RX23E-B Group

RSSKRX23E-B Board Control Program

Introduction

Renesas Solution Starter Kit for RX23E-B (hereafter, referred to as RSSKRX23E-B) is a solution kit to make it possible to easily evaluate Analog Front End (hereafter, referred to as AFE), 24 bit Δ-ΣA/D converter (hereafter, referred to as DSAD), and 16 bit D/A converter (hereafter, referred to as R16DA) equipped in 32 bit MCU with high performance and high efficiency, RX23E-B.

This program operates at MCU RX23E-B on the RSSKRX23E-B board, communicates with QE for AFE via USB-UART, and provides functions to set registers of AFE, DSAD, and R16A, start and stop A/D conversion, and acquire and transmit the A/D conversion value based on the command from QE for AFE.

With this program, users can evaluate the AFE, DSAD, and R16DA without writing their own programs.

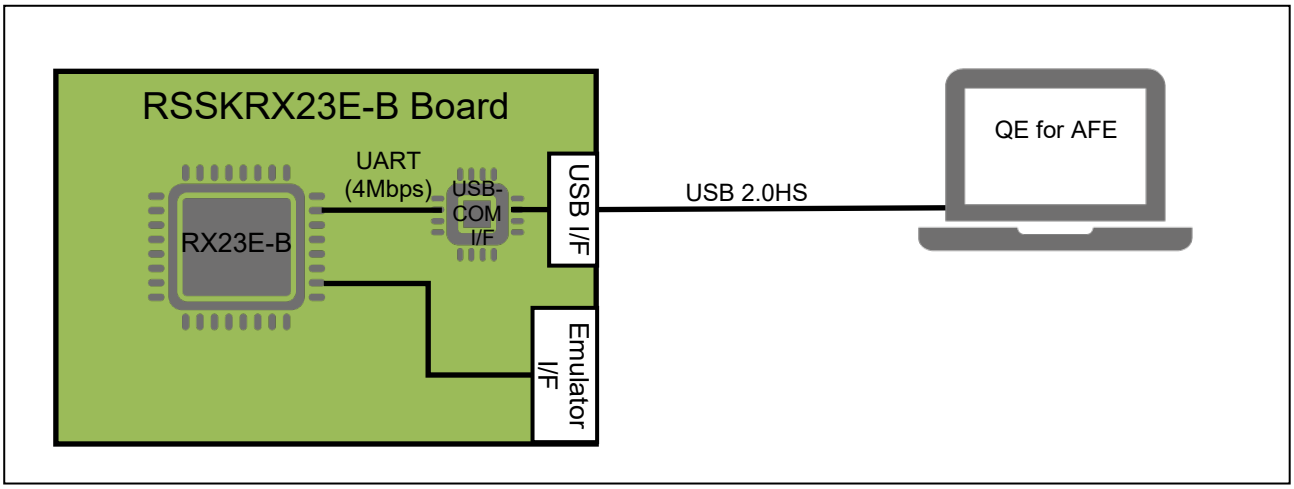

Figure for System connection

Target Device

RX23E-B (R5F523E6LDFP)

Contents

1. Overview

This program operates at MCU RX23E-B on the RSSKRX23E-B board, communicates with QE for AFE in UART via USB, and provides functions to set registers of AFE, DSAD, and R16DA, start and stop A/D conversion, and Tabacquire and transmit the A/D conversion value based on the command from QE for AFE. Figure 1-1 shows the data flow related to QE for AFE communication.

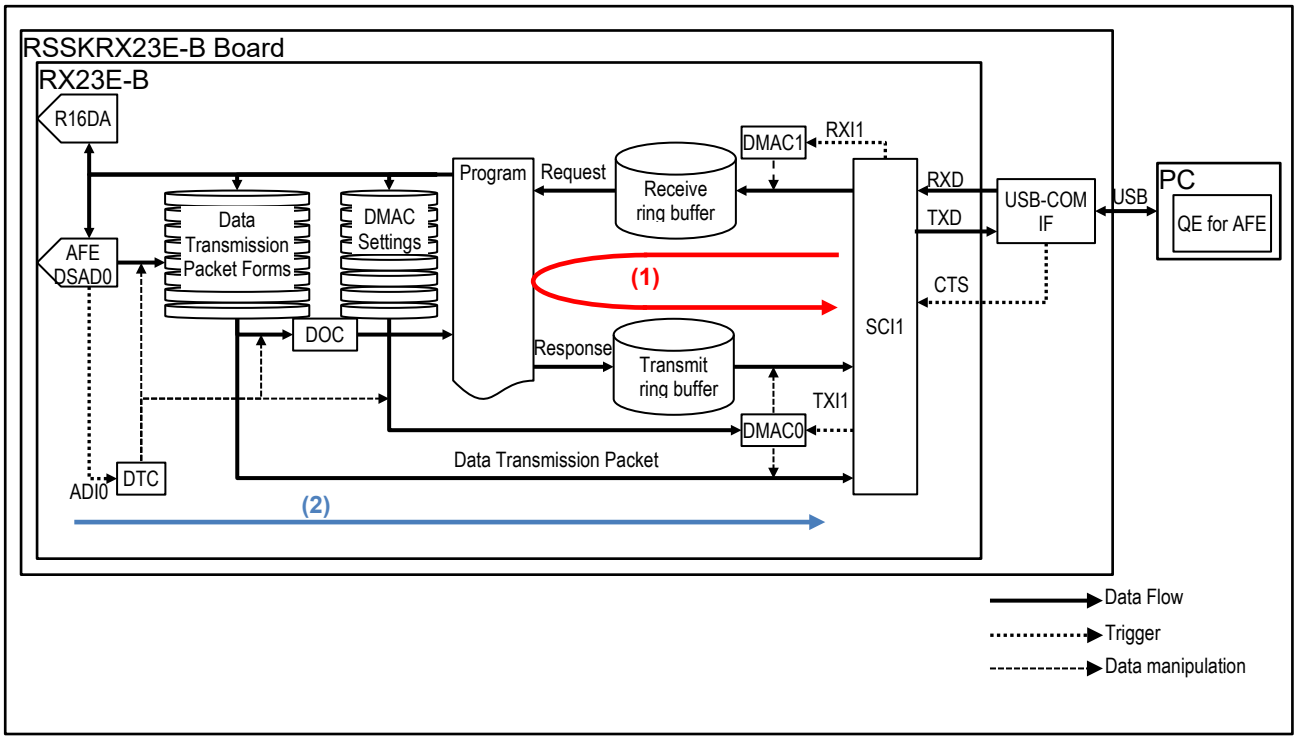

Figure 1-1 This Program and Data Flow

The following describes the data flow (1) and (2) in Figure 1-1.

(1) Command and response from/to QE for AFE

The data which SCI1 received from QE for AFE are transferred to the receive ring buffer from SCI1 by DMAC1. The program acquires and analyzes the data stored in the receive ring buffer in units of packets. According to the command in the packet, the program sets registers of AFE, DSAD, and R16DA, starts and stops A/D conversion, stores the response packet in the transmit ring buffer, and transmits it to QE for AFE from SCI1 by using DMAC0.

(2) Transmission of DSAD0 conversion value

With completion of A/D conversion (ADI0) as a trigger, DTC stores the A/D conversion result of DSAD0 in the A/D conversion value transmission packet, and DMAC0 is set to transfer the A/D conversion value transmission packet. DMAC0 transmits the A/D conversion value transmission packet from SCI1 to QE for AFE. Additionally, DTC sets the A/D conversion result in the data operation circuit (DOC) in order to extract the A/D conversion error information from the acquired A/D conversion result.

2. Operating Conditions

Table 2-1 shows the operating conditions.

Table 2-1 Operating Conditions

Item	Description
Board	RSSKRX23E-B Board (RTK0ES1001C00001BJ)
MCU	RX23E-B (R5F523E6BDFM) Power voltage (VCC, AVCC0): 5V Operating frequency (ICLK): 32MHz Peripheral operating frequency (PCLKB, PCLKC): 32MHz DSAD0 operating frequency (f_{OP}): 16MHz DSAD0 modulator clock frequency (f_{MOD}): 4MHz
IDE	Renesas e^2 studio Version 2023-04 Renesas RX Smart Configurator V23.4.0
Tool Chain	Renesas CC-RX V3.05.00
Emulator	E2 Emulator Lite
PC Program	QE for AFE V2.1.1

3. Package Contents

The package contents are shown in Table 3-1.

Table 3-1 Package Contents

File/folder name	Description
r01an6364jj0100-rx23e-b.pdf	This document (Japanese)
r01an6364ej0100-rx23e-b.pdf	This document (English)
rx23eb_rssk_fw.mot	Firmware binary file written at the board shipment
rx23eb_qe	Communication module source
rx23eb_rssk_fw	This sample project
readme_j.txt	Package explanation (Japanese)
readme_e.txt	Package explanation (English)

4. Related Documents

- R01UH0972 RX23E-B Group User's Manual: Hardware
- R20UZ0108 RSSKRX23E-B User's Manual

5. QE for AFE Communication Specification

This chapter describes specifications of communication with QE for AFE.

5.1 Serial Communication Setting

Table 5-1 Serial Communication setting

Item	Setting
Method	Asynchronous mode (UART)
Transfer rate	Up to 4Mbps
Data length	8 bits
Parity	None
Stop bit	1

5.2 Sequence

Figure 5-1 shows the communication sequence.

In the normal communication sequence, the RX23E-B program returns a response for a command from QE for AFE. When the command is supported by the program, the response is ACK. When the command is not supported, the response is NACK.

QE for AFE sends a Negotiation command at first, then receives the functions that the RX23E-B program supports. Following this, QE for AFE sends only commands that the program supports.

Acquisition of A/D conversion values and measurement values is performed in a different sequence. When QE for AFE sends a Run command, the RX23E-B program continues transmitting data packets of A/D conversion values or measurement values until it receives a Stop command.

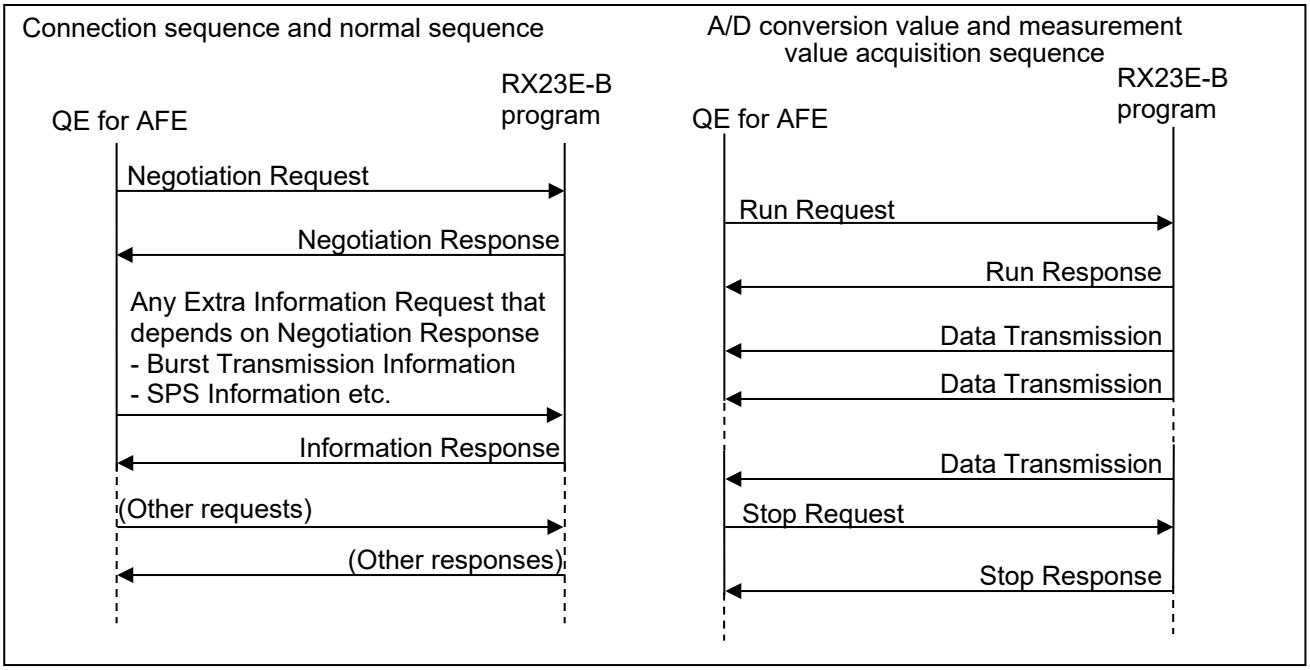

Figure 5-1 Communication Sequence

5.3 Endian

The order of communicating multibyte data is based on the MCU endian setting. Table 5-2 shows the order of communicating multibyte data for each endian setting.

