RYZ014A and RA MCU

Firmware Upgrade from Host MCU

Introduction

This document describes a sample application that upgrades the RYZ014A firmware from the host MCU.

This sample application works in a configuration that uses the EK-RA6M5 board with RA6M5 as the host MCU and connects the PMOD™ Expansion Board for RYZ014A to the PMOD connector. Store the RYZ014A firmware file on a USB flash drive and connect it to the USB FS connector on the EK-RA6M5 board. Transfer the firmware from the USB flash drive to the RYZ014A for upgrade.

Target Device

RYZ014A

EK-RA6M5

Related Documents

- Renesas LTE Cat-M1 Cellular IoT Module RYZ014A Pmod™ Expansion Board (R21QS0004)
- RYZ014 Module System Integration Guide (R19AN0074)
- RA6M5 Group User's Manual: Hardware (R01UH0891)
- RA6M5 Group Evaluation Kit for RA6M5 Microcontroller Group EK-RA6M5 v1 User's Manual (R20UT4829)
- Renesas Flexible Software Package (FSP) User's Manual (R11UM0155)

Pmod™ is registered to Digilent Inc.

Contents

1. Overview

During a product lifetime, at least one modem software upgrade will most likely be required. There are three types of firmware upgrade methods for RYZ014A, and this application note describes how to upgrade from the control microcomputer (Host MCU) among them.

- Connect to LTE network and upgrade wirelessly (Firmware Upgrade Over-the-Air (FOTA))
- Upgrade from PC
- Upgrade from the control microcomputer (Host MCU)

It works in a configuration where the EK-RA6M5 board with RA6M5 is used as the host MCU and the PMOD Expansion Board for RYZ014A is connected to the PMOD connector. Store the RYZ014A firmware file on a USB flash drive and connect it to the USB FS connector on the EK-RA6M5 board. Transfer the firmware from the USB flash drive to the RYZ014A for upgrade.

Figure 1-1 shows a configuration diagram of the firmware upgrade sample application.

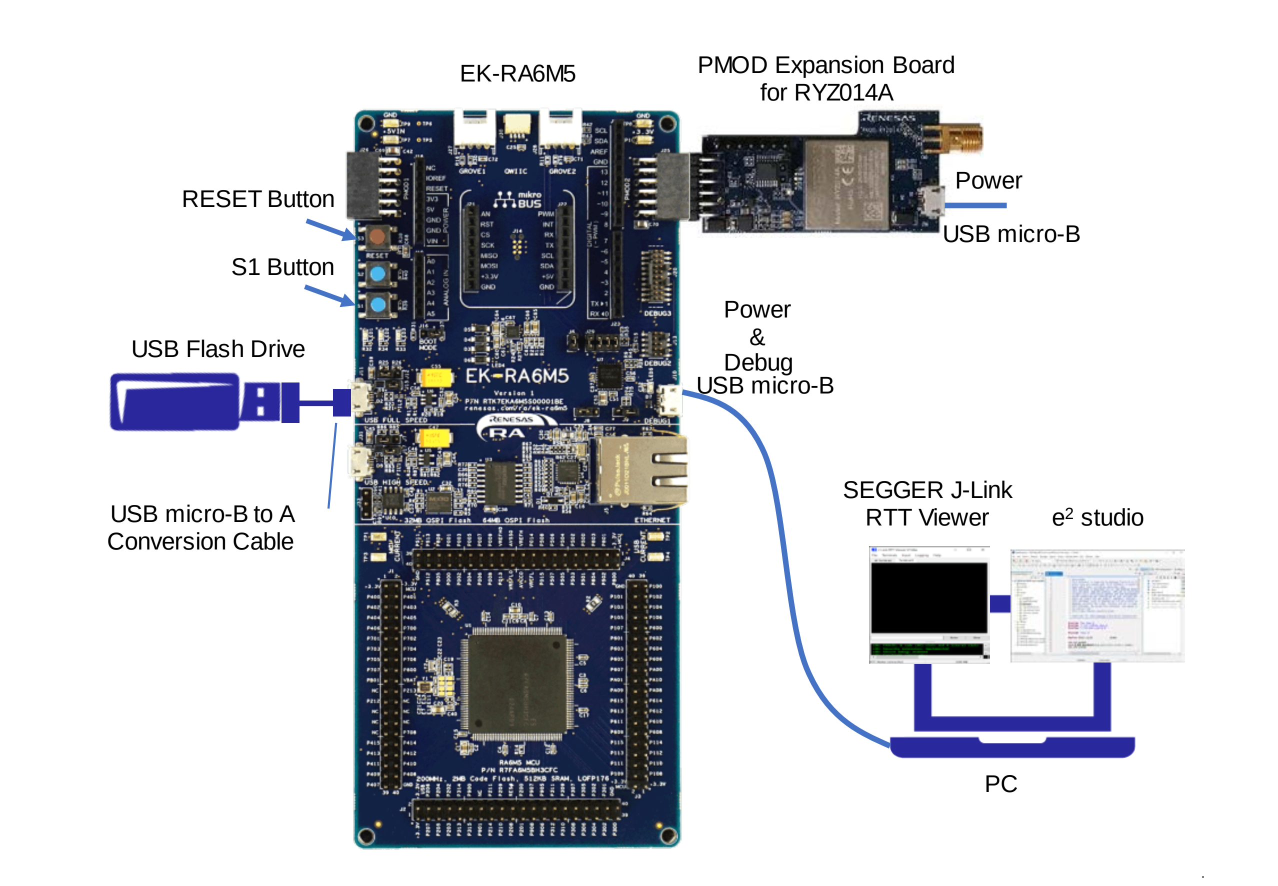


Figure 1-1. Firmware upgrade sample application configuration diagram

2. Operating Environment

2.1 Hardware

The hardware requirements used in the sample application are shown below.

Table 2-1. Hardware requirements

Hardware	Description
PMOD Expansion Board for RYZ014A	RTKYZ014A0B00000BE
EK-RA6M5	RTK7EKA6M5S00001BE
USB Flash Drive	10MB or more
Windows 10 PC	---
USB Cable	● for EK-RA65M USB Debug connecter (micro-B) ● for PMOD Expansion Board for RYZ014A (micro-B) ● for USB Flash Drive (USB micro-B to A Conversion)

2.2 Software

The software requirements used in the sample application are shown below.

Table 2-2. Software requirements

Software	Version
e2 studio	2022-04
GCC Compiler	10.3.1
FSP	3.7.0
Renesas Flash Programmer	V3.09
SEGGER J-Link RTT Viewer	V7.66 (It is used in the execution log display for operation check.) (Download J-Link RTT Viewer)

2.3 Microcontroller Peripheral Functions

The microcontroller peripheral functions used in the sample application are shown below.

Table 2-3. Microcontroller peripheral functions

Peripheral function name		Description
Serial Communication Interface	SCI0(UART0)	UART communication with RYZ014A Baudrate :921600 bps Data length :8 bit Parity :none Stop bit :1 bit Flow control :Hardware CTS/Software RTS RTS : P412 CTS : P413
Low power Asynchronous General Purpose Timer	AGT0	UART communication timeout with RYZ014A
	AGT1	LED blinking cycle
USB2.0 Full-Speed Module	USBFS	Communication with the USB flash drive where the firmware file is stored.
I/O Port	P404	Reset pin control of RYZ014A
	P412	RTS signal used for UART communication with RYZ014A.
	P413	CTS signal used for UART communication with RYZ014A.
	P006	User LED1 (Blue)
	P008	User LED3 (Red)
External Interrupt Request	IRQ10	External interrupt request of user button S1

2.4 FSP Modules

The FSP modules used in the sample application are shown below.

