RL78/G15

HS300x sensor data communication with Bluetooth LE DA14531

Introduction

This application note describes a sample program to control the DA14531MOD * on the RL78/G15 to perform wireless communication. Humidity and temperature data acquired from the HS300x (humidity and temperature sensor) is communicated wirelessly. Humidity and temperature data is sent to the smartphone via Bluetooth® Low Energy (LE). This application note also describes how to control the HS300x on the RL78/G15.

* DA14531MOD Firmware uses Codeless DA14531 Data Pump Hex.

Target Device

RL78/G15

DA14531MOD

HS300x

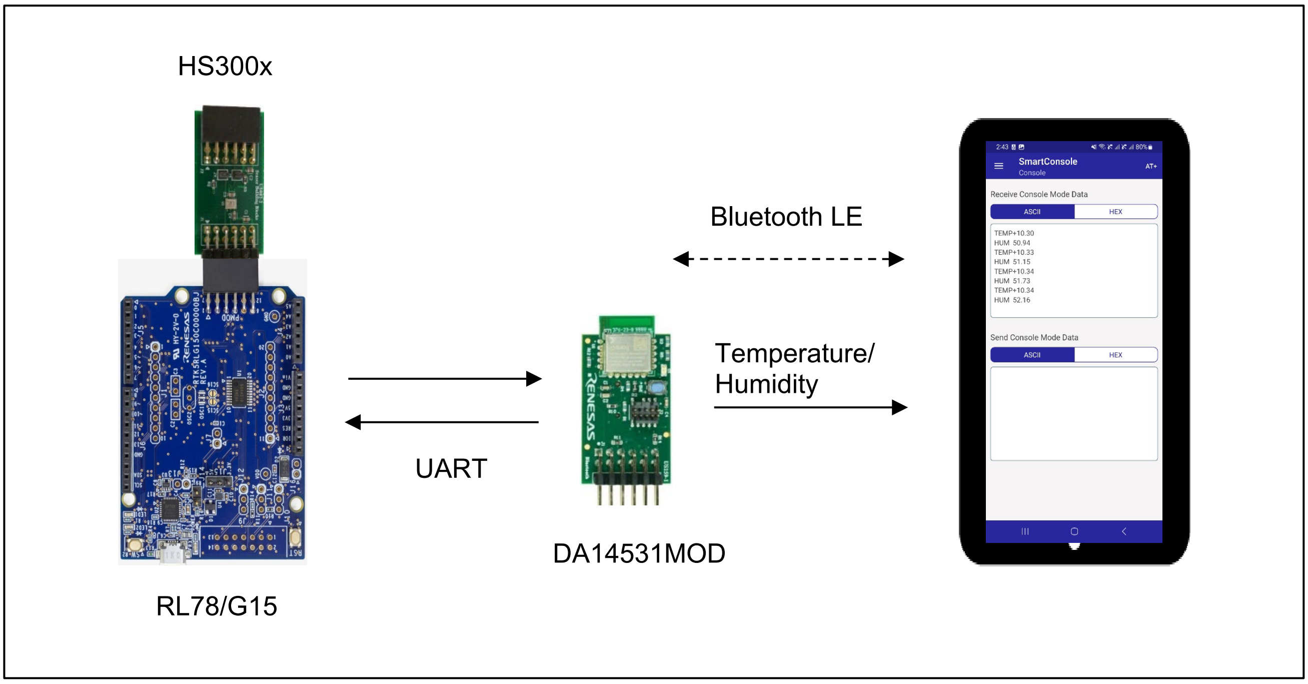
Contents

1. Overview

This application note describes how to connect the RL78/G15 with the DA14531MOD and wirelessly communicate humidity and temperature data acquired from the HS300x. The sample program uses the DA14531MOD by the UART function incorporated into the RL78/G15 to wirelessly communicate humidity and temperature data to your smartphone using Bluetooth® Low Energy (LE).

In addition, this application note describes how to acquire and calculate humidity and temperature data from the HS300x humidity and temperature sensor on the RL78/G15. The sample program controls the HS300x by the built-in I2C driver of the RL78/G15, and acquires humidity and temperature ADC data from the HS300x. Additionally, this sample program calculates acquired data on the RL78/G15.

Figure 1-1 Schematic Diagram of the Entire System



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/G15 （R5F12068）
Operating frequency	16 MHz
Memory size (memory used)	ROM 8KB （6526 byte used） RAM 1KB （287 byte used）
Operating voltage	2.4V～5.5V SPOR operation At rising edge TYP. 2.57 V (2.44 V to 2.68 V) At falling edge TYP. 2.52 V (2.40 V to 2.62 V)
Integrated development environment (e2studio)	e2 studio V2024-01 from Renesas Electronics Corporation
C compiler (e2studio)	CC-RL V1.13.00 from Renesas Electronics Corporation
Integrated development environment (CS+)	CS+ V8.11.00 from Renesas Electronics Corporation
C compiler (CS+)	CC-RL V1.13.00 from Renesas Electronics Corporation
Smart configurator (SC)	V1.9.0 from Renesas Electronics Corp.
Board support package (BSP)	BSP v1.62
Board used	RL78/G15 Fast Prototyping Board (RTK5RLG150C00000BJ)
Temperature/Humidity Sensor Modules	Relative Humidity Sensor Pmod™ Board (US082-HS3001EVZ)
Data transmission module	Low Power Bluetooth® Pmod™ Board (US159-DA14531EVZ)

