

Renesas RX Family

Renesas Boot Firmware for RX74M and RX74N MCU Groups

Introduction

This application note describes the communication protocol, command set, and usage of the boot firmware provided with Renesas RX74M and RX74N MCU Group.

Target Device

RX74M MCU Group

RX74N MCU Group

Contents

1.	Terminology	11
1.1	Boot Firmware	11
1.2	Memory (Flash memory / MRAM)	11
1.3	Device Lifecycle Management (DLM)	11
1.4	Authentication Level (AL)	11
1.5	Protection Level (PL)	12
1.6	Secure/Non-secure	12
1.7	Block Protection	13
1.8	Image	13
2.	System Architecture	14
2.1	RX74M, RX74N MCU Group	14
2.2	Precautions of Power Supply	15
3.	Communication Methods	15
3.1	2-wire UART communication	16
3.2	Universal Serial Bus (USB) Communication	16
3.3	JTAG/SWD Communication	17
3.4	FINE Communication	18
4.	General Procedure	19
4.1	Sequence Diagram (Generic Sequence)	19
4.2	State Transition Diagram (Generic State Transition)	20
4.3	Initialization Phase	20
4.3.1	Processing Procedure	20
4.4	Communication Setting Phase	21
4.4.1	Processing Procedure	21
4.4.2	Settings of the 2-wire UART Communication	22
4.4.3	Settings of the USB Communication	23

4.5	Command Acceptable Phase.....	24
4.5.1	Processing Procedure	24
5.	Packet Format.....	24
5.1.1	Elements in the Packet	24
5.1.2	Command Packet.....	24
5.1.3	Data Packet.....	25
5.1.4	CMD: Command Code.....	25
5.1.5	RES: Response Code	26
5.1.6	STS: Status Code.....	26
5.1.7	ST2: Status Details.....	27
5.1.8	ADR: Failure Address	27
5.1.9	DLM: Device Lifecycle Management State Code	28
6.	Command List.....	28
6.1	Device Lifecycle Management	30
6.2	DLM State Transit Command.....	31
6.2.1	Packets.....	31
6.2.1.1	Command Packet.....	31
6.2.1.2	Data Packet [Status OK]	32
6.2.1.3	Data Packet [Status ERR].....	32
6.2.2	Processing Procedure	32
6.2.3	Status Information from the Microcontroller	33
6.2.4	DLM State Transition	33
6.3	DLM State Request Command	34
6.3.1	Sequence Diagram.....	34
6.3.2	Packets.....	34
6.3.2.1	Command Packet.....	34
6.3.2.2	Data Packet [DLM State].....	34
6.3.2.3	Data Packet [Status ERR].....	35
6.3.3	Processing Procedure	35
6.3.4	Status Information from the Microcontroller	35
6.4	Protection Level Transit Command.....	35
6.4.1	Sequence Diagram.....	36
6.4.2	Packets.....	36
6.4.2.1	Command Packet.....	36
6.4.2.2	Data Packet [Status OK]	36
6.4.2.3	Data Packet [Status ERR].....	37
6.4.2.4	Processing Procedure	37
6.4.3	Status Information from the Microcontroller	38
6.4.4	Protection Level Transition.....	38

6.5	Protection Level Request Command	39
6.5.1	Sequence Diagram.....	39
6.5.2	Packets.....	39
6.5.2.1	Command Packet.....	39
6.5.2.2	Data Packet [Protection Level].....	39
6.5.2.3	Data Packet [Status ERR].....	40
6.5.3	Processing Procedure	40
6.5.4	Status Information from the Microcontroller	40
6.6	Authentication Level Request Command.....	40
6.6.1	Sequence Diagram.....	41
6.6.2	Packets.....	41
6.6.2.1	Command Packet.....	41
6.6.2.2	Data Packet [Authentication Level]	41
6.6.2.3	Data Packet [Status ERR].....	41
6.6.3	Processing procedure	42
6.6.4	Status Information from the Microcontroller	42
6.7	Authentication Command.....	42
6.7.1	Sequence Diagram.....	43
6.7.2	Packets.....	44
6.7.2.1	Command Packet.....	44
6.7.2.2	Data Packet [Challenge Value or Unique ID].....	44
6.7.2.3	Data Packet [Response Value or Authentication Code]	45
6.7.2.4	Data Packet [Status OK]	45
6.7.2.5	Data Packet [Status ERR].....	45
6.7.3	Processing Procedure	45
6.7.4	Status Information from the Microcontroller	48
6.7.5	Precautions	49
6.7.6	DLM State Transition	49
6.7.7	Authentication Level Transition	49
6.7.8	Response Value Calculation	50
6.8	Key Setting Command	50
6.8.1	Sequence Diagram.....	50
6.8.2	Packets.....	51
6.8.2.1	Command Packet.....	51
6.8.2.2	Data Packet [Key Data].....	52
6.8.2.3	Data Packet [Status OK]	53
6.8.2.4	Data Packet [Status ERR].....	53
6.8.3	Processing Procedure	53
6.8.4	Status Information from the Microcontroller	55
6.8.5	Precautions	55

6.8.6	Key type that can be set in each Authentication Level	55
6.9	User Key Setting Command.....	56
6.9.1	Sequence Diagram.....	56
6.9.2	Packets.....	57
6.9.2.1	Command Packet.....	57
6.9.2.2	Data Packet [Key Data (1st)].....	58
6.9.2.3	Data Packet [Key Data (2 nd)].....	59
6.9.2.4	Data Packet [Status OK]	59
6.9.2.5	Data Packet [Status ERR].....	59
6.9.3	Processing Procedure	59
6.9.4	Status Information from the Microcontroller	61
6.9.4.1	User Key List.....	62
6.10	Key Verify Command	63
6.10.1	Sequence Diagram.....	63
6.10.2	Packets.....	64
6.10.2.1	Command Packet.....	64
6.10.2.2	Data Packet [Status OK]	64
6.10.2.3	Data Packet [Status ERR].....	64
6.10.2.4	Processing Procedure	64
6.10.3	Status Information from the Microcontroller	65
6.11	User Key Verify Command.....	65
6.11.1	Sequence Diagram.....	66
6.11.2	Packets.....	66
6.11.2.1	Command Packet.....	66
6.11.2.2	Data Packet [Status OK]	66
6.11.2.3	Data Packet [Status ERR].....	67
6.11.3	Processing Procedure	67
6.11.4	Status Information from the Microcontroller	68
6.12	Initialize Command.....	68
6.12.1	Sequence Diagram.....	69
6.12.2	Packets.....	69
6.12.2.1	Command Packet.....	69
6.12.2.2	Data Packet [Status OK]	69
6.12.2.3	Data Packet [Status ERR].....	70
6.12.3	Processing Procedure	70
6.12.4	Status Information from the Microcontroller	72
6.12.5	Precautions	72
6.12.6	Protection Level Transition.....	73
6.13	Boundary Setting Command	73
6.13.1	Sequence Diagram.....	73

6.13.2	Packets.....	74
6.13.2.1	Command Packet.....	74
6.13.2.2	Data Packet [Status OK]	74
6.13.2.3	Data Packet [Status ERR].....	74
6.13.3	Processing Procedure	75
6.13.4	Status Information from the Microcontroller	75
6.13.5	Example of Use	76
6.14	Boundary Request Command.....	77
6.14.1	Sequence Diagram.....	77
6.14.2	Packets.....	77
6.14.2.1	Command Packet.....	77
6.14.2.2	Data packet [Boundary Setting Data].....	77
6.14.2.3	Data Packet [Status ERR].....	78
6.14.3	Processing Procedure	78
6.14.4	Status Information from the Microcontroller	78
6.15	Parameter Setting Command.....	78
6.15.1	Sequence Diagram.....	79
6.15.2	Packets.....	80
6.15.2.1	Command Packet.....	80
6.15.2.2	Data Packet [Status OK]	81
6.15.2.3	Data Packet [Status ERR].....	81
6.15.3	Processing Procedure	81
6.15.4	Status Information from the Microcontroller	82
6.16	Parameter Request Command	82
6.16.1	Sequence Diagram.....	83
6.16.2	Packets.....	83
6.16.2.1	Command Packet.....	83
6.16.2.2	Data Packet [Parameter Data]	84
6.16.2.3	Data Packet [Status ERR].....	84
6.16.3	Processing Procedure	85
6.16.4	Status Information from the Microcontroller	85
6.17	Inquiry Command	86
6.17.1	Sequence Diagram.....	86
6.17.2	Packets.....	86
6.17.2.1	Command Packet.....	86
6.17.2.2	Data Packet [Status OK]	86
6.17.2.3	Data Packet [Status ERR].....	87
6.17.3	Processing Procedure	87
6.17.4	Status Information from the Microcontroller	87
6.18	Signature Request Command.....	87

6.18.1	Sequence Diagram.....	88
6.18.2	Packets.....	88
6.18.2.1	Command Packet.....	88
6.18.2.2	Data Packet [Signature]	89
6.18.2.3	Data Packet [Status ERR].....	89
6.18.3	Processing Procedure	89
6.18.4	Status Information from the Microcontroller	90
6.19	Area Information Request Command.....	90
6.19.1	Sequence Diagram.....	90
6.19.2	Packets.....	90
6.19.2.1	Command Packet.....	90
6.19.2.2	Data Packet [Area Information]	91
6.19.2.3	Data Packet [Status ERR].....	92
6.19.3	Processing Procedure	92
6.19.4	Status Information from the Microcontroller	92
6.19.5	Specific value of Area Information	93
6.20	Baudrate Setting Command.....	94
6.20.1	Sequence Diagram.....	95
6.20.2	Packets.....	95
6.20.2.1	Command Packet.....	95
6.20.2.2	Data Packet [Status OK]	96
6.20.2.3	Data Packet [Status ERR].....	96
6.20.3	Processing Procedure	96
6.20.4	Status Information from the Microcontroller	97
6.21	Endian Notice Command	98
6.21.1	Sequence Diagram.....	98
6.21.2	Packets.....	98
6.21.2.1	Command Packet.....	98
6.21.2.2	Data Packet [Status OK]	99
6.21.2.3	Data Packet [Status ERR].....	99
6.21.3	Processing Procedure	99
6.21.4	Status Information from the Microcontroller	100
6.21.5	Affected commands by Endian notice command	100
6.22	Write Command	100
6.22.1	Sequence Diagram.....	101
6.22.2	Packets.....	101
6.22.2.1	Command Packet.....	101
6.22.2.2	Data Packet [Write Data].....	102
6.22.2.3	Data Packet [Status OK]	102
6.22.2.4	Data Packet [Status ERR].....	102

6.22.3	Processing Procedure	102
6.22.4	Status Information from the Microcontroller	104
6.22.5	Precautions	105
6.23	Read Command	106
6.23.1	Sequence Diagram.....	106
6.23.2	Packets.....	107
6.23.2.1	Command Packet.....	107
6.23.2.2	Data Packet [Status OK]	107
6.23.2.3	Data Packet [Status ERR].....	107
6.23.2.4	Data Packet [Read Data]	107
6.23.3	Processing Procedure	107
6.23.4	Status Information from the Microcontroller	109
6.24	CRC Command	110
6.24.1	Sequence Diagram.....	110
6.24.2	Packets.....	110
6.24.2.1	Command Packet.....	110
6.24.2.2	Data Packet [CRC Data]	111
6.24.2.3	Data Packet [Status ERR].....	111
6.24.3	Processing Procedure	111
6.24.4	Status Information from the Microcontroller	112
6.25	Code Certificate Update Command	112
6.25.1	Sequence Diagram.....	113
6.25.2	Packets.....	113
6.25.2.1	Command Packet.....	113
6.25.2.2	Data Packet [Key/Code certificate data]	114
6.25.2.3	Data Packet [status OK].....	114
6.25.2.4	Data Packet [status ERR]	114
6.25.3	Processing Procedure	114
6.25.4	Status Information from the Microcontroller	117
6.25.5	Precautions	118
6.26	Code Certificate Check Command.....	118
6.26.1	Sequence Diagram.....	119
6.26.2	Packets.....	119
6.26.2.1	Command Packet.....	119
6.26.2.2	Data Packet [Anti-rollback counter value]	120
6.26.2.3	Data Packet [Status ERR].....	120
6.26.3	Processing Procedure	120
6.26.4	Status Information from the Microcontroller	121
6.27	Encrypted Data Write Command	121
6.27.1	Sequence Diagram.....	122

6.27.2	Packets.....	123
6.27.2.1	Command Packet.....	123
6.27.2.2	Data Packet [Parameter].....	123
6.27.2.3	Data Packet [AL Key].....	126
6.27.2.4	Data Packet [Nonce and MAC for Encrypted User Data]	127
6.27.2.5	Data Packet [Encrypted User Data]	127
6.27.2.6	Data Packet [Status OK]	128
6.27.2.7	Data Packet [Status ERR].....	128
6.27.3	Processing Procedure	128
6.27.3.1	Data Packet Reception	131
6.27.4	Status Information from the Microcontroller	131
6.27.5	Precautions	133
6.27.6	Device State after Command Execution	134
6.27.7	DLM State Transitions.....	134
7.	Flow Examples.....	135
7.1	Beginning Communication	135
7.2	Acquisition of Device Information/Baudrate Settings/Endian Notice	136
7.3	Transiting DLM State.....	137
7.4	Transiting Protection Level.....	138
7.5	Transiting Authentication Level	139
7.6	Data Programming	140
7.7	Encrypted Data Programming	141
7.8	Initialize Memory	142
7.9	Storing Keys	143
7.10	Updating Boundary or Parameter Setting	144
7.11	Storing Code Certificate	145
7.12	Downloading Whole Image	146
7.13	Downloading Non-secure Image.....	148
7.14	Command Cancel.....	149
8.	AC Characteristics	149
8.1.1	Communication Setting Phase.....	149
8.1.2	DLM State Transit Command.....	150
8.1.3	DLM State Request Command	150
8.1.4	Protection Level Transit Command.....	150
8.1.5	Protection Level Request Command	151
8.1.6	Authentication Level Request Command.....	151
8.1.7	Authentication Command.....	151
8.1.8	Key Setting Command	152
8.1.9	User Key Setting Command.....	152

8.1.10	Key Verify Command	152
8.1.11	User Key Verify Command.....	152
8.1.12	Initialize Command.....	153
8.1.13	Boundary Setting Command.....	153
8.1.14	Boundary Request Command.....	154
8.1.15	Parameter Setting Command.....	154
8.1.16	Parameter Request Command	154
8.1.17	Inquiry Command.....	155
8.1.18	Signature Request Command.....	155
8.1.19	Area Information Request Command.....	155
8.1.20	Baudrate Setting Command.....	156
8.1.21	Endian Notice Command	156
8.1.22	Write Command	156
8.1.23	Read Command	156
8.1.24	CRC Command.....	157
8.1.25	Code Certificate Update Command	157
8.1.26	Code Certificate Check Command.....	157
8.1.27	Encrypted Data Write Command	158
9.	Sequencer Command List.....	158
10.	Precaution List	159
10.1.1	Authentication command.....	159
10.1.2	Key setting command.....	159
10.1.3	Initialize Command.....	159
10.1.4	Parameter setting command	159
10.1.5	Write Command	160
10.1.6	Code Certificate Update Command	160
10.1.7	Encrypted Data Write Command	161
11.	Causes for Operation Stop.....	162
11.1	Initialization Phase	162
11.2	Communication Setting Phase.....	162
11.3	Command Acceptable Phase.....	162
11.4	DLM State Transit Command.....	162
11.5	Authentication Command.....	162
11.6	Key Setting Command	162
11.7	User Key Setting Command.....	162
11.8	Key Verify Command	162
11.9	User Key Verify Command.....	162
11.10	Initialize Command.....	163
11.11	Code certificate update command	163

11.12 Code Certificate Check Command.....	163
11.13 Encrypted Data Write Command	163
12. Causes for Software Reset	163
12.1 Communication Setting Phase.....	163
Revision History	164

1. Terminology

1.1 Boot Firmware

The program embedded in the microcontroller that rewrites non-volatile memories such as flash memory and MRAM is called Boot firmware.

This rewrite operations are executed by sending boot firmware commands though serial programming interface for example 2-wire UART.

1.2 Memory (Flash memory / MRAM)

In this document, the Flash memory area and the MRAM area that can be rewritten by boot firmware may be generically referred to as "memory".

There are several types of memory such as User area and Data area and what types of the memories are supported depends on the device specifications.

Refer to the User's manual of the device or the specifications of Area information command for more information.

1.3 Device Lifecycle Management (DLM)

RX Family MCUs adopt the concept of device lifecycle and maintain the lifecycle 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, the boot firmware has a user-executable command as the only way to transition lifecycle states.

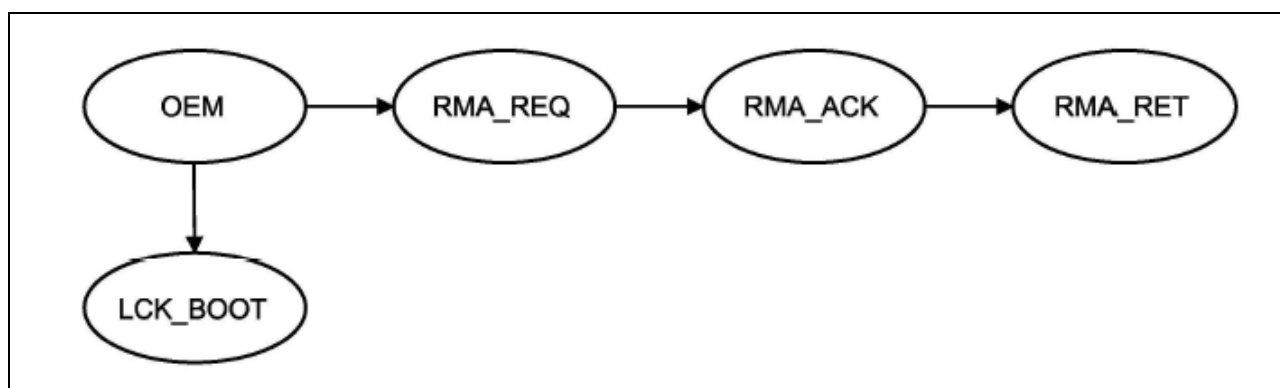


Figure 1. Device Lifecycle States

Table 1. DLM States

DLM State Name	Description
OEM	Original Equipment Manufacturer
LCK_BOOT	LoCKed BOOT interface
RMA_REQ	Return Material Authorization REQuest
RMA_ACK	Return Material Authorization ACKnowledged
RMA_RET	Return Material Authorization RETurn

1.4 Authentication Level (AL)

In RX Series, the executable commands and the range of operations that can be performed with each command are determined not only by the DLM state but also by the authentication level. There are three authentication levels: AL2, AL1, and AL0. The executable operation range is the widest at AL2, and the narrowest at AL0.

Changing the authentication level is possible only when the DLM state is OEM, so that the executable operation range at OEM is more subdivided. On the other hand, changing the authentication level when the DLM state is not OEM is not possible because the DLM state and authentication level are uniquely associated at DLM states other than OEM. To change the authentication level at OEM, change the protection level, then reset the device, or use dedicated boot firmware commands. Level change by the

boot firmware command is a temporary change, and the authentication level returns to the level before the change when resetting the device.

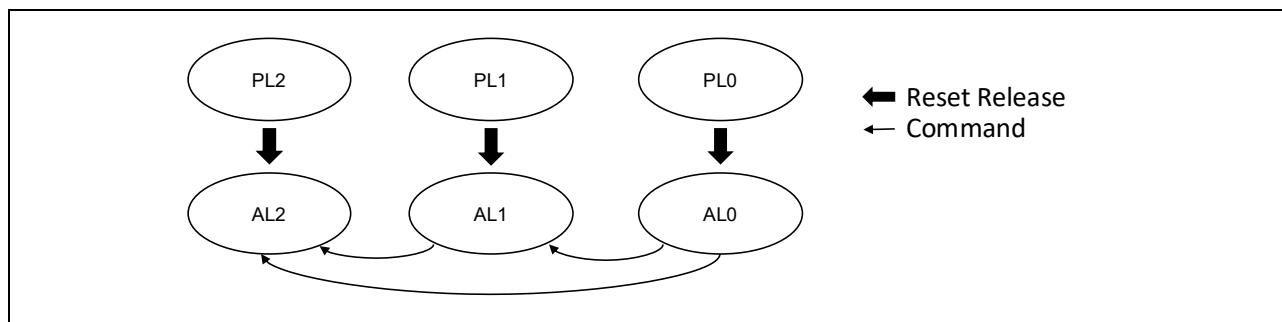


Figure 2. Authentication Level Transitions

Table 2. Authentication Level States

AL State Name	Description
AL2	Authentication Level 2
AL1	Authentication Level 1
AL0	Authentication Level 0

1.5 Protection Level (PL)

Protection level is the initial authentication level: the authentication level when the device boots is determined by the protection level. Like the authentication level, it is not possible to change the protection level when the DLM state is not OEM. To change the protection level at OEM, use dedicated boot firmware commands.

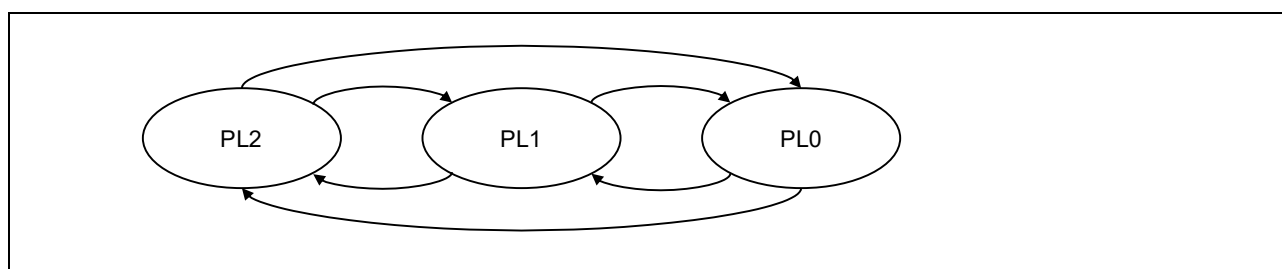


Figure 3. Protection Level Transitions

Table 3. Protection Level States

PL State Name	Description
PL2	Protection Level 2
PL1	Protection Level 1
PL0	Protection Level 0

1.6 Secure/Non-secure Attributes

This MCU has the concept of security attribute; some HW resources can be allocated as secure attribute or non-secure attribute.

Memory area also can be divided into secure attribute and non-secure attribute. Area allocated as each attribute is called secure area and non-secure.

Unauthorized access from non secure program to secure program is prevented, by writing secure program in secure area and non-secure in non-secure area.

The allocation setting of secure area and non-secure area is referred to as "boundary" in this document. Boundary can be set using Boundary setting command of boot firmware.

1.7 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 as listed in Table 4.

Table 4. Block Protection Types

Types of protection	Description
Block protection	Protection that can be released after enabled.
Permanent block protection	Protection that permanently disables releasing the Block protection setting.

Although MRAM does not have the concept of “block”, the device has block protection. Refer to the User's Manual (hardware) for the range and the unit of this protection for MRAM products. In addition, this protection protects only rewriting for MRAM products because MRAM does not have the concept of “erase”.

1.8 Image

The term “Image” is data written to flash memory or MRAM using boot firmware.

“Write data” refers to each data to be written and “image” refers to a set of write data to be written to a device or an area.

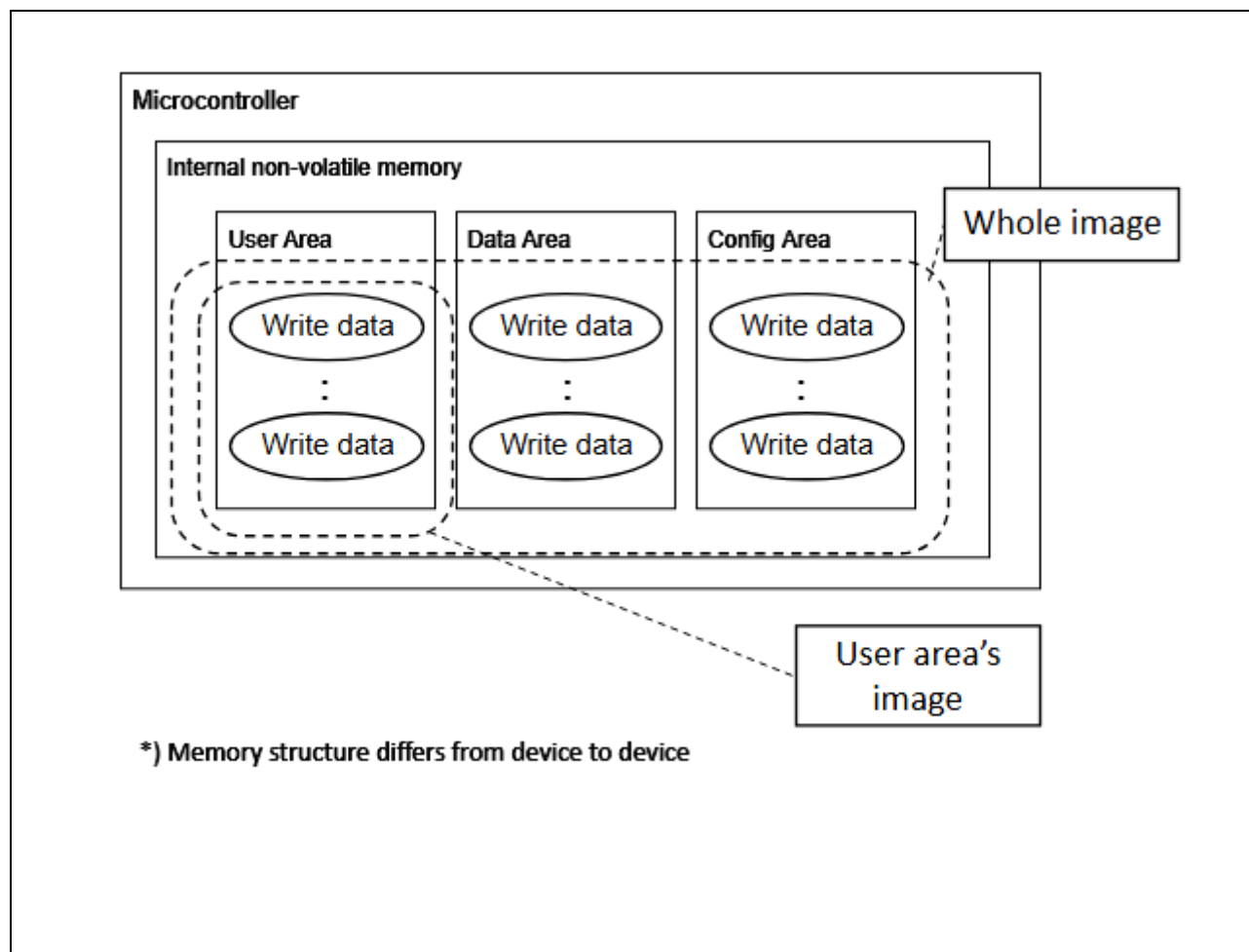


Figure 4. Memory Image Concept

2. System Architecture

Boot firmware has a serial programming interface to send and receive memory control commands between the microcontroller and the host in the serial programming mode. Boot firmware is embedded into the device.

2.1 RX74M, RX74N MCU Group

This chapter describes the system architecture of the RX74M, RX74N MCU Groups regarding the MRAM control.

Table 5. Operating Environment

System Architecture	Specifications		
CPU core	RXv3e		
Max CPU operating frequency	600 MHz (Boot firmware operating frequency: 300 MHz)		
Main-OSC	8, 10, 12, 15, 16, 20, 24, 32, 48 MHz • If neither is set, boot firmware generates its operating frequency not from Main-OSC but from HOCO. • However, if a Main-OSC whose frequency is around +/- 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 one of the following options when using USB communication: — Use a Main-OSC whose frequency is the value listed above. — Do not use a Main-OSC and use a Sub-OSC with a frequency that is supported by the specifications of the device.		
Operating voltage	VCC		1.62 V - 3.63 V
	VCC2		1.62 V - 3.63V
	VCC_USB (only USB boot mode)		3.0 - 3.6 V
	VDD		0.9 V or 0.93 V
Operating mode	Boot mode		
MRAM	Code MRAM	User area	Max.1 MB * Programming for both of Linear mode and Dual mode are supported
	Extra MRAM	User Extra MRAM	Max. 128KB
		Data area	Max. 48KB
RAM	Boot firmware uses 32KB of SRAM0, from the top (= lower) address. * The device's maximum SRAM0 size, for RX74M, is 1024KB		
Communication method	• [2-wire UART communication] — (Initial/Min) 9600 bps — (Max) 6 Mbps • [USB communication] — 12 Mbps — When performing USB communication without Main-OSC, Sub-OSC must be oscillating stably. — USB communication operation is only guaranteed for use with Windows 10 as the host OS. Use with other host OS systems is not guaranteed. • [JTAG/SWD communication]		

	— 25 MHz • [FINE communication]
--	------------------------------------

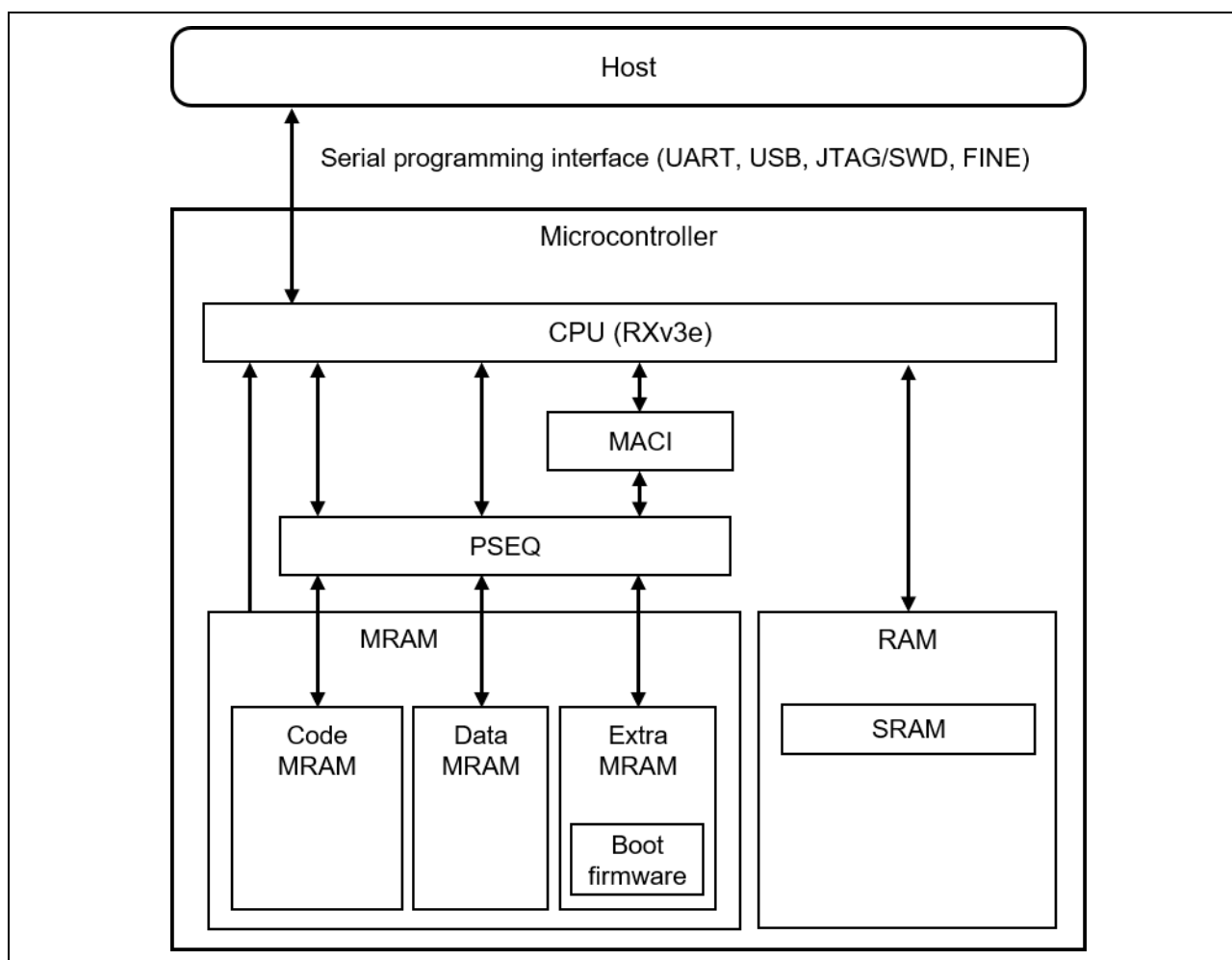


Figure 5. Block Diagram

2.2 Precautions of Power Supply

Programming during the mass production process using power supplied from the host *1 is not recommended.

If power is supplied from the host, ensure that thorough verification is conducted on the user environment before execution.

*1: "Power supplied from the host" refers to power supply from an emulator, USB bus power, etc.

3. Communication Methods

Boot firmware has interfaces for the following communication methods:

- 2-wire UART communication
- Universal Serial Bus (USB) communication
- JTAG/SWD communication
- FINE communication

3.1 2-wire UART communication

Boot firmware supports the 2-wire UART communication as shown in Figure 6.

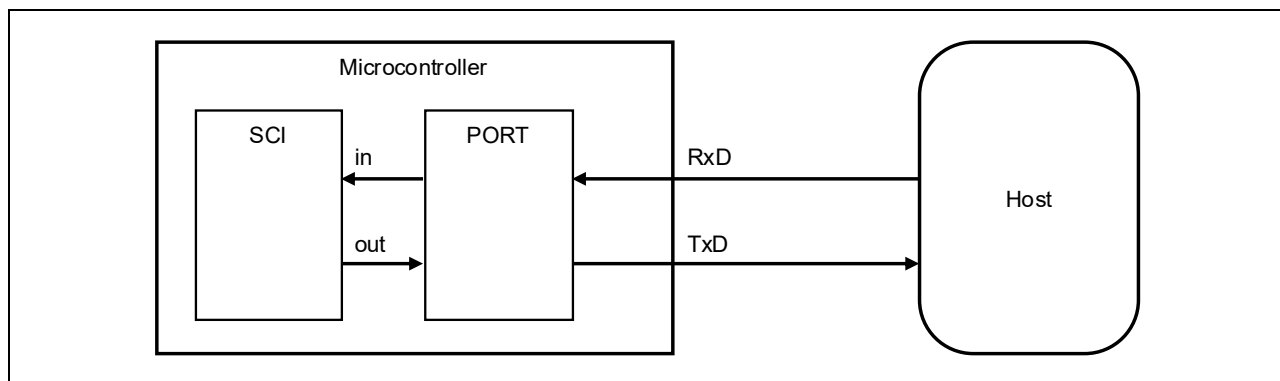


Figure 6. 2-Wire UART Communication

Table 6. UART Settings

Interface	RSCI-3 ch1
RxD	P303, input mode
TxD	P302, output mode
Transfer rate	9600 bps (minimum, until the baudrate setting command) 6 Mbps (maximum)
Data length	8 bits (LSB first)
Parity bit	None
Stop bit	1 bit

Communication is performed at 9600 bps until the baudrate setting command. After the baudrate setting command is completed, 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.

Note: If the communication cable is disconnected during communication, subsequent operations may not be guaranteed.

3.2 Universal Serial Bus (USB) Communication

Boot firmware supports USB communication as shown in Figure 7.

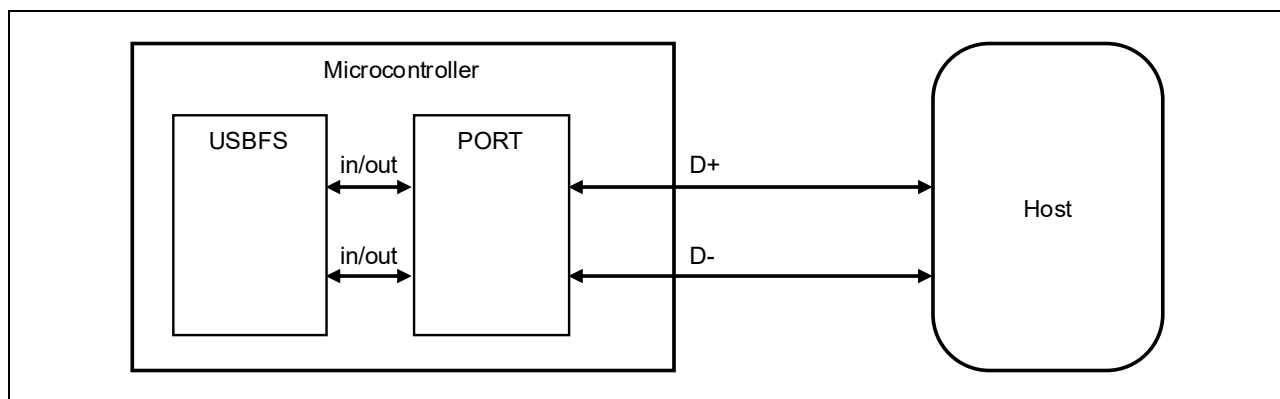


Figure 7. USB Communication

Table 7. USB Settings

Interface	USBFS
VBUS	P101, input mode
D+	Input-output mode
D-	Input-output mode
Transfer rate	12 Mbps (USB 2.0 Full-Speed)
Device class	Communication device class (CDC) - Sub-Class: Abstract Control Model (ACM) - Protocol: Common AT commands
Vendor ID	045Bh (Renesas)
Product ID	0244h
Transfer mode	Control (in/out) Bulk (in/out) Interrupt (in)
Endpoint	EP0: Default control pipe, control transfers (in/out) EP1: TxD pipe, bulk transfers (in) 64 bytes EP2: RxD pipe, bulk transfers (out) 64 bytes EP6: Control pipe, interrupt transfers (in)

Notes:

- If the USB cable is disconnected during communication, subsequent operations are not guaranteed.
- Operation of the boot firmware in bus-powered mode is not guaranteed.
The boot firmware always returns "1b" (Self-Powered) as the value of bit 6 in bmAttributes of the Configuration Descriptor, regardless of the actual power supply state.
As a result, even when the device is operating in bus-powered mode, the host recognizes it as a self-powered device.
If the boot firmware is used in a bus-powered environment, sufficient evaluation must be performed, and such use shall be at the user's own responsibility.

3.3 JTAG/SWD Communication

Boot firmware supports JTAG/SWD communication.

