RX Family

DSMIF Module Using Firmware Integration Technology

Introduction

This application note describes the DSMIF module which uses Firmware Integration Technology (FIT). This module uses DSMIF to convert a high sampling rate 1-bit digital data stream of delta-sigma modulation into low sampling rate 16-bit digital data by filtering. In this document, this module is referred to as the DSMIF FIT module.

Target Device

The following is a list of devices that are currently supported by this API:

- RX72M Group

Contents

1. Overview

The DSMIF FIT module provides a means to receive delta-sigma modulated 1-bit digital streams by using the delta-sigma modulator interface (DSMIF).

This module supports the following features:

- Filters a 1-bit digital data input stream and converts it to 16-bit digital data
- Operates as a master or slave in terms of the delta-sigma clock
- Detects and handles three types of error: Overcurrent, short-circuit, and current sum abnormality

1.1 DSMIF FIT Module

The DSMIF FIT module is implemented in a project and used as the API. Refer to 2.11 Adding the FIT Module for details on implementing the module to the project.

2. API Information

The API functions of the DSMIF FIT module adhere to the Renesas API naming standards.

2.1 Hardware Requirements

This driver requires your MCU supports the following feature:

- DSMIF

2.2 Hardware Resource Requirements

This section details the hardware peripherals that this driver requires. Unless explicitly stated, these resources must be reserved for the driver, and cannot be used elsewhere in the application.

2.2.1 Peripheral Required

Delta-Sigma Modulator Interface (DSMIF)

2.2.2 Other Peripherals Used

The driver requires I/O port pins to be assigned for DSMIF receive and transmit signals. Assigned pins may not be used for GPIO.

2.3 Software Requirements

This driver is dependent upon the following packages:

- Renesas Board Support Package (r_bsp) v5.50 or higher
- CMT Module (r_cmt) v3.40 or higher

2.4 Supported Toolchain

The operation of the DSMIF FIT module has been confirmed with the toolchain listed as C compiler in 6.1 Confirmed Operation Environment.

2.5 Usage of Interrupt Vector

The DSMIF interrupts are enabled by calling R_DSMIF_Start function.

Table 2.1 List of Usage of Interrupt Vectors

Device	Contents
RX72M	GROUPBL2 interrupt (Vector number: 107) • OCDI0 (overcurrent detection): group interrupt source number 1 • SUMEI0 (current sum error): group interrupt source number 2 • SCDI0 (short-circuit detection): group interrupt source number 3 • OCDI1 (overcurrent detection): group interrupt source number 4 • SUMEI1 (current sum error): group interrupt source number 5 • SCDI1 (short-circuit detection): group interrupt source number 6

2.6 Header Files

All API calls and their supporting interface definitions are located in “r_dsmif_rx_if.h”.

Build-time configuration options are selected or defined in the file "r_dsmif_rx_config.h”.

To reference DSMIF API elements in this FIT Module from your code include the following:

#include “r_dsmif_rx_if.h”

2.7 Integer Types

This software uses ANSI C99. These types are defined in stdint.h.

2.8 Configuration Overview

The configuration options in the DSMIF FIT module are specified in r_dsmif_rx_config.h.

The option names and setting values are listed in the table below.

Configuration option in *r_dsmif_rx_config.h*	
DSMIF_ERR_INTR_PRI_LVL Note: Default value = 15	Sets the GROUPBL2 interrupt level.
DSMIF_SETTING_WAIT_CNT Note: Default value = 250	Set the time for waiting the filter settling time (unit: usec)
DSMIF_OCTLR_VALUE Note: Default value = 0	Set the lower limit for judging the detection of overcurrent error.
DSMIF_SHORT_CNT_1 Note: Default value = 0x1FFE	Set the threshold for the input of 1-value data for judging the detection of short-circuit error.
DSMIF_SHORT_CNT_0 Note: Default value = 0x1FFE	Set the threshold for the input of 0-value data for judging the detection of short-circuit error.

2.9 Code size

Typical code sizes associated with this module are listed below.

