RL78/G11

Indoor air quality alarm GCC

Introduction

This application note describes how to measure Indoor Air Quality (IAQ) with the metal-oxide (MOX) gas sensor ZMOD4410 using RL78/G11.

The RL78/G11 sends I2C commands through the Serial Interface IICA to control ZMOD4410 and LCD character display.

Target Device

RL78/G11

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

In this application note, the RL78/G11 controls the gas sensor module ZMOD4410 of IDT to measure indoor air quality.

Calculate the IAQ rating from the output of ZMOD4410 with the software library of IDT and display the IAQ rating on LCD character display.

Turn on the alarm LED According to the IAQ rating.

Table 1.1 Used Peripheral Functions and those Purposes

Peripheral Functions	Purposes
IICA0	To communicate with the ZMOD4410
IICA1	To communicate with the LCD Character Display

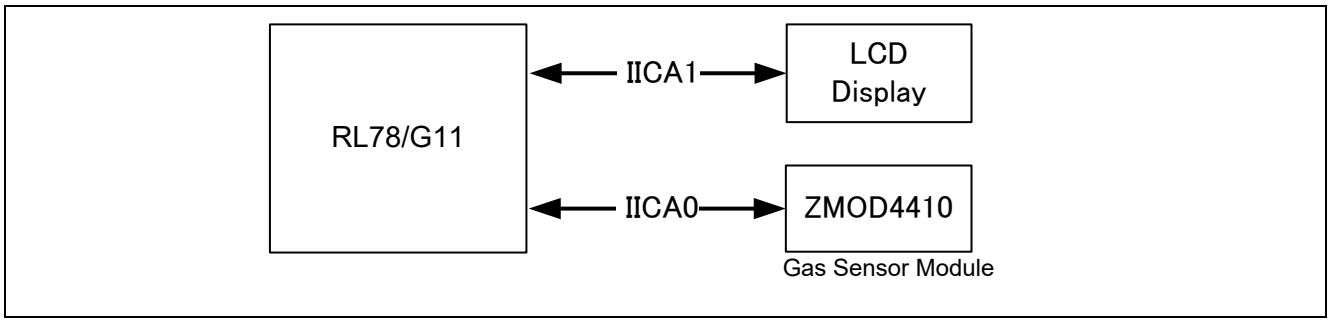


Figure 1.1 Outline of Indoor-Air Quality System

1.1 Indoor Air Quality (IAQ)

Indoor Air Quality (IAQ) indicates the cleanliness, comfortability of the air for human body.

The IDT's indoor air quality sensor (ZMOD4410) measures the total volatile organic compounds (TVOC) in the air. And applying the IDT evaluation enables to indicate the indoor air quality.

Please refer to the ZMOD4410 manual for the evaluation index.

1.2 I2C Data Communication Protocol for ZMOD4410

In this application note, I2C is used to communicate between the RL78/G11 and the ZMOD4410.

The RL78/G11 sets operation mode of ZMOD4410 and receives measurement results which is stored in internal RAM of the ZMOD4410.

ZMOD4410 supports I2C standard mode and fast mode, and the slave address is 32H by default. Please refer to the ZMOD4410 manual for the data protocol structure.

1.3 Software Library for ZMOD4410

Please get the software library for ZMOD4410 and the details of the library from the IDT's website.

https://en.idt.com/products/sensor-products/gas-sensors/zmod4410-indoor-air-quality-sensor-platform (provided on May 14, 2019)

In this application note, the library extension is renamed from ".lib" to ".a".

To compress source code, printf functions in the sample program bundled with the library are commented out, and GCC compiler optimization option (-O1) is used for the entire project.

In addition, Continuous-mode is used as the operation mode of ZMOD4410 (Low-power-mode is not used).

1.4 LCD Character Display

For the LCD character display, I2C connection and 16 × 2 display HD44780 compatible model is used.

The calculated IAQ rating is displayed in the format shown in Figure 1.2.

M	E	A	S	U	R	E	M	E	N	T					
I	A	Q	:		3	.	4	5							

Figure 1.2 Display Pattern of the LCD Character Display.

1.5 Warning Lights

Toggle the three alarm LEDs as shown in Table 1.2 according to calculated IAQ rating from ZMOD4410's measurement result.

Table 1.2 Lighting Pattern of Warning Lights

IAQ rating	TVOC (mg/m3)	LED1 (GREEN)	LED2 (YELLOW)	LED3 (RED)
≤ 2.99	< 1.0	ON	OFF	OFF
3.00 ～ 4.99	1.0 ～ 10.0	OFF	ON	OFF
≥ 5.00	> 10.0	OFF	OFF	ON

2. Conditions of Operation Confirmation Test

The sample code with this application note runs properly under the conditions below.

Table 2.1 Operation Confirmation Conditions

Items	Contents
MCU	RL78/G11（R5F1056A）
Operating frequencies	High-speed on-chip oscillator (HOCO) clock: 48 [MHz] CPU/peripheral hardware clock: 24 [MHz]
Operating voltage	3.3 [V] LVD operations (VLVD): reset mode TYP. 2.75 [V] Rising edge 2.76 to 2.87 [V] Falling edge 2.70 to 2.81 [V]
Integrated development environment (e2studio)	e2studio V7.4.0 from Renesas Electronics Corp.
C compiler (e2studio)	GCC for Renesas RL78 4.9.2.201801 Optimization options: -O1
ZMOD4410 Software Libraries	IDT_ZMOD4410-Air-Quality-eCO2-Firmware_SWR_20190514 IDT_ZMOD4410-Odor-Firmware_SWR_20190514 Confirmed operation only in CONTINUOUS_MODE

Table 2.2 Parameters for Code Generation Tool

Items	Contents
IICA0	Transfer mode: Single master Count clock setting: fCLK/2 Local address setting: 16 Operation mode setting: standard, 50000 [bps] Communication end interrupt priority: Low Callback function setting: None
IICA1	Transfer mode: Single master Count clock setting: fCLK/2 Local address setting: 16 Operation mode setting: standard, 50000 [bps] Communication end interrupt priority: Low Callback function setting: None

3. Hardware

3.1 Example of Hardware Configuration

Figure 3.1 Shows an example of the hardware configuration used in this application note.