Table 2-4. FSP modules

Module Type	Module Name	
System	I/O Port	(r_ioport)
Storage	FreeRTOS+FAT	(rm_freertos_plus_fat) (Note
Storage	Block Media USB	(rm_block_media_usb)
Connectivity	USB HMSC	(r_usb_hmsc)
Connectivity	USB	(r_usb_basic)
Transfer	Transfer	(r_dmac)
Input	External IRQ	(r_icu)
Connectivity	UART	(r_sci_uart)
Timers	Timer, Low-Power	(r_agt)

Note: This program is a bare metal version.

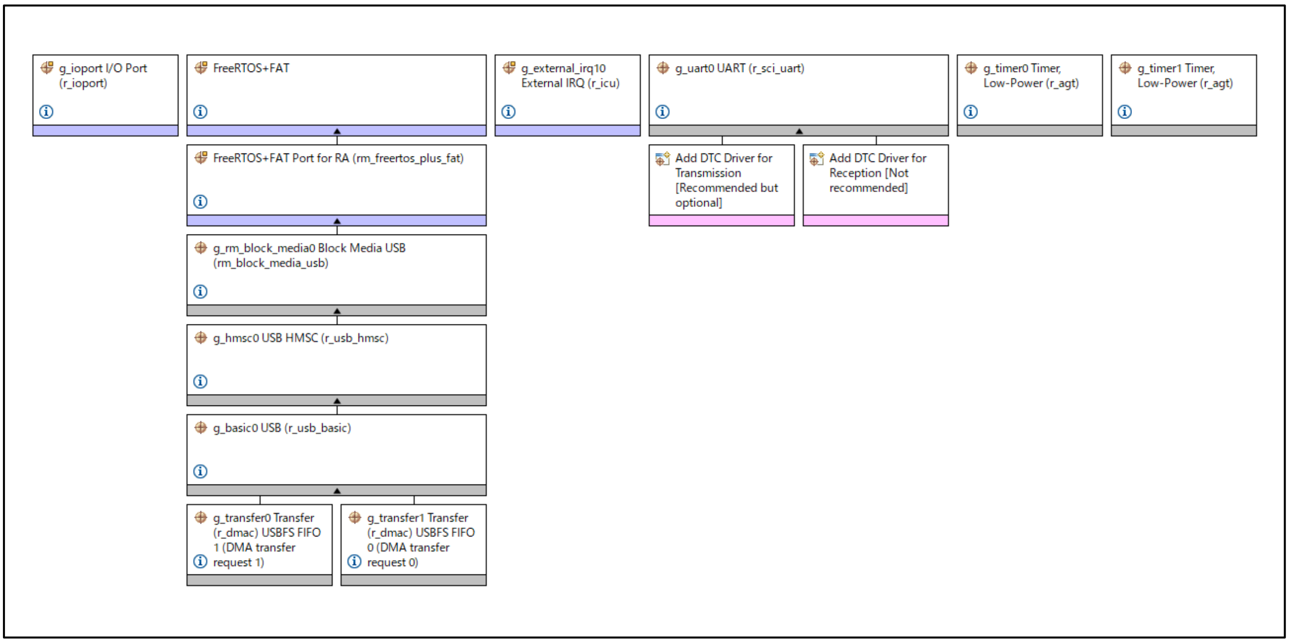

g_ioport I/O Port (r_ioport)
FreeRTOS+FAT
g_external_irq10 External IRQ (r_icu)
g_uart0 UART (r_sci_uart)
g_timer0 Timer, Low-Power (r_agt)
g_timer1 Timer, Low-Power (r_agt)
FreeRTOS+FAT Port for RA (rm_freertos_plus_fat)
Add DTC Driver for Transmission [Recommended but optional]
Add DTC Driver for Reception [Not recommended]
g_rm_block_media0 Block Media USB (rm_block_media_usb)
g_hmsc0 USB HMSC (r_usb_hmsc)
g_basic0 USB (r_usb_basic)
g_transfer0 Transfer (r_dmac) USBFS FIFO 1 (DMA transfer request 1)
g_transfer1 Transfer (r_dmac) USBFS FIFO 0 (DMA transfer request 0)

Figure 2-1 FSP module configuration

2.5 Directory/File Structure

Directory/file structure of sample application are shown below.

Table 2-5. Directory/File structure

Directory/ File Structure				Description
RYZ014A_FWUP_from_HostMCU\		.api_xml .cproject .project .secure_azone .secure_xml configuration.xml R7FA6M5BH3CFC.pincfg ra_cfg.txt RYZ014A_FWUP_from_HostMCU Debug_Flat.jlink RYZ014A_FWUP_from_HostMCU Debug_Flat.launch		Project files for GCC. Files for RA configurator.
	.settings\	com.renesas.cdt.ddsc.content.prefs com.renesas.cdt.ddsc.packs.componentfiles.prefs com.renesas.cdt.ddsc.settingseditor.prefs com.renesas.cdt.ddsc.threads.configurator.prefs com.renesas.cdt.managedbuild.gnuarm.prefs CoverageSetting.xml DebugVirtualConsoleSetting.xml e2studio_project.prefs IORegisterSetting.xml language.settings.xml org.eclipse.cdt.core.prefs org.eclipse.cdt.managedbuilder.core.prefs		e^2 studio setting files.
	script\	fsp.ld		Linker setting file.
	src\	hal_entry.c r_lte_ryz014a_fwup.c r_lte_ryz014a_fwup.h r_lte_user_config.h	Firmware upgrade sample application source codes.	
		FreeRTOSConfig.h	Header files required when using FreeRTOS+FAT without FreeRTOS.	
		SEGGER_RTT\	SEGGER_RTT.c SEGGER_RTT.h SEGGER_RTT_Conf.h SEGGER_RTT_printf.c	SEGGER J-Link RTT Viewer source coeds.

3. How to Use this project

Shows how to use the sample application. Refer to "Figure 1-1 Firmware upgrade sample application configuration diagram" for the execution environment.

Please execute this sample application while disconnected from the LTE network.

3.1 Preparation of RYZ014A firmware

Rename the RYZ014A firmware file(Note) to the following name and copy it to the root directory of the USB flash drive.

ryz014a_firmware.dup

Note: Firmware files will be provided directly to you by Renesas as needed. Please contact renesas and get it. Please note that the file will differ depending on the firmware version.

3.2 EK-RA6M5 jumper setting

The USB Full Speed interface of EK-RA6M5 is used as a USB Host to connect a USB flash drive. Set the jumper as follows according to "RA6M5 Group Evaluation Kit for RA6M5 Microcontroller Group EK-RA6M5 v1 User's Manual" (R20UT4829).

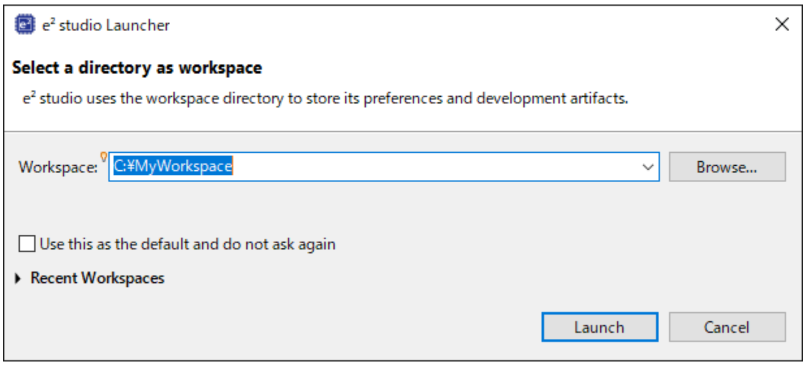
J12: Jumper on pins 1-2

J15: Open

3.3 Import of project

The steps to import a sample application project into e^2 studio are shown below.

1. Launch e2 studio, specify the workspace directory, and click the "Launch" button.



e² studio Launcher

Select a directory as workspace

e² studio uses the workspace directory to store its preferences and development artifacts.

