPS0L = 0b10010000;          /* Operation clock selection */
   /*        ||||++++------------ CK00: fCLK */
   /*        ||||++++------------ CK01: fCLK/2^9 */

···(omitted)···

<5>  TMR00 = 0b0000000000000000; /* Operation mode setting */
   /*        ||||||||||||++++---- MD003 to MD000: Interval timer mode */
   /*        ||||||||||++-------- <Fixed to 00> */
   /*        ||||||||++---------- CIS001 and CIS000: Unused */
   /*        |||||+++------------ STS002 to STS000: Only software trigger start
   /*        |||||                                  enabled */
   /*        ||||+--------------- MASTER00: Standalone operation */
   /*        |||+---------------- CSS00: Macro clock MCK specified by using
   /*        |||                         the CKS00 bit */
   /*        |++----------------- <Fixed to 00> */
   /*        +------------------- CKS00: Operating clock CK00 set by using
   /*                                    the PRS register */
<6>  TDR00 = 625-1;               /* Interval set: 16 kHz output (32 kHz) */

···(omitted)···

<7>  SAU0EN = 1;                  /* Input clock of the serial array unit supplied */

   NOP();                         /* Waiting */
   NOP();
   NOP();
   NOP();

<8>  SPS0L = 0b00000000;         /* Operation clock selection: fCLK */
   /*        ||||++++------------ PRS003 to PRS000: fCLK */
   /*        ++++---------------- PRS013 to PRS010: Unused */

<9>  SDR00 = (12-1) << 9;  /* Bits 15 to 7: Transfer clock set (833 kHz) */

<10> SMR00 = 0b0000000000100001;/* Operation mode selected: CSI mode */
   /*        ||||||||||||||||+--- MD000: Buffer empty interrupt */
   /*        ||||||||||||||++---- MD002 and MD001: CSI mode */
   /*        ||||||||||+++------ <Fixed to 100> */
   /*        |||||||||+--------- SIS000: Unused */
   /*        ||||||||+---------- <Fixed to 0> */
   /*        |||||||+----------- STS00: Only software trigger enabled
   /*        |||||||                    (fixed in CSI mode) */
   /*        ||+++++------------ <Fixed to 00000> */
   /*        |+----------------- CSS00: Transfer clock set to a clock
   /*        |                           obtained by dividing operation clock
   /*        |                           MCK as specified by using the CKS00
   /*        |                           bit */
   /*        +------------------ CKS00: Operation clock set to prescaler
   /*                                    output clock CK00 set by using
   /*                                    the PRS register */
```

```
        ···(omitted)···

            /* Initial data output (at the same time as setting CSI10) */
<11>        SO0 = 0b00000001 00000001;   /* Setting of initial outputs of the SO and SCK pins */
            /*      ||||||||++++++++------ SO0n to SO00: Used for transmission */
            /*      ++++++++-------------- CKOn to CKO0: Serial clock output of channels n
            /*                                            to 0 */

<12>        SOE0L.0 = 1;                 /* Output enabled */

<13>        P_SCK00 = 1;                 /* SCK00 latch: High level */
<14>        PM_SCK00 = 0;                /* SCK00 pin output set */
<14>        PM_SI00 = 1;                 /* SI00 pin input set */
<13>        P_SO00 = 1;                  /* SO00 latch: High level */
<14>        PM_SO00 = 0;                 /* SO00 pin output set */

<15>        SS0L.0 = 1;                  /* CSI00 operation starts (trigger bit) */
<16>        CSIIF00 = 0;                 /* INTCSI00 interrupt request cleared */
<17>        CSIMK00 = 0;                 /* INTCSI00 interrupt servicing enabled */

        ···(omitted)···

<18>          EI                         /* Enabling interrupts */

        ···(omitted)···

            /* Preparation of the I2S bus interface */
            ucI2sByteCouter = 0;         /* CSI00 reception counter */
<19>        SDR00 = (12-1) << 9;         /* Bits 15 to 7: Transfer clock set (833 kHz) */
<20>        SCR00 = 0b0100000000000111; /* Communication format setting */
            /*        |||||||||||||+++---- DLS002 to DLS000: 8-bit data length */
            /*        |||||||||||||+------- <Fixed to 0> */
            /*        |||||||||||++-------- SLC001 and SLC000: Unused (fixed to 0) */
            /*        |||||||||||+---------- <Fixed to 0> */
            /*        ||||||||||+----------- DIR00: Input and output performed MSB first */
            /*        ||||||||++------------ PTC001 and PTC000: Unused (fixed to 00) */
            /*        |||||||+-------------- EOC00: Unused (fixed to 0) */
            /*        ||||||+--------------- <Fixed to 0> */
            /*        ||||++---------------- CKP00/DAP00: Phases of data and clock in
            /*        ||||                                 CSI mode selected */
            /*        ||||                    [00 selected] */
            /*        ||||
            /*        ++++------------------ TXE00/RXE00: Only reception performed */

<21>        TOE0L.0 = 1;                 /* Operation of TO00 enabled by a count operation
                                            (LRCLK) */
<22>        TMIF00 = 0;                  /* INTTM00 interrupt request cleared */
<23>        TMMK00 = 0;                  /* INTTM00 interrupt servicing enabled */

        ···(omitted)···

            /* Starting I2S operation */
<24>        SS0L.0 = 1;                  /* CSI00 operation starts (trigger bit) */
<25>        P_LRCLK = 0;                 /* LRCLK operation starts */
<26>        TS0L.0 = 1;                  /* TO00 output operation starts (trigger bit) */
<27>        SIO00 = 0x0FF;               /* Dummy data set (INTCSI00 occurs) */

        ···(omitted)···
```

Remark <1> to <27> above correspond to steps <1> to <27> in the setting procedure.

4.2 Definitions of Variables and Constants

The variables and constants used in assembly language are defined below.

● Port definitions

Port definitions are used to clarify the ports to be used in the program and make it easier to understand the functions of the port and port mode settings.

```
;==============================================================================
;
;  Port definitions
;
;==============================================================================
;Audio codec reset
P_RESETB        EQU      P7.3            ;Reset
PM_RESETB       EQU      PM7.3           ;Reset output

;I2C interface used for setting the registers in the audio codec
P_SCL           EQU      P6.0            ;SCL0 latch
PM_SCL          EQU      PM6.0           ;SCL0 output
P_SDA           EQU      P6.1            ;SDA0 latch
PM_SDA          EQU      PM6.1           ;SDA0 I/O

;Audio codec I2S bus interface
P_LRCLK         EQU      P0.1            ;LRCLK
PM_LRCLK        EQU      PM0.1           ;LRCLK output
P_SCK00         EQU      P1.0            ;SCK00
PM_SCK00        EQU      PM1.0           ;SCK00 output
PM_SI00         EQU      PM1.1           ;SI00 input
P_SO00          EQU      P1.2            ;S000
PM_SO00         EQU      PM1.2           ;SO00 output

;Key inputs (used: 2)
P_KEY           EQU      P7              ;KR (P70 and P71 used)
PM_KEY          EQU      PM7             ;KR input (PM70 and PM71 used)
 P_PLAYKY       EQU      P7.0            ;Playback key
 PM_PLAYKY      EQU      PM7.0           ;Playback key input
 P_RECKY        EQU      P7.1            ;Record key
 PM_RECKY       EQU      PM7.1           ;Record key input

;Output to the LED used as an operation-in-progress indicator (used: 1)
P_LED           EQU      P7.2            ;Operation-in-progress indicator (active low)
PM_LED          EQU      PM7.2           ;Output to the operation-in-progress indicator
                                         ;(LED)
                                         ;LED lit during operation and turned off when
                                         ;key input is possible.
;EEPROM SPI interfaces (used: 4)
P_CS            EQU      P7              ;Chip select CS (P73 to P77 used)
PM_CS           EQU      PM7             ;Chip select CS output (PM73 to PM77 used)
 P_CS3          EQU      P7.7            ;CS3
 PM_CS3         EQU      PM7.7           ;CS3 output
 P_CS2          EQU      P7.6            ;CS2
 PM_CS2         EQU      PM7.6           ;CS2 output
 P_CS1          EQU      P7.5            ;CS1
 PM_CS1         EQU      PM7.5           ;CS1 output
 P_CS0          EQU      P7.4            ;CS0
 PM_CS0         EQU      PM7.4           ;CS0 output
P_SCK10         EQU      P0.4            ;SCK10
PM_SCK10        EQU      PM0.4           ;SCK10 output
PM_SI10         EQU      PM0.3           ;SI10 input
P_SO10          EQU      P0.2            ;SO10
PM_SO10         EQU      PM0.2           ;SO10 output
```

● Definitions of variables and constants

The definitions of variables and constants are largely classified into those related to the operation of the entire system, those related to the operation of the I²S bus interface between the microcontroller and audio codec, those related to the operation of the SPI between the microcontroller and the EEPROMs, and those of the memory area in the internal RAM used for temporarily saving data during recording or playback.

```
;==============================================================================
;
;  RAM definitions
;
;==============================================================================
DPLAY DSEG   SADDR
;Overall operation
RPLAYMOD:       DS        1               ;Operation status
 CRESET         EQU       0               ; Reset
 CSTOP          EQU       1               ; Stopped
 CREC           EQU       2               ; Recording under execution
 CREC_END       EQU       3               ;  Recording finished
 CPLAY_START    EQU       4               ; Preparing for playback started
 CPLAY_SET      EQU       5               ;  Preparing for playback
 CPLAY          EQU       6               ;  Playback under execution
 CPLAY_END      EQU       7               ;  Playback finished

;Definitions related to the operation of the I2S bus interface used between the
;microcontroller and audio codec

RCSI00CNT:      DS        1               ;Counter for transmitting and receiving 1-byte
                                          ;I2S data
RPLAYINFO       DS        1               ;Playback information
FI2SPAGE        EQU       RPLAYINFO.7     ; Page being transmitted or received via the I2S
                                          ; bus: Page 1 (0)/page 2 (1)

;Definitions related to the operation of the SPI interface used between the
;microcontroller and EEPROMs
RP_CS:          DS        1               ;Chip select CS setting value of the EEPROMs
REEPSEQ:        DS        1               ;EEPROM transfer sequence
 CEEPSEQ_RESET  EQU       0               ; Reset status
 CEEPSEQ_WREN   EQU       1               ; Write enable signal transmitted
 CEEPSEQ_INST   EQU       2               ; Instruction bytes transmitted
 CEEPSEQ_ADDRH  EQU       3               ; Higher 8 bits of 24-bit addresses transmitted
 CEEPSEQ_ADDRM  EQU       4               ; Middle 8 bits of 24-bit addresses transmitted
 CEEPSEQ_ADDRL  EQU       5               ; Lower 8 bits of 24-bit addresses transmitted
 CEEPSEQ_DATA   EQU       6               ; Data transmitted and received by using DMA

DPLAYP   DSEG   SADDRP
; Definitions related to the operation of the I2S bus interface used between the
; microcontroller and audio codec
RI2SADDR:       DS        2               ;Address to which data is transmitted or
                                          ;received via I2S saved

;Definitions related to the operation of the SPI interface used between the
;microcontroller and EEPROMs
REEPADDR:       DS        2               ;EEPROM read or write address (REEPADDR*100H)

DMEM     DSEG   UNITP
;Memory area of internal RAM
RRECMEM1:                                 ;Page 1
                DS        3*2*42          ; 42 LRCLK clocks
RRECMEM1E:                                ; Last address + 1
RRECMEM2:                                 ;Page 2
                DS        3*2*42          ; 42 LRCLK clocks
RRECMEM2E:                                ; Last address + 1
```

The variables and constants used in C language are defined below.

● Port definitions

As in assembly language, port definitions are used to clarify the ports to be used in the program and make it easier to understand the functions of the port and port mode settings.

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

   Port definitions

==============================================================================*/
/* Audio codec reset */
#define  P_RESETB     P7.3      /* Reset */
#define  PM_RESETB    PM7.3     /* Reset output */

/* I2C interface used for setting the registers in the audio codec */
#define  P_SCL        P6.0      /* SCL0 latch */
#define  PM_SCL       PM6.0     /* SCL0 output */
#define  P_SDA        P6.1      /* SDA0 latch */
#define  PM_SDA       PM6.1     /* SDA0 I/O */

/* Audio codec I2S bus interface */
#define  P_LRCLK      P0.1      /* LRCLK */
#define  PM_LRCLK     PM0.1     /* LRCLK output */
#define  P_SCK00      P1.0      /* SCK00 */
#define  PM_SCK00     PM1.0     /* SCK00 output */
#define  PM_SI00      PM1.1     /* SI00 input */
#define  P_SO00       P1.2      /* SO00 */
#define  PM_SO00      PM1.2     /* SO00 output */

/* Key inputs (used: 2) */
#define  P_KEY        P7        /* KR (P70 and P71 used) */
#define  PM_KEY       PM7       /* KR input (PM70 and PM71 used) */
#define   P_PLAYKY    P7.0      /* Playback key */
#define   PM_PLAYKY   PM7.0     /* Playback key input */
#define   P_RECKY     P7.1      /* Record key */
#define   PM_RECKY    PM7.1     /* Record key input */

/* Output to the LED used as an operation-in-progress indicator (used: 1) */
#define  P_LED        P7.2      /* Operation-in-progress indicator (active low) */
#define  PM_LED       PM7.2     /* Output to the operation-in-progress
                                   indicator (LED) */
                               /* LED lit during operation and turned off when key
                                   input is possible. */

/* EEPROM SPI interfaces (used: 4) */
#define  P_CS         P7        /* Chip select CS (P73 to P77 used) */
#define  PM_CS        PM7       /* Chip select CS output (PM73 to PM77 used) */
#define   P_CS3       P7.7      /* CS3 */
#define   PM_CS3      PM7.7     /* CS3 output */
#define   P_CS2       P7.6      /* CS2 */
#define   PM_CS2      PM7.6     /* CS2 output */
#define   P_CS1       P7.5      /* CS1 */
#define   PM_CS1      PM7.5     /* CS1 output */
#define   P_CS0       P7.4      /* CS0 */
#define   PM_CS0      PM7.4     /* CS0 output */
#define  P_SCK10      P0.4      /* SCK10 */
#define  PM_SCK10     PM0.4     /* SCK10 output */
#define  PM_SI10      PM0.3     /* SI10 input */
#define  P_SO10       P0.2      /* SO10 */
#define  PM_SO10      PM0.2     /* SO10 output */
```

● Definitions of variables and constants

As in assembly language, the definitions of variables and constants are largely classified into those related to the operation of the entire system, those related to the operation of the I²S bus interface between the microcontroller and audio codec, those related to the operation of the SPI between the microcontroller and the EEPROMs, and those of the memory area in the internal RAM used for temporarily saving data during recording or playback.

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

   RAM definitions

==============================================================================*/
/* Overall operation */
unsigned char ucPlayMode;            /* Operation status                  */
#define  CRESET            0         /*  Reset                            */
#define  CSTOP             1         /*  Stopped                          */
#define  CREC              2         /*  Recording under execution        */
#define  CREC_END          3         /*   Recording finished              */
#define  CPLAY_START       4         /*  Preparing for playback started   */
#define  CPLAY_SET         5         /*   Preparing for playback          */
#define  CPLAY             6         /*   Playback under execution        */
#define  CPLAY_END         7         /*   Playback finished               */

/* Definitions related to the operation of the I2S bus interface used between the
   microcontroller and audio codec */
unsigned char *ucI2sAddress;         /* Address to which data is transmitted or received
                                        via I2S */
unsigned char ucI2sByteCouter;       /* Counter for transmitting and receiving 1-byte
                                        I2S data */
boolean  bI2sMemoryPage;             /* Page being transmitted or received via the
                                        I2S bus: Page 1 (0)/page 2 (1) */

/* Definitions related to the operation of the SPI interface used between the
   microcontroller and EEPROMs */
unsigned short ushEepromAddress;     /* EEPROM read or write address
                                        (ushEepromAddress*0x100) */
unsigned char ucEepromCs;            /* Chip select CS setting value of the EEPROMs */
unsigned char ucEepromSeq;           /* EEPROM transfer sequence */
#define  CEEPSEQ_RESET     0         /*  Reset status                                */
#define  CEEPSEQ_WREN      1         /*  Write enable signal transmitted             */
#define  CEEPSEQ_INST      2         /*  Instruction bytes transmitted               */
#define  CEEPSEQ_ADDR      3         /*  Higher 8 bits of 24-bit addresses
                                        transmitted                                  */
#define  CEEPSEQ_ADDR      4         /*  Middle 8 bits of 24-bit addresses
                                        transmitted                                  */
#define  CEEPSEQ_ADDRL     5         /*  Lower 8 bits of 24-bit addresses transmitted  */
#define  CEEPSEQ_DATA      6         /*  Data transmitted and received by using DMA   */

/* Memory area of internal RAM */
unsigned char ucMemoryPage1[3*2*42]; /* Page 1 (42 LRCLK clocks) */
unsigned char ucMemoryPage2[3*2*42]; /* Page 2 (42 LRCLK clocks) */
```

4.3 Initial Settings of Peripheral Hardware to Be Used

The following operations are performed when the initial settings of the peripheral hardware to be used are specified in assembly language.

<1> Interrupts are disabled.
<2> The register bank is set.
<3> The stack pointer is set.
<4> The CPU or peripheral hardware clock frequency is set to operate on the X1 oscillation clock (20 MHz).
<5> The ports are set.
<6> The registers in the audio codec are set.
(a) The TO02 output of channel 2 of the timer array unit (TAU) is set to supply the clock to the audio codec.
(b) The interval timer mode of channel 1 of the timer array unit (TAU) is set to insert the wait operations to be used in the program.
(c) A wait is inserted until the power supply of the audio codec stabilizes and the audio codec is reset.
(d) Serial interface IIC0 is used to enable the registers in the audio codec to be set.
(e) The system is turned on by setting the registers in the audio codec.
<7> The statuses of the operations used in the program are set to "stopped".
<8> The I^2S bus interface for the audio data of the audio codec is set.
(a) The TO00 output of channel 0 of the timer array unit (TAU) is set to output LRCLK.
(b) CSI00 of channel 0 of serial array unit 0 (SAU0) is set to transmit and receive data.
<9> The SPI interface of the EEPROMs is set.
(a) DMA channel 0 is set to receive data via the CSI used as the SPI interface.
(b) DMA channel 1 is set to set dummy data when transmitting or receiving data via the CSI used as the SPI interface.
(c) CSI10 of channel 2 of serial array unit 0 (SAU0), which is used as the SPI interface, is set.
(d) All EEPROMs are erased (all set to 0FFH).
<10> Key interrupts are enabled and key retrieval is started.
<11> The operation-in-progress indicator is turned off.
<12> Interrupts are enabled.
<13> The program shifts to the main processing.

```
;*****************************************************************************
;
;
;     Initial settings of the peripheral hardware to be used
;
;
;*****************************************************************************
XMAIN CSEG  UNIT
RESET_START:

;-----------------------------------------------------------------------------
;     Disabling interrupts
;-----------------------------------------------------------------------------
<1>   DI
;-----------------------------------------------------------------------------
;     Register bank setting
;-----------------------------------------------------------------------------
<2>   SEL   RB0
;-----------------------------------------------------------------------------
;     Stack pointer setting
;-----------------------------------------------------------------------------
<3>   MOVW  SP,       #LOWW STACKTOP     ; Set the stack pointer

;-----------------------------------------------------------------------------
;     Clock frequency settings
;-----------------------------------------------------------------------------
;     Setting so that operations can be performed using the 20 MHz X1 oscillator
;-----------------------------------------------------------------------------
      MOV   CMC,      #01000001B         ;Clock operation mode
<4>                   ;|||||||+---------- AMPH: 10MHz<fMX≤20MHz
                      ;||||+++----------- <000>
                      ;|||+--------------- OSCSELS: P123 and P124 pins set as input ports
                      ;||+---------------- <0>
                      ;++----------------- EXCLK/OSCSEL: X1 oscillation mode (20 MHz)

      MOV   CSC,      #01000000B         ;Clock operation status control
                      ;|||||||+---------- HIOSTOP: Internal high-speed oscillator
                      ;|||||||                     operated
                      ;||+++++----------- <00000>
                      ;|+----------------- XTSTOP: XT1 oscillator stopped
                      ;+------------------ MSTOP: X1 oscillator operated

      MOVO  SMC,      #00000001B         ;Operation speed mode
                      ;|||||||+---------- FSEL: Operated at a frequency exceeding 10 MHz
                      ;+++++++----------- <00000>

      MOV   OSTS,     #00000101B         ;Oscillation stabilization time: 2^15/fX

HRST300:
      NOP
      BF    OSTC.2,   $HRST300           ;Waiting for clock oscillation to stabilize

      MOV   CKC,      #00011000B         ;Clock selection
                      ;|||||+++---------- MDIV2-0: CPU/peripheral hardware
                      ;|||||                       clock (fCLK) = fMX
                      ;||||+------------- <1>
                      ;|||+--------------- MCM0: High-speed system clock (fMX)
                      ;||+---------------- <R>
                      ;|+----------------- CSS: Main system clock (fMAIN) = fCLK
                      ;+------------------ <R>

<5>   ;-----------------------------------------------------------------------------
      ;     Port 0 settings
      ;-----------------------------------------------------------------------------
      MOV   P0,       #00011010B         ;Output latches of P00, P02, P05, and P06 set
                                         ;to low leveland those of P01, P03, and P04 set
                                         ;to high level
      MOV   PM0,      #10000000B         ;P00 to P06 set as output ports
                                         ;P01: Output LRCLK to the audio codec
                                         ;P02: Output SO10 to the EEPROM
```

```
                                        ;P03: Input SI10 from the EEPROM
                                        ;P04: Output SCK10 to the EEPROM
                                        ;P00, P05, and P06: Unused

;-------------------------------------------------------------------------------
;     Port 1 settings
;-------------------------------------------------------------------------------
      MOV   P1,       #00000000B        ;Output latches of P10 to P17 set to low level
      MOV   PM1,      #00000000B        ;P10 to P17 set as output ports
                                        ;P10: Output SCK00 to the audio codec
                                        ;P11: Input SI00 from the audio codec
                                        ;P12: Output SO00 to the audio codec
                                        ;P13 to P17: Unused

;-------------------------------------------------------------------------------
;     Port 2 settings
;-------------------------------------------------------------------------------
      MOV   P2,       #00000000B        ;Output latches of P20 to P27 set to low level
      MOV   PM2,      #00000000B        ;P20 to P27 set as output ports
                                        ;P20 to P27: Unused

;-------------------------------------------------------------------------------
;     Port 3 settings
;-------------------------------------------------------------------------------
      MOV   P3,       #00000000B        ;Output latches of P30 and P31 set to low level
      MOV   PM3,      #11111100B        ;P30 and P31 set as output ports
                                        ;P30 and P31: Unused

;-------------------------------------------------------------------------------
;     Port 4 settings
;-------------------------------------------------------------------------------
      MOV   P4,       #00000000B        ;Output latches of P40 to P47 set to low level
      MOV   PM4,      #00000000B        ;P40 to P47 set as output ports
                                        ;P40 to P47: Unused

;-------------------------------------------------------------------------------
;     Port 5 settings
;-------------------------------------------------------------------------------
      MOV   P5,       #00000000B        ;Output latches of P50 to P57 set to low level
      MOV   PM5,      #00000000B        ;P50 to P57 set as output ports
                                        ;P50 to P57: Unused

;-------------------------------------------------------------------------------
;     Port 6 settings
;-------------------------------------------------------------------------------
      MOV   P6,       #00000000B        ;Output latches of P60 to P67 set to
                                        ;low level
      MOV   PM6,      #00000000B        ;P60 to P67 set as output ports
                                        ;P60: Output SCL0 for setting the
                                        ;     registers in the audio codec
                                        ;P61: Input and output SDA0 for
                                        ;     setting the registers in the audio codec
                                        ;P62 to P67: Unused

;-------------------------------------------------------------------------------
;     Port 7 settings
;-------------------------------------------------------------------------------
      MOV   P7,       #00000000B        ;Output latches of P70 to P77 set to
                                        ;low level
      MOV   PM7,      #00000011B        ;P70 and P71 set as input ports and
                                        ;P72 to P77 set as output ports
                                        ;P70: Playback key input
                                        ;P71: Record key input
                                        ;P72: Output a signal to the
                                        ;     operation-in-progress indicator (LED)
```

```
                                        ;P73: Output a reset signal to the audio codec
                                        ;P74: Output CS0 to the EEPROM
                                        ;P75: Output CS1 to the EEPROM
                                        ;P76: Output CS2 to the EEPROM
                                        ;P77: Output CS3 to the EEPROM

;-------------------------------------------------------------------------------
;     Port 8 settings
;-------------------------------------------------------------------------------
      MOV   P8,         #00000000B      ;Output latches of P80 to P87 set to low level
      MOV   PM8,        #00000000B      ;P80 to P87 set as output ports
                                        ;P80 to P87: Unused

;-------------------------------------------------------------------------------
;     Port 11 settings
;-------------------------------------------------------------------------------
      MOV   P11,        #00000000B      ;Output latches of P110 and P111 set
                                        ;to low level
      MOV   PM11,       #11111100B      ;P110 and P111 set as output ports
                                        ;P110 and P111: Unused

;-------------------------------------------------------------------------------
;     Port 12 settings
;-------------------------------------------------------------------------------
      MOV   P12,        #00000000B      ;Output latch of P120 set to low level
      MOV   PM12,       #11111110B      ;P120 set as an output port
                                        ;P120: Unused

;-------------------------------------------------------------------------------
;     Port 13 settings
;-------------------------------------------------------------------------------
      MOV   P13,        #00000000B      ;Output latches of P130 and P131 set
                                        ;to low level
      MOV   PM13,       #11111100B      ;P131 set as an output port
                                        ;P130 and P131: Unused
;-------------------------------------------------------------------------------
;     Port 14 settings
;-------------------------------------------------------------------------------
      MOV   P14,        #00000000B      ;Output latches of P140 to P145 set
                                        ;to low level
      MOV   PM14,       #11000000B      ;P140 to P145 set as output ports
                                        ;P140 to P145: Unused

;-------------------------------------------------------------------------------
;     Port 15 settings
;-------------------------------------------------------------------------------
      MOV   P15,        #00000000B      ;Output latches of P150 to P157 set
                                        ;to low level
      MOV   PM15,       #00000000B      ;P150 to P157 set as output ports
                                        ;P150 to P157: Unused
```

<6>

```
;-----------------------------------------------------------------------------
;     Settings of the registers in the audio codec
;-----------------------------------------------------------------------------
;
;     The following operations are performed.
;     ·Supplying a clock to the audio codec
;     ·Setting a timer for inserting waits in the program
;     ·Outputting a reset signal to the audio codec
;     ·Using the I2C interface for setting the registers in the audio codec
;
;-----------------------------------------------------------------------------
```

(a)

```
      MOV      RPLAYMOD,#CRESET         ;Reset

      ; Timer array unit timer clock selection
      SET1     !TAU0EN                  ;Input clock supplied to the timer array unit
      MOV      !TPS0L,   #10010000B     ;Operation clock selection
                         ;||||++++------- CK00: fCLK
                         ;++++----------- CK01: fCLK/2^9

      ;----------------------------------------
      ; Clock supply to the audio codec
      ;----------------------------------------
      ; TO02 output (10 MHz)
      ;----------------------------------------
      MOVW     AX,       #0000000000000000B;Operation mode setting
                         ;||||||||||||++++- MD023 to MD020: Interval timer mode
                         ;||||||||||++----- <Fixed to 00>
                         ;||||||||++------- CIS021 and CIS020: Unused
                         ;|||||+++--------- STS022 to STS020: Only software trigger
                         ;|||||                               start enabled
                         ;||||+------------ MASTER02: Standalone operation
                         ;|||+------------- CSS02: Macro clock MCK specified by using
                         ;|||                       the CKS02 bit
                         ;|++-------------- <Fixed to 00>
                         ;+---------------- CKS02: Operation clock CK00 set by using
                         ;                          the PRS register
      MOVW     !TMR02,   AX
      MOVW     TDR02,    #0                 ;Interval setting: 10 MHz output (20 MHz)
      SET1     TOE0L.2                      ;TO02 output
      SET1     TS0L.2                       ;Operation starts (trigger bit)

      ;----------------------------------------
      ; Waits used in the program
      ;----------------------------------------
      ; Using TM01
      ;----------------------------------------
```

(b)

```
      MOVW     AX,       #1000000000000000B;Operation mode setting
                         ;||||||||||||++++- MD013 to MD010: Interval timer mode
                         ;||||||||||++----- <Fixed to 00>
                         ;||||||||++------- CIS011 and CIS010: Unused
                         ;|||||+++--------- STS012 to STS010: Only software trigger
                         ;|||||                               start enabled
                         ;||||+------------ MASTER01: Standalone operation
                         ;|||+------------- CSS01: Macro clock MCK specified by using
                         ;|||                       the CKS01 bit
                         ;|++-------------- <Fixed to 00>
                         ;+---------------- CKS01: Operation clock CK01 set by using
                         ;                          the PRS register
      MOVW     !TMR01,   AX
```

(c)

```
      ; Waiting for power supply to stabilize
      MOVW     TDR01,    #195*210/5-1       ;Interval set (210 ms)
      SET1     TS0L.1                       ;Timer operation starts
      CLR1     TMIF01                       ;INTTM00 interrupt request cleared
      BF       TMIF01,   $$

      ;Reset
      CLR1     P_RESETB
```

```
          ;Waiting about 5 us (calculated by 50 ns*5*Breg)
          MOV       B,        #5000/50/5
HRST550:
          DEC       B                       ;1clk
          BNZ       $HRST550                ;4clk(Z=0)
          SET1      P_RESETB

          ;Waiting for power supply to stabilize
          MOVW      TDR01,    #195*160/5-1 ;Interval set (160 ms)
          SET1      TS0L.1                  ;Timer operation starts
          CLR1      TMIF01                  ;INTTM00 interrupt request cleared
          BF        TMIF01,   $$
          SET1      TT0L.1                  ;Timer operation stopped

          ;-----------------------------------------
          ; Transmission of the values set to the
          ; registers in the audio codec
          ;-----------------------------------------
          ; Using IIC0
          ;-----------------------------------------
(d)       SET1      !IIC0EN                 ;Input clock of serial interface IIC0 supplied

          CLR1      IICE0                   ;Operation stopped

          CLR1      P_SCL                   ;SCL0 latch
          CLR1      PM_SCL                  ;SCL0 pin
          CLR1      P_SDA                   ;SDA0 pin
          CLR1      PM_SDA                  ;SDA0 pin

          MOV      IICX0,    #00000000B   ;Transfer clock selection
                             ;|||||||+---- CLX0
                             ;+++++++----- <0000000>
          MOV       IICCL0,   #00001110B
                             ;||||||++---- CL01-CL00(+CLX0): fCLK/96(first mode)
                             ;|||||+------ DFC0: Turn on the digital filter
                             ;||||+------- SMC0: Specify operation in the first mode
                             ;|||+-------- DAD0: <R>Detect the SDA0 pin level
                             ;||+--------- CLD0: <R>Detect the SCL0 pin level
                             ;++---------- <00>

          MOV       IICF0,    #00000011B   ;Communication reservation disable
                             ;|||||||+---- IICRSV: Disable communication reservation
                             ;||||||+----- STCEN: Enable initial start
                             ;||++++------ <Fixed to 0000>
                             ;|+---------- IICBSY: <R>IIC bus status flag
                             ;+----------- STCF: <R>STT0 clear flag


          MOV       IICC0,    #00001000B   ;Initial settings during master operation
                             ;|||||||+---- SPT0: Stop condition trigger
                             ;||||||+----- STT0: Start condition trigger
                             ;|||||+------ ACKE0: Control acknowledgment
                             ;||||+------- WTIM0: Control wait insertion and interrupt
                             ;||||                request issuance: 9 clocks
                             ;|||+-------- SPIE0: Disable issuing of interrupt requests
                             ;|||                 by detecting a stop condition
                             ;||+--------- WREL0: Do not cancel waiting
                             ;|+---------- LREL0: Save the communication: Normal
                             ;|                   operation
                             ;+----------- IICE0: Enable operation of I2C

          SET1      SPT0                    ;Stop condition set

(e)     ;Settings of the registers in the audio codec: Setting to turn on the system
          MOV       ES,       #HIGHW TSYSON   ;Higher 4 bits of the start address of the
                                              ;ROM table holding the register setting
                                              ;values
          MOVW      HL,       #LOWW TSYSON    ;Lower 16 bits
          MOVW      DE,       #LOWW TSYSONE   ;Lower 8 bits of the last address of the ROM
                                              ;table holding the register setting values
          CALL   !!SI2CWRITE                  ;I2C write processing

<7>       MOV    RPLAYMOD,    #CSTOP          ;Stopped
```

```
<8>  ---------------------------------------------------------------------------
     ;     Settings of the I2S bus interface used for the audio data of the
     ;     audio codec
     ;-------------------------------------------------------------------------------
     ;
     ;     The following operations are performed.
     ;     ·Setting the TO00 output and INTTM00 interrupt for outputting LRCLK
     ;     ·Setting CSI00 for transmitting and receiving data
     ;
     ;-------------------------------------------------------------------------------
            ;---------------------------------------
            ; LRCLK output settings
            ;---------------------------------------
            ; TO00 output (16 kHz)
            ;---------------------------------------
(a)         MOVW     AX,       #0000000000000000B;Operation mode setting
                               ;||||||||||||++++- MD003 to MD000: Interval timer mode
                               ;||||||||||++----- <Fixed to 00>
                               ;||||||||++------- CIS001 and CIS000: Unused
                               ;|||||+++--------- STS002 to STS000: Only software trigger
                               ;|||||                                start enabled
                               ;||||+------------ MASTER00: Standalone operation
                               ;|||+------------- CSS00: Macro clock MCK specified by using
                               ;|||                       the CKS00 bit
                               ;|++-------------- <Fixed to 00>
                               ;+---------------- CKS00: Operating clock CK00 set by using
                               ;                         the PRS register
            MOVW     !TMR00,   AX
            MOVW     TDR00,    #625-1             ;Interval set: 16 kHz output (32 kHz)


            CLR1     TMPR100                      ;Priority order set to the highest level
            CLR1     TMPR000

            ;---------------------------------------
            ; CSI00 settings
            ;---------------------------------------
(b)         SET1     !SAU0EN                      ;Input clock of the serial array unit
                                                  ;supplied

            NOP                                   ;Waiting
            NOP
            NOP
            NOP

            MOV      !SPS0L,   #00000000B         ;Operation clock selection: fCLK
                               ;||||++++---------- PRS003 to PRS000: fCLK
                               ;++++-------------- PRS013 to PRS010: Unused

            MOVW     SDR00,    #(12-1) shl 9      ;Bits 15 to 7: Transfer clock set (833 kHz)


            MOVW     AX,       #0000000000100001B ;Operation mode selected: CSI mode
                               ;||||||||||||||||+-- MD000: Buffer empty interrupt
                               ;||||||||||||||++--- MD002 and MD001: CSI mode
                               ;||||||||||+++----- <Fixed to 100>
                               ;|||||||||+-------- SIS000: Unused
                               ;||||||||+--------- <Fixed to 0>
                               ;|||||||+---------- STS00: Only software trigger enabled
                               ;|||||||                  (fixed in CSI mode)
                               ;||+++++---------- <Fixed to 00000>
                               ;|+--------------- CSS00: Transfer clock set to a clock
                               ;|                        obtained by dividing operation
                               ;|                        clock specified by using the CKS00
                               ;|                        bit
                               ;+---------------- CKS00: Operation clock set to prescaler
                               ;                         output clock CK00 set by using
                               ;                         the PRS register
      MOVW   !SMR00,   AX
```

```
        ;Initial data output (at the same time as setting CSI10)
        ;MOVW    AX,      #0000000100000001B ;Setting of initial outputs of the SO and SCK pins
        ;                 ;||||||||++++++++-- SO0n to SO00: Used for transmission
        ;                 ;++++++++---------- CKOn to CKO0: Serial clock output of channels
        ;                 ;                                 n to 0

        ;MOVW    !SO0,    AX

        SET1     SOE0L.0                     ;Output enabled

        SET1     P_SCK00                     ;SCK00 latch: High level
        CLR1     PM_SCK00                    ;SCK00 pin output set
        SET1     PM_SI00                     ;SI00 pin input set
        SET1     P_SO00                      ;SO00 latch: High level
        CLR1     PM_SO00                     ;SO00 pin output set

        SET1     SS0L.0                      ;CSI00 operation starts (trigger bit)
        CLR1     CSIIF00                     ;INTCSI00 interrupt request cleared
        CLR1     CSIMK00                     ;INTCSI00 interrupt servicing enabled

        CLR1     CSIPR100                    ;Priority order set to next after INTTM00
        SET1     CSIPR000

      ;-----------------------------------------------------------------------------
      ;     Setting of the SPI interface for EEPROM
      ;-----------------------------------------------------------------------------
      ;
      ;     The following operations are performed.
      ;     ·Setting CSI10 for transmitting and receiving data
      ;     ·Setting DMA0 for successive reception and DMA1 for successive
      ;      transmission
      ;
      ;-----------------------------------------------------------------------------
<9>   OR       P_CS,    #11110000B          ;All EEPROMs deactivated

        ;----------------------------------------
        ; DMA0 settings (for CSI reception)
        ;----------------------------------------
(a)     SET1     DEN0                        ;Operation of DMA channel 0 enabled

        MOV      DSA0,    #044H               ;DMA SFR address: SDR02(SIO10)=0FFF44H
        MOV      DMC0,    #00001000B          ;Setting of transfer mode of DMA channel 0
                          ;||||++++---------- IFC03 to IFC00: DMA start source: INTCSI10
                          ;||||                               transfer complete interrupt
                          ;|||+-------------- DWAIT0: DMA transfer suspension: DMA
                          ;|||                        transfer performed according to
                          ;|||                        a DMA start request
                          ;|||                        (not suspended)
                          ;||+--------------- DS0: Transfer data size: 8 bits
                          ;|+---------------- DRS0: DMA transfer direction selected: SFR
                          ;|                        → Internal RAM
                          ;+----------------- STG0: Software trigger not operated

        ;-----------------------------------------------------
        ;  DMA1 settings
        ;  (for CSI transmission and setting dummy data
        ;  during CSI reception)
        ;-----------------------------------------------------
(b)     SET1     DEN1                        ; Operation of DMA channel 1 enabled

        MOV      DSA1,    #044H               ;DMA SFR address: SDR02(SIO10)=0FFF44H
        MOV      DMC1,    #01001000B          ;Setting of transfer mode of DMA channel 0
                          ;||||++++---------- IFC13 to IFC10: DMA start source: INTCSI10
                          ;||||                               transfer complete interrupt
                          ;|||+-------------- DWAIT1: DMA transfer suspension: DMA
                          ;|||                        transfer performed according to a
                          ;|||                        DMA start request (not suspended)
                          ;||+--------------- DS1: Transfer data size: 8 bits
                          ;|+---------------- DRS1: DMA transfer direction selected:
                          ;|                        Internal RAM →SFR
                          ;+----------------- STG1: Software trigger not operated
```

```
        ;---------------------------------------
        ; CSI10 settings
        ;---------------------------------------
(c)     MOVW      SDR02,    #(4-1) shl 9       ;Bits 15 to 7: Transfer clock set (2.5 MHz)

        MOVW      AX,       #0000000000100000B ;Operation mode selected: CSI mode
                            ;|||||||||||||||+-- MD020: Transfer complete interrupt
                            ;||||||||||||||++--- MD022 and MD021: CSI mode
                            ;||||||||||+++----- <Fixed to 100>
                            ;|||||||||+-------- SIS020: Unused
                            ;||||||||+--------- <Fixed to 0>
                            ;|||||||+---------- STS02: Only software trigger enabled
                            ;|||||||                  (fixed in CSI mode)
                            ;||+++++----------- <Fixed to 00000>
                            ;|+---------------- CSS02: Transfer clock set to a clock
                            ;|                         obtained by dividing operation
                            ;|                         clock MCK as specified by using
                            ;|                         the CKS00 bit
                            ;+----------------- CKS02: Operation clock set to
                            ;                          prescaler output clock
                            ;                          CK00 set by using the PRS register
        MOVW      !SMR02,   AX