Table 5-2 Order of multibyte data

Data Size	Endianness	Data			
		Byte0	Byte1	Byte2	Byte3
4bytes	Value	0x654321FF			
	Little	0xFF	0x21	0x43	0x65
	Big	0x65	0x43	0x21	0xFF
3bytes	Value	0x4321FF			
	Little	0xFF	0x21	0x43	-
	Big	0x43	0x21	0xFF	-
2bytes	Value	0x21FF			
	Little	0xFF	0x21	-	-
	Big	0x21	0xFF	-	-

Note: QE for AFE V2.1.1 supports only little endian.

5.4 Packet Structure

There are two packet structures, normal and extended packets. Table 5-3 shows the packet structure, Table 5-4 shows the header bit constitution, and Table 5-5 lists the commands in the header.

Table 5-3 Packet Structure

Offset [Byte]		Contents	Description
Normal	Extended		
0		Header	Packet header, see Table 5-4
+1		Data length	Attached data byte counts H'00 – H'FE If it exceeds H'FE, set H'FF and specify the data length (number of bytes) in the following 4 bytes.
-	+2	Extended data length	Data length if the above data length exceeds H'FE.
+2	+6	Data	Data. Defined for each command

Table 5-4 Header Constitution

bit	Name	Description
b7-b6	Identifier	B'10
b5	Type	0: Request, 1: Response
b4	ACK/NACK	Request Packet: 0 Response Packet 0: ACK 1: NACK
b3-b0	Command	See Table 5-5

Table 5-5 Command List

Value	Command	Description
H'0	Negotiation	Acquire supported functions
H'1	Read	Read registers
H'2	Write	Write registers
H'3	Run	Start data acquisition
H'4	Stop	Stop data acquisition
H'5-H'7	-	Reserved
H'8	User Value Setting	Set user setting value
H'9	Data Transmission	Request data transmission
H'A	Extra Information	Acquire extra information
H'B-H'F	-	Reserved

5.5 Command Details

5.5.1 Negotiation

QE for AFE sends this command to request supported functions, and the RX23E-B program returns them. Table 5-6 shows the packet structure, and Table 5-7 shows information on the supported functions.

Table 5-6 Negotiation Command Packet Structure

Type	Header	Data length	Data
Request	H'80	H'00	-
Response	H'A0	H'07	Information on supported functions (see Table 5-7)

Table 5-7 Information on Supported Functions

Offset	Supported function	bit	Name	Description
+0	Function 1	b7	Endianness	Endian Note1 0: Little 1: Big
		b6-b3	-	Reserved (0)
		b2	Read	Register reading support 0: Not supported 1: Supported
		b1	Write	Register writing support 0: Not supported 1: Supported
		b0	-	Reserved (0)
+1	Function 2	b7	User Value 7	User value Note2 usage 0: Disable 1: Enable
		b6	User Value 6	
		b5	User Value 5	
		b4	User Value 4	
		b3	User Value 3	
		b2	User Value 2	
		b1	User Value 1	
		b0	User Value 0	
+2	Function 3	b7-b6	Data Ch0	Transmission data type Note3 B'00: None B'10: DSAD A/D conversion value (integer) B'11: Measurement value (float)
		b5-b4	Data Ch1	
		b3-b2	Data Ch2	
		b1-b0	Data Ch3	
+3	Function 4	b7-b6	Data Ch4	
		b5-b4	Data Ch5	
		b3-b2	Data Ch6	
		b1-b0	Data Ch7	
+4	Function 5	b7-b6	Data Ch8	
		b5-b4	Data Ch9	
		b3-b2	Data Ch10	
		b1-b0	Data Ch11	
+5	Function 6	b7-b6	Data Ch12	
		b5-b4	Data Ch13	
		b3-b2	Data Ch14	
		b1-b0	Data Ch15	
+6	Function 7	b7	User button 7	User button Note4 usage 0: Not used 1: Used
		b6	User button 6	
		b5	User button 5	
		b4	User button 4	
		b3	User button 3	
		b2	User button 2	
		b1	User button 1	
		b0	User button 0	

Notes: 1. QE for AFE V2.1.1 supports Little Endian only.
2. User Values on QE for AFE are displayed as 1 to 8.
3. Mixture of different data types is not supported except for None (B'00).
4. User Buttons on QE for AFE are displayed as 1 to 8.

5.5.2 Read

Reads the value at the address in the MCU that QE for AFE specifies. QE for AFE specifies the start address and the number of data to read. The RX23E-B program returns the value of the specified number of data from the specified start address. The unit of reading is 4 bytes.

Table 5-8 shows the packet structure of Read command.

Table 5-8 Read Command Packet Structure

Type	Header	Data length	Data			
Request	H'81	H'05	Start Address (4byte)	Number to read: N (0 < N < 32) (1byte)		
ACK Response	H'A1	4+4N (0 < N < 32)	Start Address (4byte)	Register 1 (4byte)	…	Register N (4byte)
NACK Response	H'B1	H'00	-			

5.5.3 Write

Writes the value to the address specified by QE for AFE. QE for AFE specifies the start address and the value to write in 4-byte unit. The RX23E-B program returns the result of writing.

Table 5-9 shows the packet structure of Write command.

Table 5-9 Write Command Packet Structure

Type	Header	Data length	Data			
Request	H'82	4+4N (0 < N < 32)	Start Address (4byte)	Register 1 (4byte)	…	Register N (4byte)
ACK Response	H'A2					
NACK Response	H'B2	H'00	-			

5.5.4 Run

QE for AFE requests to start measurement, and the RX23E-B program starts measurement processing and returns a response packet. If the data transmission Option is enabled, the RX23E-B program sends the measurement results in data transmission packets.

Table 5-10 and Table 5-11 show the packet structure of Run command.

Table 5-10 Run Command Packet Structure

Type	Header	Data length	Data
Request	H'83	H'01	Option (1Byte)
ACK Response	H'A3		
NACK Response	H'B3		

Table 5-11 Run Command Option

Name	bit	Description
Option information	b7-b1	reserved (0)
	b0	Data transmission 0: Enable 1: Disable

5.5.5 Stop

QE for AFE requests to stop measurement, and the RX23E-B program stops the measurement processing and returns a response packet. If data transmission is enabled with Run command, the program stops sending data transmission packets.

Table 5-12 shows the packet structure of Stop command.

Table 5-12 Stop Command Packet Structure

Type	Header	Data length	Data
Request	H'84	H'00	-
ACK Response	H'A4		
NACK Response	H'B4		

5.5.6 Data Transmission

The RX23E-B program sends A/D conversion values or measurement values of the available channels in Negotiation response. Table 5-13 shows the packet structure, and Table 5-15 shows the data structure.

When the transmission data type is A/D conversion value, the program can support the extended packet structure shown in Table 5-14. In the extended packet structure, the data size of the A/D conversion value to send is specified with Data Transmission information in Extra Information described later. If the user required data length exceeds 254 (H'FE), specify it H'FF and data length which is up to 4 bytes. If the user cannot specify it, set it to H'01.

Table 5-13 Data Transmission Packet Structure

Header	Data length	Data				
H'A9	1+4N (1byte)	Channel (1byte)	Data 1 (4bytes)	Data 2 (4bytes)	…	Data N (4bytes)

Table 5-14 Data Transmission Extended Packet Structure for A/D Conversion Value Transmission

Format	Header	Data length		Data				
1	H'A9	1+N×data size (1byte)		Channel (1byte)	Data1	Data 2	…	Data N
2		H'01 (data length unfixed) (1byte)						
3		H'FF (1byte)	1+N×data size (4byte)					

Table 5-15 Data Structure

Name	bit	Description	
		DSAD A/D conversion value	**Measurement value**
Channel	b7-b5	DSAD No. 0: DSAD0 Others: reserved	0
	b4	Packet structure information 0: Data Transmission packet structure 1: Data Transmission extended packet structure	
	b3-b0	DSAD channel No.: 0 - 7	Channel No.: 0 - 15
data1…dataN	-	A/D conversion value	Measurement value

5.5.7 User Value Setting

Sets the user setting value that is enabled in Negotiation response in single precision floating point format.

Table 5-16 User Value Setting Command Packet Structure

Type	Header	Data length	Data	
			User value No.	Value
Request	H'88	H'05	0 - 7 (1byte)	User setting value (float format) (4bytes)
ACK Response	H'A8			
NACK Response	H'B8	H'01		-

5.5.8 Extra Information

Acquires extra information that the program can provide. The type of extra information is defined as Class. Table 5-17 shows the packet structure, and Table 5-18 lists each Class of extra information.

Table 5-17 Extra Information Command Packet Structure

Type	Header	Data length	Data	
Request	H'8A	H'02	Class (2byte)	-
ACK Response	H'AA	(1Byte)	Class (2Byte)	Class information
NACK Response	H'BA	H'02	Class (2Byte)	-
		H'00[Note]	-	

Note: If no extra information is supported, the data length of NACK response is 0.

Table 5-18 Class List

Class		Description
Value	Name	
H'0001	SPS Information	Returns transmission rate of measurement value per channel.
H'0002	Data Transmission Information	If transmission data type for the channel is A/D conversion value in Negotiation response, returns data size of the extended packet structure as byte counts.
H'0003	DSAD Error Information	If A/D transmission data size is 3 bytes, returns DSAD error information.
H'000E	Part Number Information	Returns device model name in ASCII character string.
H'000F	Text Information	Returns RX23E-B program information in ASCII character string.
H'0010 ~H'0017	User Button Status	Returns the status of user button operation.
Others	-	Reserved

5.5.8.1 SPS Information

When the transmission data type is measurement value in Negotiation response, acquires the transmission rate of measurement value for each channel. The data format of the transmission rate is single precision floating point format, and the data disabled channel are represented by NaN.

Table 5-19 SPS Information Packet Structure

Type	Header	Data length	Data				
			Class	Transmission rate [SPS]			
Request	H'8A	H'02	H'0001	-			
ACK Response	H'AA	H'42		ch.0 (4byte)	ch.1 (4byte)	…	ch.15 (4byte)

5.5.8.2 Data Transmission Information

When the transmission data type is A/D conversion value in Negotiation response, acquires the data size of each channel to be transmitted in data transmission extended packet structure as byte counts.

Table 5-20 Data Transmission Information Packet Structure

Type	Header	Data length	Data				
			Class	Data transmission extended packet structure data size			
Request	H'8A	H'02	H'0002	-			
ACK Response	H'AA	2+N		ch.0 (1byte)	ch.1 (1byte)	…	ch.(N-1) (1byte)

Table 5-21 Data Size of Data Transmission Extended Packet Structure

Name	bit	Description
ch.n	b7-b4	Reserved (0)
	b3-b0	Byte counts of A/D conversion value

5.5.8.3 DSAD Error Information

When the transmission data type in Navigation response is A/D conversion value and the transmission data size is 3 bytes, extracts the information on DSAD conversion errors that occur during A/D conversion after A/D conversion stops.

Table 5-22 DSAD Error Information Packet Structure

Type	Header	Data length	Data				
			Class	DSAD conversion error information			
Request	H'8A	H'02	H'0003	-			
ACK Response	H'AA	2+N		ch.0 (1byte)	ch.1 (1byte)	…	ch.(N-1) (1byte)

Table 5-23 DSAD Conversion Error Information

Name	bit	Description
ch.n	b7-b4	Reserved (0)
	b3-b0	DSAD conversion error cumulative information for each channel

5.5.8.4 Part Number Information

Acquires the device model name of RX23E-B in ASCII character string.

Table 5-24 Part Number Information Packet Structure

Type	Header	Data length	Data	
			Class	Device model name
Request	H'8A	H'02	H'000E	-
ACK Response	H'AA	H'0E		ASCII character string

5.5.8.5 Text Information

Acquires the information defined by the RX23E-B program in ASCII character string. The maximum length of ASCII character string is 253 bytes.

Table 5-25 Text Information Packet Structure

Type	Header	Data length	Data	
			Class	Class information
Request	H'8A	H'02	H'000F	-
ACK Response	H'AA	2+ASCII character string length		ASCII character string

5.5.8.6 User Button Status

Sets and acquires the status of User Button. The lower 3 bits of the Class information indicate the Button number.

Table 5-26 User Button Status Packet Structure

Type	Header	Data length	Data	
			Class	Class information
Request	H'8A	H'03	H'0010~ H'0017	User Button status (1byte)
ACK Response	H'AA			

Table 5-27 User Button Status

Name	bit	Description
-	b7-b4	Reserved (0)
User Button Status	b3-b0	0: off 1: on, QE for AFE normally sends "on".

