RL78/G13

Serial Array Unit for 3-Wire Serial I/O (Slave Transmission/Reception) CC-RL

R01AN2711EJ0200
Rev. 2.00
July 01, 2015

Introduction

This application note explains how the serial array unit (SAU) performs slave transmission and reception by 3-wire serial I/O communication (CSI). Using the CSI, this unit receives/transmits data from/to the master.

Target Device

RL78/G13

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.

Contents

1. Specifications

The serial array unit (SAU) described in this application note performs slave transmission and reception by 3-wire serial I/O communication (CSI). As the CSI slave, this unit transmits and receives data in synchronization with the clock signals from the master.

Table 1.1 lists the peripheral functions to be used and their uses. Figure 1.1 presents an overview of CSI operation.

Figures 1.2 to 1.4 show timing charts for explaining the CSI communication.

Table 1.1 Peripheral Functions to be Used and Their Uses

Peripheral Function	Use
Serial array unit 0 channel 0	CSI00 slave transmission/reception

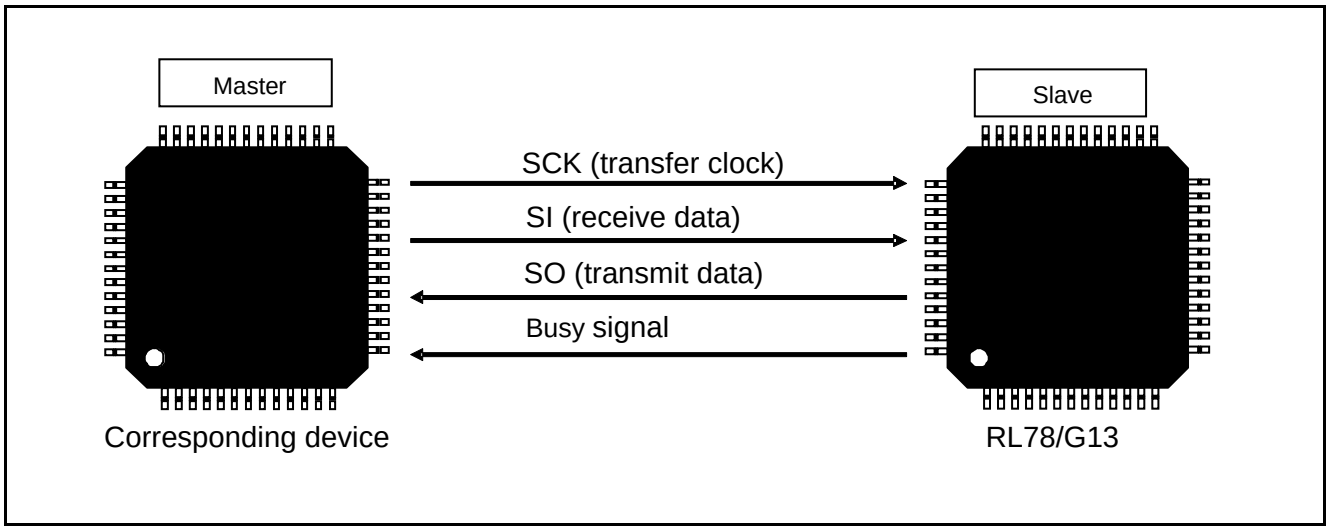


Figure 1.1 Overview of CSI Operation

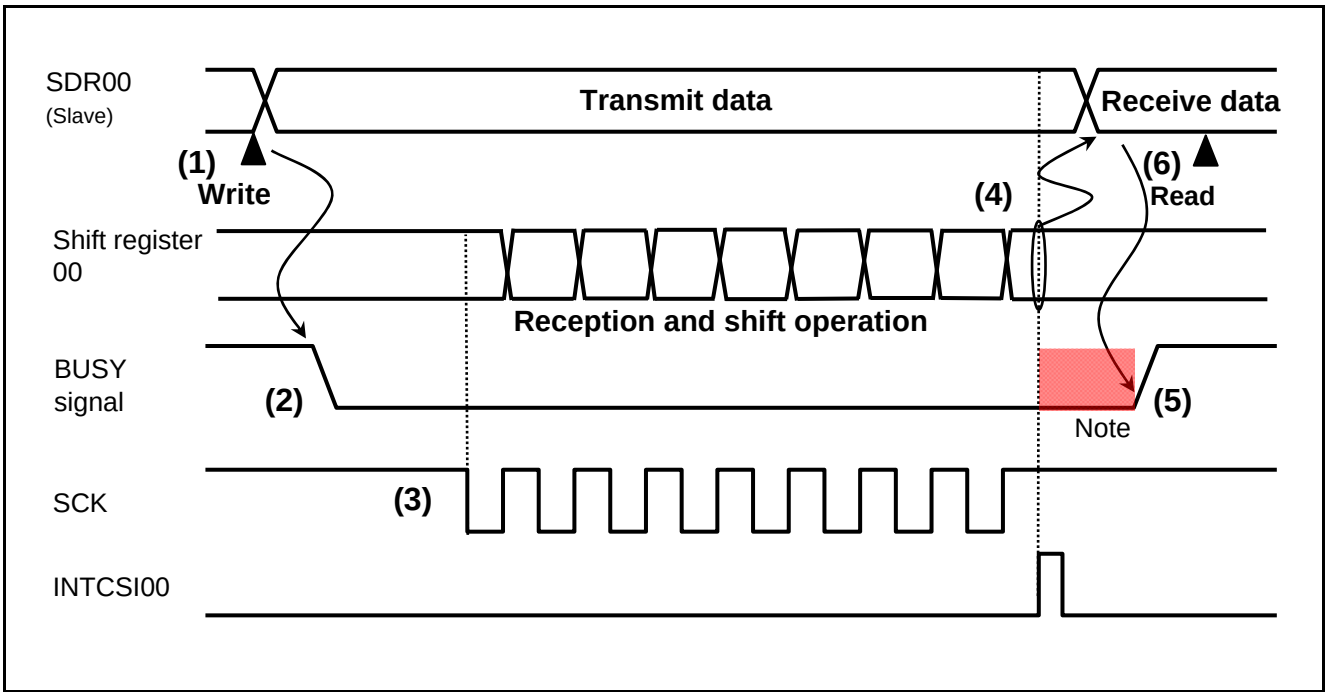


Figure 1.2 Handshake Operation and Communication

(1) [Software processing] Write the transmit data (slave → master) to the SDR00 register.
(2) [Software processing] Make the BUSY signal fall to notify the master that communication is possible.
(3) [Hardware processing] Input serial clock from the master and then enter the communication state.
(4) [Hardware processing] Transfer receive data from the shift register 00 to the SDR00 and then generate a transfer end interrupt.
(5) [Software processing] Raise the BUSY signal to notify the master that transfer is impossible.
(6) [Software processing] Read receive data from the SDR00 register.

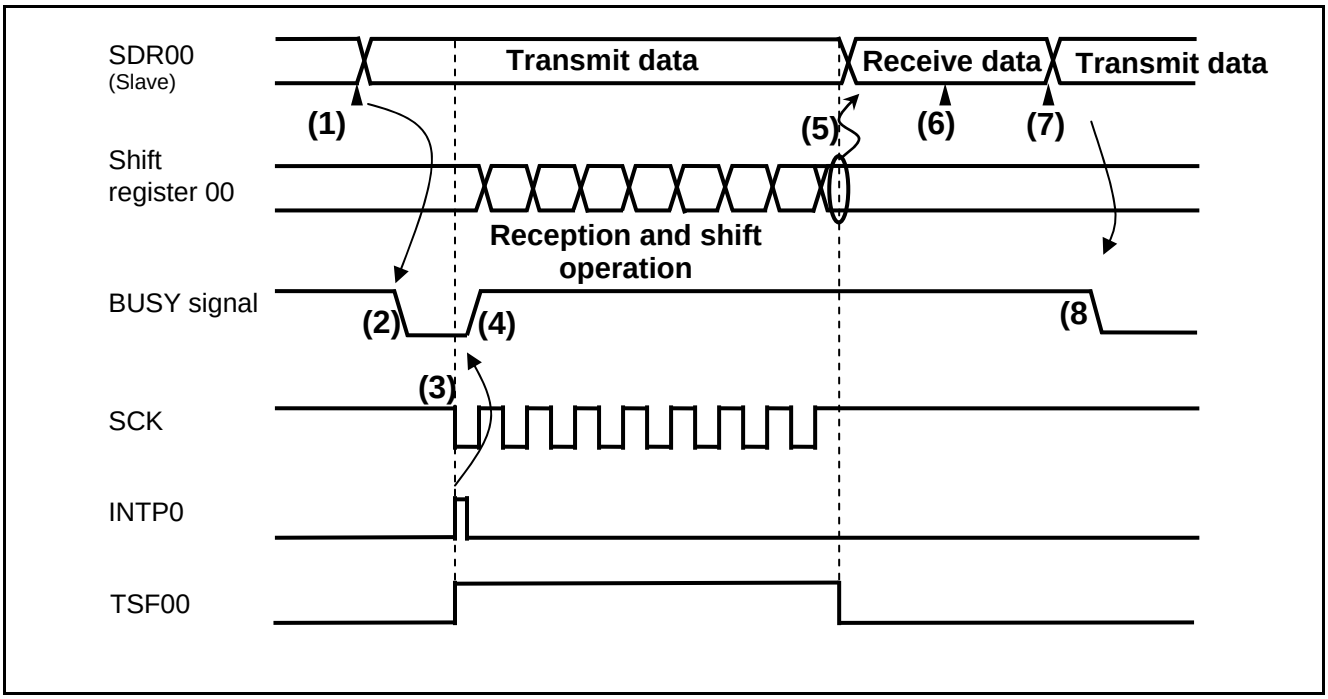


Figure 1.3 Example of BUSY Signal Control by SCK Detection in the Slave

- Serial clock edge detection
  This is an example of setting the BUSY signal to a BUSY state by detecting a falling edge of the serial clock (SCK) through the external interrupt pin (INTP0) to detect the start of communication.

(1) [Software processing in the slave]
Write transmit data (slave → master) to the SDR00 register.
(2) [Software processing in the slave]
Make the BUSY signal fall to notify the master that communication is possible.
(3) [Software processing in the master]
Confirm that communication with the slave is possible and then start communication.
[Hardware processing in the master]
Start transmission and reception and then output serial clock (SCK).
(4) [Software processing in the slave]
Detect a falling edge of SCK on INTP0 to raise the BUSY signal Note.
(5) [Hardware processing in the slave]
After completion of the transfer, store the value of the shift register 00 in the SDR00 register.
(6) [Software processing in the slave]
Read the receive data from the SDR00 register.
(7) [Software processing in the slave]
Write the next transmit data to the SDR00 register.
(8) [Software processing in the slave]
Notify the master that communication with the slave is possible.

Note: The BUSY signal is controlled only to transmit and receive the next data. This signal control has no effect on the data transmission/reception during communication.