        MOVW      AX,       #1011000000000111B; Communication format setting
                            ;||||||||||||+++-- DLS022 to DLS020: 8-bit data length
                            ;||||||||||||+----- <Fixed to 0>
                            ;||||||||||++------ SLC021 and SLC020: Unused (fixed to 0)
                            ;|||||||||+-------- <Fixed to 0>
                            ;||||||||+--------- DIR02: Input and output performed MSB
                            ;||||||||                  first
                            ;||||||++---------- PTC021 and PTC020: Unused (fixed to 00)
                            ;|||||+------------ EOC02: Unused (fixed to 0)
                            ;||||+------------- <Fixed to 0>
                            ;||++-------------- CKP02/DAP02: Phases of data and clock in
                            ;||                             CSI mode selected
                            ;||                 [11 selected]
                            ;++---------------- TXE00/RXE00: Only transmission performed
        MOVW      !SCR02,   AX

        ; Initial data output
        MOVW      AX,       #0000000100000001B;Setting of initial outputs of the SO and SCK pins
                            ;||||||||++++++++- SO0n to SO00: Used for transmission
                            ;++++++++--------- CKOn to CKO0: Serial clock output of
                            ;                              channels n to 0
        MOVW      !SO0,     AX
        SET1      SOE0L.2                      ;Output enabled

        SET1      P_SCK10                      ;SCK00 latch: High level
        CLR1      PM_SCK10                     ;SCK00 pin output set
        SET1      PM_SI10                      ;SI00 pin input set
        SET1      P_SO10                       ;SO00 latch: High level
        CLR1      PM_SO10                      ;SO00 pin output set

        SET1      SS0L.2                       ;CSI10 operation starts (trigger bit)

        ;---------------------------------------
        ; Erasing all EEPROMs (all 0FFH)
        ;---------------------------------------
(d)     ;Enabling writing to an EEPROM
        AND       P_CS,     #00001111B         ;All EEPROMs selected
        CLR1      CSIIF10
        MOV       SIO10,    #00000110B         ;Write enable (WREN) instruction (INTCSI10
                                               ;occurs)
        BF        CSIIF10,$$
        OR        P_CS,     #11110000B         ;Setting completed

        ;Erasing all EEPROMs
        AND       P_CS,     #00001111B         ;All EEPROMs set
        CLR1      CSIIF10
        MOV       SIO10,    #11000111B         ;All CEs erased (INTCSI10 occurs)
        BF        CSIIF10,$$
        OR        P_CS,     #11110000B         ;Setting completed

        SET1      ST0L.2                       ;CSI10 operation stopped (trigger bit)
        SET1      CSIMK10                      ;INTCSI10 interrupt servicing disabled
```

```
        ;-----------------------------------------------------------------------------
        ;  Starting key retrieval
        ;-----------------------------------------------------------------------------
<10>    MOV      KRM,      #00000011B         ;KR0 and KR1 enabled
        CLR1     KRIF                         ;INTKR interrupt request cleared
        SET1     KRMK                         ;INTKR interrupt servicing disabled

<11>    SET1     P_LED                        ;Operation-in-progress indicator turned off
                                              ;(key input possible)

        ;-----------------------------------------------------------------------------
        ;  Enabling interrupts
        ;-----------------------------------------------------------------------------
<12>       EI
          <13>
```

The operations performed when the initial settings for the peripheral hardware to be used are specified in C language are similar to those performed when the settings are specified in assembly language.
In C language, the initial settings can be performed earlier by generating the hdwinit function.
The hdwinit function is a function used to perform the initial settings for peripheral devices (sfr) and be generated by the user as required.

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


      Initial settings of the peripheral hardware to be used


******************************************************************************/
void hdwinit(void)
{

/*----------------------------------------------------------------------------
      Disabling interrupts
----------------------------------------------------------------------------*/
      DI();
/*----------------------------------------------------------------------------
      Clock frequency settings
----------------------------------------------------------------------------
      Setting so that operations can be performed using the 20 MHz X1 oscillator
----------------------------------------------------------------------------*/
      CMC = 0b01000001;        /* Clock operation mode */
      /*      |||||||+----------- AMPH: 10 MHz<fMX≤20 MHz */
      /*      ||||+++------------ <000> */
      /*      |||+--------------- OSCSELS: P123 and P124 pins set as input ports */
      /*      ||+---------------- <0> */
      /*      ++----------------- EXCLK/OSCSEL: X1 oscillation mode (20 MHz) */

      CSC = 0b01000000;        /* Clock operation status control */
      /*      |||||||+----------- HIOSTOP: Internal high-speed oscillator operated */
      /*      ||+++++------------ <00000> */
      /*      |+----------------- XTSTOP: XT1 oscillator stopped */
      /*      +------------------ MSTOP: X1 oscillator operated */

      OSMC = 0b00000001;       /* Operation speed mode */
      /*       |||||||+---------- FSEL: Operated at a frequency exceeding 10 MHz */
      /*       +++++++----------- <00000> */

      OSTS = 0b00000101;       /* Oscillation stabilization time: 2^15/fX */

      while(!OSTC.2)  {}       /* Waiting for clock oscillation to stabilize */

      CKC = 0b00011000;        /* Clock selection */
      /*      |||||+++----------- MDIV2 to MDIV0: CPU/peripheral hardware clock (fCLK) = fMX */
      /*      ||||+-------------- <1> */
      /*      |||+--------------- MCM0: High-speed system clock (fMX) */
      /*      ||+---------------- <R> */
      /*      |+----------------- CSS: CSS: Main system clock (fMAIN) = fCLK */
      /*      +------------------ <R> */

/*----------------------------------------------------------------------------
      Port 0 settings
----------------------------------------------------------------------------*/
      P0  = 0b00011010;        /* Output latches of P00, P02, P05, and P06 set to low
                               /* level and those of P01, P03, and P04 set to high
                                  level */
      PM0 = 0b10000000;        /* P00 to P06 set as output ports */
                               /* P01: Output LRCLK to the audio codec */
                               /* P02: Output SO10 to the EEPROM */
                               /* P03: Input SI10 from the EEPROM */
                               /* P04: Output SCK10 to the EEPROM */
                               /* P00, P05, and P06: Unused */
```

```
/*-----------------------------------------------------------------------------
      Port 1 settings
-----------------------------------------------------------------------------*/
      P1  = 0b00000000;        /* Output latches of P10 to P17 set to low level */
      PM1 = 0b00000000;        /* P10 to P17 set as output ports */
                               /* P10: Output SCK00 to the audio codec */
                               /* P11: Input SI00 from the audio codec */
                               /* P12: Output SO00 to the audio codec */
                               /* P13 to P17: Unused */

/*-----------------------------------------------------------------------------
      Port 2 settings
-----------------------------------------------------------------------------*/
      P2  = 0b00000000;        /* Output latches of P20 to P27 set to low level */
      PM2 = 0b00000000;        /* P20 to P27 set as output ports */
                               /* P20 to P27: Unused */

/*-----------------------------------------------------------------------------
      Port 3 settings
-----------------------------------------------------------------------------*/
      P3  = 0b00000000;        /* Output latches of P30 and P31 set to low level */
      PM3 = 0b11111100;        /* P30 and P31 set as output ports */
                               /* P30 and P31: Unused */

/*-----------------------------------------------------------------------------
      Port 4 settings
-----------------------------------------------------------------------------*/
      P4  = 0b00000000;        /* Output latches of P40 to P47 set to low level */
      PM4 = 0b00000000;        /* P40 to P47 set as output ports */
                               /* P40 to P47: Unused */

/*-----------------------------------------------------------------------------
      Port 5 settings
-----------------------------------------------------------------------------*/
      P5  = 0b00000000;        /* Output latches of P50 to P57 set to low level */
      PM5 = 0b00000000;        /* P50 to P57 set as output ports */
                               /* P50 to P57: Unused */

/*-----------------------------------------------------------------------------
      Port 6 settings
-----------------------------------------------------------------------------*/
      P6  = 0b00000000;        /* Output latches of P60 to P67 set to low level */
      PM6 = 0b00000000;        /* P60 to P67 set as output ports */
                               /* P60: Output SCL0 for setting the registers in the
                                       audio codec */
                               /* P61: Input and output SDA0 for setting the registers
                                       in the audio codec */
                               /* P62 to P67: Unused */

/*-----------------------------------------------------------------------------
      Port 7 settings
-----------------------------------------------------------------------------*/
      P7  = 0b00000000;        /* Output latches of P70 to P77 set to low level */
      PM7 = 0b00000011;        /* P70 and P71 set as input ports and P72 to P77 set as
                                  output ports */
                               /* P70: Playback key input */
                               /* P71: Record key input */
                               /* P72: Output a signal to the operation-in-progress
                                       Indicator (LED) */
                               /* P73: Output a reset signal to the audio codec */
                               /* P74: Output CS0 to the EEPROM */
                               /* P75: Output CS1 to the EEPROM */
                               /* P76: Output CS2 to the EEPROM */
                               /* P77: Output CS3 to the EEPROM */

/*-----------------------------------------------------------------------------
      Port 8 settings
-----------------------------------------------------------------------------*/
      P8  = 0b00000000;        /* Output latches of P80 to P87 set to low level */
      PM8 = 0b00000000;        /* P80 to P87 set as output ports */
                               /* P80 to P87: Unused */
```

```
/*-----------------------------------------------------------------------------
      Port 11 settings
-----------------------------------------------------------------------------*/
      P11  = 0b00000000;       /* Output latches of P110 and P111 set to low level */
      PM11 = 0b11111100;       /* P110 and P111 set as output ports */
                               /* P110 and P111: Unused */

/*-----------------------------------------------------------------------------
      Port 12 settings
-----------------------------------------------------------------------------*/
      P12  = 0b00000000;       /* Output latch of P120 set to low level */
      PM12 = 0b11111110;       /* P120 set as an output port */
                               /* P120: Unused */

/*-----------------------------------------------------------------------------
      Port 13 settings
-----------------------------------------------------------------------------*/
      P13  = 0b00000000;       /* Output latches of P130 and P131 set to low level */
      PM13 = 0b11111100;       /* P131 set as an output port */
                               /* P130 and P131: Unused */

/*-----------------------------------------------------------------------------
      Port 14 settings
-----------------------------------------------------------------------------*/
      P14  = 0b00000000;       /* Output latches of P140 to P145 set to low level */
      PM14 = 0b11000000;       /* P140 to P145 set as output ports */
                               /* P140 to P145: Unused */

/*-----------------------------------------------------------------------------
      Port 15 settings
-----------------------------------------------------------------------------*/
      P15  = 0b00000000;       /* Output latches of P150 to P157 set to low level */
      PM15 = 0b00000000;       /* P150 to P157 set as output ports */
                               /* P150 to P157: Unused */

/*-----------------------------------------------------------------------------
      Settings of the registers in the audio codec
-----------------------------------------------------------------------------

      The following operations are performed.
      ·Supplying a clock to the audio codec
      ·Setting a timer for inserting waits in the program
      ·Outputting a reset signal to the audio codec
      ·Using the I2C interface for setting the registers in the audio codec

-----------------------------------------------------------------------------*/

      /* Timer array unit timer clock selection */
      TAU0EN = 1;              /* Input clock supplied to the timer array unit */
      TPS0L = 0b10010000;      /* Operation clock selection */
      /*        ||||++++--------- CK00: fCLK */
      /*        ++++------------- CK01: fCLK/2^9 */

      /*---------------------------------------
      Clock supply to the audio codec
      ---------------------------------------
      TO02 output (10 MHz)
      ---------------------------------------*/
      TMR02 = 0b0000000000000000;/* Operation mode setting */
      /*        ||||||||||||++++--- MD023 to MD020: Interval timer mode */
      /*        ||||||||||++-------- <Fixed to 00> */
      /*        ||||||||++---------- CIS021 and CIS020: unused */
      /*        |||||+++------------ STS022-020: Only software trigger start enabled */
      /*        ||||+--------------- MASTER02: Standalone operation */
      /*        |||+---------------- CSS02: Macro clock MCK specified by using
      /*        |||                         the CKS02 bit */
      /*        |++----------------- <Fixed to 00> */
      /*        +------------------- CKS02: Operation clock CK00 set by using
      /*                                    the PRS register */

      TDR02 = 0;               /* Interval setting: 10 MHz output (20 MHz) */
      TOE0L.2 = 1;             /* TO02 output */
```

```
TS0L.2 = 1;              /* Operation starts (trigger bit) */

/*-------------------------------------
Waits used in the program
---------------------------------------
Using TM01
---------------------------------------*/

TMR01 = 0b1000000000000000;/* Operation mode setting */
/*        |||||||||||||++++--- MD013 to MD010: Interval timer mode */
/*        ||||||||||||++------- <Fixed to 00> */
/*        ||||||||||++--------- CIS011 and CIS010: Unused */
/*        |||||+++----------- STS012 to STS010: Only software trigger start
/*        |||||                                  enabled */
/*        ||||+-------------- MASTER01: Standalone operation */
/*        |||+--------------- CSS01: Macro clock MCK specified by using
/*        |||                        the CKS01 bit */
/*        |++---------------- <Fixed to 00> */
/*        +------------------ CKS01: Operation clock CK01 set by using
/*                                   the PRS register*/

/* Waiting for power supply to stabilize */
TDR01 = 195*210/5-1;     /* Interval set (210 ms) */
TS0L.1 = 1;              /* Timer operation starts */
TMIF01 = 0;              /* INTTM00 interrupt request cleared */
while(!TMIF01)  {}

/* Reset */
P_RESETB = 0;
/* Waiting about 5 us */
TDR01 = 0;               /* Interval set */
TS0L.1 = 1;              /* Timer operation starts */
TMIF01 = 0;              /* INTTM00 interrupt request cleared */
while(!TMIF01)  {}
P_RESETB = 1;

/* Waiting for power supply to stabilize */
TDR01 = 195*160/5-1;     /* Interval set (160 ms) */
TS0L.1 = 1;              /* Timer operation starts */
TMIF01 = 0;              /* INTTM00 interrupt request cleared */
while(TMIF01)   {}
TT0L.1 = 1;              /* Timer operation stopped */

/*-------------------------------------
Transmission of the values set to
the registers in the audio codec
---------------------------------------
Using IIC0
---------------------------------------*/
IIC0EN = 1;              /* Input clock of serial interface IIC0 supplied */

IICE0 = 0;               /* Operation stopped */

P_SCL = 0;               /* SCL0 latch */
PM_SCL = 0;              /* SCL0 pin */
P_SDA = 0;               /* SDA0 pin */
PM_SDA = 0;              /* SDA0 pin */

IICX0 = 0b00000000;      /* Transfer clock selection */
/*        |||||||+--------- CLX0 */
/*        +++++++---------- <0000000> */
IICCL0 = 0b00001110;
/*         ||||||++-------- CL01-CL00(+CLX0): fCLK/96(first mode) */
/*         |||||+---------- DFC0: Turn on the digital filter */
/*         ||||+----------- SMC0: Specify operation in the first mode */
/*         |||+------------ DAD0: <R>Detect the SDA0 pin level */
/*         ||+------------- CLD0: <R>Detect the SCL0 pin level */
/*         ++-------------- <00> */
```

```
      IICF0 = 0b00000011;      /* Communication reservation disable */
      /*        |||||||+--------- IICRSV: Disable communication reservation */
      /*        ||||||+---------- STCEN: Enable initial start */
      /*        ||++++----------- <Fixed to 0000> */
      /*        |+--------------- IICBSY: <R>IIC bus status flag */
      /*        +---------------- STCF: <R>STT0 clear flag */

      IICC0 = 0b00001000;      /* Initial settings during master operation */
      /*        |||||||+--------- SPT0: Stop condition trigger */
      /*        ||||||+---------- STT0: Start condition trigger */
      /*        |||||+----------- ACKE0: Control acknowledgment */
      /*        ||||+------------ WTIM0: Control wait insertion and interrupt
      /*        ||||                      request issuance: 9 clocks */
      /*        |||+------------- SPIE0: Disable issuing of interrupt requests by
      /*        |||                       detecting a stop condition */
      /*        ||+-------------- WREL0: Do not cancel waiting */
      /*        |+--------------- LREL0: Save the communication: Normal operation */
      /*        +---------------- IICE0: Enable operation of I2C */

      SPT0 = 1;                /* Stop condition set */

      /* Settings of the registers in the audio codec: Setting to turn on the system */
      fn_I2cWrite(&aSystemOnTbl[0][0],sizeof(aSystemOnTbl));

/*-----------------------------------------------------------------------------
      Settings of the I2S bus interface used for the audio data of the
      audio codec
-----------------------------------------------------------------------------

      The following operations are performed.
       ·Setting the TO00 output and INTTM00 interrupt for outputting LRCLK
       ·Setting CSI00 for transmitting and receiving data

-----------------------------------------------------------------------------*/
      /*--------------------------------------
       LRCLK output settings
      ---------------------------------------
       TO00 output (16 kHz)
      --------------------------------------*/
      TMR00 = 0b0000000000000000;/* Operation mode setting */
      /*        ||||||||||||++++--- MD003 to MD000: Interval timer mode */
      /*        ||||||||||++------- <Fixed to 00> */
      /*        ||||||||++--------- CIS001 and CIS000: Unused */
      /*        |||||+++----------- STS002 to STS000: Only software trigger start
      /*        |||||                                 enabled */
      /*        ||||+-------------- MASTER00: Standalone operation */
      /*        |||+--------------- CSS00: Macro clock MCK specified by using
      /*        |||                        the CKS00 bit */
      /*        |++---------------- <Fixed to 00> */
      /*        +------------------ CKS00: Operating clock CK00 set by using
      /*                                   the PRS register */
      TDR00 = 625-1;           /* Interval set: 16 kHz output (32 kHz) */

      TMPR100 = 0;             /* Priority order set to the highest level */
      TMPR000 = 0;

      /*--------------------------------------
       CSI00 settings
      --------------------------------------*/
      SAU0EN = 1;              /* Input clock of the serial array unit supplied */

      NOP();                   /* Waiting */
      NOP();
      NOP();
      NOP();

      SPS0L = 0b00000000;      /* Operation clock selection: fCLK */
      /*        ||||++++--------- PRS003 to PRS000: fCLK */
      /*        ++++------------- PRS013 to PRS010: Unused */

      SDR00 = (12-1) << 9;     /* Bits 15 to 7: Transfer clock set (833 kHz) */
```

```
      SMR00 = 0b0000000000100001;/* Operation mode selected: CSI mode */
      /*        |||||||||||||||+--- MD000: Buffer empty interrupt */
      /*        ||||||||||||||++---- MD002 and MD001: CSI mode */
      /*        |||||||||||+++------ <Fixed to 100> */
      /*        |||||||||+--------- SIS000: Unused */
      /*        ||||||||+---------- <Fixed to 0> */
      /*        |||||||+----------- STS00: Only software trigger enabled
      /*        |||||||                    (fixed in CSI mode) */
      /*        ||+++++------------ <Fixed to 00000> */
      /*        |+----------------- CSS00: Transfer clock set to a clock obtained
      /*        |                           by dividing operation clock MCK as
      /*        |                           specified by using the CKS00 bit */
      /*        +------------------ CKS00: Operation clock set to prescaler output
      /*                                    clock CK00 set by using the PRS register */

      /* Initial data output (at the same time as setting CSI10) */
      /* SO0 = 0b0000000100000001;/* Setting of initial outputs of the SO and SCK pins */
      /*         ||||||||++++++++--- SO0n to SO00: Used for transmission */
      /*         ++++++++----------- CKOn to CKO0: Serial clock output of channels n to 0 */

      SOE0L.0 = 1;             /* Output enabled */

      P_SCK00 = 1;             /* SCK00 latch: High level */
      PM_SCK00 = 0;            /* SCK00 pin output set */
      PM_SI00 = 1;             /* SI00 pin input set */
      P_SO00 = 1;              /* SO00 latch: High level */
      PM_SO00 = 0;             /* SO00 pin output set */

      SS0L.0 = 1;              /* CSI00 operation starts (trigger bit) */
      CSIIF00 = 0;             /* INTCSI00 interrupt request cleared */
      CSIMK00 = 0;             /* INTCSI00 interrupt servicing enabled */

      CSIPR100 = 0;            /* Priority order set to next after INTTM00 */
      CSIPR000 = 1;

/*-----------------------------------------------------------------------------
      Setting of the SPI interface for EEPROM
-----------------------------------------------------------------------------

      The following operations are performed.
      ·Setting CSI10 for transmitting and receiving data
      ·Setting DMA0 for successive reception and DMA1 for successive transmission

-----------------------------------------------------------------------------*/
      P_CS |= 0b011110000;     /* All EEPROMs deactivated */

      /*--------------------------------------
       DMA0 settings (for CSI reception)
      --------------------------------------*/
      DEN0 = 1;                /* Operation of DMA channel 0 enabled */

      DSA0 = 0x044;            /* DMA SFR address: SDR02(SIO10)=0FFF44H */
      DMC0 = 0b00001000;       /* Setting of transfer mode of DMA channel 0 */
      /*       ||||++++---------- IFC03 to IFC00: DMA start source: INTCSI10
      /*       ||||                                transfer complete interrupt */
      /*       |||+-------------- DWAIT0: DMA transfer suspension: DMA transfer
      /*       |||                         performed according to a DMA start request
      /*       |||                         (not suspended) */
      /*       ||+--------------- DS0: Transfer data size: 8 bits */
      /*       |+---------------- DRS0: DMA transfer direction selected: SFR →
      /*       |                         Internal RAM */
      /*       +----------------- STG0: Software trigger not operated */

      /*-------------------------------------------------
       DMA1 settings
       (for CSI transmission and setting dummy data
        during CSI reception)
      -------------------------------------------------*/
      DEN1 = 1;                /* Operation of DMA channel 1 enabled */

      DSA1 = 0x044;            /* DMA SFR address: SDR02(SIO10)=0FFF44H */
```

```
DMC1 = 0b01001000;       /* Setting of transfer mode of DMA channel 1 */
/*       ||||++++---------- IFC13 to IFC10: DMA start source: INTCSI10
/*       ||||                                transfer complete interrupt */
/*       |||+-------------- DWAIT1: DMA transfer suspension: DMA transfer
/*       |||                        performed according to a DMA start request
/*       |||                        (not suspended) */
/*       ||+--------------- DS1: Transfer data size: 8 bits */
/*       |+---------------- DRS1: DMA transfer direction selected: Internal
/*       |                        RAM→ SFR */
/*       +----------------- STG1: Software trigger not operated */

/*--------------------------------------
 CSI10 settings
--------------------------------------*/
SDR02 = (4-1) << 9;      /* Bits 15 to 7: Transfer clock set (2.5 MHz) */

SMR02 = 0b0000000000100000;/* Operation mode selected: CSI mode */
/*        ||||||||||||||||+--- MD020: Transfer complete interrupt */
/*        ||||||||||||||++---- MD022 and MD021: CSI mode */
/*        |||||||||||+++------ <Fixed to 100> */
/*        |||||||||+---------- SIS020: Unused */
/*        ||||||||+----------- <Fixed to 0> */
/*        |||||||+------------ STS02: Only software trigger enabled
/*        |||||||                     (fixed in CSI mode) */
/*        ||+++++------------- <Fixed to 00000> */
/*        |+------------------ CSS02: Transfer clock set to a clock obtained by
/*        |                           dividing operation clock MCK as specified
/*        |                           by using the CKS00 bit */
/*        +------------------- CKS02: Operation clock set to prescaler output
/*                                    clock CK00 set by using the PRS register */
SCR02 = 0b1011000000000111;/* Communication format setting */
/*        ||||||||||||||+++--- DLS022 to DLS020: 8-bit data length */
/*        ||||||||||||+------ <Fixed to 0> */
/*        ||||||||||++------- SLC021 and SLC020: Unused (fixed to 0) */
/*        |||||||||+---------- <Fixed to 0> */
/*        ||||||||+----------- DIR02: Input and output performed MSB first */
/*        ||||||++------------ PTC021 and PTC020: Unused (fixed to 00) */
/*        |||||+-------------- EOC02: Unused (fixed to 0) */
/*        ||||+--------------- <Fixed to 0> */
/*        ||++---------------- CKP02/DAP02: Phases of data and clock in CSI mode
/*        ||                                selected */
/*        ||                   [11 selected] */
/*        ++------------------ TXE00/RXE00: Only transmission performed */

/* Initial data output */
SO0 = 0b0000000100000001;   /* Setting of initial outputs of the SO and SCK pins */
/*      ||||||||++++++++------ SO0n to SO00: Used for transmission */
/*      ++++++++-------------- CKOn to CKO0: Serial clock output of channels n to 0 */
SOE0L.2 = 1;                /* Output enabled */

P_SCK10 = 1;                /* SCK00 latch: High level */
PM_SCK10 = 0;               /* SCK00 pin output set */
PM_SI10 = 1;                /* SI00 pin input set */
P_SO10 = 1;                 /* SO00 latch: High level */
PM_SO10 = 0;                /* SO00 pin output set */

SS0L.2 = 1;                 /* CSI10 operation starts (trigger bit) */

/*--------------------------------------
 Erasing all EEPROMs (all 0FFH)
--------------------------------------*/
/* Enabling writing to an EEPROM */
P_CS &= 0b00001111;         /* All EEPROMs selected */
CSIIF10 = 0;
SIO10 = 0b00000110;         /* Transmit data set (Write enable (WREN) instruction
                               INTCSI10 occurs) */
while(!CSIIF10) {}
P_CS |= 0b11110000;         /* Setting completed */

/* Erasing all EEPROMs */
P_CS &= 0b00001111;         /* All EEPROMs selected */
CSIIF10 = 0;
SIO10 = 0b11000111;         /* Transmit data set (All CEs erased INTCSI10 occurs) */
```

```
      while(!CSIIF10) {}
      P_CS |= 0b11110000;         /* Setting completed */

      ST0L.2 = 1;                 /* CSI10 operation stopped (trigger bit) */
      CSIMK10 = 1;                /* INTCSI10 interrupt servicing disabled */

/*-----------------------------------------------------------------------------
      Starting key retrieval
-----------------------------------------------------------------------------*/
      KRM = 0b00000011;           /* KR0 and KR1 enabled */
      KRIF = 0;                   /* INTKR interrupt request cleared */
      KRMK = 1;                   /* INTKR interrupt servicing disabled */

      P_LED = 1;                  /* Operation-in-progress indicator turned off (key input
                                     possible) */

/*-----------------------------------------------------------------------------
      Enabling interrupts
-----------------------------------------------------------------------------*/
      EI();
}
```

4.4 Main Processing

The following operations are performed in the main processing specified in assembly language.

<1> Key processing
<2> If the key processing above returns the judgment that the record key was pressed, key input is disabled and recording starts.
<3> If the key processing above returns the judgment that the playback key was pressed, key input is disabled and playback starts.
<4> If the operating status indicates that recording has finished, recording is terminated.
<5> If the operating status indicates that playback has finished, playback is terminated.
<6> The operating status is set to "stopped" and key input is enabled.
<7> The processing returns to <1>.

```
;*****************************************************************************
;
;
;
;        Main processing
;
;
;*****************************************************************************
MAIN_LOOP:
<1>     ;---------------------------------------
        ;  Key processing
        ;---------------------------------------
        BT       KRIF,     $LMAIN100       ;Key input? Yes,
        BR       LMAINRECE                 ; No,
LMAIN100:
        CLR1     KRIF                      ;INTKR interrupt request cleared

        CMP      RPLAYMOD, #CSTOP          ;Stopped?
        BZ       $LMAIN200                 ; Yes,
        BR       LMAINRECE                 ; No, (key disabled)
LMAIN200:
        MOV      A,        P_KEY
        AND      A,        #00000011B      ;Is key valid?
        SKNZ                               ; Yes,
        BR       LMAINRECE                 ; No, (key disabled: pressed multiple times)

        ;Key judged to have been pressed
        ;Waiting about 10 ms (chattering removed)
        MOVW     TDR01,    #195*10/5-1     ;Interval set (10 ms)
        SET1     TS0L.1                    ;Timer operation starts
        CLR1     TMIF01                    ;INTTM00 interrupt request cleared
        BF       TMIF01,   $$
        SET1     TT0L.1                    ;Timer operation stopped

        MOV      A,        P_KEY           ;KR port
        AND      A,        #00000011B
        CMP      A,        #00000011B      ;Is key valid?
        BNZ      $LMAIN400                 ; Yes,
        BR       LMAINRECE                 ; No, (key disabled: pressed multiple times)
LMAIN400:
        BF       P_RECKY,  $LMAINREC       ;Was record key pressed? Yes,
        BF       P_PLAYKY, $LMAINPLAY      ;Was playback key pressed? Yes,
        BR       LMAINRECE                 ; No,

        ;---------------------------------------
        ; Starting recording
<2>     ;---------------------------------------
LMAINREC:
        CLR1     P_LED                     ;Operation-in-progress indicator turned on
                                           ;(key input disabled)

        MOV      KRM,      #00000000B      ;KR0 and KR1 disabled
        MOV      RPLAYMOD, #CREC           ;Voice data recorded

        ;Settings of the registers in the audio codec: Recording set
        MOV      ES,       #HIGHW TRECON   ;Higher 4 bits of the start address of the
                                           ;ROM table holding the register
                                           ;setting values
        MOVW     HL,       #LOWW TRECON    ;Lower 16 bits
        MOVW     DE,       #LOWW TRECONE   ;Lower 8 bits of the last address of the
                                           ;ROM table holding the register setting
                                           ;values
        CALL     !!SI2CWRITE               ;I2C write processing

        ;Preparation of the I2S bus interface
        MOV      RCSI00CNT, #0             ;CSI00 reception counter
        MOVW     SDR00,    #(12 to 1) shl 9;Bits 15 to 7: Transfer clock set
                                           ;(833 kHz)
```

```
        MOVW    AX,     #0100000000000111B;Communication format setting
                        ;||||||||||||+++- DLS002 to DLS000: 8-bit data length
                        ;||||||||||||+---- <Fixed to 0>
                        ;||||||||||++----- SLC001 and SLC000: Unused (fixed to 0)
                        ;|||||||||+------- <Fixed to 0>
                        ;||||||||+-------- DIR00: Input and output performed MSB
                        ;||||||||                 first
                        ;||||||++--------- PTC001 and PTC000: Unused (fixed to 00)
                        ;|||||+----------- EOC00: Unused (fixed to 0)
                        ;||||+------------ <Fixed to 0>
                        ;||++------------- CKP00/DAP00: Phases of data and clock
                        ;||                             in CSI  mode selected
                        ;||               [00 selected]
                        ;++--------------- TXE00/RXE00: Only reception performed
        MOVW    !SCR00, AX

        MOV     TO0L,   #00000000B      ;Initial output set to low level
        SET1    TOE0L.0                 ;Operation of TO00 enabled by a count
                                        ;operation (LRCLK)
        CLR1    TMIF00                  ;INTTM00 interrupt request cleared
        CLR1    TMMK00                  ;INTTM00 interrupt servicing enabled

        ;Setting writing to an EEPROM
        MOVW    RI2SADDR,#RRECMEM1      ;Address to which the data received via
                                        ;I2S is saved
        CLR1    FI2SPAGE
        MOV     RP_CS,  #11100000B      ;EEPROM selected
        MOV     REEPSEQ, #CEEPSEQ_RESET ;EEPROM transfer sequence
        MOVW    REEPADDR,#0             ;EEPROM write address (REEPADDR*100H)

        MOVW    AX,     #1011000000000111B;Communication format setting
                        ;||||||||||||+++- DLS022 to DLS020: 8-bit data length
                        ;||||||||||||+---- <Fixed to 0>
                        ;||||||||||++----- SLC021 and SLC020: Unused (fixed to 0)
                        ;|||||||||+------- <Fixed to 0>
                        ;||||||||+-------- DIR02: Input and output performed MSB
                        ;||||||||                 first
                        ;||||||++--------- PTC021 and PTC020: Unused (fixed to 00)
                        ;|||||+----------- EOC02: Unused (fixed to 0)
                        ;||||+------------ <Fixed to 0>
                        ;||++------------- CKP02/DAP02: Phases of data and clock
                        ;||                             in CSI  mode selected
                        ;||               [11 selected]
                        ;++--------------- TXE00/RXE00: Only transmission
                        ;                               performed
        MOVW    !SCR02, AX

        ;Starting I2S operation
        SET1    SS0L.0                  ;CSI00 operation starts (trigger bit)
        CLR1    P_LRCLK                 ;LRCLK operation starts
        SET1    TS0L.0                  ;TO00 output operation starts (trigger bit)
        MOV     SIO00,  #0FFH           ;Dummy data set (INTCSI00 occurs)
        BR LMAINRET

        ;---------------------------------------
        ; Starting playback
        ;---------------------------------------
<3>
LMAINPLAY:
        CLR1    P_LED                   ;Operation-in-progress indicator turned  ON
                                        ;(key input disabled)

        MOV     KRM,    #00000000B      ;KR0 and KR1 disabled
        MOV     RPLAYMOD,#CPLAY_START   ;Preparing for playback started

        ;Settings of the registers in the audio codec: Playback set
        MOV     ES,     #HIGHW TPLAYON  ;Higher 4 bits of the start address of
                                        ;the ROM table holding the register setting
                                        ;values
        MOVW    HL,     #LOWW TPLAYON   ;Lower 16 bits
        MOVW    DE,     #LOWW TPLAYONE  ;Lower 8 bits of the last address of the
                                        ;ROM table holding the register setting
                                        ;values
        CALL    !!SI2CWRITE             ;I2C write processing
```

```
        ;Preparation of the I2S bus interface
        MOV      RCSI00CNT,#0              ;CSI00 transmission counter
        MOVW     SDR00,    #(12-1) shl 9   ;Bits 15 to 7: Transfer clock set (833 kHz)
        MOVW     AX,       #1000000000000111B;Communication format setting
                           ;||||||||||||+++- DLS002 to DLS000: 8-bit data length
                           ;||||||||||||+---- <Fixed to 0>
                           ;||||||||||++----- SLC001 and SLC000: Unused (fixed to 0)
                           ;||||||||||+------- <Fixed to 0>
                           ;||||||||+-------- DIR00: Input and output performed MSB
                           ;||||||||                 first
                           ;||||||++--------- PTC001 and PTC000: Unused (fixed to 00)
                           ;|||||+----------- EOC00: Unused (fixed to 0)
                           ;||||+------------ <Fixed to 0>
                           ;||++------------- CKP00/DAP00: Phases of data and clock
                           ;||                              in CSI mode selected
                           ;||                [00 selected]
                           ;++--------------- TXE00/RXE00: Only transmission
                           ;                               performed
        MOVW   !SCR00,     AX

        MOV    TO0L,       #00000000B      ;Initial output set to low level
        SET1   TOE0L.0                     ;Operation of TO00 enabled by a count
                                           ;operation
        CLR1   TMIF00                      ;INTTM00 interrupt request cleared
        CLR1   TMMK00                      ;INTTM00 interrupt servicing enabled

        ;Setting reading from an EEPROM (writing set at first)
        MOV    A,          P_CS            ;EEPROM selected
        AND    A,          #00001111B
        OR     A,          #11100000B
        MOV    P_CS,       A
        MOV    R  P_CS,       #10110000B   ;EEPROM selected (enabled after 2 pages are
                                           ;read)
        SET1   FI2SPAGE                    ;Data of the EEPROM set to be saved to page 1
        MOV    REEPSEQ,    #CEEPSEQ_INST   ;Instruction bytes transmitted
        MOVW   REEPADDR,   #0              ;EEPROM read address (REEPADDR*100H)
        MOVW   AX,         #1011000000000111B;Communication format setting
                           ;||||||||||||+++- DLS022 to DLS020: 8-bit data length
                           ;||||||||||||+---- <Fixed to 0>
                           ;||||||||||++----- SLC021 and SLC020: Unused (fixed to 0)
                           ;||||||||||+------- <Fixed to 0>
                           ;||||||||+-------- DIR02: Input and output performed MSB
                           ;||||||||                 first
                           ;||||||++--------- PTC021 and PTC020: Unused (fixed to 00)
                           ;|||||+----------- EOC02: Unused (fixed to 0)
                           ;||||+------------ <Fixed to 0>
                           ;||++------------- CKP02/DAP02: Phases of data and clock in
                           ;||                              CSI mode selected
                           ;||                [11 selected]
                           ;++--------------- TXE00/RXE00: Only transmission performed
        MOVW   !SCR02,     AX

        ;Starting reading from the EEPROM
        CLR1   CSIIF10                     ;INTCSI10 interrupt request cleared
        CLR1   CSIMK10                     ;INTCSI10 interrupt servicing enabled
        SET1   SS0L.2                      ;CSI10 operation starts (trigger bit)
        MOV    SIO10,      #00000011B      ;Transmit data (reading specified) (INTCSI10
                                           ;occurs)
        BR     LMAINRET

        ;----------------------------------------
        ; Recording end settings
<4>     ;----------------------------------------
LMAINRECE:
        CMP    RPLAYMOD,   #CREC_END       ;Recording finished?
        BNZ    $LMAINPLAYE                 ; No,

        ;Settings of the registers in the audio codec: Setting termination of
        ;recording
        MOV    ES,         #HIGHW TRECOFF  ;Higher 4 bits of the start address of the
                                           ;ROM table holding the register setting
                                           ;values
        MOVW   HL,         #LOWW TRECOFF   ;Lower 16 bits
```

```
          MOVW   DE,          #LOWW TRECOFFE  ;Lower 8 bits of the last address of the ROM
                                              ;table holding the register setting values
          CALL   !!SI2CWRITE                  ;I2C write processing
          BR     LMAINSTOP
<5>
          ;---------------------------------------
          ; Playback end settings
          ;---------------------------------------
LMAINPLAYE:
          CMP    RPLAYMOD,    #CPLAY_END      ;Playback finished?
          BNZ    $LMAINRET                   ; No,

          ;Settings of the registers in the audio codec: Setting termination of playback
          MOV    ES,          #HIGHW TPLAYOFF ;Higher 4 bits of the start address of the
                                              ;ROM table holding the register setting
                                              ;values
          MOVW   HL,          #LOWW TPLAYOFF  ;Lower 16 bits
          MOVW   DE,          #LOWW TPLAYOFFE ;Lower 8 bits of the last address of the ROM
                                              ;table holding the register setting values
          CALL   !!SI2CWRITE                  ;I2C write processing
LMAINSTOP:
<6>
          MOV    KRM,         #00000011B      ;KR0 and KR1 enabled
          MOV    RPLAYMOD,    #CSTOP          ;Stopped
          SET1   P_LED                        ;Operation-in-progress indicator turned off
                                              ;(key input possible)
LMAINRET:
<7>
          BR     MAIN_LOOP
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

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


      Main processing


*****************************************************************************/
void main(void)
{
   ucPlayMode = CSTOP;      /* Stopped */

   while(1){
      /*--------------------------------------
      Key processing
      --------------------------------------*/
      if(KRIF){
      /* Key input */
         KRIF = 0;    /* INTKR interrupt request cleared */
         if(ucPlayMode==CSTOP){
         /* Stopped */
           if(P_KEY & 0b00000011){
           /* key valid */
              /* Key judged to have been pressed */
              /* Waiting about 10 ms (chattering removed) */
              TDR01 = 195*10/5-1;    /* Interval set (10 ms) */
              TS0L.1 = 1;      /* Timer operation starts */
              TMIF01 = 0;      /* INTTM00 interrupt request cleared */
              while(!TMIF01)   {}

              if((P_KEY & 0b00000011)&&!P_RECKY){ /* KR port */
              /* Record key valid */
              /*--------------------------------------
                Starting recording
              --------------------------------------*/
                  P_LED = 0;   /* Operation-in-progress indicator turned on (key input
                                  disabled) */
                  KRM = 0b00000000;     /* KR0 and KR1 disabled */
                  ucPlayMode = CREC;    /* Voice data recorded */

                  /* Settings of the registers in the audio codec: Recording set */
                   fn_I2cWrite(&aRecordOnTbl[0][0],sizeof(aRecordOnTbl));

                  /* Preparation of the I2S bus interface */
                  ucI2sByteCouter = 0;  /* CSI00 reception counter */
                  SDR00 = (12-1) << 9;  /* Bits 15 to 7: Transfer clock set (833 kHz) */
                  SCR00 = 0b0100000000000111;/* Communication format setting */
                  /*        |||||||||||||+++--- DLS002 to DLS000: 8-bit data length */
                  /*        |||||||||||||+------ <Fixed to 0> */
                  /*        ||||||||||++-------- SLC001 and SLC000: Unused (fixed to 0) */
                  /*        |||||||||+---------- <Fixed to 0> */
                  /*        ||||||||+----------- DIR00: Input and output performed
                  /*        ||||||||                    MSB first */
                  /*        ||||||++------------ PTC001 and PTC000: Unused (fixed to 00)*/
                  /*        |||||+-------------- EOC00: Unused (fixed to 0) */
                  /*        ||||+--------------- <Fixed to 0> */
                  /*        ||++---------------- CKP00/DAP00: Phases of data and  clock
                  /*        ||                                 in CSI mode selected */
                  /*        ||                   [00 selected] */
                  /*        ++----------------- TXE00/RXE00: Only reception performed */

                  TOE0L.0 = 1; /* Operation of TO00 enabled by a count operation (LRCLK) */
                  TMIF00 = 0;  /* INTTM00 interrupt request cleared */
                  TMMK00 = 0;  /* INTTM00 interrupt servicing enabled */
```

```
            /* Setting writing to an EEPROM */
            ucI2sAddress = &ucMemoryPage1[0];/* Address to which the data
                                                received via I2S is saved */
            bI2sMemoryPage = 0;
            ucEepromCs = 0b11100000; /* EEPROM selected */
            ucEepromSeq = CEEPSEQ_RESET;/* EEPROM transfer sequence */
            ushEepromAddress = 0; /* EEPROM write address */
                                  /* ushEepromAddress*100H) */
            SCR02 = 0b1011000000000111;/* Communication format setting */
            /*        |||||||||||||+++--- DLS022 to DLS020: 8-bit data length */
            /*        ||||||||||||+------ <Fixed to 0> */
            /*        |||||||||||++------- SLC021 and SLC020: Unused (fixed to 0) */
            /*        |||||||||+--------- <Fixed to 0> */
            /*        ||||||||+---------- DIR02: Input and output
            /*        ||||||||                   performed MSB first */
            /*        ||||||++----------- PTC021 and PTC020: Unused  (fixed to 00) */
            /*        |||||+-------------- EOC02: Unused (fixed to 0) */
            /*        ||||+--------------- <Fixed to 0> */
            /*        ||++---------------- CKP02/DAP02: Phases of data and
            /*        ||                                clock in CSI mode
            /*        ||                                selected */
            /*        ||                   [11 selected] */
            /*        ++----------------- TXE00/RXE00: Only transmission
            /*                                           performed */

            /* Starting I2S operation */
            SS0L.0 = 1;     /* CSI00 operation starts (trigger bit) */
            P_LRCLK = 0;    /* LRCLK operation starts */
            TS0L.0 = 1;     /* TO00 output operation starts (trigger bit) */
            SIO00 = 0x0FF;  /* Dummy data set (INTCSI00 occurs) */

        }
        else if((P_KEY & 0b00000011)&&!P_PLAYKY){ /* KR port */
        /* Playback key valid */
        /*--------------------------------------
         Starting playback
        --------------------------------------*/
            P_LED = 0;      /* Operation-in-progress indicator turned on
                               (key input disabled) */

            KRM = 0b00000000;        /* KR0 and KR1 disabled */
            ucPlayMode = CPLAY_START;   /* Preparing for playback started */

            /* Settings of the registers in the audio codec: Playback set */
            fn_I2cWrite(&aPlayOnTbl[0][0],sizeof(aPlayOnTbl));

            /* Preparation of the I2S bus interface */
            ucI2sByteCouter = 0;  /* CSI00 transmission counter */
            SDR00 = (12-1) << 9;  /* Bits 15 to 7: Transfer clock set (833 kHz) */
            SCR00 = 0b1000000000000111; /* Communication format setting */
            /*        |||||||||||||+++---- DLS002 to DLS000: 8-bit data Length */
            /*        ||||||||||||+------- <Fixed to 0> */
            /*        |||||||||||++-------- SLC001 and SLC000: Unused (fixed to 0) */
            /*        |||||||||+---------- <Fixed to 0> */
            /*        ||||||||+----------- DIR00: Input and output
            /*        ||||||||                    performed MSB first */
            /*        ||||||++------------ PTC001 and PTC000: Unused (fixed to 00) */
            /*        |||||+-------------- EOC00: Unused (fixed to 0) */
            /*        ||||+---------------- <Fixed to 0> */
            /*        ||++----------------- CKP00/DAP00: Phases of data and
            /*        ||                                 clock in CSI mode
            /*        ||                                 selected */
            /*        ||                    [00 selected] */
            /*        ++------------------ TXE00/RXE00: Only Transmission
            /*                                           performed */
```

```
                    TOE0L.0 = 1; /* Operation of TO00 enabled by a count operation */
                    TMIF00 = 0;  /* INTTM00 interrupt request cleared */
                    TMMK00 = 0;  /* INTTM00 interrupt servicing enabled */

                    /* Setting reading from an EEPROM (writing set at first) */
                    P_CS = ((P_CS & 0b00001111)|0b11100000);/* EEPROM selected */
                    ucEepromCs = 0b10110000;/* EEPROM selected (enabled after 2 pages are
                                               read) */
                    bI2sMemoryPage = 1;     /* Data of the EEPROM set to be saved to
                                              page 1 */
                    ucEepromSeq = CEEPSEQ_INST;/* Instruction bytes transmitted */
                    ushEepromAddress = 0;        /* EEPROM read address */
                                                 /* (ushEepromAddress*100H) */
                    SCR02 = 0b1011000000000111 ;/* Communication format setting */
                    /*        |||||||||||||+++---- DLS022 to DLS020: 8-bit data length */
                    /*        ||||||||||||+------- <Fixed to 0> */
                    /*        |||||||||||++-------- SLC021 and SLC020: Unused (fixed to 0) */
                    /*        ||||||||||+---------- <Fixed to 0> */
                    /*        |||||||||+----------- DIR02: Input and output
                    /*        |||||||                      performed MSB first */
                    /*        ||||||++------------ PTC021 and PTC020: Unused (fixed to 00) */
                    /*        |||||+-------------- EOC02: Unused (fixed to 0) */
                    /*        ||||+--------------- <Fixed to 0> */
                    /*        ||++---------------- CKP02/DAP02: Phases of data and
                    /*        ||                                 clock in CSI mode
                    /*        ||                                 selected  */
                    /*        ||                    [11 selected] */
                    /*        ++------------------ TXE00/RXE00: Only transmission
                    /*                                           performed */

                    /* Starting reading from the EEPROM */
                    CSIIF10 = 0; /* INTCSI10 interrupt request cleared */
                    CSIMK10 = 0; /* INTCSI10 interrupt servicing enabled */
                    SS0L.2 = 1;  /* CSI10 operation starts (trigger bit) */
                    SIO10 = 0b00000011;   /* Transmit data (reading specified)
                                             (INTCSI10 occurs) */
                }
            }
        }
    }
    /*--------------------------------------
     Recording end settings
    ---------------------------------------*/
    if(ucPlayMode==CREC_END){
    /* Recording finished */
        /* Settings of the registers in the audio codec: Setting termination of recording */
        fn_I2cWrite(&aRecordOffTbl[0][0],sizeof(aRecordOffTbl));

        KRM = 0b00000011;     /* KR0 and KR1 enabled */
        ucPlayMode = CSTOP;   /* Stopped */
        P_LED = 1;   /* Operation-in-progress indicator turned off (key input possible) */
    }
    /*--------------------------------------
     Playback end settings
    ---------------------------------------*/
    else if(ucPlayMode==CPLAY_END){
    /* Playback finished */
        /* Settings of the registers in the audio codec: Setting termination of playback */
        fn_I2cWrite(&aPlayOffTbl[0][0],sizeof(aPlayOffTbl));

        KRM =0b00000011;         /* KR0 and KR1 enabled */
        ucPlayMode = CSTOP;      /* Stopped */
        P_LED = 1;   /* Operation-in-progress indicator turned off (key input possible) */
    }
  }
}
```