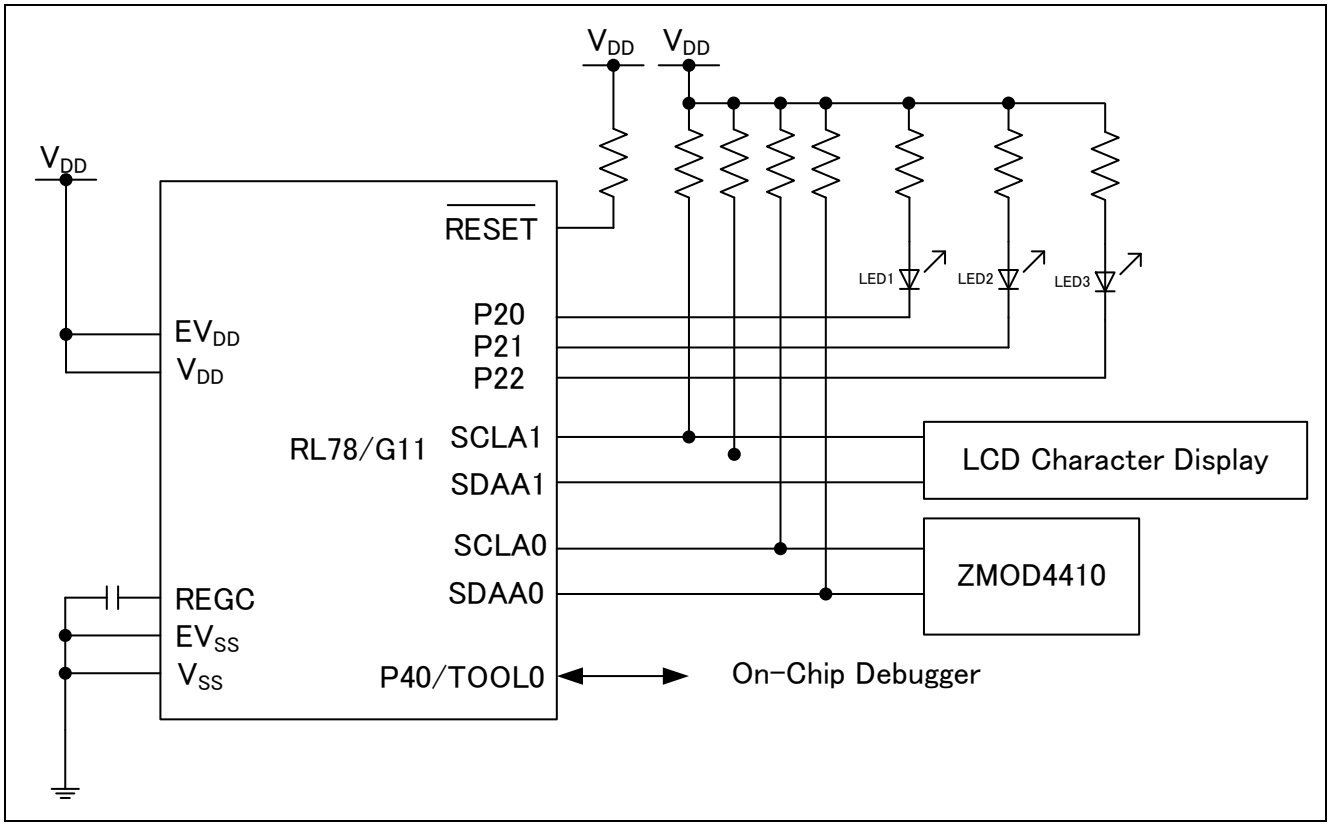


Figure 3.1 Hardware Configuration

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

2. V_{DD} must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

3.2 Used Pins

Table 3.1 shows list of used Pins and assignment functions.

Table 3.1 List of Pins and Functions

Pin Name	Input / Output	Function
SCLA0	Output	I²C serial clock (connected with ZMOD4410)
SDAA0	Output	I²C serial data bus (connected with ZMOD4410)
SCLA1	Output	I²C serial clock (connected with LCD Display)
SDAA1	Output	I²C serial data bus (connected with LCD Display)
P20	Output	LED1 (GREEN)
P21	Output	LED2 (YELLOW)
P22	Output	LED3 (RED)

4. Software

4.1 Folder Configuration

Table 4.1 shows a folder configuration of this application note.

Table 4.1 Folder configuration

Folder configuration	Outline	Bundled with application	Created by code generation tool
¥an-r01an4903xx0100-rl78g11 <DIR>	Root folder of this application note		
r01an4903xx0100-rl78g11.pdf	PDF file of this application note	✓	
¥src <DIR>	Folder for program source		
¥api <DIR>	Folder for ZMOD4410 software library APIs		
continuous.c	ZMOD4410 software library API		
continuous.h	ZMOD4410 software library API header		
zmod44xx_config.h	ZMOD4410 software library API		
zmod44xx_types.h	ZMOD4410 software library API header		
zmod44xx.c	ZMOD4410 software library API		
zmod44xx.h	ZMOD4410 software library API header		
¥lib <DIR>	Folder for ZMOD4410 software library		
eco2.h	ZMOD4410 software library header		
iaq.h	ZMOD4410 software library header		
odor.h	ZMOD4410 software library header		
r_cda.h	ZMOD4410 software library header		
tvoc.h	ZMOD4410 software library header		
zmod4410_calibration.h	ZMOD4410 software library header		
libeco2_rl78.a	ZMOD4410 software library		
libiaq_rl78.a	ZMOD4410 software library		
libodor_rl78.a	ZMOD4410 software library		
libr_cda_rl78.a	ZMOD4410 software library		
libtvoc_rl78.a	ZMOD4410 software library		
libzmod4410_calibration_co_rl78.a	ZMOD4410 software library		
¥src <DIR>	Folder for source file		
r_cg_cgc_user.c	CGC module	✓	✓
r_cg_cgc.c	CGC module	✓	✓
r_cg_cgc.h	CGC module header	✓	✓
r_cg_hardware_setup.c	System initialization module	✓	✓
r_cg_iica_user.c	IICA module	✓	✓
r_cg_iica.c	IICA module	✓	✓ (Partly)
r_cg_iica.h	IICA module header	✓	✓
r_cg_interrupt_handlers.h	Interrupt handler header	✓	✓
r_cg_macrodriver.h	Header	✓	✓
r_cg_main.c	Main initialization module	✓	✓
r_cg_port_user.c	Port module	✓	✓
r_cg_port.c	Port module	✓	✓
r_cg_port.h	Port module header	✓	✓
r_cg_reset_program.asm	Reset program	✓	✓
r_cg_userdefine.h	User definition file	✓	✓
r_cg_vector_table.c	Vector table	✓	✓
r_lcd.c	LCD character display controlling module	✓	
r_lcd.h	LCD character display controlling module header	✓	
r_rl78_i2c.c	IICA interface	✓	
r_rl78_i2c.h	IICA interface header	✓	

Note: <DIR> means directory.

4.2 Changes on continuous.c

In this application note, some of source code is changed as follows for IDT’s software library API.

- All printf() functions in "continuous.c" are commented out

- Call the function “R_DRAW_IAQ()” defined in "r_cg_main.c", that draws the IAQ value to the LCD display and toggles LED.

The change is as follow.