The table lists reference values when the C compiler’s compile options are set to their default values, as described in 2.4, Supported Toolchain. The compile option default values are optimization level: 2, optimization type: for size, and data endianness: little-endian. The code size varies depending on the C compiler version and compile options.

ROM and RAM Sizes		
Device	**Category**	**Size**
RX72M	ROM	2040 bytes
	RAM	50 bytes

2.10 Arguments

This section documents the enumerations, unions, and structures used as arguments to API functions. These are included in the r_dsmif_rx_if.h header file along with the API function prototype declarations.

```
typedef struct
{
    uint32_t unit_no;           /* Select the DSMIF unit to open (0/1) */
    uint32_t mode;              /* Select the I/O direction of DSMCLK (Master/Slave) */
    uint32_t ckdiv;             /* Select the clock divisor when the DSMIF serves as a master */
    uint32_t channel;           /* Select the channel for which the pin setting is to be made
                                   (as the logical OR of the values on the given pins) */
    uint32_t dis_error;         /* Select the DSMIF errors for which detection is to be disabled
                                   (as the logical OR of the values on the given pins) */
    void (*pcallback)(uint32_t); /* Specify the callback function to indicate the occurrence of an error. */
} st_dsmif_config_t;

/* DSMIF Unit No */
#define DSMIF_UNIT_DSMIF0           (0U)    /* DSMIF unit 0 */
#define DSMIF_UNIT_DSMIF1           (1U)    /* DSMIF unit 1 */

/* DSMIF Mode Setting */
#define DSMIF_MODE_SLAVE            (0U)    /* Slave operation */
#define DSMIF_MODE_MASTER           (1U)    /* Master operation */

/* DSMIF Channel & Error Disable No */
#define DSMIF_CH0                   (1U)    /* channel 0 */
#define DSMIF_CH1                   (2U)    /* channel 1 */
#define DSMIF_CH2                   (4U)    /* channel 2 */
#define DSMIF_SUM                   (8U)    /* Disables detection of current sum errors */

/* DSMIF Filter Setting Code -----------------------------------------------*/
#define DSMIF_FILTER_SET_0          (0U)    /* filter number 0 */
#define DSMIF_FILTER_SET_1          (1U)    /* filter number 1 */
#define DSMIF_FILTER_SET_2          (2U)    /* filter number 2 */
#define DSMIF_FILTER_SET_3          (3U)    /* filter number 3 */
#define DSMIF_FILTER_SET_4          (4U)    /* filter number 4 */
#define DSMIF_FILTER_SET_5          (5U)    /* filter number 5 */
#define DSMIF_FILTER_SET_6          (6U)    /* filter number 6 */
#define DSMIF_FILTER_SET_7          (7U)    /* filter number 7 */
#define DSMIF_FILTER_SET_8          (8U)    /* filter number 8 */
#define DSMIF_FILTER_SET_9          (9U)    /* filter number 9 */
#define DSMIF_FILTER_SET_10         (10U)   /* filter number 10 */
#define DSMIF_FILTER_SET_11         (11U)   /* filter number 11 */
#define DSMIF_FILTER_SET_12         (12U)   /* filter number 12 */
```

```
#define DSMIF_FILTER_SET_13            (13U)   /* filter number 13 */
#define DSMIF_FILTER_SET_14            (14U)   /* filter number 14 */

/* DSMIF Target Register to Read -------------------------------------------*/
#define DSMIF_READ_DSMIF0_CDR0         (1U)    /* unit 0 CDR0 register */
#define DSMIF_READ_DSMIF0_CDR1         (2U)    /* unit 0 CDR1 register */
#define DSMIF_READ_DSMIF0_CDR2         (3U)    /* unit 0 CDR2 register */
#define DSMIF_READ_DSMIF1_CDR0         (4U)    /* unit 1 CDR0 register */
#define DSMIF_READ_DSMIF1_CDR1         (5U)    /* unit 1 CDR1 register */
#define DSMIF_READ_DSMIF1_CDR2         (6U)    /* unit 1 CDR2 register */
#define DSMIF_READ_DSMIF0_CCDR0        (7U)    /* unit 0 CCDR0 register */
#define DSMIF_READ_DSMIF0_CCDR1        (8U)    /* unit 0 CCDR1 register */
#define DSMIF_READ_DSMIF0_CCDR2        (9U)    /* unit 0 CCDR2 register */
#define DSMIF_READ_DSMIF1_CCDR0        (10U)   /* unit 1 CCDR0 register */
#define DSMIF_READ_DSMIF1_CCDR1        (11U)   /* unit 1 CCDR1 register */
#define DSMIF_READ_DSMIF1_CCDR2        (12U)   /* unit 1 CCDR2 register */
#define DSMIF_READ_DSMIF0_TCDR0        (13U)   /* unit 0 TCDR0 register */
#define DSMIF_READ_DSMIF0_TCDR1        (14U)   /* unit 0 TCDR1 register */
#define DSMIF_READ_DSMIF0_TCDR2        (15U)   /* unit 0 TCDR2 register */
#define DSMIF_READ_DSMIF1_TCDR0        (16U)   /* unit 1 TCDR0 register */
#define DSMIF_READ_DSMIF1_TCDR1        (17U)   /* unit 1 TCDR1 register */
#define DSMIF_READ_DSMIF1_TCDR2        (18U)   /* unit 1 TCDR2 register */
#define DSMIF_READ_DSMIF0_OCDR0        (19U)   /* unit 0 OCDR0 register */
#define DSMIF_READ_DSMIF0_OCDR1        (20U)   /* unit 0 OCDR1 register */
#define DSMIF_READ_DSMIF0_OCDR2        (21U)   /* unit 0 OCDR2 register */
#define DSMIF_READ_DSMIF1_OCDR0        (22U)   /* unit 1 OCDR0 register */
#define DSMIF_READ_DSMIF1_OCDR1        (23U)   /* unit 1 OCDR1 register */
#define DSMIF_READ_DSMIF1_OCDR2        (24U)   /* unit 1 OCDR2 register */
```

