RZ/T1 Group

RSPI Sample Program

R01AN2601EJ0140
Rev.1.40
Jun. 07, 2018

Introduction

This application note explains a Renesas Serial Peripheral Interface (RSPI) sample driver that supports the RSK RZ/T1 Evaluation Board equipped with RZ/T1 Group MCU.

- Channels
  Supports only channel 1 of all four RSPI channels and does not support channel 0, 2, or 3
- Interrupts
  Supports transmit buffer empty interrupts and transfer error interrupts
  Does not support receive buffer full interrupts or RSPI idle interrupts
- Operating modes
  - Supports clock-synchronous operation (three lines)
  - Supports master mode
  - Supports a transfer data bit length of 16 bits
  - Supports a bit rate of 5 Mbps when the High-Speed Serial Clock (SERICLK) is 150 MHz
  - Supports MSB-first format

  Other operating modes are not supported.
- DMA transfer
  Not supported
- Serial communication
  Supports only transmission processing and does not support reception processing
  Controls RSPI slave select signals by software
- WM8978 settings
  Supports initial settings, headphone volume level settings, microphone volume level settings, and termination settings

Target Devices

RZ/T1

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

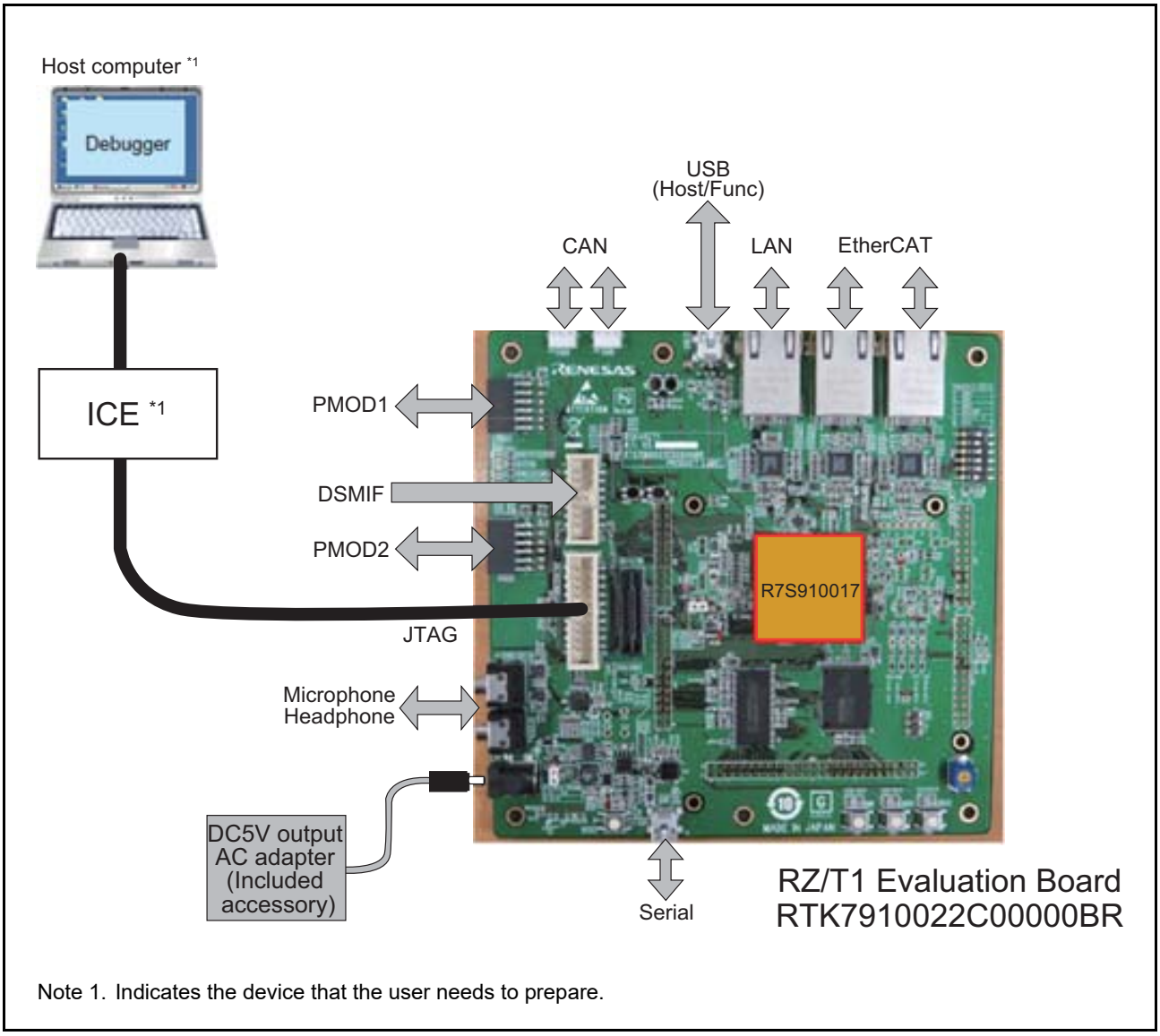
Table of Contents

1. Specifications

Table 1.1 Peripheral Functions and Applications lists the peripheral functions to be used and their applications, and Figure 1.1 shows the Operating Environment where the sample code is executed.

Table 1.1 Peripheral Functions and Applications

Peripheral Function	Application
RZ/T1 RSPI channel 1	Clock and transmission data control for WM8978 Stereo CODEC with Speaker Driver
RZ/T1 general I/O port (P43)	Slave selection control for WM8978 Audio CODEC with Speaker Driver
RZ/T1 ICUA (RSPI Unit1)	Interrupt control • Transmit buffer empty interrupt (SPTI) source: Vector number 85 • RSPI error interrupt (SPEI) source: Vector number 86
RZ/T1 clock generation circuit	Clock supply control for the RSPI (SERICLK)
RZ/T1 RSPI power saving function	Saving power consumption for the RSPI (RSPIa Units 0, 1, 2, and 3)
WM8978 (mounted on the evaluation board)	AudioCODEC&SpeakerDriver



Note 1. Indicates the device that the user needs to prepare.

Figure 1.1 Operating Environment

2. Operating Environment

The sample code covered in this application note is for the environment below.

Table 2.1 Operating Environment

Item	Description
Microcomputer	RZ/T1 Group
Operating frequency	CPUCLK = 450 MHz
Operating voltage	3.3 V
Integrated Development Environment	Manufactured by IAR Systems Embedded Workbench® for Arm Version 8.20.2 Manufactured by Arm DS-5™ 5.26.2 Manufactured by RENESAS e2studio 6.1.0
Operating mode	SPI boot mode 16-bit bus boot mode
Board	RZ/T1 Evaluation Board (RTK7910022C00000BR)
Device (functions to be used on the board)	• NOR flash memory (connected to CS0 and CS1 spaces) Manufacturer: Macronix International Co., Ltd. Model: MX29GL512FLT2I-10Q • SDRAM (connected to CS2 and CS3 spaces) Manufacturer: Integrated Silicon Solution Inc. Model: IS42S16320D-7TL • Serial flash memory Manufacturer: Macronix International Co., Ltd. Model: MX25L51245G • Audio CODEC Manufacturer: Wolfson, Model: WM8978GEFL/V

3. Related Application Note

The application note related to this application note is listed below for reference.

- Application Note: RZ/T1 Group Initial Settings (R01AN2554EJ)

Note: For any registers not covered by this application note, use the values specified in the Application Note: RZ/T1 Group Initial Settings.

4. Peripheral Functions

The basics of the operating modes, serial peripheral interface (RSPIa), general I/O port, interrupt controller (ICUA), clock generation circuit, and power saving function are described in RZ/T1 Group User's Manual: Hardware.

The basics of the WM8978 CODEC are described in the WM8978 datasheets.

5. Hardware

5.1 Hardware Configuration

Figure 5.1 shows the Hardware Configuration for the RSPI.