	Before changes	**After changes**

cont_run()	`/* calculate IAQ index */`
	`iaq = calc_iaq(r_mox, r_cda, &tvoc_par);`
		`/* calculate IAQ index */`
		`iaq = calc_iaq(r_mox, r_cda, &tvoc_par);`
		`R_DrawIAQ(iaq);`

4.3 Changes in r_cg_iica.c

In this application note, the colored parts are commented out as follows for the code generated by the code generation tool.

When the code is generated by the code generation tool again, please comment out the colored part as follows.

	Before changes	After changes

R_IICA0_Master_Receive	`MD_STATUS R_IICA0_Master_Receive(uint8_t adr, uint8_t * const rx_buf, uint16_t rx_num, uint8_t wait)`
	`{`
	`  MD_STATUS status = MD_OK;`
	`  IICAMK0 = 1U;  /* disable INTIICA0 interrupt */`
	`  if (1U == IICBSY0)`
	`  {`
	`  /* Check bus busy */`
	`  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`
	`  status = MD_ERROR1;`
	`  }`
	`  else`
	`  {`
	`  STT0 = 1U;  /* generate a start condition */`
	`  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`
		`MD_STATUS R_IICA0_Master_Receive(uint8_t adr, uint8_t * const rx_buf, uint16_t rx_num, uint8_t wait)`
		`{`
		`  MD_STATUS status = MD_OK;`
		`  IICAMK0 = 1U;  /* disable INTIICA0 interrupt */`
		`  //if (1U == IICBSY0)`
		`  //{`
		`  //  /* Check bus busy */`
		`  //  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`
		`  //  status = MD_ERROR1;`
		`  //}`
		`  //else`
		`  {`
		`  STT0 = 1U;  /* generate a start condition */`
		`  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`

R_IICA0_Master_Send	`MD_STATUS R_IICA0_Master_Send(uint8_t adr, uint8_t * const tx_buf, uint16_t tx_num, uint8_t wait)`
	`{`
	`  MD_STATUS status = MD_OK;`
	`  IICAMK0 = 1U;  /* disable INTIICA0 interrupt */`
	`  if (1U == IICBSY0)`
	`  {`
	`  /* Check bus busy */`
	`  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`
	`  status = MD_ERROR1;`
	`  }`
	`  else`
	`  {`
	`  STT0 = 1U;  /* generate a start condition */`
	`  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`
		`MD_STATUS R_IICA0_Master_Send(uint8_t adr, uint8_t * const tx_buf, uint16_t tx_num, uint8_t wait)`
		`{`
		`  MD_STATUS status = MD_OK;`
		`  IICAMK0 = 1U;  /* disable INTIICA0 interrupt */`
		`  //if (1U == IICBSY0)`
		`  //{`
		`  //  /* Check bus busy */`
		`  //  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`
		`  //  status = MD_ERROR1;`
		`  //}`
		`  //else`
		`  {`
		`  STT0 = 1U;  /* generate a start condition */`
		`  IICAMK0 = 0U;  /* enable INTIICA0 interrupt */`

4.4 Overview of the sample program

In this application note, after initial settings, measurement of IAQ rating will be started in ZMOD4410.

The measurement result is displayed on the LCD display and the LED is turned on according to the IAQ rating.

(1) Initialize the IICA and LCD character display.

(2) Read device information from ZMOD4410.

(3) Calculate the parameters for ZMOD4410 calibration.

(4) Make Initial settings and measurement settings for ZMOD4410.

(5) Start ZMOD4410 measurement.

(6) Wait for ZMOD4410 initialization to complete.

(7) Wait for ZMOD4410 to complete measurement.

(8) Read the measurement result from ZMOD4410.

(9) The measurement results from the first time after initialization to the 10th measurement are discarded (calibration process).

(10) Calculate the IAQ rating from the measurement result.

(11) The IAQ rating is displayed on the LCD display, and the LED is turned on according to the IAQ rating.

(12) Wait until ZMOD4410 enters standby mode.

(13) Repeat steps to (7) to (12)

4.5 Option Byte Settings

Table 4.2 list the option byte settings.

Table 4.2 Option Byte Settings

Address	Setting Value	Contents
000C0H/010C0H	11101111B	Operation of Watchdog timer is stopped (counting is stopped after reset)
000C1H/010C1H	01111111B	LVD operation (VLVD): reset mode TYP. 2.75V Rising edge = 2.76V to 2.87V Falling edge = 2.70V to 2.81V
000C2H/010C2H	11101000B	HS mode High-speed on-chip oscillator clock: 24MHz
000C3H/010C3H	10000100B	On-chip debugging is enabled

4.6 Constants

Table 4.3 lists the constants that are used in this sample program

Table 4.3 Constants

Constant Name	Setting Value	Contents	Defined in
LCD_SLAVE_ADDR	A0H	I2C slave address of the LCD character display	r_ccg_userdefine.h
FCLK_MHZ	24	CPU clock frequency [MHz]	lcd.h
LOOPNUM_1MS	FCLK_MHZ * 143	Loop number for 1ms waiting function	lcd.h
CONTINUOUS_MODE	—	—	Project setting

4.7 Functions

Table 4.4 lists the function used in this sample program.