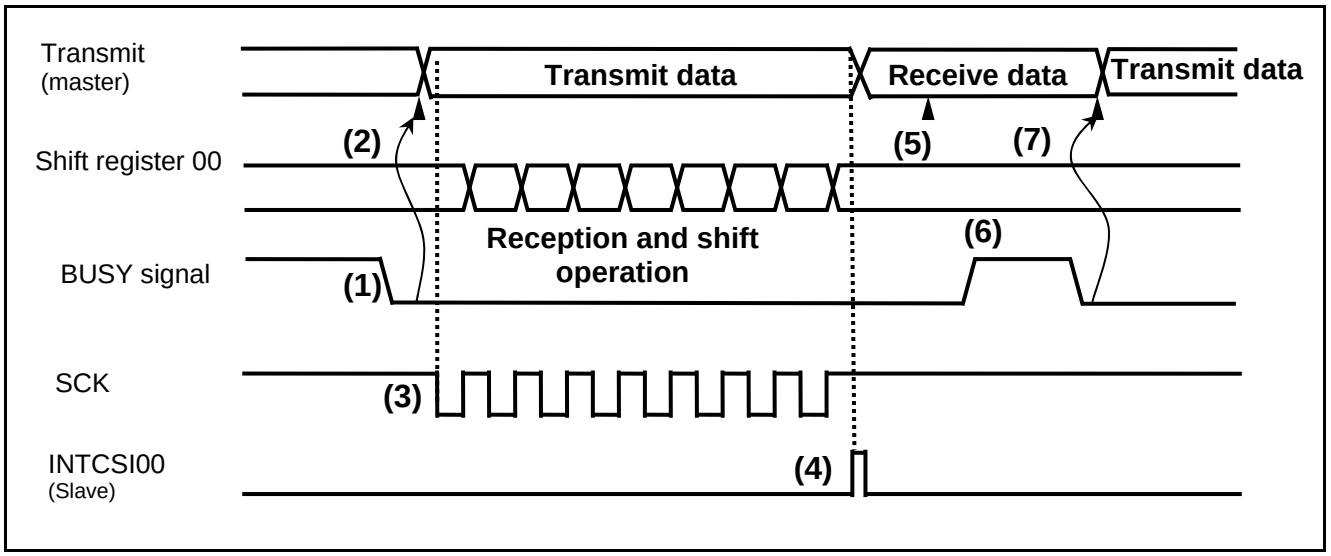


Figure 1.4 Example of BUSY Signal Edge Detection in the Master

- BUSY signal edge detection
  In this example, the master starts communication upon detection of the falling edge of the BUSY signal from the slave.

(1) [Software processing in the slave] Write the next transmit data (slave → master) to the master and make the BUSY signal fall.
(2) [Software processing in the master]
Detect the falling edge of the BUSY signal and write transmit data to the transmit register.
(3) [Hardware processing in the master] Start transmission/reception and then output serial clock (SCK) signals.
(4) [Hardware processing in the slave] After completion of the transfer, set the value of shift register 00 in the SDR00 register and then generate a transfer end interrupt (INTCSI00).
(5) [Software processing in the master]
Read the receive data from the SDR00 register and wait for the falling edge of BUSY signal Note.
(6) [Software processing in the slave]
Raise BUSY signal to read the receive data from the SDR00 register, write the next transmit data to the SDR00 register, and make the BUSY signal fall.
(7) [Software processing in the master]
Detect the falling edge of the BUSY signal and then write the transmit data to the transmit register.

Note: If the BUSY signal is held at the high level for a short period, the software may be unable to detect the edge. In this case, input the BUSY signal to an external interrupt pin (such as the INTP0 pin) so that the hardware detects the edge.

2. Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the table below.

Table2.1 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G13 (R5F100LEA)
Operating frequency	• High-speed on-chip oscillator (HOCO) clock: 32 MHz • CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (Operation is possible over a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVD}): Reset mode which uses 2.81 V (2.76 V to 2.87 V)
Integrated development environment (CS+)	CS+ for CC V3.01.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.01.00 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio V4.0.0.26 from Renesas Electronics Corp.
C compiler (e^2 studio)	CC-RL V1.01.00 from Renesas Electronics Corp.

3. Related Application Note

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

- RL78/G13 Initialization (R01AN2575EJ0100) Application Note
- RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) (R01AN2547EJ0100) Application Note

4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

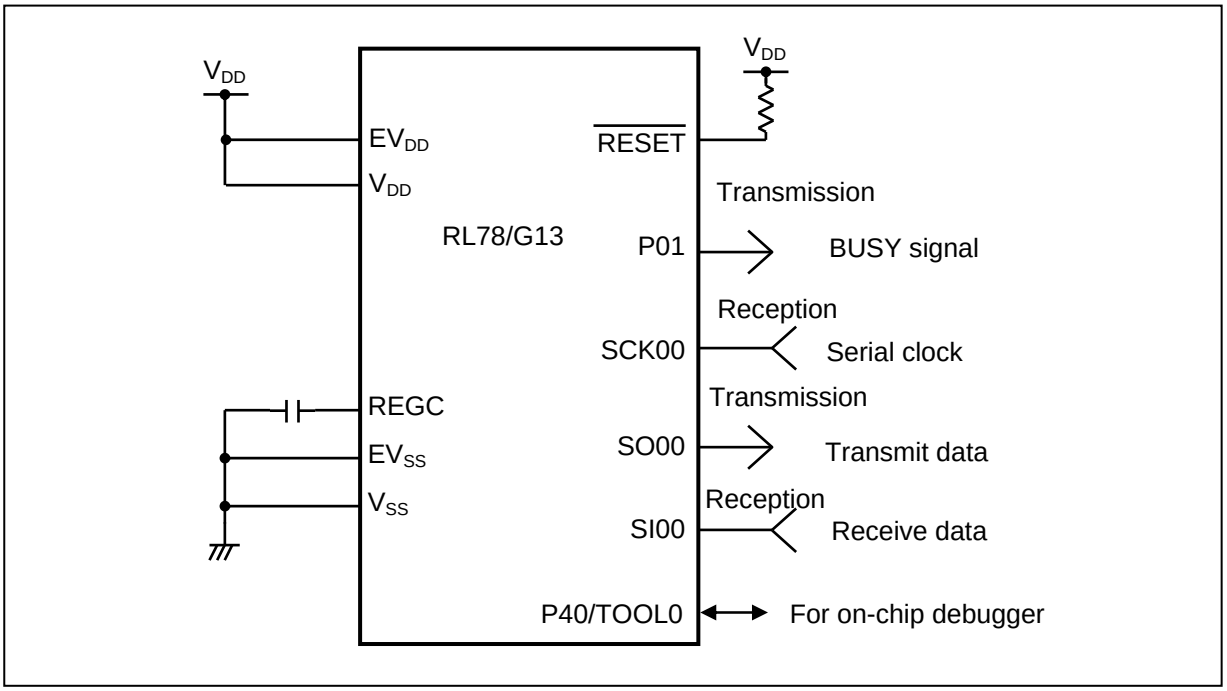


Figure 4.1 Hardware Configuration

Cautions:
1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).
2. Connect any pins whose name begins with EV_{SS} to V_{SS} and any pins whose name begins with EV_{DD} to V_{DD}, respectively.
3. V_{DD} must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

4.2 List of Pins to be Used

Table 4.1 lists the pins to be used and their functions.

Table 4.1 Pins to be Used and Their Functions

Pin Name	I/O	Description
P10/SCK00/SCL00	Input	Serial clock input pin
P11/SI00/RxD0/TOOL RxD/SDA00	Input	Data reception pin
P12/SO00/TxD0/TOOL TxD	Output	Data transmission pin
P01/ANI16/TO00/RxD1	Output	BUSY signal pin

5. Description of the Software

5.1 Operation Outline

The sample program covered in this application note transmits and receives data to and from the corresponding device (master) via the CSI (slave transmission/reception).

(1) Initialize SAU0.
<Conditions for setting>

- Use SAU0 channel 0 as the CSI.
- Select the single transfer mode as the operation mode.
- Select type 1 as the phase between data and clock signals.
- The length of data should be 8 bits.
- A serial transfer end interrupt (INTCSI00) should occur in single transfer mode.
- Use the P10/SCK00 pin for clock input.
- Use the P12/SO00 pin for data output and set the initial output value to 1.
- Use the P11/SI00 pin for data input.
- Enable output for serial communication.

(2) Write transmit data (slave → master) to the SDR00 register.

(3) Set the BUSY signal to low level to notify the master that communication is possible.

(4) Execute a HALT instruction to enter HALT mode and then wait for the occurrence of a transfer end interrupt (INTCSI00).

(5) Update the receive data when a transfer end interrupt (INTCSI00) occurs. Then, output a BUSY state to the BUSY signal pin to cancel HALT mode.

(6) Wait until the serial I/O reenters a state in which communication is possible.

Note: Able to communicate at up to 4MHz. It depends on the delay of the communication line and the master to be connected.

5.2 List of Option Byte Settings

Table 5.1 summarizes the settings of the option bytes.

Table 5.1 Option Byte Settings

Address	Value	Description
000C0H/010C0H	01101110B	Disables the watchdog timer. (Stops counting after the release from the reset state.)
000C1H/010C1H	01111111B	LVD reset mode, 2.81 V (2.76 V to 2.87 V)
000C2H/010C2H	11101000B	HS mode, HOCO: 32 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger.

5.3 List of Constants

Table 5.2 lists the constants that are used in this sample program.

Table 5.2 Constants for the Sample Program

Constant	Setting	Description
_0001_SAU_CH0_DATA_OUTPUT_1	0x0001U	Serial data output setting for SAU0 channel 0
_0001_SAU_CH0_OUTPUT_ENABLE	0x0001U	Setting for enabling output through serial communication on SAU0 channel 0
_0001_SAU_CH0_START_TRG_ON	0x0001U	Setting for starting operation of SAU0 channel 0
_0001_SAU_OVERRUN_ERROR	0x0001U	Overrun error occurrence (SSR)
MD_STATUSBASE	0x00U	Communication status base value
MD_OK	MD_STATUSBASE+ 0x00U	Successful completion
MD_ERRORBASE	0x80U	Communication error status base value
MD_ARGERROR	MD_ERRORBASE+ 01U	Parameter error

5.4 List of Variables

Table 5.3 list the global variables that are used in this sample program.

Table 5.3 Global Variables

Type	Variable Name	Contents	Function Used
unsigned char	g_tx_data	Serial transmit data	main()
unsigned char	g_rx_data	Serial receive data	main()
uint8_t	gp_csi00_rx_address	CSI00 receive buffer address	R_CSI00_Send_Receive() R_CSI00_Interrupt()
uint8_t	gp_csi00_tx_address	CSI00 transmit buffer address	R_CSI00_Send_Receive() R_CSI00_Interrupt()
uint16_t	g_csi00_tx_count	CSI00 transmit data size	R_CSI00_Send_Receive() R_CSI00_Interrupt()

5.5 List of Functions

Table 5.4 lists the global variable that is used by this sample program.

Table 5.4 Functions

Function Name	Outline
R_CSI00_Start	CSI00 operation start processing
R_CSI00_Send_Receive	CSI00 data transmission/reception function
r_csi00_interrupt	CSI00 transfer end interrupt function
r_csi00_callback_receiveend	CSI00 data reception completion processing

5.6 Function Specifications

This section describes the specifications for the functions that are used in the sample code.

