Renesas RA Family

Standard Boot Firmware for the RA family MCUs Based on Arm® Cortex®-M33

Introduction

This document describes the Standard Boot Firmware Specifications for RA family MCUs flash memory based on Arm Cortex-M33 flash memory. Make sure you understand the target devices and FPSYS / FACI specifications before reading this document. We do not guarantee the operation of any usage not described in this document.

Target Device

Product Group-A (Hereafter referred to as GrpA)
- RA4M2 Group
- RA4M3 Group
- RA6M4 Group
- RA6M5 Group

Product Group-B (Hereafter referred to as GrpB)
- RA4E1 Group
- RA6E1 Group

Product Group-C (Hereafter referred to as GrpC)
- RA6T2 Group

Product Group-D (Hereafter referred to as GrpD)
- RA4E2 Group
- RA6E2 Group
- RA4T1 Group
- RA6T3 Group

Contents

1. Definition of term

A definition of the terminology used in this specification is indicated below.

1. Boot firmware
2. Flash memory
3. Secure / Non-secure / Non-secure callable
4. Device Lifecycle management (DLM)
5. Block Protection

1.1 Boot firmware

The program included in the microcontroller to rewrite the flash memory is called Boot firmware.

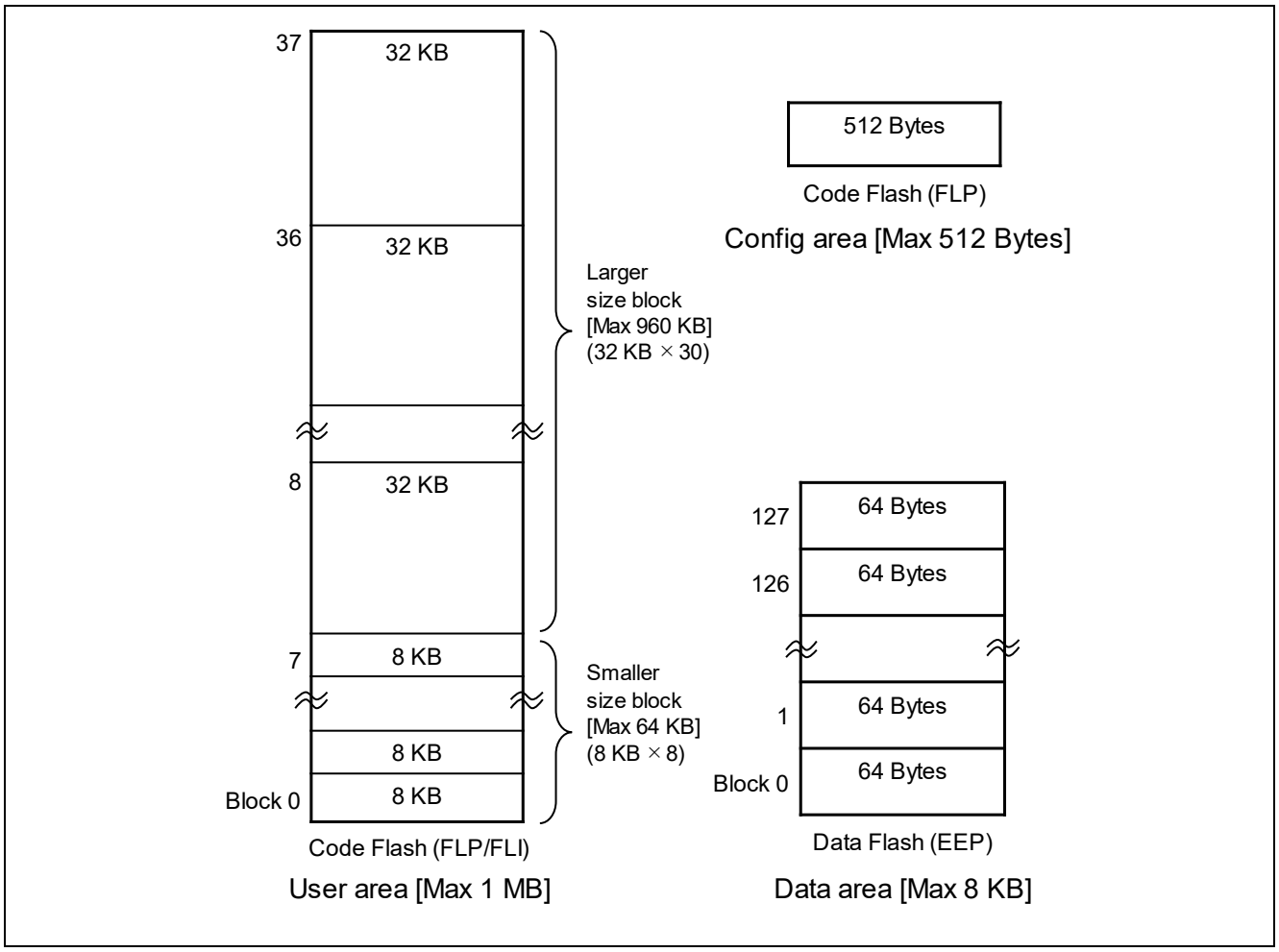
1.2 Flash memory

The following areas are collectively called flash memory.

- Code flash : The ROM area where program code is written (FLP/FLI)
- Data flash : The ROM area where data is written (EEP)

The Code flash area used by user is called ”User area”, the Data flash area used by user is called “Data area”, and the area to store configuration data is called “Config area”. The boot firmware rewrites and reads these User areas, Data area, and Config area according to commands given by the user.

Example of 1 MB flash memory structure (memory structure will differ from device to device)



1.3 Secure / Non-secure

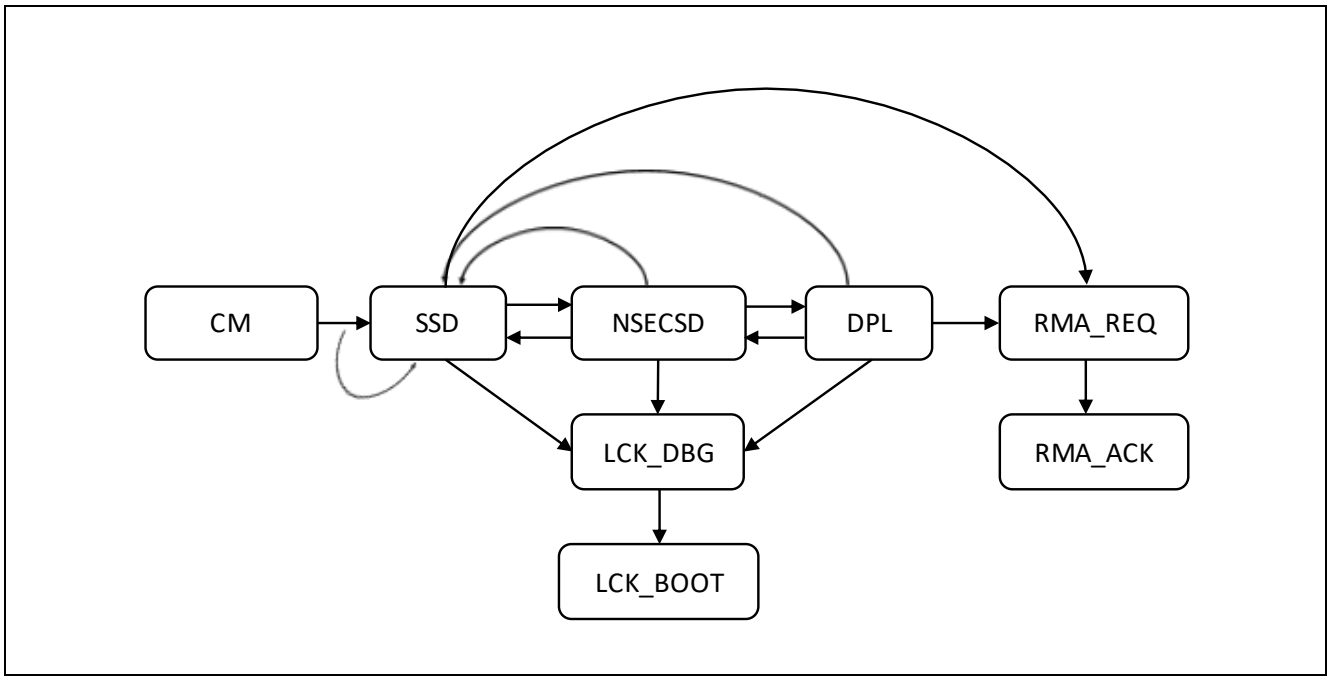
The Renesas Advanced (RA) Family MCUs have the attributes of secure and non-secure. In particular, the memory area is divided into two exclusive areas, a secure area and a non-secure area. The CPU core has two secure states, a secure state and a non-secure state, and the secure state of the CPU changes depending on the secure attribute of the memory where the execution code exists. When the CPU core processes the execution code in the secure area, it is in the secure state, and when it processes the execution code in the non-secure area, it is in the non-secure state. Then, without going through special procedures, the CPU core transitions from the non-secure state to the secure state, or the CPU core is in the non-secure state and accesses the memory in the secure area. By preventing it by the mechanism of the CPU core, it bears a part of the security function of the Renesas Advanced (RA) Family MCUs.

The boot firmware specifies a secure area and a non-secure area for the User area, Data area, and SRAM by a user command.

1.4 Device Lifecycle Management (DLM) for GrpA, GrpB, and GrpC

RA family MCUs based on Arm Cortex-M33 microcontrollers adopt the concept of device life cycle and maintain the life cycle state inside the device.

The boot firmware controls the executable commands and the range of operations that can be performed with each command in each lifecycle state. In addition, it has a user-executable command as the only way to transition lifecycles.



DLM state name	Description
CM	Chip Manufacturing.
SSD	Secure Software Development.
NSECSD	Non-SECure Software Development.
DPL	DePLoyed.
LCK_DBG	LoCKed DeBuG.
LCK_BOOT	LoCKed BOOT interface.
RMA_REQ	Return Material Authorization REQuest.
RMA_ACK	Return Material Authorization ACKnowledged.

1.5 Block Protection

Block protection refers to a function that prohibits erasing/writing the specified range of Flash memory. The specified range is done in blocks, and there are two types of protection listed below.

Types of protection	Description
Block protection (BPS)	Protection that can temporarily enable erasing/writing by the register setting of flash sequencer
Permanent block protection (PBPS)	Protection that permanently disables erasing/writing

2. System Architecture

Boot firmware has a serial programming interface that sends and receives flash control commands between the microcontroller and the host in serial programming mode. Boot firmware is embedded into the device.

2.1 GrpA(RA4M2, RA4M3, RA6M4, RA6M5), GrpB(RA4E1, RA6E1), GrpC(RA6T2), and GrpD(RA4E2, RA6E2, RA4T1, RA6T3) groups

This chapter describes the system architecture regarding the flash memory control.

2.1.1 Operating environment

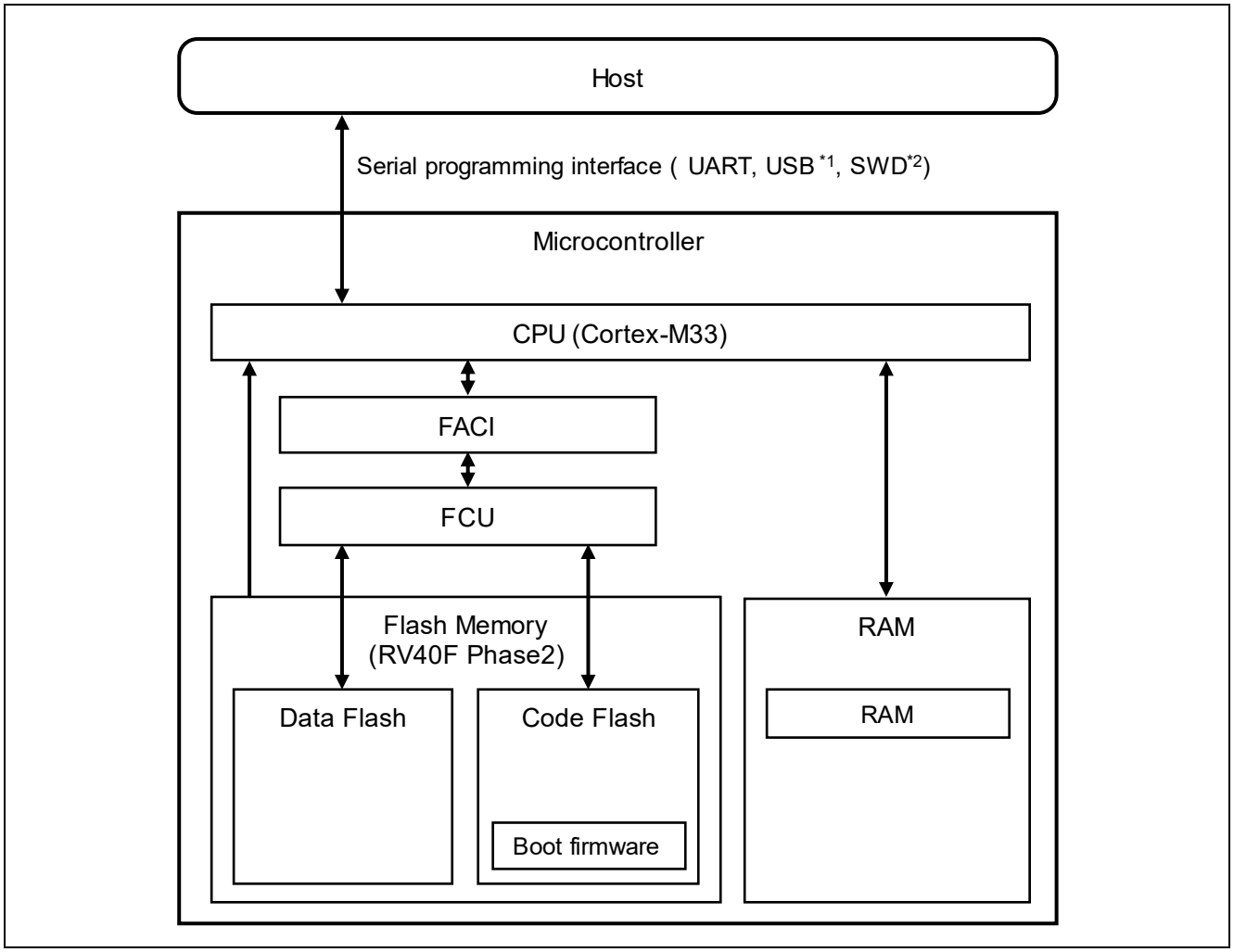
CPU core	Arm Cortex-M33
Max. CPU operating frequency	RA4M2, RA4M3, RA4E1: 100MHz (Boot firmware operating frequency: 100MHz) RA6M4, RA6M5, RA6E1: 200MHz (Boot firmware operating frequency: 100MHz) RA6T2: 240MHz (Boot firmware operating frequency: 100MHz) RA4E2, RA4T1 : 100MHz (Boot firmware operating frequency: 60MHz) RA6E2, RA6T3 : 200MHz (Boot firmware operating frequency: 60MHz)
Clock Source	RA4M2/3, RA6M4/5, RA4E1/2, RA6E1/2, RA4T1, RA6T3 8, 10, 16, 20, 24 MHz, * If neither is set, operates with HOCO * However, a Main-OSC whose frequency is around plus-minus 3% or less of the frequency above is set, there is a possibility that the frequency is misjudged and therefore USB communication fails. To avoid this, it is recommended to choose any of the followings when using USB communication. - Use a Main-OSC whose frequency is the very value listed above - Not use a Main-OSC and also use a Sub-OSC whose frequency is supported by the device's specifications • RA6T2 HOCO 20MHz (Doesn't use Main-OSC)
Operating voltage	VCC = 2.7 to 3.6 V
Operating mode	Boot mode
Flash memory	• Code flash User area: 256KB(Max.) : RA4E2, RA6E2, RA4T1, RA6T3 512 KB(Max.) : RA4M2, RA4E1, RA6T2 1 MB(Max.) : RA4M3, RA6M4, RA6E1 2 MB(Max.) : RA6M5 • Data flash 4KB : RA4E2, RA6E2, RA4T1, RA6T3 8KB(Max.) : RA4M2, RA4M3, RA6M4, RA6M5, RA4E1, RA6E2 16KB : RA6T2:
RAM	40KB : RA4E2, RA6E2, RA4T1, RA6T3 64KB : RA6T2 128KB : RA4M2, RA4M3, RA4E1 256KB : RA6M4, RA6E1, RA6M5

Communication method

Method	GrpA GrpB	GrpC	GrpD
2-wire UART communication			
(Initial/Min) 9600 bps (Max) 6 Mbps	✔	✔	
(Initial/Min) 9600 bps (Max) 2 Mbps			✔
USB communication* 12 Mbps	✔		✔
SWD communication (Max) 6MHz			✔

* When performing USB communication with HOCO, Sub-OSC must be oscillating stably.
* USB communication is confirmed under the condition that the host OS is Windows 10.

2.1.2 Block diagram



*1: GrpC doesn't support.
*2: GrpD supports

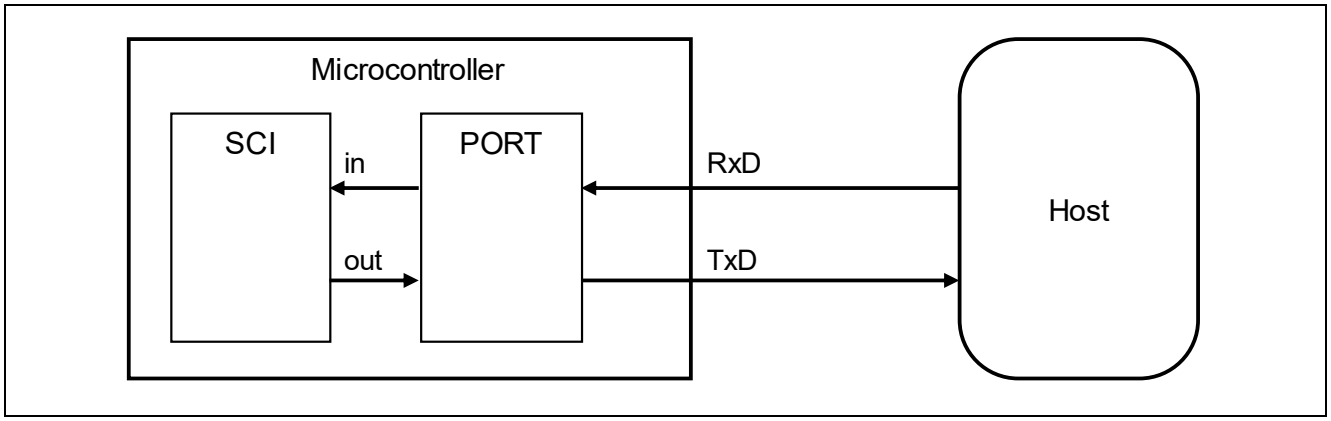
3. Communication Method

Boot firmware has interfaces for the following communication methods:

	GrpA, GrpB	GrpC	GrpD
2-wire UART communication	✔	✔	✔
Universal Serial Bus (USB) communication	✔		✔
SWD communication			✔

3.1 2-wire UART communication

Boot firmware supports the 2-wire UART communication.



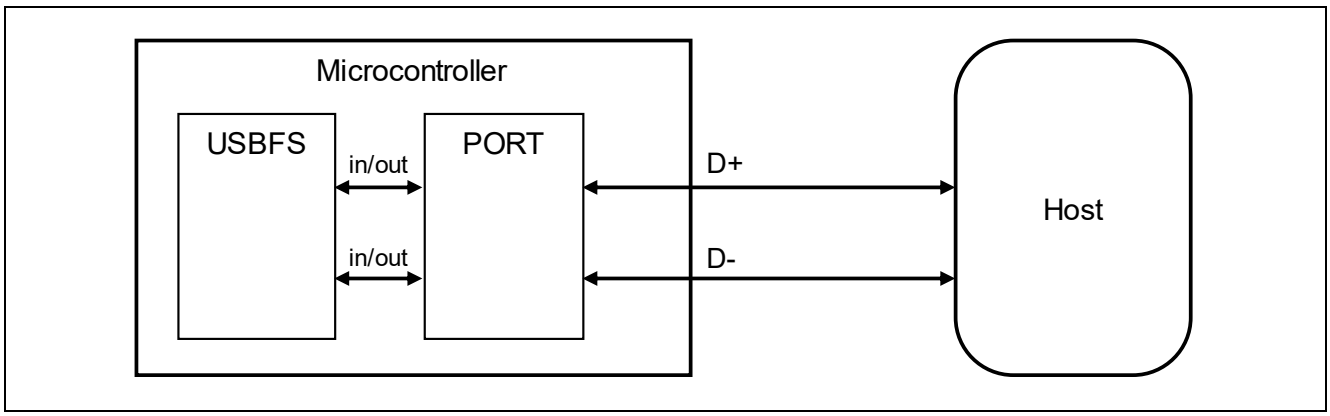
General settings		
Interface	SCI (ch9)	
RxD	• (GrpA, GrpB, GrpD) P110, Input mode • (GrpC) PA15, Input mode	
TxD	• (GrpA, GrpB, GrpD) P109, Output mode • (GrpC) PB03, Output mode	
Transfer rate	9600bps	(Min, until the baud rate setting command)
Data length	8 bits	(LSB first)
Parity bit	none	
Stop bit	1 bit	

Communication is performed at 9600bps until the transfer rate setting command. After the baud rate setting command is completed normally, communication is performed at the desired transfer rate. The maximum transfer rate that can be communicated with the device is returned by “RMB” of the signature request command.

* If the communication cable is disconnected during communication, subsequent operations are not guaranteed.

3.2 Universal Serial Bus (USB) communication

Boot firmware supports the USB communication.



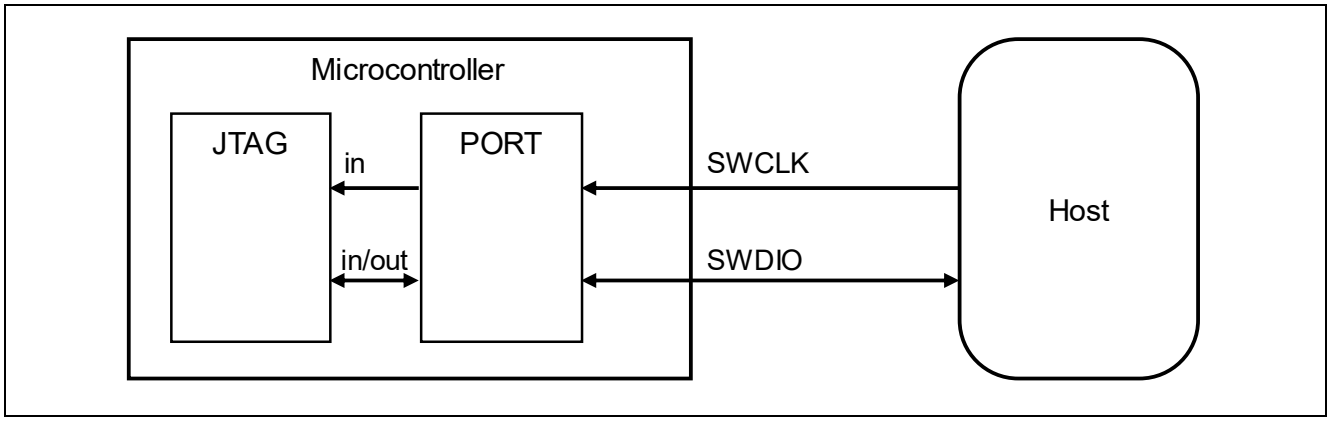
General settings	
Interface	USBFS
VBUS	P407, Input mode
D+	Input-Output mode
D-	Input-Output mode
Transfer rate	12 Mbps (USB2.0 Full Speed)
Device class	Communication Device Class (CDC) SubClass : Abstract Control Model (ACM) Protocol : Common AT commands
Vender ID	0x045B (Renesas)
Product ID	0x0261
Transfer mode	Control (in/out) Bulk (in, out) Interrupt (in)
End point	:

	GrpA, GrpB	GrpC	GrpD
Default control pipe, Control transfers (in/out)	EP0	N/A	EP0
TxD pipe, Bulk transfers (in) 64 bytes	EP1	N/A	EP4
RxD pipe, Bulk transfers (out) 64 bytes	EP2	N/A	EP5
Control pipe, Interrupt transfers (in)	EP6	N/A	EP6

* If the USB cable is disconnected during communication, subsequent operations are not guaranteed.
* When performing USB communication, the host is notified as self power mode.
* USB boot does not guarantee operation with bus power (when rewriting with bus power, please execute it after verifying sufficiently by users).
* GrpC does not support USB communication.

3.3 SWD communication

Boot firmware supports the SWD communication. SWD communication is enabled by setting a magic code in the JBMDR register during terminal reset.



General settings	
[SWD] SWCLK	P300, Input mode
[SWD] SWDIO	P108, Input-Output mode
Transfer rate	6MHz (Max)
Data length	32bit
Magic code	0xA5

[Endien of transmission and reception data]
Store the data transmitted from the Host in the JBRDR register by 4 bytes in order from the lower byte.
The data transmitted from the Microcontroller is stored in the JBTDR register by 4 bytes in order from the lower byte.

[example: 1byte data transmission from the Host to the Microcontroller]
sending data: 0x55

JBRDR[31:24]	JBRDR[23:16]	JBRDR[15:8]	JBRDR[7:0]
Don't care	Don't care	Don't care	0x55

[example: 7byte data transmission from the Microcontroller to the Host]
sending data: 0x00, 0x01, 0x02, 0x03

JBTDR[31:24]	JBTDR[23:16]	JBTDR[15:8]	JBTDR[7:0]
0x03	0x02	0x01	0x00

sending data: 0x04, 0x05, 0x06

JBTDR[31:24]	JBTDR[23:16]	JBTDR[15:8]	JBTDR[7:0]
Don't care	0x06	0x05	0x04

[Communication handshake]
Host and microcontroller perform handshake by using JBSTR register in SWD communication.
Host must check JBSTR.RDF=0 before writing data to JBRDR, and JBSTR.TDE=0 before reading data from JBTDR.

However, this handshake is omittable when transmitting and receiving 5th-byte or after in a packet(*), namely host can write JBRDR and read JBTDR without checking JBSTR.

*) 5th-byte or after in a packet means the following specifically.

Command packet	Command information - ETX
Data packet	Data - ETX

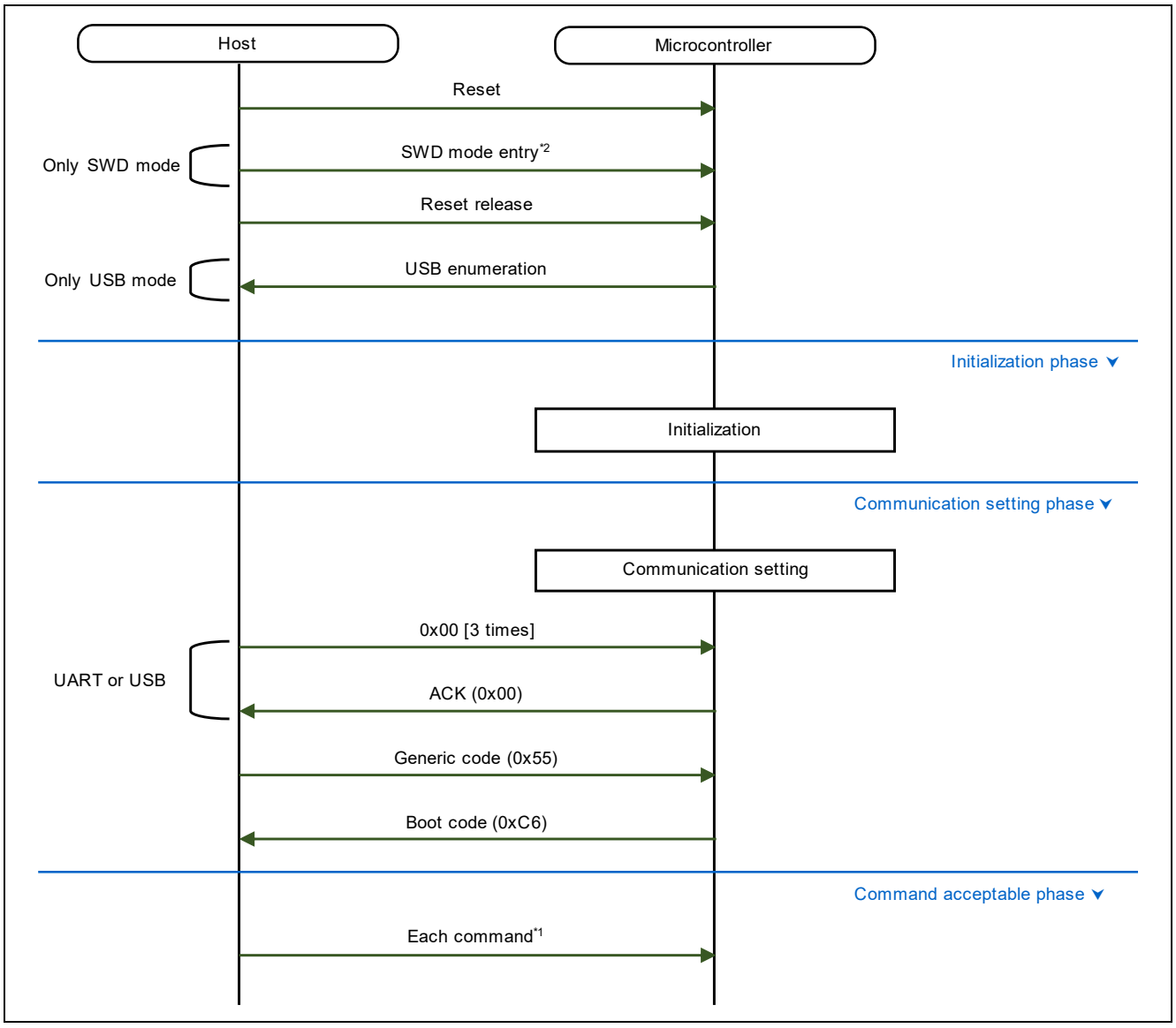
Only GrpD supports SWD communication.

4. General Procedure

Boot firmware transits in the following order after the reset release. This sequence is non-invertible.

