RX Family

Data Transmission (Thermo Sensor) Using Wireless Module (Wi-Fi/Bluetooth)

Introduction

This application note describes a sample program for wireless communication by combining an RX microcontroller and a wireless module. Connecting RX microcontroller to wireless module can easily communicate wirelessly (Wi-Fi Note1/ Bluetooth® Note2). This application note provides sample programs using the following boards.

Board name	Board designation	Model
Renesas Solution Starter Kit for RX23E-A	RSSKRX23E-A	RTK0ESXB10C00001BJ
Low Power Bluetooth® Pmod™ Board	DA14531 Pmod™ Board	US159-DA14531EVZ
Ultra-Low Power Wi-Fi Pmod™ Board	DA16600 Pmod™ Board	US159-DA16600EVZ

This application note describes how to perform temperature measurement based on the "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747) and transmit the measured temperature data to the interface terminal via wireless communication using the Bluetooth® function of the DA14531 Pmod™ Board or the Wi-Fi function of the DA16600 Pmod™ Board.

It also explains how to connect to Amazon Web Service (AWS)Note3, one of the cloud services, over the Internet via Wi-Fi.

Note1 Wi-Fi is registered trademarks of Wi-Fi Alliance.

Note2 The Bluetooth® word mark and logo are registered trademarks owned by Bluetooth SIG, Inc. and Renesas Electronics Corporation uses these marks under license. Other trademarks and registered trademarks are the property of their respective owners.

Note3 Amazon Web Service, AWS are registered trademarks of Amazon.com,Inc. or its affiliated companies.

The board combinations and communication targets in each sample program are shown below.

Project Name	Board combination	Dara destination	Communication method
r01an6677_rx23ea_ble	RSSKRX23E-A + DA14531 Pmod™ Board	Smartphone	Bluetooth®
r01an6677_rx23ea_wifi	RSSKRX23E-A + DA16600 Pmod™ Board	PC	Wi-Fi
r01an6677_rx23ea_aws	RSSKRX23E-A + DA16600 Pmod™ Board	AWS	Wi-Fi

Target Device

RX23E-A Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

This application note requires the board (RSSKRX23E-A) modification. Please refer to the operational preparation for each sample program (3.1.5, 3.2.5 and 3.3.5 Hardware Preparation).

DA14531 Pmod™ Board and DA16600 Pmod™ Board are not included with the RSSKRX23E-A. They must be purchased separately.

Contents

1. Outline 5
1.1 Case of Bluetooth® 5
1.2 Case of Wi-Fi 5
1.3 Case of AWS 6

2. Operation Confirmation Conditions 7

3. Sample Programs 9
3.1 Bluetooth® Demonstration Project (r01an6677_rx23ea_ble) 10
3.1.1 System Structure 10
3.1.2 Software Structure 10
3.1.3 Overview Flowchart 11
3.1.4 Sample Program Structure 12
3.1.4.1 Pins Used 12
3.1.4.2 Peripheral Functions Used 12
3.1.4.3 Peripheral Function Settings 13
3.1.4.4 File Structure 14
3.1.4.5 Variables 15
3.1.4.6 Constants 15
3.1.4.7 Functions 16
3.1.4.8 Function Specifications 16
3.1.4.9 Bluetooth® Demonstration Flowchart 17
3.1.5 Hardware Preparation 18
3.1.5.1 Chip resistance removal 18
3.1.5.2 Pin header implementation 19
3.1.5.3 Connecting to RSSKRX23E-A to the DA14531 Pmod™ Board 19
3.1.6 Software Preparation 20
3.1.6.1 Advance Preparation of Smartphone Application 20
3.1.7 Sample Program Operation Overview 21
3.1.8 Sample Program Operation Details 22
3.2 Wi-Fi Demonstration Project (r01an6677_rx23ea_wifi) 24
3.2.1 System Structure 24
3.2.2 Software Structure 24
3.2.3 Overview Flowchart 25
3.2.4 Sample Program Structure 26
3.2.4.1 Pins Used 26
3.2.4.2 Peripheral Functions Used 26
3.2.4.3 Peripheral Function Settings 27
3.2.4.4 File Structure 28
3.2.4.5 Variables 29

3.2.4.6 Constants 29
3.2.4.7 Functions 30
3.2.4.8 Function Specification 30
3.2.4.9 Wi-Fi Demonstration Flowchart 32
3.2.5 Hardware Preparation 34
3.2.5.1 Chip resistance removal 34
3.2.5.2 Pin header implementation 35
3.2.5.3 Connecting to the RSSKRX23E-A to the DA16600 Pmod™ Board 35
3.2.6 Software Preparation 35
3.2.6.1 Tera Term Preparation 35
3.2.7 Sample Program Operation Overview 36
3.2.8 Sample Program Operation Details 37
3.3 AWS Demonstration Project (r01an6677_rx23ea_aws) 40
3.3.1 System Structure 40
3.3.2 Software Structure 41
3.3.3 Overview Flowchart 42
3.3.4 Sample Program Structure 43
3.3.4.1 Pins Used 43
3.3.4.2 Peripheral Functions Used 43
3.3.4.3 Peripheral Function Settings 44
3.3.4.4 File Structure 45
3.3.4.5 Variables 46
3.3.4.6 Constants 46
3.3.4.7 Functions 47
3.3.4.8 Function Specifications 47
3.3.4.9 AWS Demonstration Flowchart 49
3.3.5 Hardware Preparation 52
3.3.5.1 Chip resistance removal 52
3.3.5.2 Pin header implementation 53
3.3.5.3 Connecting to RSSKRX23E-A to the DA16600 Pmod™ Board 53
3.3.6 AWS Preparation 54
3.3.6.1 Applying Root CA Certificate and Device Certificate and Private Key in Source Code 55
3.3.7 Sample Program Operation Overview 57
3.3.8 Sample Program Operation Details 58
3.3.8.1 How to check temperature data in AWS 60

4. Preparing a Project for Execution 61
4.1 Procedure in e^2 studio 61
4.1.1 Importing a Project in e2-studio 61
4.1.2 Set build options 62
4.1.3 Build the project. 62

4.1.4 Debug 62
4.1.5 Run 63
4.2 Procedure in CS+ 64
4.2.1 Build the project. 65
4.2.2 Debug 65
4.2.3 Run 66

5. Troubleshooting 67
5.1 Unable to connect to AWS 67
5.1.1 How to check the DA16600 Pmod™ Board logs 67
5.1.1.1 Required Parts 67
5.1.1.2 Connection method 67
5.1.1.3 Open Tera Term 68
5.1.1.4 Tera Term Terminal Settings 68
5.1.1.5 Tera Term Serial Settings 69
5.1.1.6 Check operation 70
5.1.2 Update the firmware version of the DA16600 Pmod™ Board 70
5.1.2.1 Download firmware 71
5.1.2.2 Unzip the downloaded file 72
5.1.2.3 Connect with Pmod USBUART 72
5.1.2.4 Power supply 72
5.1.2.5 Open Tera Term 73
5.1.2.6 Tera Term Terminal Settings 73
5.1.2.7 Tera Term Serial Settings 73
5.1.2.8 Update the firmware 74
5.1.2.9 Check the version 75
5.1.3 If "Fail to establish tls-sess(0x7200)" is displayed 76
5.1.3.1 Buffer reconfiguration for MQTT 76
5.1.3.2 Check the setting result 77

6. Reference Documents 78

Revision History 79

1. Outline

This application note describes how to perform temperature measurement based on the "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747) and transmit the measured temperature data to the interface terminal via wireless communication using the Bluetooth® function of the DA14531 Pmod™ Board or the Wi-Fi function of the DA16600 Pmod™ Board.

Sample programs are available for three cases: Bluetooth®, Wi-Fi and AWS.

1.1 Case of Bluetooth®

In the case of Bluetooth®, RSSKRX23E-A is connected to the DA14531 Pmod™ Board and communicates with a smartphone. The data from the Bluetooth® communication is viewed via a smartphone application. The name of the smartphone application is SmartConsole from Renesas.

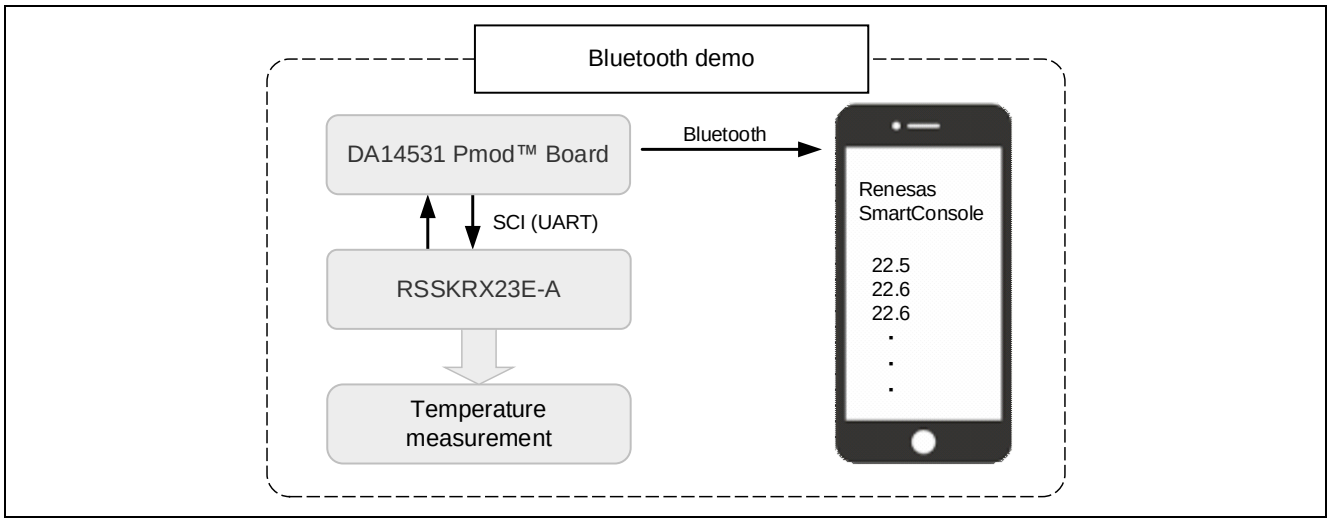

Figure 1-1 Bluetooth® Demonstration Structure Image

1.2 Case of Wi-Fi

In the case of Wi-Fi, RSSKRX23E-A is connected to the DA16600 Pmod™ Board and communicates with a PC. The data from the Wi-Fi communication is checked using Tera Term.

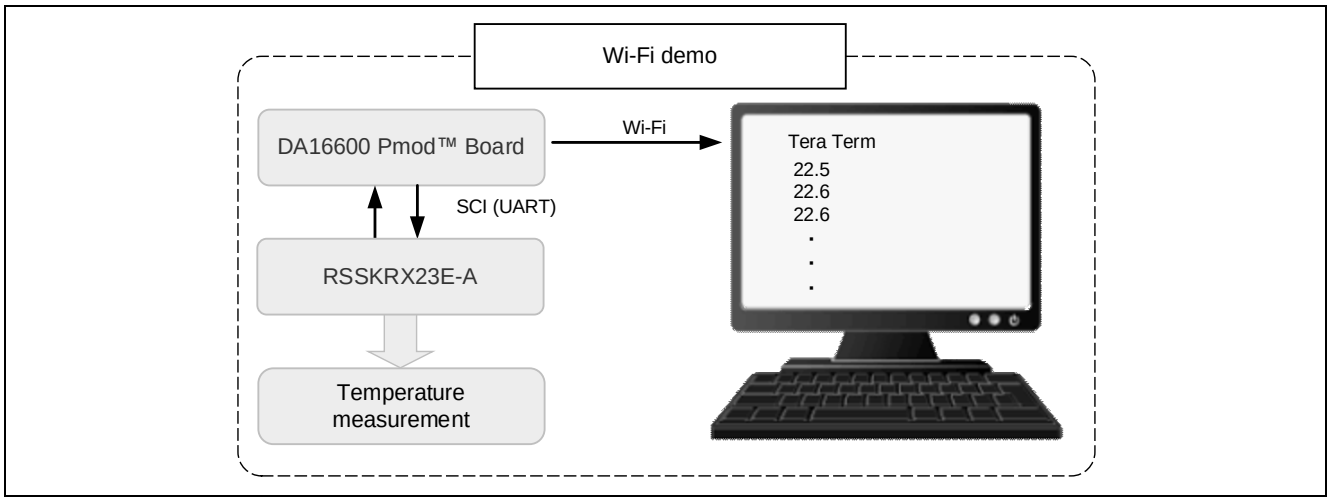

Figure 1-2 Wi-Fi Demonstration Structure Image

1.3 Case of AWS

In the case of AWS, you will need to create your own account. Please refer to "3.3.6 AWS Preparation" on how to create an account.

In the AWS demonstration, RSSKRX23E-A is connected to the DA16600 Pmod™ Board and communicates via the MQTT protocol using the AWS IoT Core service from AWS. The AWS communication data is viewed in the AWS console.

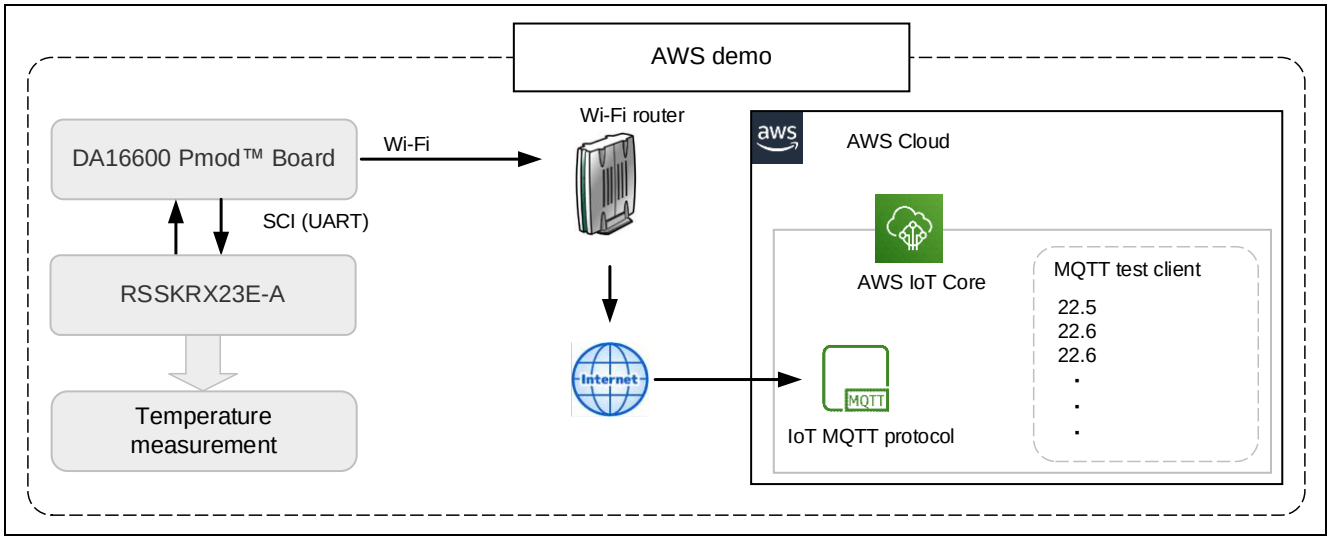

Figure 1-3 AWS Demonstration Structure Image

2. Operation Confirmation Conditions

The operation of the sample program has been confirmed under the following conditions.

Table 2-1 Operation Confirmation Conditions of microcontroller (RX23E-A)

Item	Contents
MCU used	R5F523E6ADFL (RX23E-A Group)
CPU max. operating frequency	32MHz
Bit count	32bit
Package/pin count	LFQFP / 48 Pins
ROM	256K Byte
RAM	32K Byte
Power voltage	3.3V

Table 2-2 Tools Used

Item	Contents
Integrated development environment	Renesas Electronics e^2 studio Version 2023.07
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V3.05.00
	Compiler option Default settings of integrated development environment
Smart Configurator	V2.18.0
Board support package (r_bsp)	V7.41
Endian order	Little Endian
Operating mode	Single chip mode
Processor mode	Super visor mode
Emulator	E2 Emulator Lite
Board used	RSSKRX23E-A Board (RTK0ESXB10C00001BJ)
Communication software	Tera Term (Version 4.106)
OS	None

Table 2-3 Operation Confirmation Conditions (DA16600 Pmod™ Board)

Item	Contents
Board used	US159-DA16600EVZ
Firmware	DA 16600 v3.2.8.0 • Connecting to Wi-Fi and TLS communication to AWS are performed by this firmware • If the version is different, refer to "5.1.2 Updating the Firmware Version of the DA16600 Pmod™ Board"

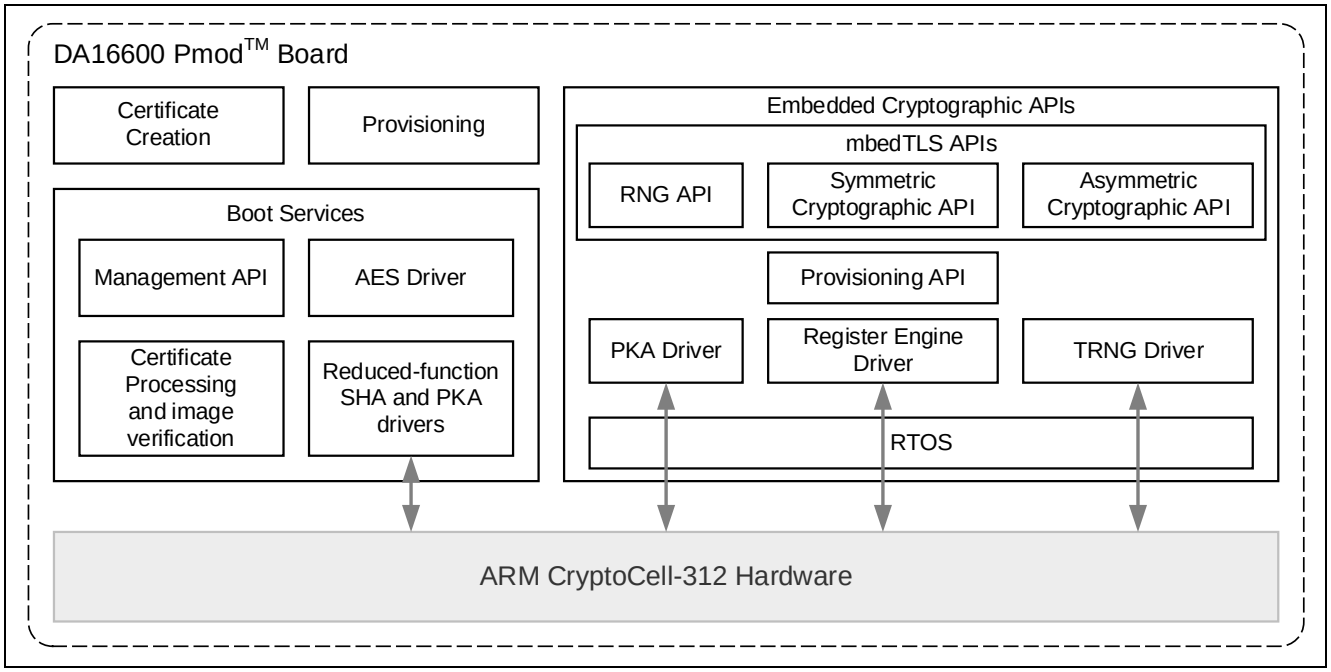

Figure 2-1 DA16600 Software Stack

Table 2-4 Operation Confirmation Conditions (DA14531 Pmod™ Board)

Item	Contents
Board used	US159-DA14531EVZ
Firmware	Non-public • Advertise communication to Bluetooth®, etc. is performed by this firmware

3. Sample Programs

This application note provides the following sample programs. These sample programs have been checked to work in e^2 studio.

Table 3-1 Sample Programs

Project name	Contents	Reference
r01an6677_rx23ea_ble	Connect DA14531 Pmod™ Board and perform Bluetooth® demonstration	3.1
r01an6677_rx23ea_wifi	Connect DA16600 Pmod™ Board and perform Wi-Fi demonstration	3.2
r01an6677_rx23ea_aws	Connect DA16600 Pmod™ Board and perform AWS demonstration	3.3

3.1 Bluetooth® Demonstration Project (r01an6677_rx23ea_ble)

Connect RSSKRX23E-A to the DA14531 Pmod™ Board for the Bluetooth® demonstration.

The project to perform Bluetooth® demonstration is r01an6677_rx23ea_ble. To execute this project, hardware modification is necessary. If you wish to proceed with the project, please proceed to section 3.1.5Hardware Preparation.

3.1.1 System Structure

The following shows the system structure of this sample program.

For the connection of the RSSKRX23E-A Board, refer to Page4 Figure 4-1 of the "RX23E-A Group Temperature Measurement Example Using a Thermocouple" Application Note (R01AN4747).

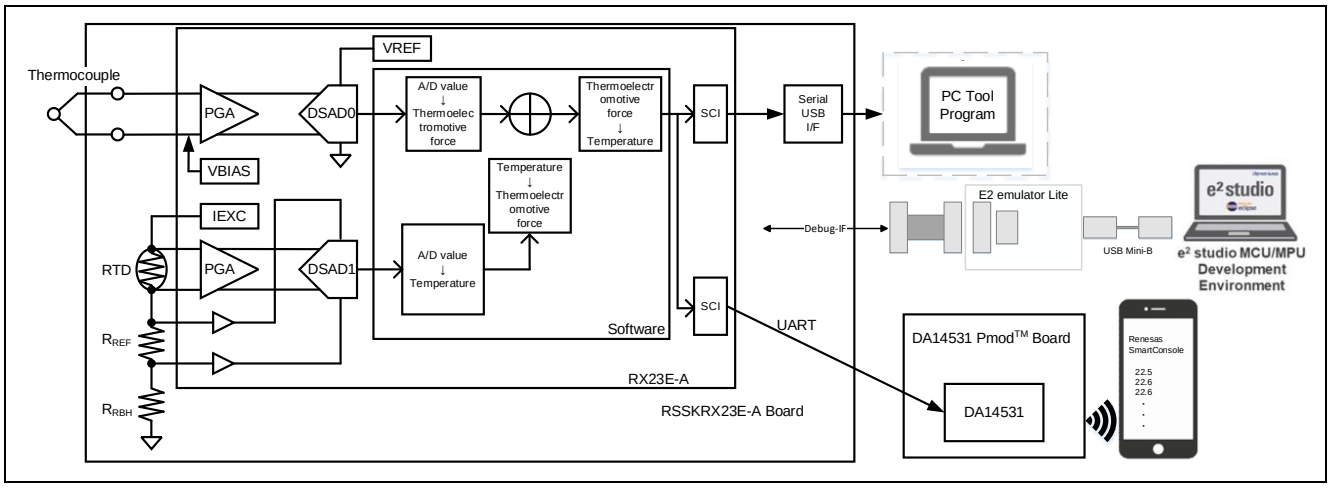

Figure 3-1 System Structure of Bluetooth® Demonstration

Note: This PC tool is not required for this operation, but it can be used if needed.

3.1.2 Software Structure

The following shows the software structure of this sample program. The blue part of the RSSKRX23E-A Board is the unchanged part from the original sample program. All Bluetooth® control is performed by the DA14531 Pmod™ Board. The software structure of DA14531 Pmod™ Board is not in public.

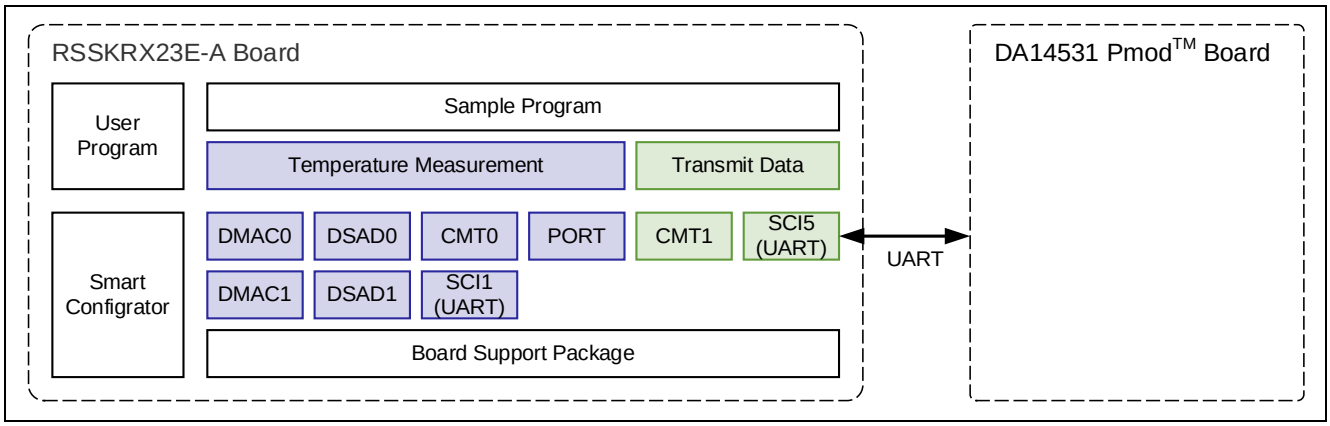

Figure 3-2 Software Structure of Bluetooth® Demonstration

3.1.3 Overview Flowchart

The following is an overview flowchart of this sample program.

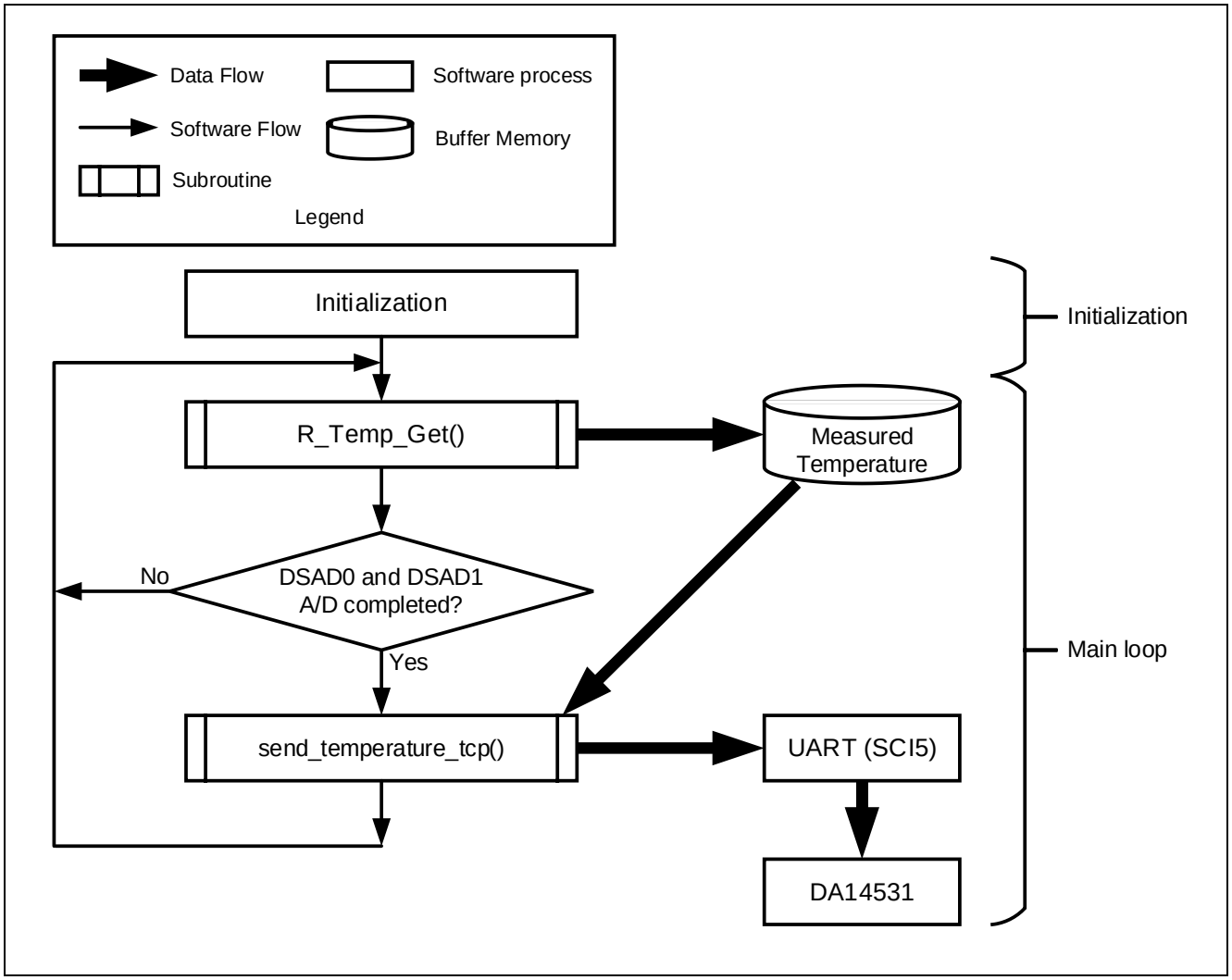

Figure 3-3 Overview Flowchart

3.1.4 Sample Program Structure

3.1.4.1 Pins Used

The following is a list of pins used on RX23E-A in this sample program.