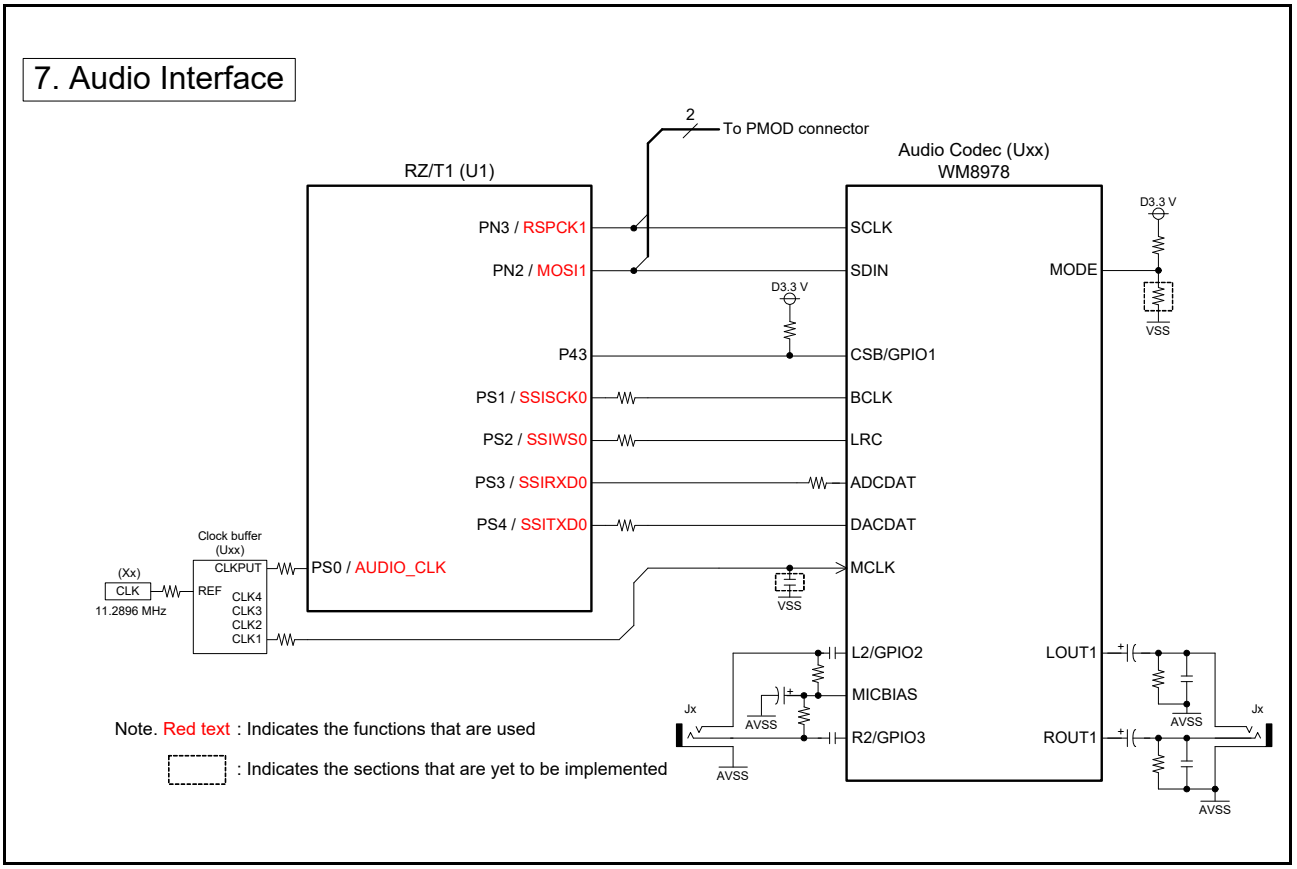


Figure 5.1 Hardware Configuration for the RSPI

5.2 Pins

Table 5.1 shows the Pins Used and Their Functions.

Table 5.1 Pins Used and Their Functions

Pin Name	Input/Output	Function
RSPCK1	Output	Clock signal for the RSPI (channel 1)
MOSI1	Output	Data signal for the RSPI (channel 1)
P43	Output	Slave select signal (software-controlled) for the RSPI (channel 1)

6. Software

6.1 Operation Outline

Table 6.1 Operation Outline presents a functional overview of the RSPI sample program. Figure 6.1 shows the System Block Diagram for this program.

Table 6.1 Operation Outline

Function	Outline
Channel	Setting channel 1
Operating mode	• SPI operation: 3-line system • Communication mode: Master mode (transmission only) • Data length: 16 bits • Bit rate: 5 Mbps
Communication start conditions	Starting serial communications by software activation
Sample program operation outline	Sending commands to the WM8978 (to control the input and output volume levels)

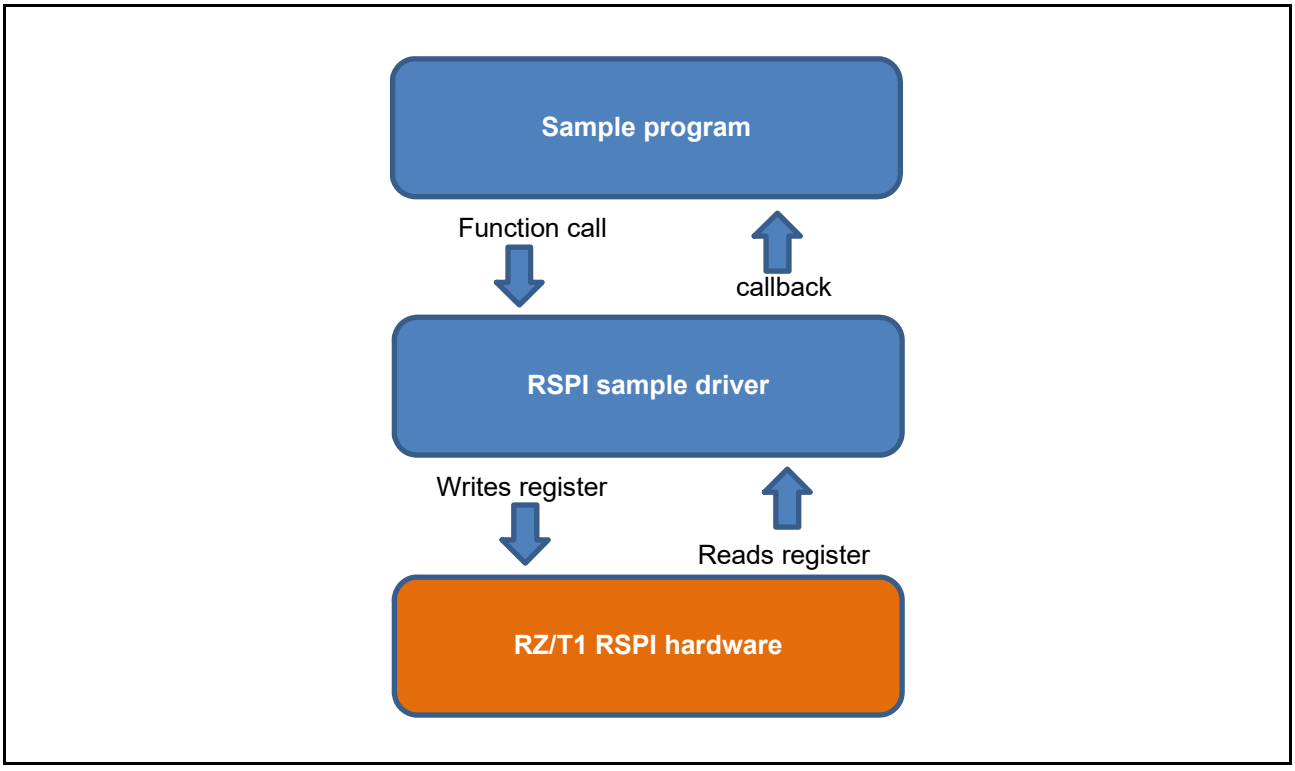


Figure 6.1 System Block Diagram

6.1.1 Project Setup

How to set up projects used in the EWARM development environment is described in the Application Note: RZ/T1 Group Initial Settings.

6.1.2 Preparation

There is no need to prepare for executing this sample program.

6.2 Memory Mapping

Memory mapping for the address spaces in the RZ/T1 Group MCU and the memory in the RZ/T1 Evaluation Board is described in the Application Note: RZ/T1 Group Initial Settings.

6.2.1 Section Allocation for the Sample program

The sections used by the sample program, the section allocation for the sample program in the initial state (load view), and the section allocation for the sample program after the scatter loading function is used (execution view) are described in the Application Note: RZ/T1 Group Initial Settings.

6.2.2 MPU Setup

MPU setup is described in the Application Note: RZ/T1 Group Initial Settings.

6.2.3 Exception Processing Vector Table

Exception processing vector tables are described in the Application Note: RZ/T1 Group Initial Settings.

6.3 Interrupts

Table 6.2 lists the interrupts for this sample code.

Table 6.2 Interrupts for the Sample Code

Interrupt (Source ID)	Priority	Process Outline
RSPI transmit buffer empty interrupt (SPTI)	3	Transmission completion processing
RSPI error interrupt (SPEI)	3	Transmission error processing

6.4 Fixed-Width Integer Types

Table 6.3 shows the Fixed-Width Integer Types for the Sample Code.

Table 6.3 Fixed-Width Integer Types for the Sample Code

Symbol	Description
int8_t	8-bit signed integer (defined in the standard library)
int16_t	16-bit signed integer (defined in the standard library)
int32_t	32-bit signed integer (defined in the standard library)
int64_t	64-bit signed integer (defined in the standard library)
uint8_t	8-bit unsigned integer (defined in the standard library)
uint16_t	16-bit unsigned integer (defined in the standard library)
uint32_t	32-bit unsigned integer (defined in the standard library)
uint64_t	64-bit unsigned integer (defined in the standard library)

6.5 Constants/Error Codes

Table 6.4 to Table 6.10 show the constants for the sample code, and Table 6.8 shows the API Error Codes for the RSPI Sample Driver.

Table 6.4 RSPI Register Bit Definition Constants for the RSPI Sample Driver

Constant Name	Setting Value	Description
RSPI_RZT_VERSION_MAJOR	1	Major Version
RSPI_RZT_VERSION_MINOR	0	Minor Version
RSPI_BYTE_DATA	0x01	RSPI data length judgment result (in bytes)
RSPI_WORD_DATA	0x02	RSPI data length judgment result (in words)
RSPI_LONG_DATA	0x04	RSPI data length judgment result (in long words)

Table 6.5 RSPI Register Bit Definition Constants for the RSPI Sample Driver (1 / 2)