4.5 INTTM00 Interrupt Servicing

The following operations are performed in the INTTM00 interrupt servicing in assembly language.

<1> The register bank is switched.

<2> Reception performed via CSI00 used for receiving data via the I²S bus interface is restarted and synchronized with LRCLK after the third byte is received during recording.

<3> Transmission performed via CSI00 used for transmitting data via the I²S bus interface is restarted and synchronized with LRCLK after the first byte is transmitted during playback.

<4> The address to which the data to be transmitted or received via the I2S bus interface is saved is updated.

```
;*****************************************************************************
;
;        INTTM00 interrupt servicing
;        (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and SDIN)
;
;-----------------------------------------------------------------------------
;
;        LRCLK output via the I2S bus interface and BCLK, SDOUT, and SDIN used for
;        transmitting and receiving data are synchronized by restarting the
;        operation of CSI00, which was stopped by INTCSI00 interrupt servicing,
;        When LRCLK changes from high to low level and from low to high level.
;
;                    ________                 ________________
;        LRCLK               |________________|                |________
;
;                    _________                __              __
;        BCLK                |||||||||||||||  |||||||||||||||  |||||||
;
;                    _________ _ _      _ _ __ _ _      _ _ __ _ _
;        SDOUT/SDIN          |_|_. . . _|_|  |_|_. . . _|_|  |_|_. .
;
;
;        INTTM00     ________|________________|________________|________
;
;        Register bank 1 used
;
;*****************************************************************************
IINTTM00:
         SEL         RB1

         CMP         RPLAYMOD, #CREC     ;During recording? No, (during playback)
<1>      BNZ         $HTM00PLAY
         ;---------------------------------------
         ; During recording
         ;---------------------------------------
<2>      ;Starting the operation of the I2S bus interface
         MOV         A,        #0FFH     ;Dummy data
         SET1        SS0L.0              ;CSI00 operation starts (trigger bit)
         XCH         A,        SIO00     ;Third byte received and dummy data set
                                         ;(INTCSI00 occurs)
         MOVW        HL,       RI2SADDR  ;Address to which the data received
                                         ;via I2S is saved
         MOV         [HL],     A         ;Data saved
         BR          HTM00RET

HTM00PLAY:
```

```
            ;-------------------------------------
            ; During playback
            ;-------------------------------------
<3>         ;Starting the operation of the I2S bus interface
            MOVW        HL,   RI2SADDR     ;Address to which the data transmitted via I2S
                                           ;is saved
            MOV         A,    [HL]         ;Transmit data (first byte)
            SET1        SS0L.0             ;CSI00 operation starts (trigger bit)
            MOV         SIO00,A            ;Transmit data set (INTCSI00 occurs)
HTM00RET:
<4>         INCW        RI2SADDR           ;Address to which the data transmitted or
                                           ;received via I2S is saved updated
            RETI
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

```
/*****************************************************************************
   INTTM00 interrupt servicing
   (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and SDIN)

-----------------------------------------------------------------------------

       LRCLK output via the I2S bus interface and BCLK, SDOUT, and SDIN used for
       transmitting and receiving data are synchronized by restarting the
       operation of CSI00, which was stopped by INTCSI00 interrupt servicing,
       when LRCLK changes from high to low level and from low to high level.

                 ________                 _______________
      LRCLK              |_______________|               |_______

                 _________                __             __
      BCLK                |||||||||||||  |||||||||||||  |||||||

                 _________ _ _     _ _ __ _ _     _ _ __ _ _
      SDOUT/SDIN          |_|_· · · _|_|  |_|_· · · _|_|  |_|_· ·


      INTTM00    ________|_______________|_______________|_______

      Register bank 1 used

*****************************************************************************/
__interrupt void fn_inttm00(void)
{
      if(ucPlayMode==CREC){
      /*--------------------------------------
       During recording
      --------------------------------------*/
      /* Starting the operation of the I2S bus interface */
         SS0L.0 = 1;                    /* CSI00 operation starts (trigger bit) */
         *ucI2sAddress = SIO00;         /* Third byte received and dummy data set (data
                                           saved to address) */
      SIO00 = 0x0FF;                    /* Dummy data set (INTCSI00 occurs) */
      }
      else{
      /*--------------------------------------
       During playback
      --------------------------------------*/
      /* Starting the operation of the I2S bus interface */
         SS0L.0 = 1;                    /* CSI00 operation starts (trigger bit) */
         SIO00 = *ucI2sAddress;         /* Transmit data (first byte) set (INTCSI00
                                           occurs) */
      }
      ucI2sAddress++;                   /* Address to which the data transmitted or
                                           received via I2S is saved updated */
}
```

4.6 INTCSI00 Interrupt Servicing

The following operations are performed in the INTCSI00 interrupt servicing in assembly language.

<1> The register bank is switched.
<2> Whether the address to which the received data is saved has reached one page is determined. If the address has reached one page, writing recorded data to an EEPROM is started.
<3> Reception of each byte (first and second bytes) via CSI00 during recording
<4> Transmission of each byte (second and third bytes) via CSI00 during playback
<5> Whether the reference address of the transmitted data has reached one page is determined. If the address has reached one page, the selection of the EEPROM is changed and reading data from an EEPROM is started.

```
;*****************************************************************************
;
;        INTCSI00 interrupt servicing
;        (using INTCSI00 for CSI transmission or reception)
;
;-----------------------------------------------------------------------------
;
;        Data is transmitted and received via the I2S bus interface between the
;        microcontroller and the audio codec.
;        During recording, CSI00 is used to save the received data.  Normally, data is
;        successively received by setting dummy data.  To achieve synchronization with
;        LRCLK, however, only reception is performed every three bytes and successive
;        reception operation is stopped.  The stopped operation is restarted by INTTM00
;        interrupt servicing and can be synchronized with LRCLK.  When the received
;        data reaches one page, writing to an EEPROM starts.  During playback, CSI00 is
;        used to successively transmit data, but successive transmission operation is
;        stopped every three bytes to achieve synchronization with LRCLK, similarly as
;        during recording.  The stopped operation is restarted by INTTM00 interrupt
;        servicing and can be synchronized with LRCLK.  When the transmitted data
;        reaches one page, reading data of a different page from the EEPROM starts.
;
;                        ________                              ______
;        LRCLK                  |______________________________|
;
;                        _________                             _
;        BCLK                    |||||||||||||||||||||||||||| ||||||
;
;                        _________ _      _ _      _ _      _ _ _
;        SDOUT/SDIN              |_. . . _|_. . . _|_. . . _| |_. .
;
;        INTCSI00        _________|________|________|__________|_____
;
;                                 1        2        3          1
;
;        Register bank 1 used
;
;*****************************************************************************
IINTCSI00:
<1>     SEL         RB1

        INC         RCSI00CNT          ;CSI00 reception counter updated

        CMP         RPLAYMOD, #CREC    ;During recording?
        BNZ         $HI2SPLAY          ; No,

        ;---------------------------------------
        ; During recording
        ;---------------------------------------
<2>     ;Determining the last position of the page
        MOVW        AX,       RI2SADDR     ;Address to which the data received via
                                           ;I2S is saved
        BT          FI2SPAGE, $HI2SR030
        CMPW        AX,       #RRECMEM1E   ;End of page 1?
        BC          $HI2SR100              ; No,
        MOVW        RI2SADDR, #RRECMEM2    ;Address to which the data received via
                                           ;I2S is saved:Page 2
        SET1        FI2SPAGE
        BR          HI2SR050
HI2SR030:
        CMPW        AX,       #RRECMEM2E   ;End of page 2?
        BC          $HI2SR100              ; No,
        MOVW        RI2SADDR, #RRECMEM1    ;Address to which the data received via
                                           ;I2S is saved:Page 1
        CLR1        FI2SPAGE
HI2SR050:
        ;Starting writing to the EEPROM
        MOV         REEPSEQ,  #CEEPSEQ_WREN;Write enable (WREN) signal transmitted
        MOV         A,        P_CS         ;EEPROM CS set
        AND         A,        #00001111B
        OR          A,        RP_CS
        MOV         P_CS,     A
        CLR1        CSIIF10                ;INTCSI10 interrupt request cleared
```

```
        CLR1        CSIMK10                 ;INTCSI10 interrupt servicing enabled
        SET1        SS0L.2                  ;CSI10 operation starts (trigger bit)
        MOV         SIO10,    #00000110B    ;Write enable (WREN) instruction (INTCSI10
                                            ;occurs)
HI2SR100:
        ;Buffer empty interrupt immediately after starting operation
<3>     CMP         RCSI00CNT,#1            ;Buffer empty interrupt immediately after
                                            ;starting reception?
        BNZ         $HI2SR200               ; No,
        MOV         SIO00,    #0FFH         ;Dummy data set
        BR          HI2SRET                 ; Yes, (ignored because the data is
                                            ;undefined)
HI2SR200:
        ;Reception of the first byte
        CMP         RCSI00CNT,#2            ;First byte received?
        BNZ         $HI2SR300               ; No,
        MOV         A,        #0FFH         ;Dummy data
        XCH         A,        SIO00         ;Dummy data set and data received
        BR          HI2SR500
HI2SR300:
        ;Reception of the second byte
        MOV         RCSI00CNT,#0            ;CSI00 reception counter initialized
        MOV         A,        SIO00         ;Data reception only
        ;BR         HI2SR500
HI2SR500:
        MOVW        HL,       RI2SADDR      ;Address to which the data received via
                                            ;I2S is saved
        MOV         [HL],     A             ;Received data saved
HI2SR800:
        INCW        RI2SADDR                ;Address to which the data received via
                                            ;I2S is saved updated
        BR          HI2SRET

        ;---------------------------------------
        ; During playback
        ;---------------------------------------
HI2SPLAY:
<4>     CMP         RPLAYMOD, #CPLAY        ;During playback?
        BNZ         $HI2SRET                ; No,

        CMP         RCSI00CNT,#3            ;Timing of synchronization with LRCLK?
                                            ;(Immediately after transmission of the
                                            ;third byte starts?)
        BNC         $HI2SP100

        MOVW        HL,       RI2SADDR      ;Address to which the data transmitted via
                                            ;I2S is saved
        MOV         A,        [HL]          ;Transmit data (second or third byte)
        MOV         SIO00,    A             ;Transmit data set (INTCSI00 occurs)
        INCW        RI2SADDR                ;Address to which the data transmitted via
                                            ;I2S is ;saved updated
        BR          HI2SP200
HI2SP100:
<5>     MOV         RCSI00CNT,#0            ;CSI00 transmission counter initialized
HI2SP200:
        MOVW        AX,       RI2SADDR      ;Address to which the data transmitted via
                                            ;I2S is saved
        BT          FI2SPAGE, $HI2SP300     ;Page 2 being transmitted?
        CMPW        AX,       #RRECMEM1E    ;End of page 1?
        BC          $HI2SRET                ; No,
        MOVW        RI2SADDR, #RRECMEM2     ;Address to which the data transmitted via
                                            ;I2S is saved: Page 2
        SET1        FI2SPAGE
        BR          HI2SP500
HI2SP300:
        CMPW        AX,       #RRECMEM2E    ;End of page 2?
        BC          $HI2SRET                ; No,
        MOVW        RI2SADDR, #RRECMEM1     ;Address to which the data transmitted via
                                            ;I2S is saved: Page 1
        CLR1        FI2SPAGE
HI2SP500:
        ;Starting reading from the EEPROM
        MOV         A,        P_CS          ;EEPROM selected
        AND         A,        #00001111B
```

```
        OR          A,        RP_CS
        MOV         P_CS,     A
        MOV         REEPSEQ,  #CEEPSEQ_INST;Instruction bytes transmitted
        CLR1        CSIIF10               ;INTCSI10 interrupt request cleared
        CLR1        CSIMK10               ;INTCSI10 interrupt servicing enabled
        SET1        SS0L.2                ;CSI10 operation starts (trigger bit)
        MOV         SIO10,    #00000011B  ;Transmit data (reading specified) set
                                          ;(INTCSI10 occurs)
HI2SRET:
        RETI
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

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

      INTCSI00 interrupt servicing
      (using INTCSI00 for CSI transmission or reception)

-----------------------------------------------------------------------------

         Data is transmitted and received via the I2S bus interface between the
         microcontroller and the audio codec.  During recording, CSI00 is used to
         save the received data.  Normally, data is successively received by
         setting dummy data.
         To achieve synchronization with LRCLK, however, only reception is
         Performed every three bytes and successive reception operation is stopped.
         The stopped operation is restarted by INTTM00 interrupt servicing and
         can be synchronized with LRCLK.  When the received data reaches one page,
         writing to an EEPROM starts.  During playback, CSI00 is used to
         successively transmit data, but successive transmission operation is
         stopped every three bytes to achieve synchronization with LRCLK,
         similarly as during recording.
         The stopped operation is restarted by INTTM00 interrupt servicing and
         can be synchronized with LRCLK.  When the transmitted data reaches one
         page, reading data of a different page from the EEPROM starts.

                     ________                              ______
         LRCLK               |____________________________|

                     _________                             _
         BCLK                 |||||||||||||||||||||||||||| ||||||

                     _________ _      _ _      _ _      _ _ _
         SDOUT/SDIN           |_. . . _|_. . . _|_. . . _| |_. .

         INTCSI00    _________|________|________|__________|_____

                              1        2        3          1

         Register bank 1 used

*****************************************************************************/
__interrupt void fn_intcsi00(void)
{
   ucI2sByteCouter++; /* CSI00 reception counter updated */

   if(ucPlayMode==CREC){
   /*-------------------------------------
    During recording
   -------------------------------------*/
       /* Determining the last position of the page */
       if(bI2sMemoryPage){
       /* Page 2 being transmitted */
         if(ucI2sAddress>=&ucMemoryPage2[3*2*42]){
         /* End of page 2 */
            ucI2sAddress = &ucMemoryPage1[0];/* Address to which the data
                                                received via I2S is saved: page 1 */
            bI2sMemoryPage = 0;

            /* Starting writing to the EEPROM */
            ucEepromSeq = CEEPSEQ_WREN; /* Write enable (WREN) signal transmitted */
            P_CS = (P_CS & 0b00001111)|ucEepromCs;   /* EEPROM CS set */
            CSIIF10 = 0;/* INTCSI10 interrupt request cleared */
            CSIMK10 = 0;/* INTCSI10 interrupt servicing enabled */
            SS0L.2 = 1;/* CSI10 operation starts (trigger bit) */
            SIO10 = 0b00000110;   /* Transmit data set (Write enable (WREN)
                                     instruction (INTCSI10 occurs) */
         }
       }
```

```
        else{
        /* Page 1 being transmitted */
          if(ucI2sAddress>=&ucMemoryPage1[3*2*42]){
          /* End of page 1 */
             ucI2sAddress = &ucMemoryPage2[0];/* Address to which the data received via
                                                I2S is saved:Page 2 */
             bI2sMemoryPage = 1;

             /* Starting writing to the EEPROM */
             ucEepromSeq = CEEPSEQ_WREN;    /* Write enable (WREN) signal transmitted */
             P_CS = (P_CS & 0b00001111)|ucEepromCs;   /* EEPROM CS set */
             CSIIF10 = 0; /* INTCSI10 interrupt request cleared */
             CSIMK10 = 0; /* INTCSI10 interrupt servicing enabled */
             SS0L.2 = 1;  /* CSI10 operation starts (trigger bit) */
             SIO10 = 0b00000110;   /* Transmit data set (Write enable (WREN) instruction
                                      INTCSI10 occurs) */
          }
        }

        if(ucI2sByteCouter==1){
        /* Buffer empty interrupt immediately after starting operation */
           SIO00 = 0x0FF; /* Dummy data set */
        }
        else{
             if(ucI2sByteCouter==2){
             /* Reception of the first byte */
                *ucI2sAddress = SIO00;   /* Data received via I2S received and saved */
                SIO00 = 0x0FF;           /* Dummy data set */
             }
             else/*if(ucI2sByteCouter==3)*/{
             /* Reception of the second byte */
                *ucI2sAddress = SIO00;   /* Data received via I2S received and saved
                                            (data reception only) */
                ucI2sByteCouter = 0;  /* CSI00 reception counter initialized */
             }
             ucI2sAddress++;
        }
    }
    else if(ucPlayMode==CPLAY){
    /*-------------------------------------
     During playback
    --------------------------------------*/
        if(ucI2sByteCouter<3){
        /* Transmission of the second or third byte */
             SIO00 = *ucI2sAddress;   /* Transmit data set (I2S Transmit
                                         data,INTCSI00 occurs) */
             ucI2sAddress++; /* Address to which the data transmitted via I2S is
                                saved updated */
        }
        else/*if(ucI2sByteCouter==3)*/{
        /* Timing of synchronization with LRCLK? (Immediately after transmission
           of the third byte starts) */
           ucI2sByteCouter = 0;      /* CSI00 transmission counter initialized */
        }

        if(bI2sMemoryPage){
        /* Page 2 being transmitted */
           if(ucI2sAddress>=&ucMemoryPage2[3*2*42]){
           /* End of page 2 */
              ucI2sAddress = &ucMemoryPage1[0];
              /* Address to which the data transmitted via I2S is saved: Page 1 */
              bI2sMemoryPage = 0;
              /* Starting reading from the EEPROM */
              P_CS = (P_CS & 0b00001111)|ucEepromCs;/* EEPROM selected */
```

```
            ucEepromSeq = CEEPSEQ_INST;/* Instruction bytes transmitted */
            CSIIF10 = 0;   /* INTCSI10 interrupt request cleared */
            CSIMK10 = 0;   /* INTCSI10 interrupt servicing enabled */
            SS0L.2 = 1; /* CSI10 operation starts (trigger bit) */
            SIO10 = 0b00000011;/* Transmit data (reading specified) set
                                   (INTCSI10 occurs) */
         }
      }
      else{
      /* Page 1 being transmitted */
         if(ucI2sAddress>=&ucMemoryPage1[3*2*42]){
         /* End of page 1 */
            ucI2sAddress = &ucMemoryPage2[0];
            /* Address to which the data transmitted via I2S is saved: Page 2 */
               bI2sMemoryPage = 1;
            /* Starting reading from the EEPROM */
            P_CS = (P_CS & 0b00001111)|ucEepromCs;/* EEPROM selected */
            ucEepromSeq = CEEPSEQ_INST;/* Instruction bytes transmitted */
            CSIIF10 = 0;   /* INTCSI10 interrupt request cleared */
            CSIMK10 = 0;   /* INTCSI10 interrupt servicing enabled */
            SS0L.2 = 1; /* CSI10 operation starts (trigger bit) */
            SIO10 = 0b00000011;/* Transmit data (reading specified) set
                                   (INTCSI10 occurs) */
         }
      }
   }
}
```

4.7 INTCSI10 Interrupt Servicing

The following operations are performed in the INTCSI10 interrupt servicing in assembly language.

<1> Multiple interrupts are enabled and the register bank is switched.
<2> Section for transmitting instruction bytes used for writing data to the EEPROMs during recording
<3> Section for transmitting the 24-bit address used for writing data to the EEPROMs during recording. The address is transmitted in three sections of one byte each.
<4> Timing of starting transmitting the data section used for writing data to the EEPROMs during recording. Transmission using DMA is started.
<5> Section for transmitting the 24-bit address used for reading data from the EEPROMs during playback. The address is transmitted in three sections of one byte each.
<6> Timing of starting to receive the data section used for reading data from the EEPROMs during playback. Reception using DMA is started.

```
;*****************************************************************************
;
;        INTCSI10 interrupt servicing
;        (using INTCSI10 for CSI transmission or reception)
;
;-----------------------------------------------------------------------------
;
;        During recording, instruction bytes and 24-bit addresses are transmitted
;        to an EEPROM, DMA channel 1 starts when transmitting data to the EEPROM
;        starts, and transmission continues in CSI10 single transmission mode.
;        During playback, 24-bit addresses are transmitted to an EEPROM, DMA
;        channels 0 and 1 start when receiving data from the EEPROM starts, and
;        reception continues in CSI10 single reception mode.  However, DMA
;        channel 1 is used for setting dummy data.
;
;        Register bank 1 used
;
;*****************************************************************************
IINTCSI10:
<1>     EI                                  ;Multiple interrupts enabled
        SEL       RB2

        INC       REEPSEQ                   ;EEPROM transfer sequence updated

        CMP       RPLAYMOD, #CREC           ;During recording? No, (during playback)
        BZ        $HCSI1REC
        BR        HCSI1PLAY

HCSI1REC:
        ;---------------------------------------
        ; During recording
        ;---------------------------------------
<2>     ;Instruction byte transmission
        CMP       REEPSEQ,#CEEPSEQ_INST     ;Instruction byte transmission timing?
        BNZ       $HCSI1100                 ; No,
        OR        P_CS,     #11110000B      ;Non-active
        NOP
        MOV       A,        P_CS            ;EEPROM selected
        AND       A,        #00001111B
        OR        A,        RP_CS
        MOV       P_CS,     A
        MOV       A,        #00000010B      ;Writing specified
        BR        HCSI1650                  ;Data transmitted
HCSI1100:
<3>     ;Transmission of the higher bits of the 24-bit address
        CMP       REEPSEQ,  #CEEPSEQ_ADDRH  ;Transmitting the higher bits of the 24-bit
                                            ;address?
        BNZ       $HCSI1200                 ; No,
        MOV       A,        REEPADDR+1
        BR        HCSI1650                  ;Data transmitted
HCSI1200:
        ;Transmission of the middle bits of the 24-bit address
        CMP       REEPSEQ,  #CEEPSEQ_ADDRM  ;Transmitting the middle bits of the 24-bit
                                            ;address?
        BNZ       $HCSI1300                 ; No,
        MOV       A,        REEPADDR
        BR        HCSI1650                  ;Data transmitted
HCSI1300:
        ;Transmission of the lower bits of the 24-bit address
        CMP       REEPSEQ,  #CEEPSEQ_ADDRL  ;Transmitting the lower bits of the 24-bit
                                            ;address?
        BNZ       $HCSI1400                 ; No,
        MOV       A,        #000H
        BR        HCSI1650                  ;Data transmitted
HCSI1400:
<4>     ;Starting data transmission
        ;Starting DMA transfer
        BT        FI2SPAGE,$HCSI1430        ;Is I2S transmitting page 2?
        MOVW      DRA1,     #RRECMEM2+1     ;Transmit buffer
        MOV       A,        !RRECMEM2       ;First transmit data
        BR        HCSI1450
```

```
HCSI1430:
        MOVW    DRA1,   #RRECMEM1+1     ;Transmit buffer
        MOV     A,      !RRECMEM1       ;First transmit data
HCSI1450:
        MOVW    DBC1,   #3*2*42-1       ;One page
        SET1    DST1                    ;DMA1 transfer enabled (for CSI10
                                        ;transmission)

        CLR1    DMAIF1                  ;INTDM1 interrupt request cleared
        CLR1    DMAMK1                  ;INTDM1 interrupt servicing enabled
        CLR1    CSIIF10                 ;INTCSI10 interrupt request cleared
        SET1    CSIMK10                 ;INTCSI10 interrupt servicing disabled

        SET1    DMAMK0                  ;INTDM0 interrupt servicing disabled

HCSI1650:
        SET1    SS0L.2                  ;CSI00 operation starts (trigger bit)
        MOV     SIO10,  A               ;Transmit data set (INTCSI00 occurs)
HCSI1800:
        BR      HCSI1RET

        ;---------------------------------------
        ; During playback
        ;---------------------------------------
HCSI1PLAY:
        ;Transmission of the higher bits of the 24-bit address
<5>     CMP     REEPSEQ, #CEEPSEQ_ADDRH ;Transmitting the higher bits of the
                                        ;24-bit address?
        BNZ     $HCSI1P200              ; No,
        MOV     A,      REEPADDR+1
        BR      HCSI1P350               ;Data transmitted
HCSI1P200:
        ;Transmission of the middle bits of the 24-bit address
        CMP     REEPSEQ, #CEEPSEQ_ADDRM ;Transmitting the middle bits of the
                                        ;24-bit address?
        BNZ     $HCSI1P300              ; No,
        MOV     A,      REEPADDR
        BR      HCSI1P350               ;Data transmitted
HCSI1P300:
        ;Transmission of the lower bits of the 24-bit address
        CMP     REEPSEQ, #CEEPSEQ_ADDRL ;Transmitting the lower bits of the
                                        ;24-bit address?
        BNZ     $HCSI1P400              ; No,
        MOV     A,      #000H
HCSI1P350:
        ;Data transmission
        MOV     SIO10,  A               ;Transmit data set (INTCSI10 occurs)
        BR      HCSI1RET
HCSI1P400:
        ;Starting data reception
<6>     MOVW    AX,     #0111000000000111B;Communication format setting
                        ;||||||||||||+++- DLS022 to DLS020: 8-bit data length
                        ;||||||||||||+---- <Fixed to 0>
                        ;||||||||||++----- SLC021 and SLC020: Unused  (fixed to 0)
                        ;|||||||||+------- <Fixed to 0>
                        ;||||||||+-------- DIR02: Input and output performed
                        ;||||||||                 with MSB first
                        ;||||||++--------- PTC021 and PTC020: Unused (fixed to 00)
                        ;|||||+----------- EOC02: Unused (fixed to 0)
                        ;||||+------------ <Fixed to 0>
                        ;||++------------- CKP02/DAP02: Phases of data and clock
                        ;||                              in CSI mode selected
                        ;||        [11 selected]
                        ;++--------------- TXE00/RXE00: Only reception performed
        MOVW    !SCR02, AX

        MOVW    DRA1,   #RRECMEM1         ;Address for setting dummy data
```

```
        MOVW    DBC1,     #3*2*42-1          ;Page size set

        BT      FI2SPAGE,$HCSI1P430          ;Is I2S transmitting page 2?
        MOVW    DRA0,     #RRECMEM2          ;Receive buffer set to page 1
        BR      HCSI1P450
HCSI1P430:
        MOVW    DRA0,     #RRECMEM1          ;Receive buffer set to page 2
HCSI1P450:
        MOVW    DBC0,     #3*2*42            ;Page size set

        ;Starting DMA transfer
        SET1    DST1                         ;DMA1 transfer enabled (for CSI10
                                             ;transmission)
        SET1    DST0                         ;DMA0 transfer enabled (for CSI10
                                             ;reception)

        SET1    DMAMK1                       ;INTDM1 interrupt servicing disabled
        CLR1    CSIIF10                      ;INTCSI10 interrupt request cleared
        SET1    CSIMK10                      ;INTCSI10 interrupt servicing disabled

        CLR1    DMAIF0                       ;INTDM0 interrupt request cleared
        CLR1    DMAMK0                       ;INTDM0 interrupt servicing enabled

        SET1    SS0L.2                       ;CSI10 operation starts (trigger bit)
        MOV     SIO10,    #0FFH              ;Dummy data set (INTCSI00 occurs)
HCSI1RET:
        RETI
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

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

   INTCSI10 interrupt servicing
   (using INTCSI10 for CSI transmission or reception)

-------------------------------------------------------------------------------

   During recording, instruction bytes and 24-bit addresses are transmitted
   to an EEPROM, DMA channel 1 starts when transmitting data to the
   EEPROM starts, and transmission continues in CSI10 single transmission mode.
   During playback, 24-bit addresses are transmitted to an EEPROM,
   DMA channels 0 and 1 start when receiving data from the EEPROM starts,
   and reception continues in CSI10 single reception mode.  However,
   DMA channel 1 is used for setting dummy data.

   Register bank 1 used

*******************************************************************************/
__interrupt void fn_intcsi10(void)
{
   EI();/* Multiple interrupts enabled */

      if(ucPlayMode==CREC){
      /*--------------------------------------
       During recording
      --------------------------------------*/
        ucEepromSeq++;   /* EEPROM transfer sequence updated */
        /* Instruction byte transmission */
        switch(ucEepromSeq){
         case CEEPSEQ_INST:
            /* Instruction byte transmission timing */
            P_CS |= 0b11110000;      /* Non-active */
            NOP();
            P_CS = (P_CS & 0b00001111)|ucEepromCs;      /* EEPROM selected */
            SIO10 = 0b00000010;   /* Transmission of write specification */
            break;

         case CEEPSEQ_ADDRH:
            /* Transmission of the higher bits of the 24-bit address */
            SIO10 = (unsigned char)(ushEepromAddress>>8);
            break;

         case CEEPSEQ_ADDRM:
            /* Transmission of the middle bits of the 24-bit address */
            SIO10 = (unsigned char)(ushEepromAddress);
            break;

         case CEEPSEQ_ADDRL:
            /* Transmission of the lower bits of the 24-bit address */
            SIO10 = 0x000;
            break;

         case CEEPSEQ_DATA:
         default:
            /* Starting data transmission */
            /* Starting DMA transfer */
            if(bI2sMemoryPage){
            /* Page 2 being transmitted */
                  DRA1 = (unsigned short)&ucMemoryPage1[1];
                  /* Transmit buffer set to page 1 */
            }
            else{
            /* Page 1 being transmitted */
                DRA1 = (unsigned short)&ucMemoryPage2[1];
                /* Transmit buffer set to page 2 */
            }
            DBC1 = 3*2*42-1;      /* One page */
            DST1 = 1;    /* DMA1 transfer enabled (for CSI10 transmission) */
```

```
            DMAIF1 = 0;     /* INTDM1 interrupt request cleared */
            DMAMK1 = 0;     /* INTDM1 interrupt servicing enabled */
            CSIIF10 = 0;    /* INTCSI10 interrupt request cleared */
            CSIMK10 = 1;    /* INTCSI10 interrupt servicing disabled */
            DMAMK0 = 1;     /* INTDM0 interrupt servicing disabled */
            SS0L.2 = 1;     /* CSI00 operation starts (trigger bit) */

            if(bI2sMemoryPage){
            /* Page 2 being transmitted */
                  SIO10 = *ucMemoryPage1;  /* Transmit data set (INTCSI00 occurs) */
            }
            else{
                  SIO10 = *ucMemoryPage2;  /* Transmit data set (INTCSI00 occurs) */
            /* Page 1 being transmitted */
            }

            break;
         }
      }
      else{
      /*---------------------------------------
       During playback
      ---------------------------------------*/
         ucEepromSeq++;     /* EEPROM transfer sequence updated */
         /* Instruction byte transmission */
         switch(ucEepromSeq){
            case CEEPSEQ_ADDRH:
               /* Transmission of the higher bits of the 24-bit address */
               SIO10 = (unsigned char)(ushEepromAddress>>8);
               break;

            case CEEPSEQ_ADDRM:
               /* Transmission of the middle bits of the 24-bit address */
               SIO10 = (unsigned char)(ushEepromAddress);
               break;

            case CEEPSEQ_ADDRL:
               /* Transmission of the lower bits of the 24-bit address */
               SIO10 = 0x000;
               break;

            case CEEPSEQ_DATA:
            default:
               /* Starting data reception */
               SCR02 = 0b0111000000000111;    /* Communication format setting */
               /*        |||||||||||||+++------- DLS022 to DLS020: 8-bit data length */
               /*        |||||||||||||+---------- <Fixed to 0> */
               /*        ||||||||||||++---------- SLC021 and SLC020: Unused (fixed to 0) */
               /*        |||||||||||+------------- <Fixed to 0> */
               /*        ||||||||+--------------- DIR02: Input and output performed with MSB
               /*        ||||||||                       first */
               /*        ||||||++---------------- PTC021 and PTC020: Unused (fixed to 00) */
               /*        |||||+------------------ EOC02: Unused (fixed to 0) */
               /*        ||||+------------------- <Fixed to 0> */
               /*        ||++-------------------- CKP02/DAP02: Phases of data and clock
               /*        ||                                   in CSI mode selected */
               /*        ||                        [11 selected] */
               /*        ++--------------------- TXE00/RXE00: Only reception performed */

               DRA1 = (unsigned short)&ucMemoryPage1[0];/* Address for setting dummy
                                                           data */
               DBC1 = 3*2*42-1;      /* Page size set */

               if(bI2sMemoryPage){
               /* Page 2 being transmitted */
                  DRA0 = (unsigned short)&ucMemoryPage1[0];/* Receive buffer set to
                                                              page 1 */
               }
               else{
               /* Page 1 being transmitted */
               DRA0 = (unsigned short)&ucMemoryPage2[0];/* Receive buffer set to
                                                            page 2 */
               }
```

```
                DBC0 = 3*2*42;           /* Page size set */

                /* Starting DMA transfer */
                DST1 = 1;    /* DMA1 transfer enabled (for CSI10 transmission) */
                DST0 = 1;    /* DMA0 transfer enabled (for CSI10 reception) */

                DMAMK1 = 1;  /* INTDM1 interrupt servicing disabled */
                CSIIF10 = 0; /* INTCSI10 interrupt request cleared */
                CSIMK10 = 1; /* INTCSI10 interrupt servicing disabled */

                DMAIF0 = 0;  /* INTDM0 interrupt request cleared */
                DMAMK0 = 0;  /* INTDM0 interrupt servicing enabled */

                SS0L.2 = 1;  /* CSI10 operation starts (trigger bit) */
                SIO10 = 0x0FF;/* Dummy data set (INTCSI00 occurs) */
                break;
            }
        }
}
```

4.8 INTDMA0 Interrupt Servicing

The following operations are performed in the INTDMA0 interrupt servicing in assembly language.

<1> The register bank is switched.
<2> A wait is inserted until communication via CSI10 ends.
<3> If the operating status indicates that preparing for playback has started, the operating status is set to "preparation for playback", and reading from the EEPROM of the second page is started.
<4> If the operating status indicates that playback is ready, because two pages of playback data has been acquired, the operating status is set to "playback under execution", operation of the I^2S bus interface is started, and data is transmitted to the audio codec.
<5> If the operating status indicates that playback is under execution, the EEPROM is selected. When the data of all EEPROM pages has been read, operation of the I^2S bus interface is stopped and the operation status is set to "playback stopped".
<6> DMA operation is stopped.

