RL78/G22

Realtime Clock

Introduction

This application note shows usage examples of the fixed-cycle interrupt function and the alarm interrupt function of the realtime clock (RTC).

The fixed-cycle interrupt function inverts the outputs of output ports and displays the clock time on the LCD. The alarm interrupt function generates an alarm interrupt five seconds after the set clock time.

Continuous RTC operation can be monitored with the clock time display on the LCD even during a reset period.

Target Device

RL78/G22

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

1.1 Overview of Specifications

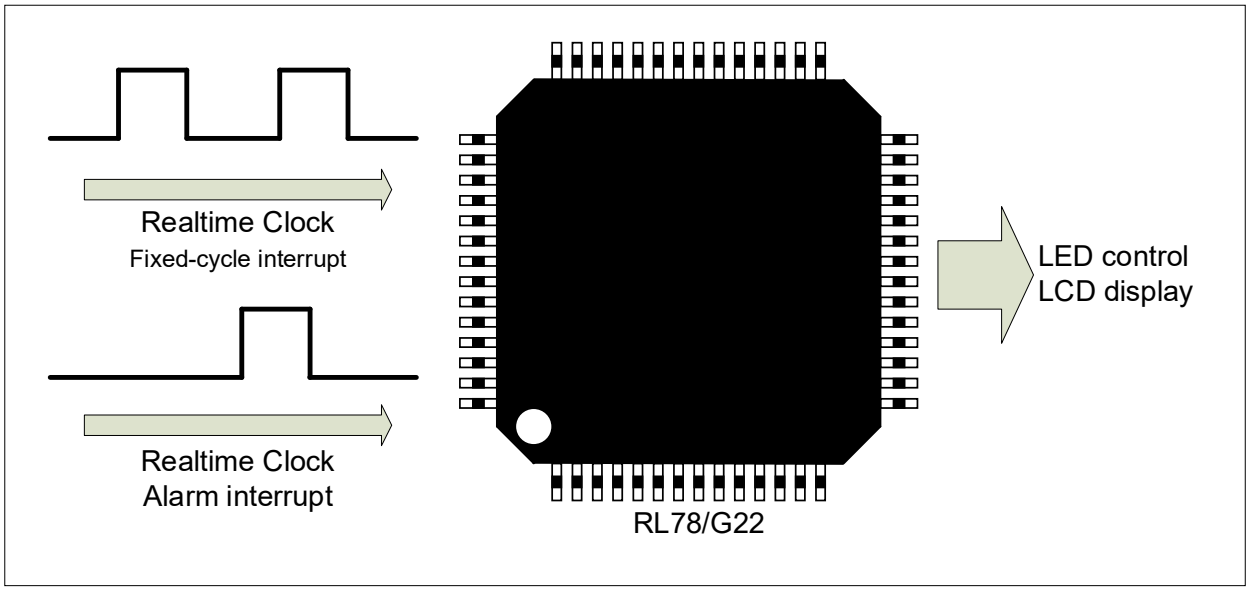
The fixed-cycle interrupt function inverts the outputs of output ports and displays the clock time on the LCD. The alarm interrupt function generates an alarm interrupt five seconds after the set clock time.

Table 1-1 lists peripheral functions to be used and Figure 1-1 shows an overview of sample code operation.

Table 1-1 Peripheral Function and Use

Peripheral Function	Use
Realtime clock	RTC interrupt (INTRTC)
P62	Set to port output in the fixed-cycle interrupt processing (inverted output)
P63	Set to port output in the alarm interrupt processing (high level output)
Serial Interface IICA0 P60/SCLA0, P61/SDAA0	I2C communication with the LCD module (LCM)
$\overline{\text{RESET}}$	External reset input
P123/XT1	RTC operating clock

Figure 1-1 Overview of Sample Code Operation



1.2 Outline of operation

In this application note, the clock time of the RTC is set to “2024/1/1 (Mon) 15:59:55” and the alarm time is set to “16:00:00 every day”. Furthermore, the following interrupt processing is performed.

- P62 output inversion and clock time display on the LCD in the fixed-cycle interrupt processing
- Low-level output of P63 (LED ON) in the alarm interrupt processing

(1) Initialize the realtime clock (RTC)

- Select the subsystem clock (f_{SXR}) at the RTC operation clock.
- Present the time in 24-hour system.
- Disable the RTC1HZ pin output.
- Enable fixed-cycle interrupt and set their cycle time to 0.5 second.
- Enable alarm interrupt.
- Enable INTRTC interrupts.

(2) Initialize the I/O ports.

- Set P62 to the output port for the fixed-cycle interrupt processing. (The LED is lit when the initial value is low level.)
- Set P63 to the output port for the alarm interrupt processing. (The LED is unlit when the initial value is high level.)

(3) Initialize the serial interface IICA

- Use IICA0 (P60 set to SCLA0 and P61 set to SDAA0).
- Select f_{CLK}/2 at the IICA0 operation clock.
- Set the local address to 0x10.
- Set the standard mode as the operation mode.
- Set the transfer clock to 80000 bps.
- Enable INTIICA0 interrupt.

(4) Initialize the LCD module

- Set to 8 bits, bus mode, 2-line display, and font type 5x8 dots.
- Make settings to enable display indication, disable cursor display, and disable cursor blinking.
- Set the cursor shift direction to right.

(5) Control the LEDs and perform communication with the LCD module according to interrupts of the RTC.

Note 1. Refer to the RL78/G22 User’s Manual: Hardware for usage notes concerning this device.
Note 2. Make initial settings for the RTC only when a power-on reset is generated. To enable this, some codes generated by the smart configurator are modified for source codes. Note that, if the smart configurator re-generates codes, this modification becomes invalid. The constants defined in r_cg_userdefine.h are used for the RTC’s clock time and the initial set value of alarm time.

2. Operation Confirmation Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2-1 Operation Confirmation Conditions

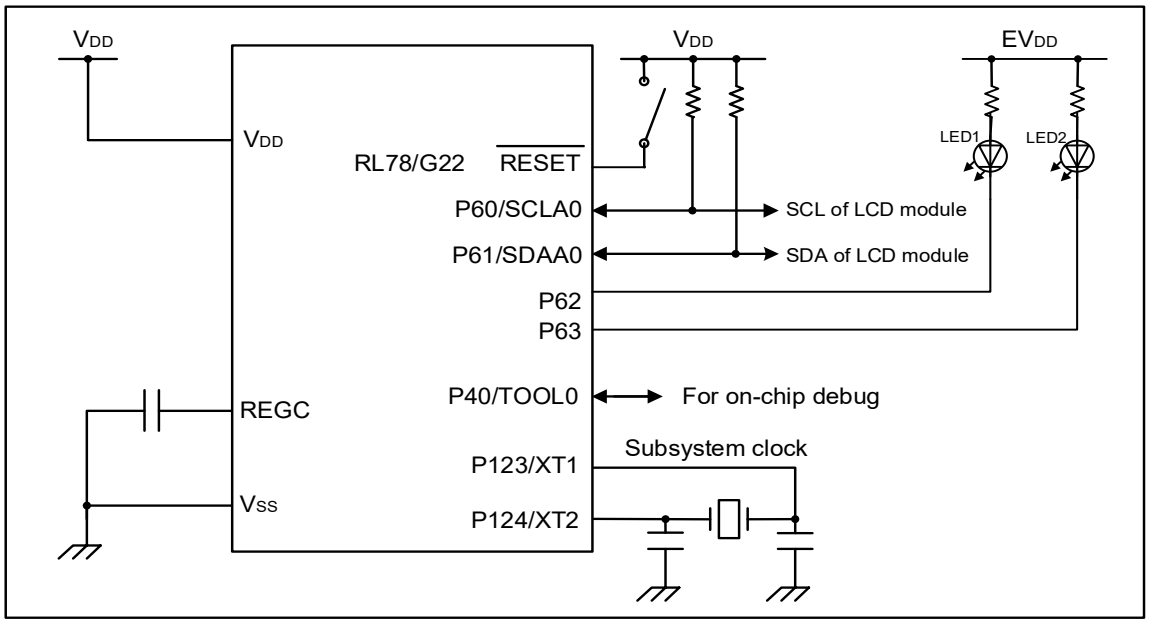
Item	Description
MCU used	RL78/G22 (R7F102GGE2DFB)
Board used	RL78/G22-64p Fast Prototyping Board ((RTK7RLG220C00000BJ)
Operating frequency	● High-speed on-chip oscillator clock (f_{IH}): 32 MHz ● Subsystem clock (XT1 clock (f_{XT})): 32.768 kHz
Operating voltage	3.3 V (can be operated at 2.4 V to 5.5 V) LVD0 operations (V_{LVD0}): Reset mode At rising edge TYP. 1.90 V (1.84 V to 1.95 V) At falling edge TYP. 1.86 V (1.80 V to 1.91 V)
Integrated development environment (CS+)	CS+ for CC V8.12.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.14.00 from Renesas Electronics Corp.
Integrated development environment (e2studio)	e2studio V2024-10 from Renesas Electronics Corp.
C compiler (e2studio)	CC-RL V1.14.00 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 V5.10.3 from IAR Systems Corp.
C compiler (IAR)	IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716 from IAR Systems Corp.
Smart configurator (SC)	V1.11.0 from Renesas Electronics Corp.
Board support package (BSP)	V1.62 from Renesas Electronics Corp.
LCD module	ACM1602NI-FLW-FBW-M01

3. Hardware Descriptions

3.1 Example of Hardware Configuration

Figure 3-1 shows an example of the hardware configuration used in the application note.

Figure 3-1 Hardware Configuration



Note 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes appropriate pin handling and meets electrical characteristic requirements (connect each input-only port to V_{DD} or V_{SS} through a resistor).

Note 2. VDD must not be lower than the reset release voltage (VLVD0) that is specified for the LVD0.

3.2 List of Pins to be Used

Table 3-1 lists the pins to be used and their functions.

Table 3-1 Pins to be Used and Their Functions

Pin name	I/O	Function
P62	Output	LED1 control
P63	Output	LED2 control
P60 / SCLA0, P61 / SDAA0	Input/ Output	I2C communication with LCD module
$\overline{\text{RESET}}$	Input	External reset input
P123 / X T1, P124 / XT2	Input	RTC operating clock

Caution In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

4. Software Explanation

4.1 Setting of Option Byte

Table 4-1 shows the option byte settings.