6. QE for AFE Communication Module

QE for AFE communication module (hereafter, referred to as communication module) performs communication processing based on “5. QE for AFE Communication Specification”. The communication module uses one SCI channel and two DMAC channels to transmit/receive packets. It is also possible to detect transmission timeout by using one CMT.

The communication module is set with r_qe_cfg.h, and peripheral functions are set with Smart Configurator.

User data definitions for user program extension are set with st_qe_api_t structure in r_qe_cfg_typedef.h. User defined processes such as judgement of command acceptability and reading/writing user data described above are performed by user functions in r_qe_api_user.c corresponding to each command.

6.1 Overview

Figure 6-1 shows an example of using communication module API functions and the relation to peripheral functions.

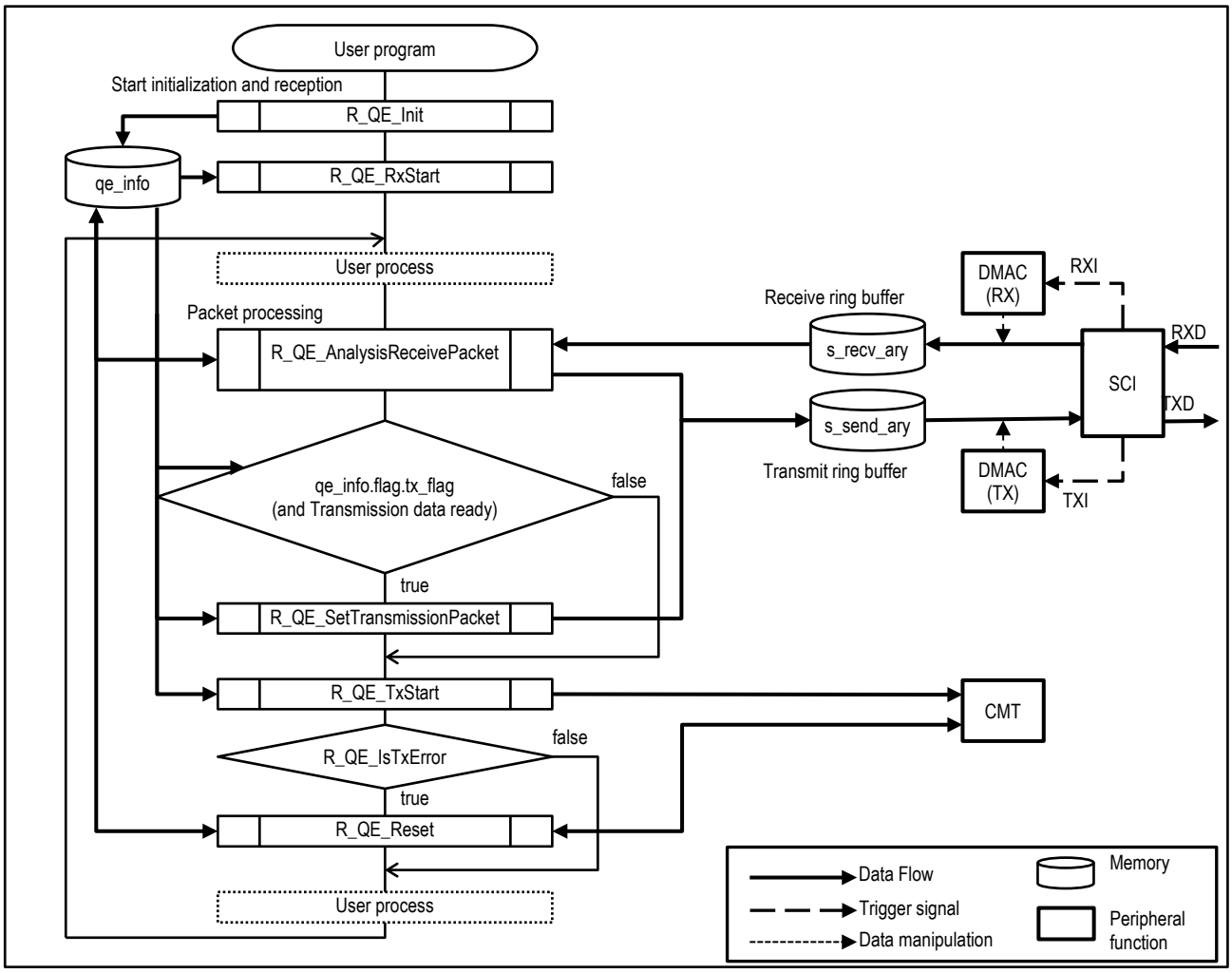

Figure 6-1 Example of Communication Module Usage and Peripheral Functions

Communication with QE for AFE is performed in the asynchronous mode , and the received data are transferred from SCI to the receive ring buffer, and the transmission data are transferred from the transmit ring buffer to SCI by each DMAC respectively. Each ring buffer is secured by the communication module.

It is possible to check the communication module status and exchange data with user programs via st_qe_api_t type structure variables by adding user defined members to st_qe_api_t structure variables.

The following describes an example of user program that uses communication module API functions shown in Figure 6-1.

Initialization and reception start

Starts initialization of the communication module with R_QE_Init function and starts reception with R_QE_RxStart function.

Packet processing

Received packet processing

Acquires the received packet stored in the receive ring buffer with R_QE_AnalysisReceivePacket function, processes it corresponding to the command, and stores the response packet in the transmit ring buffer.

Transmission data packet processing

If the variable flag.tx_flag, member of the st_qe_api_t structure is set, and there are transmission data, creates a data transmission packet and stores it in the transmit ring buffer with R_QE_SetTransmissionPacket function.

Transmission processing

When a transmission packet is stored in the transmit ring buffer, performs transmission start settings with R_QE_TxStart function. If transmission timeout is used, resets and starts CMT at transmission start settings, and stops CMT at the detection of transmission completion.

Transmission error processing

Detects transmission errors including transmission timeout with R_QE_IsTxError function, and resets the transmission processing with R_QE_Reset function if a transmission error occurs.

With R_QE_AnalysisReceivePacket function, executes user functions corresponding to each command and creates an ACK or NACK response packet according to the return value of each user function. The user functions judge whether the command is acceptable or unacceptable and process it. Figure 6-2 shows a flowchart of R_QE_AnalysisReceivePacket function.

For user functions corresponding to each command, see Table 6-13.

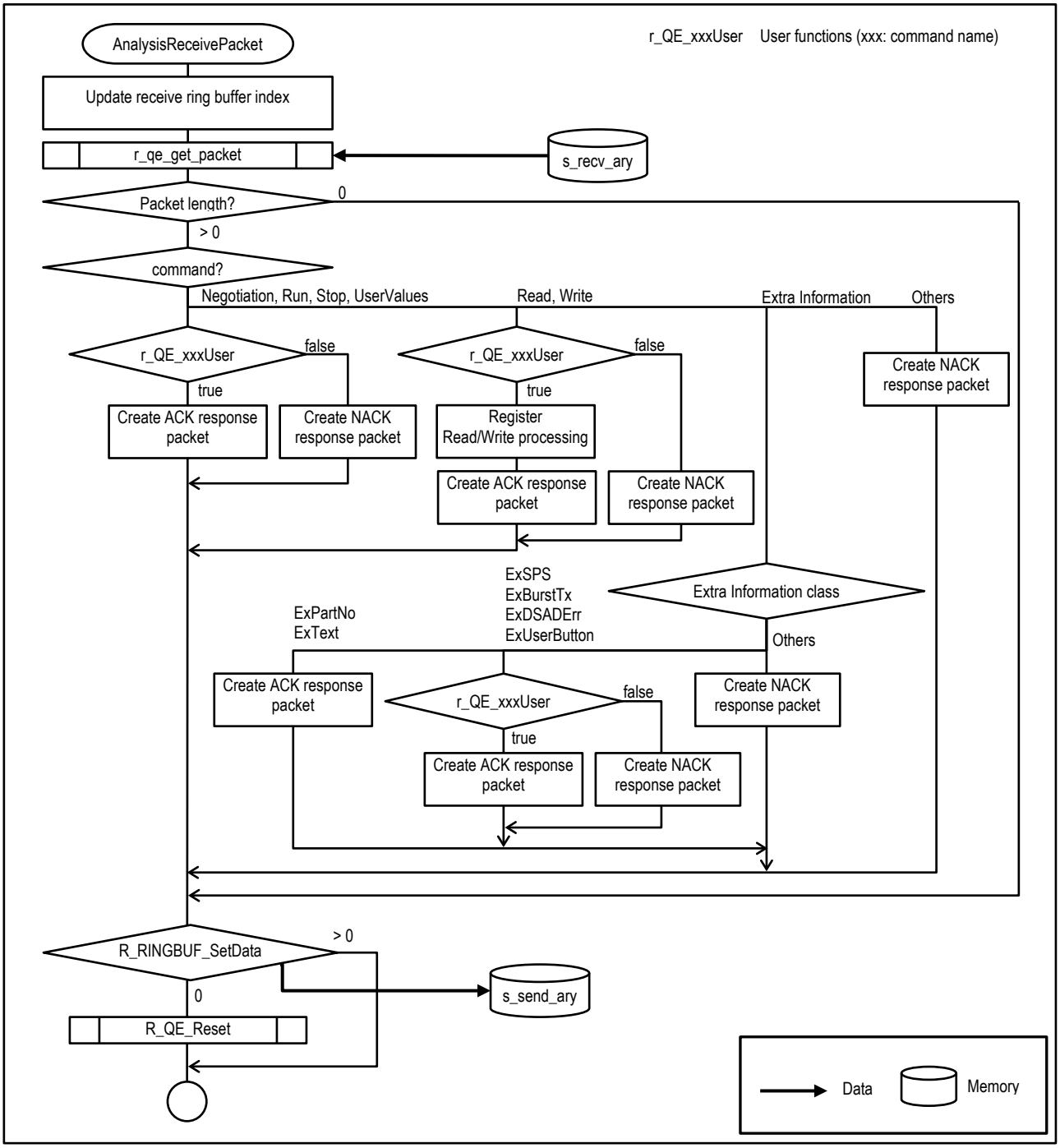

Figure 6-2 R_QE_AnalysisReceivePacket Function Flowchart

6.2 Peripheral Functions Used

The communication module uses one SCI channel and two DMAC channels to transmit/receive packets. It is also possible to detect transmission timeout by using one CMT. The channel selections for peripheral functions to use is specified with r_qe_cfg.h. For the details, see Table 6-5.

Operation settings for peripheral functions are performed by Smart Configurator. The followings show Smart Configurator settings which the communication module is based on.

Table 6-1 SCI Settings

Asynchronous mode
Operation mode: Transmission/reception

Item		Setting
Start bit edge detection setting		Falling edge on RXD1 pin
Data length setting		8 bits
Parity setting		None
Stop bit length setting		1 bit
Transfer direction setting		LSB-first
Transfer rate setting	Transfer clock	(Determined on user program conditions)
	Bit rate	
	Enable modulation duty correction	
	SCK1 pin function	
Noise filter setting		
Hardware flow control setting		
Data handling setting	Transfer data handling	Data handled by DMAC
	Receive data handling	Data handled by DMAC
Interrupt setting	Enable reception error interrupt (ERI1)	Not used
	TXI1, RXI1, TEI1, ERI1 priority	Level 0 (Determined on user program conditions)
Callback function setting		Not used

Table 6-2 DMAC Settings

Item		Setting	
		For Reception	**For Transmission**
Transfer setting	Activation source	SCIx (RXIx)	SCIx (TXIx)
	Activation source flag control	Clear interrupt flag of the activation source	
	Transfer mode	Free running mode	Free running mode
	Transfer data size	8 bits	
	Transfer count / Repeat size / Block size	-	(Setting on execution)
Source address setting	Source address	Address of SCIx.RXDx Fixed	(Setting on execution) Incremented
	Specify the transfer source as extended repeat area	-	Enable
	Extended repeat area		Adjust to module settings Note
Destination address setting	Destination address	(Set by the program) Incremented	Address of SCIx.TXDx Fixed
	Specify the transfer destination as extended repeat area	Enable	-
	Extended repeat area	Adjust to module settings Note	
Interrupt setting		Not used	

Note: Adjust to the setting value of D_QE_CFG_TX_RINGBUF_SIZE of r_qe_cfg.h shown in Table 6-5.

Table 6-3 CMT Settings

PCLKB: 32MHz

Item		Setting
Count clock setting		PCLKB/512
Compare match setting	Interval value	1000ms
	Compare match interrupt (CMIx)	Enable Priority: Level 0 (disabled)

6.3 Communication Module Composition

6.3.1 File Composition

Table 6-4 shows the file configuration of the module. Users can customize the files that are indicated as “User extension: Enable” in this table.

