RL78/G13 Group

Sensirion environmental sensor module control sample software

Introduction

This application note explains the communication control sample software between "Renesas Starter Kit for RL78/G13 (RSK)" manufactured by Renesas Electronics Corporation and environmental sensor modules (SVM30, SCD30, and SPS30) manufactured by Sensirion.

Target Device

RL78/G13 Group

Contents

1. Outline 3
2. Operation Confirmation Environment 3
2.1 Connection Preparation 3
2.2 Connection 3
2.3 Connection Example 4
3. Software Functions 5
3.1 Sensor Data 5
3.1.1 SGP30 5
3.1.2 SHTC1 5
3.2 LCD Display 6
3.3 Data Flash Memory 6
4. Code Generator Module 7
4.1 Module Lists 7
5. Software Specifications 10
5.1 Software Structure 10
5.2 File Structure 10
5.3 API Function Specifications 11
5.3.1 R_FLASH_Init 11
5.3.2 R_FLASH_Save 11
5.3.3 R_FLASH_Load 11
5.3.4 R_SHTC1_Init 12
5.3.5 R_SHTC1_GetData 12
5.3.6 R_SGP30_Init 12
5.3.7 R_SGP30_GetData 13

5.3.8 R_SGP30_SetHumi 13
5.3.9 R_SGP30_GetBaseline 13
5.3.10 R_SGP30_SetBaseline 13
5.3.11 R_SCD30_Init 14
5.3.12 R_SCD30_GetData 14
5.3.13 R_SCD30_GetReady 14
5.3.14 R_SCD30_Recalibration 14
5.3.15 R_SPS30_Init 15
5.3.16 R_SPS30_GetData 15
5.3.17 R_SPS30_GetReady 15

1. Outline

This software is communication control sample software of RSK and SVM30.

The RL78 microcomputer gets the data of the temperature and humidity sensor (SHTC1) and Multi-Pixel gas sensor (SGP30) mounted on the SVM30, CO2 data of SCD30, and PM2.5 data of SPS30 by I²C communication and displays it on the RSK LCD.

2. Operation Confirmation Environment

Table 2.1 lists the confirmed operation conditions for this software.

Table 2.1 Operation Confirmation Environment

Item	Description
Evaluation board	R0K50100LC000BE
MCU used	RL78/G13
Operating frequency	20MHz
Operating voltage	5.0V
Integrated Development Environment	e² Studio v7.4.0
C compiler	CCRL v1.08.00

2.1 Connection Preparation

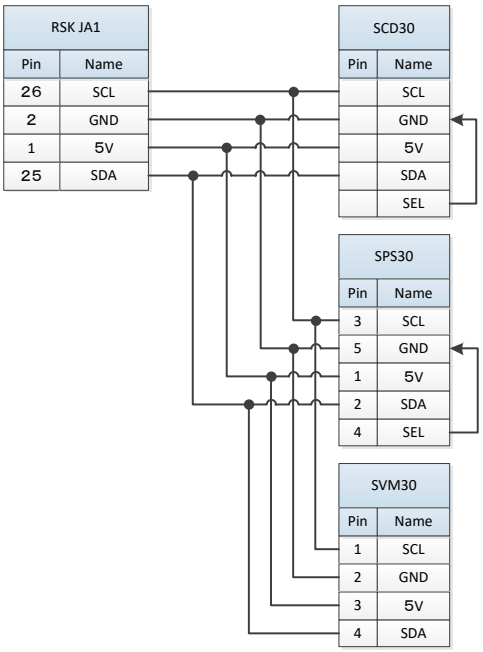
In order to I²C communicate with sensor modules, it is necessary to change the pull-up of RSK to 5V.

Remove the 0[orm] resistor of R96 and add a 0[orm] resistor to R97.

2.2 Connection

Connect sensor modules using JA1 of RSK.

The connection between JA1 of RSK and sensor modules terminals is as follows.



2.3 Connection Example

An example of connection between RSK and sensor modules is shown below.