Table 3-2 List of Pins and Functions

Pin name	Input / Output	Functions
PH2	Output	LED1 lighting control
P26/TXD1	Output	UART1 transmit pin
P30/RXD1	Input	UART1 receive pin
P31/CTS1#	Input	CTS signal input pin
AIN11	Input	Thermocouple + side input pin
AIN10	Input	Thermocouple - side input pin
AIN9	Output	RTD excitation current output pin
AIN7	Input	RTD + side input pin
AIN6	Input	RTD - side input pin
AIN5/REF1P	Input	RTD measurement DSAD + side reference voltage
AIN4/REF1N	Input	RTD measurement DSAD - side reference voltage
PH1/TXD5Note1	Output	Connect to TXD on DA14531 Pmod™ Board
PH0/RXD5Note1	Input	Connect to RXD on DA14531 Pmod™ Board
VCCNote1	-	Supply 3.3V to DA14531 Pmod™ Board
VSSNote1	-	Connect to VSS on DA14531 Pmod™ Board

Note1. Pins added from the base " RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.1.4.2 Peripheral Functions Used

The following lists peripheral functions used in sample program.

Table 3-3 List of Peripheral Functions Used and Functions

Peripheral function	Functions	Addition
AFE,DSAD0,DSAD1	Driving thermocouples and RTDs (AFE), A/D conversion of thermocouples (DSAD0), A/D conversion of RTDs (DSAD1)	-
SCI1	UART communication with PC tool programs	-
DMAC0	Data transfer triggered by SCI1 receive completion interrupt	-
DMAC3	Data transfer triggered by SCI1 buffer empty interrupt	-
CMT0	Communication timeout detection for SCI	-
PH2	LED1 lighting control	-
SCI5Note1	UART communication with DA14531 Pmod™ Board	yes
CMT1Note1	Interval control of temperature data transmission	yes

Note1. Peripheral functions added from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.1.4.3 Peripheral Function Settings

The peripheral function settings used in this sample program are based on the code generation function of the Smart Configurator. The following are the setting conditions for Smart Configurator. The following describes the peripheral functions added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747).

Table 3-4 SCI5 Settings

Item	Settings
Serial communication method	Start/Stop Synchronization
Start bit detection setting	Low level on RXD5 pin
Data bit length	8bit
Parity setting	Disabled
Stop bit setting	1bit
Data transfer direction setting	LSB First
Transfer rate setting	• Transfer clock : Internal clock • Bit rate : 115200bps • Bit rate modulation Function enabled • SCK5 pin function: SCK5 is disabled
Noise filter setting	Noise filter disabled
Hardware flow control setting	Hardware flow control setting : Disabled
Data processing setting	Transmit data processing : processed by interrupt service routine Received data processing : processed by interrupt service routine
Interrupt setting	Receive error interrupt enabled Priority : Level 15
Callback function setting	Disabled
Input / output pins	• Output : TXD5 (PH1) • Input : RXD5 (PH0)

Table 3-5 CMT1 Settings

Item	Settings
Clock setting	PCLKB/512
Compare match setting	• Interval time : 10 ms • Compare match interrupt enabled (CMI1) • Priority: Level 15 (interrupt disabled)

3.1.4.4 File Structure

The following shows the file structure of this sample program.

Table 3-6 File Structure

Folder name, File name	Description
src	Folder for storing program
├ main.c[Note1]	Main process
├ r_ring_buffer_control_api.c	Ring buffer control program
├ r_ring_buffer_control_api.h	Ring buffer control API definition
├ r_sensor_common_api.c	Table search, linear interpolation process program
├ r_sensor_common_api.h	Table search, linear interpolation process API definition
├ r_thermocouple_api.c	Thermocouple measurement calculation program, temperature vs. thermoelectromotive force table
├ r_thermocouple_api.h	Thermocouple measurement calculation API definition
├ r_rtd_api.c	Resistance temperature detector measurement calculation program, temperature vs. resistance value table
├ r_rtd_api.h	Resistance temperature detector measurement calculation API definition
├ r_communication_control_api.c	Communication control program
├ r_communication_control_api.h	Communication control API definition
├ string_func.c[Note1]	AT command control program
├ string_func.h[Note1]	AT command control API definition
└ smc_gen	Smart Configurator generation
├ Config_AFE	
├ Config_CMT0	
├ Config_CMT1[Note1]	
├ Config_DMAC0	
├ Config_DMAC3	
├ Config_DSAD0	
├ Config_DSAD1	
├ Config_PORT	
├ Config_SCI1	
├ Config_SCI5[Note1]	
├ general	
├ r_bsp	
├ r_config	
└ r_pincfg	

Note1. The file with additions and modified from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.1.4.5 Variables

The following shows the variables that are used in this sample program.

The following describes the variables added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747). For variables other than the ones added, please refer to R01AN4747.

Table 3-7 List of variables used in the sample code

Variable name	Type	Contents
g_temp	volatile float	Temperature data
g_send_flg	volatile uint8_t	Temperature data transmission completion flag
g_rcv_buf	uint8_t	Buffer storing receive data

3.1.4.6 Constants

There are no constants added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747). For the list of constants, please refer to R01AN4747.

3.1.4.7 Functions

The following shows a list of functions used in this sample program.

The following describes the functions added and modified from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747).

Table 3-8 List of functions used in the sample code

Function name	Outline	
main	Main process	Modification
send_temperature_ble	Transmit temperature data	Addition

3.1.4.8 Function Specifications

The following shows function specifications that are used in this sample program.

[Function name] main

Outline	Main process
Header	None
Declaration	void main (void)
Description	Initialize peripheral functions. Measure temperature using a thermocouple, control the DA14531 Pmod™ Board, and transmit temperature data.
Arguments	None
Return Value	None
Remarks	None

[Function name] send_temperature_ble

Outline	Transmit temperature data
Header	string_func.h
Declaration	void send_temperature_ble (void)
Description	Set the command for transmission and temperature data in the transmission buffer and transmit to the DA14531 Pmod™ Board.
Arguments	None
Return Value	None
Remarks	None

3.1.4.9 Bluetooth® Demonstration Flowchart

The following shows the main function flowchart for the Bluetooth® demonstration.

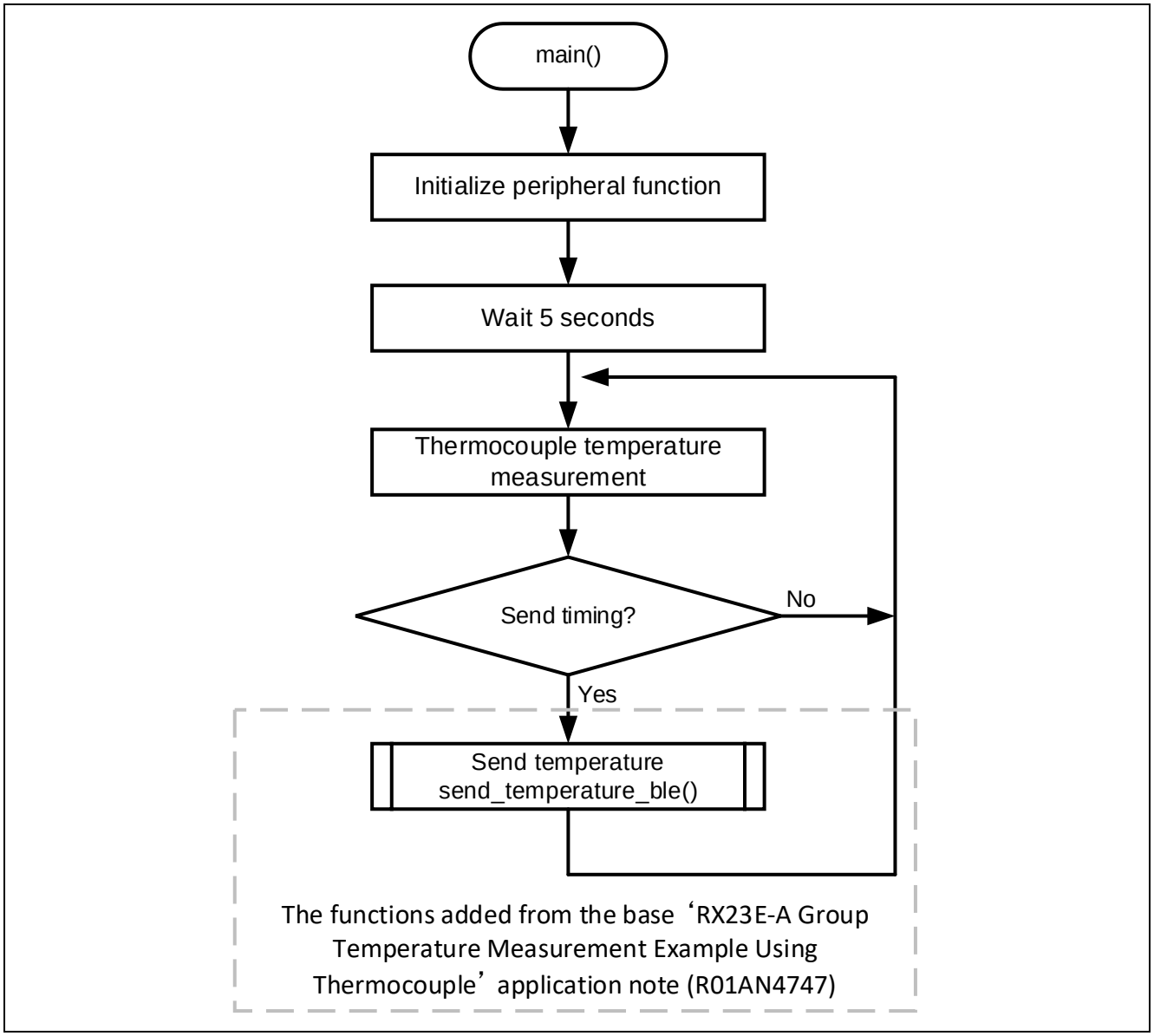

Figure 3-4 main function Flowchart

3.1.5 Hardware Preparation

This application note uses the thermocouple measurement circuit on the RSSKRX23E-A board. For usage details, refer to "2.4 Using the Analog Input Circuit" in the "RSSKRX23E-A User's Manual".

To connect RSSKRX23E-A to the DA14531 Pmod™ Board, the RSSKRX23E-A board must be modified.

The following is a list of pins to be modified. For details on pin numbers, refer to the "RSSKRX23E-A User's Manual".

Table 3-9 List of Pins to be modified

Pin number	MCU pin number	Function	Input / Output	Description
1	-	VSS	Output	VSS pin
2	-	VCC	Output	VCC pin Used for external power supply
3	23	PH1/TXD5	Input / Output	PH1/TXD5 pin
4	24	PH0/RXD5	Input / Output	PH0/RXD5 pin

3.1.5.1 Chip resistance removal

To use TXD5 and RXD5, the chip resistors must be removed. The chip resistors to be removed are R91 and R90.

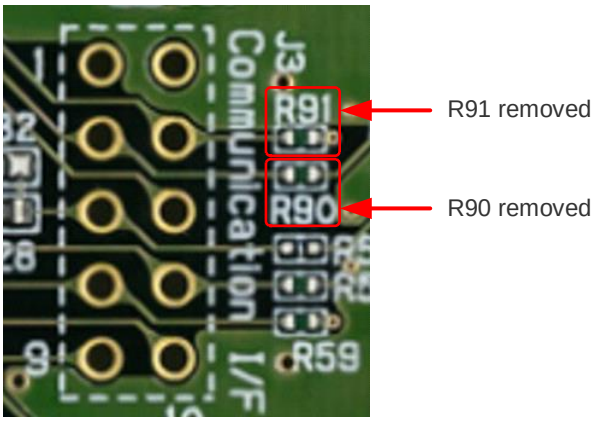

Figure 3-5 Chip resistance removal

3.1.5.2 Pin header implementation

Implement pin headers to use VSS, VCC, TXD5, and RXD5.

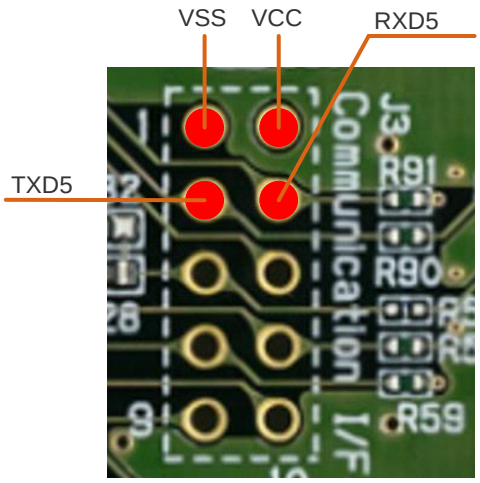

Figure 3-6 Connecting to the DA14531 Pmod™ Board

3.1.5.3 Connecting to RSSKRX23E-A to the DA14531 Pmod™ Board

Connect VSS, VCC, TXD5, RXD5 and DA14531 Pmod™ Board as follows.

Note to connect TXD5 on the RSSKRX23E-A to TXD on the DA14531 Pmod™ Board.

Table 3-10 Connection Table

RSSKRX23E-A		DA14531 Pmod™ Board		Supplyment
Pin number	Pin name	Pin number	Signal	
-	-	1	CTS	OPEN
3	PH1/TXD5	2	TXD	
4	PH0/RXD5	3	RXD	
-	-	4	RTS	OPEN
1	VSS	5	GND	
2	VCC	6	VCC	
-	-	7	GPIO	OPEN
-	-	8	GPIO	OPEN
-	-	9	GPIO	OPEN
-	-	10	GPIO	OPEN
-	-	11	GND	OPEN
-	-	12	VCC	OPEN

3.1.6 Software Preparation

3.1.6.1 Advance Preparation of Smartphone Application

In the Bluetooth® demonstration, the temperature data measured by the RSSKRX23E-A is transmitted to a smartphone. Therefore, the application (Renesas SmartConsole) must be installed on the smartphone. The smartphone application is provided globally in both the Apple App Store and the Google Play Store. The Smartphone application can be downloaded from the following link.

For details on how to use the Renesas SmartConsole, refer to
http://lpccs-docs.renesas.com/UM-140-DA145x-CodeLess/smartconsole.html#

Figure 3-7 Link to Smartphone application

3.1.7 Sample Program Operation Overview

The following shows an overview of the sample program operation. For detailed procedures to operate the sample program, Refer to "3.1.8 Sample Program Operation Details".

(1) Start
Import the sample program and execute it.

(2) Initialization
RSSKRX23E-A automatically initializes itself.

(3) Advertise & Scan
The DA14531 Pmod™ Board will automatically start advertising when power is applied. Now use the smartphone application and scan the DA14531 Pmod™ Board.

(4) Connection
Scan the DA14531 Pmod™ Board by smartphone and connect smartphone to the DA14531 Pmod™ Board.

(5) Temperature data transmission
Once the connection is completed, temperature data is automatically transmitted from the RSSKRX23E-A to the smartphone.

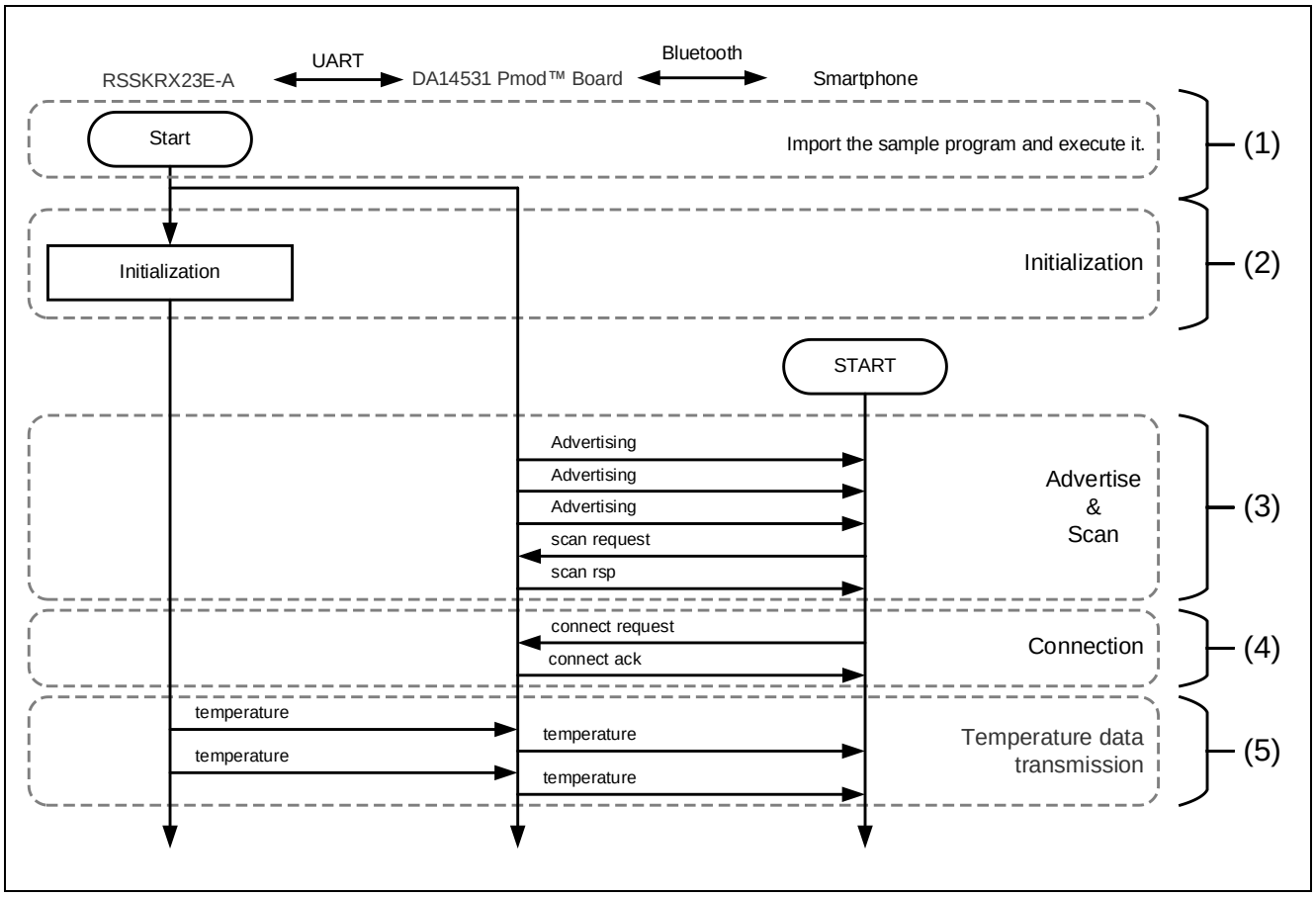

Figure 3-8 Bluetooth® Demonstration Flowchart

3.1.8 Sample Program Operation Details

Follow these procedures to perform the demonstration.

Note If you have updated the firmware of the DA14531 Pmod™ Board, please use Renesas SmartConfig(app) to reconfigure the baud rate of the DA14531 Pmod™ Board to 115200bps.

(1) Steps from Importing to Executing the Sample Program
Please follow the '4 Preparing a Project for Execution' and execute the project.
When power is supplied from the dubugger,LED2 will illuminate.

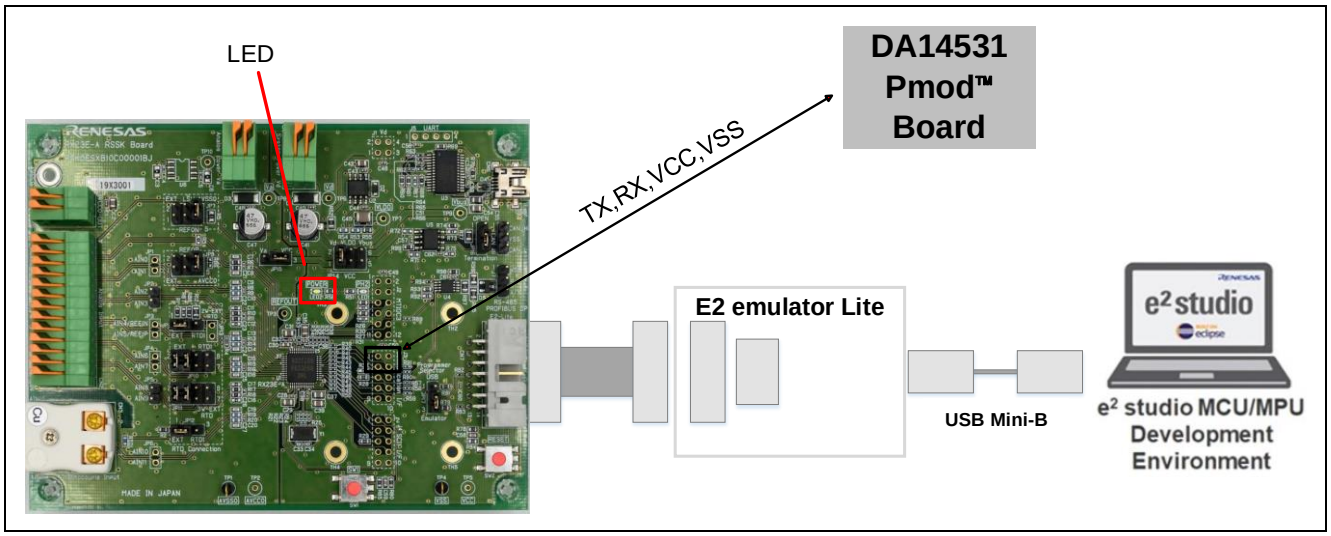

Figure 3-9 LED2 (Power) Position

(2) Initialization
RSSKRX23E-A automatically initializes its internal state.

(3) Advertise & Scan
The DA14531 Pmod™ Board will automatically start advertising when power is applied. Now scan the DA14531 Pmod Board by using application (Renesas SmartConsole) in your smartphone.

(4) Connection
On the Renesas SmartConsole screen, tap DA14531's device name to connect automatically.

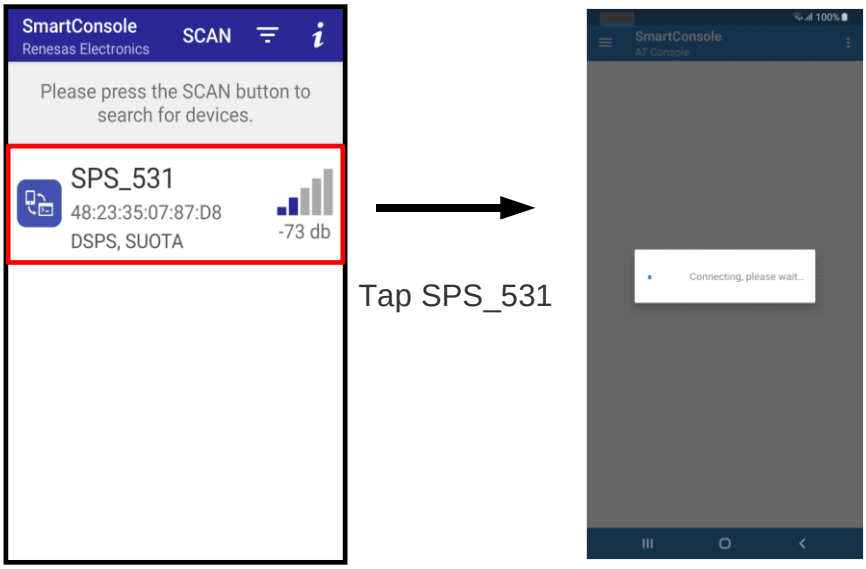

Figure 3-10 Scanning DA14531 Pmod™ Board

(5) Temperature data transmission
Once the connection is completed, temperature data is automatically transmitted from the RSSKRX23E-A to the smartphone.

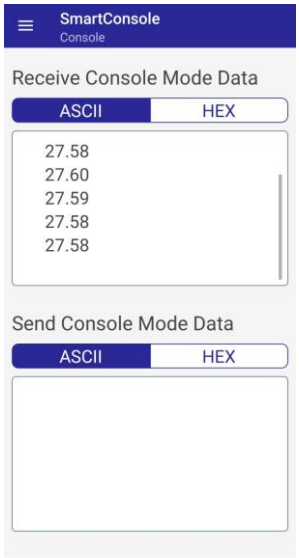

Figure 3-11 Scanning DA14531 Pmod™ Board

3.2 Wi-Fi Demonstration Project (r01an6677_rx23ea_wifi)

Connect RSSKRX23E-A to the DA16600 Pmod™ Board for the Wi-Fi demonstration.

The project to perform Wi-Fi demonstration is r01an6677_rx23ea_wifi. To execute this project, hardware modification is necessary. If you wish to proceed with the project, please proceed to section 3.2.5 Hardware Preparation.

3.2.1 System Structure

The following shows the system structure of this sample program.

For the connection of the RSSKRX23E-A Board, refer to Page4 Figure 4-1 of the "RX23E-A Group Temperature Measurement Example Using a Thermocouple" Application Note (R01AN4747).

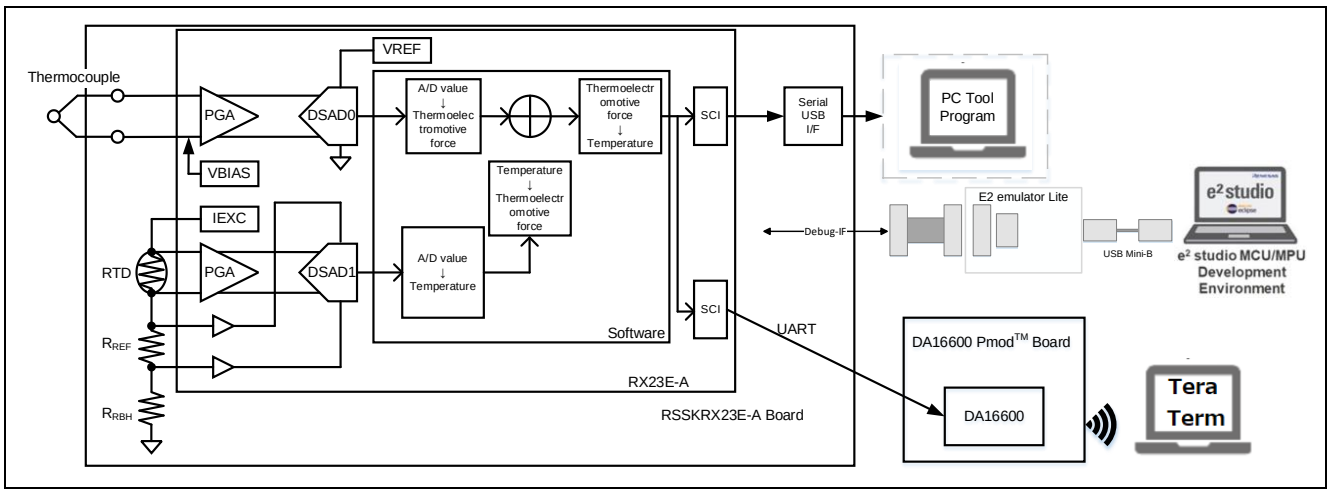

Figure 3-12 System Structure for Wi-Fi Demonstration

Note: This PC tool is not required for this operation, but it can be used if needed.

3.2.2 Software Structure

The following shows the software structure of this sample program. The blue part of the RSSKRX23E-A Board is the unchanged part from the original sample program. All communication with Wi-Fi and AWS is performed by the DA16600 Pmod™ Board.

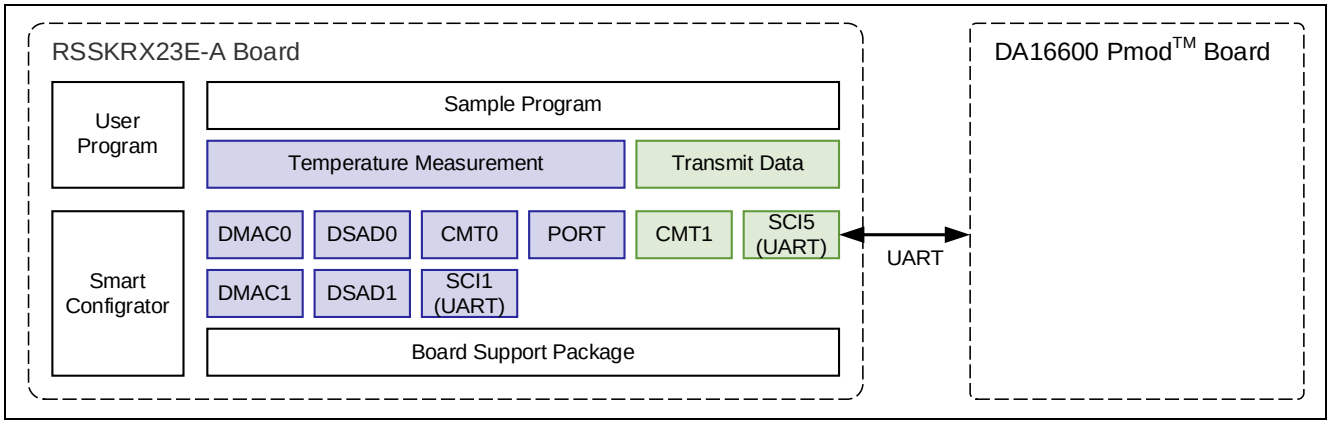

Figure 3-13 Software Structure for Wi-Fi Demonstration

3.2.3 Overview Flowchart

The following is an overview flowchart of this sample program.