2.11 Adding the FIT Module to Your Project

This module must be added to each project in which it is used. Renesas recommends using “Smart Configurator” described in (1) or (3). However, “Smart Configurator” only supports some RX devices. Please use the methods of (2) or (4) for unsupported RX devices.

(1) Adding the FIT module to your project using “Smart Configurator” in e^2 studio
By using the “Smart Configurator” in e^2 studio, the FIT module is automatically added to your project. Refer to “Renesas e^2 studio Smart Configurator User Guide (R20AN0451)” for details.

(2) Adding the FIT module to your project using “FIT Configurator” in e^2 studio
By using the “FIT Configurator” in e^2 studio, the FIT module is automatically added to your project. Refer to “Adding Firmware Integration Technology Modules to Projects (R01AN1723)” for details.

(3) Adding the FIT module to your project using “Smart Configurator” on CS+
By using the “Smart Configurator Standalone version” in CS+, the FIT module is automatically added to your project. Refer to “Renesas e^2 studio Smart Configurator User Guide (R20AN0451)” for details.

(4) Adding the FIT module to your project in CS+
In CS+, please manually add the FIT module to your project. Refer to “Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826)” for details.

3. API Functions

3.1 R_DSMIF_Create

Initial settings of the DSMIF.

Parameters

void R_DSMIF_Create(st_dsmif_config_t* config);

Parameters

config

A pointer to the information on DSMIF settings

Refer to "2.10 Arguments" for information on the structure member settings.

Return Values

None

Properties

Prototyped in file “r_dsmif_rx_if.h”

Description

This function makes the following initial settings of the DSMIF.

- Release the DSMIF from the module-stop state
- Initialize the DSMIF registers
- Set error interrupts
- Set ports to input or output

Example

```
st_dsmif_config_t dsmif_config;

dsmif_config.unit_no = DSMIF_UNIT_DSMIF1;
dsmif_config.mode   = DSMIF_MODE_SLAVE;
dsmif_config.channel = DSMIF_CH0;
dsmif_config.dis_error = (DSMIF_CH1| DSMIF_CH2 | DSMIF_SUM);
dsmif_config.pcallback = &dsmif_error_callback;
R_DSMIF_Create(&dsmif_config);
```

3.2 R_DSMIF_Start

Start filtering by the DSMIF.