1. Initialization phase
2. Communication setting phase
3. Command acceptable phase

4.1 Sequence diagram (Generic sequence)



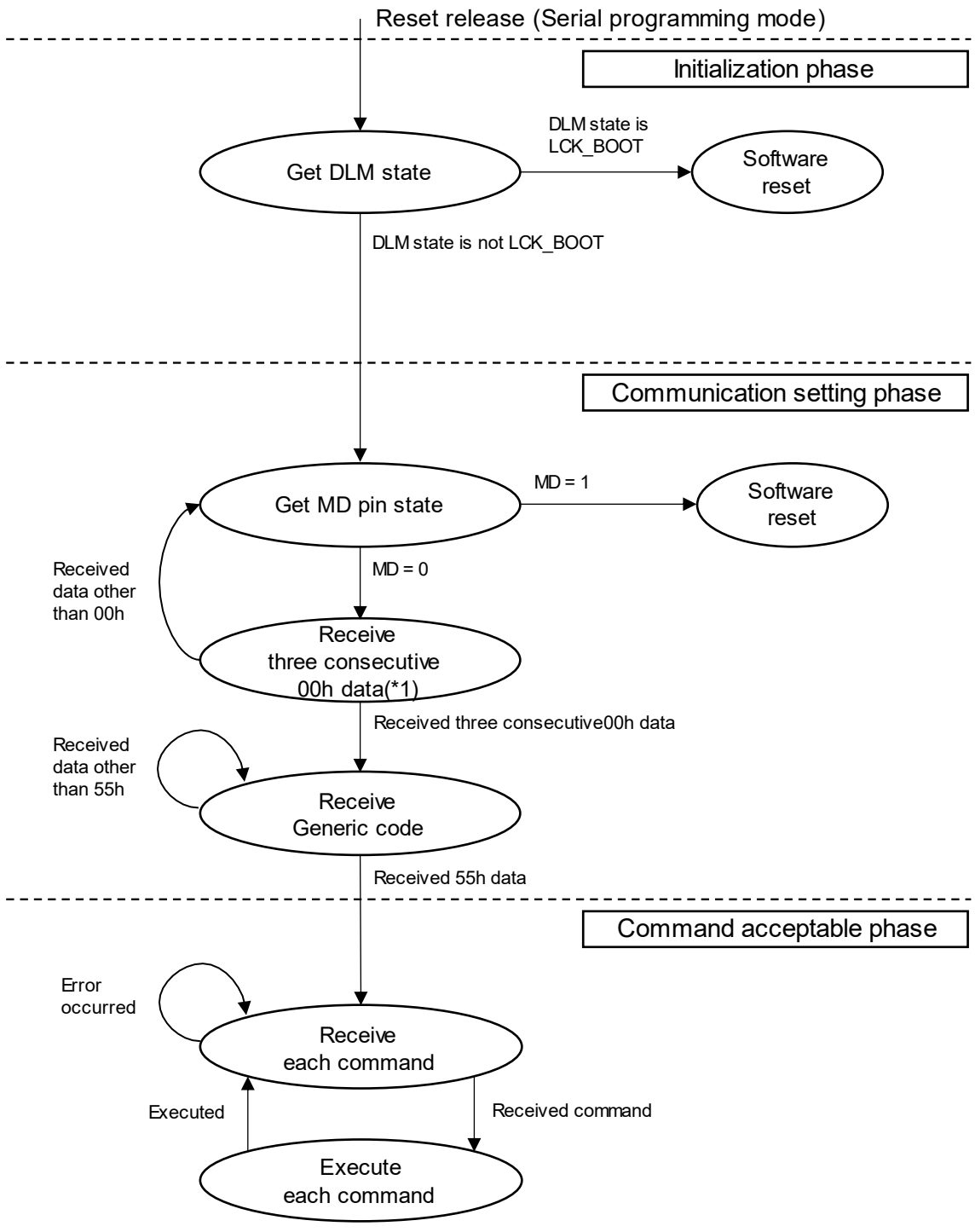
*1 From the Command acceptable phase, the host and microcontroller send data in turn unless otherwise noted. The host executes data transmission after data reception from the microcontroller.

*2: If the magic code "0xA5" is set in the JBMDR register during a pin reset, the microcontroller will boot into SWD mode.

GrpC supports only UART communication, and only GrpD supports SWD communication.

4.2 State transition diagram (generic state transition)

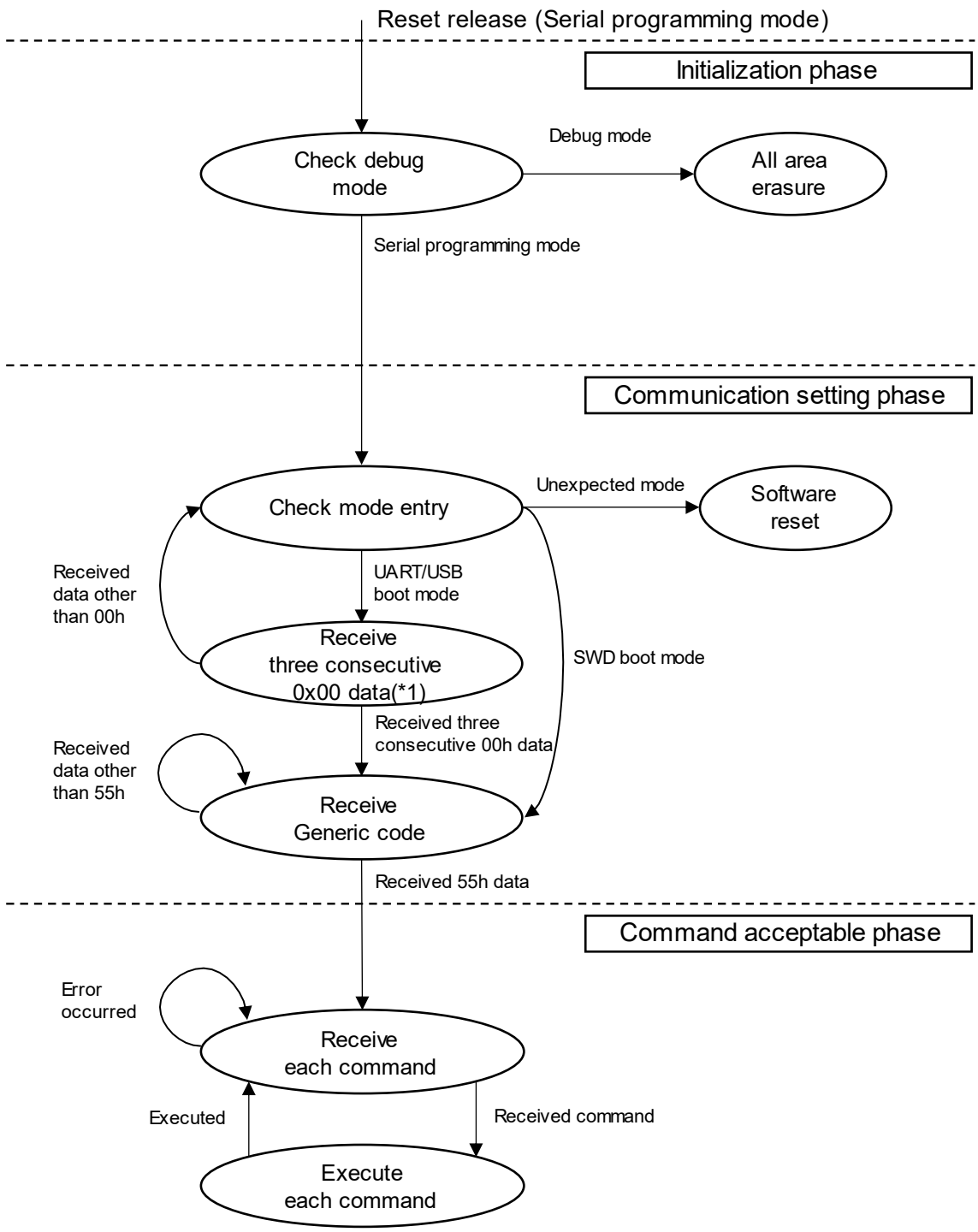
(GrpA, GrpB, and GrpC)



*1 : Boot firmware adopts IP(SCI orUSBFS) that received 00h consecutively three times earlier as the communication method.

*2 : GrpC support UART communication only.

(GrpD)



*1：Boot firmware adopts IP(SCI or USBFS) that received 0x00 consecutively three times earlier as the communication method.

4.3 Cause for operation to stop

The boot firmware enters an infinite loop in the following cases:

- When DLM state transitions to LCK_BOOT
- When DLM state transitions due to the Authentication command (after replying with completion status)
- When DLM state transits to SSD with the Initialize command (after replying with completion status)
- When DLM state has an abnormal value after DLM state transition
- When Trusted system goes into an abnormal state
- When the USB cable is disconnected with the USB status of “Configured” for GrpA and GrpB
- When the following CPU exceptions occur:
  NMI / HardFault / MemManage / BusFault / UsageFault / SecureFault / SVCall / DebugMonitor / PendSV / SysTick
- When Main-OSC and Sub-OSC are not oscillating at the specified frequency

4.4 Cause for software reset

Boot firmware performs software reset in the following cases.

- When the DLM state is LCK_BOOT after startup
- When DLM state is abnormal after startup
- When MD = 1 is detected during communication mode judgement

4.5 Initialization phase

Boot firmware initializes hardware modules in this phase. After that, boot firmware transits to the “Communication setting phase”.

4.5.1 Processing procedure

Boot firmware initializes after reset release.

- Boot firmware initializes hardware modules, and transits to the “Communication setting phase”.

The following features are only available in GrpD.

- If the operation mode is debug mode, debug mode processing is performed.
  If the stored ID[127:126] in the device is not 11b, boot firmware sets to standby mode and become an infinite loop.
  If the stored ID[127:126] in the device is 11b, boot firmware erases all the flash memory, and if the erasure succeeds, sets sleep mode and enters an infinite loop.
  Also, if erasing fails, boot firmware sets to standby mode and enters an infinite loop.

  If there is a block with permanent block protection, boot firmware sets to standby mode without executing all erasure, enters an infinite loop.
  Also, if the FSPR in the config area is set 0, boot firmware sets to standby mode without executing all erasure, enters an infinite loop.
  * Flash memory status does not change before command reception.

4.6 Communication setting phase

The boot firmware establishes communication with the host in this phase. Check the connection of each communication method under the conditions shown in the table below. After receiving the generic code using the established communication method, the boot firmware transitions to the "Command acceptable phase".

Condition	Communication method	GrpA	GrpB	GrpC	GrpD
- Data "0x00" was continuously received 3 times by 2-wire UART communication	2-wire UART communication	✔	✔	✔	✔
- Data "0x00" was continuously received 3 times by USB communication	USB communication	✔	✔		✔
- DBGSTR.CDBGPWRUPREQ=1 is set during terminal reset - Magic code "0xA5" was set in the JBMDR register during terminal reset - MD pin level is High	SWD communication				✔

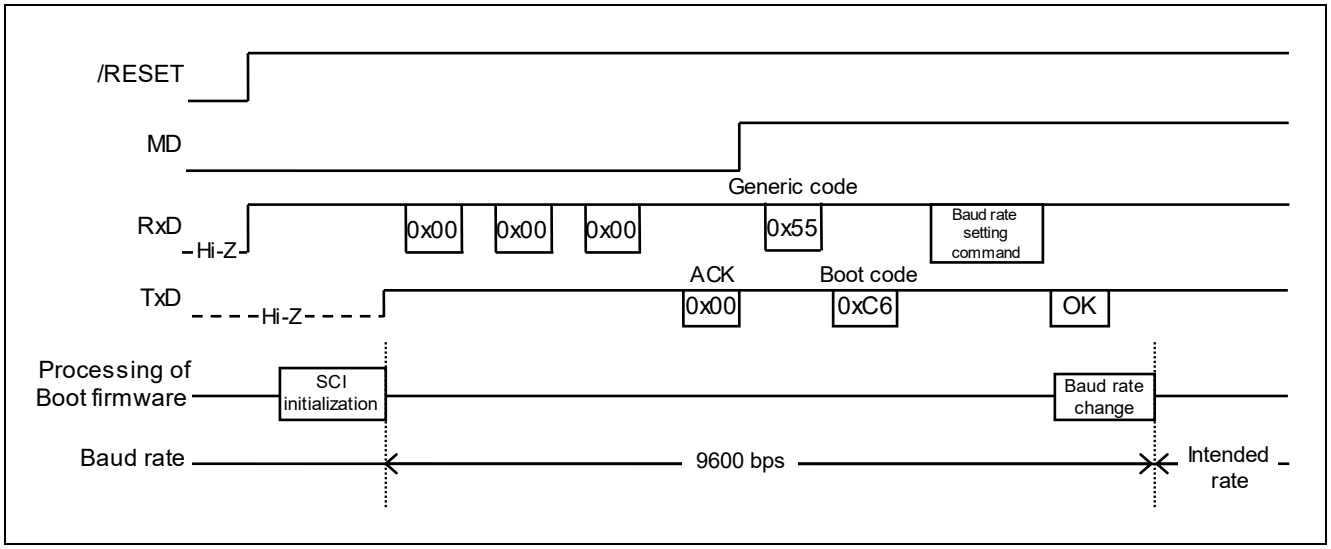
4.6.1 Processing procedure

Boot firmware performs communication settings.

- When all the following conditions are met, the boot firmware will perform a software reset.
  - ・MD=1
  - ・JBMDR≠0xA5 (Only GrpD)
- When all the following conditions are met, JTAG/SWD communication is determined to be selected for GrpD.
  * When JTAG/SWD communication is selected, wait for Generic code without waiting for 0x00.
  - ・MD=1
  - ・JBMDR=0xA5
- When JTAG/SWD communication is not selected or not supported, waiting for 0x00 to be received.
  If 00h is received continuously for 3 bytes in either 2-wire UART communication(GrpA, GrpB, Grp,C GrpD) or USB communication(GrpA, GrpB, GrpD), “ACK” is transmitted. (* data is received until the communication mode is determined)
  * The time from when reset is released until 0x00 can be received is shown AC characteristics.
- After that, when Generic code is received, it sends a “Boot code”.
  If a code other than generic code is received, it will wait to receive Generic code again.
  * The time from when reset is released until Generic code can be received is shown AC characteristics.
- The boot firmware transitions to the “Command acceptable phase” when the transmission of “Boot code” is completed.

4.6.2 Settings of the 2-wire UART communication (For GrpA, GrpB, GrpC, and GrpD)

When the device operating mode is serial programming mode, the boot firmware initializes SCI and waits for reception. By receiving 0x00 three times consecutively, it is determined that asynchronous 2-wire communication is selected as the communication method. Before receiving 3 bytes, if data other than 0x00 is received or some data is received from USB, the count value will be reset. (without GrpC due to not support USB communication)



* Boot firmware of GrpA, GrpB, and GrpC outputs High from TxD after SCI initialization.
  Boot firmware of GrpD enables pull-up of TxD after SCI initialization, and outputs High from TxD after 3-byte 00h reception. After SCI initialization, the boot firmware outputs High from TxD.

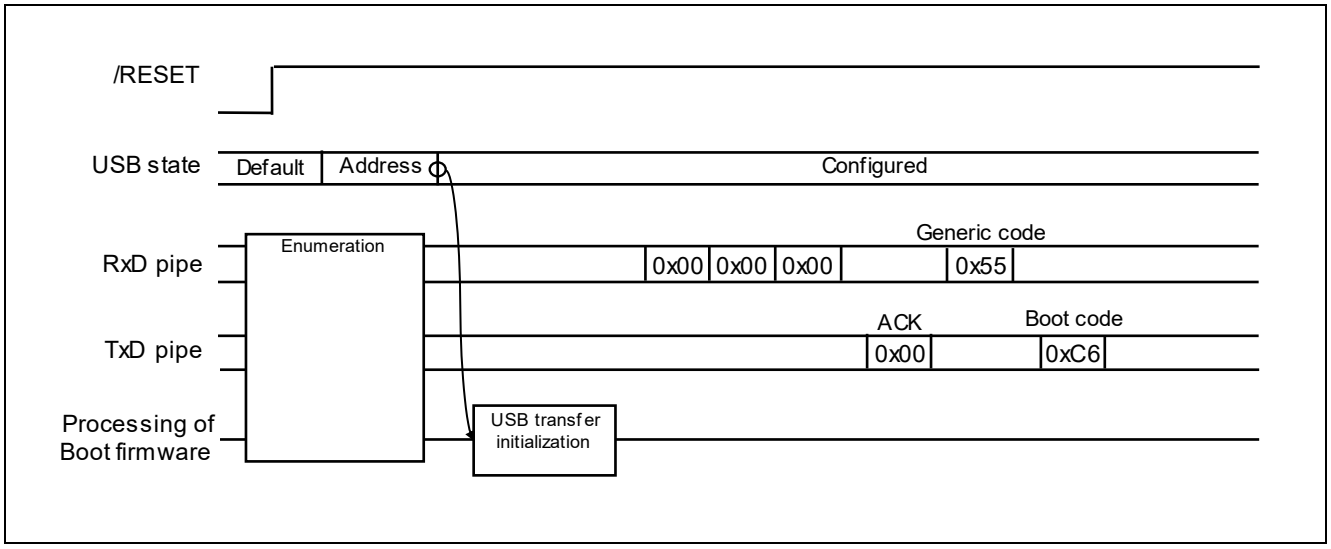
By performing the following procedure, communication establishment is completed and the process moves to the ”Command acceptable phase”.

1. Receive 3 bytes of 0x00 data (9600 bps) from the host
   (Perform 0x00 data transmission until ACK is received in step 2).
2. Send 0x00 data (ACK) from boot firmware.
3. Receive 0x55 data (generic code) from the host.
4. Send 0xC6 data (boot code) from boot firmware.

* If ACK is not returned even after sending 0x00 data, check the communication environment and try again from reset release.

4.6.3 Settings of the USB communication (For GrpA, GrpB, and GrpD)

When the device's operating mode is serial programming mode, the boot firmware configures the USB into an enumerable state. Set the data communication start by USB configured status detection. By receiving 0x00 three times consecutively, it is determined that USB communication is selected as the communication method. Before receiving 3 bytes, if data other than 0x00 is received or some data is received from UART, the count value will be reset.



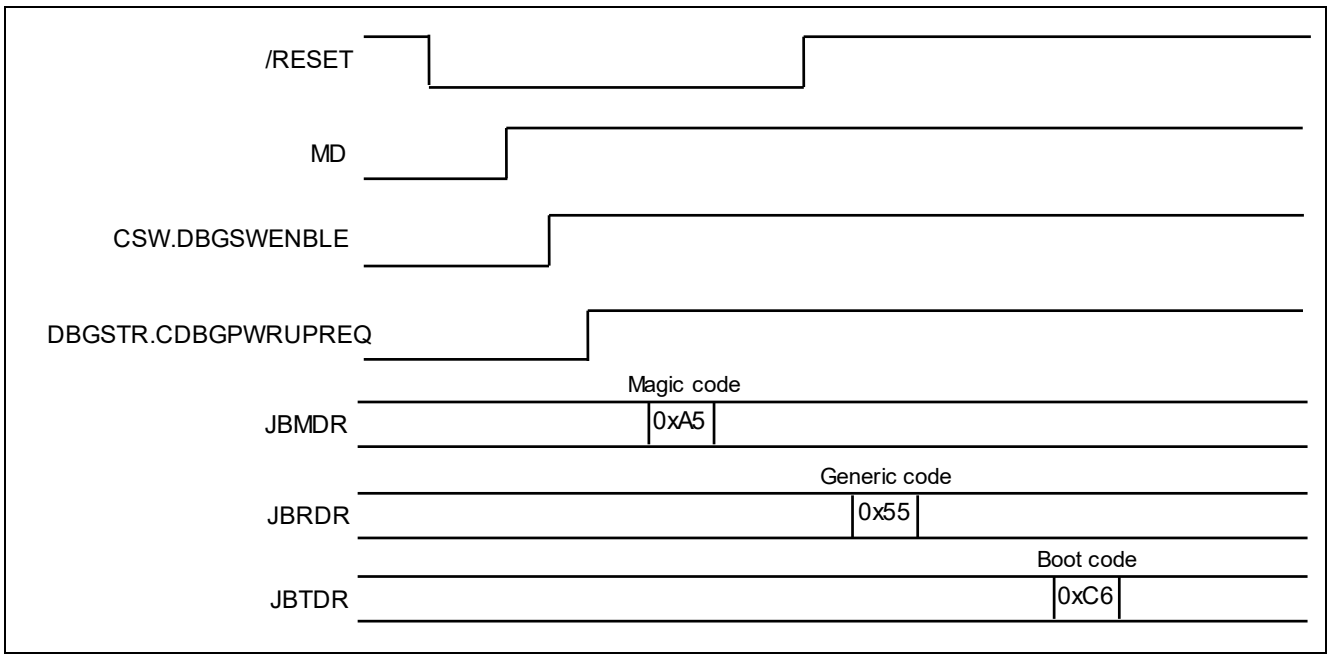
By performing the following procedure, communication establishment is completed and the process moves to the ‘Command acceptable phase’.

1. When the boot firmware detects the USB configured state, the USB communication start setting is performed.
2. Receive 3 bytes of 0x00 data from the host
   (Perform 00h data transmission until ACK is received in step 3.)
3. Send 0x00 data (ACK) from boot firmware.
4. Receive 0x55 data (generic code) from the host.
5. Send 0xC6 data (boot code) from boot firmware.

* If ACK is not returned even after sending 0x00 data, check the communication environment and try again from reset release.

4.6.4 Settings of the SWD communication (For GrpD)

When the boot firmware detects MD=1 and JBMDR=0xA5, establish communication with SWD communication.



By performing the following procedure, communication establishment is completed and the process moves to the "Command acceptable phase".

1. Assert the terminal reset
2. Set 1 to CSW.DBGSWENBLE
3. Set 1 to DBGSTR.CDBGPWRUPREQ
4. Wait until DBGSTR.CDBGPWRUPACK becomes 1
5. Set 0xA5 to JBMDR
6. Release the terminal reset
7. If MD detects 1 after following the above procedure, the boot firmware will set the SWD communication start setting
8. Receive 0x55 data (Generic code) from the host
9. Send 0xC6 data (Boot code) from the boot firmware

Moreover, follow the steps below to disconnect SWD communication with boot firmware.

1. Assert the terminal reset
2. Set 0x00 to JBMDR
3. Set 0 to DBGSTR.CDBGPWRUPREQ
4. Wait until DBGSTR.CDBGPWRUPACK becomes 0
5. Set 0 to CSW.DBGSWENBLE

4.7 Command acceptable phase

Boot firmware accepts the commands on this phase.

4.7.1 Processing procedure

When the boot firmware receives a command packet, it performs packet analysis.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the CMD in the received command packet is an undefined code, the boot firmware sends an ”Unsupported command error”

- .
- If the received command packet’sfLNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.

When the processing above is successfully completed, boot firmware executes command processing. (Refer to explanation of each command for detail)

- When a command is normally finished boot firmware stays on the ”Command acceptable phase”.

5. Packet Format

Be sure to follow this format.
1. Command packet
2. Data packet

<Elements in the packet>
- CMD : Command code
- RES : Response code
- STS : Status code
- DLM : Device Lifecycle Management state code
- FST : Flash status
- ADR : Failure address

5.1 Command packet

The host sends data of a command packet to microcontroller by the following format.

Symbol	Size	Value	Description
SOH	1 byte	0x01	Start of command packet
LNH	1 byte	-	Packet length (length of ”CMD + command information”) [High]
LNL	1 byte	-	Packet length (length of ”CMD + command information”) [Low]
CMD	1 byte	-	Command code
Command information	0 to 255 bytes	-	Command information e.g.) For write command: Start/End address e.g.) For erase command: Start/End address e.g.) For DLM state transit command: Source/Destination DLM state code (DLM) (DLM is supported on GrpA, GrpB, and GrpD) e.g.) for Baudrate setting command: UART baudrate
SUM	1 byte	-	Sum data of “LNH + LNL + CMD + Command information” (expressed as two's complement) e.g.) LNH + LNL + CMD + Command information(1) + Command information(2) + ... + Command information(n) + SUM = 0x00
ETX	1 byte	0x03	End of packet

*1: If the host sends data that exceeds 261 bytes, subsequent operations are not guaranteed.

5.2 Data packet

Host and boot firmware send data to each other by the following format.

Symbol	Size	Value	Description
SOD	1 byte	0x81	Start of data packet
LNH	1 byte	-	Packet length (length of ”RES + Data”) [High] (*1)
LNL	1 byte	-	Packet length (length of ”RES + Data”) [Low] (*1)
RES	1 byte	-	Response code
Data	(*3)	-	Transmit data e.g.) For write data transmission: Write data e.g.) For status transmission: Status code (STS), Status detail (ST2), and Failure address (ADR) e.g.) For DLM state requesting: DLM state code (DLM)
SUM	1 byte	-	Sum data of “LNH + LNL + RES + Data” (expressed as two's complement) e.g.) LNH + LNL + RES + Data(1) + Data(2) + ... + Data(n) + SUM = 0x00
ETX	1 byte	0x03	End of packet

*1: If the host sends a packet whose length is 0 byte or over 1025 bytes, the microcontroller will return a packet with indefinite RES value.
*2: If the host sends data that exceeds 1030 bytes, subsequent operations are not guaranteed.
*3: The size is 1~1024byte.

5.3 CMD : Command code

Value	Device			Name	Description
	GrpA GrpC	GrpB	GrpD		
0x71	✔	✔		DLM state transit command	Authentication-free DLM transition
0x30	✔	✔	✔	Authentication command	DLM transition for authentication
0x28	✔	✔		Key setting command	Insert the key
0x2A	✔			User key setting command	Insert the user custom key
0x29	✔	✔		Key verify command	Verify the key
0x2B	✔			User key verify command	Verify the user custom key
0x50	✔	✔		Initialize command	Initialize all User area, Data area, and Config area
0x2C	✔	✔		DLM state request command	Request the current DLM state
0x4E	✔	✔		Boundary setting command	Set the boundary
0x4F	✔	✔		Boundary request command	Get the boundary setting
0x51	✔	✔		Parameter setting command	Set the parameter
0x52	✔	✔		Parameter request command	Get the parameter setting
0x00	✔	✔	✔	Inquiry command	Return ACK
0x3A	✔	✔	✔	Signature request command	Get the signature information
0x3B	✔	✔	✔	Area information request command	Get the area information
0x34	✔	✔	✔	Baud rate setting command	Set baud rate (only UART)
0x12	✔	✔	✔	Erase command	Erase data on target area
0x13	✔	✔	✔	Write command	Write data to target area
0x15	✔	✔	✔	Read command	Read data from target area
0x18	✔	✔	✔	CRC command	Cyclic Redundancy Check of target area

5.4 RES : Response code

Value	Name	Description
0x00 | CMD	OK (ongoing normally)	-
0x80 | CMD	ERR (occurrence of an error)	-

5.5 STS : Status code

Value	GrpA GrpB GrpC	GrpD	Name	Description
0x00	✔	✔	Communication is normal [OK]	-
0xC0	✔	✔	Unsupported command error	(*1), Received an unsupported command
0xC1	✔	✔	Packet error	(*1), Abnormality of packet format
0xC2	✔	✔	Checksum error	(*1), Abnormality of packet's checksum value
0xD0	✔	✔	Parameter error	(*1), Abnormality of packet parameter
0xD5	✔	✔	Command acceptance error	(*1), A command cannot execute in current state (*4)
0xD6	✔		DLM state unmatched error	(*1), Different from info2's value and DLMMON register's value
0xD7	✔		Hardware error	(*1), Abnormality of flash memory (DLM state) value
0xDA	✔	✔	Protection error	(*1), Accessing protected areas or performing prohibited actions.
0xE4	✔		Secure error	(*1), Included ”Secure area” as targets for erase, write, and key injection
0xDB	✔	✔	Trusted system error	(*1), Abnormality from the Trusted system(TSIP)
0xDD		✔	ID discord error	(*1), ID authentication failed.
0xDE		✔	Serial programming disable error	(*1), If serial programming is disabled. (stored ID[127] = 0)
0xE5	✔	✔	Flash access error	(*1), (*2), (*3), Abnormality from the Flash sequencer

*1: When this error occurs, response code (RES) will be ERR.
*2: Boot firmware also returns the Status details (ST2) and the failure address (ADR) as additional error information.
*3: This error occurs when the flash sequencer becomes ”command lock” state after execution flash sequencer command.
*4: GrpA, GrpB, GrpC : DLM state, GrpD : ID authentication state

5.6 DLM : Device Lifecycle Management state code (GrpD is not supported)

Value	Name	Description
0x01	CM	Chip Manufacturing
0x02	SSD	Secure Software Development
0x03	NSECSD	Non-Secure Software Development
0x04	DPL	DePLoyed
0x05	LCK_DBG	LoCKed DeBuG
0x06	LCK_BOOT	LoCKed BOOT interface
0x07	RMA_REQ	Return Material Authorization REQuest
0x08	RMA_ACK	Return Material Authorization ACKnowledged

5.7 ST2 : Status details

Value	Name	Description
FSTATR[31:0]	Flash Status	When a flash access error occurs, boot firmware returns the value of the FSTATR register. When not, boot firmware returns 0xFFFFFFFF. Boot firmware clears the FSTATR register after the status sending, so even when Error occurs, host can retry the next command without reset release.

5.8 ADR : Failure address

Value	Name	Description
0x00000000h to 0xFFFFFFFF	Failure address	When a flash access error occurs, boot firmware returns the value of the start address of the flash sequencer command. When not, boot firmware returns 0xFFFFFFFF.

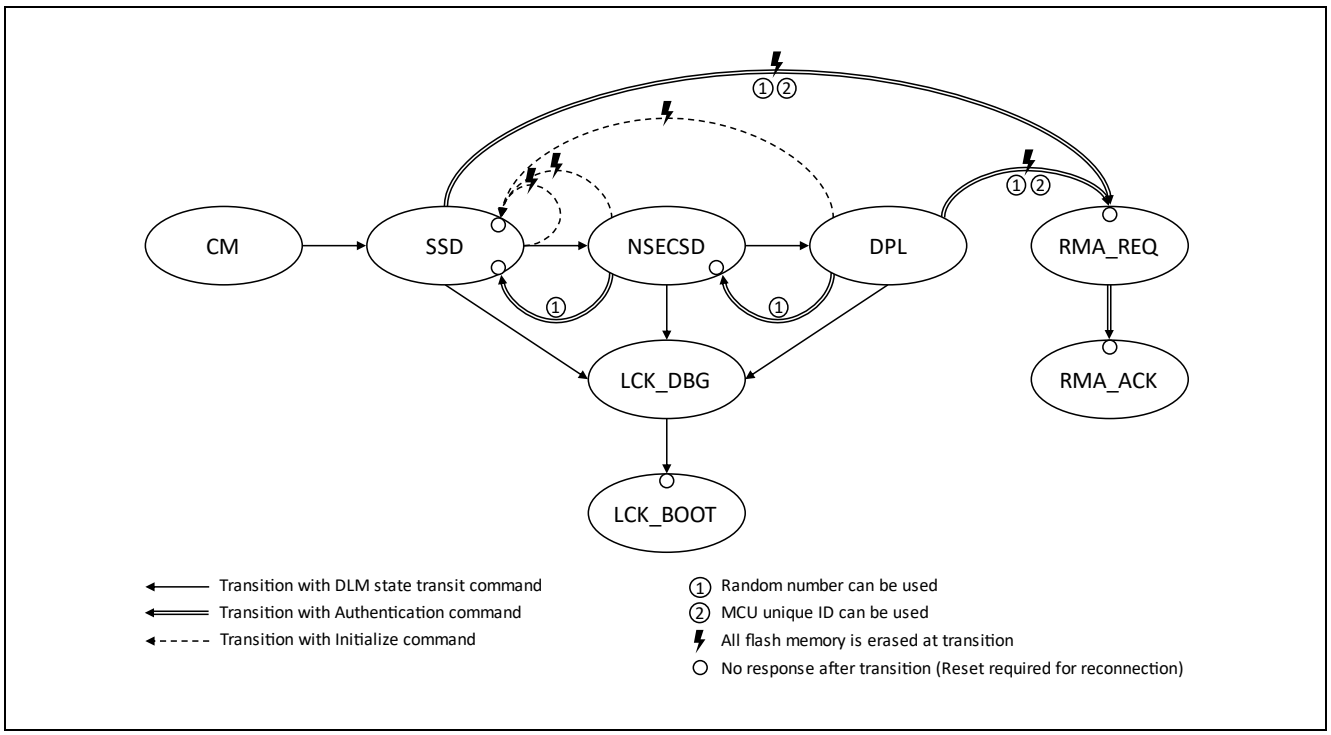
6. Command List

Name	Device (*1)			DLM state(*2) for GrpA, GrpB, and GrpC							ID Authentication for GrpD	
	GrpA GrpC	GrpB	GrpD	CM	SSD	NSECSD	DPL	LCK_DBG	LCK_BOOT	RMA_REQ	Unauthenti-ficated	Authenti-ficated
DLM state transit command	✔	✔		✔	✔	✔	✔	✔	(*3)			
Authentication command	✔	✔	✔	✔	✔	✔	✔		(*3)	✔	(*4)	
Key setting command	✔	✔			✔	✔			(*3)			
User key setting command	✔				✔	✔			(*3)			
Key verify command	✔	✔		✔	✔	✔	✔	✔	(*3)	✔		
User key verify command	✔			✔	✔	✔	✔	✔	(*3)	✔		
Initialize command	✔	✔			✔	✔	✔		(*3)			
DLM state request command	✔	✔		✔	✔	✔	✔	✔	(*3)	✔		
Boundary setting command	✔	✔			✔				(*3)			
Boundary request command	✔	✔		✔	✔	✔	✔	✔	(*3)	✔		
Parameter setting command	✔	✔			✔	✔	✔		(*3)			
Parameter request command	✔	✔		✔	✔	✔	✔	✔	(*3)	✔		
Inquiry command	✔	✔	✔	✔	✔	✔	✔	✔	(*3)	✔		✔
Signature request command	✔	✔	✔	✔	✔	✔	✔	✔	(*3)	✔	✔	✔
Area information request command	✔	✔	✔	✔	✔	✔	✔	✔	(*3)	✔	✔	✔
Baud rate setting command	✔	✔	✔	✔	✔	✔	✔	✔	(*3)	✔	✔	✔
Erase command	✔	✔	✔		✔	✔			(*3)			✔
Write command	✔	✔	✔		✔	✔			(*3)			✔
Read command	✔	✔	✔		✔	✔			(*3)			✔
CRC command	✔	✔	✔	✔	✔	✔	✔	✔	(*3)	✔	✔	✔

*1 : The command is available in the device. If an unavailable command is sent, boot firmware returns “Unsupported command error”.
*2 : The command is available in the state. If an unavailable command is sent, boot firmware returns “Command acceptance error”.
*3 : LCK_BOOT state never transits to the command acceptable phase because boot firmware executes software reset in the initialization phase.
*4 : Available when ID(Bit127 = 0b1) is set and ID is unauthenticated.