Table 6-4 File Composition

File Name	Description	User extension
r_qe_api_user.c	Communication module user functions	Enable
r_qe_api.c	Communication module API	Disable
r_qe_api.h		
r_qe_cfg_thpedef.h	Communication module information structure st_qe_api_t definition	Enable
r_qe_cfg.h	Communication module settings	Disable
r_qe_packet.h	Packet structure definition	
r_qe_sc_if.h	Function definitions generated by Smart Configurator	
r_ring_buffer_control_api.c	Ring buffer module	
r_ring_buffer_control_api.h	Ring buffer module header file	

6.3.2 Macro Definitions

The following macros are used to perform settings of the communication module according to the user system.

Table 6-5 r_qe_cfg.h Definitions (1/2)

Definition name	Description
D_QE_CFG_TX_RINGBUF_SIZE	Byte counts of transmit ring buffer, a power of two
D_QE_CFG_RX_RINGBUF_SIZE	Byte counts of receive ring buffer, a power of two
D_QE_CFG_FORMAT_REV	Communication specification revision. Specify 3.
D_QE_CFG_READ	Read command support 0: Not supported 1: Supported
D_QE_CFG_WRITE	Write command support 0: Not supported 1: Supported
D_QE_CFG_USER_VAL0	User Value 0: Not used 1: Used
D_QE_CFG_USER_VAL1	
D_QE_CFG_USER_VAL2	
D_QE_CFG_USER_VAL3	
D_QE_CFG_USER_VAL4	
D_QE_CFG_USER_VAL5	
D_QE_CFG_USER_VAL6	
D_QE_CFG_USER_VAL7	
D_QE_CFG_EX_SPS	SPS information 0: Not used 1: Used
D_QE_CFG_EX_USER_BTN0	User Button 0: Not used 1: Used
D_QE_CFG_EX_USER_BTN1	
D_QE_CFG_EX_USER_BTN2	
D_QE_CFG_EX_USER_BTN3	
D_QE_CFG_EX_USER_BTN4	
D_QE_CFG_EX_USER_BTN5	
D_QE_CFG_EX_USER_BTN6	
D_QE_CFG_EX_USER_BTN7	
D_QE_CFG_CH0	Transmission data type Note 0x0: Not transmitted 0x2: A/D conversion value (integer) 0x3: Measurement value (float)
D_QE_CFG_CH1	
D_QE_CFG_CH2	
D_QE_CFG_CH3	
D_QE_CFG_CH4	
D_QE_CFG_CH5	
D_QE_CFG_CH6	
D_QE_CFG_CH7	
D_QE_CFG_CH8	
D_QE_CFG_CH9	
D_QE_CFG_CH10	
D_QE_CFG_CH11	
D_QE_CFG_CH12	
D_QE_CFG_CH13	
D_QE_CFG_CH14	
D_QE_CFG_CH15	

Note: Mixture of different data types is not available except for 0x0.

Table 6-6 r_qe_cfg.h Definitions (2/2)

Definition name	Description
D_QE_CFG_TXT_INFO	Text of extra information, ASCII character string of 253 bytes or lower
D_QE_CFG_TXERRCHK_EN	Transmission error check 0: Not checked 1: Checked
D_QE_CFG_TIMEOUT	Transmission error detection method 0: Timmer (CMT) 1: Detection of CTS signal
D_QE_CFG_SCI	SCI channel number
D_QE_CFG_DMAC_RX	DMAC channel number for reception
D_QE_CFG_DMAC_TX	DMAC channel number for transmission
D_QE_CFG_CMT	CMT channel number

6.3.3 Structures and Unions

Table 6-7 r_qe_cfg_typedef.h

Structure type name	st_qe_api_t		
Description	Communication module information		
Member	**Type**	**Name**	**Description**
	union	flag	Flag for API operation control
	uint8_t	byte	Entire flag
	struct	bit	Access in bit units
	uint8_t:1	tx_flag	Data transmission enable flag
	uint8_t	dsad_err[D_QE_DSAD_CH_NUM]	DSAD conversion error information storage array for DSAD error information
	uint8_t	burst_tx[D_QE_DSAD_CH_NUM]	A/D conversion value data size storage array for burst transmission information
	float	sps[16]	Transmission rate storage array for SPS information
	union	button	Button status
	uint8_t	buttons	Entire status
	struct	bit	Access in bit units
	int8_t:1	b0	Button 0 status
	int8_t:1	b1	Button 1 status
	int8_t:1	b2	Button 2 status
	int8_t:1	b3	Button 3 status
	int8_t:1	b4	Button 4 status
	int8_t:1	b5	Button 5 status
	int8_t:1	b6	Button 6 status
	int8_t:1	b7	Button 7 status

Note: D_QE_DSAD_CH_NUM is defined according to the module settings of r_qe_cfg.h.

Table 6-8 r_ring_buffer_control_api.h

Structure type name	st_ring_buf_t		
Description	Ring buffer management		
Member	**Type**	**Name**	**Description**
	uint8_t *	buf	Pointer to buffer
	size_t	length	Buffer length
	uint32_t	r_index	Read index
	uint32_t	w_index	Write index

Table 6-9 r_qe_packet.h (1/2)

Structure type name	st_qe_userval_t		
Description	User Value Setting command packet data		
Member	**Type**	**Name**	**Description**
	uint8_t	no	User variable number
	float	fval	User variable
Union type name	u_data_t		
Description	Packet data part		
Member	**Type**	**Name**	**Description**
	uint8_t	buf[255]	Entire data array
	struct	r_req	Read request data
	uint32_t *	p_start_addr	Start address
	uint8_t	num	Number to read
	struct	w_req	Write request data
	uint32_t *	p_start_addr	Start address
	uint32_t	w_data[32]	Data array to write
	struct	rw_rsp	Read/Write response data
	uint32_t *	p_start_addr	Start address
	uint32_t	reg[32]	Data array to read
	struct	tx_mes	Measurement value transmission data
	uint8_t	ch	Channel
	float	fdata[32]	Data part of data transmission packet for measurement value
	struct	tx_dsad	A/D conversion value transmission data
	uint8_t	ch	Channel
	uint32_t	data[32]	Data part of data transmission packet for A/D conversion value
	st_qe_userval_t	user_val	User Value Setting data
	struct	ex	Extra Information data
	uint16_t	cls	Class
	union	info	Each class data of extra information
	uint8_t	data[253]	Entire data of each class response
	float	sps[16]	Transmission rate array of SPS information
	uint8_t	burst_tx[16]	Data size array for burst transmission information
	uint8_t	dsad_err[16]	DSAD error information array for DSAD error information
	char	text[253]	Text information in ASCII character string

Table 6-10 r_qe_packet.h (2/2)

Structure type name	st_qe_pkt_t		
Description	Packet		
Member	**Type**	**Name**	**Description**
	union	-	Packet structure
	uint8_t	buf[261]	Entire packet array
	struct	pkt	Packet element
	uint8_t	header	Header
	uint8_t	data_len	Data length
	union	-	Data part
	u_data_t	data	Data
	struct	-	Extended format
	union	-	Extended data length
	uint32_t	len	4-byte access
	uint8_t	byte[4]	1-byte access
	u_data_t	data	Data

6.3.4 Function List

Table 6-11 r_qe_api.h QE for AFE Communication Module API Functions (1/2)

Function name	**R_QE_Init**			
Description	Initializes the communication module.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
Return value	-	void	-	
Function name	**R_QE_Reset**			
Description	Ends transmission and resets.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
Return value	-	void	-	
Function name	**R_QE_RxStart**			
Description	Starts reception.			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	**void**	**-**	**-**
Return value	-	void	-	
Function name	**R_QE_TxStart**			
Description	If there is an untransmitted packet in the transmit ring buffer, starts transmission. If detection of transmission timeout is enabled, starts CMT counting.			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Start transmission false: Not transmit	
Function name	**R_QE_AnalysisReceivePacket**			
Description	Acquires a reception packet, executes the corresponding processing, creates a response packet, and stores it in the transmit ring buffer.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
Return value	O	size_t	Byte counts of the response packet stored in the transmit ring buffer	
Function name	**R_QE_IsTxError**			
Description	Checks communication errors caused during transmission.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	void	-	-
Return value	O	bool	true: Error occurs false: Normal	

Table 6-12 r_qe_api.h QE for AFE Communication Module API Functions (2/2)

Function name	**R_QE_SetTxData**			
Description	Stores data of the specified byte counts in the transmit ring buffer.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	data[]	Data array to store
	I	size_t	len	Byte counts
Return value	O	size_t	Byte counts stored in the transmit ring buffer	
Function name	**R_QE_SetExButtonStatusPacket**			
Description	Stores user button status packet of extra information in the transmit ring buffer.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	btn_no	Button number (0 to 7)
	I	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
Return value	O	size_t	Byte counts stored in the transmit ring buffer	
Function name	**R_QE_SetTransmissionPacket**			
Description	Creates a data transmission packet whose transmission data type is measurement value and stores it in the transmit ring buffer.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	uint8_t	channel	Channel
	I	float	fval[]	Transmission data storage array
	I	uint8_t	num	Number of transmission data
Return value	O	size_t	Byte counts of the packet stored in the transmit ring buffer	

Table 6-13 r_qe_api_user.c Functions (1/2)

Function name	**r_QE_NegotiationUser**			
Description	Judges whether a Negotiation command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	
Function name	**r_QE_ReadUser**			
Description	Judges whether a Read command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	
Function name	**r_QE_WriteUser**			
Description	Judges whether a Write command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	
Function name	**r_QE_RunUser**			
Description	Judges whether a Run command is acceptable or unacceptable, sets qe_info member flag.tx_flag, and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	
Function name	**r_QE_StopUser**			
Description	Judges whether a Stop command is acceptable or unacceptable, clears qe_info member flag.tx_flag, and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable.
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	

Table 6-14 r_qe_api_user.c Functions (2/2)

Function name	**r_QE_UserValueUser**			
Description	Judges whether a User Value Setting command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	

Function name	**r_QE_ExSpsInfoUser**			
Description	Judges whether SPS Information class of Extra Information command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	

Function name	**r_QE_ExBurstTxInfoUser**			
Description	Judges whether Burst Transmission Information class of Extra information command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	

Function name	**r_QE_ExDsadErrInfoUser**			
Description	Judged whether DSAD Error Information class of Extra Information command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	

Function name	**r_QE_ExUseButtonStatusUser**			
Description	Judges whether User Button Status class of Extra Information command is acceptable or unacceptable and performs user processing.			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_qe_api_t *	qe_info	Pointer to the communication module information structure variable
	I	st_qe_pkt_t *	rcv	Pointer to reception packet structure
Return value	O	bool	true: Acceptable false: Unacceptable	

Table 6-15 r_qe_sc_if.h Macro Functions Corresponding to Functions Generated by Smart Configurator

Peripheral function	Definition function	Corresponding function	Supplement
CMT	R_QE_CMT_CntStart	R_Config_CMTn_Start	n: CMT channel number
	R_QE_CMT_CntStop	R_Config_CMTn_Stop	
SCI	R_QE_SCI_Start	R_Config_SCIn_Start	n: SCI channel number
	R_QE_SCI_RxStart	R_Config_SCIn_Stop	
	R_QE_SCI_SendStart	R_Config_SCIn_Serial_Send	
DMAC	R_QE_DMAC_TX_Start	R_Config_DMACn_Start	n: DMAC channel number for transmission
	R_QE_DMAC_TX_Stop	R_Config_DMACn_Stop	
	R_QE_DMAC_RX_Start	R_Config_DMACn_Start	n: DMAC channel number for reception

Note: Each channel number is specified with r_qe_cfg.h. See Table 6-6.

Table 6-16 r_qe_sc_if.h User-Defined Functions

Function name	**R_QE_DMAC_RX_SetDstAddr**			
Description	Sets transfer destination address to DMAC for reception. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	addr	Destination address
Return value	-	void	-	
Function name	**R_QE_DMAC_RX_GetDstAddr**			
Description	Acquires transfer destination address of DMAC for reception. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	uint32_t	Destination address	
Function name	**R_QE_DMAC_TX_SetSrcAddr**			
Description	Sets transfer source address to DMAC for transmission. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint32_t	addr	Source address
Return value	-	void	-	
Function name	**R_QE_DMAC_TX_SetTxCnt**			
Description	Sets transfer count to DMAC for transmission. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint16_t	cnt	Transfer count
Return value	-	void	-	
Function name	**R_QE_SCI_IsTransferEnd**			
Description	Detects end of SCI transmission. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Transmission end false: During transmission	
Function name	**R_QE_SCI_SendStop**			
Description	Ends SCI transmission. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	-	void	-
Return value	-	void	-	
Function name	**R_QE_SCI_SetTDR**			
Description	Sets transmission data to SCI. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	uint8_t	data	Transmission data
Return value	-	void	-	
Function name	**R_QE_CMT_CntClear**			
Description	Clears CMT counter. (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	-	void	-
Return value	-	void	-	

7. Program Explanation

7.1 Overview

This program processes requests from QE for AFE with main function, acquiring and transmitting A/D conversion values by peripheral functions and by interrupt processing. Transmission of A/D conversion values is described in “7.2 Acquisition and Transmission of DSAD Conversion Results”. Figure 7-1 shows the flowchart of the main functions of this program.