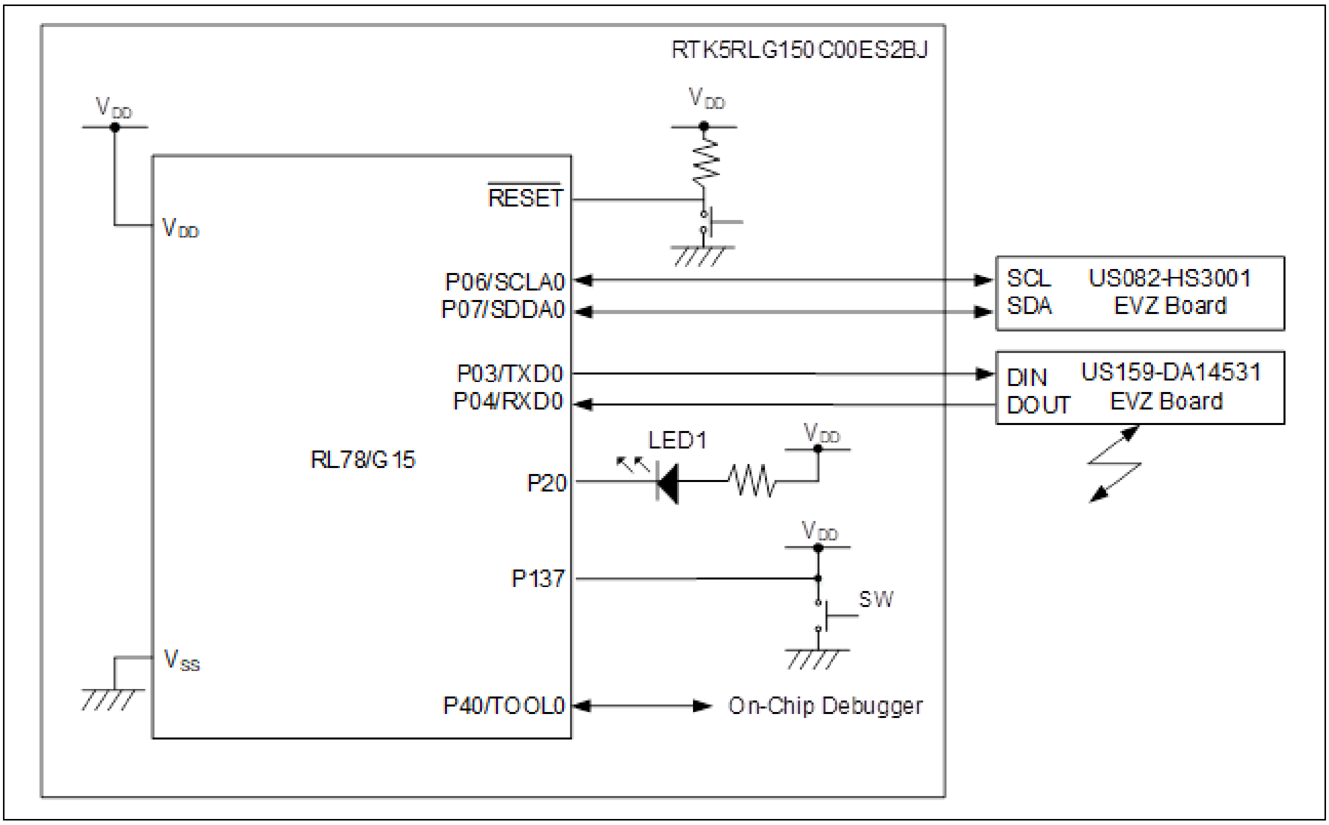
Note. The optimization options used by the C compiler in this application are as follows:
The optimization level of the CC-RL compiler : -Olite
The optimization level of the IAR compiler : [High] , Balance

3. Hardware Descriptions

3.1. Example of Hardware Configuration

The following shows an example of the hardware configuration used in the application note.

Figure 3-1 Hardware Configuration



Note 1. This schematic circuit diagram is simplified to show the outline of connections. When creating circuits, design them so that they meet electrical characteristics by properly performing pin processing. (Connect input-only ports to VDD or VSS individually through a resistor.)

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

3.2. List of Pins to be Used

The following lists the pins to be used and their functions.

Table 3-1 Pins to be Used and Their Functions

Pin name	I/O	Function
P03/TXD0	Output	UART Data transmission pin (Via DA14531MOD)
P04/RXD0	Input	UART Data reception pin (From DA14531MOD)
P06/SCLA0	Input/Output	IICA serial clock (with HS300x)
P07/SDDA0	Input/Output	IICA serial data bus (with HS300x)
P20	Output	LED1 control pin (Low Active)
P137 / INTP0	Input	Input terminal for the switch (external interrupt request input terminal)

4. Module Specifications

This section describes the specifications of the DA14531MOD and HS300x.

4.1. Specifications of DA14531MOD

Table4-1 outlines the specifications of the DA14531MOD.

Table 4-1 Specifications of DA14531MOD

Item	Description
Communication standard	Bluetooth 5.1 core qualified
Receiver sensitivity	-93dBm
Receiver current consumption	2mA at 3V
Transmission current consumption	4mA at 3V at 0dBm
Serial data interface	UART, SPI, I^2C
Configuration method	API or AT command, local or wireless
Frequency band	ISM 2.4 GHz
Operating voltage	1.8 to 3.3 V

4.2.1. UART communication interface

UART communication, which communicates humidity and temperature data, is sent in the Binary Mode of DA1531MOD (Codeless DA14531 Data Pump Hex). For more information on UART communication, refer to DA145xx CodeLess User Manual.

Website: SmartBond™ - CodeLess™ AT Commands | Renesas

DA145xx CodeLess User Manual — DA145XX Tutorial SDK Getting started (renesas.com)

7. CodeLess Host Application — DA145XX Tutorial SDK Getting started (renesas.com)

4.2. Specifications of HS300x

Table 4-2 outlines the specifications of the HS300x humidity and temperature sensor.

Table 4-2 Specifications of HS300x Sensor

Item	Description
Humidity measurement range	0 to 100%RH
Humidity accuracy	±1.5%RH (typ.) (HS3001, 10 to 90%RH, 25℃)
ADC resolution	0.01%RH (typ.), (14 bits)
Measurement resolution	8, 10, 12, 14 bits
RH response speed	1 second (typ.) (with 1 m/s air flow) 4 seconds (typ.) (in a sealed space)
Temperature sensor accuracy	±0.2℃ (typ.) (HS3001, HS3002, -10 to 80℃)
Average current	Average 24.4 uA (14-bit resolution, power supply voltage 3.3 V supplied), 1 RH and temperature measurement per second
Sleep current	-40 to 85℃: 0.6 μA (typ.) -40 to 125℃: 1 μA (typ.)
Power supply voltage	2.3 V to 5.5 V: 3.3 V (typ.)
Extended power supply voltage	1.8 V (-20℃ to +125℃)
Operating temperature	-40℃ to +125℃

Table 4-3 List of Sensor Functions

Item	Description
I^2C communication	Sensor data are transferred through I^2C communication.
Measurement mode	This software operates with the sensor in sleep mode. On completion of measurement triggered by a measurement request, the sensor enters sleep mode.
Measurement request	The sensor in sleep mode is placed in the measurement state upon receiving a measurement request.
Data fetch	At the end of a measurement cycle, valid data can be acquired.
Status bits	The status bits for the results of measurement indicate whether the current data are valid.