6.1 Device Lifecycle Management (GrpA, GrpB, and GrpC)

The following DLM state transitions can be triggered by each command:

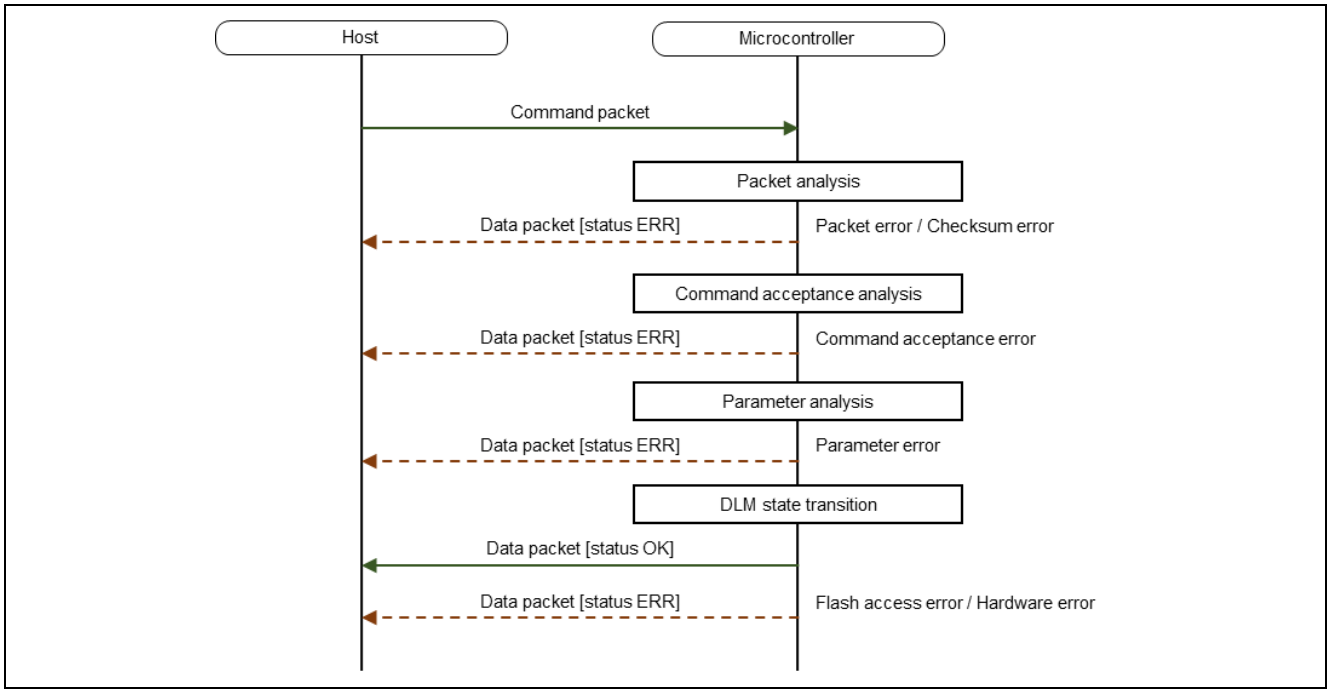


6.2 DLM state transit command (GrpA, GrpB, and GrpC)

This command transitions the DLM state without authentication. The boot firmware is hung when the DLM state transitions to LCK_BOOT.

This command requires adherence to conditions described in Command List.

6.2.1 Sequence diagram



6.2.2 Packets

6.2.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x03
CMD	(1 byte)	0x71 (DLM state transit command)
SDLM	(1 byte)	Source DLM state code
DDLM	(1 byte)	Destination DLM state code
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.2.2.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x71 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0x8D
ETX	(1 byte)	0x03

6.2.2.3 Data packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xF1 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.2.2.4 Data packet [status ERR] (Flash access error in disclosed area)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xF1 (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.2.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- If this command cannot be executed in the current DLM state, the boot firmware sends a “Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the command acceptance analysis is successfully completed, boot firmware analyzes the parameters.

- When SDLM is different from the current DLM state, boot firmware returns “Parameter error”.
- When DDLM is a DLM state that cannot transition from the current DLM state without authentication, boot firmware returns “Parameter error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the parameter analysis has successfully completed, boot firmware transitions to the DLM state.

- If an error occurs during transition to the DLM state, a “Flash access error” is returned but waits for the next command.
  * Check the DLM state after the Flash access error occurred with the DLM state request command.
- If the DLM state after the transition is an invalid value, the boot firmware sends a “Hardware error” and becomes unresponsive.
  Also, if the DLM state after transition is LCK_BOOT, the boot firmware sends “OK” and does not respond.
- When the DLM state transitions successfully, “OK” is returned and waits for the next command.

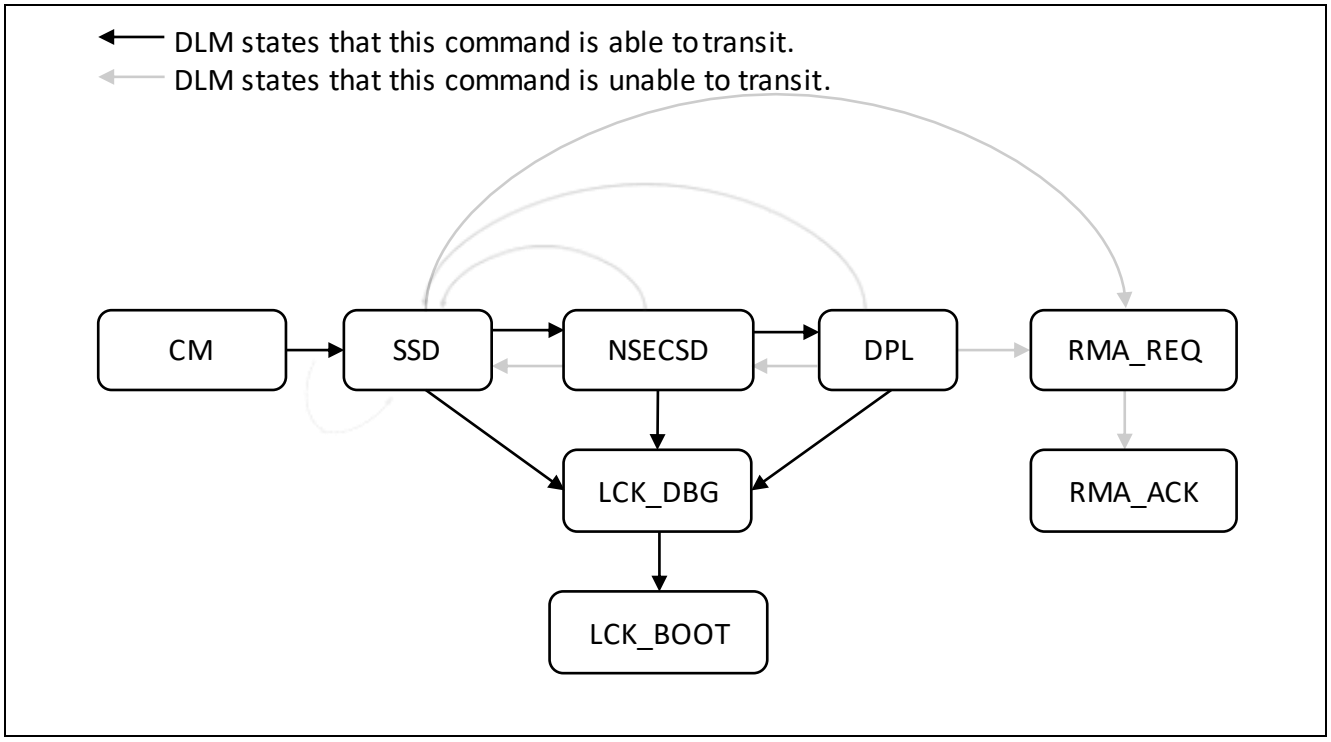
6.2.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in the current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
SDLM is different from the current DLM state.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
DDLM is not transitionable to DLM state.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after command execution in a nondisclosed area.	Flash access error	Flash status	0xFFFFFFFF
DLM state is abnormal.	Hardware error	0xFFFFFFFF	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.2.5 DLM state transition

The following shows the DLM state that can be transit by this command.



6.3 Authentication command

Command outline for GrpA, GrpB, and GrpC, Command outline for GrpD

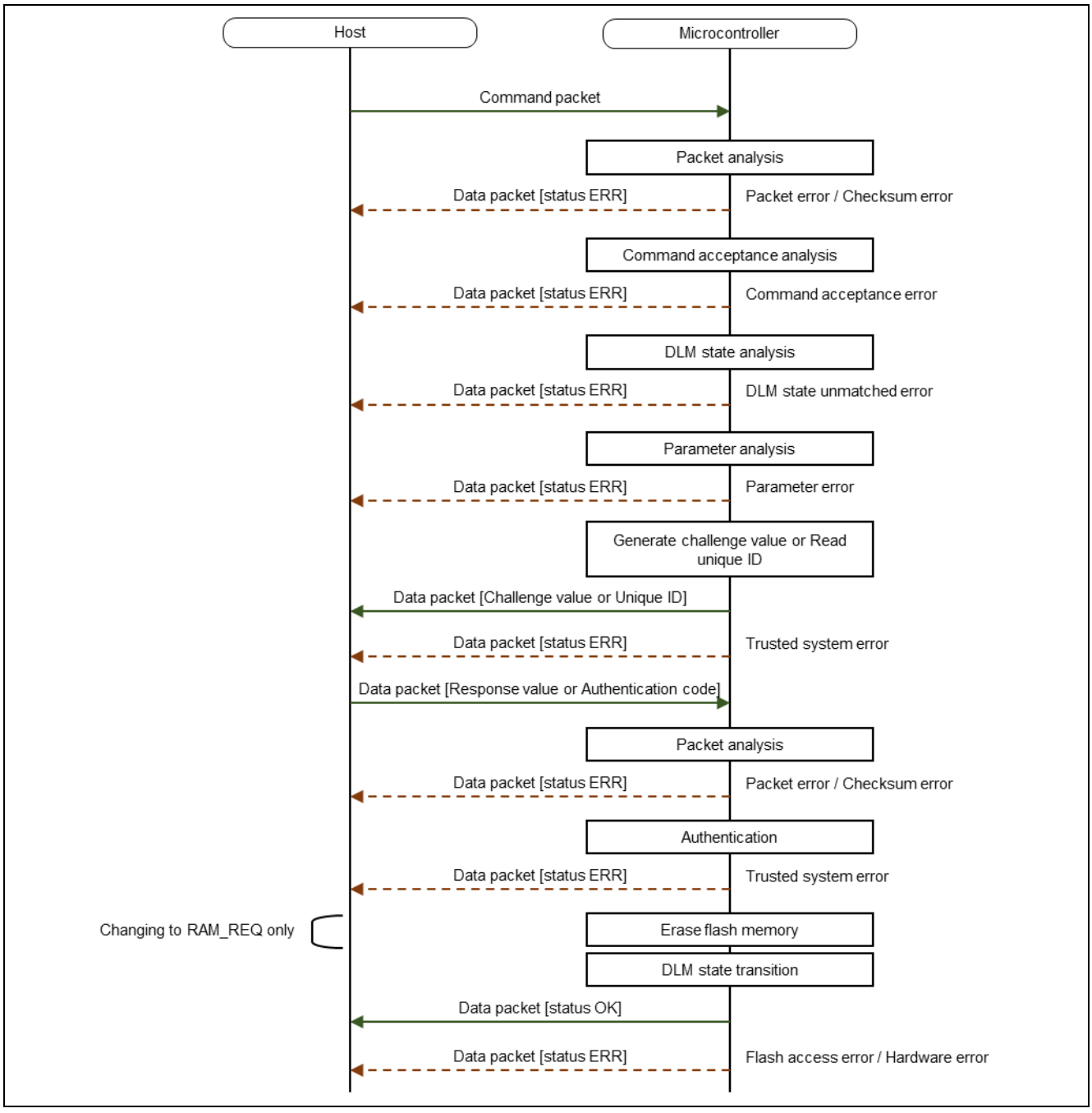
6.3.1 Command outline for GrpA, GrpB, and GrpC

This command authenticates using a key and transition the DLM state. Authentication is executed by the challenge and response method. Boot firmware erases the flash memory when the DLM state transits to RMA_REQ. Erase processing at this time is not affected by the block protection settings (BPS, BPS_SEC).

This command requires adherence to conditions described in Command List.

GrpA, GrpB : Response = HMAC-SHA256(Key, 128-bit challenge || Fixed value(256bit));
GrpC : Response = AES-128 CMAC(Key, 128-bit challenge);

6.3.2 Sequence diagram for GrpA, GrpB, and GrpC



6.3.3 Packets for GrpA, GrpB, and GrpC

6.3.3.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x04	
CMD	(1 byte)	0x30 (Authentication command)	
SDLM	(1 byte)	Source DLM state code	
DDLM	(1 byte)	Destination DLM state code	
CHCT	(1 byte)	Authentication type	0x00 : Random number (can be used in all transit cases.) 0x01 : MCU unique ID (can be used only in transit to RMA_REQ.)
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.3.3.2 Data packet [Challenge value or Unique ID]

SOD	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x11	
RES	(1 byte)	0x30 (OK)	
CHCD	(16 bytes)	Challenge value or Unique ID	e.g.) 0x01234567_89AB ... 2233_44556677 -> 0x01, 0x23, 0x45, ... , 0x55, 0x66, 0x77
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.3.3.3 Data packet [Response value or Authentication code]

SOD	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x21	
RES	(1 byte)	0x30 (OK)	
MAC	(32 bytes) (*1)	Response value or Authentication code	e.g.) 0x01234567_89AB ... 2233_44556677 -> 0x01, 0x23, 0x45, ... , 0x55, 0x66, 0x77
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

*1: Please fill "1" to lower 16 byte of MAC on Data Packet due to calculated Response by AES-128 CMAC in GrpC is 16 bytes .

6.3.3.4 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x30 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xCE
ETX	(1 byte)	0x03

6.3.3.5 Data packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xB0 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.3.3.6 Data packet [status ERR] (Flash access error in disclosed area)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xB0 (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details e.g.) FSTATR[31:0] = 0x0120A000 -> 0x01, 0x20, 0xA0, 0x00
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.3.3.7 Data packet [status ERR] (Flash access error in not disclosed area)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xB0 (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.3.4 Processing procedure for GrpA, GrpB, and GrpC

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- If this command cannot be executed in the current DLM state, the boot firmware sends a “Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the command reception analysis ends normally, the boot firmware performs DLM state analysis.

- If the currently active DLM state does not match the stored DLM state, the boot firmware sends a “DLM state unmatched error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware analyzes the parameters.

- When SDLM is different from the current DLM state, boot firmware returns “Parameter error” after 5 seconds.
- When DDLM is a DLM state that cannot transit from the current DLM state with authentication, boot firmware returns “Parameter error” after 5 seconds.
- When CHCT is not of challenge type that can be used for transition of DLM state, boot firmware returns “Parameter error” after 5 seconds.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the parameter analysis has successfully completed, boot firmware sends data packet [Challenge value or Unique ID].

- If the Challenge value / Unique ID is successfully generated, the boot firmware sends the value.
- If the Trusted system becomes abnormal after the Challenge value / Unique ID generation, the boot firmware returns nothing and no response.

* Flash memory status does not change before command reception.

- If the Challenge value / Unique ID generation fails, the boot firmware sends a “Trusted system error” and returns to the command wait state.
  * Flash memory status does not change before command reception.

When sending the data packet [Challenge value or Unique ID] has successfully completed, boot firmware receives and analyzes a data packet [Response value or Authentication code].

- Boot firmware detects the beginning of a data packet by receiving SOD.
  When boot firmware receives other data than SOD, it discards the data and waits for the next data until SOD is sent.
- When the received data packet does not have ETX, “Packet error” is returned after 5 seconds.
- When SUM in the received data packet is different from the value calculated by boot firmware, “Checksum error” is returned after 5 seconds.
- When LNH and LNL in the received data packet do not comply with the packet format, “Packet error” is returned after 5 seconds.
- When RES in the received data packet is different from defined values, “Packet error” is returned after 5 seconds.
- When LNH and LNL in the received data packet do not comply with the specifications of this command, “Packet error” is returned after 5 seconds.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware authenticates with received Response value or Authentication code.

- If the Trusted system becomes abnormal after authentication, the boot firmware returns nothing and no response.

* Flash memory status does not change before command reception.

- When authentication fails, “Trusted system error” is returned and waits for the next command.
  * Flash memory status does not change before command reception.

When authentication has successfully completed and DLM state transit to RMA_REQ, boot firmware erases the flash memory.

  * This command erases the flash memory even if initialization is disabled. Refer to the Parameter request command.

- If an error occurs during erasure in the Block protect the setting, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the Block protect setting is undefined.
- If an error occurs during erasure in the User area, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the area after ADR (failure address) of the User area is undefined.
- If an error occurs during erasure in the Data area, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the Data area is undefined.
- If an error occurs during erasure in the Config area, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the Config area is undefined.
- If an error occurs during boundary setting and key index (wrapped key) erasure in the User area, the boot firmware sends a ”Flash access error” and returns to the command wait state.

When the Authentication has successfully completed (when transition to RMA_REQ, erase of flash memory is also successful), boot firmware transits the DLM state.

- If an error occurs during transition of the DLM state, “Flash access error” is returned but waits for the next command.
  * Check the DLM state after the Flash access error occurred with the DLM state request command.
- If the DLM state after the transition is an invalid value, the boot firmware sends a “Hardware error” and becomes unresponsive.
- If the specified error does not occur, the boot firmware sends “OK” and becomes unresponsive.
  * When the DLM state transitions to RMA_REQ, each area of the flash memory is in the following state:
  — User area is erased except for the following:
    - Blocks for which “0” is set for permanent block protection setting (PBPS, PBPS_SEC)
      * Not affected by block protection settings (BPS, BPS_SEC)
  — All Data areas are erased
  — The Config area is written with “1” except for the following:
    - Permanent block protection setting (PBPS, PBPS_SEC)
    - Block protection setting (BPS, BPS_SEC) for block which “0” is set for permanent block protection setting (PBPS, PBPS_SEC)
    - Secure Attribute setting for block protection (BPS_SEL)
    - FSPR and BTFLG when FSPR = 0

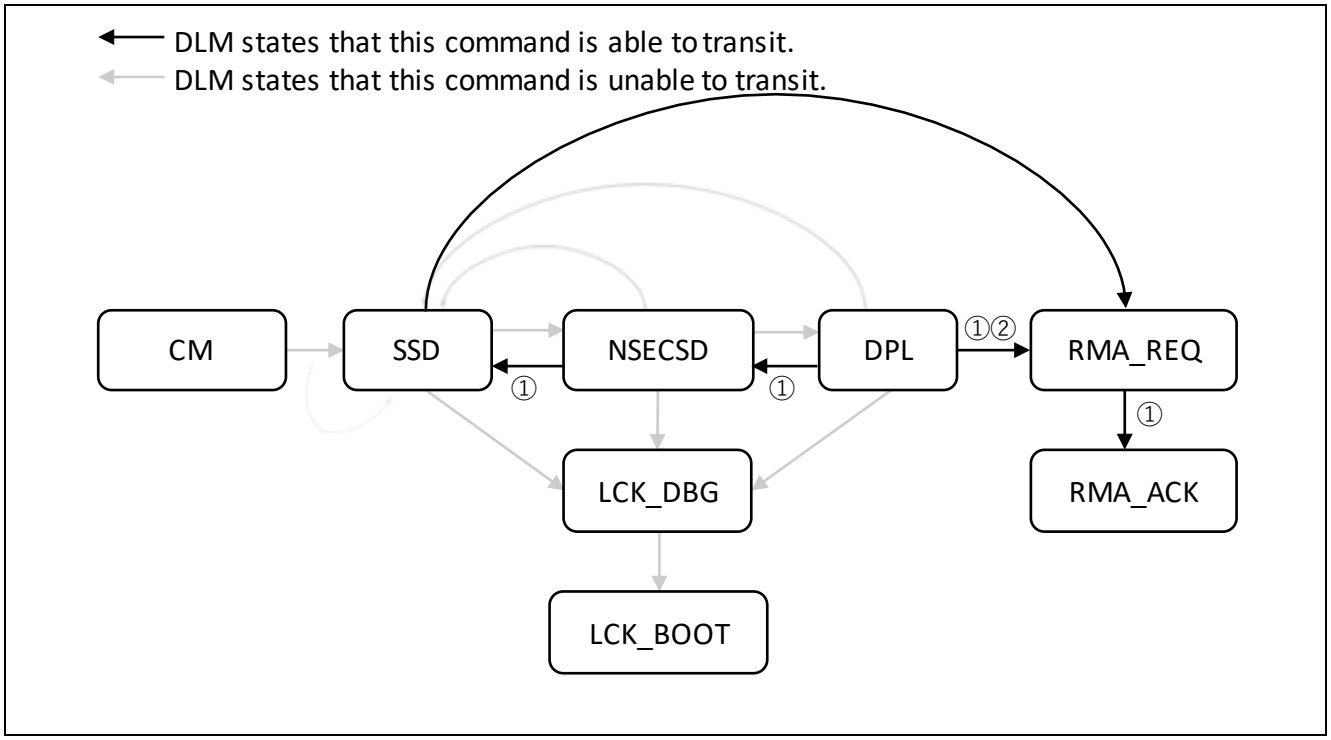
6.3.5 Status information from microcontroller for GrpA, GrpB, and GrpC

(listed in descending order of priority)

Condition	STS	FST	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
The currently active DLM state does not match the stored DLM state.	DLM state unmatched error	0xFFFFFFFF	0xFFFFFFFF
SDLM is different from current DLM state.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
DDLM is not transitionable DLM state.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
CHCT is a nonexistent challenge type.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Challenge value/Unique ID generation failed.	Trusted system error	0xFFFFFFFF	0xFFFFFFFF
The RES of the received data packet is different from the value specified by this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Authentication failed.	Trusted system error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after the command execution in disclosed area.	Flash access error	Flash status	Failure address
FACI detected an error after command execution in a nondisclosed area.	Flash access error	Flash status	0xFFFFFFFF
DLM state is abnormal.	Hardware error	0xFFFFFFFF	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.3.6 DLM state transition for GrpA, GrpB, and GrpC

The following diagram shows the DLM state that can be transit by this command.



6.3.7 Command outline for GrpD

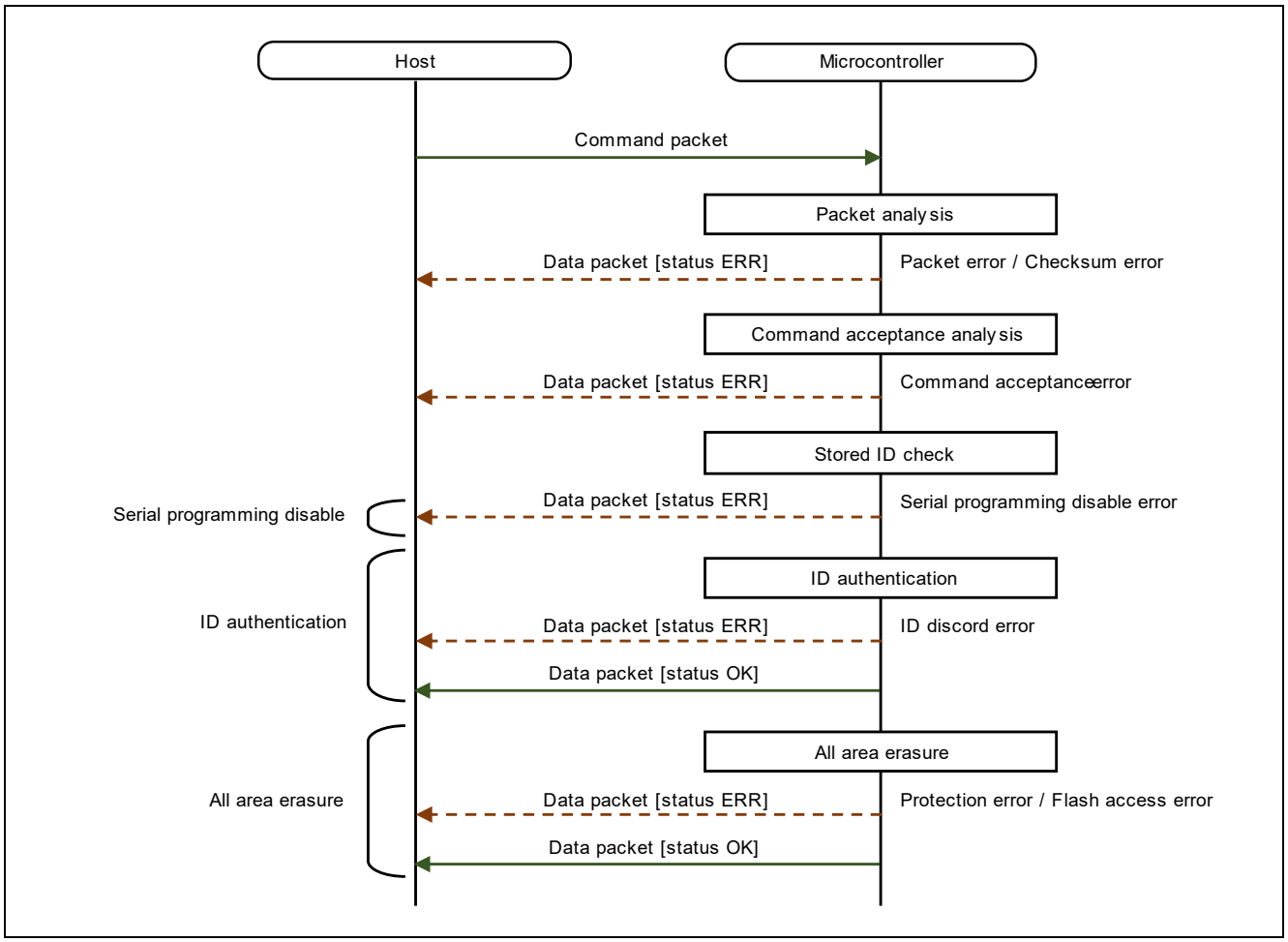
This command compares the ID-code stored in the microcontroller with the ID-code received from the flash programmer, and sends the result to the flash programmer. If the received ID code is "ALeRASE(*1)", all areas of the flash memory are erased without ID authentication. Erase processing at this time is not affected by the block protection settings (BPS, BPS_SEC). This command cannot be executed if the ID-code stored in the microcontroller is all "1". Also, this command cannot be executed after successful authentication until reset if the ID-code stored in the microcontroller is NOT all "1".

This command require adherence to conditions described in Command List.

*1: "ALeRASE" = 0x41, 0x4C, 0x65, 0x52, 0x41, 0x53, 0x45, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF

Condition		Processing	
stored ID[127:0] = All "1"		Non-secure device	-> Return to command acceptance (Other commands become executable)
stored ID[127] = 0b		Serial programming disable	-> Serial programming disable error (Enter an infinite loop)
stored ID[127:126] = 10b		ID authentication	when ID mismatch -> ID discord error (Enter an infinite loop) when ID match -> Return to command acceptance (Other commands become executable)
stored ID[127:126] = 11b	received ID != "ALeRASE"		
	received ID = "ALeRASE"	All area erasure	when FSPR = 0 -> Protection error (Enter an infinite loop) when Error occurred -> Flash access error (Enter an infinite loop) when Successful erasure -> Return to command acceptance (Other commands become executable)

6.3.8 Sequence diagram for GrpD



6.3.9 Packets for GrpD

6.3.9.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x11	
CMD	(1 byte)	0x30 (Authentication command)	
IDC	(16 bytes)	ID code	e.g.) If stored ID is as follows, the Host should send IDC in order shown in the lower table. stored ID: ID[127:96] | ID[95:64] 0xF0 0xF1 0xF2 0xF3 | 0xE4 0xE5 0xE6 0xE7 ID[63:32] | ID[31:0] 0xD8 0xD9 0xDA 0xDB | 0xCC 0xCD 0xCE 0xCF Order of sending IDC for ID authentication: 1st 2nd 3rd 4th 5th 6th 7th 8th 0xF0 0xF1 0xF2 0xF3 0xE4 0xE5 0xE6 0xE7 9th 10th 11th 12th 13th 14th 15th 16th 0xD8 0xD9 0xDA 0xDB 0xCC 0xCD 0xCE 0xCF e.g.) for All area erasure, the Host should send "ALeRASE" as IDC. Order of sending IDC for All area erasure: 1st 2nd 3rd 4th 5th 6th 7th 8th 0x41 0x4C 0x65 0x52 0x41 0x53 0x45 0xFF 9th 10th 11th 12th 13th 14th 15th 16th 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.3.9.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x30 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 byte)	0xFFFFFFFF (unused code)
ADR	(4 byte)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.3.9.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xB0 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.3.10 Processing procedure for GrpD

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a "Packet error".
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a "Checksum error".
- If the received command packet's LNH and LNL are different from the values specified in the packet format, the boot firmware sends a "Packet error".
- If the received command packet's LNH and LNL are different from the values specified in each command, the boot firmware sends a "Packet error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, boot firmware executes the acceptance analysis.