Format

```
void R_DSMIF_Start(uint32_t unit_no);
```

Parameters

unit_no

Select the DSMIF unit (0 to 1)

Return Values

None

Properties

Prototyped in file “r_dsmif_rx_if.h”

Description

This function makes the following settings necessary to start filtering (monitoring the value for current) by the DSMIF

- Set the DSMIF error conditions
- Start filtering by the DSMIF
- Wait for the filter to start operating
- Clear the DSMIF error status
- Enable error interrupts

Example

```
/* Start DSMIF filtering */
R_DSMIF_Start(DSMIF_UNIT_DSMIF1);
```

3.3 R_DSMIF_Stop

Stop filtering by the DSMIF.

Format

void R_DSMIF_Stop(uint32_t unit_no);

Parameters

unit_no

Select the DSMIF unit (0 to 1)

Return Values

None

Properties

Prototyped in file “r_dsmif_rx_if.h”

Description

This function makes the following settings necessary to stop filtering (monitoring the value for current) by the DSMIF.

- Disable error interrupts
- Stop filtering by the DSMIF

Example

```
/* Stop DSMIF filtering */
R_DSMIF_Stop(DSMIF_UNIT_DSMIF1);
```

3.4 R_DSMIF_SetFilter

Change the DSMIF filter setting.

Format

void R_DSMIF_SetFilter(uint32_t unit_no, uint8_t filter_setting);

Parameters

unit_no

Select the DSMIF unit (0 to 1)

filter_setting

Select the filter number (0 to 14)

Refer to "Table 3.1 Setting contents by filter number" for the contents set by filter number.

Return Values

None

Properties

Prototyped in file “r_dsmif_rx_if.h”

Description

Change the filter settings of DSMIF.

The following items are changed according to the specified filter number.

- Sinc Filter Order
- Decimation Ratio
- Resolution

Example

R_DSMIF_SetFilter(DSMIF_UNIT_DSMIF1, DSMIF_FILTER_SET_14);

Table 3.1 Setting contents by filter number

Filter Number	Sinc Filter Order	Decimation Ratio	Resolution
0	3 ($Sinc^3$)	8	9 bits
1		16	12 bits
2		32	15 bits
3		64	16 bits
4		128	16 bits
5		256	16 bits
6	2 ($Sinc^2$)	16	8 bits
7		32	10 bits
8		64	12 bits
9		128	14 bits
10		256	16 bits
11	1 ($Sinc^1$)	32	5 bits
12		64	6 bits
13		128	7 bits
14		256	8 bits

3.5 R_DSMIF_ReadData

Reads data converted by DSMIF filtering operation.

Format

uint16_t R_DSMIF_ReadData(uint8_t data_reg);

Parameters

data_reg

Select the register to read the value.
For details on constants used as register names, see "2.10 Arguments".

Return Values

Data register value

Properties

Prototyped in file “r_dsmif_rx_if.h”

Description

This function reads the 16-bit digital data that were converted through the DSMIF filter. The value is read from the applicable register specified by the argument.

Example

```
/* Read DSMIF Unit1 CDR0 register */

uint32_t read_data;

read_data = R_DSMIF_ReadData(DSMIF_READ_DSMIF1_CDR0);
```

3.6 R_DSMIF_GetErrorStatus

Reads the DSMIF error status register.

Format

uint32_t R_DSMIF_GetErrorStatus(uint32_t unit_no);

Parameters

unit_no

Select the DSMIF unit (0 to 1)

Return Values

DSMIF status register (DSSR) value

Properties

Prototyped in file “r_dsmif_rx_if.h”

Description

Reads the value of DSMIF status register (DSSR) that stores error detection status.