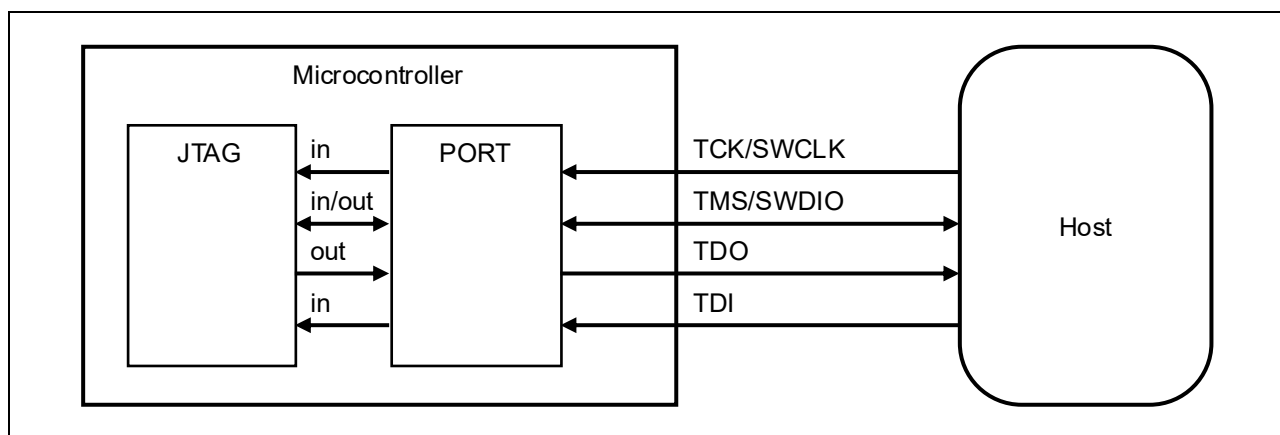
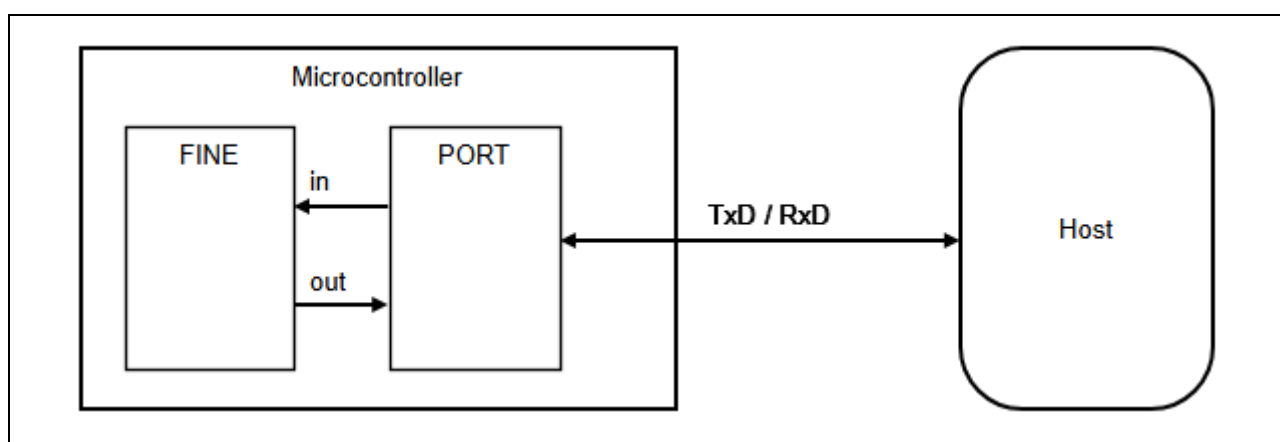
**Figure 8. JTAG/SWD Communication**

Table 8. JTAG/SWD Settings

[JTAG] TCK	P301, input mode
[JTAG] TMS	P300, input mode
[JTAG] TDO	P302, output mode
[JTAG] TDI	P303, input mode
[SWD] SWCLK	P301, input mode
[SWD] SWDIO	P300, input-output mode
Transfer rate	25 MHz (maximum)
Data length	32 bits

3.4 FINE Communication

Boot firmware supports FINE communication.

**Figure 9. FINE Communication****Table 9. JTAG/SWD Settings**

TxD / RxD	P315 (MD/FINED), input-output mode
-----------	------------------------------------

4. General Procedure

Boot firmware transitions phases in the following order after a reset release:

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

The above sequence cannot be altered.

4.1 Sequence Diagram (Generic Sequence)

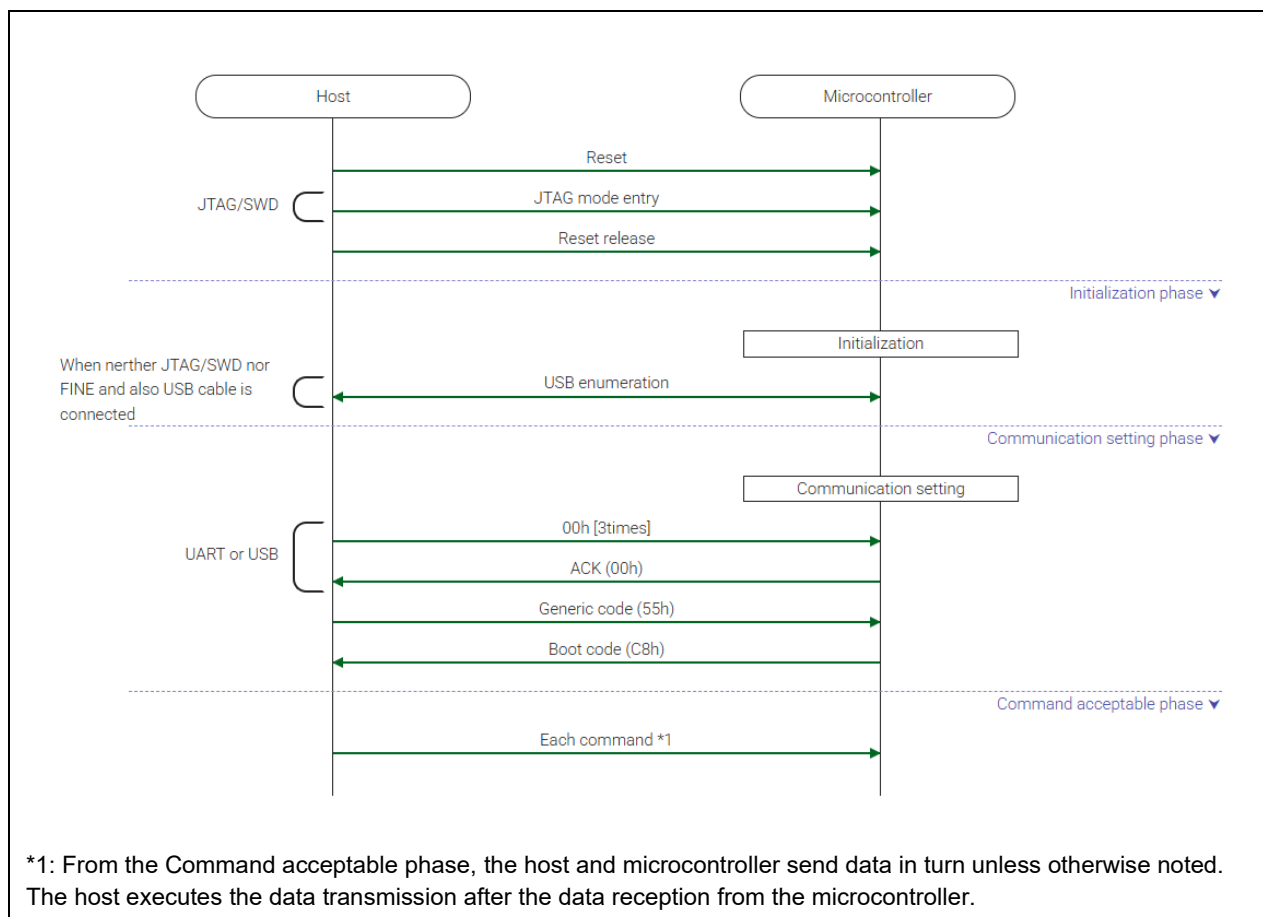


Figure 10. Sequence Diagram (Generic Sequence)

4.2 State Transition Diagram (Generic State Transition)

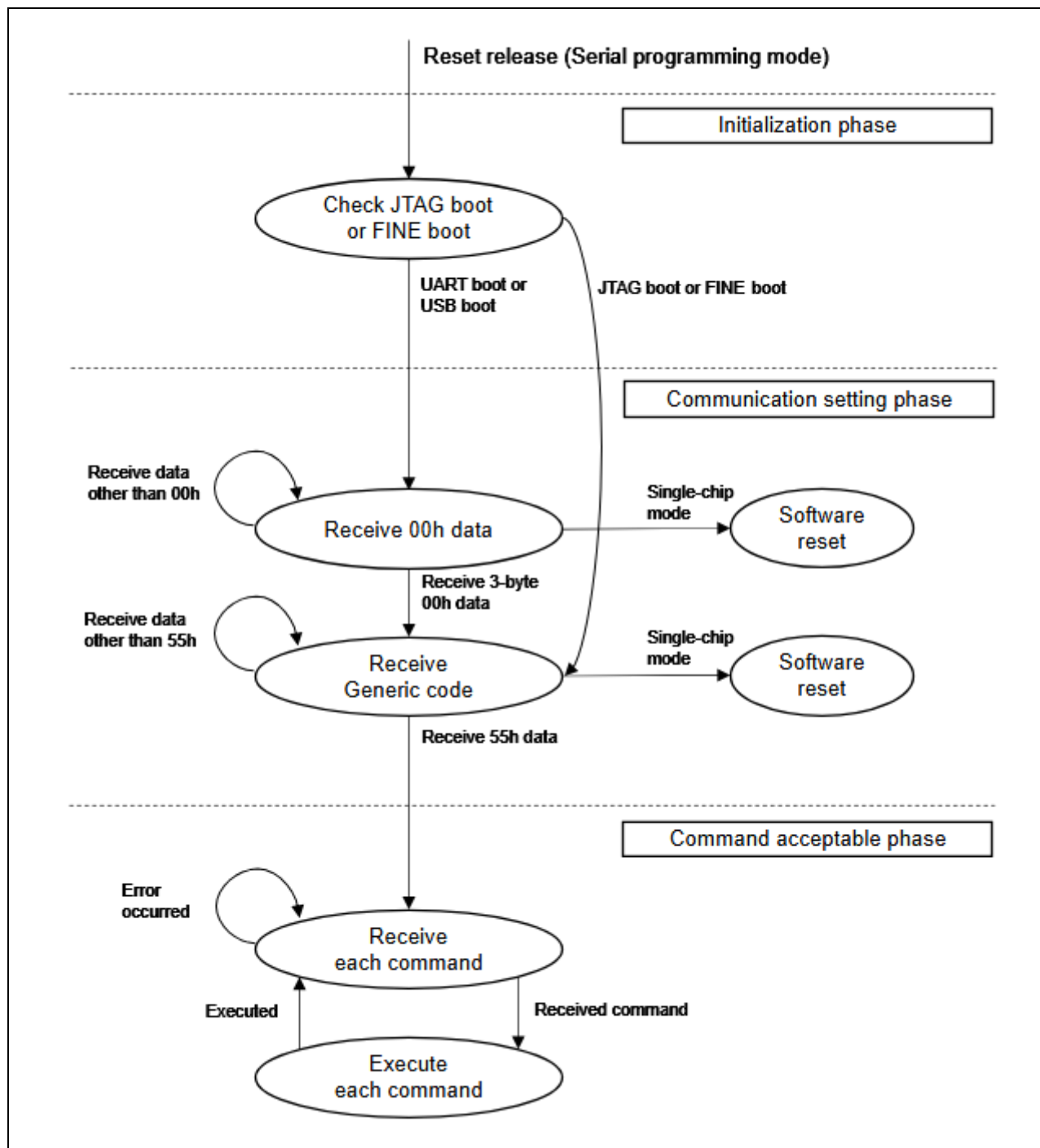


Figure 11. State Transition Diagram (Generic State Transition)

4.3 Initialization Phase

Boot firmware initializes hardware modules in this phase. After that, boot firmware transitions to the "Communication setting phase".

4.3.1 Processing Procedure

Boot firmware initializes after reset release.

The boot firmware initializes hardware modules and then transitions to the "Communication setting phase".

4.4 Communication Setting Phase

Boot firmware establishes communication with the host in this phase.

Boot firmware receives Generic code from the host, sends Boot code to the host and transitions to the "Command acceptable phase".

In addition, when the communication method is not determined in Initialization phase, boot firmware determines the communication method by receiving 00h data before Generic code reception.

4.4.1 Processing Procedure

Boot firmware waits that 00h is sent by host to determine the communication method.

However when the communication method has been determined to JTAG/SWD or FINE, boot firmware skips 00h waiting and proceed to Generic code waiting.

- Boot firmware determines SCI boot or USB boot by 00h data wait.
Boot firmware waits that 00h data is sent by the Host with both of these communication.
When 3-byte consecutive 00h data is sent by either of the communications, boot firmware decides to use the communication by which the 00h is sent, and disables the other one.

Boot firmware sends 1-byte 00h after the communication is determined, then the Host send the Generic code.

In addition, boot firmware resets the device when detected that MD pin is Hi during the 00h data wait.

- Boot firmware waits that Generic code is sent.

4.4.2 Settings of the 2-wire UART Communication

Perform the communication setting as the steps below to use SCI boot (2-wire UART communication).
Perform the data transmission in the steps below in 9600bps.

1. Release device reset with MD pin = Lo.
2. Host sends 3-bytes 00h data. *1
(Perform 00h data transmission until ACK is received in step 3.)
3. Boot firmware sends 00h data (ACK).
4. Host sends 55h data (Generic code).
5. Boot firmware sends C8h data (Boot code).

*1: If ACK is not returned even after sending 00h data, check the communication environment and try again from reset release.

Timing chart of the communication setting is shown below.

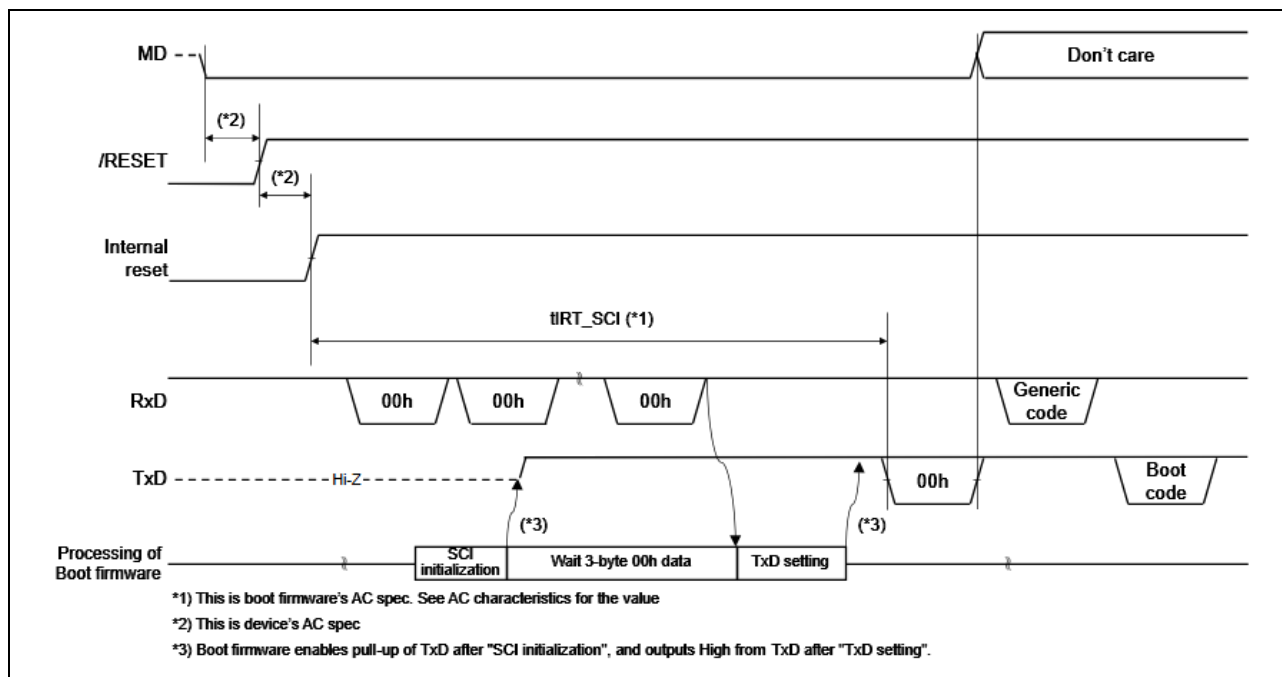


Figure 12. 2-wire UART Communication Setting

4.4.3 Settings of the USB Communication

Perform the communication setting as the steps below to use USB boot.

1. Release device reset with MD pin = Lo.

(USB cable can be connected both beforehand the device reset or after the device reset.)

2. Boot firmware executes communication initialization of USBFS, then performs enumeration with the host after detected VBUS=Hi.

(USB boot is recognized as a virtual COM port after the enumeration is completed.)

3. Host sends 3-bytes 00h data. *1

(Perform 00h data transmission until ACK is received in step 3.)

4. Boot firmware sends 00h data (ACK).

5. Host sends 55h data (Generic code).

6. Boot firmware sends C8h data (Boot code).

*1: If ACK is not returned even after sending 00h data, check the communication environment and try again from reset release.

Timing chart of the communication setting is shown below.

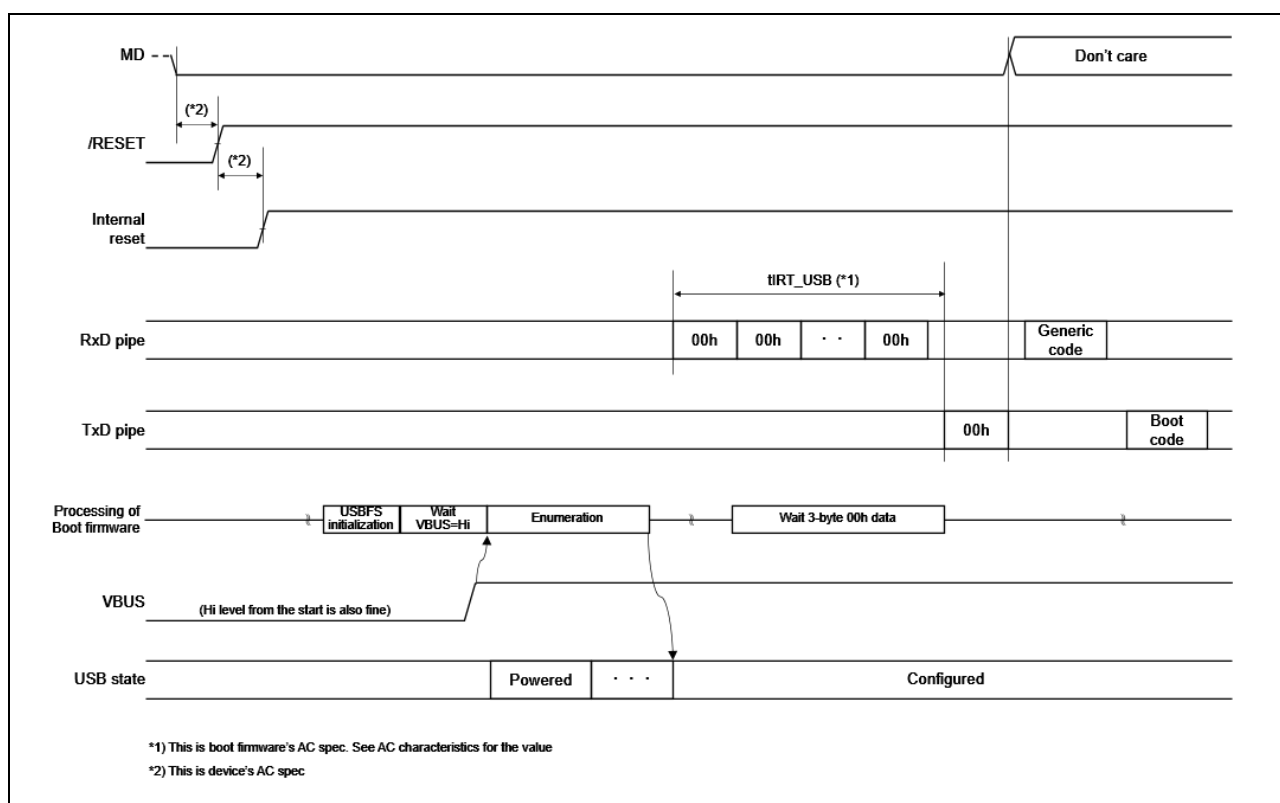


Figure 13. USB Communication Setting

4.5 Command Acceptable Phase

Boot firmware accepts the commands in this phase.

4.5.1 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 LNH and LNL of the received command packet 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 LNH and LNL of the received command packet are different from the values specified in each command, the boot firmware sends a "Packet error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

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

When a command is normally finished, boot firmware stays on the "Command acceptable phase".

5. Packet Format

Use the following packet types:

- Command packet
- Data packet

5.1.1 Elements in the Packet

- CMD: Command code
- RES: Response code
- STS: Status code
- ST2: Status details
- ADR: Failure address
- DLM: Device Lifecycle Management state code

5.1.2 Command Packet

The host sends a command packet to the microcontroller in the following format.

Table 10. Command Packet Format

Symbol	Size	Value	Description
SOH	1 byte	01h	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	0-255 bytes	-	Command information. Examples: • For Write command: Start/End address • For Baudrate setting command: UART baudrate
SUM	1 byte	-	Sum data of "LNH + LNL + CMD + Command information" (expressed as two's complement).

			For example: LNH + LNL + CMD + Command information(1) + Command information(2) + ... + Command information(n) + SUM = 00h.
ETX	1 byte	03h	End of packet.

Note: If the host sends a packet whose LNH/LNL is 0 or greater than 256, the subsequent operations are not guaranteed.

5.1.3 Data Packet

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

Table 11. Data Packet Format

Symbol	Size	Value	Description
SOD	1 byte	81h	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. Examples: - For Write data transmission: Write data. - For Status transmission: Status code (STS), Status details (ST2) and Failure address (ADR).
SUM	1 byte	-	Sum data of "LNH + LNL + RES + Data" (expressed as two's complement) For example: LNH + LNL + RES + Data(1) + Data(2) + ... + Data(n) + SUM = 00h.
ETX	1 byte	03h	End of packet.

Notes:

1. 1040byte is the maximum size among all boot firmware commands.
The Data's length of Data packet for each command is specified by the command's specification individually. See the command's specification for details.

If the host sends a packet whose LNH/LNL is 0 or greater than 1041, the subsequent operations are not guaranteed.

5.1.4 CMD: Command Code

Table 12. Command Codes

Value	Name	Description
71h	DLM state transit command	Authentication-free DLM transition.
2Ch	DLM state request command	Get the current DLM state.
72h	Protection level transit command	Protection level transition.
73h	Protection level request command	Get the Protection level.
75h	Authentication level request command	Get the Authentication level.
30h	Authentication command	Authentication-required DLM and AL transition.
28h	Key setting command	Insert the key for authentication of DLM/AL transition
2Ah	User key setting command	Insert the user custom key.
29h	Key verify command	Verify the key for authentication of DLM/AL transition
2Bh	User key verify command	Verify the user custom key.
50h	Initialize command	Initialize all the memory areas.
4Eh	Boundary setting command	Set the boundary.
4Fh	Boundary request command	Get the boundary setting.

Value	Name	Description
51h	Parameter setting command	Set the parameter.
52h	Parameter request command	Get the parameter setting.
00h	Inquiry command	Return ACK.
3Ah	Signature request command	Get the signature information.
3Bh	Area information request command	Get the area information.
34h	Baudrate setting command	Set baudrate for UART communication.
36h	Endian notice command	Specify the endian of the data that to be written or read by boot firmware commands
13h	Write command	Write data to target area.
15h	Read command	Read data from target area
18h	CRC command	Cyclic Redundancy Check of target area.
26h	Code certificate update command	Update the code certificate.
27h	Code certificate check command	Check the code certificate.
1Ah	Encrypted data write command	Write encrypted data to target area.

5.1.5 RES: Response Code

Table 13. Response Codes

Value	Name	Description
00h CMD	OK (ongoing normally)	This is returned when the status code is OK (00h).
80h CMD	ERR (occurrence of an error)	This is returned when the status code is not OK (00h).

5.1.6 STS: Status Code

Table 14. Status Codes

Value	Name	Description
00h	Communication is normal [OK]	-
C0h	Unsupported command error	Received an unsupported command.
C1h	Packet error	Abnormality of packet format.
C2h	Checksum error	Abnormality of packet's checksum value.
D0h	Parameter error	Abnormality of packet parameter.
D2h	Invalid address error	Invalid address in the current boundary settings.
D3h	Certificate storage error	Certificate storage area is invalid.
D5h	Command acceptance error	The command cannot be executed in the current state.
D6h	DLM state unmatched error	Device reset is not asserted after DLM state is changed.
D7h	Hardware error	Abnormality of memory value.
DAh	Protection error	Accessing protected areas or performing prohibited actions.
DBh	Trusted system error	Abnormality from the Trusted system.
DCh	Boot loader version error	Abnormality of the program version indicated by Code certificate.
E4h	Secure error	Access to an area which is inaccessible with current privilege.
E8h	Verify error	Verification of the written data fails.
EDh	MRAM sequencer error	(*1), (*2), MACI command abnormally terminates.
EEh	MRAM write error	(*1), MRAM write without MACI commands abnormally terminates.

Notes:

1. Boot firmware also returns additional error information as below:
MRAM sequencer error: Status details (ST2) and Failure address (ADR)
MRAM Write error : Failure address (ADR)
2. This error occurs when MACI becomes "command lock" state after execution MRAM sequencer command.

5.1.7 ST2: Status Details

Table 15. Status Details

Value	Name	Description
MSTATR [31:0]	MRAM status	When a MRAM sequencer error occurs, boot firmware returns the value of the MSTATR register. When not, boot firmware returns FFFFFFFFh. Boot firmware clears the MSTATR register after the status sending, so even when an error occurs, the host can retry the next command without reset release.
AAAA0000h – AAAAFFFFh	Trusted system status	When a Trusted system error occurs, boot firmware returns the following detailed information: • AAAA0100h: An invalid magic number is set. • AAAA0101h: Unsupported version is set. • AAAA0102h: Out of range TLV Length is set. • AAAA0103h: Missing required TLV field. • AAAA0104h: The length exceeding the end of the manifest is specified in Length of the TLV field. • AAAA0105h: An invalid image length is set. • AAAA0106h: There is a wrong combination of signature algorithms. • AAAA0107h : An algorithm was specified that the manifest does not support • AAAA0200h: Cryptographic processing failure. • AAAA0201h: Verification failed. • AAAA0202h: Unsupported algorithm. • AAAA0204h: Parameter error. • AAAA0300h: CRC mismatch

5.1.8 ADR: Failure Address

Table 16. Failure Address

Value	Name	Description
00000000h–FFFFFFFFh	Failure address	When a MRAM sequencer error, or a MRAM write error occurs, boot firmware returns the error occurrence address. When not, boot firmware returns FFFFFFFFh.

5.1.9 DLM: Device Lifecycle Management State Code**Table 17. DLM State Codes**

Value	Name	Description
04h	OEM	Original Equipment Manufacturer
06h	LCK_BOOT	LoCKed BOOT interface
07h	RMA_REQ	Return Material Authorization REQuest
08h	RMA_ACK	Return Material Authorization ACKnowledged
09h	RMA_RET	Return Material Authorization RETurn

6. Command List**Table 18. Command List**

Name	Communication Method	DLM State			Prerequisite Command
		OEM	LCK_BOOT	RMA_REQ	
DLM State Transit Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
DLM State Request Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Protection Level Transit Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
Protection Level Request Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Authentication Level Request Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Authentication Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Key Setting Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
User Key Setting Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
Key Verify Command	2-wire UART, USB,	⊙	(*1)	⊙	➤ Endian notice command

	JTAG/SWD, FINE				
User Key Verify Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Initialize Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
Boundary Setting Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
Boundary Request Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Parameter Setting Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
Parameter Request Command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Inquiry command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Signature request command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	-
Area information request command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	-
Baudrate setting command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	-
Endian notice command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	-
Write command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command

Read command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
CRC command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Code certificate update command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command
Code certificate check command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)	⊙	➤ Endian notice command
Encrypted data write command	2-wire UART, USB, JTAG/SWD, FINE	⊙	(*1)		➤ Endian notice command

⊙ : Command is available in the state. (If an unavailable command is sent, boot firmware returns "Command acceptance error".)

*1: No commands are available in LCK_BOOT because boot firmware does not operate in this state.

6.1 Device Lifecycle Management

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

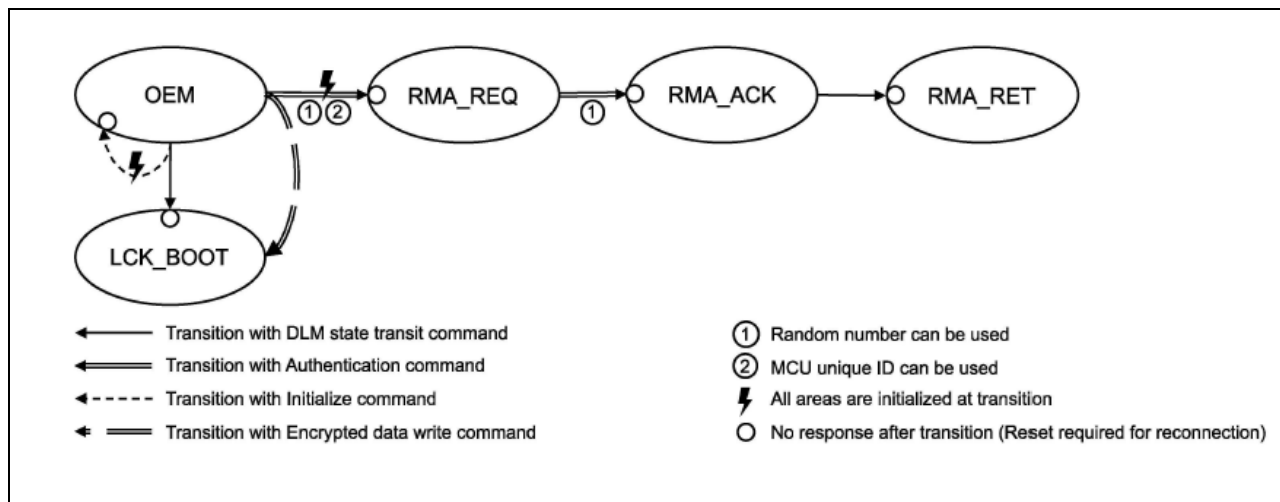


Figure 14. DLM State Transitions

6.2 DLM State Transit Command

This command transitions the DLM state without authentication.

Boot firmware will enter an infinite loop when the DLM state transitions to LCK_BOOT or RMA_RET.

This command requires adherence to conditions described in Command List.

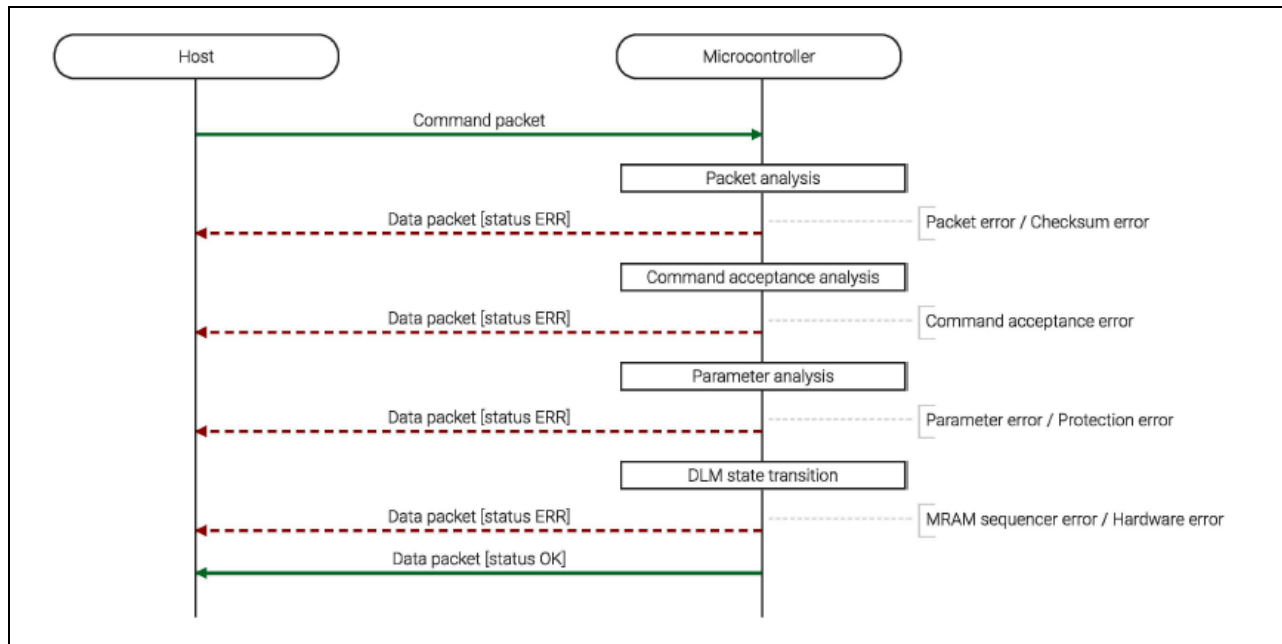


Figure 15. DLM State Transit Command Sequence Diagram

6.2.1 Packets

6.2.1.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	03h
CMD	(1 byte)	71h (DLM state transit command)
SDLM	(1 byte)	Source DLM state code: 04h: OEM 08h: RMA_ACK
DDLML	(1 byte)	Destination DLM state code: 06h: LCK_BOOT 09h: RMA_RET
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.2.1.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	71h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	8Dh
ETX	(1 byte)	03h

6.2.1.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	F1h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.2.2 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

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

- When SDLM is different from the current DLM state, boot firmware returns "Parameter error".
- When DDLM is a DLM state that cannot be entered from the current DLM state without authentication, boot firmware returns "Parameter error".
- If LCK_BOOT is specified for DDLM while the transition to LCK_BOOT is disabled, boot firmware returns "Protection error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware transitions to the DLM state.

- If an error occurs during DLM state transitioning, boot firmware returns "MRAM sequencer error" and waits for the next command.
* Check the DLM state after the error has 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 or RMA_RET, the boot firmware sends "OK" and does not respond.

- When DLM state transit successful completion, "OK" is returned and the boot firmware waits for the next command.

6.2.3 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Source DLM state code is different from the current DLM state.	Parameter error	FFFFFFFFh	FFFFFFFFh
Destination DLM state code is not a transitionable DLM state.	Parameter error	FFFFFFFFh	FFFFFFFFh
LCK_BOOT was specified for the Destination DLM state code with the transition to LCK_BOOT disabled.	Protection error	FFFFFFFFh	FFFFFFFFh
MACI detected an error after the command execution in an undisclosed area.	MRAM sequencer error	MRAM status	FFFFFFFFh
DLM state is abnormal.	Hardware error	FFFFFFFFh	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.2.4 DLM State Transition

Figure 16 shows the DLM states that can be transitioned by the DLM State Transit command.

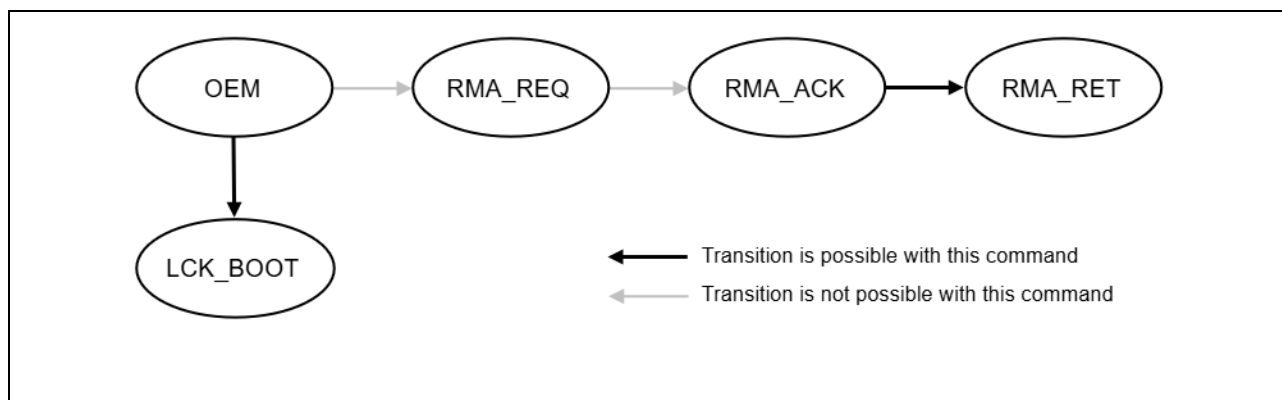


Figure 16. Valid State Transitions for DLM State Transit Command

Source DLM	Destination DLM	Requirements for transition
OEM	LCK_BOOT	Transition to LCK_BOOT (Parameter ID: 02h) is enabled.
RMA_ACK	RMA_RET	-

6.3 DLM State Request Command

This command is used to get the current DLM state.

This command requires adherence to conditions described in Command List.

6.3.1 Sequence Diagram

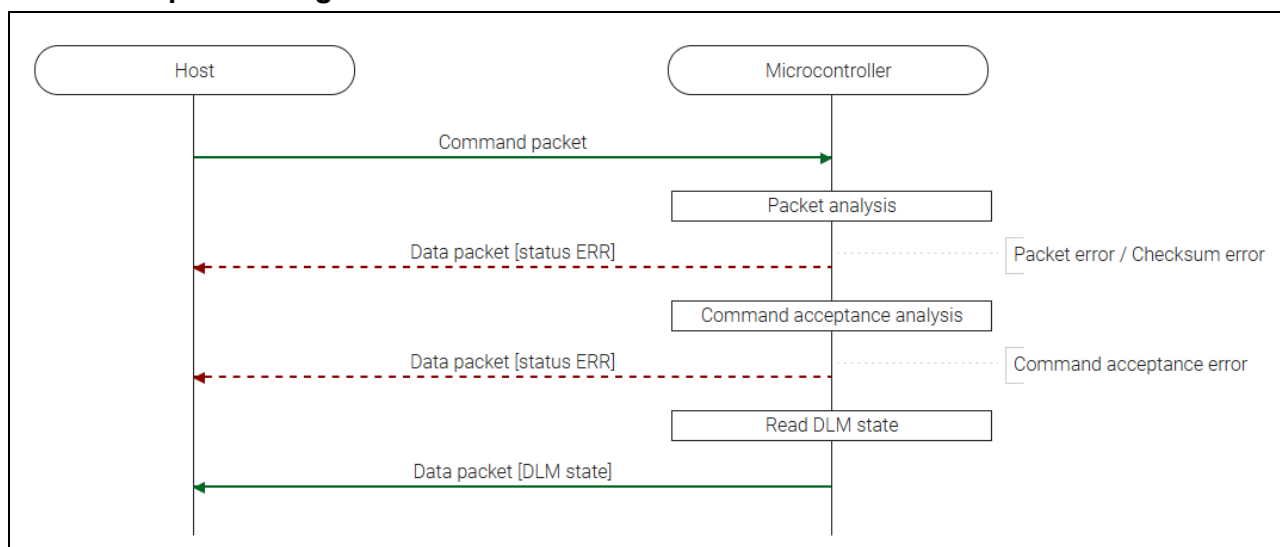


Figure 17. DLM State Request Command Sequence Diagram

6.3.2 Packets

6.3.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
CMD	(1 byte)	2Ch (DLM state request command)
SUM	(1 byte)	D3h
ETX	(1 byte)	03h

6.3.2.2 Data Packet [DLM State]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	2Ch (OK)
DLM	(1 byte)	DLM state code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.3.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	ACh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.3.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, 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.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware returns the current DLM state:

- Send DLM state and return to command wait state.
* Memory contents do not change before command reception.

6.3.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.4 Protection Level Transit Command

This command transitions to the Protection level.

This command requires adherence to conditions described in Command List.