Constant Name	Setting Value	Description
RSPI_SPCR_SPMS	0x01	RSPI mode selection bits for the SPCR register
RSPI_SPCR_TXMD	0x02	Communication operation mode selection bits for the SPCR register
RSPI_SPCR_MODFEN	0x04	Mode fault error detection enable bits for the SPCR register
RSPI_SPCR_MSTR	0x08	Master/slave mode selection bits for the SPCR register
RSPI_SPCR_SPEIE	0x10	Error interrupt enable bits for the SPCR register
RSPI_SPCR_SPTIE	0x20	Transmit-buffer-empty interrupt enable bits for the SPCR register
RSPI_SPCR_SPE	0x40	RSPI function enable bits for the SPCR register
RSPI_SPCR_SPRIE	0x80	Receive-buffer-full interrupt enable bits for the SPCR register
RSPI_SSLP_MASK	0x0F	Non-reserve bit mask for the SPCR register
RSPI_SPPCR_MASK	0x37	Non-reserve bit mask for the SPPCR register
RSPI_SPSR_OVRF	0x01	Overrun error flag for the SPSR register
RSPI_SPSR_IDLNF	0x02	RSPI idle flag for the SPSR register
RSPI_SPSR_MODF	0x04	Mode fault error flag for the SPSR register
RSPI_SPSR_PERF	0x08	Parity error flag for the SPSR register
RSPI_SPSR_MASK	0xA0	Non-reserve bit mask for the SPPCR register
RSPI_SPDCR_SPFC	0x03	Frame count setting bits for the SPDCR register
RSPI_SPDCR_SPRDTD	0x10	RSPI send/receive data selection bits for the SPDCR register
RSPI_SPDCR_SPLW	0x20	RSPI long-word-access/word-access setting bits for the SPDCR register
RSPI_SPCKD_MASK	0x07	RSPCK delay setting bits for the SPCKD register
RSPI_SSLND_MASK	0x07	SLL negation delay setting bits for the SSLND register
RSPI_SPND_MASK	0x07	RSPI next-access delay setting bits for the SPND register
RSPI_SPCR2_SPPE	0x01	Parity enable bits for the SPCR2 register
RSPI_SPCR2_SPOE	0x02	Parity mode bits for the SPCR2 register
RSPI_SPCR2_SPIIE	0x04	RSPI idle interrupt enable bits for the SPCR2 register
RSPI_SPCR2_PTE	0x08	Parity self-diagnosis bits for the SPCR2 register
RSPI_SPCR2_MASK	0x1F	Non-reserve bit mask for the SPCR2 register
RSPI_SPCMD_CPHA	0x0001	RSPCK phase setting bits for the SPCMD register
RSPI_SPCMD_CPOL	0x0002	RSPCK polarity setting bits for the SPCMD register
RSPI_SPCMD_BRDV	0x000c	Bit rate division setting bits for the SPCMD register
RSPI_SPCMD_SSLA	0x0070	SSL signal assertion setting bits for the SPCMD register

Table 6.5 RSPI Register Bit Definition Constants for the RSPI Sample Driver (2 / 2)

Constant Name	Setting Value	Description
RSPI_SPCMD_SSLKP	0x0080	SSL signal level hold bits for the SPCMD register
RSPI_SPCMD_SPB	0x0F00	RSPI data length setting bits for the SPCMD register
RSPI_SPCMD_LSBF	0x1000	RSPI LSB-first bits for the SPCMD register
RSPI_SPCMD_SPNDEN	0x2000	RSPI next-access delay setting bits for the SPCMD register
RSPI_SPCMD_SLNDEN	0x4000	SLL negation delay setting enable bits for the SPCMD register
RSPI_SPCMD_SCKDEN	0x8000	RSPCK delay setting enable bits for the SPCMD register

Table 6.6 RSPI Register Default Constants for the RSPI Sample Driver (1 / 2)

Constant Name	Setting Value	Description
RSPI_SPCR_SPMS_DEF	0x00	SPMS bit default setting for the SPCR register
RSPI_SPCR_TXMD_DEF	0x00	TXMD bit default setting for the SPCR register
RSPI_SPCR_MODFEN_DEF	0x00	MODFEN bit default setting for the SPCR register
RSPI_SPCR_MSTR_DEF	0x08	MSTR bit default setting for the SPCR register
RSPI_SPCR_SPEIE_DEF	0x00	SPEIE bit default setting for the SPCR register
RSPI_SPCR_SPTIE_DEF	0x00	SPTIE bit default setting for the SPCR register
RSPI_SPCR_SPE_DEF	0x00	SPE bit default setting for the SPCR register
RSPI_SPCR_SPRIE_DEF	0x00	SPRIE bit default setting for the SPCR register
RSPI_SPCR_DEF	0x08	SPCR register default setting
RSPI_SSLP_SSL0P_DEF	0x00	SSL0P bit default setting for the SSLP register
RSPI_SSLP_SSL1P_DEF	0x00	SSL1P bit default setting for the SSLP register
RSPI_SSLP_SSL2P_DEF	0x00	SSL2P bit default setting for the SSLP register
RSPI_SSLP_SSL3P_DEF	0x00	SSL3P bit default setting for the SSLP register
RSPI_SSLP_DEF	0x00	SSLP register default setting
RSPI_SPPCR_SPLP_DEF	0x00	SPLP bit default setting for the SPPCR register
RSPI_SPPCR_SPLP2_DEF	0x00	SPLP2 bit default setting for the SPPCR register
RSPI_SPPCR_MOIFV_DEF	0x00	MOIFV bit default setting for the SPPCR register
RSPI_SPPCR_MOIFE_DEF	0x00	MOIFE bit default setting for the SPPCR register
RSPI_SPPCR_DEF	0x00	SPPCR register default setting
RSPI_SPSCR_DEF	0x00	SPSCR register default setting
RSPI_SPBR_DEF	0x01	SPBR register default setting
RSPI_SPDCR_SPFC_DEF	0x00	SPFC bit default setting for the SPDCR register
RSPI_SPDCR_SPRDTD_DEF	0x00	SPRDTD bit default setting for the SPDCR register
RSPI_SPDCR_SPLW_DEF	0x20	SPLW bit default setting for the SPDCR register
RSPI_SPDCR_DEF	0x20	SPDCR register default setting
RSPI_SSLND_DEF	0x00	SSLND register default setting
RSPI_SPND_DEF	0x00	SPND register default setting
RSPI_SPCR2_SPPE_DEF	0x00	SPPE bit default setting for the SPCR2 register
RSPI_SPCR2_SPOE_DEF	0x00	SPOE bit default setting for the SPCR2 register
RSPI_SPCR2_SPIIE_DEF	0x00	SPIIE bit default setting for the SPCR2 register
RSPI_SPCR2_PTE_DEF	0x00	PTE bit default setting for the SPCR2 register
RSPI_SPCR2_DEF	0x00	SPCR2 register default setting
RSPI_SPCMD_CPHA_DEF	0x0000	CPHA bit default setting for the SPCMD register
RSPI_SPCMD_CPOL_DEF	0x0000	CPOL bit default setting for the SPCMD register
RSPI_SPCMD_BRDV_DEF	0x0000	BRDV bit default setting for the SPCMD register

Table 6.6 RSPI Register Default Constants for the RSPI Sample Driver (2 / 2)

Constant Name	Setting Value	Description
RSPI_SPCMD_SSLA_DEF	0x0000	SSLA bit default setting for the SPCMD register
RSPI_SPCMD_SSLKP_DEF	0x0000	SSLKP bit default setting for the SPCMD register
RSPI_SPCMD_SPB_DEF	0x0400	SPB bit default setting for the SPCMD register
RSPI_SPCMD_LSBF_DEF	0x0000	LSBF bit default setting for the SPCMD register
RSPI_SPCMD_SPNDEN_DEF	0x0000	SPNDEN bit default setting for the SPCMD register
RSPI_SPCMD_SLNDEN_DEF	0x0000	SLNDEN bit default setting for the SPCMD register
RSPI_SPCMD_SCKDEN_DEF	0x0000	SCKDEN bit default setting for the SPCMD register
RSPI_SPCMD_DEF	0x0400	SPCMD register default setting

Table 6.7 Configuration Constants for the RSPI Sample Driver

Constant Name	Setting Value	Description
RSPI_DUMMY_TXDATA	0xFFFFFFFF	Dummy transfer data
RSPI_CFG_USE_CH0	0	RSPI channel 0 not used
RSPI_CFG_USE_CH1	1	RSPI channel 1 used
RSPI_CFG_USE_CH2	0	RSPI channel 2 not used
RSPI_CFG_USE_CH3	0	RSPI channel 3 not used
RSPI_IR_PRIORITY_CHAN0	3	RSPI channel 0 priority order
RSPI_IR_PRIORITY_CHAN1	3	RSPI channel 1 priority order
RSPI_IR_PRIORITY_CHAN2	3	RSPI channel 2 priority order
RSPI_IR_PRIORITY_CHAN3	3	RSPI channel 3 priority order
RSPI_CFG_MASK_UNUSED_BITS	0	Transfer bit length mask setting
RSPI_NUM_CHANNELS	4	Number of RSPI channels
RSPI_POWER_ON	0	RSPI power saving function ON
RSPI_POWER_OFF	1	RSPI power saving function OFF
RSPI_PN3PFS_SET_VAL	0x0e	PN3PFS setting (RSPCK1 selection)
RSPI_PN2PFS_SET_VAL	0x0e	PN2PFS setting (MOSI1 selection)

Table 6.8 API Error Codes for the RSPI Sample Driver

Constant Name	Setting Value	Description
RSPI_SUCCESS	0	Normal termination
RSPI_ERR_BAD_CHAN	1	Channel number error
RSPI_ERR_CH_NOT_OPENED	2	Open error
RSPI_ERR_CH_NOT_CLOSED	3	Closing error
RSPI_ERR_UNKNOWN_CMD	4	Command error
RSPI_ERR_INVALID_ARG	5	Parameter error
RSPI_ERR_ARG_RANGE	6	Parameter range error
RSPI_ERR_NULL_PTR	7	Null pointer
RSPI_ERR_LOCK	8	Lock error
RSPI_ERR_UNDEF	9	Unknown error
RSPI_SUCCESS_ALL_TX_COMPLETE	10	Normal termination (all data transfers completed)

Table 6.9 Internal State Codes for the RSPI Sample Driver

Constant Name	Setting Value	Description
RSPI_EVT_TRANSFER_COMPLETE	0	Transfer complete
RSPI_EVT_TRANSFER_ABORTED	1	Transfer suspended
RSPI_EVT_ERR_MODE_FAULT	2	Mode fault error
RSPI_EVT_ERR_READ_OVF	3	Overflow error
RSPI_EVT_ERR_PARITY	4	Parity error
RSPI_EVT_ERR_UNDEF	5	Unknown error

Table 6.10 Constants for the Sample Program

Constant Name	Setting Value	Description
WM8978_BITRATE_5MBPS	5000000	RSPI transfer bit rate: 5 Mbps
WM8978_CMD_BUF_WORD_SIZE	32	WM8978 command buffer size (in words) The size of the buffer for storing word data to be transferred to the WM8978 (in words)
RSPI_WRITE_CMD	0x0f04	RSPI Write command [b0] = 0b Specifies data sampling on odd edge and data variation on even edge [b1] = 0b Specifies RSPCK = Low when idle [b3-2] = 01b Specifies 1/2 frequency division of the base bit rate [b11-b8] = 1111b Specifies 16 bits for the RSPI data length [b12] = 0 Specifies MSB-first
CMCR0_REG_DEF_VAL	0x0000	Default value of the CMCR0 register
CMCOR0_REG_DEF_VAL	0xFFFF	Default value of the CMCOR0 register
SSL_WAIT_TIME_10US	94	10 microseconds in the compare match timer

6.6 Structures, Unions, and Enumerated Types

Table 6.11 to Table 6.14 show the structures, unions, and enumerated types for the sample code.