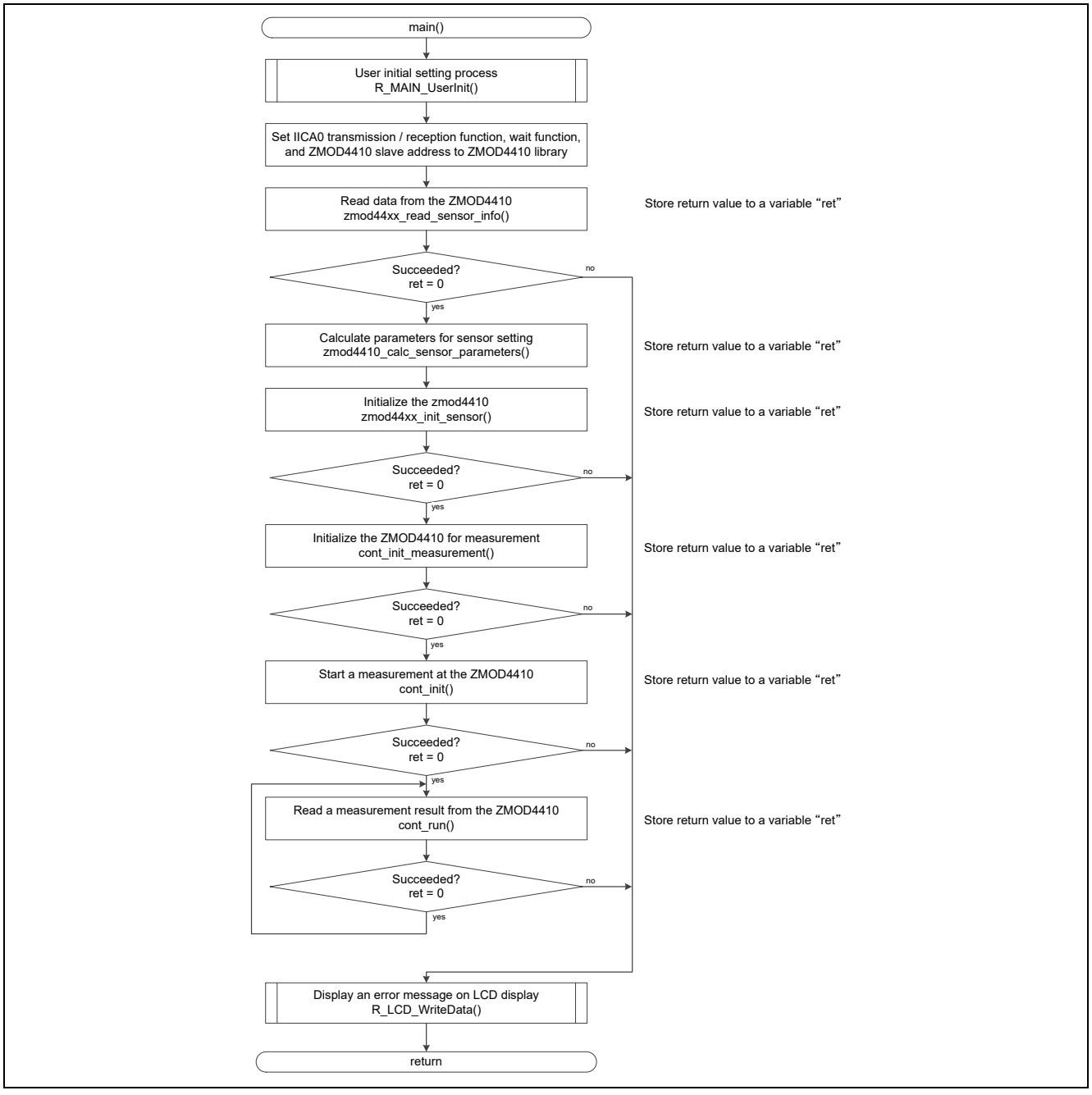
Table 4.4 Functions

Function Name	Outline
main	Main process (generated by Code Generation Tool)
R_MAIN_UserInit	User initialize process
R_RL78_I2C_Read	Read data from ZMOD4410
R_RL78_I2C_Write	Write data to ZMOD4410
R_LCD_Init	Initialize process for LCD character display
R_LCD_WriteCommand	Send command to LCD character display
R_LCD_WriteData	Send draw pattern to LCD character display
R_WaitMilliSeconds	Wait several milliseconds specified by argument
R_Wait1MilliSecond	Wait for 1 millisecond

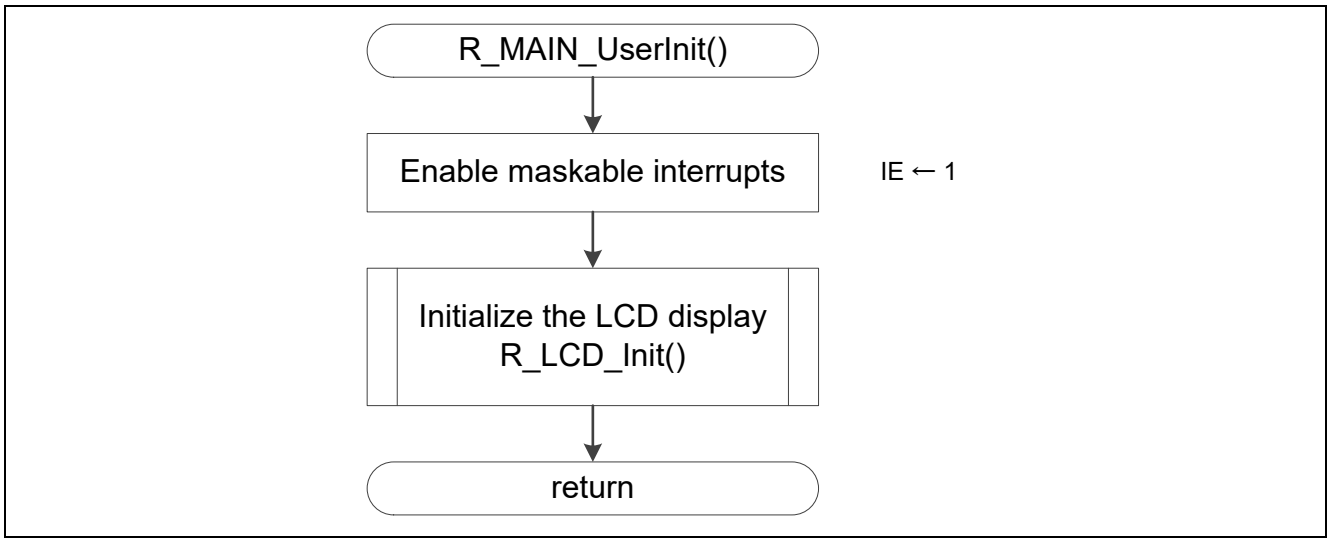
4.8 Function Specifications

This part describes function specifications of the sample code.