6.4.1 Sequence Diagram

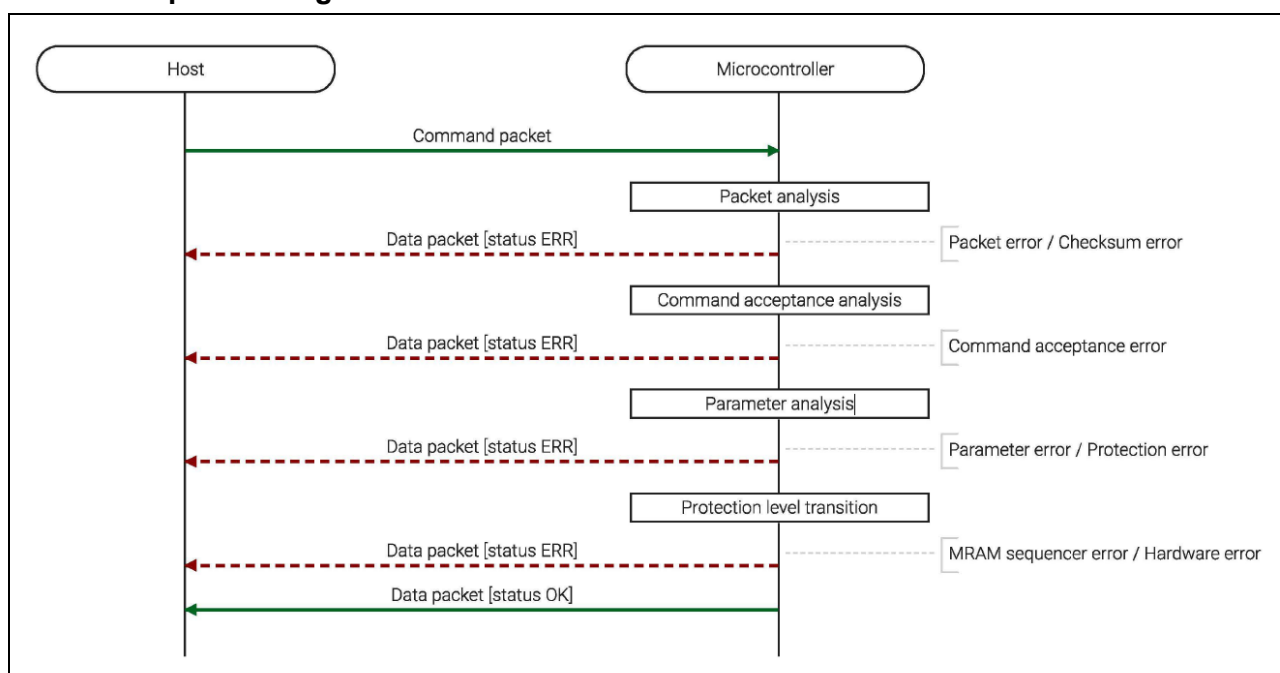


Figure 18. Protection Level Transit Command Sequence Diagram

6.4.2 Packets

6.4.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	03h
CMD	(1 byte)	72h (Protection level transit command)
SPL	(1 byte)	Source PL code: • 02h: Protection level 2 • 03h: Protection level 1 • 04h: Protection level 0
DPL	(1 byte)	Destination PL code: • 02h: Protection level 2 • 03h: Protection level 1 • 04h: Protection level 0
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.4.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	72h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	8Ch
ETX	(1 byte)	03h

6.4.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	F2h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.4.2.4 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

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

- When SPL is different from the current Protection level, boot firmware returns "Parameter error".
- When DPL is a Protection level that cannot be transitioned to from the current Protection level, boot firmware returns "Parameter error".
- If it is not allowed to transit to the specified DPL in the current Authentication level, boot firmware returns "Protection error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware transitions the Protection level:

- If an error occurs during transition of the Protection level, boot firmware returns "MRAM sequencer error" and waits for the next command.
* Check the DLM state after the error has occurred with the DLM state request command.
- When the Protection level transit successfully completes, "OK" is returned and the boot firmware waits for the next command.

6.4.3 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Source PL code is different from current Protection level.	Parameter error	FFFFFFFFh	FFFFFFFFh
Destination PL code is not transitional Protection level.	Parameter error	FFFFFFFFh	FFFFFFFFh
Does not meet the Authentication level required for transition to the specified destination PL code.	Protection error	FFFFFFFFh	FFFFFFFFh
MACI detected an error after the command execution in an undisclosed area.	MRAM sequencer error	MRAM status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.4.4 Protection Level Transition

Figure 19 shows the Protection level that can be transited by this command.

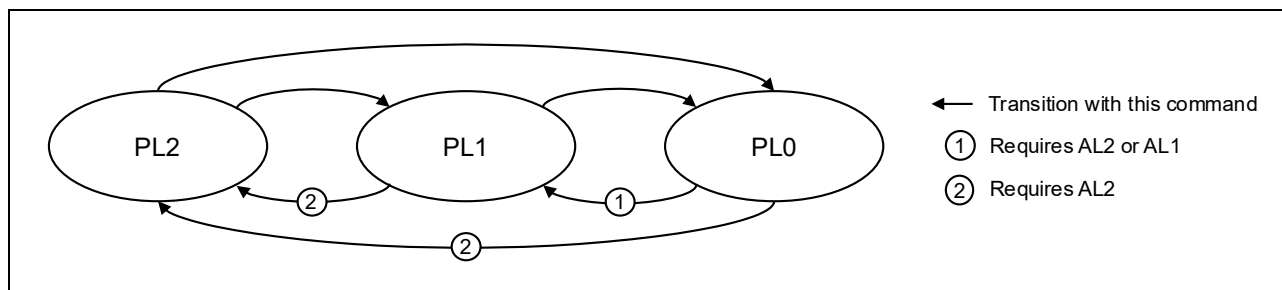


Figure 19. Valid Protection Level Transitions

Source PL	Destination PL	Current Authentication level		
		AL2	AL1	AL0
PL0	PL1	OK	OK	Protection error
	PL2		Protection error	
PL1	PL0		OK	N/A (Impossible combination)
	PL2		Protection error	
PL2	PL0		N/A (Impossible combination)	
	PL1			

6.5 Protection Level Request Command

This command is used to get the current Protection level.

This command requires adherence to conditions described in Command List.

6.5.1 Sequence Diagram

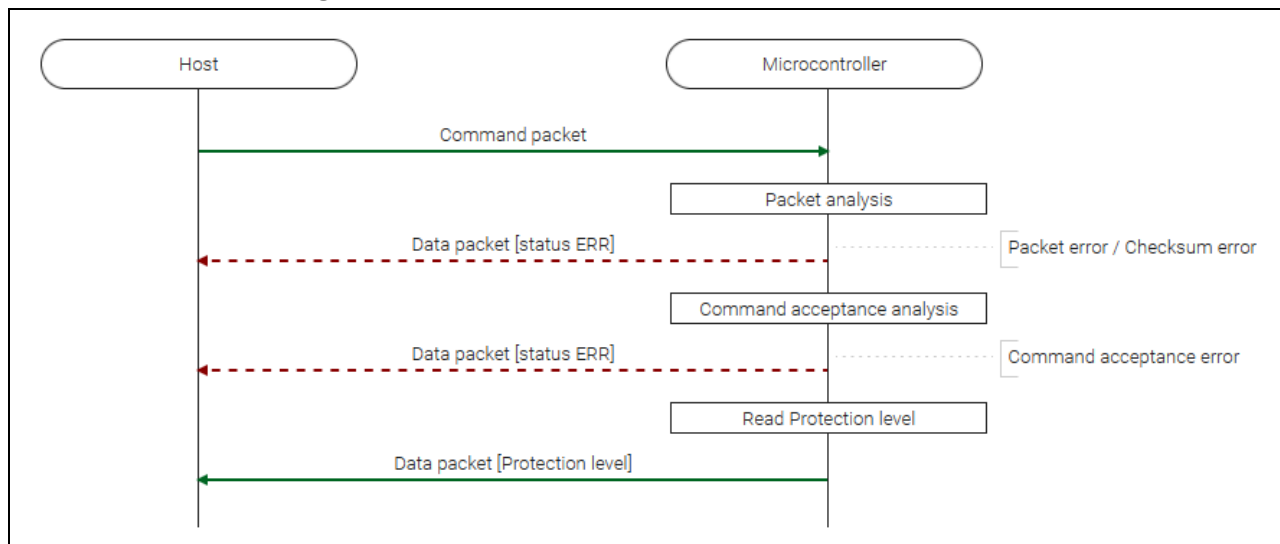


Figure 20. Protection Level Request Sequence Diagram

6.5.2 Packets

6.5.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
CMD	(1 byte)	73h (Protection level request command)
SUM	(1 byte)	8Ch
ETX	(1 byte)	03h

6.5.2.2 Data Packet [Protection Level]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	73h (OK)
CPL	(1 byte)	Current PL code • 02h: Protection level 2 • 03h: Protection level 1 • 04h: Protection level 0
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.5.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	F3h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.5.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If device reset is not asserted after the Encrypted data write command execution, 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.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware returns Protection level:

- Send Protection level and return to command wait state.
* Memory contents do not change before command reception.

6.5.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.6 Authentication Level Request Command

This command is used to get the current Authentication level.

This command requires adherence to conditions described in Command List.

6.6.1 Sequence Diagram

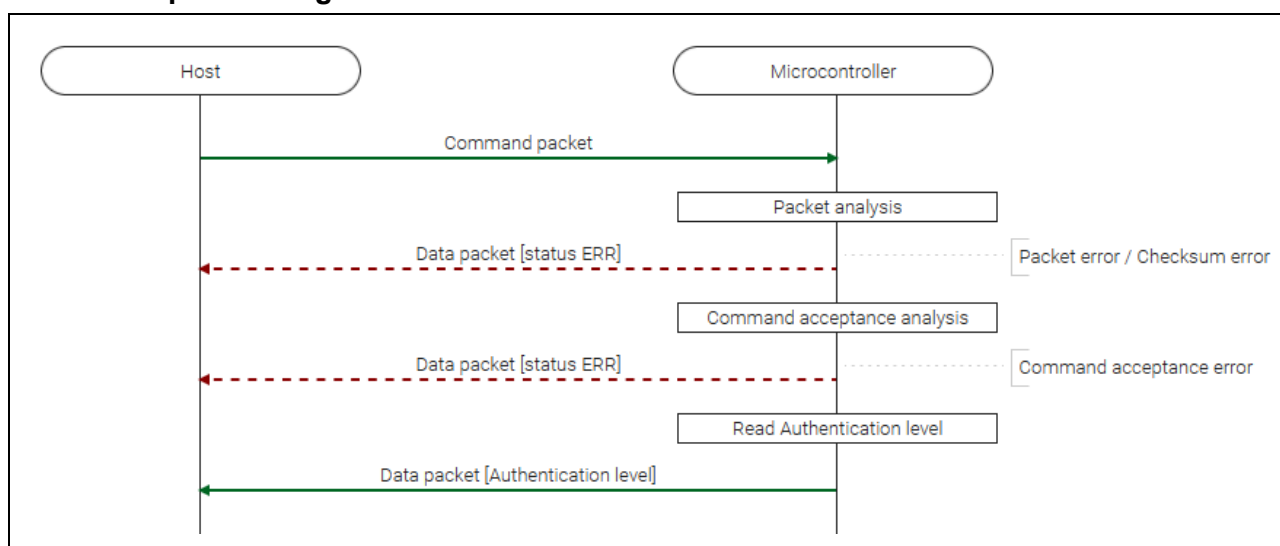


Figure 21. Authentication Level Request Command Sequence Diagram

6.6.2 Packets

6.6.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
CMD	(1 byte)	75h (Authentication level request command)
SUM	(1 byte)	8Ah
ETX	(1 byte)	03h

6.6.2.2 Data Packet [Authentication Level]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	75h (OK)
CAL	(1 byte)	Current AL code • 02h: Authentication level 2 • 03h: Authentication level 1 • 04h: Authentication level 0
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.6.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	F5h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.6.3 Processing procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, 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.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware returns Authentication level:

- Send Authentication level and return to the command wait state.
* Memory contents do not change before command reception.

6.6.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.7 Authentication Command

This command authenticates using a key and transitions the DLM state or the Authentication level.

Authentication is executed by the challenge and response method or Unique ID.

When transiting to RMA_REQ, Boot firmware initializes the MRAM before executing DLM state transition. Initialization process at this time is not affected by the block protection settings.

This command requires adherence to conditions described in Command List.

6.7.1 Sequence Diagram

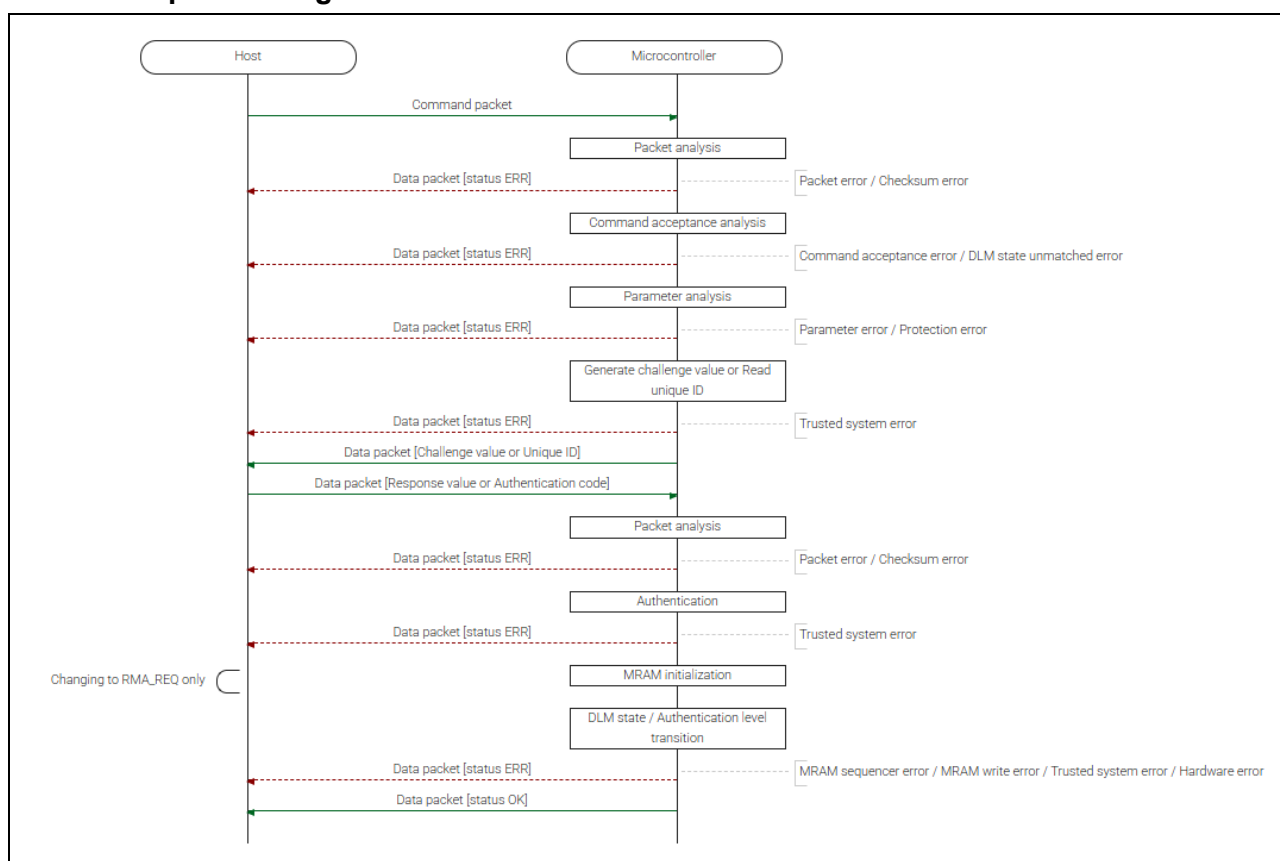


Figure 22. Authentication Command Sequence Diagram

6.7.2 Packets**6.7.2.1 Command Packet**

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	04h
CMD	(1 byte)	30h (Authentication command)
SDLM	(1 byte)	Source DLM/AL code. For DLM transitions: • 04h: OEM • 07h: RMA_REQ For AL transitions: • 03h: AL1 • 04h: AL0
DDLm	(1 byte)	Destination DLM/AL code. For DLM transitions: • 07h: RMA_REQ • 08h: RMA_ACK For AL transitions: • 02h: AL2 • 03h: AL1
CHCT	(1 byte)	Authentication type: • 00h: Random number (Can be used all transit cases.) • 01h: MCU unique ID (Can be used only transit to RMA_REQ.)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.7.2.2 Data Packet [Challenge Value or Unique ID]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	11h
RES	(1 byte)	30h (OK)
CHCD	(16 bytes)	Challenge value or Unique ID For example: 01234567_89AB ... 2233_44556677h -> 01h, 23h, 45h, ... , 55h, 66h, 77h
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.7.2.3 Data Packet [Response Value or Authentication Code]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	21h
RES	(1 byte)	30h (OK)
MAC	(32 bytes)	Response value or Authentication code For example: 01234567_89AB ... 2233_44556677h -> 01h, 23h, 45h, ..., 55h, 66h, 77h For more detail on Response value, please refer to 6.7.8 Response Value Calculation .
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.7.2.4 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	30h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	CEh
ETX	(1 byte)	03h

6.7.2.5 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	B0h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.7.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If HUK has been zeroized, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- If the currently active DLM state does not match the stored DLM state, boot firmware sends "DLM state unmatched error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

* Memory contents do not change before command reception.

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

- When SDLM is different from the current DLM state or Authentication level, boot firmware returns "Parameter error".
- When SDLM and DDLM are not a transitional combination, boot firmware returns "Parameter error".
- When any of the following conditions is met, boot firmware returns "Protection error":
 - Authentication with AL2_KEY is disabled and DDLM is RMA_REQ.
 - Authentication with AL2_KEY is disabled and DDLM is AL2.
 - Authentication with AL1_KEY is disabled and DDLM is AL1.
- When CHCT is not a challenge type that can be used to transition the DLM state, boot firmware returns "Parameter error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory contents do not change before command reception.

When the processing above is 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 does not respond.
 - * Memory contents do 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.
 - * Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware receives and analyzes a 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.
- 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 with the specifications of this command, "Packet error" is returned.
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory contents do not change before command reception.

When the processing above is 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 does not respond.
 - * Memory contents do not change before command reception.
- When authentication fails, "Trusted system error" is returned and the boot firmware waits for the next command.
 - * Memory contents do not change before command reception.

When authentication is successfully completed and the DLM state transits to RMA_REQ, boot firmware initializes the memory.

* This command initializes the memory even if initialization is disabled. (refer to the Parameter request command).

- If an error occurs during initialization, the boot firmware sends a “MRAM sequencer error” or “MRAM write error” and returns to the command wait state. Also, if the Trusted system becomes abnormal after initialization of the Trusted system, the boot firmware returns nothing and no response.

When the Authentication is successfully completed (for transition to RMA_REQ, erase of memory is also successful), boot firmware executes transition:

- If the Trusted system becomes abnormal during transition, the boot firmware returns nothing and does not respond.
 - * Check the DLM state after the error has occurred with the DLM state request command.
- If an error occurs during transition, boot firmware returns “MRAM sequencer error” or “Trusted system error” and waits for the next command.
 - * Check the DLM state after the error has 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 above error does not occur, the boot firmware sends “OK” and becomes unresponsive (DLM transition) or waits for the next command (Authentication level transition).
 - * When the DLM state transitions to RMA_REQ, each area of the memory is in the following state:
 - User area is all-FFh except for the following:
 - Blocks for which 0 is set for permanent block protection setting.
 - * Not affected by block protection settings.
 - User Extra MRAM is all-FFh except following cases:
 - Protection for User Extra MRAM is enabled by Parameter setting command, and also cannot be unset. (refer to the Parameter setting command).
 - User Extra MRAM is changed to OTP by memory type select.
 - The Config area is written the value at shipment, except for the following:
 - Area protected by MOFSPSn_S, MOFSPS0_NS, SOFSPSn_S or SOFSPS0_NS.
 - Area changed to OTP by memory type select.
 - Block protection setting for blocks in which 0 is set for permanent block protection setting.

6.7.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The currently active DLM state does not match the stored DLM state.	DLM state unmatched error	FFFFFFFFh	FFFFFFFFh
SDLM is different from current DLM state or AL.	Parameter error	FFFFFFFFh	FFFFFFFFh
SDLM and DDLM are not a transitionable combination.	Parameter error	FFFFFFFFh	FFFFFFFFh
AL2 or RMA_REQ specified for DDLM with AL2_KEY disabled.	Protection error	FFFFFFFFh	FFFFFFFFh
AL1 specified for DDLM with AL1_KEY disabled.	Protection error	FFFFFFFFh	FFFFFFFFh
Authentication type is different from the value specified by this command.	Parameter error	FFFFFFFFh	FFFFFFFFh
Challenge value/Unique ID generation failed.	Trusted system error	FFFFFFFFh	FFFFFFFFh
The response code of the received data packet is different from the value specified by this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Authentication failed.	Trusted system error	FFFFFFFFh	FFFFFFFFh
An error detected when MRAM programming which does not use MACI command.	MRAM write error	FFFFFFFFh	Failure address
MACI detected an error after the command execution in the disclosed area.	MRAM sequencer error	MRAM status	Failure address
MACI detected an error after the command execution in an undisclosed area.	MRAM sequencer error	MRAM status	FFFFFFFFh
DLM state is abnormal.	Hardware error	FFFFFFFFh	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.7.5 Precautions

- (1) This device has 4 storage areas each for authentication keys which are to transit to AL2, AL1 or RMA_REQ (AL2_KEYn, AL1_KEYn and RMA_KEYn where n=0~3).

Boot firmware searches a storage area that satisfies the both of followings from the 4 areas:

- Authentication key is stored in the area
- The stored authentication key is not revoked

Boot firmware performs this search in order from n=3, n=2, n=1 to n=0, and use the authentication key stored in the firstly found area for the authentication.

Therefore, for example, when all the 4 keys are stored and also no one is revoked, only the key n=3 can be used for authentication.

To change the used authentication key to n=2 from this state, enable the revoke setting of key n=3 mapped in Config area.

6.7.6 DLM State Transition

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

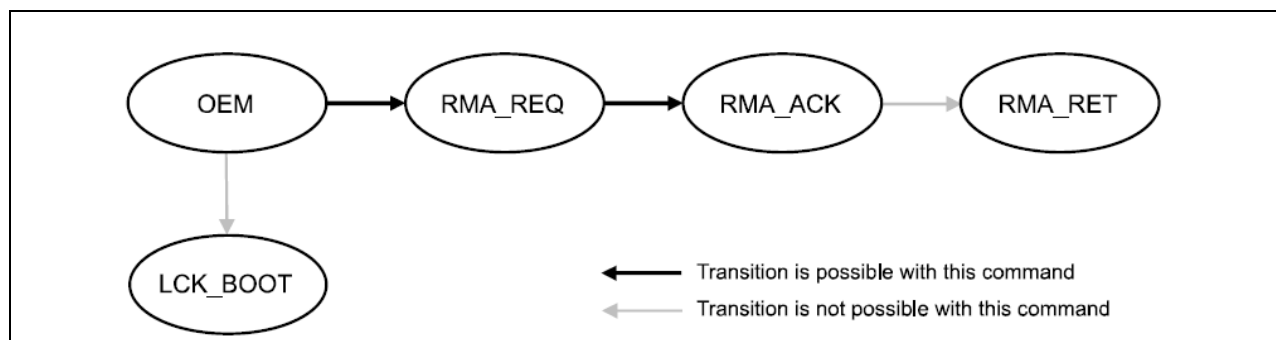


Figure 23. DLM State Transition

Source DLM state	Destination DLM state	Required key	Requirements for transition
OEM	RMA_REQ	RMA_KEYn (n=0~3)	Authentication using AL2_KEY (Parameter ID: 03h) is enabled.
RMA_REQ	RMA_ACK	RMA_ACK_KEY	

6.7.7 Authentication Level Transition

Figure 24 shows the Authentication level that can be transition by this command.

(Authentication level transition is possible only when DLM state is "OEM".)

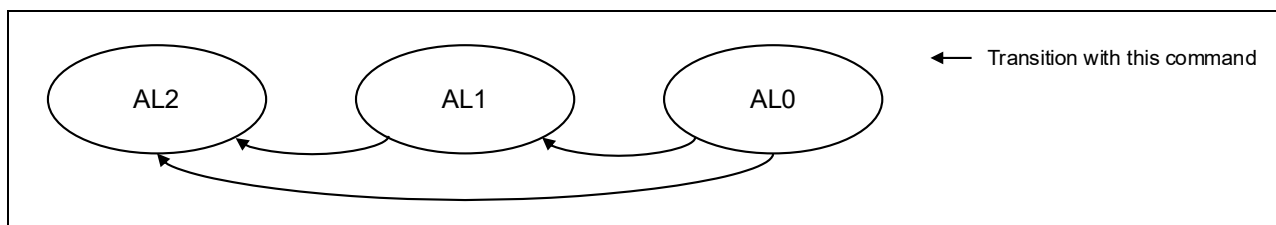


Figure 24. Valid Authentication Level Transitions

Source AL	Destination AL	Required key	Requirements for transition
AL0	AL1	AL1_KEYn (n=0~3)	Authentication using AL1_KEY (Parameter ID: 04h) is enabled.
	AL2	AL2_KEYn (n=0~3)	Authentication using AL2_KEY (Parameter ID: 03h) is enabled.
AL1	AL2	AL2_KEYn (n=0~3)	Authentication using AL2_KEY (Parameter ID: 03h) is enabled.

6.7.8 Response Value Calculation

Response = AES -128 CMAC (Key, 128-bit challenge)

*Fill "1" to the lower 16 bytes of the MAC on the Data Packet because the calculated Response is 16 bytes.

6.8 Key Setting Command

This command generates Wrapped DLM/AL key or Hash of OEM root public key depending on the specified key type, and stores the generated value to device. Which key type can be specified depends on the current AL.

This command requires adherence to the conditions described in Command List.

6.8.1 Sequence Diagram

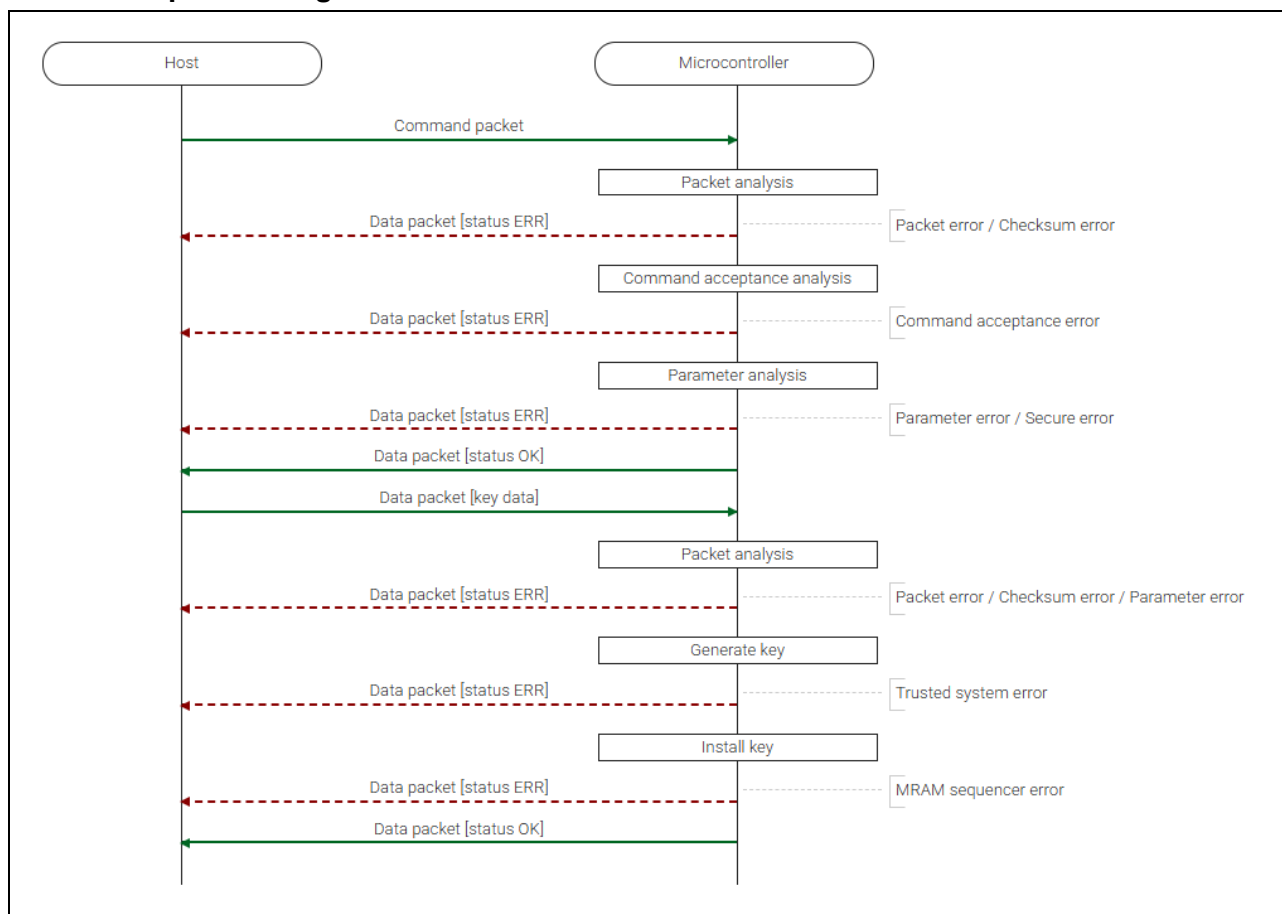


Figure 25. Key Setting Command Sequence Diagram

6.8.2 Packets**6.8.2.1 Command Packet**

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	07h
CMD	(1 byte)	28h (Key setting command)
KYTY	(1 byte)	Key type: • 01h: AL2_KEY • 02h: AL1_KEY • 03h: RMA_KEY • FDh: OEM_ROOT_PK
KID	(1 byte)	Key storage ID This device has 4 storage areas for each key type. This parameter specifies which area to store the generated value to. 0~3 are specifiable.
KLEN	(2 bytes)	Key data length Length of key data to send with Data packet. Specify the following depending on the KYTY. • KYTY=01h: 00h, 20h (32 bytes) • KYTY=02h: 00h, 20h (32 bytes) • KYTY=03h: 00h, 20h (32 bytes) • KYTY=FDh: 00h, 50h (80 bytes)
RSV	(2 bytes)	0000h (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.8.2.2 Data Packet [Key Data]

SOD	(1 byte)	81h																																																																																																																
LNH	(1 byte)	00h																																																																																																																
LNL	(1 byte)	N + 53																																																																																																																
RES	(1 byte)	28h (OK)																																																																																																																
SKR	(4 bytes)	Shared key ring number. For example: 01234567h -> 01h, 23h, 45h, 67h																																																																																																																
ESKY	(32 bytes)	Wrapped install key (W-UFPK). For example: 01234567_89AB ... 2233_44556677h -> 01h, 23h, 45h, ... , 55h, 66h, 77h																																																																																																																
IVEC	(16 bytes)	Initialization Vector. For example: 01234567_89AB ... 2233_44556677h -> 01h, 23h, 45h, ... , 55h, 66h, 77h																																																																																																																
ENKY	(N bytes (*1))	Key data (Encrypted key MAC). Encrypted key + MAC For example: If the install data is as follows, the host should send ENKY in the order shown in the lower table. Install data: <table><tr><th colspan="8">Encrypted key</th></tr><tr><td>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td></tr><tr><td>08</td><td>09</td><td>0A</td><td>0B</td><td>0C</td><td>0D</td><td>0E</td><td>0F</td></tr><tr><th colspan="8">MAC</th></tr><tr><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td></tr><tr><td>18</td><td>19</td><td>1A</td><td>1B</td><td>1C</td><td>1D</td><td>1E</td><td>1F</td></tr></table> Order of sending ENKY: <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>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</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>08</td><td>09</td><td>0A</td><td>0B</td><td>0C</td><td>0D</td><td>0E</td><td>0F</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>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</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>18</td><td>19</td><td>1A</td><td>1B</td><td>1C</td><td>1D</td><td>1E</td><td>1F</td></tr></table>	Encrypted key								00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	MAC								10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	1st	2nd	3rd	4th	5th	6th	7th	8th	00	01	02	03	04	05	06	07	9th	10th	11th	12th	13th	14th	15th	16th	08	09	0A	0B	0C	0D	0E	0F	17th	18th	19th	20th	21st	22nd	23rd	24th	10	11	12	13	14	15	16	17	25th	26th	27th	28th	29th	30th	31st	32nd	18	19	1A	1B	1C	1D	1E	1F
Encrypted key																																																																																																																		
00	01	02	03	04	05	06	07																																																																																																											
08	09	0A	0B	0C	0D	0E	0F																																																																																																											
MAC																																																																																																																		
10	11	12	13	14	15	16	17																																																																																																											
18	19	1A	1B	1C	1D	1E	1F																																																																																																											
1st	2nd	3rd	4th	5th	6th	7th	8th																																																																																																											
00	01	02	03	04	05	06	07																																																																																																											
9th	10th	11th	12th	13th	14th	15th	16th																																																																																																											
08	09	0A	0B	0C	0D	0E	0F																																																																																																											
17th	18th	19th	20th	21st	22nd	23rd	24th																																																																																																											
10	11	12	13	14	15	16	17																																																																																																											
25th	26th	27th	28th	29th	30th	31st	32nd																																																																																																											
18	19	1A	1B	1C	1D	1E	1F																																																																																																											
SUM	(1 byte)	Sum data																																																																																																																
ETX	(1 byte)	03h																																																																																																																

*1) N must be the same value of KLEN specified by the command packet.

6.8.2.3 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	28h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	D6h
ETX	(1 byte)	03h

6.8.2.4 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	A8h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.8.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If HUK has been zeroized, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- When any of the above errors occur, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware executes the parameter analysis:

- When KYTY is an unspecified value, boot firmware returns "Parameter error" and waits for the next command.
* Memory contents do not change before command reception.
- When KID is unspecified value, boot firmware sends "Parameter error" and waits for the next command.
* Memory contents do not change before command reception.
- When combination of KYTY and KLEN does not match, boot firmware sends "Parameter error" and waits for the next command.
* Memory contents do not change before command reception.
- When KYTY cannot be set in the current Authentication level, boot firmware returns "Secure error" and waits for the next command.
* Memory contents do not change before command reception.
- If the above error does not occur, the boot firmware sends "OK".

When the processing above is successfully completed, the boot firmware receives and analyzes the 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 following data until SOD is sent.
- When the received data packet does not have ETX, "Packet error" is returned.
- When the SUM in the received data packet is different from the value calculated by the 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 the defined values, "Packet error" is returned.
- When the length of ENKY in the received data packet does not match KLEN, "Parameter error" is returned.
- When any of the above errors occur, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware generates Key index (Wrapped key) or hash value of OEM_ROOT_PK:

- If the Trusted system becomes abnormal after creating a key index (Wrapped key) or hash value of OEM_ROOT_PK, the boot firmware returns nothing and does not respond.
* Memory contents do not change before command reception.
- If the generation of the Key index (wrapped key) or hash value of OEM_ROOT_PK fails, the boot firmware sends a "Trusted system error" and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware writes Key index (Wrapped key) or hash value of OEM_ROOT_PK to memory:

- If an error occurs while writing, the boot firmware sends a "MRAM sequencer error" and returns to the command wait state.
- When the key index (Wrapped key) or hash value of OEM_ROOT_PK is written successfully completed, boot firmware sends "OK" and returns to command wait state.

6.8.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The specified Key type is an unspecified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Key storage ID is unspecified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Key type and the Key data length does not match.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Key type cannot be inserted at the current Authentication level.	Secure error	FFFFFFFFh	FFFFFFFFh
The response code of the received data packet is different from the value specified by this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Length of ENKY sent by Data packet does not match the value of KLEN specified by Command packet	Parameter error	FFFFFFFFh	FFFFFFFFh
Generation of Key index or hash value failed.	Trusted system error	FFFFFFFFh	FFFFFFFFh
MACI detected an error after the command execution in an undisclosed area.	MRAM sequencer error	MRAM status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.8.5 Precautions

- (1) Written hash value of OEM root public key cannot be rewritten because it is written to OTP area.
Specify a Key storage ID which has not been written yet to change the value of OEM root public key.
- (2) Storage area of DLM/AL key's key index can be switched to OTP area. When the storage area has been switched to OTP area, written key index cannot be rewritten.
Specify a Key storage ID which has not been written yet to change the value of DLM/AL key.

6.8.6 Key type that can be set in each Authentication Level

Table 19 shows the Key types that can be set in each Authentication level.

Table 19. Key Types for each Authentication Level

Authentication level	Key type
AL2	AL2_KEY AL1_KEY RMA_KEY OEM_ROOT_PK
AL1	AL1_KEY

6.9 User Key Setting Command

This command generates Key index (Wrapped key) using Wrapped install 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.

This command requires adherence to conditions described in Command List.