Workspace: C:¥MyWorkspace

□ Use this as the default and do not ask again

Figure 3-1. Workspace selection

2. Select "File" ➔ "Import ..." from the menu bar.

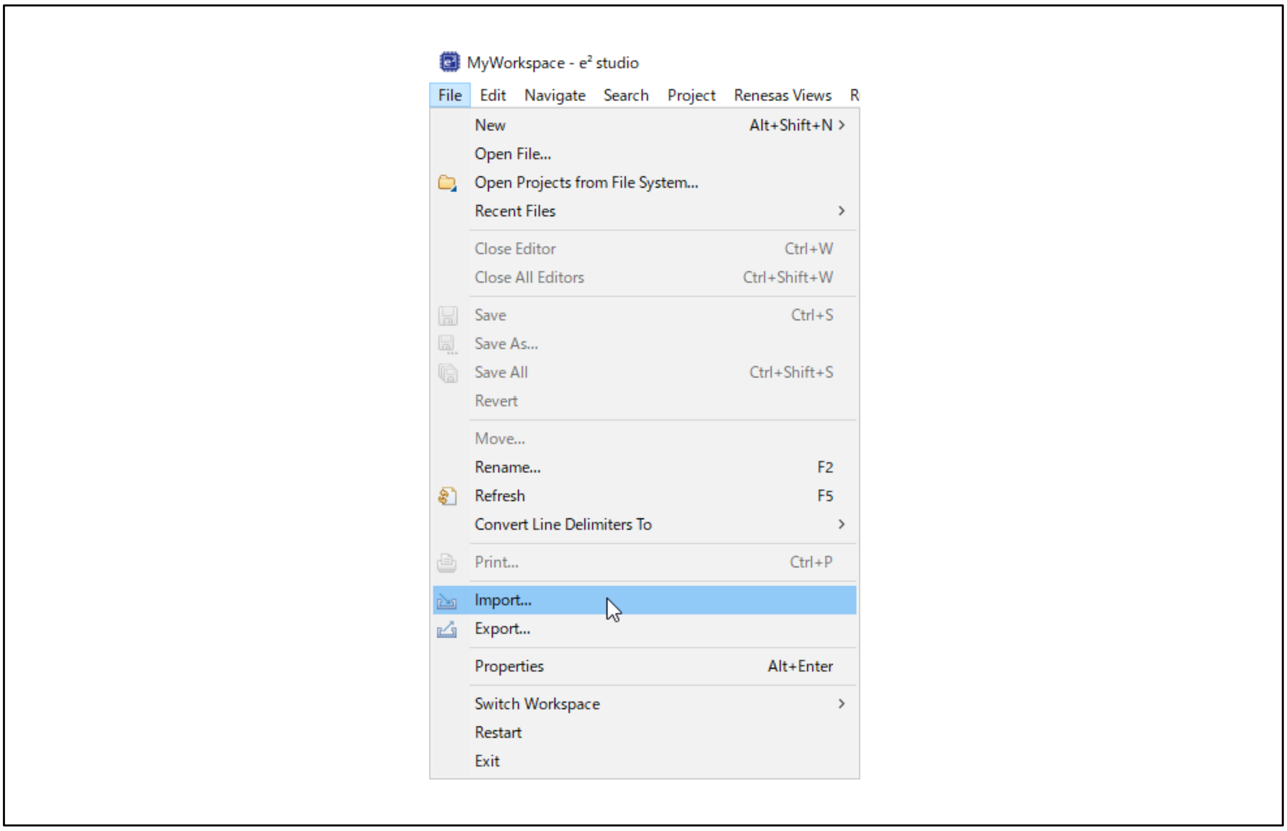
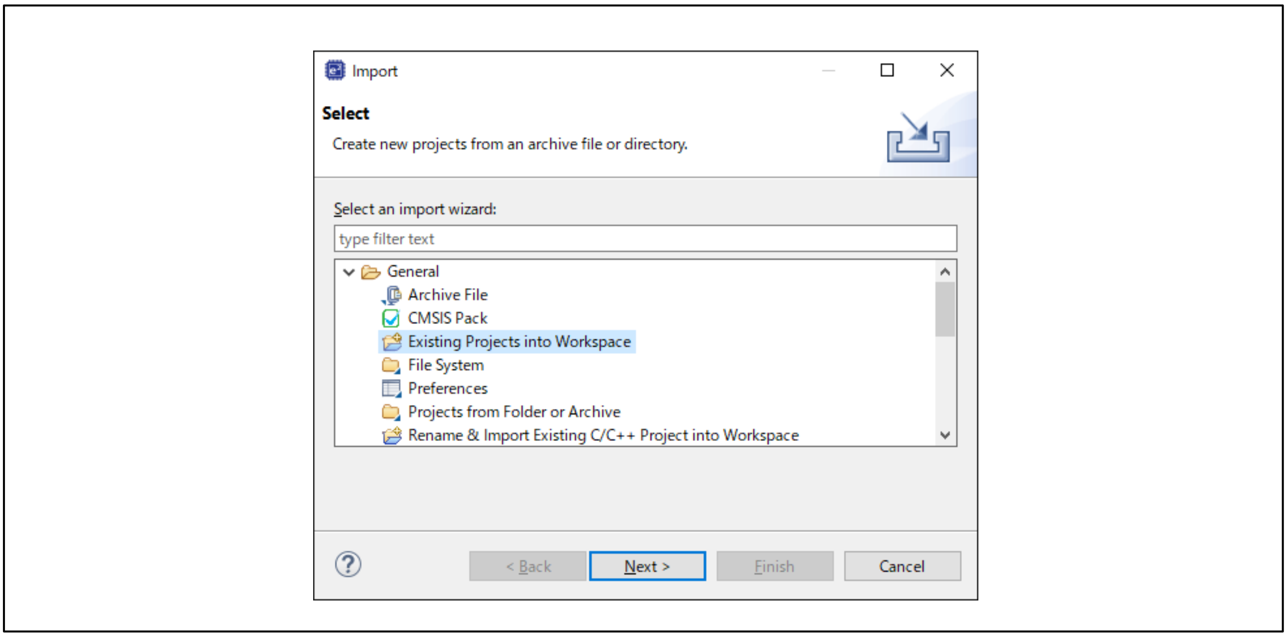


Figure 3-2. File menu

3. Select "Existing Projects into Workspace" and click "Next".

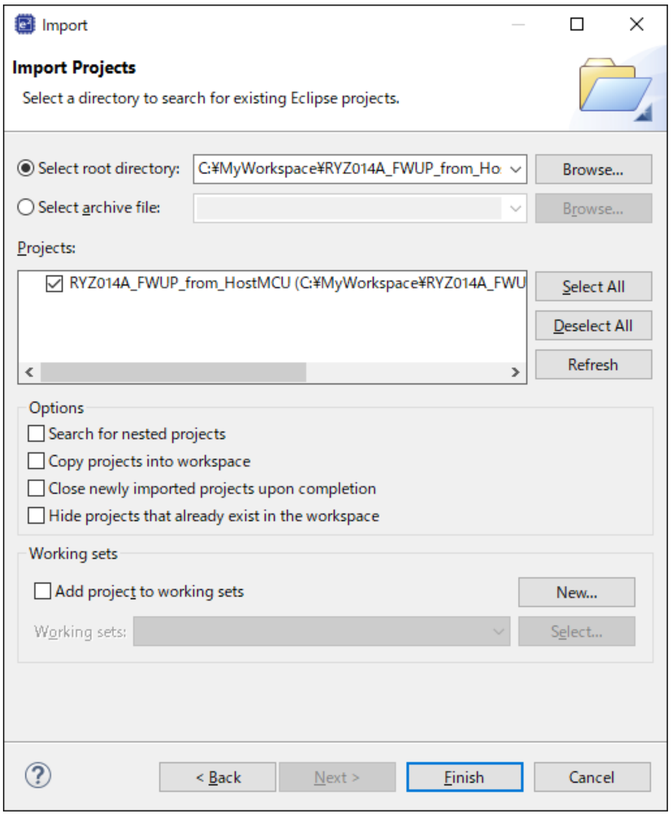


Create new projects from an archive file or directory.