- No setting OCD/Serial ID, or already authenticated, the boot firmware sends a "Command acceptance error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware checks for "Serial programming disable error" after the processing above.

- When bit 127 of the ID code stored in the device is 0, boot firmware sends a "Serial programming disable error" and finishes command processing.
  Boot firmware does not accept any commands nor response for the commands after sending this error.
  * Flash memory status does not change before command reception.

When no error occurs, boot firmware decides whether to execute ID code authentication.

- Boot firmware execute ID code authentication when at least one of the following conditions are met.
  - Bit 126 of the ID code stored in the device is 0b0
  - Bit 126 of the ID code stored in the device is 0b1 but received IDC is not "ALeRASE"
- When ID code authentication fails, boot firmware sends "ID discord error" and finishes command processing.
  Boot firmware does not accept any commands nor response for the commands after sending this error.
  * Flash memory status does not change before command reception
- When ID code authentication passes, boot firmware send "OK" and finishes command processing.
  Then boot firmware transits to Command acceptable phase and wait for the next command.
  * Flash memory status does not change before command reception

When no error occurs and also ID code authentication is not executed, boot firmware executes flash memory all erasure.

- If there is a permanent protected block, boot firmware send "Protection error" and does not execute all erasure but finishes command processing.
  Boot firmware does not accept any commands nor response for the commands after sending this error.
  * Flash memory status does not change before command reception
- When FSPR in Config area is set, e.g. the value is 0, boot firmware send "Protection error" and does not execute all erasure but finishes command processing.
  Boot firmware does not accept any commands nor response for the commands after sending this error.
  * Flash memory status does not change before command reception
- If the above error does not occur, boot firmware execute all erasure.
- When all erasure fails, boot firmware receives a data packet and returns "Erase error", "Write error" or "Sequencer error" and finishes command processing.
  Boot firmware does not accept any commands nor response for the commands after sending this error.
  * Refer to Status code for conditions of the errors
  * Flash memory area are undefined.
- When all erasure passes, boot firmware send "OK" and finishes command processing.
  Then boot firmware transits to Command acceptable phase and wait for the next command.
  * All flash memory area are erased state

6.3.11 Status information from microcontroller for GrpD

(listed in descending order of priority)

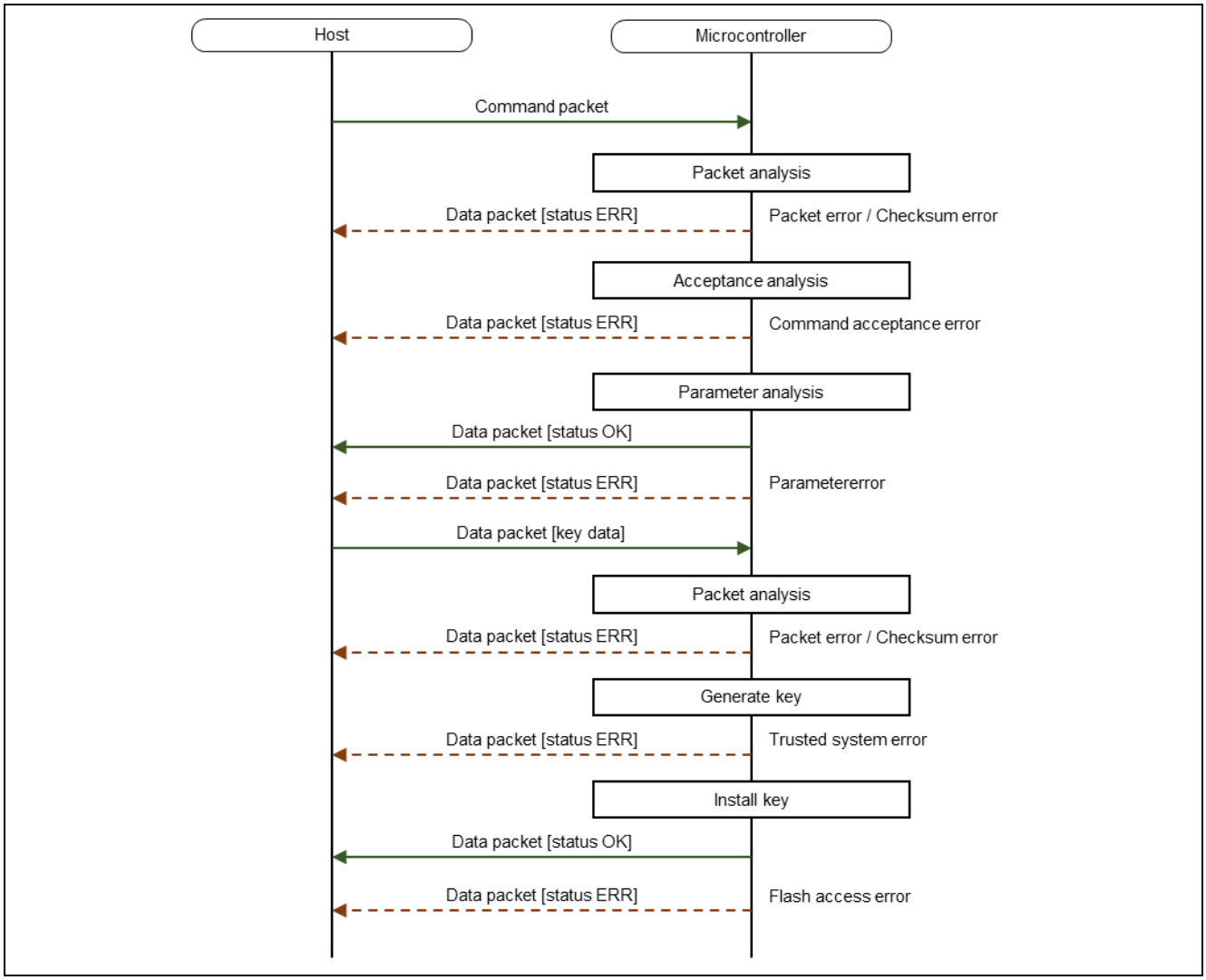
Condition	STS	ST2	ADR
The received packet doesn't have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the received packet is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet do not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet do not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
No setting OCD/Serial ID, or already authenticated. ("Status OK" has been sent as a response to the command packet.)	Command acceptance error	FFFFFFFFh	FFFFFFFFh
If the Highest-order bit of stored ID is "0".	Serial programming disable error	FFFFFFFFh	FFFFFFFFh
The ID authentication fails.	ID discord error	FFFFFFFFh	FFFFFFFFh
Permanent block protection is set even in 1bit.	Protection error	FFFFFFFFh	FFFFFFFFh
The FSPR bit is set. (FSPR = 0)	Protection error	FFFFFFFFh	FFFFFFFFh
FACI detected an error after the command execution in disclosed area.	Flash access error	Flash status	Failure address
FACI detected an error after the command execution in not disclosed area.	Flash access error	Flash status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.4 Key setting command (GrpA, GrpB, and GrpC)

This command sets the authentication key to the device. The authentication key must be specified in the DLM state.

This command requires adherence to conditions described in Command List.

6.4.1 Sequence diagram



6.4.2 Packets

6.4.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x02	
CMD	(1 byte)	0x28 (Key setting command)	
KYTY	(1 byte)	Key type	0x01 : SECDBG_KEY 0x02 : NONSECDBG_KEY 0x03 : RMA_KEY
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.4.2.2 Data packet [key data]

<table>
<tr><td>SOD</td><td>(1 byte)</td><td>0x81</td><td></td></tr>
<tr><td>LNH</td><td>(1 byte)</td><td>0x00</td><td></td></tr>
<tr><td>LNL</td><td>(1 byte)</td><td>0x51</td><td></td></tr>
<tr><td>RES</td><td>(1 byte)</td><td>0x28 (OK)</td><td></td></tr>
<tr><td>ESKY</td><td>(32 bytes)</td><td>Session key (W-UFPK)</td><td>e.g.) 0x01234567_89AB ... 2233_44556677 -> 0x01, 0x23, 0x45, ... , 0x55, 0x66, 0x77</td></tr>
<tr><td>IVEC</td><td>(16 bytes)</td><td>Initialization vector</td><td>e.g.) 0x01234567_89AB ... 2233_44556677 -> 0x01, 23, 0x45, ... , 0x55, 0x66, 0x77</td></tr>
<tr><td>EOKY</td><td>(32 bytes)</td><td>Install data (encrypted key | MAC)</td><td>Encrypted key (0-15 bytes) + MAC (16-31 bytes)
e.g.) If install data is as follows, the host must send EOKY in the order shown in the table that follows:

Install data:
<table>
<tr><th colspan="8">Encrypted key</th></tr>
<tr><td>0x00</td><td>0x01</td><td>0x02</td><td>0x03</td><td>0x04</td><td>0x05</td><td>0x06</td><td>0x07</td></tr>
<tr><td>0x08</td><td>0x09</td><td>0x0A</td><td>0x0B</td><td>0x0C</td><td>0x0D</td><td>0x0E</td><td>0x0F</td></tr>
<tr><th colspan="8">MAC</th></tr>
<tr><td>0x10</td><td>0x11</td><td>0x12</td><td>0x13</td><td>0x14</td><td>0x15</td><td>0x16</td><td>0x17</td></tr>
<tr><td>0x18</td><td>0x19</td><td>0x1A</td><td>0x1B</td><td>0x1C</td><td>0x1D</td><td>0x1E</td><td>0x1F</td></tr>
</table>
Order of sending EOKY:
<table>
<tr><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th><th>5th</th><th>6th</th><th>7th</th><th>8th</th></tr>
<tr><td>0x00</td><td>0x01</td><td>0x02</td><td>0x03</td><td>0x04</td><td>0x05</td><td>0x06</td><td>0x07</td></tr>
<tr><th>9th</th><th>10th</th><th>11th</th><th>12th</th><th>13th</th><th>14th</th><th>15th</th><th>16th</th></tr>
<tr><td>0x08</td><td>0x09</td><td>0x0A</td><td>0x0B</td><td>0x0C</td><td>0x0D</td><td>0x0E</td><td>0x0F</td></tr>
<tr><th>17th</th><th>18th</th><th>19th</th><th>20th</th><th>21st</th><th>22nd</th><th>23rd</th><th>24th</th></tr>
<tr><td>0x10</td><td>0x11</td><td>0x12</td><td>0x13</td><td>0x14</td><td>0x15</td><td>0x16</td><td>0x17</td></tr>
<tr><th>25th</th><th>26th</th><th>27th</th><th>28th</th><th>29th</th><th>30th</th><th>31st</th><th>32nd</th></tr>
<tr><td>0x18</td><td>0x19</td><td>0x1A</td><td>0x1B</td><td>0x1C</td><td>0x1D</td><td>0x1E</td><td>0x1F</td></tr>
</table>
</td></tr>
<tr><td>SUM</td><td>(1 byte)</td><td>Sum data</td><td></td></tr>
<tr><td>ETX</td><td>(1 byte)</td><td>0x03</td><td></td></tr>
</table>

6.4.2.3 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x28 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xD6
ETX	(1 byte)	0x03

6.4.2.4 Data packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xA8 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.4.2.5 Data packet [status ERR] (Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xA8 (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.4.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- If this command cannot be executed in the current DLM state, the boot firmware sends a “Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the acceptance analysis has successfully completed, boot firmware executes the parameter analysis.

- When KYTY is not of Key type that can be set in the current DLM state, boot firmware returns “Parameter error” and waits for the next command.
  * Flash memory status does not change before command reception.
- If the specified error does not occur, the boot firmware sends “OK”.

When parameter analysis has successfully completed, boot firmware received and analyzes data packet.

- Boot firmware detects the beginning of a data packet by receiving SOD.
  When boot firmware receives other data than SOD, it discards the data and waits for the next data until SOD is sent.
- When the received data packet does not have ETX, “Packet error” is returned after 5 seconds.
- When SUM in the received data packet is different from the value calculated by boot firmware, “Checksum error” is returned.
- When LNH and LNL in the received data packet do not comply with the packet format, “Packet error” is returned.
- When RES in the received data packet is different from defined values, “Packet error” is returned.
- When the number of received data exceeds the value specified in the command in the received data packet, “Parameter error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware generates and set authentication key after parameter analysis.

- If the Trusted system becomes abnormal after creating a key index (wrapped key), the boot firmware returns nothing and no response.
  * Flash memory status does not change before command reception.

- If creation of key index (wrapped key) fails, the boot firmware sends a “Trusted system error” and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware writes authentication key after key generation.

- If an error occurs while writing key index (wrapped key), the boot firmware sends a “Flash access error” and. returns to the command wait state.
  * Use Key verify command to check the status of the key index (wrapped key) after Flash access error occurred.
- When authentication key setting has successfully completed, boot firmware returns “OK” and waits for the next command.

6.4.4 Status information from microcontroller

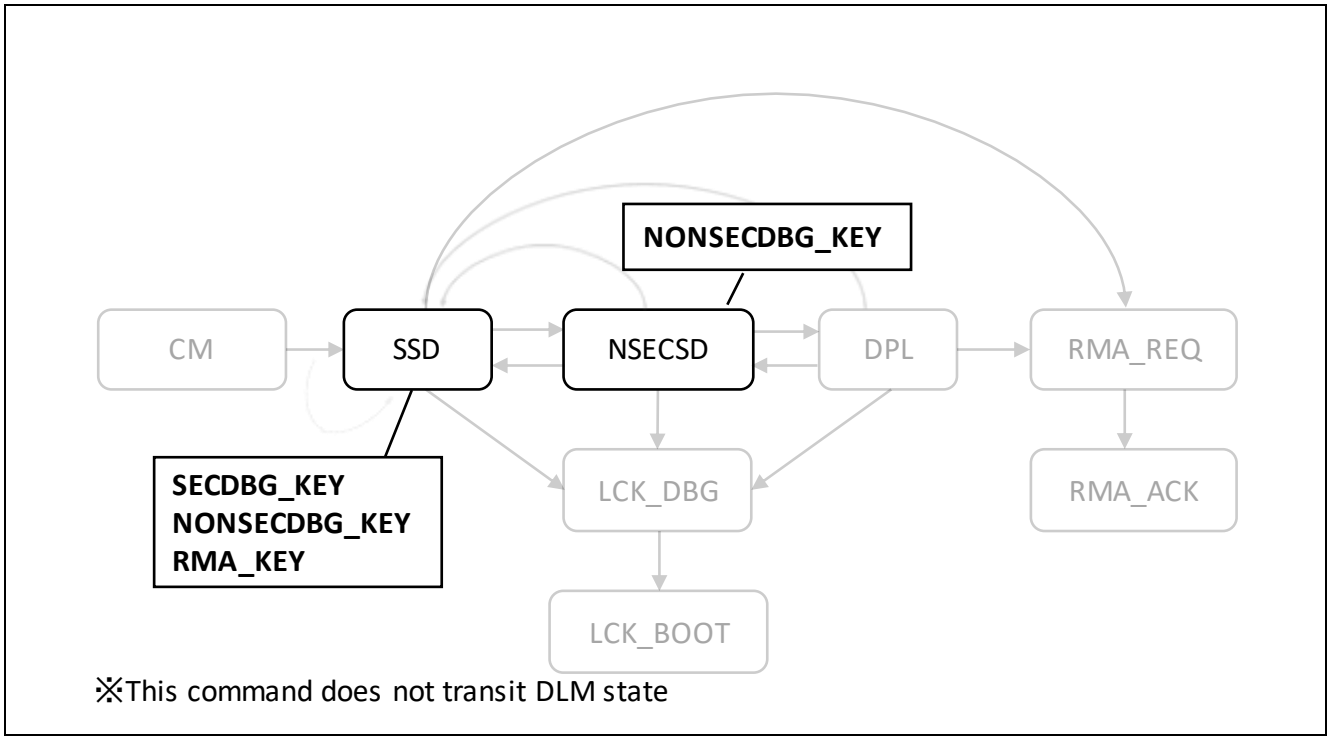
(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in the current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
KYTY cannot set the current DLM state.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The RES of the received data packet is different from the value specified by this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
The total length of received data of data packets exceeds the value specified in the command.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Authentication key generated failed.	Trusted system error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after command execution in a nondisclosed area.	Flash access error	Flash status	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.4.5 DLM state that can be set

The following table shows the DLM state that can be set for Key type.

Key type	DLM state
SECDBG_KEY	SSD
NONSECDBG_KEY	SSD, NSECSD
RMA_KEY	SSD

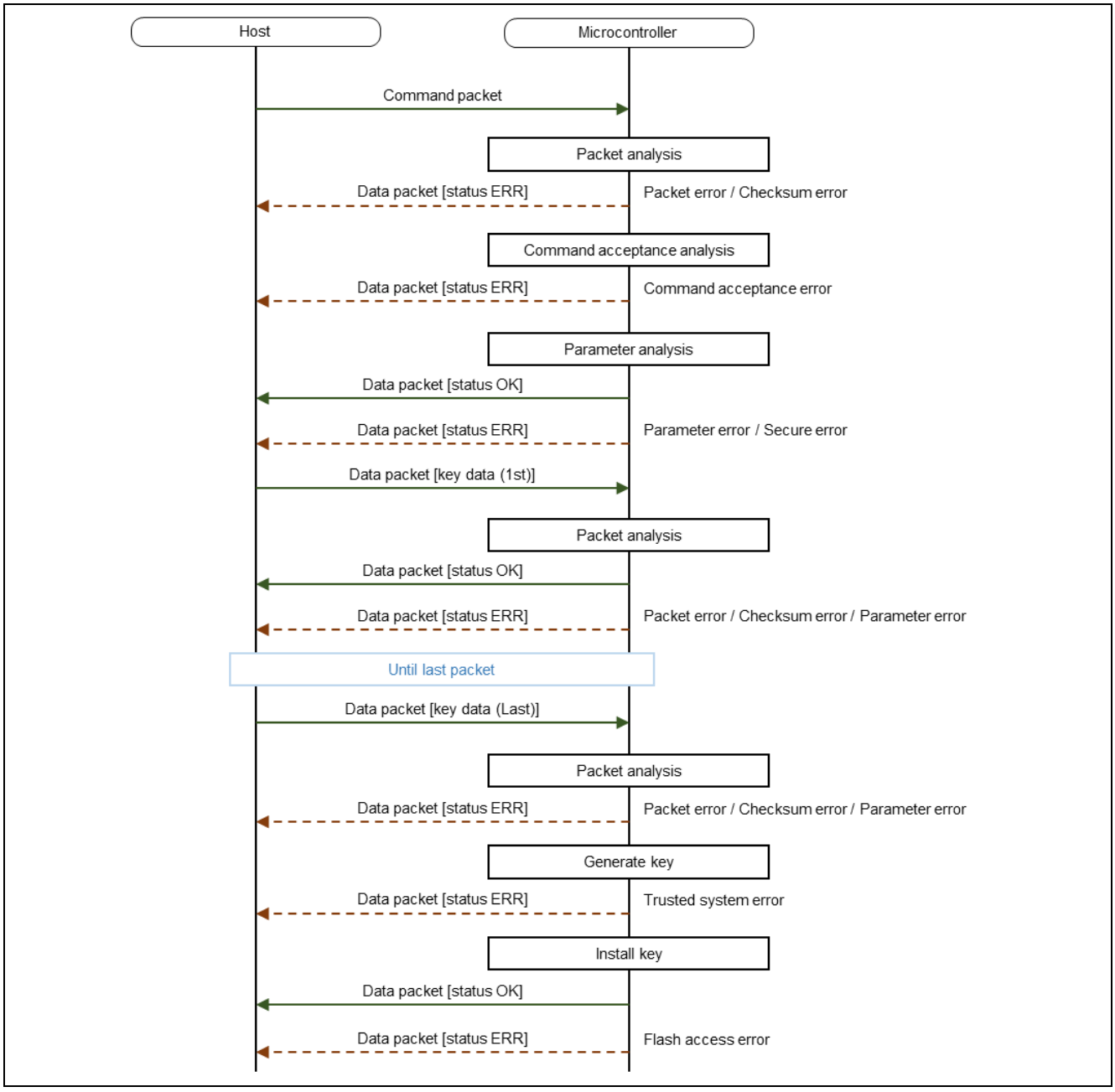


6.5 User key setting command (GrpA and GrpC)

This command generates key index (wrapped key) using session key (W-UFPK) and install data (encrypted key | MAC) received from the host, and saves it in the specified area. Write processing at this time is not affected by the block protection settings (BPS, BPS_SEC). The storage area must be erased in advance.

This command requires adherence to conditions described in Command List.

6.5.1 Sequence diagram



6.5.2 Packets

6.5.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x06	
CMD	(1 byte)	0x2A (User key setting command)	
KADR	(4 bytes)	Key setting address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
ENTY	(1 byte)	User key type	Reference to User key list
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.5.2.2 Data packet [key data (1st)]

SOD	(1 byte)	0x81	
LNH	(1 byte)	N + 49 (Higher 1 byte)	
LNL	(1 byte)	N + 49 (Lower 1 byte)	
RES	(1 byte)	0x2A (OK)	
ESKY	(32 bytes)	Session key (W-UFPK)	e.g.) 0x01234567_89AB ... 2233_44556677 -> 0x01, 0x23, 0x45, ... , 0x55, 0x66, 0x77
IVEC	(16 bytes)	Initialization vector	e.g.) 0x01234567_89AB ... 2233_44556677 -> 0x01, 0x23, 0x45, ... , 0x55, 0x66, 0x77
ENKY	(N byte)	Install data (encrypted key | MAC)	e.g) If the key type is ECC P-192 Private Key, the host must send ENKY in the order shown in the table that follows.

Install data:

Encrypted Key							
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
MAC							
0x20	0x21	0x22	0x23	0x24	25	0x26	0x27
0x28	0x29	0x2A	0x2B	0x2C	2D	0x2E	0x2F

Order of sending ENKY:

1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
9th	**10th**	**11th**	**12th**	**13th**	**14th**	**15th**	**16th**
0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
17th	**18th**	**19th**	**20th**	**21st**	**22nd**	**23rd**	**24th**
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
25th	**26th**	**27th**	**28th**	**29th**	**30th**	**31st**	**32nd**
0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
33rd	**34th**	**35th**	**36th**	**37th**	**38th**	**39th**	**40th**
0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
41st	**42nd**	**43rd**	**44th**	**45th**	**46th**	**47th**	**48th**
0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F

SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

N = 1 to 976

6.5.2.3 Data packet [key data (2nd to last)]

SOD	(1 byte)	0x81
LNH	(1 byte)	N + 1 (Higher 1 byte)
LNL	(1 byte)	N + 1 (Lower 1 byte)
RES	(1 byte)	0x2A (OK)
ENTY	(N byte)	Install data (encrypted key | MAC) *Order of sending: Low -> ... -> High
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

N = 1 to 1024

6.5.2.4 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x2A (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xD4
ETX	(1 byte)	0x03

6.5.2.5 Data packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xAA (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.5.2.6 Data packet [status ERR] (Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xAA (ERR)
STS	(1 byte)	0xE5
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.5.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- If this command cannot be executed in the current DLM state, the boot firmware sends a “Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the command acceptance analysis has successfully completed, boot firmware analyzes the parameters.

- If ENTY is not specified as Key type, the boot firmware sends a “Parameter error”.
- If the area for key index size from KADR is not included in the User area or Data area specified in the area information, the boot firmware sends a “Parameter error”.
- If the area from KADR to key index size is across different KOAs, the boot firmware sends a “Parameter error”.
- If the WAU for the specified area is 0, the boot firmware sends a “Parameter error”.
- If KADR is not specified in the WAU of the area, the boot firmware sends a “Parameter error”.
- If the current DLM state is NSECSD and the specified range includes a secure area, the boot firmware sends a “Secure error”.
- If the area for the key index size from KADR contains a block with permanent block protection, the boot firmware sends a “Protection error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.
- If the above error does not occur, the boot firmware sends “OK”.

When the DLM state check has successfully completed, boot firmware changes the DLM state.

- Boot firmware detects the beginning of a data packet by receiving SOD.
  When boot firmware receives data other than SOD, it discards the data and waits for the next data until SOD is sent.
- When the received data packet does not have ETX, “Packet error” is returned after 5 seconds.
- When SUM in the received data packet is different from the value calculated by boot firmware, “Checksum error” is returned after 5 seconds.
- When LNH and LNL in the received data packet do not comply with the packet format, “Packet error” is returned after 5 seconds.
- When RES in the received data packet is different from defined values, “Packet error” is returned after 5 seconds.
- When the number of accumulated ENKY data exceeds the install data size indicated by ENTY in the received data packet, the boot firmware sends a “Parameter error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.
- If the key data has not been received, the boot firmware receives the next data packet.

When all key data has been received, the boot firmware generates a key index (wrapped key).

- If the Trusted system becomes abnormal after creating a key index (wrapped key), the boot firmware returns nothing and no response.
  * Flash memory status does not change before command reception.
- If creation of key index (wrapped key) fails, the boot firmware sends a “Trusted system error” and returns to the command waiting state.
  * Flash memory status does not change before command reception.

If the key index (wrapped key) is successfully generated, the boot firmware writes the key index to the specified area.

- If an error occurs while writing key index (wrapped key), the boot firmware sends a “Flash access error” and returns to the command wait state.
  * WAU size from failure address (ADR) of flash memory area is undefined.
- If the key index (wrapped key) is successfully saved to the device, the boot firmware sends “OK” and returns to the command wait state.

6.5.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
ENTY is not specified as Key type.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The area from KADR to key index size does not fit in the range of User area and Data area specified by area information.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The area from KADR to key index size spans different KOA.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The key storage area WAU is 0.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
KADR is not specified in the WAU for the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The current DLM state is NSECSD and KADR contains a secure area.	Secure error	0xFFFFFFFF	0xFFFFFFFF
There is a block with permanent block protection in the area from KADR to key index size.	Protection error	0xFFFFFFFF	0xFFFFFFFF
The RES of the received data packet is different from the value specified by this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
KDAT in the received packet is different from KLEN.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Key index (wrapped key) generation failed.	Trusted system error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after the command execution.	Flash access error	Flash status	Failure address
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.5.5 User key list

The following table shows a list of user keys specified by this command.

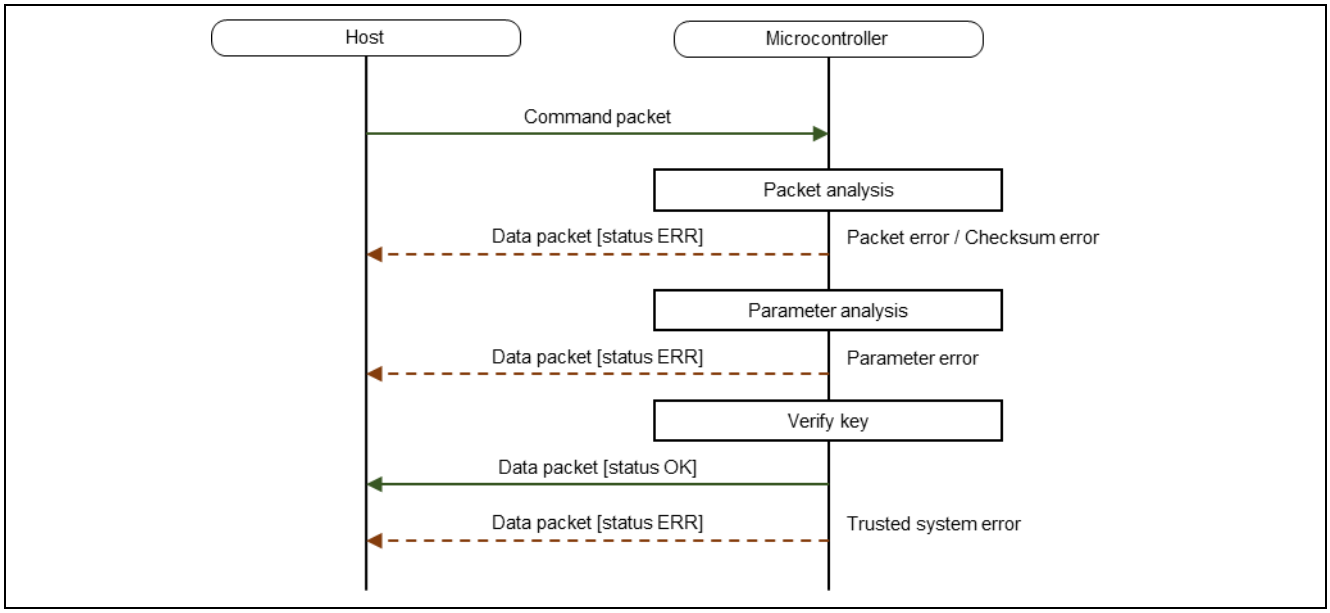
Key type	Device		Installation key		
	GrpA GrpB	GrpC		Install data size (byte)	Key index size (byte)
0x05	✔	✔	AES-128	32	36
0x06	✔		AES-192	48	52
0x07	✔	✔	AES-256	48	52
0x08	✔	✔	AES-128 XTS	48	52
0x09	✔	✔	AES-256 XTS	80	84
0x0A	✔		RSA-1024 Public Key	160	292
0x0B	✔		RSA-1024 Private Key	272	404
0x0C	✔		RSA-2048 Public Key	288	548
0x0D	✔		RSA-2048 Private Key	528	788
0x0E	✔		RSA-3072 Public Key	416	420
0x0F	✔		RSA-3072 Private Key	784	788
0x10	✔		RSA-4096 Public Key	544	548
0x11	✔		RSA-4096 Private Key	1040	1044
0x12	✔		ECC P192 Public Key	80	84
0x13	✔		ECC P192 Private Key	48	52
0x14	✔		ECC P224 Public Key	80	84
0x15	✔		ECC P224 Private Key	48	52
0x16	✔		ECC P256 Public Key	80	84
0x17	✔		ECC P256 Private Key	48	52
0x18	✔		ECC P384 Public Key	112	116
0x19	✔		ECC P384 Private Key	64	68
0x1A	✔		HMAC-SHA224	48	52
0x1B	✔		HMAC-SHA256	48	52
0x1C	✔		ECC P256r1 Public Key	80	84
0x1D	✔		ECC P256r1 Private Key	48	52
0x1E	✔		ECC P384r1 Public Key	112	116
0x1F	✔		ECC P384r1 Private Key	64	68
0x20	✔		ECC P512r1 Public Key	144	148
0x21	✔		ECC P512r1 Private Key	80	84
0x22	✔		ECC secp256k1 Public Key	80	84
0x23	✔		ECC secp256k1 Private Key	48	52
0xFF	✔	✔	Key update key	48	52

6.6 Key verify command (GrpA, GrpB, and GrpC)

This command verifies the authentication key that is set for the device.

This command requires adherence to conditions described in Command List.

6.6.1 Sequence diagram



6.6.2 Packets

6.6.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x02	
CMD	(1 byte)	0x29 (Key verify command)	
KYTY	(1 byte)	Key type	0x01: SECDBG_KEY 0x02: NONSECDBG_KEY 0x03: RMA_KEY
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.6.2.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x29 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xD6
ETX	(1 byte)	0x03

6.6.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xA9 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.6.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the parameter analysis.

