RYZ024A and RX MCU

Firmware Upgrade from Host MCU for RX

Introduction

This document describes a sample application that upgrades the RYZ024A firmware from the host MCU.

This sample application works in a configuration that uses the CK-RX65N board with RX65N as the host MCU and connects the Pmod™ Expansion Board for RYZ024A to the Pmod connector. Store the RYZ024A firmware file on a USB flash drive and connect it to the USB Full Speed connector on the CK-RX65N board. Transfer the firmware from the USB flash drive to the RYZ024A for upgrade.

This sample application works on bare metal.

Target Device

RYZ024A

CK-RX65N

Related Documents

- Renesas LTE Cat-M1 Cellular IoT Module RYZ024A Pmod™ Expansion Board (R21QS0004)
- RYZ024 Module System Integration Guide (R19AN0101)
- RX65N Group, RX651 Group User's Manual: Hardware (R01UH0590)
- Cloud Kit for RX65N Microcontroller Group CK-RX65N v1 User's Manual (R20UT5100)
- Firmware Integration Technology User's Manual (R01AN1833)

Pmod™ is registered to Digilent Inc.

Contents

1. Overview 3
2. Operating Environment 4
2.1 Hardware 4
2.2 Software 4
2.3 Microcontroller Peripheral Functions 5
2.4 FIT Modules 6
2.5 Directory/File Structure 7
3. How to Use this project 8
3.1 Preparation of RYZ024A firmware 8
3.2 CK-RX65N jumper setting 8
3.3 Import of project 8
3.4 Build and Download 11
3.5 Launch Tera Term 12
3.6 Firmware Upgrade Execution 12
4. Program processing 14
4.1 API 15
4.1.1 User API 15
4.1.1.1 Firmware Upgrade Function 15
4.1.2 Internal Processing Function 15
4.1.2.1 AT Command Processing Function 15
4.1.2.2 Firmware Data Transfer processing function 16
4.2 AT Command Processing 17
4.2.1 AT Command Flow 17
4.2.2 AT Command 19
4.3 Firmware Data Transfer Processing 20
4.3.1 Packet Configuration 20
4.3.1.1 STP packet 20
4.3.1.2 Payload 20
4.3.1.3 Checksum 21
4.3.2 Operation Flow 22
5. Appendix 26
5.1 Change operating parameters 26
5.1.1 Differential firmware upgrade timeout time 26
5.1.2 Software wait time until AT command is sent 26
5.2 AT Commands Definition 27
Revision History 32

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 RYZ024A, 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 CK-RX65N board with RX65N is used as the host MCU and the Pmod Expansion Board for RYZ024A is connected to the Pmod connector. Store the RYZ024A firmware file on a USB flash drive and connect it to the USB Full Speed connector on the CK-RX65N board. Transfer the firmware from the USB flash drive to the RYZ024A for upgrade.

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

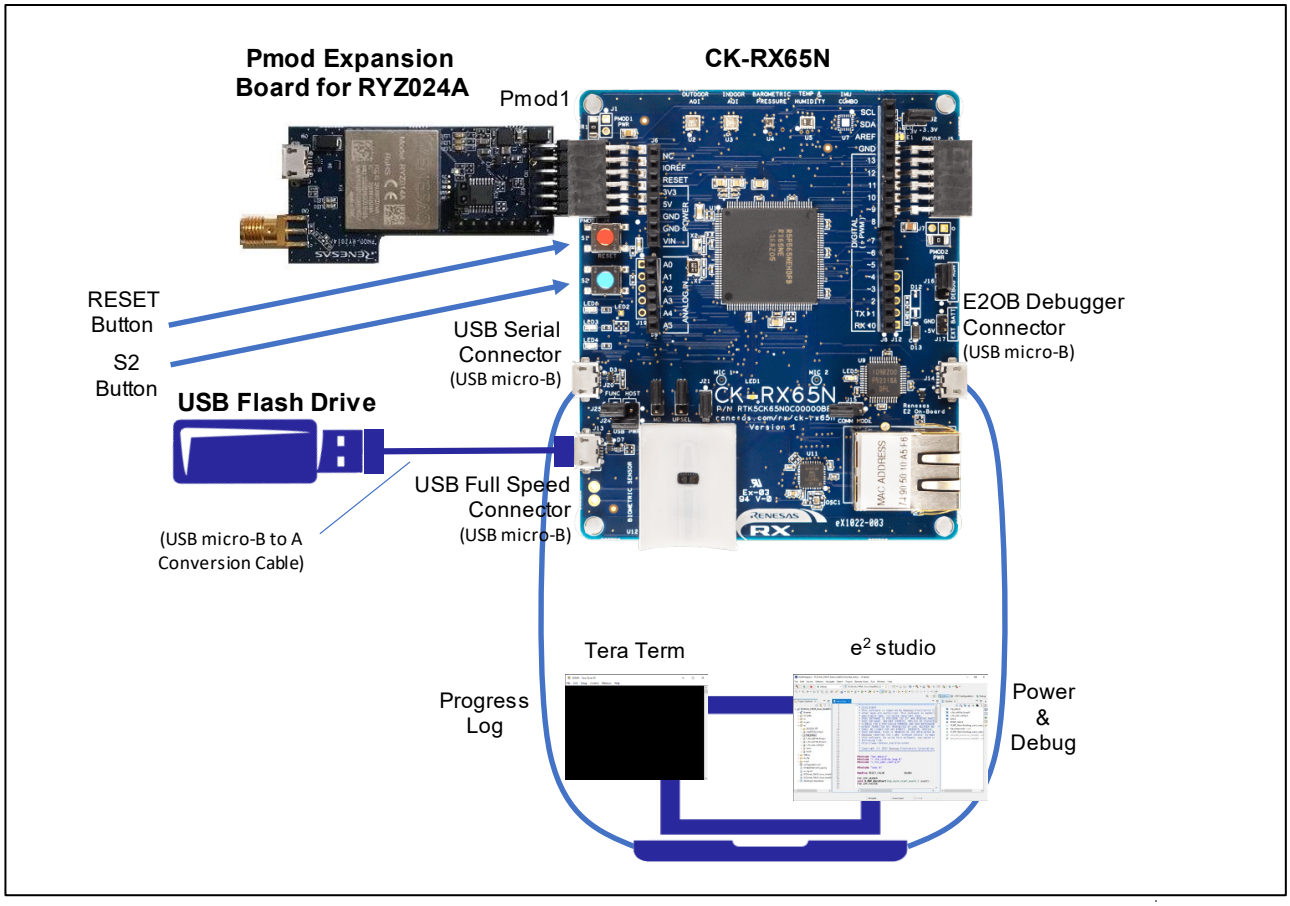

Figure 1-1 Firmware upgrade sample application configuration diagram

2. Operating Environment

2.1 Hardware

The hardware requirements used in the sample application are shown in "Table 2-1 Hardware requirements".