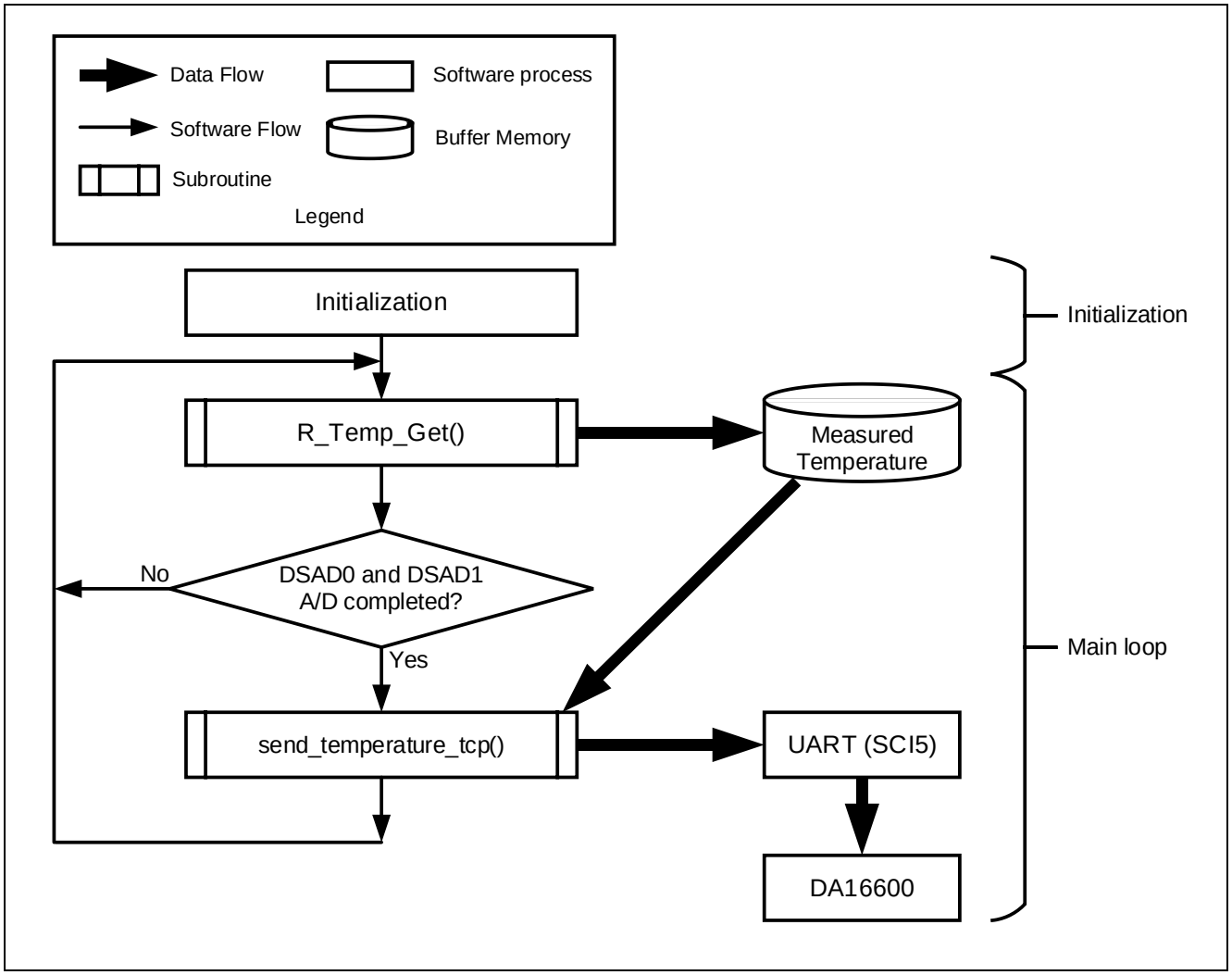

Figure 3-14 Overview Flowchart

3.2.4 Sample Program Structure

3.2.4.1 Pins Used

The following is a list of pins used on RX23E-A in this sample program.

Table 3-11 List of Pins and Functions

Pin name	Input / Output	Functions
PH2	Output	LED1 lighting control
P26/TXD1	Output	UART1 transmit pin
P30/RXD1	Input	UART1 receive pin
P31/CTS1#	Input	CTS signal input pin
AIN11	Input	Thermocouple + side input pin
AIN10	Input	Thermocouple - side input pin
AIN9	Output	RTD excitation current output pin
AIN7	Input	RTD + side input pin
AIN6	Input	RTD - side input pin
AIN5/REF1P	Input	RTD measurement DSAD + side reference voltage
AIN4/REF1N	Input	RTD measurement DSAD - side reference voltage
PH1/TXD5Note1	Output	Connect to TXD on DA16600 Pmod™ Board
PH0/RXD5Note1	Input	Connect to RXD on DA16600 Pmod™ Board
VCCNote1	-	Supply 3.3V to DA16600 Pmod™ Board
VSSNote1	-	Connect to VSS on DA16600 Pmod™ Board

Note1. Pins added from the base " RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.2.4.2 Peripheral Functions Used

The following shows lists peripheral functions used by sample program.

Table 3-12 List of Peripheral Functions Used and Functions

Peripheral functions	Functions	Addition
AFE,DSAD0,DSAD1	Driving thermocouples and RTDs (AFE), A/D conversion of thermocouples (DSAD0), A/D conversion of RTDs (DSAD1)	-
SCI1	UART communication with PC tool programs	-
DMAC0	Data transfer triggered by SCI1 receive completion interrupt	-
DMAC3	Data transfer triggered by SCI1 buffer empty interrupt	-
CMT0	Communication timeout detection for SCI	-
PH2	LED1 lighting control	-
SCI5Note1	UART communication with DA16600 Pmod™ Board	yes
CMT1Note1	Interval control of temperature data transmission	yes

Note1. Peripheral functions added from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.2.4.3 Peripheral Function Settings

The peripheral function settings used in this sample program are based on the code generation function of the Smart Configurator. The following are the setting conditions for Smart Configurator. The following describes the peripheral functions added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747).

Table 3-13 SCI5 Settings

Item	Settings
Serial communication method	Start/Stop Synchronization
Start bit detection setting	Low level on RXD5 pin
Data bit length	8bit
Parity setting	Disabled
Stop bit setting	1bit
Data transfer direction setting	LSB First
Transfer rate setting	• Transfer clock : Internal clock • Bit rate : 115200bps • Bit rate modulation Function enabled • SCK5 pin function: SCK5 is disabled
Noise filter setting	Noise filter disabled
Hardware flow control setting	Hardware flow control setting : Disabled
Data processing setting	Transmit data processing : processed by interrupt service routine Received data processing : processed by interrupt service routine
Interrupt setting	Receive error interrupt enabled Priority : Level 15
Callback function setting	Disabled
Input / output pins	• Output : TXD5 (PH1) • Input : RXD5 (PH0)

Table 3-14 CMT1 Settings

Item	Settings
Clock setting	PCLKB/512
Compare match setting	• Interval time : 10 ms • Enable compare match interrupt (CMI1) • Priority: Level 15 (interrupt disabled)

3.2.4.4 File Structure

The following shows the file structure of this sample program.

Table 3-15 File Structure

Folder name, File name	Description
src	Folder for storing program
├ main.c Note1	Main process
├ r_ring_buffer_control_api.c	Ring buffer control program
├ r_ring_buffer_control_api.h	Ring buffer control API definition
├ r_sensor_common_api.c	Table search, linear interpolation process program
├ r_sensor_common_api.h	Table search, linear interpolation process API definition
├ r_thermocouple_api.c	Thermocouple measurement calculation program, temperature vs. thermoelectromotive force table
├ r_thermocouple_api.h	Thermocouple measurement calculation API definition
├ r_rtd_api.c	Resistance temperature detector measurement calculation program, temperature vs. resistance value table
├ r_rtd_api.h	Resistance temperature detector measurement calculation API definition
├ r_communication_control_api.c	Communication control program
├ r_communication_control_api.h	Communication control API definition
├ string_func.cNote1	AT command control program
├ string_func.hNote1	AT command control API definition
└ smc_gen	Smart Configurator generation
├ Config_AFE	
├ Config_CMT0	
├ Config_CMT1Note1	
├ Config_DMAC0	
├ Config_DMAC3	
├ Config_DSAD0	
├ Config_DSAD1	
├ Config_PORT	
├ Config_SCI1	
├ Config_SCI5 Note1	
├ general	
├ r_bsp	
├ r_config	
└ r_pincfg	

Note1. The file with additions and changes from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.2.4.5 Variables

The following shows the variables that are used in this sample program.

The following describes the variables added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747). For variables other than the ones added, please refer to R01AN4747.

Table 3-16 List of variables used in sample code

Variable name	Type	Contents
g_temp	volatile float	Temperature data
g_rcv_end_flg	volatile uint8_t	DA16600 Pmod™ Board command response receive flag
g_send_flg	volatile uint8_t	Temperature data transmission completion flag
g_rcv_buf	uint8_t	Buffer storing receive data

3.2.4.6 Constants

There are no constants added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747). For the list of constants, please refer to R01AN4747.

3.2.4.7 Functions

The following shows a list of functions used in this sample program.

The following describes the functions added and changed from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747).

Table 3-17 List of functions used in the sample code

Function name	Description	
main	Main process	Modification
start_softap_mode	Set DA16600 Pmod™ Board to enable TCP communication	Addition
check_sci_rcv_end	Detect DA16600 Pmod™ Board response	Addition
check_rcv_cmd	Check contents received from DA16600 Pmod™ Board	Addition
reset_rcv_buf	Clear receive buffer	Addition
send_temperature_tcp	Transmit temperature data	Addition

3.2.4.8 Function Specification

The following shows function specifications that are used in this sample program.

[Function name]main

Outline	Main process
Header	None
Declaration	void main (void)
Description	Initialize peripheral functions. Measure temperature using a thermocouple, control the DA16600 Pmod™ Board, and perform transmission of temperature data.
Arguments	None
Return Value	None
Remarks	None

[Function name] start_softap_mode

Outline	Set DA16600 Pmod™ Board to enable TCP communication
Header	string_func.h
Declaration	void start_softap_mode (void)
Description	Transmit AT commands to the DA16600 Pmod™ Board. Set the DA16600 Pmod™ Board to soft-AP mode to enable TCP communication.
Arguments	None
Return Value	None
Remarks	None

[Function name] check_sci_rcv_end

Outline	Detect DA16600 Pmod™ Board response
Header	string_func.h
Declaration	void check_sci_rcv_end (void)
Description	Detects DA16600 Pmod™ Board responses from line feed codes.
Arguments	None
Return Values	None
Remarks	None

[Function name] check_rcv_cmd

Outline	Check contents received from DA16600 Pmod™ Board
Header	string_func.h
Declaration	void check_rcv_cmd (void)
Decription	Check contents received from DA16600 Pmod™ Board.
Arguments	None
Return Value	None
Remarks	None

[Function name] reset_rcv_buf

Outline	Clear receive buffer
Header	string_func.h
Declaration	void reset_rcv_buf (void)
Description	Clear receive buffer
Arguments	None
Return Value	None
Remarks	None

[Function name] send_temperature_tcp

Outline	Transmit temperature data
Header	string_func.h
Declaration	void send_temperature_tcp (void)
Description	Set the command for transmission and temperature data in the transmission buffer and transmit to the DA16600 Pmod™ Board.
Arguments	None
Return Value	None
Remarks	None

3.2.4.9 Wi-Fi Demonstration Flowchart

The following shows the main function flowchart for the Wi-Fi demonstration.

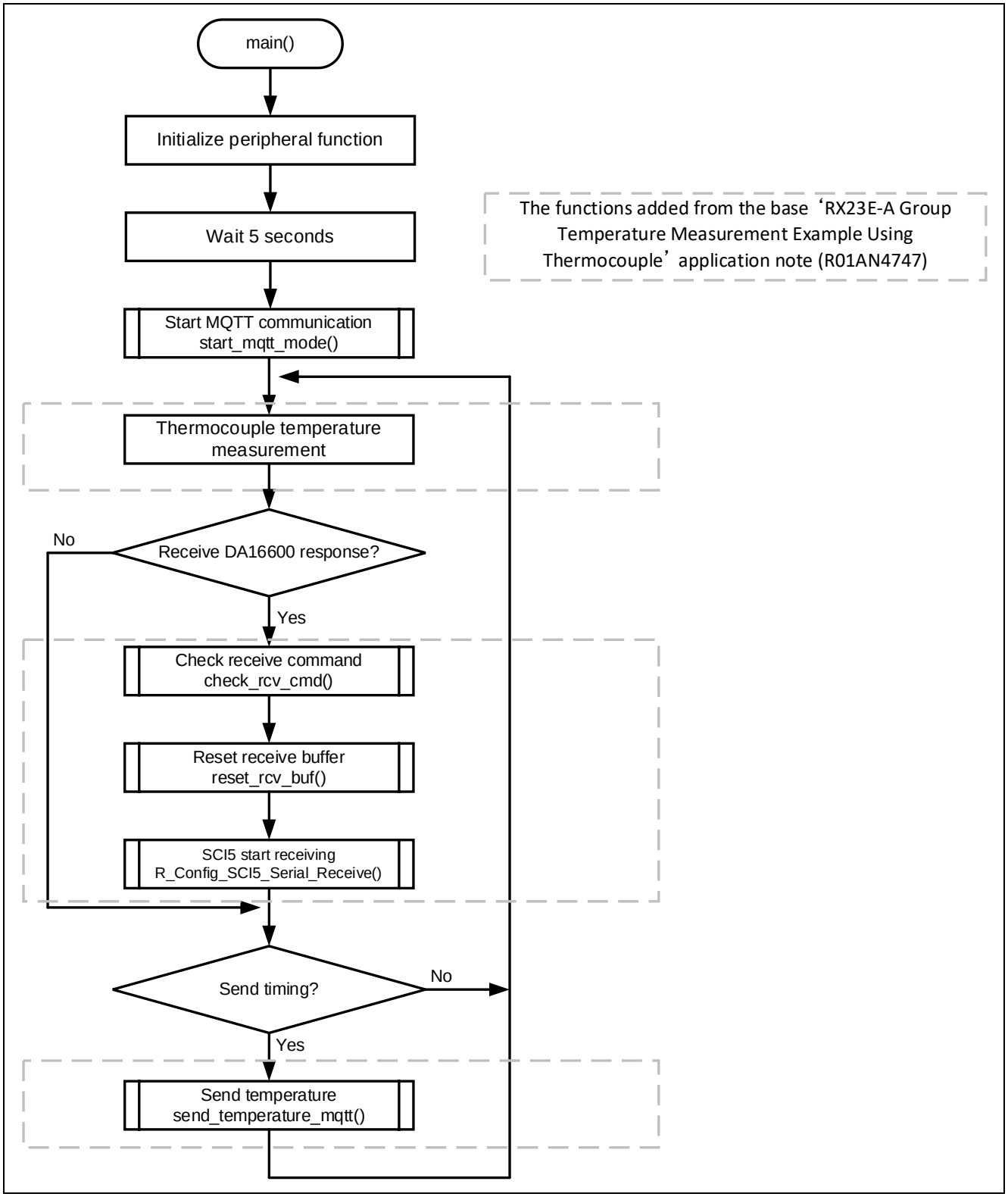

Figure 3-15 main function Flowchart

The following shows the start_softap_mode function flowchart for the Wi-Fi demonstration.

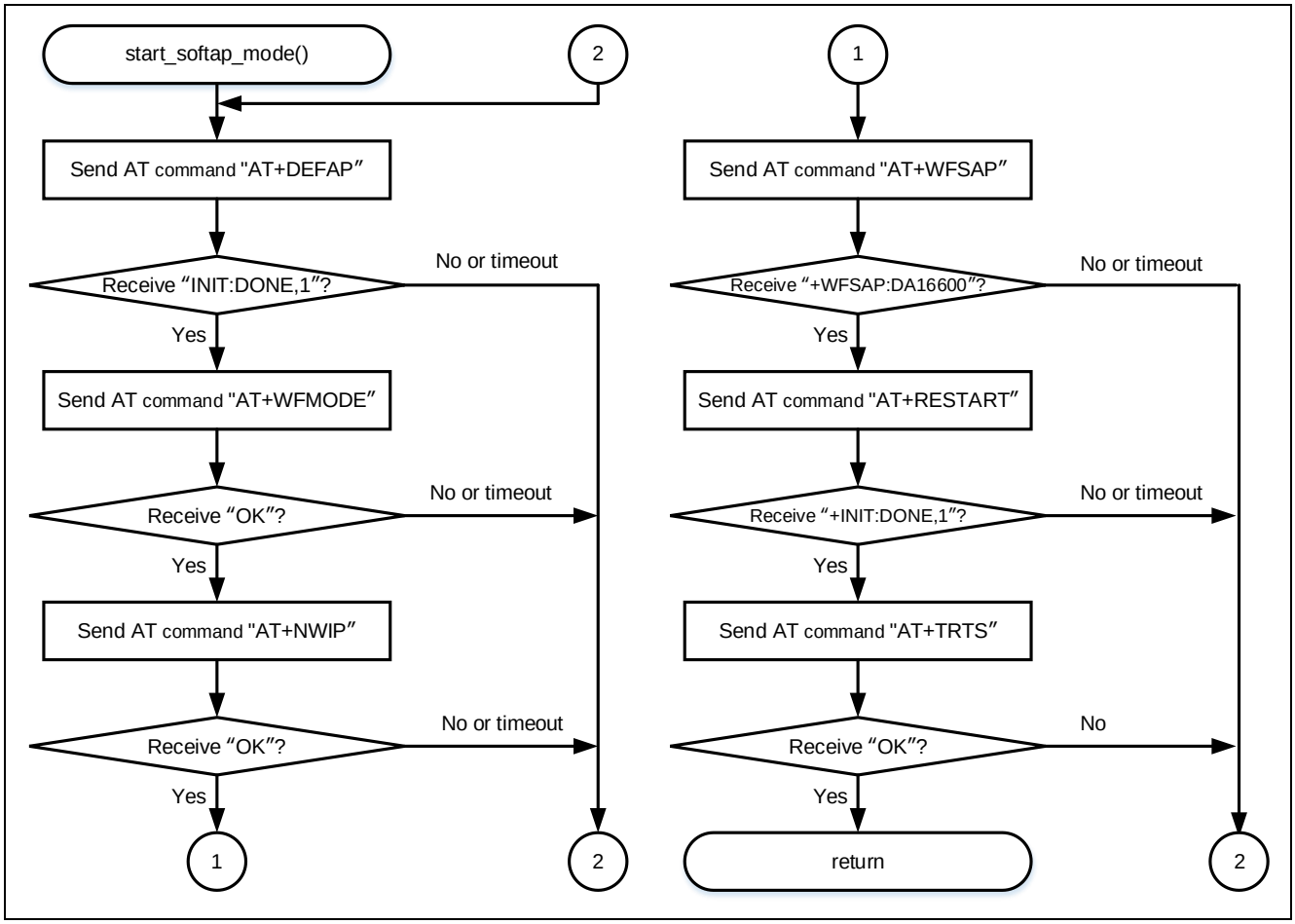

Figure 3-16 start_softap_mode function Flowchart

3.2.5 Hardware Preparation

This application note uses the thermocouple measurement circuit on the RSSKRX23E-A board. For usage details, refer to "2.4 Using the Analog Input Circuit" in the "RSSKRX23E-A User's Manual".

To connect RSSKRX23E-A to the DA16600 Pmod™ Board, the RSSKRX23E-A board must be modified.

The following is a list of pins to be modified. For details on pin numbers, refer to the "RSSKRX23E-A User's Manual".

Table 3-18 List of Pins to be modified

Pin number	MCU pin number	Function	Input / Output	Description
1	-	VSS	Output	VSS pin
2	-	VCC	Output	VCC pin Used for external power supply
3	23	PH1/TXD5	Input / Output	PH1/TXD5 pin
4	24	PH0/RXD5	Input / Output	PH0/RXD5 pin

3.2.5.1 Chip resistance removal

To use TXD5 and RXD5, the chip resistors must be removed. The chip resistors to be removed are R91 and R90.

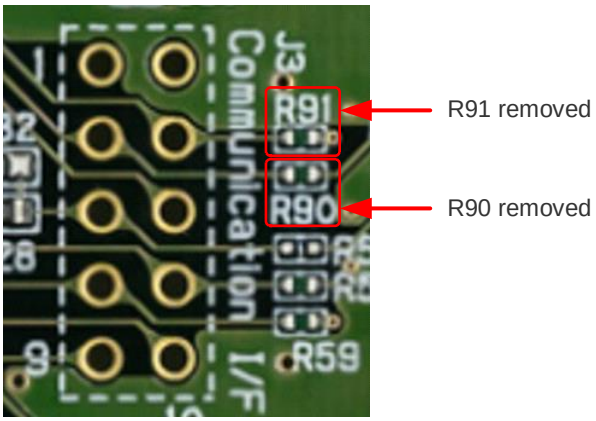

Figure 3-17 Chip resistance removal

3.2.5.2 Pin header implementation

Implement pin headers to use VSS, VCC, TXD5, and RXD5.

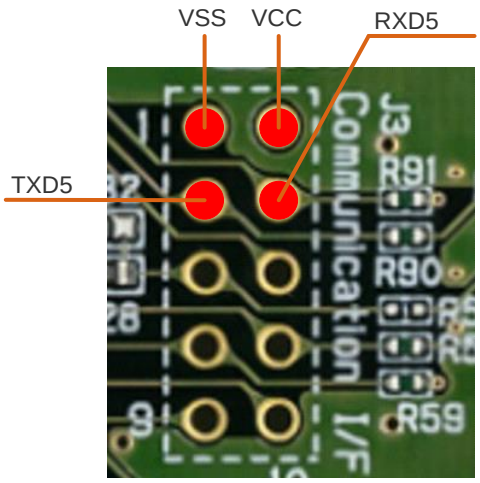

Figure 3-18 Connecting to the DA16600 Pmod™ Board

3.2.5.3 Connecting to the RSSKRX23E-A to the DA16600 Pmod™ Board

Connect VSS, VCC, TXD5, RXD5 and DA16600 Pmod™ Board as follows.

Note to connect TXD5 on the RSSKRX23E-A to TXD on the DA16600 Pmod™ Board.

Table 3-19 Connection Table

RSSKRX23E-A		DA16600 Pmod™ Board		Supplement
Pin number	Pin name	Pin number	Signal	
-	-	1	CTS	OPEN
3	PH1/TXD5	2	TXD	
4	PH0/RXD5	3	RXD	
-	-	4	RTS	OPEN
1	VSS	5	GND	
2	VCC	6	VCC	
-	-	7	GPIO	OPEN
-	-	8	GPIO	OPEN
-	-	9	GPIO	OPEN
-	-	10	GPIO	OPEN
-	-	11	GND	OPEN
-	-	12	VCC	OPEN

3.2.6 Software Preparation

3.2.6.1 Tera Term Preparation

In the Wi-Fi demonstration, the temperature data measured by RSSKRX23E-A is transmitted to a Windows PC. Tera Term must be installed on the Windows PC in order to check the data on the Windows PC.

3.2.7 Sample Program Operation Overview

The following shows an overview of the sample program operation. For detailed procedures to operate the sample program, Refer to "3.2.8 Sample Program Operation Details".

(1) Start
Import the sample program and execute it.

(2) Initialization
RSSKRX23E-A automatically initializes itself. RSSKRX23E-A also automatically sets the DA16600 Pmod™ Board to Soft AP mode.

(3) Wi-Fi Connection
From a Windows PC, connect to the DA16600 Pmod™ Board via Wi-Fi.

(4) Tera Term TCP/IP Connection
From Tera Term, connect to the DA16600 Pmod™ Board via TCP/IP.

(5) Temperature data translation
Once the connection is completed, temperature data is automatically transmitted from RSSKRX23E-A to Tera Term.

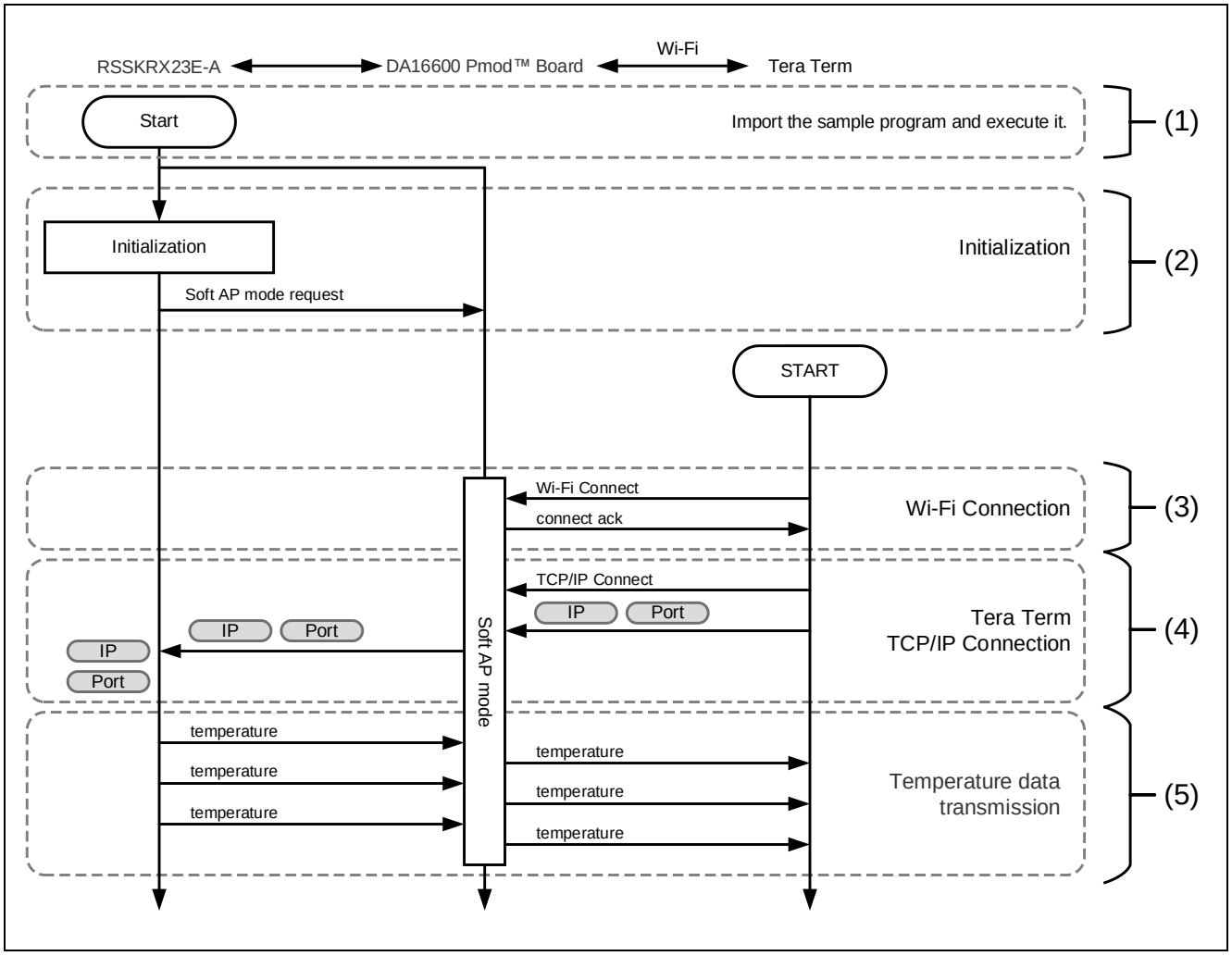

Figure 3-19 Wi-Fi Demonstration Flowchart

3.2.8 Sample Program Operation Details

Follow these procedures to perform the demonstration.

Note that DA16600 Pmod™ Board retains some settings in NVRAM. Therefore, if you have previously run a different demo, the previous demo operation may continue when executing this demo. Performing several restarts in the debugger can write the conditions of this demo to DA16600's NVRAM, allowing it to operate correctly. If the demo does not work well even after multiple restarts in the debugger, please perform a Factory Reset of DA16600 Pmod™ Board.

(1) Steps from Importing to Executing the Sample Program
Please follow the '4 Preparing a Project for Execution' and execute the project. When power is supplied from the debugger, LED2 will illuminate.