Table 6.11 Structures and Unions for the RSPI Sample Driver

Structure/Union Definition	Outline	Definition File
rspi_callback_data_t	RSPI callback information (handle, error details)	r_rspi_rx_if.h
rspi_chnl_settings_t	RSPI setting information (3-line/4-line system, master/slave, bit rate)	r_rspi_rx_if.h
rspi_cmd_baud_t	Bit rate information	r_rspi_rx_if.h
rspi_cmd_setregs_t	Register information set in the R_RSPI_Control function	r_rspi_rx_if.h
rspi_command_word_t	SPCMD register setting information	r_rspi_rx_if.h
rspi_tcb_t	Transmission/reception control information for the RSPI sample driver: Transmission/reception access size, the number of sent and received data items, and transmitted and received information	r_rspi_rx.c
rspi_ctrl_reg_values_t	RSPI register information SPCR, SSLP, SPPCR, SPSCR, SPBR, SPDCR, SPCKD, SSLND, SPND, SPCR2	r_rspi_rx.c
rspi_config_block_t	Control information for the RSPI sample driver: Channels and callback	r_rspi_rx_private.h

Table 6.12 Structure/Union Members for the RSPI Sample Driver (1 / 3)

Structure/Union Definition	Member		Description
rspi_callback_data_t	rspi_handle_t handle	uint8_t channel	Channel specification Specify 0, 1, 2, or 3.
		uint8_t current_slave	The number of currently allocated slaves
		bool rspi_chnl_opened	Channel open state true: Already opened false: Not open
		void (*pcallback)(void *pcbdat)	Callback upon the completion of reception
		void (*pcallbacktx)(void)	Callback upon the completion of transmission
	rspi_evt_t event_code	enum	See Table 6.14.
rspi_chnl_settings_t	rspi_interface_mode_t gpio_ssl	enum	See Table 6.14.
	rspi_master_slave_mode_t master_slave_mode	enum	See Table 6.14.
	uint32_t bps_target	–	Target bit rate Specify a value other than 0.
rspi_cmd_baud_t	uint32_t bps_target	–	Target bit rate Specify a bit rate less than or equal to 1/2 frequency division of the SERICLK (150 MHz, 120 MHz).
rspi_cmd_setregs_t	uint8_t sslp_val	–	SSLP register value
	uint8_t sppcr_val	–	SPPRC register value
	uint8_t spckd_val	–	SPCKD register value
	uint8_t sslnd_val	–	SSLND register value
	uint8_t spnd_val	–	SPND register value
	uint8_t spcr2_val	–	SPCR2 register value

Table 6.12 Structure/Union Members for the RSPI Sample Driver (2 / 3)

Structure/Union Definition	Member		Description
rspi_command_word_t	union uint16_t word	–	SPCMD register value (union)
	union	rspi_spcmd_cpha_t cpha	RSPCK phase setting bit 0: Specifies data sampling on odd edge and data variation on even edge 1: Specifies data sampling on even edge and data variation on odd edge
		rspi_spcmd_cpol_t cpol	RSPCK polarity setting bit 0: Specifies RSPCK = Low when idle 1: Specifies RSPCK = High when idle
		rspi_spcmd_br_div_t br_div	Bit rate division setting bits 00b: Specifies the base bit rate 01b: Specifies 1/2 frequency division of the base bit rate 10b: Specifies 1/4 frequency division of the base bit rate 11b: Specifies 1/8 frequency division of the base bit rate
		rspi_spcmd_ssl_assert_t ssl_assert	SSL signal assertion setting bits 00b: SSLy0 (y = 0, 1, 2, 3) 01b: SSLy1 (y = 0 or 1 only) 10b: SSL02 11b: SSL03
		rspi_spcmd_ssl_negation_t ssl_negate	SSL signal level hold bit 0: Negates the SSL signal upon the completion of transfer 1: Holds the SSL signal level from when transfer finishes until next access starts
		rspi_spcmd_bit_length_t bit_length	RSPI data length setting bits 0100b to 0111b: 8bit 1000b: 9bit 1001b: 10bit 1010b: 11bit 1011b: 12 bit 1100b: 13bit 1101b: 14bit 1110b: 15bit 1111b: 16bit 0000b: 20bit 0001b: 24bit 0001b,0011b: 32bit
		rspi_spcmd_bit_order_t bit_order	Specifies RSPI MSB- or LSB-first 0: MSB-first 1: LSB-first
		rspi_spcmd_spnden_t next_delay	RSPI next-access delay enable bit 0: Sets the next-access delay to 1 RSPCK clock + 2 SERICLK clocks 1: Sets the next-access delay to the value of the RSPI next-access delay register (SPND)
		rspi_spcmd_slnden_t ssl_neg_delay	SLL negation delay setting enable bit 0: Sets the SSL negation delay to 1 RSPCK clock 1: Sets the SSL negation delay to the value of the RSPI slave select negate delay register (SSLND)
		rspi_spcmd_sckden_t clock_delay	RSPCK delay setting enable bit 0: Sets the RSPCK delay to 1 RSPCK clock 1: Sets the RSPCK delay to the value of the RSPI clock delay register (SPCKD)

Table 6.12 Structure/Union Members for the RSPI Sample Driver (3 / 3)

Structure/Union Definition	Member		Description
rspi_tcb_t	void *psrc	–	Pointer to the buffer for storing the data to be sent to the WM8978 Supported transmission data length: 8 bits, 16 bits, and 32 bits
	void *pdest;	–	Pointer to the buffer for storing the data received from the WM8978 Supported reception data length: 8 bits, 16 bits, and 32 bits
	uint16_t tx_count	–	Transmission data counter
	uint16_t rx_count	–	Reception data counter
	uint16_t xfr_length	–	Number of transmitted and received data items Total number of data items regardless of the data length (8, 16, and 32 bits)
	uint8_t bytes_per_transfer	–	Data length specification One of the following values can be specified: 8, 16, and 32 bits The value can be specified in rspi_command_word_t.
	bool do_rx_now	–	Specifying whether to enable or disable received data true: Enables received data false: Disables received data
	bool do_tx	–	Transmit mode status true: Transmit mode false: Non-transmit mode
	rspi_operation_t transfer_mode	enum	See Table 6.14.
rspi_ctrl_reg_values_t	uint8_t spcr_val	–	SPCR register value
	uint8_t sslp_val	–	SSLP register value
	uint8_t sppcr_val	–	SPPCR register value
	uint8_t spscr_val	–	SPSCR register value
	uint8_t spbr_val	–	SPBR register value
	uint8_t spdcr_val	–	SPDCR register value
	uint8_t spckd_val	–	SPCKD register value
	uint8_t sslnd_val	–	SSLND register value
	uint8_t spnd_val	–	SPND register value
	uint32_t spcr2_val	–	SPCR2 register value
rspi_config_block_t	uint8_t channel	–	Channel specification Specify 0, 1, 2, or 3.
	uint8_t current_slave	–	The number of currently allocated slaves
	bool rspi_chnl_opened	–	Channel open state true: Already opened false: Not open
	void (*pcallback)(void *pcbdat)	–	Callback upon the completion of reception
	void (*pcallbacktx)(void)	–	Callback upon the completion of transmission

Table 6.13 Enumerated Types for the RSPI Sample Driver

Enumerated Type Definition	Outline	Definition File
rspi_err_t	RSPI API error code	r_rspi_rx_if.h
rspi_evt_t	Transmission/reception error code	r_rspi_rx_if.h
rspi_interface_mode_t	SPI 3-line/4-line system selection	r_rspi_rx_if.h
rspi_master_slave_mode_t	Master/slave selection	r_rspi_rx_if.h
rspi_cmd_t	R_RSPI_Control command information	r_rspi_rx_if.h
rspi_spcmd_cpha_t	SPCMD register CPHA (odd/even edge) information	r_rspi_rx_if.h
rspi_spcmd_cpol_t	SPCMD register CPOL (clock polarity) information	r_rspi_rx_if.h
rspi_spcmd_br_div_t	SPCMD register BRDV (bit rate division) information	r_rspi_rx_if.h
rspi_spcmd_ssl_assert_t	SPCMD register SSL (SSL signal assertion) information	r_rspi_rx_if.h
rspi_spcmd_ssl_negation_t	SPCMD register SSLK (SSL signal level hold) information	r_rspi_rx_if.h
rspi_spcmd_bit_length_t	SPCMD register SPB (RSPI data length) information	r_rspi_rx_if.h
rspi_spcmd_bit_order_t	SPCMD register LSBF (MSB/LSB-first) information	r_rspi_rx_if.h
rspi_spcmd_spnden_t	SPCMD register SPNDEN (RSPI next-access delay enable) information	r_rspi_rx_if.h
rspi_spcmd_slnden_t	SPCMD register SLNDEN (RSPI negation delay enable) information	r_rspi_rx_if.h
rspi_spcmd_sckden_t	SPCMD register SCKDEN (RSPCK delay setting enable) information	r_rspi_rx_if.h
rspi_operation_t	Transmit/receive mode	r_rspi_rx.c