Table 4-1 Option Byte Settings

Address	Setting Value	Contents
000C0H / 020C0H	11101111B	Disables the watchdog timer. (Counting stopped after reset)
000C1H / 020C1H	11111110B	LVD0 detection voltage: reset mode At rising edge TYP. 1.90 V At falling edge TYP. 1.86 V
000C2H / 020C2H	11101000B	HS mode, High-speed on-chip oscillator clock (f_{IH}): 32 MHz
000C3H / 020C3H	10000100B	Enables on-chip debugging

4.2 List of Constants

Table 4-2 and Table 4-3 lists the constants that are used in the sample code.

Table 4-2 Constants (1/2)

Constant Name	Setting Value	Description
_0xA0_LCM_SLAVE_ADDR	0xA0	Slave address for LCM commands
_0x00_LCM_SLAVE_ADDR_RW_LOW	0x00	Data write flag
_0x00_LCM_CONTROL_BYTE	0x00	Control byte for LCM commands
_0x80_LCM_CONTROL_BYTE_RS_HIGH	0x80	Control byte: Data transfer
_0x00_LCM_CONTROL_BYTE_RS_LOW	0x00	Control byte: Command transfer
_0x00_LCM_COMMAND_CLEAR_DISPLAY	0x00	Command: Clear Display
_0x04_LCM_COMMAND_ENTRY_MODE_SET	0x04	Command: Entry Mode Set
_0x02_LCM_COMMAND_ENTRY_MODE_SET_ID_HIGH	0x02	Entry Mode Set: Address increment
_0x00_LCM_COMMAND_ENTRY_MODE_SET_S_LOW	0x00	Entry Mode Set: Display shift OFF
_0x08_LCM_COMMAND_DISPLAY_ONOFF	0x08	Command: Display ON/OFF Control
_0x04_LCM_COMMAND_DISPLAY_ONOFF_D_HIGH	0x04	Display ON/OFF control: Display ON
_0x00_LCM_COMMAND_DISPLAY_ONOFF_C_LOW	0x00	Display ON/OFF control: Cursor display OFF
_0x00_LCM_COMMAND_DISPLAY_ONOFF_B_LOW	0x00	Display ON/OFF control: Cursor blinking OFF
_0x20_LCM_COMMAND_FUNCTION_SET	0x20	Command: Function Set
_0x10_LCM_COMMAND_FUNCTION_SET_DL_HIGH	0x10	Function Set: Mpu 8-bit bus mode
_0x08_LCM_COMMAND_FUNCTION_SET_N_HIGH	0x08	Function Set: 2-line display
_0x00_LCM_COMMAND_FUNCTION_SET_F_LOW	0x00	Function Set: 5x8-dot font
_0x80_LCM_COMMAND_SET_DDRAM_ADDRESS	0x80	Command: Set DDRAM Address

Table 4-3 Constants (2/2)

Constant Name	Setting Value	Description
LCM_COMMAND_EXEC_WAIT	26600	LCD module command execution wait time 5 ms (32 MHz operation)
LCM_CONFIG_FUNCTION_SET_PARAMS	0x18	Parameters for function set _0x10_LCM_COMMAND_FUNCTION_SET_DL_HIGH | _0x08_LCM_COMMAND_FUNCTION_SET_N_HIGH | _0x00_LCM_COMMAND_FUNCTION_SET_F_LOW
LCM_CONFIG_ENTRY_MODE_SET_PARAMS	0x20	Parameters for entry mode set _0x02_LCM_COMMAND_ENTRY_MODE_SET_ID_HIGH | _0x00_LCM_COMMAND_ENTRY_MODE_SET_S_LOW
LCM_CONFIG_DISPLAY_ONOFF_PARAMS	0x40	Display ON/OFF control command parameters _0x04_LCM_COMMAND_DISPLAY_ONOFF_D_HIGH | _0x00_LCM_COMMAND_DISPLAY_ONOFF_C_LOW | _0x00_LCM_COMMAND_DISPLAY_ONOFF_B_LOW
LCM_CONFIG_MAX_CHAR_PER_LINE	16	Maximum number of characters per line
LCM_CONFIG_WAIT_COUNT	13	IIIA0 wait count
LCM_POSITION_TOP	0x00	LCD module display line (top)
LCM_POSITION_BOTTOM	0x40	LCD module display line (bottom)
R_RTC_INIT_SEC	0x55	Initial value of the current clock time (second)
R_RTC_INIT_MIN	0x59	Initial value of the current clock time (minute)
R_RTC_INIT_HOUR	0x15	Initial value of the current clock time (hour)
R_RTC_INIT_WEEK	0x06	Initial value of the current clock time (day of the week)
R_RTC_INIT_DAY	0x01	Initial value of the current clock time (day)
R_RTC_INIT_MONTH	0x01	Initial value of the current clock time (month)
R_RTC_INIT_YEAR	0x21	Initial value of the current clock time (year)
R_RTC_ALARM_MIN	0x00	Set value of alarm occurrence time (minute)
R_RTC_ALARM_HOUR	0x16	Set value of alarm occurrence time (hour)
R_RTC_ALARM_WEEK	0xFF	Set value of alarm occurrence time (day of the week)
R_INTERRUPT_OFF	0	Interrupt flag cleared
R_INTERRUPT_ON	1	Interrupt flag ON

4.3 List of Variables

Table 4-4 lists global variables.

Table 4-4 Global Variables

Type	Variable Name	Description	Function Used
uint8_t	g_rtc_constperiod	Fixed-cycle interrupt flag	main, r_Config_RTC_Create_UserInit, r_Config_RTC_callback_constperiod, r_rtc_is_constperiod_flag_on, r_rtc_clear_constperiod_flag
uint8_t	g_LCM_is_sendend	I2C communication (with the LCD module) completion flag	r_LCM_init, r_LCM_turn_sendend_on, r_LCM_wait_sendend

4.4 List of Functions

Table 4-5 shows a list of functions.

Table 4-5 Functions

Function Name	Outline
R_Config_RTC_Create_UserInit()	User-specified RTC initialization processing
r_Config_RTC_callback_alarm()	Alarm interrupt processing.
r_Config_RTC_callback_constperiod()	Fixed-cycle interrupt processing.
r_rtc_init_current_time()	Initial setting for the current clock time
r_rtc_init_alarm_time()	Initial setting for alarm time
r_rtc_is_constperiod_flag_on()	Check the fixed-cycle interrupt flag.
r_rtc_clear_constperiod_flag()	Clear the fixed-cycle interrupt flag.
r_rtc_display_current_time()	Display the current clock time on the LCD.
convert_BCD_to_2chars()	Convert 2-digit BCD to two characters.
convert_week_to_3chars()	Convert day-of-the-week code to three characters.
r_Config_IICA0_callback_master_sendend()	IICA0 send end callback processing.
r_Config_IICA0_callback_master_error()	IICA0 error callback processing.
R_Config_PORT_Create_UserInit()	User-specified port initialization processing
r_LCM_init()	LCD module initialization.
r_LCM_clear()	LCD module display clear processing
r_LCM_send_string()	LCD module character string transmission processing
r_LCM_send_command()	LCD module command transmission processing
r_LCM_send_data()	LCD module data transmission processing
r_LCM_turn_sendend_on()	LCD module communication end flag setting
r_LCM_wait_sendend()	LCD module communication end wait processing

4.5 Specification of Function

The function specifications of the sample code are shown below.

R_Config_RTC_Create_UserInit()	
Outline	User-specified RTC initialization processing
Header	r_cg_userdefine.h
Declaration	void R_Config_RTC_Create_UserInit(void);
Description	Perform user-specified processing among initialization required before starting the RTC.
Argument	None
Return Value	None

r_Config_RTC_callback_alarm()	
Outline	Alarm interrupt processing.
Header	r_cg_macrodriver.h
Declaration	static void r_Config_RTC_callback_alarm(void);
Description	This callback function is called when an RTC alarm interrupt is generated.
Argument	None
Return Value	None

r_Config_RTC_callback_constperiod()	
Outline	Fixed-cycle interrupt processing.
Header	r_cg_macrodriver.h, r_cg_userdefine.h
Declaration	static void r_Config_RTC_callback_constperiod(void);
Description	This callback function is called when an RTC fixed-cycle interrupt is generated.
Argument	None
Return Value	None

r_rtc_init_current_time()	
Outline	Initial setting for the current clock time
Header	r_cg_macrodriver.h, r_cg_userdefine.h
Declaration	void r_rtc_init_current_time(void);
Description	Set the current clock time in the RTC register.
Argument	None
Return Value	None

r_rtc_init_alarm_time()	
Outline	Initial setting for the alarm clock time
Header	r_cg_macrodriver.h, r_cg_userdefine.h, Config_RTC.h
Declaration	void r_rtc_init_alarm_time(void);
Description	Set the alarm time in the RTC register.
Argument	None
Return Value	None

r_rtc_is_constperiod_flag_on()

Outline	Check the fixed-cycle interrupt flag.
Header	r_cg_userdefine.h
Declaration	uint8_t r_rtc_is_constperiod_flag_on(void);
Description	Check g_ rtc_constperiod and return the check result.
Argument	None
Return Value	1: g_rtc_constperiod is R_INTTERUPT_ON 0: g_ rtc_constperiod is R_INTERRUPT_OFF

r_rtc_clear_constperiod_flag()

Outline	Clear the fixed-cycle interrupt flag.
Header	r_cg_userdefine.h
Declaration	void r_rtc_clear_constperiod_flag(void);
Description	Clear g_ rtc_constperiod (set to R_INTERRUPT_OF).
Argument	None
Return Value	None

convert_BCD_to_2chars()

Outline	Conversion of two-digit BCD value to two characters
Header	r_cg_macrodriver.h
Declaration	static void convert_BCD_to_2chars(uint8_t bcd, uint8_t * const str);
Description	Convert a two-digit BCD value to two characters.
Argument	uint8_t bcd: Two-digit BCD value to be converted (0x00 to 0x99) uint8_t *str: Area to store converted characters (‘¥0’ is not added to the end.)
Return Value	None

convert_week_to_3chars()