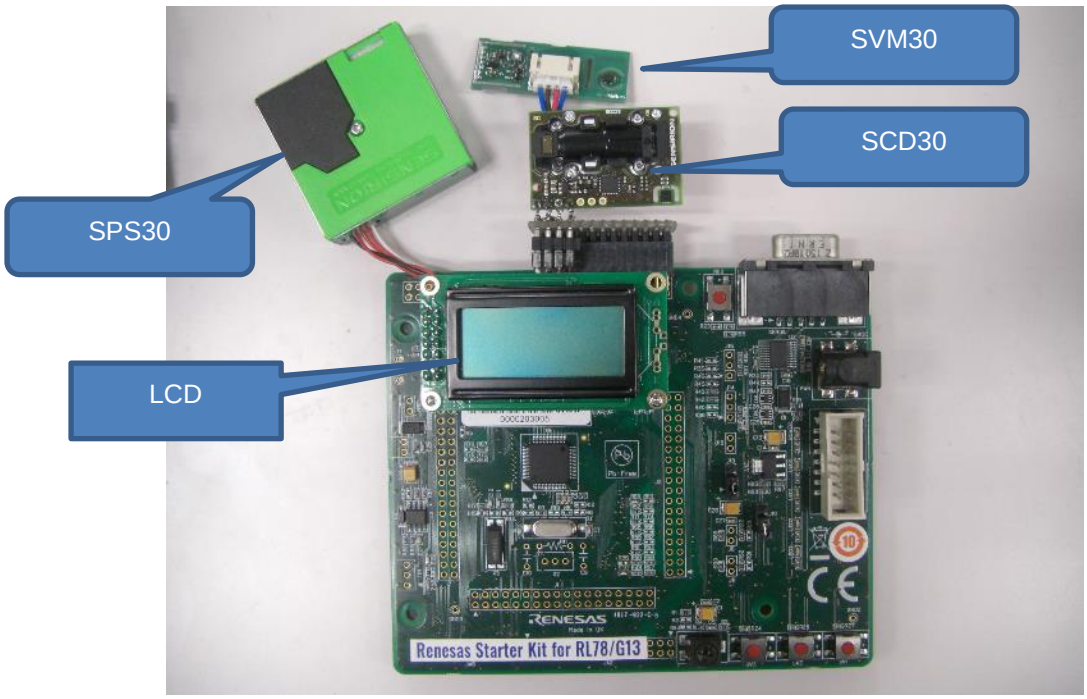


Figure 2.1 Example of connection between RSK and sensor modules

The parts used is shown below.

- PHR-4 (JST : connector housing)
- SPH-002T-P0.5S (JST : crimp tab contact)

3. Software Functions

This software features the following functions.

1. Sensor control (Initialize, get data, set data)
2. Display LCD of sensor data
3. Read/Write data flash memory
4. Calibration of SCD30 with RSK SW1

3.1 Sensor Data

This chapter explains the process of getting data from sensors.

3.1.1 SGP30

SGP30 has humidity correction function. If absolute humidity is set before getting data, the data can be corrected. Calculate the absolute humidity with the following formula.

(Refer to “Driver Integration Guide Software Gas Sensors SGP30”)

$$AH = 216.7 \cdot \frac{\frac{RH}{100.0} \cdot 6.112 \cdot \exp\frac{17.62 \cdot t}{243.12 + t}}{273.15 + t}$$

From SGP30, 6 bytes of data are returned in the following order.

eCO2(High byte), eCO2(Low byte), CRC、TVOC(High byte), TVOC(Low byte), CRC

3.1.2 SHTC1

From SHTC1, 6 bytes of data are returned in the following order.

Temperature(High byte)、Temperature(Low byte)、CRC、Humidity(High byte)、Humidity(Low byte)、CRC

Calculate the getting data with the following formula.