- If KYTY is an unsupported key type, the boot firmware sends a “Parameter error” and returns to the command wait state.
  * Flash memory status does not change before command reception.

Boot firmware verifies the authentication key after parameter analysis.

- If verification of key index (wrapped key) fails, the boot firmware sends a “Trusted system error” and returns to the command wait state.
  * Flash memory status does not change before command reception.
- If the verification of the key index (wrapped key) has completed successfully, the boot firmware sends ”OK” and returns to the command wait state.
  * Flash memory status does not change before command reception.

6.6.4 Status information from microcontroller

(listed in descending order of priority)

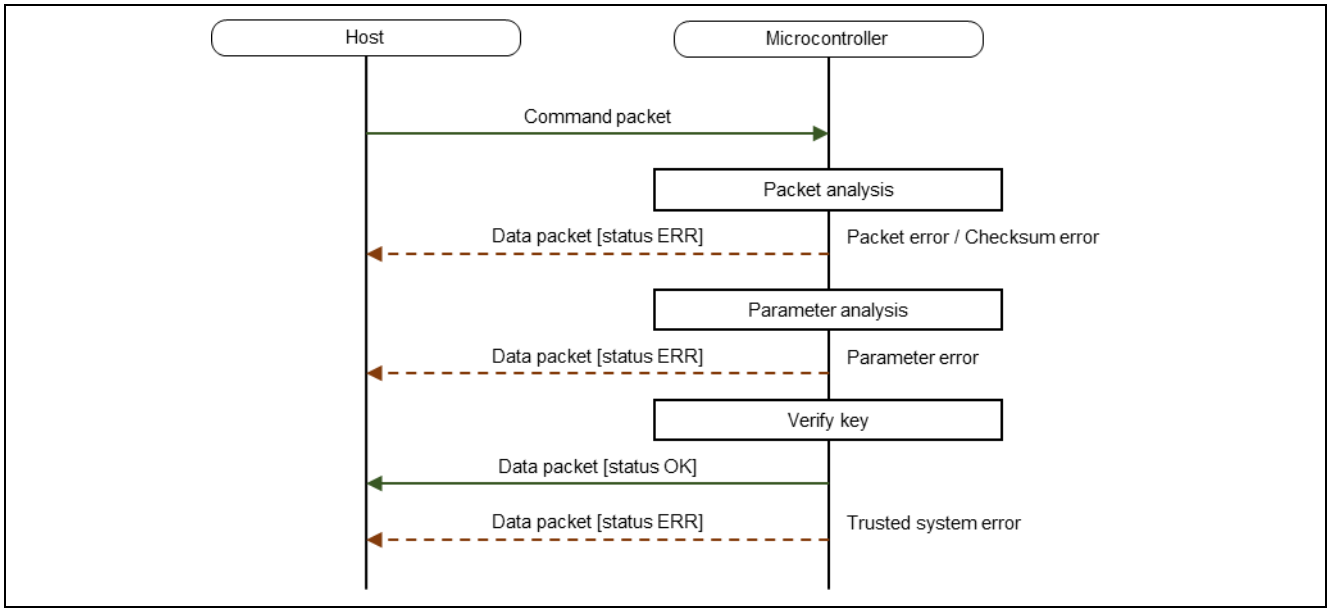
Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
KYTY is not supported key type.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Verification of the authentication key failed.	Trusted system error	0xFFFFFFFF	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.7 User key verify command (GrpA and GrpC)

This command verifies the authentication key that is set for the device. The authentication key must be specified in the DLM state to be read.

This command requires adherence to conditions described in Command List.

6.7.1 Sequence diagram



6.7.2 Packets

6.7.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x06	
CMD	(1 byte)	0x2B (User key verify command)	
KADR	(4 bytes)	Key address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
ENTY	(1 byte)	User key type	Supports the same key type as User key setting command
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.7.2.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x2B (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xD6
ETX	(1 byte)	0x03

6.7.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xAB (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.7.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the parameter analysis.

- If ENTY is not specified as key type, the boot firmware sends a “Parameter error”.
- If the area for key index size from KADR is not included in the User area or Data area specified in the area information, the boot firmware sends a “Parameter error”.
- If the area from KADR to key index size is across different KOAs, the boot firmware sends a “Parameter error”.
- If the WAU for the specified area is 0, the boot firmware sends a “Parameter error”.
- If KADR is not specified in the WAU of the area, the boot firmware sends a “Parameter error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware verifies the authentication key after parameter analysis.

- When there is a mismatch in the authentication key stored in the device, boot firmware returns “Trusted system error”.
  * Flash memory status does not change before command reception.
- If the above error does not occur, the boot firmware sends “OK”.
  * Flash memory status does not change before command reception.

6.7.4 Status information from microcontroller

(listed in descending order of priority)

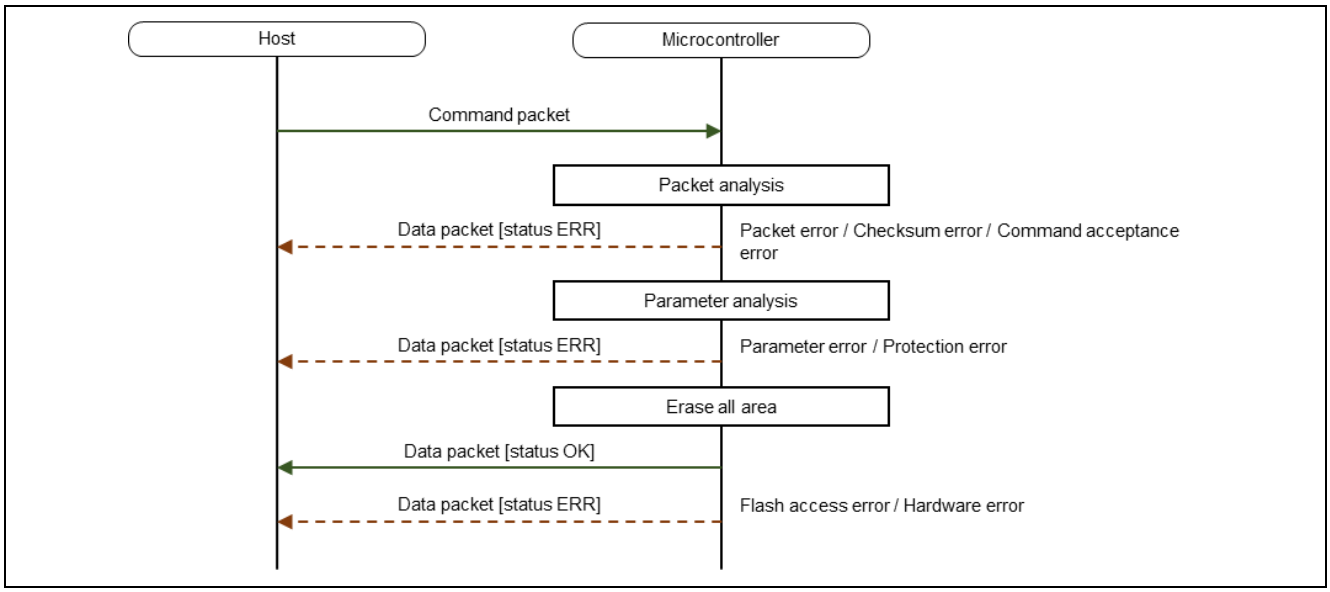
Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
ENTY is not specified as key type.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The area from KADR to key index size does not fit in the range of User area and Data area specified by area information.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Verify the authentication key failed.	Trusted system error	0xFFFFFFFF	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.8 Initialize command (GrpA, GrpB, and GrpC)

This command clears the User area, Data area, Config area, boundary setting, and key index (wrapped key). In addition, the DLM state transitions to SSD. Erase processing at this time is not affected by the block protection settings (BPS, BPS_SEC).

This command requires adherence to conditions described in Command List.

6.8.1 Sequence diagram



6.8.2 Packets

6.8.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x03	
CMD	(1 byte)	0x50 (Initialize command)	
SDLM	(1 byte)	Source DLM state code	0x02 : SSD 0x03 : NSECSD 0x04 : DPL
DDLM	(1 byte)	Destination DLM state code	0x02 : SSD
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.8.2.2 Data Packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x50 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xAE
ETX	(1 byte)	0x03

6.8.2.3 Data Packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xD0 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.8.2.4 Data packet [status ERR] (Flash access error in disclosed area)

SOH	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x0A	
RES	(1 byte)	0xD0 (ERR)	
STS	(1 byte)	0xE5 (Flash access error)	
ST2	(4 bytes)	Status details	e.g.) FSTATR[31:0] = 0x0120A000 -> 0x01, 0x20, 0xA0, 0x00
ADR	(4 bytes)	Failure address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.8.2.5 Data packet [status ERR] (Flash access error in not disclosed area)

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0Ah
RES	(1 byte)	0xD0 (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.8.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- If this command cannot be executed in the current DLM state, the boot firmware sends a “Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware analyzes the received parameters after packet analysis.

- When SDLM does not match with the current DLM state, “Parameter error” is returned.
- When DDLM is not SSD, “Parameter error” is returned.
- When initialization is disabled, “Protection error” is returned.
- When Permanent protected block exists (there is a bit that is “0” in PBPS[139:0] and PBPS_SEC[139:0]), “Protection error” is returned.
- When FSPR bit is 0, “Protection error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware executes the all erase processing after parameter analysis.

- If an error occurs during erasure in the block protect setting, the boot firmware sends a “Flash access error” and returns to the command wait state.

* The value of the block protect setting is undefined.

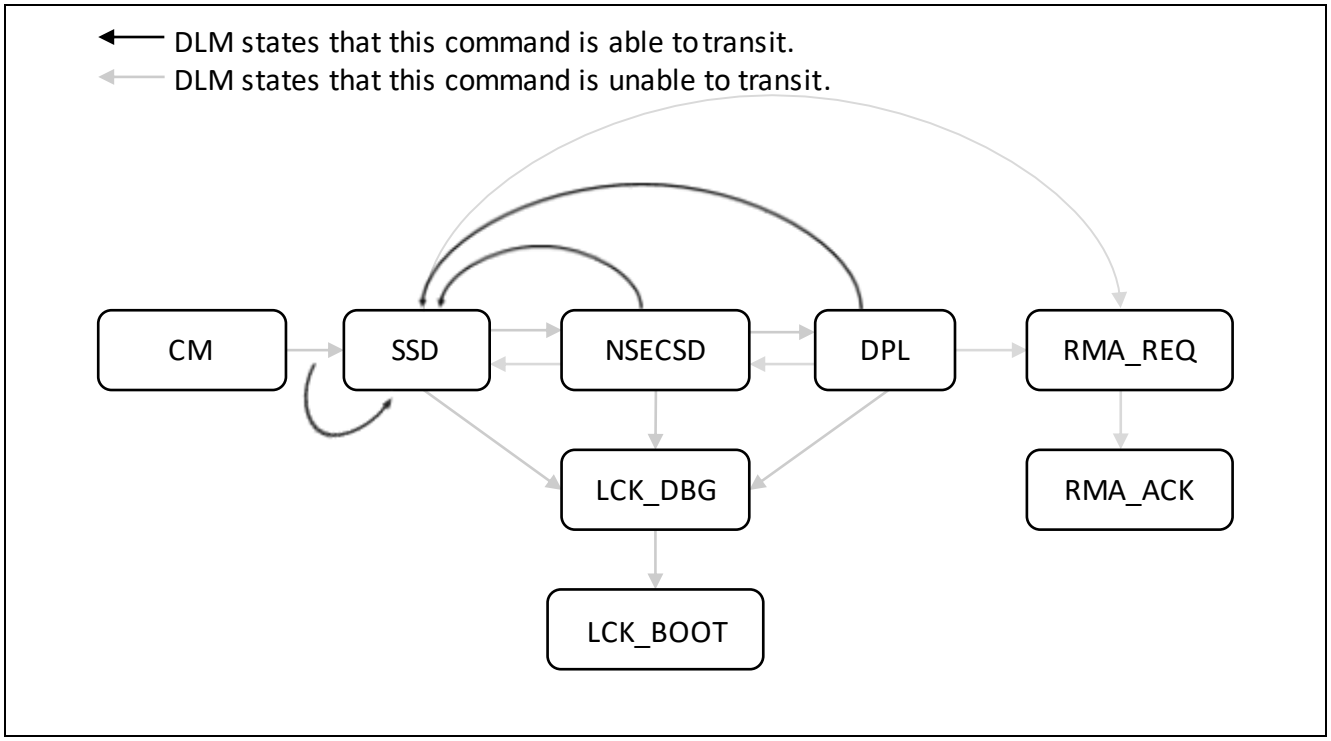
- If an error occurs during erasure in the User area, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the area after ADR (Failure address) of the User area is undefined.
- If an error occurs during erasure in the Data area, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the Data area is undefined.
- If an error occurs during erasure in the Config area, the boot firmware sends a “Flash access error” and returns to the command wait state.
  * The value of the Config area is undefined.
- If an error occurs during boundary setting and key index (wrapped key) erasure in the User area, the boot firmware sends a “Flash access error” and returns to the command wait state.
- If an error occurs during transition DLM state, “Flash access error” is returned but waits for the next command.
  * Check the DLM state after the Flash access error has occurred with the DLM state request command.
- If the DLM state is an invalid value, the boot firmware sends a “Hardware error” and becomes unresponsive.
- If all erasure process has completed normally, the boot firmware sends “OK” and no response.
  * The User area and Data area of flash memory are erased, the Config area is written (All-F), and the DLM state is SSD.

6.8.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
SDLM is different from current DLM state.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
DDLM is not SSD.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Initialization is disabled.	Protection error	0xFFFFFFFF	0xFFFFFFFF
There is a permanently protected block.	Protection error	0xFFFFFFFF	0xFFFFFFFF
The FPSR bit is set. (FSPR = 0)	Protection error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after the command execution in disclosed area.	Flash access error	Flash status	Failure address
FACI detected an error after the command execution in not disclosed area.	Flash access error	Flash status	0xFFFFFFFF
DLM state is abnormal.	Hardware error	0xFFFFFFFF	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.8.5 DLM state transition

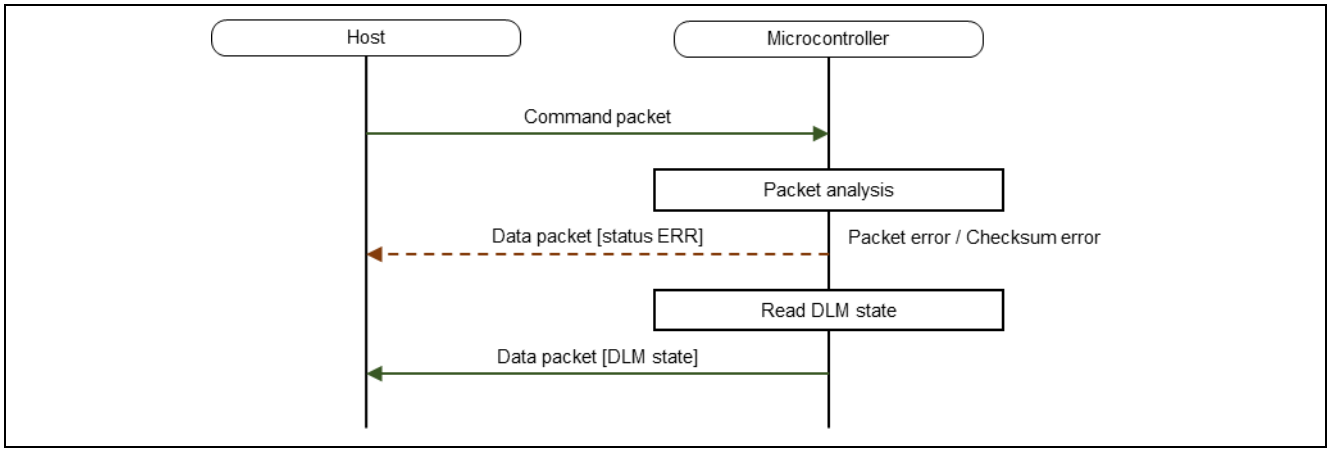


6.9 DLM state request command (GrpA, GrpB, and GrpC)

This command is used to get the current DLM state.

This command requires adherence to conditions described in Command List.

6.9.1 Sequence diagram



6.9.2 Packets

6.9.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x01
CMD	(1 byte)	0x2C (DLM state request command)
SUM	(1 byte)	0xD3
ETX	(1 byte)	0x03

6.9.2.2 Data packet [DLM state]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x2C (OK)
DLM	(1 byte)	DLM state code
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.9.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xAC (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.9.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.

* Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the inquiry processing, sends DLM state and returns to the command wait state.

* Flash memory status does not change before command reception.

6.9.4 Status information from microcontroller

(listed in descending order of priority)

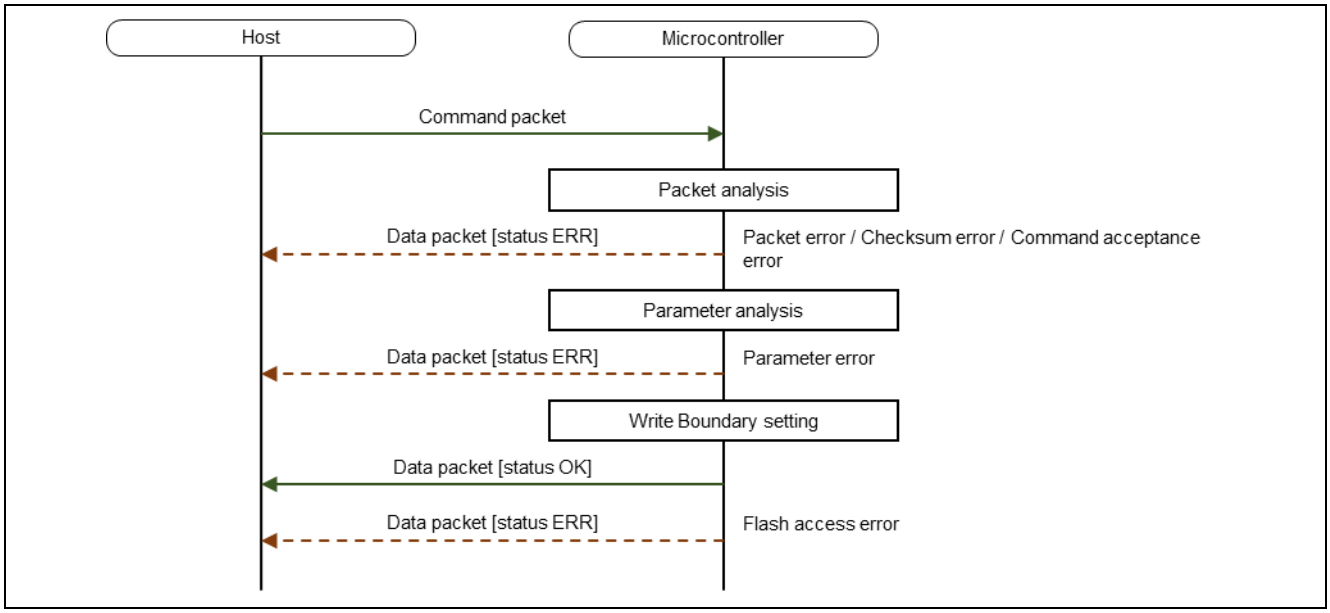
Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF

6.10 Boundary setting command (GrpA, GrpB, and GrpC)

This command receives the boundary setting and stores it in the device. The stored boundary setting becomes effective after resetting the device.

This command requires adherence to conditions described in Command List.

6.10.1 Sequence diagram



6.10.2 Packets

6.10.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x0B	
CMD	(1 byte)	0x4E (Boundary setting command)	
CFS1	(2 bytes)	Size of code flash secure region (without NSC) [KB]	e.g.) 0x0100 -> 0x01, 0x00 (256 KB)
CFS2	(2 bytes)	Size of code flash secure region [KB]	e.g.) 0x0100 -> 0x01, 0x00 (256 KB) * 32 KB align
DFS1	(2 bytes)	Size of Data Flash Secure region [KB]	e.g.)0x0004 -> 0x00, 0x04 (4 KB)
SRS1	(2 bytes)	Size of SRAM Secure region (without NSC) [KB]	e.g.) 0x0040 -> 0x00, 0x40 (64 KB)
SRS2	(2 bytes)	Size of SRAM secure region [KB]	e.g) 0x0040 -> 0x00, 0x40 (64 KB) * 8 KB align
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

NSC: Non-secure callable regions
* If CFS2 or SRS2 does not comply with the alignment, the boot firmware rounds them down.

6.10.2.2 Data Packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x4E (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.10.2.3 Data Packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xCE (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.10.2.4 Data Packet [status ERR] (Flash access error in not disclosed area)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xCE (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.10.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- If this command cannot be executed in the current DLM state, the boot firmware sends a ”Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.

* Flash memory status does not change before command reception.

Boot firmware analyzes the received parameters after packet analysis.

- When CFS1 is bigger than CFS2, “Parameter error” is returned.
- When SRS1 is bigger than SRS2, “Parameter error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.

* Flash memory status does not change before command reception.

Boot firmware executes the write processing of boundary setting after parameter analysis.

* When CFS2 or SRS2 in the received command packet does not comply with the alignment, the boot firmware rounds them down.

- If an error occurs while writing, the boot firmware sends a “Flash access error” and returns to the command wait state.
- When the write processing has finished normally, boot firmware returns “OK” and waits for the next command.

6.10.4 Status information from microcontroller

(listed in descending order of priority)

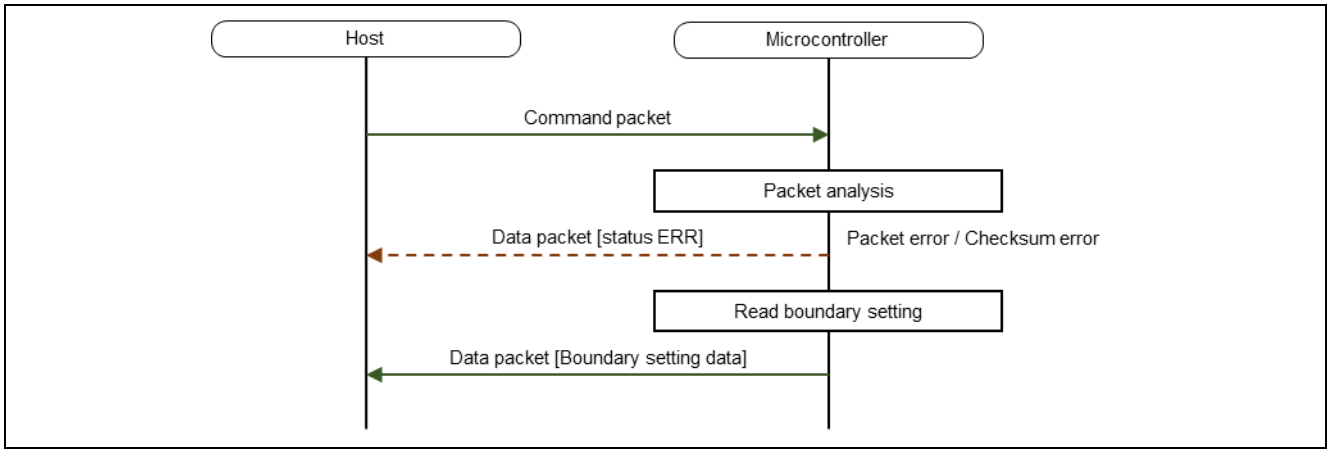
Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
CFS1 is larger than CFS2.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
SRS1 is larger than SRS2.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after command execution in a non disclosed area.	Flash access error	Flash status	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.11 Boundary request command (GrpA, GrpB, and GrpC)

This command sends the boundary setting value to the host (returns the current value stored in the device).

This command requires adherence to conditions described in Command List.

6.11.1 Sequence diagram



6.11.2 Packets

6.11.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x01
CMD	(1 byte)	0x4F (Boundary request command)
SUM	(1 byte)	0xB0
ETX	(1 byte)	0x03

6.11.2.2 Data packet [Boundary setting data]

SOD	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x0B	
RES	(1 byte)	0x4F (OK)	
CFS1	(2 bytes)	Size of code flash secure region (without NSC) [KB]	e.g.) 0x0100 -> 0x01, 0x00 (256 KB)
CFS2	(2 bytes)	Size of code flash secure region [KB]	e.g.) 0x0100 -> 0x01, 0x00 (256 KB)
DFS1	(2 bytes)	Size of data flash secure region [KB]	e.g.) 0x0004 -> 0x00, 0x04 (4 KB)
SRS1	(2 bytes)	Size of SRAM secure region (without NSC) [KB]	e.g.) 0x0040 -> 0x00, 0x40 (64 KB)
SRS2	(2 bytes)	Size of SRAM secure region [KB]	e.g.) 0x0040 -> 0x00, 0x40 (64 KB)
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

NSC: Non-secure callable regions

6.11.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xCF (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.11.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware returns boundary setting after packet analysis.

- Boot firmware sends “Boundary information” and waits for next command.
  * Flash memory status does not change before command reception.

6.11.4 Status information from microcontroller

(listed in descending order of priority)

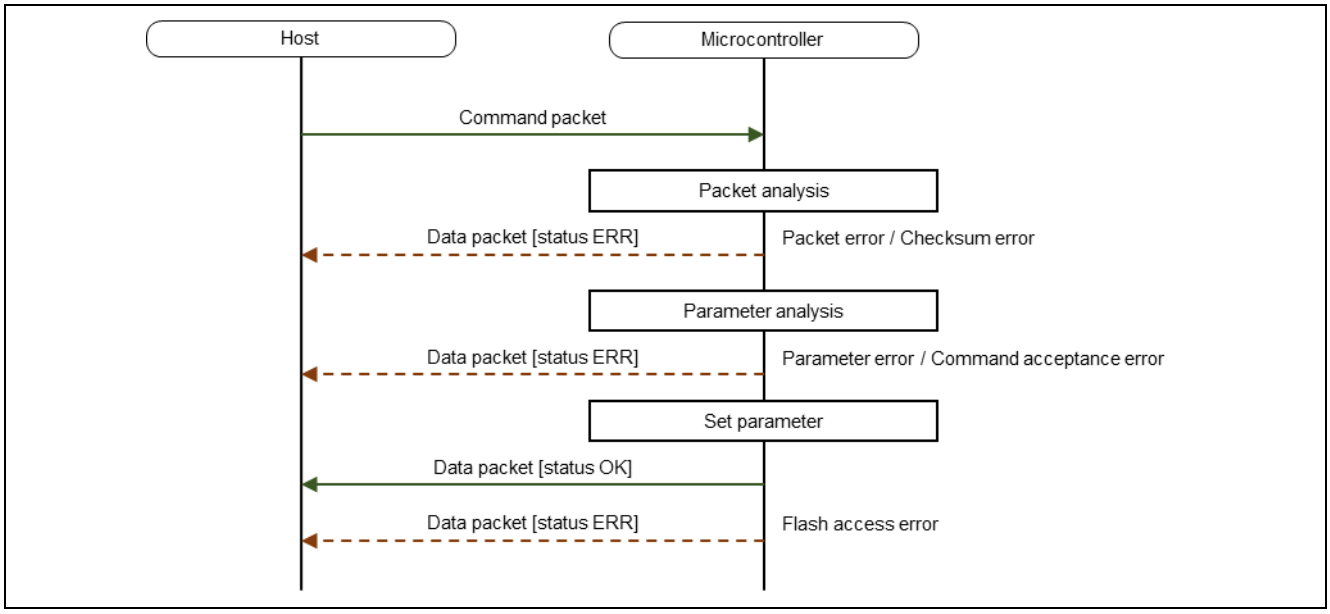
Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF

6.12 Parameter setting command (GrpA, GrpB, and GrpC)

This command stores the received parameter in the device.

This command requires adherence to conditions described in Command List.

6.12.1 Sequence diagram



6.12.2 Packets

6.12.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	N + 2	
CMD	(1 byte)	0x51 (Parameter setting command)	
PMID	(1 byte)	Parameter ID	0x01 : Setting of disable initialization
PRMT	(N byte)	Parameter data	[PMID = 0x01] 0x00 : Disable initialization
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

N = 1 to 16

6.12.2.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x51 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.12.2.3 Data packet [status ERR] (except Flash access error)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xD1 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.12.2.4 Data packet [status ERR] (Flash access error in not disclosed area)

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xD1 (ERR)
STS	(1 byte)	0xE5 (Flash access error)
ST2	(4 bytes)	Status details
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.12.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware analyzes the received parameters after packet analysis.

- When designated PMID is unsupported, “Parameter error” is returned.
- When this command is unavailable in the current DLM state, “Command acceptance error” is returned.
- If PRMT is not the specified value, the boot firmware sends a “Parameter error” and returns to the command wait state.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware executes write processing of parameter data after parameter analysis.

- If an error occurs while writing, the boot firmware sends a “Flash access error” and returns to the command wait state.
- When the write processing has finished normally, boot firmware returns “OK” and waits for the next command.

6.12.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Designated PMID is unsupported.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Executing this command is unavailable in current DLM state.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
PRMT is not the specified value.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after command execution in a non disclosed area.	Flash access error	Flash status	0xFFFFFFFF
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.12.5 Parameter Details by PMID

Parameter details for each PMID are as follows:

［PMID = 0x01］

Setting of disable initialization.

Example: Disable initialization

N = 1
PRMT[2:0] : 000b
PRMT[7:3] : Any value can be specified (ignored when writing).

* PRMT[2:0] accepts only 0x000. If 0x000 is already written, the boot firmware does not write and returns OK.

* Once disabled, initialization cannot be enabled again.

［PMID other than the above］

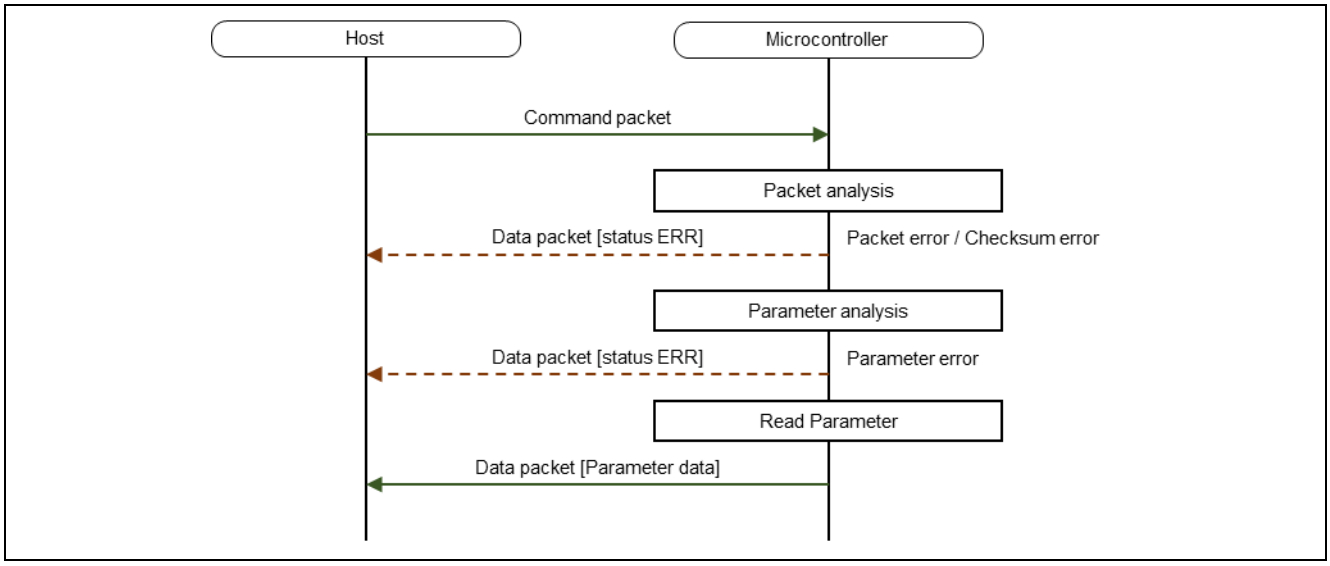
Boot firmware is unsupported and returns “Parameter error”.