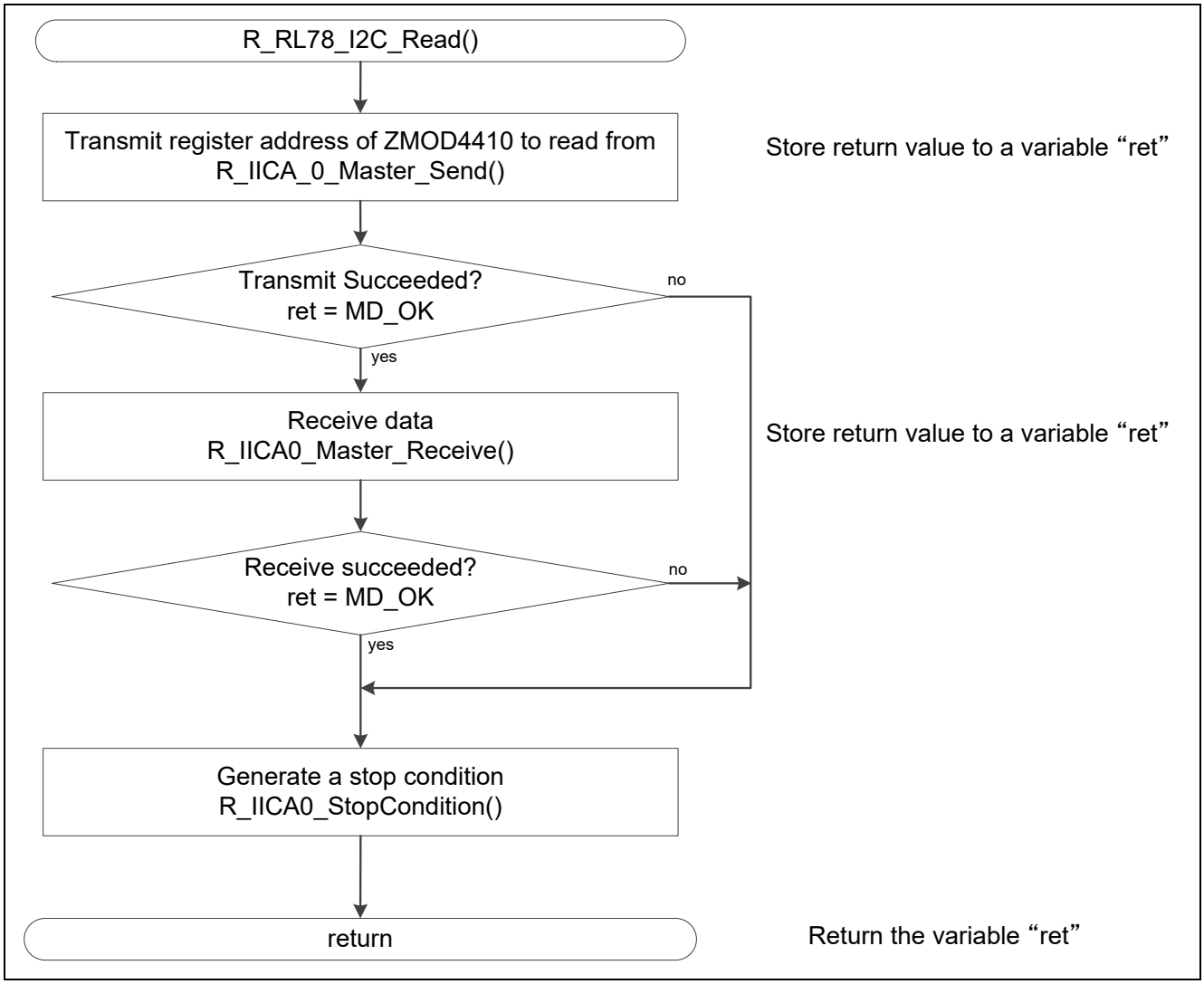
[Function Name]	main
Outline	Main processing
Header	—
Declaration	—
Description	After calling R_MAIN_UserInit(), start to control ZMOD4410 for measuring IAQ rating.
Arguments	None
Return value	None
Remarks	None



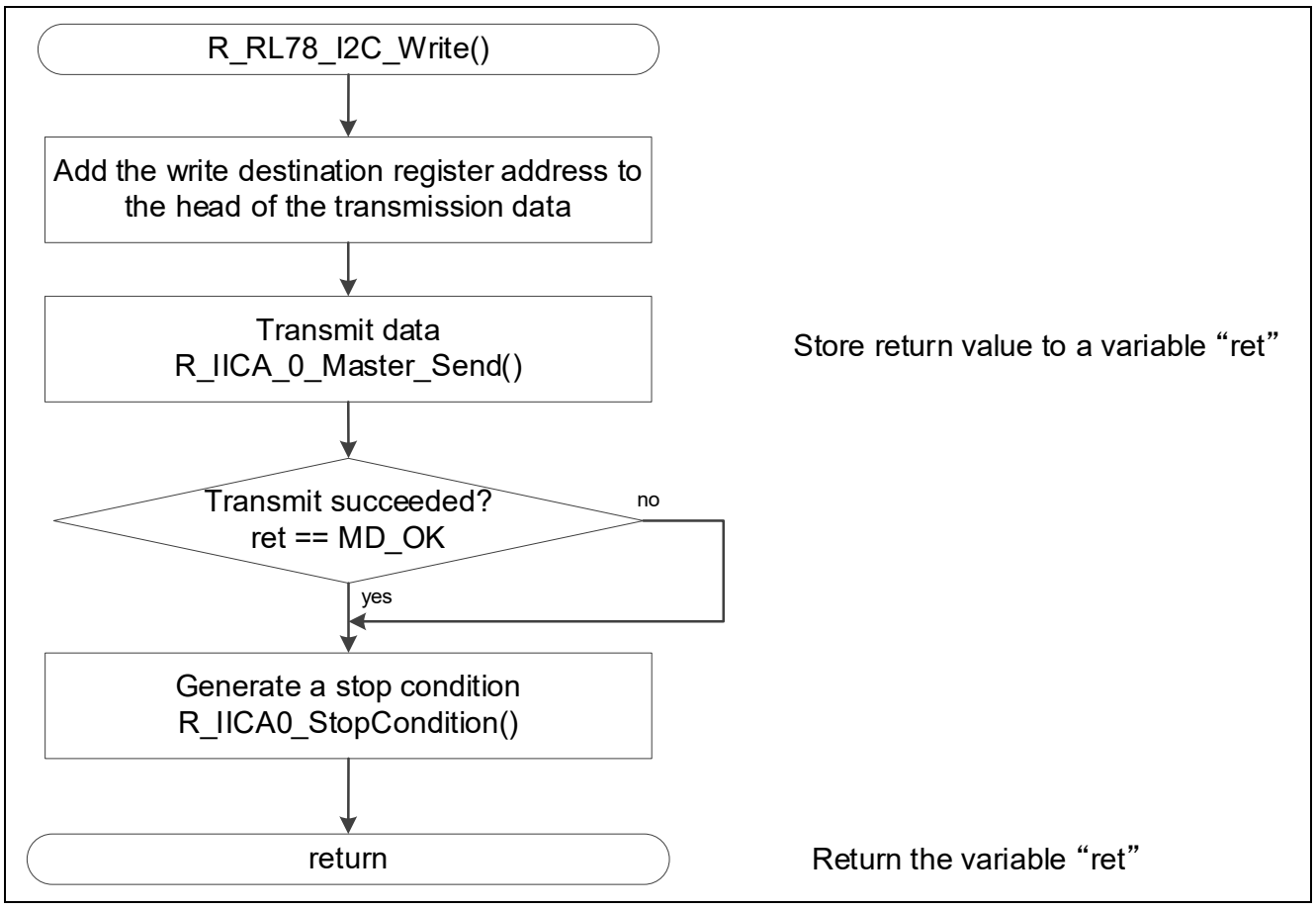
[Function Name]	R_MAIN_UserInit
Outline	User Initialization processing
Header	—
Declaration	static void R_MAIN_UserInit(void);
Description	After enabling maskable interrupts by EI(), initialize the LCD module.
Arguments	None
Return value	None
Remarks	None