[Function Name] R_CSI00_Start	
Synopsis	CSI00 operation start
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	void R_CSI00_Start(void)
Explanation	This function starts SAU0 channel 0 as CSI00 and sets it to a communication standby state.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CSI00_Send_Receive	
Synopsis	CSI00 data transmission/reception function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	MD_STATUS R_CSI00_Send_Receive(uint8_t *tx_buf, uint16_t tx_num, uint8_t *rx_buf)
Explanation	This function assigns the transmit/receive data buffer addresses and sizes that are specified by the arguments to global variables and then starts data transmission.
Arguments	uint8_t *tx_buf : [Transmit data buffer address] uint16_t tx_num : [Transmit data buffer size] uint8_t *rx_buf : [Receive data buffer address]
Return value	[MD_OK]: Transmission/reception setting has completed. [MD_ARGERROR]: Transmission/reception setting has failed.
Remarks	None

[Function Name] r_csi00_interrupt	
Synopsis	CSI00 transfer end interrupt function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void __near r_csi00_interrupt(void)
Explanation	If there is data not transmitted, this function reads receive data and then starts transmitting the data not transmitted. Otherwise, this function reads receive data.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_csi00_callback_receiveend	
Synopsis	CSI00 data reception completion processing
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void r_csi00_callback_receiveend(void)
Explanation	If the reception of data is completed, set the BUSY signal to BUSY state.
Arguments	None
Return value	None
Remarks	None

5.7 Flowcharts

Figure 5.1 shows the overall flow of the sample program described in this application note.

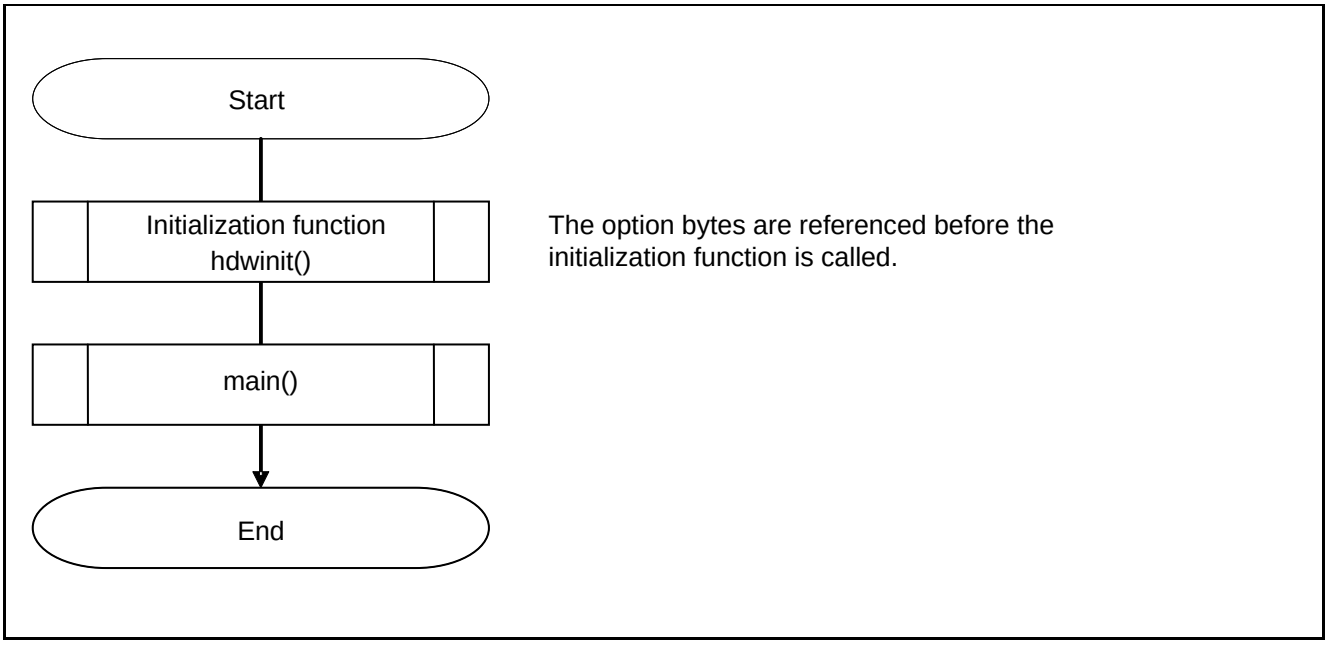


Figure 5.1 Overall Flow

Note: Startup routine is executed before and after the initialization function.

5.7.1 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.

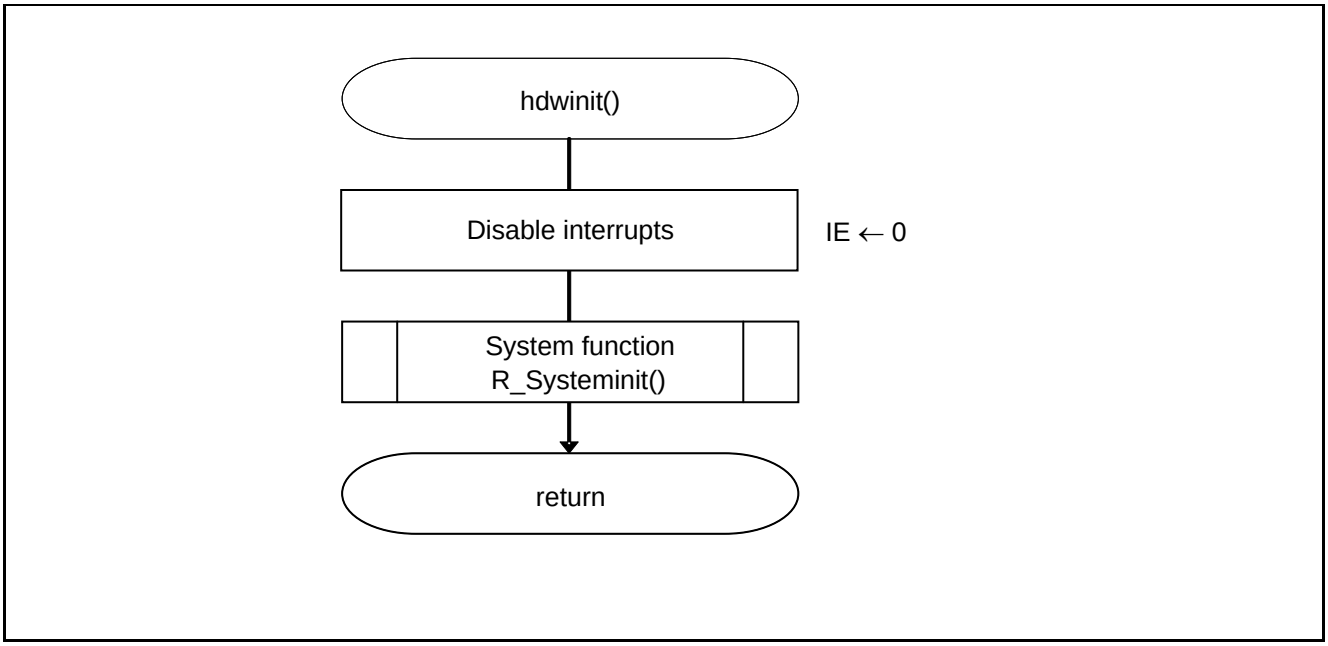


Figure 5.2 Initialization Function

5.7.2 System Function

Figure 5.3 shows the flowchart for the system function.

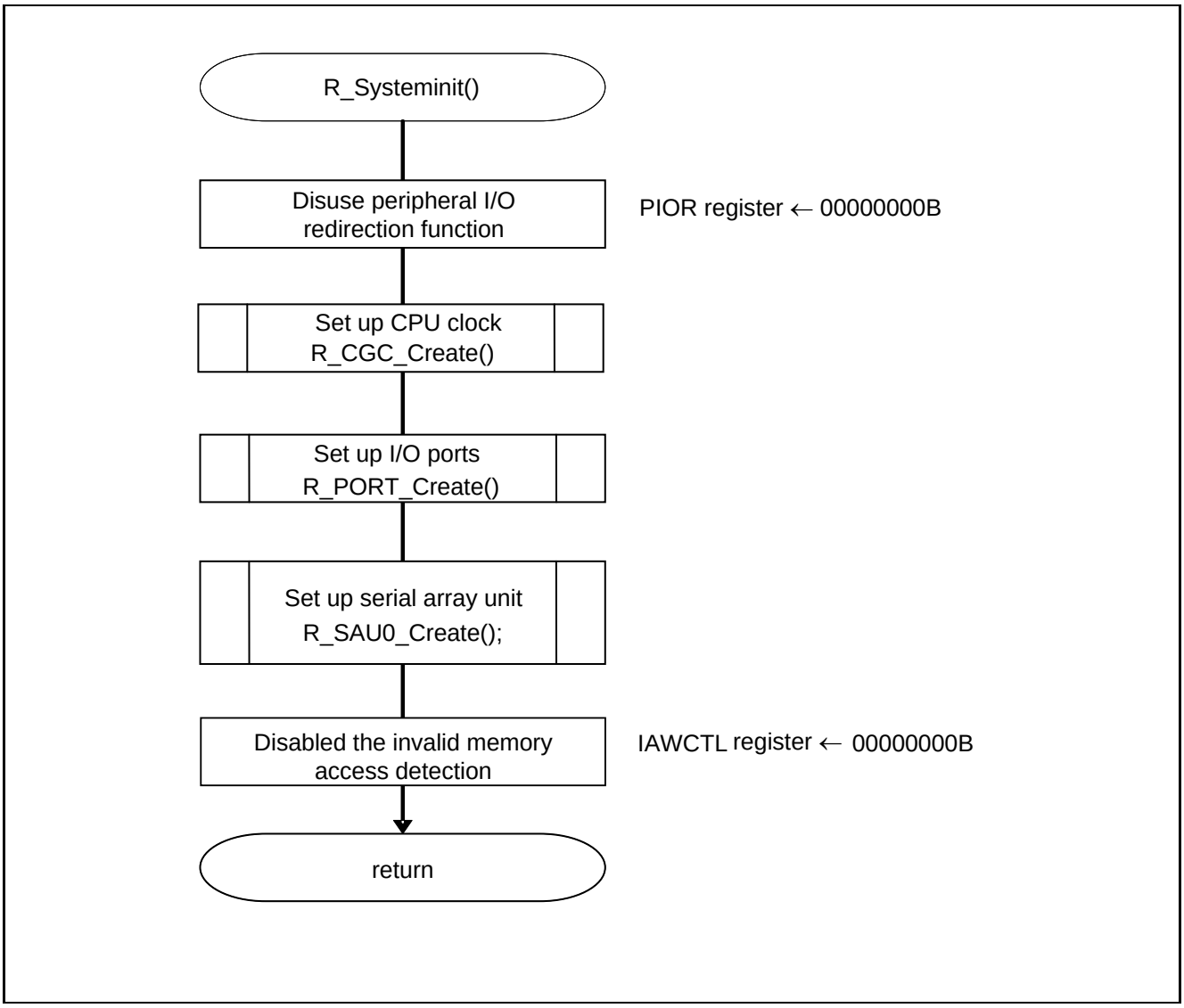


Figure 5.3 System Function

5.7.3 I/O Port Setup

Figure 5.4 shows the flowchart for I/O port setup.