Table 2-1 Hardware requirements

Hardware	Description
Pmod Expansion Board for RYZ024A	RTKYZ024A0B00000BE
CK-RX65N	RTK5CK65N0S04000BE
USB Flash Drive	10MB or more
Windows 10 PC	---
USB Cable	• for CK-RX65N E2OB Debugger(Power & Debug) (micro-B) • for CK-RX65N USB Serial Connecter(Progress Log) (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 in "Table 2-2 Software requirements".

Table 2-2 Software requirements

Software	Version
e^2 studio	2023-10
CC-RX Compiler	3.05
RX Driver Package	V1.41
Renesas Flash Programmer	V3.11 or later
Tera Term	V4.106 or later (It is used in the execution log display for operation check.)

2.3 Microcontroller Peripheral Functions

The microcontroller peripheral functions used in the sample application are shown in "Table 2-3 Microcontroller peripheral functions".

Table 2-3 Microcontroller peripheral functions

Function Name		Description
Serial Communication Interface	SCI6(UART6)	Pmod1: UART communication with RYZ024A Baudrate :115200 bps Data length :8 bit Parity :none Stop bit :1 bit Flow control :Hardware CTS/Software RTS RTS : P02 CTS : PJ3
	SCI5(UART5)	USB Serial Connector: Display progress log in Tera Term Baudrate :115200 bps Data length :8 bit Parity :none Stop bit :1 bit Flow control :none
Compare Match Timer W	CMTW0	UART communication timeout with RYZ024A
	CMTW1	LED blinking cycle
USB2.0 FS Host	USB2.0 FS Host	USB Full Speed Connector: Communication with the USB flash drive where the firmware file is stored.
I/O Port	P55	Reset pin control of RYZ024A
	P02	RTS signal used for UART communication with RYZ024A.
	PJ3	CTS signal used for UART communication with RYZ024A.
	PA3	User LED Blue
	P25	User LED Red
External Interrupt Request	IRQ1	External interrupt request of user button S2

2.4 FIT Modules

The FIT modules used in the sample application are shown in "Table 2-4 FIT modules". The FIT module can be obtained automatically by generating code with Smart Configurator.

Table 2-4 FIT modules

Function	Component		Version
BSP	Board Support Package	r_bsp	7.41
Port	Config_PORT	Config_PORT	2.4.1
SCI	SCI Driver	r_sci_rx	4.90
	Byte-based circular buffer library	r_byteq	2.10
	GPIO Driver	r_gpio_rx	5.00
IRQ	IRQ Driver	r_irq_rx	4.40
USB	USB Basic	r_usb_basic	1.41
	USB Host Mass Storage Class	r_usb_hmsc	1.41
File System	Open Source Fat File System	r_tfat_rx	4.02
	Memory Driver Interface for Open Source FAT File System	r_tfat_driver_rx	2.20
	CMT driver	r_cmt_rx	5.60
	Generic system timer for RX MCUs using CMT module	r_sys_timer_rx	1.01
Timer	CMTW Driver	r_cmtw_rx	2.80

2.5 Directory/File Structure

Directory/file structure of sample application are shown in "Table 2-5 Directory/File structure".

Table 2-5 Directory/File structure

Directory/ File Structure				Description
RYZ024A_FWUP_from_HostMCU\		.cproject .project RYZ024A_FWUP_from_HostMCU HardwareDebug.launch RYZ024A_FWUP_from_HostMCU.rcpc RYZ024A_FWUP_from_HostMCU.scfg		Project files.
	.settings\	com.renesas.cdt.managedbuild.renesas.ccrx.prefs com.renesas.hardwaredebug.rx.e2.PerfAnalysisSettings.xml com.renesas.smc.e2studio.qe.xml com.renesas.smc.generationsetting.properties CoverageSetting.xml DebugVirtualConsoleSetting.xml Dependency_Scan_Preferences.prefs e2studio_project.prefs IORegisterSetting.xml org.eclipse.cdt.core.prefs org.eclipse.core.resources.prefs renesasPGModel.xml		e^2 studio setting files.
	src\	main.c r_lte_ryz024a_fwup.c r_lte_ryz024a_fwup.h r_lte_ryz024a_fwup_console.c r_lte_ryz024a_fwup_console.h r_lte_ryz024a_fwup_hal_data.c r_lte_ryz024a_fwup_hal_data.h r_lte_user_config.h	Firmware upgrade sample application source codes.	
		smc_gen\	FIT module source codes	

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.

This sample application executes without connecting to an LTE network.

3.1 Preparation of RYZ024A firmware

There are two types of RYZ024A firmware files. Full firmware file, which upgrades the entire firmware, and differential firmware file, which upgrades the difference between the current version and the new version. Change one of the firmware files(Note) to the following name and store it in the root directory of the USB memory.

Full firmware file : `ryz024a_firmware_full.dup`

Differential firmware file : `ryz024a_firmware_diff.dup`

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

3.2 CK-RX65N jumper setting

The USB Full Speed interface of CK-RX65N is used as a USB Host to connect a USB flash drive. Set the jumper as follows according to "RX65N Group Cloud Kit for RX65N Microcontroller Group CK-RX65N v1 User's Manual" (R20UT5100).

J25: Jumper on pins 1-2 (USB Host)

J24: Closed (Bus-powered operation)

Other jumpers should be set as described in "Table 2. Default Jumper Settings" in "RX65N Group Cloud Kit for RX65N Microcontroller Group CK-RX65N v1 User's Manual" (R20UT5100).

3.3 Import of project

The steps to import a sample application project into e² studio are shown below.