Outline	Conversion of day-of-the-week value to three characters
Header	r_cg_macrodriver.h
Declaration	static void convert_week_to_3chars(uint8_t week, uint8_t * const str);
Description	Convert a day-of-the-week value to three characters. The conversion result is as follows:

Source value	0	1	2	3	4	5	6
Conversion result	Sun	Mon	Tue	Wed	Thu	Fri	Sat

Argument	uint8_t week: Day-of-the-week value to be converted (0 to 6) uint8_t *str: Area to store converted characters (‘¥0’ is not added to the end.)
Return Value	None

r_Config_IICA0_callback_master_sendend()	
Outline	IICA0 send end callback processing.
Header	r_cg_macrodriver.h, Config_IICA0.h, LCM_driver.h
Declaration	static void r_Config_IICA0_callback_master_receiveend(void);
Description	Callback function called when an IICA0 transmission completion interrupt is generated Generate stop conditions, and then call the LCD module communication end flag setting function.
Argument	None
Return Value	None

r_Config_IICA0_callback_master_error()	
Outline	IICA0 error callback processing.
Header	r_cg_macrodriver.h, Config_IICA0.h, LCM_driver.h
Declaration	static void r_Config_IICA0_callback_master_error(MD_STATUS flag);
Description	Callback function called when an IICA0 transmission error interrupt is generated Call the LCD module communication end flag setting function.
Argument	MD_STATUS flag : error type
Return Value	None

R_Config_PORT_Create_UserInit()	
Outline	User-specified port initialization processing
Header	r_cg_macrodriver.h, Config_PORT.h
Declaration	void R_Config_PORT_Create_UserInit(void);
Description	Perform user-specified processing among initialization required before using the port.
Argument	None
Return Value	None

r_LCM_init()	
Outline	LCD module Initialization
Header	LCM_driver.h, Config_IICA0.h
Declaration	void r_LCM_init(void);
Description	Initializes LCD module.
Argument	None
Return Value	None

r_LCM_clear()	
Outline	LCD module display clear processing
Header	LCM_driver.h, Config_IICA0.h
Declaration	void r_LCM_clear(void);
Description	Transmit the display clear command to the LCD module.
Argument	None
Return Value	None

r_LCM_send_string()	
Outline	LCD module character string transmission processing
Header	LCM_driver.h, Config_IICA0.h
Declaration	void r_LCM_send_string(uint8_t * const str, lcm_position_t pos);
Description	Display the character string transferred with str on the LCD module. The line to display the character string is specified by pos.
Argument	uint8_t * const str: Character string to be displayed lcm_position_t pos: LCM_POSITION_TOP: Displayed at the top LCM_POSITION_BOTTOM: Displayed at the bottom
Return Value	None

r_LCM_send_command()	
Outline	LCD module command transmission processing
Header	LCM_driver.h, Config_IICA0.h
Declaration	void r_LCM_send_command(uint8_t command);
Description	Send the command transferred with command to the LCD module.
Argument	uint8_t command: Command to be sent to the LCD module
Return Value	None

r_LCM_send_data()	
Outline	LCD module data transmission processing
Header	LCM_driver.h, Config_IICA0.h
Declaration	void r_LCM_send_data(uint8_t data);
Description	Send the data transferred with data to the LCD module.
Argument	uint8_t data: Data to be sent to the LCD module
Return Value	None

r_LCM_turn_sendend_on()	
Outline	LCD module communication end flag setting
Header	LCM_driver.h, Config_IICA0.h
Declaration	void r_LCM_turn_sendend_on(void);
Description	Set the I2C communication (with the LCD module) end flag for g_LCM_is_sendend.
Argument	None
Return Value	None

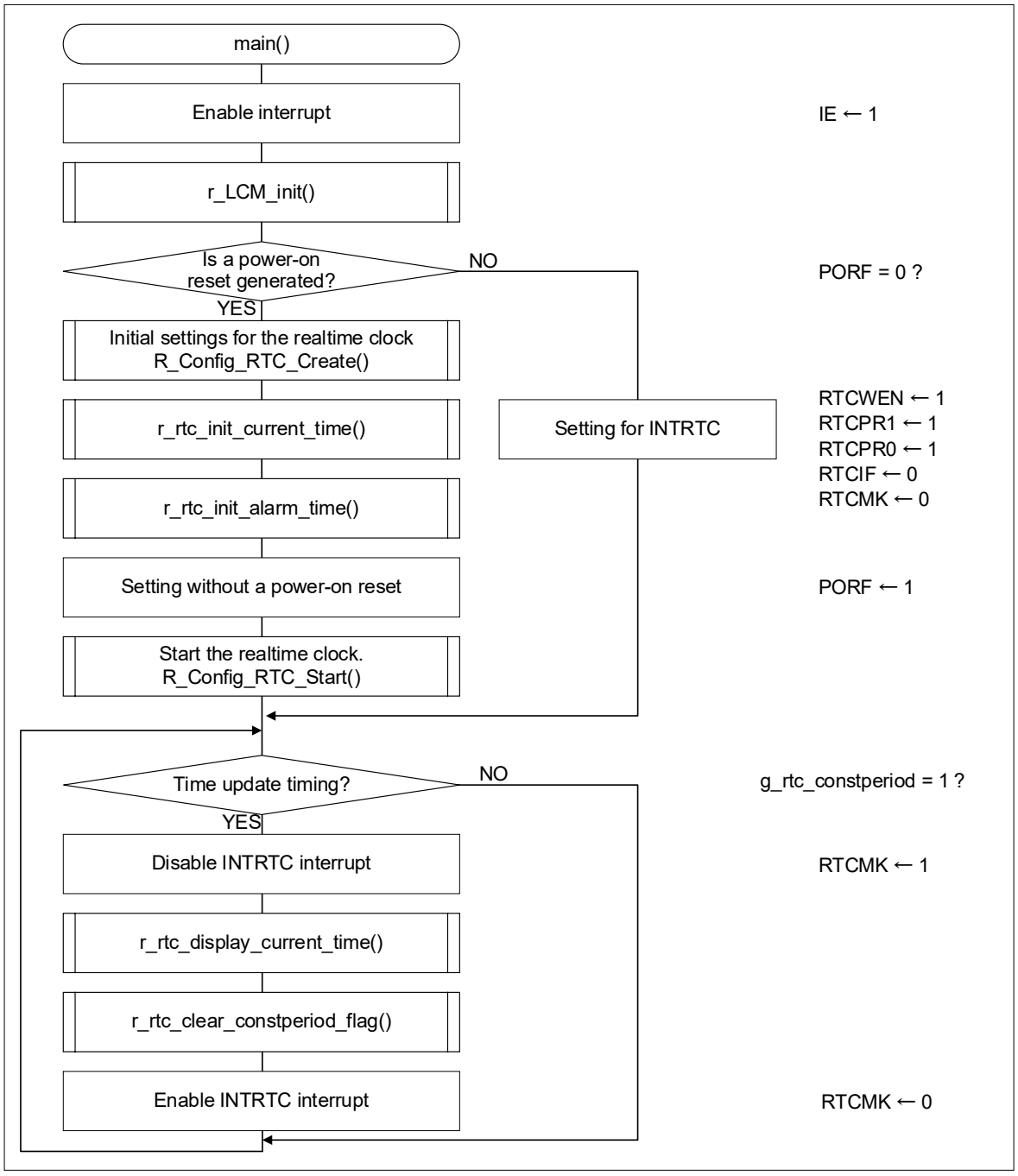
r_LCM_wait_sendend()	
Outline	LCD module communication end wait processing
Header	LCM_driver.h, Config_IICA0.h
Declaration	static void r_LCM_wait_sendend(void);
Description	Wait until the I2C communication (with the LCD module) ends, and then perform wait processing during the command execution wait time period (5 ms).
Argument	None
Return Value	None

4.6 Flowcharts

4.6.1 Main Processing

Figure 4-1 shows the flowchart of the main processing.

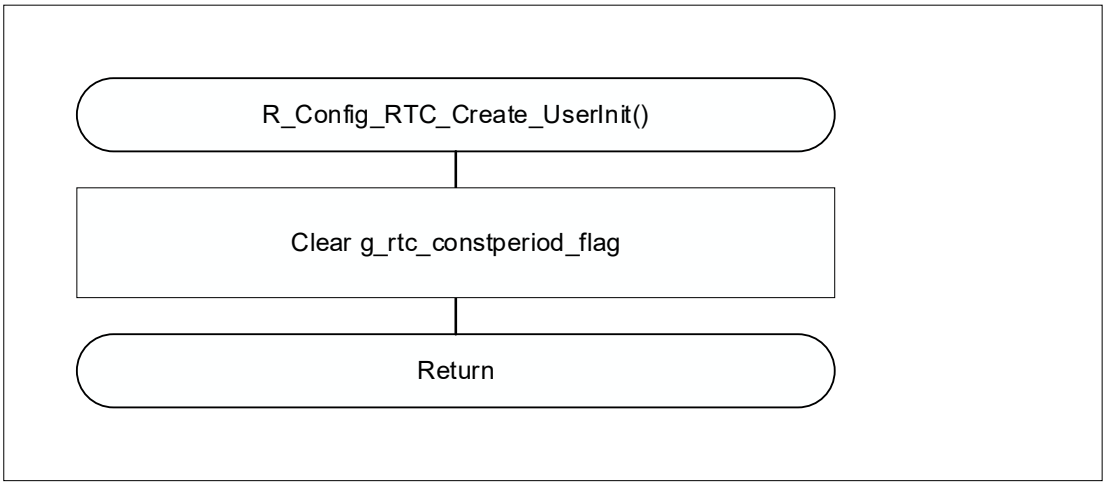
Figure 4-1 Main Processing



4.6.2 RTC Initialization Processing (User-Defined)

Figure 4-2 shows the flowchart of the user-defined RTC initialization processing.