(Refer to “Datasheet Gas Sensor Module SVM30”)

Temperature conversion formula (result in °C)

$$T = -45.68 + 175.7 * \frac{S_T}{2^{16}}$$

Relative humidity conversion formula (result in %RH)

$$RH = \left(103.7 - 3.2 * \frac{S_T}{2^{16}}\right) * \frac{S_{RH}}{2^{16}}$$

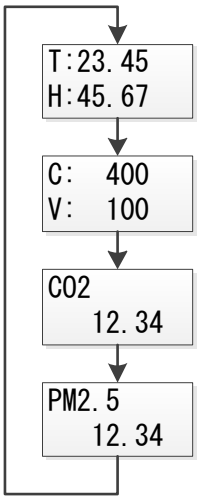
S_{RH} and S_T denote the 16-bit sensor outputs (as decimal values) for relative humidity and temperature, respectively.

3.2 LCD Display

Temperature and humidity, eCO2 and TVOC, CO2, PM2.5 are displayed in order on the LCD.

Display switching interval is one second.

Temperature is T, humidity is H, eCO 2 is C, TVOC is V.



When forced calibration is executed by pressing RSK SW1, the following is displayed for 1 second.

3.3 Data Flash Memory

At startup, if the baseline value is stored in the data flash memory, set the baseline value to SGP30.

After startup, get the baseline value of SGP30 every hour.

This baseline value is overwritten and saved in data flash memory block 0.

4. Code Generator Module

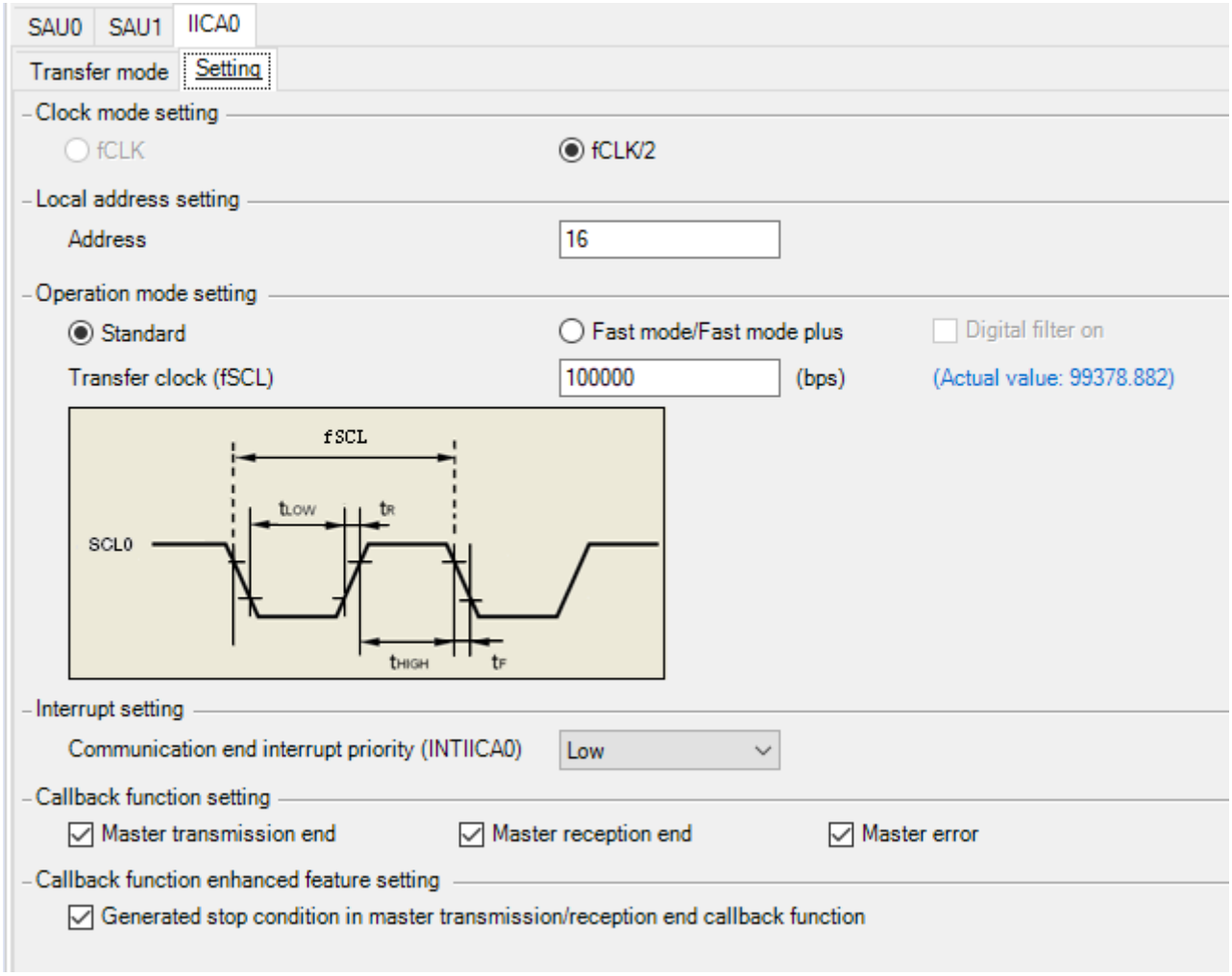
This chapter describes the modules generated by Code Generator.