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

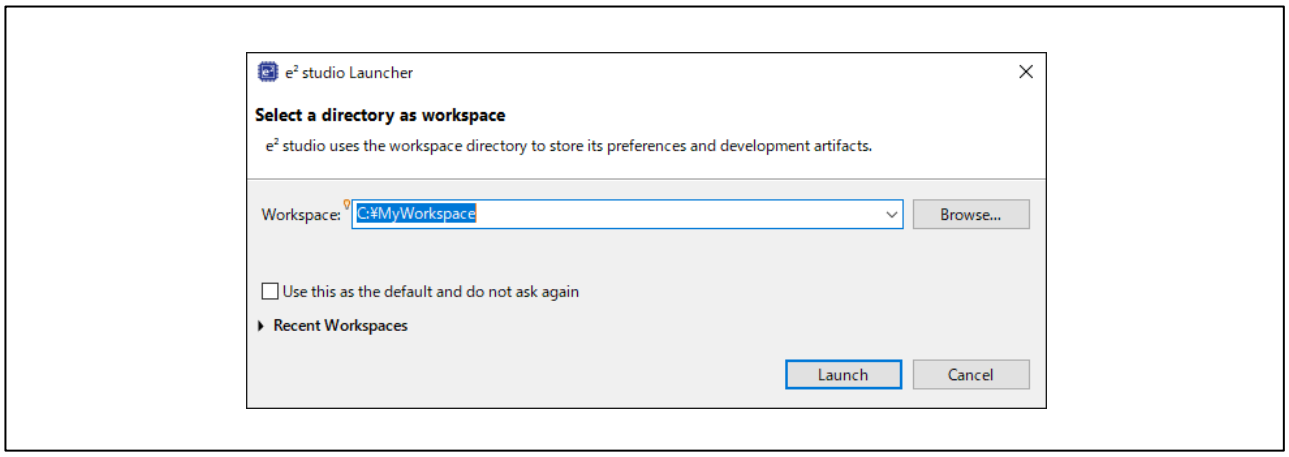

Figure 3-1 Workspace selection

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

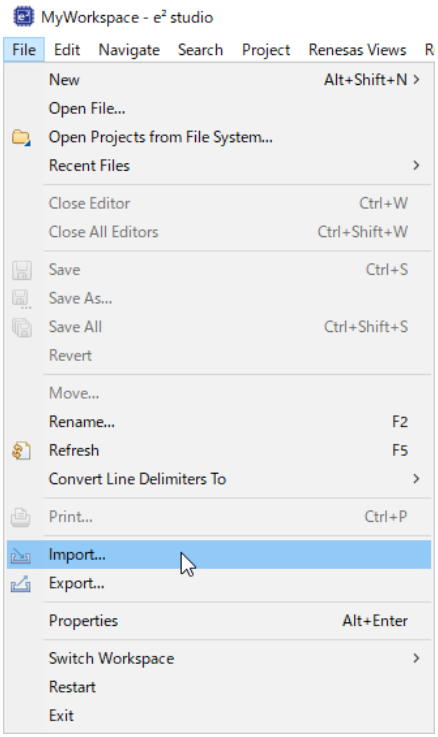

Figure 3-2 File menu

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

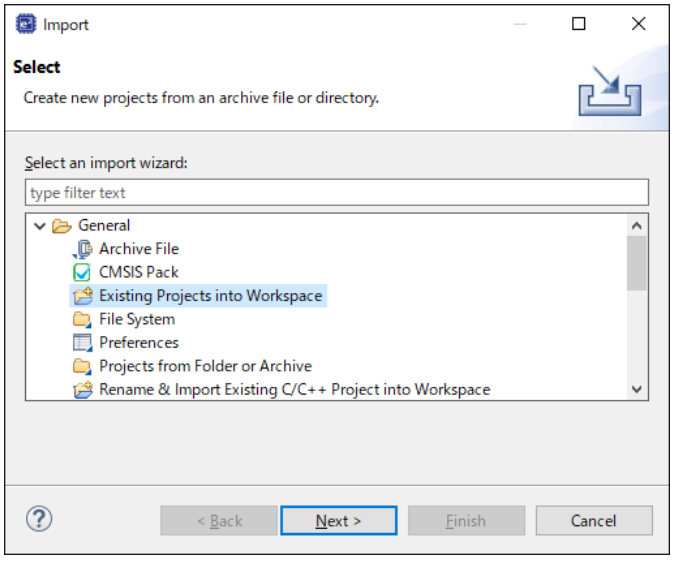

Figure 3-3 Import wizard selection

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

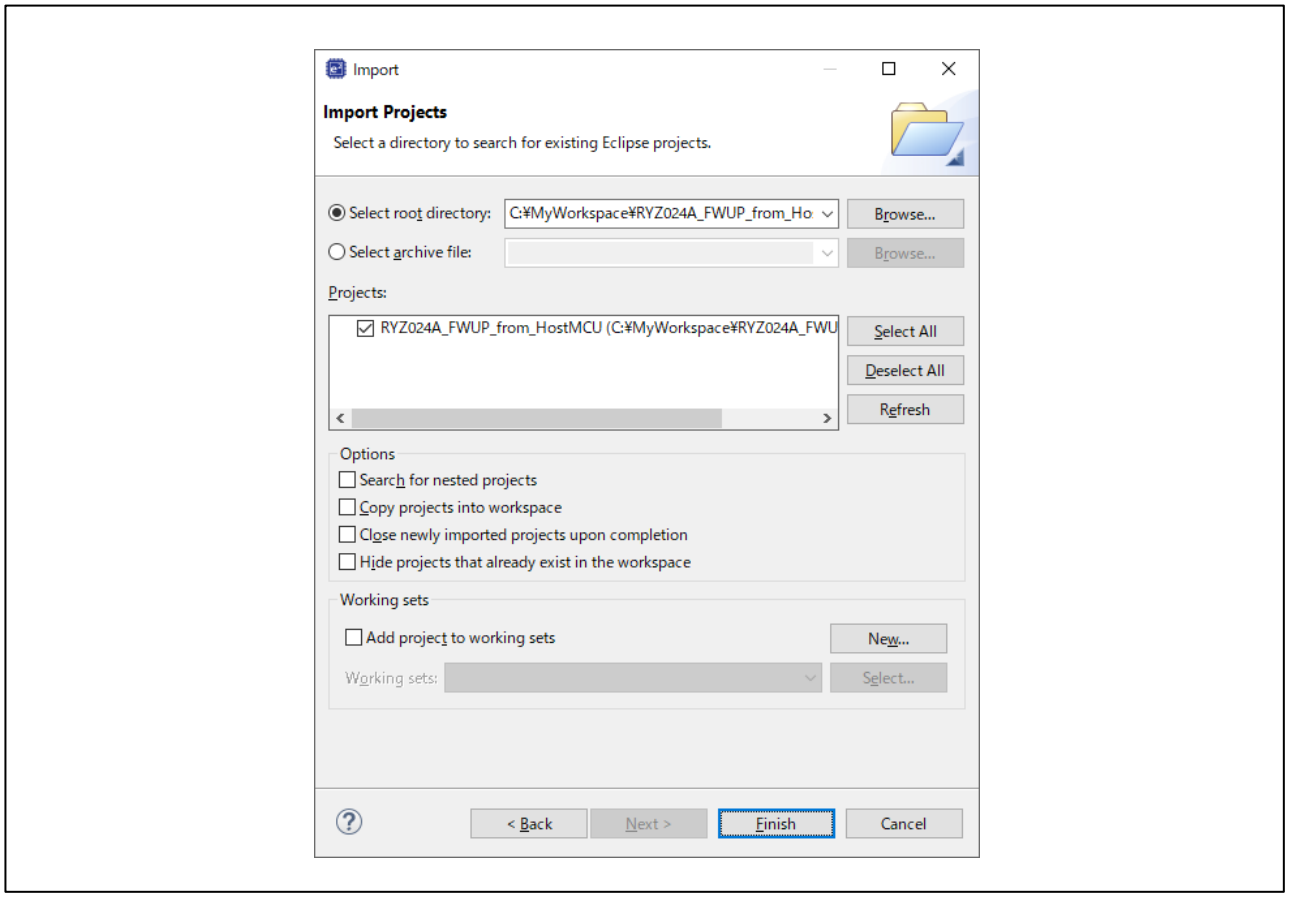

Figure 3-4 Import project

3.4 Build and Download

1. Double-click RYZ024A_FWUP_from_HostMCU.scfg to open the Smart Configurator window.

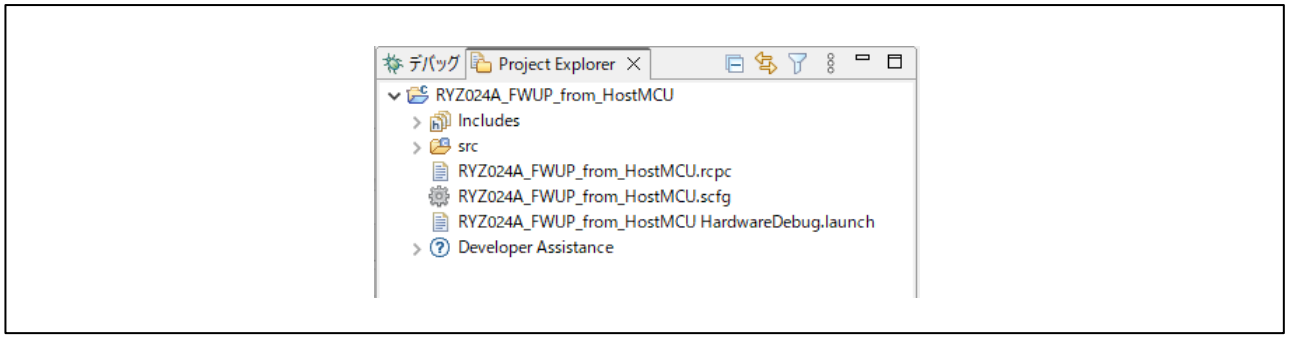

Figure 3-5 configuration.xml

2. Click the Generate Code icon

in the upper right corner of the Smart Configurator window. The required files are extracted from the FIT 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 CK-RX65N.

3.5 Launch Tera Term

1. Launch Tera Term and Select "Setup" ➔ "Serial port" from the menu bar.