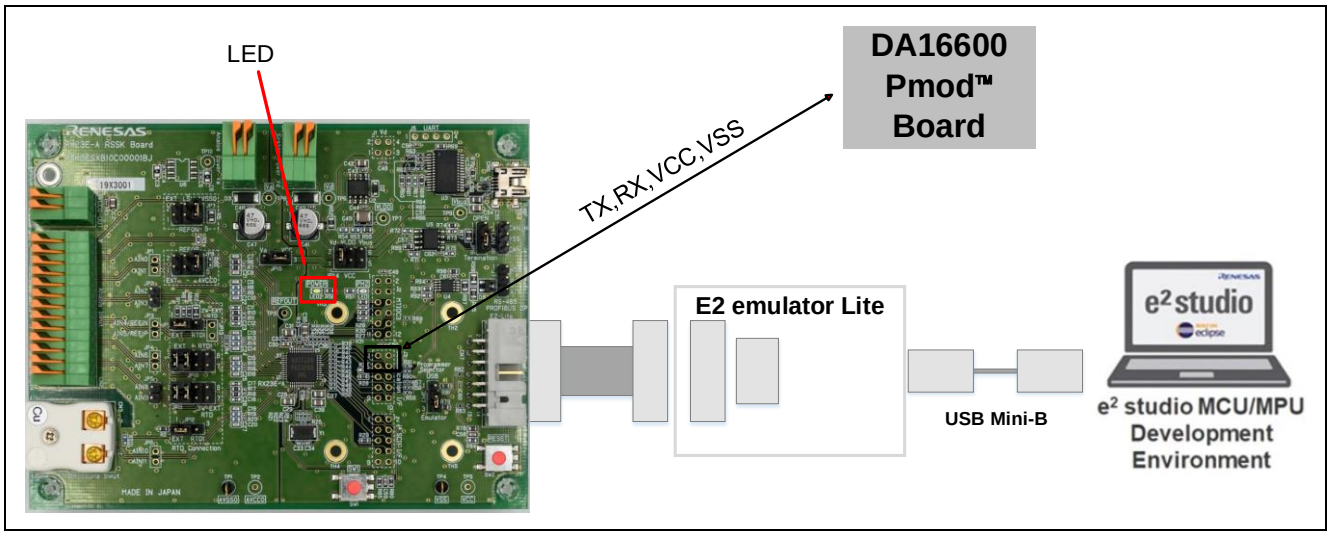

Figure 3-20 LED2 (Power) Position

(2) Initialization
RSSKRX23E-A automatically initializes itself. RSSKRX23E-A also automatically sets the DA16600 Pmod™ Board to Soft AP mode. The logs will be output to the Renesas Debug Virtual Console.

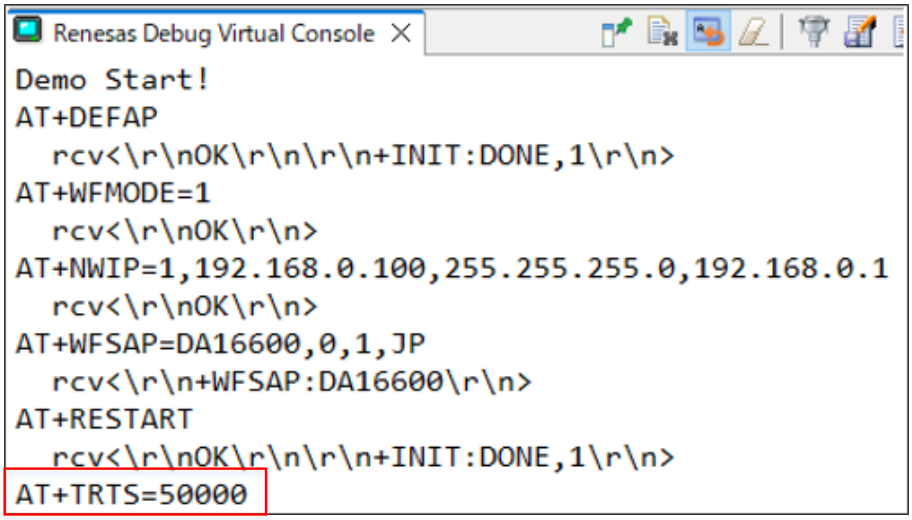

Figure 3-21 Soft AP mode

(3) Wi-Fi Connection
- Wait for 'T+TRTS=50000' to be displayed on the Renesas Debug Virtual Console.
- On the PC, select "Settings" → "Network and Internet" → "Wi-Fi".
- When Wi-Fi is set to "On," available Wi-Fi is searched.
- Select "SSID of DA16600" in the Wi-Fi searched.

Note: Before connecting to the network, you may be require entering the network password or agreement to the Terms of Use.

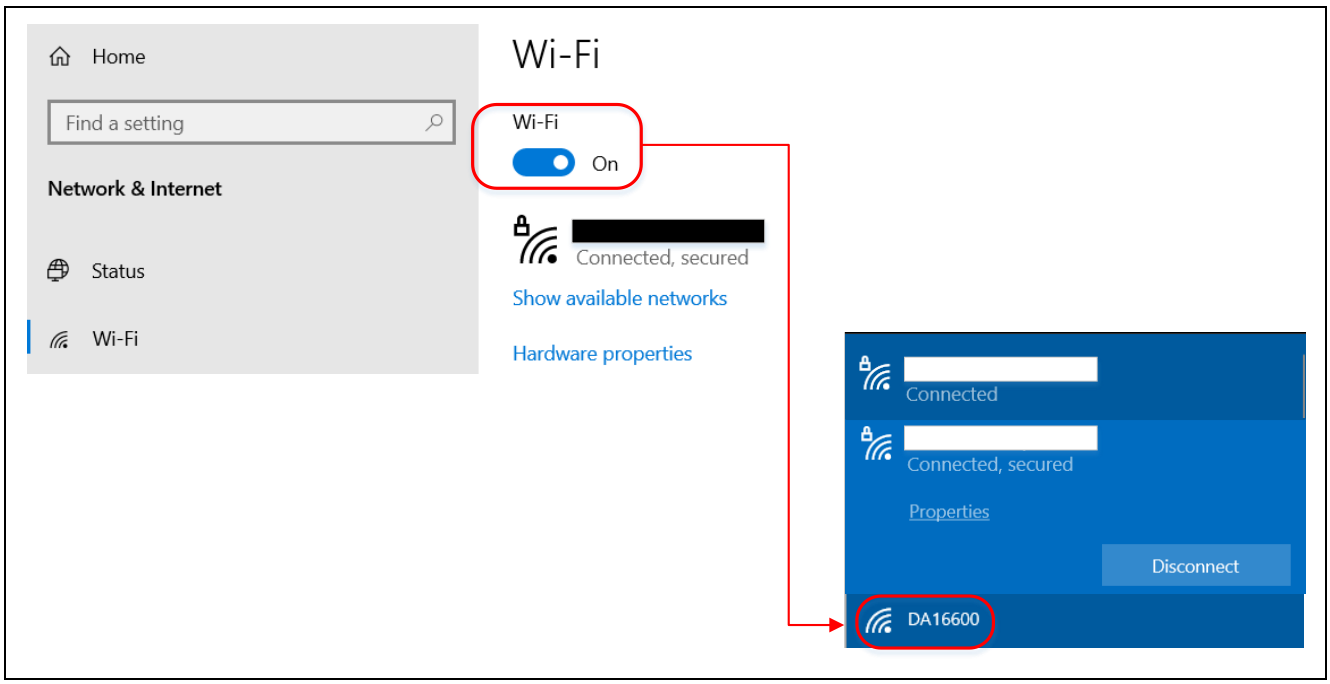

Figure 3-22 Wi-Fi Connection

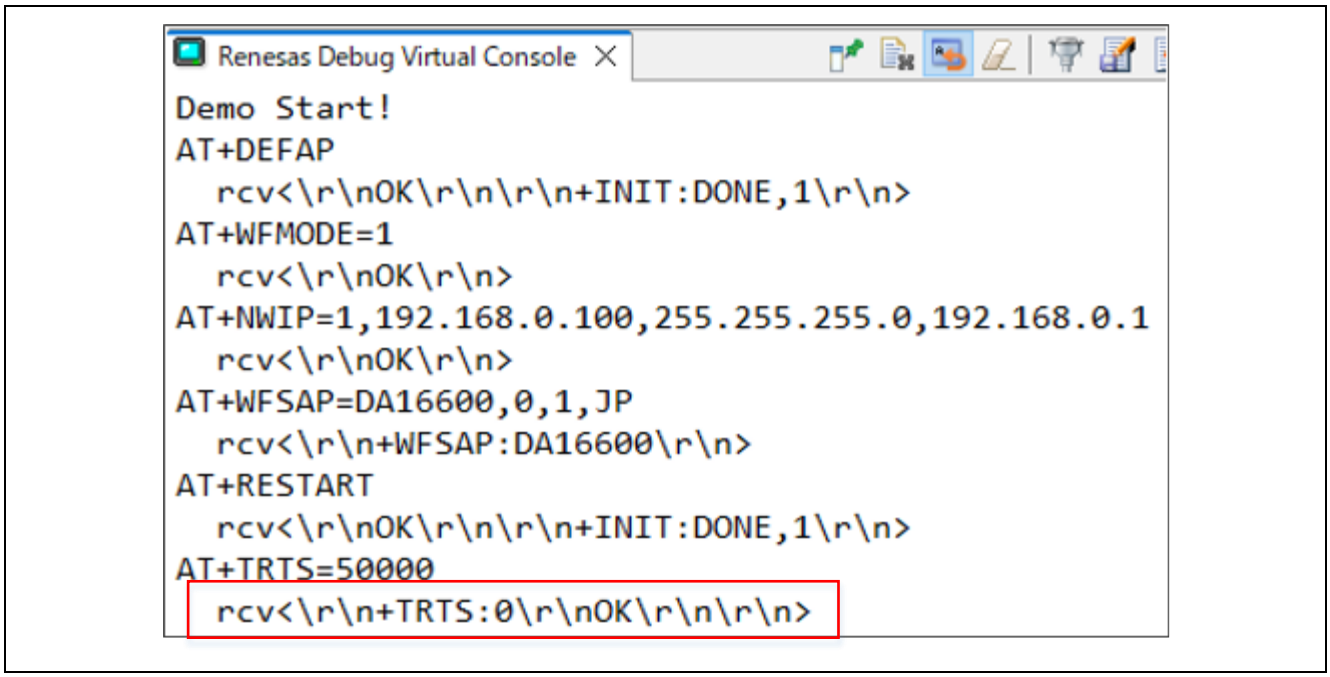

Figure 3-23 Wi-Fi Connection

(4) Tera Term TCP/IP Connection
After connecting the DA16600 Pmod™ Board to Wi-Fi, open Tera Term and start TCP communication with the following settings.

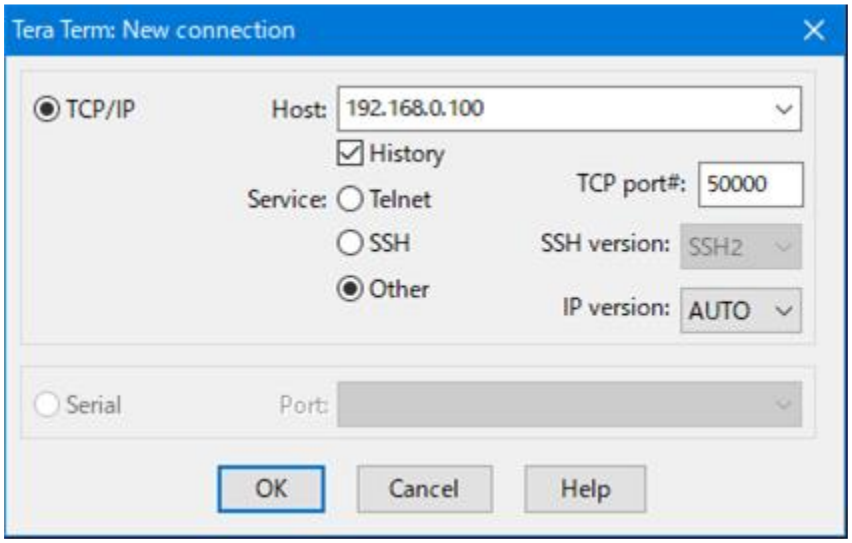

Figure 3-24 Tera Term Settings

(5) Temperature data transmission
After the TCP/IP connection is established, the temperature (unit is °C) measured by the RSSKRX23E-A is displayed as shown in the following.

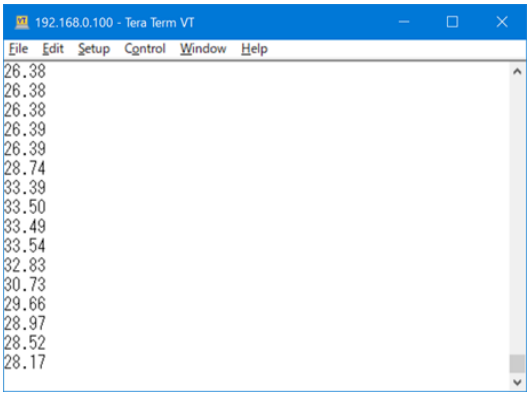

Figure 3-25 How to check temperature data in Tera Term

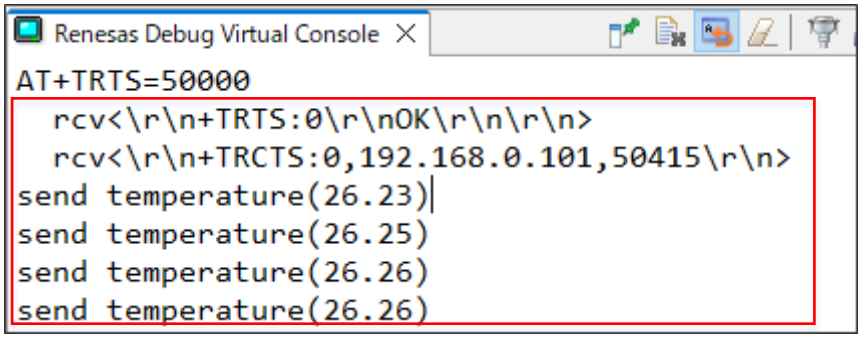

Figure 3-26 Example of Renesas Debug Virtual Console Log

3.3 AWS Demonstration Project (r01an6677_rx23ea_aws)

Connect RSSKRX23E-A to the DA16600 Pmod™ Board for the Wi-Fi demonstration.

The project to perform AWS demonstration is r01an6677_rx23ea_aws. To execute this project, hardware modification is necessary. If you wish to proceed with the project, please proceed to section 3.3.5Hardware Preparation.

In order to transmit data to AWS, advance preparation with AWS is required. AWS accounts must be prepared by customers themselves. After preparing AWS accounts, several procedures are required, such as "registering a Thing" and "issuing a Certificate". Refer to "3.3.6 AWS Preparation" for details on these procedures.

For details on "registering a Thing" and "issuing a Certificate", refer to "3.3 Network Environment and Certificate Issuance" in "Failure Detection and Movement Analysis Demonstration Using AWS Cloud and FFT".

3.3.1 System Structure

The following shows the system structure of this sample program.

For the connection of the RSSKRX23E-A Board, refer to Page4 Figure 4-1 of the "RX23E-A Group Temperature Measurement Example Using a Thermocouple" Application Note (R01AN4747).

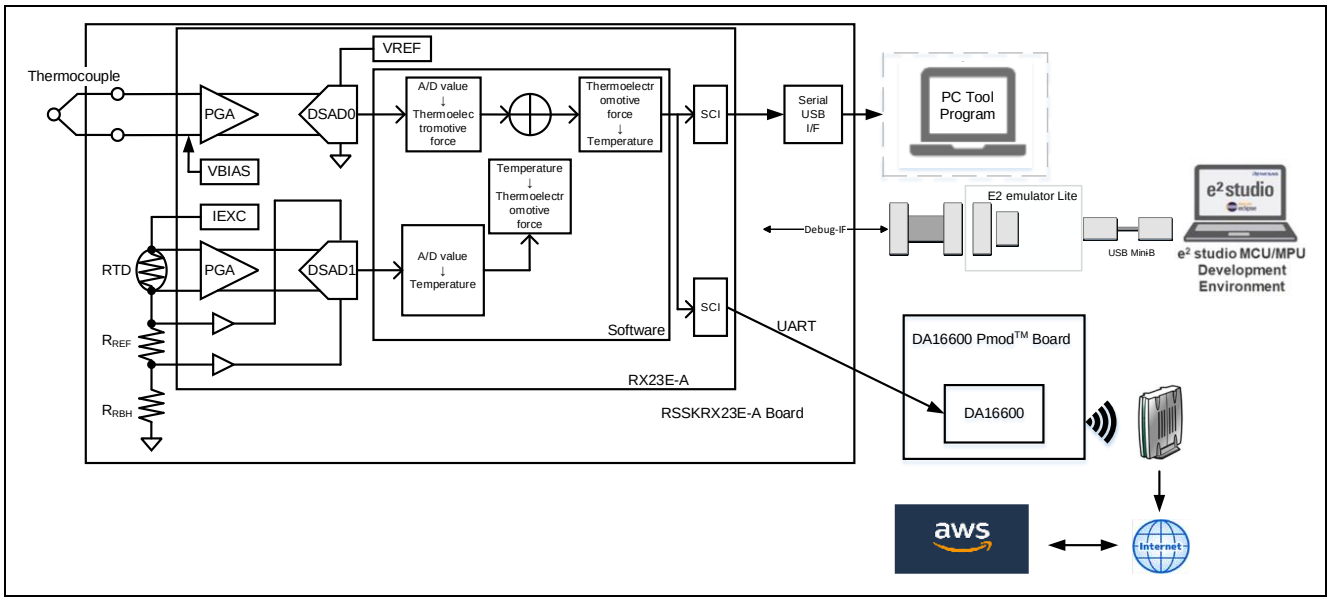

Figure 3-27 System Structure for AWS Demonstration

Note: This PC tool is not required for this operation, but it can be used if needed.

3.3.2 Software Structure

The following shows the software structure of this sample program. The blue part of the RSSKRX23E-A Board is the unchanged part from the original sample program. All communication with Wi-Fi and AWS is performed by the DA16600 Pmod™ Board.

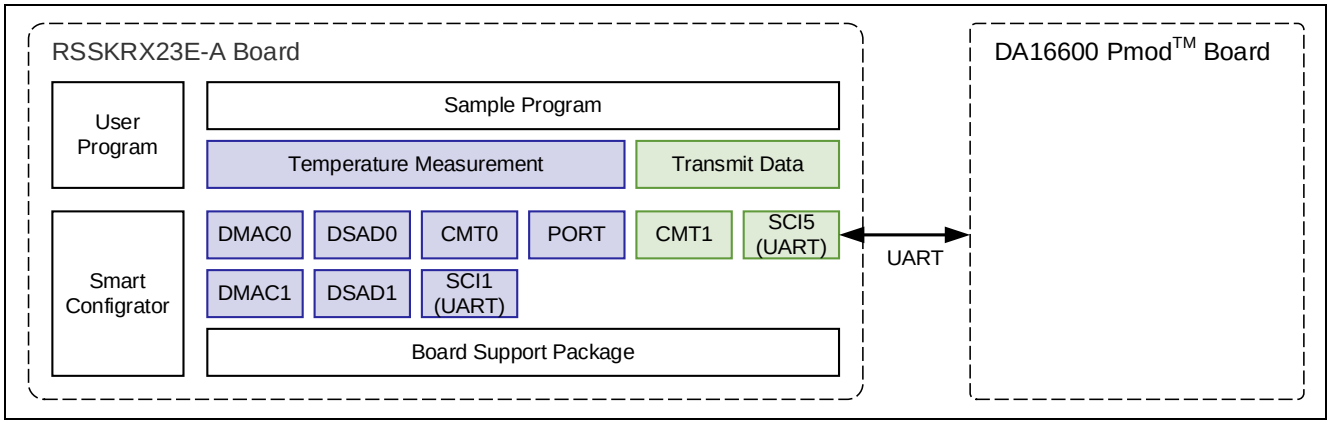

Figure 3-28 Software Structure for AWS Demonstration

3.3.3 Overview Flowchart

The following is an overview flowchart of this sample program.

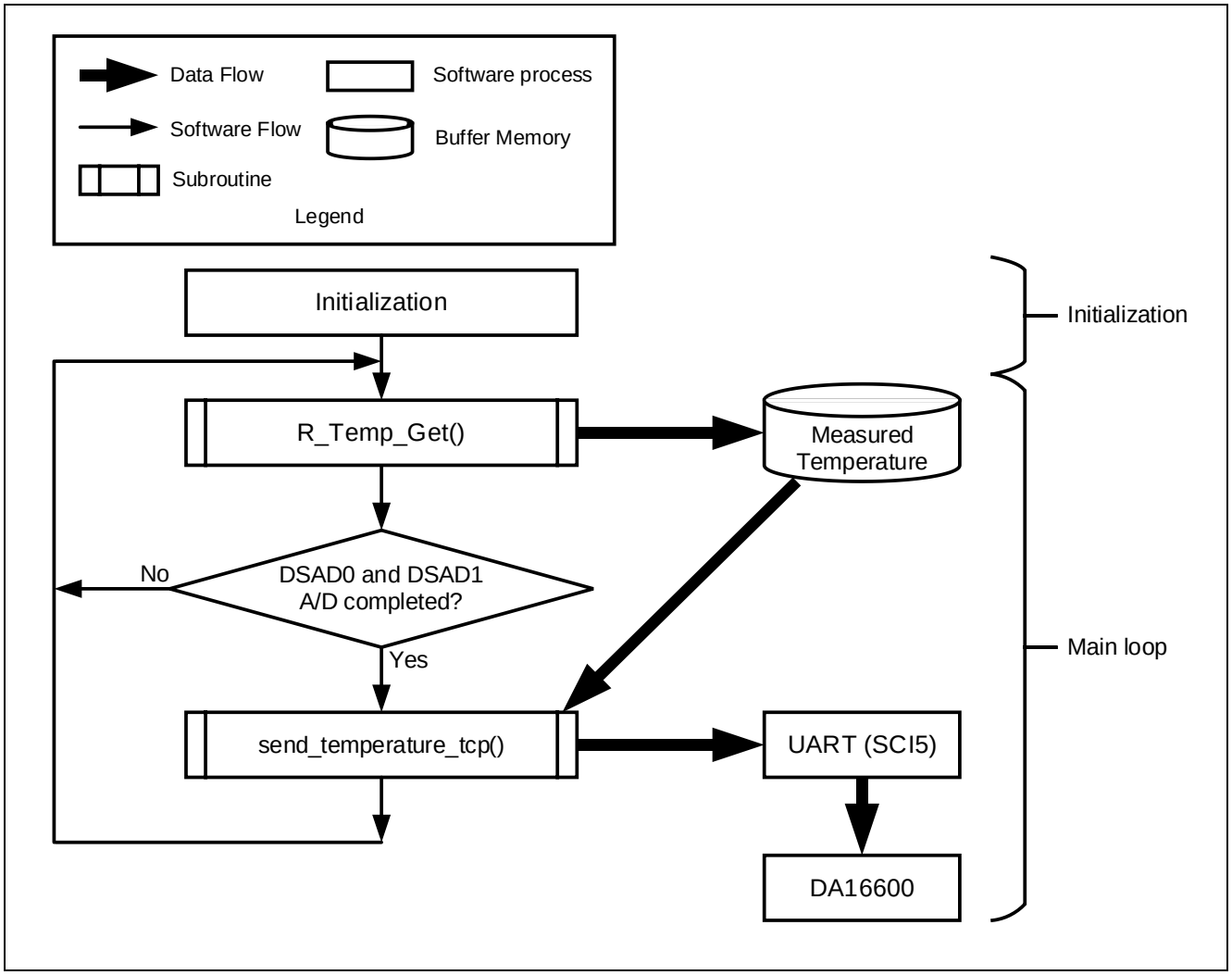

Figure 3-29 Overview Flowchart

3.3.4 Sample Program Structure

3.3.4.1 Pins Used

The following is a list of pins used on RX23E-A in this sample program.

Table 3-20 List of Pins and Functions

Pin name	Input / Output	Functions
PH2	Output	LED1 lighting control
P26/TXD1	Output	UART1 transmit pin
P30/RXD1	Input	UART1 receive pin
P31/CTS1#	Input	CTS signal input pin
AIN11	Input	Thermocouple + side input pin
AIN10	Input	Thermocouple - side input pin
AIN9	Output	RTD excitation current output pin
AIN7	Input	RTD + side input pin
AIN6	Input	RTD - side input pin
AIN5/REF1P	Input	RTD measurement DSAD + side reference voltage
AIN4/REF1N	Input	RTD measurement DSAD - side reference voltage
PH1/TXD5Note1	Output	Connect to TXD on DA16600 Pmod™ Board
PH0/RXD5Note1	Input	Connect to RXD on DA16600 Pmod™ Board
VCCNote1	-	Supply 3.3V to DA16600 Pmod™ Board
VSSNote1	-	Connect to VSS on DA16600 Pmod™ Board

Note1. Pins added from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.3.4.2 Peripheral Functions Used

The following shows lists peripheral functions used by sample program.

Table 3-21 List of Peripheral Functions Used and Functions

Peripheral functions	Functions	Addition
AFE、DSAD0、DSAD1	Driving thermocouples and RTDs (AFE), A/D conversion of thermocouples (DSAD0), A/D conversion of RTDs (DSAD1)	-
SCI1	UART communication with PC tool programs	-
DMAC0	Data transfer triggered by SCI1 receive completion interrupt	-
DMAC3	Data transfer triggered by SCI1 buffer empty interrupt	-
CMT0	Communication timeout detection for SCI	-
PH2	LED1 lighting control	-
SCI5Note1	UART communication with DA16600 Pmod™ Board	yes
CMT1Note1	Interval control of temperature data transmission	yes

Note1. Peripheral functions added from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.3.4.3 Peripheral Function Settings

The peripheral function settings used in this sample program are based on the code generation function of the Smart Configurator. The following are the setting conditions for Smart Configurator. The following describes the peripheral functions added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747).

Table 3-22 SCI5 Settings

Item	Settings
Serial communication method	Start/Stop Synchronization
Start bit detection setting	Low level on RXD5 pin
Data bit length	8bit
Parity setting	Disabled
Stop bit setting	1bit
Data transfer direction setting	LSB First
Transfer rate setting	• Transfer clock : Internal clock • Bit rate : 115200bps • Bit rate modulation Function enabled • SCK5 pin function: SCK5 is disabled
Noise filter setting	Noise filter disabled
Hardware flow control setting	Hardware flow control setting : Disabled
Data processing setting	Transmit data processing : processed by interrupt service routine Received data processing : processed by interrupt service routine
Interrupt setting	Receive error interrupt enabled Priority : Level 15
Callback function setting	Disabled
Input / output pins	• Output : TXD5 (PH1) • Input : RXD5 (PH0)

Table 3-23 CMT1 Settings

Item	Settings
Clock setting	PCLKB/512
Compare match setting	• Interval time : 10 ms • Enable compare match interrupt (CMI1) • Priority: Level 15 (interrupt disabled)

3.3.4.4 File Structure

The following shows the file structure of this sample program.

Table 3-24 File Structure

Folder name, File name	Description
src	Folder for storing program
├ main.c[Note1]	Main process
├ r_ring_buffer_control_api.c	Ring buffer control program
├ r_ring_buffer_control_api.h	Ring buffer control API definition
├ r_sensor_common_api.c	Table search, linear interpolation process program
├ r_sensor_common_api.h	Table search, linear interpolation process API definition
├ r_thermocouple_api.c	Thermocouple measurement calculation program, temperature vs. thermoelectromotive force table
├ r_thermocouple_api.h	Thermocouple measurement calculation API definition
├ r_rtd_api.c	Resistance temperature detector measurement calculation program, temperature vs. resistance value table
├ r_rtd_api.h	Resistance temperature detector measurement calculation API definition
├ r_communication_control_api.c	Communication control program
├ r_communication_control_api.h	Communication control API definition
├ r_mqtt_config.c[Note1]	MQTT communication setting information definition
├ r_mqtt_config.h[Note1]	MQTT communication setting information variable declaration
├ string_func.c[Note1]	AT command control program
├ string_func.h[Note1]	AT command control API definition
└ smc_gen	Smart Configurator generation
├ Config_AFE	
├ Config_CMT0	
├ Config_CMT1[Note1]	
├ Config_DMAC0	
├ Config_DMAC3	
├ Config_DSAD0	
├ Config_DSAD1	
├ Config_PORT	
├ Config_SCI1	
├ Config_SCI5[Note1]	
├ general	
├ r_bsp	
├ r_config	
└ r_pincfg	

Note1. The file with additions and changes from the base "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747)

3.3.4.5 Variables

The following shows the variables that are used in this sample program.

The following describes the variables added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747). For variables other than the ones added, please refer to R01AN4747.

Table 3-25 List of variables in sample code

Variable name	Type	Contents
g_temp	volatile float	Temperature data
g_rcv_end_flg	volatile uint8_t	DA16600 Pmod™ Board command response receive flag
g_send_flg	volatile uint8_t	Temperature data transmission completion flag
g_rcv_buf	uint8_t	Buffer to store received data
g_mqtt_root_certificate_pem	uint8_t	Root certificate
g_mqtt_certificate_pem_cert	uint8_t	Code signing certificate
g_mqtt_private_pem_key	uint8_t	Private key
g_mqtt_broker_endpoint	uint8_t	AWS endpoint
g_mqtt_broker_port	uint8_t	MQTT broker port number
g_mqtt_subscriber	uint8_t	AWS thing name
g_mqtt_publisher	uint8_t	MQTT topic
g_wifi_ssid	uint8_t	SSID for access point
g_wifi_password	uint8_t	Password for access point
g_root_certificate_pem_size	uint16_t	Root certificate characters
g_certificate_pem_cert_size	uint16_t	Number of characters for code signing certificate
g_private_pem_key_size	uint16_t	Characters for private key
g_broker_endpoint_size	uint16_t	Number of AWS endpoint characters
g_broker_port_size	uint16_t	Number of characters for MQTT broker port number
g_subscriber_size	uint16_t	AWS thing name number of characters
g_publisher_size	uint16_t	MQTT topic number of characters
g_wifi_size	uint16_t	Characters of access point SSID
g_password_size	uint16_t	Number of access point password characters

3.3.4.6 Constants

There are no constants added from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747). For the list of constants, please refer to R01AN4747.