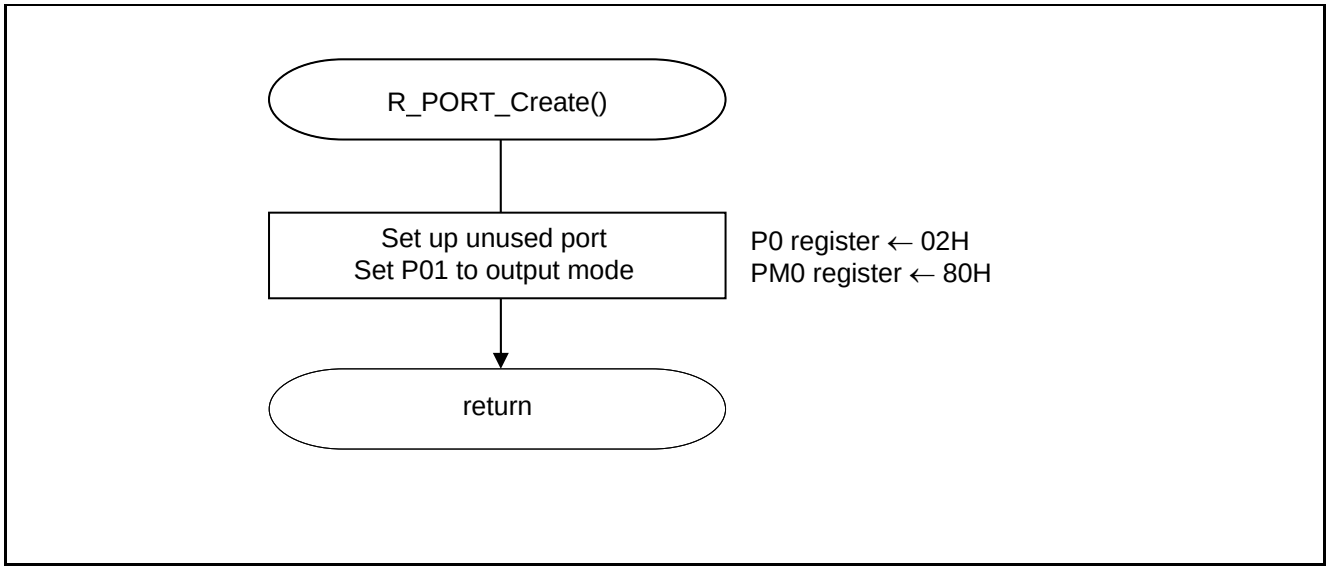


Figure 5.4 I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575EJ0100) for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to V_{DD} or V_{SS} via a separate resistor.

Setting up the BUSY signal output ports

- Port register 0 (P0)
- Port mode register 0 (PM0)
  Select an I/O mode and output latch for each port.

Symbol: P0

7	6	5	4	3	2	1	0
0	P06	P05	P04	P03	P02	P01	P00
0	x	x	x	x	x	**1**	x

Bit 1

P01	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	**Input high level**

Symbol: PM0

7	6	5	4	3	2	1	0
1	PM06	PM05	PM04	PM03	PM02	PM01	PM00
1	x	x	x	x	x	**0**	x

Bit 1

PM01	PM11 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.7.4 CPU Clock Setup

Figure 5.5 shows the flowchart for setting up the CPU clock.

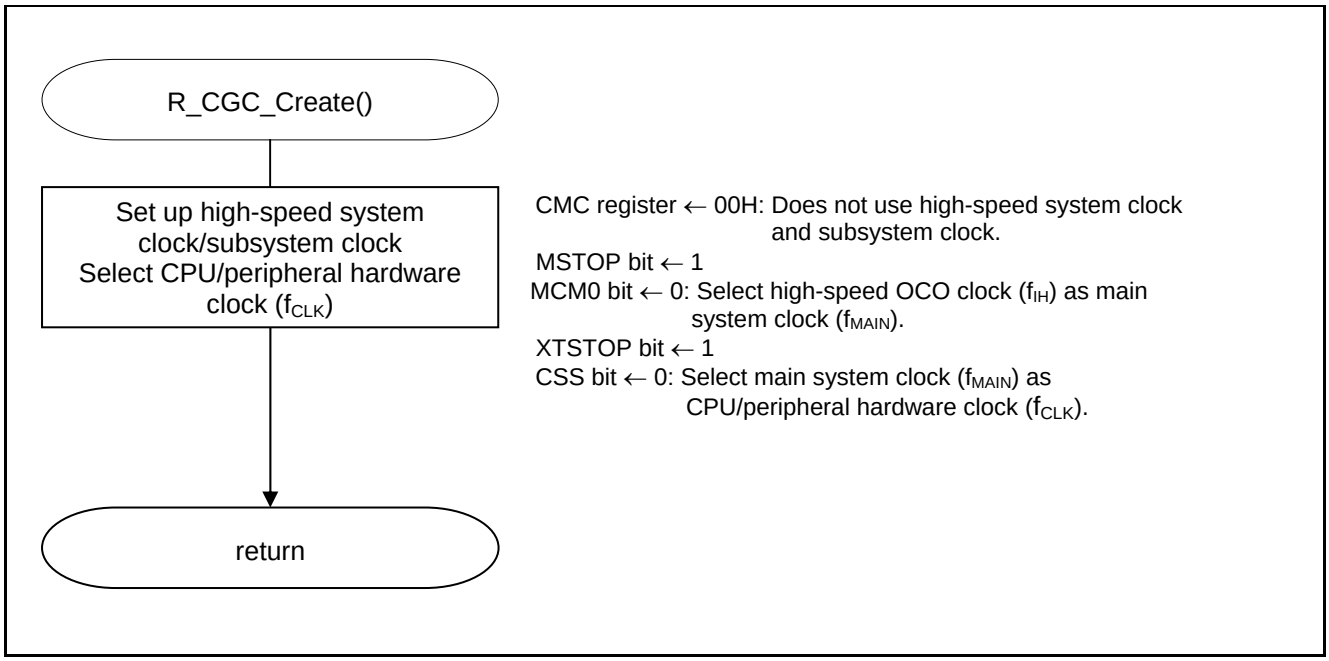


Figure 5.5 CPU Clock Setup

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575EJ0100).

5.7.5 SAU0 Setup

Figure 5.6 shows the flowchart for SAU0 setup.

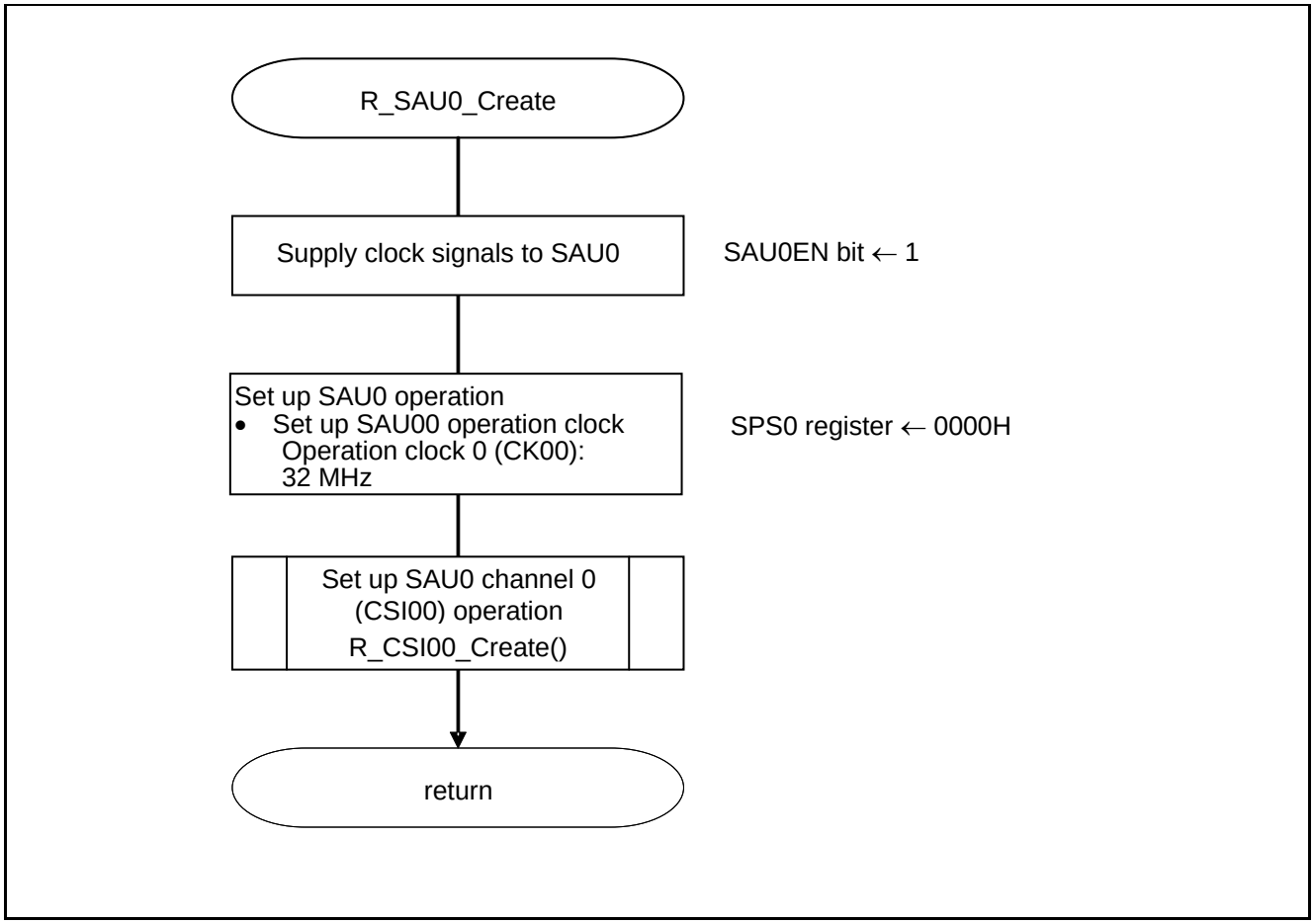


Figure 5.6 SAU0 Setup

Enabling supply of clock signals to the SAU

- Peripheral enable register 0 (PER0)
  Enable supply of clock signals to the SAU0.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
x	x	x	x	x	**1**	x	x

Bit 2

SAU0EN	Control of serial array unit 0 and input clock supply
0	Stops input clock supply.
1	**Enables input clock supply.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Select serial clock

- Serial clock selection register 0 (SPS0)
  Select an operation clock for the SAU0.

Symbol: SPS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000
0	0	0	0	0	0	0	0	x	x	x	x	**0**	**0**	**0**	**0**

Bits 3 to 0

PRS 003	PRS 002	PRS 001	PRS 000	Selection of operation clock (CK00)					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	**f_{CLK} = 32 MHz**
0	**0**	**0**	**0**	**f_{CLK}**	2 MHz	5 MHz	10 MHz	20 MHz	**32 MHz**
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1,25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^1$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.7.6 SAU0 Channel 0 (CSI00) Operation Setup

Figure 5.7 shows the flowchart for setting up SAU0 channel 0 (CSI00) operation.