Example

```
/* Read DSMIF Unit1 status register */
uint32_t err_sts;
err_sts = R_DSMIF_GetErrorStatus(DSMIF_UNIT_DSMIF1);
```

4. Pin Setting

To use the DSMIF FIT module, assign input/output signals of the peripheral function to pins with the multi-function pin controller (MPC). The pin assignment is referred to as the “Pin Setting” in this document. Please perform the pin settings after calling the R_DSMIF_Open function and before calling the R_DSMIF_Start function.

When performing the Pin Setting in the e^2 studio, the Pin Setting feature of the FIT configurator or the Smart

Configurator can be used. When using the Pin Setting feature, a source file is generated according to the option selected in the Pin Setting window in the FIT configurator or the Smart Configurator. Pins are configured by calling the function defined in the source file. Refer to Table 4.1 for details.

Table 4.1 Function Output by the FIT Configurator

MCU Used	Function to be Output	Remarks
RX72M	R_DSMIF_PinSet_DSMIFx	x: Unit number

5. Demo Projects

DSMIF demo projects are complete stand-alone programs. They include function main() that utilizes the module and its dependent modules.

5.1 dsmif_demo_rskrx72m

The dsmif_demo_rskrx72m program outputs the current measurement value received from the Photo coupler (SDIP8) Evaluation Sub Board (the DSMIF sub board) connected to the Renesas Starter Kit+ for RX72M board (the RSKRX72M board) to the virtual console of e[2] studio.

5.1.1 Board Settings and its Connection

For the details on the board, refer to the Renesas Starter Kit+ for RX72M Board User's Manual (the RSKRX72M board manual).

5.1.1.1 Configuring the RSKRX72M Board

To run the sample program described in this application note on the RSKRX72M board, enable DSMDAT3 in the board configuration. Specifically, remove the chip resistor mounted by default as R345 and mount it on as R344.

For details, refer to the Section 6.9, DSMIF Settings of the CPU board manual.

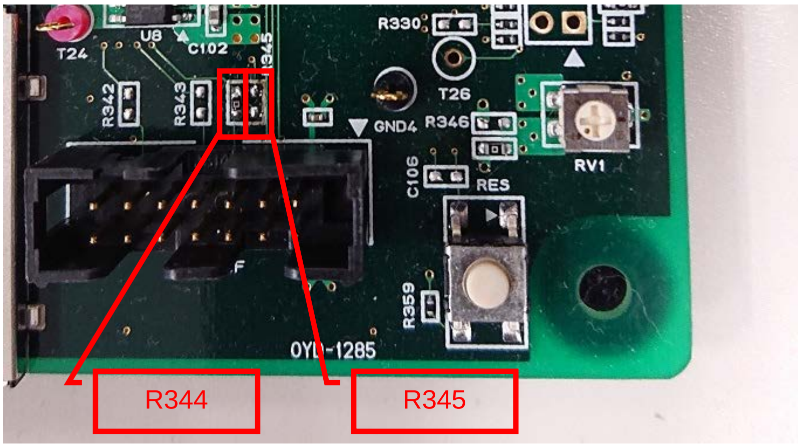

Figure 5.1 DSMDAT3 Resistor Setting

5.1.1.2 Connecting the DSMIF Sub-Board to the RSKRX72M Board

Connect the DSMIF sub-board to the J9 connector of the RSKRX72M board in the direction shown in Figure 5.2 below.

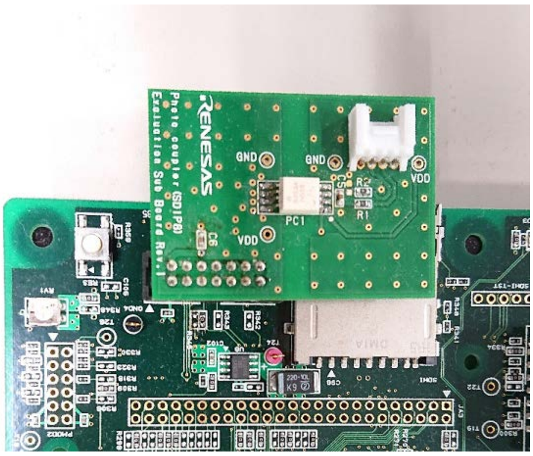

Figure 5.2 Connecting the DSMIF Sub-Board to the RSKRX72M Board

5.1.1.3 Connecting Power to the DSMIF Sub-Board

Supply DC5V to the VDD and GND pins of the DSMIF sub-board.