3.3.4.7 Functions

The following shows a list of functions used in this sample program.

The following describes the functions added and changed from the based "RX23E-A Group Temperature Measurement Example Using a Thermocouple" application note (R01AN4747).

Table 3-26 List of functions used in the sample code

Function name	Outline	
main	Main process	Modification
start_mqtt_mode	Set DA16600 Pmod™ Board to enable MQTT communication	Addition
check_sci_rcv_end	Detect DA16600 Pmod™ Board response	Addition
check_rcv_cmd	Check contents received from DA16600 Pmod™ Board	Addition
reset_rcv_buf	Clear receive buffer	Addition
send_temperature_mqtt	Transmit temperature data	Addition

3.3.4.8 Function Specifications

The following shows function specifications that are used in this sample program.

[Function name]main

Outline	Main process
Header	None
Declaration	void main (void)
Description	Initialize peripheral functions. Measure temperature using a thermocouple, control the DA16600 Pmod™ Board, and perform transmission of temperature data.
Arguments	None
Return Value	None
Remarks	None

[Function name] start_mqtt_mode

Outline	Set DA16600 Pmod™ Board to enable MQTT communication
Header	string_func.h
Declaration	void start_mqtt_mode (void)
Description	Transmit AT commands to the DA16600 Pmod™ Board. Set the DA16600 Pmod™ Board to STA mode to enable MQTT communication.
Arguments	None
Return Value	None
Remarks	None

[Function name] check_sci_rcv_end

Outline	Detect DA16600 Pmod™ Board response
Header	string_func.h
Declaration	void check_sci_rcv_end (void)
Description	Detects DA16600 Pmod™ Board responses from line feed codes.
Arguments	None
Return Values	None
Remarks	None

[Function name] reset_rcv_buf

Outline	Clear receive buffer
Header	string_func.h
Declaration	void reset_rcv_buf (void)
Description	Clear receive buffer
Arguments	None
Return Value	None
Remarks	None

[Function name] send_temperature_mqtt

Outline	Transmit temperature data
Header	string_func.h
Declaration	void send_temperature_tcp (void)
Description	Set the command for transmission and temperature data in the transmission buffer and transmit to the DA16600 Pmod™ Board.
Arguments	None
Return Value	None
Remarks	None

3.3.4.9 AWS Demonstration Flowchart

The following shows the main function flowchart for the AWS demonstration.

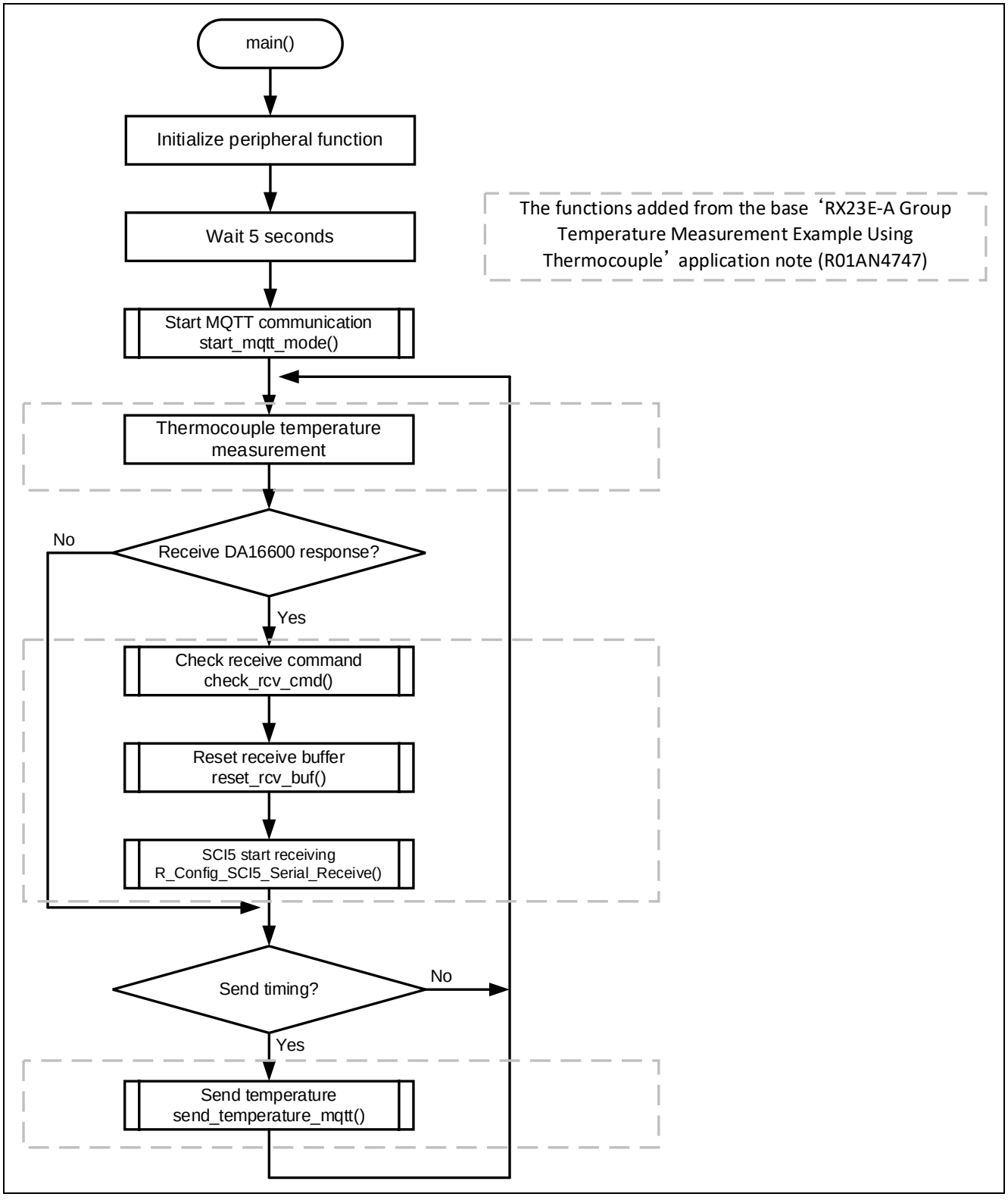

Figure 3-30 main function Flowchart

The following shows start_mqtt_mode function flowchart for the AWS demonstration.

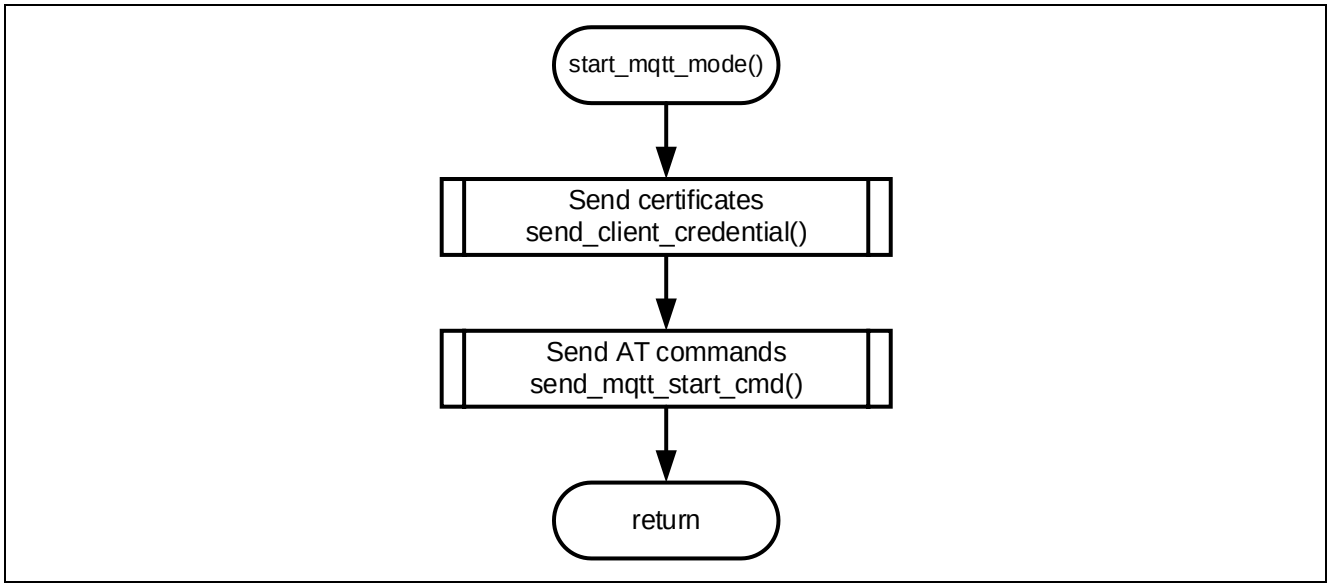

Figure 3-31 start_mqtt_mode function Flowchart

The following shows send_client_credential function flowchart for the AWS demonstration.

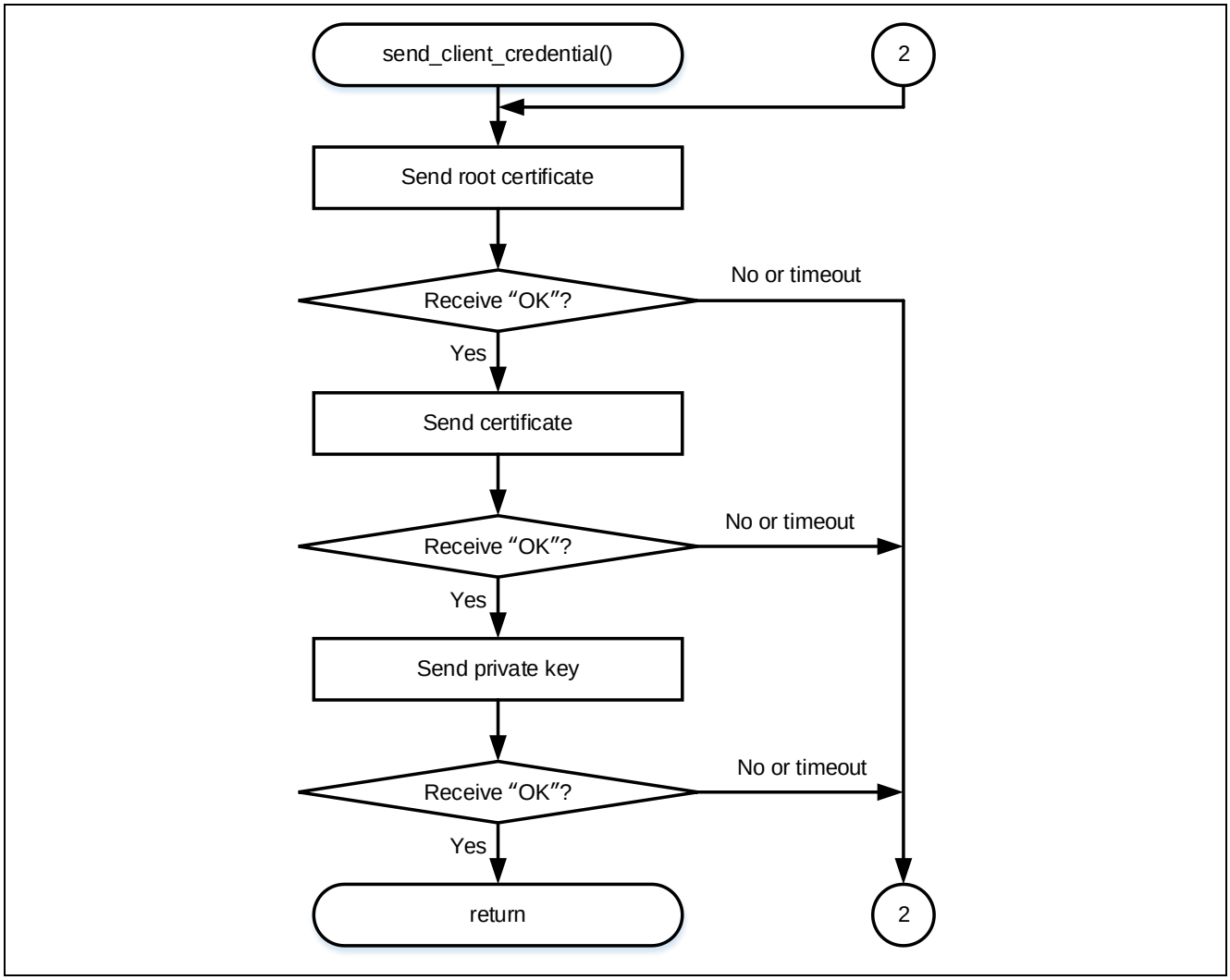

Figure 3-32 send_client_credential function Flowchart

The following shows send_mqtt_start_cmd function flowchart for the AWS demonstration.

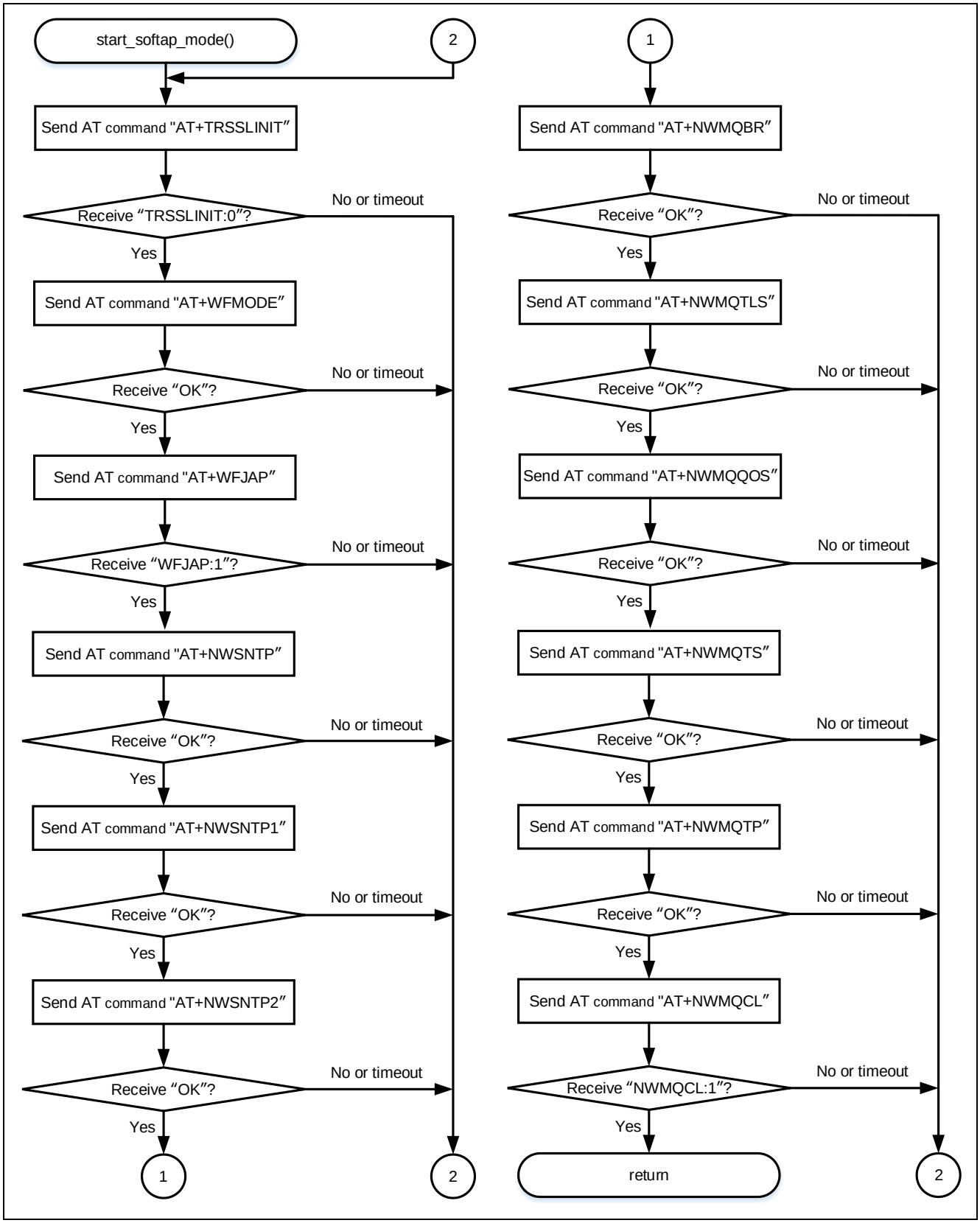

Figure 3-33 send_mqtt_start_cmd function Flowchart

3.3.5 Hardware Preparation

This application note uses the thermocouple measurement circuit on the RSSKRX23E-A board. For usage details, refer to "2.4 Using the Analog Input Circuit" in the "RSSKRX23E-A User's Manual".

To connect RSSKRX23E-A to the DA16600 Pmod™ Board, the RSSKRX23E-A board must be modified.

The following is a list of pins to be modified. For details on pin numbers, refer to the "RSSKRX23E-A User's Manual".

Table 3-27 List of Pins to be modified

Pin number	MCU pin number	Function	Input / Output	Description
1	-	VSS	Output	VSS pin
2	-	VCC	Output	VCC pin Used for external power supply
3	23	PH1/TXD5	Input / Output	PH1/TXD5 pin
4	24	PH0/RXD5	Input / Output	PH0/RXD5 pin

3.3.5.1 Chip resistance removal

To use TXD5 and RXD5, the chip resistors must be removed. The chip resistors to be removed are R91 and R90.

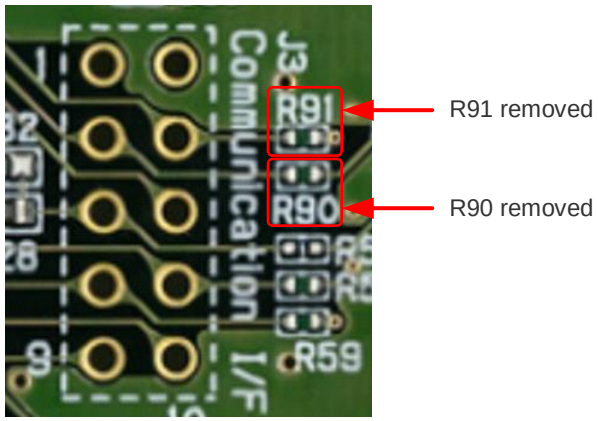

Figure 3-34 Chip resistance removal

3.3.5.2 Pin header implementation

Implement pin headers to use VSS, VCC, TXD5, and RXD5.

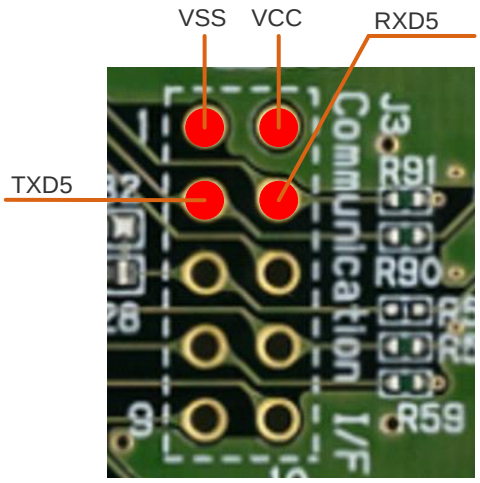

Figure 3-35 Connecting to the DA14531 Pmod™ Board

3.3.5.3 Connecting to RSSKRX23E-A to the DA16600 Pmod™ Board

Connect VSS, VCC, TXD5, RXD5 and DA16600 Pmod™ Board as follows.

Note to connect TXD5 on the RSSKRX23E-A to TXD on the DA16600 Pmod™ Board.

Table 3-28 Connection Table

RSSKRX23E-A		DA16600 Pmod™ Board		Supplement
Pin number	Pin name	Pin number	Signal	
-	-	1	CTS	OPEN
3	PH1/TXD5	2	TXD	
4	PH0/RXD5	3	RXD	
-	-	4	RTS	OPEN
1	VSS	5	GND	
2	VCC	6	VCC	
-	-	7	GPIO	OPEN
-	-	8	GPIO	OPEN
-	-	9	GPIO	OPEN
-	-	10	GPIO	OPEN
-	-	11	GND	OPEN
-	-	12	VCC	OPEN

3.3.6 AWS Preparation

Set up AWS by referring to the following tutorial.

- Resister a device with AWS IoT
  https://github.com/renesas/amazon-freertos/wiki/Register-device-to-AWS-IoT

Note: Proceed to "Check AWS IoT endpoints".

- Enter AWS connection information in source code
  Set up the five variables in {rx23ea_thermocouple_aws/src/r_mqtt_config.c}.

- g_mqtt_broker_endpoint[] ➡ Name of the endpoint checked in "3.3.6 AWS Preparation".
- g_mqtt_suscriber[] ➡ Name of the thing registered in "3.3.6 AWS Preparation".
- g_mqtt_publisher[] ➡ Name of the thing registered in "3.3.6 AWS Preparation"/send
- g_wifi_ssid ➡ SSID of the access point to be connected
- g_wifi_password ➡ Password of the access point to be connected
  (The above variables should be entered in " " as shown in the following figure.)

```
uint8_t  g_mqtt_broker_endpoint[] = [redacted]iot.ap-northeast-1.amazonaws.com";
uint16_t g_broker_endpoint_size   = sizeof(g_mqtt_broker_endpoint);

uint8_t  g_mqtt_subscriber[] = "[redacted]";
uint16_t g_subscriber_size   = sizeof(g_mqtt_subscriber);

uint8_t  g_mqtt_publisher[] = "[redacted]/send";
uint16_t g_publisher_size   = sizeof(g_mqtt_publisher);

uint8_t  g_wifi_ssid[] = "[redacted]";
uint16_t g_wifi_size   = sizeof(g_wifi_ssid);

uint8_t  g_wifi_password[] = "[redacted]";
uint16_t g_password_size   = sizeof(g_wifi_password);
```

Figure 3-36 r_mqtt_config.c

3.3.6.1 Applying Root CA Certificate and Device Certificate and Private Key in Source Code

Apply the three variables in {rx23ea_thermocouple_aws/src/r_mqtt_config.c} as follows.

- g_mqtt_root_cetificate_pem[] ➡Downloaded root CA Certificate according to Figure 3-37
- g_mqtt_cetificate_pem_cert[] ➡Downloaded Device Certificate according to Figure 3-38
- g_mqtt_private_pem_key[] ➡Downloaded Private Key according to Figure 3-39

Note: Please note the following

"\n" is required at the end of each line except the last line
Each line must be enclosed in double quotation marks
The last line of each line except the last line must end with a backslash

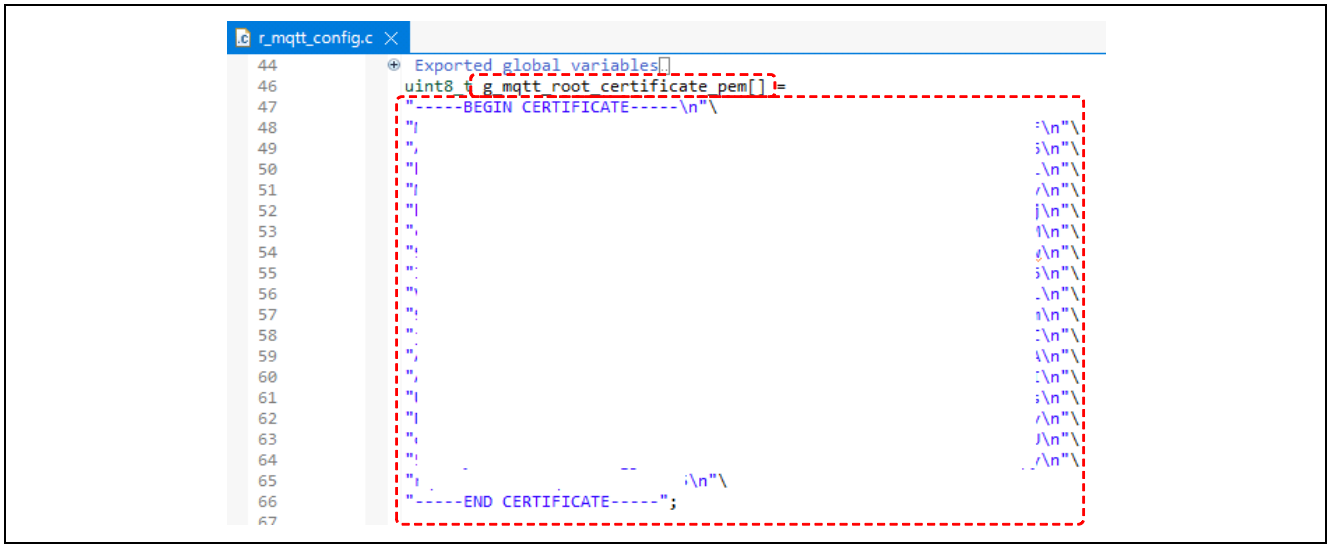

Figure 3-37 Applying Root CA Certificate

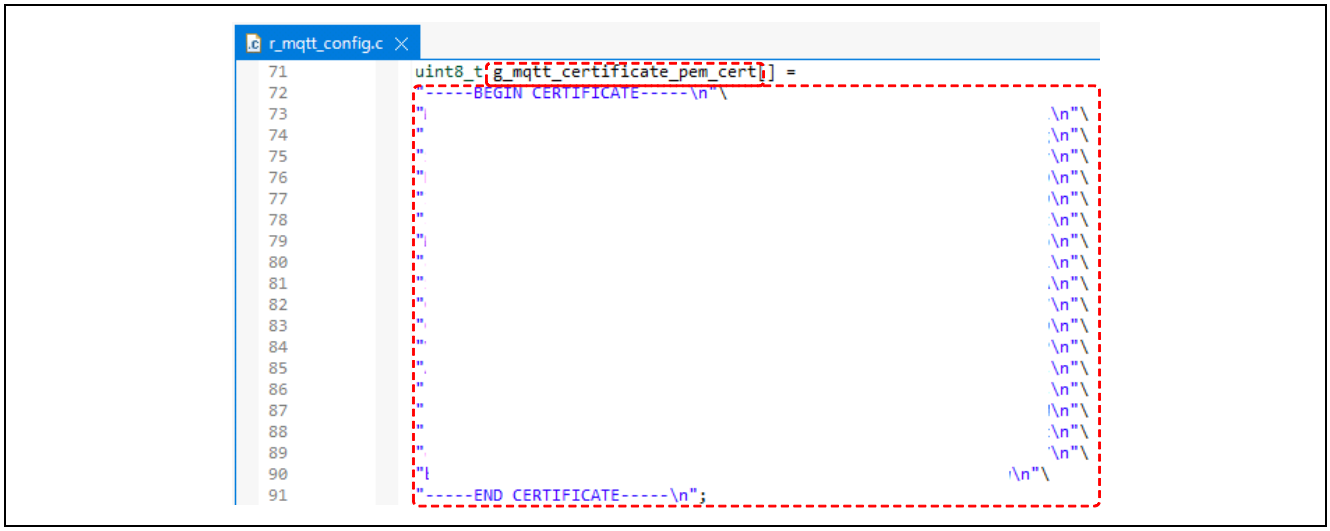

Figure 3-38 Applying Device Certificate

Figure 3-39 Applying Private Key

3.3.7 Sample Program Operation Overview

The following shows an overview of the sample program operation. For detailed procedures to operate the sample program, Refer to "3.3.8 Sample Program Operation Details".

(1) Start
Import the sample program and execute it.
(2) Initialization
RSSKRX23E-A automatically initializes itself.
(3) Storage of Information
RSSKRX23E-A automatically sets the DA16600 Pmod™ Board to STA (Station) mode.
RSSKRX23E-A also automatically writes certificates and private keys to the DA16600 Pmod™ Board.
(4) Wi-Fi Connection
RSSKRX23E-A automatically requests a Wi-Fi connection to the DA16600 Pmod™ Board after writing certificates etc.
(5) AWS Connection
RSSKRX23E-A requests an AWS connection to the DA16600 Pmod™ Board after checking that it is connected to Wi-Fi.
(6) MQTT Connection
RSSKRX23E-A requests an MQTT connection to the DA16600 Pmod™ Board after checking that it is connected to AWS.
(7) Temperature data translation
RSSKRX23E-A transmits temperature data to the DA16600 Pmod™ Board after checking that it is connected to MQTT.