2. Set the Serial port by referring to SCI5 (UART5) in "Table 2-3 Microcontroller peripheral functions".

3.6 Firmware Upgrade Execution

(1) Click the

icon on the top right of e^2 studio to open the debug perspective. Click the resume icon in the debug perspective to run the sample application.

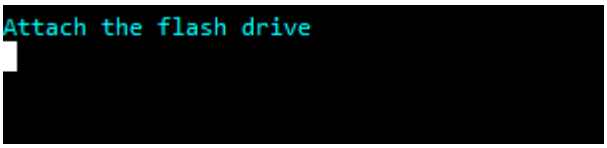

Figure 3-6 Sample application execution

(2) Connect the USB flash drive to the USB micro-B to A conversion cable connected to the USB Full Speed connector of the CK-RX65N. The RYZ024A firmware file stored on the USB flash drive will be found and the file size will be displayed.

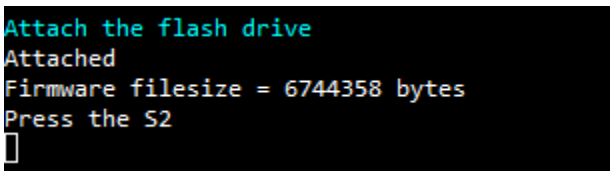

Figure 3-7 RYZ024A Firmware file detection

(3) Press the S2 button on CK-RX65N. The firmware upgrade will be executed. The firmware upgrade is successfully executed and "R_LTR_FWUpgrade success" is displayed.
(Below are the results of running an upgrade using the full firmware file.)