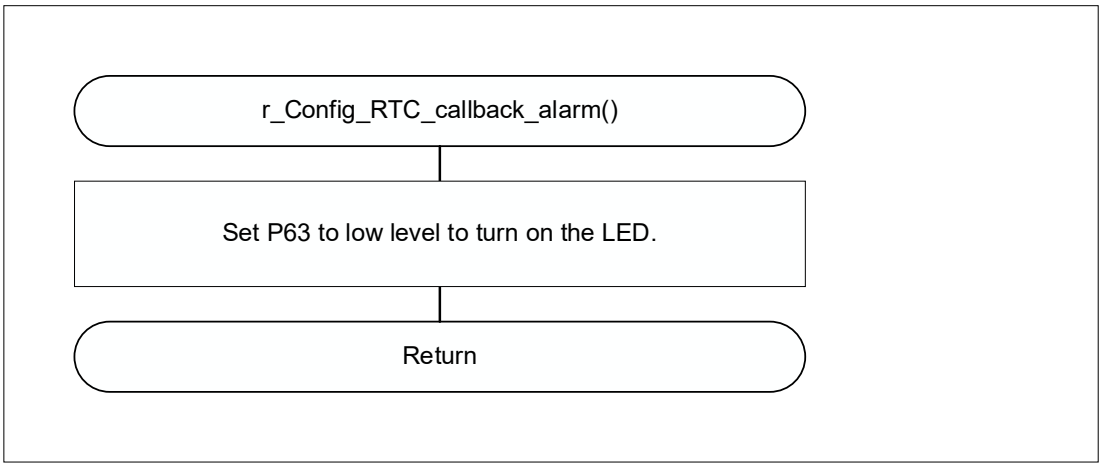
Figure 4-2 RTC Initialization Processing (User-Defined)



4.6.3 Alarm Interrupt Processing

Figure 4-3 shows the flowchart of the alarm interrupt processing.

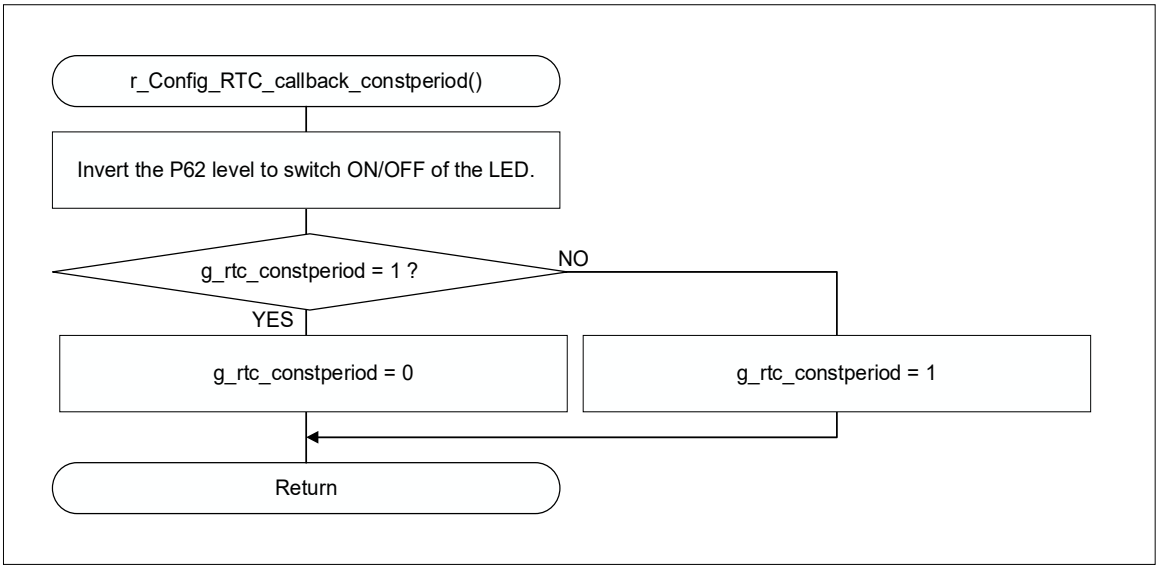
Figure 4-3 Alarm Interrupt Processing



4.6.4 Fixed-cycle Interrupt Processing

Figure 4-4 shows the flowchart of the fixed-cycle interrupt processing.

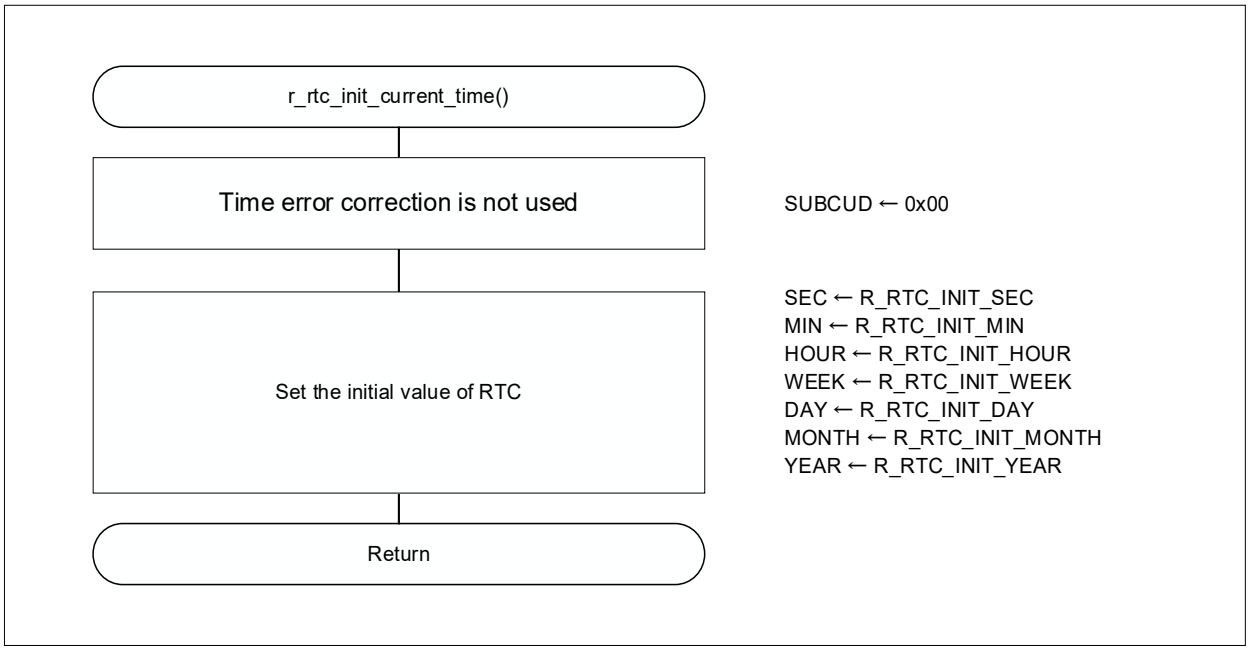
Figure 4-4 Fixed-cycle Interrupt Processing



4.6.5 Initial Settings for the Current Clock Time

Figure 4-5 shows the flowchart of the initial settings for the current clock time.

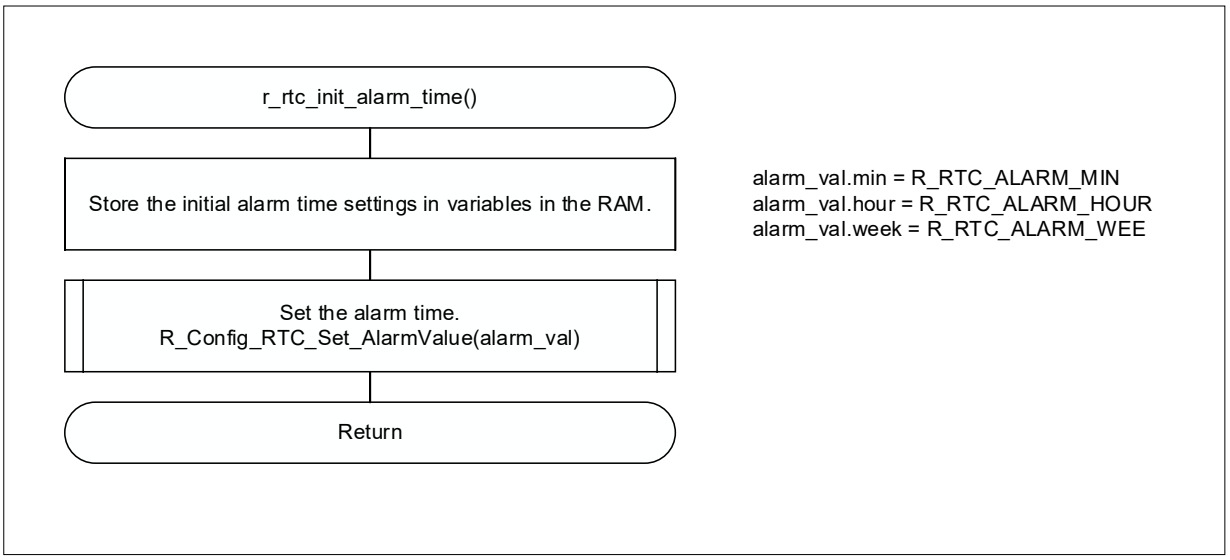
Figure 4-5 Initial Settings for the Current Clock Time



4.6.6 Initial Settings for Alarm Time

Figure 4-6 shows the flowchart of the initial settings for alarm time.

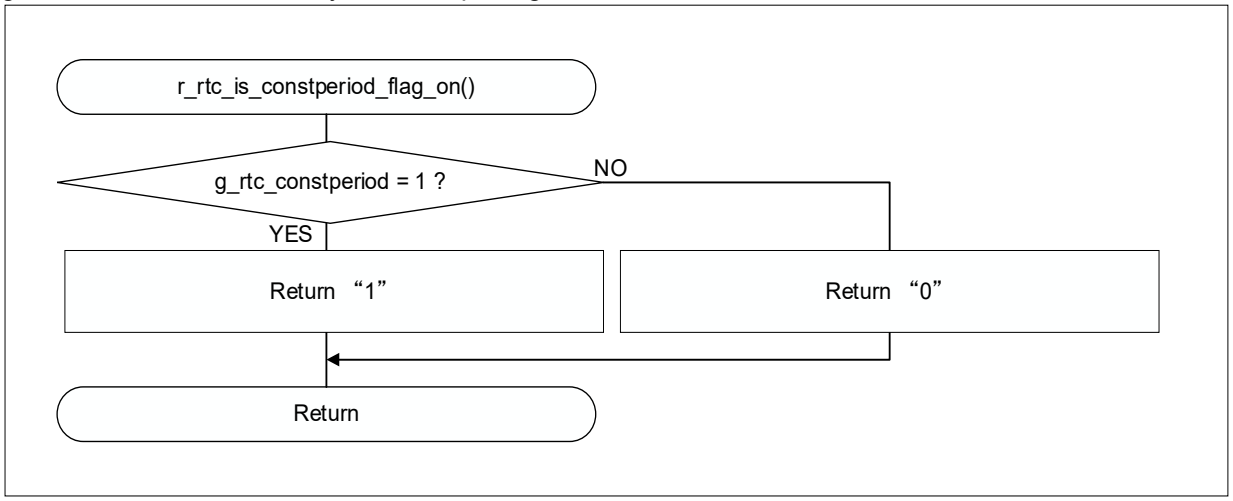
Figure 4-6 Initial Settings for Alarm Time



4.6.7 Check the Fixed-Cycle Interrupt Flag

Figure 4-7 shows the flowchart of the check the fixed-cycle interrupt flag.