4.1 Module Lists

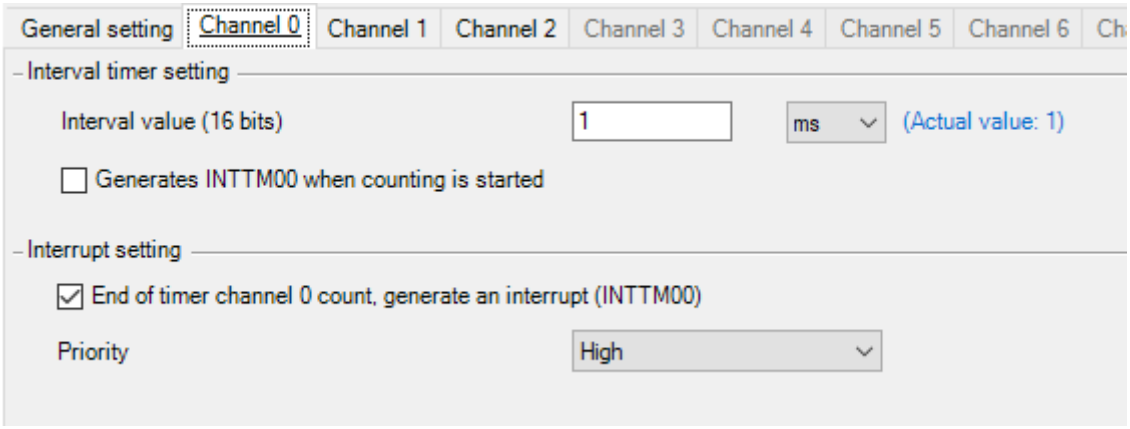
The modules used are as follows.

Folder name	Function	Description
cg_serial	Serial Interface IICA	Sensor Communication
cg_timer	Timer Array Unit Channel 0 (1ms interval timer)	Initialize wait
	Timer Array Unit Channel 1 (10ms interval timer)	One second timer
	Timer Array Unit Channel 2 (1ms interval timer)	Sensor process wait
cg_intc	External interrupt (INTP1)	SW1
pfdl	Data flash memory Library	Store of Baseline