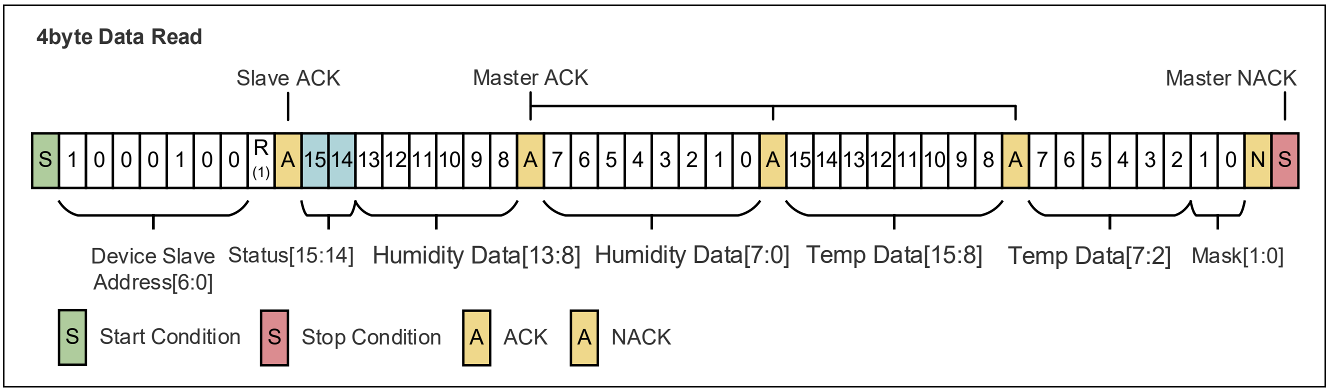
Note: This software does not support the following functions of the HS300x:

- Access to non-volatile memory
- Setting of the measurement resolution
- Reading of the HS300x ID number

4.2.1. I²C Communication Interface

The following figure shows the format of measurement data transferred through I²C communications.

Figure 4-1 Format of I2C Communication



The status bits indicate the state of data as follows.

00B: Valid data acquired at the end of a measurement cycle

01B: Invalid data that have already been acquired

4.2.2. Expressions for Converting Output Value to Humidity and Temperature

The HS300x software converts the acquired ADC data to values for humidity and temperature, and then outputs them.

The humidity conversion expression is as follows.

$$\text{Humidity [\%RH]} = \left(\frac{\text{Humidity}[13:0]}{2^{14}-1}\right) * 100$$

The temperature conversion expression is as follows.

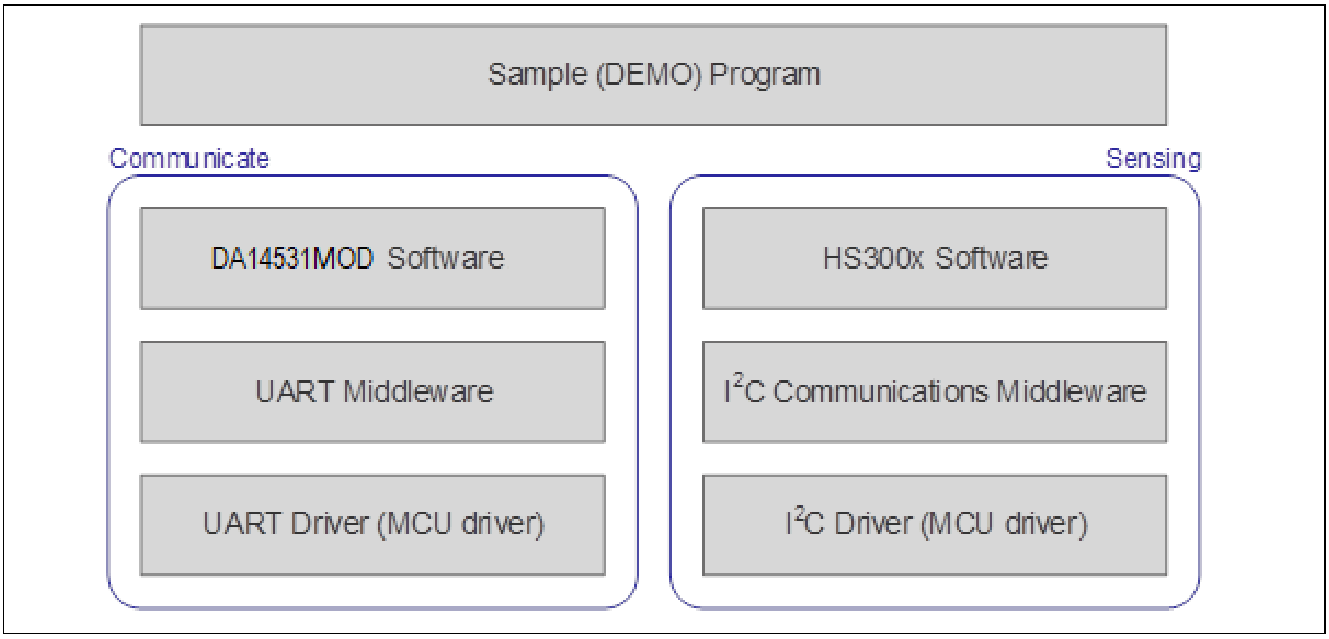
$$\text{Temperature [°C]} = \left(\frac{\text{Temperature}[15:2]}{2^{14}-1}\right) * 165 - 40$$

5. Sample Program

5.1. Sample Program Structure

Figure 5-1 is a block diagram of the sample program structure.

Figure 5-1 Block Diagram of Software



5.2. Specifications of HS300x API Functions

5.2.1. List of HS300x API functions

The following table lists the sensor API functions. For details of the API functions, refer to the separately provided HS300x Sensor API FIT Module application note (R01AN5893) and Renesas Sensor Control Modules Firmware Integration Technology (R01AN5892).