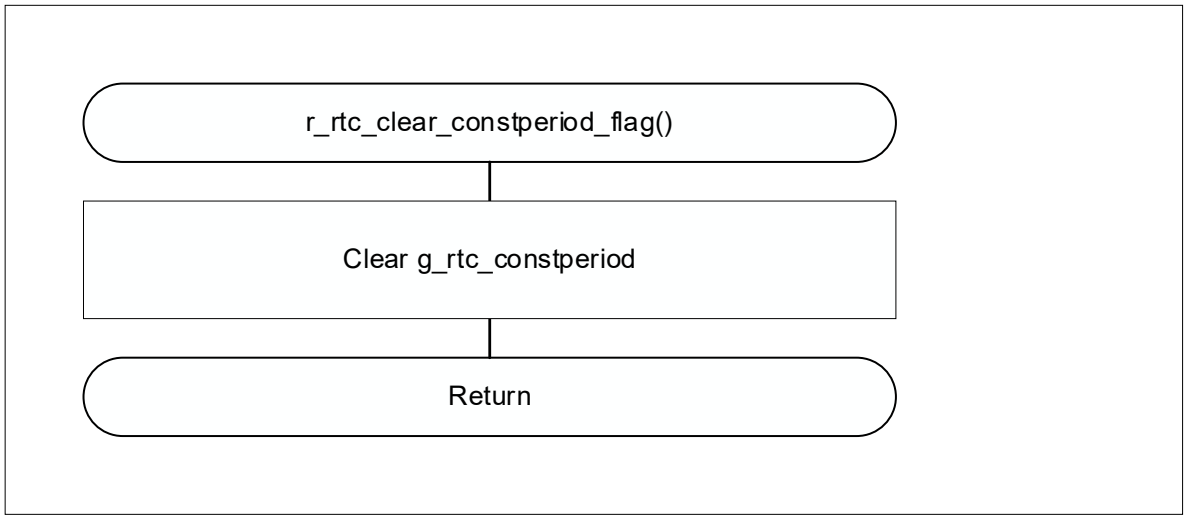
Figure 4-7 Check the Fixed-Cycle Interrupt Flag



4.6.8 Clear the Fixed-Cycle Interrupt Flag

Figure 4-8 shows the flowchart of the clear the fixed-cycle interrupt flag.

Figure 4-8 Clear the Fixed-Cycle Interrupt Flag



4.6.9 Current Clock Time Display Processing

Figure 4-9 and Figure 4-10 show the flowcharts of the current clock time display processing.

Figure 4-9 Current Clock Time Display Processing (1/2)

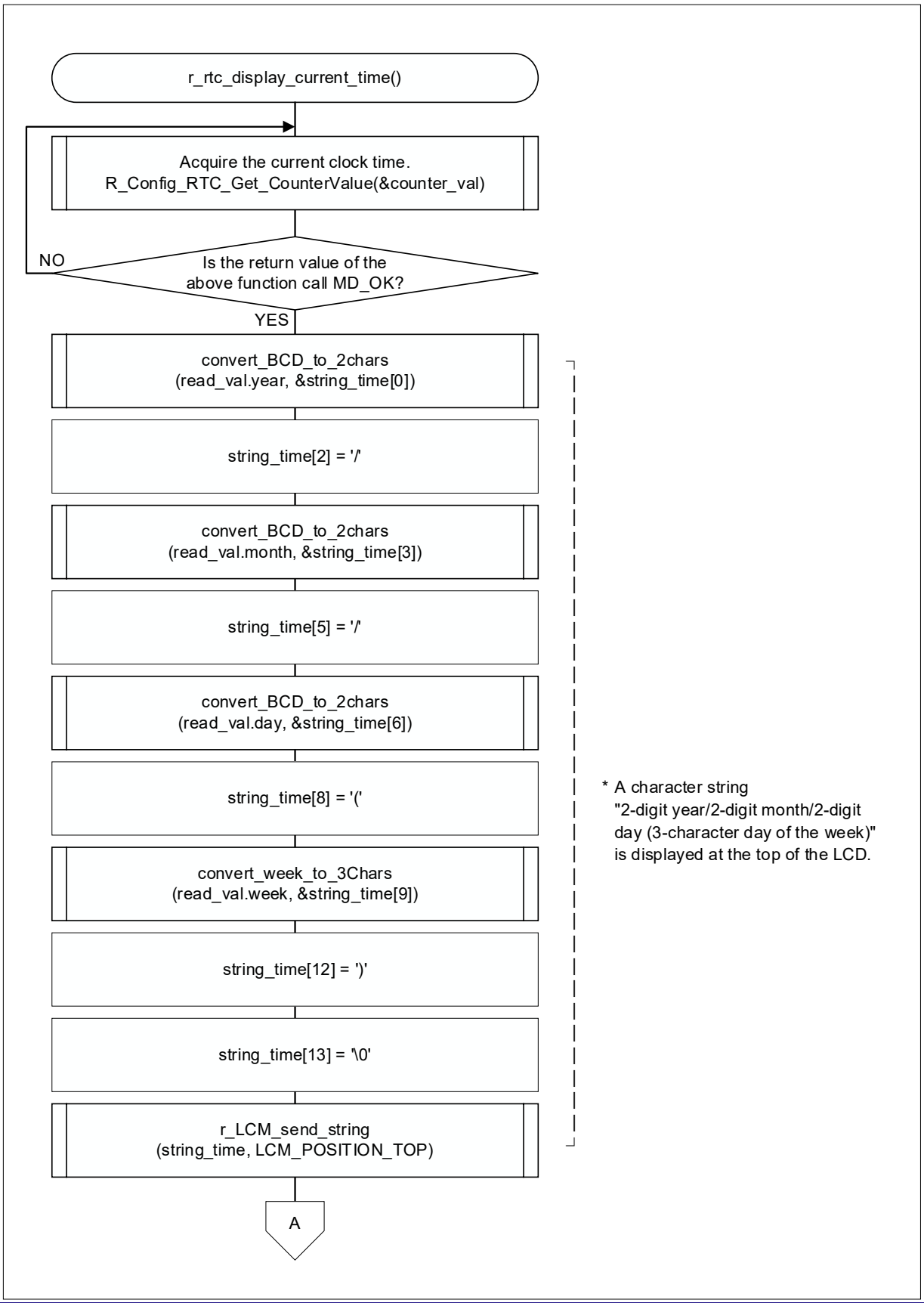
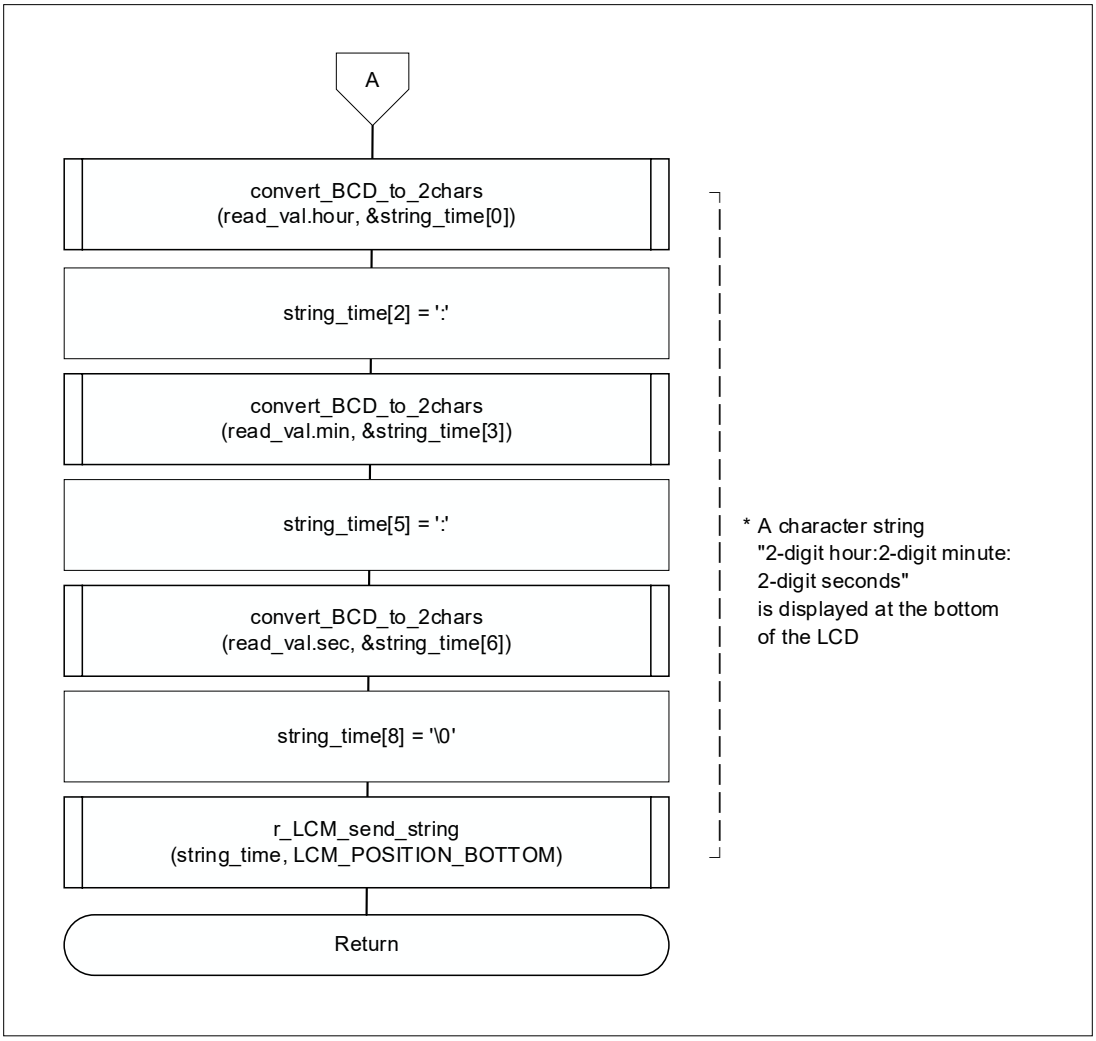