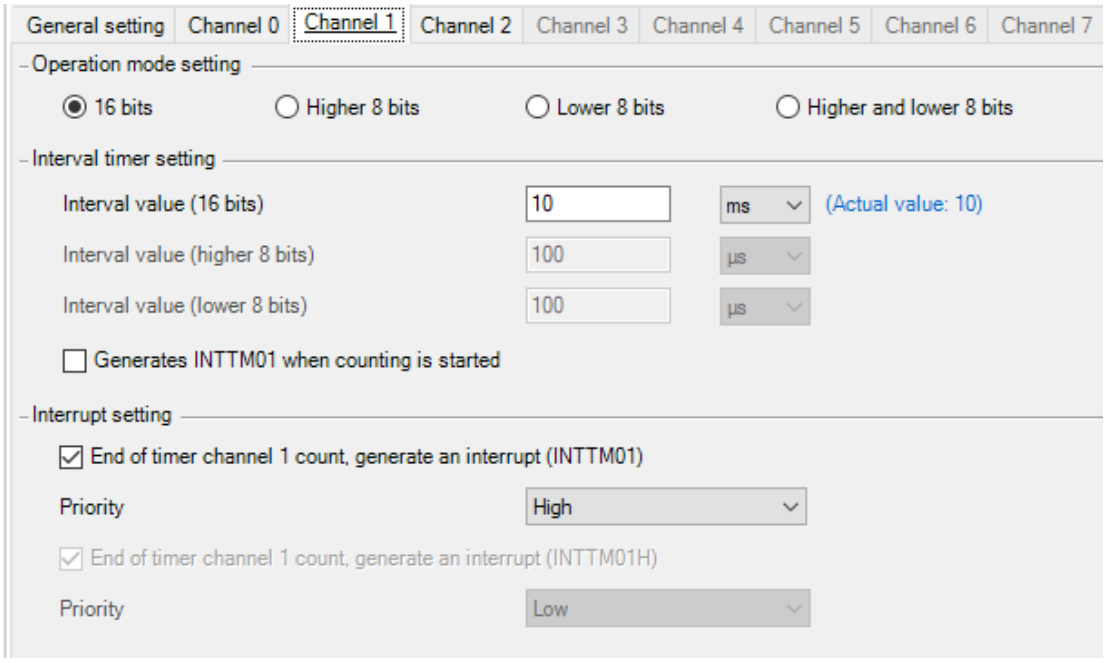
- Serial Interface IICA0 setting



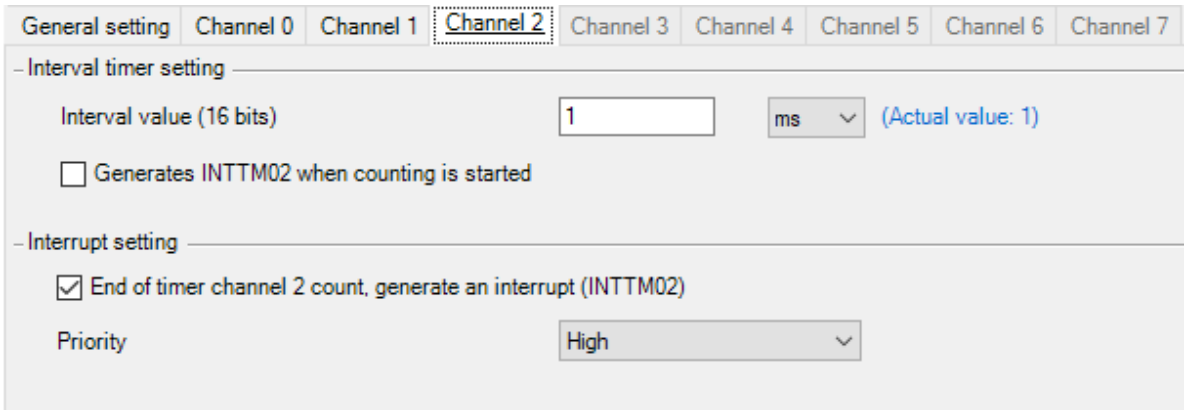
- Timer Array Unit Channel 0 setting



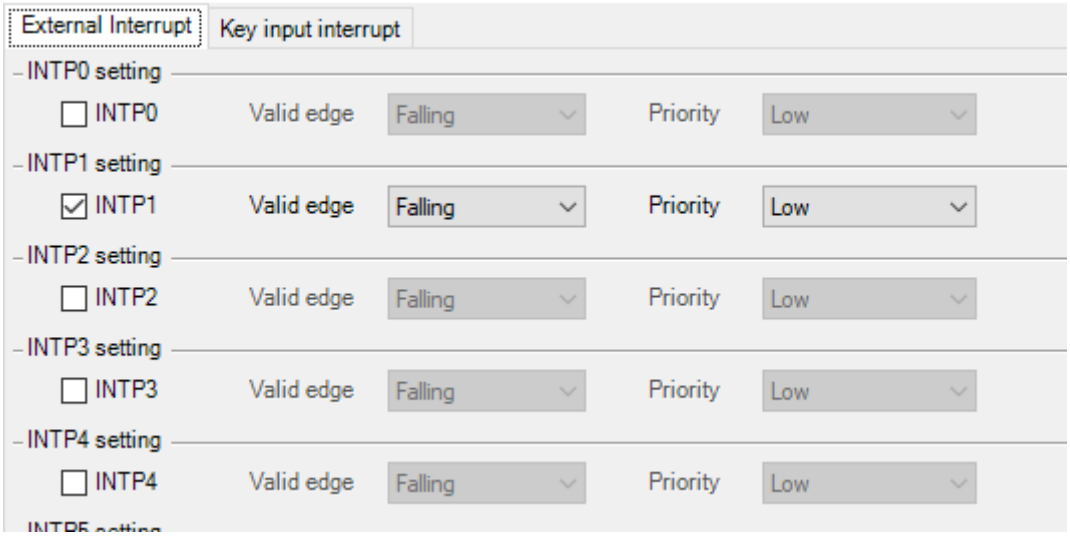
- Timer Array Unit Channel 1 setting



- Timer Array Unit Channel 2 setting



- External Interrupt setting



5. Software Specifications

5.1 Software Structure

Applications are created by using output file from Code Generator.

Figure 5.1 shows a diagram of the software structure.

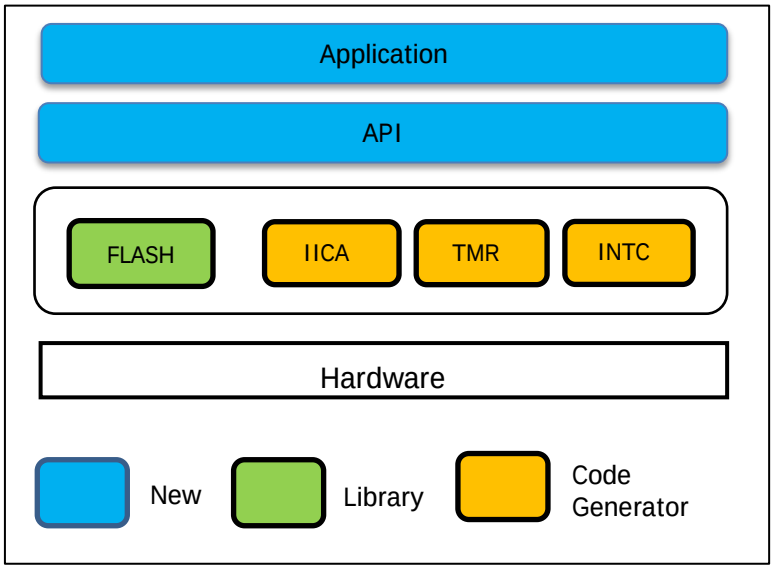


Figure 5.1 Software Structure

5.2 File Structure

Table 5.1 shows the source file tree.

Table 5.1 have omitted the files generated by Code Generator for brevity.

Table 5.1 Source File Tree

```
|-- Workspace
    |-- SensorApl_RL78G13
            |-- src
                |  sensor_main.c ......... Application source file
                |
                |-- Lcd ................... LCD folder
                |
                |-- Sensor
                      r_sgp30.c ........... SGP30 source file
                      r_sgp30.h ........... SGP30 header file
                      r_shtc1.c ........... SHTC1 source file
                      r_shtc1.h ........... SHTC1 header file
                      r_scd30.c ........... SCD30 source file
                      r_scd30.h ........... SCD30 header file
                      r_sps30.c ........... SPS30 source file
                      r_sps30.h ........... SPS30 header file
                     r_i2c_sub.c ......... I²C communication source file
                     r_i2c_sub.h ......... I²C communication header file
```

5.3 API Function Specifications

The following are the specifications for API function.