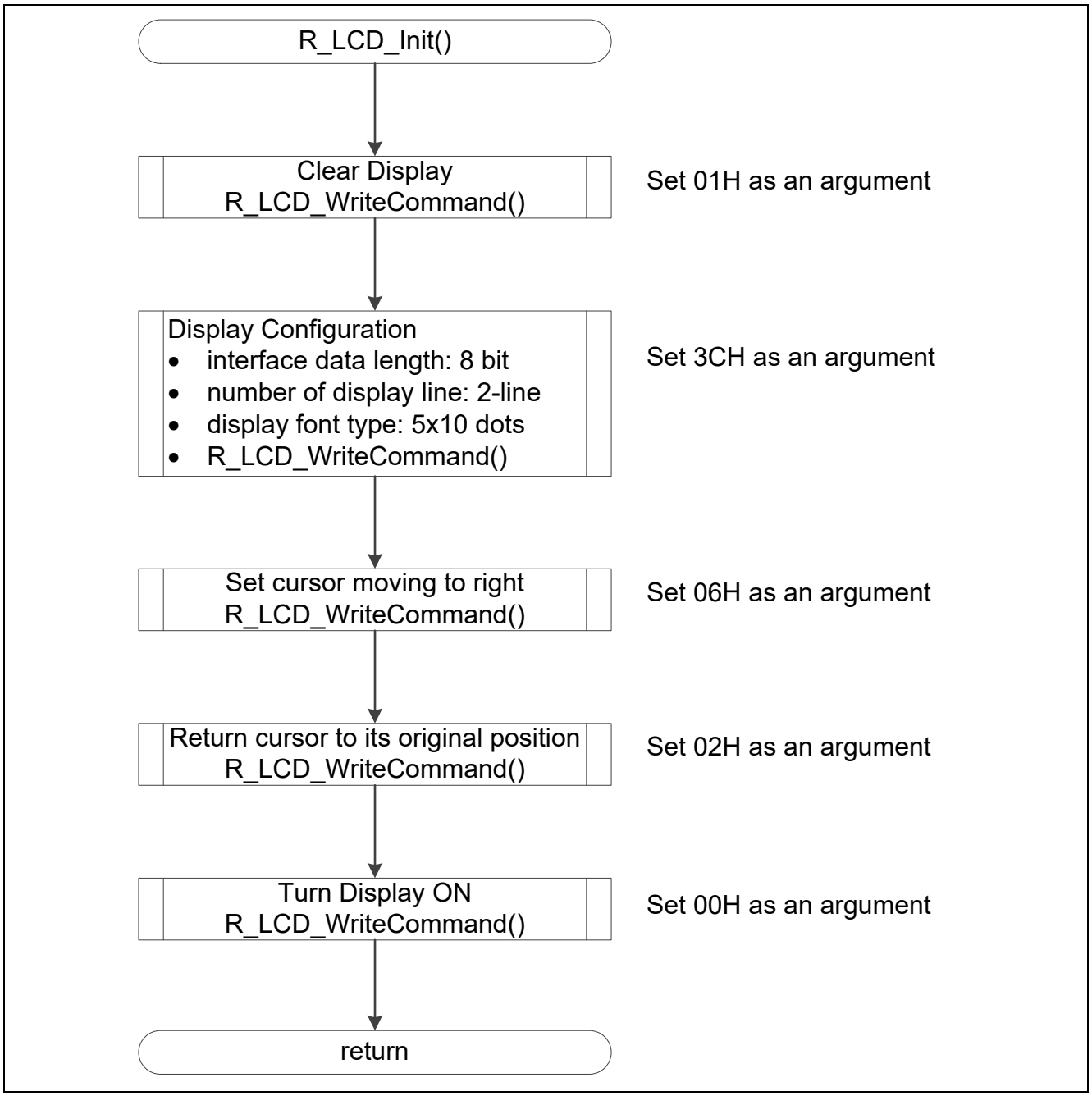
[Function Name]	R_RL78_I2C_Read	
Outline	Read data from ZMOD4410	
Header	—	
Declaration	int8_t R_RL78_I2C_Read (uint8_t i2c_addr, uint8_t reg_addr, const uint8_t *buf, uint8_t len);	
Description	Reads data from the ZMOD4410. Please set this function as a function reading I2C data for the ZMOD4410 library.	
Arguments	i2c_addr	Slave address of ZMOD4410
	reg_addr	Register address of ZMOD4410 to read data from
	buf	Buffer to store data
	len	Length of data to be received [Bytes]
Return value	0: read succeeded, else: read failed	
Remarks	“R_IICA_0_Master_Send()” and “R_IICA_0_Master_Receive()” returns “MD_OK” when the process is succeeded. (defined in r_cg_macrodriver.h)	



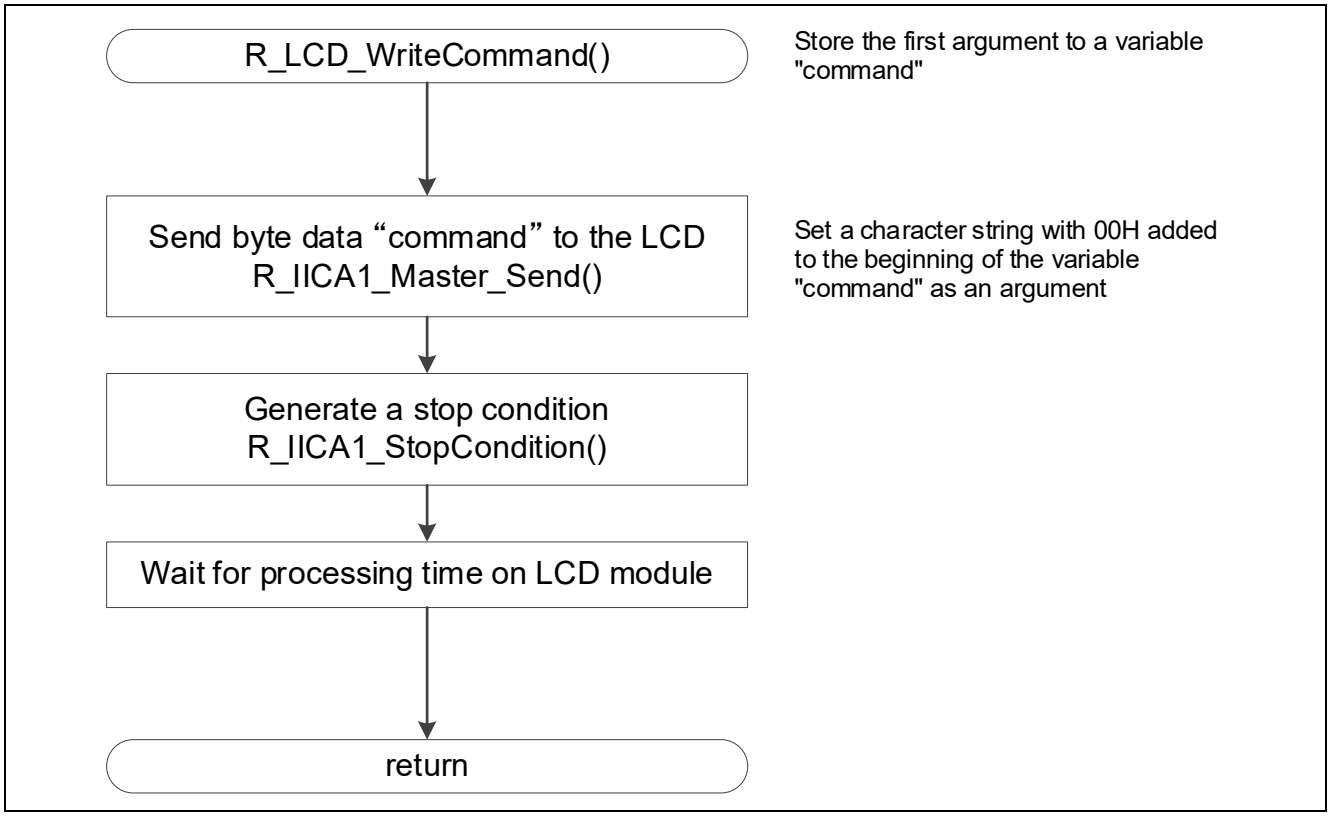
[Function Name]	R_RL78_I2C_Write	
Outline	Write data to ZMOD4410	
Header	—	
Declaration	int8_t R_RL78_I2C_Write(uint8_t i2c_addr, uint8_t reg_addr, const uint8_t *buf, uint8_t len);	
Description	Writes data to the ZMOD4410. Please set this function as a function writing I2C data for the ZMOD4410 library.	
Arguments	i2c_addr	Slave address of ZMOD4410
	reg_addr	Register address of ZMOD4410 to write data to
	buf	Buffer to store write data
	len	Length of data to be sent [Bytes]
Return value	0: write succeeded, else: write failed	
Remarks	“R_IICA_0_Master_Send()” and “R_IICA_0_Master_Receive()” returns “MD_OK” when the process is succeeded. (defined in r_cg_macrodriver.h)	