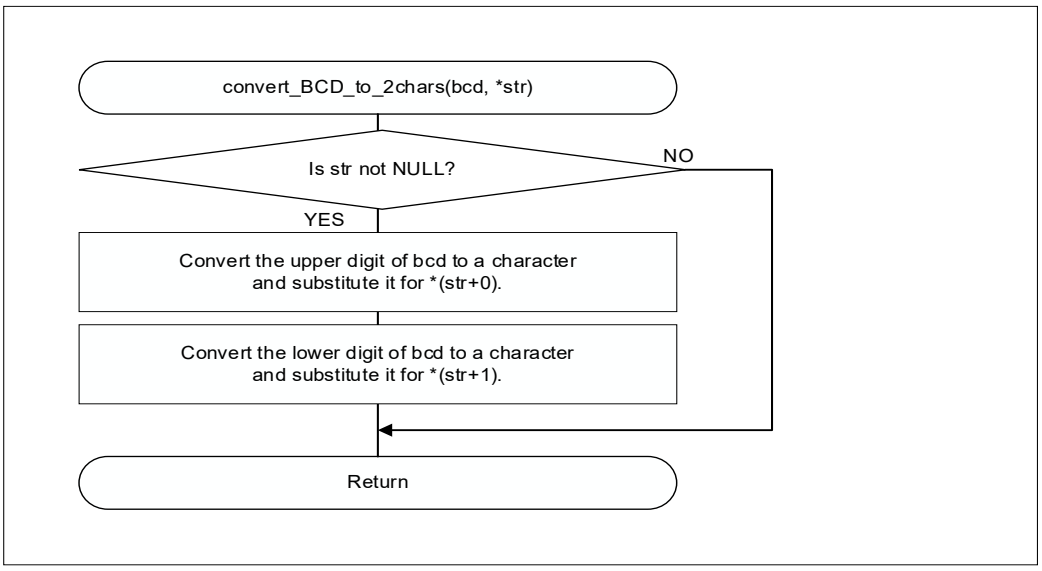
Figure 4-10 Current Clock Time Display Processing (2/2)



4.6.10 BCD-to-Character Conversion Processing

Figure 4-11 shows the flowchart of the processing for conversion from BCD to character.

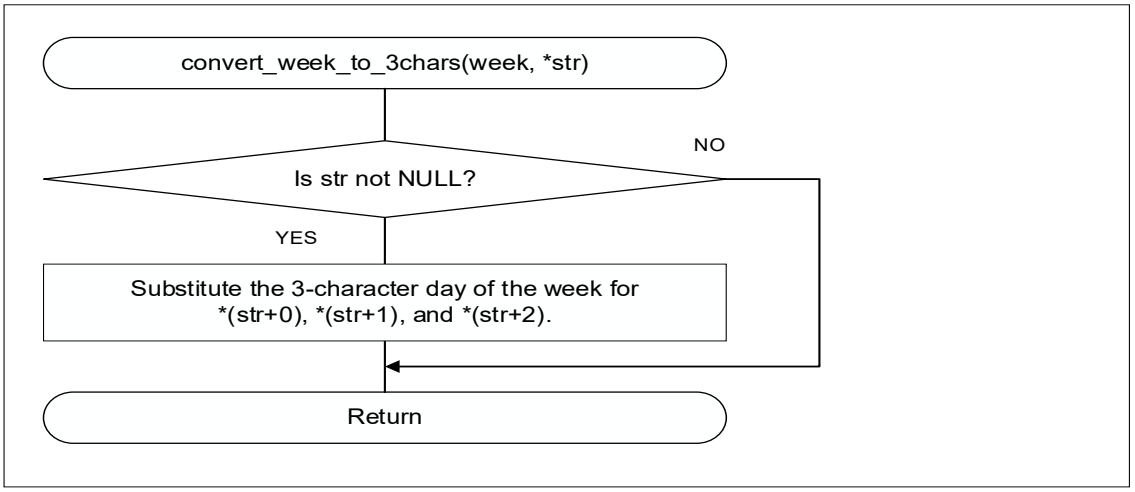
Figure 4-11 BCD-to-Character Conversion Processing



4.6.11 Day of the Week-to-Character Conversion Processing

Figure 4-12 shows the flowchart of the processing for conversion from day of the week to character.

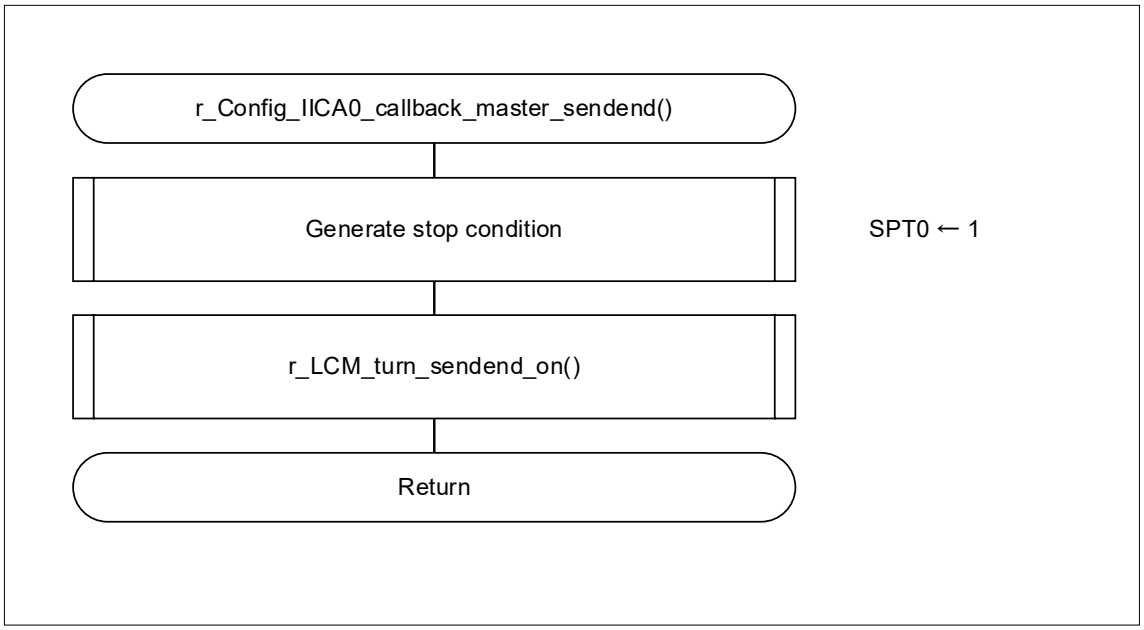
Figure 4-12 Day of the Week-to-Character Conversion Processing



4.6.12 IICA0 Send End Callback Processing

Figure 4-13 shows the flowchart of the IICA0 send end callback processing.

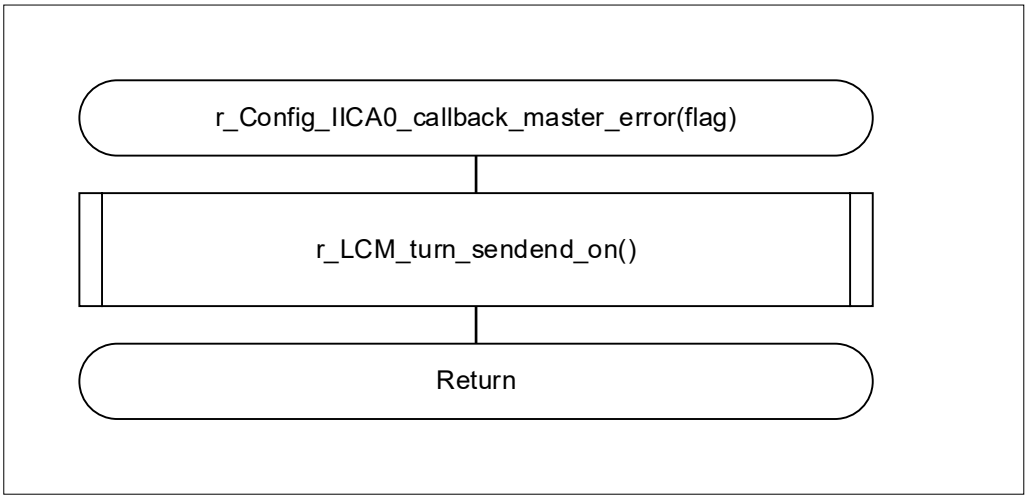
Figure 4-13 IICA0 Send End Callback Processing



4.6.13 IICA0 Error Callback Processing

Figure 4-14 shows the flowchart of the IICA0 error callback processing.

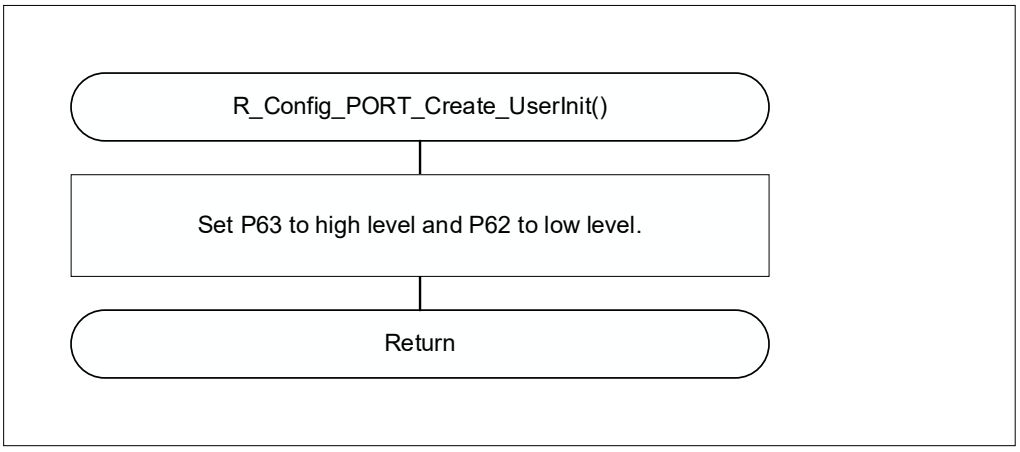
Figure 4-14 IICA0 Error Callback Processing



4.6.14 Port Initialization Processing (User-Defined)

Figure 4-15 shows the flowchart of the user-defined port initialization processing.