```
;*****************************************************************************
;
;        INTDMA0 interrupt servicing
;        (using INTDMA0 for CSI reception)
;
;-----------------------------------------------------------------------------
;
;        This interrupt occurs once data of one page has been received from an
;        EEPROM during playback.  Which of the four EEPROMs is to be used is
;        selected and operation of the I2S bus interface is started and stopped.
;
;        Register bank 1 used
;
;*****************************************************************************
IINTDMA0:
<1>     SELRB1

        ;Waiting for communication completion
<2>  HDMA0100:
        MOVA,    SSR02L                     ;Communication completed?
        BT       A.6,     $HDMA0100         ; No,

        ;Completing receiving data of one page
        OR       P_CS,    #11110000B        ;Non-active
        MOV      REEPSEQ, #CEEPSEQ_INST

        CMP      RPLAYMOD, #CPLAY           ;During playback?
        BZ       $HDMA0500                  ; Yes,
        CMP      RPLAYMOD, #CPLAY_SET       ;Preparing for playback?
        BZ       $HDMA0300                  ; Yes,
        CMP      RPLAYMOD, #CPLAY_START     ;Starting preparing for playback?
        BNZ      $HDMA0RET                  ; No,

        ;Starting reading from an EEPROM
<3>     MOV      RPLAYMOD, #CPLAY_SET       ;Prepared for playback
        MOV      A,       P_CS              ;EEPROM selected
        AND      A,       #00001111B
        OR       A,       #11010000B
        MOV      P_CS,    A
        CLR1     FI2SPAGE
        CLR1     CSIIF10                    ;INTCSI10 interrupt request cleared
        CLR1     CSIMK10                    ;INTCSI10 interrupt servicing enabled
        SET1     SS0L.2                     ;CSI10 operation starts (trigger bit)
        MOV      SIO10,   #00000011B        ;Transmit data (reading specified)
                                            ;set (INTCSI10 occurs)
        BR       HDMA0RET

    ;-----------------------------------------------
    ; Starting the operation of the I2S bus interface
    ;-----------------------------------------------
HDMA0300:
        MOV   RPLAYMOD,    #CPLAY           ;During playback
<4>     MOVW  RI2SADDR,    #RRECMEM1+1      ;Start of the transmit buffer set
        CLR1  FI2SPAGE
        MOV   A,           !RRECMEM1
        ;Starting the operation of the I2S bus interface
        SET1  SS0L.0                        ;CSI00 operation starts (trigger bit)
        CLR1  P_LRCLK                       ;LRCLK operation starts
        SET1  TS0L.0                        ;TO00 output operation starts
                                            ;(trigger bit)
        MOV   SIO00,       A                ;Transmit data set (INTCSI00 occurs)

        BR    HDMA0RET
HDMA0500:
<5>     ;EEPROM selection
        MOV   A,           RP_CS
        SET1  A.3
        ROLC  A,           1
```

```
          BC     $HDMA0700                        ;Have all EEPROMs been selected once? No,
          INCW   REEPADDR
          MOVW   AX,          REEPADDR
          CMPW   AX,          #128000/256        ;Last EEPROM?
          BNC    $HDMA0800                        ; Yes,
          MOV    A,           #11100000B
HDMA0700:
          AND    A,           #11110000B
          MOV    RP_CS,       A
          BR     HDMA0RET
HDMA0800:
          ;Stopping the operation of the I2S bus interface
          SET1   P_LRCLK                          ;LRCLK output disabled (high level)
          SET1   TT0L.0                           ;TO00 output operation stopped
                                                  ;(trigger bit)
          SET1   ST0L.0                           ;CSI000 output operation stopped
                                                  ;(trigger bit)
          SET1   ST0L.2                           ;CSI010 output operation stopped
                                                  ;(trigger bit)
          MOV    RPLAYMOD,    #CPLAY_END          ;Playback being terminated
HDMA0RET:
<6>       CLR1   DST1                             ;DMA1 transfer stopped
          CLR1   DST0                             ;DMA0 transfer stopped
          SET1   DMAMK0                           ;INTDM0 interrupt servicing disabled
          RETI
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

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

   INTDMA0 interrupt servicing
   (using INTDMA0 for CSI reception)

------------------------------------------------------------------------------

   This interrupt occurs once data of one page has been received from an
   EEPROM during playback.  Which of the four EEPROMs is to be used is
   selected and operation of the I2S bus interface is started and stopped.

   Register bank 1 used

******************************************************************************/
__interrupt void fn_intdma0(void)
{
   /* Waiting for completion of waiting for communication completion */
   while(SSR02 & 0b0000000001000000) {}

   /* Completing receiving data of one page */
   P_CS |= 0b11110000;      /* Non-active */
   ucEepromSeq = CEEPSEQ_INST;

   if(ucPlayMode==CPLAY){
   /* During playback */
   /*---------------------------------------------
    Starting the operation of the I2S bus interface
   ---------------------------------------------*/
       /* Selecting EEPROM */
       if((ucEepromCs & 0b10000000)==0b00000000){
       /* Selecting all EEPROMs once */
          ushEepromAddress++;
          if(ushEepromAddress>=128000/256){
            /* Stopping the operation of the I2S bus interface */
            P_LRCLK = 1; /* LRCLK output disabled (high level) */
            TT0L.0 = 1;  /* TO00 output operation stopped (trigger bit) */
            ST0L.0 = 1;  /* CSI000 output operation stopped (trigger bit) */
            ST0L.2 = 1;  /* CSI010 output operation stopped (trigger bit) */
            ucPlayMode = CPLAY_END;  /* Playback being terminated */
          }
          else{
            ucEepromCs = 0b11100000;
          }
       }
       else{
            ucEepromCs = (ucEepromCs|0b00001000) <<1;
       }
   }
   else if(ucPlayMode==CPLAY_START){
   /* Starting preparing for playback */
   /* Starting reading from an EEPROM */
      ucPlayMode = CPLAY_SET;  /* Preparing for playback */
      P_CS = (P_CS & 0b00001111)|0b11010000;   /* EEPROM selected */
      bI2sMemoryPage = 0;
      CSIIF10 = 0;    /* INTCSI10 interrupt request cleared */
      CSIMK10 = 0;    /* INTCSI10 interrupt servicing enabled */
      SS0L.2 = 1;/* CSI10 operation starts (trigger bit) */
      SIO10 = 0b00000011;   /* Transmit data (reading specified) set (INTCSI10 occurs) */
   }
   else if(ucPlayMode==CPLAY_SET){
   /* Preparing for playback */
      ucPlayMode = CPLAY;/* During playback */
      ucI2sAddress = &ucMemoryPage1[1];/* Start of the transmit buffer set */
      bI2sMemoryPage = 0;/* page 1 transmit */
      /* Starting the operation of the I2S bus interface */
      SS0L.0 = 1;/* CSI00 operation starts (trigger bit) */
      P_LRCLK = 0;    /* LRCLK operation starts */
      TS0L.0 = 1;/* TO00 output operation starts (trigger bit) */
```

```
        SIO00 = ucMemoryPage1[0];/* Transmit data set (INTCSI00 occurs) */
    }
    DST1 = 0;    /* DMA1 transfer stopped */
    DST0 = 0;    /* DMA0 transfer stopped */
    DMAMK0 = 1;  /* INTDM0 interrupt servicing disabled */
}
```

4.9 INTDMA1 Interrupt Servicing

The following operations are performed in the INTDMA1 interrupt servicing in assembly language.

<1> The register bank is switched.

<2> A wait is inserted until communication via CSI10 ends.

<3> The EEPROM is selected. When the data of all EEPROM pages is written, operation of the I²S bus interface is stopped and the operation status is set to "recording stopped".

<4> DMA operation is stopped.

```
;*****************************************************************************
;
;        INTDMA1 interrupt servicing
;        (using INTDMA1 for CSI transmission)
;
;-----------------------------------------------------------------------------
;
;        This interrupt occurs when data of one page has been transmitted to an
;        EEPROM during recording.    Which of the four EEPROMs is to be used is
;        selected and operation of the I2S bus interface is stopped.
;
;        Register bank 1 used
;
;*****************************************************************************
IINTDMA1:
<1>     SEL    RB1

        ;Waiting for communication completion
HDMA1100:
<2>     MOV    A,          SSR02L            ;Communication completed?
        BT     A.6,        $HDMA1100         ; No,

<3>     ;Completing transmitting data of one page
        OR     P_CS,       #11110000B        ;Data of one page transmitted
        MOV    REEPSEQ,    #CEEPSEQ_RESET

        ;EEPROM selection
        MOV    A,          RP_CS
        SET1   A.3
        ROLC   A, 1
        BC     $HDMA1300                     ;Have all EEPROMs been selected once? No,
        INCW   REEPADDR
        MOVW   AX,         REEPADDR
        CMPW   AX,         #128000/256       ;Last EEPROM?
        BNC    $HDMA1500                     ; Yes,
        MOV    A,          #11100000B
HDMA1300:
        AND    A,          #11110000B
        MOV    RP_CS,      A
        BR     HDMA1RET
HDMA1500:
        ;Stopping the operation of the I2S bus interface
        SET1   P_LRCLK                       ;LRCLK output disabled (high level)
        SET1   TT0L.0                        ;TO00 output operation stopped
                                             ;(trigger bit)
        SET1   ST0L.0                        ;CSI000 output operation stopped
                                             ;(trigger bit)
        SET1   ST0L.2                        ;CSI010 output operation stopped
                                             ;(trigger bit)
        MOV    RPLAYMOD,   #CREC_END         ;Recording stopped
HDMA1RET:
```

<4>

```
        CLR1   DST1                          ;DMA1 transfer stopped
        CLR1   DST0                          ;DMA0 transfer stopped
        SET1   DMAMK1                        ;INTDM1 interrupt servicing disabled
        RETI
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

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

   INTDMA1 interrupt servicing
   (using INTDMA1 for CSI transmission)

------------------------------------------------------------------------------

   This interrupt occurs when data of one page has been transmitted to an EEPROM
   during recording.  Which of the four EEPROMs is to be used is selected and
   operation of the I2S bus interface is stopped.

   Register bank 1 used

******************************************************************************/
__interrupt void fn_intdma1(void)
{
   /* Waiting for completion of waiting for communication completion */
   while(SSR02 & 0b0000000001000000) {}

   /* Completing transmitting data of one page */
   P_CS |= 0b11110000;   /* Completing transmitting data of one page */
   ucEepromSeq = CEEPSEQ_RESET;

   /* EEPROM selection */
   if((ucEepromCs & 0b10000000)==0b00000000){
       ushEepromAddress++;
       if(ushEepromAddress>=128000/256){
          /* Stopping the operation of the I2S bus interface */
          P_LRCLK = 1;   /* LRCLK output disabled (high level) */
          TT0L.0 = 1; /* TO00 output operation stopped (trigger bit) */
          ST0L.0 = 1; /* CSI000 output operation stopped (trigger bit) */
          ST0L.2 = 1; /* CSI010 output operation stopped (trigger bit) */
          ucPlayMode = CPLAY_END; /* Playback stopped */
       }
       else{
           ucEepromCs = 0b11100000;
       }
   }
   else{
       ucEepromCs = (ucEepromCs|0b00001000) <<1;
   }
   DST1 = 0;    /* DMA1 transfer stopped */
   DST0 = 0;    /* DMA0 transfer stopped */
   DMAMK1 = 1;  /* INTDM1 interrupt servicing disabled */
}
```

4.10 I²C Bus Interface Write Processing

The following operations are performed in the I²C bus interface write processing for setting the registers in the audio codec specified in assembly language.

<1> A start condition is issued.
<2> The slave address is transmitted.
<3> The addresses of the registers in the audio codec to be written are transmitted.
<4> The setting data to be written is transmitted.
<5> A stop condition is issued.
<6> If a wait is required, a wait is inserted.
<7> Whether transmission is to be terminated is determined. If it is not to be terminated, the processing returns to <1>, because subsequent data exists.

```
;*****************************************************************************
;
;        I2C bus interface write processing for setting the registers in the audio
;        codec
;
;-----------------------------------------------------------------------------
;
;        The data set to the registers in the audio codec is written by using the
;        functions of I2C bus interface IIC0.
;
;-----------------------------------------------------------------------------
;
;        [I N] ES          :Higher 4 bits of the start address of the ROM table
;                           holding the register setting values
;              HL          :Lower 8 bits of the start address of the ROM table
;                           holding the register setting values
;              DE          :Lower 8 bits of the last address of the ROM table
;                           holding the register setting values
;        [OUT] -
;
;*****************************************************************************
SI2CWRITE:
         SET1  IICE0                          ; Operation enabled
JI2CW100:
         ;Start condition issuance
<1>      SET1  STT0

         ;Slave address transmission
<2>      BF    STD0,       $$                 ;Address transmission period? No,
         CLR1  IICIF0
         MOV   IIC0,       #00110100B         ;Address transmission starts (writing
                                              ;specified)
         BF    IICIF0,     $$                 ;Transmission completed? No,
         BT    ACKD0,      $JI2CW200          ;ACK detected? Yes,
         SET1  SPT0                           ; No, stop condition issued
         BR    $JI2CW100                      ;Retransmission

JI2CW200:
         ;Write address transmission
<3>      MOV   A,          ES:[HL]            ;Set data acquired
         MOV   IIC0,       A                  ;Data transmission starts
         BF    IICIF0,     $$                 ;Transmission completed? No,
         BT    ACKD0,      $JI2CW300          ;ACK detected? Yes,
         SET1  SPT0                           ; No, stop condition issued
         BR    $JI2CW100                      ;Retransmission

JI2CW300:
         ;Write data transmission
         INCW  HL                             ;Reference position updated
<4>      MOV   A,          ES:[HL]            ;Set value acquired
         CLR1  IICIF0                         ;INTIIC10 interrupt request cleared
         MOV   IIC0,       A                  ;Data transmission starts
         BF    IICIF0,     $$                 ;Transmission completed? No,
         BT    ACKD0,      $JI2CW400          ;ACK detected? Yes,
         SET1  SPT0                           ; No, stop condition issued
         DECW  HL                             ;Retransmission
         BR    $JI2CW100

JI2CW400:
         ;Stop condition issuance
<5>      SET1  SPT0

         ;Wait control
<6>      INCW  HL                             ;Reference position updated
         MOV   A,          ES:[HL]            ;Wait information acquired
         CMP0  A                              ;Waiting required?
         BZ    $JI2CW600                      ; No,
         MOV   X,          #195               ;5 ms
         MULU  X                              ;A register*5 ms
         DECW  AX                             ;AX register: Timer count value
         MOVW  TDR01,      AX                 ;Interval set
```

```
            CLR1   TMIF01                          ;INTTM01 interrupt request cleared
            SET1   TS0L.1                          ;Timer operation starts
            BF     TMIF01,      $$                 ;Interval elapsed? No,
            SET1   TT0L.1                          ;Timer operation stopped

JI2CW600:
            INCW   HL                              ;Reference position updated
            MOVW   AX,          HL
            CMPW   AX,          DE                 ;Register setting completed?
            BC                  $JI2CW100          ; No,
JI2CWRET:
            RET
```

Similar operations to those performed in assembly language are performed in the processing specified in C language.

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

   I2C bus interface write processing for setting the registers
   in the audio codec

-----------------------------------------------------------------------------

   The data set to the registers in the audio codec is written by using the
   functions of I2C bus interface IIC0.

-----------------------------------------------------------------------------

   [I N]*addr   :Higher 4 bits of the start address of the ROM table holding
                 the register setting values
         size   :ROM table size
   [OUT] -

*****************************************************************************/
void fn_I2cWrite(unsigned char *addr, unsigned char size)
{
   register unsigned char cnt;

   IICE0 = 1;   /* Operation enabled */

   for(cnt=0; cnt<=size/3; cnt++){
      /* Start condition issuance */
      STT0 = 1;

      /* Slave address transmission */
      while(!STD0)       {} /* Address transmission period? No, */
      IICIF0 = 0;
      IIC0 = 0b00110100;    /* Address transmission starts (writing specified) */
      while(!IICIF0)     {} /* Transmission completed? No, */
      if(ACKD0){
          /* Write address transmission */
          IICIF0 = 0;       /* INTIIC10 interrupt request cleared */
          IIC0 = *addr;     /* Data transmission starts */
          while(!IICIF0) {} /* Transmission completed? No, */

          if(ACKD0){
            /* Write data transmission */
            addr++;         /* Reference position updated */
            IICIF0 = 0;     /* INTIIC10 interrupt request cleared */
            IIC0 = *addr;   /* Data transmission starts */
            while(!IICIF0)  {} /* Transmission completed? No, */

            if(ACKD0){
               /* stop condition issued */
               SPT0 = 1;

               /* Wait control */
               addr++;      /* Reference position updated */
               if(*addr!=0){
                  TDR01 = 195*(*addr)-1;   /* Interval set (5ms*addr) */
                  TMIF01 = 0;  /* INTTM01 interrupt request cleared */
                  TS0L.1 = 1;  /* Timer operation starts */
                  while(!TMIF01)     {} /* Interval elapsed? No, */
                  TT0L.1 = 1;  /* Timer operation stopped */
               }
               addr++;         /* Reference position updated */
            }
            else{
               SPT0 = 1;    /* Stop condition issuance */
               cnt--;       /* Retransmission */
               addr--;
```

```
            }
        }
        else{
            SPT0 = 1;    /* Stop condition issuance */
            cnt--;       /* Retransmission */
        }
    }
    else{
        SPT0 = 1;      /* Stop condition issuance */
        cnt--;         /* Retransmission */
    }
  }
}
```

CHAPTER 5 RELATED DOCUMENTS

Document Name		Japanese/English
78K0R/KE3 User's Manual		PDF
78K0R/KF3 User's Manual		PDF
78K0R/KG3 User's Manual		PDF
78K0R/KH3 User's Manual		PDF
78K0R/KJ3 User's Manual		PDF
78K0R Microcontrollers Instructions User's Manual		PDF
RA78K0R Assembler Package	Language	PDF
User's Manual	Operation	PDF
CC78K0R C Compiler	Language	PDF
User's Manual	Operation	PDF
PM+ Project Manager User's Manual		PDF

APPENDIX A PROGRAM LIST

The source program used for the 78K0R/KG3 microcontroller is shown below as a program list example.

• main.asm (assembly language version)

```
;*****************************************************************************
;
;    NEC Electronics     78K0R/KG3 Series
;
;*****************************************************************************
;    78K0R/KG3 Series
;*****************************************************************************
;    I2S bus interface between microcontroller and audio codec
;*****************************************************************************
;
; [Overview]
;
;This sample program uses the I2S bus interface to transfer audio data between
;the microcontroller and an audio codec.  The LR clock (LRCLK) for selecting whether to
;use channel L or channel R of the I2S bus interface is used in the interval timer mode of
;the timer array unit (TAU) and is output from TO00.  CSI00 of channel 0 of serial array unit
;0 (SAU0) is used for outputting the clock used for transferring data (BCLK) and for
;receiving data (SDOUT), and transmitting data (SDIN).
;ML2612 made by OKI Semiconductor is used as the audio codec.  The I2C bus interface is used
;for setting the registers in the audio codec and the I2S bus interface is used for
;transferring audio data.  When the record key is pressed, the audio codec receives the
;voice data input from a microphone and saves the data to EEPROM.  When the playback key
;is pressed, the recorded data is transmitted to the audio codec and the output as sound
;from the speakers.  The SPI interface between the microcontroller and EEPROM uses CSI10
;of channel 2 of serial array unit 0 (SAU0) to continuously perform CSI transmission
;or reception by using the DMA controller.  Whether recording or playback is in progress
;is indicated by an LED (operation-in-progress indicator) to which a signal is output from P72.
;
;The following are used to execute the sample program.
;
; I2C bus interface for setting the registers in the audio codec
; •Serial interface IIC0
;
; I2S bus interface used for transferring audio data between the microcontroller and
; audio codec
;  •Using the output from TO00 of the timer array unit (TAU) as LRCLK
;  •Using CSI00 of channel 0 of serial array unit 0 (SAU0) for transmitting and receiving data
;
; SPI interface used between the microcontroller and EEPROMs
;  •Using P74 to P77 as chip select pins because four EEPROMs are used
;  •Using CSI10 of channel 2 of serial array unit 0 (SAU0)
;  •Using DMA0 and DMA1 of the DMA controller for performing successive transmission or
;   reception
;
; Wait timer
;  •Using the interval timer mode of channel 1 of the timer array unit (TAU) to insert waits of
;   at least 5 ms in the program
;
; Record and playback keys
;  •Key interrupt input pins KR0 and KR1
;
; Operation-in-progress indicator (LED)
;  •Outputting a signal from P72 to light the LED used as an operation-in-progress indicator
;
;
;<Main initial settings for the peripheral hardware to be used>
; •Disabling interrupts
; •Setting the CPU or peripheral hardware clock frequency to the X1 oscillation clock (when
;  used at 20 MHz)
; •Setting ports
; •Setting the audio codec
;  •Supplying a clock to the audio codec
```

```
;  •Setting a timer for inserting waits in the program
;  •Outputting a reset signal to the audio codec
;  •Setting the I2C interface used for setting registers
;  •Turning on the system by setting the registers in the audio codec
; ●Setting the I2S bus interface used for the audio data of the audio codec
;  •Using TO00 output (16 kHz) for setting the output of LRCLK
;  •Setting CSI00
; ●Setting the SPI interface of EEPROM
;  •Setting CSI10 for transmitting and receiving data
;  •Setting DMA0 for successive reception and DMA1 for successive transmission
;  •Erasing all EEPROMs (all 0FFH)
; ●Starting key data retrieval
; ●Enabling interrupts
;
;
;<Main processing>
; •Key processing
; •Setting start of recording
; •Setting start of playback
; •Setting termination of recording
; •Setting termination of playback
;
;
;<Main INTTM00 interrupt servicing (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and
;SDIN)>
; •Starting the I2S bus interface
;
;
;<Main INTCSI00 interrupt servicing (using INTCSI00 for CSI transmission or reception)>
; •Transmitting or receiving data via the I2S bus interface
; •Saving the received data
; •Starting writing to EEPROM
; •Starting reading from EEPROM
;
;
;<Main INTCSI10 interrupt servicing (using INTCSI10 for CSI transmission or reception)>
; •Transmitting instruction bytes to EEPROM
; •Transmitting 24-bit addresses to EEPROM
; •Starting receiving data from EEPROM by using DMA0 or DMA1
; •Starting transmitting data to EEPROM by using DMA1
;
;
;<Main INTDMA0 interrupt servicing (using INTDMA0 for CSI reception)>
; •Starting reading from EEPROM
; •Starting the I2S bus interface
; •Selecting EEPROM
; •Stopping the I2S bus interface
;
;
;<Main INTDMA1 interrupt servicing (using INTDMA1 for CSI transmission)>
; •Selecting EEPROM
; •Stopping the I2S bus interface
;
;
;*****************************************************************************

;=============================================================================
;
;    Vector table settings
;
;=============================================================================
TVCT1CSEG    AT      000000H
     DW      RESET_START                  ;(00)  Reset input, POC, LVI, WDT, TRAP
TVCT2CSEG    AT      000004H
     DW      RESET_START                  ;(04)  INTWDTI
     DW      RESET_START                  ;(06)  INTLVI
     DW      RESET_START                  ;(08)  INTP0
```

```
     DW      RESET_START                    ;(0A)  INTP1
     DW      RESET_START                    ;(0C)  INTP2
     DW      RESET_START                    ;(0E)  INTP3
     DW      RESET_START                    ;(10)  INTP4
     DW      RESET_START                    ;(12)  INTP5
     DW      RESET_START                    ;(14)  INTST3
     DW      RESET_START                    ;(16)  INTSR3
     DW      RESET_START                    ;(18)  INTSRE3
     DW      IINTDMA0                       ;(1A)  INTDMA0
     DW      IINTDMA1                       ;(1C)  INTDMA1
     DW      IINTCSI00                      ;(1E)  INTST0/INTCSI00
     DW      RESET_START                    ;(20)  INTSR0/INTCSI01
     DW      RESET_START                    ;(22)  INTSRE0
     DW      IINTCSI10                      ;(24)  INTST1/INTCSI10/INTIIC10
     DW      RESET_START                    ;(26)  INTSR1
     DW      RESET_START                    ;(28)  INTSRE1
     DW      RESET_START                    ;(2A)  INTIIC0
     DW      IINTTM00                       ;(2C)  INTTM00
     DW      RESET_START                    ;(2E)  INTTM01
     DW      RESET_START                    ;(30)  INTTM02
     DW      RESET_START                    ;(32)  INTTM03
     DW      RESET_START                    ;(34)  INTAD
     DW      RESET_START                    ;(36)  INTRTC
     DW      RESET_START                    ;(38)  INTRTCI
     DW      RESET_START                    ;(3A)  INTKR
     DW      RESET_START                    ;(3C)  INTST2/INTCSI20/INTIIC20
     DW      RESET_START                    ;(3E)  INTSR2
     DW      RESET_START                    ;(40)  INTSRE2
     DW      RESET_START                    ;(42)  INTTM04
     DW      RESET_START                    ;(44)  INTTM05
     DW      RESET_START                    ;(46)  INTTM06
     DW      RESET_START                    ;(48)  INTTM07
     DW      RESET_START                    ;(4A)  INTP6
     DW      RESET_START                    ;(4C)  INTP7
     DW      RESET_START                    ;(4E)  INTP8
     DW      RESET_START                    ;(50)  INTP9
     DW      RESET_START                    ;(52)  INTP10
     DW      RESET_START                    ;(54)  INTP11

TBRK CSEG    AT      00007EH
     DW      RESET_START                    ;(7E)  BRK

;==============================================================================
;
;    Securing a stack area
;
;==============================================================================
DSTK DSEG    AT      0FFEC0H
STACKEND:
             DS      20H                    ;32-byte stack area secured
STACKTOP:                                   ;Stack area start address = FFEE0H

;==============================================================================
;
;    Port definitions
;
;==============================================================================
;Audio codec reset
P_RESETB            EQU     P7.3            ;Reset
PM_RESETB           EQU     PM7.3           ;Reset output

;I2C interface used for setting the registers in the audio codec
P_SCL               EQU     P6.0            ;SCL0 latch
PM_SCL              EQU     PM6.0           ;SCL0 output
P_SDA               EQU     P6.1            ;SDA0 latch
PM_SDA              EQU     PM6.1           ;SDA0 I/O
```

```
;Audio codec I2S bus interface
P_LRCLK             EQU     P0.1            ;LRCLK
PM_LRCLK            EQU     PM0.1           ;LRCLK output
P_SCK00             EQU     P1.0            ;SCK00
PM_SCK00            EQU     PM1.0           ;SCK00 input
PM_SI00             EQU     PM1.1           ;SI00 input
P_SO00              EQU     P1.2            ;SO00
PM_SO00             EQU     PM1.2           ;SO00 output

;Key inputs (used: 2)
P_KEY               EQU     P7              ;KR (P70 and P71 used)
PM_KEY              EQU     PM7             ;KR input (PM70 and PM71 used)
 P_PLAYKY           EQU     P7.0            ;Playback key
 PM_PLAYKY          EQU     PM7.0           ;Playback key input
 P_RECKY            EQU     P7.1            ;Record key
 PM_RECKY           EQU     PM7.1           ;Record key input

;Output to the LED used as an operation-in-progress indicator (used: 1)
P_LED               EQU     P7.2            ;Operation-in-progress indicator (active low)
PM_LED              EQU     PM7.2           ;Output to the operation-in-progress indicator (LED)
                                            ;LED lit during operation and turned off when key
                                            ;input is possible.
;EEPROM SPI interfaces (used: 4)
P_CS                EQU     P7              ;Chip select CS (P73 to P77 used)
PM_CS               EQU     PM7             ;Chip select CS output (PM73 to PM77 used)
 P_CS3              EQU     P7.7            ;CS3
 PM_CS3             EQU     PM7.7           ;CS3 output
 P_CS2              EQU     P7.6            ;CS2
 PM_CS2             EQU     PM7.6           ;CS2 output
 P_CS1              EQU     P7.5            ;CS1
 PM_CS1             EQU     PM7.5           ;CS1 output
 P_CS0              EQU     P7.4            ;CS0
 PM_CS0             EQU     PM7.4           ;CS0 output
P_SCK10             EQU     P0.4            ;SCK10
PM_SCK10            EQU     PM0.4           ;SCK10 output
PM_SI10             EQU     PM0.3           ;SI10 input
P_SO10              EQU     P0.2            ;SO10
PM_SO10             EQU     PM0.2           ;SO10 output

;==============================================================================
;
;    RAM definitions
;
;==============================================================================
DPLAYDSEG   SADDR
;Overall operation
RPLAYMOD:           DS      1               ;Operation status
 CRESET             EQU     0               ; Reset
 CSTOP              EQU     1               ; Stopped
 CREC               EQU     2               ; Recording under execution
 CREC_END           EQU     3               ;  Recording finished
 CPLAY_START        EQU     4               ; Preparing for playback started
 CPLAY_SET          EQU     5               ;  Preparing for playback
 CPLAY              EQU     6               ;  Playback under execution
 CPLAY_END          EQU     7               ;  Playback finished

;Definitions related to the operation of the I2S bus interface used between the microcontroller
;and audio codec

RCSI00CNT:          DS      1               ;Counter for transmitting and receiving 1-byte I2S
                                            ;data
RPLAYINFO:          DS      1               ;Playback information
 FI2SPAGE           EQU     RPLAYINFO.7     ; Page being transmitted or received via the I2S
                                            ; bus: Page 1 (0)/page 2 (1)

;Definitions related to the operation of the SPI interface used between the microcontroller and
;EEPROMs
```

```
RP_CS:              DS      1               ;Chip select CS setting value of the EEPROMs
REEPSEQ:            DS      1               ;EEPROM transfer sequence
 CEEPSEQ_RESET      EQU     0               ; Reset status
 CEEPSEQ_WREN       EQU     1               ; Write enable signal transmitted
 CEEPSEQ_INST       EQU     2               ; Instruction bytes transmitted
 CEEPSEQ_ADDRH      EQU     3               ; Higher 8 bits of 24-bit addresses transmitted
 CEEPSEQ_ADDRM      EQU     4               ; Middle 8 bits of 24-bit addresses transmitted
 CEEPSEQ_ADDRL      EQU     5               ; Lower 8 bits of 24-bit addresses transmitted
 CEEPSEQ_DATA       EQU     6               ; Data transmitted and received by using DMA

DPLAYP       DSEG   SADDRP
;Definitions related to the operation of the I2S bus interface used between the
;microcontroller and audio codec
RI2SADDR:           DS      2               ;Address to which data is transmitted or received
                                            ;via I2S saved

;Definitions related to the operation of the SPI interface used between the microcontroller and
;EEPROMs
REEPADDR:           DS      2               ;EEPROM read or write address (REEPADDR*100H)

DMEM DSEG   UNITP
;Memory area of internal RAM
RRECMEM1:                                   ;Page 1
                    DS      3*2*42          ; 42 LRCLK clocks
RRECMEM1E:                                  ; Last address + 1
RRECMEM2:                                   ;Page 2
                    DS      3*2*42          ; 42 LRCLK clocks
RRECMEM2E:                                  ; Last address + 1
;==============================================================================
;
;    ROM definitions
;
;==============================================================================
CREGACC      CSEG   UNITP
;------------------------------------------------------------------------------
;
;    These are the ROM tables that hold the values to be used for setting the registers
;    in the audio codec used to transition the status of the ML2612 audio codec made by OKI
;    Semiconductor.
;    These tables are used in the write processing performed via the I2C bus interface.
;
;    [Example]
;                   DB      0E5H,   00000111B,      0       ;Trimming 1
;                   DB      0E9H,   00000001B,      0       ;Trimming 2
;                           ~~~~|   ~~~~~~~~~|      ~|
;                               |           |       +----- Wait information (Note)
;                               |           +---------- Write data
;                               +---------------------- Write address
;
;           Note: Indicates the wait time after data is written.
;                 A wait of (5 ms * (described value)) is inserted.
;                 No wait is inserted if the value is 0.
;
;    * The setting data shown in this sample program is simply an example
;      of the settings that can be used to execute recording or playback.
;      See the data sheet and other materials for details.
;
;------------------------------------------------------------------------------
TSYSON:
;-----------------------------------------------------------------------
;    Power off → System on
;-----------------------------------------------------------------------
     ;-------------------------------------------------------------
     ; Trimming
     ;-------------------------------------------------------------
     DB      0E5H,   00000111B,      0       ;Trimming 1
     DB      0E9H,   00000001B,      0       ;Trimming 2
```

```
     ;-----------------------------------------------------------------
     ; Digital Block ON
     ;-----------------------------------------------------------------
     DB      001H,   003H,           0       ;Sampling Rate
     DB      003H,   001H,           0       ;PLLNL
     DB      005H,   000H,           0       ;PLLNH
     DB      007H,   000H,           0       ;PLLML
     DB      009H,   002H,           0       ;PLLMH
     DB      00BH,   002H,           0       ;PLLDIV
     DB      00FH,   00000001B,      0       ;CLK Input/Output control
     DB      00DH,   00000001B,      0       ;Clock Enable
     DB      00DH,   00000101B,      150/5   ;Clock Enable
TSYSONE:

TRECON:
;-----------------------------------------------------------------------
;    System on → Recording start
;-----------------------------------------------------------------------
     ;-----------------------------------------------------------------
     ; Record Function Setting
     ;-----------------------------------------------------------------
     ; Volume settings
     DB      033H,   03FH,           0       ;Mic Input Volume
     DB      039H,   00000000B,      0       ;Mic Boost Volume
     DB      049H,   00000001B,      0       ;AMP Volume Control Function Enable
     DB      04BH,   00000000B,      0       ;Amplifier Volume Fader Control
     DB      069H,   00011110B,      0       ;Volume Control Func Enable
     DB      06BH,   00000010B,      0       ;Mixer & Volume Control
     DB      06DH,   0FFH,           0       ;Record Digital Volume Control

     ;HPF1 ON
     DB      067H,   00000001B,      0       ;Filter Func Enable

     ;HPF2 ON
     ;DB     067H,   00000000B,      0       ;Filter Func Enable
     ;DB     07FH,   00000000B,      0       ;HPF2 CutOff

     ;Programmable Equalizer settings
     ;DB     06DH,   11111111B,      0       ;Record Digital Volume
     ;DB     067H,   00000001B,      0       ;Filter Func Enable
     ;DB     067H,   00000000B,      0       ;Filter Func Enable
     ;DB     07FH,   00000000B,      0       ;HPF2 CutOff
     ;DB     067H,   00000000B,      0       ;Filter Func Enable
     ;DB     075H,   00000000B,      0       ;EQ gain1 Band0
     ;DB     077H,   00000000B,      0       ;EQ gain1 Band1
     ;DB     079H,   00000000B,      0       ;EQ gain1 Band2
     ;DB     07BH,   00000000B,      0       ;EQ gain1 Band3
     ;DB     07DH,   00000000B,      0       ;EQ gain1 Band4
     ;DB     083H,   00000000B,      0       ;EQ Band0 Coef0H
     ;DB     085H,   00000000B,      0       ;EQ Band0 Coef1L
     ;DB     087H,   00000000B,      0       ;EQ Band0 Coef1H
     ;DB     089H,   00000000B,      0       ;EQ Band1 Coef0L
     ;DB     08BH,   00000000B,      0       ;EQ Band1 Coef0H
     ;DB     08DH,   00000000B,      0       ;EQ Band1 Coef1L
     ;DB     08FH,   00000000B,      0       ;EQ Band1 Coef1H
     ;DB     091H,   00000000B,      0       ;EQ Band2 Coef0L
     ;DB     093H,   00000000B,      0       ;EQ Band2 Coef0H
     ;DB     095H,   00000000B,      0       ;EQ Band2 Coef1L
     ;DB     097H,   00000000B,      0       ;EQ Band2 Coef1H
     ;DB     099H,   00000000B,      0       ;EQ Band3 Coef0L
     ;DB     09BH,   00000000B,      0       ;EQ Band3 Coef0H
     ;DB     09DH,   00000000B,      0       ;EQ Band3 Coef1L
     ;DB     09FH,   00000000B,      0       ;EQ Band3 Coef1H
     ;DB     0A1H,   00000000B,      0       ;EQ Band4 Coef0L
     ;DB     0A3H,   00000000B,      0       ;EQ Band4 Coef0H
     ;DB     0A5H,   00000000B,      0       ;EQ Band4 Coef1L
```

```
     ;DB     0A7H,   00000000B,     0       ;EQ Band4 Coef1H

     ;ALC settings
     DB      069H,   00000010B,     0       ;Volume Control Func Enable
     ;DB     0B1H,   00000000B,     0       ;ALC Mode
     ;DB     0B3H,   00000000B,     0       ;ALC Attack Time
     ;DB     0B5H,   00000000B,     0       ;ALC Decay Time
     ;DB     0B7H,   00000000B,     0       ;ALC Hold Time
     ;DB     0B9H,   00000000B,     0       ;ALC Target Level
     ;DB     0BBH,   00000000B,     0       ;ALC Max/Min Gain
     ;DB     0BDH,   00000000B,     0       ;Noise Gate Threshold
     ;DB     0BFH,   00000000B,     0       ;ALC Zero Cross Time OutPlayback Limiter Control
                                            ;Register
     ;DB     02FH,   00000000B,     0       ;ZC-CMP Power Management

     ;-----------------------------------------------------------------
     ; Record Analog Block ON
     ;-----------------------------------------------------------------
     ;Turning on the VMID generator
     DB      021H,   00000001B,     5/5     ;Reference Power Management
     DB      021H,   00000010B,     5/5     ;Reference Power Management

     ;Turning on the microphone bias circuit
     DB      021H,   00000110B,     0       ;Reference Power Management

     ;Minimum input setting (differential input)
     DB      05BH,   00000010B,     0       ;Mic IF Control

     ;Power Management Register
     DB      023H,   00001000B,     0       ;Input Power Management
     DB      023H,   00001010B,     0       ;Input Power Management

     ;-----------------------------------------------------------------
     ; Record start
     ;-----------------------------------------------------------------
     DB      069H,   00010000B,     0       ;Volume Control Func Enable
     DB      069H,   00010000B,     0       ;Volume Control Func Enable
     DB      013H,   00000001B,     5/5     ;Record/Playback Run
     DB      06BH,   00110010B,     0       ;Mixer & Volume Control
     DB      069H,   00011000B,     0       ;Volume Control Func Enable
     DB      069H,   00001000B,     0       ;Volume Control Func Enable
TRECONE:

TRECOFF:
;-----------------------------------------------------------------------
;    Recording → System on
;-----------------------------------------------------------------------
     ;-----------------------------------------------------------------
     ; Record stop
     ;-----------------------------------------------------------------
     DB      06BH,   00000010B,     0       ;Mixer & Volume Control
     DB      069H,   00001000B,     0       ;Volume Control Func Enable
     DB      069H,   00011000B,     200/5   ;Volume Control Func Enable
     DB      013H,   00000000B,     10/5    ;Record/Playback Run

     ;-----------------------------------------------------------------
     ; Record Analog Block OFF
     ;-----------------------------------------------------------------
     DB      023H,   00001000B,     0       ;Input Power Management
     DB      023H,   00000000B,     0       ;Input Power Management
     DB      021H,   00000001B,     0       ;Reference Power Management
     DB      021H,   00000000B,     0       ;Reference Power Management
TRECOFFE:

TPLAYON:
;-----------------------------------------------------------------------
;    System on → Playback start
```

```
;-----------------------------------------------------------------------
     ;-------------------------------------------------------------
     ; Playback Function Setting
     ;-------------------------------------------------------------
     ; Volume settings
     DB      03BH,   00000000B,      0       ;Speaker AMP Volume
     DB      0C9H,   00000000B,      0       ;Boost Volume
     DB      049H,   00000000B,      0       ;AMP Volume Control Function Enable
     DB      04BH,   00000000B,      0       ;Amplifier Volume Fader Control
     DB      069H,   00000000B,      0       ;Volume Control
     DB      06BH,   00000010B,      0       ;Mixer & Volume
     DB      071H,   11111111B,      0       ;Playback Digital Volume Control

     ;Programmable Equalizer settings
     ;DB     067H,   00000000B,      0       ;Filter Func Enable
     ;DB     075H,   00000000B,      0       ;EQ gain1 Band0
     ;DB     077H,   00000000B,      0       ;EQ gain1 Band1
     ;DB     079H,   00000000B,      0       ;EQ gain1 Band2
     ;DB     07bH,   00000000B,      0       ;EQ gain1 Band3
     ;DB     07dH,   00000000B,      0       ;EQ gain1 Band4
     ;DB     083H,   00000000B,      0       ;EQ Band0 Coef0H
     ;DB     085H,   00000000B,      0       ;EQ Band0 Coef1L
     ;DB     087H,   00000000B,      0       ;EQ Band0 Coef1H
     ;DB     089H,   00000000B,      0       ;EQ Band1 Coef0L
     ;DB     08BH,   00000000B,      0       ;EQ Band1 Coef0H
     ;DB     08DH,   00000000B,      0       ;EQ Band1 Coef1L
     ;DB     08FH,   00000000B,      0       ;EQ Band1 Coef1H
     ;DB     091H,   00000000B,      0       ;EQ Band2 Coef0L
     ;DB     093H,   00000000B,      0       ;EQ Band2 Coef0H
     ;DB     095H,   00000000B,      0       ;EQ Band2 Coef1L
     ;DB     097H,   00000000B,      0       ;EQ Band2 Coef1H
     ;DB     099H,   00000000B,      0       ;EQ Band3 Coef0L
     ;DB     09BH,   00000000B,      0       ;EQ Band3 Coef0H
     ;DB     09DH,   00000000B,      0       ;EQ Band3 Coef1L
     ;DB     09FH,   00000000B,      0       ;EQ Band3 Coef1H
     ;DB     0A1H,   00000000B,      0       ;EQ Band4 Coef0L
     ;DB     0A3H,   00000000B,      0       ;EQ Band4 Coef0H
     ;DB     0A5H,   00000000B,      0       ;EQ Band4 Coef1L
     ;DB     0A7H,   00000000B,      0       ;EQ Band4 Coef1H

     ;Playback Limiter settings
     DB      069H,   00011001B,      0       ;Volume Control Func Enable
     ;DB     0C1H,   00000000B,      0       ;Attack Time
     ;DB     0C3H,   00000000B,      0       ;Decay Time
     ;DB     0C5H,   00000000B,      0       ;Target Level
     ;DB     0C7H,   00000000B,      0       ;Max/Min Gain

     ;-------------------------------------------------------------
     ; Playback Analog Block ON
     ;-------------------------------------------------------------
     ;Turning on the VMID generator
     DB      021H,   00000001B,      5/5     ;Reference Power Management
     DB      021H,   00000010B,      5/5     ;Reference Power Management

     ;Mute settings
     DB      049H,   00000000B,      0       ;AMP Volume Control Function Enable
     DB      049H,   00000010B,      0       ;AMP Volume Control Function Enable

     ;DAC settings
     DB      025H,   00000010B,      0       ;DAC Power
     DB      055H,   00000010B,      0       ;Speaker AMP Output Control

     ;Turning on the speaker amplifier
     DB      027H,   013H,           0       ;Power Management
     DB      027H,   01FH,           0       ;Reference Power Management
     DB      03BH,   033H,           0       ;Speaker AMP Volume
     DB      04BH,   00000000B,      0       ;Amplifier Volume Fader Control
```

```
     DB      049H,   00000011B,     0       ;AMP Volume Control Function Enable
     DB      049H,   00000001B,     500/5   ;AMP Volume Control Function Enable

     ;-------------------------------------------------------------
     ; Playback start
     ;-------------------------------------------------------------
     DB      069H,   00011101B,     0       ;Volume Control Func Enable
     DB      069H,   00010101B,     0       ;Volume Control Func Enable
     DB      013H,   00000010B,     5/5     ;Record/Playback Run
     DB      06BH,   00110010B,     0       ;Mixer & Volume Control
     DB      069H,   00011101B,     0       ;Volume Control Func Enable
     DB      069H,   00001101B,     0       ;Volume Control Func Enable
TPLAYONE:

TPLAYOFF:
;-----------------------------------------------------------------------
;    Playback → System on
;-----------------------------------------------------------------------
     ;-------------------------------------------------------------
     ; Playback Stop
     ;-------------------------------------------------------------
     DB      06BH,   00000010B,     0       ;Mixer & Volume Control
     DB      069H,   00001000B,     0       ;Volume Control Func Enable
     DB      069H,   00011000B,     100/5   ;Volume Control Func Enable
     DB      013H,   00000000B,     0       ;Record/Playback Run

     ;-------------------------------------------------------------
     ; Playback Analog Block OFF
     ;-------------------------------------------------------------
     ; Turning off the speaker amplifier BTL mode
     DB      04BH,   00000000B,     0       ;Amplifier Volume Fader Control
     DB      049H,   00000001B,     0       ;AMP Volume Control Function Enable
     DB      049H,   00000011B,     100/5   ;AMP Volume Control Function Enable
     DB      027H,   013H,          0       ;AMP Power Management
     DB      027H,   000H,          0       ;AMP Power Management
     DB      025H,   00000000B,     0       ;DAC Power Management
     DB      021H,   00000000B,     0       ;Reference Power Management
TPLAYOFFE:


;*****************************************************************************
;
;
;    Initial settings of the peripheral hardware to be used
;
;
;*****************************************************************************
XMAINCSEG   UNIT
RESET_START:

;-----------------------------------------------------------------------------
;    Disabling interrupts
;-----------------------------------------------------------------------------
     DI
;-----------------------------------------------------------------------------
;    Register bank setting
;-----------------------------------------------------------------------------
     SEL     RB0
;-----------------------------------------------------------------------------
;    Stack pointer setting
;-----------------------------------------------------------------------------
     MOVW    SP,     #LOWW STACKTOP         ; Set the stack pointer

;-----------------------------------------------------------------------------
;    Clock frequency settings
;-----------------------------------------------------------------------------
;    Setting so that operations can be performed using the 20 MHz X1 oscillator
```