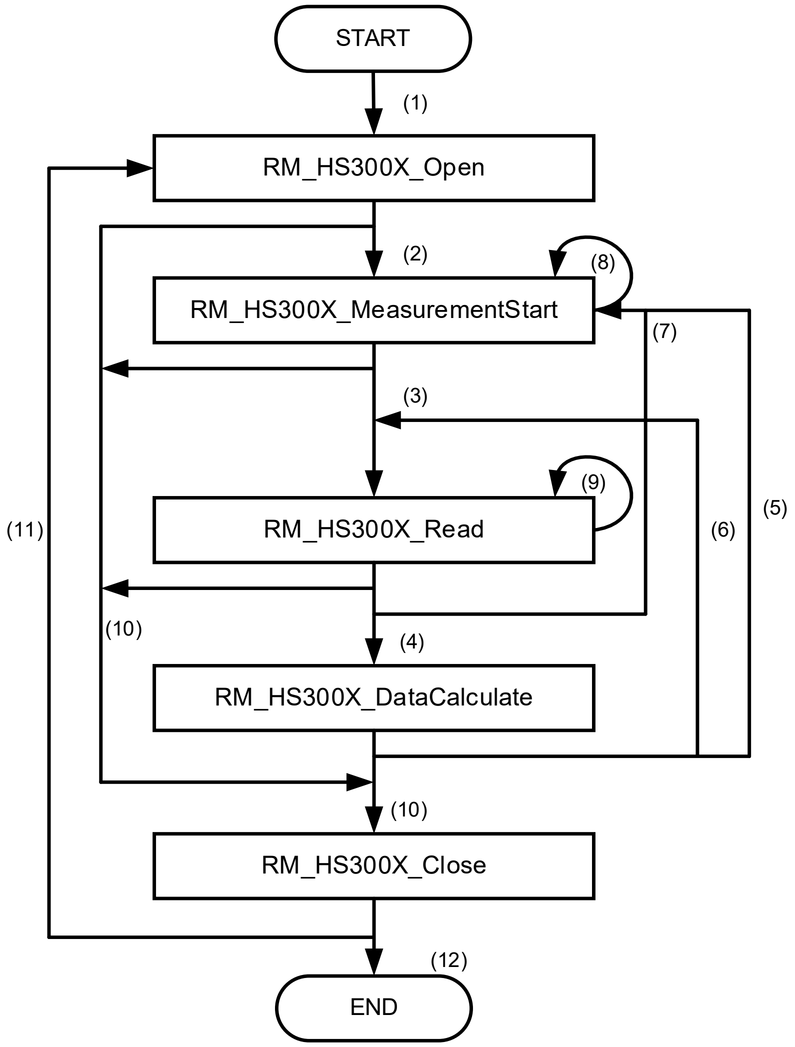
Table 5-1 List of HS300x API Functions

Function	Function
RM_HS300X_Open	Starts control of the sensor.
RM_HS300X_Close	Terminates control of the sensor.
RM_HS300X_MeasurementStart	Starts measurement by the sensor.
RM_HS300X_Read	Acquires data from the sensor.
RM_HS300X_DataCalculate	Calculates values from the data acquired from the sensor.

5.2.2. Guide to Using the API Functions

The following diagram of API function transitions shows the conditions on the usage of the individual HS300x API functions and the expected orders of function calls.

Figure 5-2 Diagram of Transitions between API Function Calls



Flow No.	Function	Transition Case
(1)	Start HS300x	not open HS300x
(2)	Start Measurement HS300x	open success
(3)	ADC Read	Measurement success (callback state success)
(4)	Humidity & Temp. calculate	ADC Read success (callback state success)
(5)	next measurement start	DataCalculate return success and next ADC data Get
(6)	get retry ADC data	User get retry ADC data or DataCalculate return invalid data
(7)	next measurement (Not calculate)	The user requests only ADC data and measures the next data
(8)	Retry Measure Start	Measurement start Waiting for response (callback none and polling)
(9)	Retry ADC Read	Read ADC Waiting for response (callback none and polling)
(10)	Close HS300x	Error state or user stop
(11)	Restart HS300x	Close success and user restart
(12)	Stop HS300x	Close success or Error

The conditions for calling the individual functions are shown below.

・RM_HS300X_Open:	(1) Activation of HS300x, or (11) restart after a call of RM_HS300X_Close
・RM_HS300X_Close:	(10) Successful completion or abnormal end of individual processing
・RM_HS300X_MeasurementStart:	(2) Start of measurement after a call of RM_HS300X_Open, (5) (7) acquisition of the next measured data, or (8) retry after waiting for the response to the measurement start request
・RM_HS300X_Read:	(3) Acquisition of measured data after a call of RM_HS300X_MeasurementStart, or (9) retry after waiting for the response to the data acquisition request
・RM_HS300X_DataCalculate:	(4) Calculation of humidity and temperature data after a call of RM_HS300X_Read

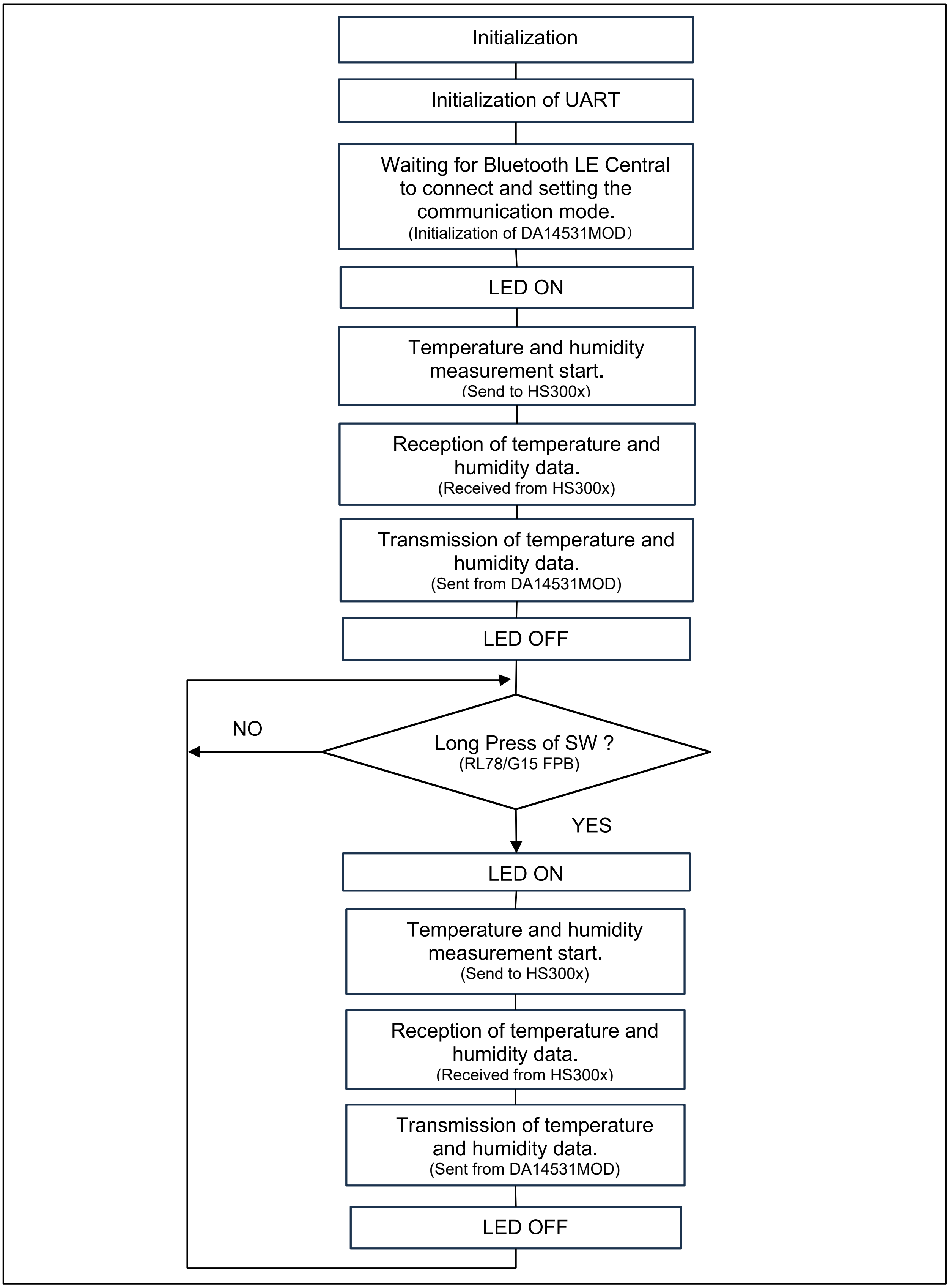
Notes:

1. Since RM_HS300X_Open checks the state of the I^2C driver, the I^2C driver must be opened before the RM_HS300X_Open processing.
2. If you perform RM_HS300X_Read processing, the measurement will be stopped, so if you want to perform RM_HS300X_Read processing again, execute RM_HS300X_MeasurementStart.

5.3. Flowcharts

The following figure shows the overall flowchart.

Figure 5-3 Overall Flowchart



5.4. Sample Program Structure

5.2.1. Peripheral Functions to Be Used

The following table lists the peripheral functions used in the sample program.

Table 5-2 Peripheral Functions to Be Used

Peripheral Function	Use
PORT	Used to control LED1 mounted on RL78/G15_FPB.
IICA0	Used for I^{2}C communication with HS300x and acquisition of humidity and temperature data from HS300x.
UART0	Used for UART communication with DA14531MOD, command transmission to DA14531MOD, and confirmation of response results returned from DA14531MOD.

5.2.2. Settings of Peripheral Functions

The following table lists the settings of Smart Configurator used in the sample program. The items and settings in each table of Smart Configurator Settings are described with the names displayed on the actual setting screen.

Table 5-3 Smart Configurator Settings (1/2)

Category	Item	Settings
Smart Configurator >> Clock		The settings on the [Clock] tab are as follows:
	VDD setting	2.4 V ≦ VDD ≦ 5.5 V
	High-speed on-chip oscillator	Checked Frequency: 16 MHz
	X1 oscillator circuit	Unchecked
	Low-speed on-chip oscillator	15 kHz
	Source selection for main system clock (f_{MAIN})	Select [High-speed on-chip oscillator (f_{IH})]. (f_{MAIN}: 16 MHz)
	Operating clock supply for 12-bit interval timer	Unchecked (f_{IL}: 15 kHz)
Smart Configurator >> System		The settings on the [System] tab are as follows:
	Pseudo RRM/DMM function setting	Not used
	Start/Stop function setting	Not used
	Security ID setting	Set security ID. Security ID "0x00000000000000000000"
	RESET pin setting	Used
	Operation mode setting	2.52 V
Smart Configurator >> Component >> r_bsp		Use the default settings except for the following changes:
	API functions disable(R_BSP_StartClock, R_BSP_StopClock)	Disable
	API functions disable(R_BSP_SetClockSource)	Disable
	API functions disable(R_BSP_ChangeClockSetting)	Disable
	API functions disable(R_BSP_SoftwareDelay)	Disable

Table 5-4 Smart Configurator Settings (2/2)

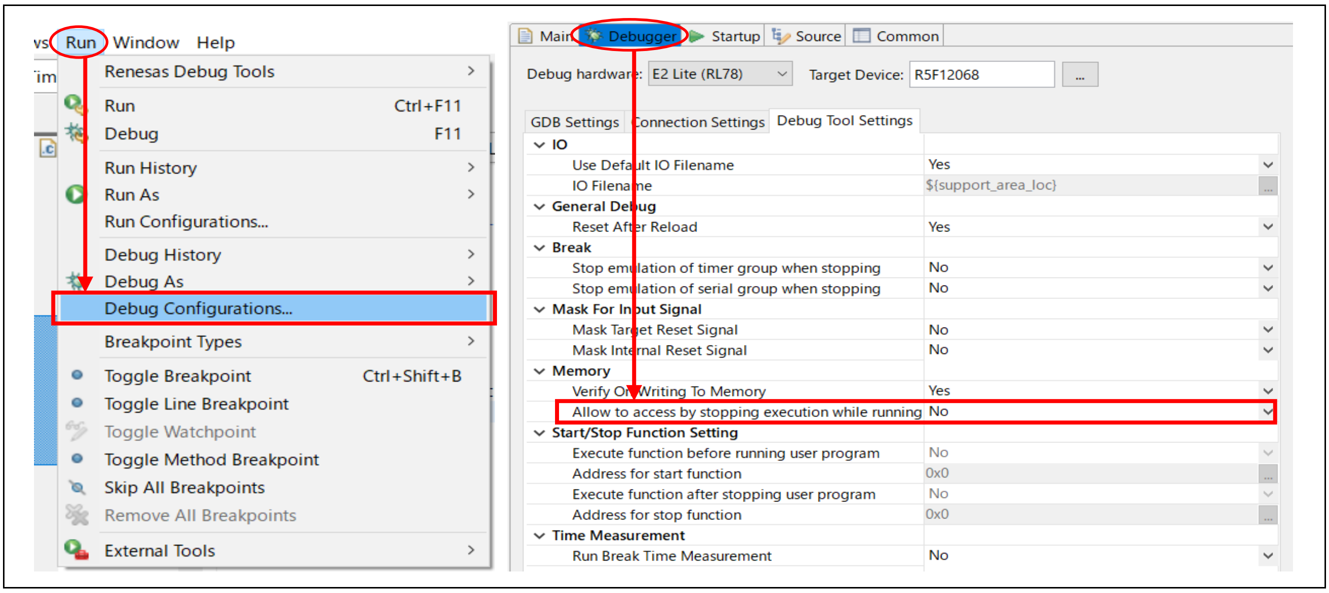
Category	Item		Settings
Smart Configurator >> Component >> PORT			Use the default settings except for the following changes:
	Port selection		PORT2
	PORT2		P20: Select [Output], [Output 1].
Smart Configurator >> Component >> Config_IICA0			The settings on [Config_IICA0] are as follows:
	Local address		16
	Operation mode setting		Set [Standard] and transfer clock (f_{SCL}): 100000 (bps)
	tR and tF settings		Unchecked
	Interrupt setting		Level 3 (lowest priority)
	Callback function setting		Select [Master transmission end], [Master reception end], and [Master error].
	Callback extension setting		Unchecked
Smart Configurator >> Component >> Config_UART			The settings on [Config_UART0] are as follows:
	Transmission	UART0 clock setting	Operating clock: CK00 Clock source: f_{CLK}/2^3
		Transfer mode setting	Continuous transfer mode
		Data bit length setting	8 bits
		Data transfer direction setting	LSB
		Parity setting	No parity bit
		Stop bit length setting	1 bit
		Transmit data level setting	Non-inversion (normal)
		Transfer rate	57600 (bps)
		Interrupt setting	Level 3 (lowest priority)
		Callback function setting	End of transmission
	Reception	UART0 clock setting	Operating clock: CK00 Clock source: f_{CLK}/2
		Data bit length setting	8 bits
		Data transfer direction setting	LSB
		Parity setting	No parity bit
		Transmit data level setting	Non-inversion (normal)
		Transfer rate	9600 (bps)
		Interrupt setting	Level 3 (lowest priority)
		Callback function setting	Reception end, reception error
Smart Configurator >> Pin			The settings on the [Pin] tab are as follows:
	Serial array unit (SAU0)		SAU00 RxD0: P04 TxD0: P03
	Serial interface IICA (IICA0)		IICA0 SCLA0: P06 SDAA0: P07