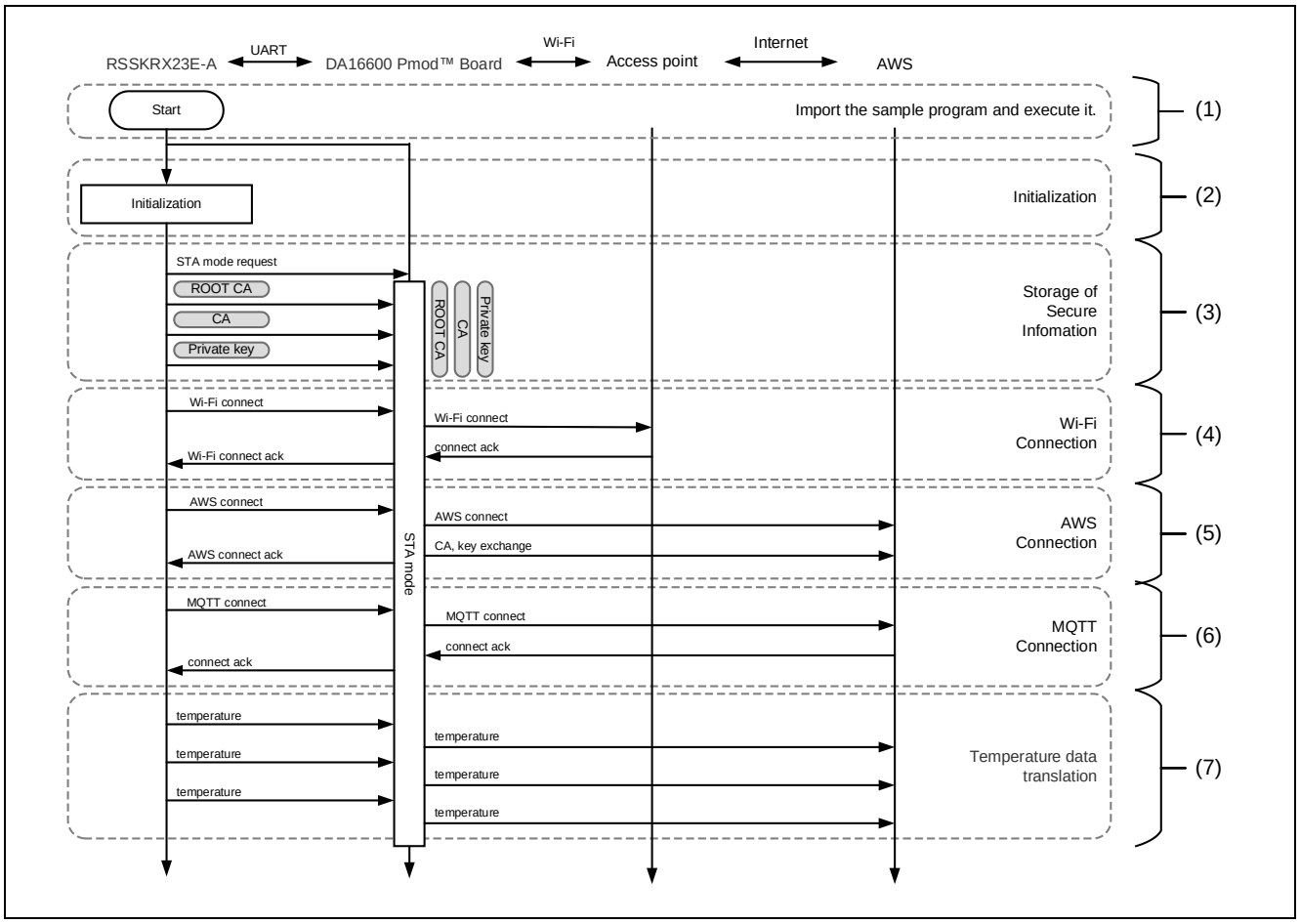

Figure 3-40 AWS Demonstration Flowchart

3.3.8 Sample Program Operation Details

Execute the demonstration by the following steps.

Note that DA16600 Pmod™ Board retains some settings in NVRAM. Therefore, if you have previously run a different demo, the previous demo operation may continue when executing this demo. Performing several restarts in the debugger can write the conditions of this demo to DA16600's NVRAM, allowing it to operate correctly. If the demo does not work well even after multiple restarts in the debugger, please perform a Factory Reset of DA16600 Pmod™ Board.

(1) Steps from Importing to Executing the Sample Program
Please follow the '4 Preparing a Project for Execution' and execute the project. When power is supplied from the debugger, LED2 will illuminate.

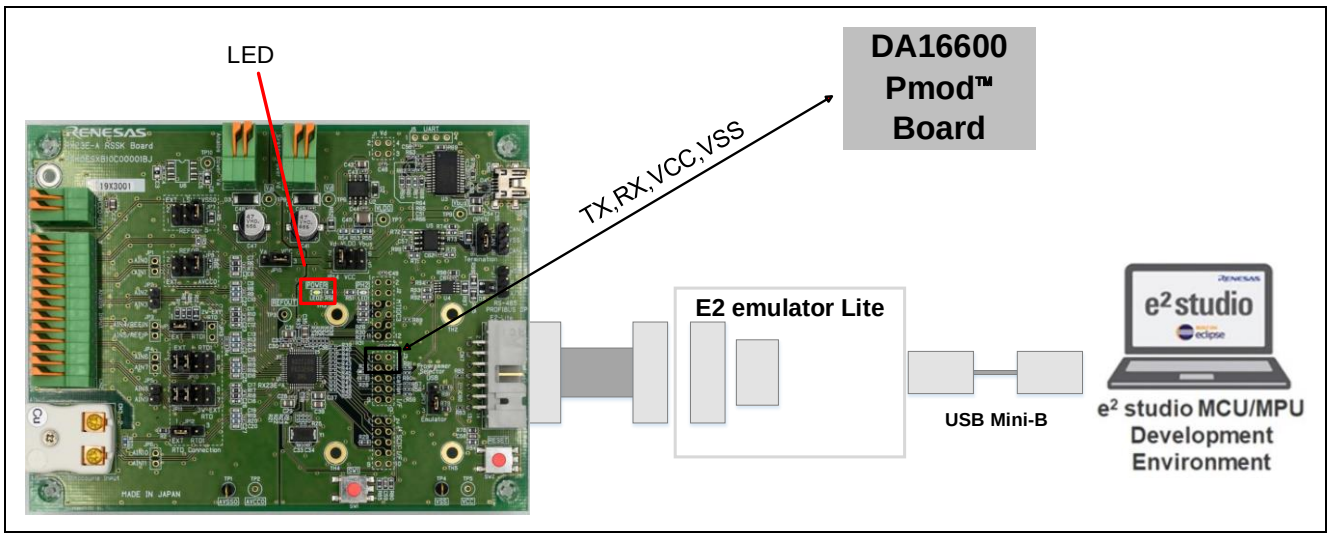

Figure 3-41 LED2 (Power) Position

(2) Initialization
RSSKRX23E-A automatically initializes itself. After this, all operations until the data is transferred to AWS are performed automatically by RSSKRX23E-A.

(3) Storage of Information
RSSKRX23E-A automatically sets the DA16600 Pmod™ Board to STA mode. Then, RSSKRX23E-A automatically writes the certificate and private key to the DA16600 Pmod™ Board.

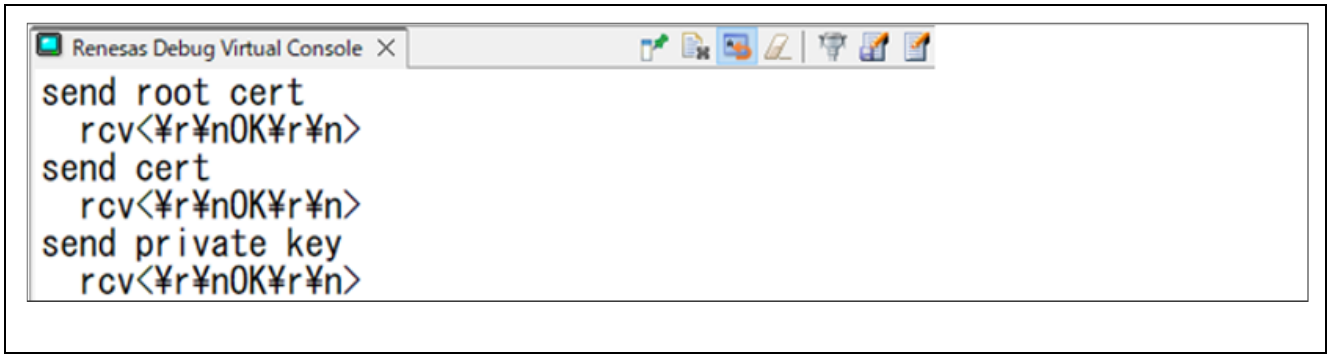

Figure 3-42 Log of Certificates and Private Keys

(4) Wi-Fi Connection
RSSKRX23E-A automatically requests a Wi-Fi connection to the DA16600 Pmod™ Board.

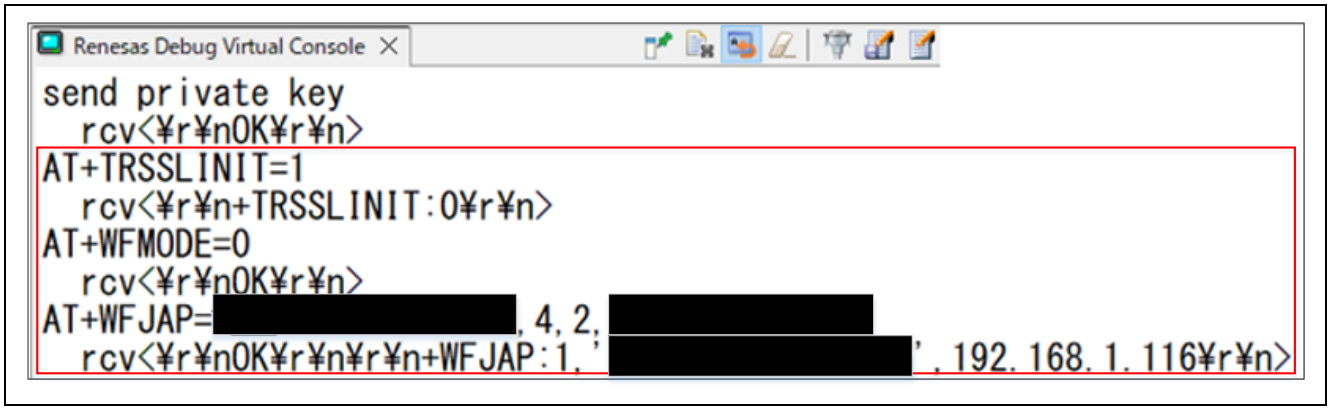

Figure 3-43 Log of Wi-Fi Requests

(5) RSSKRX23E-A automatically requests an AWS connection to the DA16600 Pmod™ Board.
(6) RSSKRX23E-A automatically requests an MQTT connection to the DA16600 Pmod™ Board.

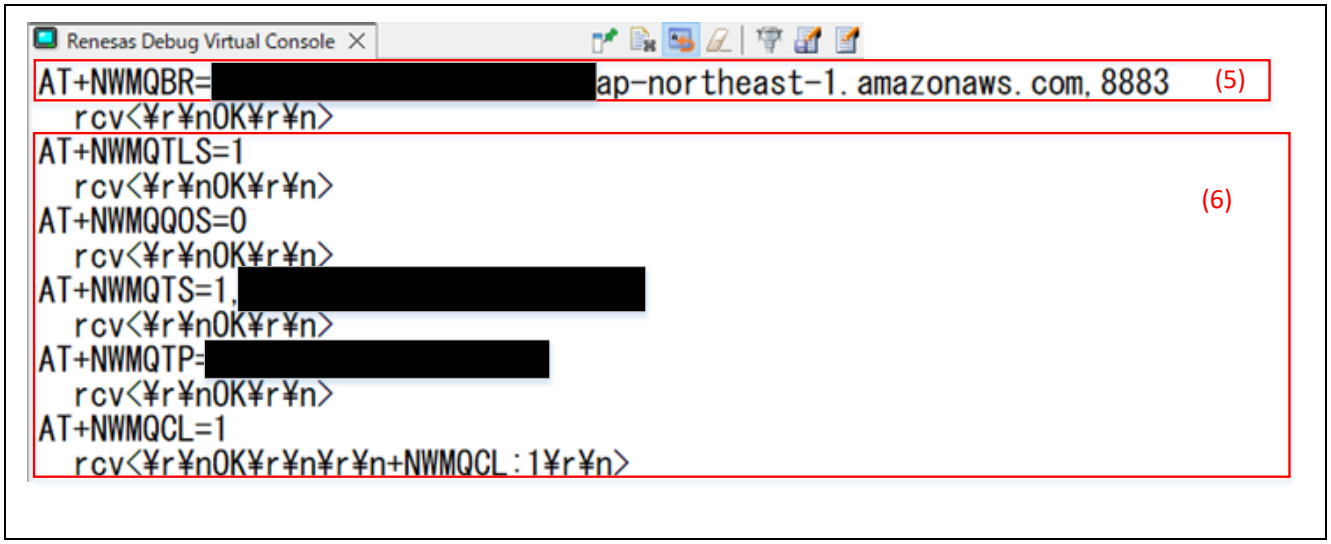

Figure 3-44 Logs of AWS and MQTT

(7) RSSKRX23E-A transmits temperature data to the DA16600 Pmod™ Board after checking that it is connected to MQTT.

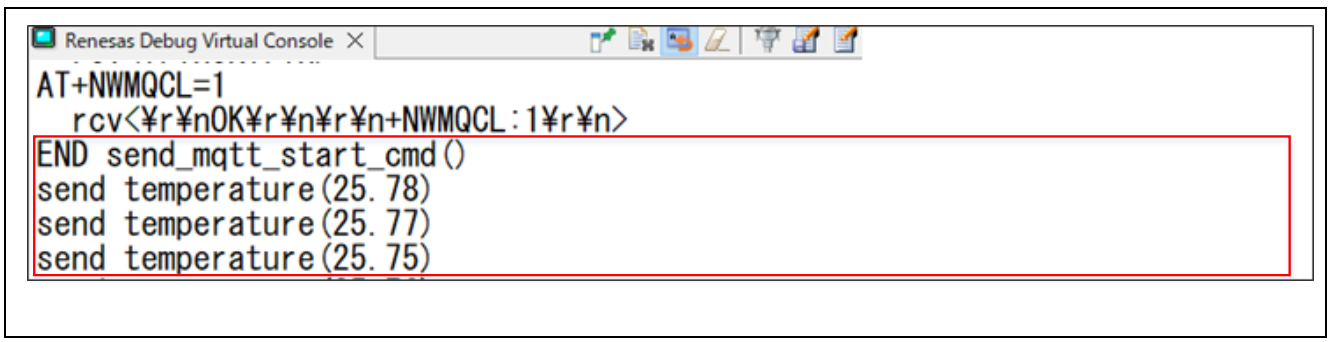

Figure 3-45 Send temperature data to AWS.

3.3.8.1 How to check temperature data in AWS

The following shows how to check temperature data in AWS.

Open the tab "IoT Core" → "Test" → "MQTT Test Client" → "Subscribe to Topic". Enter the name of the thing in the "Filter Topic." "/#" is a wildcard. Click the Subscribe button and the received temperature data is displayed.

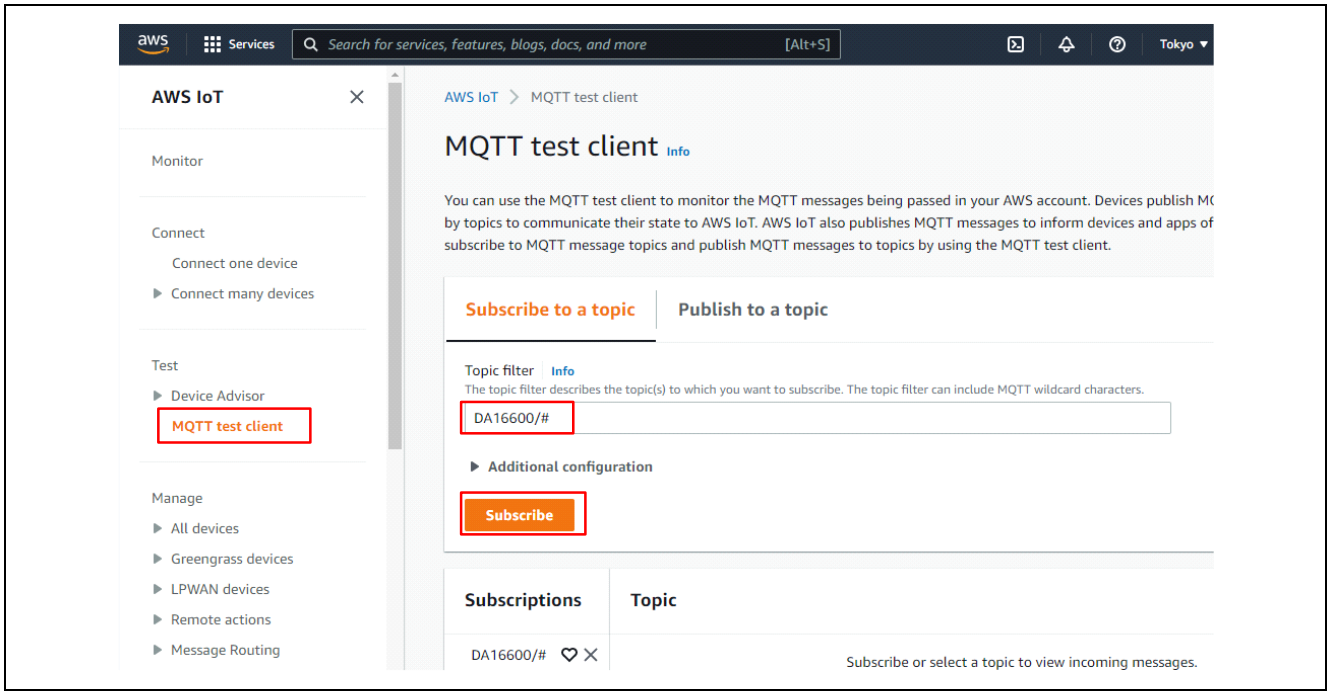

Figure 3-46 How to check temperature data in AWS

4. Preparing a Project for Execution

The sample programs are distributed in e² studio project format. This section shows how to import a project into e² studio or CS+. After importing a project, check the build and debug settings.

4.1 Procedure in e² studio

4.1.1 Importing a Project in e2-studio

To use sample programs in e² studio, follow the steps below to import them into e² studio. In projects managed by e² studio, do not use space codes, multibyte characters, and symbols such as "$", "#", "%" in folder names or paths to them.

(Note that depending on the version of e² studio you are using, the interface may appear somewhat different from the screenshots below.)

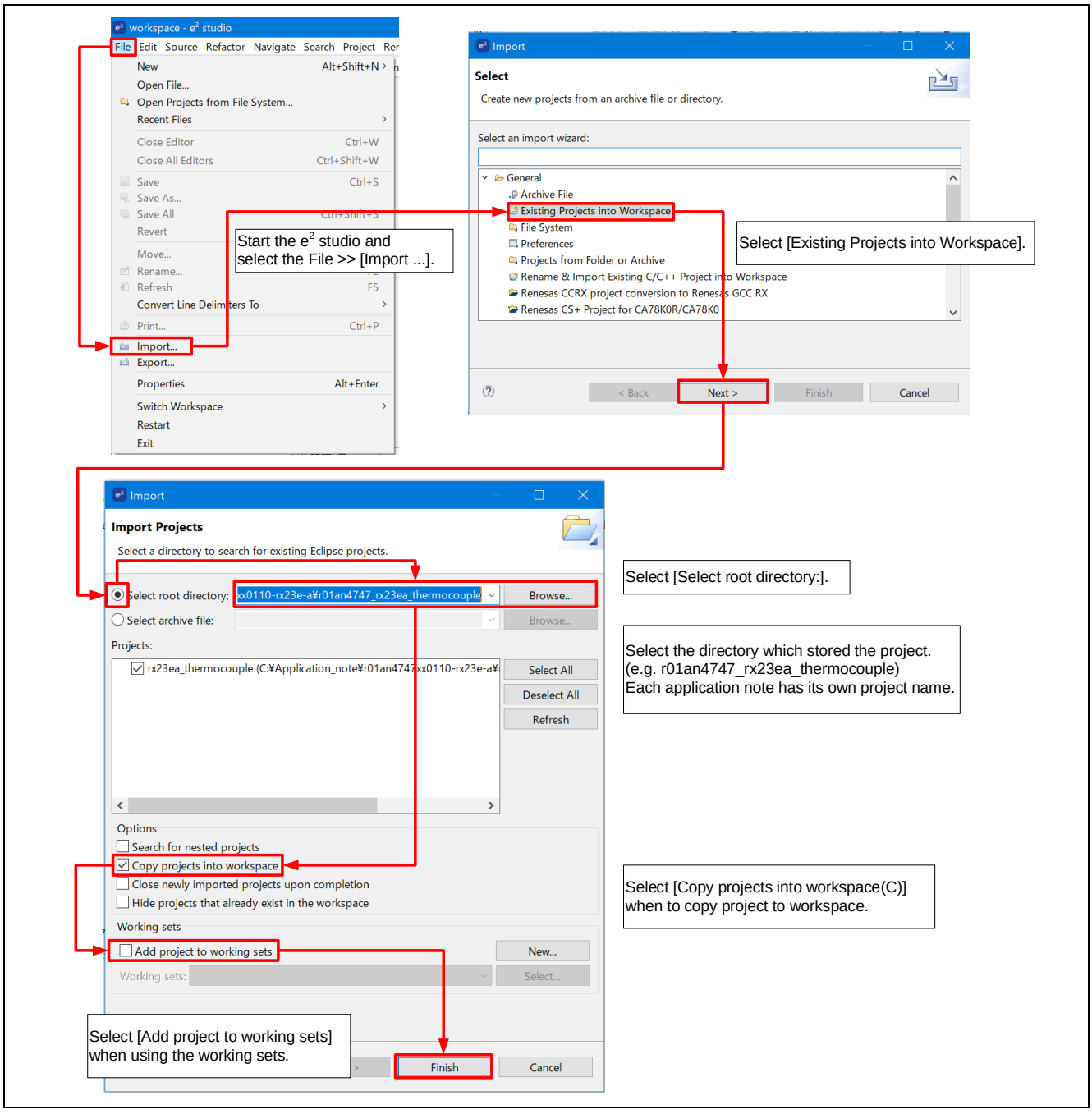

Figure 4-1 Import a Project into e² Studio

4.1.2 Set build options

1. Right-click on the project name and select menu -> [Property].
2. Click [C/C ++ Build] -> [Settings] -> [Toolchain] tab, and check the toolchain and version.

— · Toolchain : Renesas CC-RX

— · Version : v3.05.00

4.1.3 Build the project.

1. Right-click the project in the Project Explorer and select [Build project].
2. The build starts and the console displays the status of the build. When the message “Build completed” is displayed, the build is complete.

4.1.4 Debug

1. Select [Run] -> [Debug Configuration…] to open the [Debug configuration] window.
2. In the [Debug configuration] window, expand the display of the [Renesas GDB Hardware Debugging] debug configuration and click on “Project name(HardwareDebug) ” configuration.
3. Switch to [Debugger] -> [Connection Settings] tab and check that the settings are as shown below.
4. When you click 'Debug,' the program will be downloaded to the RX23E-A.

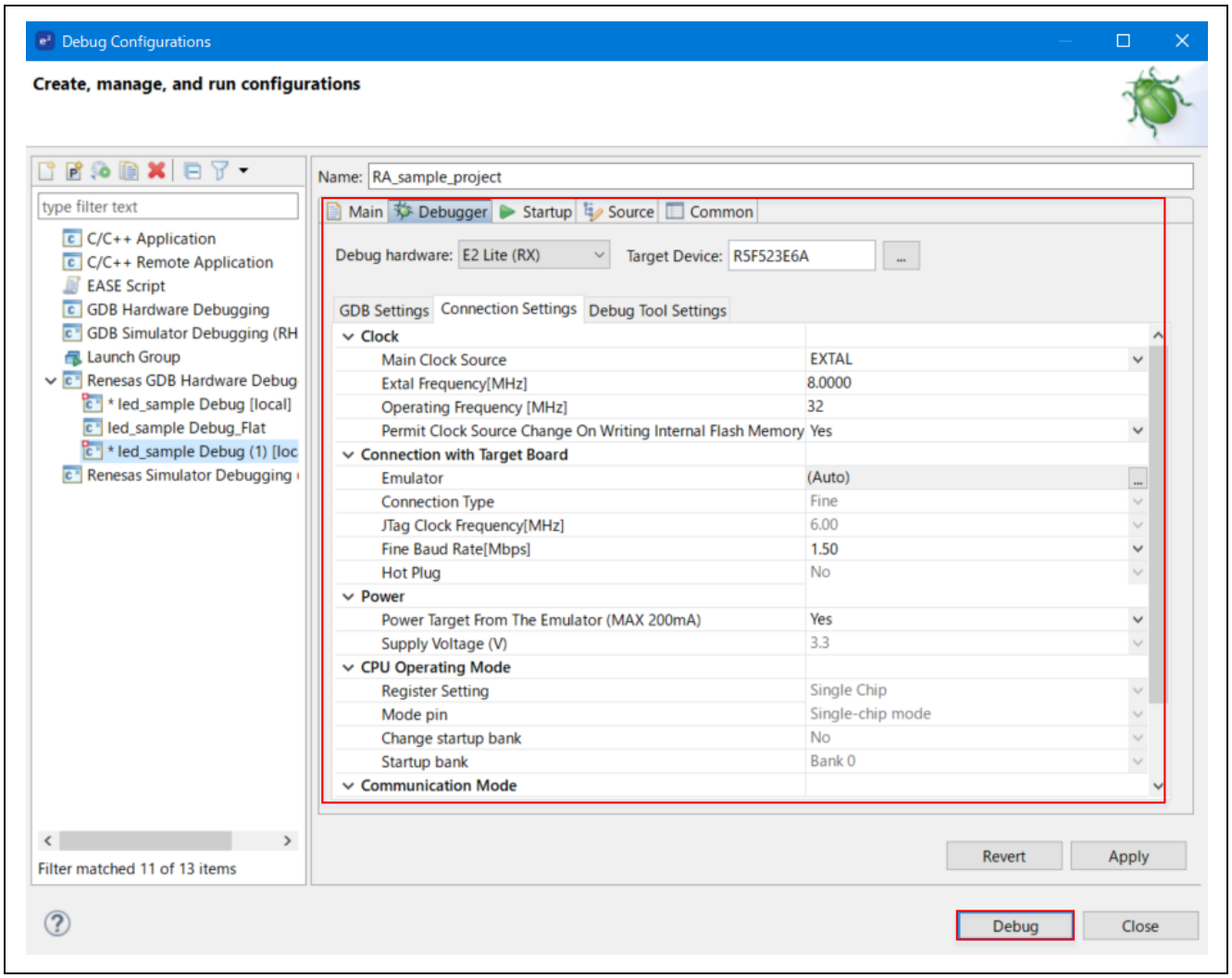

Figure 4-2 Debug screen settings pickup

5.Select [Renesas View] -> [Debug] -> [Renesas Debug Birtual Console].

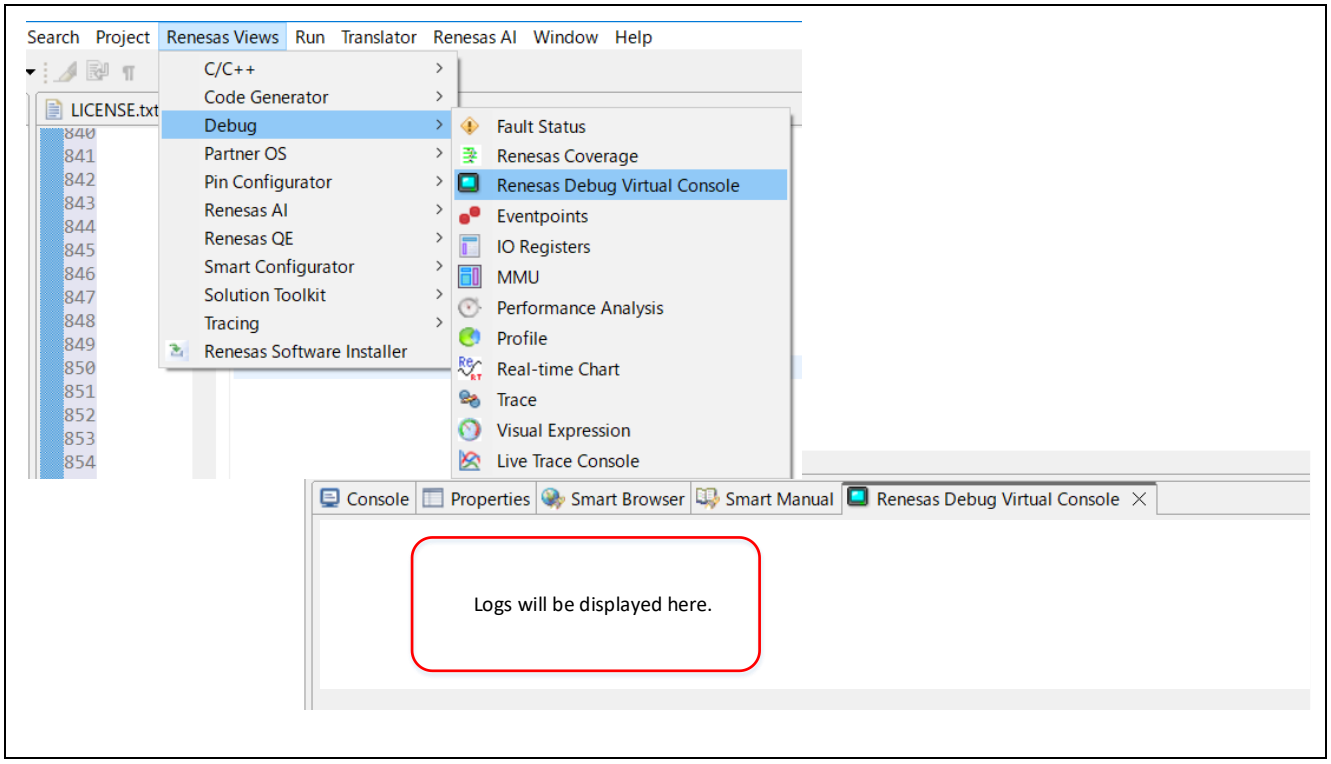

Figure 4-3 Renesas Debug Virtual Console

4.1.5 Run

1. Run the demo project by clicking the button or pressing the "F8" key.

For details on how to operate the debug screen, refer to the following user's manual, section 5.4.