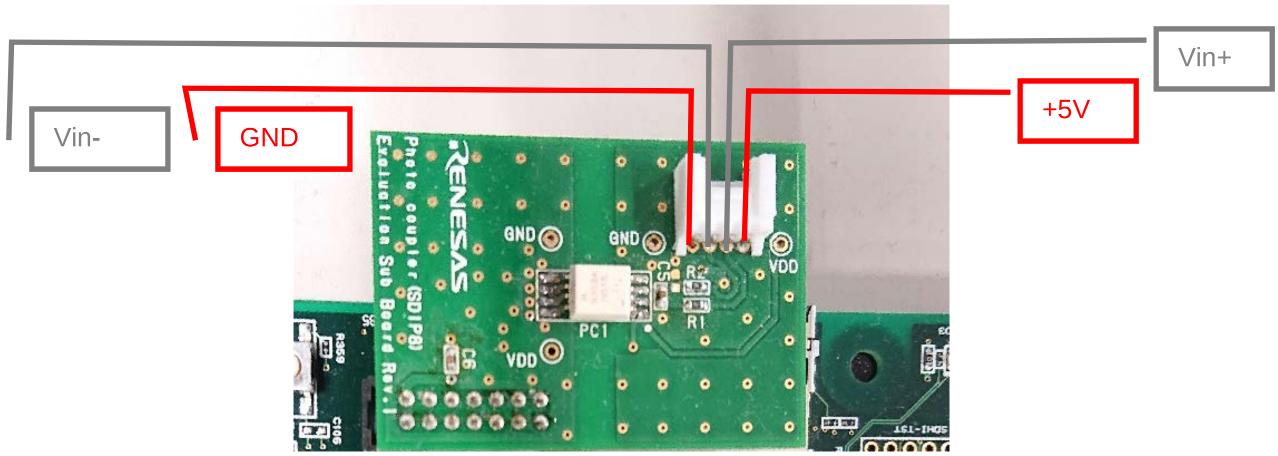

Figure 5.3 Connecting Power to the DSMIF Sub-Board

5.1.1.4 Connecting the Measurement Signal to the DSMIF Sub-Board

Supply the analog voltage to be measured to the Vin+ and Vin- pins of the DSMIF sub-board.

The input value must fall within the range for which the PS9352A mounted on the sub-board is rated (within ±200 mV is recommended).