6.9.1 Sequence Diagram

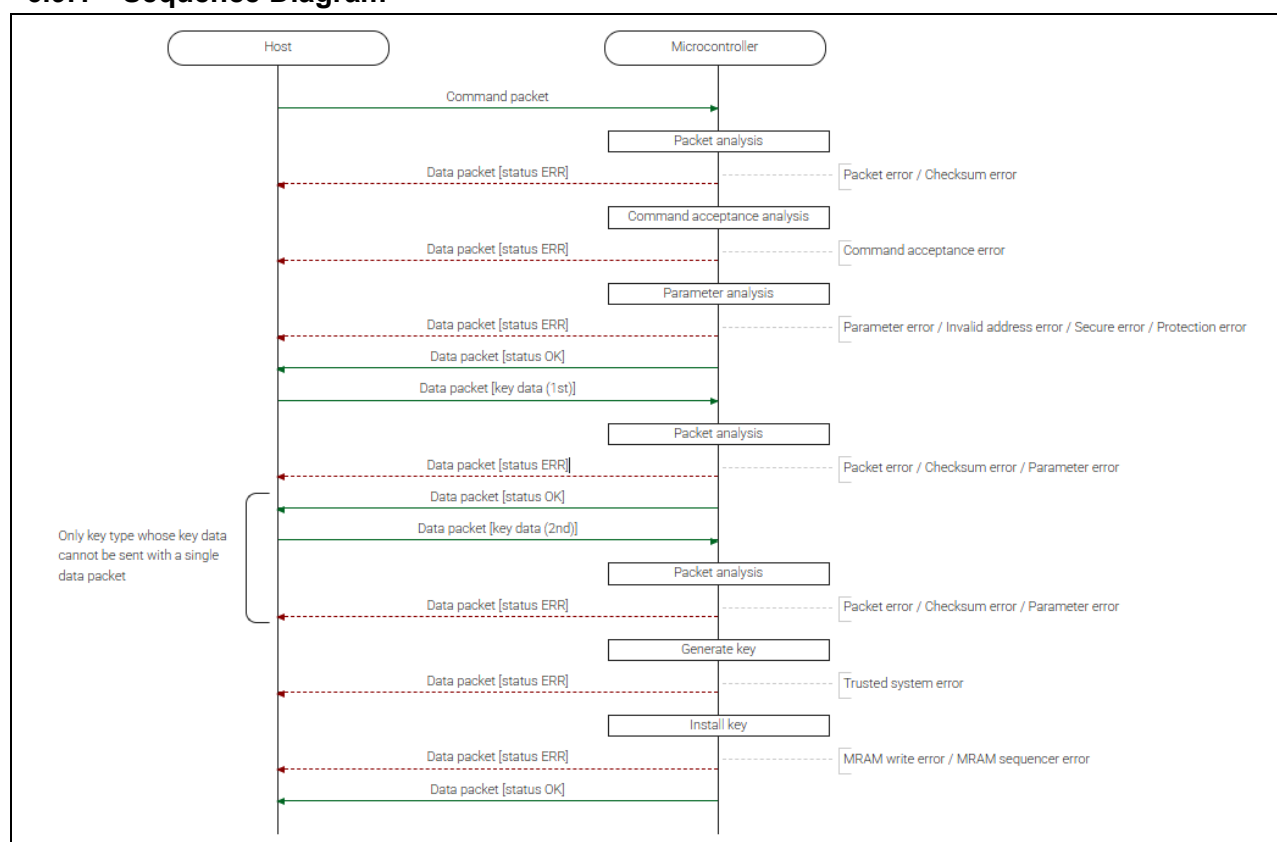


Figure 26. User Key Setting Command Sequence Diagram

6.9.2 Packets

6.9.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	08h
CMD	(1 byte)	2Ah (User key setting command)
KADR	(4 bytes)	Key setting address. For example: 00004000h -> 00h, 00h, 40h, 00h
ENTY	(1 byte)	User key type. Refer to User key list.
KLEN	(2 bytes)	Key data length Total length of key data to send with Data packet. Specify the same value of column "Install data size" in the table "User key list". e.g.) 00h, 20h (32bytes) for AES-128
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.9.2.2 Data Packet [Key Data (1st)]

SOD	(1 byte)	81h																																																																																																																
LNH	(1 byte)	N + 53 (Higher 1 byte)																																																																																																																
LNL	(1 byte)	N + 53 (Lower 1 byte)																																																																																																																
RES	(1 byte)	2Ah (OK)																																																																																																																
SKR	(4 bytes)	Shared key ring number. For example: 01234567h -> 01h, 23h, 45h, 67h																																																																																																																
ESKY	(32 bytes)	Wrapped install key (W-UFPK). For example: 01234567_89AB ... 2233_44556677h -> 01h, 23h, 45h, ... , 55h, 66h, 77h																																																																																																																
IVEC	(16 bytes)	Initialization vector. For example: 01234567_89AB ... 2233_44556677h -> 01h, 23h, 45h, ... , 55h, 66h, 77h																																																																																																																
ENKY	(N bytes(*1))	Install data (Encrypted key MAC) . For example: If the key type is AES-128, the host should send ENKY in the order shown in the lower table. Install data: <table><tr><th colspan="8">Encrypted Key</th></tr><tr><td>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td></tr><tr><td>08</td><td>09</td><td>0A</td><td>0B</td><td>0C</td><td>0D</td><td>0E</td><td>0F</td></tr><tr><th colspan="8">MAC</th></tr><tr><td>F0</td><td>F1</td><td>F2</td><td>F3h</td><td>F4h</td><td>F5h</td><td>F6h</td><td>F7h</td></tr><tr><td>F8</td><td>F9</td><td>FA</td><td>FBh</td><td>FCh</td><td>FDh</td><td>FEh</td><td>FFh</td></tr></table> Order of sending ENKY: <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>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</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>08</td><td>09</td><td>0A</td><td>0B</td><td>0C</td><td>0D</td><td>0E</td><td>0F</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>F0</td><td>F1</td><td>F2h</td><td>F3</td><td>F4</td><td>F5</td><td>F6</td><td>F7</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>F8</td><td>F9</td><td>FA</td><td>FB</td><td>FC</td><td>FD</td><td>FE</td><td>FF</td></tr></table>	Encrypted Key								00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	MAC								F0	F1	F2	F3h	F4h	F5h	F6h	F7h	F8	F9	FA	FBh	FCh	FDh	FEh	FFh	1st	2nd	3rd	4th	5th	6th	7th	8th	00	01	02	03	04	05	06	07	9th	10th	11th	12th	13th	14th	15th	16th	08	09	0A	0B	0C	0D	0E	0F	17th	18th	19th	20th	21st	22nd	23rd	24th	F0	F1	F2h	F3	F4	F5	F6	F7	25th	26th	27th	28th	29th	30th	31st	32nd	F8	F9	FA	FB	FC	FD	FE	FF
Encrypted Key																																																																																																																		
00	01	02	03	04	05	06	07																																																																																																											
08	09	0A	0B	0C	0D	0E	0F																																																																																																											
MAC																																																																																																																		
F0	F1	F2	F3h	F4h	F5h	F6h	F7h																																																																																																											
F8	F9	FA	FBh	FCh	FDh	FEh	FFh																																																																																																											
1st	2nd	3rd	4th	5th	6th	7th	8th																																																																																																											
00	01	02	03	04	05	06	07																																																																																																											
9th	10th	11th	12th	13th	14th	15th	16th																																																																																																											
08	09	0A	0B	0C	0D	0E	0F																																																																																																											
17th	18th	19th	20th	21st	22nd	23rd	24th																																																																																																											
F0	F1	F2h	F3	F4	F5	F6	F7																																																																																																											
25th	26th	27th	28th	29th	30th	31st	32nd																																																																																																											
F8	F9	FA	FB	FC	FD	FE	FF																																																																																																											
SUM	(1 byte)	Sum data																																																																																																																
ETX	(1 byte)	03h																																																																																																																

*1) N depends on the specified ENTY:

Must be 972 when KLEN specified by the command packet when KLEN is greater than 972.

Must be the same value of KLEN specified by the command packet when KLEN is less than or equal to 972.

6.9.2.3 Data Packet [Key Data (2nd)]

SOD	(1 byte)	81h
LNH	(1 byte)	N + 1 (Higher 1 byte)
LNL	(1 byte)	N + 1 (Lower 1 byte)
RES	(1 byte)	2Ah (OK)
ENKY	(N bytes)	Install data (Encrypted key MAC). *Order of sending: Low -> ... -> High
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

N must be the left size to send of ENKY (= KLEN - 972)

6.9.2.4 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	2Ah (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	D4h
ETX	(1 byte)	03h

6.9.2.5 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	AAh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.9.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If HUK has been zeroized, 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.
* Memory contents do not change before command reception.

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

- If ENTY is not specified as Key type, the boot firmware sends a "Parameter error".
- When the specified Key type and Key length do not match, boot firmware sends "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 specified range contains addresses that are inaccessible with the current boundary settings, the boot firmware sends an "Invalid address error".
- If the current Authentication level is AL1 and the specified range includes a secure area, the boot firmware sends a "Secure error".
- If the current Authentication level is AL0, the boot firmware sends a "Secure error".
- If both the following conditions are met, boot firmware sends "Protection error":
 - The area for the key index size from KADR includes area protected by block protection
 - KADR specifies Data area.
- If the area for the key index size from KADR contains a permanent protected block, the boot firmware sends a "Protection error".
- When the KADR specifies User Extra MRAM and write protect is set for User Extra MRAM, boot firmware sends "Protection error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory contents do not change before command reception.
- If the above errors do not occur, the boot firmware sends "OK".

When the processing above is successfully completed, boot firmware receives 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.
- 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 accumulated ENKY data exceeds the Install data size indicated by ENTY in the received data packet, the boot firmware sends a "Parameter error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory contents do 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 does not respond.
 - * Memory contents do not change before command reception.
- If the generation of Key index (Wrapped key) fails, the boot firmware sends a "Trusted system error" and returns to the command waiting state.
 - * Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware writes Key index to the dedicated area:

- If an error occurs while writing, the boot firmware sends a “MRAM write error” and returns to the command wait state.
- If the key index (wrapped key) is successfully saved to the device, the boot firmware sends “OK” and returns to the command wait state.

6.9.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
User key type is not specified as Key type.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Key type and the Key data length does not match.	Parameter error	FFFFFFFFh	FFFFFFFFh
Write area of key index includes an area which is neither User area, Data area nor User Extra MRAM.	Parameter error	FFFFFFFFh	FFFFFFFFh
The area from Key setting address to Key index size spans different types of area.	Parameter error	FFFFFFFFh	FFFFFFFFh
Key setting address is not specified in the WAU for the area.	Parameter error	FFFFFFFFh	FFFFFFFFh
The area from the Key setting address to the Key index size contains addresses that are inaccessible with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh
The current Authentication level is AL1, and the Key setting address contains a Secure region.	Secure error	FFFFFFFFh	FFFFFFFFh
The current Authentication level is AL0.	Secure error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: - Write area of key index includes area protected by block protection - Write area of key index specifies Data area	Protection error	FFFFFFFFh	FFFFFFFFh
Write area of key index includes a permanent protected block.	Protection error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: - The write protection for User Extra MRAM is set	Protection error	FFFFFFFFh	FFFFFFFFh

- Key setting address specifies User Extra MRAM			
The response code of the received data packet is different from the value specified by this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Length of ENKY sent by Data packet does not match the specifications of this command	Parameter error	FFFFFFFFh	FFFFFFFFh
Key index (wrapped key) generation failed.	Trusted system error	FFFFFFFFh	FFFFFFFFh
An error detected when MRAM programming which does not use MACI command.	MRAM write error	FFFFFFFFh	Failure address
MACI detected an error after the command execution.	MRAM sequencer error	MRAM status	Failure address
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.9.4.1 User Key List

The list of user keys specified by this command is shown in Table 20.

Table 20. User Key List

Key type	Installation key	Install data size (bytes)	Key index size (bytes)
05h	AES-128	32	36
06h	AES-192	48	52
07h	AES-256	48	52
08h	AES-128 XTS	48	52
09h	AES-256 XTS	80	84
0Ah	RSA-1024 Public key	160	164
0Bh	RSA-1024 Private key	272	276
0Ch	RSA-2048 Public key	288	292
0Dh	RSA-2048 Private key	528	532
0Eh	RSA-3072 Public key	416	420
0Fh	RSA-3072 Private key	784	788
10h	RSA-4096 Public key	544	548
11h	RSA-4096 Private key	1040	1044
12h	ECC P192 Public key	80	84
13h	ECC P192 Private key	48	52
14h	ECC P224 Public key	80	84
15h	ECC P224 Private key	48	52
16h	ECC P256 Public key	80	84
17h	ECC P256 Private key	48	52
18h	ECC P384 Public key	112	116
19h	ECC P384 Private key	64	68
1Ah	HMAC-SHA224	48	52
1Bh	HMAC-SHA256	48	52
1Ch	ECC P256r1 Public Key	80	84
1Dh	ECC P256r1 Private Key	48	52
1Eh	ECC P384r1 Public Key	112	116
1Fh	ECC P384r1 Private Key	64	68
20h	ECC P512r1 Public Key	144	148
21h	ECC P512r1 Private Key	80	84
22h	ECC secp256k1 Public Key	80	84
23h	ECC secp256k1 Private Key	48	52
24h	ECC P521 Public Key	176	180

25h	ECC P521 Private Key	96	100
26h	Ed25519 Public Key	48	52
27h	Ed25519 Private Key	48	52
28h	HMAC-SHA384	64	68
29h	HMAC-SHA512	80	84
2Ah	HMAC-SHA512-224	80	84
2Bh	HMAC-SHA512-256	80	84
2Ch	HMAC-SHA3-224	48	52
2Dh	HMAC-SHA3-256	48	52
2Eh	HMAC-SHA3-384	64	68
2Fh	HMAC-SHA3-512	80	84
30h	ChaCha20-Poly-1305	48	52
31h	Ed448 Public Key	80	84
32h	Ed448 Private Key	80	84
FEh	RSA-2048 Public Key for TLS	288	292
FFh	Key update key	48	52

6.10 Key Verify Command

This command verifies the authentication key that setting to device.

This command requires adherence to conditions described in Command List.

6.10.1 Sequence Diagram

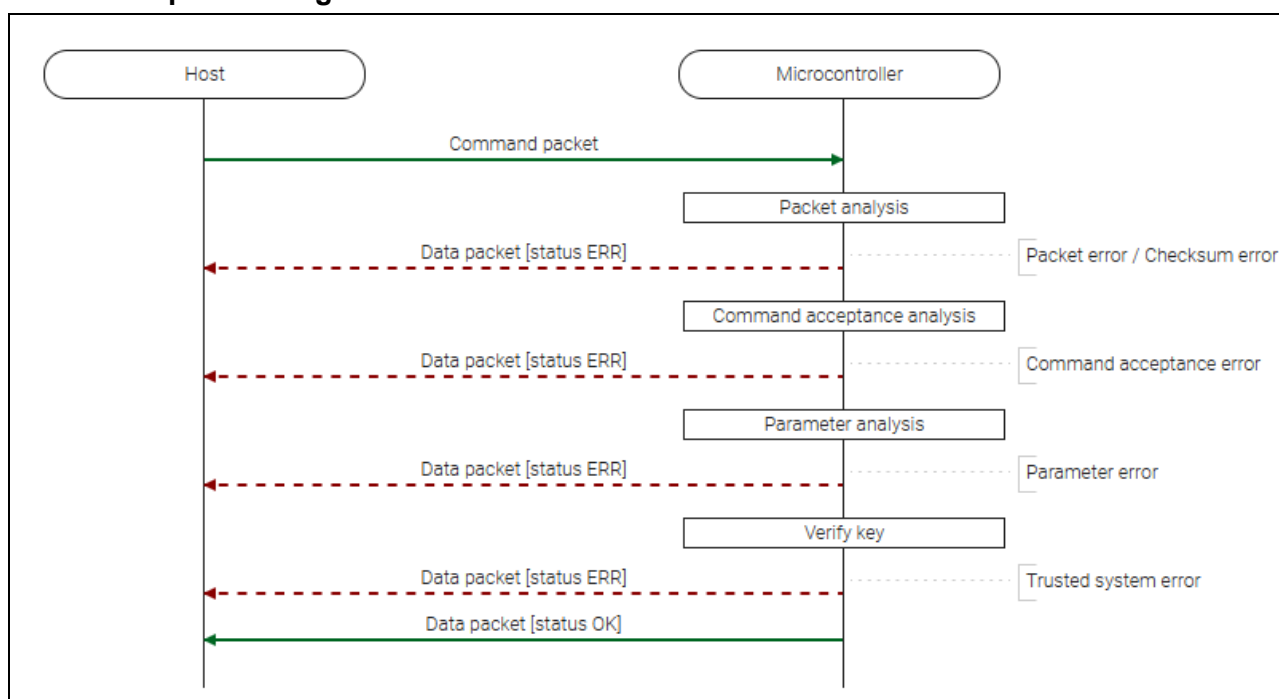


Figure 27. Key Verify Command Sequence Diagram

6.10.2 Packets**6.10.2.1 Command Packet**

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	05h
CMD	(1 byte)	29h (Key verify command)
KYTY	(1 byte)	Key type: • 01h: AL2_KEY • 02h: AL1_KEY • 03h: RMA_KEY • FDh: OEM_ROOT_PK
KID	(1 byte)	Key storage ID This device has 4 storage areas for each key type. This parameter specifies which area to verify the stored data. 0~3 are specifiable.
RSV	(2 bytes)	0000h (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.10.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	29h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	D5h
ETX	(1 byte)	03h

6.10.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	A9h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.10.2.4 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If HUK has been zeroized, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, 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.
 - * Memory contents do not change before command reception.

When the processing above is 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.
 - * Memory contents do not change before command reception.
- If KID is an unsupported key storage ID, boot firmware sends "Parameter error" and returns to command wait state.
 - * Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware verifies Key index (Wrapped key) or hash value of OEM_ROOT_PK..

- If verification of key index (Wrapped key) or hash value of OEM_ROOT_PK fails, the boot firmware sends a "Trusted system error" and returns to the command wait state.
If the Trusted system becomes abnormal during verification of key index (Wrapped key) or hash value of OEM_ROOT_PK, the boot firmware returns nothing and does not respond.
 - * Memory contents do not change before command reception.
- If the verification of the key index (Wrapped key) or hash value of OEM_ROOT_PK is completed successfully, the boot firmware sends "OK" and returns to the command wait state.
 - * Memory contents do not change before command reception.

6.10.3 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Key type is not supported key type.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Key storage ID is unspecified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
Verification of Key index or hash value failed.	Trusted system error	FFFFFFFFh	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.11 User Key Verify Command

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

This command requires adherence to conditions described in Command List.

6.11.1 Sequence Diagram

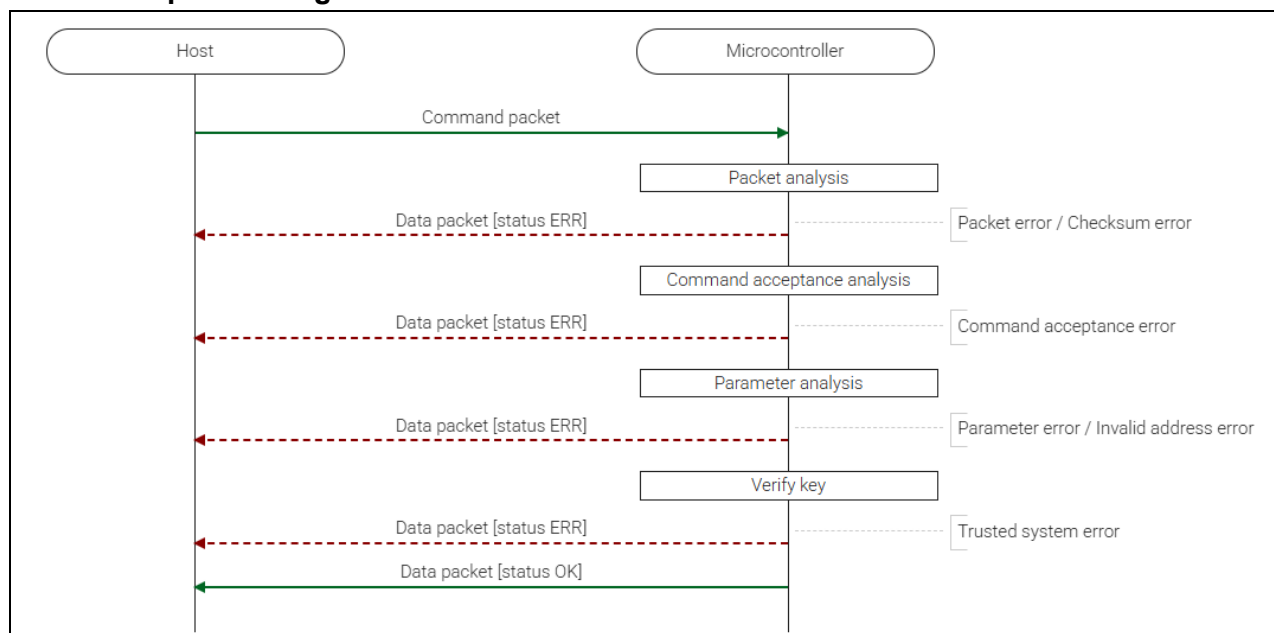


Figure 28. User Key Verify Command Sequence Diagram

6.11.2 Packets

6.11.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	06h
CMD	(1 byte)	2Bh (User key verify command)
KADR	(4 bytes)	Key address. For example: 00004000h -> 00h, 00h, 40h, 00h
ENTY	(1 byte)	User key type. Supports the same key type as User key setting command.
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.11.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	2Bh (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	D3h
ETX	(1 byte)	03h

6.11.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	ABh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.11.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis.

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If HUK has been zeroized, 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.
*Memory contents do not change before command reception.

When the processing above is 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 Extra MRAM 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 specified range contains addresses that are inaccessible with the current boundary settings, the boot firmware sends an "Invalid address error".
- When any of the above errors occur, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware verifies the authentication key:

- When there is a mismatch in the authentication key stored in the device, boot firmware returns "Trusted system error".
If the Trusted system becomes abnormal during key verification, the boot firmware returns nothing and does not respond.
* Memory contents do not change before command reception.
- If the above error does not occur, the boot firmware sends "OK".
* Memory contents do not change before command reception.

6.11.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
User key type is not specified as the Key type.	Parameter error	FFFFFFFFh	FFFFFFFFh
Storage area of key index includes an area which is neither User area, Data area nor User Extra MRAM.	Parameter error	FFFFFFFFh	FFFFFFFFh
The area from the Key address to the Key index size spans different types of area.	Parameter error	FFFFFFFFh	FFFFFFFFh
Key address is not specified in the WAU for the area.	Parameter error	FFFFFFFFh	FFFFFFFFh
The area from the Key address to the Key index size contains addresses that are inaccessible with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh
Key index verify failed.	Trusted system error	FFFFFFFFh	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.12 Initialize Command

This command initializes the following areas and transits the Protection level state to PL2:

- User area
- Data area
- User Extra MRAM
- Config area
- Boundary setting
- Key index (Wrapped key)

Initialization used here means that erasure for erasable areas and writing initial values for non-erasable areas. Initialization processing at this time is not affected by the block protection settings.

This command requires adherence to conditions described in Command List.

6.12.1 Sequence Diagram

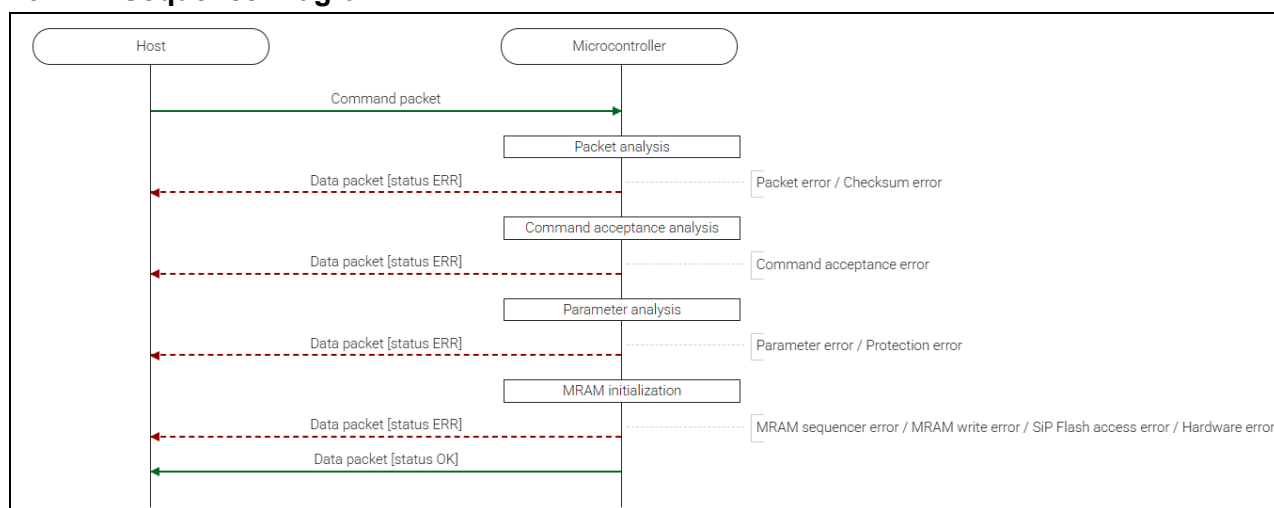


Figure 29. Initialize Command Sequence Diagram

6.12.2 Packets

6.12.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	03h
CMD	(1 byte)	50h (Initialize command)
SDLM	(1 byte)	Source DLM state code: • 04h: OEM
DDLm	(1 byte)	Destination DLM state code: • 04h: OEM
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.12.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	50h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Aeh
ETX	(1 byte)	03h

6.12.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	D0h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.12.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

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

- When SDLM does not match with the current DLM state, "Parameter error" is returned.
- When DDLM is not OEM, "Parameter error" is returned.
- If any of the following conditions are met, boot firmware sends "Protection error":
 - Initialization has been disabled
 - AL2_KEY has been disabled
 - General purpose OTP for user is not All "1"
 - Even one bit in the protection settings below has been set:
 - Permanent block protection
 - MOFSPSn_S
 - MOFSPS0_NS
 - SOFSPSn_S
 - SOFSPS0_NS
 - Even one area in the areas below has been changed to OTP:
 - Storage area for RMA_KEY
 - Storage area for AL2_KEY
 - Storage area for AL1_KEY
 - User Extra MRAM
 - Config area (OTP selectable area) N
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception

When the processing above is successfully completed, boot firmware executes memory initialization:

- If an error occurs during initialization, the boot firmware sends a “MRAM sequencer error” or “MRAM write error” and returns to the command wait state.
- If initialization is completed normally, the boot firmware sends “OK” and does not respond.
 - * The memory is in the following state and the Protection level is PL2:
 - User area : Intialized
 - Data area : Intialized
 - User Extra MRAM : Intialized
 - Config area : Value at shipment
 - Key index : Intialized
 - Boudary setting : Intialized

6.12.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Source DLM state code is different from the current DLM state.	Parameter error	FFFFFFFFh	FFFFFFFFh
Destination DLM state code is not OEM.	Parameter error	FFFFFFFFh	FFFFFFFFh
Any of the following conditions are met: - Initialization has been disabled - AL2_KEY has been disabled - General purpose OTP for user is not All "1" - Even one bit in the protection settings below has been set: - Permanent block protection - SOFSPSn_S - SOFSPS0_NS - MOFSPSn_S - MOFSPS0_NS - Even one area in the areas below has been changed to OTP: - Storage area for RMA_KEY - Storage area for AL2_KEY - Storage area for AL1_KEY - User Extra MRAM - Config area (OTP selectable area) N	Protection error	FFFFFFFFh	FFFFFFFFh
An error detected when MRAM programming which does not use MACI command.	MRAM write error	FFFFFFFFh	Failure address
MACI detected an error after the command execution in disclosed area.	MRAM sequencer error	MRAM status	Failure address
MACI detected an error after the command execution in an undisclosed area.	MRAM sequencer error	MRAM status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.12.5 Precautions

- (1) The parameters, that cannot be unset using Parameter setting command, cannot be unset by this command too.
- (2) Anti-rollback counter area is not initialized by this command.

6.12.6 Protection Level Transition

The transition of Protection level by this command is shown below.

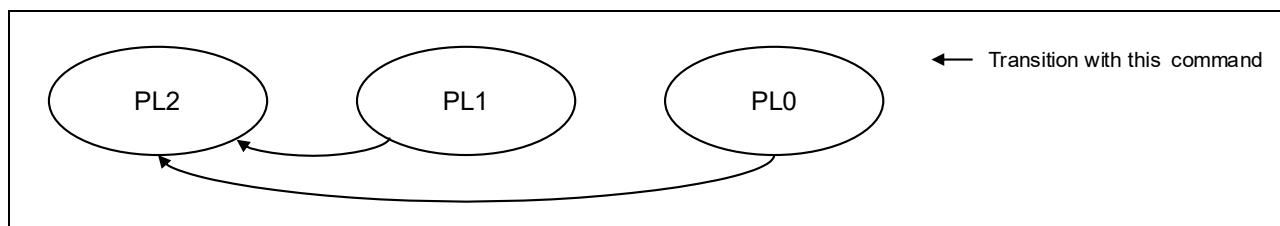


Figure 30. Protection Level Transitions

6.13 Boundary Setting Command

This command receives the boundary setting and stores it in the device.

The accessible addresses of the following areas change depending on the boundary settings:

- User area
- Data area

This command requires adherence to conditions described in Command List.

6.13.1 Sequence Diagram

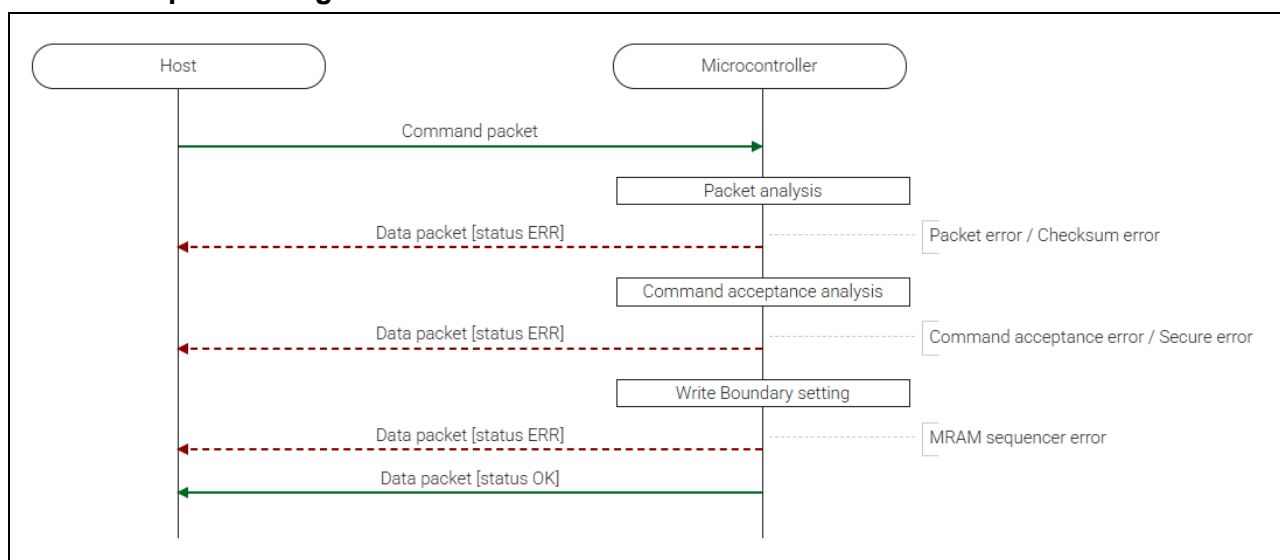


Figure 31. Boundary Setting Command Sequence Diagram

6.13.2 Packets**6.13.2.1 Command Packet**

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	0Bh
CMD	(1 byte)	4Eh (Boundary setting command)
RSV	(2 bytes)	0000h (unused code)
CMS	(2 bytes)	Size of Code MRAM Secure region [KB]. For example: 0100h -> 01h, 00h (256 KB) * 32 KB align
RSV	(1 byte)	00h (unused code)
DMS	(1 byte)	Size of Data MRAM Secure region [KB] For example: 08h (8KB) * 2 KB align
RSV	(2 bytes)	0000h (unused code)
RSV	(2 bytes)	0000h (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

* If CMS does not comply with alignment, boot firmware rounds down them.

6.13.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	4Eh (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.13.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	CEh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.13.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- If the current Authentication level is AL1 or AL0, the boot firmware sends a "Secure error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware writes the boundary setting:

- If an error occurs while writing, the boot firmware sends a "MRAM sequencer error" and returns to the command wait state.
- When the write processing is normally finished, boot firmware returns "OK" and waits for the next command.

6.13.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Authentication level is AL1 or AL0.	Secure error	FFFFFFFFh	FFFFFFFFh
MACI detected an error after the command execution in an undisclosed area.	MRAM sequence error	MRAM status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.13.5 Example of Use

The relationship between boundary settings and secure regions are shown below.

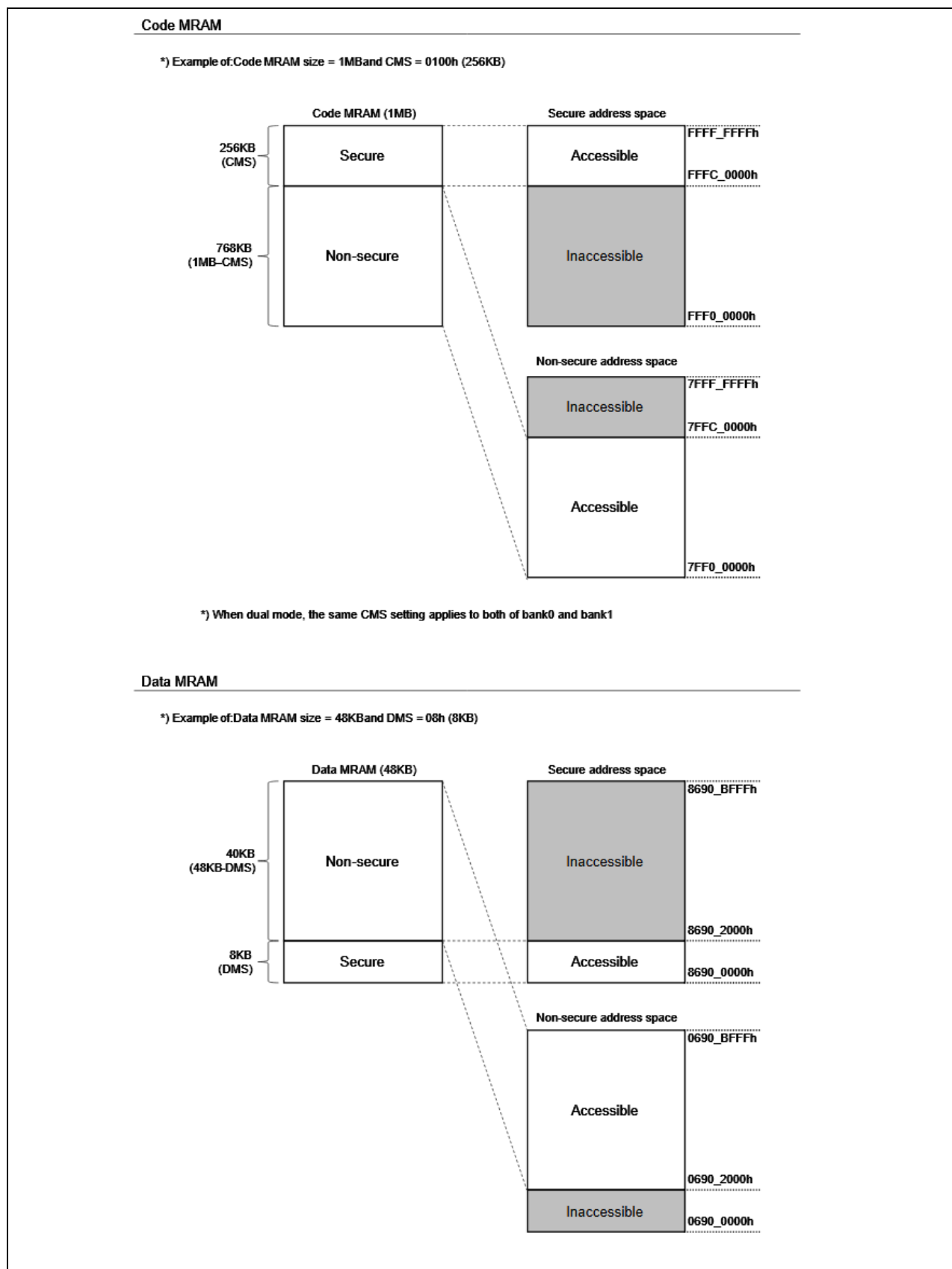


Figure 32. Boundary Setting Example

6.14 Boundary Request Command

This command sends the boundary setting value to the host. (Returns the value currently stored in the device.)

This command requires adherence to conditions described in Command List.

6.14.1 Sequence Diagram

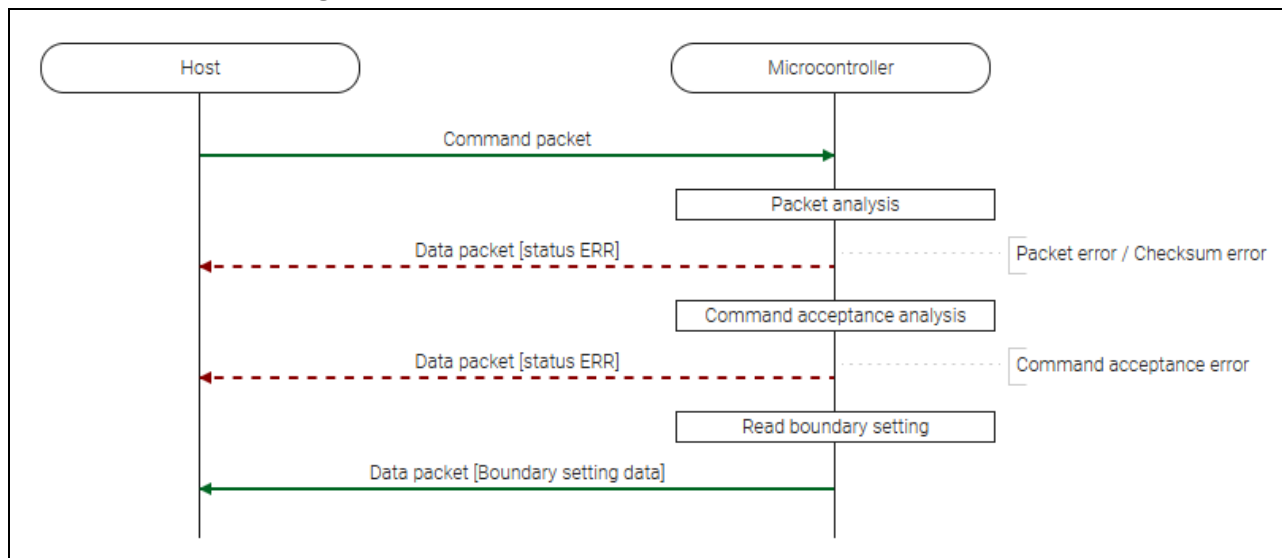


Figure 33. Boundary Request Command Sequence Diagram

6.14.2 Packets

6.14.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
CMD	(1 byte)	4Fh (Boundary request command)
SUM	(1 byte)	B0h
ETX	(1 byte)	03h

6.14.2.2 Data packet [Boundary Setting Data]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Bh
RES	(1 byte)	4Fh (OK)
RSV	(2 bytes)	0000h (unused code)
CMS	(2 bytes)	Size of Code MRAM Secure region [KB]. For example: 0100h->01h, 00h (256 KB)
RSV	(1 byte)	00h (unused code)
DMS	(1 byte)	Size of Data MRAM Secure region [KB] For example: 08h (8 KB)
RSV	(2 bytes)	0000h (unused code)
RSV	(2 bytes)	0000h (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.14.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	CFh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.14.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, 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.
* Memory contents do not change before command reception.

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

- Boot firmware sends "Boundary information" and waits for next command.
* Memory contents do not change before command reception.

6.14.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.15 Parameter Setting Command

This command stores the received parameter in the device.

This command requires adherence to conditions described in Command List.

6.15.1 Sequence Diagram

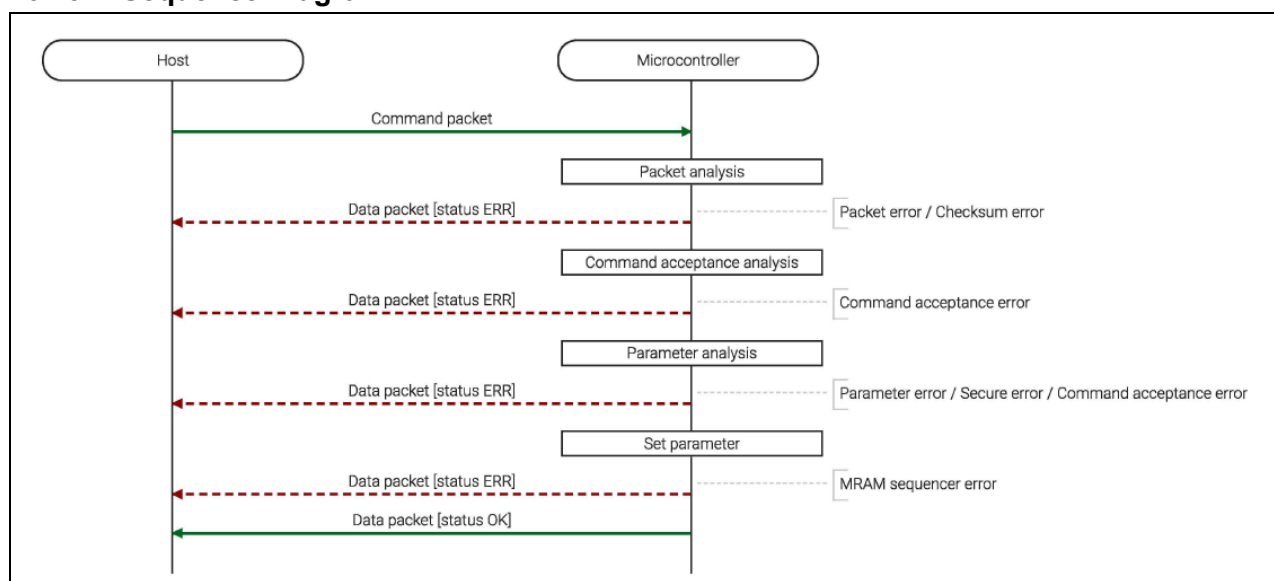


Figure 34. Parameter Setting Command Sequence Diagram

6.15.2 Packets**6.15.2.1 Command Packet**

SOH	(1 byte)	01h			
LNH	(1 byte)	00h			
LNL	(1 byte)	03h			
CMD	(1 byte)	51h (Parameter setting command)			
PMID	(1 byte)	Parameter ID Specifiable parameter:			
		PMID	Parameter description	Specifiable at	Specifiable after Encrypted data write command
		00h	Write protection for User Extra MRAM	AL2	Specifiable
		01h	Disable initialization	AL2/AL1/AL0	Specifiable
		02h	Disable LCK_BOOT	AL2/AL1	Specifiable
		03h	Disable AL2_key	AL2	Specifiable
		04h	Disable AL1_key	AL2/AL1	Non-specifiable
		10h	RMA_KEY OTP selection (*1)	AL2	Non-specifiable
		11h	AL2_KEY OTP selection (*1)	AL2	Non-specifiable
		12h	AL1_KEY OTP selection (*1)	AL2/AL1	Non-specifiable
*1) KEY OTP selection is specifiable when whole the target KEY area is all FFh.					
PRMT	(1 byte)	Parameter data: [PMID=00h] 00h: Enable write protection for User Extra MRAM 07h: Disable write protection for User Extra MRAM [PMID=01h] 00h: Disable initialization [PMID=02h] 00h: Disable transition to LCK_BOOT [PMID=03h] 00h: Disable of authentication using AL2_KEY (*1) [PMID=04h] 00h: Disable of authentication using AL1_KEY [PMID=10h] 00h : Change the storage area for RMA_KEY to OTP [PMID=11h] 00h: Change the storage area for AL2_KEY to OTP [PMID=12h] 00h: Change the storage area for AL1_KEY to OTP			
SUM	(1 byte)	Sum data			
ETX	(1 byte)	03h			

*1: When disabled, initialization and transition to RMA_REQ are also impossible.