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

     MOV    CMC,    #01000001B            ;Clock operation mode
                    ;|||||||+-------------- AMPH: 10MHz<fMX≤20MHz
                    ;||||+++--------------- <000>
                    ;|||+------------------ OSCSELS: P123 and P124 pins set as input ports
                    ;||+------------------- <0>
                    ;++-------------------- EXCLK/OSCSEL: X1 oscillation mode (20 MHz)

     MOV    CSC,    #01000000B            ;Clock operation status control
                    ;|||||||+-------------- HIOSTOP: Internal high-speed oscillator operated
                    ;||+++++--------------- <00000>
                    ;|+-------------------- XTSTOP: XT1 oscillator stopped
                    ;+--------------------- MSTOP: X1 oscillator operated

     MOV    OSMC,   #00000001B            ;Operation speed mode
                    ;|||||||+-------------- FSEL: Operated at a frequency exceeding 10 MHz
                    ;+++++++--------------- <00000>

     MOV    OSTS,   #00000101B            ;Oscillation stabilization time: 2^15/fX

HRST300:
     NOP
     BF     OSTC.2,$HRST300               ;Waiting for clock oscillation to stabilize

     MOV    CKC,    #00011000B            ;Clock selection
                    ;|||||+++-------------- MDIV2-0: CPU/peripheral hardware clock (fCLK) = fMX
                    ;||||+----------------- <1>
                    ;|||+------------------ MCM0: High-speed system clock (fMX)
                    ;||+------------------- <R>
                    ;|+-------------------- CSS: Main system clock (fMAIN) = fCLK
                    ;+--------------------- <R>

;-----------------------------------------------------------------------------
;    Port 0 settings
;-----------------------------------------------------------------------------
     MOV    P0,     #00011010B            ;Output latches of P00, P02, P05, and P06 set to low
                                          ;level and those of P01, P03, and P04 set to high
                                          ;level
     MOV    PM0,    #10000000B            ;P00 to P06 set as output ports
                                          ;P01: Output LRCLK to the audio codec
                                          ;P02: Output SO10 to the EEPROM
                                          ;P03: Input SI10 from the EEPROM
                                          ;P04: Output SCK10 to the EEPROM
                                          ;P00, P05, and P06: Unused

;-----------------------------------------------------------------------------
;    Port 1 settings
;-----------------------------------------------------------------------------
     MOV    P1,     #00000000B            ;Output latches of P10 to P17 set to low level
     MOV    PM1,    #00000000B            ;P10 to P17 set as output ports
                                          ;P10: Output SCK00 to the audio codec
                                          ;P11: Input SI00 from the audio codec
                                          ;P12: Output SO00 to the audio codec
                                          ;P13 to P17: Unused

;-----------------------------------------------------------------------------
;    Port 2 settings
;-----------------------------------------------------------------------------
     MOV    P2,     #00000000B            ;Output latches of P20 to P27 set to low level
     MOV    PM2,    #00000000B            ;P20 to P27 set as output ports
                                          ;P20 to P27: Unused

;-----------------------------------------------------------------------------
;    Port 3 settings
;-----------------------------------------------------------------------------
     MOV    P3,     #00000000B            ;Output latches of P30 and P31 set to low level
```

```
     MOV     PM3,    #11111100B            ;P30 and P31 set as output ports
                                           ;P30 and P31: Unused

;-------------------------------------------------------------------------------
;    Port 4 settings
;-------------------------------------------------------------------------------
     MOV     P4,     #00000000B            ;Output latches of P40 to P47 set to low level
     MOV     PM4,    #00000000B            ;P40 to P47 set as output ports
                                           ;P40 to P47: Unused

;-------------------------------------------------------------------------------
;    Port 5 settings
;-------------------------------------------------------------------------------
     MOV     P5,     #00000000B            ;Output latches of P50 to P57 set to low level
     MOV     PM5,    #00000000B            ;P50 to P57 set as output ports
                                           ;P50 to P57: Unused

;-------------------------------------------------------------------------------
;    Port 6 settings
;-------------------------------------------------------------------------------
     MOV     P6,     #00000000B            ;Output latches of P60 to P67 set to low level
     MOV     PM6,    #00000000B            ;P60 to P67 set as output ports
                                           ;P60: Output SCL0 for setting the registers in the
                                           ;audio codec
                                           ;P61: Input and output SDA0 for setting the
                                           ;registers in the audio codec
                                           ;P62 to P67: Unused

;-------------------------------------------------------------------------------
;    Port 7 settings
;-------------------------------------------------------------------------------
     MOV     P7,     #00000000B            ;Output latches of P70 to P77 set to low level
     MOV     PM7,    #00000011B            ;P70 and P71 set as input ports and P72 to P77 set
                                           ;as output ports
                                           ;P70: Playback key input
                                           ;P71: Record key input
                                           ;P72: Output a signal to the operation-in-progress
                                           ;indicator (LED)
                                           ;P73: Output a reset signal to the audio codec
                                           ;P74: Output CS0 to the EEPROM
                                           ;P75: Output CS1 to the EEPROM
                                           ;P76: Output CS2 to the EEPROM
                                           ;P77: Output CS3 to the EEPROM

;-------------------------------------------------------------------------------
;    Port 8 settings
;-------------------------------------------------------------------------------
     MOV     P8,     #00000000B            ;Output latches of P80 to P87 set to low level
     MOV     PM8,    #00000000B            ;P80 to P87 set as output ports
                                           ;P80 to P87: Unused

;-------------------------------------------------------------------------------
;    Port 11 settings
;-------------------------------------------------------------------------------
     MOV     P11,    #00000000B            ;Output latches of P110 and P111 set to low level
     MOV     PM11,   #11111100B            ;P110 and P111 set as output ports
                                           ;P110 and P111: Unused

;-------------------------------------------------------------------------------
;    Port 12 settings
;-------------------------------------------------------------------------------
     MOV     P12,    #00000000B            ;Output latch of P120 set to low level
     MOV     PM12,   #11111110B            ;P120 set as an output port
                                           ;P120: Unused
```

```
;-------------------------------------------------------------------------------
;    Port 13 settings
;-------------------------------------------------------------------------------
     MOV    P13,    #00000000B            ;Output latches of P130 and P131 set to low level
     MOV    PM13,   #11111100B            ;P131 set as an output port
                                          ;P130 and P131: Unused
;-------------------------------------------------------------------------------
;    Port 14 settings
;-------------------------------------------------------------------------------
     MOV    P14,    #00000000B            ;Output latches of P140 to P145 set to low level
     MOV    PM14,   #11000000B            ;P140 to P145 set as output ports
                                          ;P140 to P145: Unused

;-------------------------------------------------------------------------------
;    Port 15 settings
;-------------------------------------------------------------------------------
     MOV    P15,    #00000000B            ;Output latches of P150 to P157 set to low level
     MOV    PM15,   #00000000B            ;P150 to P157 set as output ports
                                          ;P150 to P157: Unused

;-------------------------------------------------------------------------------
;    Settings of the registers in the audio codec
;-------------------------------------------------------------------------------
;
;    The following operations are performed.
;     •Supplying a clock to the audio codec
;     •Setting a timer for inserting waits in the program
;     •Outputting a reset signal to the audio codec
;     •Using the I2C interface for setting the registers in the audio codec
;
;-------------------------------------------------------------------------------

     MOV    RPLAYMOD,#CRESET                     ;Reset

     ; Timer array unit timer clock selection
     SET1   !TAU0EN                              ;Input clock supplied to the timer array unit
     MOV    !TPS0L,        #10010000B            ;Operation clock selection
                           ;||||++++-------------- CK00: fCLK
                           ;++++----------------- CK01: fCLK/2^9

     ;----------------------------------------
     ; Clock supply to the audio codec
     ;----------------------------------------
     ; TO02 output (10 MHz)
     ;----------------------------------------
     MOVW   AX,            #0000000000000000B    ;Operation mode setting
                           ;||||||||||||++++------ MD023 to MD020: Interval timer mode
                           ;||||||||||++---------- <Fixed to 00>
                           ;||||||||++------------ CIS021 and CIS020: Unused
                           ;|||||+++-------------- STS022 to STS020: Only software trigger
                           ;|||||                                    start enabled
                           ;||||+----------------- MASTER02: Standalone operation
                           ;|||+------------------ CSS02: Macro clock MCK specified by using t
                           ;|||                           the CKS02 bit
                           ;|++------------------- <Fixed to 00>
                           ;+--------------------- CKS02: Operation clock CK00 set by using
                           ;                             the PRS register

     MOVW   !TMR02,        AX
     MOVW   TDR02,#0                             ;Interval setting: 10 MHz output (20 MHz)
     SET1   TOE0L.2                              ;TO02 output
     SET1   TS0L.2                               ;Operation starts (trigger bit)