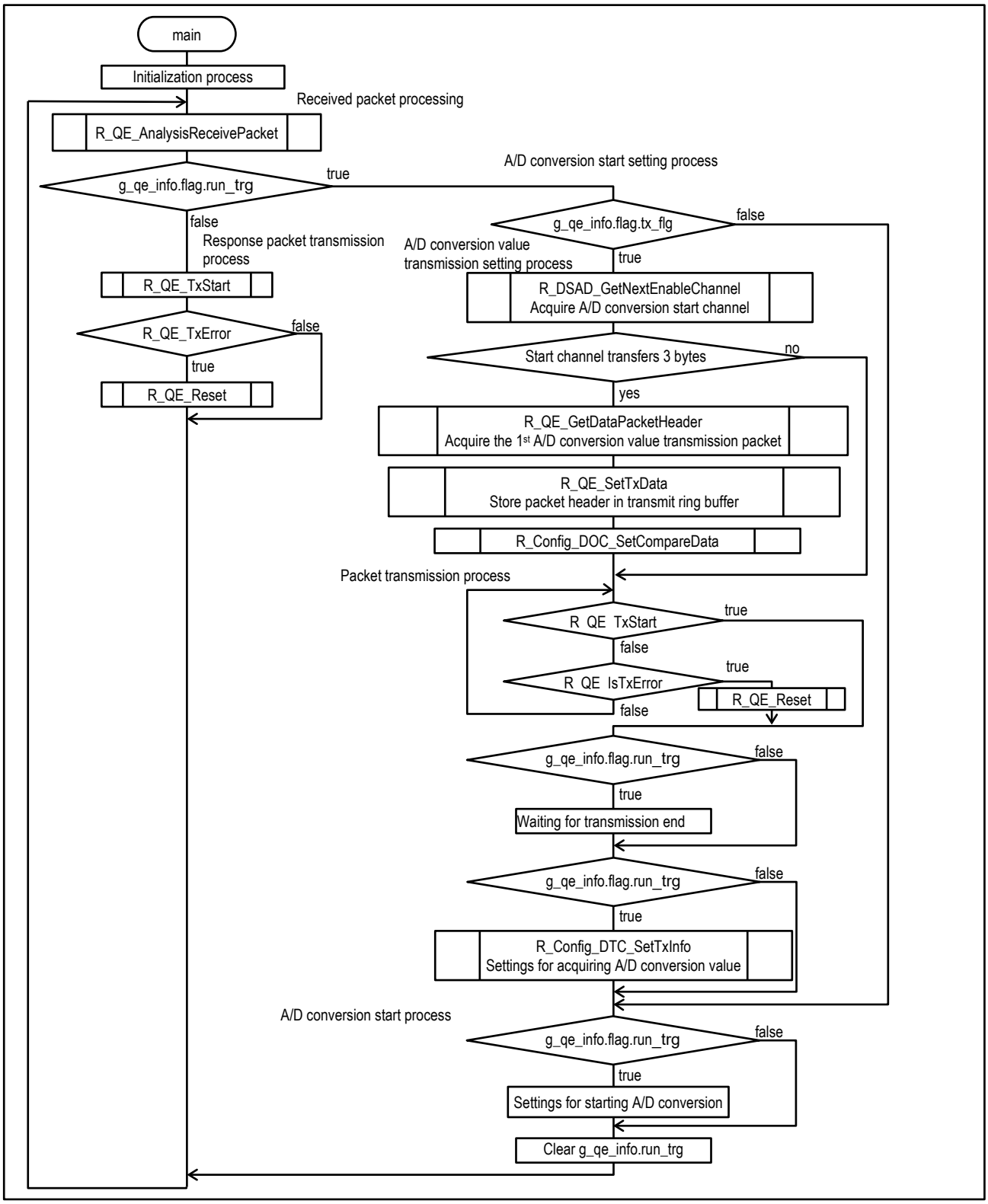

Figure 7-1 main Functions Flowchart

The following describes each process in Figure 7-1.

- Initialization process
  - DTC initial setting
    For automatic transmission of A/D conversion values, initializes the values to be set in DTC vector as many as enabled A/D channels.
  - Enabling IRQ5 interrupt
    Makes it possible to detect the communication busy signal CTS on QE for AFE side.
  - Communication module initialization and reception start
  - Turning LED0 and LED1 on.

- Received packet processing
  Processes a received packet and stores a response packet in the transmit ring buffer with R_QE_AnalysisReceivePacket function.

- Response packet transmission process
  - Transmission setting
    If a packet is stored in transmit ring buffer, performs settings for transmission start with R_QE_TxStart function.
  - Transmission error processing
    If a transmission error is detected with R_QE_IsTxError function, cancels transmission of the packet which is being transmitted.

- A/D conversion start setting process
  When the flag g_qe_info.flag.run_trg, which notifies reception of Run command, is set, processes A/D conversion value transmission setting and A/D conversion start.
  - A/D conversion value transmission start setting
    When the transmission data size of A/D conversion value of the enabled DSAD0 start channel is 3 bytes, creates a data transmission packet excluding data part and stores it in the transmit ring buffer. In addition, performs initial settings for data operation circuit to create DSAD0 error cumulative information.
  - Packet transmission process
    Transmits the packet stored in the transmit ring buffer and waits for transmission completion. If transmission can not completed, stops the transmission with R_QE_Reset function and clears g_qe_info.flag.run_trg.
  - A/D conversion value acquisition setting
    Sets DTC to acquire A/D conversion values.
  - A/D conversion start process
    Starts DTC transfer and A/D conversion to acquire A/D conversion values.

7.2 Acquisition and Transmission of DSAD Conversion Result

After A/D conversion starts, with the A/D conversion completion interrupt request of DSAS0 (ADI0) as a trigger, A/D conversion value acquisition and transmission setting are performed by DTC. With the transmission data empty interrupt request of SCI1 (TXI1) as a trigger, transmission data are provided to SCI1 by DMAC0.

At the A/D conversion value transmission, the data transmission packet structure and the transmission data size vary depending on the number of enabled channels and the A/D conversion count at each channel. Table 7-1 shows the packet structure and the transmission data size for DSAD0 conversion conditions.

Table 7-1 Data Transmission Packet Structure and Transmission Data Size for A/D conversion value transmission

Condition		**Packet structure, transmission data size**	
	A/D conversion value output rate/channel	**≦31250SPS**	**>31250SPS**
Single channel conversion		Normal 4 bytes	Extended: format 2 3 bytes
Multiple-channel conversion	A/D conversion count/channel ＝ 1		Normal 4 bytes
	2＜A/D conversion count/channel ≦84		Extended: format 1 3 bytes
	A/D conversion count/channel ＞ 85		Extended: format 3 3 bytes

When the transmission data size is 4 bytes, with ADI0 as a trigger, the A/D conversion result is sent as one packet.

When the transmission data size is 3 bytes, after transmitting the header, data length, and channel of the A/D conversion value transmission packet, with ADI0 as a trigger, performs A/D conversion result acquisition and A/D conversion value transmission. The flags of OVF and ERR are accumulated by comparing the upper 1 byte of A/D conversion result to the normal value in the data operation circuit (DOC) and by logical OR with data operation circuit interrupt request (DOPCI) handler due to ‘Not equal to’. The result can be acquired in DSAD Error Information after A/D conversion stops.

Figure 7-2 shows the peripheral functions used and the timing chart of 3-byte transmission. In the A/D conversion start process in Figure 7-1, after transmitting the header, data length, and channel of A/D conversion value transmission packet, starts A/D conversion value transmission by DTC transfer described below.

- DTC transfer
  0. A/D conversion result acquisition
     Transfers the A/D conversion result to the specified buffer.
  1-3. DMAC0 transfer setting
     Sets the source address, transfer count, and transfer permission to DMAC0 that transfers transmission data to SCI1.
  4. Transmission start
     Writes the first 1 byte of A/D conversion value to SCI1.TDR1 and starts transmission.
  5. Setting error flag part to DOC
     To acquire OVF and ERR flags for 3-byte transmission, writes the upper 1 byte of the acquired A/D conversion result to DOC.

When switching DSAD0 channels, performs transfer setting for next channel by the TXI1 interrupt handler of A/D conversion value DMAC transfer completion. Figure 7-3 shows the flowchart of TXI1 interrupt handler.