Select an import wizard:
type filter text
General
Archive File
CMSIS Pack
Existing Projects into Workspace
File System
Preferences
Projects from Folder or Archive
Rename & Import Existing C/C++ Project into Workspace
< Back
Next >
Finish
Cancel

Figure 3-3. Import wizard selection

4. Select "Select root directory", click "Browse..." and select the directory for your sample project. Click the "Finish" button to import the project.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:¥MyWorkspace¥RYZ014A_FWUP_from_Ho [Browse...]

○ Select archive file: [Browse...]

Projects:

☑ RYZ014A_FWUP_from_HostMCU (C:¥MyWorkspace¥RYZ014A_FWU

[Select All] [Deselect All] [Refresh]

Options

☐ Search for nested projects

☐ Copy projects into workspace

☐ Close newly imported projects upon completion

☐ Hide projects that already exist in the workspace

Working sets

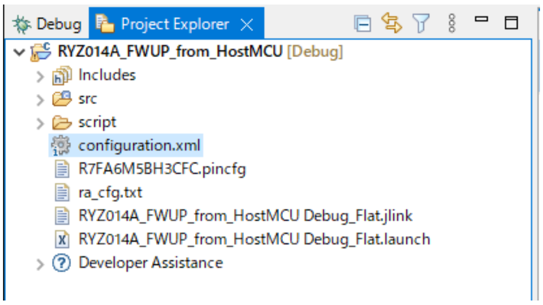
☐ Add project to working sets [New...]

Working sets: [Select...]

Figure 3-4. Import project

3.4 Build and Download

1. Double-click configuration.xml to open the FSP Configuration window.



- RYZ014A_FWUP_from_HostMCU [Debug]
  - Includes
  - src
  - script
  - configuration.xml
  - R7FA6M5BH3CFC.pincfg
  - ra_cfg.txt
  - RYZ014A_FWUP_from_HostMCU Debug_Flat.jlink
  - RYZ014A_FWUP_from_HostMCU Debug_Flat.launch
  - Developer Assistance

Figure 3-5. Configuration.xml

2. Click the Generate Project Content icon in the upper right corner of the FSP Configuration window. The required files are extracted from the FSP and the source and header files are added to the project.

3. Select "Project" ➔ "Build Project" from the menu bar or click the Build icon to build the project.

4. Click the debug icon to launch the project. When the project starts, the sample application will be downloaded to EK-RA6M5.

3.5 Launch J-Link RTT Viewer

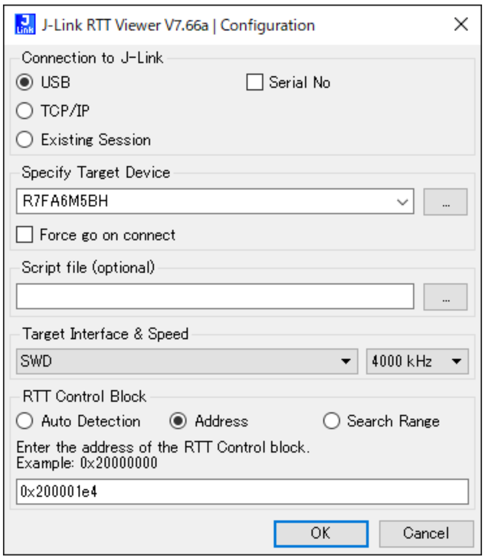
1. Launch J-Link RTT Viewer, set as follows, and click the "OK" button.

   Connection to J-Link : USB

   Specify Target Device : R7FA6M5BH

   Target Interface & Speed : SWD, 4000 kHz

   RTT Control Block : Address, 0x200001e4 (Note)



J-Link RTT Viewer V7.66a | Configuration

Connection to J-Link
● USB □ Serial No
○ TCP/IP
○ Existing Session

Specify Target Device
R7FA6M5BH
□ Force go on connect

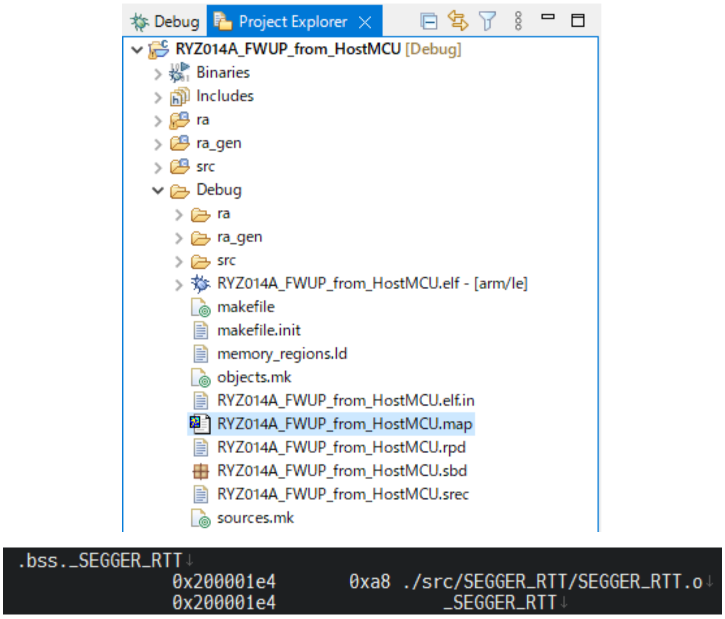
Script file (optional)

Target Interface & Speed
SWD 4000 kHz

RTT Control Block
○ Auto Detection ● Address ○ Search Range
Enter the address of the RTT Control block.
Example: 0x20000000
0x200001e4

Figure 3-6. J-Link RTT Viewer configuration

Note: The RTT Control Block Address sets the ".bss._SEGGER_RTT" section address of the map file generated in the Debug directory.



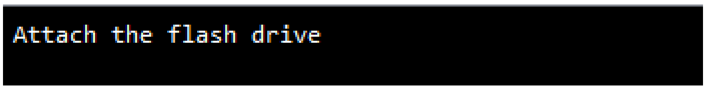
- RYZ014A_FWUP_from_HostMCU [Debug]
  - Binaries
  - Includes
  - ra
  - ra_gen
  - src
  - Debug
    - ra
    - ra_gen
    - src
    - RYZ014A_FWUP_from_HostMCU.elf - [arm/le]
    - makefile
    - makefile.init
    - memory_regions.ld
    - objects.mk
    - RYZ014A_FWUP_from_HostMCU.elf.in
    - RYZ014A_FWUP_from_HostMCU.map
    - RYZ014A_FWUP_from_HostMCU.rpd
    - RYZ014A_FWUP_from_HostMCU.sbd
    - RYZ014A_FWUP_from_HostMCU.srec
    - sources.mk

```
.bss._SEGGER_RTT
                0x200001e4       0xa8 ./src/SEGGER_RTT/SEGGER_RTT.o
                0x200001e4                _SEGGER_RTT
```

Figure 3-7. RTT Control Block Address

3.6 Firmware Upgrade Execution

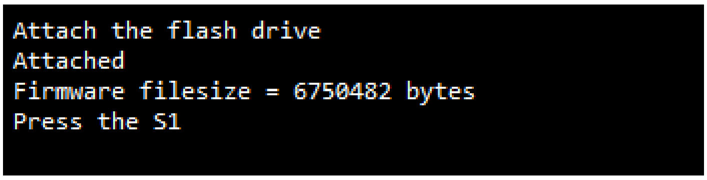
(1) Click the resume icon ▶ in the debug perspective to run the sample application.



```
Attach the flash drive
```