     ;----------------------------------------
     ; Waits used in the program
     ;----------------------------------------
     ; Using TM01
```

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

MOVW    AX,             #1000000000000000B     ;Operation mode setting
                        ;||||||||||||++++------ MD013 to MD010: Interval timer mode
                        ;||||||||||||++---------- <Fixed to 00>
                        ;||||||||++------------ CIS011 and CIS010: Unused
                        ;|||||+++-------------- STS012 to STS010: Only software trigger
                        ;|||||                                    start enabled
                        ;||||+----------------- MASTER01: Standalone operation
                        ;|||+------------------ CSS01: Macro clock MCK specified by using
                        ;|||                        the CKS01 bit
                        ;|++------------------- <Fixed to 00>
                        ;+--------------------- CKS01: Operation clock CK01 set by using
                        ;                              the PRS register
MOVW    !TMR01, AX

; Waiting for power supply to stabilize
MOVW    TDR01,          #195*210/5-1           ;Interval set (210 ms)
SET1    TS0L.1                                 ;Timer operation starts
CLR1    TMIF01                                 ;INTTM00 interrupt request cleared
BF      TMIF01,         $$

;Reset
CLR1    P_RESETB
;Waiting about 5 us (calculated by 50 ns*5*Breg)
MOV     B,              #5000/50/5
HRST550:
DEC     B                                      ;1clk
BNZ     $HRST550                               ;4clk(Z=0)
SET1    P_RESETB

;Waiting for power supply to stabilize
MOVW    TDR01,          #195*160/5-1           ;Interval set (160 ms)
SET1    TS0L.1                                 ;Timer operation starts
CLR1    TMIF01                                 ;INTTM00 interrupt request cleared
BF      TMIF01,         $$
SET1    TT0L.1                                 ;Timer operation stopped

;-----------------------------------------
; Transmission of the values set to the
; registers in the audio codec
;-----------------------------------------
; Using IIC0
;-----------------------------------------
SET1    !IIC0EN                                ;Input clock of serial interface IIC0
                                               ;supplied

CLR1    IICE0                                  ;Operation stopped

CLR1    P_SCL                                  ;SCL0 latch
CLR1    PM_SCL                                 ;SCL0 pin
CLR1    P_SDA                                  ;SDA0 pin
CLR1    PM_SDA                                 ;SDA0 pin

MOV     IICX0,          #00000000B                    ;Transfer clock selection
                        ;|||||||+--------------------- CLX0
                        ;+++++++--------------------- <0000000>
MOV     IICCL0,         #00001110B
                        ;||||||++-------------------- CL01-CL00(+CLX0): fCLK/96(first
                        ;||||||                                          mode)
                        ;|||||+----------------------- DFC0: Turn on the digital filter
                        ;||||+------------------------ SMC0: Specify operation in the first
                        ;||||                                 mode
                        ;|||+------------------------- DAD0: <R>Detect the SDA0 pin level
                        ;||+-------------------------- CLD0: <R>Detect the SCL0 pin level
                        ;++--------------------------- <00>
```

```
      MOV     IICF0,  #00000011B                    ;Communication reservation disable
                      ;|||||||+--------------------- IICRSV: Disable communication reservation
                      ;||||||+---------------------- STCEN: Enable initial start
                      ;||++++------------------------ <Fixed to 0000>
                      ;|+--------------------------- IICBSY: <R>IIC bus status flag
                      ;+---------------------------- STCF: <R>STT0 clear flag


      MOV     IICC0,  #00001000B                    ;Initial settings during master operation
                      ;|||||||+--------------------- SPT0: Stop condition trigger
                      ;||||||+---------------------- STT0: Start condition trigger
                      ;|||||+----------------------- ACKE0: Control acknowledgment
                      ;||||+------------------------ WTIM0: Control wait insertion and interrupt
                      ;||||                              request issuance: 9 clocks
                      ;|||+------------------------- SPIE0: Disable issuing of interrupt requests
                      ;|||                                  by detecting a stop condition
                      ;||+-------------------------- WREL0: Do not cancel waiting
                      ;|+--------------------------- LREL0: Save the communication: Normal
                      ;|                                    operation
                      ;+------------------------- IICE0: Enable operation of I2C

      SET1    SPT0                                  ;Stop condition set

      ;Settings of the registers in the audio codec: Setting to turn on the system
      MOV     ES,     #HIGHW TSYSON                 ;Higher 4 bits of the start address of the
                                                    ;ROM table holding the register setting
                                                    ;values
      MOVW    HL,     #LOWW TSYSON                  ;Lower 16 bits
      MOVW    DE,     #LOWW TSYSONE                 ;Lower 8 bits of the last address of the ROM
                                                    ;table holding the register setting values
      CALL    !!SI2CWRITE                           ;I2C write processing

      MOV     RPLAYMOD,#CSTOP                       ;Stopped

;-------------------------------------------------------------------------------
;     Settings of the I2S bus interface used for the audio data of the
;     audio codec
;-------------------------------------------------------------------------------
;
;     The following operations are performed.
;      •Setting the TO00 output and INTTM00 interrupt for outputting LRCLK
;      •Setting CSI00 for transmitting and receiving data
;
;-------------------------------------------------------------------------------
      ;----------------------------------------
      ; LRCLK output settings
      ;----------------------------------------
      ; TO00 output (16 kHz)
      ;----------------------------------------
      MOVW    AX,             #0000000000000000B    ;Operation mode setting
                              ;||||||||||||++++------ MD003 to MD000: Interval timer mode
                              ;||||||||||++---------- <Fixed to 00>
                              ;||||||||++------------ CIS001 and CIS000: Unused
                              ;|||||+++-------------- STS002 to STS000: Only software trigger
                              ;|||||                                   start enabled
                              ;||||+----------------- MASTER00: Standalone operation
                              ;|||+------------------ CSS00: Macro clock MCK specified by using
                              ;|||                          the CKS00 bit
                              ;||++------------------- <Fixed to 00>
                              ;+--------------------- CKS00: Operating clock CK00 set by using
                              ;                             the PRS register
      MOVW    !TMR00,         AX
      MOVW    TDR00,#625-1                          ;Interval set: 16 kHz output (32 kHz)


      CLR1    TMPR100                               ;Priority order set to the highest level
      CLR1    TMPR000
```

```
;---------------------------------------
; CSI00 settings
;---------------------------------------
SET1    !SAU0EN                             ;Input clock of the serial array unit
                                            ;supplied

NOP                                         ;Waiting
NOP
NOP
NOP

MOV     !SPS0L,         #00000000B          ;Operation clock selection: fCLK
                        ;||||++++-------------- PRS003 to PRS000: fCLK
                        ;++++------------------ PRS013 to PRS010: Unused

MOVW    SDR00,          #(12-1) shl 9               ;Bits 15 to 7: Transfer clock set
                                                    ;(833 kHz)


MOVW    AX,             #0000000000100001B  ;Operation mode selected: CSI mode
                        ;||||||||||||||||+------ MD000: Buffer empty interrupt
                        ;||||||||||||||++------- MD002 and MD001: CSI mode
                        ;||||||||||+++---------- <Fixed to 100>
                        ;||||||||||+------------ SIS000: Unused
                        ;||||||||+-------------- <Fixed to 0>
                        ;||||||||+-------------- STS00: Only software trigger enabled (fixed
                        ;||||||||                       in CSI mode)
                        ;||+++++---------------- <Fixed to 00000>
                        ;|+--------------------- CSS00: Transfer clock set to a clock
                        ;|                               obtained by dividing operation clock
                        ;|                               MCK as specified by using the CKS00
                        ;|                               bit
                        ;+---------------------- CKS00: Operation clock set to prescaler
                        ;                               output clock CK00 set by using the
                        ;                               PRS register

MOVW    !SMR00,         AX

;Initial data output (at the same time as setting CSI10)
;MOVW   AX,             #0000000100000001B  ;Setting of initial outputs of the SO and SCK pins
;                       ;||||||||++++++++------ SO0n to SO00: Used for transmission
;                       ;++++++++-------------- CKOn to CKO0: Serial clock output of
                        ;                                     channels n to 0
;MOVW   !SO0,           AX

SET1    SOE0L.0                             ;Output enabled

SET1    P_SCK00                             ;SCK00 latch: High level
CLR1    PM_SCK00                            ;SCK00 pin output set
SET1    PM_SI00                             ;SI00 pin input set
SET1    P_SO00                              ;SO00 latch: High level
CLR1    PM_SO00                             ;SO00 pin output set

SET1    SS0L.0                              ;CSI00 operation starts (trigger bit)
CLR1    CSIIF00                             ;INTCSI00 interrupt request cleared
CLR1    CSIMK00                             ;INTCSI00 interrupt servicing enabled

CLR1    CSIPR100                            ;Priority order set to next after INTTM00
SET1    CSIPR000
```

```
;-------------------------------------------------------------------------------
;    Setting of the SPI interface for EEPROM
;-------------------------------------------------------------------------------
;
;    The following operations are performed.
;     •Setting CSI10 for transmitting and receiving data
;     •Setting DMA0 for successive reception and DMA1 for successive
;      transmission
;
;-------------------------------------------------------------------------------
     OR      P_CS,   #11110000B                    ;All EEPROMs deactivated

     ;----------------------------------------
     ; DMA0 settings (for CSI reception)
     ;----------------------------------------
     SET1    DEN0                                  ;Operation of DMA channel 0 enabled

     MOV     DSA0,   #044H                         ;DMA SFR address: SDR02(SIO10)=0FFF44H
     MOV     DMC0,   #00001000B                    ;Setting of transfer mode of DMA channel 0
                     ;||||++++---------------------- IFC03 to IFC00: DMA start source: INTCSI10
                     ;||||                                            transfer complete interrupt
                     ;|||+------------------------- DWAIT0: DMA transfer suspension: DMA
                     ;|||                                   transfer performed according to a
                     ;|||                                   DMA start request (not suspended)
                     ;||+-------------------------- DS0: Transfer data size: 8 bits
                     ;|+--------------------------- DRS0: DMA transfer direction selected: SFR →
                     ;|                                   Internal RAM
                     ;+---------------------------- STG0: Software trigger not operated

     ;-----------------------------------------------------
     ;  DMA1 settings
     ;  (for CSI transmission and setting dummy data
     ;  during CSI reception)
     ;-----------------------------------------------------
     SET1    DEN1                                  ; Operation of DMA channel 1 enabled

     MOV     DSA1,   #044H                         ;DMA SFR address: SDR02(SIO10)=0FFF44H
     MOV     DMC1,   #01001000B                    ;Setting of transfer mode of DMA channel 1
                     ;||||++++---------------------- IFC13 to IFC10: DMA start source: INTCSI10
                     ;||||                                            transfer complete interrupt
                     ;|||+------------------------- DWAIT1: DMA transfer suspension: DMA
                     ;|||                            transfer performed according to a DMA start
                     ;|||                            request (not suspended)
                     ;||+-------------------------- DS1: Transfer data size: 8 bits
                     ;|+--------------------------- DRS1: DMA transfer direction selected:
                     ;|                                   Internal RAM →SFR
                     ;+---------------------------- STG1: Software trigger not operated

     ;----------------------------------------
     ; CSI10 settings
     ;----------------------------------------
     MOVW    SDR02,  #(4-1) shl 9                  ;Bits 15 to 7: Transfer clock set (2.5 MHz)
```

```
MOVW    AX,     #0000000000100000B            ;Operation mode selected: CSI mode
                ;|||||||||||||||+------------- MD020: Transfer complete interrupt
                ;||||||||||||||++-------------- MD022 and MD021: CSI mode
                ;||||||||||+++---------------- <Fixed to 100>
                ;|||||||||+------------------- SIS020: Unused
                ;||||||||+-------------------- <Fixed to 0>
                ;|||||||+--------------------- STS02: Only software trigger enabled
                ;|||||||                             (fixed in CSI mode)
                ;||+++++----------------------- <Fixed to 00000>
                ;|+--------------------------- CSS02: Transfer clock set to a clock
                ;|                                    obtained by dividing operation clock
                ;|                                    MCK as specified by using the CKS00
                ;|                                    bit
                ;+---------------------------- CKS02: Operation clock set to prescaler
                ;                                    output clock CK00 set by using
                ;                                    the PRS register
MOVW    !SMR02, AX

MOVW    AX,     #1011000000000111B            ; Communication format setting
                ;||||||||||||+++-------------- DLS022 to DLS020: 8-bit data length
                ;||||||||||||+----------------- <Fixed to 0>
                ;||||||||||++------------------ SLC021 and SLC020: Unused (fixed to 0)
                ;|||||||||+-------------------- <Fixed to 0>
                ;||||||||+--------------------- DIR02: Input and output performed MSB first
                ;||||||++----------------------- PTC021 and PTC020: Unused (fixed to 00)
                ;|||||+------------------------ EOC02: Unused (fixed to 0)
                ;||||+------------------------- <Fixed to 0>
                ;||++-------------------------- CKP02/DAP02: Phases of data and clock in
                ;||                                          CSI mode selected
                ;||                             [11 selected]
                ;||                                  _   _   _   _   _   _   _   _
                ;||                             SCK02____| |_| |_| |_| |_| |_| |_| |_| |
                ;||                                  __ ___ ___ ___ ___ ___ ___ ___ ____
                ;||                             SO02 __|___|___|___|___|___|___|___|____
                ;||                                      D7  D6  D5  D4  D3  D2  D1  D0
                ;||                 SI02 input timing ____|___|___|___|___|___|___|___|__
                ;||
                ;++---------------------------- TXE00/RXE00: Only transmission performed
MOVW    !SCR02, AX

; Initial data output
MOVW    AX,     #0000000100000001B            ;Setting of initial outputs of the SO and SCK pins
                ;||||||||++++++++------------- SO0n to SO00: Used for transmission
                ;++++++++--------------------- CKOn to CKO0: Serial clock output of
                ;                                            channels n to 0
MOVW    !SO0,   AX
SET1    SOE0L.2                               ;Output enabled

SET1    P_SCK10                               ;SCK00 latch: High level
CLR1    PM_SCK10                              ;SCK00 pin output set
SET1    PM_SI10                               ;SI00 pin input set
SET1    P_SO10                                ;SO00 latch: High level
CLR1    PM_SO10                               ;SO00 pin output set

SET1    SS0L.2                                ;CSI10 operation starts (trigger bit)

;---------------------------------------
; Erasing all EEPROMs (all 0FFH)
;---------------------------------------
;Enabling writing to an EEPROM
AND     P_CS,   #00001111B                    ;All EEPROMs selected
CLR1    CSIIF10
MOV     SIO10,  #00000110B                    ;Write enable (WREN) instruction (INTCSI10
                                              ;occurs)
BF      CSIIF10,$$
OR      P_CS,   #11110000B                    ;Setting completed
```

```
     ;Erasing all EEPROMs
     AND     P_CS,   #00001111B                  ;All EEPROMs set
     CLR1    CSIIF10
     MOV     SIO10,  #11000111B                  ;All CEs erased (INTCSI10 occurs)
     BF      CSIIF10,$$
     OR      P_CS,   #11110000B                  ;Setting completed

     SET1    ST0L.2                              ;CSI10 operation stopped (trigger bit)
     SET1    CSIMK10                             ;INTCSI10 interrupt servicing disabled

;-------------------------------------------------------------------------------
;    Starting key retrieval
;-------------------------------------------------------------------------------
     MOV     KRM,    #00000011B                  ;KR0 and KR1 enabled
     CLR1    KRIF                                ;INTKR interrupt request cleared
     SET1    KRMK                                ;INTKR interrupt servicing disabled

     SET1    P_LED                               ;Operation-in-progress indicator turned off
                                                 ;(key input possible)

;-------------------------------------------------------------------------------
;    Enabling interrupts
;-------------------------------------------------------------------------------
     EI

;*******************************************************************************
;
;
;    Main processing
;
;
;*******************************************************************************
MAIN_LOOP:
     ;---------------------------------------
     ; Key processing
     ;---------------------------------------
     BT      KRIF,   $LMAIN100                   ;Key input? Yes,
     BR      LMAINRECE                           ; No,
LMAIN100:
     CLR1    KRIF                                ;INTKR interrupt request cleared

     CMP     RPLAYMOD,#CSTOP                     ;Stopped?
     BZ      $LMAIN200                           ; Yes,
     BR      LMAINRECE                           ; No, (key disabled)
LMAIN200:
     MOV     A,      P_KEY
     AND     A,      #00000011B                  ;Is key valid?
     SKNZ                                        ; Yes,
     BR      LMAINRECE                           ; No, (key disabled: pressed multiple times)

     ;Key judged to have been pressed
     ;Waiting about 10 ms (chattering removed)
     MOVW    TDR01,  #195*10/5-1                 ;Interval set (10 ms)
     SET1    TS0L.1                              ;Timer operation starts
     CLR1    TMIF01                              ;INTTM00 interrupt request cleared
     BF      TMIF01, $$
     SET1    TT0L.1                              ;Timer operation stopped

     MOV     A,      P_KEY                       ;KR port
     AND     A,      #00000011B
     CMP     A,      #00000011B                  ;Is key valid?
     BNZ     $LMAIN400                           ; Yes,
     BR      LMAINRECE                           ; No, (key disabled: pressed multiple times)
LMAIN400:
     BF      P_RECKY, $LMAINREC                  ;Was record key pressed? Yes,
     BF      P_PLAYKY,$LMAINPLAY                 ;Was playback key pressed? Yes,
     BR      LMAINRECE                           ; No,
```

```
     ;---------------------------------------
     ; Starting recording
     ;---------------------------------------
LMAINREC:
     CLR1    P_LED                              ;Operation-in-progress indicator turned on
                                                ;(key input disabled)

     MOV     KRM,    #00000000B                 ;KR0 and KR1 disabled
     MOV     RPLAYMOD,#CREC                     ;Voice data recorded

     ;Settings of the registers in the audio codec: Recording set
     MOV     ES,     #HIGHW TRECON              ;Higher 4 bits of the start address of the
                                                ;ROM table holding the register setting
                                                ;values
     MOVW    HL,     #LOWW TRECON               ;Lower 16 bits
     MOVW    DE,     #LOWW TRECONE              ;Lower 8 bits of the last address of the ROM
                                                ;table holding the register setting values
     CALL    !!SI2CWRITE                        ;I2C write processing

     ;Preparation of the I2S bus interface
     MOV     RCSI00CNT,#0                       ;CSI00 reception counter
     MOVW    SDR00,  #(12 to 1) shl 9           ;Bits 15 to 7: Transfer clock set (833 kHz)

     MOVW    AX,     #0100000000000111B         ;Communication format setting
                     ;||||||||||||+++------------- DLS002 to DLS000: 8-bit data length
                     ;||||||||||||+---------------- <Fixed to 0>
                     ;||||||||||++----------------- SLC001 and SLC000: Unused (fixed to 0)
                     ;|||||||||+------------------- <Fixed to 0>
                     ;||||||||+-------------------- DIR00: Input and output performed MSB first
                     ;||||||++--------------------- PTC001 and PTC000: Unused (fixed to 00)
                     ;|||||+----------------------- EOC00: Unused (fixed to 0)
                     ;||||+------------------------ <Fixed to 0>
                     ;||++------------------------- CKP00/DAP00: Phases of data and clock in CSI
                     ;||                                          mode selected
                     ;||                             [00 selected]
                     ;||                                  __   _   _   _   _   _   _   _   __
                     ;||                             SCK00  |_| |_| |_| |_| |_| |_| |_| |_|
                     ;||                                 __ ___ ___ ___ ___ ___ ___ ___ ____
                     ;||                             SO00 __|___|___|___|___|___|___|___|____
                     ;||                                      D7  D6  D5  D4  D3  D2  D1  D0
                     ;||                 SI00 input timing ____|___|___|___|___|___|___|___|__
                     ;||
                     ;++--------------------------- TXE00/RXE00: Only reception performed
     MOVW    !SCR00, AX

     MOV     TO0L,   #00000000B                 ;Initial output set to low level
     SET1    TOE0L.0                            ;Operation of TO00 enabled by a count
                                                ;operation (LRCLK)
     CLR1    TMIF00                             ;INTTM00 interrupt request cleared
     CLR1    TMMK00                             ;INTTM00 interrupt servicing enabled

     ;Setting writing to an EEPROM
     MOVW    RI2SADDR,#RRECMEM1                 ;Address to which the data received via I2S
                                                ;is saved
     CLR1    FI2SPAGE
     MOV     RP_CS,  #11100000B                 ;EEPROM selected
     MOV     REEPSEQ,#CEEPSEQ_RESET             ;EEPROM transfer sequence
     MOVW    REEPADDR,#0                        ;EEPROM write address (REEPADDR*100H)
```

```
     MOVW    AX,     #1011000000000111B            ;Communication format setting
                     ;||||||||||||+++------------- DLS022 to DLS020: 8-bit data length
                     ;||||||||||||+---------------- <Fixed to 0>
                     ;||||||||||++----------------- SLC021 and SLC020: Unused (fixed to 0)
                     ;|||||||||+------------------- <Fixed to 0>
                     ;||||||||+-------------------- DIR02: Input and output performed MSB first
                     ;||||||++--------------------- PTC021 and PTC020: Unused (fixed to 00)
                     ;|||||+----------------------- EOC02: Unused (fixed to 0)
                     ;||||+------------------------ <Fixed to 0>
                     ;||++------------------------- CKP02/DAP02: Phases of data and clock in CSI
                     ;||                                          mode selected
                     ;||                             [11 selected]
                     ;||                                      _   _   _   _   _   _   _   _
                     ;||                            SCK02 ____| |_| |_| |_| |_| |_| |_| |_| |
                     ;||                                  __ ___ ___ ___ ___ ___ ___ ___ ____
                     ;||                            SO02  __|___|___|___|___|___|___|___|____
                     ;||                                      D7  D6  D5  D4  D3  D2  D1  D0
                     ;||                 SI02 input timing ____|___|___|___|___|___|___|___|__
                     ;||
                     ;++--------------------------- TXE00/RXE00: Only transmission performed
     MOVW    !SCR02, AX

     ;Starting I2S operation
     SET1    SS0L.0                                ;CSI00 operation starts (trigger bit)
     CLR1    P_LRCLK                               ;LRCLK operation starts
     SET1    TS0L.0                                ;TO00 output operation starts (trigger bit)
     MOV     SIO00,  #0FFH                         ;Dummy data set (INTCSI00 occurs)
     BR      LMAINRET

     ;---------------------------------------
     ; Starting playback
     ;---------------------------------------
LMAINPLAY:
     CLR1    P_LED                                 ;Operation-in-progress indicator turned on
                                                   ;(key input disabled)

     MOV     KRM,    #00000000B                    ;KR0 and KR1 disabled
     MOV     RPLAYMOD,#CPLAY_START                 ;Preparing for playback started

     ;Settings of the registers in the audio codec: Playback set
     MOV     ES,     #HIGHW TPLAYON                ;Higher 4 bits of the start address of the
                                                   ;ROM table holding the register setting
                                                   ;values
     MOVW    HL,     #LOWW TPLAYON                 ;Lower 16 bits
     MOVW    DE,     #LOWW TPLAYONE                ;Lower 8 bits of the last address of the ROM
                                                   ;table holding the register setting values
     CALL    !!SI2CWRITE                           ;I2C write processing

     ;Preparation of the I2S bus interface
     MOV     RCSI00CNT,#0                          ;CSI00 transmission counter
     MOVW    SDR00,  #(12-1) shl 9                 ;Bits 15 to 7: Transfer clock set (833 kHz)
```

```
MOVW    AX,     #1000000000000111B            ;Communication format setting
                ;||||||||||||||+++------------- DLS002 to DLS000: 8-bit data length
                ;||||||||||||||+---------------- <Fixed to 0>
                ;||||||||||||++----------------- SLC001 and SLC000: Unused (fixed to 0)
                ;|||||||||||+------------------- <Fixed to 0>
                ;||||||||||+-------------------- DIR00: Input and output performed MSB first
                ;||||||||++---------------------- PTC001 and PTC000: Unused (fixed to 00)
                ;|||||||+------------------------ EOC00: Unused (fixed to 0)
                ;||||||+------------------------- <Fixed to 0>
                ;|||||++-------------------------- CKP00/DAP00: Phases of data and clock in CSI
                ;|||                                            mode selected
                ;|||                             [00 selected]
                ;|||                                  __   _   _   _   _   _   _   _   __
                ;|||                             SCK00  |_| |_| |_| |_| |_| |_| |_| |_|
                ;|||                                  __ ___ ___ ___ ___ ___ ___ ___ ____
                ;|||                             SO00 __|___|___|___|___|___|___|___|____
                ;|||                                     D7  D6  D5  D4  D3  D2  D1  D0
                ;|||                 SI00 input timing ____|___|___|___|___|___|___|___|__
                ;|||
                ;|||------------------------- TXE00/RXE00: Only transmission performed
MOVW    !SCR00, AX

MOV     TO0L,   #00000000B                    ;Initial output set to low level
SET1    TOE0L.0                               ;Operation of TO00 enabled by a count
                                              ;operation
CLR1    TMIF00                                ;INTTM00 interrupt request cleared
CLR1    TMMK00                                ;INTTM00 interrupt servicing enabled

;Setting reading from an EEPROM (writing set at first)
MOV     A,      P_CS                          ;EEPROM selected
AND     A,      #00001111B
OR      A,      #11100000B
MOV     P_CS,   A
MOV     RP_CS,  #10110000B                    ;EEPROM selected (enabled after 2 pages are
                                              ;read)
SET1    FI2SPAGE                              ;Data of the EEPROM set to be saved to page 1
MOV     REEPSEQ,#CEEPSEQ_INST                 ;Instruction bytes transmitted
MOVW    REEPADDR,#0                           ;EEPROM read address (REEPADDR*100H)
MOVW    AX,     #1011000000000111B            ;Communication format setting
                ;||||||||||||||+++------------- DLS022 to DLS020: 8-bit data length
                ;||||||||||||||+---------------- <Fixed to 0>
                ;||||||||||||++----------------- SLC021 and SLC020: Unused (fixed to 0)
                ;|||||||||||+------------------- <Fixed to 0>
                ;||||||||||+-------------------- DIR02: Input and output performed MSB first
                ;||||||||++---------------------- PTC021 and PTC020: Unused (fixed to 00)
                ;|||||||+------------------------ EOC02: Unused (fixed to 0)
                ;||||||+------------------------- <Fixed to 0>
                ;|||||++-------------------------- CKP02/DAP02: Phases of data and clock in CSI
                ;|||                                            mode selected
                ;|||                             [11 selected]
                ;|||                                       _   _   _   _   _   _   _   _
                ;|||                             SCK02____| |_| |_| |_| |_| |_| |_| |_| |
                ;|||                                  __ ___ ___ ___ ___ ___ ___ ___ ____
                ;|||                             SO02 __|___|___|___|___|___|___|___|____
                ;|||                                     D7  D6  D5  D4  D3  D2  D1  D0
                ;|||                 SI02 input timing____|___|___|___|___|___|___|___|__
                ;|||
                ;|||--------------------------- TXE00/RXE00: Only transmission performed
MOVW    !SCR02,AX

;Starting reading from the EEPROM
CLR1    CSIIF10                               ;INTCSI10 interrupt request cleared
CLR1    CSIMK10                               ;INTCSI10 interrupt servicing enabled
SET1    SS0L.2                                ;CSI10 operation starts (trigger bit)
MOV     SIO10,  #00000011B                    ;Transmit data (reading specified) (INTCSI10
                                              ;occurs)
BR      LMAINRET
```

```
     ;---------------------------------------
     ; Recording end settings
     ;---------------------------------------
LMAINRECE:
     CMP     RPLAYMOD,#CREC_END                  ;Recording finished?
     BNZ     $LMAINPLAYE                         ; No,

     ;Settings of the registers in the audio codec: Setting termination of recording
     MOV     ES,     #HIGHW TRECOFF              ;Higher 4 bits of the start address of the
                                                 ;ROM table holding the register setting
                                                 ;values
     MOVW    HL,     #LOWW TRECOFF               ;Lower 16 bits
     MOVW    DE,     #LOWW TRECOFFE              ;Lower 8 bits of the last address of the ROM
                                                 ;table holding the register setting values
     CALL    !!SI2CWRITE                         ;I2C write processing
     BR      LMAINSTOP

     ;---------------------------------------
     ; Playback end settings
     ;---------------------------------------
LMAINPLAYE:
     CMP     RPLAYMOD,#CPLAY_END                 ;Playback finished?
     BNZ     $LMAINRET                           ; No,

     ;Settings of the registers in the audio codec: Setting termination of playback
     MOV     ES,     #HIGHW TPLAYOFF             ;Higher 4 bits of the start address of the
                                                 ;ROM table holding the register setting
                                                 ;values
     MOVW    HL,     #LOWW TPLAYOFF              ;Lower 16 bits
     MOVW    DE,     #LOWW TPLAYOFFE             ;Lower 8 bits of the last address of the ROM
                                                 ;table holding the register setting values
     CALL    !!SI2CWRITE                         ;I2C write processing
LMAINSTOP:
     MOV     KRM,    #00000011B                  ;KR0 and KR1 enabled
     MOV     RPLAYMOD,#CSTOP                     ;Stopped
     SET1    P_LED                               ;Operation-in-progress indicator turned off
                                                 ;(key input possible)
LMAINRET:
     BR      MAIN_LOOP

;*******************************************************************************
;
;    INTTM00 interrupt servicing
;    (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and SDIN)
;
;-------------------------------------------------------------------------------
;
;    LRCLK output via the I2S bus interface and BCLK, SDOUT, and SDIN used for
;    transmitting and receiving data are synchronized by restarting the
;    operation of CSI00, which was stopped by INTCSI00 interrupt servicing,
;    When LRCLK changes from high to low level and from low to high level .
;
;                    ________                 _______________
;    LRCLK                   |_______________|               |_______
;
;                    _________                __              __
;    BCLK                     |||||||||||||  |||||||||||||  |||||||
;
;                    _________ _ _      _ _ __ _ _      _ _ __ _ _
;    SDOUT/SDIN               |_|_• • • _|_|  |_|_• • • _|_|  |_|_• •
;
;
;    INTTM00         ________|_______________|_______________|_______
;
;    Register bank 1 used
;
```

```
;*****************************************************************************
IINTTM00:
     SEL     RB1

     CMP     RPLAYMOD,#CREC                      ;During recording? No, (during playback)
     BNZ     $HTM00PLAY
     ;---------------------------------------
     ; During recording
     ;---------------------------------------
     ;Starting the operation of the I2S bus interface
     MOV     A,      #0FFH                       ;Dummy data
     SET1    SS0L.0                              ;CSI00 operation starts (trigger bit)
     XCH     A,      SIO00                       ;Third byte received and dummy data set
                                                 ;(INTCSI00 occurs)
     MOVW    HL,     RI2SADDR                    ;Address to which the data received via I2S
                                                 ;is saved
     MOV     [HL],   A                           ;Data saved
     BR      HTM00RET

HTM00PLAY:
     ;---------------------------------------
     ; During playback
     ;---------------------------------------
     ;Starting the operation of the I2S bus interface
     MOVW    HL,     RI2SADDR                    ;Address to which the data transmitted via
                                                 ;I2S is saved
     MOV     A,      [HL]                        ;Transmit data (first byte)
     SET1    SS0L.0                              ;CSI00 operation starts (trigger bit)
     MOV     SIO00,  A                           ;Transmit data set (INTCSI00 occurs)
HTM00RET:
     INCW    RI2SADDR                            ;Address to which the data transmitted or
                                                 ;received via I2S is saved updated
     RETI
```

```
;*****************************************************************************
;
;    INTCSI00 interrupt servicing
;    (using INTCSI00 for CSI transmission or reception)
;
;-----------------------------------------------------------------------------
;
;    Data is transmitted and received via the I2S bus interface between the
;    microcontroller and the audio codec.
;    During recording, CSI00 is used to save the received data.  Normally, data is
;    successively received by setting dummy data.  To achieve synchronization with
;    LRCLK, however, only reception is performed every three bytes and successive
;    reception operation is stopped.  The stopped operation is restarted by INTTM00
;    interrupt servicing and can be synchronized with LRCLK.  When the received data
;    reaches one page, writing to an EEPROM starts.  During playback, CSI00 is used
;    to successively transmit data, but successive transmission operation is
;    stopped every three bytes to achieve synchronization with LRCLK, similarly as
;    during recording.  The stopped operation is restarted by INTTM00 interrupt
;    servicing and can be synchronized with LRCLK.  When the transmitted data
;    reaches one page, reading data of a different page from the EEPROM starts.
;
;                    ________                             ______
;    LRCLK                   |____________________________|
;
;                    _________                            _
;    BCLK                     |||||||||||||||||||||||||||| ||||||
;
;                    _________ _      _ _      _ _      _ _ _
;    SDOUT/SDIN               |_• • • _|_• • • _|_• • • _| |_• •
;
;    INTCSI00        _________|________|________|__________|_____
;
;                             1        2        3          1
;
;    Register bank 1 used
;
;*****************************************************************************
IINTCSI00:
     SEL     RB1

     INC     RCSI00CNT                           ;CSI00 reception counter updated

     CMP     RPLAYMOD,#CREC                      ;During recording?
     BNZ     $HI2SPLAY                           ; No,

     ;---------------------------------------
     ; During recording
     ;---------------------------------------
     ;Determining the last position of the page
     MOVW    AX,     RI2SADDR                    ;Address to which the data received via I2S
                                                 ;is saved
     BT      FI2SPAGE,$HI2SR030
     CMPW    AX,     #RRECMEM1E                  ;End of page 1?
     BC      $HI2SR100                           ; No,
     MOVW    RI2SADDR,#RRECMEM2                  ;Address to which the data received via I2S
                                                 ;is saved: Page 2
     SET1    FI2SPAGE
     BR      HI2SR050
HI2SR030:
     CMPW    AX,     #RRECMEM2E                  ;End of page 2?
     BC      $HI2SR100                           ; No,
     MOVW    RI2SADDR,#RRECMEM1                  ;Address to which the data received via I2S
                                                 ;is saved: Page 1
     CLR1    FI2SPAGE
HI2SR050:
     ;Starting writing to the EEPROM
     MOV     REEPSEQ,#CEEPSEQ_WREN               ;Write enable (WREN) signal transmitted
```

```
     MOV     A,      P_CS                        ;EEPROM CS set
     AND     A,      #00001111B
     OR      A,      RP_CS
     MOV     P_CS,   A
     CLR1    CSIIF10                             ;INTCSI10 interrupt request cleared
     CLR1    CSIMK10                             ;INTCSI10 interrupt servicing enabled
     SET1    SS0L.2                              ;CSI10 operation starts (trigger bit)
     MOV     SIO10,  #00000110B                  ;Write enable (WREN) instruction (INTCSI10
                                                 ;occurs)
HI2SR100:
     ;Buffer empty interrupt immediately after starting operation
     CMP     RCSI00CNT,#1                        ;Buffer empty interrupt immediately after
                                                 ;starting reception?
     BNZ     $HI2SR200                           ; No,
     MOV     SIO00,  #0FFH                       ;Dummy data set
     BR      HI2SRET                             ; Yes, (ignored because the data is
                                                 ;undefined)
HI2SR200:
     ;Reception of the first byte
     CMP     RCSI00CNT,#2                        ;First byte received?
     BNZ     $HI2SR300                           ; No,
     MOV     A,      #0FFH                       ;Dummy data
     XCH     A,      SIO00                       ;Dummy data set and data received
     BR      HI2SR500
HI2SR300:
     ;Reception of the second byte
     MOV     RCSI00CNT,#0                        ;CSI00 reception counter initialized
     MOV     A,      SIO00                       ;Data reception only
     ;BR     HI2SR500
HI2SR500:
     MOVW    HL,     RI2SADDR                    ;Address to which the data received via I2S
                                                 ;is saved
     MOV     [HL],   A                           ;Received data saved
HI2SR800:
     INCW    RI2SADDR                            ;Address to which the data received via I2S
                                                 ;is saved updated
     BR      HI2SRET

     ;---------------------------------------
     ; During playback
     ;---------------------------------------
HI2SPLAY:
     CMP     RPLAYMOD,#CPLAY                     ;During playback?
     BNZ     $HI2SRET                            ; No,

     CMP     RCSI00CNT,#3                        ;Timing of synchronization with LRCLK?
                                                 ;(Immediately after transmission of the third
                                                 ;byte starts?)
     BNC     $HI2SP100

     MOVW    HL,     RI2SADDR                    ;Address to which the data transmitted via
                                                 ;I2S is saved
     MOV     A,      [HL]                        ;Transmit data (second or third byte)
     MOV     SIO00,  A                           ;Transmit data set (INTCSI00 occurs)
     INCW    RI2SADDR                            ;Address to which the data transmitted via
                                                 ;I2S is saved updated
     BR      HI2SP200
HI2SP100:
     MOV     RCSI00CNT,#0                        ;CSI00 transmission counter initialized
HI2SP200:
     MOVW    AX,     RI2SADDR                    ;Address to which the data transmitted via
                                                 ;I2S is saved
     BT      FI2SPAGE,$HI2SP300                  ;Page 2 being transmitted?
     CMPW    AX,     #RRECMEM1E                  ;End of page 1?
     BC      $HI2SRET                            ; No,
     MOVW    RI2SADDR,#RRECMEM2                  ;Address to which the data transmitted via
                                                 ;I2S is saved: Page 2
```

```
     SET1    FI2SPAGE
     BR      HI2SP500
HI2SP300:
     CMPW    AX,     #RRECMEM2E                  ;End of page 2?
     BC      $HI2SRET                            ; No,
     MOVW    RI2SADDR,#RRECMEM1                  ;Address to which the data transmitted via
                                                 ;I2S is saved: Page 1
     CLR1    FI2SPAGE
HI2SP500:
     ;Starting reading from the EEPROM
     MOV     A,      P_CS                        ;EEPROM selected
     AND     A,      #00001111B
     OR      A,      RP_CS
     MOV     P_CS,   A
     MOV     REEPSEQ,#CEEPSEQ_INST               ;Instruction bytes transmitted
     CLR1    CSIIF10                             ;INTCSI10 interrupt request cleared
     CLR1    CSIMK10                             ;INTCSI10 interrupt servicing enabled
     SET1    SS0L.2                              ;CSI10 operation starts (trigger bit)
     MOV     SIO10,  #00000011B                  ;Transmit data (reading specified) set
                                                 ;(INTCSI10 occurs)
HI2SRET:
     RETI

;*******************************************************************************
;
;    INTCSI10 interrupt servicing
;    (using INTCSI10 for CSI transmission or reception)
;
;-------------------------------------------------------------------------------
;
;    During recording, instruction bytes and 24-bit addresses are transmitted
;    to an EEPROM, DMA channel 1 starts when transmitting data to the EEPROM
;    starts, and transmission continues in CSI10 single transmission mode.
;    During playback, 24-bit addresses are transmitted to an EEPROM, DMA
;    channels 0 and 1 start when receiving data from the EEPROM starts, and
;    reception continues in CSI10 single reception mode.  However, DMA
;    channel 1 is used for setting dummy data.
;
;    Register bank 1 used
;
;*******************************************************************************
IINTCSI10:
     EI                                          ;Multiple interrupts enabled
     SEL     RB2

     INC     REEPSEQ                             ;EEPROM transfer sequence updated

     CMP     RPLAYMOD,#CREC                      ;During recording? No, (during playback)
     BZ      $HCSI1REC
     BR      HCSI1PLAY

HCSI1REC:
     ;--------------------------------------
     ; During recording
     ;--------------------------------------
     ;Instruction byte transmission
     CMP     REEPSEQ,#CEEPSEQ_INST               ;Instruction byte transmission timing?
     BNZ     $HCSI1100                           ; No,
     OR      P_CS,   #11110000B                  ;Non-active
     NOP
     MOV     A,      P_CS                        ;EEPROM selected
     AND     A,      #00001111B
     OR      A,      RP_CS
     MOV     P_CS,   A
     MOV     A,      #00000010B                  ;Writing specified
     BR      HCSI1650                            ;Data transmitted
HCSI1100:
```

```
     ;Transmission of the higher bits of the 24-bit address
     CMP     REEPSEQ,#CEEPSEQ_ADDRH              ;Transmitting the higher bits of the 24-bit
                                                 ;address?
     BNZ     $HCSI1200                           ; No,
     MOV     A,      REEPADDR+1
     BR      HCSI1650                            ;Data transmitted
HCSI1200:
     ;Transmission of the middle bits of the 24-bit address
     CMP     REEPSEQ,#CEEPSEQ_ADDRM              ;Transmitting the middle bits of the 24-bit
                                                 ;address?
     BNZ     $HCSI1300                           ; No,
     MOV     A,      REEPADDR
     BR      HCSI1650                            ;Data transmitted
HCSI1300:
     ;Transmission of the lower bits of the 24-bit address
     CMP     REEPSEQ,#CEEPSEQ_ADDRL              ;Transmitting the lower bits of the 24-bit
                                                 ;address?
     BNZ     $HCSI1400                           ; No,
     MOV     A,      #000H
     BR      HCSI1650                            ;Data transmitted
HCSI1400:
     ;Starting data transmission
     ;Starting DMA transfer
     BT      FI2SPAGE,$HCSI1430                  ;Is I2S using page 2?
     MOVW    DRA1,   #RRECMEM2+1                 ;Transmit buffer
     MOV     A,      !RRECMEM2                   ;First transmit data
     BR      HCSI1450
HCSI1430:
     MOVW    DRA1,   #RRECMEM1+1                 ;Transmit buffer
     MOV     A,      !RRECMEM1                   ;First transmit data
HCSI1450:
     MOVW    DBC1,   #3*2*42-1                   ;One page
     SET1    DST1                                ;DMA1 transfer enabled (for CSI10
                                                 ;transmission)

     CLR1    DMAIF1                              ;INTDM1 interrupt request cleared
     CLR1    DMAMK1                              ;INTDM1 interrupt servicing enabled
     CLR1    CSIIF10                             ;INTCSI10 interrupt request cleared
     SET1    CSIMK10                             ;INTCSI10 interrupt servicing disabled

     SET1    DMAMK0                              ;INTDM0 interrupt servicing disabled

HCSI1650:
     SET1    SS0L.2                              ;CSI00 operation starts (trigger bit)
     MOV     SIO10,  A                           ;Transmit data set (INTCSI00 occurs)
HCSI1800:
     BR      HCSI1RET

     ;---------------------------------------
     ; During playback
     ;---------------------------------------
HCSI1PLAY:
     ;Transmission of the higher bits of the 24-bit address
     CMP     REEPSEQ,#CEEPSEQ_ADDRH              ;Transmitting the higher bits of the 24-bit
                                                 ;address?
     BNZ     $HCSI1P200                          ; No,
     MOV     A,      REEPADDR+1
     BR      HCSI1P350                           ;Data transmitted
HCSI1P200:
     ;Transmission of the middle bits of the 24-bit address
     CMP     REEPSEQ,#CEEPSEQ_ADDRM              ;Transmitting the middle bits of the 24-bit
                                                 ;address?
     BNZ     $HCSI1P300                          ; No,
     MOV     A,      REEPADDR
     BR      HCSI1P350                           ;Data transmitted
HCSI1P300:
     ;Transmission of the lower bits of the 24-bit address
```

```
     CMP     REEPSEQ,#CEEPSEQ_ADDRL                ;Transmitting the lower bits of the 24-bit
                                                   ;address?
     BNZ     $HCSI1P400                            ; No,
     MOV     A,      #000H
HCSI1P350:
     ;Data transmission
     MOV     SIO10,  A                             ;Transmit data set (INTCSI10 occurs)
     BR      HCSI1RET
HCSI1P400:
     ;Starting data reception
     MOVW    AX,     #0111000000000111B            ;Communication format setting
                     ;||||||||||||+++------------- DLS022 to DLS020: 8-bit data length
                     ;||||||||||||+---------------- <Fixed to 0>
                     ;||||||||||++----------------- SLC021 and SLC020: Unused (fixed to 0)
                     ;|||||||||+------------------- <Fixed to 0>
                     ;||||||||+-------------------- DIR02: Input and output performed with MSB
                     ;||||||||                             first
                     ;||||||++---------------------- PTC021 and PTC020: Unused (fixed to 00)
                     ;|||||+----------------------- EOC02: Unused (fixed to 0)
                     ;||||+------------------------ <Fixed to 0>
                     ;||++------------------------- CKP02/DAP02: Phases of data and clock in CSI
                     ;||                                          mode selected
                     ;||                             [11 selected]
                     ;||                                       _   _   _   _   _   _   _   _
                     ;||                             SCK02 ____| |_| |_| |_| |_| |_| |_| |_| |
                     ;||                                   __ ___ ___ ___ ___ ___ ___ ___ ____
                     ;||                             SO02  __|___|___|___|___|___|___|___|____
                     ;||                                       D7  D6  D5  D4  D3  D2  D1  D0
                     ;||                SI02 input timing  ____|___|___|___|___|___|___|___|__
                     ;||
                     ;++--------------------------- TXE00/RXE00: Only reception performed
     MOVW    !SCR02, AX

     MOVW    DRA1,   #RRECMEM1                     ;Address for setting dummy data
     MOVW    DBC1,   #3*2*42-1                     ;Page size set

     BT      FI2SPAGE,$HCSI1P430                   ;Is I2S using page 2?
     MOVW    DRA0,   #RRECMEM2                     ;Receive buffer set to page 1
     BR      HCSI1P450
HCSI1P430:
     MOVW    DRA0,   #RRECMEM1                     ;Receive buffer set to page 2
HCSI1P450:
     MOVW    DBC0,   #3*2*42                       ;Page size set

     ;Starting DMA transfer
     SET1    DST1                                  ;DMA1 transfer enabled (for CSI10
                                                   ;transmission)
     SET1    DST0                                  ;DMA0 transfer enabled (for CSI10 reception)

     SET1    DMAMK1                                ;INTDM1 interrupt servicing disabled
     CLR1    CSIIF10                               ;INTCSI10 interrupt request cleared
     SET1    CSIMK10                               ;INTCSI10 interrupt servicing disabled

     CLR1    DMAIF0                                ;INTDM0 interrupt request cleared
     CLR1    DMAMK0                                ;INTDM0 interrupt servicing enabled

     SET1    SS0L.2                                ;CSI10 operation starts (trigger bit)
     MOV     SIO10,  #0FFH                         ;Dummy data set (INTCSI00 occurs)
HCSI1RET:
     RETI
```

```
;*****************************************************************************
;
;    INTDMA0 interrupt servicing
;    (using INTDMA0 for CSI reception)
;
;-----------------------------------------------------------------------------
;
;    This interrupt occurs once data of one page has been received from an
;    EEPROM during playback.  Which of the four EEPROMs is to be used is
;    selected and operation of the I2S bus interface is started and stopped.
;
;    Register bank 1 used
;
;*****************************************************************************
IINTDMA0:
     SEL     RB1

     ;Waiting for communication completion
HDMA0100:
     MOV     A,      SSR02L                      ;Communication completed?
     BT      A.6,    $HDMA0100                   ; No,

     ;Completing receiving data of one page
     OR      P_CS,   #11110000B                  ;Non-active
     MOV     REEPSEQ,#CEEPSEQ_INST

     CMP     RPLAYMOD,#CPLAY                     ;During playback?
     BZ      $HDMA0500                           ; Yes,
     CMP     RPLAYMOD,#CPLAY_SET                 ;Preparing for playback?
     BZ      $HDMA0300                           ; Yes,
     CMP     RPLAYMOD,#CPLAY_START               ;Starting preparing for playback?
     BNZ     $HDMA0RET                           ; No,

     ;Starting reading from an EEPROM
     MOV     RPLAYMOD,#CPLAY_SET                 ;Prepared for playback
     MOV     A,      P_CS                        ;EEPROM selected
     AND     A,      #00001111B
     OR      A,      #11010000B
     MOV     P_CS,   A
     CLR1    FI2SPAGE
     CLR1    CSIIF10                             ;INTCSI10 interrupt request cleared
     CLR1    CSIMK10                             ;INTCSI10 interrupt servicing enabled
     SET1    SS0L.2                              ;CSI10 operation starts (trigger bit)
     MOV     SIO10,  #00000011B                  ;Transmit data (reading specified) set
                                                 ;(INTCSI10 occurs)
     BR      HDMA0RET

     ;-----------------------------------------------
     ; Starting the operation of the I2S bus interface
     ;-----------------------------------------------
HDMA0300:
     MOV     RPLAYMOD,#CPLAY                     ;During playback
     MOVW    RI2SADDR,#RRECMEM1+1                ;Start of the transmit buffer set
     CLR1    FI2SPAGE
     MOV     A,      !RRECMEM1
     ;Starting the operation of the I2S bus interface
     SET1    SS0L.0                              ;CSI00 operation starts (trigger bit)
     CLR1    P_LRCLK                             ;LRCLK operation starts
     SET1    TS0L.0                              ;TO00 output operation starts (trigger bit)
     MOV     SIO00,  A                           ;Transmit data set (INTCSI00 occurs)

     BR      HDMA0RET
HDMA0500:
     ;EEPROM selection
     MOV     A,      RP_CS
     SET1    A.3
     ROLC    A,      1
```

```
     BC      $HDMA0700                          ;Have all EEPROMs been selected once? No,
     INCW    REEPADDR
     MOVW    AX,     REEPADDR
     CMPW    AX,     #128000/256                ;Last EEPROM?
     BNC     $HDMA0800                          ; Yes,
     MOV     A,      #11100000B
HDMA0700:
     AND     A,      #11110000B
     MOV     RP_CS,  A
     BR      HDMA0RET
HDMA0800:
     ;Stopping the operation of the I2S bus interface
     SET1    P_LRCLK                            ;LRCLK output disabled (high level)
     SET1    TT0L.0                             ;TO00 output operation stopped (trigger bit)
     SET1    ST0L.0                             ;CSI000 output operation stopped (trigger
                                                ;bit)
     SET1    ST0L.2                             ;CSI010 output operation stopped (trigger
                                                ;bit)
     MOV     RPLAYMOD,#CPLAY_END                ;Playback being terminated
HDMA0RET:
     CLR1    DST1                               ;DMA1 transfer stopped
     CLR1    DST0                               ;DMA0 transfer stopped
     SET1    DMAMK0                             ;INTDM0 interrupt servicing disabled
     RETI

;*****************************************************************************
;
;    INTDMA1 interrupt servicing
;    (using INTDMA1 for CSI transmission)
;
;-----------------------------------------------------------------------------
;
;    This interrupt occurs when data of one page has been transmitted to an
;    EEPROM during recording.      Which of the four EEPROMs is to be used is
;    selected and operation of the I2S bus interface is stopped.
;
;    Register bank 1 used
;
;*****************************************************************************
IINTDMA1:
     SEL     RB1

     ;Waiting for communication completion
HDMA1100:
     MOV     A,      SSR02L                     ;Communication completed?
     BT      A.6,    $HDMA1100                  ; No,

     ;Completing transmitting data of one page
     OR      P_CS,   #11110000B                 ;Data of one page transmitted
     MOV     REEPSEQ,#CEEPSEQ_RESET

     ;EEPROM selection
     MOV     A,      RP_CS
     SET1    A.3
     ROLC    A,      1
     BC      $HDMA1300                          ;Have all EEPROMs been selected once? No,
     INCW    REEPADDR
     MOVW    AX,     REEPADDR
     CMPW    AX,     #128000/256                ;Last EEPROM?
     BNC     $HDMA1500                          ; Yes,
     MOV     A,      #11100000B
HDMA1300:
     AND     A,      #11110000B
     MOV     RP_CS,  A
     BR      HDMA1RET
HDMA1500:
     ;Stopping the operation of the I2S bus interface
```

```
     SET1    P_LRCLK                              ;LRCLK output disabled (high level)
     SET1    TT0L.0                               ;TO00 output operation stopped (trigger bit)
     SET1    ST0L.0                               ;CSI000 output operation stopped (trigger
                                                  ;bit)
     SET1    ST0L.2                               ;CSI010 output operation stopped (trigger
                                                  ;bit)
     MOV     RPLAYMOD,#CREC_END                   ;Recording stopped
HDMA1RET:
     CLR1    DST1                                 ;DMA1 transfer stopped
     CLR1    DST0                                 ;DMA0 transfer stopped
     SET1    DMAMK1                               ;INTDM1 interrupt servicing disabled
     RETI

;*****************************************************************************
;
;    I2C bus interface write processing for setting the registers in the audio
;    codec
;
;-----------------------------------------------------------------------------
;
;    The data set to the registers in the audio codec is written by using the
;    functions of I2C bus interface IIC0.
;
;-----------------------------------------------------------------------------
;
;    [I N]   ES      :Higher 4 bits of the start address of the ROM table
;                     holding the register setting values
;            HL      :Lower 8 bits of the start address of the ROM table
;                     holding the register setting values
;            DE      :Lower 8 bits of the last address of the ROM table
;                     holding the register setting values
;    [OUT]   -
;
;*****************************************************************************
SI2CWRITE:
     SET1    IICE0                                ; Operation enabled
JI2CW100:
     ;Start condition issuance
     SET1    STT0

     ;Slave address transmission
     BF      STD0,   $$                           ;Address transmission period? No,
     CLR1    IICIF0
     MOV     IIC0,   #00110100B                   ;Address transmission starts (writing
                                                  ;specified)
     BF      IICIF0, $$                           ;Transmission completed? No,
     BT      ACKD0,  $JI2CW200                    ;ACK detected? Yes,
     SET1    SPT0                                 ; No, stop condition issued
     BR      $JI2CW100                            ;Retransmission

JI2CW200:
     ;Write address transmission
     MOV     A,      ES:[HL]                      ;Set data acquired
     CLR1    IICIF0                               ;INTIIC10 interrupt request cleared
     MOV     IIC0,   A                            ;Data transmission starts
     BF      IICIF0, $$                           ;Transmission completed? No,
     BT      ACKD0,  $JI2CW300                    ;ACK detected? Yes,
     SET1    SPT0                                 ; No, stop condition issued
     BR      $JI2CW100                            ;Retransmission

JI2CW300:
     ;Write data transmission
     INCW    HL                                   ;Reference position updated
     MOV     A,      ES:[HL]                      ;Set value acquired
     CLR1    IICIF0                               ;INTIIC10 interrupt request cleared
     MOV     IIC0,   A                            ;Data transmission starts
     BF      IICIF0, $$                           ;Transmission completed? No,
```

```
     BT      ACKD0,  $JI2CW400                    ;ACK detected? Yes,
     SET1    SPT0                                 ; No, stop condition issued
     DECW    HL                                   ;Retransmission
     BR      $JI2CW100

JI2CW400:
     ;Stop condition issuance
     SET1    SPT0

     ;Wait control
     INCW    HL                                   ;Reference position updated
     MOV     A,      ES:[HL]                      ;Wait information acquired
     CMP0    A                                    ;Waiting required?
     BZ      $JI2CW600                            ; No,
     MOV     X,      #195                         ;5 ms
     MULU    X                                    ;A register*5 ms
     DECW    AX                                   ;AX register: Timer count value
     MOVW    TDR01,  AX                           ;Interval set
     CLR1    TMIF01                               ;INTTM01 interrupt request cleared
     SET1    TS0L.1                               ;Timer operation starts
     BF      TMIF01,$$                            ;Interval elapsed? No,
     SET1    TT0L.1                               ;Timer operation stopped

JI2CW600:
     INCW    HL                                   ;Reference position updated
     MOVW    AX,     HL
     CMPW    AX,     DE                           ;Register setting completed?
     BC      $JI2CW100                            ; No,
JI2CWRET:
     RET

end
```

• main.c (C language version)

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

     NEC Electronics     78K0R/KG3 Series

*****************************************************************************
     78K0R/KG3 Series
*****************************************************************************
     I2S bus interface between microcontroller and audio codec
*****************************************************************************

[Overview]

This sample program uses the I2S bus interface to transfer audio data between the
microcontroller and an audio codec.  The LR clock (LRCLK) for selecting whether to use channel
L or channel R of the I2S bus interface is used in the interval timer mode of the timer array
unit (TAU) and is output from TO00.  CSI00 of channel 0 of serial array unit 0 (SAU0) is used
for outputting the clock used for transferring data (BCLK) and for receiving data (SDOUT), and
transmitting data (SDIN).
ML2612 made by OKI Semiconductor is used as the audio codec.  The I2C bus interface is used for
setting the registers in the audio codec and the I2S bus interface is used for transferring
audio data.  When the record key is pressed, the audio codec receives the voice data input from
a microphone and saves the data to EEPROM.  When the playback key is pressed, the recorded data
is transmitted to the audio codec and the output as sound from the speakers.  The SPI interface
between the microcontroller and EEPROM uses CSI10 of channel 2 of serial array unit 0 (SAU0) to
continuously perform CSI transmission or reception by using the DMA controller.  Whether
recording or playback is in progress is indicated by an LED (operation-in-progress indicator)
to which a signal is output from P72.

The following are used to execute the sample program.

I2C bus interface for setting the registers in the audio codec
•Serial interface IIC0

I2S bus interface used for transferring audio data between the microcontroller and audio codec
•Using the output from TO00 of the timer array unit (TAU) as LRCLK
•Using CSI00 of channel 0 of serial array unit 0 (SAU0) for transmitting and receiving data

SPI interface used between the microcontroller and EEPROMs
•Using P74 to P77 as chip select pins because four EEPROMs are used
•Using CSI10 of channel 2 of serial array unit 0 (SAU0)
•Using DMA0 and DMA1 of the DMA controller for performing successive transmission or reception

Wait timer
•Using the interval timer mode of channel 1 of the timer array unit (TAU) to insert waits of at
 least 5 ms in the program

Record and playback keys
•Key interrupt input pins KR0 and KR1

Operation-in-progress indicator (LED)
•Outputting a signal from P72 to light the LED used as an operation-in-progress indicator


<Main initial settings for the peripheral hardware to be used>
●Disabling interrupts
●Setting the CPU or peripheral hardware clock frequency to the X1 oscillation clock (when used
  at 20 MHz)
●Setting ports
●Setting the audio codec
•Supplying a clock to the audio codec
•Setting a timer for inserting waits in the program
•Outputting a reset signal to the audio codec
•Setting the I2C interface used for setting registers
•Turning on the system by setting the registers in the audio codec
●Setting the I2S bus interface used for the audio data of the audio codec
•Using TO00 output (16 kHz) for setting the output of LRCLK
```

```
•Setting CSI00
●Setting the SPI interface of EEPROM
•Setting CSI10 for transmitting and receiving data
•Setting DMA0 for successive reception and DMA1 for successive transmission
•Erasing all EEPROMs (all 0FFH)
●Starting key data retrieval
●Enabling interrupts


<Main processing>
•Key processing
•Setting start of recording
•Setting start of playback
•Setting termination of recording
•Setting termination of playback


<Main INTTM00 interrupt servicing (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and
SDIN)>
•Starting the I2S bus interface


<Main INTCSI00 interrupt servicing (using INTCSI00 for CSI transmission or reception)>
•Transmitting or receiving data via the I2S bus interface
•Saving the received data
•Starting writing to EEPROM
•Starting reading from EEPROM


<Main INTCSI10 interrupt servicing (using INTCSI10 for CSI transmission or reception)>
•Transmitting instruction bytes to EEPROM
•Transmitting 24-bit addresses to EEPROM
•Starting receiving data from EEPROM by using DMA0 or DMA1
•Starting transmitting data to EEPROM by using DMA1


<Main INTDMA0 interrupt servicing (using INTDMA0 for CSI reception)>
•Starting reading from EEPROM
•Starting the I2S bus interface
•Selecting EEPROM
•Stopping the I2S bus interface


<Main INTDMA1 interrupt servicing (using INTDMA1 for CSI transmission)>
•Selecting EEPROM
•Stopping the I2S bus interface


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

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

     Preprocessing directives (#pragma directives)

===========================================================================*/
#pragma SFR                                        /* Enable description of special function
                                                      register (SFR) names */
#pragma DI                                         /* Enable description of DI instructions */
#pragma EI                                         /* Enable description of EI instructions */
#pragma NOP                                        /* Enable description of NOP instructions */
#pragma interrupt INTTM00  fn_inttm00     RB1      /* Interrupt function declaration:INTTM00 */
#pragma interrupt INTCSI00 fn_intcsi00    RB1      /* Interrupt function declaration:INTCSI00 */
#pragma interrupt INTCSI10 fn_intcsi10    RB2      /* Interrupt function declaration:INTCSI10 */
#pragma interrupt INTDMA0  fn_intdma0     RB1      /* Interrupt function declaration:INTDMA0 */
#pragma interrupt INTDMA1  fn_intdma1     RB1      /* Interrupt function declaration:INTDMA1 */

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

```
     Function prototype declaration

==============================================================================*/
void fn_I2cWrite(unsigned char *addr, unsigned char size);

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

     Port definitions

==============================================================================*/
/* Audio codec reset */
#define      P_RESETB       P7.3           /* Reset */
#define      PM_RESETB      PM7.3          /* Reset output */

/* I2C interface used for setting the registers in the audio codec */
#define P_SCL               P6.0           /* SCL0 latch */
#define PM_SCL              PM6.0          /* SCL0 output */
#define P_SDA               P6.1           /* SDA0 latch */
#define PM_SDA              PM6.1          /* SDA0 I/O */

/* Audio codec I2S bus interface */
#define P_LRCLK             P0.1           /* LRCLK */
#define PM_LRCLK            PM0.1          /* LRCLK output */
#define P_SCK00             P1.0           /* SCK00 */
#define PM_SCK00            PM1.0          /* SCK00 output */
#define PM_SI00             PM1.1          /* SI00 input */
#define P_SO00              P1.2           /* SO00 */
#define PM_SO00             PM1.2          /* SO00 output */

/* Key inputs (used: 2) */
#define P_KEY               P7             /* KR (P70 and P71 used) */
#define PM_KEY              PM7            /* KR input (PM70 and PM71 used) */
#define  P_PLAYKY           P7.0           /* Playback key */
#define  PM_PLAYKY          PM7.0          /* Playback key input */
#define  P_RECKY            P7.1           /* Record key */
#define  PM_RECKY           PM7.1          /* Record key input */

/* Output to the LED used as an operation-in-progress indicator (used: 1) */
#define P_LED               P7.2           /* Operation-in-progress indicator (active low) */
#define PM_LED              PM7.2          /* Output to the operation-in-progress indicator
                                              (LED) */
                                           /* LED lit during operation and turned off when key
                                              input is possible. */

/* EEPROM SPI interfaces (used: 4) */
#define P_CS                P7             /* Chip select CS (P73 to P77 used) */
#define PM_CS               PM7            /* Chip select CS output (PM73 to PM77 used) */
#define  P_CS3              P7.7           /* CS3 */
#define  PM_CS3             PM7.7          /* CS3 output */
#define  P_CS2              P7.6           /* CS2 */
#define  PM_CS2             PM7.6          /* CS2 output */
#define  P_CS1              P7.5           /* CS1 */
#define  PM_CS1             PM7.5          /* CS1 output */
#define  P_CS0              P7.4           /* CS0 */
#define  PM_CS0             PM7.4          /* CS0 output */
#define P_SCK10             P0.4           /* SCK10 */
#define PM_SCK10            PM0.4          /* SCK10 output */
#define PM_SI10             PM0.3          /* SI10 input */
#define P_SO10              P0.2           /* SO10 */
#define PM_SO10             PM0.2          /* SO10 output */

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

     RAM definitions

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

```
/* Overall operation */
unsigned char ucPlayMode;                   /* Operation status */
#define CRESET              0               /*  Reset */
#define CSTOP               1               /*  Stopped */
#define CREC                2               /*  Recording under execution */
#define CREC_END            3               /*   Recording finished */
#define CPLAY_START         4               /*  Preparing for playback started */
#define CPLAY_SET           5               /*   Preparing for playback */
#define CPLAY               6               /*   Playback under execution */
#define CPLAY_END           7               /*   Playback finished */

/* Definitions related to the operation of the I2S bus interface used between the
  microcontroller and audio codec */
unsigned char *ucI2sAddress;                /* Address to which data is transmitted or received
                                               via I2S */
unsigned char ucI2sByteCouter;              /* Counter for transmitting and receiving 1-byte I2S
                                               data */
boolean bI2sMemoryPage;                     /* Page being transmitted or received via the I2S
                                               bus: Page 1 (0)/page 2 (1) */

/* Definitions related to the operation of the SPI interface used between the microcontroller
  and EEPROMs */
unsigned short ushEepromAddress;            /* EEPROM read or write address
                                               (ushEepromAddress*0x100) */
unsigned char ucEepromCs;                   /* Chip select CS setting value of the EEPROMs */
unsigned char ucEepromSeq;                  /* EEPROM transfer sequence */
#define CEEPSEQ_RESET       0               /*  Reset status */
#define CEEPSEQ_WREN        1               /*  Write enable signal transmitted */
#define CEEPSEQ_INST        2               /*  Instruction bytes transmitted */
#define CEEPSEQ_ADDR        3               /*  Higher 8 bits of 24-bit addresses transmitted */
#define CEEPSEQ_ADDR        4               /*  Middle 8 bits of 24-bit addresses transmitted */
#define CEEPSEQ_ADDRL       5               /*  Lower 8 bits of 24-bit addresses transmitted */
#define CEEPSEQ_DATA        6               /*  Data transmitted and received by using DMA */

/* Memory area of internal RAM */
unsigned char ucMemoryPage1[3*2*42];        /* Page 1 (42 LRCLK clocks) */
unsigned char ucMemoryPage2[3*2*42];        /* Page 2 (42 LRCLK clocks) */


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

     ROM definitions

==============================================================================*/
/*------------------------------------------------------------------------------

     These are the ROM tables that hold the values to be used for setting the registers in the
     audio codec used to transition the status of the ML2612 audio codec made by OKI
     Semiconductor.
     These tables are used in the write processing performed via the I2C bus interface.