6.15.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	51h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.15.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	D1h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.15.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- 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.
 - * Memory contents do not change before command reception.

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

- When designated PMID is unsupported, "Parameter error" is returned.
- When designated PMID cannot be set in the current Authentication level, "Secure error" is returned.
- If both the following conditions are met, the boot firmware sends a "Command acceptance error":
 - Device reset is not asserted after Encrypted data write command execution.
 - Parameter ID that is non-specifiable after Encrypted data write command is specified.
- When OTP selection of key storage area is specified by PMID but the target storage area is not all area FFh, boot firmware sends "Protection error" and returns to command wait state.
- If Write protection for User Extra MRAM is specified by PMID but bit56 in SOFSPSn_S is not 1b, boot firmware sends "Protection error" and returns to command wait state.
- If PRMT is not the specified value, the boot firmware sends a "Parameter error" and returns to the command wait state.
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware writes parameter setting.

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

6.15.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The specified Parameter ID is an unsupported value.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Parameter ID cannot be set at the current Authentication level.	Secure error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: • Device reset is not asserted after Encrypted data write command execution. • Parameter ID that is non-specifiable after Encrypted data write command is specified.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
An OTP selection of KEY area is specified when whole the target KEY area is not FFh	Protection error	FFFFFFFFh	FFFFFFFFh
Write protection for User Extra MRAM is specified when bit56 in SOFSPSn_S is not 1b.	Protection error	FFFFFFFFh	FFFFFFFFh
Parameter data is not the specified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
MACI detected an error after the command execution in an undisclosed area.	MRAM sequence error	MRAM status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

Precautions

- Parameters that only 00h is specifiable as PRMT cannot be unset again once after set.
- Parameters [Write protection for User Extra MRAM], hereafter referred to as "this parameter", can be changed to OTP data by the Memory type select setting mapped in Config area Config area (OTP selectable area) 0, hereafter referred to as "the Memory type select". Therefore it is possible to lock this parameter permanently by setting this parameter after setting the Memory type select.

The reverse setting order, namely setting the Memory type select while this parameter is set is prohibited.

6.16 Parameter Request Command

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.

6.16.1 Sequence Diagram

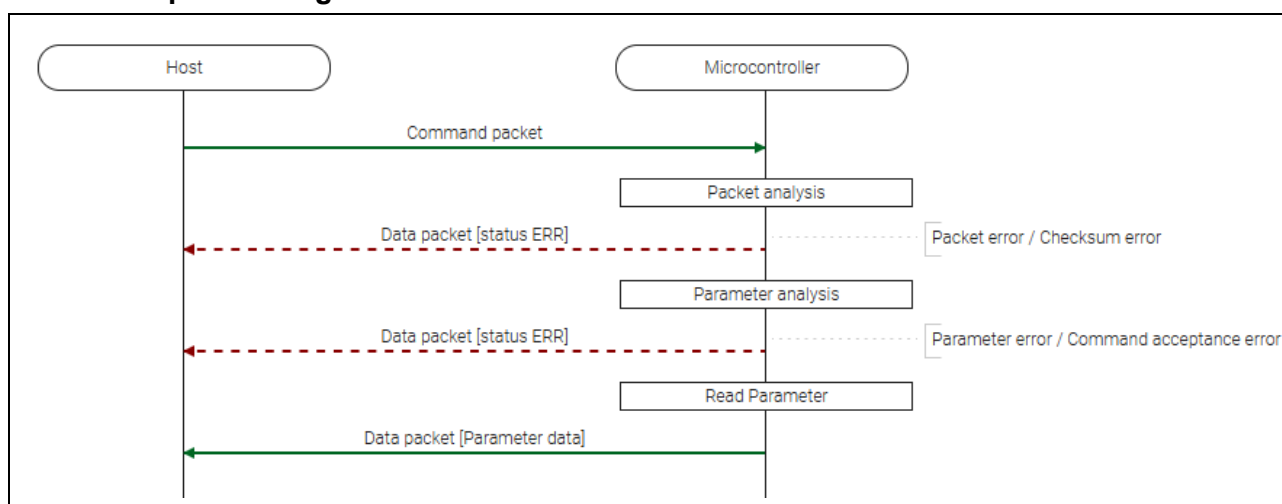


Figure 35. Parameter Request Command Sequence Diagram

6.16.2 Packets

6.16.2.1 Command Packet

SOH	(1 byte)	01h																											
LNH	(1 byte)	00h																											
LNL	(1 byte)	02h																											
CMD	(1 byte)	52h (Parameter request command)																											
PMID	(1 byte)	Parameter ID Specifiable parameter: <table border="1"> <thead> <tr> <th>PMID</th><th>Parameter description</th><th>Specifiable after Encrypted data write command</th></tr> </thead> <tbody> <tr> <td>00h</td><td>Write protection for User Extra MRAM</td><td>Specifiable</td></tr> <tr> <td>01h</td><td>Disable initialization</td><td>Specifiable</td></tr> <tr> <td>02h</td><td>Disable LCK_BOOT</td><td>Specifiable</td></tr> <tr> <td>03h</td><td>Disable AL2_key</td><td>Specifiable</td></tr> <tr> <td>04h</td><td>Disable AL1_key</td><td>Non-specifiable</td></tr> <tr> <td>10h</td><td>RMA_KEY OTP selection</td><td>Non-specifiable</td></tr> <tr> <td>11h</td><td>AL2_KEY OTP selection</td><td>Non-specifiable</td></tr> <tr> <td>12h</td><td>AL1_KEY OTP selection</td><td>Non-specifiable</td></tr> </tbody> </table>	PMID	Parameter description	Specifiable after Encrypted data write command	00h	Write protection for User Extra MRAM	Specifiable	01h	Disable initialization	Specifiable	02h	Disable LCK_BOOT	Specifiable	03h	Disable AL2_key	Specifiable	04h	Disable AL1_key	Non-specifiable	10h	RMA_KEY OTP selection	Non-specifiable	11h	AL2_KEY OTP selection	Non-specifiable	12h	AL1_KEY OTP selection	Non-specifiable
PMID	Parameter description	Specifiable after Encrypted data write command																											
00h	Write protection for User Extra MRAM	Specifiable																											
01h	Disable initialization	Specifiable																											
02h	Disable LCK_BOOT	Specifiable																											
03h	Disable AL2_key	Specifiable																											
04h	Disable AL1_key	Non-specifiable																											
10h	RMA_KEY OTP selection	Non-specifiable																											
11h	AL2_KEY OTP selection	Non-specifiable																											
12h	AL1_KEY OTP selection	Non-specifiable																											
SUM	(1 byte)	Sum data																											
ETX	(1 byte)	03h																											

6.16.2.2 Data Packet [Parameter Data]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	52h (OK)
PRMT	(1 byte)	Parameter data: • [PMID=00h] - 00h: Write protection for User Extra MRAM is enabled. - 07h: Write protection for User Extra MRAM is disabled. • [PMID=01h] - 00h: Initialization is disabled. - 07h: Initialization is enabled. • [PMID=02h] - 00h: Transition to LCK_BOOT is disabled. - 07h: Transition to LCK_BOOT is enabled. • [PMID=03h] - 00h: Authentication using AL2_KEY is disabled (*1). - 07h: Authentication using AL2_KEY is enabled. • [PMID=04h] - 00h: Authentication using AL1_KEY is disabled. - 07h: Authentication using AL1_KEY is enabled. • [PMID=10h] - 00h: Storage area for RMA_KEY is OTP. - 07h: Storage area for RMA_KEY is not OTP. • [PMID=11h] - 00h: Storage area for AL2_KEY is OTP. - 07h: Storage area for AL2_KEY is not OTP. • [PMID=12h] - 00h: Storage area for AL1_KEY is OTP. - 07h: Storage area for AL1_KEY is not OTP.
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

*1: When disabled, initialization and transition to RMA_REQ are also impossible.

6.16.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	D2h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.16.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

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

- When designated PMID is unsupported, "Parameter error" is returned.
- If both the following conditions are met, the boot firmware sends a "Command acceptance error":
 - Device reset is not asserted after Encrypted data write command execution.
 - Parameter ID that is non-specifiable after Encrypted data write command is specified.
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory contents do not change before command reception.

When the processing above is successfully completed, boot firmware returns the parameter setting:

- Boot firmware sends a parameter and waits for the next command.
* Memory contents do not change before command reception.

6.16.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The specified Parameter ID is an unsupported value.	Parameter error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: • Device reset is not asserted after Encrypted data write command execution. • Parameter ID that is non-specifiable after Encrypted data write command is specified.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.17 Inquiry Command

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

This command requires adherence to conditions described in Command List.

6.17.1 Sequence Diagram

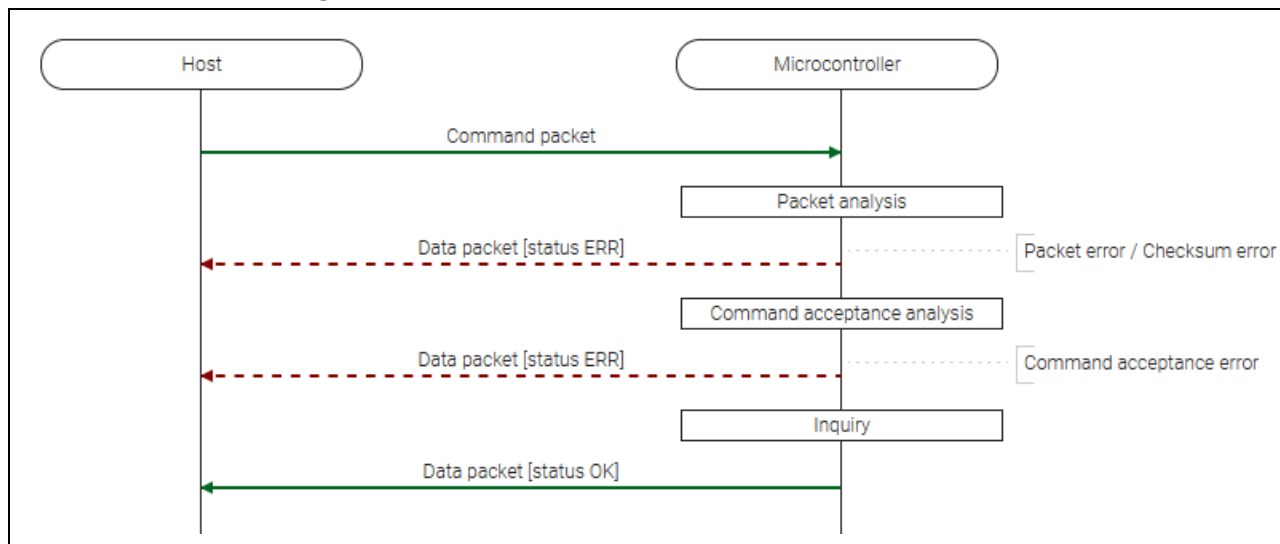


Figure 36. Inquiry Command Sequence Diagram

6.17.2 Packets

6.17.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
CMD	(1 byte)	00h (Inquiry command)
SUM	(1 byte)	FFh
ETX	(1 byte)	03h

6.17.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	00h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	FEh
ETX	(1 byte)	03h

6.17.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	80h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.17.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, 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.
* 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".
* Memory status does not change before command reception.

6.17.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The process has ended normally.	OK	FFFFFFFFh	FFFFFFFFh

6.18 Signature Request Command

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

This command requires adherence to conditions described in Command List.

6.18.1 Sequence Diagram

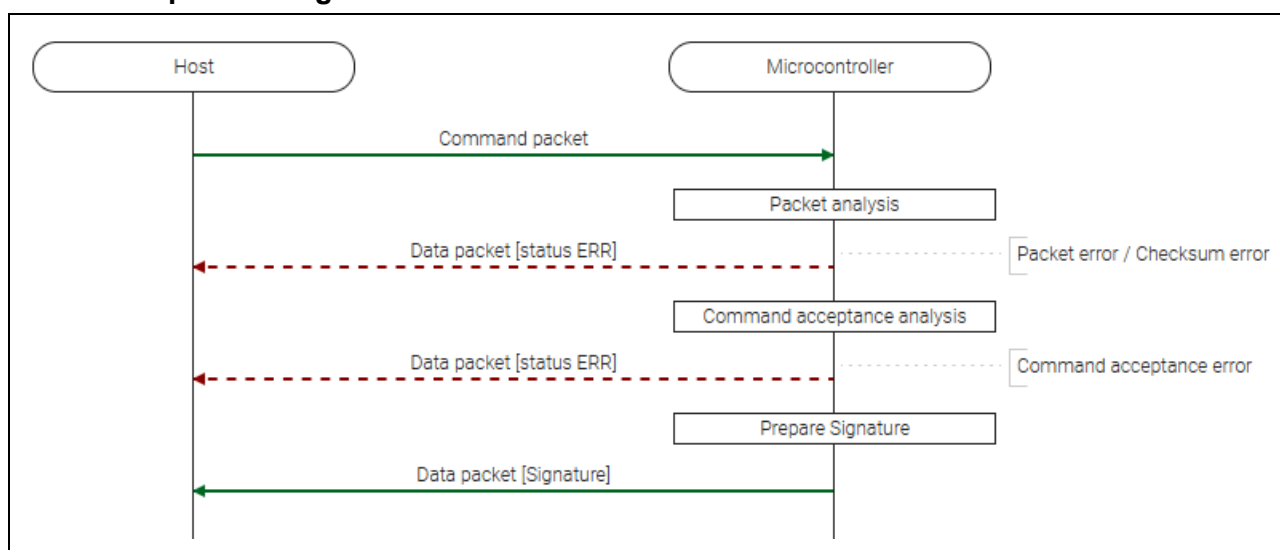


Figure 37. Signature Request Command Sequence Diagram

6.18.2 Packets

6.18.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
CMD	(1 byte)	3Ah (Signature request command)
SUM	(1 byte)	C5h
ETX	(1 byte)	03h

6.18.2.2 Data Packet [Signature]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	2Ah
RES	(1 byte)	3Ah (OK)
RMB	(4 bytes)	Recommended maximum UART baudrate of the device [bps]. *Order of sending: High -> ... -> Low For example: 6 Mbps (6000000bps) -> 00h, 5Bh, 8Dh, 80h
NOA	(1 byte)	Number of accessible areas For example, if the device has 4 areas -> 04h
TYP	(1 byte)	Type code (features and functions of the device): 01h : RX74M MCU Group
BFV	(3 byte)	Boot firmware version Order of sending: Major version -> minor version -> build For example, v2.4.1.6 -> 02h, 04h, 10h
DID	(16 bytes)	Device ID 16-byte ID code (unique ID) for identifying the particular MCU
PTN	(16 bytes)	Product type name. Character strings (20h for space) Order of sending example: R5K574MCA3FC ->52h, 35h, 4Bh, 35h, 37h, 34h, 4Dh, 43h, 41h, 33h, 46h, 43h, 20h, 20h, 20h, 20h
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.18.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	BAh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.18.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If the device reset is not asserted after the encrypted data write command execution, 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.
* 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 while waiting.
* Memory status does not change before command reception.

6.18.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.19 Area Information Request Command

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

This command requires adherence to conditions described in Command List.

6.19.1 Sequence Diagram

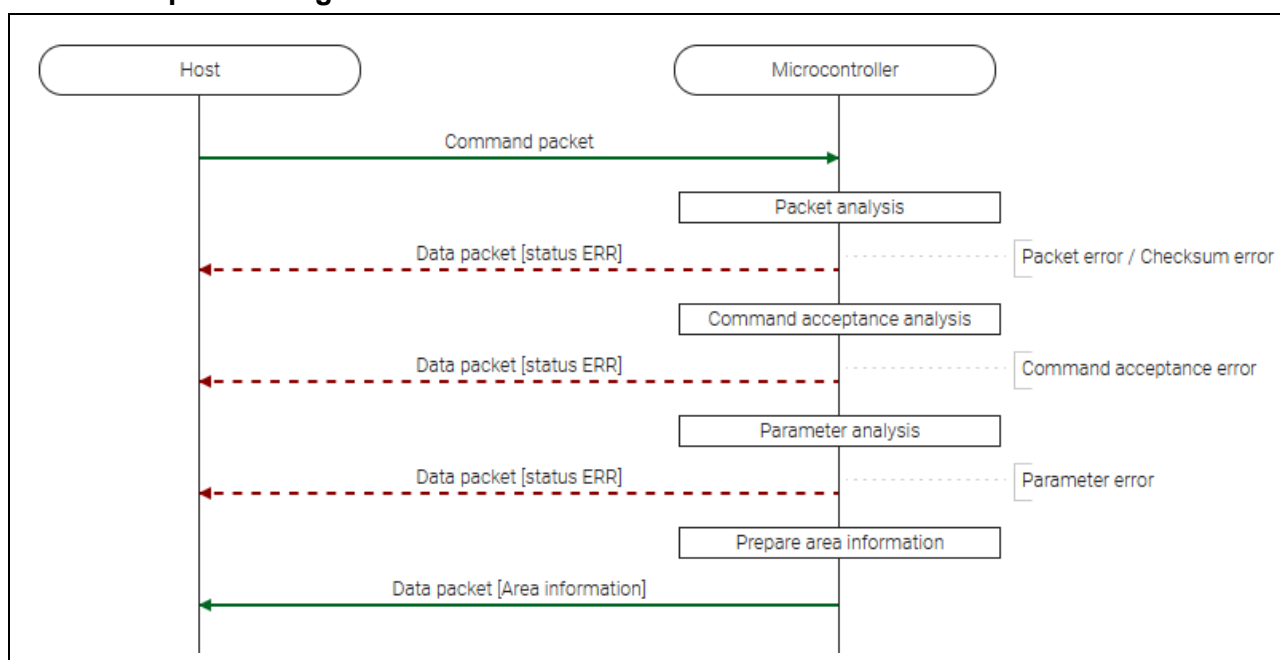


Figure 38. Area Information Request Command Sequence Diagram

6.19.2 Packets

6.19.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
CMD	(1 byte)	3Bh (Area information request command)
NUM	(1 byte)	Area number [0–NOA-1]
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.19.2.2 Data Packet [Area Information]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	1Bh
RES	(1 byte)	3Bh (OK)
KOM	(1 byte)	Kind of the memory (*3) 10h: MRAM
KOA	(1 byte)	Kind of the area: 0Nh: User area N (*2) 1Nh: Data area N (*2) 2Nh: User Extra MRAM N (*2) 3Nh: Config area (Non-OTP area) N (*2) 4Nh: Config area (OTP area) N (*2) 5Nh: Config area (OTP selectable area) N (*2)
SAD	(4 bytes)	Start address. Order of sending: High -> ... -> Low For example: 00010000h -> 00h, 01h, 00h, 00h
EAD	(4 bytes)	End address *Order of sending: High -> ... -> Low For example: 001FFFFFFh -> 00h, 1Fh, FFh, FFh
EAU	(4 bytes)	Reserved for future products and always 00000000h is returned. Target command: None.
WAU	(4 bytes)	Write access unit (alignment) [byte] (*1) Order of sending: High -> ... -> Low For example: 128 byte -> 00h, 00h, 00h, 80h Target command: Write command, User key setting command, User key verify command, Code certificate update command, Encrypted data write command.
RAU	(4 bytes)	Read access unit (alignment) [byte] (*1) Order of sending: High -> ... -> Low For example: 1 byte -> 00h, 00h, 00h, 01h Target command: Read command.
CAU	(4 bytes)	CRC access unit (alignment) [byte] (*1) Order of sending: High -> ... -> Low For example: 4 byte -> 00h, 00h, 00h, 04h Target command: CRC command
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

*1: If each access unit is 00000000h, target command is not available for the area.

*2: N = 0–F

*3: KOM shows memory kind of the area, such as Flash memory or MRAM. It is actually reserved data for future use and is not used in the specifications of the other commands.

6.19.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	BBh (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.19.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If device reset is not asserted after the Encrypted data write command execution, 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.
* Memory status does not change before command reception.

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

- If the specified NUM is "NOA" returned by "Signature request command" or more, send "Parameter error" and return to command waiting status.
* 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.
* Memory status does not change before command reception.

When the processing above is successfully completed, the area information is returned:

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

6.19.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
If Area number in the received packet is a non-existent area number.	Parameter error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

6.19.5 Specific value of Area Information

Example: [RX74M (Linear mode)] - User area=1MB, Data area=48KB

NUM	Area	KOM	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area for Secure (*2)	10h	00h	FFF0_0000h (*3)	FFFF_FF FFh	0 (*1)	32B	1B	32KB
1	User area for Non-secure (*2)	10h	01h	7FF0_0000h (*3)	7FFF_FF FFh	0 (*1)	32B	1B	32KB
2	Data area for Secure (*2)	10h	10h	8690_0000h	8690_BFFFh (*3)	0 (*1)	16B	1B	1KB
3	Data area for Non-secure (*2)	10h	11h	0690_0000h	0690_BFFFh (*3)	0 (*1)	16B	1B	1KB
4	User Extra MRAM	10h	20h	86D8_0000h	86D9_FFFFh (*3)	0 (*1)	16B	1B	1KB
5	Config area (Non-OTP area) 0	10h	30h	86C2_E820h	86C2_E9 FFh	0 (*1)	16B	1B	16B
6	Config area (Non-OTP area) 1	10h	31h	06C4_EC20h	06C4_EC7Fh	0 (*1)	16B	1B	16B
7	Config area (Non-OTP area) 2	10h	32h	86C9_F020h	86C9_F03Fh	0 (*1)	16B	1B	16B
8	Config area (Non-OTP area) 3	10h	33h	86C9_F060h	86C9_F3 FFh	0 (*1)	16B	1B	16B
9	Config area (Non-OTP area) 4	10h	34h	06C9_F400h	06C9_F7 FFh	0 (*1)	16B	1B	16B
10	Config area (OTP area) 0	10h	40h	86E0_0400h	86E0_05 FFh	0 (*1)	16B	1B	16B
11	Config area (OTP area) 1	10h	41h	86E1_0A00h	86E1_0A FFh	0 (*1)	16B	1B	16B
12	Config area (OTP area) 2	10h	42h	06E2_0C00h	06E2_0C FFh	0 (*1)	16B	1B	16B
13	Config area (OTP area) 3	10h	43h	86E3_0E00h	86E3_0E7Fh	0 (*1)	16B	1B	16B
14	Config area (OTP area) 4	10h	44h	06E3_0E80h	06E3_0E FFh	0 (*1)	16B	1B	16B
15	Config area (OTP selectable area) 0	10h	50h	86E1_0600h	86E1_06 EFh	0 (*1)	16B	1B	16B
16	Config area (OTP selectable area) 1	10h	51h	86E1_0710h	86E1_07 FFh	0 (*1)	16B	1B	16B
17	Config area (OTP selectable area) 2	10h	52h	06E2_0800h	06E2_087 Fh	0 (*1)	16B	1B	16B

*1: EAU is reserved for this product.

*2: The accessible address changes depending on the boundary settings.

*3: The address changes depending on the capacity of the MRAM of each product.

RX74M (Dual mode)] - User area=1MB, Data area=48KB

NUM	Area	KOM	KOA	SAD	EAD	EAU	WAU	RAU	CAU
0	User area for Secure front side (*2)	10h	00h	FFF8_0000h (*3)	FFFF_FF FFh	0 (*1)	32B	1B	32KB
1	User area for Secure back side (*2)	10h	01h	FFB8_0000h (*3)	FFBF_FF FFh	0 (*1)	32B	1B	32KB
2	User area for Non-secure front side (*2)	10h	02h	7FF8_0000h (*3)	7FFF_FF FFh	0 (*1)	32B	1B	32KB
3	User area for Non-secure back side (*2)	10h	03h	7FB8_0000h (*3)	7FBF_FF FFh	0 (*1)	32B	1B	32KB
4 ~	(Areas other than User area are the same as Linear mode)	-	-	-	-	-	-	-	-

*1: EAU is reserved for this product.

*2: The accessible address changes depending on the boundary settings.

*3: The address changes depending on the capacity of the MRAM of each product.

6.20 Baudrate Setting Command

This command receives baudrate data and change 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.

6.20.1 Sequence Diagram

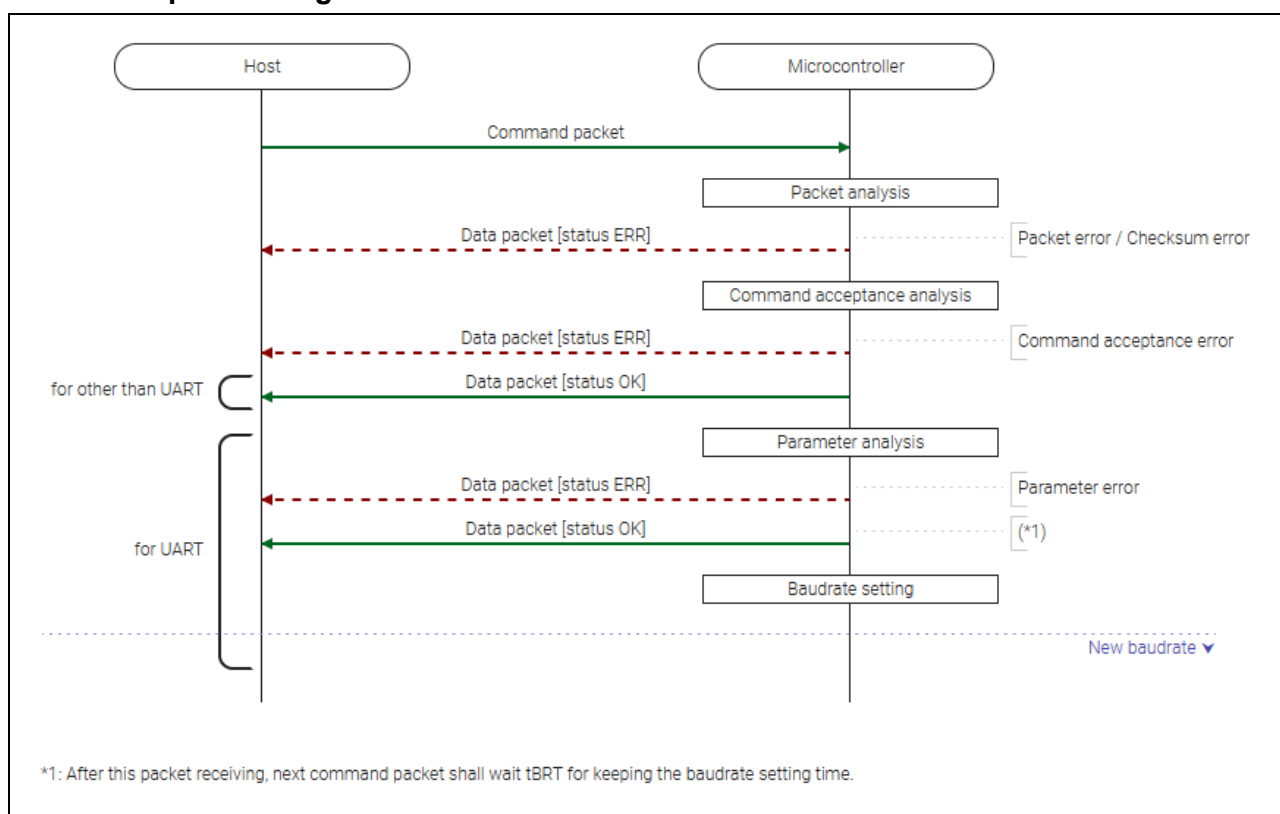


Figure 39. Baudrate Setting Command Sequence Diagram

6.20.2 Packets

6.20.2.1 Command Packet

SOH	(1 byte)	01h																																																	
LNH	(1 byte)	00h																																																	
LNL	(1 byte)	05h																																																	
CMD	(1 byte)	34h (Baudrate setting command)																																																	
BRT	(4 bytes)	UART baudrate [bps] You can set one of the following values. Order of sending BRT: <table><tr><th>Baudrate</th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th></tr><tr><td>9600 bps</td><td>00</td><td>00</td><td>25</td><td>80</td></tr><tr><td>115200 bps</td><td>00</td><td>01</td><td>C2</td><td>00</td></tr><tr><td>500 Kbps</td><td>00</td><td>07</td><td>A1</td><td>20</td></tr><tr><td>1.0 Mbps</td><td>00</td><td>0F</td><td>42</td><td>40</td></tr><tr><td>1.5 Mbps</td><td>00</td><td>16</td><td>E3</td><td>60</td></tr><tr><td>2.0 Mbps</td><td>00</td><td>1E</td><td>84</td><td>80</td></tr><tr><td>4.0 Mbps</td><td>00</td><td>3D</td><td>09</td><td>00</td></tr><tr><td>6.0 Mbps</td><td>00</td><td>5B</td><td>8D</td><td>80</td></tr></table>					Baudrate	1st	2nd	3rd	4th	9600 bps	00	00	25	80	115200 bps	00	01	C2	00	500 Kbps	00	07	A1	20	1.0 Mbps	00	0F	42	40	1.5 Mbps	00	16	E3	60	2.0 Mbps	00	1E	84	80	4.0 Mbps	00	3D	09	00	6.0 Mbps	00	5B	8D	80
Baudrate	1st	2nd	3rd	4th																																															
9600 bps	00	00	25	80																																															
115200 bps	00	01	C2	00																																															
500 Kbps	00	07	A1	20																																															
1.0 Mbps	00	0F	42	40																																															
1.5 Mbps	00	16	E3	60																																															
2.0 Mbps	00	1E	84	80																																															
4.0 Mbps	00	3D	09	00																																															
6.0 Mbps	00	5B	8D	80																																															
SUM	(1 byte)	Sum data																																																	
ETX	(1 byte)	03h																																																	

6.20.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	34h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	CAh
ETX	(1 byte)	03h

6.20.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	B4h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.20.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If the device reset is not asserted after the Encrypted data write command execution, 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.
 - * Memory status does not change before command reception.

When the communication mode is not asynchronous 2-wire communication, a response is returned when the processing above ends normally:

- If the communication mode is not asynchronous 2-wire communication, send "OK" and return to the command waiting state.
 - * Memory status does not change before command reception.

In asynchronous 2-wire communication, parameter analysis is performed when the 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 any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory status does not change before command reception.

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

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

6.20.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Received UART baudrate is greater than RMB.	Parameter error	FFFFFFFFh	FFFFFFFFh
Different from the baudrate value supported by the received UART baudrate.	Parameter error	FFFFFFFFh	FFFFFFFFh
Communication mode is different from UART.	OK	FFFFFFFFh	FFFFFFFFh
Started the baudrate setting.	OK	FFFFFFFFh	FFFFFFFFh

Table 21. Baudrate Setting Values

Intended Baudrate	ABCS	BGDM	CKS [1:0]	BRR [7:0]	MDDR [7:0]	Accuracy
9600bps	0	0	00b	66h	A2h	±0.0%
115200 bps	0	0	00b	07h	97h	±0.0%
500 Kbps	0	0	00b	01h	A4h	+0.1%
1.0 Mbps	0	0	00b	00h	A4h	+0.1%
1.5 Mbps	0	0	00b	00h	F6h	+0.1%
2.0 Mbps	1	0	00b	00h	A4h	+0.1%
4.0 Mbps	1	1	00b	00h	A4h	+0.1%
6.0 Mbps	1	1	00b	00h	F6h	+0.1%
Other	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	-

6.21 Endian Notice Command

This command is used to notify boot firmware the endianness of write data of the user. Depending on the notified endian by this command, boot firmware reverses the endian of data treated in certain commands.

This command require adherence to conditions described in Command List.

This command can be executed only once after device reset. Before executing this command, excepting this command only the following commands can be executed.

- Signature request command
- Area information request command
- Baudrate setting command

6.21.1 Sequence Diagram

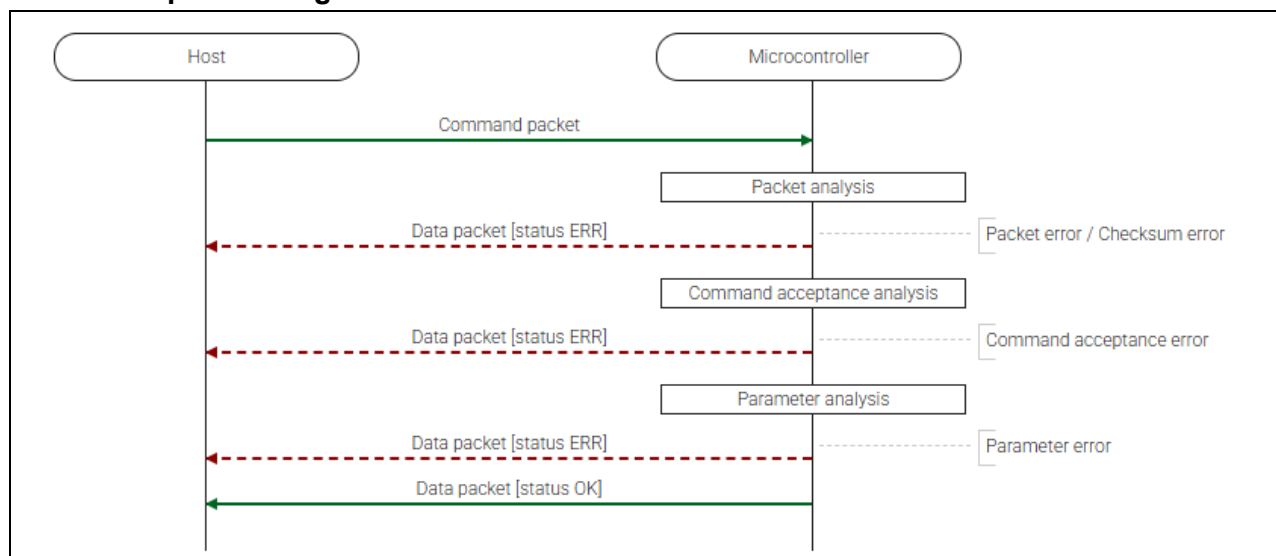


Figure 40. Endian Notice Command Sequence Diagram

6.21.2 Packets

6.21.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
CMD	(1 byte)	36h (Endian notice command)
END	(1 byte)	Endian 00h: Little endian 01h: Big endian
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.21.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	36h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	C8h
ETX	(1 byte)	03h

6.21.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	B6h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.21.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If device reset is not asserted after this command has been executed and successfully completed, boot firmware sends "Command acceptance error".
- When the errors above occur, boot firmware does not execute command processing but returns to command wait state.
 - * Memory contents do not change before command reception.

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

- When designated END is unsupported, "Parameter error" is returned.
- When the errors above occur, boot firmware does not execute command processing but returns to command wait state.
 - * Memory contents do not change before command reception.

When the processing above is successfully completed, notify endian of user write data to Boot firmware:

- Then boot firmware sends "OK" and returns to command wait state.
 - * Memory contents do not change before command reception.

6.21.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is executed and successfully completed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The specified endian is an unsupported value.	Parameter error	FFFFFFFFh	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.21.5 Affected commands by Endian notice command

The table below shows which commands are affected by Endian notice command, and what specifically boot firmware does depending on the notified endian in each command.

Command	What specifically boot firmware does depending on the notified endian in each command
Write command	Reverse the endian of the write data received from the host, then writes to the device
Read command	Reverse the endian of the data read from the device, then sends to the host
CRC command	Reverse the endian of the data read from the device, then calculates the CRC value
Encrypted data write command	Reverse the endian of the decrypted plain-text write data, then writes to the device

On the other hands the data of the commands not listed above are treated in little endian regardless of the endian notified by Endian notice command.

6.22 Write Command

This command receives data from the host and writes those data to the specified area. The alignment of the target address follows 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 for User area.

This command requires adherence to conditions described in Command List.