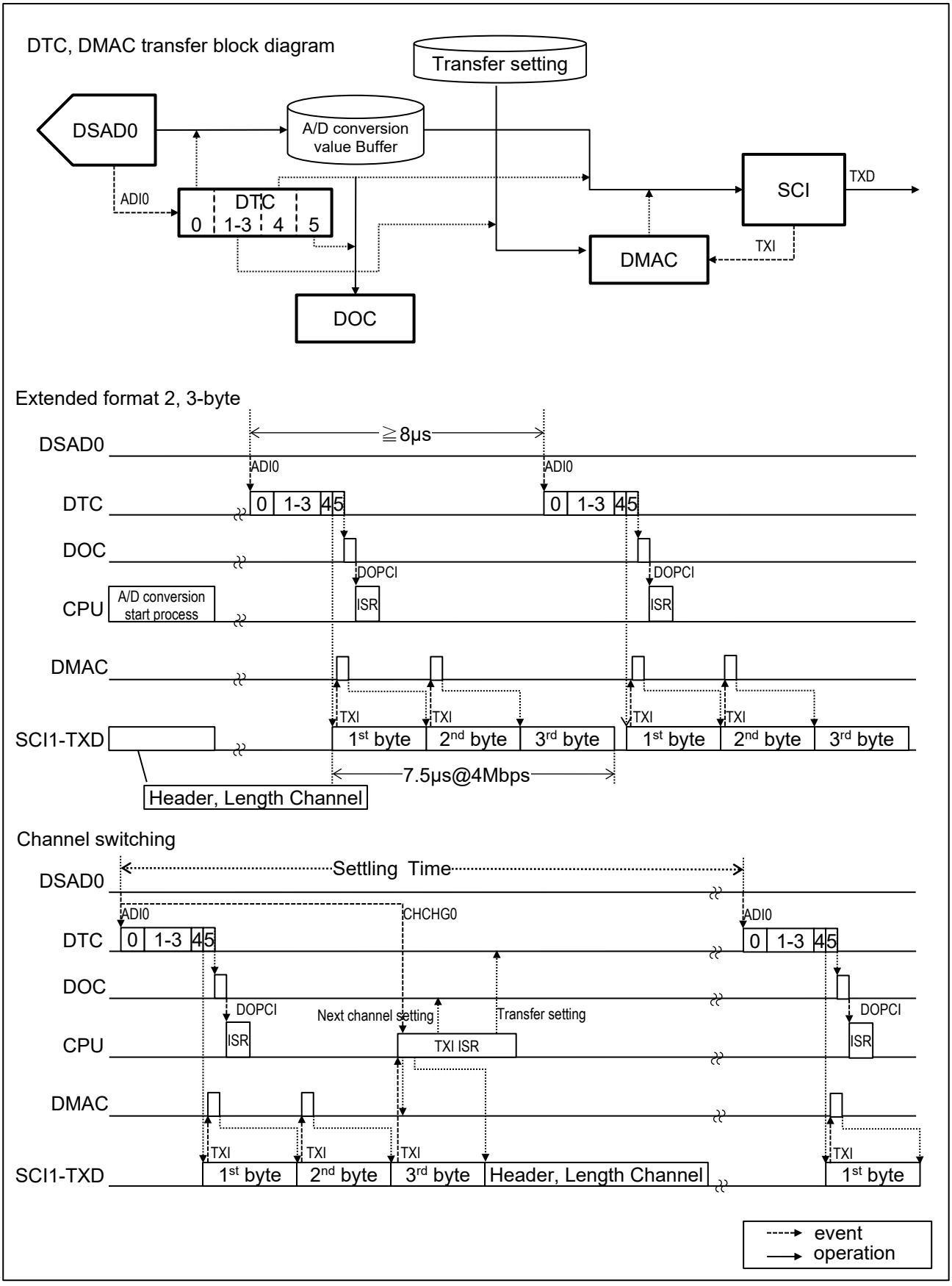

Figure 7-2 3-byteTransmission Timing Chart

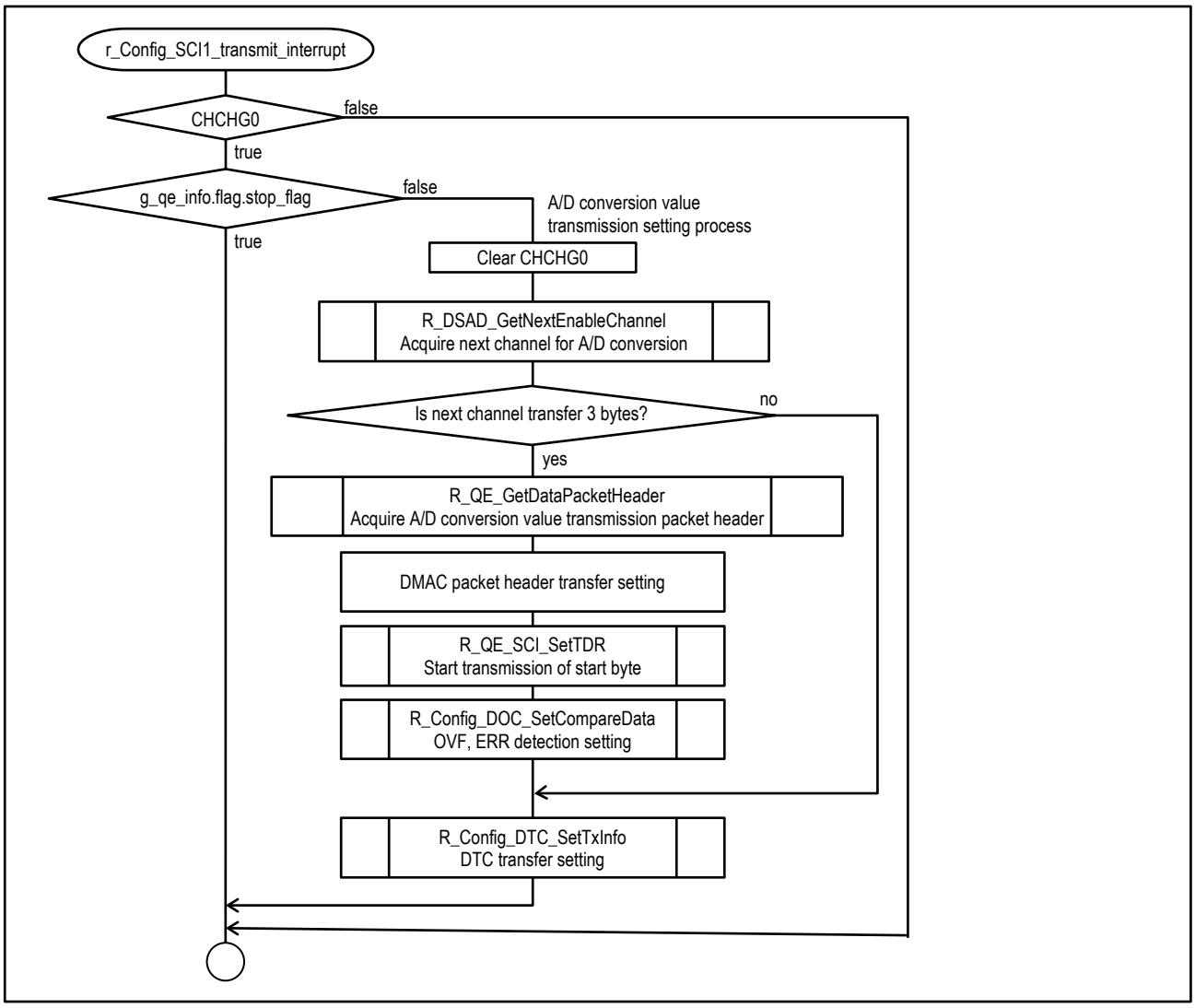

Figure 7-3 TXI1 Interrupt Handler Flowchart

7.3 Peripheral Functions and Pins

Table 7-2 lists the peripheral functions used in this program, Table 7-3 lists the pins used, Table 7-4 lists the pins unused, and Table 7-5 shows the clock settings.
The settings for the peripheral functions are performed by Smart Configurator. The following shows the settings for each peripheral function.

Table 7-2 Peripheral Functions Used

Peripheral function	Use
AFE、DSAD0	A/D conversion
R16DA	D/A output
DTC	A/D conversion result acquisition and transmission setting
DOC	Detection of DSAD conversion error
SCI1	UART communication with QE for AFE
DMAC0	Packet transmission to QE for AFE
DMAC1	Packet reception from QE for AFE
CMT0	Detection of timeout reception error from QE for AFE
ICU	UART communication CTS interrupt

Table 7-3 Pins used

Pin No.	Function	I/O	Use
5	P73	O	LED3 control
6	P72	O	LED2 control
7	P71	O	LED1 control
8	P70	O	LED0 control
28	RXD1	I	SCI1 reception
30	TXD1	O	SCI1 transmission
39	CTS1/IRQ5	I	SCI1 UART communication CTS interrupt
74	REFOUT	O	DSAD0 internal reference voltage output
76-79	AIN0-AIN3	I	DSAD0 signal input
80	REF0N	I	DSAD0 external reference voltage 0 input
81	REF0P	I	
82-89	AIN4-AIN11	I	DSAD0 signal input
90	HVCOM	I	DSAD0 signal input
91-94	HVAIN0-HVAIN3	I	
95	AIN12/REF1N	I	DSAD0 signal input, DSAD0 external reference voltage 1 input
96	AIN13/REF1P	I	
97-98	AIN14-AIN15	I	DSAD0 signal input
99	VREFL	I	Reference voltage input for 16-bit D/A converter
100	VREFH	I	

Table 7-4 Settings of unused pins

Pin No.	Function	Connection	Setting
4	P74	Open	Input, internal pull up
18-25	P60-P67		
26	NMI	Pull Up	-
27	P31	Open	Input, internal pull up
29	P27		
31-36	P20-P25		
37-38	P16-P17		
40-42	P12-P14		
43-44	P54-P55		
45	PC7		
46	PC6	Pull Down	Input (default)
47-48	PC5-PC4	Pull Up	
49	PC3	Open	Input, internal pull up
50	PC2	Pull Down	Input (default)
51	PC1	Pull Up	
52	PC0	Open	Input, internal pull up
53-57	PA0-PA4		
58,60	PB1,PB0		
62-65	PE1-PE4	SW1-SW3	Input (default)
66	PE0	Open	Input, internal pull up
67-71	PD0-PD4		

Table 7-5 Clock Settings

Item		Setting
Clock used		Main clock
	Oscillation source	Resonator
	Frequency	8MHz
	Wait time	8192 (2048μs)
PLL circuit	Frequency Division	x1/2
	Frequency Multiplication	x8
SCKCR (FCLK)		x1 (32MHz)
SCKCR (ICLK)		x1 (32MHz)
SCKCR (PCLKA)		x1 (32MHz)
SCKCR (PCLKB)		x1 (32MHz)
SCKCR (PCLKC)		x1 (32MHz)
SCKCR (PCLKD)		x1 (32MHz)

7.3.1 Analog Front End (AFE) and 24-bit Δ-Σ A/D Converter (DSAD0)

Table 7-6 shows the initial settings for DSAD0. AFE is not set because it doesn't have specific settings.

Table 7-6 DSAD0 Settings

Continuous scan mode

Item		Setting
Analog input channel setting		CH0
Operation clock setting		PCLK/2(16MHz)
Conversion start trigger source		Software trigger
Interrupt setting	Enable ΔΣA/D conversion completion interrupt (ADI0)	Enable, Priority: Level 15 (highest)
	Enable ΔΣA/D conversion scan completion interrupt (SCANEND0)	Disable
	Enable ΔΣA/D channel change interrupt (CHCHG0)	Disable
Voltage fault and disconnection setting		Not used
Analog input setting	Positive input signal	AIN1
	Negative input signal	AIN0
	Reference input	REFOUT/AVSS0
Amplifier setting	Amplifier selection	PGA
	PGA gain setting	x 1
ΔΣA/D conversion setting	A/D conversion mode	Normal operation
	Data format	Two's complement
	A/D conversion number	1
	First stage oversampling ratio	256
	Second stage oversampling ratio	16
	Set offset calibration value	Not used
	Set gain calibration value	Not used
Disconnect detection assist setting		Disable
Digital filter setting	Sinc filter select	Sinc4 + Sinc4
	Set sinc filter gain calibration	Disable

7.3.2 16-bit D/A Converter (R16DA)

Table 7-7 shows the initial settings for R16DA. These settings can be changed with QE for AFE.

Table 7-7 R16DA Settings

Item		Setting
D/A channel0 setting	Use DA0	Enable
	Buffer amplifier output pull-Down	Enable
Analog output impedance setting		Analog output pin is pulled down by a 1-kΩ resistor
D/A-A/D synchronous setting		Not used

7.3.3 Data Transfer Controller (DTC)

This is used to acquire and transmit the A/D conversion result. Table 7-8 and Table 7-9 show the settings.

Table 7-8 DTC Settings (1/2)

Item			Setting		
			DTC0	**DTC1**	**DTC0**
Base setting	Transfer data read skip		Disable		
	Address mode		Short-address mode (24 bits)		
	DTC vector base address		0x00007C00 (default value)		
Activation source setting	Activation source		DSAD0 (ADI0)	-	
	Chain transfer		Used		
Chain transfer setting			Continuous		
Transfer mode setting			Repeat mode		
Transfer data size setting			32 bits	32 bits	32 bits
Interrupt setting			An interrupt request to the CPU is disabled when specified data transfer is completed		
Block / Repeat area setting			Transfer destination		
Transfer address and count setting		Source address	0x000A1070 (DSAD0.DR) Address fixed	(Set by the program) Address fixed	
		Destination address	(Set by the program) Address fixed	0x00082000 (DMAC0.DMSAR) Address fixed	0x00082008 (DMAC0.DMCRA) Address fixed
		Count	1	1	1

Table 7-9 DTC Settings (2/2)

Item			Setting		
			DTC3	**DTC4**	**DTC5**
Base setting	Transfer data read skip		Disable		
	Address mode		Short-address mode (24 bits)		
	DTC vector base address		0x00007C00 (default value)		
Activation source setting	Activation source		-		
	Chain transfer		Used		Not used
Chain transfer setting			Continuous		-
Transfer mode setting			Repeat mode		
Transfer data size setting			8 bits	8 bits	16 bits
Interrupt setting			An interrupt request to the CPU is disabled when specified data transfer is completed		
Block / Repeat area setting			Transfer destination		
Transfer address and count setting		Source address	(Set by the program) Address fixed		
		Destination address	0x0008201C (DMAC0.DMCNT) Address fixed	0x0008A023 (SCI1.TDR) Address fixed	0x0008B082 (DOC.DODIR) Address fixed
		Count	1	1	1

7.3.4 Data Operation Circuit (DOC)

This is used to detect DSAD conversion errors. The settings are shown in Table 7-10.

Table 7-10 DOC Settings

Item		Setting
Data operation setting	Operation mode	Data comparison mode
	Detection condition select	'Not equal to' is to be detected
	Comparison reference / Initial value of addition or substruction result	0
Interrupt setting	Enable data operation circuit interrupt (DOPCF)	Enable
	Priority	Level 15 (highest)

7.3.5 Serial Communication Interface (SCI1) and DMA Controller (DMAC)

Table 7-11 shows settings of SCI1 which is used to communicate with QE for AFE, and Table 7-13 shows DMAC settings. In addition, TXI interrupt is used in fast interrupt. Table 7-12 shows the TXI1 settings in "Interrupt" tab.

Table 7-11 SCI1 Settings

Asynchronous mode
Operation mode: Transmission/reception

Item		Setting
Start bit edge detection setting		Falling edge on RXD1 pin
Data length setting		8 bits
Parity setting		None
Stop bit length setting		1 bit
Transfer direction setting		LSB-first
Transfer rate setting	Transfer clock	Internal clock
	Bit rate	4,000,000bps
	Enable modulation duty correction	Not used
	SCK1 pin function	SCK1 is not used
Noise filter setting		Not used
Hardware flow control setting		None
Data handling setting	Transmit data handling	Data handled by DMAC
	Receive data handling	Data handled by DMAC
Interrupt setting	Enable reception error interrupt (ERI1)	Not used
	TXI1, RXI1, TEI1, ERI1 priority	Level 2
Callback function setting		Not used

Table 7-12 TXI1 Interrupt Settings

Item	Setting
Vector Number	220
Interrupt	TXI1
Peripheral	SCI1
Priority	Level 2
Status	Used
Fast Interrupt	**Used**

Note: Set the setting written in bold.