6.13 Parameter request command (GrpA, GrpB, and GrpC)

This command reads the specified parameter from the device and sends it to the host (returns the value currently stored in the device).

This command requires adherence to conditions described in Command List.

6.13.1 Sequence diagram



6.13.2 Packets

6.13.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x02	
CMD	(1 byte)	0x52 (Parameter request command)	
PMID	(1 byte)	Parameter ID	0x01 : Enable / Disable of the initialization
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.13.2.2 Data packet [Parameter data]

SOD	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	N + 1	
RES	(1 byte)	0x52 (OK)	
PRMT	(N byte)	Parameter data	[PMID = 0x01] 0x00 : Initialization is disabled 0x07 : Initialization is enabled
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

N = 1 to 16

6.13.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xD2 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.13.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives data other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a “Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packets of LNH and LNL are different from the values specified in the packet format, the boot firmware sends a “Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a “Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware analyzes parameter after packet analysis.

- When designated PMID is unsupported, “Parameter error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

Boot firmware returns parameter after parameter analysis.

- Boot firmware sends parameter and waits for next command.
  * Flash memory status does not change before command reception.

6.13.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Designated PMID is unsupported.	Parameter error	0xFFFFFFFF	0xFFFFFFFF

6.13.5 Parameter Details by PMID

Parameter details for each PMID are as follows:

［PMID = 0x01］

Returns the enabled / disabled state of the Initialization.

Example) When Initialization is disabled

N = 1
PRMT[2:0] : 000b
PRMT[7:3] : Always returns 0

Example) When Initialization is enabled

N = 1
PRMT[2:0] : 111b
PRMT[7:3] : Always returns 0

［PMID other than the above］

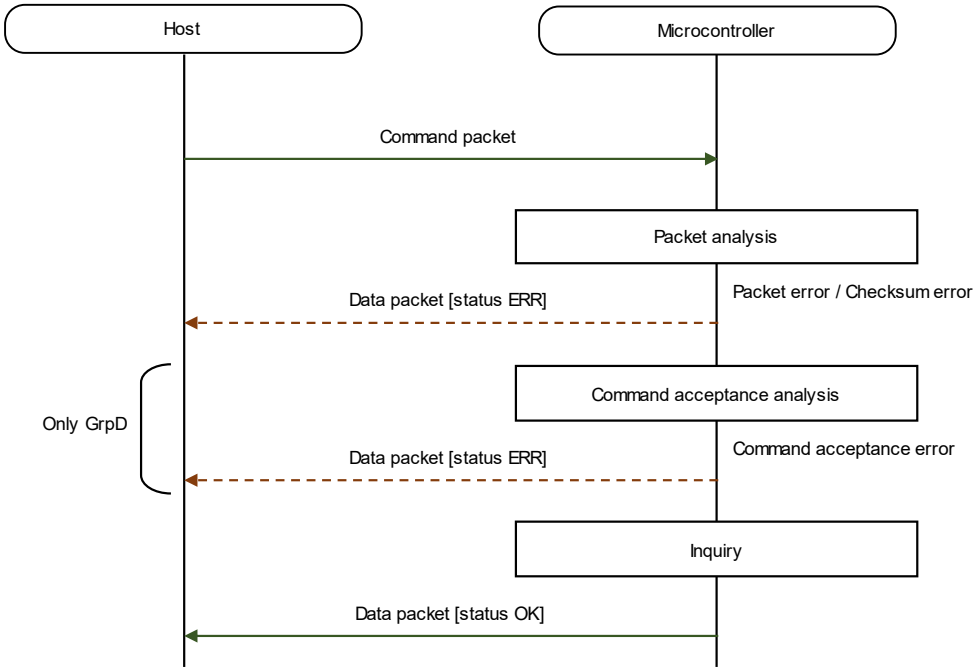
Boot firmware is unsupported and returns “Parameter error”.

6.14 Inquiry command (GrpA, GrpB, GrpC, and GrpD)

This command is used to check if boot firmware is “Command acceptable phase” or not.

This command require adherence to conditions described in Command List.

6.14.1 Sequence diagram



6.14.2 Packets

6.14.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(10xbyte)	0x00
LNL	(1 byte)	0x01
CMD	(1 byte)	0x00 (Inquiry command)
SUM	(1 byte)	0xFF
ETX	(1 byte)	0x03

6.14.2.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x00 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xFE
ETX	(1 byte)	0x03

6.14.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x80 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details)
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.14.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packets of LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

For GrpD
When the processing above is successfully completed, boot firmware executes the acceptance analysis.

- If OCD/Serial ID is set and not authenticated, the boot firmware sends a "Command acceptance error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, boot firmware executes the inquiry processing.

- The boot firmware sends ”OK” .
  * Flash memory status does not change before command reception.

6.14.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
OCD/Serial ID is set, and ID authentication by Authentication command has not been performed.(*1)	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
The process has ended normally.	OK	0xFFFFFFFF	0xFFFFFFFF

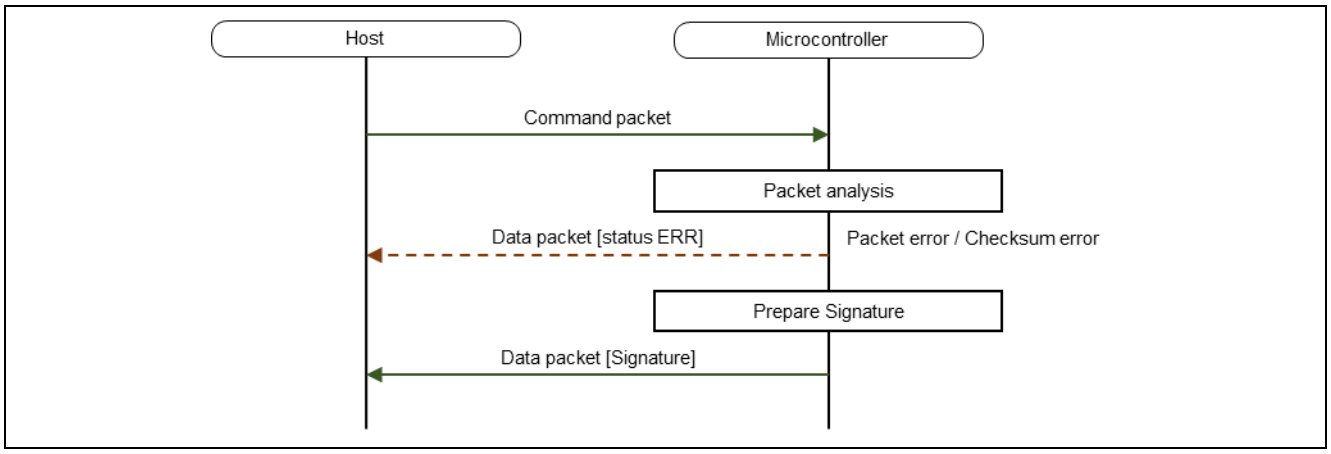
*1 : Supported on GrpD

6.15 Signature request command (GrpA, GrpB, GrpC, and GrpD)

This command sends the information of the device signature to the host.

This command requires adherence to conditions described in Command List.

6.15.1 Sequence diagram



6.15.2 Packets

6.15.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x01
CMD	(1 byte)	0x3A (Signature request command)
SUM	(1 byte)	0xC5
ETX	(1 byte)	0x03

6.15.2.2 Data packet [Signature]

SOD	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x2A	
RES	(1 byte)	0x3A (OK)	
RMB	(4 bytes)	Recommended maximum UART baudrate of the device [bps]	*Order of sending: High -> ... -> Low e.g.) 6 Mbps (6000000 bps) -> 0x00, 0x5B, 0x8D, 0x80
NOA	(1 byte)	Number of accessible areas	e.g.) If the device has 4 areas. -> 0x04
TYP	(1 byte)	Type code (features and functions of the device)	0x01:GrpA, GrpB 0x02:GrpC 0x05:GrpD
BFV	(3 bytes)	Boot firmware version	*Order of sending: Major version -> Minor version -> Build e.g.) v2.4.16 -> 0x02, 0x04, 0x10
DID	(16 bytes)	Device ID	16-byte ID code (unique ID) for identifying the individual MCU
PTN	(16 bytes)	Product type name	Character strings (0x20 for the space) *Order of sending: e.g.) R7FA6M3AH ->0x52, 0x37, 0x46, 0x41, 0x36, 0x4d, 0x33, 0x41, 0x48, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.15.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xBA (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.15.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH .
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.

* Flash memory status does not change before command reception.

When the processing above is successfully completed, boot firmware returns the signature.

- Send a signature and return to command waiting.
  * Flash memory status does not change before command reception.

6.15.4 Status information from microcontroller

(listed in descending order of priority)

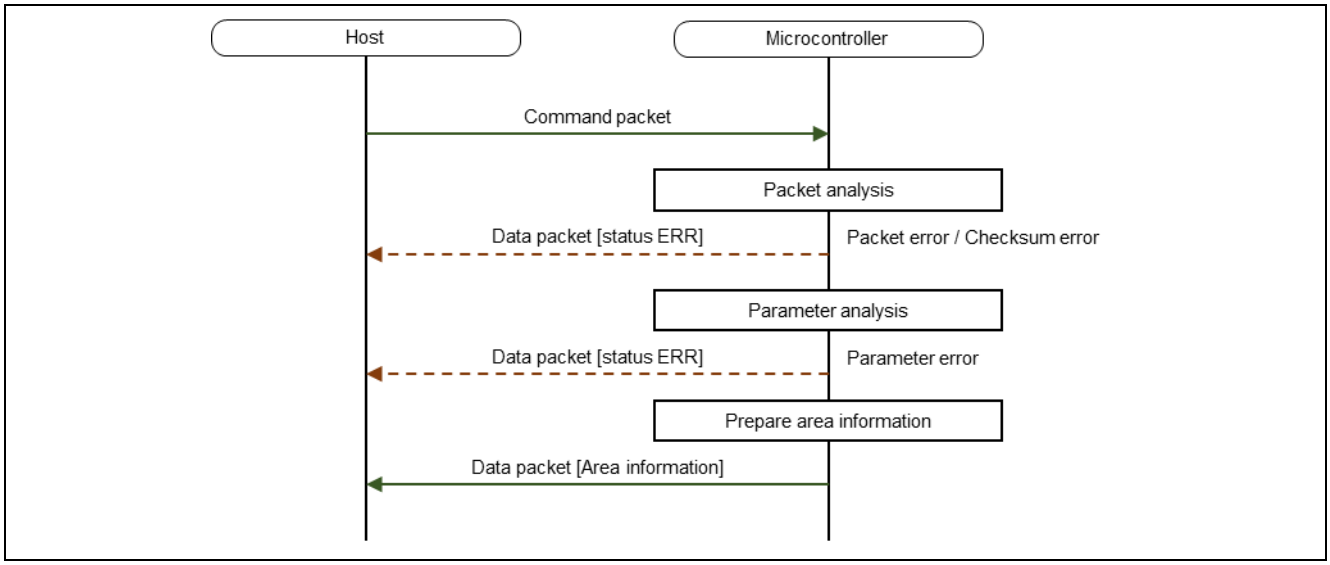
Condition	STS	ST2	ADR
The received packet doesn’t have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF

6.16 Area information request command (GrpA, GrpB, GrpC, and GrpD)

This command sends the information of the designated area to the host. The alignment of the target address of command shall follow this area information.

This command requires adherence to conditions described in Command List.

6.16.1 Sequence diagram



6.16.2 Packets

6.16.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
CMD	(1 byte)	0x3B (Area information request command)
NUM	(1 byte)	Area number [0 to NOA-1]
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.16.2.2 Data packet [Area information]

SOD	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x1A	
RES	(1 byte)	0x3B (OK)	
KOA	(1 byte)	Kind of the area	0x0N : User area N(*2) 0x1N : Data area N(*2) 0x2N : Config area N(*2)
SAD	(4 bytes)	Start address	*Order of sending: High -> ... -> Low e.g.) 0x00010000 -> 0x00, 0x01, 0x00, 0x00
EAD	(4 bytes)	End address	*Order of sending: High -> ... -> Low e.g.) 0x001FFFFF -> 0x00, 0x1F, 0xFF, 0xFF
EAU	(4 bytes)	Erase access unit (alignment) [byte] (*1)	*Order of sending: High -> ... -> Low e.g.) 32 KB (32768byte) -> 0x00, 0x00, 0x80, 0x00 Target command : Erase command
WAU	(4 bytes)	Write access unit (alignment) [byte] (*1)	*Order of sending: High -> ... -> Low e.g.) 128byte -> 0x00, 0x00, 0x00, 0x80 Target command : Write command
RAU	(4 bytes)	Read access unit (alignment) [byte] (*1)	*Order of sending: High -> ... -> Low e.g.) 1byte -> 0x00, 0x00, 0x00, 0x01 Target command : Read command
CAU	(4 bytes)	CRC access unit (alignment) [byte] (*1)	*Order of sending: High -> ... -> Low e.g.) 4byte -> 0x00, 0x00, 0x00, 0x04 Target command : CRC command
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

*1 If each access unit is 0x00000000, target command is not available for the area.
*2 N = 0x0~0xF

6.16.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xBB (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details)
ADR	(4 bytes)	Failure address)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.16.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait untilit receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”done.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, boot firmware analyzes the command parameters.

- If the specified NUM is ”NOA” returned by ”signature requests command” or more, the boot firmware send a ”Parameter error” and return to command waiting status.

* Flash memory status does not change before command reception.

- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, the area information will be returned.

- Send area information of specified NUM and return to command waiting status.
  * Flash memory status does not change before command reception.

6.16.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet doesn’t have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
If Area number in the received packet is non-existent area number.	Parameter error	0xFFFFFFFF	0xFFFFFFFF

6.16.5 Example of area information

e.g.) RA4M2 (Linear mode)

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	0x00000000	0x0000FFFF	8 KB	128 Bytes	1 Byte	32 KB
1	User area0(L)	0x00	0x00010000	0x000FFFFF	32 KB	128 Bytes	1 Byte	32 KB
2	Data area	0x10	0x08000000	0x08001FFF	64 Bytes	4 Bytes	1 Byte	1 KB
3	Config area	0x20	0x0100A100	0x0100A2FF	0 (*1)	16 Bytes	1 Byte	256Bytes(*2)

e.g.) RA6M4 (Linear mode)

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	0x00000000	0x0000FFFF	8 KB	128 Bytes	1 Byte	32 KB
1	User area0(L)	0x00	0x00010000	0x000FFFFF	32 KB	128 Bytes	1 Byte	32 KB
2	Data area	0x10	0x08000000	0x08001FFF	64 Bytes	4 Bytes	1 Byte	1 KB
3	Config area	0x20	0x0100A100	0x0100A2FF	0 (*1)	16 Bytes	1 Byte	256Bytes(*2)

e.g.) RA6M4 (Dual mode)

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	0x00000000	0x0000FFFF	8 KB	128 Bytes	1 Byte	32 KB
1	User area0(L)	0x00	0x00010000	0x0007FFFF	32 KB	128 Bytes	1 Byte	32 KB
2	User area1(S)	0x01	0x00200000	0x0020FFFF	8 KB	128 Bytes	1 Byte	32 KB
3	User area1(L)	0x01	0x00210000	0x0027FFFF	32 KB	128 Bytes	1 Byte	32 KB
4	Data area	0x10	0x08000000	0x08001FFF	64 Bytes	4 Bytes	1 Byte	1 KB
5	Config area	0x20	0x0100A100	0x0100A2FF	0 (*1)	16 Bytes	1 Byte	256Bytes(*2)

e.g.) RA4E1 (Linear mode)

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	0x00000000	0x0000FFFF	8 KB	128 Bytes	1 Byte	32 KB
1	User area0(L)	0x00	0x00010000	0x000FFFFF	32 KB	128 Bytes	1 Byte	32 KB
2	Data area	0x10	0x08000000	0x08001FFF	64 Bytes	4 Bytes	1 Byte	1 KB
3	Config area	0x20	0x0100A100	0x0100A2FF	0 (*1)	16 Bytes	1 Byte	256Bytes(*2)

e.g.) RA6E1 (Linear mode)

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	0x00000000	0x0000FFFF	8 KB	128 Bytes	1 Byte	32 KB
1	User area0(L)	0x00	0x00010000	0x000FFFFF	32 KB	128 Bytes	1 Byte	32 KB
2	Data area	0x10	0x08000000	0x08001FFF	64 Bytes	4 Bytes	1 Byte	1 KB
3	Config area	0x20	0x0100A100	0x0100A2FF	0 (*1)	16 Bytes	1 Byte	256Bytes(*2)

e.g.) RA6T2

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	00000000h	0000FFFFh	8KB	128Bytes	1Byte	32KB
1	User area0(L)	0x00	00010000h	0007FFFFh	32KB	128Bytes	1Byte	32KB
2	Data area	0x10	08000000h	08003FFFh	64B	4Bytes	1Byte	1KB
3	Config area	0x20	0100A100h	0100A2FFh	0 (*1)	16Bytes	1Byte	256Bytes(*2)

e.g.) RA6E2

NUM	Area	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area0(S)	0x00	00000000h	0000FFFFh	8KB	128Bytes	1Byte	32KB
1	User area0(L)	0x00	00010000h	0003FFFFh	32KB	128Bytes	1Byte	32KB
2	Data area	0x10	08000000h	08000FFFh	64B	4Bytes	1Byte	1KB
3	Config area	0x20	0100A100h	0100A2FFh	0 (*1)	16Bytes	1Byte	256Bytes(*2)

*1: When Access unit is 0, it indicates that the corresponding operation is not supported.

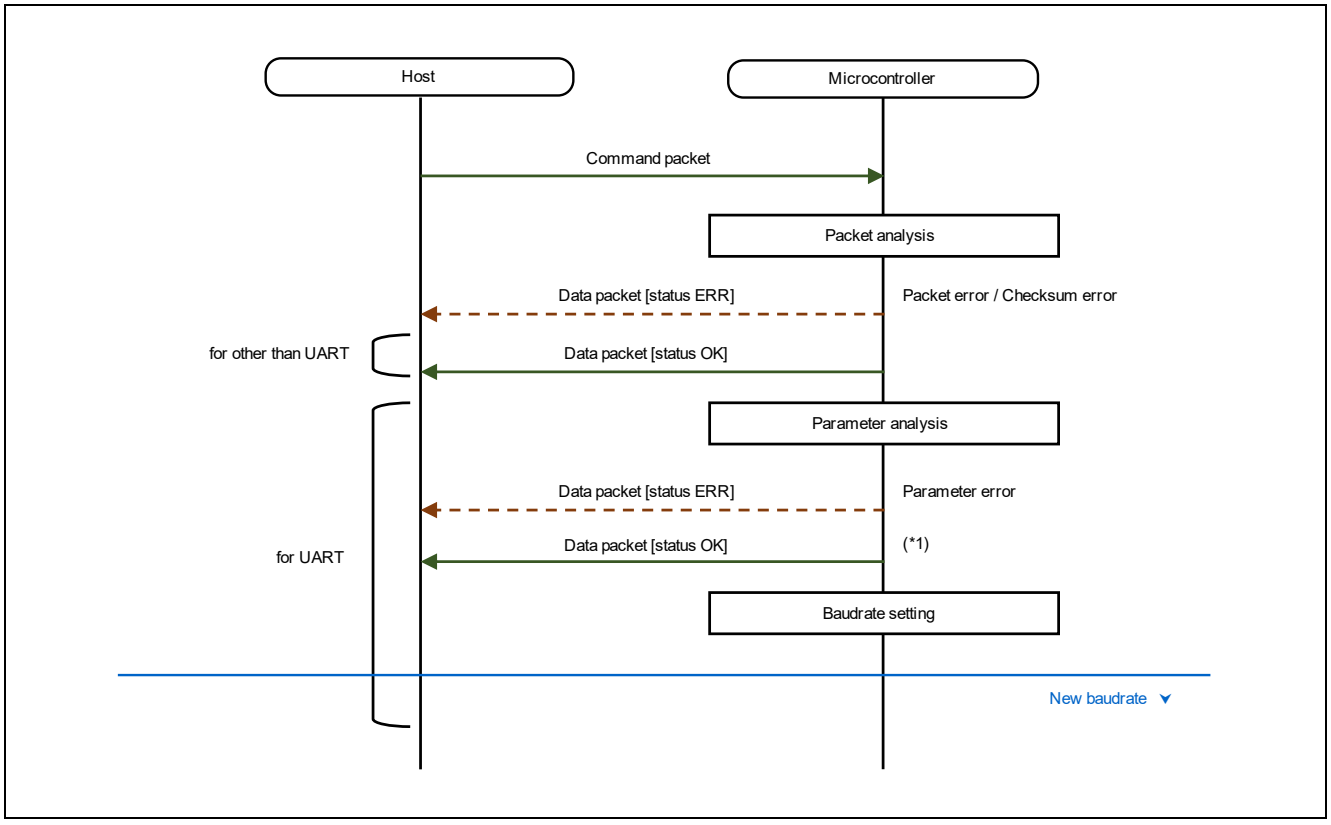
*2: CRC command to Config area is executable only for entire area, in other words only possible to specify the area's SAD/EAD as CRC command's SAD/EAD. See CRC command for details

6.17 Baud rate setting command (GrpA, GrpB, GrpC, and GrpD)

This command receives baudrate data and changes the UART baudrate of the device. If an error occurs, the baudrate is not changed. This command does not change the communication speed except for UART communication.

This command requires adherence to conditions described in Command List.

6.17.1 Sequence diagram



*1: After this packet receiving, next command packet waits tBRT for keeping the baudrate setting time.

6.17.2 Packets

6.17.2.1 Command packet

]	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x05	
CMD	(1 byte)	0x34 (Baudrate setting command)	
BRT	(4 bytes)	UART baudrate [bps]	You can set one of the following values. Order of sending BRT:

Baud rate	GrpA GrpB GrpC	GrpD	1st	2nd	3rd	4th
9600bps	✔	✔	0x00	0x00	0x25	0x80
115200bps	✔	✔	0x00	0x01	0xC2	0x00
500Kbps	✔	✔	0x00	0x07	0xA1	0x20
1.0Mbps	✔	✔	0x00	0x0F	0x42	0x40
1.5Mbps	✔	✔	0x00	0x16	0xE3	0x60
2.0Mbps	✔	✔	0x00	0x1E	0x84	0x80
4.0Mbps	✔		0x00	0x3D	0x09	0x00
6.0Mbps	✔		0x00	0x5B	0x8D	0x80

SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.17.2.2 Data packet [status OK]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x34 (OK)
STS	(1 byte)	0x00 (OK)
FST	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	0xCA
ETX	(1 byte)	0x03

6.17.2.3 Data packet [status ERR]

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0xB4 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details)
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.17.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”don.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the communication mode is not asynchronous 2-wire communication, a response will be returned when packet analysis ends normally.

- If the communication mode is not asynchronous 2-wire communication, the boot firmware sends ”OK” and returns to the command waiting state.
  * Flash memory status does not change before command reception.

In asynchronous 2-wire communication, parameter analysis is performed when processing above is completed successfully.

- Sends ”Parameter error” if the specified BRT (baudrate) is greater than the RMB in the Signature request command.
- Sends ”Parameter error” if the specified BRT (baudrate) is not a supported baudrate value.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

In asynchronous 2-wire communication, when processing above is completed normally, the baud rate is set.

- After sending ”OK”, set the baudrate and return to the command waiting state.
  * Flash memory status does not change before command reception.
  * After the boot firmware returned OK (Started the baudrate setting), wait 1ms before sendingf next command.

6.17.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet doesn't have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Received UART baudrate is greater than RMB.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Different from the baudrate value supported by the received UART baudrate.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Communication mode is different from UART.	OK	0xFFFFFFFF	0xFFFFFFFF
Started the baudrate setting.	OK	0xFFFFFFFF	0xFFFFFFFF

6.17.5 Baud rate setting values

(GrpA, GrpB, and GrpC)

Intended baud rate	ABCS	CKS[1:0]	BRR[7:0]	MDDR[7:0]	Accuracy
9600 bps	0b0	0b00	0xFF	0xC9	-0.2%
115200 bps	0b0	0b00	0x1A	0xFE	-0.3%
500 Kbps	0b0	0b00	0x05	0xF5	-0.3%
1.0 Mbps	0b0	0b00	0x02	0xF5	-0.3%
1.5 Mbps	0b0	0b00	0x01	0xF5	-0.3%
2.0 Mbps	0b0	0b00	0x00	0xA3	-0.5%
4.0 Mbps	0b1	0b00	0x00	0xA3	-0.5%
6.0 Mbps	0b1	0b00	0x00	0xF5	-0.3%
Other	unavailable	unavailable	unavailable	unavailable	-

(GrpD)

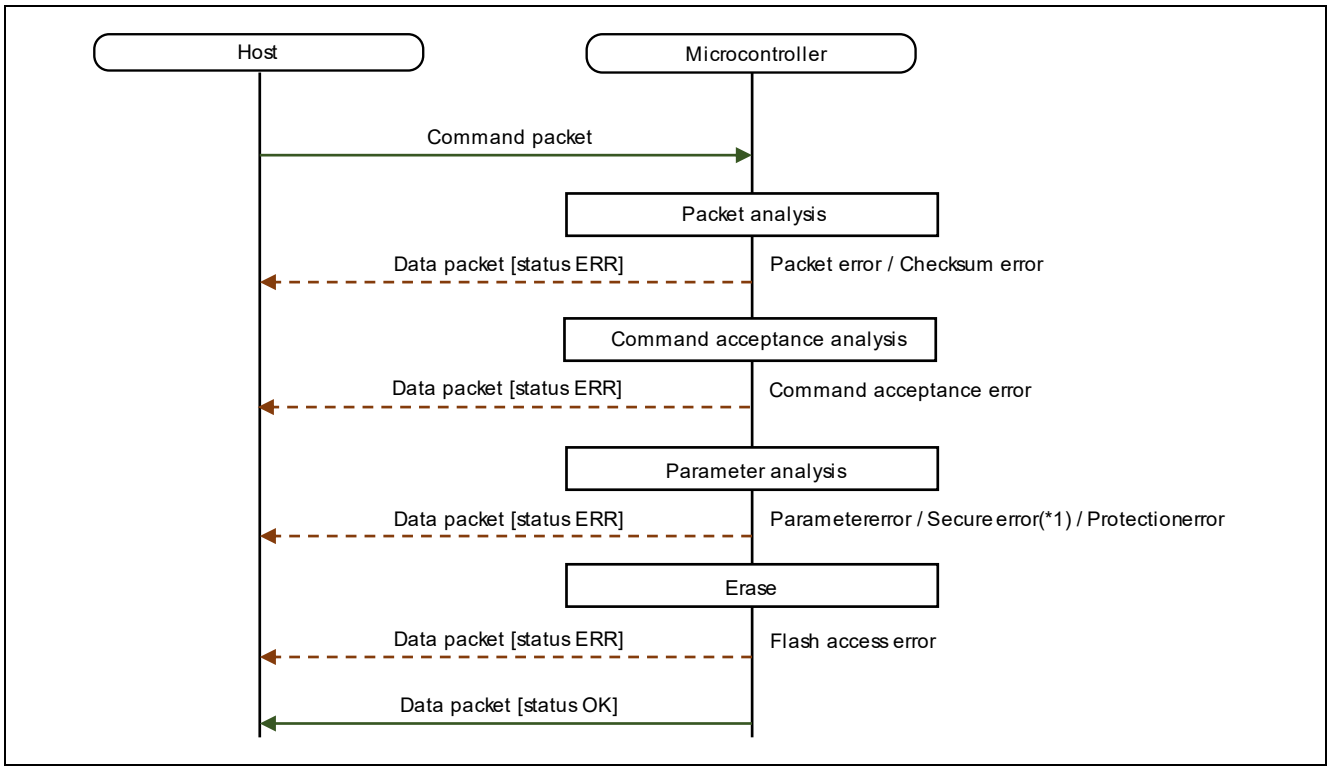
Intended baud rate	ABCS	CKS[1:0]	BRR[7:0]	MDDR[7:0]	Accuracy
9600bps	0b0	0b00	0xC2	0xFF	-0.2%
115200bps	0b0	0b00	0x0F	0xFB	-0.3%
500Kbps	0b0	0b00	0x02	0xCC	-0.3%
1.0Mbps	0b0	0b00	0x00	0x88	-0.3%
1.5Mbps	0b0	0b00	0x00	0xCC	-0.3%
2.0Mbps	0b1	0b00	0x00	0x88	-0.5%
Other	unavailable	unavailable	unavailable	unavailable	-

6.18 Erase command (GrpA, GrpB, GrpC, and GrpD)

This command erases data in the specified area of the flash memory. The alignment of the target addresses shall follow the area information returned by the Area information request command. Erasures are executed in order from the start address to the end address by the erase access unit. Erase processing at this time is not affected by the block protection settings (BPS, BPS_SEC) in case of GrpA,GrpB,GrpC, / (BPS_SEC) in case of GrpD.

This command requires adherence to conditions described in Command List.

6.18.1 Sequence diagram



*1 : In case of GrpA, GrpB, and GrpC

6.18.2 Packets

6.18.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x09	
CMD	(1 byte)	0x12 (Erase command)	
SAD	(4 bytes)	Start address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
EAD	(4 bytes)	End address	e.g.) 0x003FFFFF -> 0x00, 0x3F, 0xFF, 0xFF
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.18.2.2 Data packet [status OK]

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x12 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.18.2.3 Data packet [status ERR]

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x92 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.18.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis has successfully completed, boot firmware executes the acceptance analysis.

- (In case of GrpA, GrpB, and GrpC) : If this command cannot be executed in the current DLM state,
  (In case of GrpD) : If OCD/Serial ID is set and not authenticated,
  the boot firmware sends a ”Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, the boot firmware analyzes the command parameters.

- If the SAD is greater than EAD, the boot firmware sends a ”Parameter error”.
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends a ”Parameter error”.
- If SAD and EAD belong to different KOA, boot firmware will send a ”Parameter error”.
- If the EAU for the specified area is 0, the boot firmware sends a ”Parameter error”.
- If SAD and EAD are not specified in the EAU of the area, the boot firmware sends a ”Parameter error”.
- (In case of GrpA, GrpB, and GrpC) : If the current DLM state is NSECSD and the specified range includes a secure area, the boot firmware sends a ”Secure error”.
- When designated erasure range includes a permanent protected block, ”Protection error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When no error occurs, boot firmware executes the erase processing.

- If an error occurs during erasure, the boot firmware sends a ”Flash access error” and returns to the command wait state.
  * The value of the area after ADR (Failure address) of the Flash memory is undefined.
- When the erase processing is normally finished, boot firmware returns ”OK” and waits for the next command.
  * Specified areas in flash memory are erased state.