Figure 3-8. Sample application execution

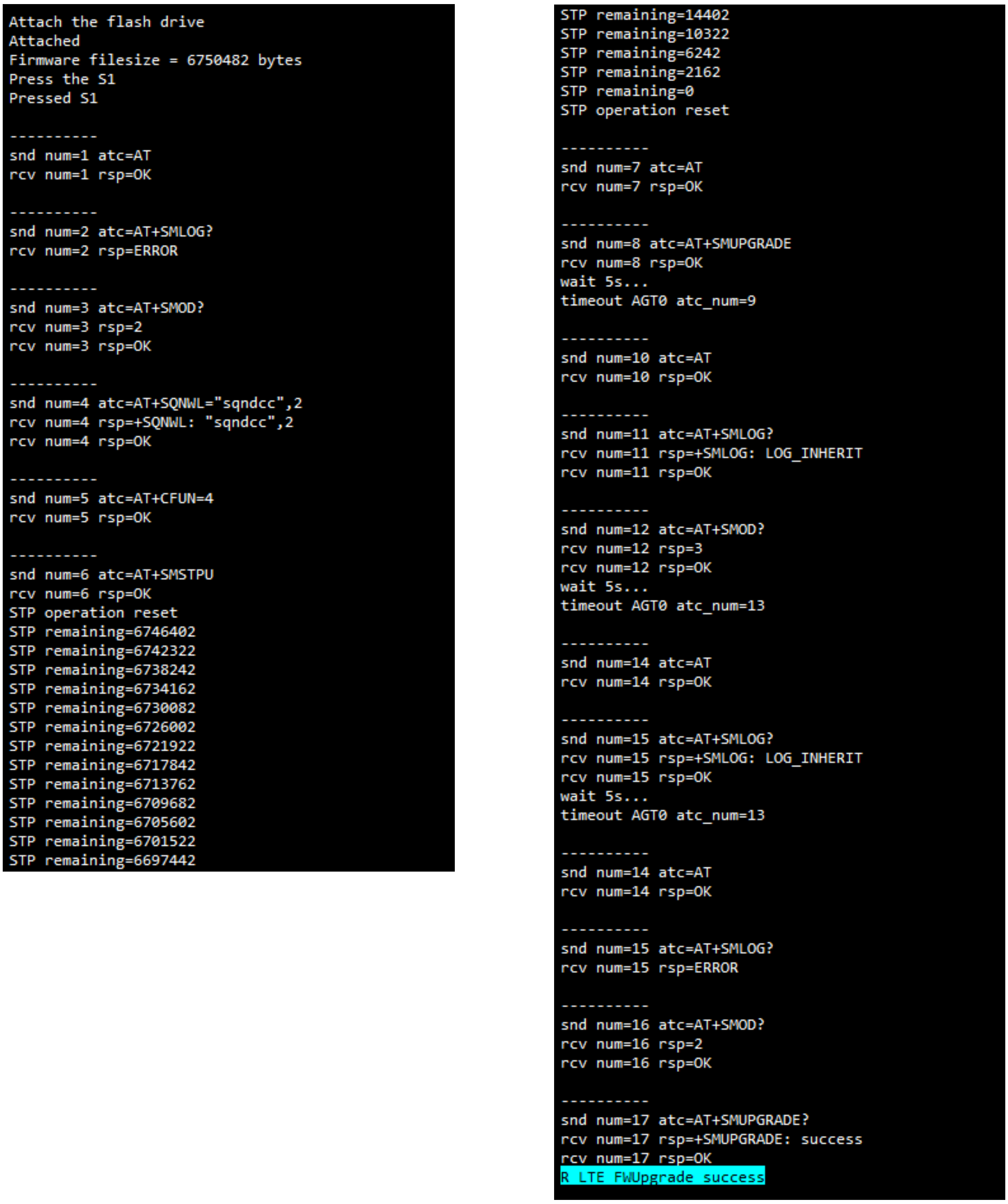
(2) Connect the USB flash drive to the USB micro-B to A conversion cable connected to the USB FS connector of the EK-RA6M5. The RYZ014A firmware file stored on the USB flash drive will be found and the file size will be displayed.



```
Attach the flash drive
Attached
Firmware filesize = 6750482 bytes
Press the S1
```

Figure 3-9. RYZ014A Firmware file detection

(3) Press the S1 button. The firmware upgrade will be executed.



```
Attach the flash drive
Attached
Firmware filesize = 6750482 bytes
Press the S1
Pressed S1

----------
snd num=1 atc=AT
rcv num=1 rsp=OK

----------
snd num=2 atc=AT+SMLOG?
rcv num=2 rsp=ERROR

----------
snd num=3 atc=AT+SMOD?
rcv num=3 rsp=2
rcv num=3 rsp=OK

----------
snd num=4 atc=AT+SQNWL="sqndcc",2
rcv num=4 rsp=+SQNWL: "sqndcc",2
rcv num=4 rsp=OK

----------
snd num=5 atc=AT+CFUN=4
rcv num=5 rsp=OK

----------
snd num=6 atc=AT+SMSTPU
rcv num=6 rsp=OK
STP operation reset
STP remaining=6746402
STP remaining=6742322
STP remaining=6738242
STP remaining=6734162
STP remaining=6730082
STP remaining=6726002
STP remaining=6721922
STP remaining=6717842
STP remaining=6713762
STP remaining=6709682
STP remaining=6705602
STP remaining=6701522
STP remaining=6697442
```

```
STP remaining=14402
STP remaining=10322
STP remaining=6242
STP remaining=2162
STP remaining=0
STP operation reset

----------
snd num=7 atc=AT
rcv num=7 rsp=OK

----------
snd num=8 atc=AT+SMUPGRADE
rcv num=8 rsp=OK
wait 5s...
timeout AGT0 atc_num=9

----------
snd num=10 atc=AT
rcv num=10 rsp=OK

----------
snd num=11 atc=AT+SMLOG?
rcv num=11 rsp=+SMLOG: LOG_INHERIT
rcv num=11 rsp=OK

----------
snd num=12 atc=AT+SMOD?
rcv num=12 rsp=3
rcv num=12 rsp=OK
wait 5s...
timeout AGT0 atc_num=13

----------
snd num=14 atc=AT
rcv num=14 rsp=OK

----------
snd num=15 atc=AT+SMLOG?
rcv num=15 rsp=+SMLOG: LOG_INHERIT
rcv num=15 rsp=OK
wait 5s...
timeout AGT0 atc_num=13

----------
snd num=14 atc=AT
rcv num=14 rsp=OK

----------
snd num=15 atc=AT+SMLOG?
rcv num=15 rsp=ERROR

----------
snd num=16 atc=AT+SMOD?
rcv num=16 rsp=2
rcv num=16 rsp=OK

----------
snd num=17 atc=AT+SMUPGRADE?
rcv num=17 rsp=+SMUPGRADE: success
rcv num=17 rsp=OK
R LTE FWUpgrade success
```

Figure 3-10. Firmware upgrade execution

4. Program processing

Firmware upgrade is performed by calling R_LTE_FWUpgrade function. When the firmware file of RYZ014A stored in the USB flash drive is detected and the S1 button is pressed, AT command processing starts.

In AT command processing, AT commands are sent in the order shown in "4.2.1 AT Command Flow" and responses are received. Then, instruct RYZ014A to execute the firmware upgrade.

Firmware data transfer processing uses the Simple Transfer Protocol (STP) shown in "4.3 Firmware Data Transfer Processing" to send firmware data to RYZ014A.

If the host MCU or RYZ014A is powered off during the firmware upgrade, power them on and then call the R_LTE_FWUpgrade function. Performing the firmware upgrade again will complete the upgrade normally.