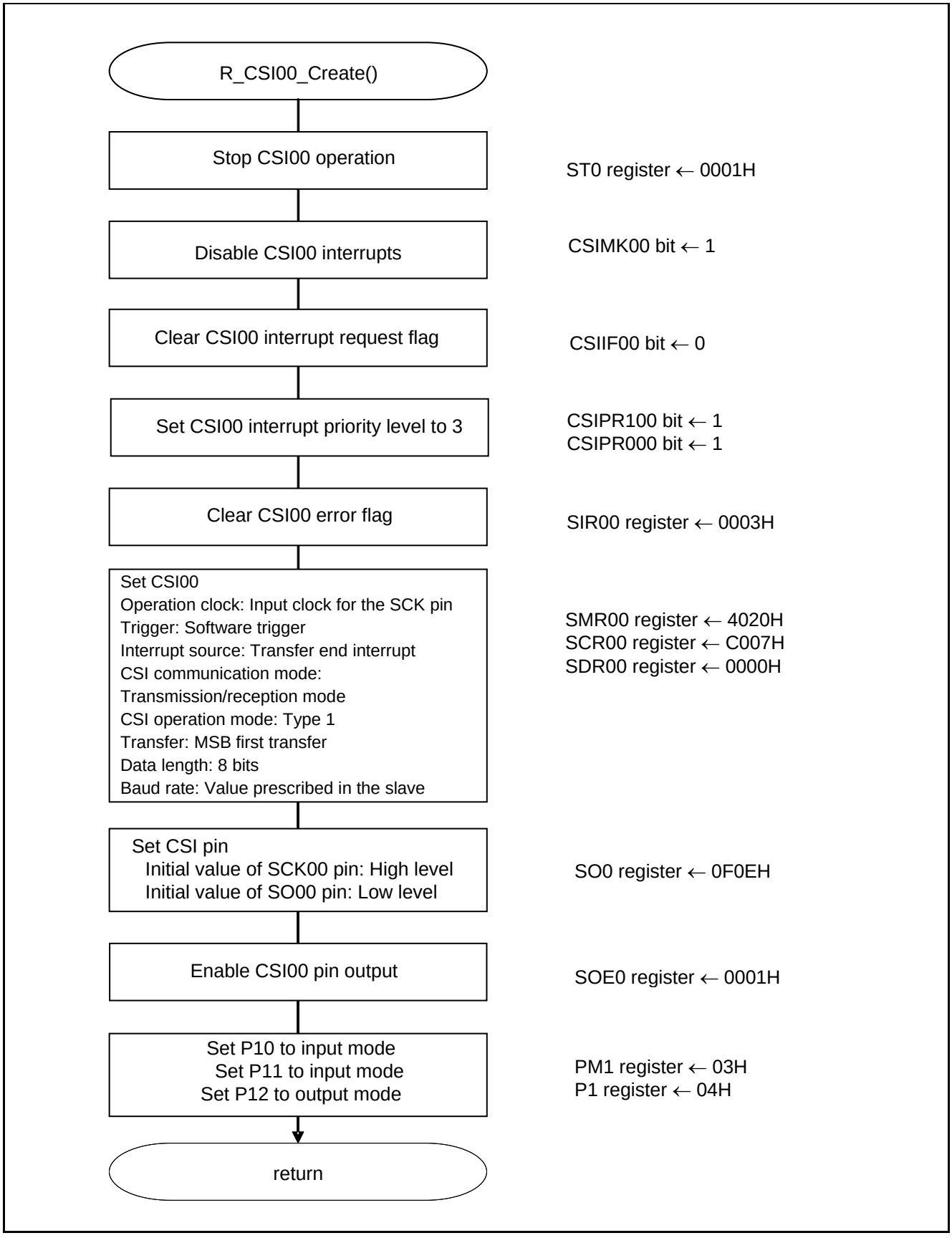


Figure 5.7 SAU0 Channel 0 (CSI00) Operation Setup

Stopping serial channel 0

- Serial channel stop register 0 (ST0)
  Stop communication/count operation of serial channel 0.

Symbol: ST0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	**1**

Bit 0

ST00	Operation stop trigger of channel 0
0	No trigger operation
1	**Clears the SE00 bit to 0 and stops the communication operation.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting a transfer end interrupt priority level

- Priority specification flag register 00H (PR00H)
- Priority specification flag register 10H (PR10H)
  Set the interrupt priority level.

Symbol: PR00H

7	6	5	4	3	2	1	0
SREPR00 TMPR001H	SRPR00 CSIPR001 IICPR001	STPR00 **CSIPR000** IICPR000	DMAPR01	DMAPR00	SREPR02 TMPR011H	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020
x	x	**1**	x	x	x	x	x

Symbol: PR10H

7	6	5	4	3	2	1	0
SREPR10 TMPR101H	SRPR10 CSIPR101 IICPR101	STPR10 **CSIPR100** IICPR100	DMAPR11	DMAPR10	SREPR12 TMPR111H	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120
x	x	**1**	x	x	x	x	x

Bit 5

CSIPR000	CSIPR100	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	**1**	**Specify level 3 (low priority level)**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Clearing the CSI00 error flags

- Serial flag clear trigger register 00 (SIR00)
  Clear the SAU0 channel 0 error flags.

Symbol: SIR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	PECT00	OVCT00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**

Bit 1

PECT00	Clear trigger of parity error flag of channel 0
0	Not cleared
1	**Clears the PEF00 bit of the SSR00 register to 0.**

Bit 0

OVCT00	Clear trigger of overrun error flag of channel 0
0	Not cleared
1	**Clears the OVF00 bit of the SSR00 register to 0.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting up the SAU0 channel 0 operation mode

- Serial mode register 00 (SMR00)
  Select an operation clock (f_{MCK}).
  Specify whether to make the serial clock (f_{SCK}) input available.
  Set the start trigger and operation mode.
  Select an interrupt source.

Symbol: SMR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS00	CCS00	0	0	0	0	0	STS00	0	SIS000	1	0	0	MD002	MD001	MD000
0	**0**	0	0	0	0	0	**0**	0	0	1	0	0	**0**	**0**	**0**

Bit 15

CKS00	Selection of operation clock (f_{MCK}) of channel n
0	**Operation clock CK00 set by the SPS0 register**
1	Operation clock CK01 set by the SPS0 register

Bit 14

CCS00	Selection of transfer clock (f_{TCLK}) of channel n
0	**Divided operation clock f_{MCK} specified by the CKS00 bit**
1	Clock input f_{SCK} from the $\overline{SCK00}$ pin (slave transfer in CSI mode)

Bit 8

STS00	Selection of start trigger source
0	**Only software trigger is valid**
1	Valid edge of the RxDq pin (selected for UART reception)

Bits 2 and 1

MD002	MD001	Setting of operation mode of channel 0
0	**0**	**CSI mode**
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

Bit 0

MD000	Selection of interrupt source of channel 0
0	**Transfer end interrupt**
1	Buffer empty interrupt

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting up the SAU0 channel 0 operation mode

- Serial communication operation setup register 00 (SCR00)
  Select an operation clock (f_{MCK}).
  Specify whether to make the serial clock (f_{SCK}) input available.
  Set up the start trigger and operation mode.
  Select an interrupt source.

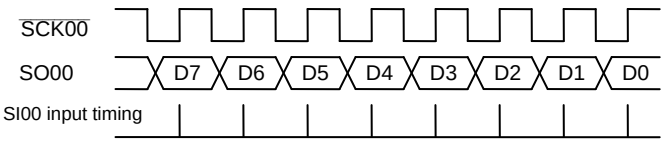
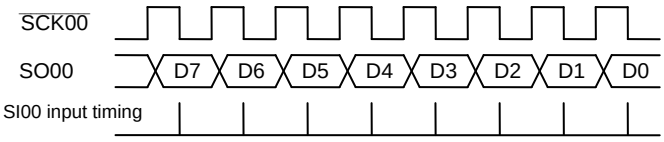
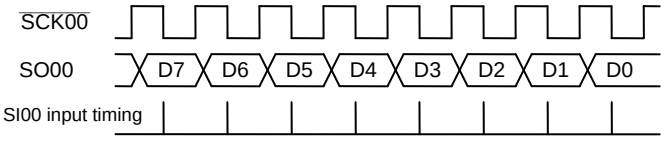
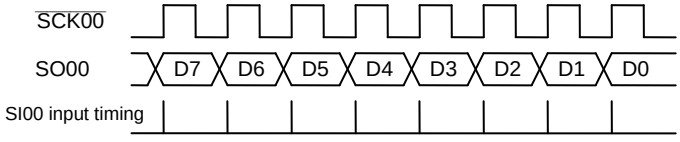
Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE00	RXE00	DAP00	CKP00	0	EOC00	PTC001	PTC000	DIR00	0	SLC001	SLC000	0	1	DLS001	DLS000
1	**1**	**0**	**0**	0	0	0	0	**0**	0	0	0	0	1	**1**	**1**

Bits 15 and 14

TXE00	RXE00	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	**1**	**Transmission/reception**

Bits 13 and 12

DAP00	CKP00	Selection of data and clock phase in CSI mode	Type
0	**0**		**1**
0	1		2
1	0		3
1	1		4

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE00	RXE00	DAP00	CKP00	0	EOC00	PTC001	PTC000	DIR00	0	SLC001	SLC000	0	1	DLS001	DLS000
1	**1**	**0**	**0**	0	0	0	0	**0**	0	0	0	0	1	**1**	**1**

Bit 7

DIR00	Selection of data transfer sequence in CSI and UART modes
0	**Inputs/outputs data with MSB first.**
1	Inputs/outputs data with LSB first.

Bits 1 and 0

DLS001	DLS000	Setting of data length in CSI and UART modes
0	0	9-bit data length (stored in bits 0 to 8 of the SDR00 register) (can be set in UART0 mode only.)
1	0	7-bit data length (stored in bits 0 to 6 of the SDR00 register)
1	**1**	**8-bit data length (stored in bits 0 to 7 of the SDR00 register)**
Other than above		Setting prohibited

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

SO00 pin output value setting

- Serial output register 0 (SO0)
  Set the output values of the serial data output pin and the serial clock output pin

Symbol: SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00
0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	**0**

Bit 0

SO00	Serial data output of channel 0
0	**Serial data output value is “0”**
1	Serial data output value is “1”

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting for enabling output through serial communication

- Serial output enable register 0 (SOE0)
  Enable output of serial communication

Symbol: SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	**1**

Bit 0

SOE00	Serial output enable/stop of channel 0
0	Stops output by serial communication operation.
1	**Enables output by serial communication operation.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting up the ports of the SCK00, SO00 and SI00 pins

- Port register 1 (P1)
- Port mode register 1 (PM1)
  Select an input/output mode and output latch for each port.

Symbol: P1

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
x	x	x	x	x	**1**	x	**1**

Bit 2

P12	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	Input high level

Symbol: PM1

7	6	5	4	3	2	1	0
PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
x	x	x	x	x	**0**	**1**	**0**

Bit 2

PM12	P12 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Bit 1

PM11	P11 pin I/O mode selection
0	Output mode (output buffer on)
1	**Input mode (output buffer off)**

Bit 0

PM10	P10 pin I/O mode selection
0	Output mode (output buffer on)
1	**Input mode (output buffer off)**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.7.7 Main Processing

Figure 5.8 shows the flowchart for main processing.