— e2studio User's Manual: Getting Started Guide for V7.0(r20ut4374)

4.2 Procedure in CS+

To use sample programs in CS+, follow the steps below to import them into CS+. In projects managed by CS+, do not use space codes, multibyte characters, and symbols such as "$", "#", "%" in folder names or paths to them.

(Note that depending on the version of CS+ you are using, the interface may appear somewhat different from the screenshots below.)

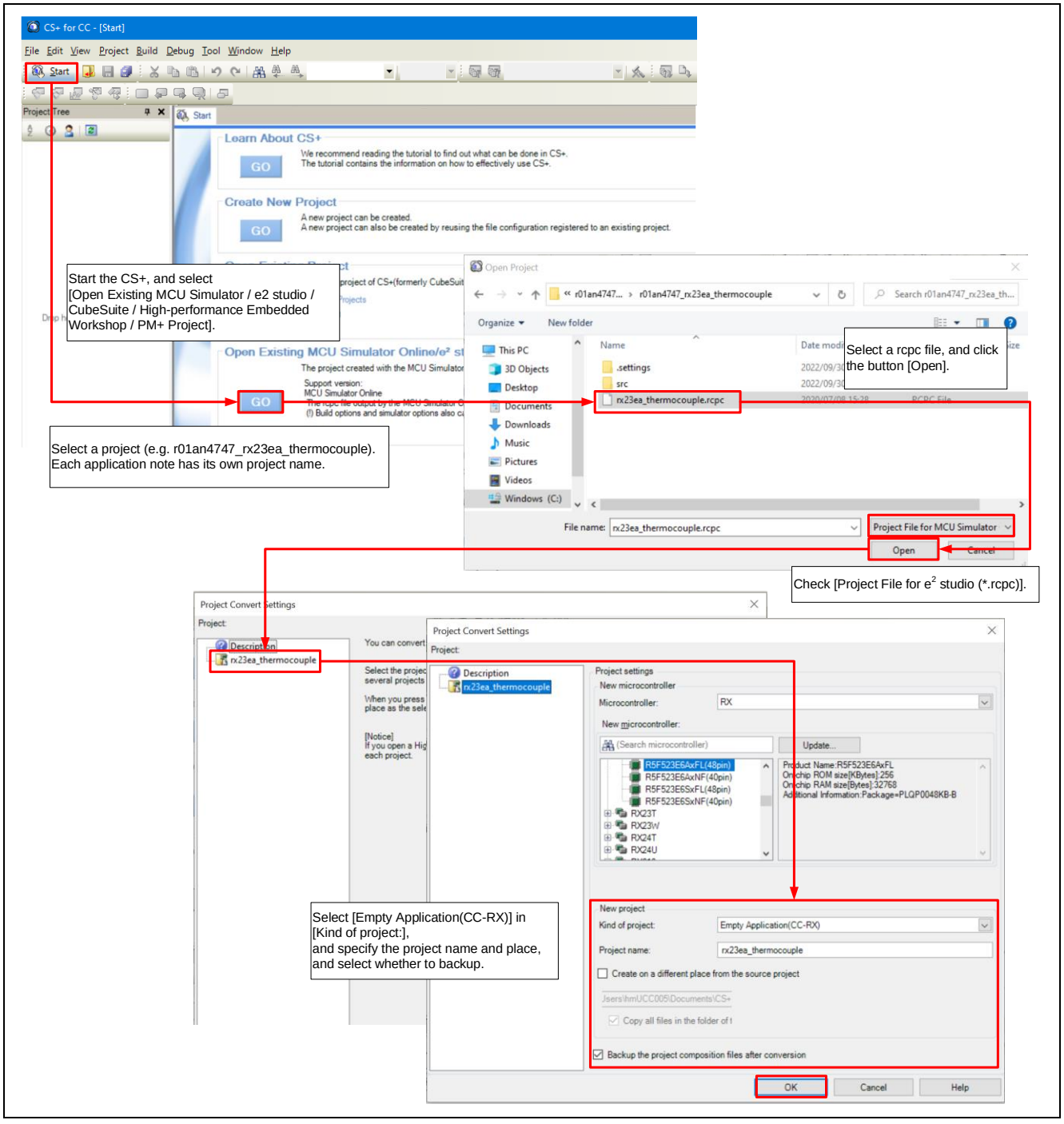

Figure 4-4 Import a Project into CS+

4.2.1 Build the project.

1. Select [Build] -> [Build project].
2. The build starts and the console displays the status of the build. When the message “Build completed” is displayed, the build is complete.

4.2.2 Debug

1. Right-click the RX simulator and select [RX E2 Lite] from [Using Debug Tool].
2. Verify that the displayed property screen is as shown in Figure 4-5.

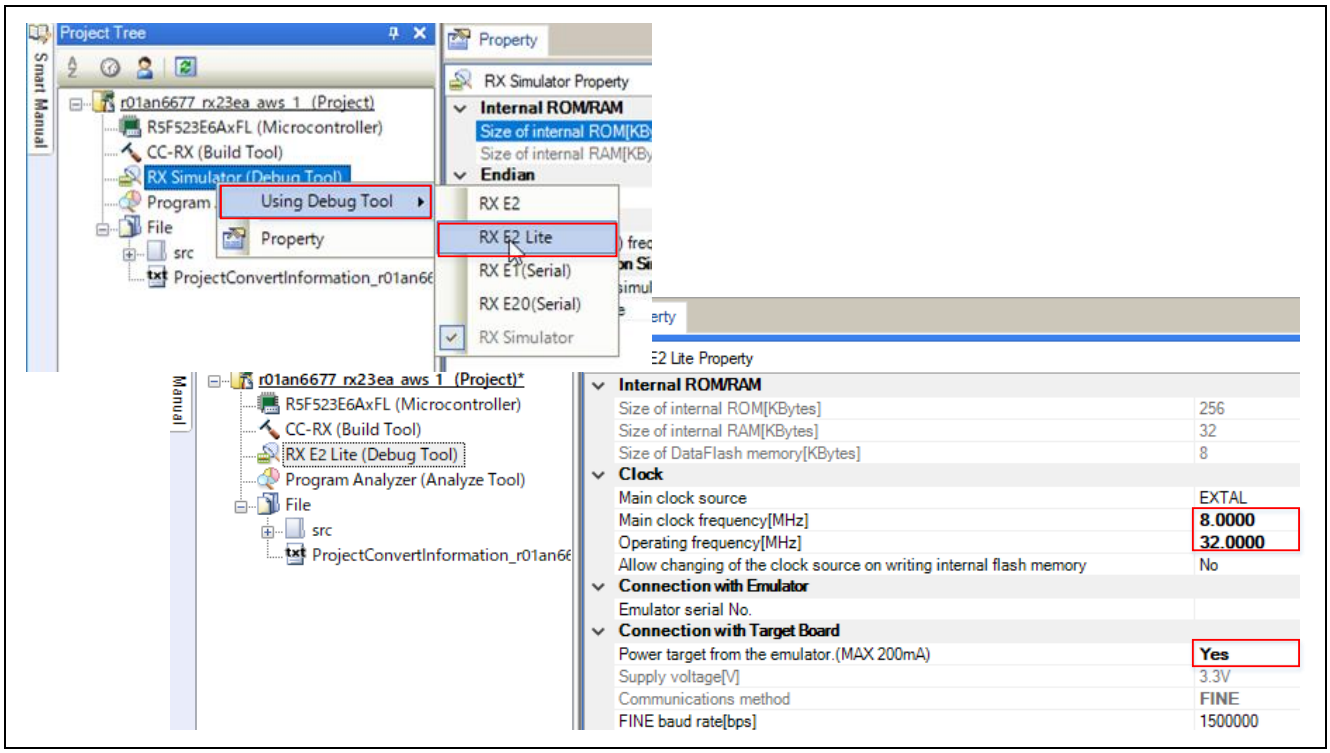

Figure 4-5 Debug Setting

4.2.3 Run

1. When you click the  button, the program will be downloaded.
2. Select［Renesas Views] -> [Debug] -> [Renesas Debug Virtual Console].
3. Clicking the  button will execute the program.

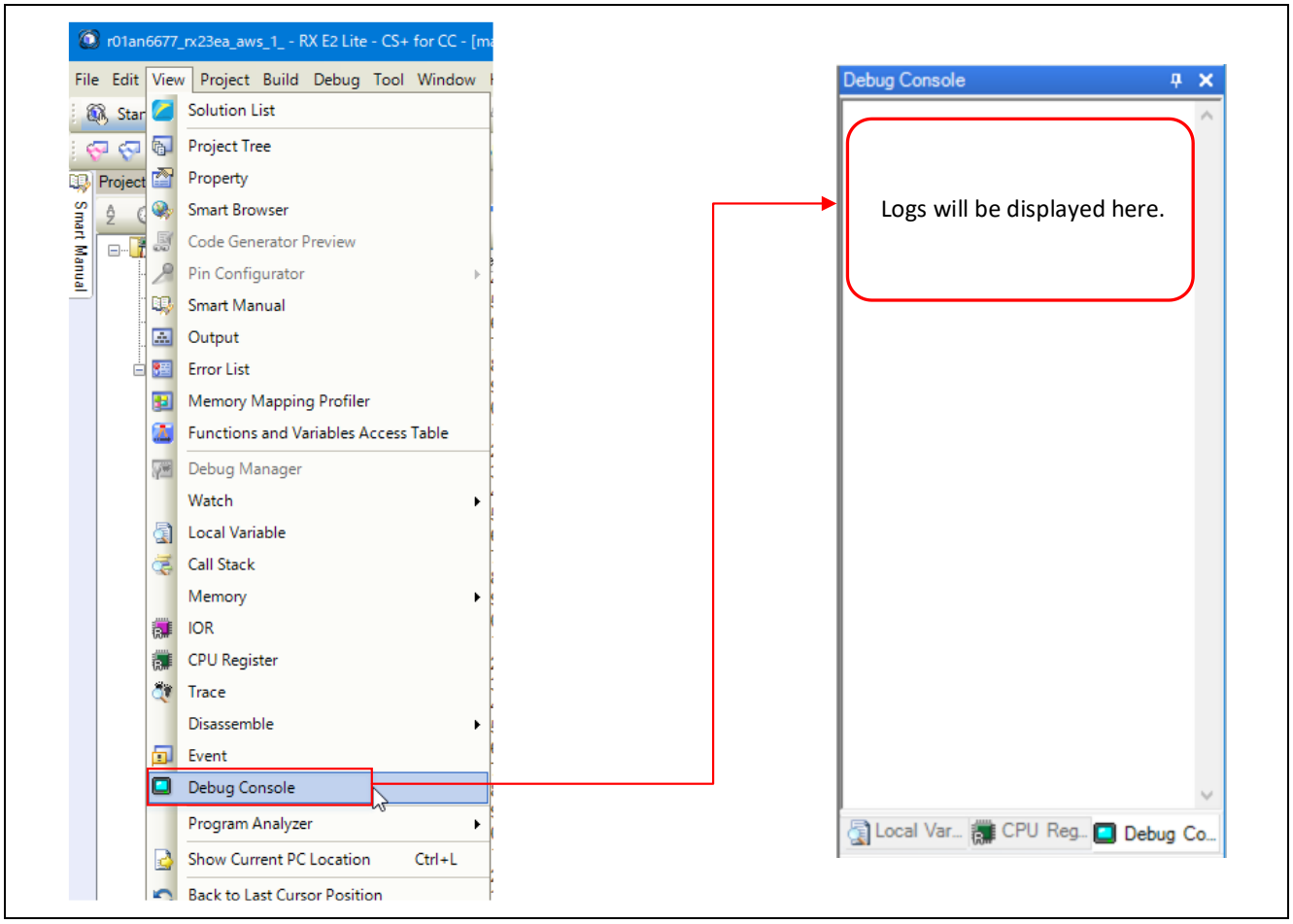

Figure 4-6 Renesas Debug Virtual Console(CS+)

5. Troubleshooting

5.1 Unable to connect to AWS

There are various reasons for being unable to connect to AWS. The main reasons are "Wrong AWS endpoint" or "Wrong certification or private key". To verify these exactly, check the logs via the debug port on the DA16600 Pmod™ Board. Here is how to check the logs and troubleshoot some of them.

5.1.1 How to check the DA16600 Pmod™ Board logs

5.1.1.1 Required Parts

The following modules are required to view the DA16600 Pmod™ Board logs.

- Pmod USBUART (manufactured by DIGILENT)
  https://digilent.com/reference/pmod/pmodusbuart/start?redirect=1

5.1.1.2 Connection method

Connect Pmod USBUART and DA16600 Pmod™ Board as follows. Keep the connection of the RSSKRX23E-A as it is. However, when connecting the Pmod USBUART, RSSKRX23E-A must be disconnected from the power supply.

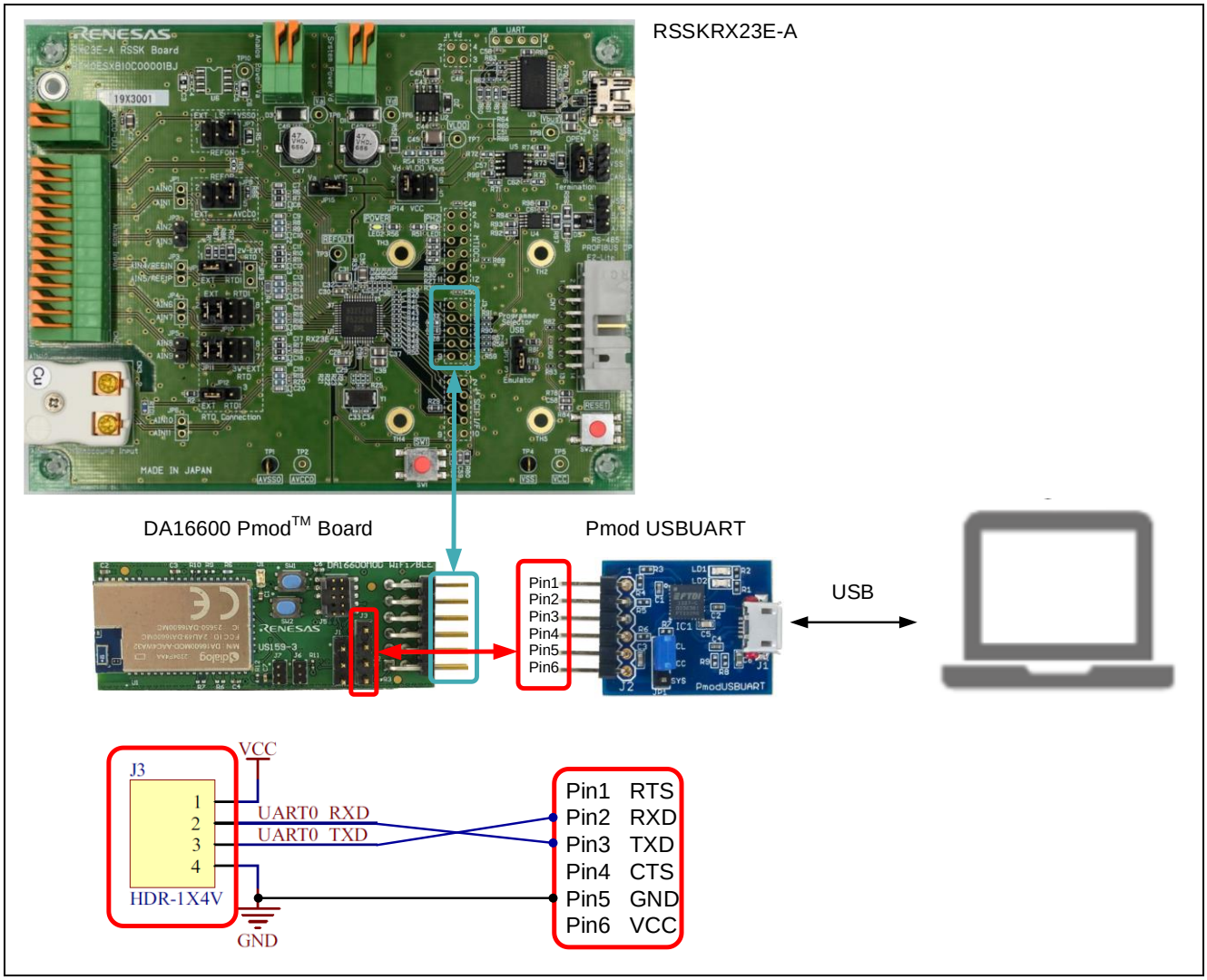

Figure 5-1 Connection Method between Pmod USBUART and DA16600 Pmod™ Board

5.1.1.3 Open Tera Term

After power on RSSKRX23E-A, open Tera Term. Check "Serial" and select the COM port which the Pmod USBUART is connected.

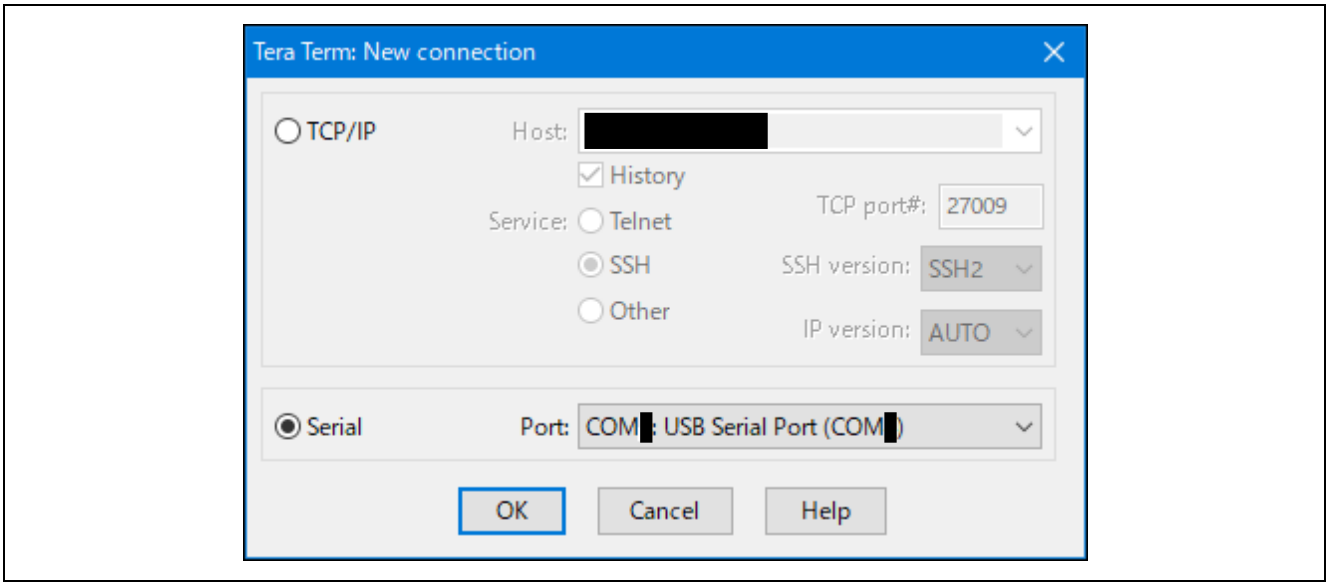

Figure 5-2 Open method for Tera Term

5.1.1.4 Tera Term Terminal Settings

From the Tera Term window, open "Setup" → "Terminal setup". Set as follows.

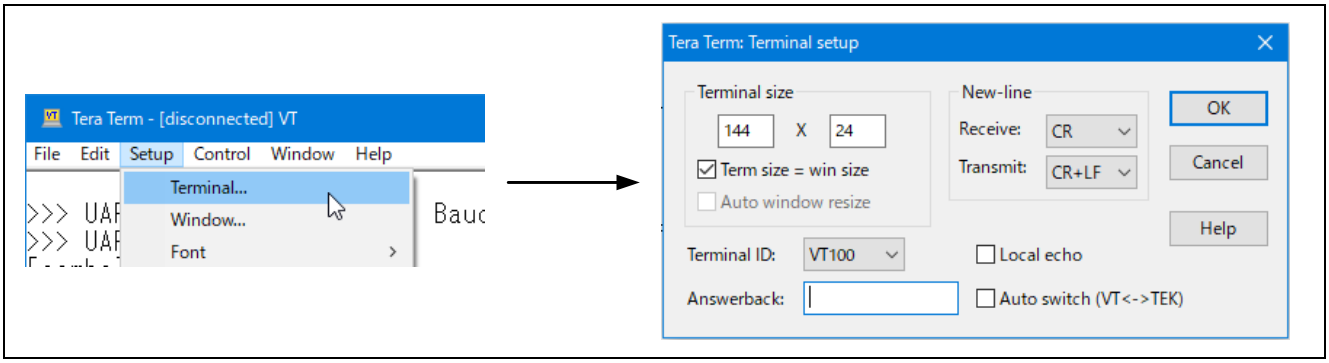

Figure 5-3 Tera Term's "Terminal setup" screen

5.1.1.5 Tera Term Serial Settings

From the Tera Term window, open "Setup" → "Serial port setup and connection". The baud rate must match the debug port setting of the DA16600 Pmod™ Board. Set as follows.

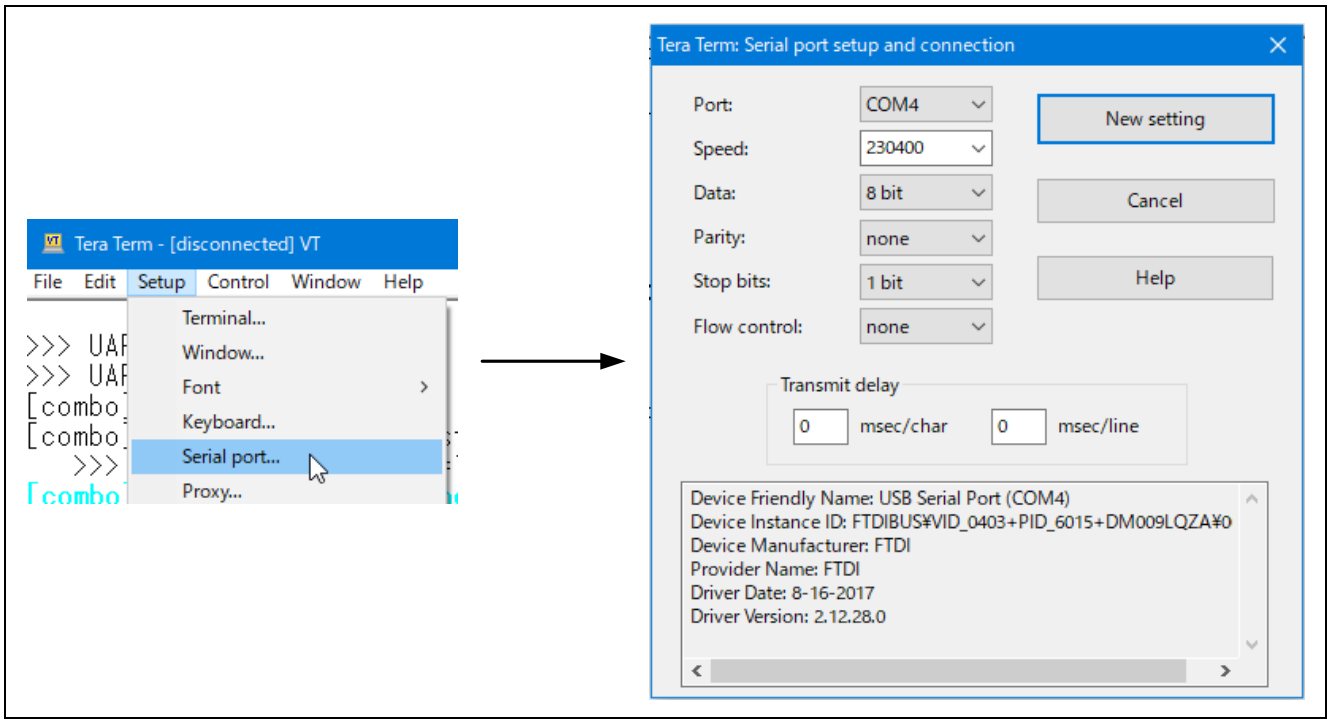

Figure 5-4 Tera Term's "Serial port setup and connection" screen

5.1.1.6 Check operation

Power on the RX23E-A and check that the logs are displayed correctly. The following is typical logs.

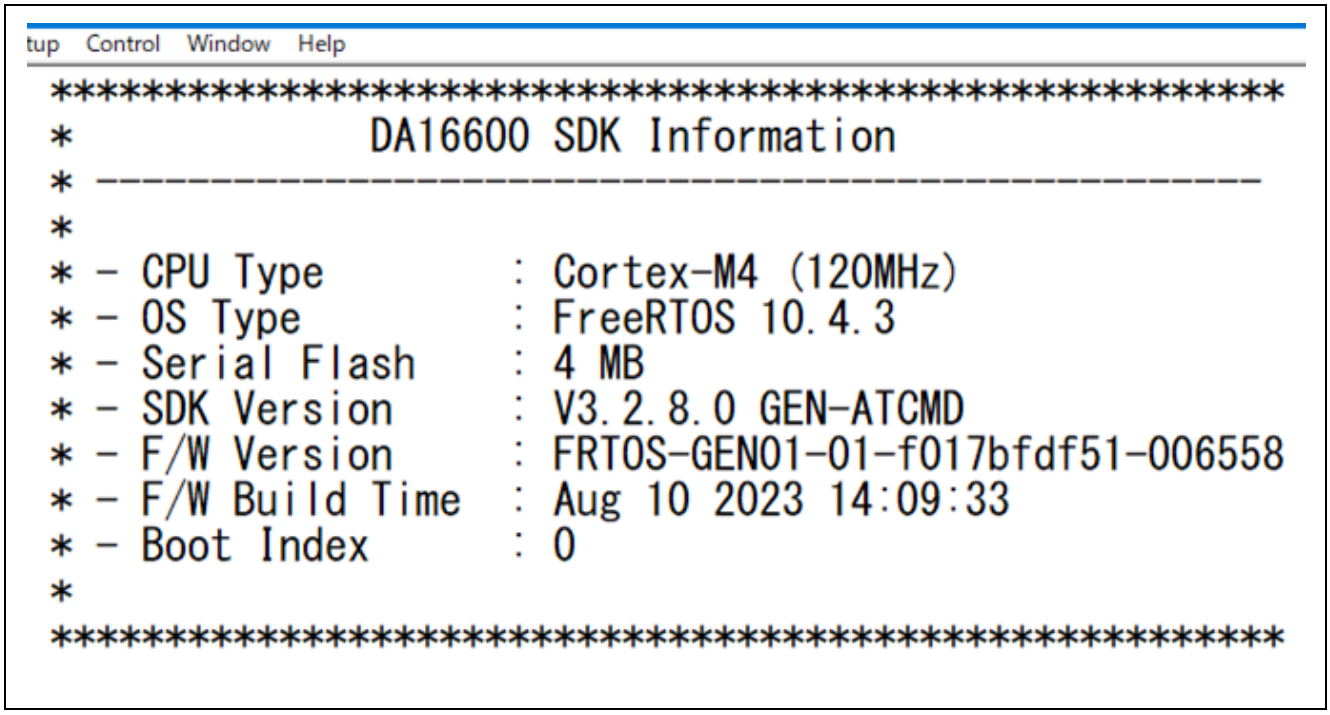

Figure 5-5 Log Example

5.1.2 Update the firmware version of the DA16600 Pmod™ Board

This application note has been confirmed to work with DA16600_IMG_FreeRTOS_ATCMD_UART2_EVK_v3.2.8.0_4. If the firmware version is different, update the firmware following the instructions below.