6.18.4 Status information from microcontroller

(listed in descending order of priority)

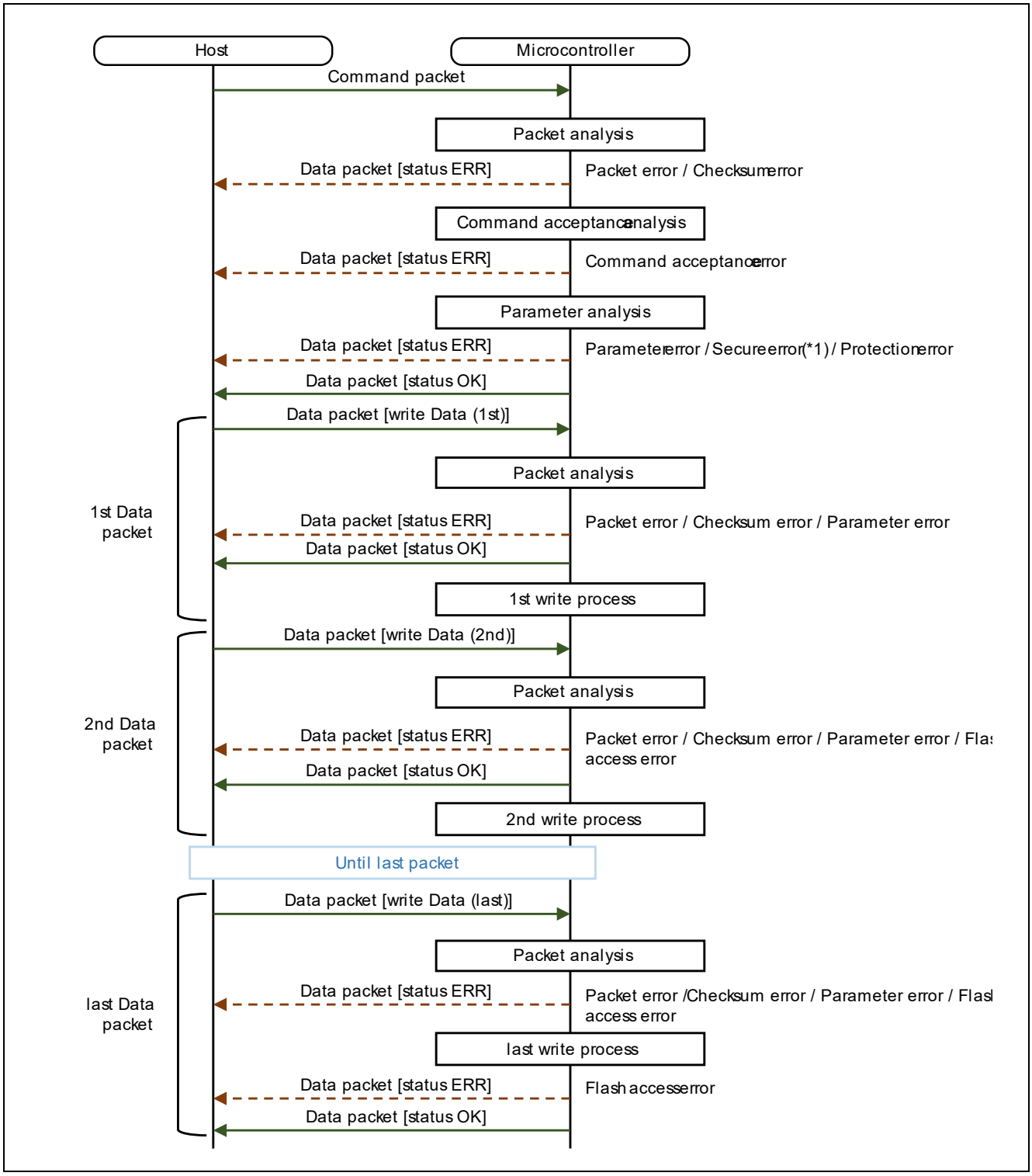
Condition	STS	ST2	ADR
The received packet doesn't have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
(In case of GrpA, GrpB, and GrpC) Executing this command is unavailable in current DLM state. (In case of GrpD) OCD/Serial ID is set, and ID authentication by Authentication command has not been performed.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
Start address is bigger than End address.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address is outside the scope of User area specified in area information.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address and End address belong to different Kind of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The access unit "EAU" of the specified area is 0.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address does not comply with EAU of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
(In case of GrpA, GrpB, and GrpC) Current DLM state is NSECSD and designated erasure range includes Secure region. (In case of GrpD) N/A	Secure error	0xFFFFFFFF	0xFFFFFFFF
Designated erasing range includes permanent protected blocks.	Protection error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after the command execution.	Flash access error	Flash status	Failure address
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.19 Write command (GrpA, GrpB, GrpC, and GrpD)

This command receives data from host and writes those data to the flash memory. The alignment of the target address shall follow the area information returned by the Area information request command. Writings are executed in order from the start address to the end address by the write access unit. Write processing at this time is not affected by the block protection settings (BPS, BPS_SEC) in case of GrpA,GrpB,GrpC, / (BPS_SEC) in case of GrpD.

This command requires adherence to conditions described in Command List.

6.19.1 Sequence diagram



*1 : In case of GrpA, GrpB, and GrpC

6.19.2 Packets

6.19.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x09	
CMD	(1 byte)	0x13 (Write command)	
SAD	(4 bytes)	Start address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
EAD	(4 bytes)	End address	e.g.) 0x003FFFFF -> 0x00, 0x3F, 0xFF, 0xFF
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.19.2.2 Data packet [write data]

SOH	(1 byte)	0x81
LNH	(1 byte)	N + 1 (Higher 1 byte)
LNL	(1 byte)	N + 1 (Lower 1 byte)
RES	(1 byte)	0x13 (OK)
DAT	(N byte)	Write data
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

N = 1 to 1024

6.19.2.3 Data packet [status OK]

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x13 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.19.2.4 Data packet [status ERR]

SOH	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x0A	
RES	(1 byte)	0x93 (ERR)	
STS	(1 byte)	Status code	
ST2	(4 bytes)	Status details	e.g.) FSTATR[31:0] = 0x0120A000 -> 0x01, 0x20, 0xA0, 0x00
ADR	(4 bytes)	Failure address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.19.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a “Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, boot firmware executes the acceptance analysis.

- (In case of GrpA, GrpB, and GrpC) : If this command cannot be executed in the current DLM state,
  (In case of GrpD) : If OCD/Serial ID is set and not authenticated,
  the boot firmware sends a ”Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, the boot firmware analyzes the command parameters.

- If the SAD is greater than EAD, the boot firmware sends a ”Parameter error”.
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends a ”Parameter error”.
- If SAD and EAD belong to different KOA, boot firmware sends a ”Parameter error”.
- If the WAU for the specified area is 0, the boot firmware sends a ”Parameter error”.
- If SAD and EAD are not specified in the WAU of the area, the boot firmware sends a ”Parameter error”.
- (In case of GrpA, GrpB, and GrpC) : If the current DLM state is NSECSD and the specified range includes a secure area, the boot firmware sends a ”Secure error”.
- When designated writing range includes PBPS block, ”Protection error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.
- If the above error does not occur, the boot firmware sends ”OK”.

When the processing above is successfully completed, boot firmware receives and analyzes a data packet.

- The boot firmware recognizes the start of the data packet by receiving SOD.
  If the boot firmware receives something other than SOD, it will wait until it receives SOD.
- If ETX is not added to the received data packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”.
- If the received data packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- When RES in the received data packet is different from defined values by each command, ”Packet error” is returned.
- When total length of the received data of data packets exceeds the size of specified area, ”Parameter error” is returned.
- If size of the write data is not specified in the WAU of the area, the boot firmware sends a ”Parameter error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the received data packet is not the last write data, boot firmware returns ”OK” and executes the write processing.

- Boot firmware returns ”OK” and executes the write processing.
- When the write processing is abnormally finished, boot firmware returns ‘Flash access error’ and waits for the next command.
  * WAU size from failure address (ADR) of Flash memory area are undefined.
- When the write processing is normally finished, boot firmware receives the next data packet.

When the received data packet is the last write data, boot firmware executes the write processing and returns status.

- Boot firmware executes the write processing.
- If an error occurs while writing, the boot firmware sends a ”Flash access error” and returns to the command wait state.
  * WAU size from failure address (ADR) of Flash memory area are undefined.
- When the write processing is normally finished, boot firmware returns ”OK” and waits for the next command.
  * Sent data are written to the specified area on flash memory.

6.19.4 Status information from microcontroller

(listed in descending order of priority)

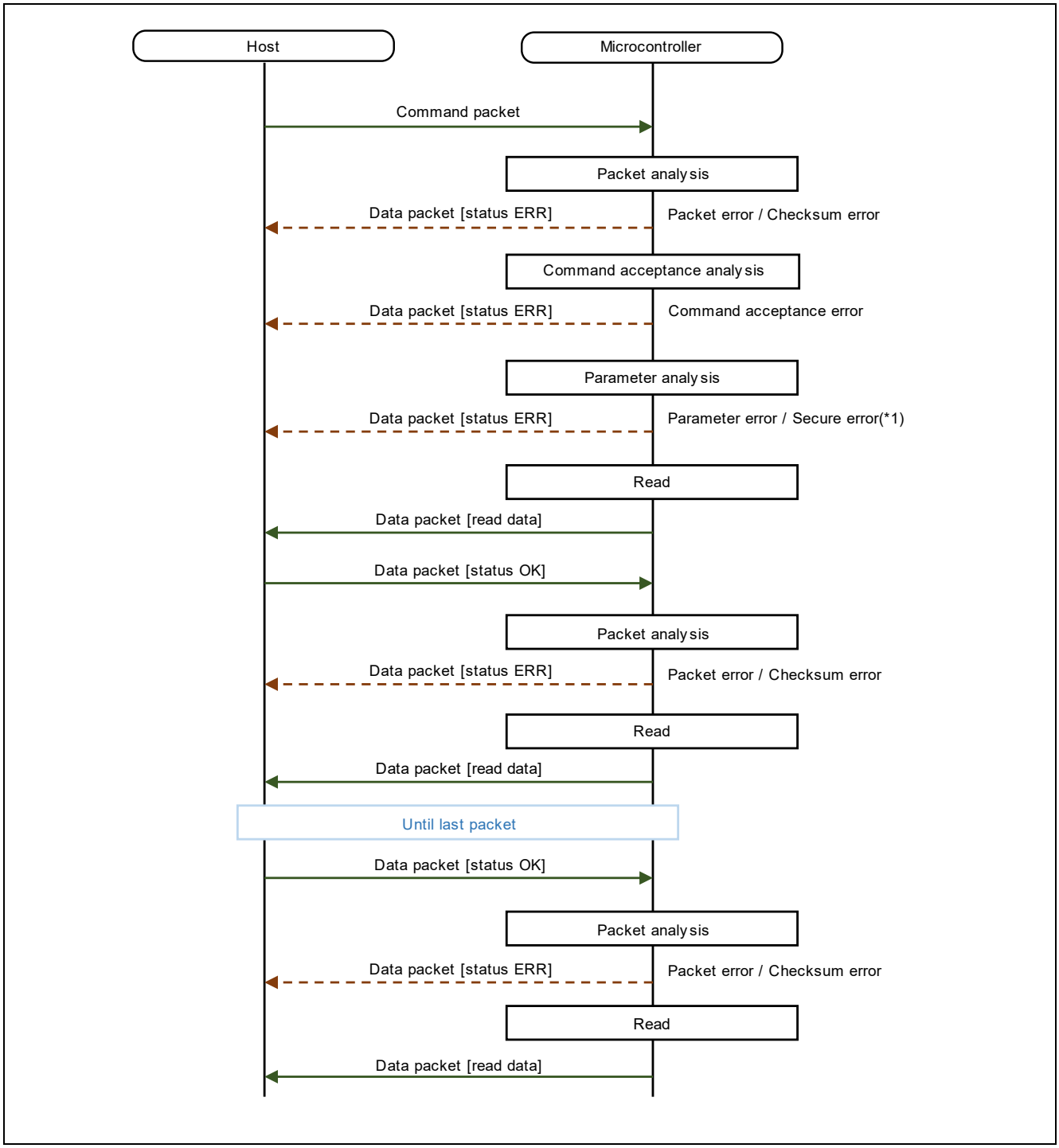
Condition	STS	ST2	ADR
The received packet doesn't have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
(In case of GrpA, GrpB, and GrpC) Executing this command is unavailable in current DLM state. (In case of GrpD) OCD/Serial ID is set, and ID authentication by Authentication command has not been performed.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
Start address is bigger than End address.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address is outside the scope of accessible area specified in area information.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address and End address belong to different Kind of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The access unit "WAU" of the specified area is 0.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address does not comply with WAU of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
(In case of GrpA, GrpB, and GrpC) Current DLM state is NSECSD and designated writing range includes secure region. (In case of GrpD) N/A	Secure error	0xFFFFFFFF	0xFFFFFFFF
Designated writing range includes permanent protected blocks.	Protection error	0xFFFFFFFF	0xFFFFFFFF
The response code of the received data packet is different from the value specified by this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
The total length of received data of data packets exceeds the specified end address.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The data size of the data packet doesn't comply with writing unit of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
FACI detected an error after the command execution.	Flash access error	Flash status	Failure address
Successful completion.	OK	0xFFFFFFFF	0xFFFFFFFF

6.20 Read command (GrpA, GrpB, GrpC, and GrpD)

This command reads data from a specified area in the flash memory, and sends those data to host. The alignment of the target addresses shall follow the area information returned by the Area information request command. Readings are executed in order from the start address to the end address by the read access unit.

This command requires adherence to conditions described in Command List.

6.20.1 Sequence diagram



*1 : In case of GrpA, GrpB, and GrpC

6.20.2 Packets

6.20.2.1 Command packet

SOH	(1 byte)	0x01	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x09	
CMD	(1 byte)	0x15 (Read command)	
SAD	(4 bytes)	Start address	e.g.) 0x00004000 -> 0x00, 0x00, 0x40, 0x00
EAD	(4 bytes)	End address	e.g.) 0x003FFFFF -> 0x00, 0x3F, 0xFF, 0xFF
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.20.2.2 Data packet [status OK]

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x15 (OK)
STS	(1 byte)	0x00 (OK)
ST2	(4 bytes)	0xFFFFFFFF (unused code)
ADR	(4 bytes)	0xFFFFFFFF (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.20.2.3 Data packet [status ERR]

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x95 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.20.2.4 Data packet [read data]

SOH	(1 byte)	0x81
LNH	(1 byte)	N + 1 (Higher 1 byte)
LNL	(1 byte)	N + 1 (Lower 1 byte)
RES	(1 byte)	0x15 (OK)
DAT	(N byte)	Read data
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

N = 1 to 1024

6.20.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it is receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, boot firmware executes the acceptance analysis.

- (In case of GrpA, GrpB, and GrpC) : If this command cannot be executed in the current DLM state,
  (In case of GrpD) : If OCD/Serial ID is set and not authenticated,
  the boot firmware sends a ”Command acceptance error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, the boot firmware analyzes the command parameters.

- If the SAD is greater than EAD, the boot firmware sends a ”Parameter error”.
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends a ”Parameter error”.
- If SAD and EAD belong to different KOA, boot firmware sends a ”Parameter error”.
- If the RAU for the specified area is 0, the boot firmware sends a ”Parameter error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, the boot firmware performs a secure analysis.

- (In case of GrpA, GrpB, and GrpC) : If the current DLM state is NSECSD and the specified range includes a secure area, the boot firmware sends a ”Secure error” and returns to the command waiting state without processing.
  * Flash memory status does not change before command reception.

When no error occurs, boot firmware executes the read processing.

- Boot firmware returns the data stored in the internal buffer (packet-length: Max. 1024 bytes.
- When all the data have been sent, boot firmware waits for the next command.
  * Flash memory status does not change before command reception.

If data transmission for the specified size is not completed, the boot firmware receives the data packet and performs packet analysis.

- Boot firmware detects the beginning of a data packet by receiving SOD.
  When boot firmware receives other data than SOD, it discards the data and waits for the next data until SOD is sent.
- When the received data packet doesn't have ETX, "Packet error" is returned .
- When SUM in the received data packet is different from the value calculated by boot firmware, "Checksum error" is returned.
- When LNH and LNL in the received data packet do not comply with the packet format, "Packet error" is returned.
- When RES in the received data packet is different from defined values, "Packet error" is returned.
- When LNH and LNL in the received data packet do not comply format with this command, "Packet error"' is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.
- If the above error does not occur, the boot firmware continues to read and send data.

6.20.4 Status information from microcontroller

(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
(In case of GrpA, GrpB, and GrpC) Executing this command is unavailable in current DLM state. (In case of GrpD) OCD/Serial ID is set, and ID authentication by Authentication command has not been performed.	Command acceptance error	0xFFFFFFFF	0xFFFFFFFF
Start address is bigger than End address.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address is outside the scope of accessible area specified in area information.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address and End address belong to different Kind of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The access unit "RAU" of the specified area is 0.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address doesn't comply with RAU of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
(In case of GrpA, GrpB, and GrpC) Current DLM state is NSECSD, designated reading range is User area or Data area and includes secure region. (In case of GrpD) N/A	Secure error	0xFFFFFFFF	0xFFFFFFFF
The response code of the received data packet is different from the value specified by this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF

6.21 CRC command (GrpA, GrpB, GrpC, and GrpD)

This command calculates CRC data from a specified area in the flash memory, and sends it to host. The alignment of the target addresses shall follow the area information returned by the Area information request command. Calculations are executed in order from the start address to the end address by the CRC access unit.

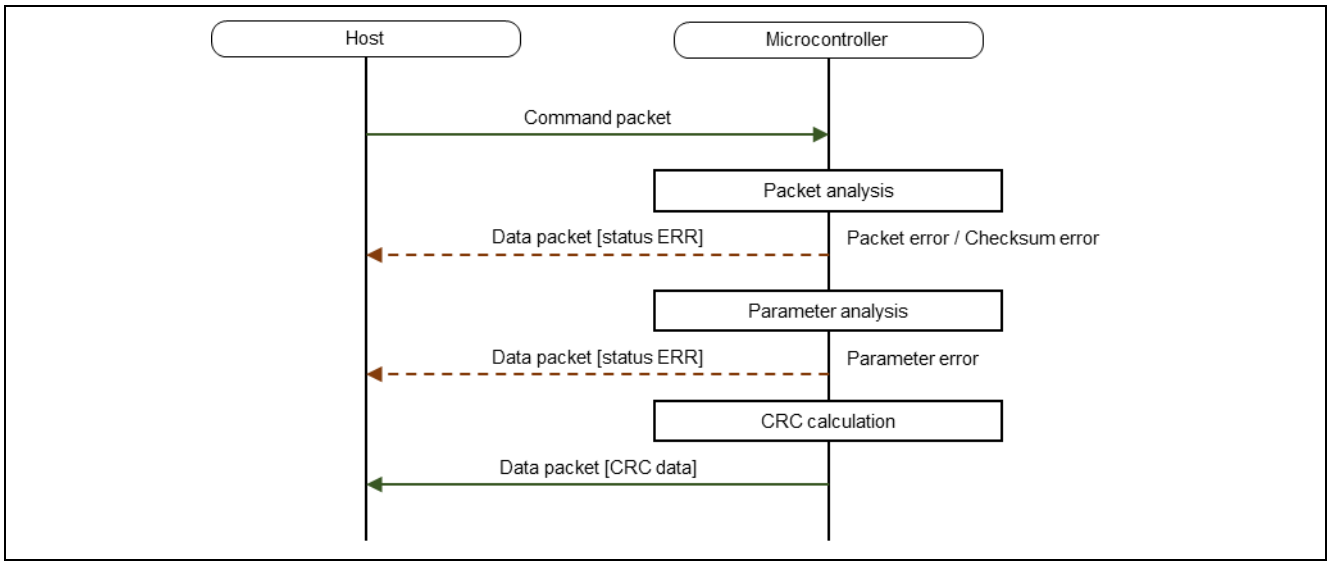
As an exception, when specifying Config area, it is able to specify only entire Config area regardless of CRC access unit returned by Area information command.

This command requires adherence to conditions described in Command List.

Boot firmware uses following CRC method.

Name	CRC-32-IEEE-802.3
Default value	0xFFFFFFFF
Shift direction	Left shift
Polynomial representations	(MSB first) 0x04C11DB7

6.21.1 Sequence diagram



6.21.2 Packets

6.21.2.1 Command packet

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x09
CMD	(1 byte)	0x18 (CRC command)
SAD	(4 bytes)	Start address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.21.2.2 Data packet [CRC data]

SOH	(1 byte)	0x81	
LNH	(1 byte)	0x00	
LNL	(1 byte)	0x05	
RES	(1 byte)	0x18 (OK)	
CRC	(4 bytes)	CRC data (result of calculation)	e.g.) 0x01234567 -> 0x01, 0x23, 0x45, 0x67
SUM	(1 byte)	Sum data	
ETX	(1 byte)	0x03	

6.21.2.3 Data packet [status ERR]

SOH	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0A
RES	(1 byte)	0x98 (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

6.21.3 Processing procedure

Boot firmware receives and analyzes a command packet.

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it will wait until it receives SOH.
- If ETX is not added to the received command packet, the boot firmware sends a ”Packet error”.
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a ”Checksum error”.
- If the received command packet’s LNH and LNL are different from the values specified in the packet format, the boot firmware sends a ”Packet error”.
- If the received command packet’s LNH and LNL are different from the values specified in each command, the boot firmware sends a ”Packet error”.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the processing above is successfully completed, the boot firmware analyzes the command parameters.

- If the SAD is greater than EAD, the boot firmware sends a ”Parameter error”.
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends a ”Parameter error”.
- If SAD and EAD belong to different KOA, boot firmware will send a ”Parameter error”.
- If the CAU for the specified area is 0, the boot firmware sends a ”Parameter error”.
- If SAD and EAD are not specified in the CAU of the area, the boot firmware sends a ”Parameter error”.
- When KOA is 0x20 (Config area) and SAD／EAD do not specify entire Config area, ”Parameter error” is returned.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the parameter analysis has successfully completed, boot firmware executes CRC calculation.

- After the CRC calculation, boot firmware returns ”CRC data” and waits for the next command.
  * Flash memory status does not change before command reception.

6.21.4 Status information from microcontroller

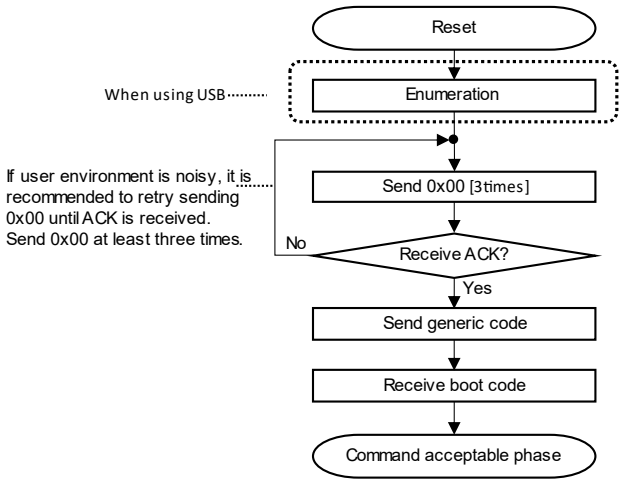
(listed in descending order of priority)

Condition	STS	ST2	ADR
The received packet doesn’t have ETX.	Packet error	0xFFFFFFFF	0xFFFFFFFF
SUM data in the received packet is different from the value calculated by the boot firmware.	Checksum error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the packet format.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Packet length in the received packet do not comply with the specifications of this command.	Packet error	0xFFFFFFFF	0xFFFFFFFF
Start address is bigger than End address.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address is outside the scope of accessible area specified in area information.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address and End address belong to different Kind of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The access unit ”CAU” of the specified area is 0.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
Start address or End address doesn’t comply with CAU of the area.	Parameter error	0xFFFFFFFF	0xFFFFFFFF
The specified area is Config area and Start addrerss or End address does not match with specified value.	Parameter error	0xFFFFFFFF	0xFFFFFFFF

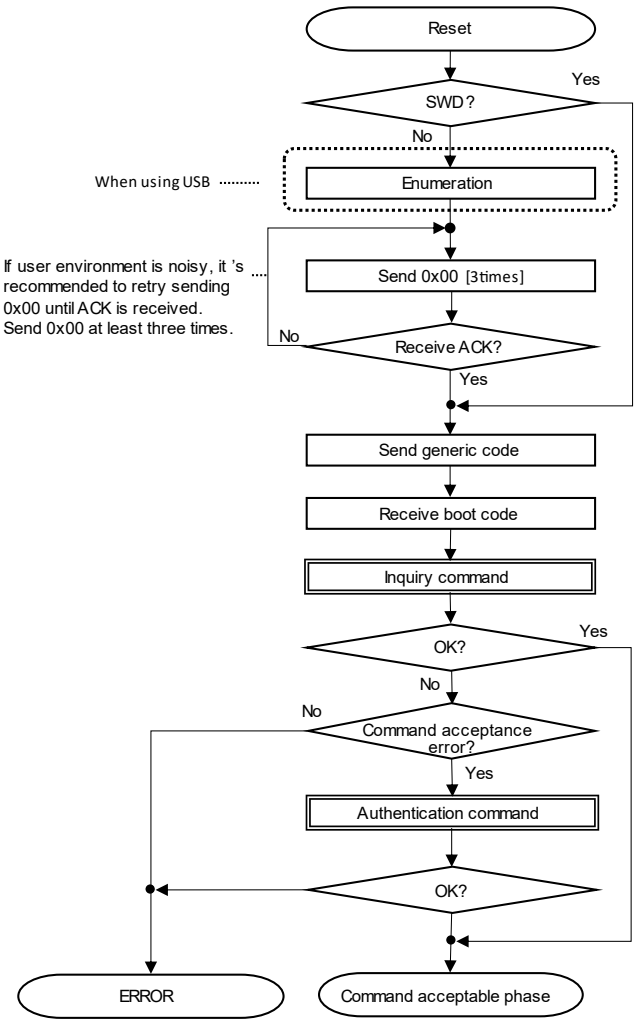
7. Flow examples

7.1 Beginning communication

(GrpA, GrpB, and GrpC)

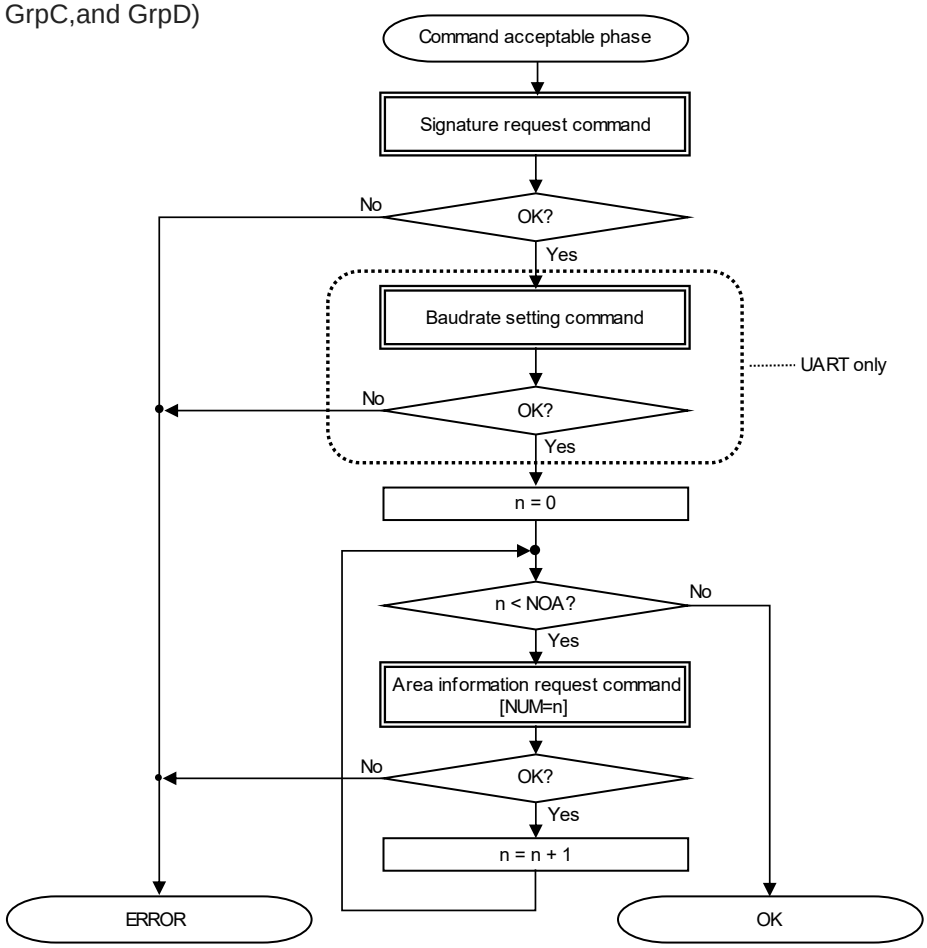


(GrpD)



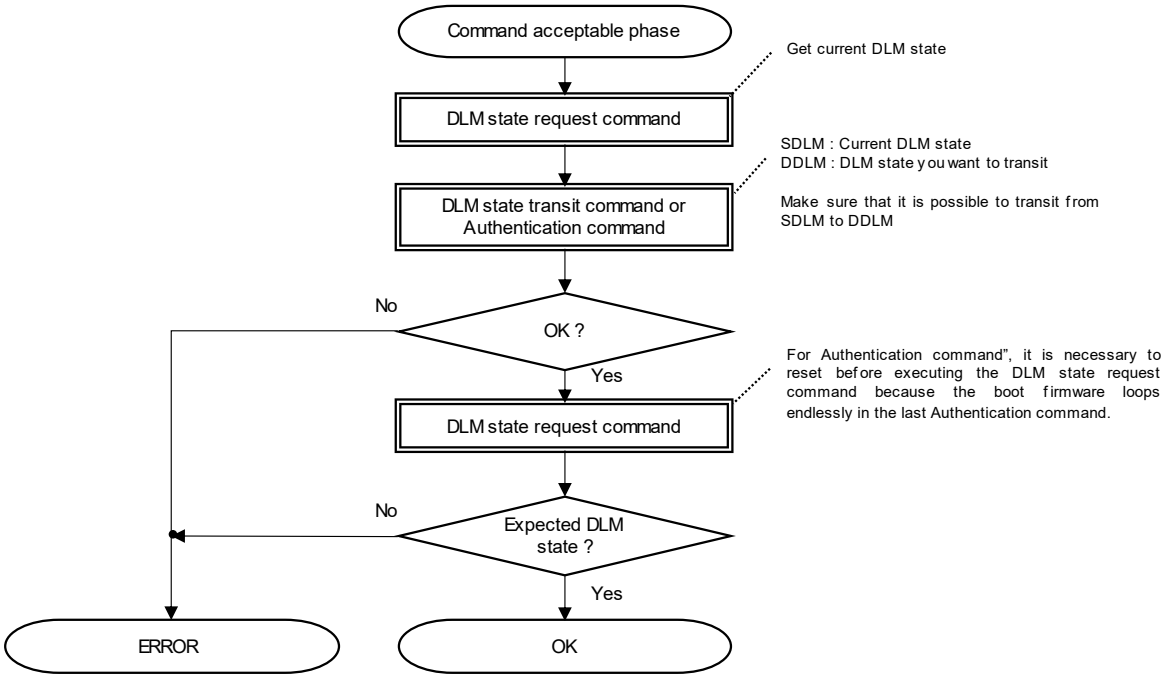
7.2 Acquisition of device information / baud rate settings