6.22.1 Sequence Diagram

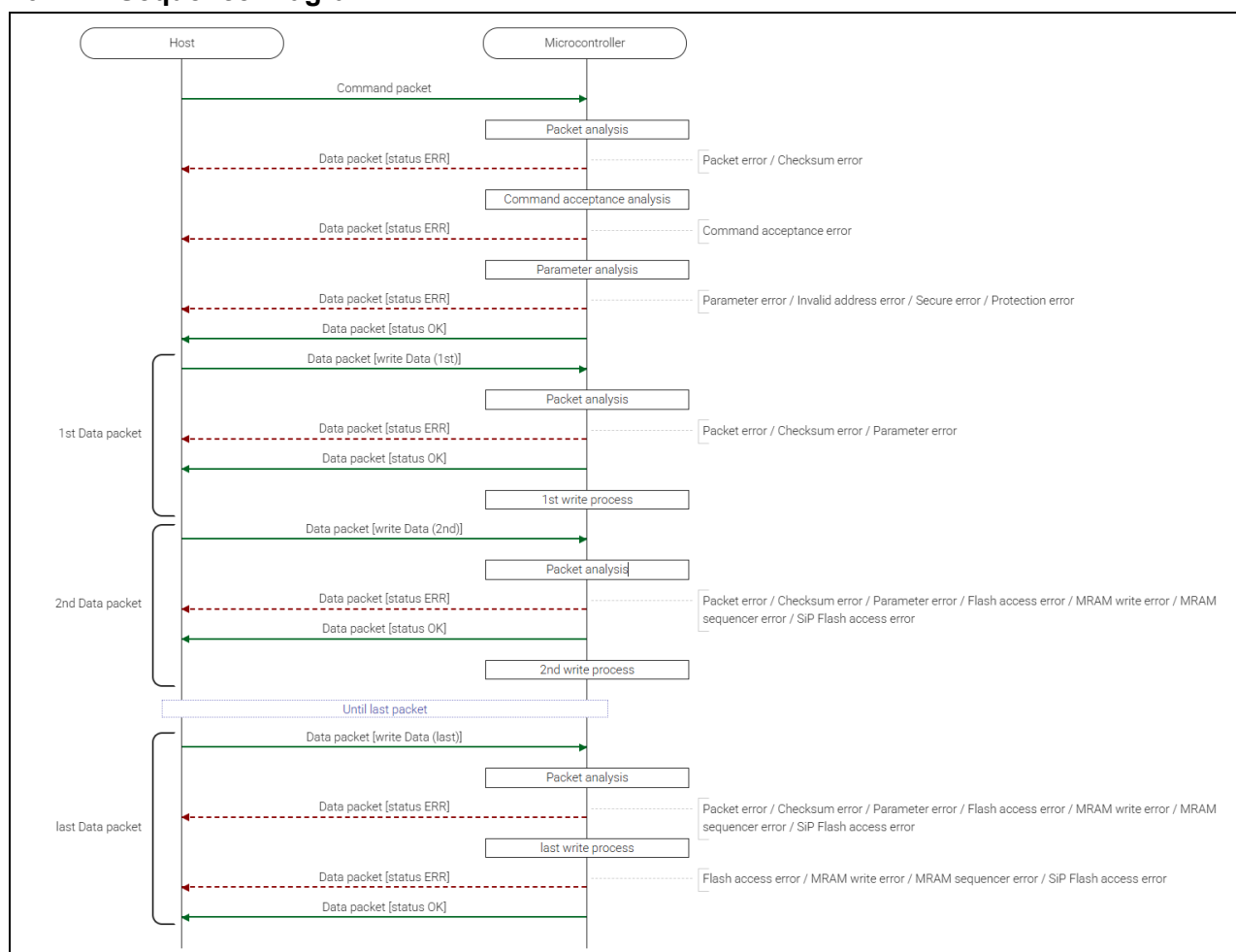


Figure 41. Write Command Sequence Diagram

6.22.2 Packets

6.22.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	13h (Write command)
SAD	(4 bytes)	Start address. For example: FFFFFFFC00h -> FFh, FFh, FCh, 00h
EAD	(4 bytes)	End address. For example: FFFFFFFFh -> FFh, FFh, FFh, FFh
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.22.2.2 Data Packet [Write Data]

SOD	(1 byte)	81h
LNH	(1 byte)	N + 1 (Higher 1 byte)
LNL	(1 byte)	N + 1 (Lower 1 byte)
RES	(1 byte)	13h (OK)
DAT	(N bytes)	Write data
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

N = 1–1024

6.22.2.3 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	13h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.22.2.4 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	93h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.22.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

* 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".
- If the area specified with SAD and EAD includes address that is inaccessible with the current boundary setting, the boot firmware sends a "Invalid address error".
- If the current Authentication level is AL1 and the specified range includes a secure area, the boot firmware sends a "Secure error".
- If the current Authentication level is AL0, the boot firmware sends a "Secure error".
- If both the following conditions are met, boot firmware sends "Protection error":
 - Designated writing range includes area protected by block protection
 - Designated writing range is Data area.
- When designated writing range includes PBPS block, "Protection error" is returned.
- When designated writing range includes MOFSPSn_S, MOFSPS0_NS, SOFSPSn_S or SOFSPS0_NS protected area, "Protection error" is returned.
- If both the following conditions are met, boot firmware sends "Protection error":
 - The write protection for User Extra MRAM is set.
 - Designated writing range is User Extra MRAM.
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * 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 waits 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".
- If all the following conditions are met, boot firmware sends "Protection error":
 - The write protection for User Extra MRAM is set
 - Designated writing range includes the memory type select setting for User Extra MRAM
 - Write data to the memory type select setting for User Extra MRAM is different from the value currently written
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * 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 receives the next data packet, returns “MRAM write error” or “MRAM sequencer error” and waits for the next command.
* WAU size from failure address (ADR) of 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.
- When the write processing is abnormally finished, boot firmware returns “MRAM write error” or “MRAM sequencer error” and waits for the next command.
* WAU size from failure address (ADR) of 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 memory.

6.22.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not 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 does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Start address is bigger than End address.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address or End address is outside the scope of accessible area specified in area information.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address and End address belong to different types of area.	Parameter error	FFFFFFFFh	FFFFFFFFh
The access unit “WAU” of the specified area is 0.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address or End address does not comply with WAU of the area.	Parameter error	FFFFFFFFh	FFFFFFFFh
The area from Start address to End address contains addresses that are inaccessible with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh
Current Authentication level is AL1, and designated writing range includes Secure region.	Secure error	FFFFFFFFh	FFFFFFFFh
Current Authentication level is AL0.	Secure error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: - Designated writing range includes area protected by block protection - Designated writing range is Data area	Protection error	FFFFFFFFh	FFFFFFFFh
Designated writing range includes permanent protected blocks.	Protection error	FFFFFFFFh	FFFFFFFFh

Designated writing range includes MOFSPSn_S, MOFSPS0_NS, SOFSPSn_S, SOFSPS0_NS protected area.	Protection error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: - The write protection for User Extra MRAM is set - Designated writing range is User Extra MRAM	Protection error	FFFFFFFFh	FFFFFFFFh
The response code of the received data packet is different from the value specified by this command.	Packet error	FFFFFFFFh	FFFFFFFFh
The total length of received data of data packets exceeds the specified end address.	Parameter error	FFFFFFFFh	FFFFFFFFh
The data size of the data packet does not comply with writing unit of the area.	Parameter error	FFFFFFFFh	FFFFFFFFh
All the following conditions are met: - The write protection for User Extra MRAM is set - Designated writing range includes the memory type select setting for User Extra MRAM - Write data to the memory type select setting for User Extra MRAM is different from the value currently written	Protection error	FFFFFFFFh	FFFFFFFFh
An error detected when MRAM programming which does not use MACI command	MRAM write error	FFFFFFFFh	Failure address
MACI detected an error after the command execution.	MRAM sequencer error	MRAM status	Failure address
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.22.5 Precautions

- (1) When the following protection is set, the protected area cannot be rewritten. Therefore, rewrite the protected area before setting the followings.
 - Permanent block protection
 - MOFSPSn_S
 - MOFSPS0_NS
 - SOFSPSn_S
 - SOFSPS0_NS
- (2) The memory type select setting for User Extra MRAM is mapped in Config area.
When changing User Extra MRAM to OTP, the memory type select setting must be written earlier than writing User Extra MRAM.
Since there is no protection setting is mapped in Config area, it is recommended that writing to User Extra MRAM is performed after writing to all the Config areas are done.

Similarly, there are memory type select settings for Config area are mapped in also Config area. Basically there is no need to pay attention to the writing order because the memory type select setting is written first when writing in the order of the address. However, for example when writing to the area multiple times separately, please be careful not to write in reverse order.

- (3) When writing to Data area, the block protection settings for the write target area must be disabled beforehand.
- (4) OTP area cannot be written back to 1 once after 0 is written.

For OTP area which has ECC, also ECC bits cannot be written back from 0 to 1. Therefore note that re-writing is not possible even when the data bits are to be changed only from 1 to 0. However when the following cases re-writing is possible as an exception:

- Re-writing the same value.
- The data already written is all "F".

- (5) When MRAM sequencer error occurs when writing Config area and the returned ST2 value is CFGSETERR=1b, an ECC error that cannot be corrected would have been occurred.

Please resolve the ECC error using one of the following ways:

- Execute Initialize command
- Identify the address where the error occurred by the returned ADR value and rewrite the identified address with the correct value.

6.23 Read Command

This command reads data from a specified area and sends that data to the host. The alignment of the target addresses 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.

6.23.1 Sequence Diagram

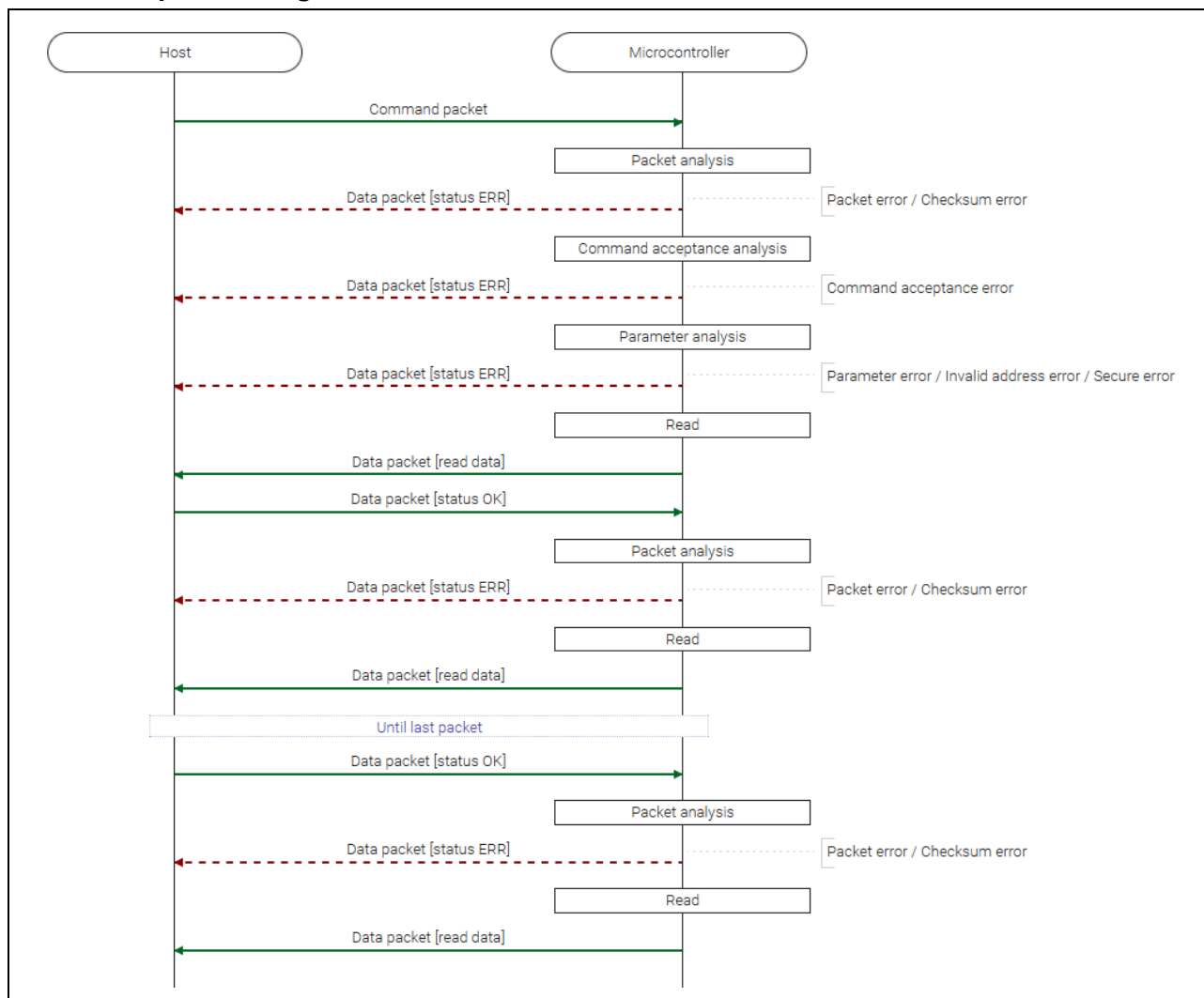


Figure 42. Read Command Sequence Diagram

6.23.2 Packets**6.23.2.1 Command Packet**

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	15h (Read command)
SAD	(4 bytes)	Start address For example: FFFFFFFC00h -> FFh, FFh, FCh, 00h
EAD	(4 bytes)	End address For example: FFFFFFFFh -> FFh, FFh, FFh, FFh
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.23.2.2 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	15h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.23.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	95h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.23.2.4 Data Packet [Read Data]

SOD	(1 byte)	81h
LNH	(1 byte)	N + 1 (Higher 1 byte)
LNL	(1 byte)	N + 1 (Lower 1 byte)
RES	(1 byte)	15h (OK)
DAT	(N byte)	Read data
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.23.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis.

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If device reset is not asserted after the Encrypted data write command execution, 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.
 - * Memory status does not change before command reception.

When the processing above is successfully completed, 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, the boot firmware sends a "Parameter error".
- If the RAU for the specified area is 0, the boot firmware sends a "Parameter error".
- If the area specified with SAD and EAD includes an address that is inaccessible with the current boundary setting, the boot firmware sends a "Invalid address error".
- If the current Authentication level is AL1 and the specified range includes a secure area, boot firmware sends "Secure error".
- If the current Authentication level is AL0, boot firmware sends "Secure error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory status does not change before command reception.

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

- Boot firmware returns the read data (packet-length: Max.1024 bytes).
- When all the data are sent, the boot firmware waits for the next command.
 - * 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 does not 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.
 - * Memory status does not change before command reception.
- If the above error does not occur, the boot firmware continues to read and send data.

6.23.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Start address is bigger than End address.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address or End address is outside the scope of accessible area specified in area information.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address and End address belong to different type of area.	Parameter error	FFFFFFFFh	FFFFFFFFh
The access unit "RAU" of the specified area is 0.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address or End address does not comply with RAU of the area.	Parameter error		
The area from Start address to End address contains addresses that are inaccessible with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh
Current Authentication level is AL1, and designated reading range includes area whose security attribution for read is Secure.	Secure error	FFFFFFFFh	FFFFFFFFh
Current Authentication level is AL0.	Secure error	FFFFFFFFh	FFFFFFFFh
The response code of the received data packet is different from the value specified by this Packet error command.	Packet error	FFFFFFFFh	FFFFFFFFh

6.24 CRC Command

This command calculates CRC data from a specified area and sends it to host. The alignment of the target addresses follows 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. This command requires adherence to conditions described in Command List.

Boot firmware uses the following CRC method.

Name	CRC-32-IEEE-802.3
Default value	FFFFFFFFh
Shift direction	Left shift
Polynomial representations	(MSB first) 04C11 DB7h

6.24.1 Sequence Diagram

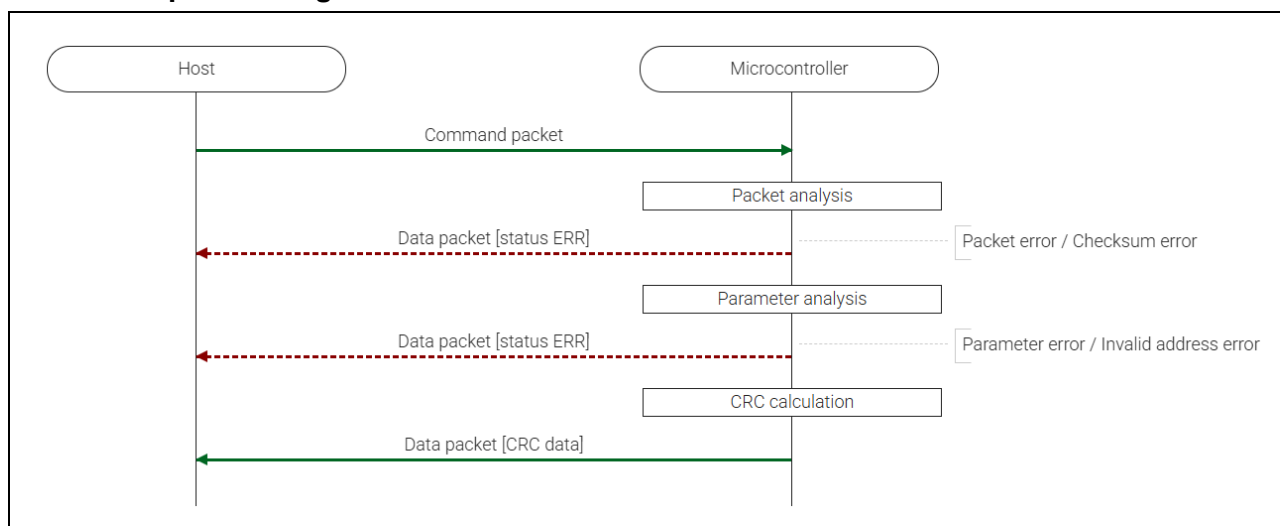


Figure 43. CRC Command Diagram

6.24.2 Packets

6.24.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	18h (CRC command)
SAD	(4 bytes)	Start address
EAD	(4 bytes)	End address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.24.2.2 Data Packet [CRC Data]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	05h
RES	(1 byte)	18h (OK)
CRC	(4 bytes)	CRC data (result of calculation) e.g.) 01234567h -> 01 h, 23h, 45h, 67h
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.24.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	98h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.24.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- When the errors above occur, boot firmware does not execute command processing but returns to command wait state.
* Memory contents do not change before command reception.

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

- If SAD is greater than EAD, the boot firmware sends a "Parameter error".
- If 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, the boot firmware sends 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".
- If the area specified with SAD and EAD includes address that is inaccessible with the current boundary setting, the boot firmware sends a "Invalid address error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
* Memory status does not change before command reception.

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

- After the CRC calculation, boot firmware returns "CRC data" and waits next command.
* Memory status does not change before command reception.

6.24.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Start address is bigger than End address.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address or End address is outside the scope of accessible area specified in area information.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address and End address belongs to different type of the area.	Parameter error	FFFFFFFFh	FFFFFFFFh
The access unit "CAU" of the specified area is 0.	Parameter error	FFFFFFFFh	FFFFFFFFh
Start address or End address doesn't comply with CAU of the area.	Parameter error		
The area from Start address to End address contains addresses that are inaccessible with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh

6.25 Code Certificate Update Command

This command executes the following depending on the specified MAC type by the Command packet.

[MAC type: HMAC-SHA256]

- Check the integrity of the following:
 - "Hash of OEM root public key" and "Key certificate".
 - "Key certificate" and "Code certificate".
 - "Code certificate" and "user program verified by FSBL".
- Write the "Code certificate" and "MAC value of Code certificate and the user program verified by FSBL" to the area indicated by Code certificate start address.
- Update the Anti-rollback counter(*1) to the value indicated by Code certificate.
 - *1) Updated counter is either of the followings depending on the Anti-rollback counter mode:
 - mode1: Anti-rollback counter for OEM boot loader.
 - mode2: Anti-rollback counter for Secure application.

[MAC type: None]

- Calculate the CRC value of the user program verified by FSBL and compare it with the CRC value included in Code certificate.
- Write the "Code certificate" to the area indicated by Code certificate start address.

This command requires adherence to the conditions described in Command List.

6.25.1 Sequence Diagram

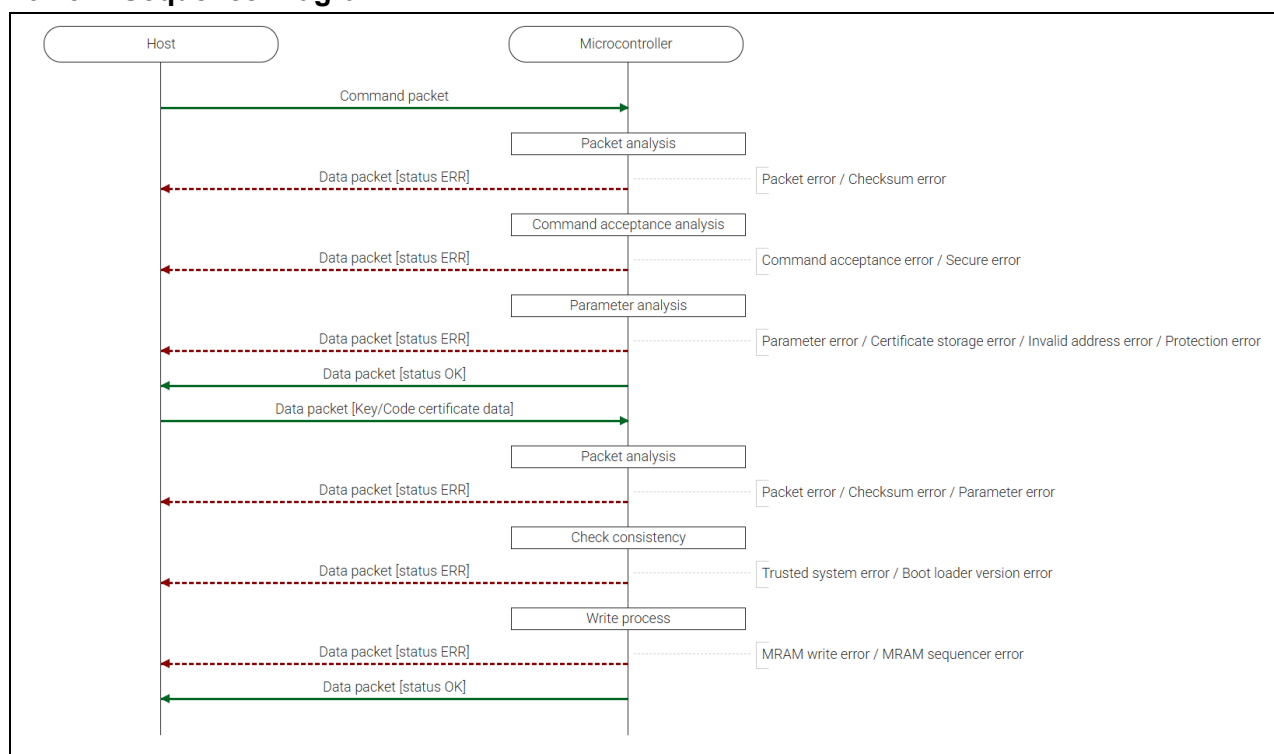


Figure 44. Code Certificate Update Command Diagram

6.25.2 Packets

6.25.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	26h (Code certificate update command)
MAC	(1 byte)	- MAC type 02h: HMAC-SHA256 - FFh: None (CRC check)
AMD	(1 byte)	01h : Anti-rollback counter mode is Mode1 00h : Anti-rollback counter mode is Mode2 *) AMD is ignored and not used when CRC boot
KCS	(2 bytes)	Key certificate size Max. 208 bytes e.g.) 208 bytes -> 00h, D0h
CCS	(2 bytes)	Code certificate size Max. 216 bytes e.g.) 216 bytes -> 00h, D8h
RSV	(2 bytes)	0000h (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.25.2.2 Data Packet [Key/Code certificate data]

SOD	(1 byte)	81h
LNH	(1 byte)	N + M + 1 (Higher 1 byte)
LNL	(1 byte)	N + M + 1 (Lower 1 byte)
RES	(1 byte)	26h (OK)
KCD	(N byte)	Key certificate data
CCD	(M byte)	Code certificate data
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

N = KCS, M = CCS

6.25.2.3 Data Packet [status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	26h (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.25.2.4 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	A6h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.25.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis.

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
 - If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
 - If HUK has been zeroized, the boot firmware sends a "Command acceptance error".
 - If the current Authentication level is AL 1 or AL0, the boot firmware sends a "Secure error".
 - When the above error occurs, the boot firmware does not process and returns to the command waiting state.
- * Memory status does not change before command reception.

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

- When any of the following conditions are met, boot firmware returns "Parameter error".
 - MAC is not specified as MAC type.
 - KCS is larger than specified max size (208 bytes).
 - CCS is larger than specified max size (216 bytes).
- If specified Anti-rollback counter mode is an unsupported value when MAC type is other than "None", boot firmware sends "Parameter error".
- If specified Anti-rollback counter mode is different from the mode that is set to the device when MAC type is other than "None", boot firmware sends "Parameter error".
- If the area for writing Code certificate and MAC value extends outside the range of User area, Data area and User Extra MRAM, boot firmware sends "Certificate storage error".
- If the area for writing Code certificate and MAC value is across different KOAs, boot firmware sends "Certificate storage error".
- If the WAU of the area for writing Code certificate and MAC value is 0, boot firmware sends "Certificate storage error".
- If the area for writing Code certificate and MAC value is not specified in the WAU for the addresses, boot firmware sends "Certificate storage error".
- If the area for writing Code certificate and MAC value includes address that is inaccessible with current boundary setting, boot firmware sends "Invalid address error".
- If both the following conditions are met, boot firmware sends "Protection error":
 - The area for writing Code certificate and MAC value includes area protected by block protection.
 - The area for writing Code certificate and MAC value is Data area.
- If the area for writing Code certificate and MAC value contains a permanent protected block, boot firmware sends "Protection error".
- If both the following conditions are met, boot firmware sends "Protection error":
 - The write protection for User Extra MRAM is set.
 - The area for writing Code certificate and MAC value is User Extra MRAM.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
 - * 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 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.
- 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 size of received KCD and CCD exceeds the size specified by KCS and CCS in the command packet, boot firmware sends "Parameter error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
 - * Memory status does not change before command reception.

When the processing above is successfully completed, boot firmware verifies the consistency.

- If Anti-rollback counter value indicated by Code certificate is in any cases below when MAC type is other than "None", boot firmware sends "Boot loader version error" and returns to command wait state.
 - Anti-rollback counter value indicated by Code certificate is greater than the maximum value of Anti-rollback counter defined by device specifications.
 - Anti-rollback counter value indicated by Code certificate is less than the current Anti-rollback

counter value.

* Memory contents do not change before command reception.

- If the Trusted system becomes abnormal during verification of consistency, the boot firmware returns nothing and no response.
 - * Memory status does not change before command reception.
- If the verification of consistency fails, the boot firmware sends a "Trusted system error" and returns to the command waiting state.
 - * Memory status does not change before command reception.

When the verification of consistency success, boot firmware writes "Code certificate" when MAC type is "None", or "Code certificate" and "MAC value of Code certificate and the verified program" when the type is other than "None" to the Code certificate start address.

- If an error occurs while writing "Code certificate" or "MAC value of Code certificate and the verified program", boot firmware sends "MRAM write error" or "MRAM sequencer error" and returns to command wait state.
 - * WAU size from failure address (ADR) of memory area are undefined.
- When the writing of Code certificate is normally finished and MAC type is None (FFh), the boot firmware returns "OK" and waits for the next command.
 - * Code certificates are written to memory.

When successful writing and MAC type is other than "None", boot firmware updates Anti-rollback counter. As an exception, when Anti-rollback counter value indicated by Code certificate is equal to the current Anti-rollback counter value, boot firmware does not updates Anti-rollback counter.

- When Anti-rollback counter value indicated by Code certificate is equal to the current Anti-rollback counter value, boot firmware sends "OK" and returns to command wait state.
 - * Code certificate and MAC value are written to memory.
- If an error occurs while updating Anti-rollback counter, boot firmware sends "MRAM sequencer error" and returns to command wait state.
 - * Anti-rollback counter of memory area are undefined.
- When the Anti-rollback counter is successfully updated, boot firmware sends "OK" and returns to command wait state.
 - * Code certificate and MAC value are written and Anti-rollback counter is updated.

6.25.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Authentication level is AL1 or AL0.	Secure error	FFFFFFFFh	FFFFFFFFh
The specified MAC type is an unsupported value.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Anti-rollback counter mode is an unsupported value.(*1)	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Anti-rollback counter mode is different from the mode that is set to the device.(*1)	Parameter error	FFFFFFFFh	FFFFFFFFh
Key certificate size exceeds the specified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
Code certificate size exceeds the specified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
The area for writing Code certificate and MAC value includes an area which is neither User area, Data area nor User Extra MRAM.	Certificate storage error	FFFFFFFFh	FFFFFFFFh
The area for writing Code certificate and MAC value spans different Kind of the area.	Certificate storage error	FFFFFFFFh	FFFFFFFFh
The area for writing Code certificate and MAC value doesn't comply with WAU of the area.	Certificate storage error	FFFFFFFFh	FFFFFFFFh
The area for writing Code certificate and MAC value contains inaccessible address with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: - The area for writing Code certificate and MAC value includes area protected by block protection - The area for writing Code certificate and MAC value is Data area	Protection error	FFFFFFFFh	FFFFFFFFh
The area for writing Code certificate and MAC value includes a permanent protected block.	Protection error	FFFFFFFFh	FFFFFFFFh
Both the following conditions are met: - The write protection for User Extra MRAM is set - The area for writing Code certificate and MAC value is User Extra MRAM	Protection error	FFFFFFFFh	FFFFFFFFh
The response code of the received data packet is different from the value specified by this command.	Packet error	FFFFFFFFh	FFFFFFFFh
The number of received KCD and CCD data in the received data packet is different from the KCS and CCS specified in the command packet.	Parameter error	FFFFFFFFh	FFFFFFFFh

Anti-rollback counter value indicated by Code certificate is greater than the maximum value of Anti-rollback counter defined by device specifications.(*1)	Boot loader version error	FFFFFFFFh	FFFFFFFFh
Anti-rollback counter indicated by Code certificate is less than to the current Anti-rollback counter value.(*1)	Boot loader version error	FFFFFFFFh	FFFFFFFFh
Verification of consistency failed.	Trusted system error	Trusted system status	FFFFFFFFh
An error detected when MRAM programming which does not use MACI command.	MRAM write error	FFFFFFFFh	Failure address
MACI detected an error after the command execution.	MRAM sequencer error	MRAM status	Failure address
MACI detected an error after the command execution when updating Anti-rollback counter.(*1)	MRAM sequencer error	MRAM status	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

(*1) This error does not occur when MAC type = None (CRC check).

6.25.5 Precautions

(1) Use this command after writing the following data in advance.

Data	Area to write	Command used for write	Remarks
User program verified by FSBL	User area, User Extra MRAM	Write command, Encrypted data write command	–
Code certificate start address	Config area		
Anti-rollback counter mode	Dedicated area	Key setting command	Not necessary for CRC boot
OEM root public key encrypted Hash value			

(2) Verification fails if data of received Key certificate or Code certificate does not conform to device specifications.

Refer to the user's manual of the device for certificates' specifications.

(3) Key certificate is not necessary when MAC type = None. Specify KCS = 0 and do not send any data as Key certificate data in this case.

6.26 Code Certificate Check Command

This command executes the followings depending on the specified MAC type by the Command packet.
[MAC type: HMAC-SHA256.

- Check the consistency of "Code certificate" and "the MAC value of Code certificate and the user program verified by FSBL" which are stored in the device.

- Read and return the Anti-rollback counter(*1) value which is stored in the device

*1) Returned counter is either of the followings depending on the Anti-rollback counter mode:

mode1: Anti-rollback counter for OEM boot loader

mode2: Anti-rollback counter for Secure application

[MAC type: None]

- Calculate the CRC value of the user program verified by FSBL and compare it with the CRC value included in Code certificate which are stored in the device.

This command requires adherence to conditions described in Command List.

6.26.1 Sequence Diagram

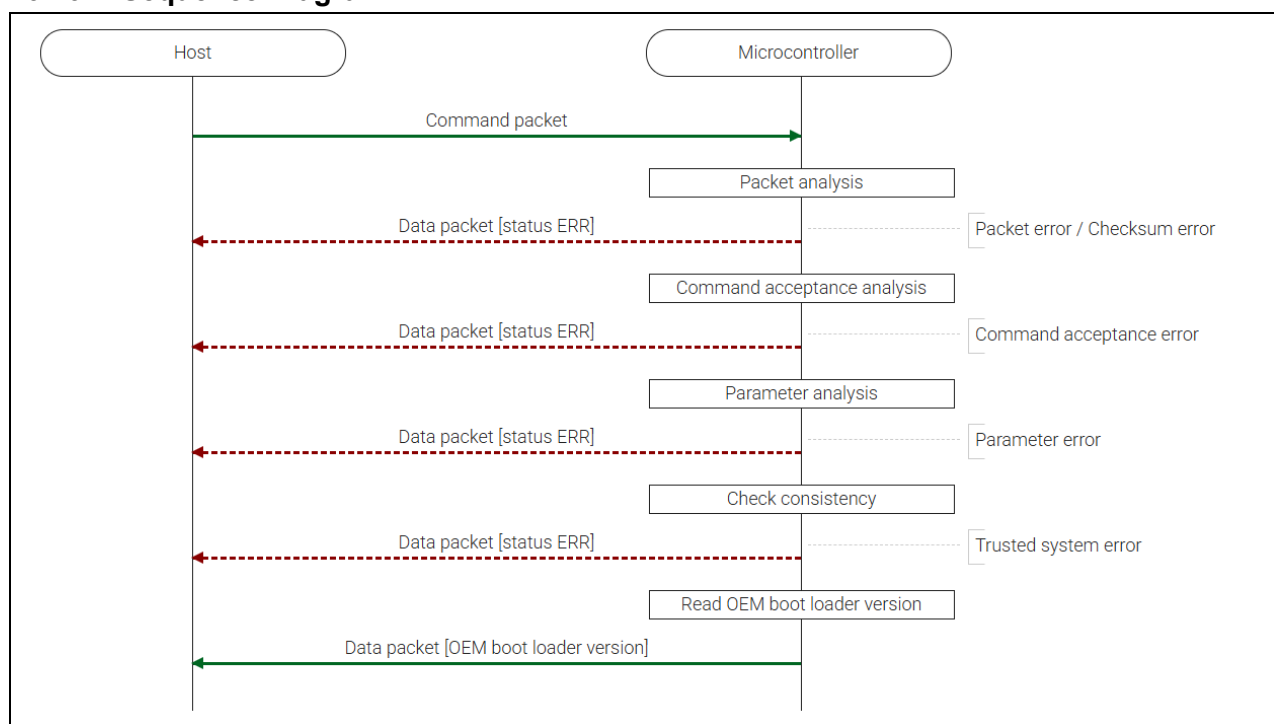


Figure 45. Code Certificate Check Command Diagram

6.26.2 Packets

6.26.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	27h (Code certificate update command)
MAC	(1 byte)	- MAC type 02h: HMAC-SHA256 - FFh : None (CRC check)
AMD	(1 byte)	Anti-rollback counter mode 01h : Anti-rollback counter mode is Mode1 00h : Anti-rollback counter mode is Mode2 *) AMD is ignored and not used when CRC boot
KCS	(2 bytes)	Key certificate size Max. 208 bytes For example: 208 bytes -> 00h, D0h Specify fixed size 208 bytes since KCS is unused in this product.
CCS	(2 bytes)	Code certificate size Max. 216 bytes For example: 216 bytes -> 00h, DBh
RSV	(2 bytes)	0000h (unused code)
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.26.2.2 Data Packet [Anti-rollback counter value]

SOH	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	05h
RES	(1 byte)	27h (OK)
ARC	(4bytes)	Anti-rollback counter value e.g.) Counter value = 10 -> 00h, 00h, 00h, 0Ah When MAC type = None, this field is not used and always FFFFFFFFh.
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.26.2.3 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	A7h (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.26.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If HUK has been zeroized, the boot firmware sends a 'Command acceptance error'.
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.
* Memory status does not change before command reception.

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

- When any of the following conditions are met, boot firmware returns 'Parameter error'.
 - MAC is not specified as MAC type.
 - KCS is larger than specified max size (208 bytes).
 - CCS is larger than specified max size (216 bytes).
- If specified Anti-rollback counter mode is an unsupported value when MAC type is other than "None", boot firmware sends "Parameter error".
- If specified Anti-rollback counter mode is different from the mode that is set to the device when MAC type is other than "None", boot firmware sends "Parameter error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
* Memory status does not change before command reception.

When the processing above is successfully completed, the boot firmware checks the consistency.

- If consistency check fails, the boot firmware sends a "Trusted system error" and returns to the command wait state.
However, boot firmware sends nothing with no response if the Trusted system becomes abnormal during the consistency check.
* Memory status does not change before command reception.

When the verification of consistency success, boot firmware returns Anti-rollback counter value.

※When MAC type is "None", all-F is returned for Anti-rollback counter value.

- When the verification of consistency is successfully completed, boot firmware sends Anti-rollback counter value and returns to command wait state.

* Memory contents do not change before command reception

6.26.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
The specified MAC type is an unsupported value.	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Anti-rollback counter mode is an unsupported value.(*1)	Parameter error	FFFFFFFFh	FFFFFFFFh
The specified Anti-rollback counter mode is different from the mode that is set to the device.(*1)	Parameter error	FFFFFFFFh	FFFFFFFFh
Key certificate size exceeds the specified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
Code certificate size exceeds the specified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
Consistency check failed.	Trusted system error	Trusted system error	FFFFFFFFh

*1) This error does not occur when MAC type = None (CRC check).

6.27 Encrypted Data Write Command

This command receives an encrypted image from the host, decrypts the image, and saves the plain-text image on the device.

In addition, this command changes the device to the specified state, PL0 or LCK_BOOT, when saving the data.

Erase processing and write processing of this command are not affected by the block protection settings for User area.

This command require adherence to conditions described in Command List.