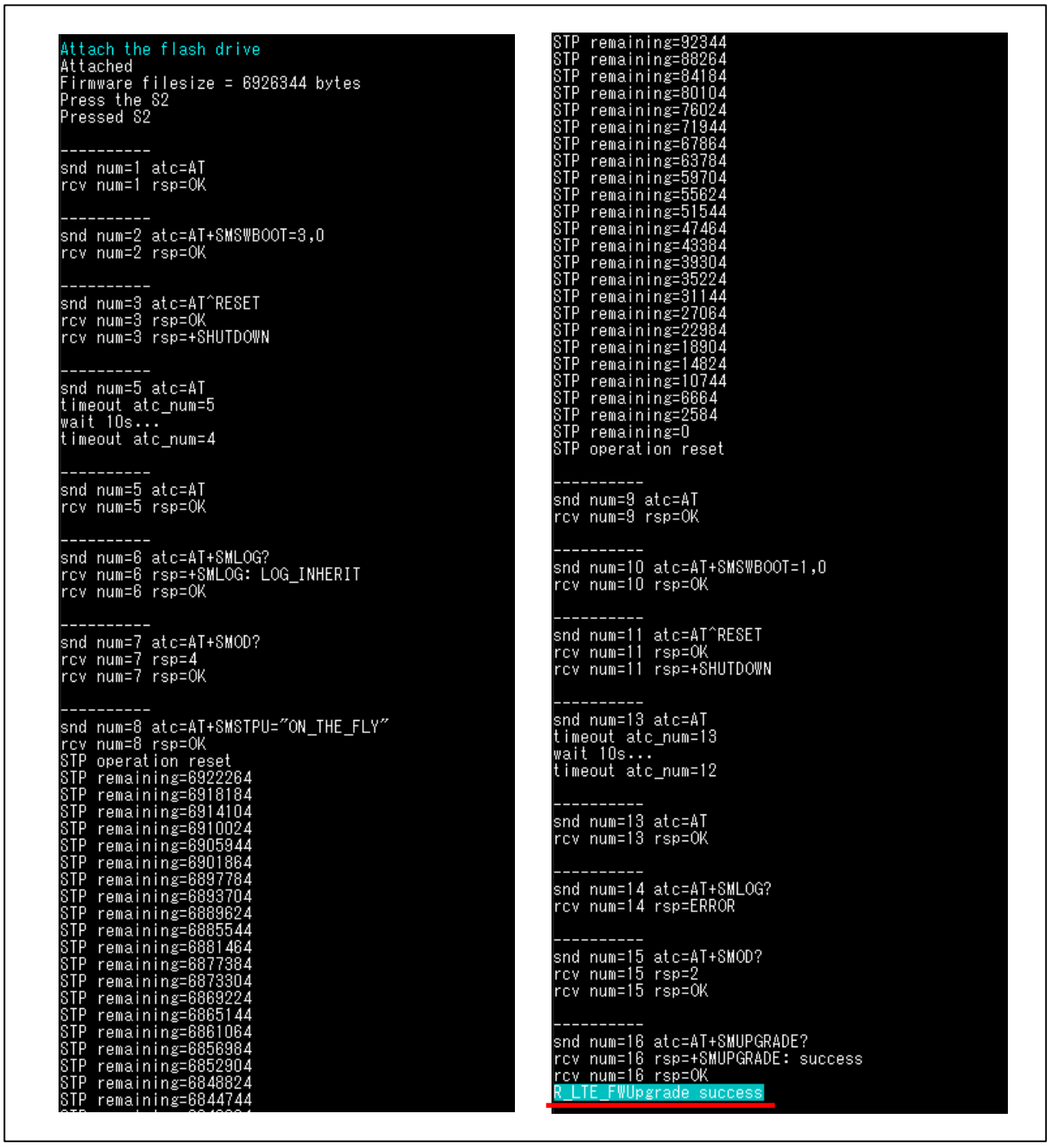

Figure 3-8 Firmware upgrade execution

4. Program processing

Firmware upgrade is performed by calling R_LTE_FWUpgrade function. When the firmware file of RYZ024A stored in the USB flash drive is detected and the S2 button is pressed, AT command (ATC) 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 RYZ024A 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 RYZ024A.

If the host MCU or RYZ024A 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.

Refer to "3 How to Use this project" for how to execute the firmware upgrade using the sample application.

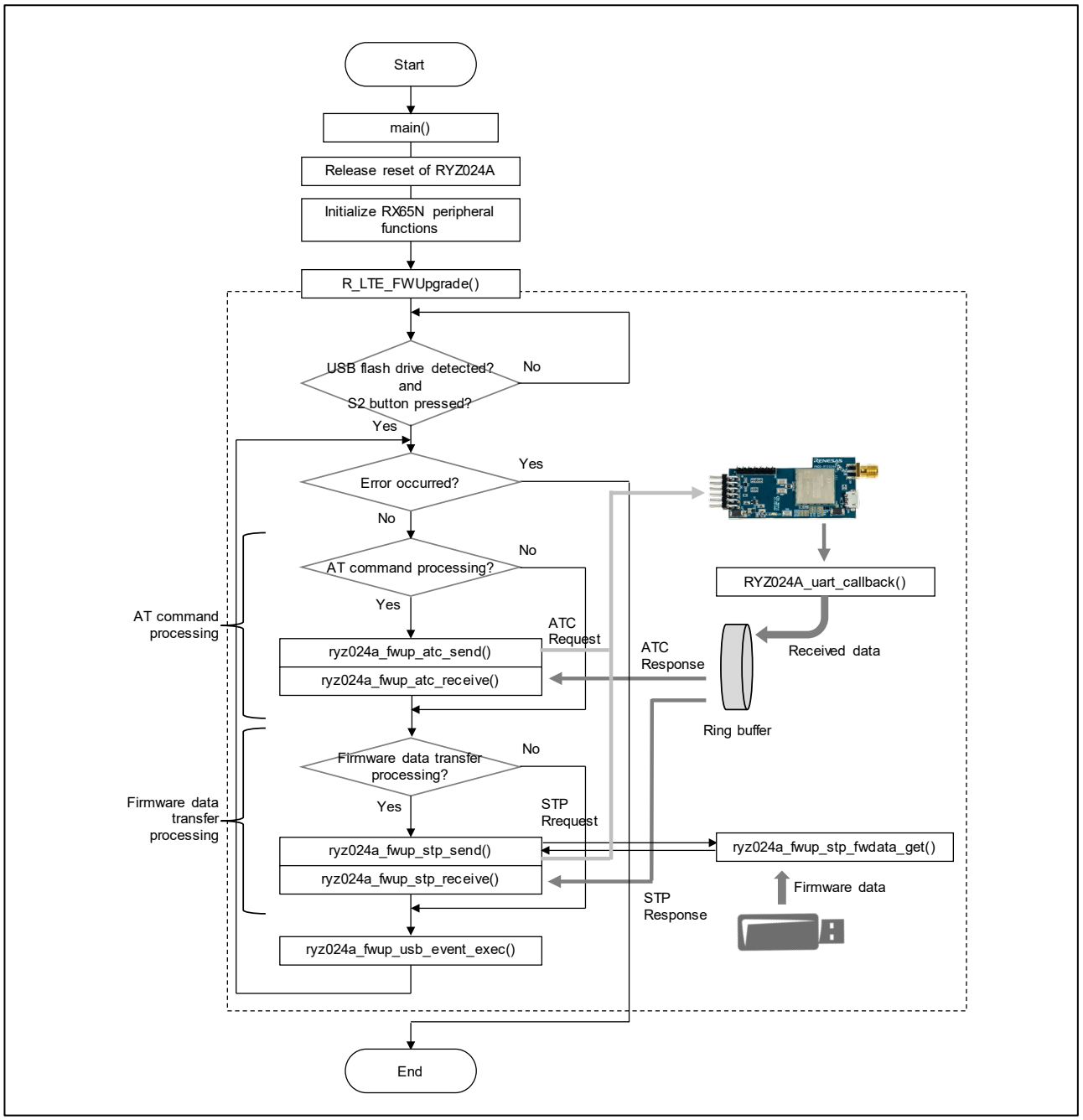

Figure 4-1 Firmware upgrade program processing overview diagram

4.1 API

4.1.1 User API

4.1.1.1 Firmware Upgrade Function

(1) R_LTE_FWUpgrade

Function	Upgrade the firmware of RYZ024A.
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_RYZ024A_fwup.h" for error definition macros.)