(GrpA, GrpB, GrpC,and GrpD)



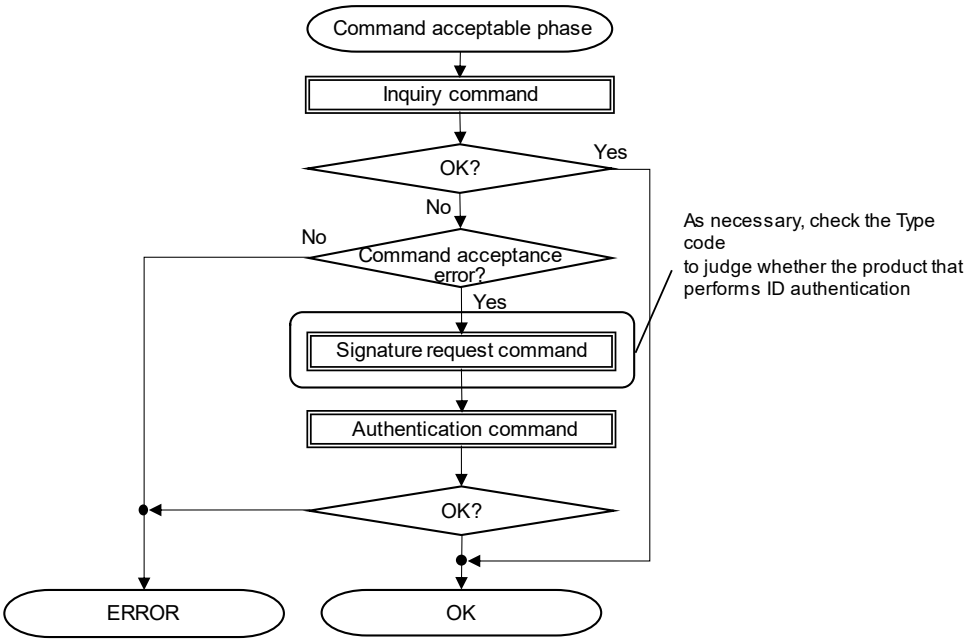
7.3 Transiting DLM state

(GrpA, GrpB, and GrpC)

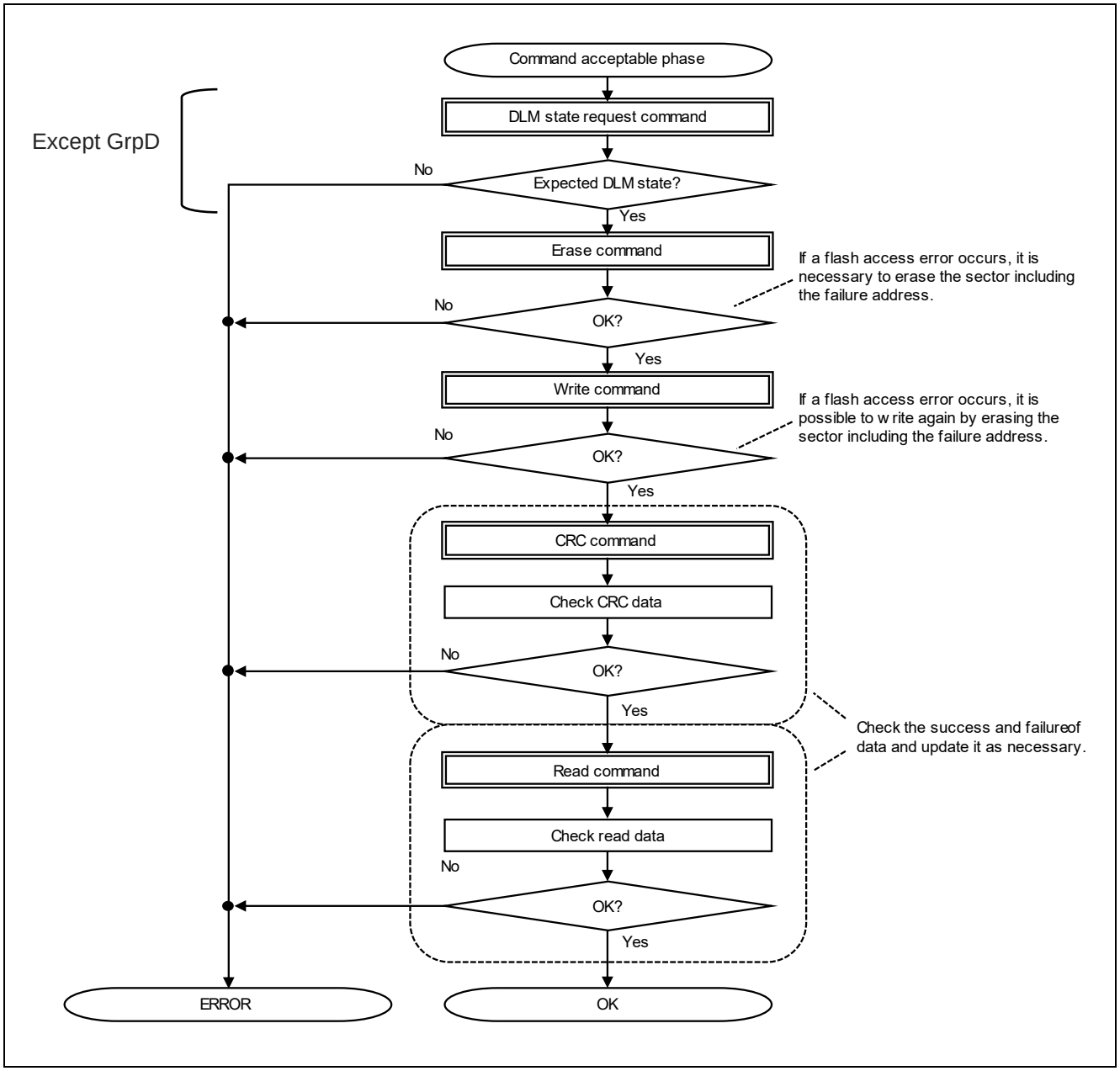


7.4 ID authentication

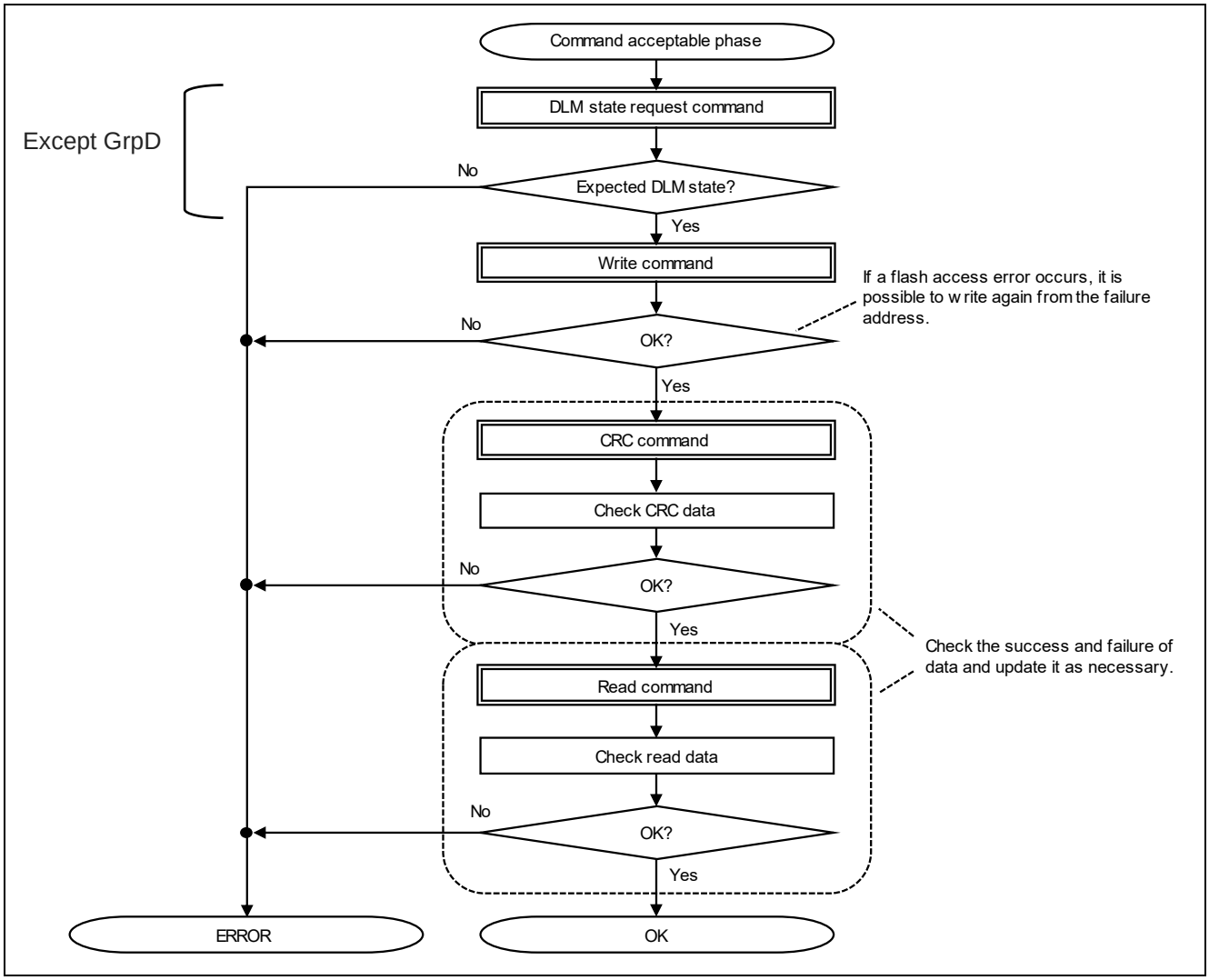
(GrpD)



7.5 User area / Data area updates

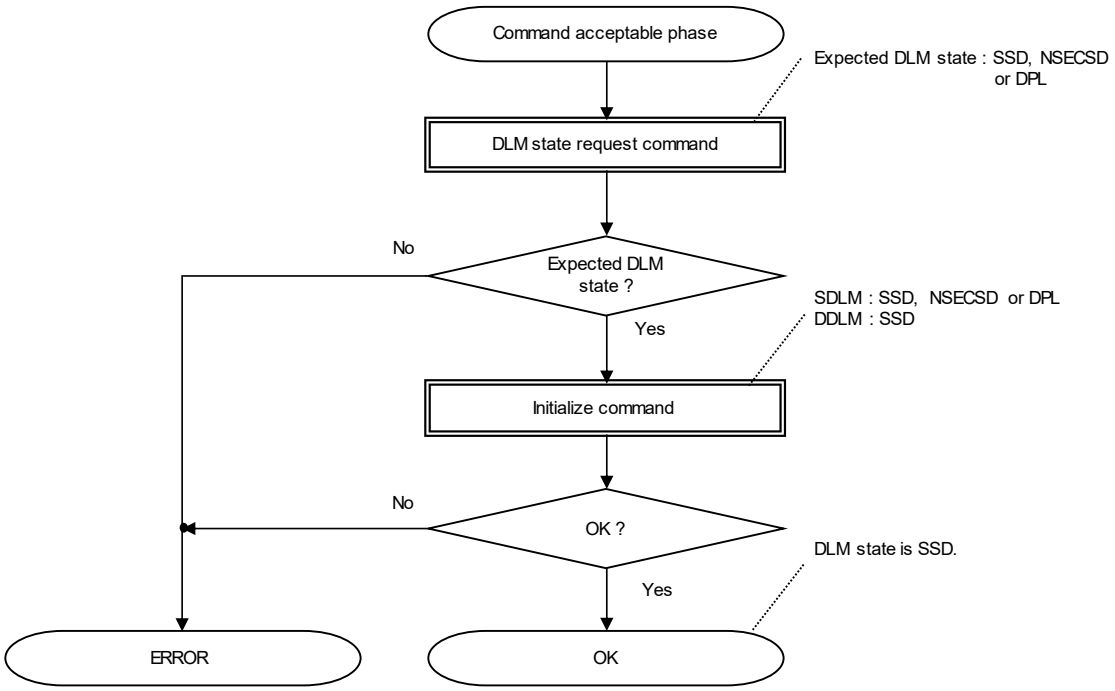


7.6 Config area updates

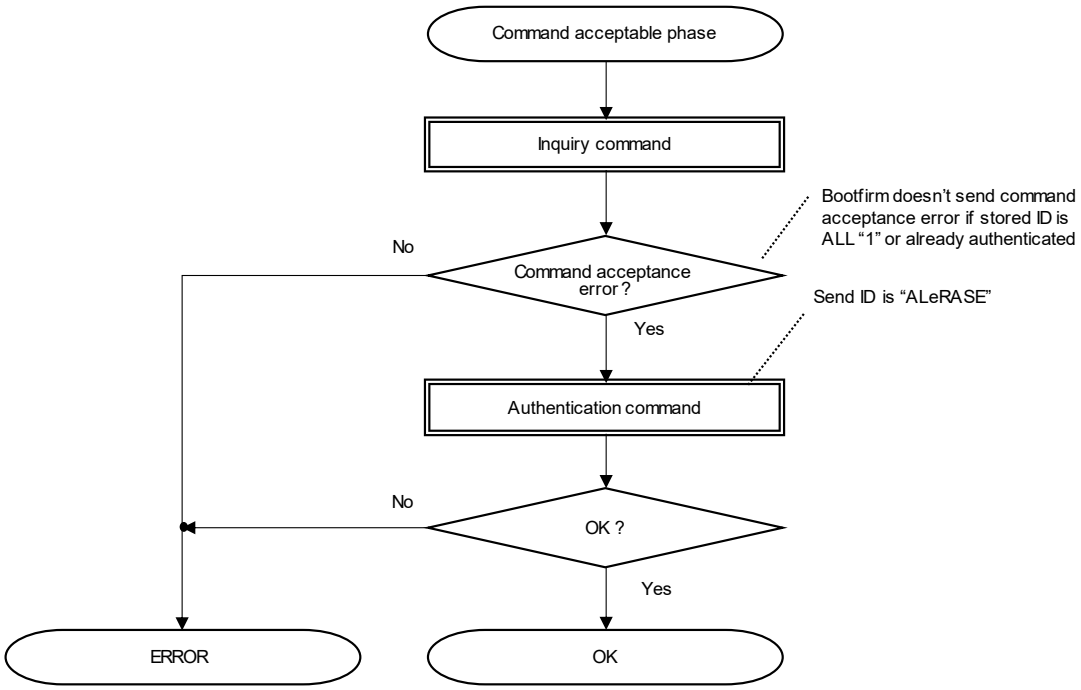


7.7 Initialize flash memory

(GrpA, GrpB, and GrpC)

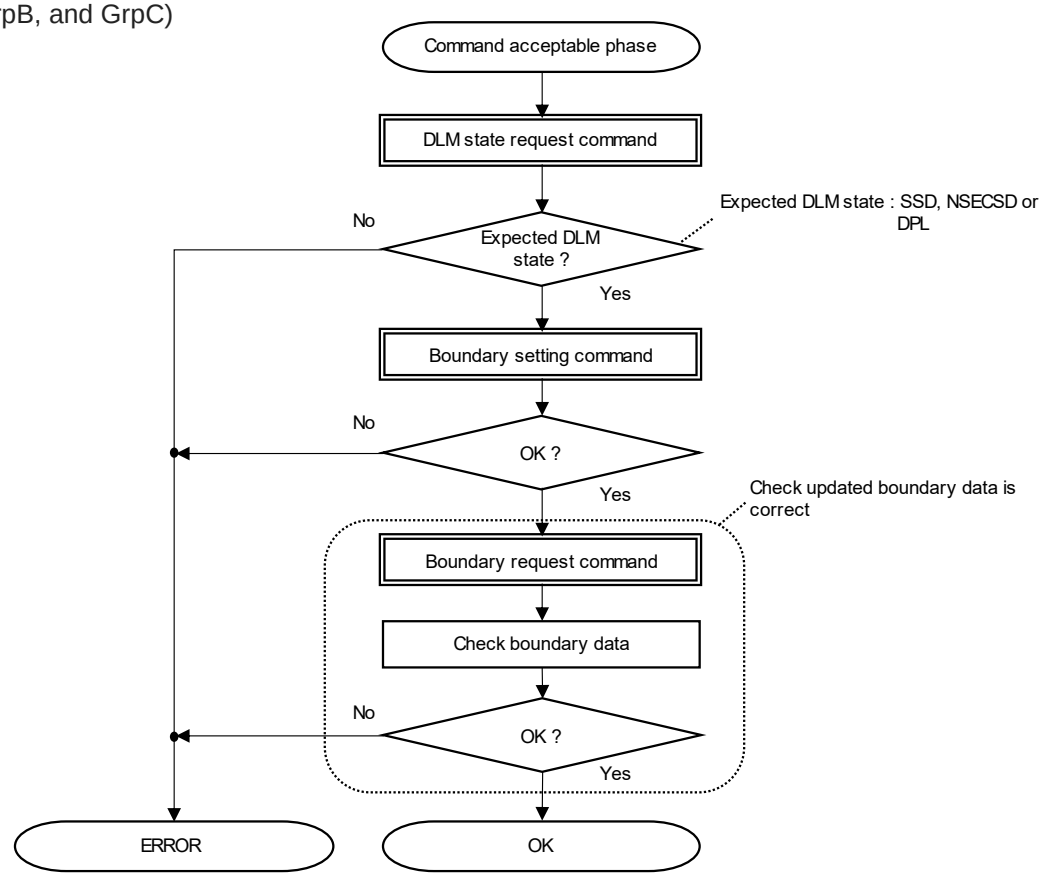


(GrpD)



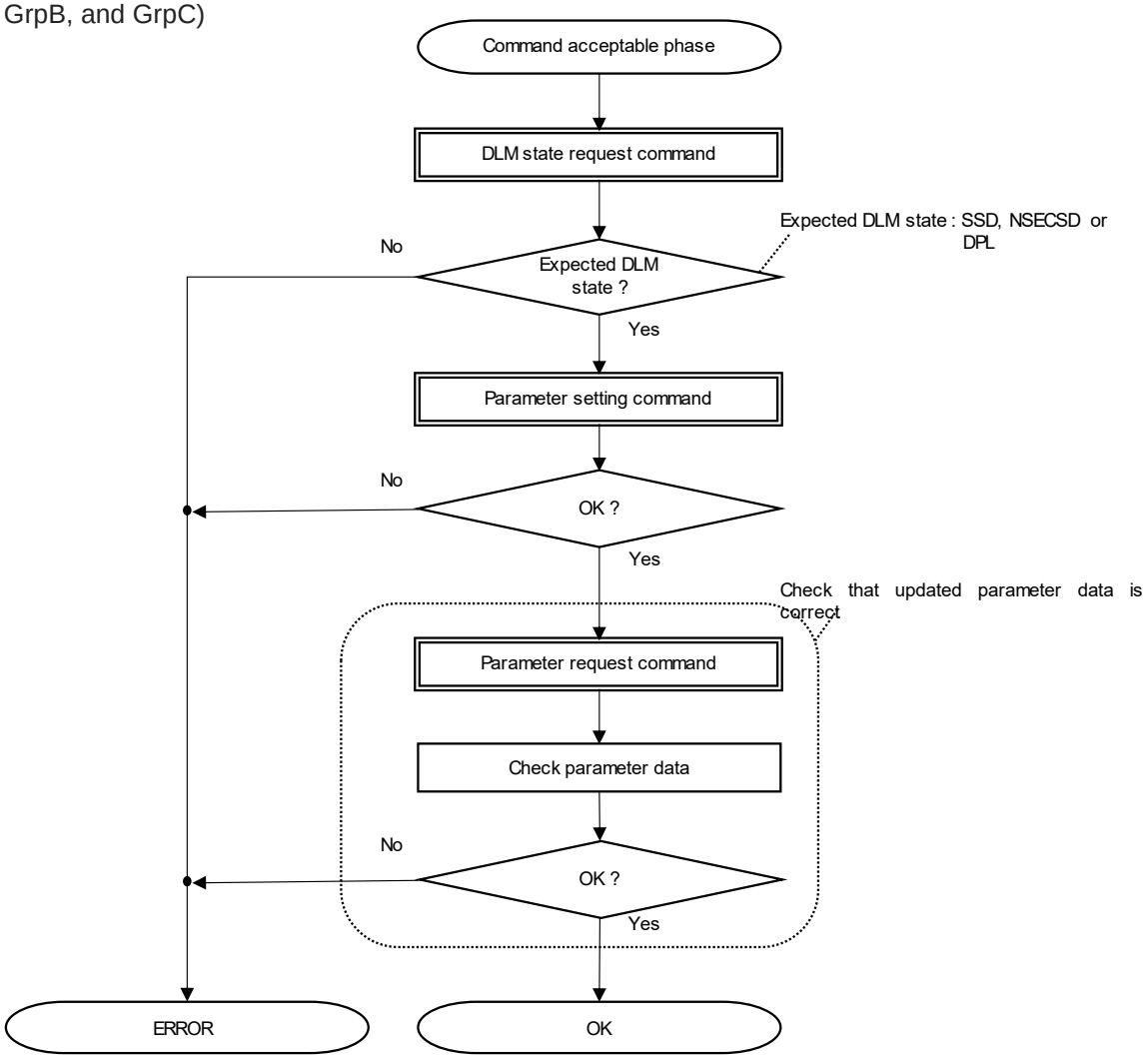
7.8 Boundary setting updates

(GrpA, GrpB, and GrpC)



7.9 Parameter setting updates

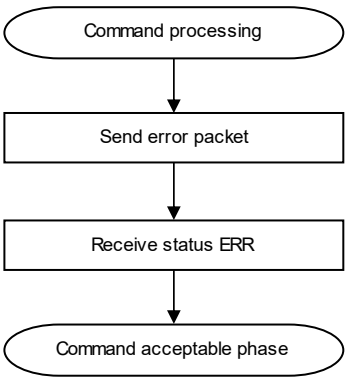
(GrpA, GrpB, and GrpC)



7.10 Command cancel

For commands that continuously send and receive packets, you can end the command by intentionally sending an error packet and return to the command acceptable phase.

(GrpA, GrpB, GrpC,and GrpD)

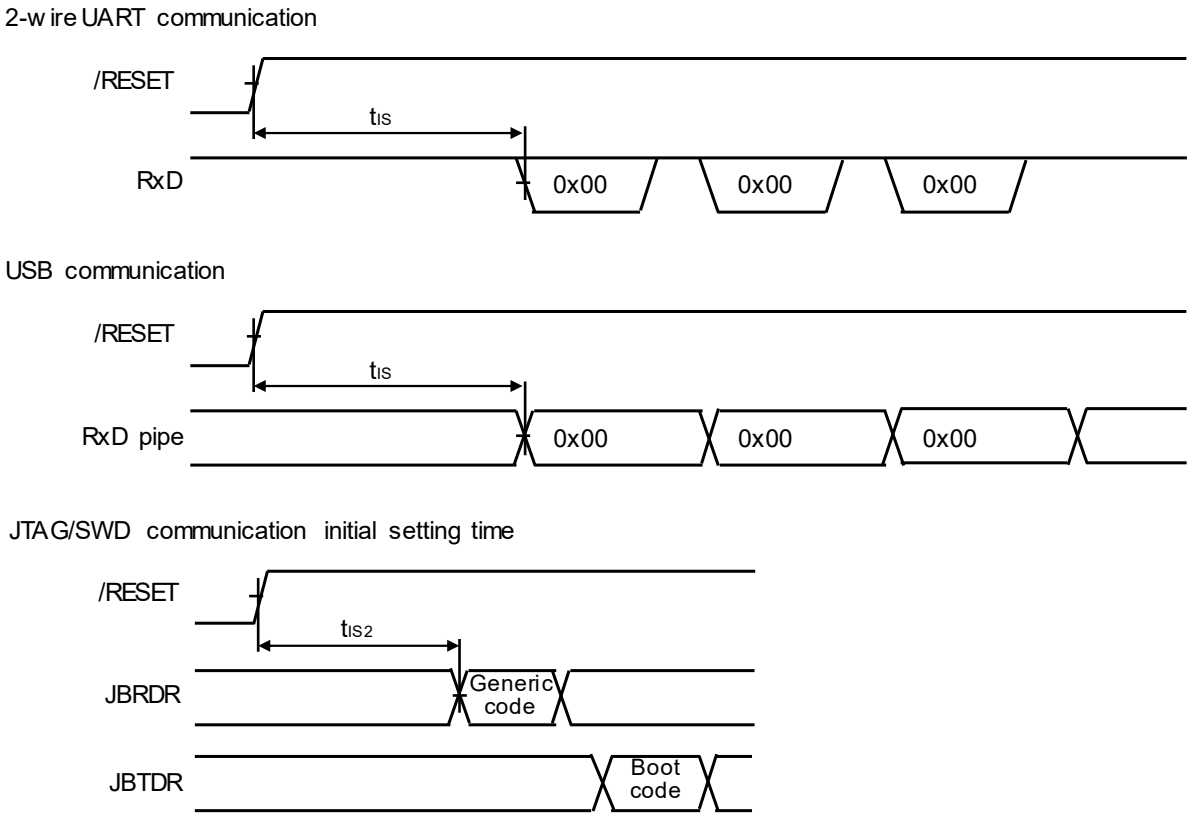


e.g.) Error packets to end the command.

Command	GrpA GrpB GrpC	GrpD	When to send error packets	Example of the error packet		
Authentication command	✔		Data packet [Response value or Authentication code]	SOD	(1 byte)	0x81
Key setting command	✔		Data packet [key data]	LNH	(1 byte)	0x00
Write command	✔	✔	Data packet [write data]	LNL	(1 byte)	0x01
Read command	✔	✔	Data packet [status OK]	RES	(1 byte)	0xFF (ERR)
				SUM	(1 byte)	0x00
				ETX	(1 byte)	0x03

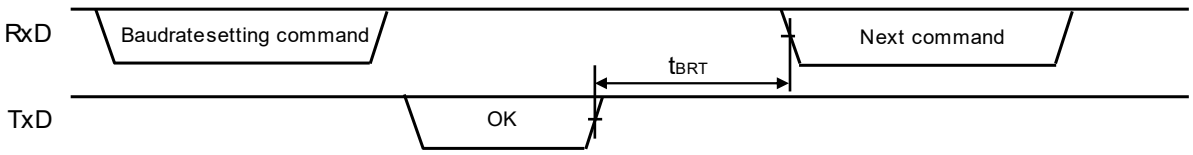
8. AC characteristics

8.1 Communication setting phase



Parameter	Product	Symbol	Min	Typ	Max	Unit
Initial setting time (when using Main-OSC (External clock)	GrpA,GrpB	tIS	-	-	280	ms
	GrpD	tIS	-	-	40	ms
Initial setting time (when using Main-OSC (Crystal resonator)	GrpA,GrpB	tIS	-	-	507	ms
	GrpC	tIS	-	-	419	ms
	GrpD	tIS	-	-	267	ms
Initial setting time (when using HOCO)	GrpA,GrpB	tIS	-	-	2613	ms
	GrpD	tIS	-	-	2336	ms
Initial setting time 2	GrpD	tIS2	-	-	36	ms

8.2 Baudrate setting command



Parameter	Product	Symbol	Min	Typ	Max	Unit
Initial setting time (when using Main-OSC (External clock)	GrpA,GrpB GrpC,GrpD	tBRT	-	-	1	ms

9. FACI command list

The FACI commands executed by each communication command are shown below

Communication command	Product	FACI command	number of issue times
DLM state transit command	GrpA,GrpB, GrpC	Configuration set	1 time
Authentication command	GrpA,GrpB, GrpC	Program	[Size of Data area / 4] times (transiting to RMA_REQ)
		Block Erase	[Size of Code S area / 8 KByte] + [Size of Code L area / 32 KByte] + [Size of Data area / 64Byte *2] times (transiting to RMA_REQ)
		Configuration set	1 time (transiting to except RMA_REQ)
			27 times (transiting to RMA_REQ and FSPR bit is 0)
			28 times (transiting to RMA_REQ and FSPR bit is 1)
	GrpD	Program	【Data Flash erasure】 1024 times
		Block Erase	【Code Flash erasure】 [Size of Code S area / 8KByte] times [Size of Code L area / 32KByte] times 【Data Flash erasure】 [Size of Data area / 64Byte] × 2 times
		Configuration set	【Config erasure】 6 times
		Forced stop *) Use to clear error status	4 times
Key setting command	GrpA,GrpB, GrpC	Configuration set	4 times
User key setting command	GrpA,GrpC	Program	Depends on designated address
Initialize command	GrpA,GrpB, GrpC	Program	[Size of Data area / 4] times
		Block Erase	[Size of User area (smaller block size) / 8 K] + [Size of User area (larger block size) / 32 K] + [Size of Data area / 64 * 2] times
		Configuration set	30 times
Boundary setting command	GrpA,GrpB, GrpC	Configuration set	1 time
Parameter setting command	GrpA,GrpB, GrpC	Configuration set	1 time
Erase command	GrpA,GrpB, GrpC	Block erase	Depends on designated address
	GrpD	Block Erase	Depends on designated address
		Forced stop *) Use to clear error status	1 time
Write command	GrpA,GrpB, GrpC	Program	Depends on designated address
		Configuration set	Depends on designated address
	GrpD	Program	Depends on designated address
		Configuration set	Depends on designated address
		Forced stop *) Use to clear error status	1 time

10. Precaution list

10.1 Write command

(1) If permanent block protection in the Config area is set, the protected area cannot be rewritten. Therefore, rewrite the protected area before setting the permanent block protection.

10.2 CRC command

(1) Since erased Data area's value is undefined, calculated CRC data would be incorrect if range of calculating CRC data includes erased Data area.

11. Cause for operation stop

The boot firmware enters an infinite loop in the following cases

11.1 Initialization phase

- When following CPU exceptions occur
  NMI / HardFault / MemManage / BusFault / UsageFault / SecureFault / SVCall / DebugMonitor / PendSV / SysTick

11.2 Communication setting phase

- When the USB cable is disconnected when the USB status is "Configured"
- When following CPU exceptions occur
  NMI / HardFault / MemManage / BusFault / UsageFault / SecureFault / SVCall / DebugMonitor / PendSV / SysTick

11.3 Command acceptance phase

- When the USB cable is disconnected when the USB status is "Configured"
- When following CPU exceptions occur
  NMI / HardFault / MemManage / BusFault / UsageFault / SecureFault / SVCall / DebugMonitor / PendSV / SysTick

12. Cause for software reset

Boot firmware performs software reset in the following cases

12.1 Communication setting phase

- When MD=1 is detected and not requested SWD boot mode during communication mode judgement

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Oct 30, 2020	—	First release
1.10	Jun 30, 2021	1	Added three product groups
1.20	Jun. 20, 2022	1, 5, 8-14, 18, 21, 27-28, 40, 47, 50, 74, 79, 89, 91, 106	Added RA6T2, Merged RA4E1 and RA6E1 from another Application Note. Corrected some misdescriptions.
1.30	Mar. 15, 2023	Many pages	Add RA4E2 and RA6E2 (SWD Boot, ID authentication)
1.40	May 30, 2023	Many pages	Add RA4T1 and RA6T3
1.50	Mar. 11, 2024	76	Change the description of “6.15.2.2 Data packet [Signature]”

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/.