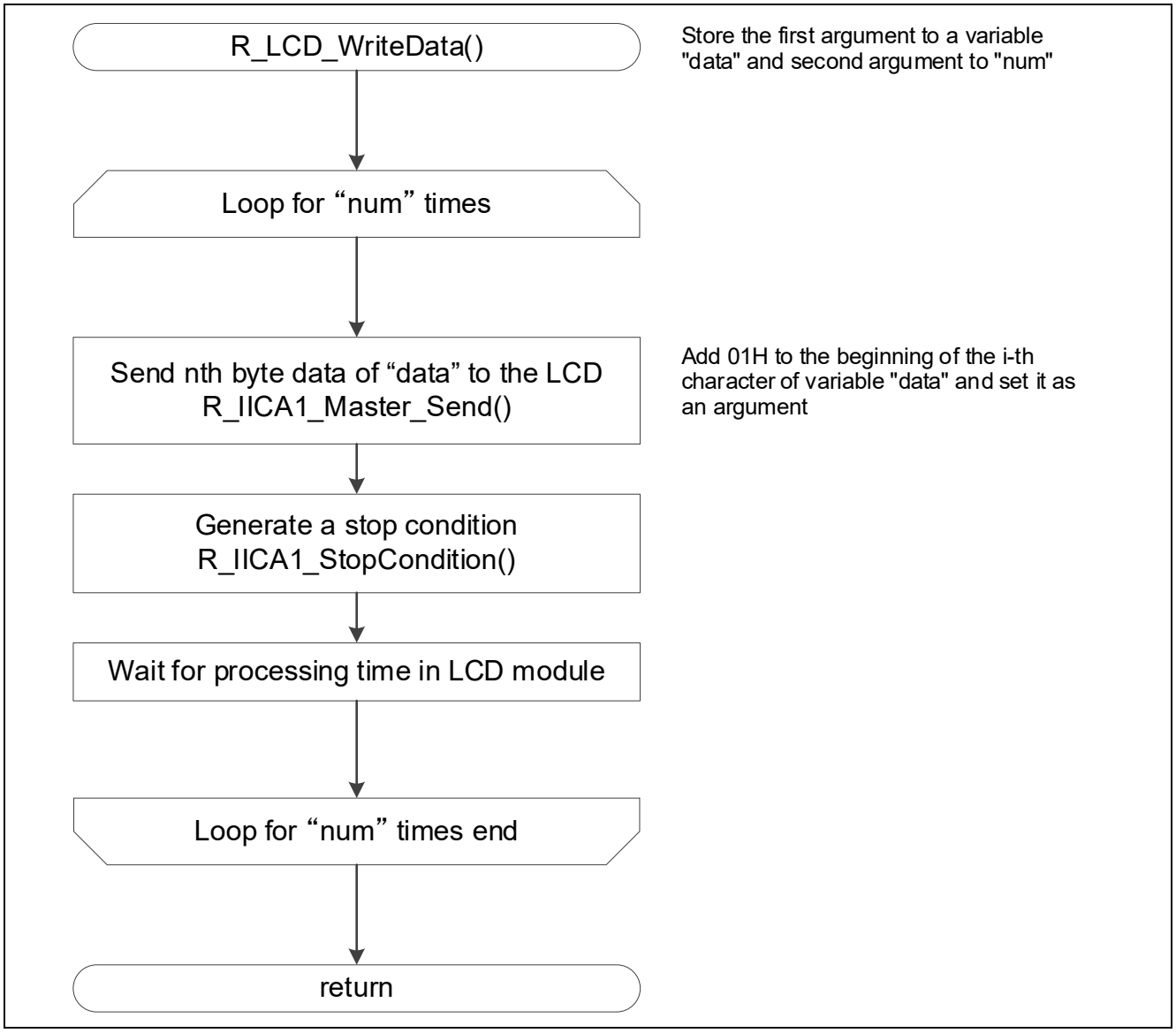
[Function Name]	R_LCD_Init
Outline	LCD character display initialization function
Header	lcd.h
Declaration	void R_LCD_Init(void);
Description	Initialize LCD character display
Arguments	None
Return value	None
Remarks	None