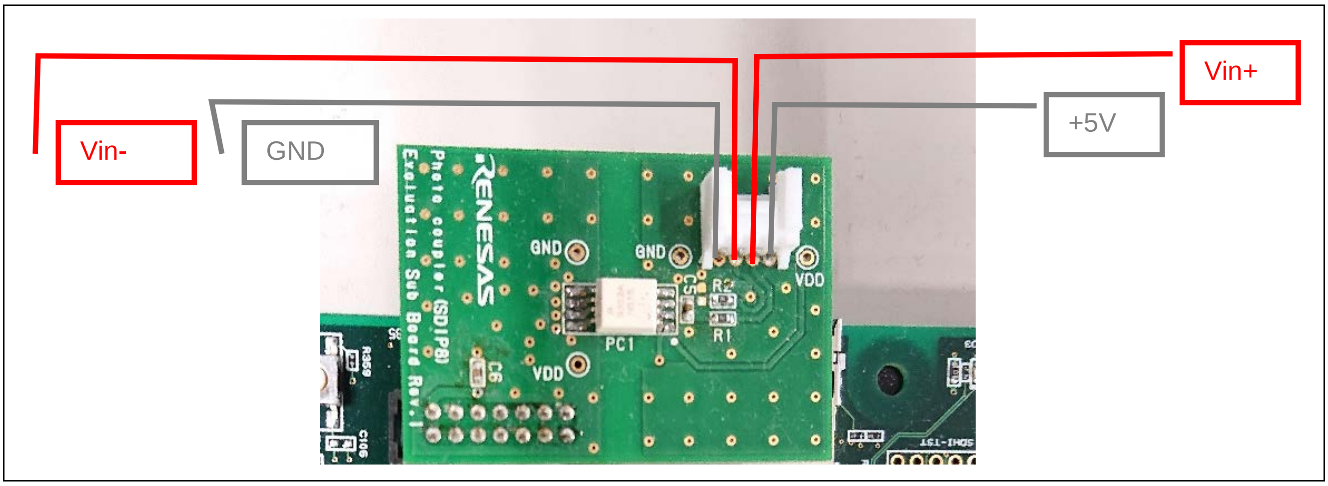

Figure 5.4 Connecting the Measurement Signals to the DSMIF Sub-Board

5.1.2 Procedure for Running the Sample Program

The procedure for running the sample program is described below.

5.1.2.1 Sample Program Menu

When the program is executed and ready for making communications, the menu appears on the console as shown below.

(1) Top Menu

After the program starts, the following top menu will appear.

```
Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]


[8] Log Output
[9] Exit
>
```

Figure 5.5 Top Menu

In the top menu, you can select any of the following operations.

- Selecting [1]: Starts the measurement of the values for current.
- Selecting [2]: Goes to the scan frequency setting menu.
- Selecting [3]: Goes to the filter setting menu.
- Selecting [8]: Outputs a log.
- Selecting [9]: Ends the sample program.

(2) Scan Frequency Setting Menu

Select [2] from the top menu to change the frequency of scanning.

```
>2[Enter]

Input the number to select the measurement mode.

[1] High Frequency
[2] Low Frequency

>2[Enter]
```

Figure 5.6 Scan Frequency Setting Menu

Select the number 1 or 2 for the required frequency.

If you want to exit without changing the frequency, press [Enter] without specifying a number.

In the example, [2], Low Frequency is selected.

A message indicating that the setting was changed is displayed. The top menu then appears again.

You can confirm that the setting is reflected in the menu.

```
> 2

Set to Low frequency mode.

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [Low]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

Figure 5.7 Updated Top Menu following Change of the Scan Frequency Setting

(3) Filter Setting Menu

Select [3] from the menu to change the filter settings.

```
>3[Enter]

Input the number to set the filter (0-14)

     SINC filter    Decimation    Data
-------------------------------------------------------
[0]    3            1/8           b8-b0, 0000000
[1]    3            1/16          b11-b0, 0000
[2]    3            1/32          b14-b0, 0
[3]    3            1/64          b17-b2
[4]    3            1/128         b20-b5
[5]    3            1/256         b23-b8
[6]    2            1/16          b7-b0, 00000000
[7]    2            1/32          b9-b0, 000000
[8]    2            1/64          b11-b0, 0000
[9]    2            1/128         b13-b0, 00
[10]   2            1/256         b15-b0
[11]   1            1/32          b4-b0, 00000000000
[12]   1            1/64          b5-b0, 0000000000
[13]   1            1/128         b6-b0, 000000000
[14]   1            1/256         b7-b0, 00000000

>5
```

Figure 5.8 Filter Setting Menu

Select the number from 0 to 14 for the combination of required settings for the filter. See, "Table 3.1 Setting contents by filter number" for the setting values.

If you want to exit without changing filter settings, press [Enter] without specifying a number.

In this example, [5] (three-stage filter, decimation clock setting 1/256, last data to be output: b23 to b8) is selected.

A message indicating that the setting was changed is displayed. The top menu then appears again.

You can confirm that the setting is reflected in the menu.

```
> 5

Set to 5.

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [Low]
[3] Filter Settings [5]

[8] Log Output
[9] Exit

>
```

Figure 5.9 Updated Top Menu following Change of the Filter Settings

(4) Starting Measurement of Current Values

By selecting [1] from the top menu, monitoring of the value of current starts with the specified scanning frequency and filter settings.