     [Example]
             {      0E5H    ,00000111B     ,0      }              ;Trimming 1
            ,{      0E9H    ,00000001B,    ,0      }              ;Trimming 2
                    ~~~~|    ~~~~~~~~~|     ~|
                        |            |      +------------ Wait information (Note)
                        |            +------------------- Write data
                        +---------------------------------- Write address

            Note: Indicates the wait time after data is written.
                  A wait of (5 ms * (described value)) is inserted.
                  No wait is inserted if the value is 0.

     * The setting data shown in this sample program is simply an example of the settings that
     can be used to execute recording or playback.  See the data sheet and other materials for
     details.
```

```
-----------------------------------------------------------------------------*/
static const unsigned char aSystemOnTbl[][3]=
{
/*-------------------------------------------------------------------------
     Power off → System on
-------------------------------------------------------------------------*/
     /*-------------------------------------------------------------
      Trimming
     -------------------------------------------------------------*/
      {      0x0E5   ,0b00000111    ,0      }       /* Trimming 1 */
     ,{      0x0E9   ,0b00000001    ,0      }       /* Trimming 2 */

     /*-------------------------------------------------------------
      Digital Block ON
     -------------------------------------------------------------*/
     ,{      0x001   ,0x003         ,0      }       /* Sampling Rate */
     ,{      0x003   ,0x001         ,0      }       /* PLLNL */
     ,{      0x005   ,0x000         ,0      }       /* PLLNH */
     ,{      0x007   ,0x000         ,0      }       /* PLLML */
     ,{      0x009   ,0x002         ,0      }       /* PLLMH */
     ,{      0x00B   ,0x002         ,0      }       /* PLLDIV */
     ,{      0x00F   ,0b00000001    ,0      }       /* CLK Input/Output control */
     ,{      0x00D   ,0b00000001    ,0      }       /* Clock Enable */
     ,{      0x00D   ,0b00000101    ,150/5  }       /* Clock Enable */
};

static const unsigned char aRecordOnTbl[][3]=
{
/*-------------------------------------------------------------------------
     System on → Recording start
-------------------------------------------------------------------------*/
     /*-------------------------------------------------------------
      Record Function Setting
     -------------------------------------------------------------*/
     /* Volume settings */
      {      0x033   ,0x03F         ,0      }       /* Mic Input Volume */
     ,{      0x039   ,0b00000000    ,0      }       /* Mic Boost Volume */
     ,{      0x049   ,0b00000001    ,0      }       /* AMP Volume Control Function Enable */
     ,{      0x04B   ,0b00000000    ,0      }       /* Amplifier Volume Fader Control */
     ,{      0x069   ,0b00011010    ,0      }       /* Volume Control Func Enable */
     ,{      0x06B   ,0b00000010    ,0      }       /* Mixer & Volume Control */
     ,{      0x06D   ,0x0FF         ,0      }       /* Playback Digital Volume Control */

     /* HPF1 ON */
     ,{      0x067   ,0b00000001    ,0      }       /* Filter Func Enable */

     /* HPF2 ON */
//   ,{      0x067   ,0b00000000    ,0      }       /* Filter Func Enable */
//   ,{      0x07F   ,0b00000000    ,0      }       /* HPF2 CutOff */

     /* Programmable Equalizer settings */
//   ,{      0x06D   ,0b11111111    ,0      }       /* Record Digital Volume */
//   ,{      0x067   ,0b00000001    ,0      }       /* Filter Func Enable */
//   ,{      0x067   ,0b00000000    ,0      }       /* Filter Func Enable */
//   ,{      0x07F   ,0b00000000    ,0      }       /* HPF2 CutOff */
//   ,{      0x067   ,0b00000000    ,0      }       /* Filter Func Enable */
//   ,{      0x075   ,0b00000000    ,0      }       /* EQ gain1 Band0 */
//   ,{      0x077   ,0b00000000    ,0      }       /* EQ gain1 Band1 */
//   ,{      0x079   ,0b00000000    ,0      }       /* EQ gain1 Band2 */
//   ,{      0x07B   ,0b00000000    ,0      }       /* EQ gain1 Band3 */
//   ,{      0x07D   ,0b00000000    ,0      }       /* EQ gain1 Band4 */
//   ,{      0x083   ,0b00000000    ,0      }       /* EQ Band0 Coef0H */
//   ,{      0x085   ,0b00000000    ,0      }       /* EQ Band0 Coef1L */
//   ,{      0x087   ,0b00000000    ,0      }       /* EQ Band0 Coef1H */
//   ,{      0x089   ,0b00000000    ,0      }       /* EQ Band1 Coef0L */
//   ,{      0x08B   ,0b00000000    ,0      }       /* EQ Band1 Coef0H */
//   ,{      0x08D   ,0b00000000    ,0      }       /* EQ Band1 Coef1L */
```

```
//   ,{      0x08F   ,0b00000000    ,0      }       /* EQ Band1 Coef1H */
//   ,{      0x091   ,0b00000000    ,0      }       /* EQ Band2 Coef0L */
//   ,{      0x093   ,0b00000000    ,0      }       /* EQ Band2 Coef0H */
//   ,{      0x095   ,0b00000000    ,0      }       /* EQ Band2 Coef1L */
//   ,{      0x097   ,0b00000000    ,0      }       /* EQ Band2 Coef1H */
//   ,{      0x099   ,0b00000000    ,0      }       /* EQ Band3 Coef0L */
//   ,{      0x09B   ,0b00000000    ,0      }       /* EQ Band3 Coef0H */
//   ,{      0x09D   ,0b00000000    ,0      }       /* EQ Band3 Coef1L */
//   ,{      0x09F   ,0b00000000    ,0      }       /* EQ Band3 Coef1H */
//   ,{      0x0A1   ,0b00000000    ,0      }       /* EQ Band4 Coef0L */
//   ,{      0x0A3   ,0b00000000    ,0      }       /* EQ Band4 Coef0H */
//   ,{      0x0A5   ,0b00000000    ,0      }       /* EQ Band4 Coef1L */
//   ,{      0x0A7   ,0b00000000    ,0      }       /* EQ Band4 Coef1H */

     /* ALC settings */
     ,{      0x069   ,0b00000010    ,0      }       /* Volume Control Func Enable */
//   ,{      0x0B1   ,0b00000000    ,0      }       /* ALC Mode */
//   ,{      0x0B3   ,0b00000000    ,0      }       /* ALC Attack Time */
//   ,{      0x0B5   ,0b00000000    ,0      }       /* ALC Decay Time */
//   ,{      0x0B7   ,0b00000000    ,0      }       /* ALC Hold Time */
//   ,{      0x0B9   ,0b00000000    ,0      }       /* ALC Target Level */
//   ,{      0x0BB   ,0b00000000    ,0      }       /* ALC Max/Min Gain */
//   ,{      0x0BD   ,0b00000000    ,0      }       /* Noise Gate Threshold */
//   ,{      0x0BF   ,0b00000000    ,0      }       /* ALC Zero Cross Time OutPlayback Limiter
                                                      Control Register */
//   ,{      0x02F   ,0b00000000    ,0      }       /* ZC-CMP Power Management

     /*-------------------------------------------------------------
      Record Analog Block ON
     -------------------------------------------------------------*/
     /* Turning on the VMID generator */
     ,{      0x021   ,0b00000001    ,5/5    }       /* Reference Power Management */
     ,{      0x021   ,0b00000010    ,5/5    }       /* Reference Power Management */

     /* Turning on the microphone bias circuit */
     ,{      0x021   ,0b00000110    ,0      }       /* Reference Power Management */

     /* Minimum input setting (differential input) */
     ,{      0x05B   ,0b00000010    ,0      }       /* Mic IF Control */

     /* Power Management Register */
     ,{      0x023   ,0b00001000    ,0      }       /* Input Power Management */
     ,{      0x023   ,0b00001010    ,0      }       /* Input Power Management */

     /*-------------------------------------------------------------
      Record start
     -------------------------------------------------------------*/
     ,{      0x069   ,0b00010000    ,0      }       /* Volume Control Func Enable */
     ,{      0x069   ,0b00010000    ,0      }       /* Volume Control Func Enable */
     ,{      0x013   ,0b00000001    ,5/5    }       /* Record/Playback Run */
     ,{      0x06B   ,0b00110010    ,0      }       /* Mixer & Volume Control */
     ,{      0x069   ,0b00011000    ,0      }       /* Volume Control Func Enable */
     ,{      0x069   ,0b00001000    ,0      }       /* Volume Control Func Enable */
};

static const unsigned char aRecordOffTbl[][3]=
{
/*-----------------------------------------------------------------------
     Recording → System on
-----------------------------------------------------------------------*/
     /*-------------------------------------------------------------
      Record stop
     -------------------------------------------------------------*/
      {      0x06B   ,0b00000010    ,0      }       /* Mixer & Volume Control */
     ,{      0x069   ,0b00001000    ,0      }       /* Volume Control Func Enable */
     ,{      0x069   ,0b00011000    ,200/5  }       /* Volume Control Func Enable */
     ,{      0x013   ,0b00000000    ,10/5   }       /* Record/Playback Run */
```

```
     /*-------------------------------------------------------------
      Record Analog Block OFF
     -------------------------------------------------------------*/
     ,{      0x023  ,0b00001000    ,0      }       /* Input Power Management */
     ,{      0x023  ,0b00000000    ,0      }       /* Input Power Management */
     ,{      0x021  ,0b00000001    ,0      }       /* Reference Power Management */
     ,{      0x021  ,0b00000000    ,0      }       /* Reference Power Management */
};

static const unsigned char aPlayOnTbl[][3]=
{
/*-----------------------------------------------------------------------
     System on → Playback start
-----------------------------------------------------------------------*/
     /*-------------------------------------------------------------
      Playback Function Setting
     -------------------------------------------------------------*/
     /* Volume settings */
      {      0x03B  ,0b00000000    ,0      }       /* Speaker AMP Volume */
     ,{      0x0C9  ,0b00000000    ,0      }       /* Boost Volume */
     ,{      0x049  ,0b00000000    ,0      }       /* AMP Volume Control Function Enable */
     ,{      0x04B  ,0b00000000    ,0      }       /* Amplifier Volume Fader Control */
     ,{      0x069  ,0b00000000    ,0      }       /* Volume Control */
     ,{      0x06B  ,0b00000010    ,0      }       /* Mixer & Volume */
     ,{      0x071  ,0b11111111    ,0      }       /* Playback Digital Volume Control */

     /* Programmable Equalizer settings */
//   ,{      0x067  ,0b00000000    ,0      }       /* Filter Func Enable */
//   ,{      0x075  ,0b00000000    ,0      }       /* EQ gain1 Band0 */
//   ,{      0x077  ,0b00000000    ,0      }       /* EQ gain1 Band1 */
//   ,{      0x079  ,0b00000000    ,0      }       /* EQ gain1 Band2 */
//   ,{      0x07b  ,0b00000000    ,0      }       /* EQ gain1 Band3 */
//   ,{      0x07d  ,0b00000000    ,0      }       /* EQ gain1 Band4 */
//   ,{      0x083  ,0b00000000    ,0      }       /* EQ Band0 Coef0H */
//   ,{      0x085  ,0b00000000    ,0      }       /* EQ Band0 Coef1L */
//   ,{      0x087  ,0b00000000    ,0      }       /* EQ Band0 Coef1H */
//   ,{      0x089  ,0b00000000    ,0      }       /* EQ Band1 Coef0L */
//   ,{      0x08B  ,0b00000000    ,0      }       /* EQ Band1 Coef0H */
//   ,{      0x08D  ,0b00000000    ,0      }       /* EQ Band1 Coef1L */
//   ,{      0x08F  ,0b00000000    ,0      }       /* EQ Band1 Coef1H */
//   ,{      0x091  ,0b00000000    ,0      }       /* EQ Band2 Coef0L */
//   ,{      0x093  ,0b00000000    ,0      }       /* EQ Band2 Coef0H */
//   ,{      0x095  ,0b00000000    ,0      }       /* EQ Band2 Coef1L */
//   ,{      0x097  ,0b00000000    ,0      }       /* EQ Band2 Coef1H */
//   ,{      0x099  ,0b00000000    ,0      }       /* EQ Band3 Coef0L */
//   ,{      0x09B  ,0b00000000    ,0      }       /* EQ Band3 Coef0H */
//   ,{      0x09D  ,0b00000000    ,0      }       /* EQ Band3 Coef1L */
//   ,{      0x09F  ,0b00000000    ,0      }       /* EQ Band3 Coef1H */
//   ,{      0x0A1  ,0b00000000    ,0      }       /* EQ Band4 Coef0L */
//   ,{      0x0A3  ,0b00000000    ,0      }       /* EQ Band4 Coef0H */
//   ,{      0x0A5  ,0b00000000    ,0      }       /* EQ Band4 Coef1L */
//   ,{      0x0A7  ,0b00000000    ,0      }       /* EQ Band4 Coef1H */

     /* Playback Limiter settings */
     ,{      0x069  ,0b00011001    ,0      }       /* Volume Control Func Enable */
//   ,{      0x0C1  ,0b00000000    ,0      }       /* Attack Time */
//   ,{      0x0C3  ,0b00000000    ,0      }       /* Decay Time */
//   ,{      0x0C5  ,0b00000000    ,0      }       /* Target Level */
//   ,{      0x0C7  ,0b00000000    ,0      }       /* Max/Min Gain */

     /*-------------------------------------------------------------
      Playback Analog Block ON
     -------------------------------------------------------------*/
     /* Turning on the VMID generator */
     ,{      0x021  ,0b00000001    ,5/5    }       /* Reference Power Management */
     ,{      0x021  ,0b00000010    ,5/5    }       /* Reference Power Management */
```

```
     /* Mute settings */
     ,{      0x049   ,0b00000000    ,0      }       /* AMP Volume Control Function Enable */
     ,{      0x049   ,0b00000010    ,0      }       /* AMP Volume Control Function Enable */

     /* DAC settings */
     ,{      0x025   ,0b00000010    ,0      }       /* DAC Power */
     ,{      0x055   ,0b00000010    ,0      }       /* Speaker AMP Output Control */

     /* Turning on the speaker amplifier */
     ,{      0x027   ,0x013         ,0      }       /* Power Management */
     ,{      0x027   ,0x01F         ,0      }       /* Reference Power Management */
     ,{      0x03B   ,0x033         ,0      }       /* Speaker AMP Volume */
     ,{      0x04B   ,0b00000000    ,0      }       /* Amplifier Volume Fader Control */
     ,{      0x049   ,0b00000011    ,0      }       /* AMP Volume Control Function Enable */
     ,{      0x049   ,0b00000001    ,500/5  }       /* AMP Volume Control Function Enable */

     /*-------------------------------------------------------------
      Playback start
     -------------------------------------------------------------*/
     ,{      0x069   ,0b00011101    ,0      }       /* Volume Control Func Enable */
     ,{      0x069   ,0b00010001    ,0      }       /* Volume Control Func Enable */
     ,{      0x013   ,0b00000010    ,5/5    }       /* Record/Playback Run */
     ,{      0x06B   ,0b00110010    ,0      }       /* Mixer & Volume Control */
     ,{      0x069   ,0b00011001    ,0      }       /* Volume Control Func Enable */
     ,{      0x069   ,0b00001001    ,0      }       /* Volume Control Func Enable */
};

static const unsigned char aPlayOffTbl[][3]=
{
/*-----------------------------------------------------------------------
     Playback → System on
-----------------------------------------------------------------------*/
     /*-------------------------------------------------------------
      Playback Stop
     -------------------------------------------------------------*/
      {      0x06B   ,0b00000010    ,0      }       /* Mixer & Volume Control */
     ,{      0x069   ,0b00001000    ,0      }       /* Volume Control Func Enable */
     ,{      0x069   ,0b00011000    ,100/5  }       /* Volume Control Func Enable */
     ,{      0x013   ,0b00000000    ,0      }       /* Record/Playback Run */

     /*-------------------------------------------------------------
      Playback Analog Block OFF
     -------------------------------------------------------------*/
     /*  Turning off the speaker amplifier BTL mode */
     ,{      0x04B   ,0b00000000    ,0      }       /* Amplifier Volume Fader Control */
     ,{      0x049   ,0b00000001    ,0      }       /* AMP Volume Control Function Enable */
     ,{      0x049   ,0b00000011    ,100/5  }       /* AMP Volume Control Function Enable */
     ,{      0x027   ,0x013         ,0      }       /* AMP Power Management */
     ,{      0x027   ,0x000         ,0      }       /* AMP Power Management */
     ,{      0x025   ,0b00000000    ,0      }       /* DAC Power Management */
     ,{      0x021   ,0b00000000    ,0      }       /* Reference Power Management */
};


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


     Initial settings of the peripheral hardware to be used


*****************************************************************************/
void hdwinit(void)
{

/*---------------------------------------------------------------------------
     Disabling interrupts
```

```
-----------------------------------------------------------------------------*/
     DI();
/*----------------------------------------------------------------------------
     Clock frequency settings
-----------------------------------------------------------------------------
     Setting so that operations can be performed using the 20 MHz X1 oscillator
-----------------------------------------------------------------------------*/
     CMC = 0b01000001;                    /* Clock operation mode */
     /*      |||||||+------------------------ AMPH: 10 MHz<fMX≤20 MHz */
     /*      ||||+++------------------------- <000> */
     /*      |||+---------------------------- OSCSELS: P123 and P124 pins set as input ports */
     /*      ||+----------------------------- <0> */
     /*      ++------------------------------ EXCLK/OSCSEL: X1 oscillation mode (20 MHz) */

     CSC = 0b01000000;                    /* Clock operation status control */
     /*      |||||||+------------------------ HIOSTOP: Internal high-speed oscillator operated */
     /*      ||+++++------------------------- <00000> */
     /*      |+------------------------------ XTSTOP: XT1 oscillator stopped */
     /*      +------------------------------- MSTOP: X1 oscillator operated */

     OSMC = 0b00000001;                   /* Operation speed mode */
     /*       |||||||+----------------------- FSEL: Operated at a frequency exceeding 10 MHz */
     /*       +++++++------------------------ <00000> */

     OSTS = 0b00000101;                   /* Oscillation stabilization time: 2^15/fX */

     while(!OSTC.2) {}                    /* Waiting for clock oscillation to stabilize */

     CKC = 0b00011000;                    /* Clock selection */
     /*      |||||+++------------------------ MDIV2 to MDIV0: CPU/peripheral hardware clock
     /*      |||||                             (fCLK) = fMX */
     /*      ||||+--------------------------- <1> */
     /*      |||+---------------------------- MCM0: High-speed system clock (fMX) */
     /*      ||+----------------------------- <R> */
     /*      |+------------------------------ CSS: CSS: Main system clock (fMAIN) = fCLK */
     /*      +------------------------------- <R> */

/*----------------------------------------------------------------------------
     Port 0 settings
-----------------------------------------------------------------------------*/
     P0  = 0b00011010;                    /* Output latches of P00, P02, P05, and P06 set to
                                             low level and those of P01, P03, and P04 set to
                                             high level */
     PM0 = 0b10000000;                    /* P00 to P06 set as output ports */
                                          /* P01: Output LRCLK to the audio codec */
                                          /* P02: Output SO10 to the EEPROM */
                                          /* P03: Input SI10 from the EEPROM */
                                          /* P04: Output SCK10 to the EEPROM */
                                          /* P00, P05, and P06: Unused */

/*----------------------------------------------------------------------------
     Port 1 settings
-----------------------------------------------------------------------------*/
     P1  = 0b00000000;                    /* Output latches of P10 to P17 set to low level */
     PM1 = 0b00000000;                    /* P10 to P17 set as output ports */
                                          /* P10: Output SCK00 to the audio codec */
                                          /* P11: Input SI00 from the audio codec */
                                          /* P12: Output SO00 to the audio codec */
                                          /* P13 to P17: Unused */

/*----------------------------------------------------------------------------
     Port 2 settings
-----------------------------------------------------------------------------*/
     P2  = 0b00000000;                    /* Output latches of P20 to P27 set to low level */
     PM2 = 0b00000000;                    /* P20 to P27 set as output ports */
                                          /* P20 to P27: Unused */
```

```
/*-----------------------------------------------------------------------------
     Port 3 settings
-----------------------------------------------------------------------------*/
     P3  = 0b00000000;                    /* Output latches of P30 and P31 set to low level */
     PM3 = 0b11111100;                    /* P30 and P31 set as output ports */
                                          /* P30 and P31: Unused */

/*-----------------------------------------------------------------------------
     Port 4 settings
-----------------------------------------------------------------------------*/
     P4  = 0b00000000;                    /* Output latches of P40 to P47 set to low level */
     PM4 = 0b00000000;                    /* P40 to P47 set as output ports */
                                          /* P40 to P47: Unused */

/*-----------------------------------------------------------------------------
     Port 5 settings
-----------------------------------------------------------------------------*/
     P5  = 0b00000000;                    /* Output latches of P50 to P57 set to low level */
     PM5 = 0b00000000;                    /* P50 to P57 set as output ports */
                                          /* P50 to P57: Unused */

/*-----------------------------------------------------------------------------
     Port 6 settings
-----------------------------------------------------------------------------*/
     P6  = 0b00000000;                    /* Output latches of P60 to P67 set to low level */
     PM6 = 0b00000000;                    /* P60 to P67 set as output ports */
                                          /* P60: Output SCL0 for setting the registers in the
                                                  audio codec */
                                          /* P61: Input and output SDA0 for setting the
                                                  registers in the audio codec */
                                          /* P62 to P67: Unused */

/*-----------------------------------------------------------------------------
     Port 7 settings
-----------------------------------------------------------------------------*/
     P7  = 0b00000000;                    /* Output latches of P70 to P77 set to low level */
     PM7 = 0b00000011;                    /* P70 and P71 set as input ports and P72 to P77 set
                                             as output ports */
                                          /* P70: Playback key input */
                                          /* P71: Record key input */
                                          /* P72: Output a signal to the operation-in-progress
                                                 indicator (LED) */
                                          /* P73: Output a reset signal to the audio codec */
                                          /* P74: Output CS0 to the EEPROM */
                                          /* P75: Output CS1 to the EEPROM */
                                          /* P76: Output CS2 to the EEPROM */
                                          /* P77: Output CS3 to the EEPROM */

/*-----------------------------------------------------------------------------
     Port 8 settings
-----------------------------------------------------------------------------*/
     P8  = 0b00000000;                    /* Output latches of P80 to P87 set to low level */
     PM8 = 0b00000000;                    /* P80 to P87 set as output ports */
                                          /* P80 to P87: Unused */

/*-----------------------------------------------------------------------------
     Port 11 settings
-----------------------------------------------------------------------------*/
     P11  = 0b00000000;                   /* Output latches of P110 and P111 set to low level */
     PM11 = 0b11111100;                   /* P110 and P111 set as output ports */
                                          /* P110 and P111: Unused */

/*-----------------------------------------------------------------------------
     Port 12 settings
-----------------------------------------------------------------------------*/
     P12  = 0b00000000;                   /* Output latch of P120 set to low level */
     PM12 = 0b11111110;                   /* P120 set as an output port */
```

```
                                        /* P120: Unused */

/*-----------------------------------------------------------------------------
     Port 13 settings
-----------------------------------------------------------------------------*/
     P13  = 0b00000000;                 /* Output latches of P130 and P131 set to low level */
     PM13 = 0b11111100;                 /* P131 set as an output port */
                                        /* P130 and P131: Unused */

/*-----------------------------------------------------------------------------
     Port 14 settings
-----------------------------------------------------------------------------*/
     P14  = 0b00000000;                 /* Output latches of P140 to P145 set to low level */
     PM14 = 0b11000000;                 /* P140 to P145 set as output ports */
                                        /* P140 to P145: Unused */

/*-----------------------------------------------------------------------------
     Port 15 settings
-----------------------------------------------------------------------------*/
     P15  = 0b00000000;                 /* Output latches of P150 to P157 set to low level */
     PM15 = 0b00000000;                 /* P150 to P157 set as output ports */
                                        /* P150 to P157: Unused */

/*-----------------------------------------------------------------------------
     Settings of the registers in the audio codec
-----------------------------------------------------------------------------

     The following operations are performed.
      •Supplying a clock to the audio codec
      •Setting a timer for inserting waits in the program
      •Outputting a reset signal to the audio codec
      •Using the I2C interface for setting the registers in the audio codec

-----------------------------------------------------------------------------*/

     /* Timer array unit timer clock selection */
     TAU0EN = 1;                        /* Input clock supplied to the timer array unit */
     TPS0L = 0b10010000;                /* Operation clock selection */
     /*        ||||++++---------------------- CK00: fCLK */
     /*        ++++------------------------- CK01: fCLK/2^9 */

     /*---------------------------------------
      Clock supply to the audio codec
     ---------------------------------------
      TO02 output (10 MHz)
     ---------------------------------------*/
     TMR02 = 0b0000000000000000;        /* Operation mode setting */
     /*        ||||||||||||++++------------- MD023 to MD020: Interval timer mode */
     /*        ||||||||||++----------------- <Fixed to 00> */
     /*        ||||||||++------------------- CIS021 and CIS020: unused */
     /*        |||||+++---------------------- STS022-020: Only software trigger start enabled */
     /*        ||||+------------------------- MASTER02: Standalone operation */
     /*        |||+-------------------------- CSS02: Macro clock MCK specified by using
     /*        |||                                   the CKS02 bit */
     /*        |++--------------------------- <Fixed to 00> */
     /*        +---------------------------- CKS02: Operation clock CK00 set by using
     /*                                             the PRS register */

     TDR02 = 0;                         /* Interval setting: 10 MHz output (20 MHz) */
     TOE0L.2 = 1;                       /* TO02 output */
     TS0L.2 = 1;                        /* Operation starts (trigger bit) */

     /*---------------------------------------
      Waits used in the program
     ---------------------------------------
      Using TM01
     ---------------------------------------*/
```

```
TMR01 = 0b1000000000000000;          /* Operation mode setting */
/*        |||||||||||||++++------------ MD013 to MD010: Interval timer mode */
/*        ||||||||||||++----------------- <Fixed to 00> */
/*        ||||||||++------------------- CIS011 and CIS010: Unused */
/*        |||||+++--------------------- STS012 to STS010: Only software trigger start
/*        |||||                                            enabled */
/*        ||||+------------------------ MASTER01: Standalone operation */
/*        |||+------------------------- CSS01: Macro clock MCK specified by using
/*        |||                                   the CKS01 bit */
/*        |++-------------------------- <Fixed to 00> */
/*        +----------------------------- CKS01: Operation clock CK01 set by using
/*                                              the PRS register*/

/* Waiting for power supply to stabilize */
TDR01 = 195*210/5-1;                 /* Interval set (210 ms) */
TS0L.1 = 1;                          /* Timer operation starts */
TMIF01 = 0;                          /* INTTM00 interrupt request cleared */
while(!TMIF01) {}

/* Reset */
P_RESETB = 0;
/* Waiting about 5 us */
TDR01 = 0;                           /* Interval set */
TS0L.1 = 1;                          /* Timer operation starts */
TMIF01 = 0;                          /* INTTM00 interrupt request cleared */
while(!TMIF01) {}
P_RESETB = 1;

/* Waiting for power supply to stabilize */
TDR01 = 195*160/5-1;                 /* Interval set (160 ms) */
TS0L.1 = 1;                          /* Timer operation starts */
TMIF01 = 0;                          /* INTTM00 interrupt request cleared */
while(TMIF01)  {}
TT0L.1 = 1;                          /* Timer operation stopped */

/*---------------------------------------
 Transmission of the values set to
 the registers in the audio codec
----------------------------------------
 Using IIC0
----------------------------------------*/
IIC0EN = 1;                          /* Input clock of serial interface IIC0 supplied */

IICE0 = 0;                           /* Operation stopped */

P_SCL = 0;                           /* SCL0 latch */
PM_SCL = 0;                          /* SCL0 pin */
P_SDA = 0;                           /* SDA0 pin */
PM_SDA = 0;                          /* SDA0 pin */

IICX0 = 0b00000000;                  /* Transfer clock selection */
/*        |||||||+--------------------- CLX0 */
/*        +++++++---------------------- <0000000> */
IICCL0 = 0b00001110;
/*         ||||||++-------------------- CL01-CL00(+CLX0): fCLK/96(first mode) */
/*         |||||+----------------------- DFC0: Turn on the digital filter */
/*         ||||+------------------------ SMC0: Specify operation in the first mode */
/*         |||+------------------------- DAD0: <R>Detect the SDA0 pin level */
/*         ||+-------------------------- CLD0: <R>Detect the SCL0 pin level */
/*         ++--------------------------- <00> */
```

```
     IICF0 = 0b00000011;                  /* Communication reservation disable */
     /*        |||||||+--------------------- IICRSV: Disable communication reservation */
     /*        ||||||+---------------------- STCEN: Enable initial start */
     /*        ||++++------------------------ <Fixed to 0000> */
     /*        |+--------------------------- IICBSY: <R>IIC bus status flag */
     /*        +---------------------------- STCF: <R>STT0 clear flag */


     IICC0 = 0b00001000;                  /* Initial settings during master operation */
     /*        |||||||+--------------------- SPT0: Stop condition trigger */
     /*        ||||||+---------------------- STT0: Start condition trigger */
     /*        |||||+------------------------ ACKE0: Control acknowledgment */
     /*        ||||+------------------------- WTIM0: Control wait insertion and interrupt
     /*        ||||                                  request issuance: 9 clocks */
     /*        |||+-------------------------- SPIE0: Disable issuing of interrupt requests by
     /*        |||                                   detecting a stop condition */
     /*        ||+--------------------------- WREL0: Do not cancel waiting */
     /*        |+---------------------------- LREL0: Save the communication: Normal operation */
     /*        +---------------------------- IICE0: Enable operation of I2C */

     SPT0 = 1;                            /* Stop condition set */

     /* Settings of the registers in the audio codec: Setting to turn on the system */
     fn_I2cWrite(&aSystemOnTbl[0][0],sizeof(aSystemOnTbl));

/*-------------------------------------------------------------------------------
     Settings of the I2S bus interface used for the audio data of the
     audio codec
-------------------------------------------------------------------------------

     The following operations are performed.
      •Setting the TO00 output and INTTM00 interrupt for outputting LRCLK
      •Setting CSI00 for transmitting and receiving data

-------------------------------------------------------------------------------*/
     /*---------------------------------------
      LRCLK output settings
     ---------------------------------------
      TO00 output (16 kHz)
     ---------------------------------------*/
     TMR00 = 0b0000000000000000;          /* Operation mode setting */
     /*        ||||||||||||++++------------- MD003 to MD000: Interval timer mode */
     /*        ||||||||||++----------------- <Fixed to 00> */
     /*        ||||||||++------------------- CIS001 and CIS000: Unused */
     /*        |||||+++---------------------- STS002 to STS000: Only software trigger start
     /*        |||||                                            enabled */
     /*        ||||+------------------------ MASTER00: Standalone operation */
     /*        |||+------------------------- CSS00: Macro clock MCK specified by using
     /*        |||                                   the CKS00 bit */
     /*        |++--------------------------- <Fixed to 00> */
     /*        +---------------------------- CKS00: Operating clock CK00 set by using
     /*                                              the PRS register */
     TDR00 = 625-1;                       /* Interval set: 16 kHz output (32 kHz) */

     TMPR100 = 0;                         /* Priority order set to the highest level */
     TMPR000 = 0;

     /*---------------------------------------
      CSI00 settings
     ---------------------------------------*/
     SAU0EN = 1;                          /* Input clock of the serial array unit supplied */

     NOP();                               /* Waiting */
     NOP();
     NOP();
     NOP();
```

```
     SPS0L = 0b00000000;                  /* Operation clock selection: fCLK */
     /*        ||||++++--------------------- PRS003 to PRS000: fCLK */
     /*        ++++------------------------- PRS013 to PRS010: Unused */

     SDR00 = (12-1) << 9;                 /* Bits 15 to 7: Transfer clock set (833 kHz) */


     SMR00 = 0b0000000000100001;          /* Operation mode selected: CSI mode */
     /*        ||||||||||||||+------------- MD000: Buffer empty interrupt */
     /*        ||||||||||||++-------------- MD002 and MD001: CSI mode */
     /*        |||||||||+++----------------- <Fixed to 100> */
     /*        |||||||||+------------------- SIS000: Unused */
     /*        ||||||||+-------------------- <Fixed to 0> */
     /*        |||||||+--------------------- STS00: Only software trigger enabled
     /*        |||||||                             (fixed in CSI mode) */
     /*        ||+++++----------------------- <Fixed to 00000> */
     /*        |+-------------------------- CSS00: Transfer clock set to a clock obtained by
     /*        |                                  dividing operation clock MCK as specified y
     /*        |                                  by using the CKS00 bit */
     /*        +---------------------------- CKS00: Operation clock set to prescaler output
     /*                                           clock CK00 set by using the PRS register */

     /* Initial data output (at the same time as setting CSI10) */
     /* SO0 = 0b0000000100000001;         /* Setting of initial outputs of the SO and SCK pins */
     /*         ||||||||++++++++------------ SO0n to SO00: Used for transmission */
     /*         ++++++++-------------------- CKOn to CKO0: Serial clock output of channels n to 0 */

     SOE0L.0 = 1;                         /* Output enabled */

     P_SCK00 = 1;                         /* SCK00 latch: High level */
     PM_SCK00 = 0;                        /* SCK00 pin output set */
     PM_SI00 = 1;                         /* SI00 pin input set */
     P_SO00 = 1;                          /* SO00 latch: High level */
     PM_SO00 = 0;                         /* SO00 pin output set */

     SS0L.0 = 1;                          /* CSI00 operation starts (trigger bit) */
     CSIIF00 = 0;                         /* INTCSI00 interrupt request cleared */
     CSIMK00 = 0;                         /* INTCSI00 interrupt servicing enabled */

     CSIPR100 = 0;                        /* Priority order set to next after INTTM00 */
     CSIPR000 = 1;

/*-----------------------------------------------------------------------------
     Setting of the SPI interface for EEPROM
-----------------------------------------------------------------------------

     The following operations are performed.
      •Setting CSI10 for transmitting and receiving data
      •Setting DMA0 for successive reception and DMA1 for successive transmission

-----------------------------------------------------------------------------*/
     P_CS |= 0b011110000;                 /* All EEPROMs deactivated */

     /*-------------------------------------
      DMA0 settings (for CSI reception)
     -------------------------------------*/
     DEN0 = 1;                            /* Operation of DMA channel 0 enabled */

     DSA0 = 0x044;                        /* DMA SFR address: SDR02(SIO10)=0FFF44H */
```

```
DMC0 = 0b00001000;                    /* Setting of transfer mode of DMA channel 0 */
/*       ||||++++---------------------- IFC03 to IFC00: DMA start source: INTCSI10
/*       ||||                                            transfer complete interrupt */
/*       |||+-------------------------- DMA transfer suspension: DMA transfer performed
/*       |||                                                     according to a DMA start
/*       |||                                                     request (not suspended) */
/*       ||+--------------------------- DS0: Transfer data size: 8 bits */
/*       |+---------------------------- DRS0: DMA transfer direction selected: SFR →
/*       |                                    Internal RAM */
/*       +----------------------------- STG0: Software trigger not operated */

/*-------------------------------------------------
 DMA1 settings
 (for CSI transmission and setting dummy data
  during CSI reception)
-------------------------------------------------*/
DEN1 = 1;                             /* Operation of DMA channel 1 enabled */

DSA1 = 0x044;                         /* DMA SFR address: SDR02(SIO10)=0FFF44H */

DMC1 = 0b01001000;                    /* Setting of transfer mode of DMA channel 1 */
/*       ||||++++---------------------- IFC13 to IFC10: DMA start source: INTCSI10
/*       ||||                                            transfer complete interrupt */
/*       |||+-------------------------- DMA transfer suspension: DMA transfer performed
/*       |||                                                     according to a DMA start
/*       |||                                                     request(not suspended) */
/*       ||+--------------------------- DS1: Transfer data size: 8 bits */
/*       |+---------------------------- DRS1: DMA transfer direction selected: Internal
/*       |                                    RAM→ SFR */
/*       +----------------------------- STG1: Software trigger not operated */