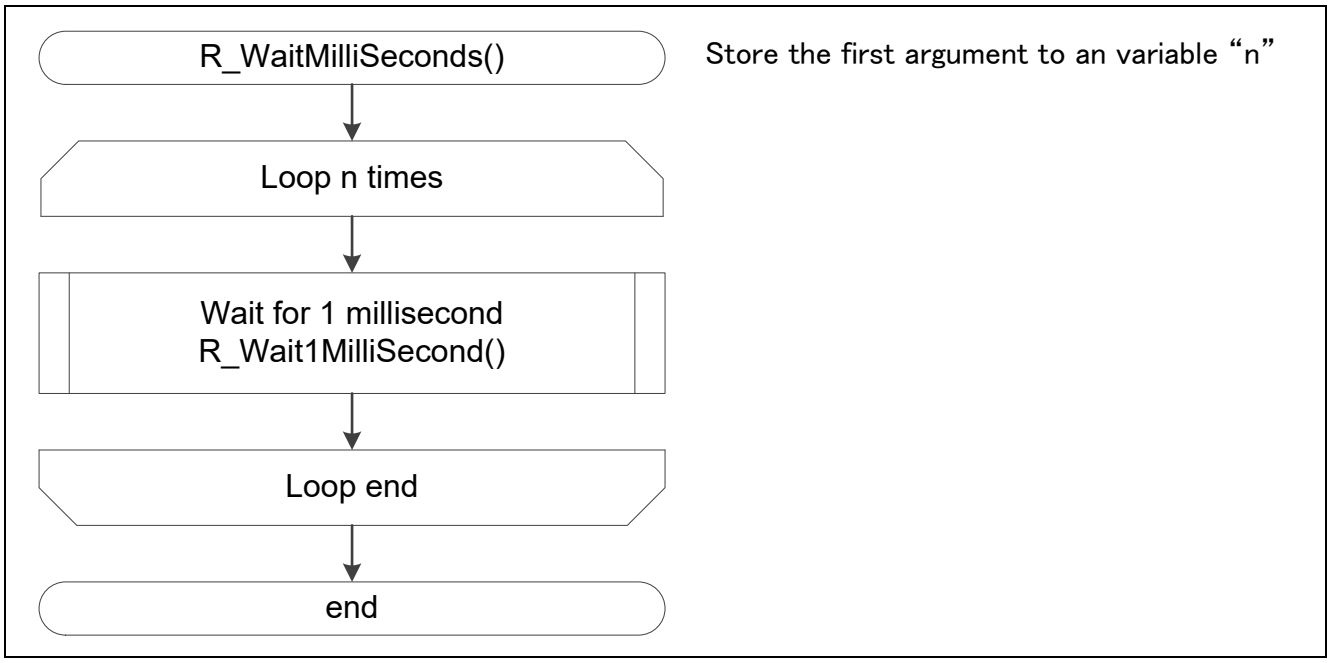
[Function Name]	R_LCD_WriteCommand	
Outline	Control command send function to LCD character display	
Header	lcd.h	
Declaration	void R_LCD_WriteCommand(uint8_t command)	
Description	Send a control command to the LCD character display	
Arguments	command	Control command
Return value	None	
Remarks	None	



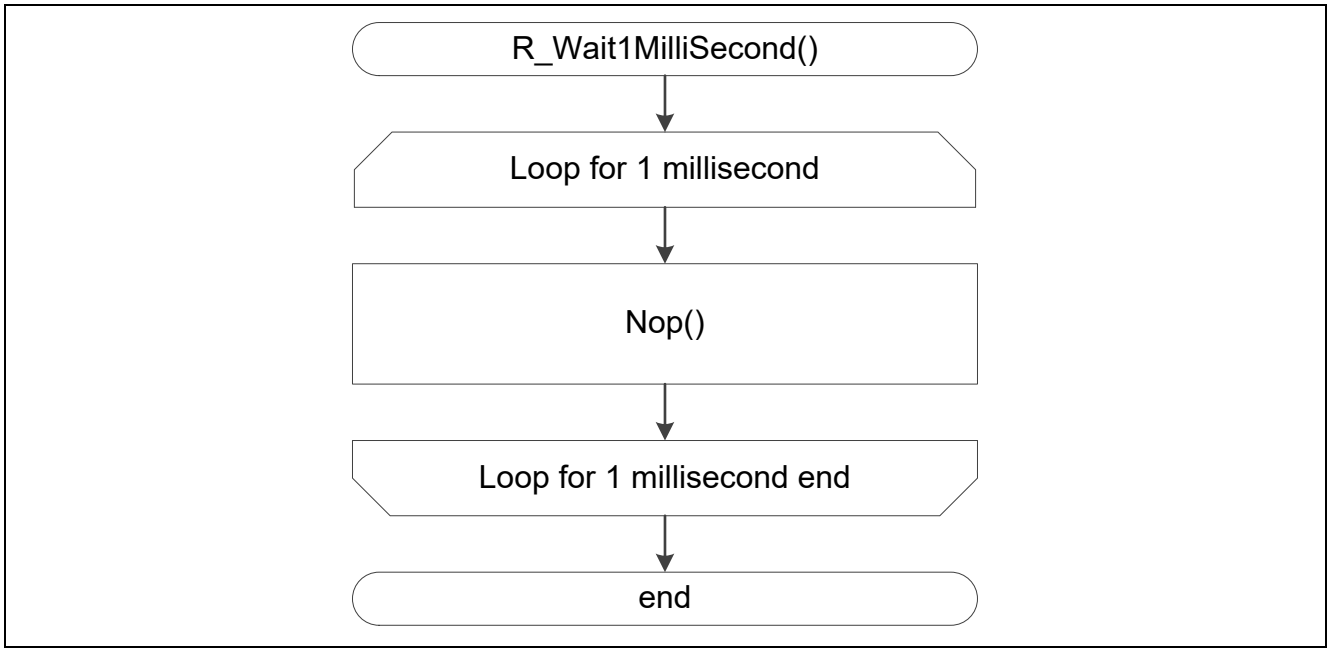
[Function Name]	R_LCD_WriteData	
Outline	Send draw pattern to LCD character display	
Header	lcd.h	
Declaration	void R_LCD_WriteData(uint8_t* data, uint8_t data_num)	
Description	Send display data to the LCD character display	
Arguments	data	Address of data buffer
Return value	data_num	Size of data [Byte]
Remarks	None	
Outline	None	



[Function Name]	R_WaitMilliSeconds
Outline	Wait several milliseconds specified by argument
Header	r_lcd.h
Declaration	void R_WaitMilliSeconds(uint16_t msec);
Description	• Calls the R_Wait1MilliSecond function for the number of milliseconds specified by the argument. • Please set this function as a wait function for the ZMOD4410 library.
Arguments	uint16_t msec milliseconds to wait for
Return value	None
Remarks	None



[Function Name]	R_Wait1MilliSecond
Outline	Wait for 1 millisecond
Header	r_lcd.h
Declaration	void R_Wait1MilliSecond(void);
Description	Wait for 1 millisecond using CPU
Arguments	None
Return value	None
Remarks	None



5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

The ZMOD4410 library can be downloaded from the following IDT website.
https://ja.idt.com/products/sensor-products/gas-sensors/zmod4410-indoor-air-quality-sensor-platform

Website and Support

Renesas Electronics Website
http://www.renesas.com/

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct. 03, 2019	-	First edition

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 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.

(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.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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

Contact information

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