Only the following commands are executable after boot firmware sends status OK to the Command packet of this command until device reset is asserted, regardless of the DLM state at the timing:

Executable command	Note
User key setting command	—
User key verify command	—
Parameter setting command	Depending on parameter ID (PMID), see the command's specifications for details.
Parameter request command	
CRC command	—
Code certificate update command	—
Code certificate check command	—

6.27.1 Sequence Diagram

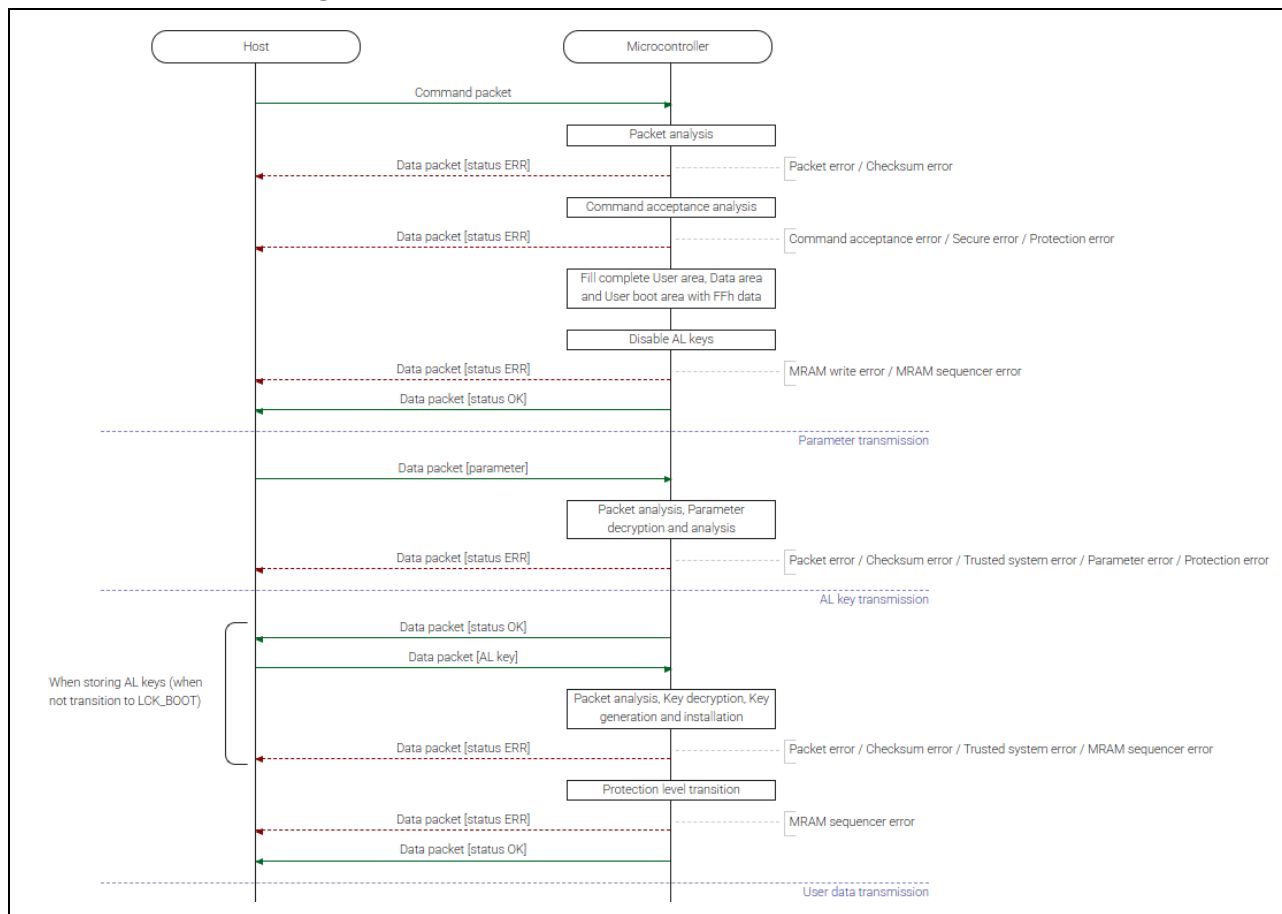


Figure 46. Encrypted Data Write Command Sequence Diagram (Part 1)

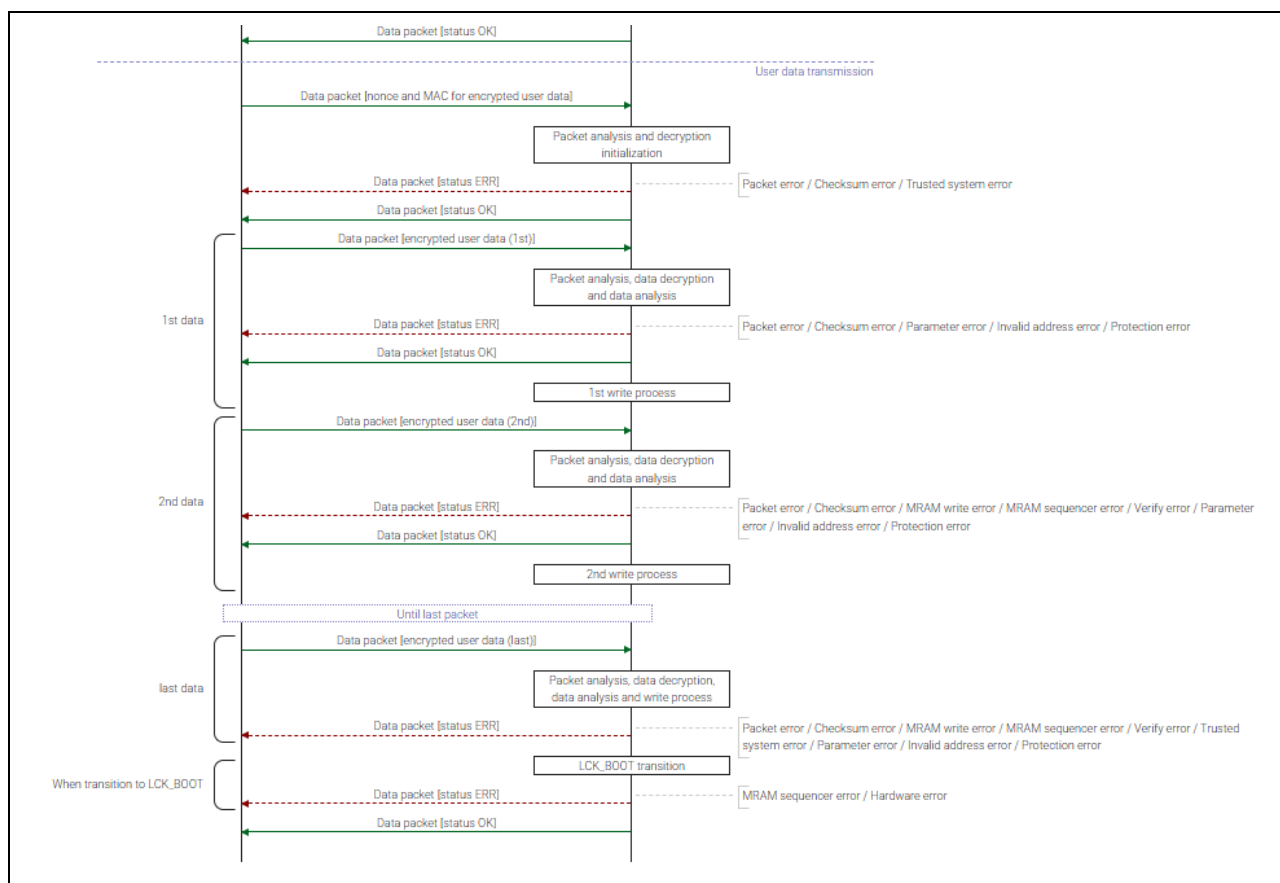


Figure 47. Encrypted Data Write Command Sequence Diagram (Part 2)

6.27.2 Packets

6.27.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	55h
CMD	(1 byte)	1Ah (Encrypted data write command)
SKR	(4 bytes)	Shared key ring number
ESKY	(32 bytes)	Wrapped install key (W-UFPK)
IVEC	(16 bytes)	Initialization Vector used for encrypting ENKY
ENKY	(32 bytes)	Encrypted encryption key MAC. Encryption method is AES128-CBC with CMAC.
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.27.2.2 Data Packet [Parameter]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	3Dh
RES	(1 byte)	1Ah (OK)
NCE	(12 bytes)	Nonce used for encrypting parameters. Nonce length is 12 bytes and counter length is 4 bytes.

PRM	(32 bytes)	Encrypted parameters Encryption method is AES128-CCM mode (NIST SP800-38C). Data format before encryption: <table><tr><td>1st ~ 4th bytes</td><td>5th byte</td><td>6th byte</td><td>7th byte</td><td>8th byte</td></tr><tr><td>LOD</td><td>TRN</td><td>KNUM2</td><td>KNUM1</td><td>END</td></tr><tr><td colspan="5">9th ~ 32th bytes</td></tr><tr><td colspan="5">(reserved)</td></tr></table> Parameter details: <table><tr><td>LOD (4 bytes)</td><td> Total length of “encrypted user data and write address/size” - This field must be from 00000020h to 00FFFFFF0h - This field must be multiple of 16, encryption block size of AES128 Note that the value of this field is not the same as the data size to be written, but calculated by the sum of the two values below: - total bytes of data to be written - 16 byte * N, where N is the total number of “Data packet [encrypted user data]” to be sent An example of how to calculate the value of this field is shown below. [Example of LOD calculation] Example when writing 524,928-byte write data to the following areas: Area1: FFF80000h ~ FFFFFFFFh (524,288 byte) Area2: 86900000h ~ 869001FFh (512 byte) Area3: 86900300h ~ 8690037Fh (128 byte) In this case the number of “Data packet [encrypted user data]” below are sent for sending write data of each area: Area1: 512 packets (*1) Area2: 1 packet (*2) Area3: 1 packet (*2) The number of Data packet [encrypted user data] is 514 in total, thus LOD is the following value. LOD = 524,928 + (16 x 514) = 533,152 (=000822A0h) *1) 512 packets are necessary to send 524,288-byte write data since up to 1024-byte data can be sent by one “Data packet [encrypted user data]”. </td></tr></table>	1st ~ 4th bytes	5th byte	6th byte	7th byte	8th byte	LOD	TRN	KNUM2	KNUM1	END	9th ~ 32th bytes					(reserved)					LOD (4 bytes)	Total length of “encrypted user data and write address/size” - This field must be from 00000020h to 00FFFFFF0h - This field must be multiple of 16, encryption block size of AES128 Note that the value of this field is not the same as the data size to be written, but calculated by the sum of the two values below: - total bytes of data to be written - 16 byte * N, where N is the total number of “Data packet [encrypted user data]” to be sent An example of how to calculate the value of this field is shown below. [Example of LOD calculation] Example when writing 524,928-byte write data to the following areas: Area1: FFF80000h ~ FFFFFFFFh (524,288 byte) Area2: 86900000h ~ 869001FFh (512 byte) Area3: 86900300h ~ 8690037Fh (128 byte) In this case the number of “Data packet [encrypted user data]” below are sent for sending write data of each area: Area1: 512 packets (*1) Area2: 1 packet (*2) Area3: 1 packet (*2) The number of Data packet [encrypted user data] is 514 in total, thus LOD is the following value. LOD = 524,928 + (16 x 514) = 533,152 (=000822A0h) *1) 512 packets are necessary to send 524,288-byte write data since up to 1024-byte data can be sent by one “Data packet [encrypted user data]”.
1st ~ 4th bytes	5th byte	6th byte	7th byte	8th byte																				
LOD	TRN	KNUM2	KNUM1	END																				
9th ~ 32th bytes																								
(reserved)																								
LOD (4 bytes)	Total length of “encrypted user data and write address/size” - This field must be from 00000020h to 00FFFFFF0h - This field must be multiple of 16, encryption block size of AES128 Note that the value of this field is not the same as the data size to be written, but calculated by the sum of the two values below: - total bytes of data to be written - 16 byte * N, where N is the total number of “Data packet [encrypted user data]” to be sent An example of how to calculate the value of this field is shown below. [Example of LOD calculation] Example when writing 524,928-byte write data to the following areas: Area1: FFF80000h ~ FFFFFFFFh (524,288 byte) Area2: 86900000h ~ 869001FFh (512 byte) Area3: 86900300h ~ 8690037Fh (128 byte) In this case the number of “Data packet [encrypted user data]” below are sent for sending write data of each area: Area1: 512 packets (*1) Area2: 1 packet (*2) Area3: 1 packet (*2) The number of Data packet [encrypted user data] is 514 in total, thus LOD is the following value. LOD = 524,928 + (16 x 514) = 533,152 (=000822A0h) *1) 512 packets are necessary to send 524,288-byte write data since up to 1024-byte data can be sent by one “Data packet [encrypted user data]”.																							

			*2) One "Data packet [encrypted user data]" is necessary each for write data of these two areas; cannot send write data for these two areas by single "Data packet [encrypted user data]". This is because even when the total size of the write data is 1024 byte or smaller, the write data cannot be sent by one "Data packet [encrypted user data]" when the write destination address is not consecutive.
		TRN (1 byte)	Transition pattern 00h: OEM (PL0) 02h: LCK_BOOT
		KNUM2 (1 byte)	Number of AL2_key to store, specifiable from 1 to 4 This device has 4 storage areas for AL2_key (AL2_KEYn where n=0~3). Boot firmware stores specified number of keys by this parameter in order from n=0 to n=3. For example KNUM2 = 2 is specified, then AL2_KEY0 and AL2_KEY1 are stored. This parameter is ignored when transiting to LCK_BOOT (TRN=02h)
		KNUM1 (1 byte)	Number of AL1_key to store, specifiable from 0 to 4 The other specifications are the same as KNUM2, see the explanation of KNUM2.
		END (1byte)	Endian of the image to be written 00h: Little endian 01h: Big endian
		reserved	Reserved data is ignored. Although any value is specifiable, but using random numbers or similarly complex values is recommended for security reasons, rather than using simple value such as 00h or FFh.
MAC	(16 bytes)	MAC for Encrypted parameters	
SUM	(1 byte)	Sum data	
ETX	(1 byte)	03h	

6.27.2.3 Data Packet [AL Key]

SOD	(1 byte)	81h																																	
LNH	(1 byte)	N + 29 (Higher 1 byte)																																	
LNL	(1 byte)	N + 29 (Lower 1 byte)																																	
RES	(1 byte)	1Ah (OK)																																	
NCE	(12 bytes)	Nonce used for encrypting AL key data Nonce length is 12 bytes and counter length is 4 bytes																																	
KEY	(N byte)	Encrypted AL key data Encryption method is AES128-CCM mode (NIST SP800-38C) AL key data before encryption is made in the steps: 1. Prepare the number of “one AL key data” specified by KNUM2 and KNUM1. 2. Combine the prepared “one AL key data’s” in order of AL2 key -> AL1 key. Data format of “one AL key data” is the same as IVEC and ENKY of Key setting command, as shown in two tables below. “One AL key data” format before encryption: <table><tr><td>1st ~ 16th bytes</td><td>17th ~ 48th bytes</td></tr><tr><td>IVEC</td><td>ENKY</td></tr></table> Where IVEC and ENKY means: <table><tr><td>IVEC</td><td>Initialization Vector</td></tr><tr><td>ENKY</td><td>Encrypted AL key (0 – 15 bytes) + MAC (16 – 31 bytes)</td></tr></table> As explained above, AL key data is made by combining multiple “one AL key data” in order of AL2 key -> AL1 key. Therefore, for example when KNUM2 = 2 and KNUM1 = 3, AL key data before encryption is as below. For example: AL key data before encryption when KNUM2 = 2 and KNUM1 = 3: <table><tr><td rowspan="2">AL2 key0 data</td><td>1st ~ 16th bytes</td><td>17th ~ 48th bytes</td></tr><tr><td>IVEC</td><td>ENKY</td></tr><tr><td rowspan="2">AL2 key1 data</td><td>49th ~ 64th bytes</td><td>65th ~ 96th bytes</td></tr><tr><td>IVEC</td><td>ENKY</td></tr><tr><td rowspan="2">AL1 key0 data</td><td>97th ~ 112th bytes</td><td>113th ~ 144th bytes</td></tr><tr><td>IVEC</td><td>ENKY</td></tr><tr><td rowspan="2">AL1 key1 data</td><td>145st ~ 160th bytes</td><td>161th ~ 192th bytes</td></tr><tr><td>IVEC</td><td>ENKY</td></tr><tr><td rowspan="2">AL1 key2 data</td><td>193th ~ 208th bytes</td><td>209th ~ 240th bytes</td></tr><tr><td>IVEC</td><td>ENKY</td></tr></table>	1st ~ 16th bytes	17th ~ 48th bytes	IVEC	ENKY	IVEC	Initialization Vector	ENKY	Encrypted AL key (0 – 15 bytes) + MAC (16 – 31 bytes)	AL2 key0 data	1st ~ 16th bytes	17th ~ 48th bytes	IVEC	ENKY	AL2 key1 data	49th ~ 64th bytes	65th ~ 96th bytes	IVEC	ENKY	AL1 key0 data	97th ~ 112th bytes	113th ~ 144th bytes	IVEC	ENKY	AL1 key1 data	145st ~ 160th bytes	161th ~ 192th bytes	IVEC	ENKY	AL1 key2 data	193th ~ 208th bytes	209th ~ 240th bytes	IVEC	ENKY
1st ~ 16th bytes	17th ~ 48th bytes																																		
IVEC	ENKY																																		
IVEC	Initialization Vector																																		
ENKY	Encrypted AL key (0 – 15 bytes) + MAC (16 – 31 bytes)																																		
AL2 key0 data	1st ~ 16th bytes	17th ~ 48th bytes																																	
	IVEC	ENKY																																	
AL2 key1 data	49th ~ 64th bytes	65th ~ 96th bytes																																	
	IVEC	ENKY																																	
AL1 key0 data	97th ~ 112th bytes	113th ~ 144th bytes																																	
	IVEC	ENKY																																	
AL1 key1 data	145st ~ 160th bytes	161th ~ 192th bytes																																	
	IVEC	ENKY																																	
AL1 key2 data	193th ~ 208th bytes	209th ~ 240th bytes																																	
	IVEC	ENKY																																	
MAC	(16 bytes)	MAC for Encrypted AL key data																																	
SUM	(1 byte)	Sum data																																	
ETX	(1 byte)	03h																																	

$$N = (KNUM2 + KNUM1) \times 48$$

6.27.2.4 Data Packet [Nonce and MAC for Encrypted User Data]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	1Dh
RES	(1 byte)	1Ah (OK)
NCE	(12byte)	Nonce used for encrypting user data. Nonce length is 12 bytes, and counter length is 4 bytes.
MAC	(16 bytes)	MAC for Encrypted user data
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.27.2.5 Data Packet [Encrypted User Data]

SOD	(1 byte)	81h																														
LNH	(1 byte)	N + 1 (Higher 1 byte)																														
LNL	(1 byte)	N + 1 (Lower 1 byte)																														
RES	(1 byte)	1Ah (OK)																														
DAT	(N bytes)	Encrypted user data and write address/size. Encryption method is AES128-CCM mode (NIST SP800-38C). Data format before encryption: <table><tr><td>1st ~ 4th bytes</td><td>5th ~ 6th bytes</td><td colspan="2">7th ~ 16th bytes</td></tr><tr><td>SAD</td><td>SIZE</td><td colspan="2">(Reserved: FFh)</td></tr><tr><td colspan="4">17th ~ (n + 16)th bytes (n = 16, 32, 48, ... 1024)</td></tr><tr><td colspan="4">User data</td></tr></table> User data and write address/size details: <table><tr><td>SAD - Write address</td><td colspan="3">Specify write address of the User data/ For example: 02000000h -> 02h, 00h, 00h, 00h</td></tr><tr><td>SIZE - Write size</td><td colspan="3">Specify size of the User data/ For example: 0400h -> 04h, 00h</td></tr><tr><td>User data</td><td colspan="3">Length must be both: • 1024 bytes or less • Least common multiple of the following: - WAU of the write address - Encryption block size (16 bytes for AES128) </td></tr></table>			1st ~ 4th bytes	5th ~ 6th bytes	7th ~ 16th bytes		SAD	SIZE	(Reserved: FFh)		17th ~ (n + 16)th bytes (n = 16, 32, 48, ... 1024)				User data				SAD - Write address	Specify write address of the User data/ For example: 02000000h -> 02h, 00h, 00h, 00h			SIZE - Write size	Specify size of the User data/ For example: 0400h -> 04h, 00h			User data	Length must be both: • 1024 bytes or less • Least common multiple of the following: - WAU of the write address - Encryption block size (16 bytes for AES128)		
1st ~ 4th bytes	5th ~ 6th bytes	7th ~ 16th bytes																														
SAD	SIZE	(Reserved: FFh)																														
17th ~ (n + 16)th bytes (n = 16, 32, 48, ... 1024)																																
User data																																
SAD - Write address	Specify write address of the User data/ For example: 02000000h -> 02h, 00h, 00h, 00h																															
SIZE - Write size	Specify size of the User data/ For example: 0400h -> 04h, 00h																															
User data	Length must be both: • 1024 bytes or less • Least common multiple of the following: - WAU of the write address - Encryption block size (16 bytes for AES128)																															
SUM	(1 byte)	Sum data																														
ETX	(1 byte)	03h																														

N = 32, 48, 64, ... 1040

6.27.2.6 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	1Ah (OK)
STS	(1 byte)	00h (OK)
ST2	(4 bytes)	FFFFFFFFh (unused code)
ADR	(4 bytes)	FFFFFFFFh (unused code)
SUM	(1 byte)	E4h
ETX	(1 byte)	03h

6.27.2.7 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	0Ah
RES	(1 byte)	9Ah (ERR)
STS	(1 byte)	Status code
ST2	(4 bytes)	Status details
ADR	(4 bytes)	Failure address
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.27.3 Processing Procedure

After receiving this command, boot firmware executes the acceptance analysis:

- If Endian notice command is not executed after device reset, boot firmware sends "Command acceptance error".
- If this command cannot be executed in the current DLM state, the boot firmware sends a "Command acceptance error".
- If HUK has been zeroized, the boot firmware sends a "Command acceptance error".
- If the device reset is not asserted after the Encrypted data write command execution, the boot firmware sends a "Command acceptance error".
- If the current Authentication level is AL1 or AL0, the boot firmware sends a "Secure error".
- If any of the following conditions is met, boot firmware sends "Protection error":
 - UMRECR.UMREEN is not 1b.
 - SAS2.BTFLG is not 1b.
 - BANKSEL_NS.BANKSWP[2:0] is not 111b (only for dual mode supported devices).
 - BANKSEL_S.BANKSWP[2:0] is not 111b (only for dual mode supported devices).
- If there are any blocks protected by block protection in Data area, boot firmware sends "Protection error".
- If Permanent protected block exists, the boot firmware sends a "Protection error".
- If write protection for User Extra MRAM is set, boot firmware sends "Protection error".
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

When the processing above is successfully completed, boot firmware erases the complete User area, Data area and User Extra MRAM:

- If erasure fails, boot firmware sends "MRAM write error" or "MRAM sequencer error" and returns to command wait state.
- When erasure completes, boot firmware disables AL keys dependin.
- If an error occurs while disabling AL keys, boot firmware sends "MRAM sequencer error" and returns to command wait state.
- If the above error does not occur, the boot firmware sends "OK".

When the processing above is successfully completed, the boot firmware receives a data packet [parameter] and decrypts and analyzes the parameters:

- Boot firmware receives data packet [parameter].
* Refer to "Data Packet Reception" below for data packet reception processing.
- Boot firmware decrypts the received parameter.
If decryption fails, the boot firmware sends a "Trusted system error".
However, boot firmware sends nothing and becomes unresponsive if the Trusted system becomes abnormal.
- When decryption is complete, boot firmware checks PRM:
 - Boot firmware sends "Parameter error" if PRM is an unspecified value.
 - Boot firmware sends "Protection error" if LCK_BOOT is specified for TRN when the transition to LCK_BOOT is disabled.
 - Boot firmware sends "Parameter error" if END specifies the other endian specified by Endian notice command.
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

When the processing above is successfully completed, the boot firmware receives and writes AL keys depending on the parameters, decrypt and write AL key:

- Boot firmware receives and writes AL keys depending on TRN and KNUM as shown in the following table:

TRN	KNUM2	KNUM1	AL2_key	AL1_key
PL0	1 to 4	1 to 4	X	X
		0	X	-
LCK_BOOT	ignore	ignore	-	-

X: Receive and write

-: Not receive or write

- Boot firmware sends "OK".
- Boot firmware receives data packet [AL key].
* Refer to Data Packet Reception below for data packet reception processing.
- Boot firmware decrypts the received AL key.
If decryption fails, the boot firmware sends a "Trusted system error".
However boot firmware sends nothing and becomes no response if the Trusted system becomes abnormal.
- Boot firmware generates Key index (Wrapped AL key).
If the generation of Key index (Wrapped AL key) fails, the boot firmware sends a "Trusted system error" and returns to the command waiting state.
However, boot firmware sends nothing and becomes unresponsive if the Trusted system becomes abnormal.
- Boot firmware writes Key index (Wrapped AL key) to the device.
If an error occurs while writing Key index (Wrapped AL key), the boot firmware sends a "MRAM sequencer error" and returns to the command wait state.

When the processing above is successfully completed, boot firmware transits the Protection level:

- If an error occurs during Protection level transition, boot firmware returns "MRAM sequencer error" and waits for the next command.
- When Protection level transition is successfully completed, boot firmware sends "OK".

When the processing above is successfully completed, boot firmware receives data packet [nonce and MAC for encrypted user data] and initializes decryption processing:

- Boot firmware receives data packet [nonce and MAC for encrypted user data].
* Refer to "Data Packet Reception" below for data packet reception processing.
- Boot firmware initializes decryption processing.
If initialization fails, the boot firmware sends a "Trusted system error" and returns to the command waiting state.
However, boot firmware sends nothing and becomes unresponsive if the Trusted system becomes abnormal.
- When initialization is successfully completed, boot firmware sends "OK".

When the processing above is successfully completed, boot firmware receives data packet [encrypted user data], decrypts received data, and analyzes decrypted data:

- Boot firmware receives data packet [encrypted user data].
* Refer to "Data Packet Reception" below for data packet reception processing.
- Boot firmware decrypts received encrypted user data.
If decryption fails, the boot firmware sends a "Trusted system error" and returns to the command waiting state.
However, boot firmware sends nothing and becomes unresponsive if the Trusted system becomes abnormal.
- Boot firmware checks SAD/EAD as described below after decryption of encrypted user data:
 - Boot firmware sends "Parameter error" if:
 - SIZE does not match the length of User data.
 - "SAD ~ (SAD + SIZE - 1)" specifies outside the areas defined in area information.
 - "SAD ~ (SAD + SIZE - 1)" spans different types of area.
 - SAD specifies area whose WAU = 0.
 - SAD is not a multiple of WAU.
 - SAD is not a multiple of encryption block size.
 - SIZE is not a multiple of WAU.
 - Boot firmware sends "Invalid address error" if:
 - The area specified by SAD and SIZE includes address that is inaccessible with the current boundary setting.
 - Boot firmware sends a "Protection error" if:
 - Area specified with SAD and SIZE includes area that the MOFSPSn_S, MOFSPS0_NS, SOFSPSn_S, or SOFSPS0_NS is set.

When the error above occurs, the boot firmware does not process and returns to the command waiting state.

When the encrypted user data is not the last write data, boot firmware returns "OK" and executes the write processing:

- Boot firmware returns "OK" and executes the write processing.
- If an error occurs while write processing of the user data, boot firmware receives Data packet, sends "MRAM write error" or "MRAM sequencer error" and returns to command wait state.
- If the write value and write result do not match at writing to Config area, boot firmware received Data packet, sends "Verify error" and returns to command wait state.
- When the write processing is normally finished, the boot firmware receives the next data packet [encrypted user data].

When the encrypted user data is the last write data, the boot firmware executes write processing and then returns a data packet:

When TRN is "LCK_BOOT", boot firmware also executes LCK_BOOT transition before data packet transmission.

- Boot firmware does not return "OK" but executes write processing.
- If an error occurs while write processing of the user data, boot firmware sends "MRAM write error" or "MRAM sequencer error" and returns to command wait state.
- If the write value and write result do not match at writing to Config area, boot firmware sends "Verify error" and returns to command wait state.
- When the write processing is successfully completed and TRN is "LCK_BOOT", boot firmware executes LCK_BOOT transition.
- If an error occurs during LCK_BOOT transition, boot firmware returns "MRAM sequencer error" and waits for the next command.
- If the DLM state after the transition is an invalid value, the boot firmware sends a "Hardware error" and becomes unresponsive.
- When the processing above is successfully completed, the boot firmware returns "OK" and waits for the next command.

6.27.3.1 Data Packet Reception

Data packet reception processing is described below:

- The boot firmware recognizes the start of the data packet by receiving SOD.
If the boot firmware receives something other than SOD, it waits 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 data packet is different from the sum value, the boot firmware sends a "Checksum error".
- If the LNH and LNL of the received command packet 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.
- If the LNH and LNL of the received command packet are different from the values specified in each command, the boot firmware sends a "Packet error".
- If the total received size of "Encrypted user data and write address/size" exceeds LOD, the boot firmware sends a "Parameter error".
*) Only when receiving data packet [encrypted user data].
- When any of the above errors occurs, the boot firmware does not process and returns to the command waiting state.

6.27.4 Status Information from the Microcontroller

(Listed in descending order of priority.)

Condition	STS	ST2	ADR
The received packet does not have ETX.	Packet error	FFFFFFFFh	FFFFFFFFh
Sum data in the packet received is different from the value calculated by the boot firmware.	Checksum error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the packet format.	Packet error	FFFFFFFFh	FFFFFFFFh
Packet length in the received packet does not comply with the specifications of this command.	Packet error	FFFFFFFFh	FFFFFFFFh
Endian notice command is not executed after device reset.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Executing this command is unavailable in the current DLM state.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
HUK has been zeroized.	Command acceptance error	FFFFFFFFh	FFFFFFFFh

Condition	STS	ST2	ADR
Device reset is not asserted after Encrypted data write command execution.	Command acceptance error	FFFFFFFFh	FFFFFFFFh
Authentication level is AL1 or AL0.	Secure error	FFFFFFFFh	FFFFFFFFh
Any of the followings are satisfied: - UMRECR.UMREEN $\neq$ 1b - SAS2.BTFLG $\neq$ 1b - BANKSEL_NS.BANKSWP[2:0] $\neq$ 111b - BANKSEL_S.BANKSWP[2:0] $\neq$ 111b	Protection error	FFFFFFFFh	FFFFFFFFh
There are any blocks protected by block protection in Data area.	Protection error	FFFFFFFFh	FFFFFFFFh
There are any blocks protected by permanent block protection.	Protection error	FFFFFFFFh	FFFFFFFFh
The write protection for User Extra MRAM is set	Protection error	FFFFFFFFh	FFFFFFFFh
An error detected when MRAM programming which does not use MACI command	MRAM write error	FFFFFFFFh	FFFFFFFFh
MACI detected an error after the command execution in the disclosed area.	MRAM sequencer error	MRAM status	FFFFFFFFh
The response code of the received data packet is different from the value specified by this command.	Packet error	FFFFFFFFh	FFFFFFFFh
The processing below fails: • Decryption processing • Wrapped AL key generation	Trusted system error	FFFFFFFFh	FFFFFFFFh
Parameter in Encrypted parameters is an unspecified value.	Parameter error	FFFFFFFFh	FFFFFFFFh
"LCK_BOOT" is specified for the Transition destination when the transition to LCK_BOOT disabled.	Protection error	FFFFFFFFh	FFFFFFFFh
Endian of the image to be written specifies the other endian specified by Endian notice command	Parameter error	FFFFFFFFh	FFFFFFFFh
Protection level or DLM state is abnormal.	Hardware error	FFFFFFFFh	FFFFFFFFh
Any condition of the followings is met: - Total received size of "Encrypted user data and write address/size" exceeds LOD - SIZE does not match the length of User data - "SAD ~ (SAD + SIZE - 1)" specifies outside the areas defined in area information - "SAD ~ (SAD + SIZE - 1)" spans different Kind of the area - SAD specifies area whose WAU = 0 - SAD is not multiple of WAU - SAD is not multiple of encryption block size (*) - SIZE is not multiple of WAU *) 16 byte for AES128	Parameter error	FFFFFFFFh	FFFFFFFFh
"SAD ~ (SAD + SIZE - 1)" contains addresses that are inaccessible with the current boundary settings.	Invalid address error	FFFFFFFFh	FFFFFFFFh
"SAD ~ (SAD + SIZE - 1)" contains an area where the MOFSPSn_S, MOFSPS0_NS, SOFSPSn_S or SOFSPSn_SR is set.	Protection error	FFFFFFFFh	FFFFFFFFh

Condition	STS	ST2	ADR
The written value and the write result do not match at writing at Config area.	Verify error	FFFFFFFFh	FFFFFFFFh
Successful completion.	OK	FFFFFFFFh	FFFFFFFFh

6.27.5 Precautions

(1) This command becomes inexecutable after permanent block protection is set.

(2) This command becomes inexecutable once after write protection for User Extra MRAM is set after changed to OTP.

See specifications of Parameter setting command for details of write protection for User Extra MRAM.

(3) This command becomes inexecutable once after User Boot start-up is enabled by setting UMRECR.UMREEN = 0b.

(4) This command becomes inexecutable if any of the followings is set and also MOFSPSn_S or MOFSPS0_NS for it is set::

- SAS2.BTFLG = 0b
- BANKSEL_NS.BANKSWP[2:0] ≠ 111b
- BANKSEL_S.BANKSWP[2:0] ≠ 111b

(5) Do not set permanent block protection of the area where user keys are to be written when the User key setting command will be used.

Do not set permanent block protection of the area where the Code certificate is to be written when the Code certificate update command will be used.

In addition, for both commands, when the area to be written is Data area, do not set not only permanent block protection but also block protection by this command.

If they are set, both commands become inexecutable due to Protection errors.

(6) Do not write 0b to bit56 of SOFSPSn_S when Write protection for User Extra MRAM is enabled.

If 0b is written, Write protection for User Extra MRAM cannot be altered due to Protection error.

*) Write protection for User Extra MRAM can be set with Parameter setting command.

(7) If the protection settings such as the followings are written before the protected area, this command abnormally finishes at writing of the protected area:

- Permanent block protection
- Block protection for Data area
- MOSFSPSn_S
- MOFSPS0_NS
- SOFSPSn_S
- SOFSPS0_NS

To avoid this, Data packet [encrypted user data] for the protected areas must be sent earlier than ones for protection settings above.

(8) The memory type select setting for User Extra MRAM is mapped in Config area.

Therefore when changing User Extra MRAM to OTP, Data packet [encrypted user data] for the memory type select setting must be sent earlier than ones for User Extra MRAM.

Since there is no protection setting mapped in Config area, it is recommended that writing to User Extra MRAM is performed after writing to all the Config areas are done.

Similarly, there are memory type select settings for Config area are mapped in also Config area.

There is no need to adhere to the writing order because the memory type select setting is written first when writing in the order of the address.

(9) OTP area cannot be written back to 1 once after 0 is written.

For OTP area which has ECC, ECC bits cannot be written back from 0 to 1. Note that re-writing is not

possible even when the data bits are to be changed only from 1 to 0. However when the following cases re-writing is possible as an exception:

- Re-writing the same value.
- The data already written is all "F".

(10) When MRAM sequencer error occurs when writing Config area and the returned ST2 value is CFGSETERR=1b, an ECC error that cannot be corrected would have been occurred.

In this case, please initialize the memory area using Initialize command to resolve the ECC error.

6.27.6 Device State after Command Execution

Table 24 shows the state of the device after this command is executed.

Table 22. Device States after Encrypted Data Write Command Execution

Command finish timing		Device state				
		User/Data/ User Extra MRAM	AL key	Protection level	Area specified by SAD	DLM
Fails at	Command acceptance analysis	No change				
	Erasing complete User/Data/User Extra MRAM	Undefined	No change			
	Disabling AL key	Filled with FFh data	Undefined	No change		
	Decrypting or analyzing the parameter		disabled			
	Decrypting, generating or writing AL key		Undefined or disabled depending on KNUM2, KNUM1 and TRN			
	Transiting Protection level	Filled with FFh data (areas specified by SAD are undefined)	Written or disabled depending on KNUM2, KNUM1 and TRN	Undefined	No change	
	Initializing decryption			PL0	Undefined	No change
	Decrypting, checking, or writing user data					
	Transiting to LCK_BOOT	User data written			User data written	Undefined
Successful completion						LCK_BOOT depending on TRN

6.27.7 DLM State Transitions

The following diagram shows the DLM states that can be transitioned by this command.