Table 6.14 Enumerated Type Details for the RSPI Sample Driver (1 / 2)

Enumerated Type Definition	List	Description
rspi_err_t	RSPI_SUCCESS	RSPI API error code 0: Normal termination
	RSPI_ERR_BAD_CHAN	1: Channel number error
	RSPI_ERR_CH_NOT_OPENED	2: Open error
	RSPI_ERR_CH_NOT_CLOSED	3: Closing error
	RSPI_ERR_UNKNOWN_CMD	4: Command error
	RSPI_ERR_INVALID_ARG	5: Parameter error
	RSPI_ERR_ARG_RANGE	6: Parameter range error
	RSPI_ERR_NULL_PTR	7: Null pointer
	RSPI_ERR_LOCK	8: Lock error
	RSPI_ERR_UNDEF	9: Unknown error
	RSPI_SUCCESS_ALL_TX_COMPLETE	10: Normal termination (all data transfers completed)
rspi_evt_t	RSPI_EVT_TRANSFER_COMPLETE	RSPI driver internal information 0: Transfer complete
	RSPI_EVT_TRANSFER_ABORTED	1: Transfer suspended
	RSPI_EVT_ERR_MODE_FAULT	2: Mode fault error
	RSPI_EVT_ERR_READ_OVF	3: Read overflow error
	RSPI_EVT_ERR_PARITY	4: Parity error
	RSPI_EVT_ERR_UNDEF	5: Unknown error
rspi_interface_mode_t	RSPI_IF_MODE_3WIRE	RSPI 3-line/4-line system specification 0x01: RSPI 3-line system (b0 = 1)
	RSPI_IF_MODE_4WIRE	0xFE: RSPI 4-line system (b0 = 0)
rspi_master_slave_mode_t	RSPI_MS_MODE_MASTER	RSPI master/slave specification 0x08: Master (b3 = 1)
	RSPI_MS_MODE_SLAVE	0xF7: Slave (b3 = 0)
rspi_cmd_t	RSPI_CMD_SET_BAUD	RSPI Control command 1: Baud rate setting command
	RSPI_CMD_ABORT	2: Transmission/reception abort command
	RSPI_CMD_SETREGS	3: RSPI register setting command
	RSPI_CMD_UNKNOWN	4: Unknown command
rspi_spcmd_cpha_t	RSPI_SPCMD_CPHA_SAMPLE_ODD	RSPCK phase setting 0: Specifies data sampling on odd edge and data variation on even edge
	RSPI_SPCMD_CPHA_SAMPLE_EVEN	1: Specifies data sampling on even edge and data variation on odd edge
rspi_spcmd_cpol_t	RSPI_SPCMD_CPOL_IDLE_LO	RSPCK polarity setting 0: Specifies RSPCK = Low when idle
	RSPI_SPCMD_CPOL_IDLE_HI	1: Specifies RSPCK = High when idle
rspi_spcmd_br_div_t	RSPI_SPCMD_BR_DIV_1	Bit rate division setting 0: Specifies the base bit rate
	RSPI_SPCMD_BR_DIV_2	1: Specifies 1/2 frequency division of the base bit rate
	RSPI_SPCMD_BR_DIV_4	2: Specifies 1/4 frequency division of the base bit rate
	RSPI_SPCMD_BR_DIV_8	3: Specifies 1/8 frequency division of the base bit rate

Table 6.14 Enumerated Type Details for the RSPI Sample Driver (2 / 2)

Enumerated Type Definition	List	Description
rspi_spcmd_ssl_assert_t	RSPI_SPCMD_ASSERT_SSL0	SSL signal assertion setting 0: SSLy0 (y = 0,1,2,3)
	RSPI_SPCMD_ASSERT_SSL1	1: SSLy1 (y = 0 or 1 only)
	RSPI_SPCMD_ASSERT_SSL2	2: SSL02
	RSPI_SPCMD_ASSERT_SSL3	3: SSL03
rspi_spcmd_ssl_negation_t	RSPI_SPCMD_SSL_NEGATE	SSL signal level hold setting 0: Negates the SSL signal upon the completion of transfer
	RSPI_SPCMD_SSL_KEEP	1: Holds the SSL signal level from when transfer finishes until next access starts
rspi_spcmd_bit_length_t	RSPI_SPCMD_BIT_LENGTH_8	RSPI data length setting bits 0x7: 8bit
	RSPI_SPCMD_BIT_LENGTH_9	0x8: 9bit
	RSPI_SPCMD_BIT_LENGTH_10	0x9: 10bit
	RSPI_SPCMD_BIT_LENGTH_11	0xA: 11bit
	RSPI_SPCMD_BIT_LENGTH_12	0xB: 12bit
	RSPI_SPCMD_BIT_LENGTH_13	0xC: 13 bit
	RSPI_SPCMD_BIT_LENGTH_14	0xD: 14 bit
	RSPI_SPCMD_BIT_LENGTH_15	0xE: 15 bit
	RSPI_SPCMD_BIT_LENGTH_16	0xF: 16 bit
	RSPI_SPCMD_BIT_LENGTH_20	0x0: 20 bit
	RSPI_SPCMD_BIT_LENGTH_24	0x1: 24 bit
	RSPI_SPCMD_BIT_LENGTH_32	0x3: 32 bit
rspi_spcmd_bit_order_t	RSPI_SPCMD_ORDER_MSB_FIRST	RSPI MSB/LSB-first specification 0: MSB-first
	RSPI_SPCMD_ORDER_LSB_FIRST	1: LSB-first
rspi_spcmd_spnden_t	RSPI_SPCMD_NEXT_DLY_1	RSPI next-access delay enable setting 0: Sets the next-access delay to 1 RSPCK clock + 2 SERICLK clocks
	RSPI_SPCMD_NEXT_DLY_SSLND	1: Sets the next-access delay to the value of the RSPI next-access delay register (SPND)
rspi_spcmd_slnden_t	RSPI_SPCMD_SSL_NEG_DLY_1	SSL negation delay setting enable setting 0: Sets the SSL negation delay to 1 RSPCK clock
	RSPI_SPCMD_SSL_NEG_DLY_SSLND	1: Sets the SSL negation delay to the value of the RSPI slave select negate delay register (SSLND)
rspi_spcmd_sckden_t	RSPI_SPCMD_CLK_DLY_1	RSPCK delay setting enable setting 0: Sets the RSPCK delay to 1 RSPCK clock
	RSPI_SPCMD_CLK_DLY_SPCKD	1: Sets the RSPCK delay to the value of the RSPI clock delay register (SPCKD)
rspi_operation_t	RSPI_DO_TX	Transmit/receive mode specification 1: Transmit mode
	RSPI_DO_RX	2: Receive mode
	RSPI_DO_TX_RX	3: Transmit/receive mode

6.7 Global Variables

Table 6.15 shows the Global Variables for the sample code.

Table 6.15 Global Variables

Type	Variable Name	Description	Function
static struct rspi_config_block_s	g_rspi_handles	Control information for the RSPI sample driver: Channels and callback	R_RSPI_Open() R_RSPI_Contro()l rspe_write_read_common() R_RSPI_Close() rspi_tx_rx_common() rspi_spti1_isr() rspi_spei_isr_common()
static volatile uint32_t	gb_rxdata[]	RSPI register read buffer	R_RSPI_ReadContinue() R_RSPI_WriteContinue rspi_tx_rx_common()
static struct rspi_tcb_s	g_rspi_tcb	Transmission/reception control information for the RSPI sample driver: Transmission/reception access size, the number of sent and received data items, and transmitted and received information	rspi_write_read_common() R_RSPI_WriteContinue() rspi_tx_rx_common()
static rspi_callback_data_t	g_rspi_cb_data	RSPI callback information (handle, error details)	R_RSPI_Cntrol() rspi_tx_rx_common() rspi_spei_isr_common()
static rspi_ctrl_reg_values_t	g_ctrl_reg_values	RSPI register information SPCR, SSLP, SPPCR, SPSCR, SPBR, SPDCR, SPCKD, SSLND, SPND, SPCR2	R_RSPI_Open() R_RSPI_Contril() rspi_baud_set()
static const uint32_t	g_unused_bits_masks	Mask information for the transfer rate setting register	rspi_write_read_common()
static volatile struct st_rspi	(* const pg_rspi_channels[])	RSPI register access variables	R_RSPI_Open() R_RSPI_Control() R_RSPI_ReadContinue() R_RSPI_WriteContinue() R_RSPI_Close() rspe_write_read_common() rspi_baud_set() rspi_tx_rx_common() rspi_spei_isr_common()
static const wm8978_write_data_t	wm8978_init_data[]	WM8978 initialization data	wm8978_init()
static bool	tx_cmp	Transfer completion information	R_RSPI_Open_Support R_RSPI_Control_Support R_RSPI_Write_Support R_RSPI_Close_Support
static rspi_chnl_settings_t	rspi_config	RSPI setting information (3-line/4-line system, master/slave, bit rate)	R_RSPI_Open_Support
static rspi_handle_t	rspi_handle	Control information for the RSPI sample driver: Channels and callback	R_RSPI_Open_Support R_RSPI_Control_Support R_RSPI_Write_Support R_RSPI_Close_Support

6.8 Functions

Table 6.16 shows the Functions for the sample code.

Table 6.16 Functions

Function Name	Page Number
R_RSPI_Open	23
R_RSPI_Control	24
R_RSPI_Write	25
R_RSPI_WriteContinue	25
R_RSPI_Close	26
R_RSPI_GetVersion	26
rspi_tx_callback	26
wm8978_init	27
wm8978_output_volume_set	27
wm8978_input_volume_set	28
wm8978_shutdown	28

6.9 Specifications of Functions

This section presents the specifications of the functions for the sample code.