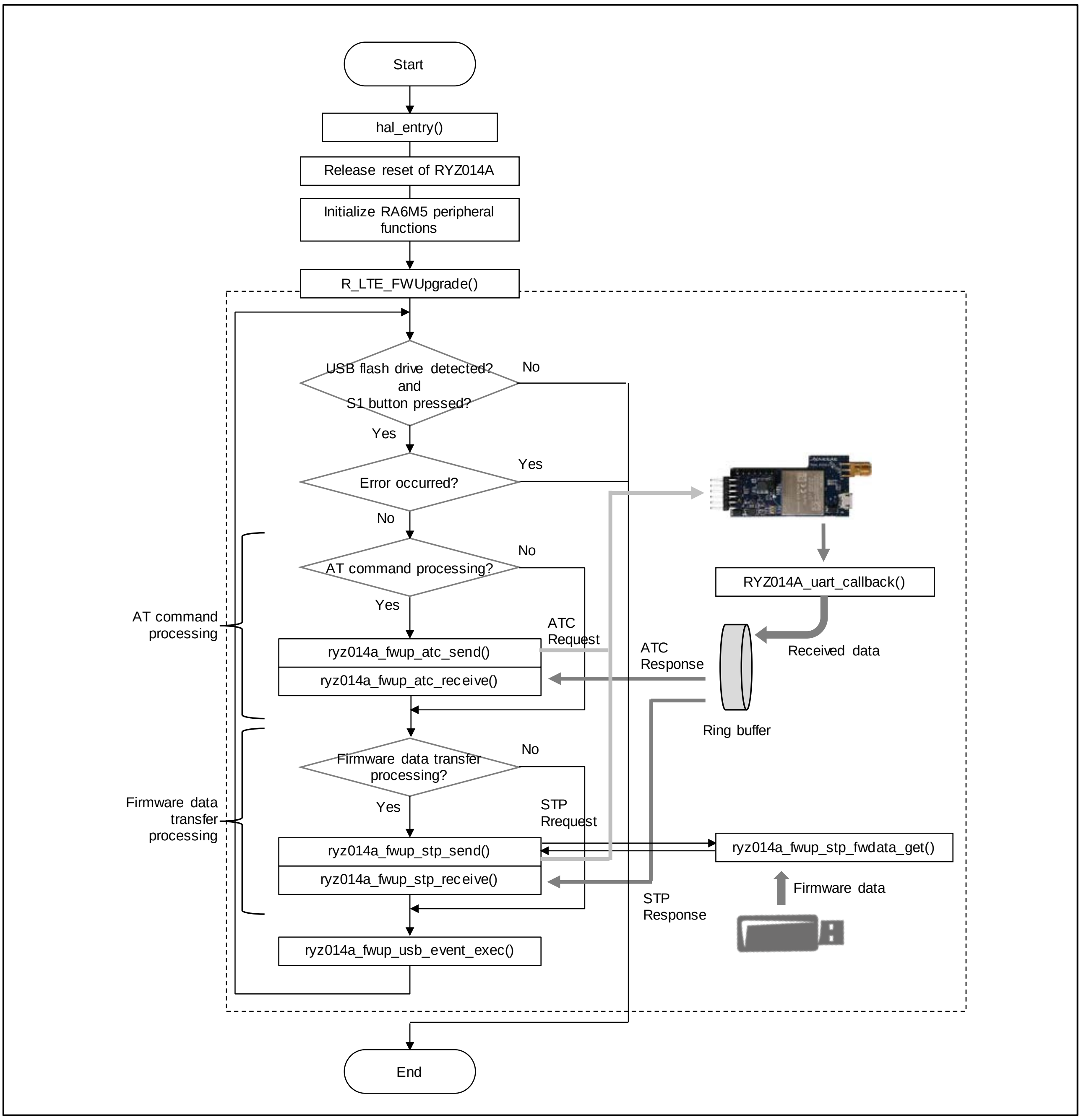


Figure 4-1. Firmware upgrade program processing overview diagram

4.1 API

4.1.1 User API

4.1.1.1 Firmware Upgrade Function

(1) R_LTE_FWUpgrade

Function	Upgrade the firmware of RYZ014A.
Arguments	None
Return	Success : LTE_SUCCESS (Successful firmware upgrade) Error : LTE_ERR_FWUPGRADE (Firmware upgrade failure) LTE_ERR_NOT_FOUND_FWFILE (Firmware file not found) (Refer to "r_lte_ryz014a_fwup.h" for error definition macros.)

4.1.2 Internal Processing Function

4.1.2.1 AT Command Processing Function

(1) ryz014a_fwup_atc_send

Function	Send an AT command to RYZ014A. For the AT command to be sent, refer to "4.2.1 AT Command Flow".
Arguments	None
Return	None

(2) ryz014a_fwup_atc_receive

Function	The response of the AT command is received from RYZ014A and the result is evaluated. For the response to be received, refer to "4.2.1 AT Command Flow". If the following error occurs, re-send the same AT command 5 times. If a normal response cannot be received during retransmission, reset RYZ014A and re-start from the beginning of AT command processing. If the same error occurs again, the firmware upgrade will fail. ● An invalid response was received. ● No response was received for 5 seconds.
Arguments	None
Return	None

4.1.2.2 Firmware Data Transfer processing function

(1) ryz014a_fwup_stp_send

Function	Send an STP request to RYZ014A. For details on the protocol, refer to "4.3 Firmware Data Transfer Processing".
Arguments	None
Return	None

(2) ryz014a_fwup_stp_receive

Function	The STP response is received from RYZ014A and the result is evaluated. For details on the protocol, refer to "4.3 Firmware data Transfer Processing". If the following error occurs, reset the same RYZ014A and re-execute from the beginning of AT command processing. If the same error occurs again, the firmware upgrade will fail. ● An invalid response was received. ● No response was received for 5 seconds.
Arguments	None
Return	None

(3) ryz014a_fwup_stp_fwdata_get

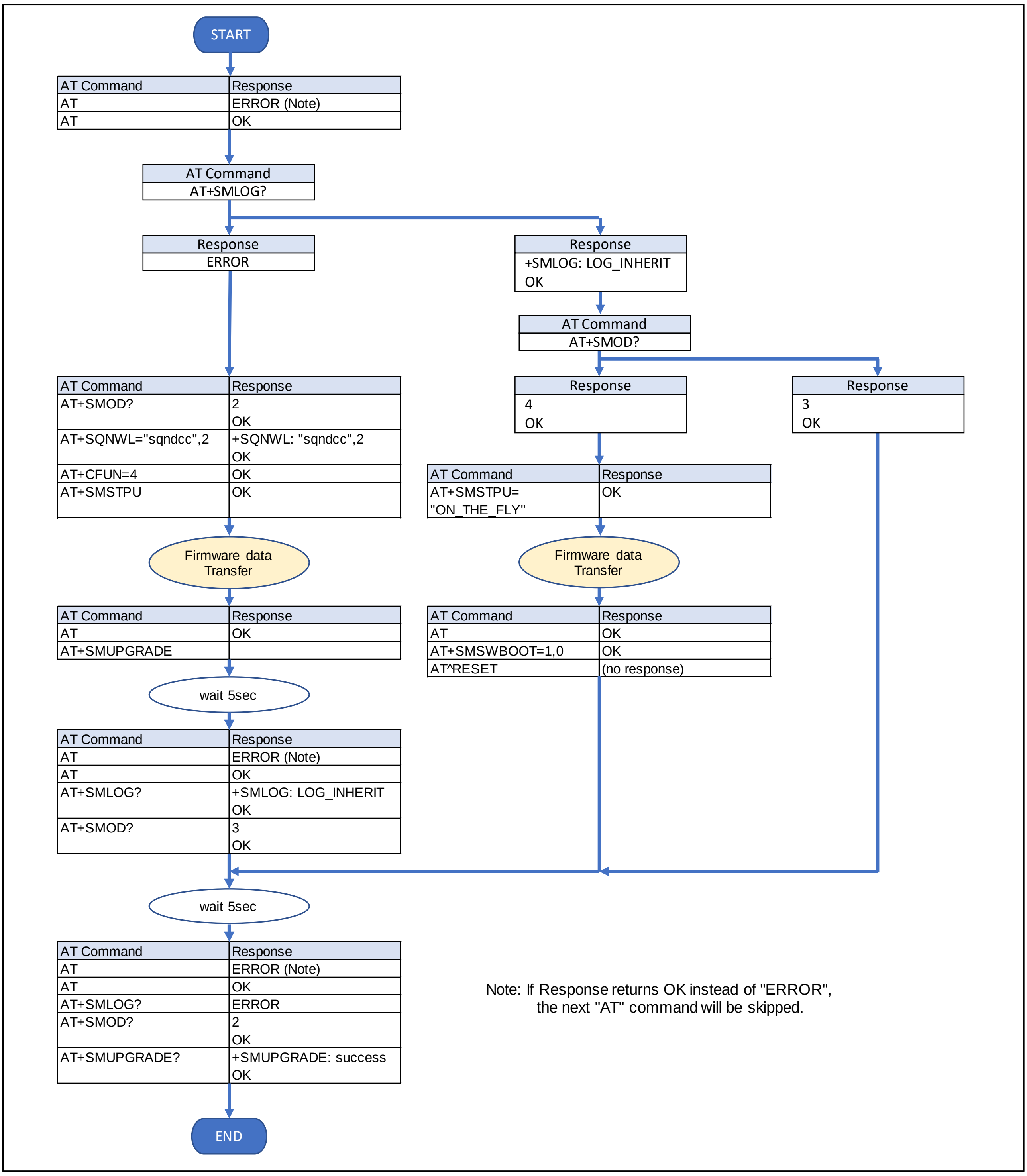
Function	Gets the firmware data of the specified size from the firmware file.
Arguments	● Firmware data storage buffer pointer. (output) ● Data size to get. (input)
Return	None