Table 7-13 DMAC Settings

Item		Setting	
		DMAC0	**DMAC1**
Transfer setting	Activation source	SCI1 (TXI1)	SCI1 (RXI1)
	Activation source flag control	Clear interrupt flag of the activation source	
	Transfer mode	Normal mode	Free running mode
	Transfer data size	8 bits	
	Transfer count / Repeat size / Block size	(Setting on execution)	-
Source address setting	Source address	(Setting on execution) Incremented	0x0008A025 (SCI1.RDR) Fixed
	Specify the transfer source as extended repeat area	Enable	-
	Extended repeat area	Lower 9 bits of the address (512 bytes)	
Destination address setting	Destination address	0x0008A023(SCI1.TDR) Fixed	(Set by the program) Incremented
	Specify the transfer destination as extended repeat area	-	Enable
	Extended repeat area		Lower 9 bits of the address (512 bytes)
Interrupt setting		Not used	

7.3.6 Interrupt Controller (ICU)

Since this program detects CTS of SCI1 with IRQ5, IRQ5 is assigned to Pin 39 and used. Table 7-14 shows settings for component “Config_ICU”, and Table 7-15 shows pin assignment in “Pin” tab.

Table 7-14 ICU Settings (only for IRQ5)

Item		Setting
IRQ5 setting	IRQ5	Enable
	Detection type	Rising edge
	Digital filter	No filter
	Priority	Level 15 (highest)

Table 7-15 IRQ5 Pin Settings

Item		Setting
IRQ5	Assignment	P15/MTIOC0C/MTIOC4D/MTCLKB/TMCI2/CTS1#/RTS1#/SS1#/SSLA0 /CRXD0/SEG14/IRQ5
	Pin Number	39

7.3.7 I/O Port (PORT)

Table 7-16 shows settings for I/O ports to use according to Table 7-3 and Table 7-4.

Table 7-16 PORT Settings

PORT	Setting							
PORT7	P70	P71	P72	P73	P74			
	Out CMOS output Output 1				In Pull-up			
PORT1	P12	P13	P14	P15	P16	P17		
	In Pull-up			-	In Pull-up			
PORT2	P20	P21	P22	P23	P24	P25	P26	P27
	In Pull-up						-	In Pull-up
PORT3	P30	P31	P35	P36	P37			
	-	In Pull-up	-					
PORT5	P54	P55						
	In Pull-up							
PORT6	P60	P61	P62	P63	P64	P65	P66	P67
	In Pull-up							
PORTA	PA0	PA1	PA2	PA3	PA4			
	In Pull-up							
PORTB	PB0	PB1						
	In Pull-up							
PORTC	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
	In Pull-up	-		In Pull-up	-			In Pull-up
PORTD	PD0	PD1	PD2	PD3	PD4			
	In Pull-up							
PORTE	PE0	PE1	PE2	PE3	PE4			
	In Pull-up	-						

7.4 Program Configuration

7.4.1 Source File Configuration

Table 7-17 Source File Configuration

Folder name, file name	Description
src	
├rx23eb_qe	QE for AFE communication module
├smc_gen	Generated by Smart Configurator
│ ├Config_DA	
│ ├Config_DMAC0	
│ ├Config_DMAC1	
│ ├Config_DOC	
│ ├Config_DSAD0	
│ ├Config_DTC	
│ ├Config_ICU	
│ ├Config_PORT	
│ ├Config_SCI1	
│ ├general	
│ ├r_bsp	
│ ├r_config	
│ └r_pincfg	
└r_main.c	Main function

7.4.2 Macro Definitions

Table 7-18 r_qe_cfg.h

Definition name	value	Description
D_QE_CFG_TX_RINGBUF_SIZE	(512)	Transmit ring buffer size [byte]
D_QE_CFG_RX_RINGBUF_SIZE	(512)	Receive ring buffer size [byte]
D_QE_CFG_FORMAT_REV	(3)	Communication specifications revision
D_QE_CFG_READ	(1)	Support Read command
D_QE_CFG_WRITE	(1)	Support Write command
D_QE_CFG_USER_VAL0	(0)	User Value is not used
D_QE_CFG_USER_VAL1	(0)	
D_QE_CFG_USER_VAL2	(0)	
D_QE_CFG_USER_VAL3	(0)	
D_QE_CFG_USER_VAL4	(0)	
D_QE_CFG_USER_VAL5	(0)	
D_QE_CFG_USER_VAL6	(0)	
D_QE_CFG_USER_VAL7	(0)	
D_QE_CFG_EX_SPS	(0)	SPS information is not used
D_QE_CFG_EX_USER_BTN0	(0)	User button is not used
D_QE_CFG_EX_USER_BTN1	(0)	
D_QE_CFG_EX_USER_BTN2	(0)	
D_QE_CFG_EX_USER_BTN3	(0)	
D_QE_CFG_EX_USER_BTN4	(0)	
D_QE_CFG_EX_USER_BTN5	(0)	
D_QE_CFG_EX_USER_BTN6	(0)	
D_QE_CFG_EX_USER_BTN7	(0)	
D_QE_CFG_CH0	(0x2)	Transmission data type: A/D conversion value
D_QE_CFG_CH1	(0x2)	
D_QE_CFG_CH2	(0x2)	
D_QE_CFG_CH3	(0x2)	
D_QE_CFG_CH4	(0x2)	
D_QE_CFG_CH5	(0x2)	
D_QE_CFG_CH6	(0x2)	
D_QE_CFG_CH7	(0x2)	
D_QE_CFG_CH8	(0)	Transmission data type: No transmission
D_QE_CFG_CH9	(0)	
D_QE_CFG_CH10	(0)	
D_QE_CFG_CH11	(0)	
D_QE_CFG_CH12	(0)	
D_QE_CFG_CH13	(0)	
D_QE_CFG_CH14	(0)	
D_QE_CFG_CH15	(0)	
D_QE_CFG_TXT_INFO	“RSSKRX23E-B FW V1.0”	Text of program information
D_QE_CFG_TXERRCHK_EN	(1)	Enable transmission error check
D_QE_CFG_TIMEOUT	(1)	Detection of transmission timeout: CTS
D_QE_CFG_SCI	1	SCI channel number
D_QE_CFG_DMAC_RX	1	DMAC channel number for reception
D_QE_CFG_DMAC_TX	0	DMAC channel number for transmission
D_QE_CFG_CMT	0	CMT channel number (not used)

Table 7-19 r_qe_cfg.h Extended Definitions

Definition name	value	Description
D_QE_CFG_BURST_SPS	(31250)	SPS threshold value for selecting Data Transmission packet structure

7.4.3 Structures and Unions

Table 7-20 r_qe_cfg_typedef.h

Bold text: Extension in this program

Structure type name	st_qe_api_t		
Description	Communication module information		
Member	**Type**	**Name**	**Description**
	union	flag	API operation control flag
	uint8_t	byte	Entire flag
	struct	bit	Access in bit units
	uint8_t:1	tx_flag	Data transmission enable flag
	uint8_t:1	**run_trg**	A/D conversion start trigger
	uint8_t:1	**stop_flag**	A/D conversion stop processing flag
	uint8_t	dsad_err[8]	
	uint8_t	burst_tx[8]	
	uint16_t	**cmp_data[8]**	Comparison reference value for error detection for each DSAD0 channel
	st_dtc_tx_info_short_t	**dtc_info[8][6]**	DTC transfer information of each DSAD0 channel
	uint32_t	**current_ch**	DSAS0 channel number under A/D conversion
	uint32_t	**dmax_src_addr[8]**	Storage destination address for A/D conversion result of each DSAD0 channel
	uint32_t	**dmax_tx_cnt[8]**	DMAC transfer count of A/D conversion value transmission packet in each channel

Table 7-21 Config_DTC.h

Structure type name	st_dtc_tx_info_short_t		
Description	Transfer information for DTC short mode		
Member	**Type**	**Name**	**Description**
	uinon	mra_sar	MRA, SAR register setting values
	uint32_t	LONG	For long word access
	struct	BIT	Access in bit units
	uint32_t:8	mra	DTC transfer source control information
	uint32_t:24	sar	DTC transfer source address
	union	mrb_dar	MRB, DAR register values
	uint32_t	LONG	For Long word access
	struct	BIT	Access in bit units
	uint32_t:8	mrb	DTC transfer destination control information
	uint32_t:24	dar	DTC transfer destination address
	union	cra_crb	CRA, CRB register values
	cra_crb	LONG	For long word access
	struct	BIT	Access in bit units
	uint32_t:16	cra	CRA register setting value
	uint32_t:16	crb	CRB register setting value

7.4.4 Functions

Table 7-22 main.c Function

Function name	main			
Description	Main function			
Argument	I/O	Type	Name	Description
	-	void	-	-
Return value	-	void	-	

Table 7-23 r_qe_api_user.c QE for AFE Communication Module User-Defined API Function

Function name	R_QE_GetDataPacketHeader			
Description	Returns pointer to the prepared response packet structure of the specified channel			
Argument	I/O	Type	Name	Description
	-	uint32_t	ch	Channel
Return value	O	uint8_t *	-	
Function name	R_QE_GetDataPacketLength			
Description	Returns byte counts of the header, data length, and channel of the specified response packet structure			
Argument	I/O	Type	Name	Description
	-	uint8_t *	ptr	Pointer to response packet structure
Return value	O	size_t	Byte counts of header, data length, and channel	

Table 7-24 r_qe_api_user.c User-Defined Processing

Description only for this program processing

Function name	r_QE_WriteUser
Description	If writing destination is DAC, turns LED1 on. Otherwise, turns LED1 off
Function name	r_QE_RunUser
Description	If DSAD0 is operable, determines A/D conversion value transmission data size of each DSAD0 enabled channel, performs setting for DTC and DMAC to acquire and transmit A/D conversion result, prepares transmission packet structure, and then turns LED0 off.
Function name	r_QE_StopUser
Description	Stops DSAD0 A/D conversion and A/D conversion value transmission and turns LED0 on
Function name	r_QE_ExBurstTxInfoUser
Description	Stores the A/D conversion value transmission data size of each DSAD0 enabled channel in st_qe_api_t type variable member burst_tx
Function name	r_QE_ResetUser
Description	Clears st_qe_api_t type variable member flag.run_trg.

Table 7-25 r_qe_api_user.c User-Defined Functions

Function name	**r_calc_burst_tx**			
Description	Calculates the A/D conversion value transmission data size of the enabled channel from the current DSAD0 settings and stores it in qe_info member burst_tx			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	st_qe_api_t *	qe_info	Pointer to communication module
Return value	-	void	-	
Function name	**r_tx_setting**			
Description	Performs settings for DTC and DMAC to acquire and transmit the A/D conversion result from the current DSAD0 settings and prepares transmission packet structure			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	st_qe_api_t *	qe_info	Pointer to communication module
Return value	-	void	-	

Table 7-26 Config_SCI1 User-Defined Function

Function name	**R_SCI1_GetCTS**			
Description	Acquires the status of CTS pin (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: CTS assert false :CTS negate	

Table 7-27 Config_DSAD0 User-Defined Functions

Function name	**R_DSAD0_IsCHCHG**			
Description	Detects occurrence of switching channels (CHCHG0). When it is detected, clears CHCHG0.			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Detected false: Not detected	
Function name	**R_DSAD0_IsEnable**			
Description	Detects enable/disable settings (OPCR.DSAD0EN) (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: Enable false: Disable	
Function name	**R_DSAD0_IsStart**			
Description	Detects A/D conversion in progress (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	bool	true: A/D conversion in progress false: Stopped	
Function name	**R_DSAD_GetNextEnabledChannel**			
Description	Acquires the enabled channel number next to the specified channel number			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	uint32_t	current_ch	Channel number
Return value	O	uint32_t	Next enabled channel number	
Function name	**R_DSAD_GetEnabledChannelCount**			
Description	Acquires the number of enabled channels.			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	uint32_t	Number of enabled channels	
Function name	**R_DSAD0_GetConversionCnt**			
Description	Acquires A/D conversion count setting of the specified channel			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	uint32_t	ch	Channel number
Return value	O	uint32_t	A/D conversion count setting value	
Function name	**R_DSAD0_GetSPS**			
Description	Acquires A/D conversion value output rate of the specified channel			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	uint32_t	ch	Channel number
Return value	O	float	Output rate [SPS]	
Function name	**R_DSAD0_GetCHmEN**			
Description	Acquires enabled channel information (MR.CH0EN-MR.CH7EN)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	void	-	-
Return value	O	uint32_t	Information on the enabled channel with CH0EN in LSB Each bit 0: Enable 1: Disable	