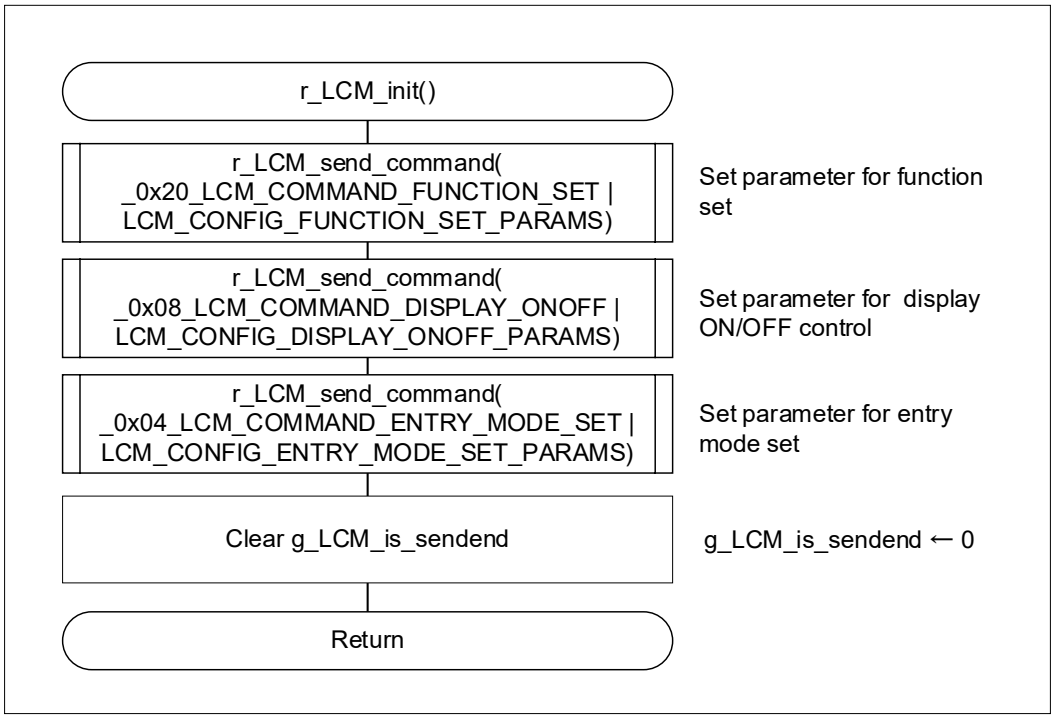
Figure 4-15 Port Initialization Processing (User-Defined)



4.6.15 LCD Module Initialization

Figure 4-16 shows the flowchart of the LCD module initialization.

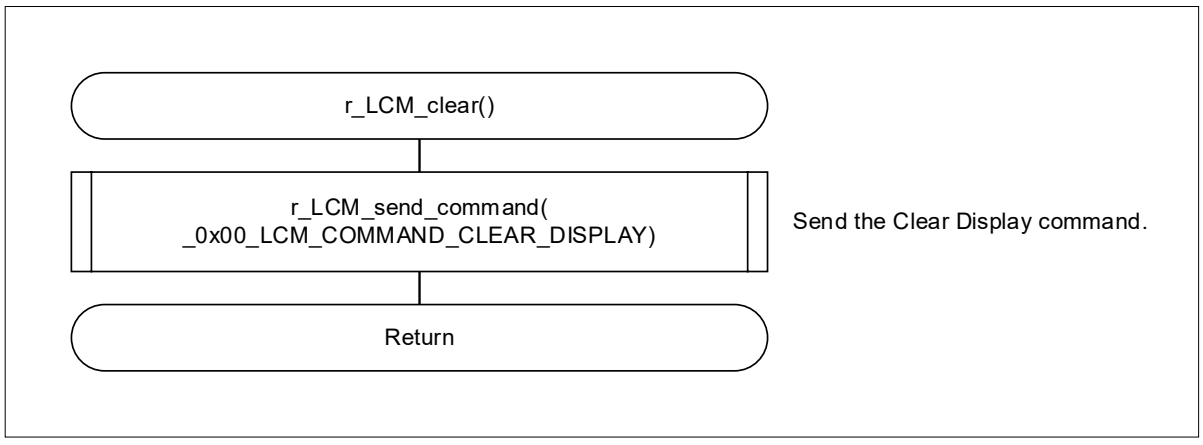
Figure 4-16 LCD Module Initialization



4.6.16 LCD Module Display Clear Processing

Figure 4-17 shows the flowchart of the display clear processing for the LCD module.

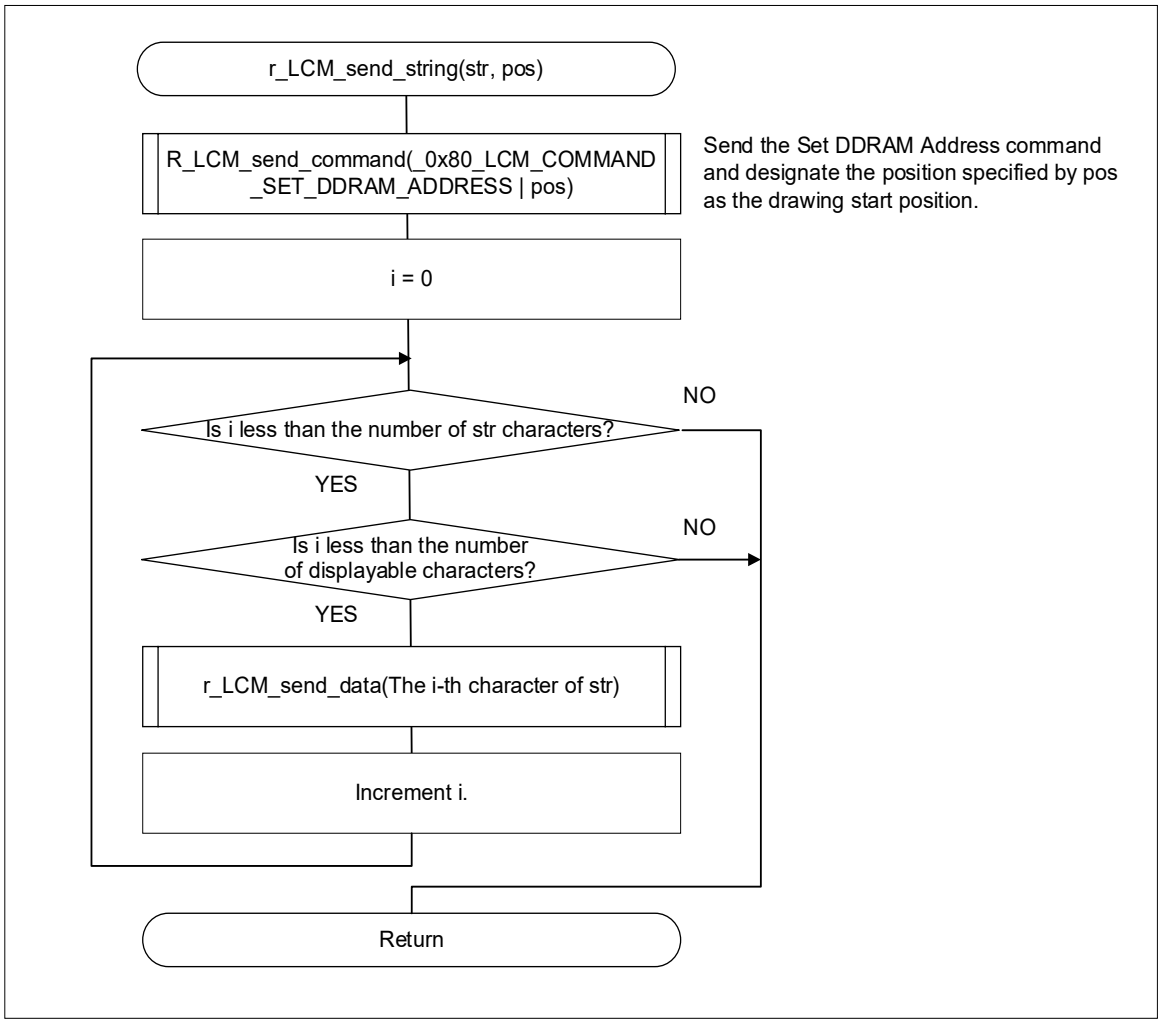
Figure 4-17 LCD Module Display Clear Processing



4.6.17 LCD Module Character String Transmission Processing

Figure 4-18 shows the flowchart of the character string transmission processing for the LCD module.