5.1.2.1 Download firmware

Connect to the following and download DA16200 DA16600 FreeRTOS SDK Image v3.2.8.0.

https://www.renesas.com/jp/en/products/interface-connectivity/wireless-communications/wi-fi/low-power-wi-fi/da16200-ultra-low-power-wi-fi-soc-battery-powered-iot-devices#design_development

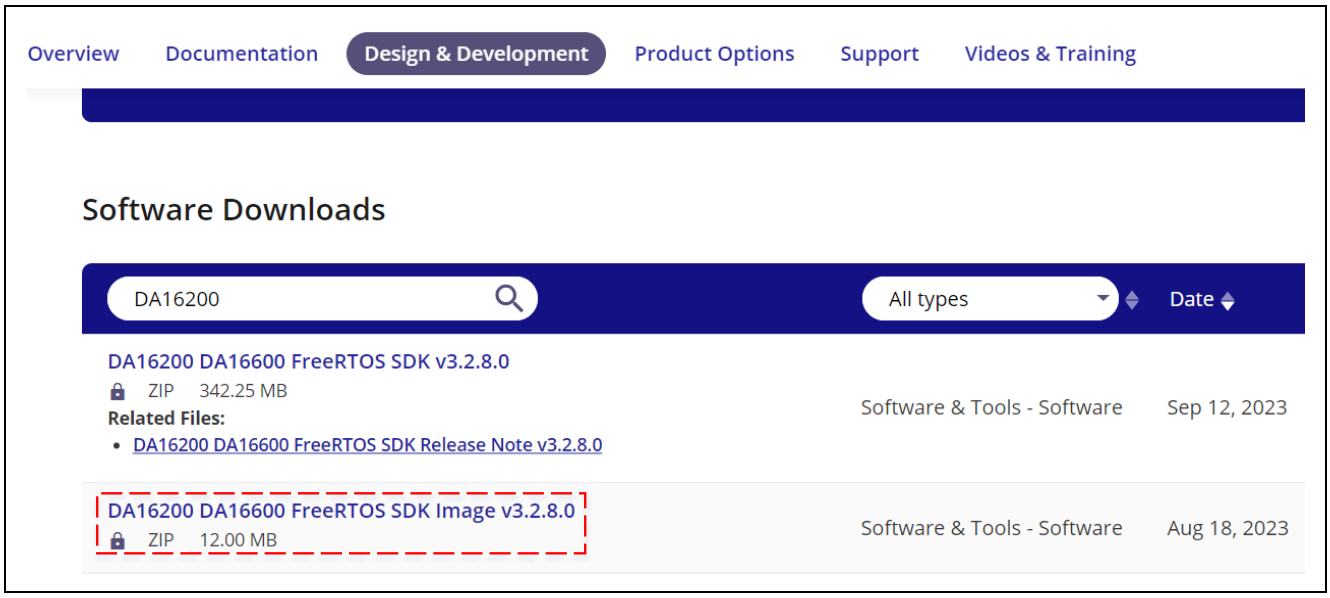

Figure 5-6 **Download firmware**

5.1.2.2 Unzip the downloaded file

Unzip the downloaded zip file to an optional folder. After unzipping, further unzip the DA16600_IMG_FreeRTOS_ATCMD_UART2_EVK_v3.2.8.0_4MB.zip

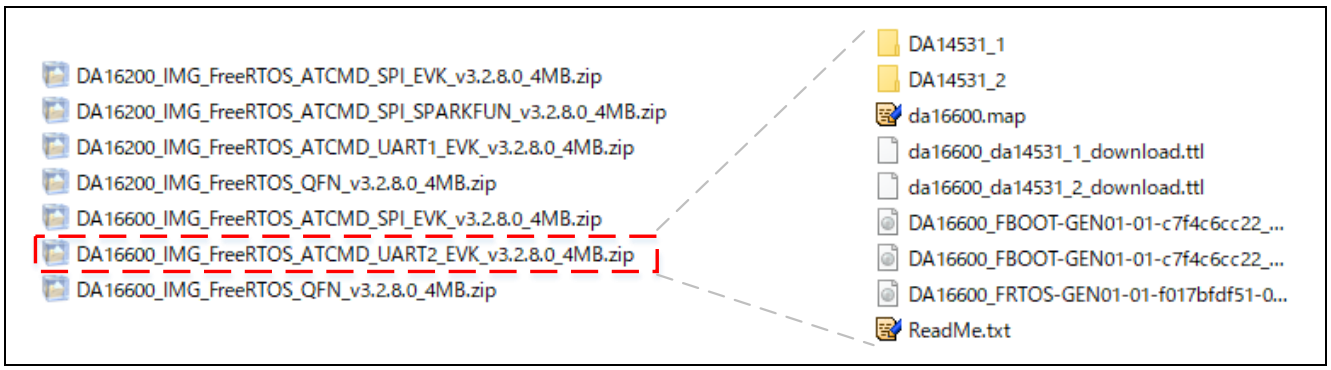

Figure 5-7 Unzip the downloaded file

5.1.2.3 Connect with Pmod USBUART

Connect Pmod USBUART and DA16600 Pmod™ Board as follows.

All Pmod pins on the DA16600 Pmod™ Board should be open. The Pmod USBUART and the PC should be connected last.

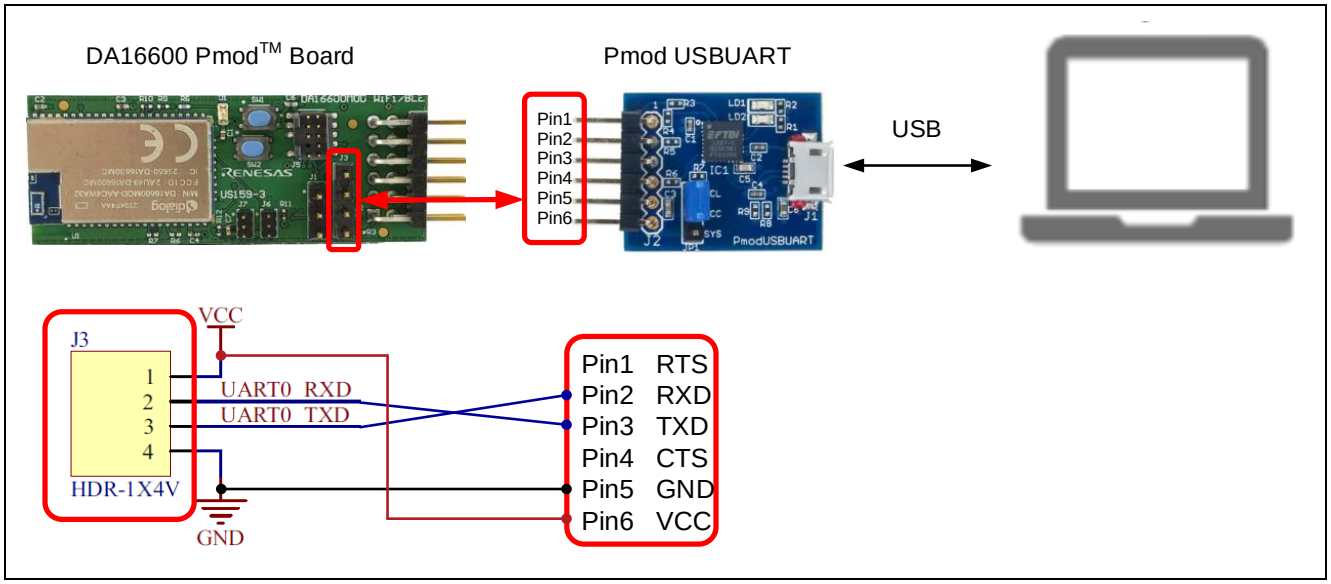

Figure 5-8 Connection Method between Pmod USBUART and DA16600 Pmod™ Board

5.1.2.4 Power supply

When Pmod USBUART and PC are connected via USB, power is supplied to Pmod USBUART and DA16600 Pmod™ Board.

5.1.2.5 Open Tera Term

After power on RSSKRX23E-A, open Tera Term. Check "Serial" and select the COM port which the Pmod USBUART is connected.

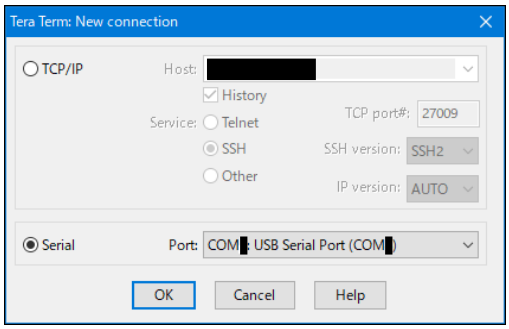

Figure 5-9 Open method for Tera Term

5.1.2.6 Tera Term Terminal Settings

From the Tera Term window, open "Setup" → "Terminal setup". Set as follows.

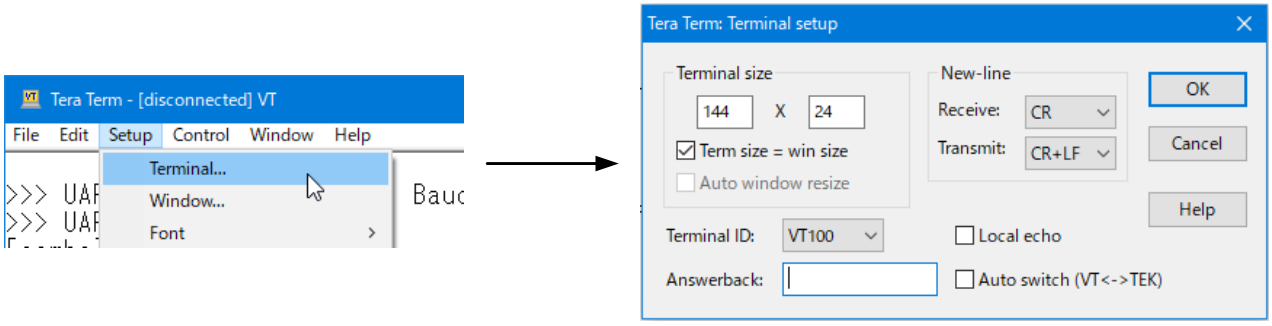

Figure 5-10 Tera Term's "Terminal setup" screen

5.1.2.7 Tera Term Serial Settings

From the Tera Term window, open "Setup" → "Serial port setup and connection". The baud rate must match the debug port setting of the DA16600 Pmod™ Board. Set as follows.

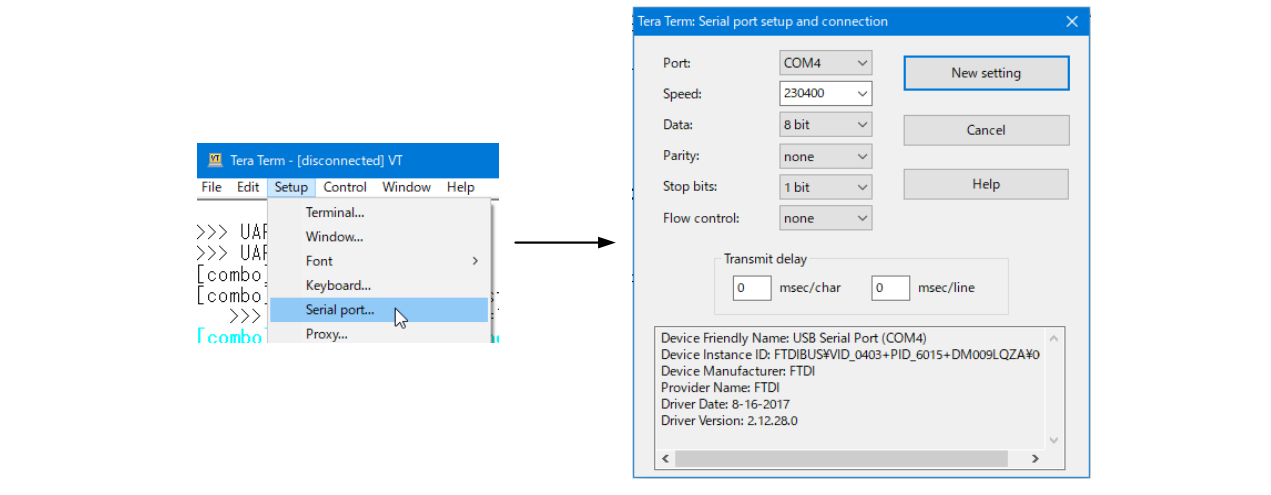

Figure 5-11 Tera Term's "Serial port setup and connection" screen

5.1.2.8 Update the firmware

From the Tera Term window, open "Control" → "Macro". Select "da16600_da14531_1_download.ttl" in the folder unzipped in "5.1.2.2 Unzip the downloaded file" and click "Open". Then, on the "Confirm" screen, select "AT25SL321" and click "OK" to execute automatically the command on Tera Term and update the firmware.

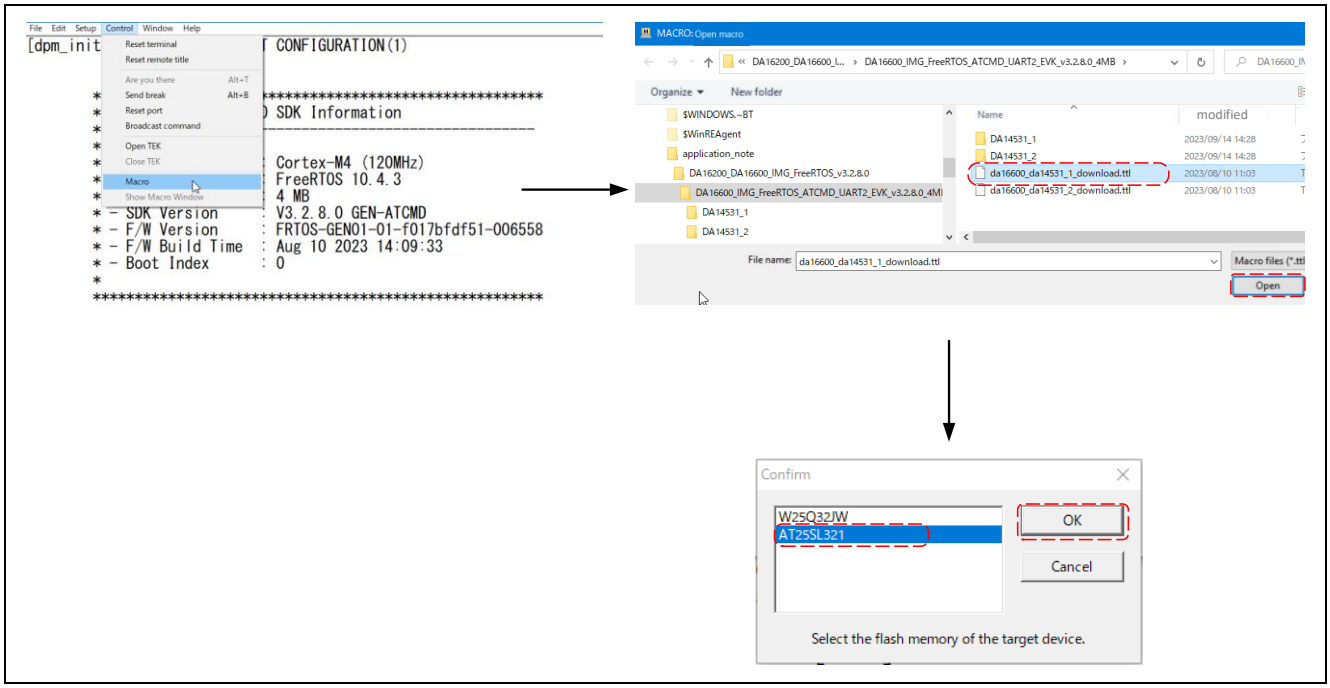

Figure 5-12 Firmware Update

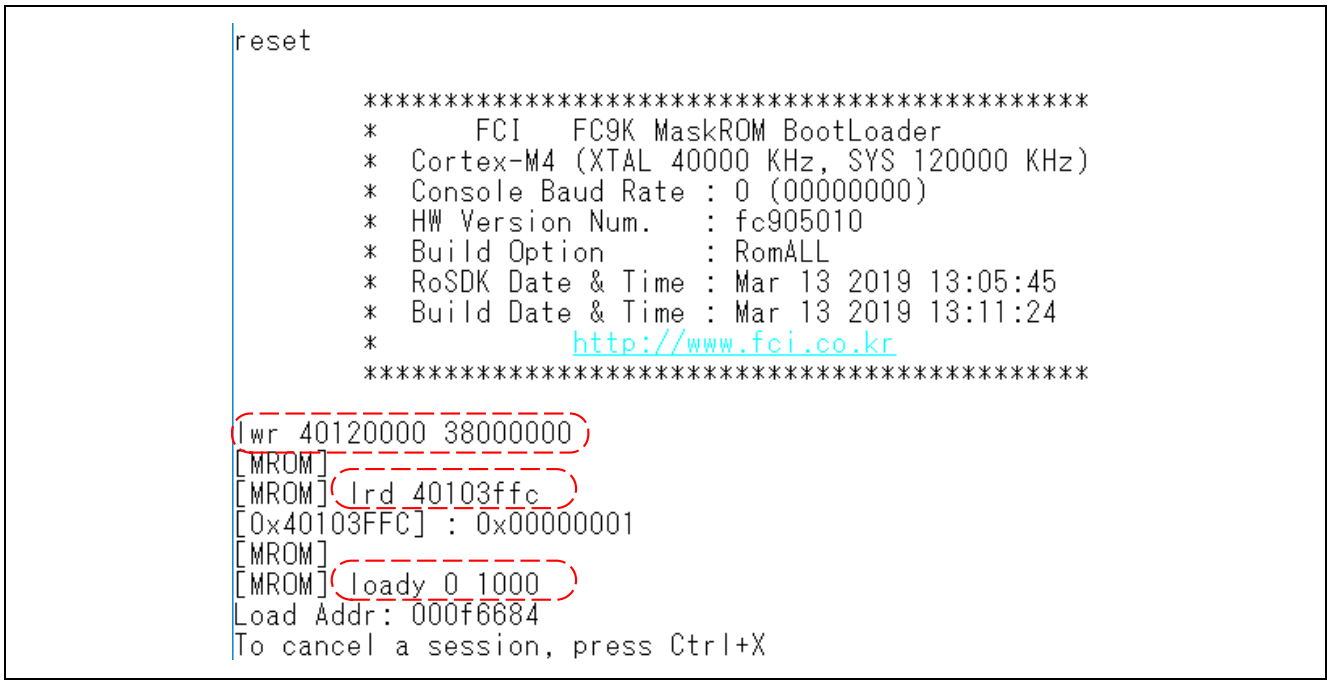

Figure 5-13 Tera Term screen Updating Firmware

5.1.2.9 Check the version

After the firmware has been updated, check that the firmware version is v3.2.8.0.

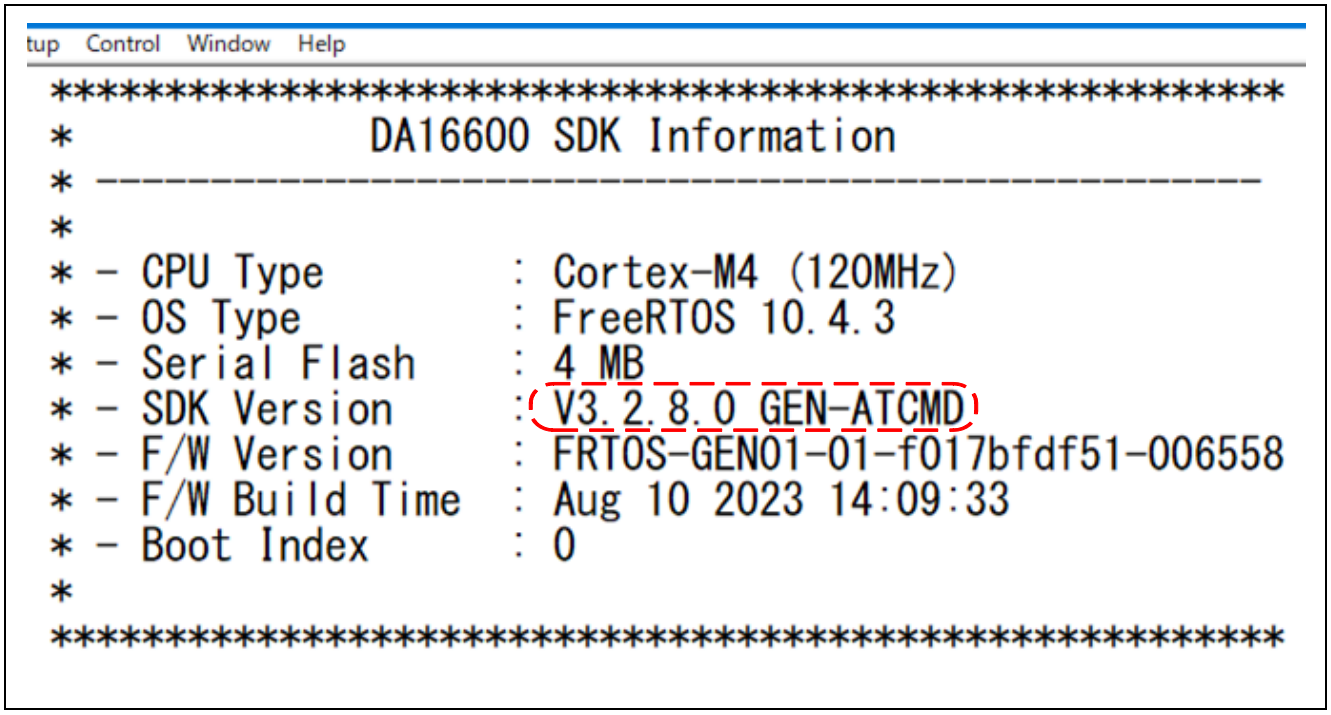

Figure 5-14 Check firmware version

5.1.3 If "Fail to establish tls-sess(0x7200)" is displayed

The following describes a case where Fail to establish tls-sess(0x7200) is displayed.

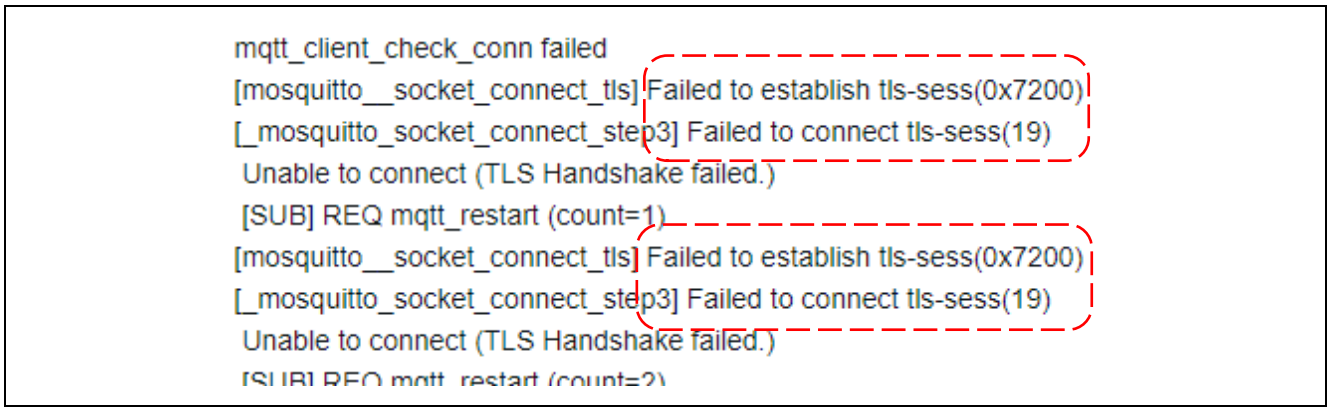

Figure 5-15 If "Fail to establish tls-sess(0x7200)" is displayed

5.1.3.1 Buffer reconfiguration for MQTT

If Fail to establish tls-sess (0x7200) is displayed, reconfigure the buffer for MQTT. Execute the following command and perform the reconfiguration.

```
setenv MQTT_TLS_INCOMING 16384
setenv MQTT_TLS_OUTGOING 16384
```

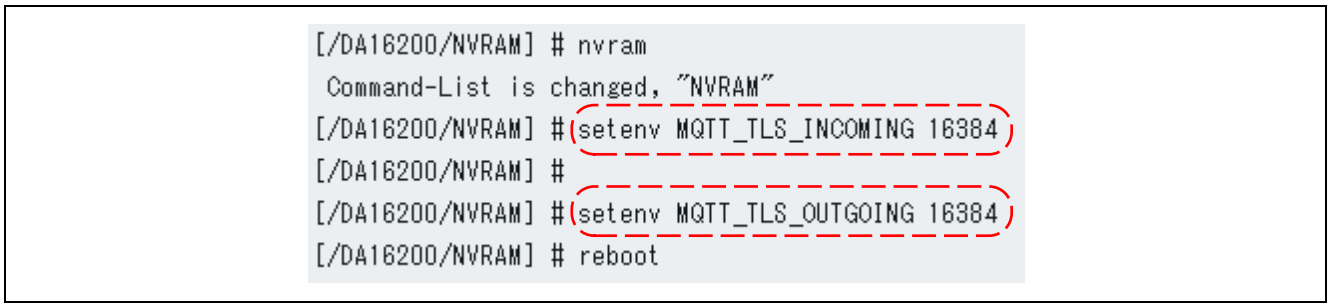

Figure 5-16 MQTT buffer reconfiguration

5.1.3.2 Check the setting result

Execute the following command to check that it has been reconfigured.

```
mqtt_config status
```

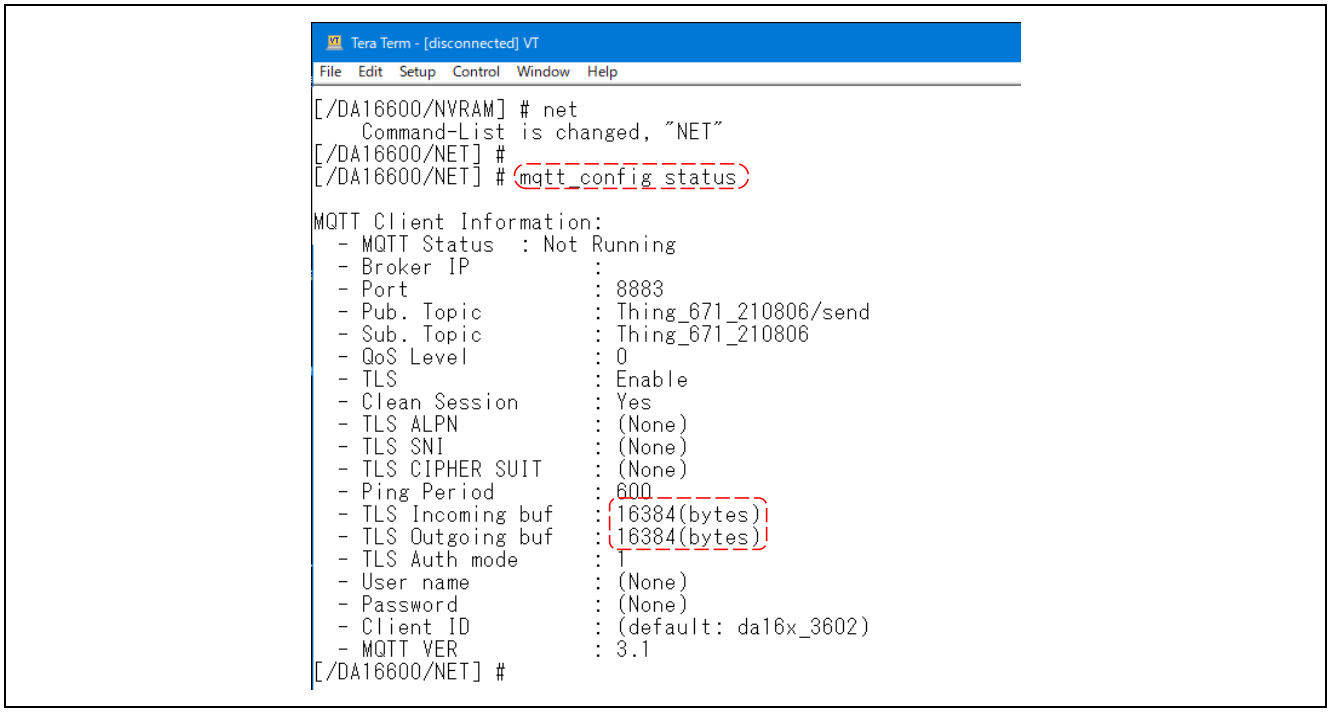

Figure 5-17 MQTT buffer reconfiguration check result

6. Reference Documents

- RX23E-A Group User's Manual (R01UH0801)
- RSSKRX23E-A User's Manual (R20UT4542)
- RX23E-A Group Temperature Measurement Example Using a Thermocouple（R01AN4747)
- US159-DA14531EVZ Evaluation Board Manual(R15UZ0004)
- US159-DA 16600 EVZ Evaluation Board Manual (R15UZ0006)
- GATTBrowser for Windows Windows Application Instruction manual (R01AN6230)
- User Manual DA16200 DA16600 AT Command (UM-WI-003)
- Failure Detection and Movement Analysis Demonstration Using AWS Cloud and FFT (R01AN5366)

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep.27.23	-	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 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

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/.