4.2 AT Command Processing

4.2.1 AT Command Flow

The flow of AT commands used for firmware upgrade is shown below.

The right branch is the recovery route if the RYZ041A is powered down during the upgrade. Normally not executed.



AT Command	Response
AT	ERROR (Note)
AT	OK

AT Command
AT+SMLOG?

Response
ERROR

AT Command	Response
AT+SMOD?	2 OK
AT+SQNWL="sqndcc",2	+SQNWL: "sqndcc",2 OK
AT+CFUN=4	OK
AT+SMSTPU	OK

AT Command	Response
AT	OK
AT+SMUPGRADE	

AT Command	Response
AT	ERROR (Note)
AT	OK
AT+SMLOG?	+SMLOG: LOG_INHERIT OK
AT+SMOD?	3 OK

AT Command	Response
AT	ERROR (Note)
AT	OK
AT+SMLOG?	ERROR
AT+SMOD?	2 OK
AT+SMUPGRADE?	+SMUPGRADE: success OK

Response
+SMLOG: LOG_INHERIT OK

AT Command
AT+SMOD?

Response
4 OK

Response
3 OK

AT Command	Response
AT+SMSTPU= "ON_THE_FLY"	OK

AT Command	Response
AT	OK
AT+SMSWBOOT=1,0	OK
AT^RESET	(no response)

Note: If Response returns OK instead of "ERROR", the next "AT" command will be skipped.

Figure 4-2. AT command flowchart

4.2.2 AT Command

Outlines the AT commands used in the firmware upgrade.

Table 4-1. AT Command

AT Command	Description
AT+SMLOG	Specifies how to output the console log.
AT+SMOD	Returns the boot mode.
AT+SQNWL	Controls the transition to sleep mode.
AT+CFUN	Select the Mobile Termination (MT) functional level.
AT+SMSTPU	Starts Simple Transfer Protocol (STP) transfer.
AT+SMUPGRADE	Execute firmware upgrade.
AT+SMSWBOOT	Boots the device in the specified mode.

4.3 Firmware Data Transfer Processing

The Firmware data transfer process uses Simple Transfer Protocol (STP) to send firmware data to the RYZ014A. Communication is performed using the request packet sent from the host MCU to the RYZ014A and the response packet sent from the RYZ014A to the host MCU.

4.3.1 Packet Configuration

4.3.1.1 STP packet

Shows the configuration of STP packet. The 16 bytes excluding the payload of the following configuration is called the STP header.

Table 4-2. STP packet configuration

Name	Size	Description
signature	4	Request : 0x66617374 ("fast") Response : 0x74736166 ("tsaf")
operation	1	0x00 : reset 0x01 : open session 0x02 : transfer block command 0x03 : transfer block Response operation is OR with 0x80.
session ID	1	set to 0 on reset, set to 1 once session is open.
payload length	2	payload length
transaction ID	4	transaction ID, set to 0 on reset, then incremented after each answer.
full header checksum	2	Request : full header checksum Response : 0
payload checksum	2	payload checksum or 0 if no payload.
payload	-	payload data Refer to "4.3.1.2 Payload" for the data structure.

4.3.1.2 Payload

The Payload configuration is shown below. The Payload used depends on the operation status.

(1) open session operation payload (Response)

This payload is added in the response of open session operation.

Table 4-3. Open session payload configuration

Name	Size	Description
success	1	Indicate if session has been open with success (1) or not (0).
version	1	protocol version, always 1.
max transfer size	2	max size for STP header + payload.

(2) transfer block command operation payload (Request)

This payload is added in the transfer block command operation request.

Table 4-4. Transfer block command payload configuration

Name	Size	Description
block size	2	block size to transfer.

(3) transfer block operation payload (Request)

This payload is added in the transfer block operation request.

Table 4-5. Transfer block operation request payload configuration

Name	Size	Description
firmware data	-	firmware data. size = max transfer size - STP header

(4) transfer block operation payload (Response)

This payload is added in the transfer block operation response.

Table 4-6. Transfer block command response payload configuration

Name	Size	Description
residue	2	always 0.

4.3.2 Operation Flow

This section describes the operation flow when sending firmware data by STP.