4.1.2 Internal Processing Function

4.1.2.1 AT Command Processing Function

(1) RYZ024A_fwup_atc_send

Function	Send an AT command to RYZ024A. For the AT command to be sent, refer to "4.2.1 AT Command Flow".
Arguments	None
Return	None

(2) RYZ024A_fwup_atc_receive

Function	The response of the AT command is received from RYZ024A 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 RYZ024A 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 10 seconds.
Arguments	None
Return	None

4.1.2.2 Firmware Data Transfer processing function

(1) RYZ024A_fwup_stp_send

Function	Send an STP request to RYZ024A. For details on the protocol, refer to "4.3 Firmware Data Transfer Processing".
Arguments	None
Return	None

(2) RYZ024A_fwup_stp_receive

Function	The STP response is received from RYZ024A 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 RYZ024A 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 10 seconds.
Arguments	None
Return	None

(3) RYZ024A_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 AT command flow for full firmware and differential firmware upgrades is shown below.

(1) Full firmware

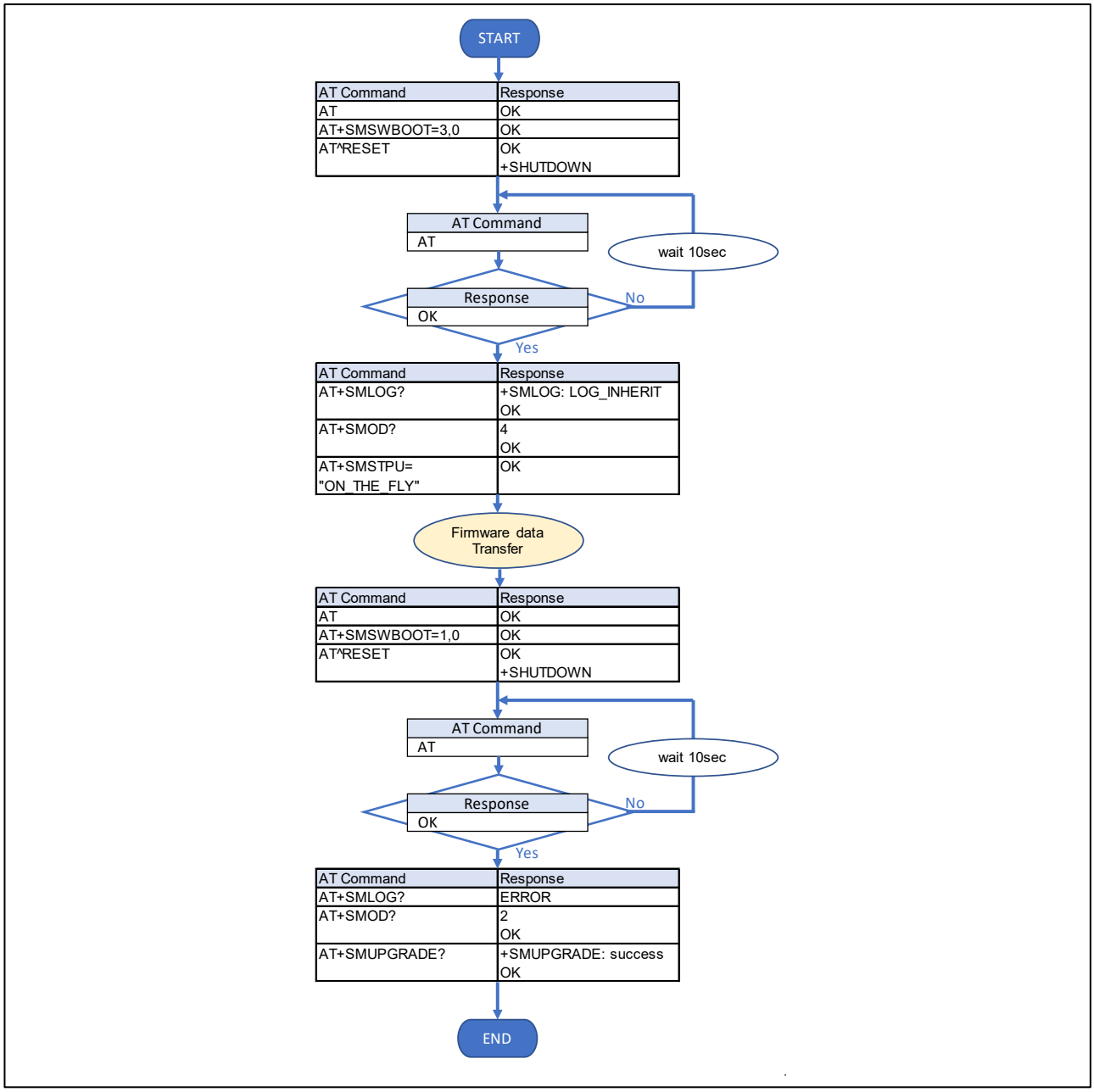

Figure 4-2 AT command flowchart (Full firmware)