6.9.1 R_RSPI_Open (RSPI Sample Driver)

R_RSPI_Open		
Synopsis	Opening the RSPI module	
Header		
Declaration	rspi_err_t R_RSPI_Open(uint8_t channel, rspi_chnl_settings_t *pconfig, void (*pcallback)(void *pcbdat), void (*pcallbacktx)(void), rspi_handle_t *phandle);	
Description	This function opens the RSPI module for the specified RSPI channel, sets RSPI options, and returns the handle (to the opened RSPI module) as an argument. It also returns the result of open processing as a return value.	
Arguments	uint8_t channel	: Specifies an RSPI channel
	rspi_chnl_settings_t *pconfig	: Specifies an RSPI option
	void (*pcallback)(void *pcbdat)	: Specifies the callback function to be executed upon the completion of reception
	void (*pcallbacktx)(void)	: Specifies the callback function to be executed upon the completion of transmission
	rspi_handle_t *phandle	: Returns the handle to the opened RSPI module
Return values	RSPI_SUCCESS(0)	: Success - The RSPI module initialized successfully
	RSPI_ERR_BAD_CHAN(1)	: Failure - Invalid channel
	RSPI_ERR_CH_NOT_CLOSED(3)	: Failure - Already initialized
	RSPI_ERR_ARG_RANGE(6)	: Failure - Invalid argument range
	RSPI_ERR_NULL_PTR(7)	: Failure - Parameter not specified
	RSPI_ERR_LOCK(8)	: Failure - The RSPI cannot lock

6.9.2 R_RSPI_Control (RSPI Sample Driver)

R_RSPI_Control		
Synopsis	Setting parameters for the RSPI module	
Header		
Declaration	rspi_err_t R_RSPI_Control(rspi_handle_t handle, rspi_cmd_t cmd, void *pcmd_data)	
Description	This function sets up the specified three commands (shown below) for the specified handle to the RSPI module: • Baud rate setting command Sets a baud rate • Command quit command Quits a command that is being executed • RSPI register setting command Sets up the following registers according to the specified parameter: SPCKD SPCR2 SPND SPPCR SSLND SSLP	
Arguments	rspi_handle_t handle	: Specifies the handle to the RSPI module
	rspi_cmd_t cmd	: Specifies the command for the module
	void *pcmd_data	: Specifies the parameter for the module
Return values	RSPI_SUCCESS(0)	: Success - The parameters for the RSPI module set successfully
	RSPI_ERR_CH_NOT_OPENED(2)	: Failure - The channel cannot open
	RSPI_ERR_ARG_RANGE(6)	: Failure - Invalid argument range
	RSPI_ERR_NULL_PTR(7)	: Failure - Parameter not specified
	RSPI_ERR_LOCK(8)	: Failure - The RSPI cannot lock
	RSPI_ERR_UNKNOWN_CMD(9)	: Failure - Unknown command specified

6.9.3 R_RSPI_Write (RSPI Sample Driver)

R_RSPI_Write		
Synopsis	Starting transmissions with the RSPI module	
Header		
Declaration	rspi_err_t R_RSPI_Write(rspi_handle_t handle, rspi_command_word_t spcmd_command_word, void *psrc, uint16_t length)	
Description	This function starts data transmissions by setting the specified transmission data for the specified handle to the RSPI module.	
Arguments	rspi_handle_t handle	: Specifies the handle to the RSPI module
	rspi_command_word_t spcmd_command_word	: Specifies the value for PMCMD
	void *psrc	: Specifies the pointer to the data to be transferred
	uint16_t length	: Specifies the length of the data to be transferred
Return values	RSPI_SUCCESS(0)	: Success - Transmissions using the RSPI module started successfully
	RSPI_ERR_CH_NOT_OPENED(2)	: Failure - The channel cannot open
	RSPI_ERR_LOCK(8)	: Failure - The RSPI cannot lock

6.9.4 R_RSPI_WriteContinue (RSPI Sample Driver)

R_RSPI_WriteContinue		
Synopsis	Controlling transmission for the RSPI module	
Header		
Declaration	rspi_err_t void R_RSPI_WriteContinue(rspi_handle_t handle)	
Description	This function verifies the continuity of data transmission and the completion of all data transmissions for the specified handle to the RSPI module.	
Arguments	uint8_t channel	: Specifies an RSPI channel
Return values	RSPI_SUCCESS(0)	: Success - Transmission control for the RSPI module completed successfully
	RSPI_ERR_CH_NOT_OPENED(2)	: Failure - The channel not opened
	RSPI_SUCCESS_ALL_TX_COMPLETE(10)	: Success - Transmission control for the RSPI module completed successfully and all data transmissions completed successfully

6.9.5 R_RSPI_Close (RSPI Sample Driver)

R_RSPI_Close		
Synopsis	Closing the RSPI module	
Header		
Declaration	rspi_err_t R_RSPI_Close(rspi_handle_t handle)	
Description	This function closes the specified handle to the RSPI module. It enables the power saving function when closing the handle.	
Arguments	rspi_handle_t handle	: Specifies the handle to the RSPI module
Return values	RSPI_SUCCESS(0) RSPI_ERR_CH_NOT_OPENED(2) RSPI_ERR_NULL_PTR(7)	: Success - The parameters for the RSPI module set successfully : Failure - The channel cannot open : Failure - Parameter not specified

6.9.6 R_RSPI_GetVersion (RSPI Sample Driver)

R_RSPI_GetVersion		
Synopsis	Obtaining version information for the RSPI module	
Header		
Declaration	uint32_t R_RSPI_GetVersion(void)	
Description	This function returns version information for the RSPI module.	
Arguments	rspi_handle_t handle	: Specifies the handle to the RSPI module
Return values	uint32_t	: 0-15bit Major Version : 16-31bit Minor Version

6.9.7 rspi_tx_callback (Sample Program)

rspi_tx_callback		
Synopsis	Executing callback processing upon the completion of transmission	
Header		
Declaration	void rspi_tx_callback(void)	
Description	This function executes callback processing when a transmission completion interrupt occurs. It sets the transmission completion flag for slave selection control.	
Arguments	None	–
Return values	None	–

6.9.8 wm8978_init (Sample Program)

wm8978_init		
Synopsis	Initializing the WM8978	
Header		
Declaration	void wm8978_init(void)	
Description	This function performs the following setup tasks: • Initializing the WM8978 - Disabling the power management functions - ADC/DAC setting - Microphone input setting - Digital filter setting - L/R DAC output setting - Headphone output setting - MCLK setting • Initializing RSPI communications - Disabling power saving mode for the RSPI module	
Arguments	None	–
Return values	None	–

6.9.9 wm8978_output_volume_set (Sample Program)

wm8978_output_volume_set		
Synopsis	Setting the headphone output volume level for the WM8978	
Header		
Declaration	void wm8978_output_volume_set (uint8_t l_vol, uint8_t r_vol)	
Description	This function sets the output volume level of the L/R channel for the WM8978.	
Arguments	uint8_t l_vol	L-channel output volume level ROUT1VOL (6-bit) value of the 0x34 register for the WM8978 Note: This value is set in the lower six bits of the argument.
	uint8_t r_vol	R-channel output volume level LOUT1VOL (6-bit) value of the 0x35 register for the WM8978 Note: This value is set in the lower six bits of the argument.
Return values	None	–

6.9.10 wm8978_input_volume_set (Sample Program)

wm8978_input_volume_set		
Synopsis	Setting the microphone input volume level for the WM8978	
Header		
Declaration	void wm8978_input_volume_set (uint8_t l_vol, uint8_t r_vol)	
Description	This function sets the input volume level of the L/R channel for the WM8978.	
Arguments	uint8_t l_vol	L-channel input volume level INPPGAVOLL (6-bit) value of the 0x2D register for the WM8978 Note: This value is set in the lower six bits of the argument.
	uint8_t r_vol	R-channel input volume level INPPGAVOLR (6-bit) value of the 0x2E register for the WM8978 Note: This value is set in the lower six bits of the argument.
Return values	None	–

6.9.11 wm8978_shutdown (Sample Program)

wm8978_shutdown		
Synopsis	Shutting down the WM8978	
Header		
Declaration	void wm8978_shutdown(void)	
Description	This function performs the following setup tasks: • Shutting down the WM8978 - Enabling the power management functions • Closing RSPI communications - Enabling power saving mode for the RSPI module	
Arguments	None	–
Return values	None	–

6.10 Flowcharts

6.10.1 wm8978_init Processing

Figure 6.2 shows the flowchart for wm8978_init processing.