5.4.2.1 How to Prevent Memory from being Overwritten

If you use the default debugger settings as is, the debugger will occupy the top address (4 bytes) of the RAM (RRM function). Therefore, you need to change the settings.

As shown in Figure 5-6, select the debugger memory in [Debug configuration] and change [Interrupt execution to access memory] to "No" to stop the RRM function and prevent the first 4 bytes of the RAM from being overwritten.

Figure 5-4 Change Debugger Settings



5.4.2.2 Notes on Generating Code with Smart Configurator

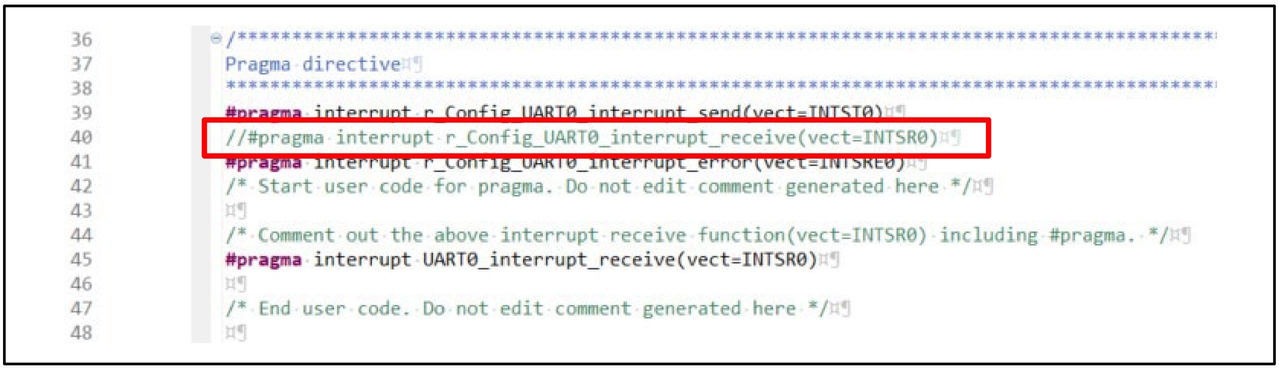
(1) UART receive interrupt function

The sample program provides a dedicated interrupt function to prepare for command recognition in the UART reception interrupt function. Although the #pragma command to call this function is also coded together, when new code is generated by Smart Configurator, the #pragma function (former reception interrupt function) is also coded. As a result, an interrupt table overload error occurs at build time.

After generating new code with the Smart Configurator, before building, make sure that the #pragma command to call the former reception interrupt function is commented out.

Target file: Config_UART0_user.c, line 40

Figure 5-5 Part to be Commented-out after Code Generation



(2) OCD Monitor Memory Area

In the "Linker > Device" settings, the memory area of the OCD monitor isIt is set with 01F00-01FFF in Smart Configurator, but in the case of RL78/G15, no additional memory area is required. For make effective use of the memory area, after generating new code with the Smart Configurator, before building, specify 0-0. (A yellow ▲ notification is displayed on e2studio, but this is not a problem.)

5.2.3. File Structure

The following table lists the file structure of the sample program.

Table 5-5 File Structure

Folder name, file name	Explanation
src	Program storage folder
├ command	Command related program storage folder
| ├ da14531_atcom.c	AT Command related source file
| └ da14531_atcom.h	AT Command related header file
| ├ command.c	Command related source file
| └ command.h	Command related header file
├ r_comms_i2c_rl	I2C communication middleware layer storage folder (the one for G23 is used)
├ r_hs300x	Sensor middleware layer storage folder (the one for G23 is used)
├ smc_gen	Smart Configurator generation folder
| ├ Config_IICA0	
| ├ Config_PORT	
| ├ Config_UART0	
| ├ general	
| ├ r_bsp	
| └ r_config	
├ main.c	Main processing source file
├ rl78_hs300x.c	Sensor software measurement flow source file
└ rl78_hs300x.h	Sensor software measurement flow header file

For details on the I^2C communication middleware layer storage folder, sensor middleware layer storage folder, and Smart Configurator generation folder, refer to the application notes stored in the doc folder in each folder.

6. Description of Software Operation

This sample program uses e2 studio as a tool for developing MCU programs and checking operation, and XCTU as a tool for controlling the DA14531MOD and checking its operation.

6.1. Initialization Using the AT Commands

DA14531MOD (Codeless DA14531 Data Pump Hex) performs the Advertising behavior with the Peripheral behavior at startup. After that, the Connect operation is executed from the Central side, and the communication mode is changed to Binary Mode for data communication.

An example of the initial configuration of the DA14531MOD is shown below.