(2) Differential firmware

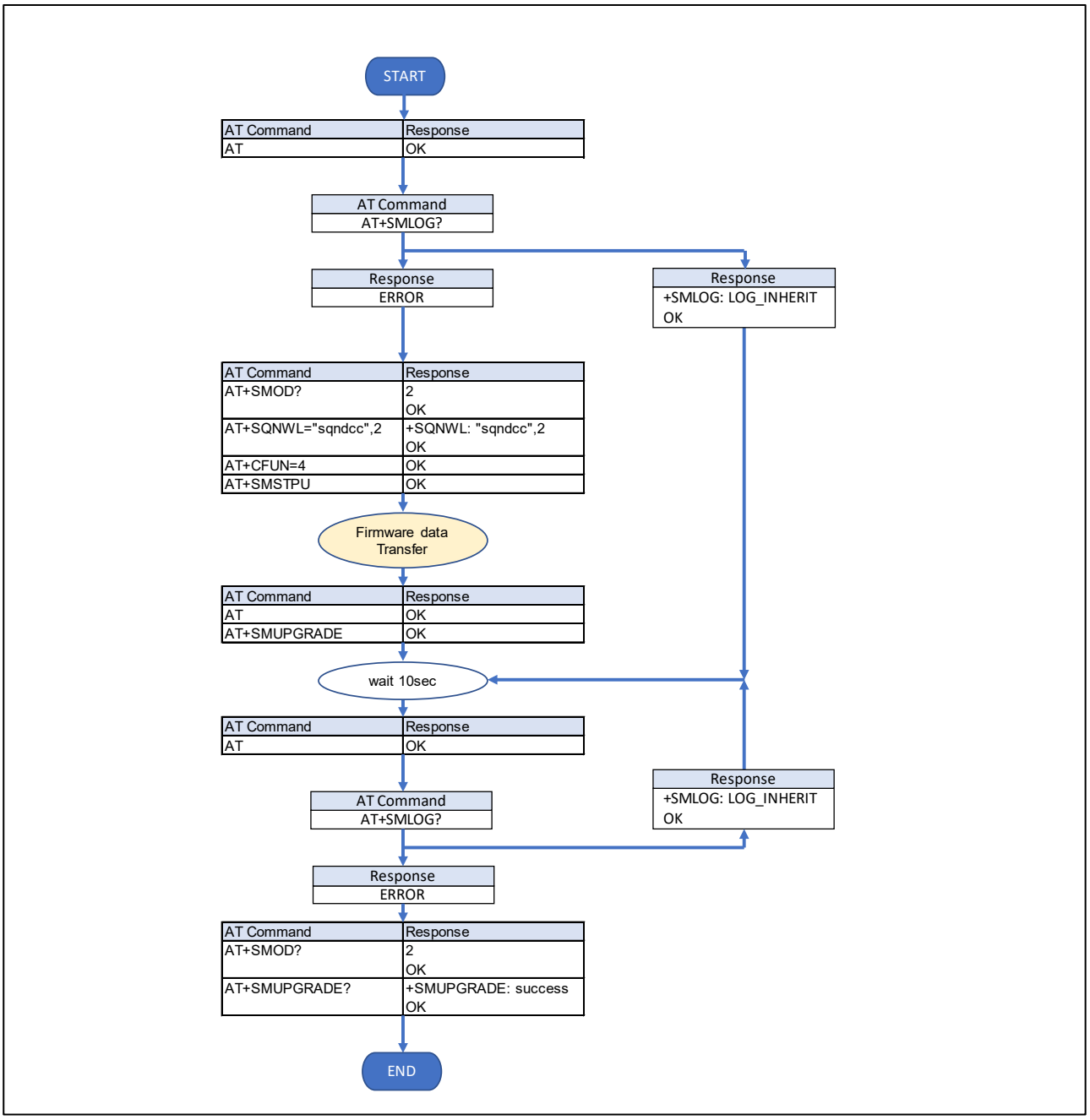

Figure 4-3 AT command flowchart (Differential firmware)

4.2.2 AT Command

An overview of the AT commands used for firmware upgrade is shown in "Table 4-1 AT Command". For details of AT commands, refer to “5.2 AT Commands Definition”.

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 original protocol called Simple Transfer Protocol (STP) to send firmware data to the RYZ024A. Communication is performed using the request packet sent from the host MCU to the RYZ024A and the response packet sent from the RYZ024A to the host MCU.

The packet configuration is shown in "4.3.1 Packet Configuration", and the STP processing details are shown in "4.3.2 Operation Flow".

4.3.1 Packet Configuration

4.3.1.1 STP packet

The STP packet configuration is shown in "Table 4-2 STP packet configuration". The 16 bytes excluding the payload in "Table 4-2 STP packet configuration" are called the STP header.

Table 4-2 STP packet configuration

Name	Size	Description
signature	4	Request : 0x66617374 ("fast") Response : 0x74736166 ("tsaf")
Operation (op)	1	0x00 : reset 0x01 : open session 0x02 : transfer block command 0x03 : transfer block Response operation is OR with 0x80.
session ID (sid)	1	set to 0 on reset, set to 1 once session is open.
payload length (plen)	2	payload length
transaction ID (tid)	4	transaction ID, set to 0 on reset, then incremented after each answer.
full header checksum (hcrc16)	2	Request : full header checksum Response : 0
payload checksum (pcrc16)	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 size used differs depending on the status of each operation (1) to (4).

(1) open session operation payload (Response)

The payload configuration added to the open session operation response is shown in "Table 4-3 open session payload configuration".

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)

The payload configuration added to the transfer block command operation request is shown in "Table 4-4 transfer block command payload configuration".

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 configuration added to the transfer block operation request is shown in "Table 4-5 transfer block operation request payload configuration".

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 configuration added to the transfer block operation response is shown in "Table 4-6 transfer block command response payload configuration".

Table 4-6 transfer block command response payload configuration

Name	Size	Description
residue	2	always 0.

4.3.1.3 Checksum

The checksum calculation method is shown below.

```
uint16_t ryz024a_fwup_stp_crc16(const void *input, size_t length)
{
    uint16_t crc = 0;
    const uint8_t *data = (const uint8_t *)input;

    while (length-- > 0)
    {
        crc  = (uint16_t)((0xFF & (crc >> 8)) | (crc << 8));
        crc ^= (uint16_t)*data++;
        crc ^= (uint16_t)((crc & 0xff) >> 4);
        crc ^= (uint16_t)(crc << 12);
        crc ^= (uint16_t)((uint8_t)(crc & 0xff) << 5);
    }
    return crc;
}
```

Figure 4-4 Checksum calculation method

4.3.2 Operation Flow

Firmware data transfer processing using STP is executed by sending and receiving STP packets for each item indicated in "operation" in "Table 4-2 STP packet configuration".