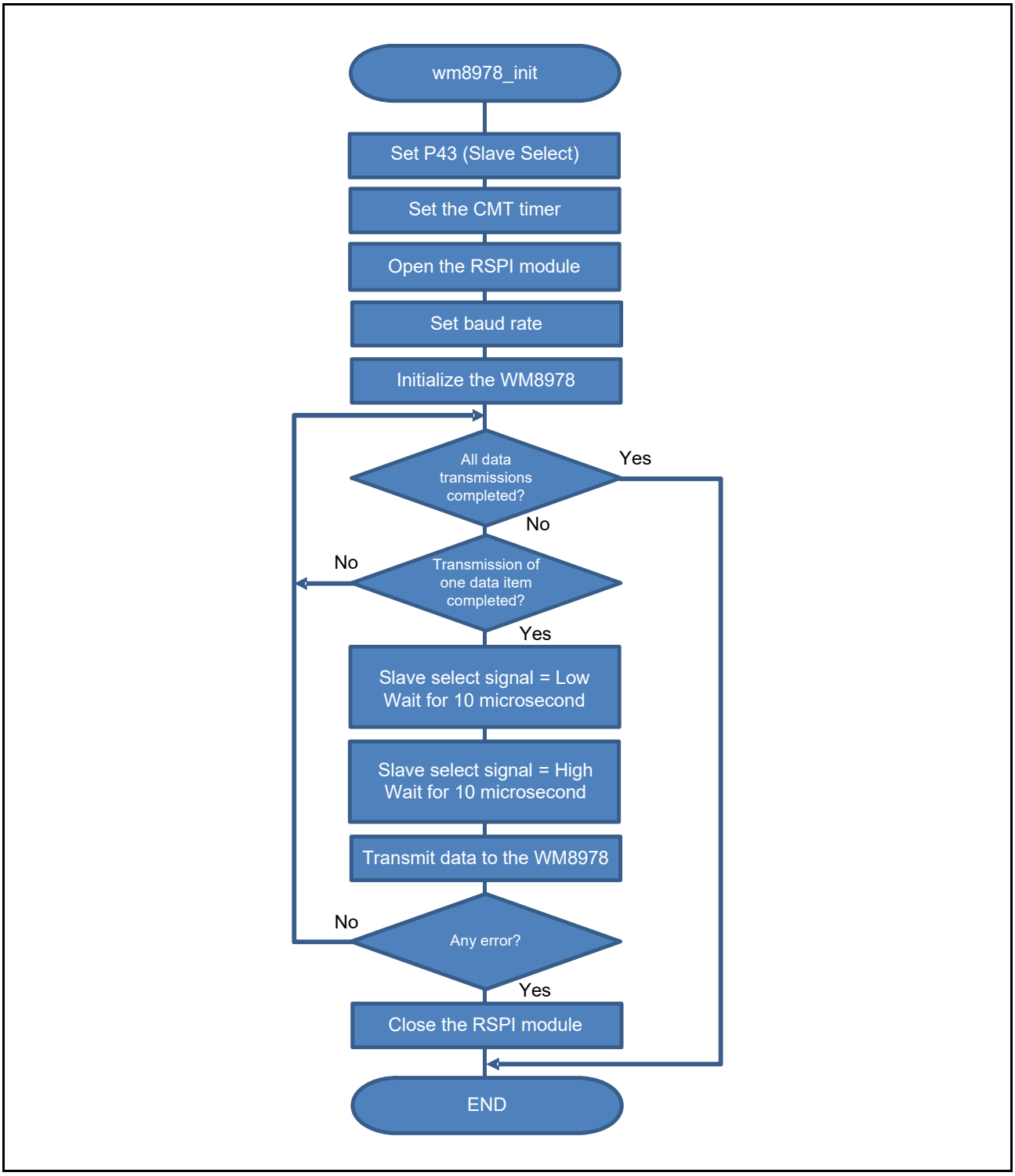


Figure 6.2 wm8978_init Processing

6.10.2 wm8978_output_volume_set Processing

Figure 6.3 shows the flowchart for wm8978_output_volume_set processing.

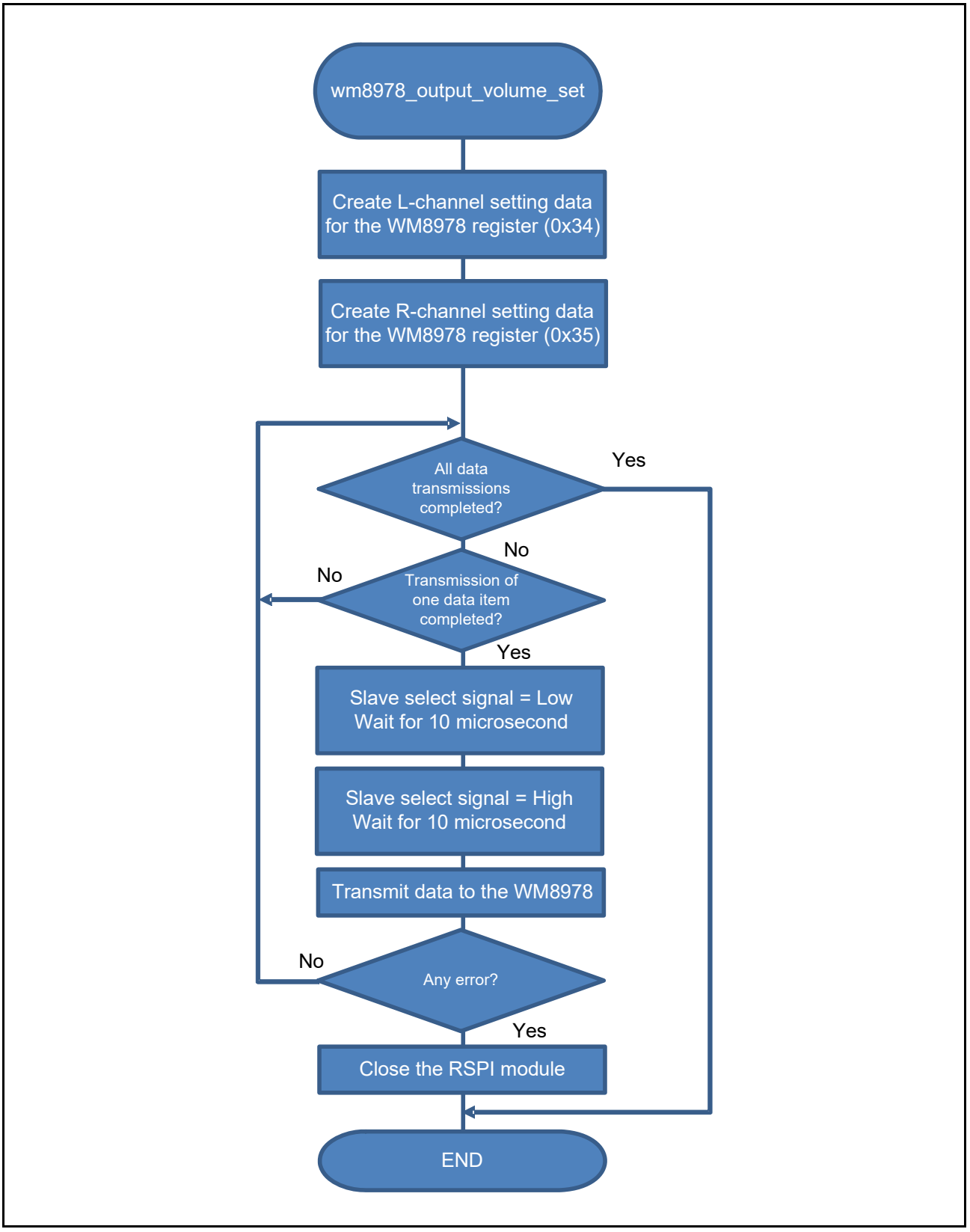


Figure 6.3 wm8978_output_volume_set Processing

6.10.3 wm8978_input_volume_set Processing

Figure 6.4 shows the flowchart for wm8978_input_volume_set processing.

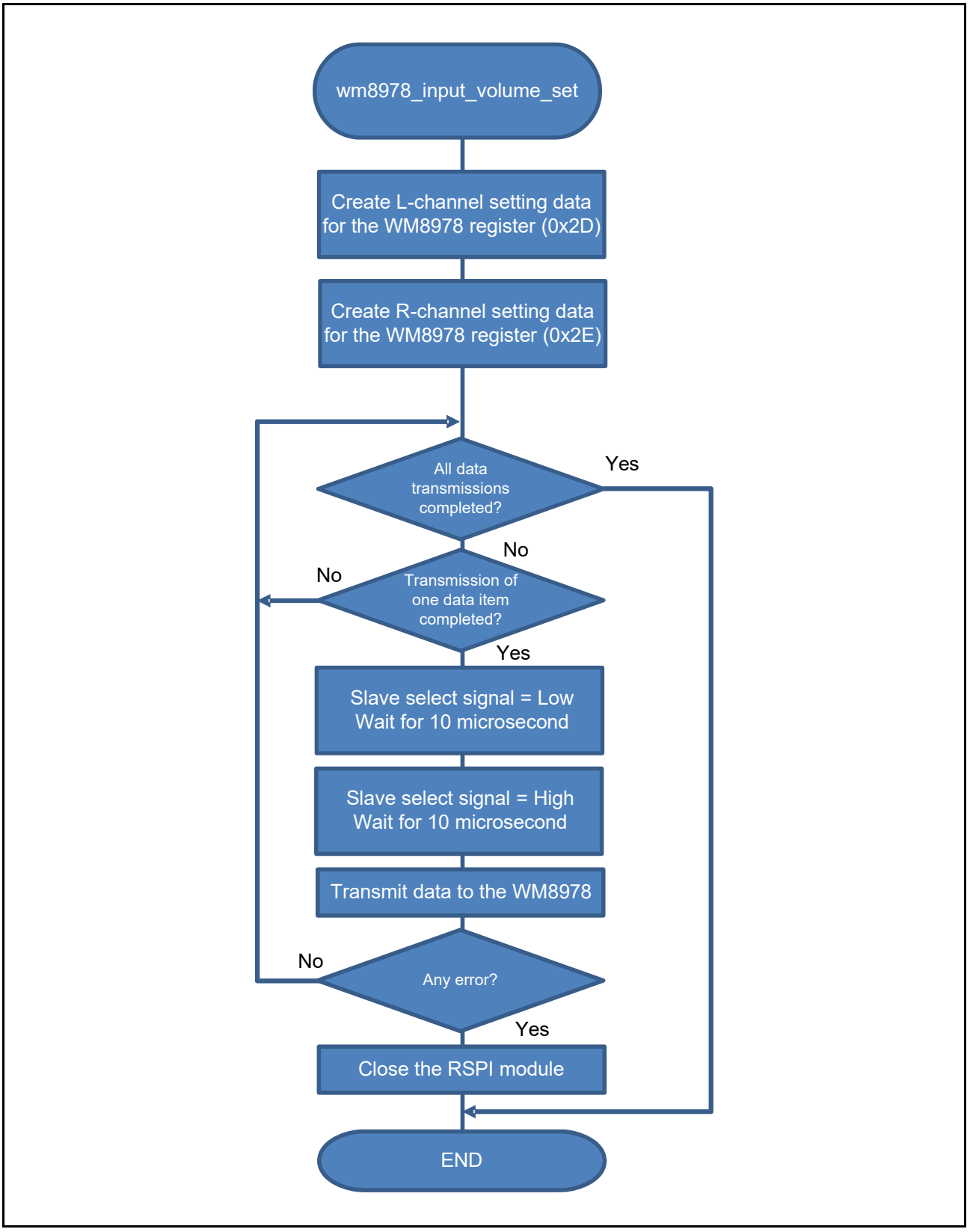


Figure 6.4 wm8978_input_volume_set Processing

6.10.4 wm8978_shutdown Processing

Figure 6.5 shows the flowchart for wm8978_shutdown processing.