5.3.1 R_FLASH_Init

Function	Data flash memory library initialization		
Format	MD_STATUS R_FLASH_Init(void)		
Argument	-	-	-
Return value	MD_OK MD_ERROR	Successful completion Abnormal completion	
Description	This API calls R_FDL_Open() to initialize.		
Note	-		

5.3.2 R_FLASH_Save

Function	Write to data flash memory		
Format	MD_STATUS R_FLASH_Save(uint8_t *dat, uint16_t len)		
Argument	uint8_t * uint16_t	dat len	Address of buffer Number of data bytes
Return value	MD_OK MD_ERROR	Successful completion Abnormal completion	
Description	This API calls R_FDL_Erase() to erase the data flash memory, then calls R_FDL_Write() to write to the data flash memory.		
Note	The block of data flash memory can be changed with VD_PRV_FLASH_BLOCK.		

5.3.3 R_FLASH_Load

Function	Read from data flash memory		
Format	MD_STATUS R_FLASH_Load(uint8_t *dat, uint16_t len)		
Argument	uint8_t * uint16_t	dat len	Address of buffer Number of data bytes
Return value	MD_OK MD_ERROR	Successful completion Abnormal completion	
Description	This API stores the data read from the data flash memory at the specified address.		
Note	The block of data flash memory can be changed with VD_PRV_FLASH_BLOCK. Please ensure that the size of the data storage buffer is larger than the number of data bytes.		

5.3.4 R_SHTC1_Init

Function	SHTC1 initialization		
Format	MD_STATUS R_SHTC1_Init(void)		
Argument	-	-	-
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received	
Description	This API sends the GET_SERIAL_ID command. If the serial ID can be read correctly, MD_OK is returned. If the serial ID cannot be read, MD_ERROR is returned.		
Note	-		

5.3.5 R_SHTC1_GetData

Function	SHTC1 data getting		
Format	MD_STATUS R_SHTC1_GetData(float *temp, float *humi)		
Argument	float * float *	temp humi	Temperature data storage address Humidity data storage address
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing	
Description	This API sends the FETCH_DATA command to get the data. Calculate the temperature and humidity from this data by calculation in 3.1.2 and store it at the specified address.		
Note	Repeat until the return value is other than MD_BUSY2. If the return value is MD_OK, sensor data is stored at the address specified by the argument.		

5.3.6 R_SGP30_Init

Function	SGP30 initialization		
Format	MD_STATUS R_SGP30_Init(void)		
Argument	-	-	-
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received	
Description	This API sends the GET_SERIAL_ID command. If the serial ID can be read correctly, MD_OK is returned. If the serial ID cannot be read, MD_ERROR is returned. Send the INIT_AIR_QUALITY command and start measurement.		
Note	-		

5.3.7 R_SGP30_GetData

Function	SGP30 data getting		
Format	MD_STATUS R_SGP30_GetData(uint16_t *eco2, uint16_t *tvoc)		
Argument	uint16_t * uint16_t *	eco2 tvoc	eCO2 data storage address TVOX data storage address
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing	
Description	This API sends the MEASURE_AIR_QUALITY command and stores eCO2 and TVOC in the specified address.		
Note	Repeat until the return value is other than MD_BUSY2. If the return value is MD_OK, sensor data is stored at the address specified by the argument.		

5.3.8 R_SGP30_SetHumi

Function	SGP30 absolute humidity setting		
Format	MD_STATUS R_SGP30_SetHumi(float_t temp, float_t humi)		
Argument	float_t float_t	temp humi	temperature data humidity data
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing	
Description	This API calculates the absolute humidity from the temperature data and humidity data by SHTC1 and sets it to SGP30.		
Note	Repeat until the return value is other than MD_BUSY2. If the return value is MD_OK, setting complete.		