Table 6-1 List of AT Commands / URC used to configure DA14531MOD

Items	AT commands and URC	Description
Basic AT command	AT	Checking communication with the module
Turns UART echo off	ATE=0	Set to Echo off to easy data decoding process
Connect from Central	+AWAKE +CONNECTED +COMMAND MODE SUPPORTED +BINARY MODE SUPPORTED	When Connect from Central, URC (Unsolicited Result Code) is sent from the Central side
Request Binary mode	AT+BINREQ	Binary Mode request
Acknowledge binary mode request	+BINREQACK	Acknowledge URC for binary mode request from Central

6.2. Preparation of Hardware

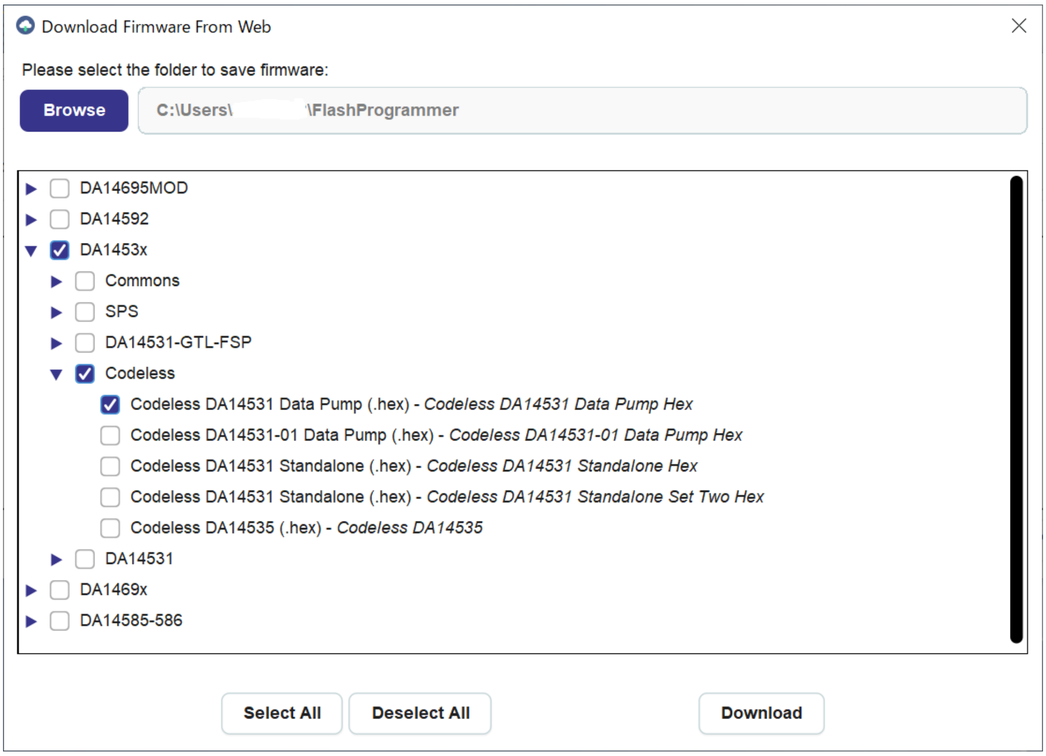
6.2.1. Preparation of DA14531MOD

DA14531MOD firmware uses Codeless DA14531 Data Pump Hex. For more information on how to upgrade the firmware of DA14531 modules, please refer to the following information.

US159-DA14531EVZ Firmware Upgrade — US159-DA14531EVZ Firmware Upgrade (renesas.com)

At Renesas SmartBond™ Flash Programmer's Download Firmware, select Codeless DA14531 Data Pump (.hex)

Figure 6-1 Select firmware of Renesas SmartBond™ Flash Programmer



6.2.2. Hardware Connectivity

RL78/G15 (RTK5RLG150C00000BJ) and DA14531MOD (US159-DA14531EVZ), HS300x (US082-HS3001EVZ) as follows. To setting the RTK5RLG150C00000BJ's operating power supply (VDD) to 3.3V, short-circuit J15 2-3.

Figure 6-2 Overall View of Connection

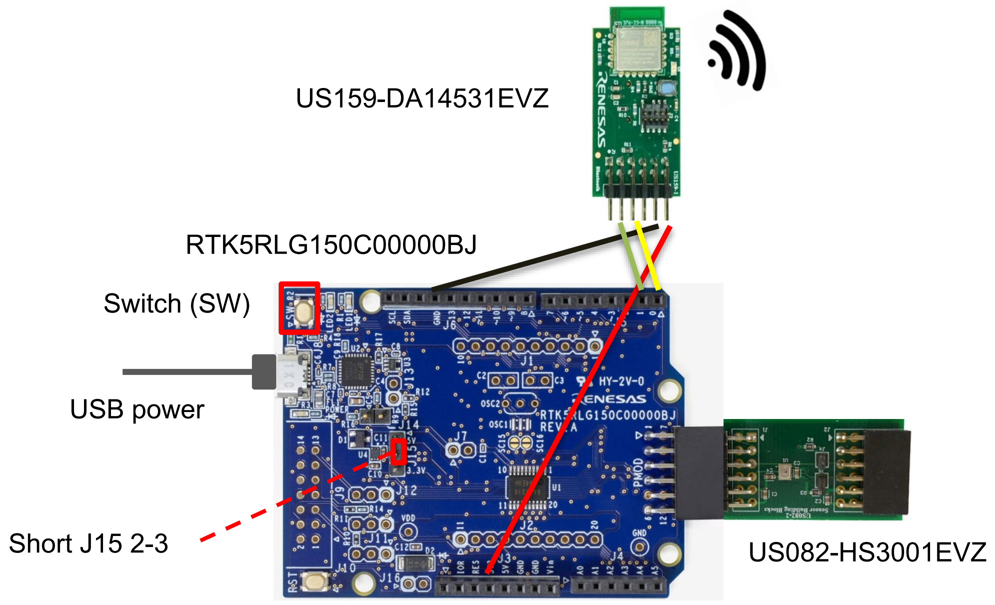


Table 6-2 Pins to which RTK5RLG150C00000BJ and US159-DA14531EVZ Connect

RTK5RLG150C00000BJ		US159-DA14531EVZ		Explanation
Pin number	Name	Pin number	Name	
J5: 1 P03	TXD0	Pmod 2pin	TXD	UART transmission from RL78/G15 to DA14531MOD
J5: 0 P04	RXD0	Pmod 3pin	RXD	UART reception from DA14531MOD to RL78/G15
J6: GND	GND	Pmod 5pin	GND	GND of RL78/G15 and DA14531MOD
J3: 3V3	3V3	Pmod 6pin	VCC	Power supply from RL78/G15 to DA14531MOD