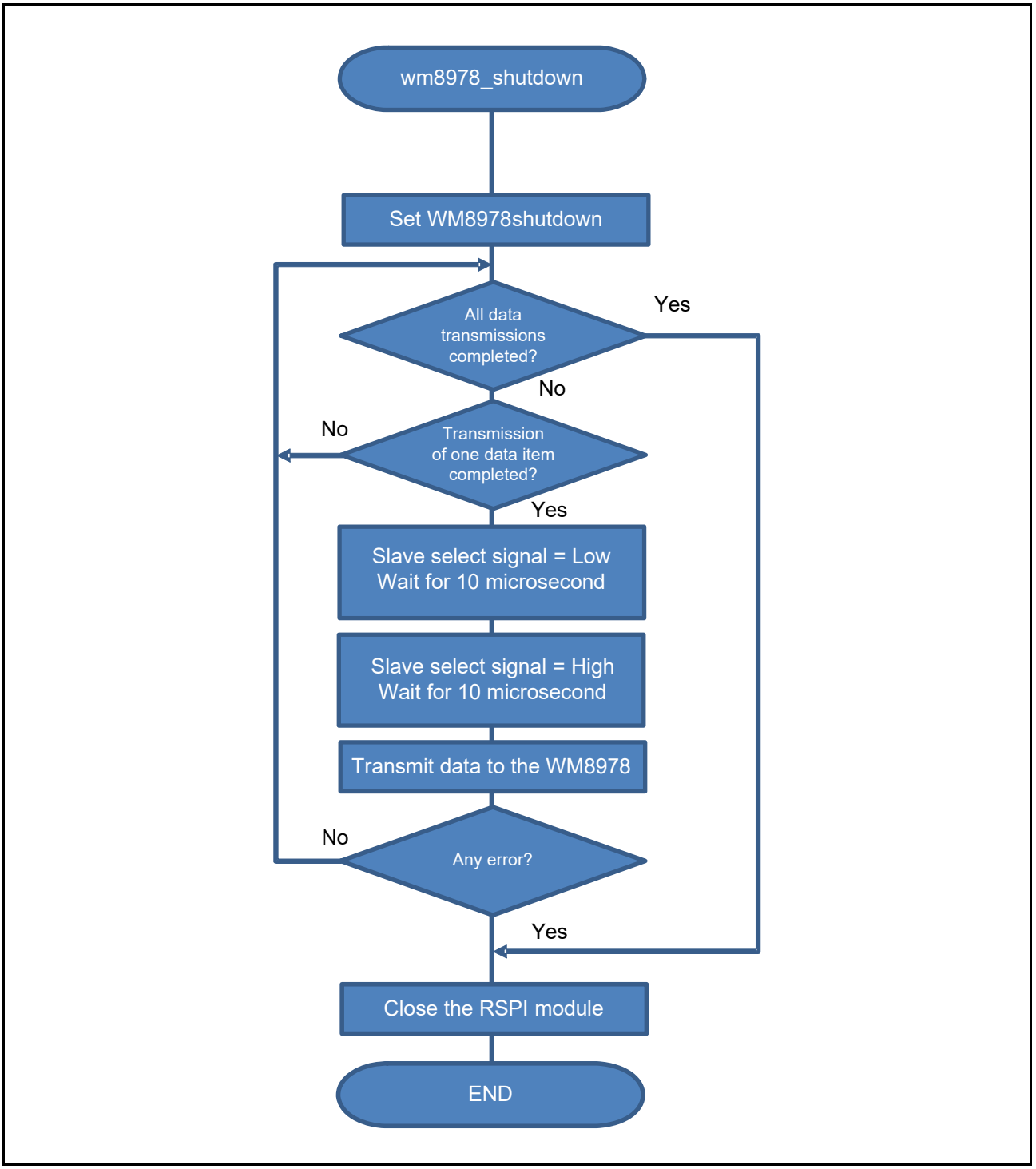


Figure 6.5 wm8978_shutdown Processing

6.11 Commands Used with R_RSPI_Control

The following table lists the commands used with R_RSPI_Control (RSPI Sample Driver)

Table 6.17 Commands

Command	Outline
RSPI_CMD_SET_BAUD	Baud rate setting command
RSPI_CMD_ABORT	RSPI communication abort command
RSPI_CMD_IDLESTATE	RSPI operating state check command
RSPI_CMD_SETREGS	RSPI register setting command

6.11.1 RSPI_CMD_SET_BAUD

RSPI_CMD_SET_BAUD			
Synopsis	Baud rate setting command		
Header	r_rspi_rzt1_if.h		
Description	This command specifies a baud rate for RSPI communications.		
Parameters	rspi_cmd_baud_t *	uint32_t bps_target	Specifies a baud rate for RSPI communications in bits per second (bps) If an invalid value is specified, an error is returned as a return value. Setting example: For 5 Mbps, specify 5000000.
Return values	RSPI_ERR_ARG_RANGE		: Failure - Parameter range error
Remarks	–		

6.11.2 RSPI_CMD_ABORT

RSPI_CMD_ABORT	
Synopsis	RSPI communication abort command
Header	r_rspi_rzt1_if.h
Description	This command forcibly closes RSPI communications. Executing this command performs the following operations: • Suspends the current serial transfers • Sets the output signal pin to high impedance while in slave mode • Initializes the internal state of the RSPI • Clears the RSPI transmit buffer • Disables the RSPI interrupts • Executes the callback function registered in the third argument of R_RSPI_Open
Parameters	None
Return values	RSPI_SUCCESS : Success - The command completed successfully
Remarks	–

6.11.3 RSPI_CMD_IDLESTATE

RSPI_CMD_IDLESTATE		
Synopsis	RSPI operating state check command	
Header	r_rspi_rzt1_if.h	
Description	This command checks the RSPI operating state and returns the result as a return value.	
Parameters	None	
Return values	RSPI_SUCCESS	: Success - The RSPI is in an idle state
	RSPI_ERR_NOTIDLE_STATE	: Failure - The RSPI is transferring data
Remarks	–	

6.11.4 RSPI_CMD_SETREGS

RSPI_CMD_SETREGS			
Synopsis	RSPI register setting command		
Header	r_rspi_rzt1_if.h		
Description	This command sets the specified value in the RSPI register.		
Parameters	rspi_cmd_setregs_t	uint8_t sslp_val	Specifies the value of the RSPI slave select polarity register (SSLP)
		uint8_t sppcr_val	Specifies the value of the RSPI pin control register (SPPCR)
		uint8_t spckd_val	Specifies the value of the RSPI clock delay register (SPCKD)
		uint8_t sslnd_val	Specifies the value of the RSPI slave select negate delay register (SSLND)
		uint8_t spnd_val	Specifies the value of the RSPI next-access delay register (SPND)
		uint8_t spcr2_val	Specifies the value of the RSPI control register 2 (SPCR2)
Return values	RSPI_SUCCESS		: Success - The command completed successfully
Remarks	For details on the registers, refer to RZ/T1 Group User's Manual: Hardware.		
	The register values specified with this command are then reflected to the channels that will be opened by executing the R_RSPI_Open function. These register values are not reflected to the channels that are already open.		

7. Sample Code

Download the sample code from the Renesas Electronics website.

8. Related Documents

- User's Manuals: Hardware
  RZ/T1 Group User's Manual: Hardware
  (Download the latest edition from the Renesas Electronics website.)

  RZ/T1 Evaluation Board RTK7910022C00000BR User's Manual
  (Download the latest edition from the Renesas Electronics website.)

- Technical Update and Technical News
  (Download the latest information from the Renesas Electronics website.)

- User's Manuals: Development Environment
  For the IAR Embedded Workbench® for Arm, download the user's manual from the IAR website.
  (Download the latest edition from the IAR website.)

Website and Support

Renesas Electronics website

http://www.renesas.com/

Inquiries

http://www.renesas.com/inquiry

Revision History	Application Note: RSPI Sample Program

Rev.	Date	Description	
		Page	Summary
0.10	Apr. 02, 2015	—	First Edition issued
1.00	Apr. 10, 2015	—	Only the revision number was changed to be posted on a website.
1.10	Jul. 16, 2015	2. Operating Environment	
		5	Table 2.1 Operating Environment: Description added to Integrated Development Environment
		6. Software	
		10	6.2.4 Required Memory Size: Description and reference added
		10	Table 6.2: Table title and size description were partially amended
		10	Table 6.2 Memory Requirements: Description on the Note, changed
		11	Table 6.3 added
		11	Table 6.4 added
1.20	Dec. 04, 2015	2. Operating Environment	
		5	Table 2.1 Operating Environment: Integrated Development Environment, information partially amended
1.30	May 18, 2017	Introduction	
		1	The description on the bit rate of operating modes, modified
		2. Operating Environment	
		5	Table 2.1 Operating Environment: Integrated Development Environment, modified
		6. Software	
		—	6.2.4 Required Memory Size, deleted
1.40	Jun. 07, 2018	2. Operating Environment	
		5	Table 2.1 Operating Environment: The description on the integrated development environment, modified
		8. Related Documents	
		36	The name of IAR Embedded Workbench, modified

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

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

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0