Table 7-28 Config_DTC User-Defined Functions

Function name	**R_Config_DTC_SetDAR**			
Description	Sets the transfer destination start address to DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_dtc_tx_info_short_t *	p_vec	Pointer to DTC transfer information
	I	uint32_t	addr	Destination start address
Function name	-	void	-	
Function name	**R_Config_DTC_SetSAR**			
Description	Sets the transfer source address to DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_dtc_tx_info_short_t *	p_vec	Pointer to DTC transfer information
	I	uint32_t	addr	Source start address
Function name	-	void	-	
Function name	**R_Config_DTC_SetMRA**			
Description	Sets the value to MRA of DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_dtc_tx_info_short_t *	p_vec	Pointer to DTC transfer information
	I	uint8_t	val	Setting value
Function name	-	void	-	
Function name	**R_Config_DTC_SetMRB**			
Description	Sets the value to MRB of DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_dtc_tx_info_short_t *	p_vec	Pointer to DTC transfer information
	I	uint8_t	val	Setting value
Function name	-	void	-	
Function name	**R_Config_DTC_SetTxInfo**			
Description	Sets the transfer information to DTC			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	st_dtc_tx_info_short_t *	p_vec	Pointer to DTC transfer information
Return value	-	void	-	
Function name	**R_Config_DTC_InitVectorTable**			
Description	Stores the current DTC transfer information in DTC transfer information			
Argument	**I/O**	**Type**	**Name**	**Description**
	I/O	st_dtc_tx_info_short_t *	p_vec	Pointer to DTC transfer information
Return value	-	void	-	

Table 7-29 Config_DOC User-Defined Function

Function name	**R_Config_DOC_SetCompareData**			
Description	Sets the value to DODSR register (macro function)			
Argument	**I/O**	**Type**	**Name**	**Description**
	-	uint16_t	data	Setting value
Return value	-	void	-	

Table 7-30 Config_PORT User-Defined Functions

Function name	**R_Config_PORT_LED0_ON**			
Description	Turns LED0 ON/OFF			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	flag	true: ON false: OFF
Return value	-	void	-	
Function name	**R_Config_PORT_LED1_ON**			
Description	Turns LED1 ON/OFF			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	flag	true: ON false: OFF
Return value	-	void	-	
Function name	**R_Config_PORT_LED2_ON**			
Description	Turns LED2 ON/OFF			
Argument	**I/O**	**Type**	**Name**	**Description**
	I	bool	flag	true: ON false: OFF
Return value	-	void	-	

8. Importing a Project

After importing the sample project, make sure to confirm build and debugger setting.

8.1 Importing a Project into e2 studio

Follow the steps below to import your project into e^2 studio. Pictures may be different depending on the version of e^2 studio to be used.

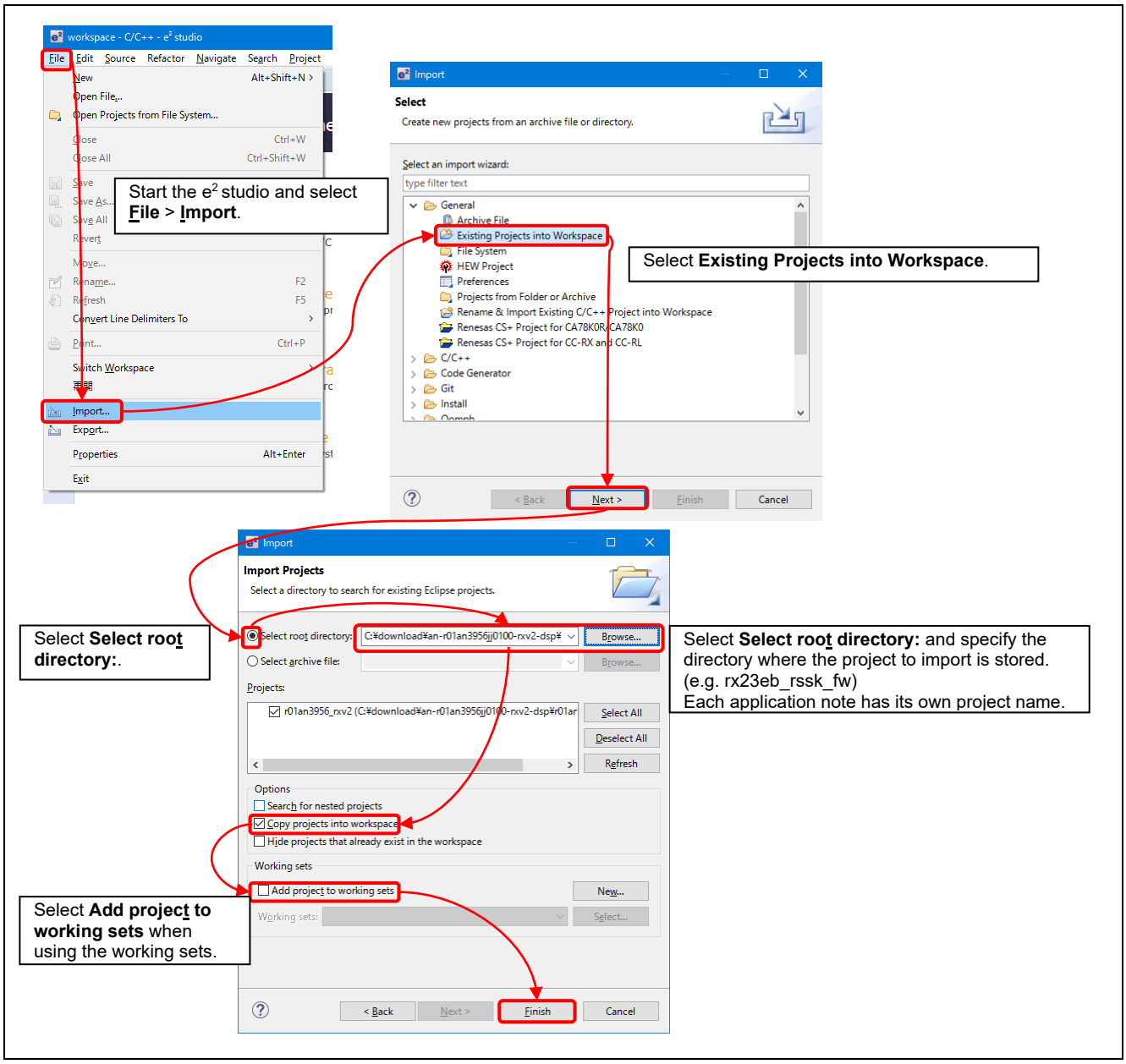

Figure 8-1 Importing a project into e^2 studio

8.2 Importing a Project into CS+

Follow the steps below to import your project into CS+. Pictures may be different depending on the version of CS+ to be used.

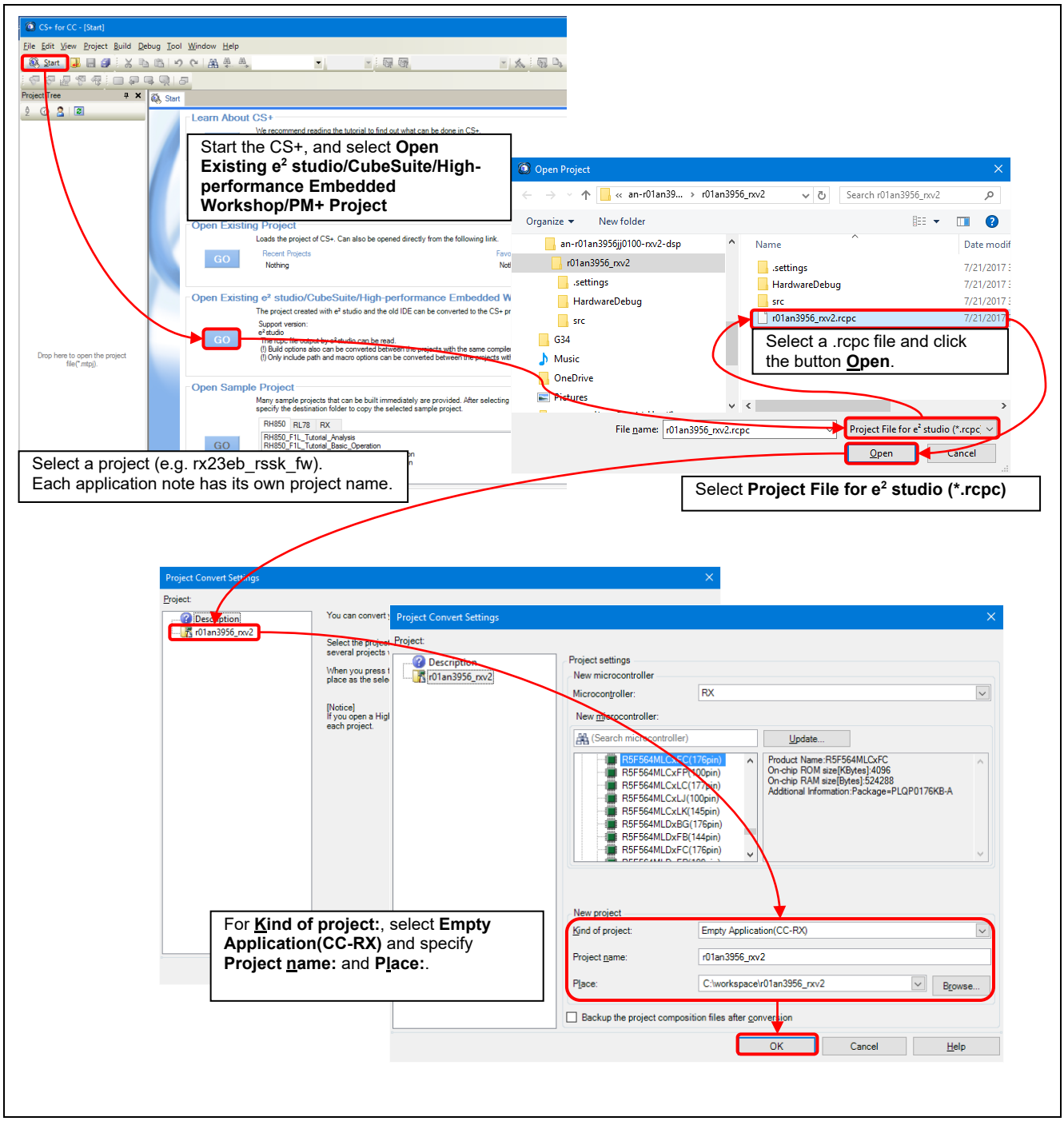

Figure 8-2 Importing a project into CS+

9. Memory Usage

9.1 Build Conditions

The build conditions for this program are listed in Table 9-1.

Table 9-1 Build Conditions

Item		Setting
Compiler		-isa=rxv2 -utf8 -nomessage -output=obj -obj_path=${workspace_loc:/${ProjName}/${ConfigName}} -debug -outcode=utf8 -listfile="$(dir $@)¥$(basename $(notdir $<)).lst" -show=source,conditionals,definitions,expansions -nologo
Linker		-noprelink -output="rx23eb_rssk_fw.abs" -form=absolute -map="rx23eb_rssk_fw.bls" -nomessage -vect=_undefined_interrupt_source_isr -list=rx23eb_rssk_fw.map -nooptimize -rom=D=R,D_1=R_1,D_2=R_2 -cpu=RAM=00000000-00007fff, FIX=00080000-00083fff,FIX=00086000-00087fff,FIX=00088000-0008dfff, FIX=00090000-0009ffff,FIX=000a0000-000bffff,FIX=000c0000-000fffff, ROM=00100000-00101fff,FIX=007fc000-007fc4ff,FIX=007ffc00-007fffff, ROM=fffc0000-ffffffff -total_size -nologo
	Section	SU,SI,B_1,R_1,B_2,R_2,B,R/04,B_DMAC_REPEAT_AREA_1/02800, PResetPRG,C_1,C_2,C,C$*,D*,W*,L, P/0FFFC0000,EXCEPTVECT/0FFFFFF80,RESETVECT/0FFFFFFFC

Note: Included paths other than user settings in compiler setting are omitted.

9.2 Memory Usage

The amount of memory usage of this program is shown in Table 9-2.

Table 9-2 Amount of Memory Usage

Item		Size [byte]	Remarks
ROM		9807	
	Code	8346	
	Data	1461	
RAM		11718 (268)	Note
	Data	6058	
	Stack	5120 (268)	Note

Note: RAM usage shown in "()" is calculated from stack usage.

,

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	Oct.16.23	-	First release

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.

Notice

1. 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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/.