Table 6-3 Pins to which RTK5RLG150C00000BJ and US082-HS3001EVZ Connect

RL78/ G15		HS300x		Explanation
Pin number	Name	Pin number	Name	
Pmod 3pin P06	SCLA0	Pmod 3pin	SCL	I^{2}C clock transmission from RL78/G15 to HS300x
Pmod 4pin P07	SDDA0	Pmod 4pin	SDA	I^{2}C data transmission from RL78/G15 to HS300x
Pmod 5pin	GND	Pmod 5pin	GND	GND of RL78/G15 and HS300x
Pmod 6pin	VCC	Pmod 6pin	3V3	Power supply from RL78/G15 to HS300x

6.2.3. Preparation of smartphone

Temperature and humidity data can be viewed with the Renesas SmartConsole app. Install the following:

Renesas SmartConsole - Google Play

Renesas SmartConsole on the App Store (apple.com)

6.3. Operation Confirmation Methods

This sample software allows the control of LED1 on the RL78/G15 Fast Prototyping Board and the HS300x sensor connected to the same board. It receives temperature and humidity data sent from the RL78/G15 Fast Prototyping Board, and displays the temperature and humidity on the smartphone.

(1) Set the power selection header of the RL78/G15 Fast Prototyping Board to 3.3V power (short J15 2-3), and supply power to the RL78/G15. At this point in time, the DA14531MOD is also powered.

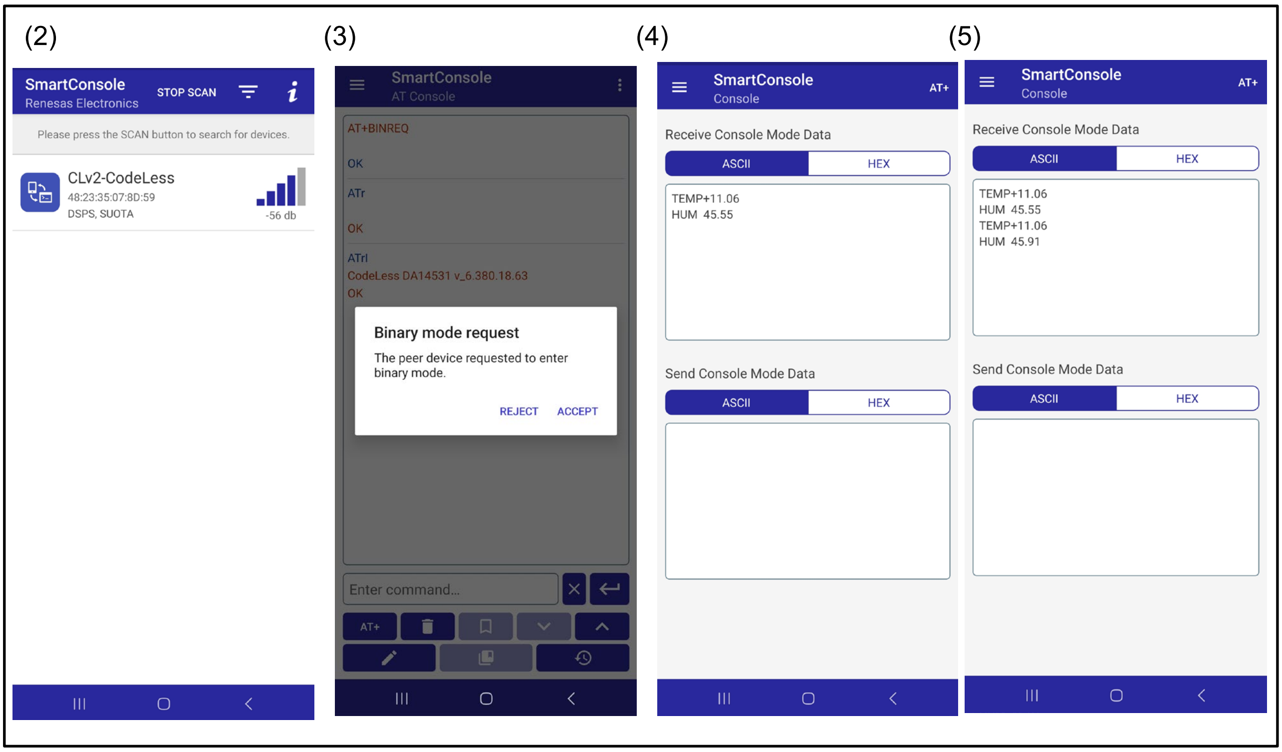
(2) Start the Renesas SmartConsole on your smartphone. Click the displayed CLv2-CodeLess to connect to the RL78/G15 Fast Prototyping Board side via Bluetooth LE.

(3) The RL78/G15 Fast Prototyping Board sends a Binary mode request and displays it on the smartphone side, so click ACCEPT.

(4) The RL78/G15 Fast Prototyping Board measures temperature and humidity data, sends it via Bluetooth LE, and displays it on the smartphone side. (When measuring temperature and humidity data, LED1 lights on.)

(5) By pressing and holding the Switch (SW) on the RL78/G15 Fast Prototyping Board, the temperature and humidity data is remeasured and sent via Bluetooth LE and displayed additionally on the smartphone side. (When measuring temperature and humidity data, LED1 lights on.)

Figure 6-3 Smartphone behavior



Remarks

The sample program does not implement the Bluetooth LE Disconnect process. The disconnect is turned off, or the Bluetooth LE communication is disconnected from the smartphone side. After disconnecting, press the RST (Reset) button on the RL78/G15 Fast Prototyping Board to reconnect.

Reference Documents

・RL78/G15 User’s Manual: Hardware (R01UH0959)

・RL78/G15 Fast Prototyping Board User’s Manual (R12UM0042)

・HS300x Datasheet (R36DS0010EU0701)

・RA Family, RX Family, RL78 Family, RE01 256KB / 1500KB Group, RZ Family HS300x Sample Software Manual (R01AN5897)

・RA Family, RX Family, RL78 Family, RZ Family Sensor Software Combination Manual (R01AN6591)

・RL78 Family Board Support Package Module Using Software Integration System（R01AN5522）

・RX Family Renesas HS300x Sensor Control Module Firmware Integration Technology (R01AN5893)

・RL78 Family Renesas Sensor Control Modules Software Integration System（R01AN6192）

・RX Family Renesas Sensor Control Modules Firmware Integration Technology（R01AN5892）

・RL78 Family Renesas Sensor I2C Communication Middleware Control Module Software Integration System (R01AN6193）

・RL78 Family Renesas HS300x Sensor Control Module Software Integration System（R01AN6194）

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb.9.2023	-	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.

Corporate Headquarters

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

Trademarks

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/.