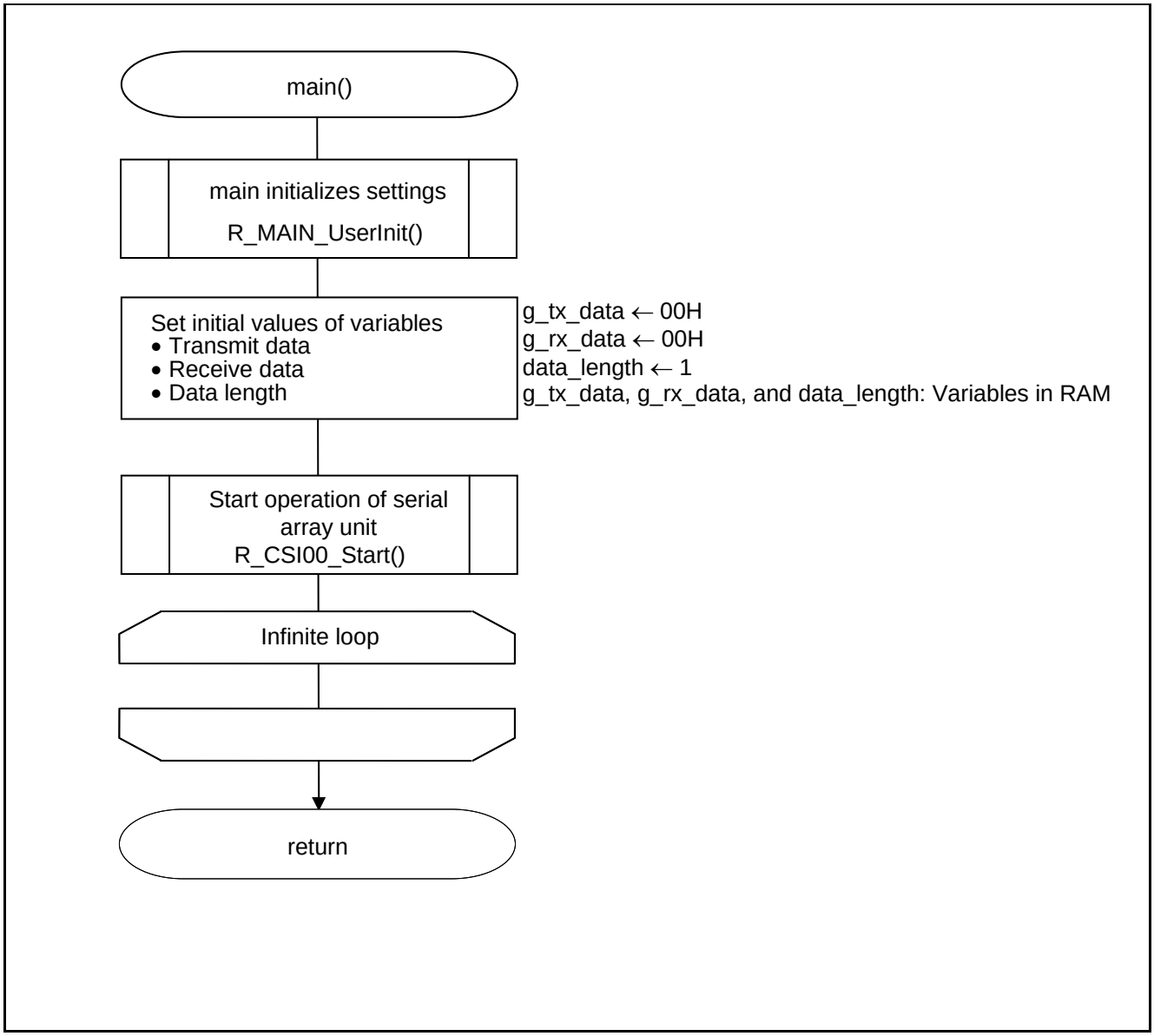


Figure 5.8 Main Processing

Caution: For infinite loop, please refer to the 5.7.9 Infinite Loop in Main Processing.

5.7.8 Main initializes settings

Figure 5.9 shows the flowchart for the main initializes settings.

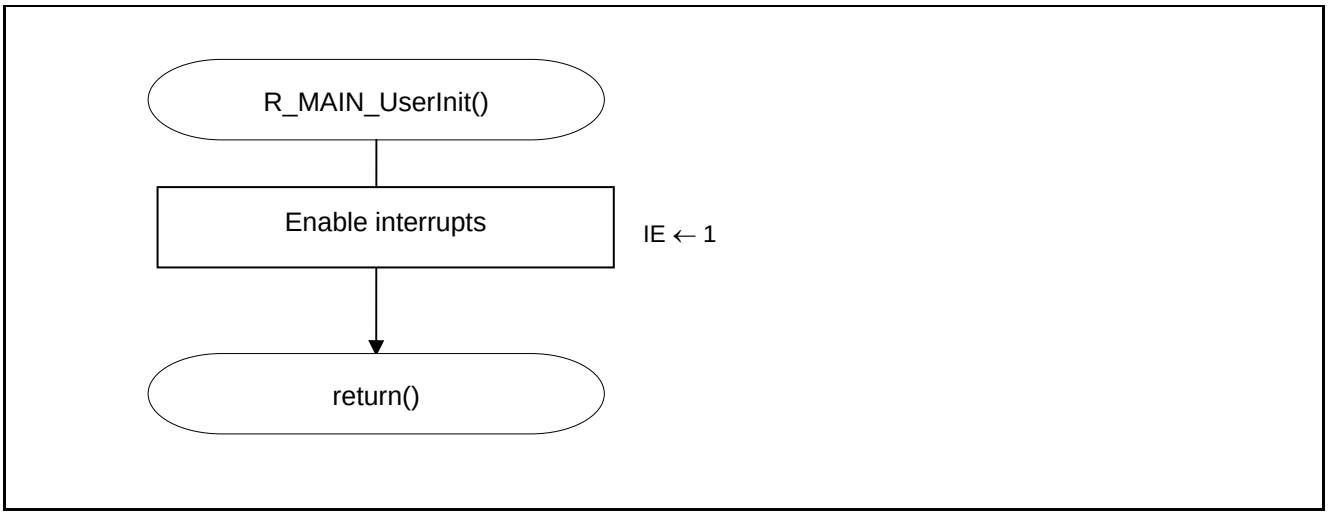


Figure 5.9 Main initializes settings

5.7.9 SAU0 Channel 0 (CSI00) Operation Start Processing

Figure 5.10 shows the flowchart for starting operation of SAU0 channel 0 (CSI00).

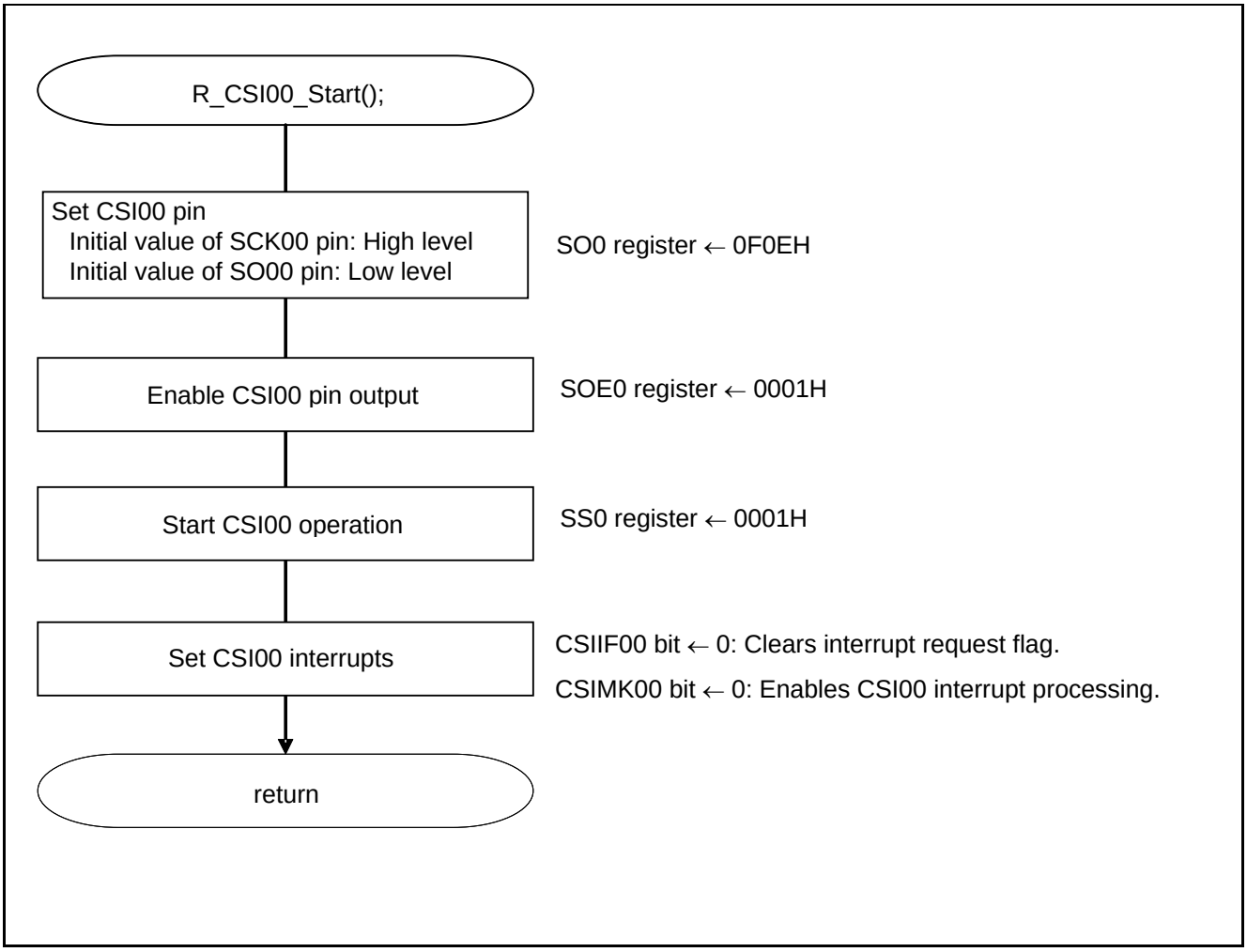


Figure 5.10 SAU0 Channel 0 (CSI00) Operation Start Processing

Setting the transfer end interrupt

- Interrupt request flag register 0H (IF0H)
  Clear the interrupt request flag.
- Interrupt mask flag register 0H (MK0H)
  Enable interrupt processing.

Symbol: IF0H

7	6	5	4	3	2	1	0
SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	DMAIF1	DMAIF0	SREIF2 TMIF11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	x	x	x	x	x

Bit 5

CSIIF00	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Symbol: MK0H

7	6	5	4	3	2	1	0
SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	DMAMK1	DMAMK0	SREMK2 TMMK11H	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20
x	x	**0**	x	x	x	x	x

Bit 5

CSIMK00	Interrupt processing control
0	**Enables interrupt processing.**
1	Disables interrupt processing.

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Enabling serial communication

- Serial channel start register 0 (SS0)
  Enable serial communication/count operation.

Symbol: SS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	**1**

Bit 0

SS00	Operation start trigger of channel 0
0	No trigger operation
1	**Sets the SE00 bit to 1 and enters the communication wait status.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Remarks: When the SS0 register is read, 0000H is always read.

5.7.10 Infinite Loop in Main Routine

Figure5.11 shows the flowchart for an infinite loop in the main routine.