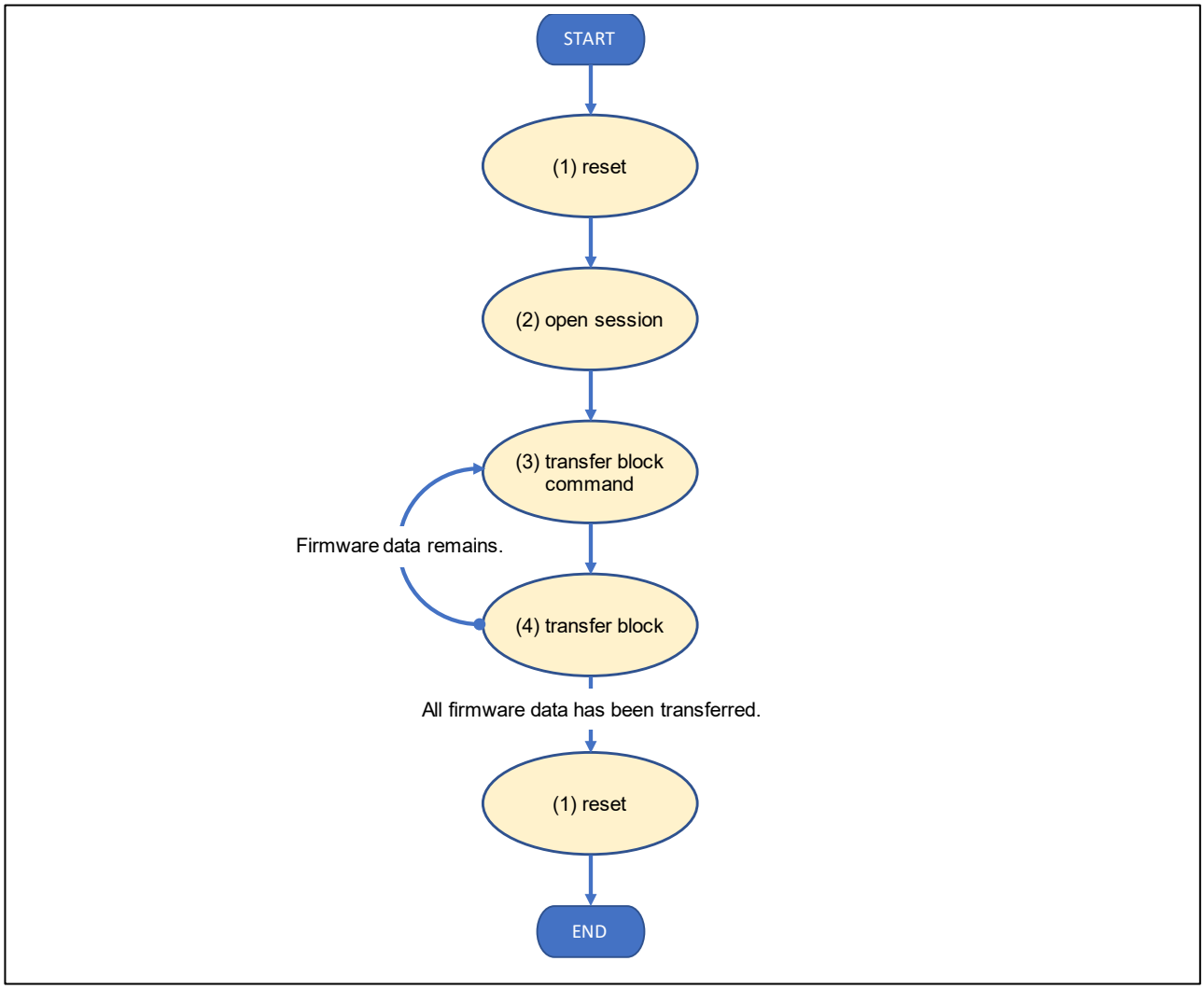

Figure 4-5 Operation flowchart

(1) reset

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

Request packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload
66	61	73	74	00	00	00	00	00	00	00	00	8A	E3	00	00	-

(Hexadecimal)

Response packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload
74	73	61	66	80	00	00	00	00	00	00	00	00	00	00	00	-

(Hexadecimal)

(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.

Request packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload
66	61	73	74	01	01	00	00	00	00	00	01	FB	8E	00	00	-

(Hexadecimal)

Response packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload			
																success	version	max transfer size	
74	73	61	66	81	01	00	04	00	00	00	01	00	00	47	62	01	01	20	00

(Hexadecimal)

(3) transfer block command

- Instructs RYZ024A 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.

Request packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload	
																block size	
66	61	73	74	02	01	00	02	00	00	00	02	E1	18	FF	21	0F	F0

(Hexadecimal)

Response packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload
74	73	61	66	82	01	00	00	00	00	00	02	00	00	00	00	-

(Hexadecimal)

(4) transfer block

- Send the firmware data to RYZ024A. 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.

Request packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload
																firmware data
66	61	73	74	03	01	0F	F0	00	00	00	03	D2	3D	xx	xx	xx xx xx xx xx...

(Hexadecimal)

Response packet

signature				op	sid	plen		tid				hcrc16		pcrc16		payload	
																residue	
74	73	61	66	83	01	00	02	00	00	00	03	00	00	00	00	00	00

(Hexadecimal)

5. Appendix

5.1 Change operating parameters

5.1.1 Differential firmware upgrade timeout time

Differential firmware upgrade requires several minutes to complete after transferring the differential firmware data to the RYZ024A. If timeouts occur during firmware upgrade, change the values defined in Table 5-1.

Table 5-1 AT Differential firmware upgrade timeout time

Name	Value	Description
DIFF_UPGRADE_TO_COUNT_MAX	18	Differential firmware upgrade timeout time. unit: 10 sec e.g.) 10 sec * 18 = 180 sec

5.1.2 Software wait time until AT command is sent

After receiving the AT command response, software waits for about 20ms before sending the next AT command. If you change the host MCU and need to change the wait value, change the value defined in Table 5-2.

Table 5-2 Software wait time until AT command is sent

Name	Value	Description
ATC_SEND_WAIT	20	Software wait time until AT command is sent. unit: 1 msec e.g.) 1 msec * 20 = 20 msec

5.2 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

1: reboot after the OK response

(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?	+SMLOG=LOG_INHERIT
	ERROR

Parameters:

- +SMLOG=LOG_INHERIT: Firmware is upgrading.
- ERROR: Firmware has been upgraded.

Command	Possible Response(s)
AT+SMLOG?	+SMLOG=LOG_INHERIT
	ERROR

Parameters:

- +SMLOG=LOG_INHERIT: Firmware is upgrading.
- ERROR: Firmware has been upgraded.

(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	Dec.18.23	-	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

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/.

© 2023 Renesas Electronics Corporation. All rights reserved.