```
Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>1[Enter]

<start>
33792
```

Figure 5.10 Screen after Starting Current Measurement

(a) High-Frequency Scanning

In the high-frequency scan mode, the values for current are scanned 1000 times per second and the results are logged on completion of each set (1000 times) of scanning.

On completion of logging of the results of scanning 10,000 times, the top menu is re-displayed.

```
<start>
33792
33280
.
.
32128
<end>

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

Figure 5.11 Result of the High-Frequency Scan

(b) Low-Frequency Scanning

In the low-frequency scan mode, the values for current are scanned once per second. The log of the results is displayed after each scan.

To stop scanning, press SW1 on the CPU board. This stop scanning and returns to the top menu.

```
<start>
34048
32512
33408
34304
33792
[Press SW1]
<end>

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [Low]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

Figure 5.12 Result of Low-Frequency Scanning

(c) Stop Scanning on Error Detection

If an error (overcurrent error, short-circuit error, or current sum error) occurred while the value for current is being measured, the ongoing measurement is stopped. After the type of error is displayed, the top menu appears again.

```
<start>
58496
63232
59392
62336
OVER CURRENT ERROR
<end>

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [High]
[3] Filter Settings [0]

[8] Log Output
[9] Exit

>
```

Figure 5.13 Stop Scanning on Error Detection

(5) Log Output

Selecting [8] from the top menu outputs the latest results of scanning, i.e., latest values for current. 10 logged values are displayed per line.

```
> 8[Enter]

Log Output
-----------------------------------------------------------------
[1] 35686 35686 32791 32779 32779 32752 32752 32661 32721 32721
[11] 32727 32727 32738 32733 32733 32736 32736 32729 32731 32731
[21] 32726 32726 32716 32723

Delta Sigma Interface test program

[1] Start Measurement
[2] Measurement Frequency [Low]
[3] Filter Settings [5]

[8] Log Output
[9] Exit

>
```

Figure 5.14 Log Output

(6) Ending the Sample Program

Selecting [9] from the top menu stops operation of the sample program and exits the menu.

6. Appendix

6.1 Confirmed Operation Environment

This section describes confirmed operation environment for the DSMIF FIT module.

Table 6.1 Confirmed Operation Environment (Rev.1.01)

Item	Contents
Integrated development environment	Renesas Electronics e2 studio Version 7.5.0 IAR Embedded Workbench for Renesas RX 4.12.1
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V3.01.00 Compiler option: The following option is added to the default settings of the integrated development environment. -lang=C99
	GCC for Renesas RX 4.8.4.201803 Compiler option: The following option is added to the default settings of the integrated development environment. -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.12.1 Compiler option: The default settings of the integrated development environment.
Endian	Big endian/little endian
Revison of the module	Rev.1.01
Board used	Renesas Starter Kit+ for RX72M (product number: RTK5572MNDCxxxxxBJ) and Photo coupler (SDIP8) Evaluation Sub Board

Table 6.2 Confirmed Operation Environment (Rev.1.00)

Item	Contents
Integrated development environment	Renesas Electronics e2 studio Version 7.5.0
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V3.01.00 Compiler option: The following option is added to the default settings of the integrated development environment. -lang=C99
	GCC for Renesas RX 4.8.4.201803 Compiler option: The following option is added to the default settings of the integrated development environment. -std=gnu99
Endian	Big endian/little endian
Revison of the module	Rev.1.00
Board used	Renesas Starter Kit+ for RX72M (product number: RTK5572MNDCxxxxxBJ) and Photo coupler (SDIP8) Evaluation Sub Board

7. Reference Documents

User’s Manual: Hardware

RX72M Group User’s Manual: Hardware (Doc No. R01UH0804)
PS9352AL2 Optically Isolated Delta-Sigma Modulator Data Sheet (Doc No. R08DS0129)
(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest information can be downloaded from the Renesas Electronics website.)

User’s Manual: Development Tools

RX Family C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package (R20UT0570)
(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug.02.19	—	First edition issued
1.01	Sep.17.19	—	Supported the following compilers: - IAR C/C++ Compiler for Renesas RX

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.