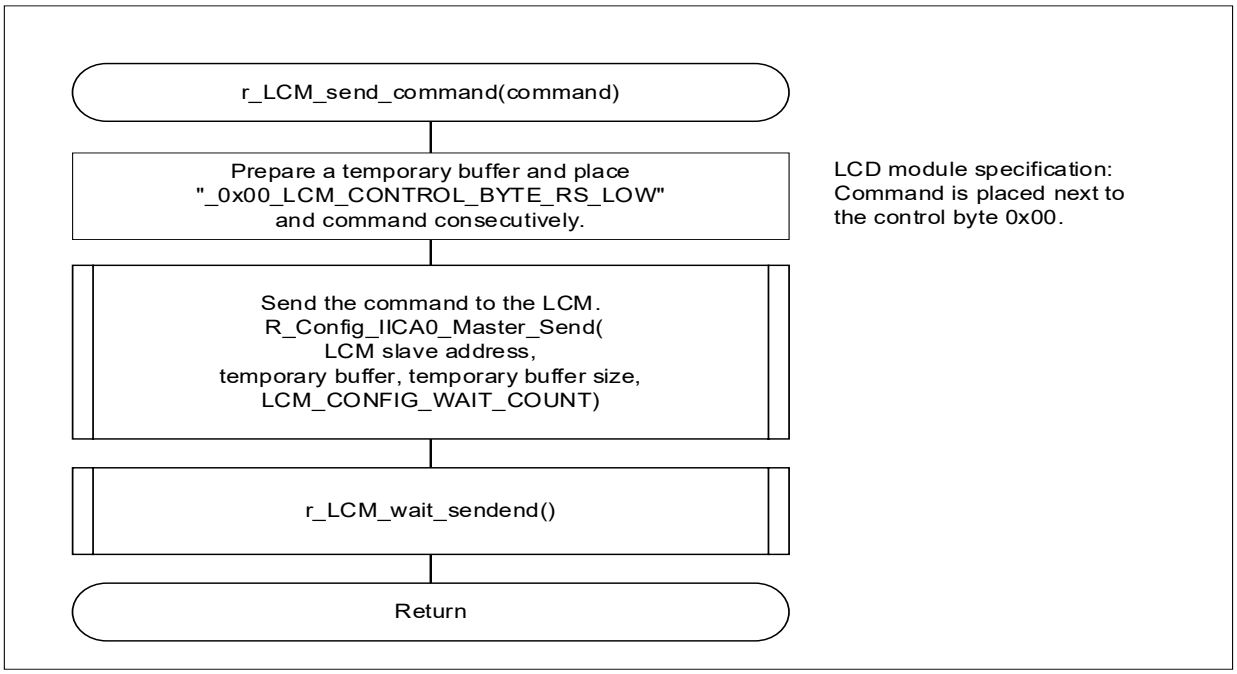
Figure 4-18 LCD Module Character String Transmission Processing



4.6.18 LCD Module Command Transmission Processing

Figure 4-19 shows the flowchart of the command transmission processing for the LCD module.

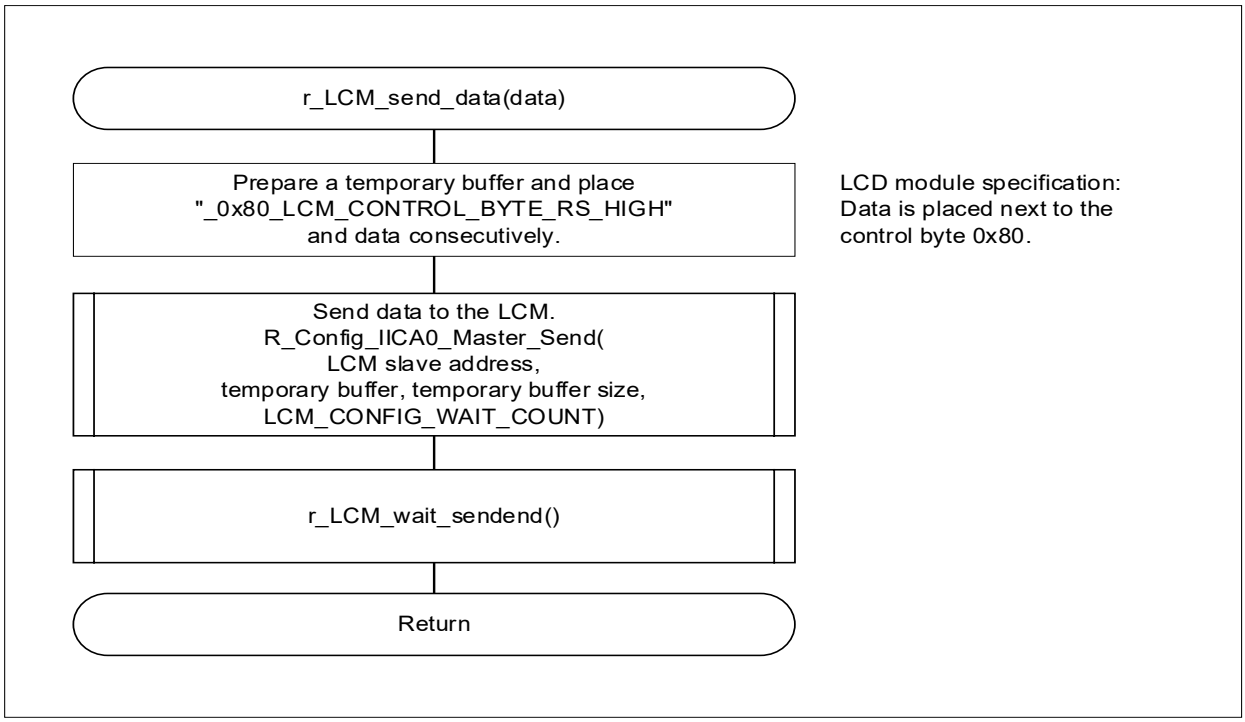
Figure 4-19 LCD Module Command Transmission Processing



4.6.19 LCD Module Data Transmission Processing

Figure 4-20 shows the flowchart of the data transmission processing for the LCD module.

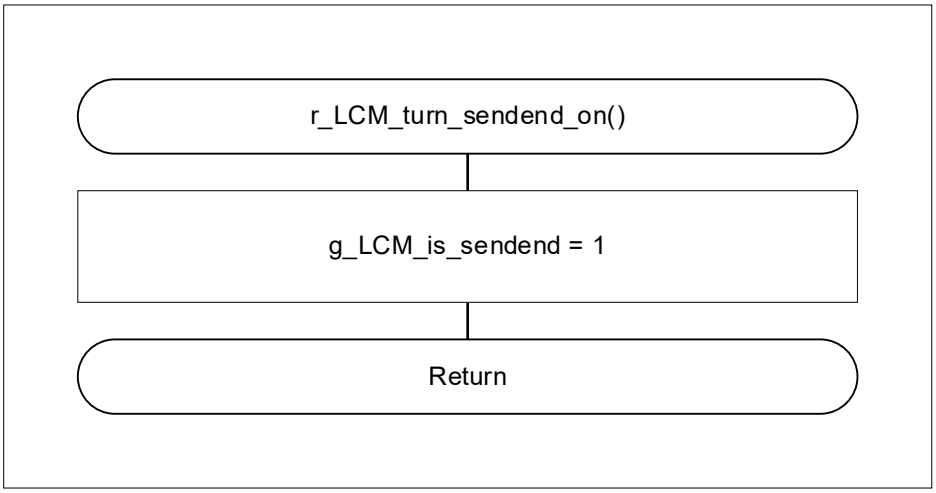
Figure 4-20 LCD Module Data Transmission Processing



4.6.20 LCD Module Communication End Flag Setting

Figure 4-21 shows the flowchart for setting the communication end flag for the LCD module.

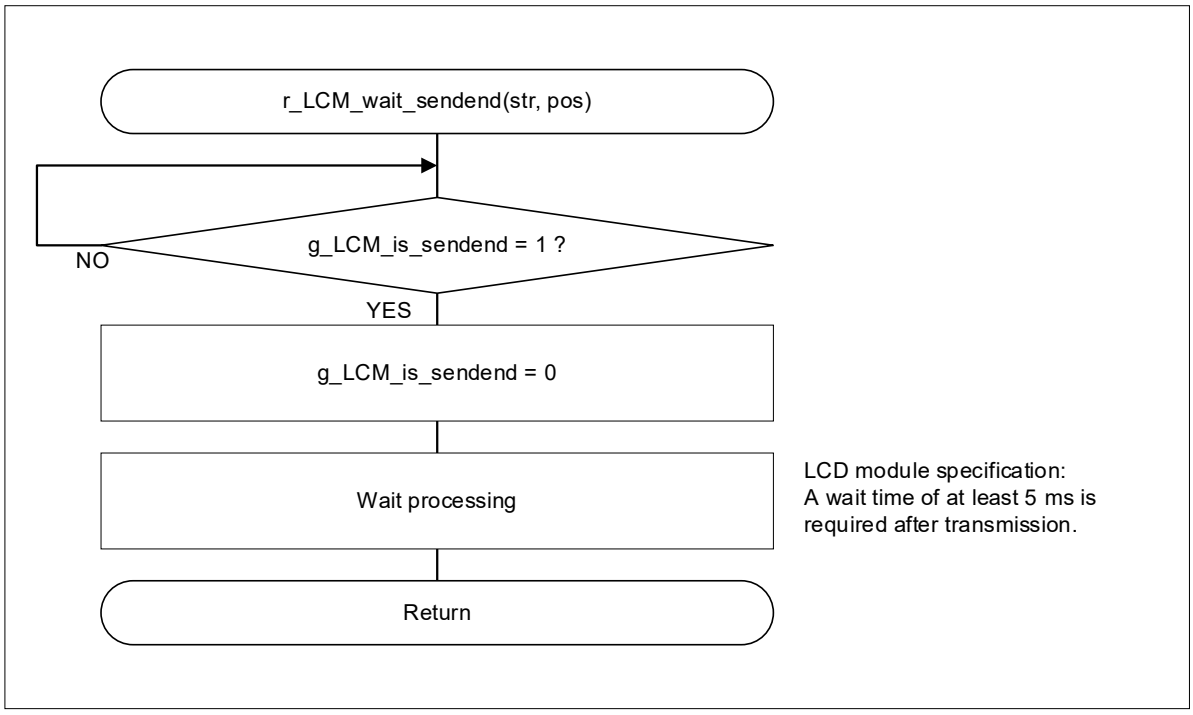
Figure 4-21 LCD Module Communication End Flag Setting



4.6.21 LCD Module Communication End Wait Processing

Figure 4-22 shows the flowchart of the communication end flag processing for the LCD module.

Figure 4-22 LCD Module Communication End Wait Processing



5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

RL78/G22 User’s Manual: Hardware (R01UH0978J)
RL78 family user's manual software (R01US0015)
The latest versions can be downloaded from the Renesas Electronics website.

Technical update
The latest versions can be downloaded from the Renesas Electronics website.
LCD module datasheet
(ACM1602NI-FLW-FBW-M01 (ZETTLER DISPLAYS) CHARACTER MODULE VER1.4)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2024.09.24	-	First Edition
2.00	2024.12.11	5	Updated Development Environment

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact information

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

Trademarks

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