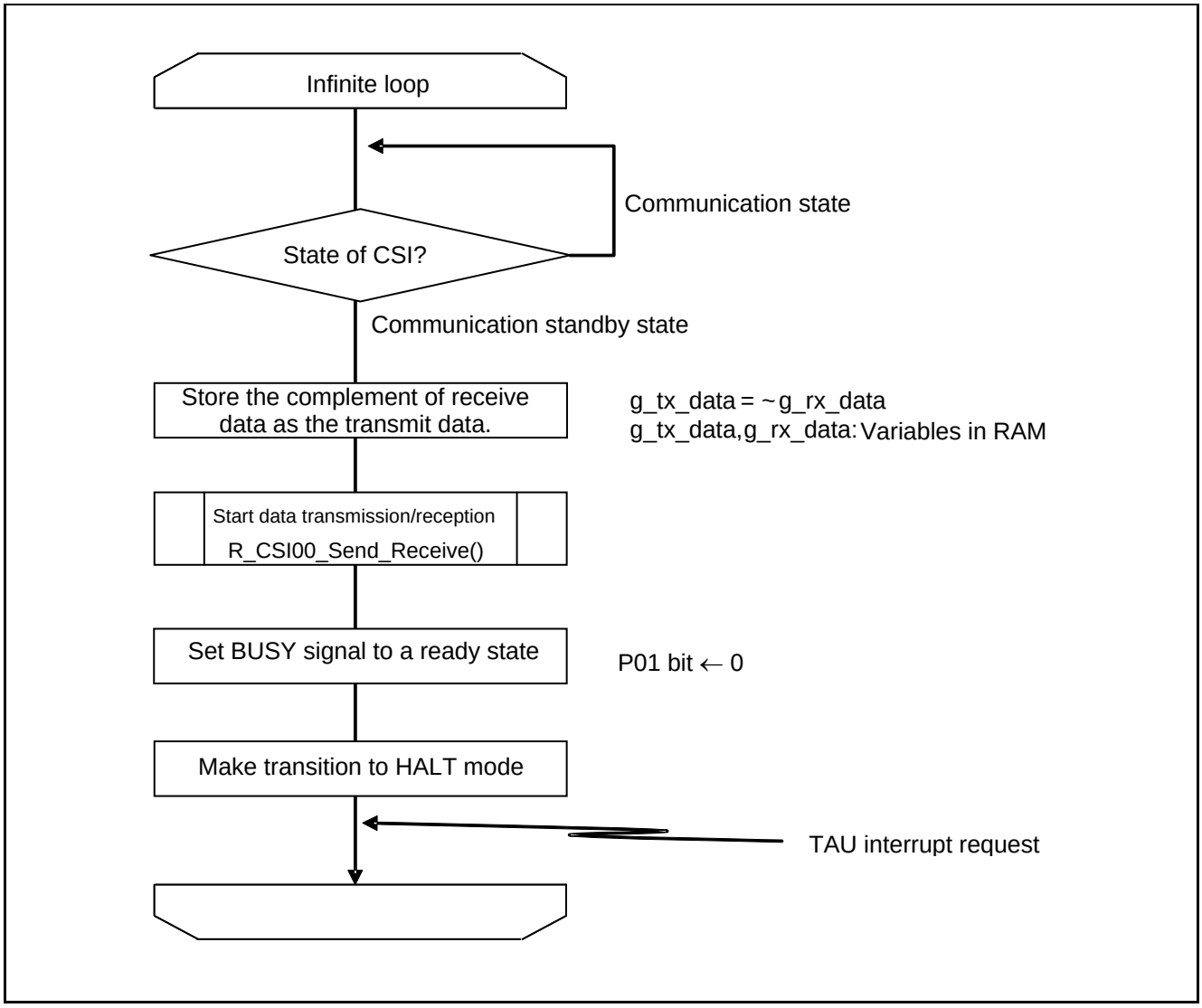


Figure 5.11 Infinite Loop in Main Processing

Confirming the communication state

- Serial status register 00 (SSR00)
  Indicate the communication status and error occurrence status of serial array unit channel 0.

Symbol: SSR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	TSF00	BFF00	0	0	0	PEF00	OVF00
0	0	0	0	0	0	0	0	0	**0/1**	x	0	0	0	x	x

Bit 6

TSF00	Communication status indication flag of channel 0
0	Communication is stopped or suspended.
1	Communication is in progress.

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.7.11 CSI00 Data Transmission/Reception Start

Figure 5.12, 5.13 shows the flowchart for starting CSI00 data transmission/reception.

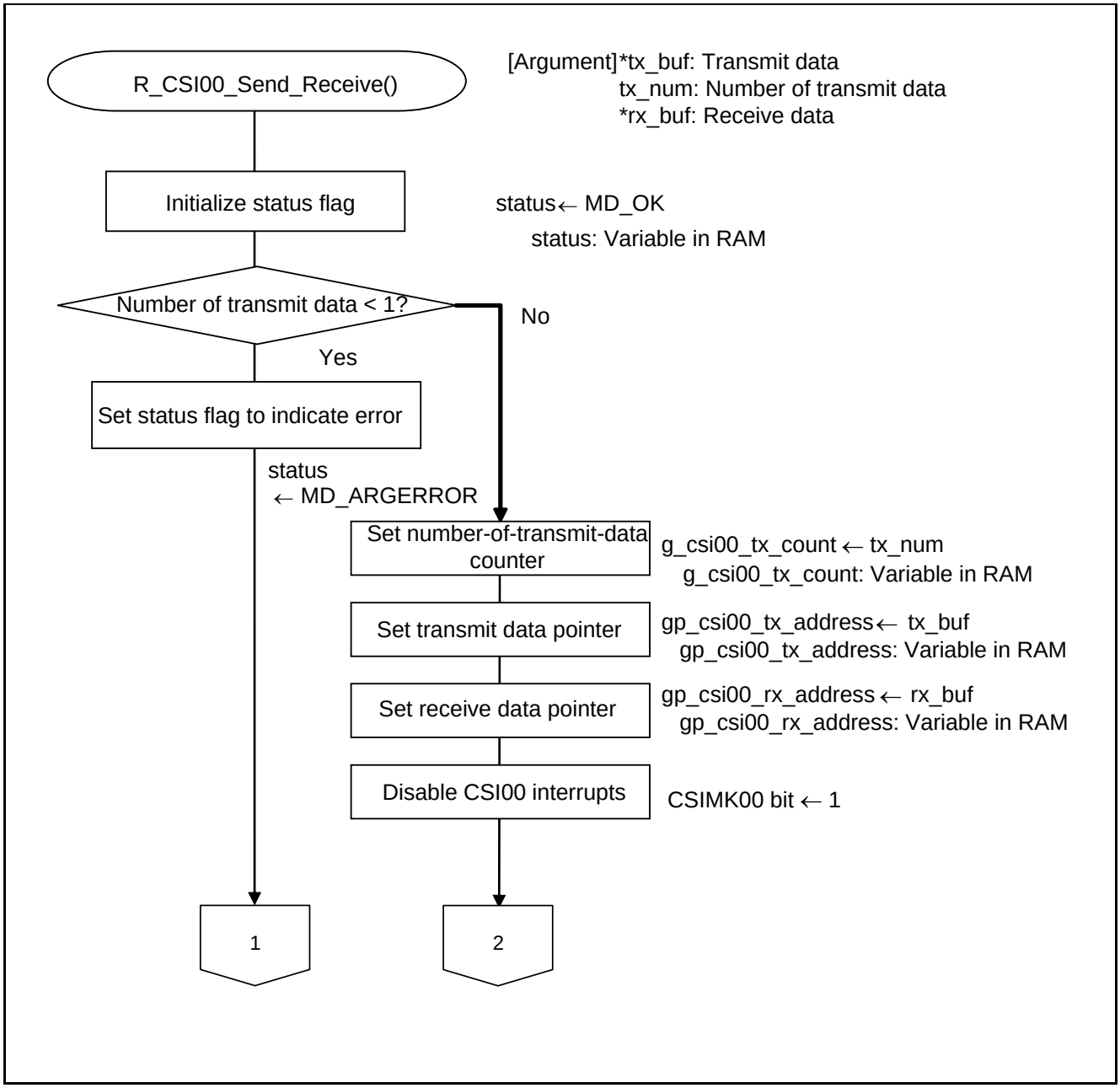


Figure 5.12 CSI00 Data Transmission/Reception Start (1/2)

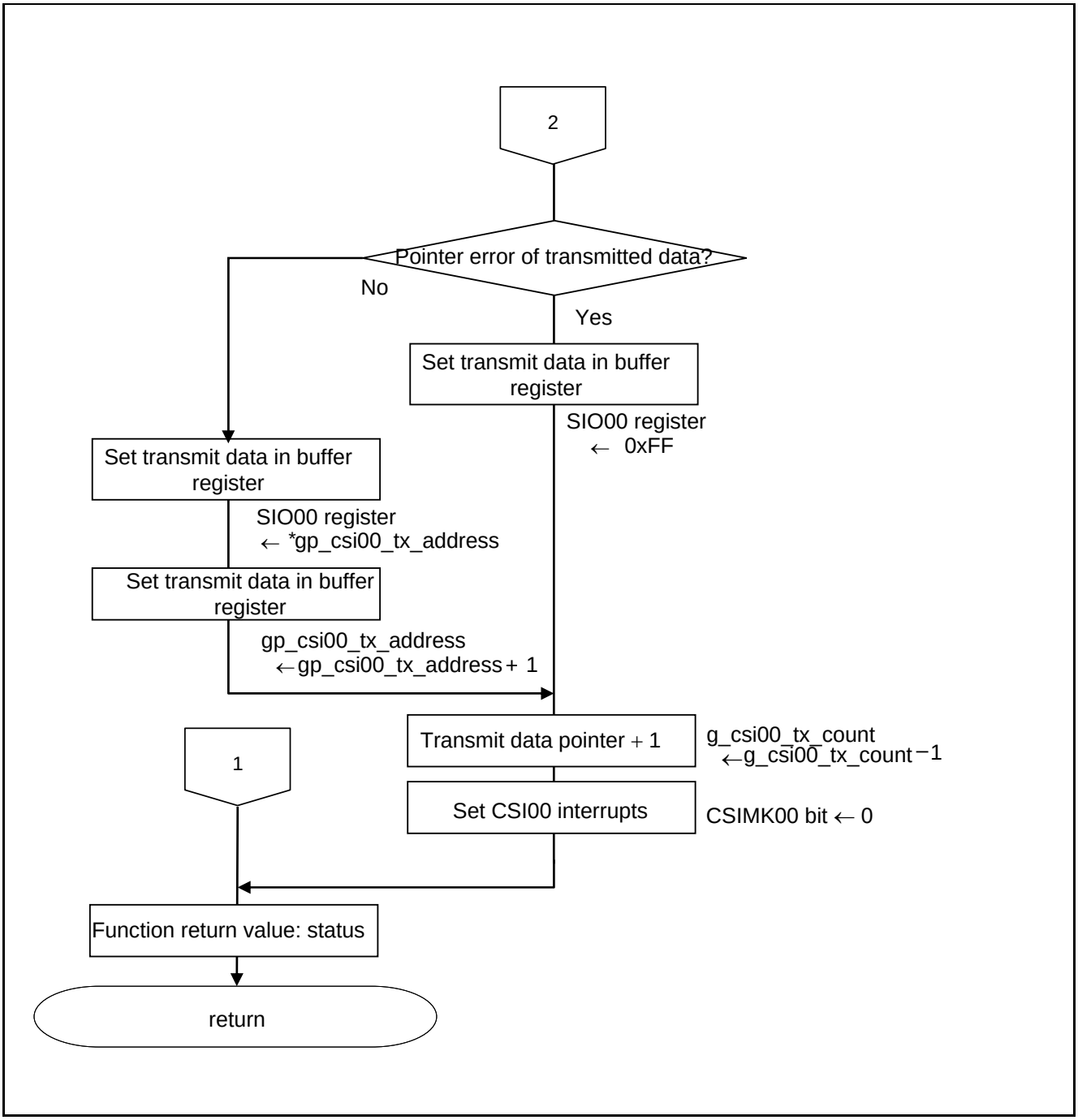
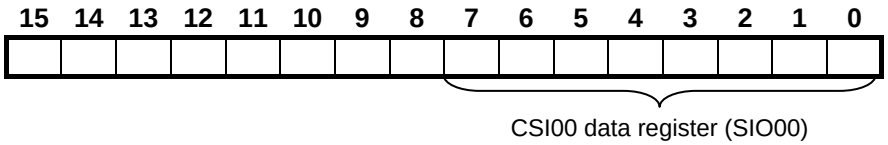


Figure 5.13 CSI00 Data Transmission/Reception Start (2/2)

Setting transmit data

- Serial data register 00 (SDR00)
  Set transmit data and start transmitting the data.