5.3.9 R_SGP30_GetBaseline

Function	SGP30 baseline value getting		
Format	MD_STATUS R_SGP30_GetBaseline(uint8_t *dat)		
Argument	uint16_t *	dat	baseline value storage address
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received	
Description	This API sends GET_BASELINE command to get the baseline value.		
Note	The baseline value storage area shall be secured at least 6 bytes.		

5.3.10 R_SGP30_SetBaseline

Function	SGP30 baseline value setting		
Format	MD_STATUS R_SGP30_SetBaseline(uint8_t *dat)		
Argument	uint16_t *	dat	baseline value storage address
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received	
Description	This API sends SET_BASELINE command to set the baseline value.		
Note	-		

5.3.11 R_SCD30_Init

Function	SCD30 initialization		
Format	MD_STATUS R_SCD30_Init(void)		
Argument	–	–	–
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received	
Description	Send the START_PERIODIC_MEASUREMENT command and start measurement.		
Note	–		

5.3.12 R_SCD30_GetData

Function	SCD30 data getting		
Format	MD_STATUS R_SCD30_GetData(float *co2, float *temp, float *humi)		
Argument	float * float * float *	eco2 temp humi	eCO2 data storage address Temperature data storage address Humidity data storage address
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing	
Description	This API sends the READ_DATA_MEASUREMENT command and stores data in the specified address.		
Note	Repeat until the return value is other than MD_BUSY2. If the return value is MD_OK, sensor data is stored at the address specified by the argument.		

5.3.13 R_SCD30_GetReady

Function	SCD30 status getting		
Format	MD_STATUS R_SCD30_GetReady(void)		
Argument	–	–	–
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2 MD_NOT_READY	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing Preparing	
Description	This API sends the GET_DATA_READY command and gets sensor status. MD_OK is returned when data preparation is complete, and MD_NOT_READY is returned when preparation is in progress.		
Note	Repeat until the return value is other than MD_BUSY2.		

5.3.14 R_SCD30_Recalibration

Function	SCD30 forced recalibration execution		
Format	MD_STATUS R_SCD30_Recalibration(void)		
Argument	–	–	–
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing	
Description	This API sends the SET_FORCED_RECALIB command and execute forced recalibration.		
Note	–		

5.3.15 R_SPS30_Init

Function	SPS30 initialization		
Format	MD_STATUS R_SPS30_Init(void)		
Argument	–	–	–
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received	
Description	Send the START_MEASUREMENT command and start measurement.		
Note	–		

5.3.16 R_SPS30_GetData

Function	SPS30 data getting		
Format	MD_STATUS R_SPS30_GetData(st_sps_t *p_sps_dat)		
Argument	st_sps_t *	p_sps_dat	SPS30 data storage address
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing	
Description	This API sends the READ_MEASURED_VALUES command and stores data in the specified address.		
Note	Repeat until the return value is other than MD_BUSY2. If the return value is MD_OK, sensor data is stored at the address specified by the argument.		

5.3.17 R_SPS30_GetReady

Function	SPS30 status getting		
Format	MD_STATUS R_SPS30_GetReady(void)		
Argument	–	–	–
Return value	MD_OK MD_ERROR MD_ARGERROR MD_ERROR1 MD_ERROR2 MD_ERROR3 MD_BUSY2 MD_NOT_READY	Successful completion Abnormal completion Argument error Arbitration lost Timeout NACK received Processing Preparing	
Description	This API sends the READ_DATA_READY_FLAG command and gets sensor status. MD_OK is returned when data preparation is complete, and MD_NOT_READY is returned when preparation is in progress.		
Note	Repeat until the return value is other than MD_BUSY2.		

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug 31, 2018	-	First release
2.00	Jul 31, 2019	-	Add SCD30 and SPS30

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/.