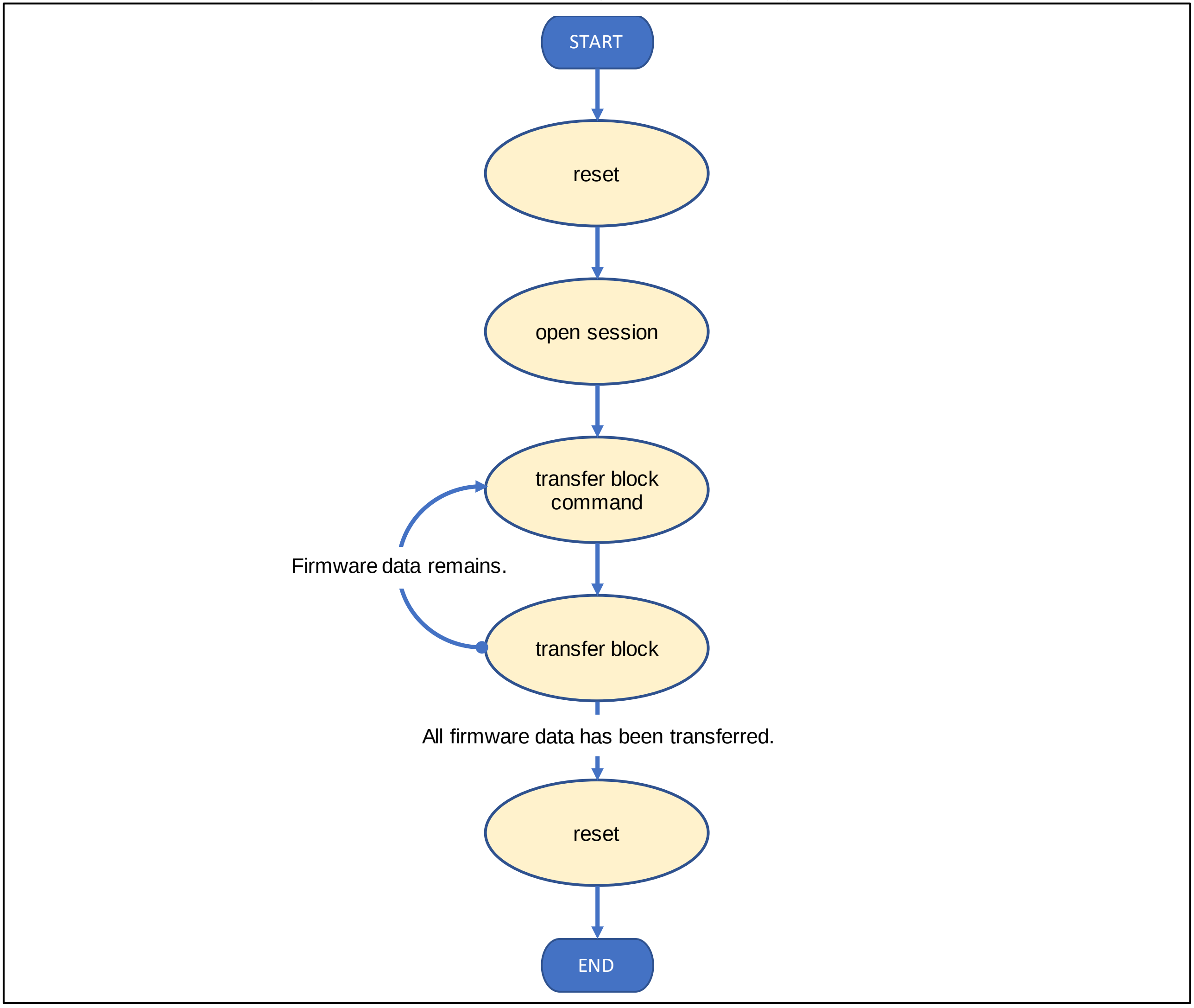


Figure 4-3. Operation flowchart

(1) reset

- Reset an open remote session.
- Reset sid and tid to 0.
- There is no request payload.
- There is no response payload.

(2) open session

- Open session with peer and get the session parameters "open session operation payload (Response)".
- Set sid and tid to 1. Increment tid once the response is received.
- There is no request payload.
- Get the session parameters "open session operation payload (Response)" in the resonse.

(3) transfer block command

- Instructs RYZ014A to receive firmware data. Increment tid once the response is received.
- Send "transfer block command operation payload (Request)" in the request.
- There is no response payload.

(4) transfer block

- Send the firmware data to RYZ014A. Increment tid once the response is received.
- Sends the firmware data "transfer block operation payload (Request)" in the request.
- Get a "transfer block operation payload (Response)" in the response.

5. Appendix

5.1 AT Commands Definition

This section describes the AT commands used in the firmware upgrade.

(1) AT+SMSWBOOT

Description:

This command forces the device to boot in mode <mode> (FFF, FFH, Updater or Recovery).

Syntax:

Command	Possible Response(s)
AT+SMSWBOOT=<mode>[,<reboot>]	OK

Parameters:

<mode>: integer 0, 1, 2 or 3. Device start-up mode at next boot

0: FFH

1: FFF

2: UPDATER

3: RECOVERY

<reboot>: integer 0 or 1. Automatic device reboot after <mode> change

0 (default): no reboot

(2) AT+SMOD

Description:

This command returns the boot mode.

Syntax:

Command	Possible Response(s)
AT+SMOD	<mode>

Parameters:

<mode>: Integer. Device start-up mode at next boot

0: FFH

1: FFF

2: UPDATER

3: RECOVERY

4: OTHER

(3) AT+SMUPGRADE

Description:

AT+SMUPGRADE parses the .dup file and directly flashes the data into the corresponding regions depending on which boot mode the module is:

- FFH: Upgrade all regions and filesystem
- RECOVERY: Upgrade all regions and filesystem
- FFF: Upgrade UPDATER and BOOTROM regions
- UPDATER: Upgrade FFF region and filesystem

When in FFF, all authorized regions are upgraded before the module reboots in UPDATER mode to finish the execution of the .dup file. Once the UPDATER mode is over, the module reboots in FFF mode. The SFU tool takes care of both steps, its use is highly recommended for any upgrade.

Syntax:

Command	Possible Response(s)
AT+SMUPGRADE	-
AT+SMUPGRADE?	Upgrade report

(4) AT+SMLOG

Description:

Available in manufacturing mode (AT+CFUN=5)

Forces console logs to be printed on the specified UART (regardless of their programmed "console" functionality).

Syntax:

Command	Possible Response(s)
AT+SMLOG=<log>	OK
AT+SMLOG?	+SMLOG=<log>
AT+SMLOG=?	+SMLOG [list of supported <log>]

Parameters:

<log>: String. Chooses the UART the console log is directed to.

- LOG_DISABLE: The logs are discarded
- LOG_INHERIT: The logs are printed on the UART configured as ‘console’ (default)
- LOG_FORCE_UART0: The logs are printed on UART 0
- LOG_FORCE_UART1: The logs are printed on UART 1
- LOG_FORCE_UART2: The logs are printed on UART 2

(5) AT+SMSTPU

Description:

This command starts a STP transfer, waiting for an FFF image containing upgrade images.

Syntax:

Command	Possible Response(s)
AT+SMSTPU[="ON_THE_FLY"]	-

Parameters:

none: Normal transfer

ON_THE_FLY: Recovery transfer

(6) AT^RESET

Description:

This command performs an hardware reset.

Syntax:

Command	Possible Response(s)
AT^RESET	Device is reset
-	+SHUTDOWN ... +SYSSTART

(7) AT+SQNWL

Description:

This command manages resources wake locks to indicate that a client application running on the Host CPU needs to secure full and immediate availability of some device resources, which implies to prevent the considered resources to enter sleep mode.

Currently, the wake-lockable system resources are:

- CPU and external interfaces (UART, GPIO)
- Device memory (RAM)

The Set command is used to set and release the wake locks based on the resource identified by the bitmask <wl_mask>. To set the wake locks and prevent the sleep mode for the specified resource(s), <wl_mask> bit(s) should be set to 1. To release the wake locks and allow sleep mode for the specified resource(s), <wl_mask> bit(s) should be set to 0.

Note: The wake locks configuration is volatile. It is lost at reboot.

The Set command entered with bitmask omitted will return the wake lock defined by the <app> client application.

The Read command returns the list of client applications using wake locks and lock status.

Caution: It is very important release wake lock as soon as possible to avoid running down the device's battery excessively. Each application

<app> setting a wake lock to 1 should explicitly reset it to 0 when the need for resource availability is not present anymore.

Syntax:

Command	Possible Response(s)
AT+SQNWL=<app>[,<wl_mask>]	+CME ERROR:<err> +SQNWL:<app>,<wl_mask> OK
AT+SQNWL?	+SQNWL:<app1>,<mask1> [<CR><LF>...<CR><LF>+SQNWL: <appN>,<maskN>[...]] OK
AT+SQNWL=?	+SQNWL:””,(0-3) OK

Parameters:

app

- Client application name, string.

wl_mask

- Bitmask as integer in range [0-3] identifying the resource to keep available. Bitis set to 1 to keep the resource available and prevent sleep mode, or reset to 0 to release.

wl_mask

Value	Description
0	Default value. No system resource locked
Bit 0 (0x01)	Keep CPU and external interfaces (UART, GPIO) active. Prevents the CPU and external interfaces to enter sleep mode.
Bit 1 (0x02)	Keep device’s RAM memory active. Prevents the device’s RAM memory to enter sleep mode.
Bits 0 and 1 (0x03)	Keep CPU, external interfaces (UART, GPIO) and device’s RAM memory active. Prevents the

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	Jun.29.22	-	Initial Release

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact information

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

Trademarks

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