/*--------------------------------------
 CSI10 settings
--------------------------------------*/
SDR02 = (4-1) << 9;                   /* Bits 15 to 7: Transfer clock set (2.5 MHz) */

SMR02 = 0b0000000000100000;           /* Operation mode selected: CSI mode */
/*        ||||||||||||||+------------- MD020: Transfer complete interrupt */
/*        ||||||||||||||++-------------- MD022 and MD021: CSI mode */
/*        ||||||||||+++---------------- <Fixed to 100> */
/*        |||||||||+-------------------- SIS020: Unused */
/*        ||||||||+--------------------- <Fixed to 0> */
/*        |||||||+---------------------- STS02: Only software trigger enabled
/*        |||||||                              (fixed in CSI mode) */
/*        ||+++++----------------------- <Fixed to 00000> */
/*        |+--------------------------- CSS02: Transfer clock set to a clock obtained by
/*        |                                     dividing operation clock MCK as specified
/*        |                                     by using the CKS00 bit */
/*        +---------------------------- CKS02: Operation clock set to prescaler output
/*                                              clock CK00 set by using the PRS register */
```

```
     SCR02 = 0b1011000000000111;          /* Communication format setting */
     /*        |||||||||||||+++------------ DLS022 to DLS020: 8-bit data length */
     /*        |||||||||||||+---------------- <Fixed to 0> */
     /*        ||||||||||++----------------- SLC021 and SLC020: Unused (fixed to 0) */
     /*        |||||||||+------------------- <Fixed to 0> */
     /*        ||||||||+-------------------- DIR02: Input and output performed MSB first */
     /*        ||||||++--------------------- PTC021 and PTC020: Unused (fixed to 00) */
     /*        |||||+------------------------ EOC02: Unused (fixed to 0) */
     /*        ||||+------------------------- <Fixed to 0> */
     /*        ||++-------------------------- CKP02/DAP02: Phases of data and clock in CSI mode
     /*        ||                                            selected */
     /*        ||                            [11 selected] */
     /*        ||                                      _   _   _   _   _   _   _   _   */
     /*        ||                            SCK02 ____| |_| |_| |_| |_| |_| |_| |_| |  */
     /*        ||                                  __ ___ ___ ___ ___ ___ ___ ___ ____  */
     /*        ||                            SO02  __|___|___|___|___|___|___|___|____  */
     /*        ||                                      D7  D6  D5  D4  D3  D2  D1  D0   */
     /*        ||                  SI02 input timing____|___|___|___|___|___|___|___|__  */
     /*        ||                                                                       */
     /*        ++---------------------------- TXE00/RXE00: Only transmission performed  */

     /* Initial data output */
     SO0 = 0b0000000100000001;            /* Setting of initial outputs of the SO and SCK pins */
     /*      ||||||||++++++++--------------- SO0n to SO00: Used for transmission */
     /*      ++++++++------------------------ CKOn to CKO0: Serial clock output of channels n
     /*                                                     to 0 */
     SOE0L.2 = 1;                         /* Output enabled */

     P_SCK10 = 1;                         /* SCK00 latch: High level */
     PM_SCK10 = 0;                        /* SCK00 pin output set */
     PM_SI10 = 1;                         /* SI00 pin input set */
     P_SO10 = 1;                          /* SO00 latch: High level */
     PM_SO10 = 0;                         /* SO00 pin output set */

     SS0L.2 = 1;                          /* CSI10 operation starts (trigger bit) */

     /*--------------------------------------
      Erasing all EEPROMs (all 0FFH)
     --------------------------------------*/
     /* Enabling writing to an EEPROM */
     P_CS &= 0b00001111;                  /* All EEPROMs selected */
     CSIIF10 = 0;
     SIO10 = 0b00000110;                  /* Transmit data set (Write enable (WREN)
                                             instruction INTCSI10 occurs) */
     while(!CSIIF10)       {}
     P_CS |= 0b11110000;                  /* Setting completed */

     /* Erasing all EEPROMs */
     P_CS &= 0b00001111;                  /* All EEPROMs selected */
     CSIIF10 = 0;
     SIO10 = 0b11000111;                  /* Transmit data set (All CEs erased INTCSI10
                                             occurs) */
     while(!CSIIF10)       {}
     P_CS |= 0b11110000;                  /* Setting completed */

     ST0L.2 = 1;                          /* CSI10 operation stopped (trigger bit) */
     CSIMK10 = 1;                         /* INTCSI10 interrupt servicing disabled */

/*-----------------------------------------------------------------------------
     Starting key retrieval
-----------------------------------------------------------------------------*/
     KRM = 0b00000011;                    /* KR0 and KR1 enabled */
     KRIF = 0;                            /* INTKR interrupt request cleared */
     KRMK = 1;                            /* INTKR interrupt servicing disabled */

     P_LED = 1;                           /* Operation-in-progress indicator turned off (key
                                             input possible) */
```

```
/*-----------------------------------------------------------------------------
     Enabling interrupts
-----------------------------------------------------------------------------*/
     EI();
}

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


     Main processing


*****************************************************************************/
void main(void)
{
     ucPlayMode = CSTOP;           /* Stopped */

     while(1){
             /*-------------------------------------
              Key processing
             -------------------------------------*/
             if(KRIF){
             /* Key input */
                KRIF = 0;   /* INTKR interrupt request cleared */
                if(ucPlayMode==CSTOP){
                /* Stopped */
                 if(P_KEY & 0b00000011){
                 /* key valid */
                     /* Key judged to have been pressed */
                     /* Waiting about 10 ms (chattering removed) */
                     TDR01 = 195*10/5-1;    /* Interval set (10 ms) */
                     TS0L.1 = 1;            /* Timer operation starts */
                     TMIF01 = 0;            /* INTTM00 interrupt request cleared */
                     while(!TMIF01) {}

                     if((P_KEY & 0b00000011)&&!P_RECKY){   /* KR port */
                     /* Record key valid */
                     /*-------------------------------------
                      Starting recording
                     -------------------------------------*/
                        P_LED = 0;  /* Operation-in-progress indicator turned on (key input
                                       disabled) */

                        KRM = 0b00000000;          /* KR0 and KR1 disabled */
                        ucPlayMode = CREC;         /* Voice data recorded */

                        /* Settings of the registers in the audio codec: Recording set */
                        fn_I2cWrite(&aRecordOnTbl[0][0],sizeof(aRecordOnTbl));

                        /* Preparation of the I2S bus interface */
                        ucI2sByteCouter = 0;               /* CSI00 reception counter */
                        SDR00 = (12-1) << 9;               /* Bits 15 to 7: Transfer clock set
                                                              (833 kHz) */
```

```
SCR00 = 0b0100000000000111;        /* Communication format setting */
/*        |||||||||||||+++---------- DLS002 to DLS000: 8-bit data length */
/*        ||||||||||||+------------- <Fixed to 0> */
/*        ||||||||||++-------------- SLC001 and SLC000: Unused (fixed
/*        ||||||||||                                    to 0) */
/*        |||||||||+---------------- <Fixed to 0> */
/*        ||||||||+----------------- DIR00: Input and output
/*        ||||||||                          performed MSB first */
/*        ||||||++------------------ PTC001 and PTC000: Unused (fixed
/*        ||||||                                        to 00) */
/*        |||||+-------------------- EOC00: Unused (fixed to 0) */
/*        ||||+--------------------- <Fixed to 0> */
/*        ||++----------------------- CKP00/DAP00: Phases of data and
/*        ||                                     clock in CSI mode
/*        ||                                     selected */
/*        ||              [00 selected] */
/*        ||                            __   _   _   _   _   _   _   _   _ */
/*        ||              SCK00       |_| |_| |_| |_| |_| |_| |_| |_| |_| */
/*        ||                            __ ___ ___ ___ ___ ___ ___ ___ ____*/
/*        ||              SO00        __|___|___|___|___|___|___|___|____*/
/*        ||                              D7  D6  D5  D4  D3  D2  D1  D0 */
/*        ||      SI00 input timing____|___|___|___|___|___|___|___|__*/
/*        ||                                                            */
/*        ++------------------------- TXE00/RXE00: Only reception
/*                                               performed */

TO0L = 0b00000000;       /* Initial output set to low level */
TOE0L.0 = 1;             /* Operation of TO00 enabled by a count operation
                            (LRCLK) */
TMIF00 = 0;              /* INTTM00 interrupt request cleared */
TMMK00 = 0;              /* INTTM00 interrupt servicing enabled */

/* Setting writing to an EEPROM */
ucI2sAddress = &ucMemoryPage1[0];       /* Address to which the data received
                                           via I2S is saved */
bI2sMemoryPage = 0;
ucEepromCs = 0b11100000;         /* EEPROM selected */
ucEepromSeq = CEEPSEQ_RESET;     /* EEPROM transfer sequence */
ushEepromAddress = 0;            /* EEPROM write address
                                   (ushEepromAddress*100H) */
SCR02 = 0b1011000000000111;      /* Communication format setting */
/*        |||||||||||||+++---------- DLS022 to DLS020: 8-bit data length */
/*        ||||||||||||+------------- <Fixed to 0> */
/*        ||||||||||++-------------- SLC021 and SLC020: Unused (fixed to
/*        ||||||||||                                    0) */
/*        |||||||||+---------------- <Fixed to 0> */
/*        ||||||||+----------------- DIR02: Input and output performed
/*        ||||||||                          MSB first */
/*        ||||||++------------------ PTC021 and PTC020: Unused (fixed to
/*        ||||||                                        00) */
/*        |||||+-------------------- EOC02: Unused (fixed to 0) */
/*        ||||+--------------------- <Fixed to 0> */
/*        ||++----------------------- CKP02/DAP02: Phases of data and clock in
/*        ||                                     CSI mode selected */
/*        ||              [11 selected]                                */
/*        ||                            _   _   _   _   _   _   _   _ */
/*        ||              SCK02     ____| |_| |_| |_| |_| |_| |_| |_| |*/
/*        ||                          __ ___ ___ ___ ___ ___ ___ ___ ____*/
/*        ||              SO02      __|___|___|___|___|___|___|___|____*/
/*        ||                            D7  D6  D5  D4  D3  D2  D1  D0 */
/*        ||     SI02 input timing ____|___|___|___|___|___|___|___|__*/
/*        ||                                                            */
/*        ++------------------------- TXE00/RXE00: Only transmission
/*                                               performed */
```

```
   /* Starting I2S operation */
   SS0L.0 = 1;         /* CSI00 operation starts (trigger bit) */
   P_LRCLK = 0;        /* LRCLK operation starts */
   TS0L.0 = 1;         /* TO00 output operation starts (trigger bit) */
   SIO00 = 0x0FF;      /* Dummy data set (INTCSI00 occurs) */

}
else if((P_KEY & 0b00000011)&&!P_PLAYKY){    /* KR port */
/* Playback key valid */
/*-------------------------------------
 Starting playback
--------------------------------------*/
   P_LED = 0;  /* Operation-in-progress indicator turned on (key input
                  disabled) */

   KRM = 0b00000000;                 /* KR0 and KR1 disabled */
   ucPlayMode = CPLAY_START;         /* Preparing for playback started */

   /* Settings of the registers in the audio codec: Playback set */
   fn_I2cWrite(&aPlayOnTbl[0][0],sizeof(aPlayOnTbl));

   /* Preparation of the I2S bus interface */
   ucI2sByteCouter = 0;              /* CSI00 transmission counter */
   SDR00 = (12-1) << 9;              /* Bits 15 to 7: Transfer clock set
                                        (833 kHz) */
   SCR00 = 0b1000000000000111;       /* Communication format setting */
   /*        |||||||||||||+++---------- DLS002 to DLS000: 8-bit data
   /*        |||||||||||||                                length */
   /*        ||||||||||||+------------- <Fixed to 0> */
   /*        |||||||||||++-------------- SLC001 and SLC000: Unused (fixed to
   /*        ||||||||||                                    0) */
   /*        |||||||||+---------------- <Fixed to 0> */
   /*        ||||||||+----------------- DIR00: Input and output performed
   /*        ||||||||                         MSB first */
   /*        ||||||++------------------ PTC001 and PTC000: Unused (fixed to
   /*        ||||||                                       00) */
   /*        |||||+-------------------- EOC00: Unused (fixed to 0) */
   /*        ||||+--------------------- <Fixed to 0> */
   /*        ||++---------------------- CKP00/DAP00: Phases of data and
   /*        ||                                      clock in CSI mode
   /*        ||                                      selected */
   /*        ||      [00 selected]                                    */
   /*        ||                        __   _   _   _   _   _   _   _   __ */
   /*        ||              SCK00    |_| |_| |_| |_| |_| |_| |_| |_|   */
   /*        ||                        __ ___ ___ ___ ___ ___ ___ ___ ____ */
   /*        ||               SO00  __|___|___|___|___|___|___|___|____ */
   /*        ||                          D7  D6  D5  D4  D3  D2  D1  D0  */
   /*        ||  SI00 input timing  ____|___|___|___|___|___|___|___|__ */
   /*        ||                                                          */
   /*        ++----------------------- TXE00/RXE00: Only transmission
   /*                                                performed */

   TO0L = 0b00000000;  /* Initial output set to low level */
   TOE0L.0 = 1;        /* Operation of TO00 enabled by a count operation */
   TMIF00 = 0;         /* INTTM00 interrupt request cleared */
   TMMK00 = 0;         /* INTTM00 interrupt servicing enabled */

   /* Setting reading from an EEPROM (writing set at first) */
   P_CS = ((P_CS & 0b00001111)|0b11100000);/* EEPROM selected */
   ucEepromCs = 0b10110000;   /* EEPROM selected (enabled after 2 pages are
                    read) */
   bI2sMemoryPage = 1;        /* Data of the EEPROM set to be saved to page
                    1 */
   ucEepromSeq = CEEPSEQ_INST;/* Instruction bytes transmitted */
   ushEepromAddress = 0;      /* EEPROM read address
                                 (ushEepromAddress*100H) */
```

```
                SCR02 = 0b1011000000000111;       /* Communication format setting */
                /*        |||||||||||||+++---------- DLS022 to DLS020: 8-bit data
                /*        |||||||||||||                                length */
                /*        ||||||||||||+------------- <Fixed to 0> */
                /*        ||||||||||++-------------- SLC021 and SLC020: Unused (fixed to
                /*        ||||||||||                                    0) */
                /*        |||||||||+---------------- <Fixed to 0> */
                /*        ||||||||+----------------- DIR02: Input and output performed
                /*        ||||||||                        MSB first */
                /*        ||||||++------------------ PTC021 and PTC020: Unused (fixed to
                /*        ||||||                                        00) */
                /*        |||||+-------------------- EOC02: Unused (fixed to 0) */
                /*        ||||+--------------------- <Fixed to 0> */
                /*        ||++----------------------- CKP02/DAP02: Phases of data and
                /*        ||                                     clock in CSI mode
                /*        ||                                     selected */
                /*        ||      [11 selected]                                  */
                /*        ||                          _   _   _   _   _   _   _   _  */
                /*        ||              SCK02  ____| |_| |_| |_| |_| |_| |_| |_| | */
                /*        ||                     __ ___ ___ ___ ___ ___ ___ ___ ____ */
                /*        ||               SO02  __|___|___|___|___|___|___|___|____ */
                /*        ||                         D7  D6  D5  D4  D3  D2  D1  D0  */
                /*        ||  SI02 input timing  ____|___|___|___|___|___|___|___|__ */
                /*        ||                                                          */
                /*        ++------------------------- TXE00/RXE00: Only transmission
                /*                                               performed */

                /* Starting reading from the EEPROM */
                CSIIF10 = 0;               /* INTCSI10 interrupt request cleared */
                CSIMK10 = 0;               /* INTCSI10 interrupt servicing enabled */
                SS0L.2 = 1;                /* CSI10 operation starts (trigger bit) */
                SIO10 = 0b00000011;        /*Transmit data (reading specified) (INTCSI10
                                             occurs) */
            }
        }
      }
    }

    /*--------------------------------------
     Recording end settings
    --------------------------------------*/
    if(ucPlayMode==CREC_END){
    /* Recording finished */
            /* Settings of the registers in the audio codec: Setting termination of
               recording */
            fn_I2cWrite(&aRecordOffTbl[0][0],sizeof(aRecordOffTbl));

            KRM = 0b00000011;      /* KR0 and KR1 enabled */
            ucPlayMode = CSTOP;    /* Stopped */
            P_LED = 1;             /* Operation-in-progress indicator turned off (key
                                      input possible) */
    }
    /*--------------------------------------
     Playback end settings
    --------------------------------------*/
    else if(ucPlayMode==CPLAY_END){
    /* Playback finished */
            /* Settings of the registers in the audio codec: Setting termination of
               playback */
            fn_I2cWrite(&aPlayOffTbl[0][0],sizeof(aPlayOffTbl));

            KRM =0b00000011;       /* KR0 and KR1 enabled */
            ucPlayMode = CSTOP;    /* Stopped */
            P_LED = 1;             /* Operation-in-progress indicator turned off (key
                                      input possible) */
    }
}
```

```
}

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

     INTTM00 interrupt servicing
     (using INTTM00 for synchronizing LRCLK, BCLK, SDOUT, and SDIN)

-----------------------------------------------------------------------------

     LRCLK output via the I2S bus interface and BCLK, SDOUT, and SDIN used for
     transmitting and receiving data are synchronized by restarting the
     operation of CSI00, which was stopped by INTCSI00 interrupt servicing,
     when LRCLK changes from high to low level and from low to high level .

                    ________                 _______________
     LRCLK                  |_______________|               |_______

                    _________              __             __
     BCLK                   |||||||||||||  |||||||||||||  |||||||

                    _________ _ _     _ _ __ _ _     _ _ __ _ _
     SDOUT/SDIN             |_|_• • • _|_|  |_|_• • • _|_|  |_|_• •


     INTTM00        ________|_______________|_______________|_______

     Register bank 1 used

*****************************************************************************/
__interrupt void fn_inttm00(void)
{
     if(ucPlayMode==CREC){
     /*-------------------------------------
      During recording
     -------------------------------------*/
     /* Starting the operation of the I2S bus interface */
            SS0L.0 = 1;                  /* CSI00 operation starts (trigger bit) */
            *ucI2sAddress = SIO00;       /* Third byte received and dummy data set (data
                                            saved to address) */
            SIO00 = 0x0FF;               /* Dummy data set (INTCSI00 occurs) */
     }
     else{
     /*-------------------------------------
      During playback
     -------------------------------------*/
     /* Starting the operation of the I2S bus interface */
            SS0L.0 = 1;                  /* CSI00 operation starts (trigger bit) */
            SIO00 = *ucI2sAddress;       /* Transmit data (first byte) set (INTCSI00 occurs) */
     }
     ucI2sAddress++;                     /* Address to which the data transmitted or received
                                            via I2S is saved updated */
}
```

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

     INTCSI00 interrupt servicing
     (using INTCSI00 for CSI transmission or reception)

------------------------------------------------------------------------------

     Data is transmitted and received via the I2S bus interface between the
     microcontroller and the audio codec.
     During recording, CSI00 is used to save the received data.  Normally, data is
     successively received by setting dummy data.  To achieve synchronization with
     LRCLK, however, only reception is performed every three bytes and successive
     reception operation is stopped.  The stopped operation is restarted by INTTM00
     interrupt servicing and can be synchronized with LRCLK.  When the received data
     reaches one page, writing to an EEPROM starts.  During playback, CSI00 is used
     to successively transmit data, but successive transmission operation is
     stopped every three bytes to achieve synchronization with LRCLK, similarly as
     during recording.  The stopped operation is restarted by INTTM00 interrupt
     servicing and can be synchronized with LRCLK.  When the transmitted data
     reaches one page, reading data of a different page from the EEPROM starts.


                    ________                              ______
     LRCLK                  |____________________________|

                    _________                             _
     BCLK                    |||||||||||||||||||||||||||| ||||||

                    _________ _      _ _      _ _      _ _ _
     SDOUT/SDIN              |_• • • _|_• • • _|_• • • _| |_• •

     INTCSI00       _________|________|________|__________|_____

                             1        2        3          1

     Register bank 1 used

******************************************************************************/
__interrupt void fn_intcsi00(void)
{
     ucI2sByteCouter++;            /* CSI00 reception counter updated */

     if(ucPlayMode==CREC){
     /*-------------------------------------
      During recording
     -------------------------------------*/
            /* Determining the last position of the page */
            if(bI2sMemoryPage){
            /* Page 2 being transmitted */
                   if(ucI2sAddress>=&ucMemoryPage2[3*2*42]){
                   /* End of page 2 */
                          ucI2sAddress = &ucMemoryPage1[0];     /* Address to which the data
                                                                   received via I2S is
                                                                   saved: page 1 */
                          bI2sMemoryPage = 0;

                          /* Starting writing to the EEPROM */
                          ucEepromSeq = CEEPSEQ_WREN;           /* Write enable (WREN) signal
                                                                   transmitted */
                          P_CS = (P_CS & 0b00001111)|ucEepromCs;/* EEPROM CS set */
                          CSIIF10 = 0;           /* INTCSI10 interrupt request cleared */
                          CSIMK10 = 0;           /* INTCSI10 interrupt servicing enabled */
                          SS0L.2 = 1;            /* CSI10 operation starts (trigger bit) */
                          SIO10 = 0b00000110;    /* Transmit data set (Write enable (WREN)
                                                    instruction (INTCSI10 occurs) */
                   }
            }
```

```
        else{
        /* Page 1 being transmitted */
                if(ucI2sAddress>=&ucMemoryPage1[3*2*42]){
                /* End of page 1 */
                        ucI2sAddress = &ucMemoryPage2[0];     /* Address to which the data
                                                                 received via I2S is
                                                                 saved: Page 2 */
                        bI2sMemoryPage = 1;

                        /* Starting writing to the EEPROM */
                        ucEepromSeq = CEEPSEQ_WREN;           /* Write enable (WREN) signal
                                                                 transmitted */
                        P_CS = (P_CS & 0b00001111)|ucEepromCs;/* EEPROM CS set */
                        CSIIF10 = 0;          /* INTCSI10 interrupt request cleared */
                        CSIMK10 = 0;          /* INTCSI10 interrupt servicing enabled */
                        SS0L.2 = 1;           /* CSI10 operation starts (trigger bit) */
                        SIO10 = 0b00000110;   /* Transmit data set (Write enable (WREN)
                                                 instruction INTCSI10 occurs) */
                }
        }

        if(ucI2sByteCouter==1){
        /* Buffer empty interrupt immediately after starting operation */
                SIO00 = 0x0FF;        /* Dummy data set */
        }
        else{
                if(ucI2sByteCouter==2){
                /* Reception of the first byte */
                        *ucI2sAddress = SIO00; /* Data received via I2S received and saved */
                        SIO00 = 0x0FF;        /* Dummy data set */
                }
                else/*if(ucI2sByteCouter==3)*/{
                /* Reception of the second byte */
                        *ucI2sAddress = SIO00; /* Data received via I2S received and saved
                                     (data reception only) */
                        ucI2sByteCouter = 0;  /* CSI00 reception counter initialized */
                }
                ucI2sAddress++;
        }
}
else if(ucPlayMode==CPLAY){
/*-------------------------------------
 During playback
--------------------------------------*/
        if(ucI2sByteCouter<3){
        /* Transmission of the second or third byte */
                SIO00 = *ucI2sAddress; /* Transmit data set (I2S Transmit data, INTCSI00
                                          occurs) */
                ucI2sAddress++;       /* Address to which the data transmitted via I2S is
                                          saved updated */
        }
        else/*if(ucI2sByteCouter==3)*/{
        /* Timing of synchronization with LRCLK? (Immediately after transmission of the
           third byte starts) */
                ucI2sByteCouter = 0;  /* CSI00 transmission counter initialized */
        }

        if(bI2sMemoryPage){
        /* Page 2 being transmitted */
                if(ucI2sAddress>=&ucMemoryPage2[3*2*42]){
                /* End of page 2 */
                        ucI2sAddress = &ucMemoryPage1[0];
                        /* Address to which the data transmitted via I2S is saved: Page 1 */
                        bI2sMemoryPage = 0;
                        /* Starting reading from the EEPROM */
                        P_CS = (P_CS & 0b00001111)|ucEepromCs;       /* EEPROM selected */
```

```
                        ucEepromSeq = CEEPSEQ_INST;   /* Instruction bytes
                                                          transmitted */
                        CSIIF10 = 0;          /* INTCSI10 interrupt request cleared */
                        CSIMK10 = 0;          /* INTCSI10 interrupt servicing enabled */
                        SS0L.2 = 1;           /* CSI10 operation starts (trigger bit) */
                        SIO10 = 0b00000011;   /* Transmit data (reading specified) set
                                                 (INTCSI10 occurs) */
                }
            }
            else{
            /* Page 1 being transmitted */
                if(ucI2sAddress>=&ucMemoryPage1[3*2*42]){
                /* End of page 1 */
                        ucI2sAddress = &ucMemoryPage2[0];
                        /* Address to which the data transmitted via I2S is saved: Page 2 */
                        bI2sMemoryPage = 1;
                        /* Starting reading from the EEPROM */
                        P_CS = (P_CS & 0b00001111)|ucEepromCs;/* EEPROM selected */
                        ucEepromSeq = CEEPSEQ_INST;           /* Instruction bytes
                                                                 transmitted */
                        CSIIF10 = 0;          /* INTCSI10 interrupt request cleared */
                        CSIMK10 = 0;          /* INTCSI10 interrupt servicing enabled */
                        SS0L.2 = 1;           /* CSI10 operation starts (trigger bit) */
                        SIO10 = 0b00000011;   /* Transmit data (reading specified) set
                                                 (INTCSI10 occurs) */
                }
            }
    }
}

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

     INTCSI10 interrupt servicing
     (using INTCSI10 for CSI transmission or reception)

-----------------------------------------------------------------------------

     During recording, instruction bytes and 24-bit addresses are transmitted
     to an EEPROM, DMA channel 1 starts when transmitting data to the
     EEPROM starts, and transmission continues in CSI10 single transmission
     mode.
     During playback, 24-bit addresses are transmitted to an EEPROM,
     DMA channels 0 and 1 start when receiving data from the EEPROM starts,
     and reception continues in CSI10 single reception mode.  However, DMA
     channel 1 is used for setting dummy data.

     Register bank 1 used

*****************************************************************************/
__interrupt void fn_intcsi10(void)
{
     EI();          /* Multiple interrupts enabled */

     if(ucPlayMode==CREC){
     /*-------------------------------------
      During recording
     -------------------------------------*/
       ucEepromSeq++;                      /* EEPROM transfer sequence updated */
        /* Instruction byte transmission */
        switch(ucEepromSeq){
          case CEEPSEQ_INST:
                /* Instruction byte transmission timing */
                P_CS |= 0b11110000;                           /* Non-active */
                NOP();
                P_CS = (P_CS & 0b00001111)|ucEepromCs;        /* EEPROM selected */
                SIO10 = 0b00000010;                           /* Transmission of write
                                                                 specification */
```

```
            break;

        case CEEPSEQ_ADDRH:
            /* Transmission of the higher bits of the 24-bit address */
            SIO10 = (unsigned char)(ushEepromAddress>>8);
            break;

        case CEEPSEQ_ADDRM:
            /* Transmission of the middle bits of the 24-bit address */
            SIO10 = (unsigned char)(ushEepromAddress);
            break;

        case CEEPSEQ_ADDRL:
            /* Transmission of the lower bits of the 24-bit address */
            SIO10 = 0x000;
            break;

        case CEEPSEQ_DATA:
        default:
            /* Starting data transmission */
            /* Starting DMA transfer */
            if(bI2sMemoryPage){
            /* Page 2 being transmitted */
                DRA1 = (unsigned short)&ucMemoryPage1[1];
                /* Transmit buffer set to page 1 */
            }
            else{
            /* Page 1 being transmitted */
                DRA1 = (unsigned short)&ucMemoryPage2[1];
                /* Transmit buffer set to page 2 */
            }
            DBC1 = 3*2*42-1;            /* One page */
            DST1 = 1;                   /* DMA1 transfer enabled (for CSI10 transmission) */

            DMAIF1 = 0;                 /* INTDM1 interrupt request cleared */
            DMAMK1 = 0;                 /* INTDM1 interrupt servicing enabled */
            CSIIF10 = 0;                /* INTCSI10 interrupt request cleared */
            CSIMK10 = 1;                /* INTCSI10 interrupt servicing disabled */

            DMAMK0 = 1;                 /* INTDM0 interrupt servicing disabled */

            SS0L.2 = 1;                 /* CSI00 operation starts (trigger bit) */

            if(bI2sMemoryPage){
            /* Page 2 being transmitted */
                SIO10 = *ucMemoryPage1;       /* Transmit data set (INTCSI00 occurs) */
            }
            else{
                SIO10 = *ucMemoryPage2;       /* Transmit data set (INTCSI00 occurs) */
            /* Page 1 being transmitted */
            }

            break;
        }
    }
    else{
    /*-------------------------------------
     During playback
    -------------------------------------*/
      ucEepromSeq++;                    /* EEPROM transfer sequence updated */
       /* Instruction byte transmission */
       switch(ucEepromSeq){
        case CEEPSEQ_ADDRH:
            /* Transmission of the higher bits of the 24-bit address */
            SIO10 = (unsigned char)(ushEepromAddress>>8);
            break;
```

```
            case CEEPSEQ_ADDRM:
                  /* Transmission of the middle bits of the 24-bit address */
                  SIO10 = (unsigned char)(ushEepromAddress);
                  break;

            case CEEPSEQ_ADDRL:
                  /* Transmission of the lower bits of the 24-bit address */
                  SIO10 = 0x000;
                  break;

            case CEEPSEQ_DATA:
            default:
             /* Starting data reception */

             SCR02 = 0b0111000000000111;     /* Communication format setting */
             /*        |||||||||||||+++------- DLS022 to DLS020: 8-bit data length */
             /*        |||||||||||||+---------- <Fixed to 0> */
             /*        ||||||||||++----------- SLC021 and SLC020: Unused (fixed to 0) */
             /*        |||||||||+------------- <Fixed to 0> */
             /*        ||||||||+-------------- DIR02: Input and output performed with MSB first */
             /*        ||||||++--------------- PTC021 and PTC020: Unused (fixed to 00) */
             /*        |||||+----------------- EOC02: Unused (fixed to 0) */
             /*        ||||+------------------ <Fixed to 0> */
             /*        ||++------------------- CKP02/DAP02: Phases of data and clock in CSI mode
             /*        ||                                   selected */
             /*        ||                      [11 selected] */
             /*        ||                                _   _   _   _   _   _   _   _    */
             /*        ||                      SCK02____| |_| |_| |_| |_| |_| |_| |_| |   */
             /*        ||                           __ ___ ___ ___ ___ ___ ___ ___ ____   */
             /*        ||                      SO02 __|___|___|___|___|___|___|___|____   */
             /*        ||                              D7  D6  D5  D4  D3  D2  D1  D0    */
             /*        ||         SI02 input timing ____|___|___|___|___|___|___|___|__   */
             /*        ||                                                                 */
             /*        ++--------------------- TXE00/RXE00: Only reception performed     */

            DRA1 = (unsigned short)&ucMemoryPage1[0];       /* Address for setting dummy data */
            DBC1 = 3*2*42-1;                                /* Page size set */

            if(bI2sMemoryPage){
            /* Page 2 being transmitted */
                DRA0 = (unsigned short)&ucMemoryPage1[0];/* Receive buffer set to page 1 */
            }
            else{
            /* Page 1 being transmitted */
                DRA0 = (unsigned short)&ucMemoryPage2[0];/* Receive buffer set to page 2 */
            }
                DBC0 = 3*2*42;                           /* Page size set */

                /* Starting DMA transfer */
                DST1 = 1;     /* DMA1 transfer enabled (for CSI10 transmission) */
                DST0 = 1;     /* DMA0 transfer enabled (for CSI10 reception) */

                DMAMK1 = 1;   /* INTDM1 interrupt servicing disabled */
                CSIIF10 = 0;  /* INTCSI10 interrupt request cleared */
                CSIMK10 = 1;  /* INTCSI10 interrupt servicing disabled */

                DMAIF0 = 0;   /* INTDM0 interrupt request cleared */
                DMAMK0 = 0;   /* INTDM0 interrupt servicing enabled */

                SS0L.2 = 1;   /* CSI10 operation starts (trigger bit) */
                SIO10 = 0x0FF;/* Dummy data set (INTCSI00 occurs) */
                break;
            }
        }
    }
```

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

     INTDMA0 interrupt servicing
     (using INTDMA0 for CSI reception)

-----------------------------------------------------------------------------

     This interrupt occurs once data of one page has been received from an
     EEPROM during playback.  Which of the four EEPROMs is to be used is
     selected and operation of the I2S bus interface is started and stopped.

     Register bank 1 used

*****************************************************************************/
__interrupt void fn_intdma0(void)
{
     /* Waiting for completion of waiting for communication completion */
     while(SSR02 & 0b0000000001000000)    {}

     /* Completing receiving data of one page */
     P_CS |= 0b11110000;           /* Non-active */
     ucEepromSeq = CEEPSEQ_INST;

     if(ucPlayMode==CPLAY){
     /* During playback */
     /*----------------------------------------------
      Starting the operation of the I2S bus interface
     ----------------------------------------------*/
            /* Selecting EEPROM */
            if((ucEepromCs & 0b10000000)==0b00000000){
            /* Selecting all EEPROMs once */
                   ushEepromAddress++;
                   if(ushEepromAddress>=128000/256){
                          /* Stopping the operation of the I2S bus interface */
                          P_LRCLK = 1;   /* LRCLK output disabled (high level) */
                          TT0L.0 = 1;    /* TO00 output operation stopped (trigger bit) */
                          ST0L.0 = 1;    /* CSI000 output operation stopped (trigger bit) */
                          ST0L.2 = 1;    /* CSI010 output operation stopped (trigger bit) */
                          ucPlayMode = CPLAY_END;       /* Playback being terminated */
                   }
                   else{
                          ucEepromCs = 0b11100000;
                   }
            }
            else{
                   ucEepromCs = (ucEepromCs|0b00001000) <<1;
            }
     }
     else if(ucPlayMode==CPLAY_START){
     /* Starting preparing for playback */
     /* Starting reading from an EEPROM */
            ucPlayMode = CPLAY_SET;                      /* Preparing for playback */
            P_CS = (P_CS & 0b00001111)|0b11010000;       /* EEPROM selected */
            bI2sMemoryPage = 0;
            CSIIF10 = 0;                  /* INTCSI10 interrupt request cleared */
            CSIMK10 = 0;                  /* INTCSI10 interrupt servicing enabled */
            SS0L.2 = 1;                   /* CSI10 operation starts (trigger bit) */
            SIO10 = 0b00000011;           /* Transmit data (reading specified) set (INTCSI10
                                             occurs) */
     }
     else if(ucPlayMode==CPLAY_SET){
     /* Preparing for playback */
            ucPlayMode = CPLAY;                   /* During playback */
            ucI2sAddress = &ucMemoryPage1[1];     /* Start of the transmit buffer set */
            bI2sMemoryPage = 0;                   /* page 1 transmit */
            /* Starting the operation of the I2S bus interface */
            SS0L.0 = 1;                   /* CSI00 operation starts (trigger bit) */
```

```
            P_LRCLK = 0;                  /* LRCLK operation starts */
            TS0L.0 = 1;                   /* TO00 output operation starts (trigger bit) */
            SIO00 = ucMemoryPage1[0];     /* Transmit data set (INTCSI00 occurs) */
     }
     DST1 = 0;      /* DMA1 transfer stopped */
     DST0 = 0;      /* DMA0 transfer stopped */
     DMAMK0 = 1;    /* INTDM0 interrupt servicing disabled */
}

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

     INTDMA1 interrupt servicing
     (using INTDMA1 for CSI transmission)

-----------------------------------------------------------------------------

     This interrupt occurs when data of one page has been transmitted to an EEPROM
     during recording.  Which of the four EEPROMs is to be used is selected and
     operation of the I2S bus interface is stopped.

     Register bank 1 used

*****************************************************************************/
__interrupt void fn_intdma1(void)
{
     /* Waiting for completion of waiting for communication completion */
     while(SSR02 & 0b0000000001000000)    {}

     /* Completing transmitting data of one page */
     P_CS |= 0b11110000;           /* Completing transmitting data of one page */
     ucEepromSeq = CEEPSEQ_RESET;

     /* EEPROM selection */
     if((ucEepromCs & 0b10000000)==0b00000000){
            ushEepromAddress++;
            if(ushEepromAddress>=128000/256){
                   /* Stopping the operation of the I2S bus interface */
                   P_LRCLK = 1;           /* LRCLK output disabled (high level) */
                   TT0L.0 = 1;            /* TO00 output operation stopped (trigger bit) */
                   ST0L.0 = 1;            /* CSI000 output operation stopped (trigger bit) */
                   ST0L.2 = 1;            /* CSI010 output operation stopped (trigger bit) */
                   ucPlayMode = CPLAY_END;       /* Playback stopped */
            }
            else{
                   ucEepromCs = 0b11100000;
            }
     }
     else{
            ucEepromCs = (ucEepromCs|0b00001000) <<1;
     }
     DST1 = 0;      /* DMA1 transfer stopped */
     DST0 = 0;      /* DMA0 transfer stopped */
     DMAMK1 = 1;    /* INTDM1 interrupt servicing disabled */
}

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

     I2C bus interface write processing for setting the registers
     in the audio codec

-----------------------------------------------------------------------------

The data set to the registers in the audio codec is written by using the
functions of I2C bus interface IIC0.

-----------------------------------------------------------------------------
```

```
     [I N]   *addr   :Higher 4 bits of the start address of the ROM table holding
                      the register setting values
              size   :ROM table size
     [OUT]   -

*****************************************************************************/
void fn_I2cWrite(unsigned char *addr, unsigned char size)
{
     register unsigned char cnt;

     IICE0 = 1;                          /* Operation enabled */

     for(cnt=0; cnt<=size/3; cnt++){
             /* Start condition issuance */
             STT0 = 1;

             /* Slave address transmission */
             while(!STD0)   {}      /* Address transmission period? No, */
             IICIF0 = 0;
             IIC0 = 0b00110100;     /* Address transmission starts (writing specified) */
             while(!IICIF0) {}      /* Transmission completed? No, */
             if(ACKD0){
                     /* Write address transmission */
                     IICIF0 = 0;            /* INTIIC10 interrupt request cleared */
                     IIC0 = *addr;          /* Data transmission starts */
                     while(!IICIF0) {}      /* Transmission completed? No, */

                     if(ACKD0){
                             /* Write data transmission */
                             addr++;                /* Reference position updated */
                             IICIF0 = 0;            /* INTIIC10 interrupt request cleared */
                             IIC0 = *addr;          /* Data transmission starts */
                             while(!IICIF0) {}      /* Transmission completed? No, */

                             if(ACKD0){
                                     /* stop condition issued */
                                     SPT0 = 1;

                                     /* Wait control */
                                     addr++;                /* Reference position updated */
                                     if(*addr!=0){
                                             TDR01 = 195*(*addr)-1; /* Interval set (5ms*addr) */
                                             TMIF01 = 0;            /* INTTM01 interrupt request
                                                                        cleared */
                                             TS0L.1 = 1;            /* Timer operation starts */
                                             while(!TMIF01) {}      /* Interval elapsed? No, */
                                             TT0L.1 = 1;            /* Timer operation stopped */
                                     }
                                     addr++;                /* Reference position updated */
                             }
                             else{
                                     SPT0 = 1;      /* Stop condition issuance */
                                     cnt--;         /* Retransmission */
                                     addr--;
                             }
                     }
                     else{
                             SPT0 = 1;      /* Stop condition issuance */
                             cnt--;         /* Retransmission */
                     }
             }
             else{
                     SPT0 = 1;      /* Stop condition issuance */
                     cnt--;         /* Retransmission */
             }
     }
}
```

APPENDIX B REVISION HISTORY

Editon	Data Published	Page	Revision
1st edition	February 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