Symbol: SDR00



Write transmit data to the lower eight bits.

These eight bits should be accessed as the CSI00 register.

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.7.12 CSI00 Transfer End Interrupt Processing

Figure 5.14, 5.15 shows the flowchart for CSI00 transfer end interrupt processing.

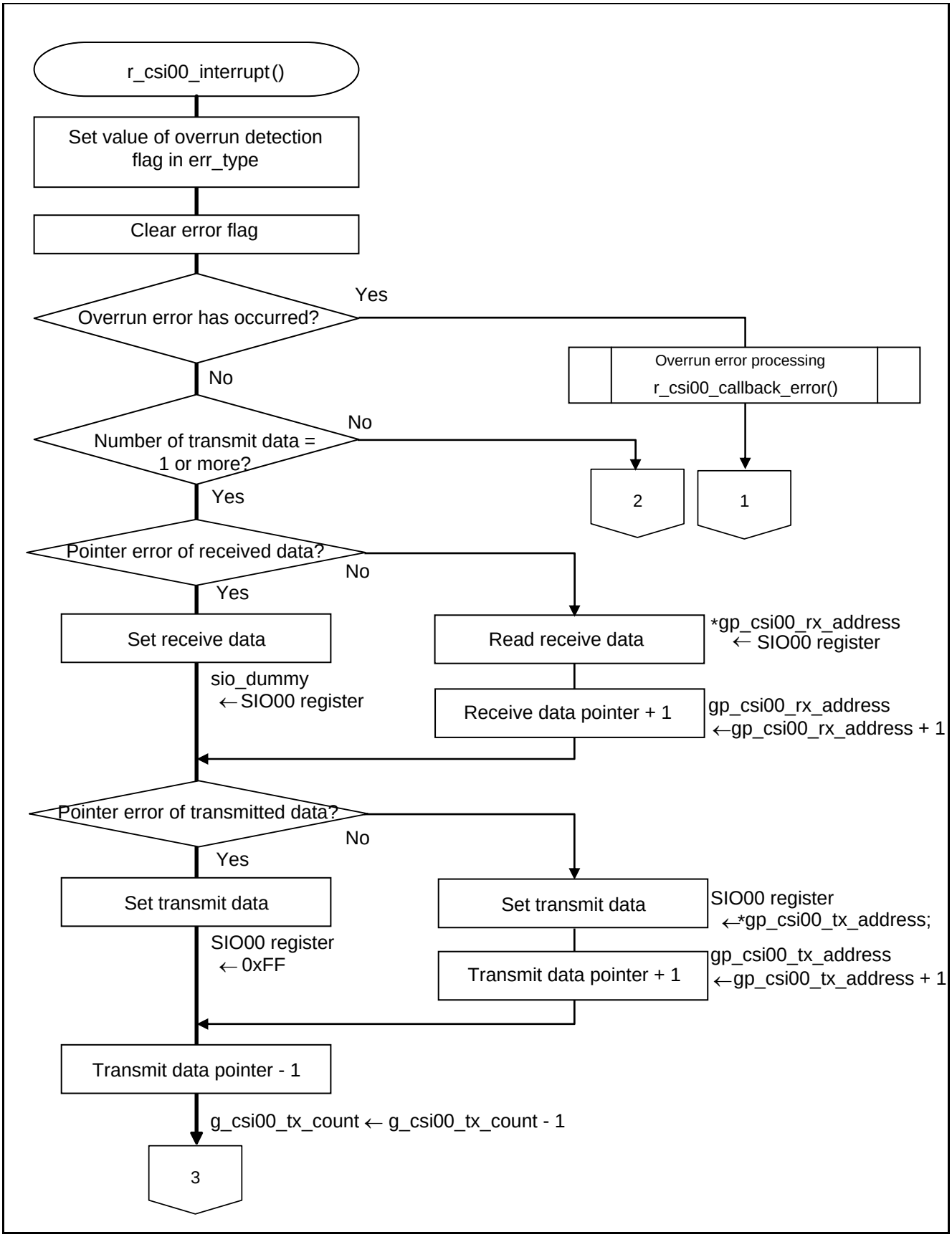


Figure 5.14 CSI00 Transfer End Interrupt Processing (1/2)

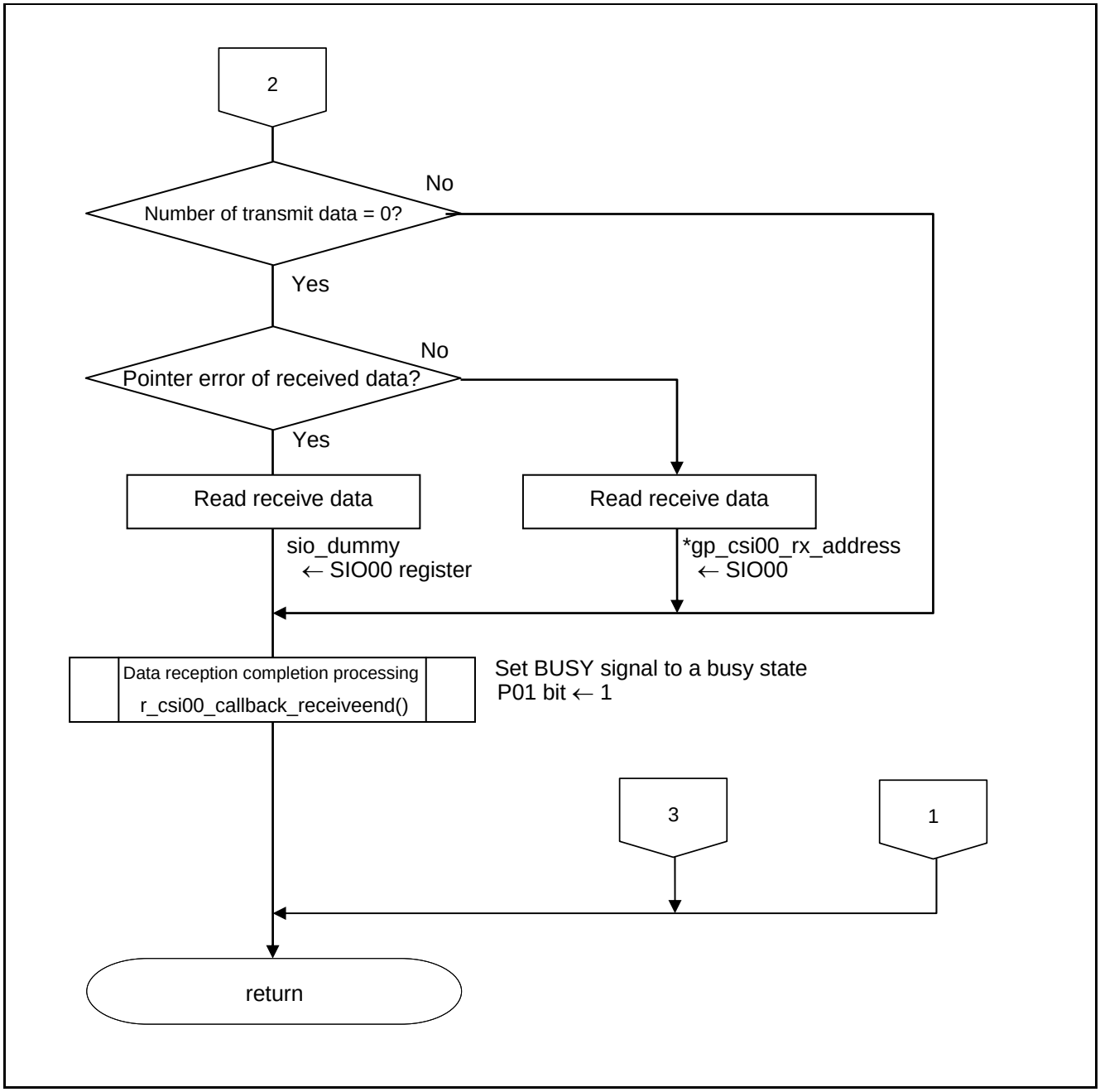


Figure 5.15 CSI00 Transfer End Interrupt Processing (2/2)

5.7.13 Data reception completion processing

Figure 5.16 shows the flowchart for the data reception completion processing.

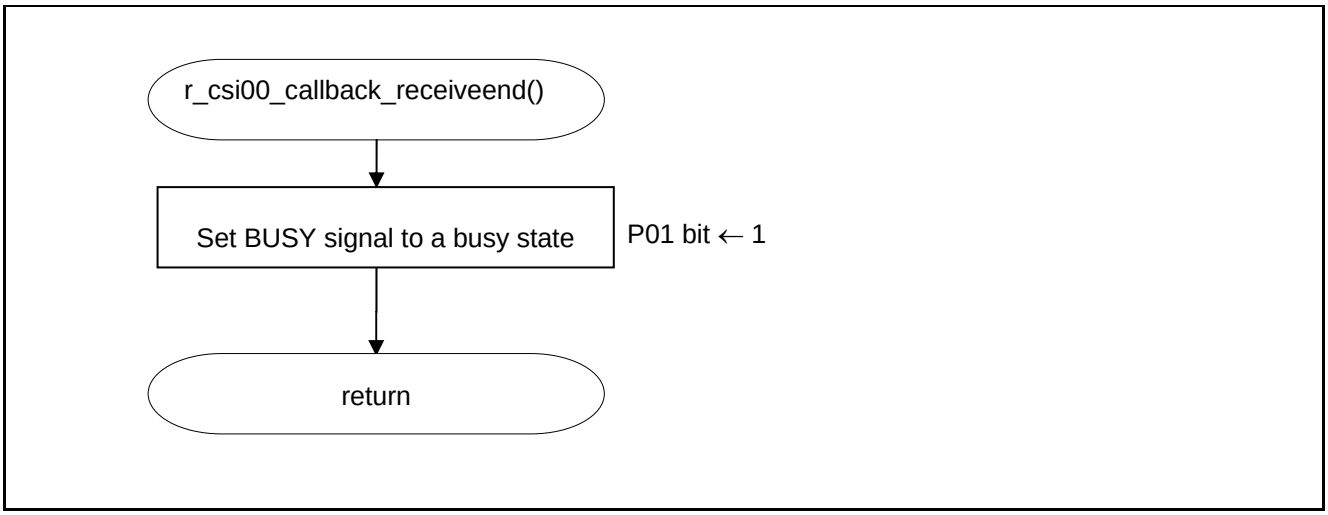


Figure 5.16 Data reception completion processing

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

User's Manual:

RL78/G13 User's Manual: Hardware (R01UH0146EJ)
RL78 Family User's Manual: Software (R01US0015EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Updates/Technical News

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

Website and Support

Renesas Electronics Website

http://www.renesas.com/index.jsp

Inquiries

http://www.renesas.com/contact/

REVISION HISTORY	RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Slave Transmission/Reception) CC-RL

Rev.	Date	Description	
		Page	Summary
1.00	Apr. 16, 2015	—	First edition issued
2.00	July 01, 2015	7	Table 2.1: Added e^2 studio

0

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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 an MPU or MCU 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
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, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0