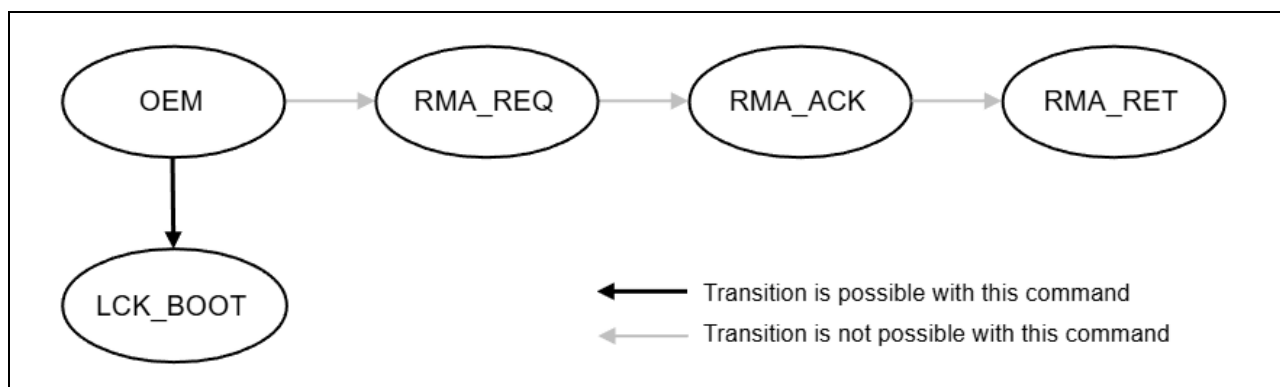


Figure 48. Valid DLM State Transitions for Encrypted Data Write Command

7. Flow Examples

7.1 Beginning Communication

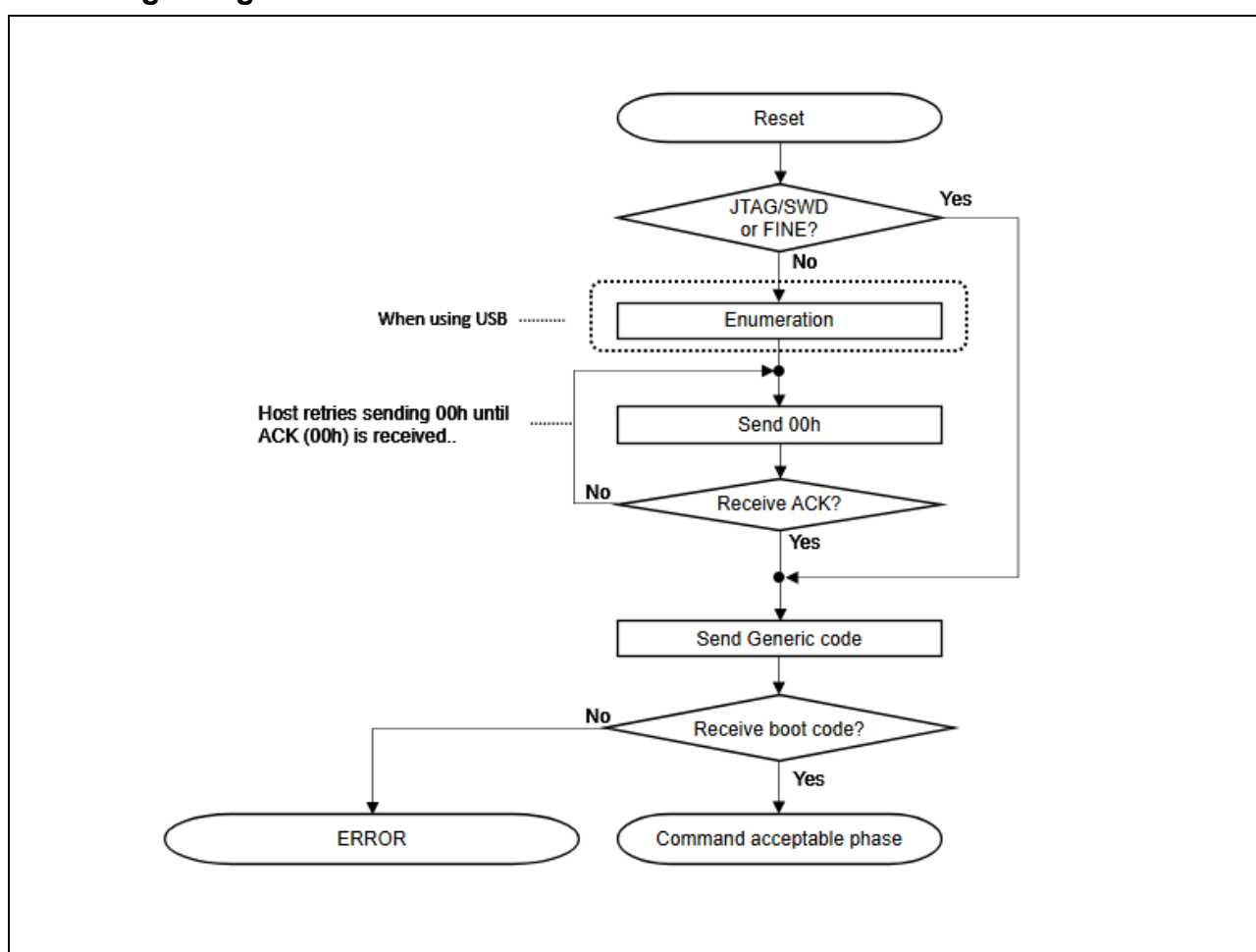


Figure 49. Beginning Communication

7.2 Acquisition of Device Information/Baudrate Settings/Endian Notice

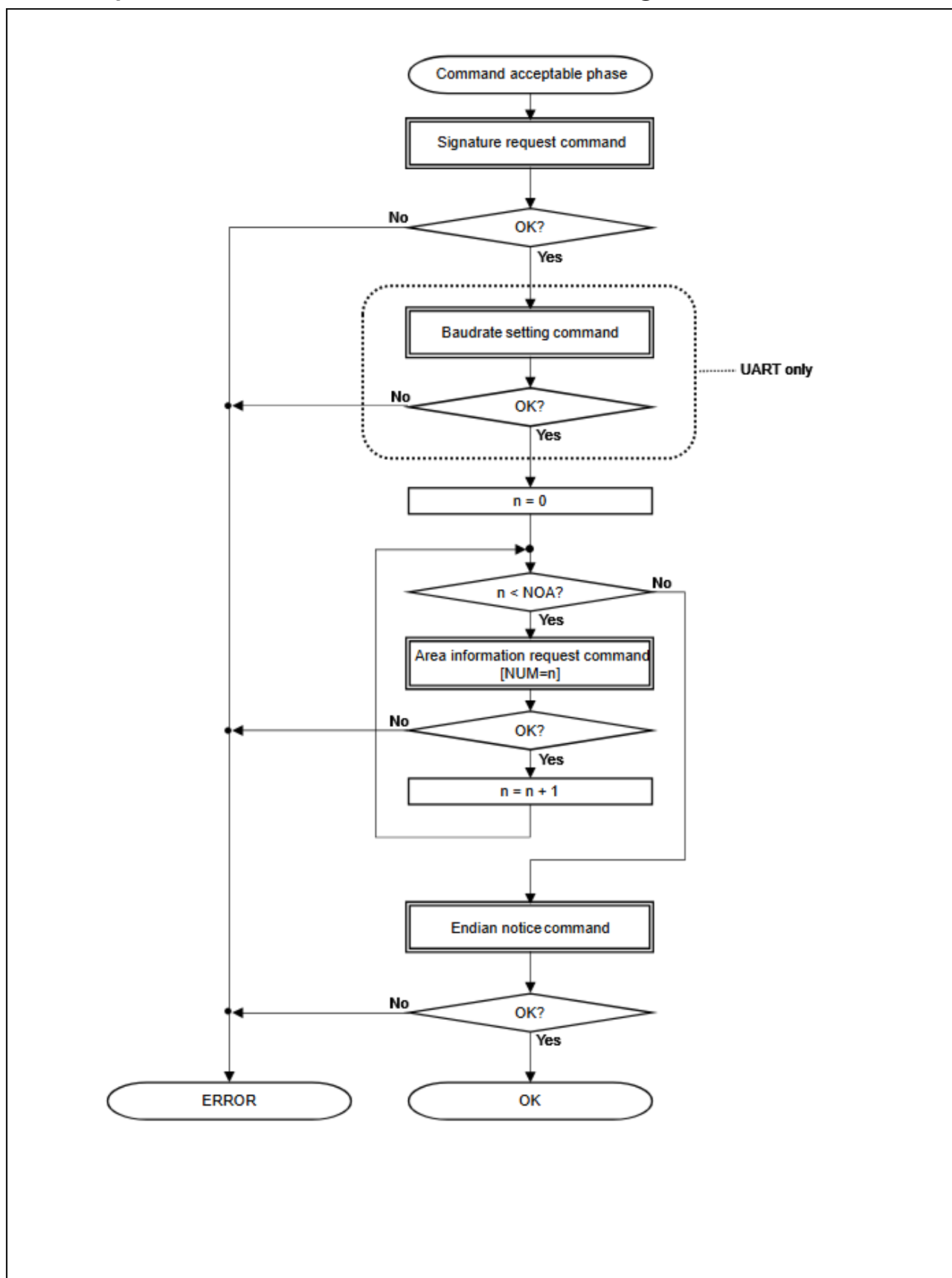


Figure 50. Acquisition of Device Information / Baudrate Settings / Endian Notice

7.3 Transiting DLM State

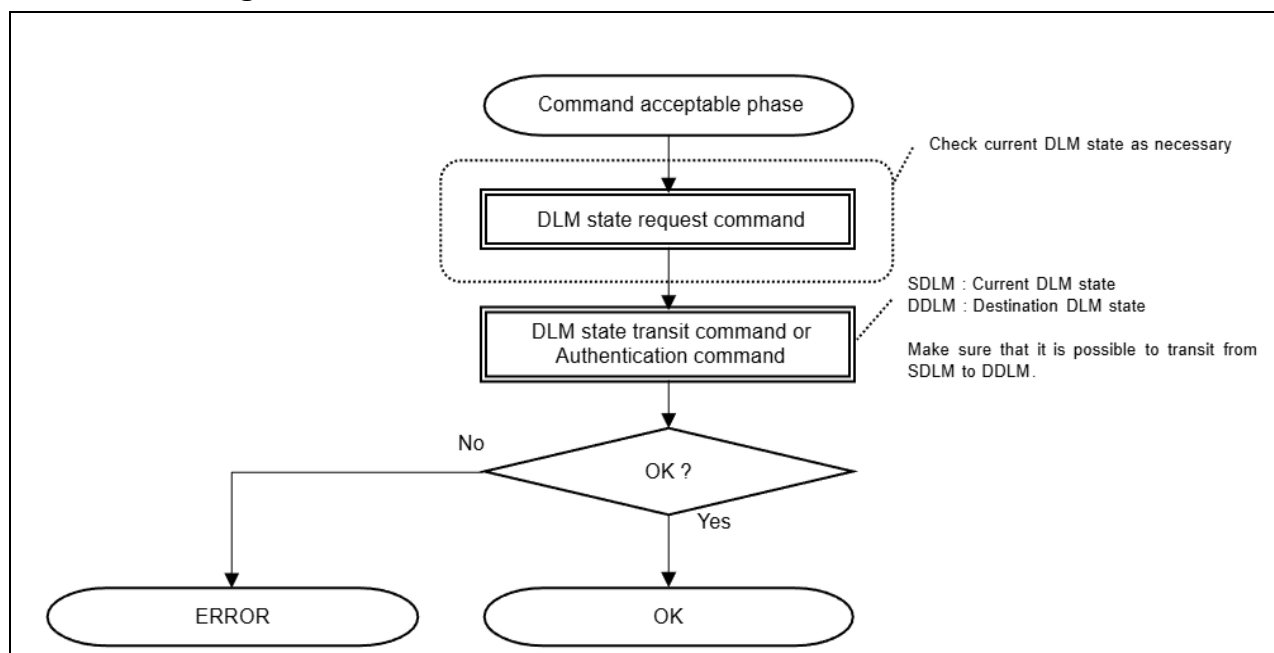


Figure 51. Transiting DLM State

7.4 Transiting Protection Level

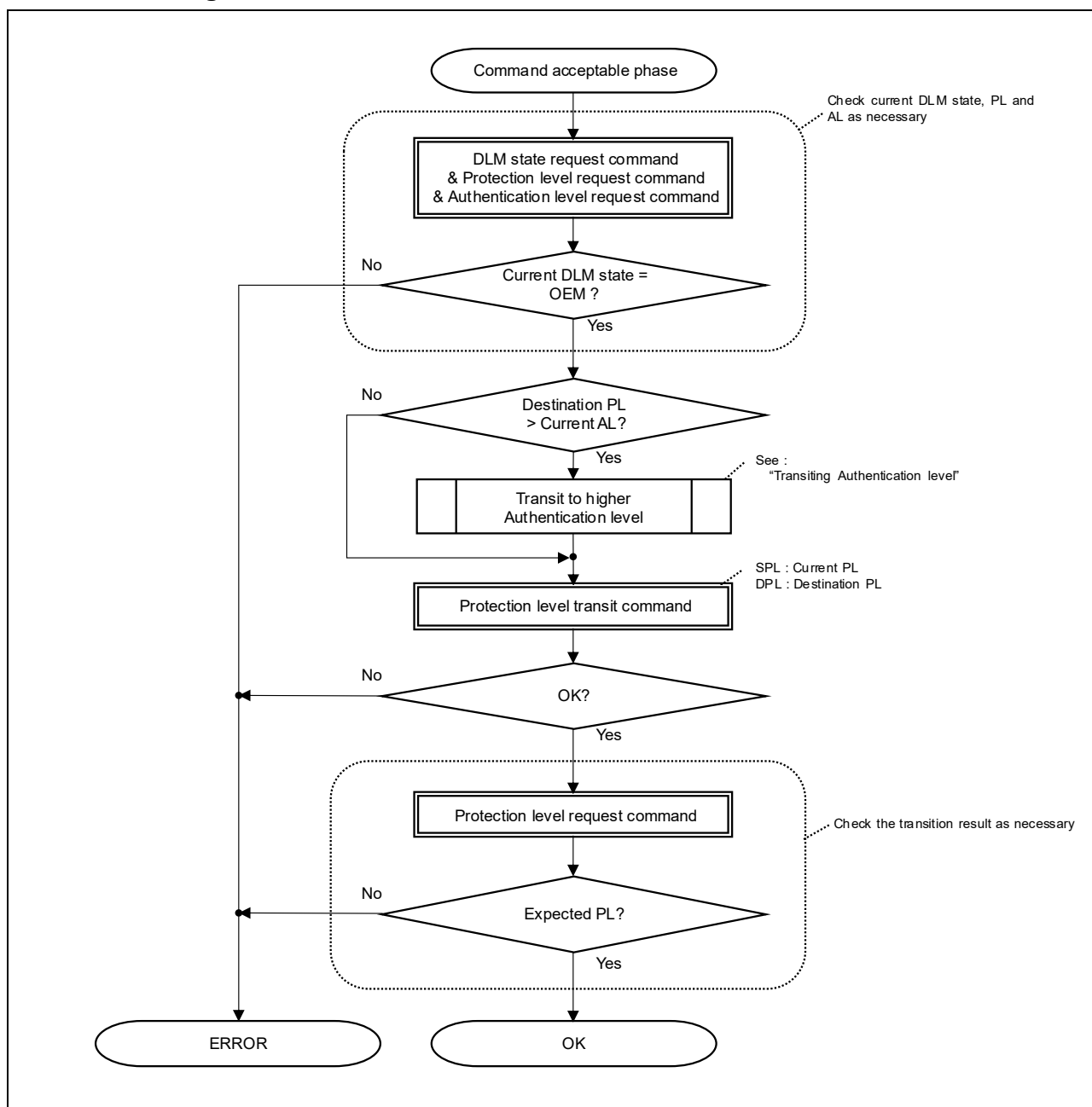


Figure 52. Transiting Protection Level

7.5 Transiting Authentication Level

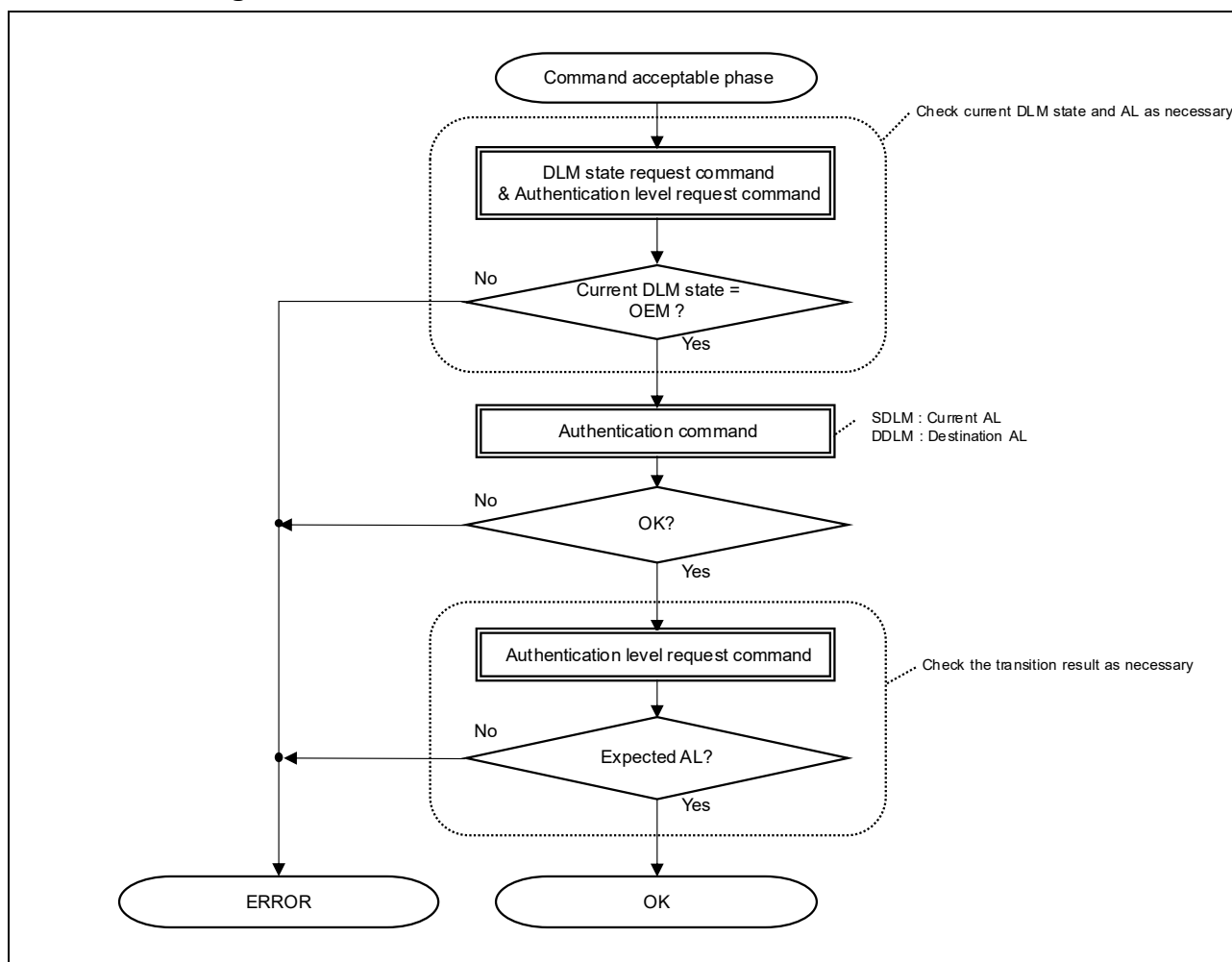


Figure 53. Transiting Authentication Level

7.6 Data Programming

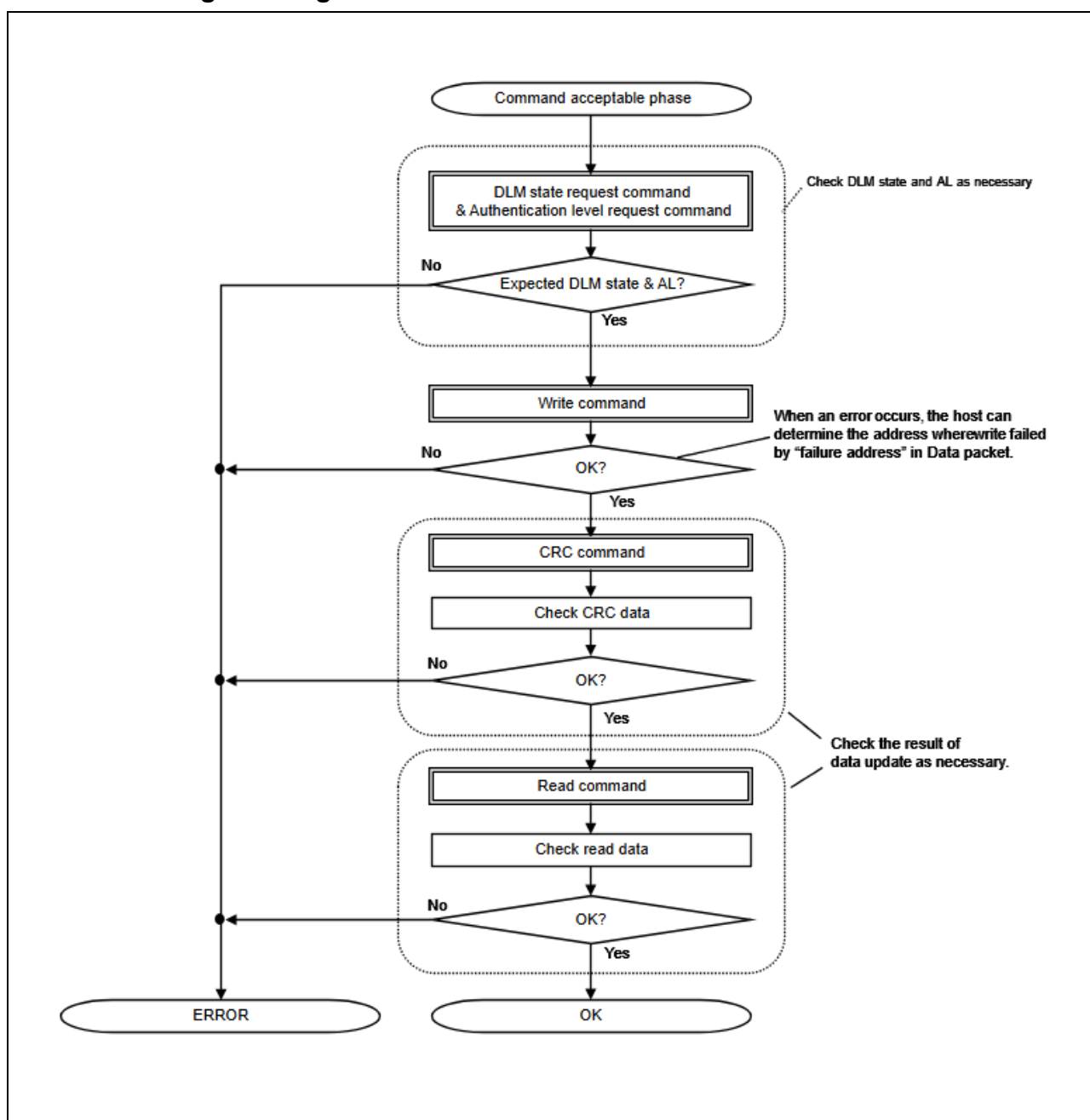


Figure 54. Data Programming

7.7 Encrypted Data Programming

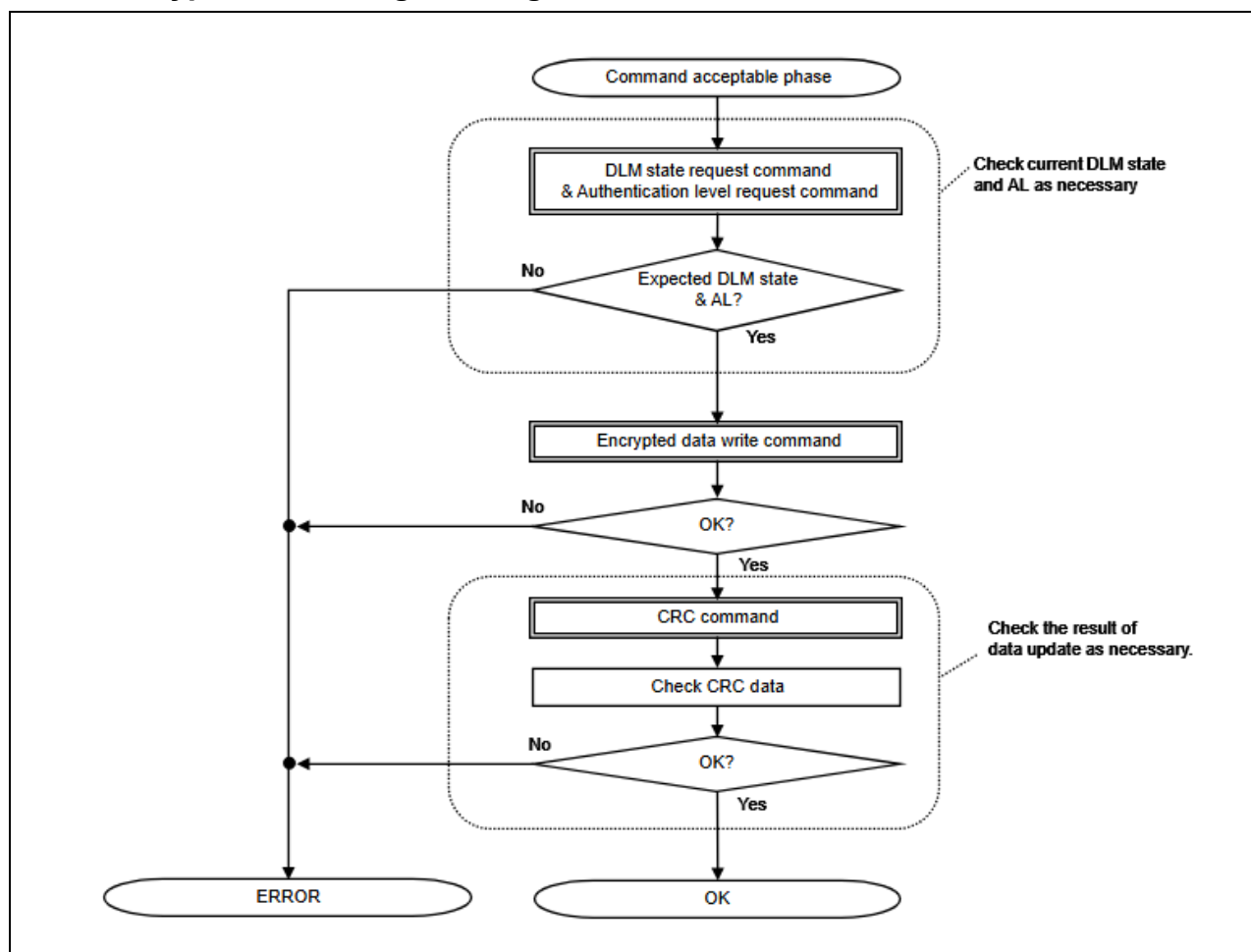


Figure 55. Encrypted Data Programming

7.8 Initialize Memory

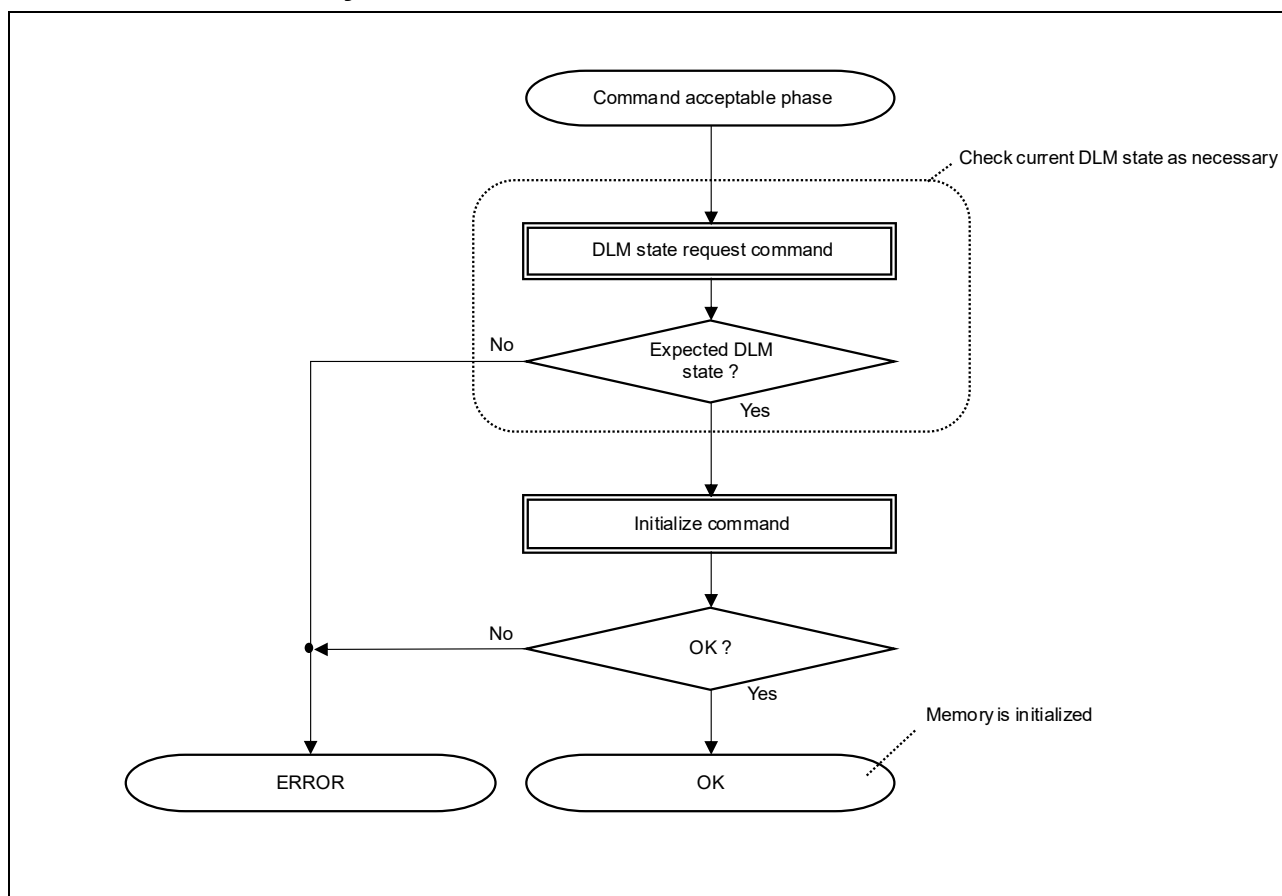


Figure 56. Initializing Memory

7.9 Storing Keys

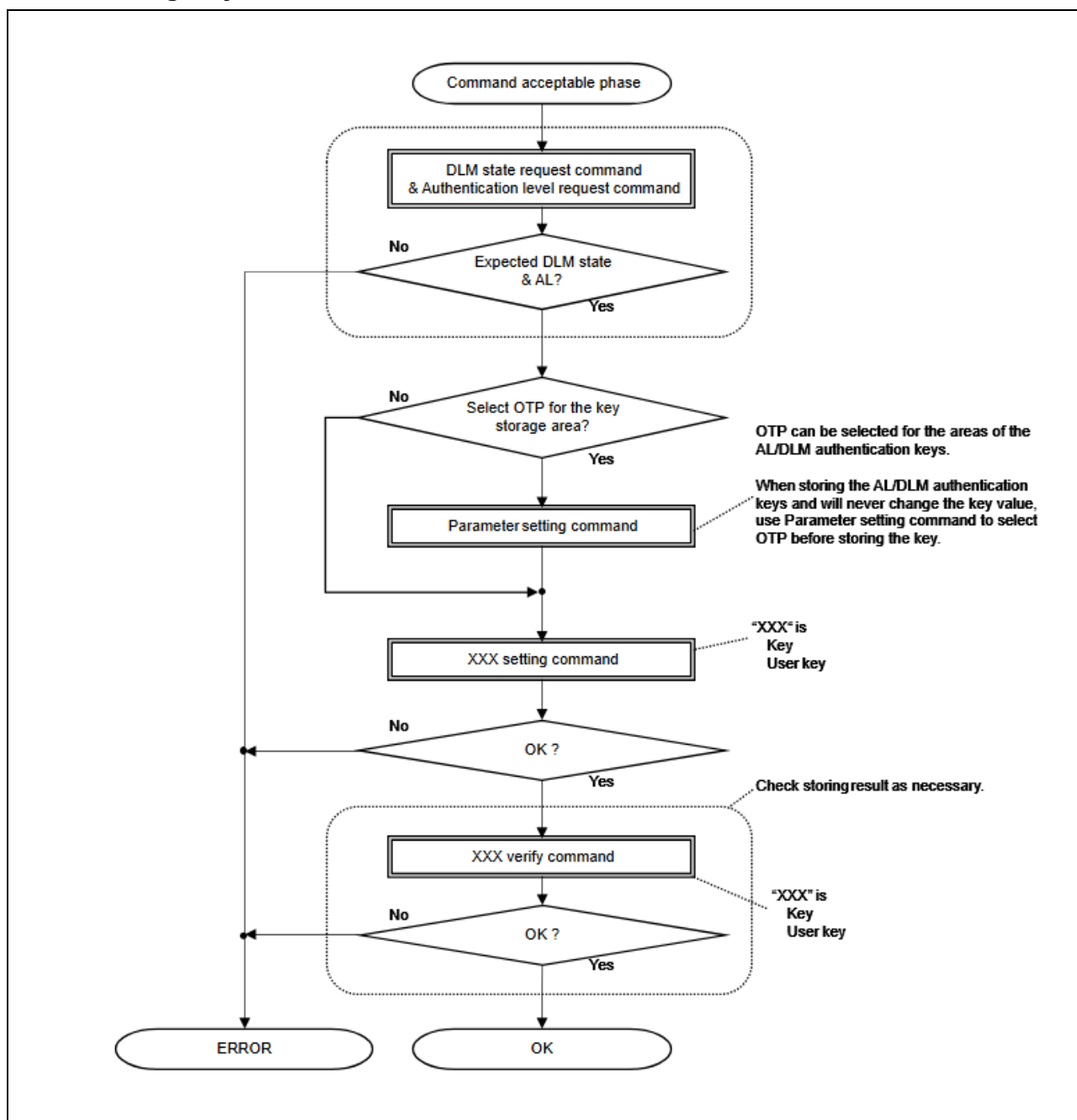


Figure 57. Storing Keys

7.10 Updating Boundary or Parameter Setting

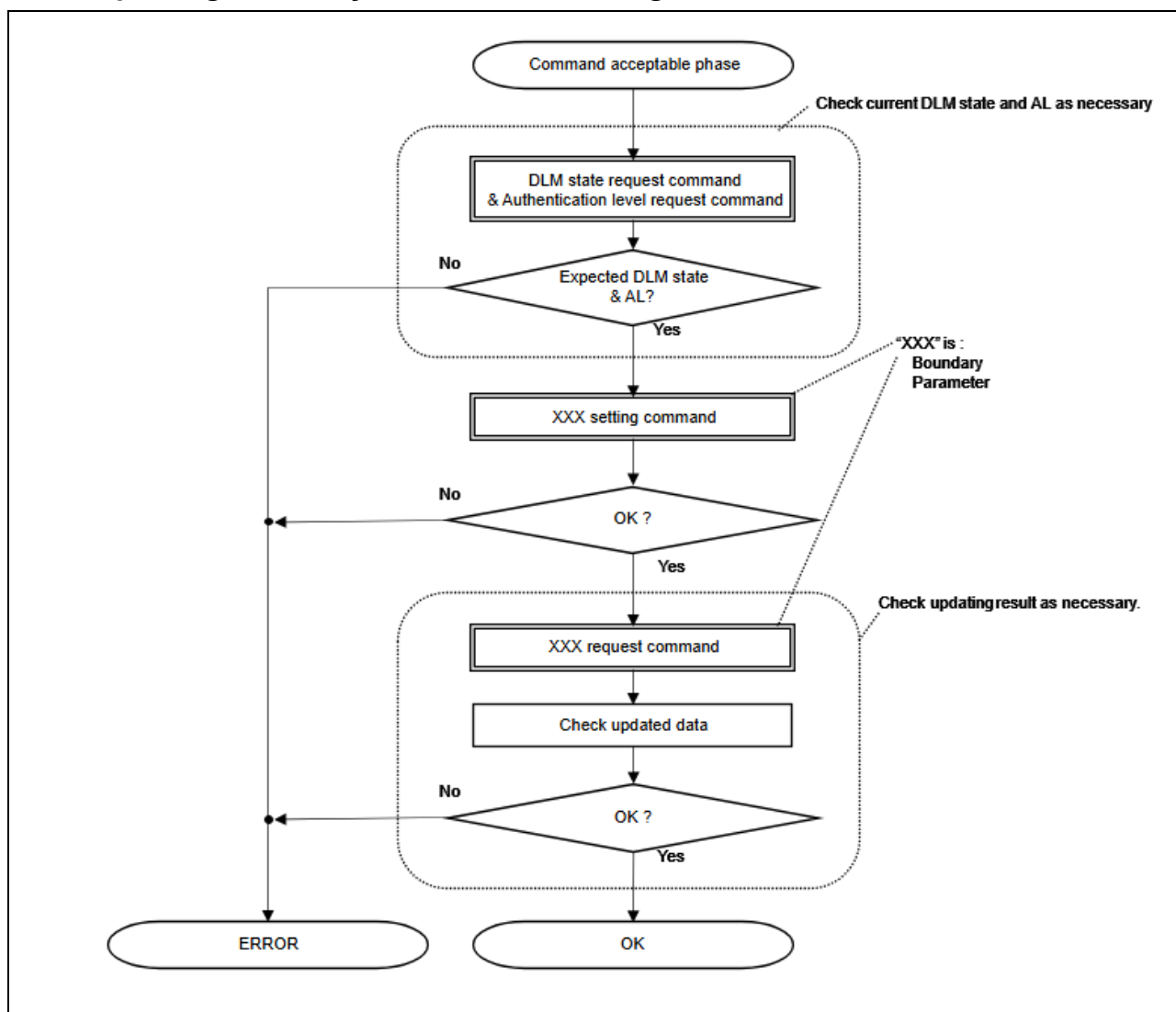


Figure 58. Updating Boundary or Parameter Setting

7.11 Storing Code Certificate

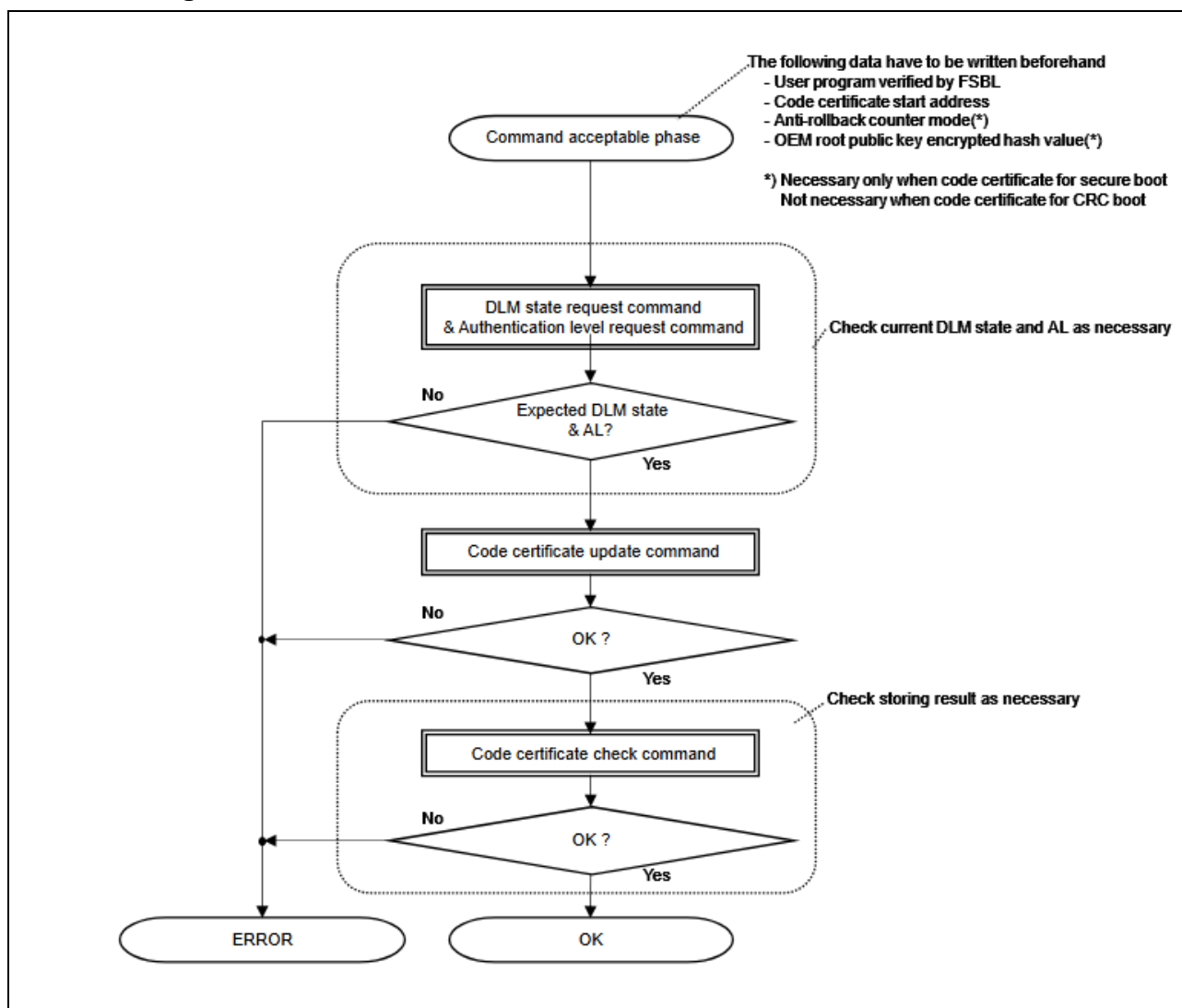


Figure 59. Storing Code Certificate

7.12 Downloading Whole Image

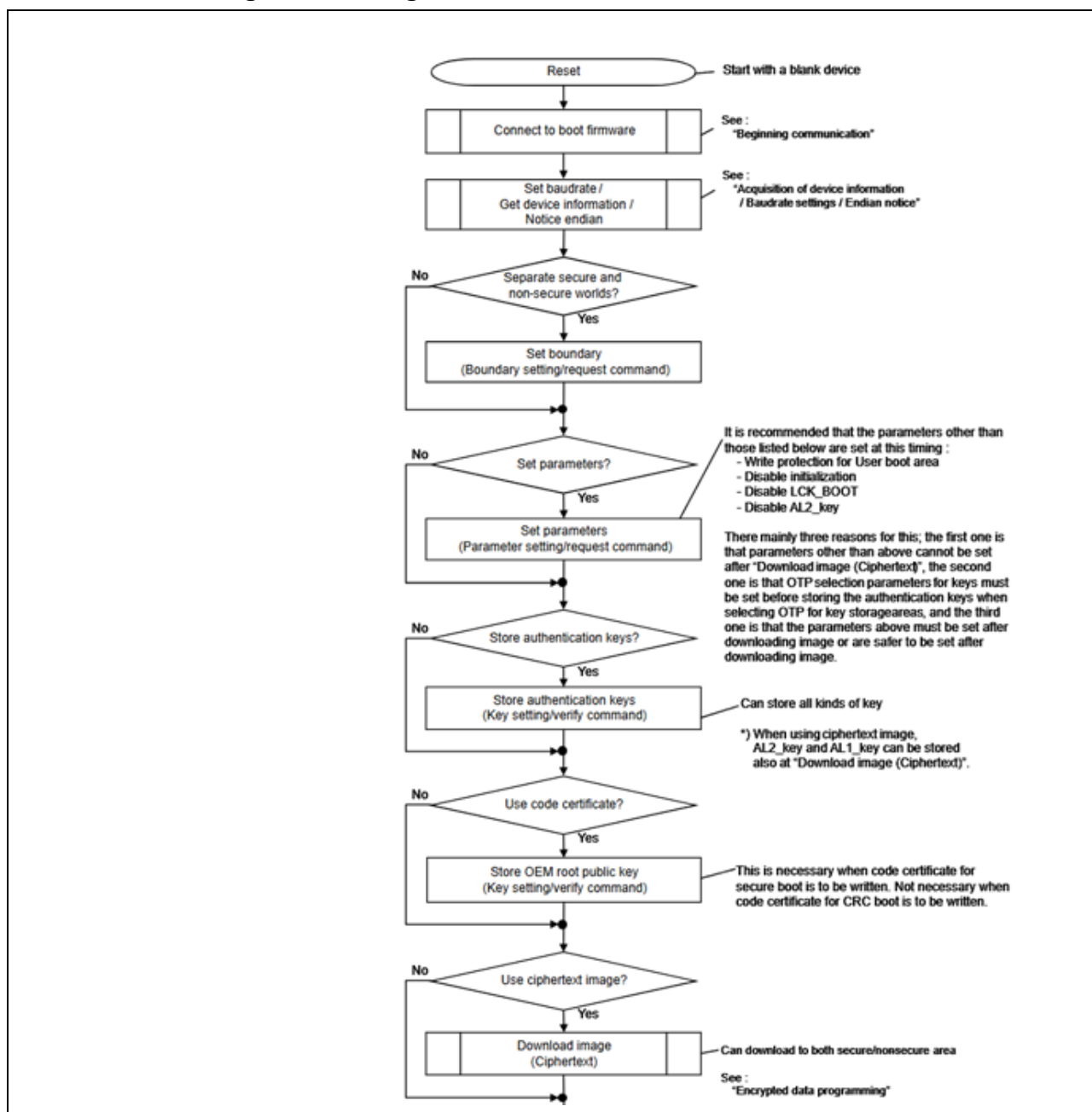


Figure 60. Downloading Whole Image (Part 1)

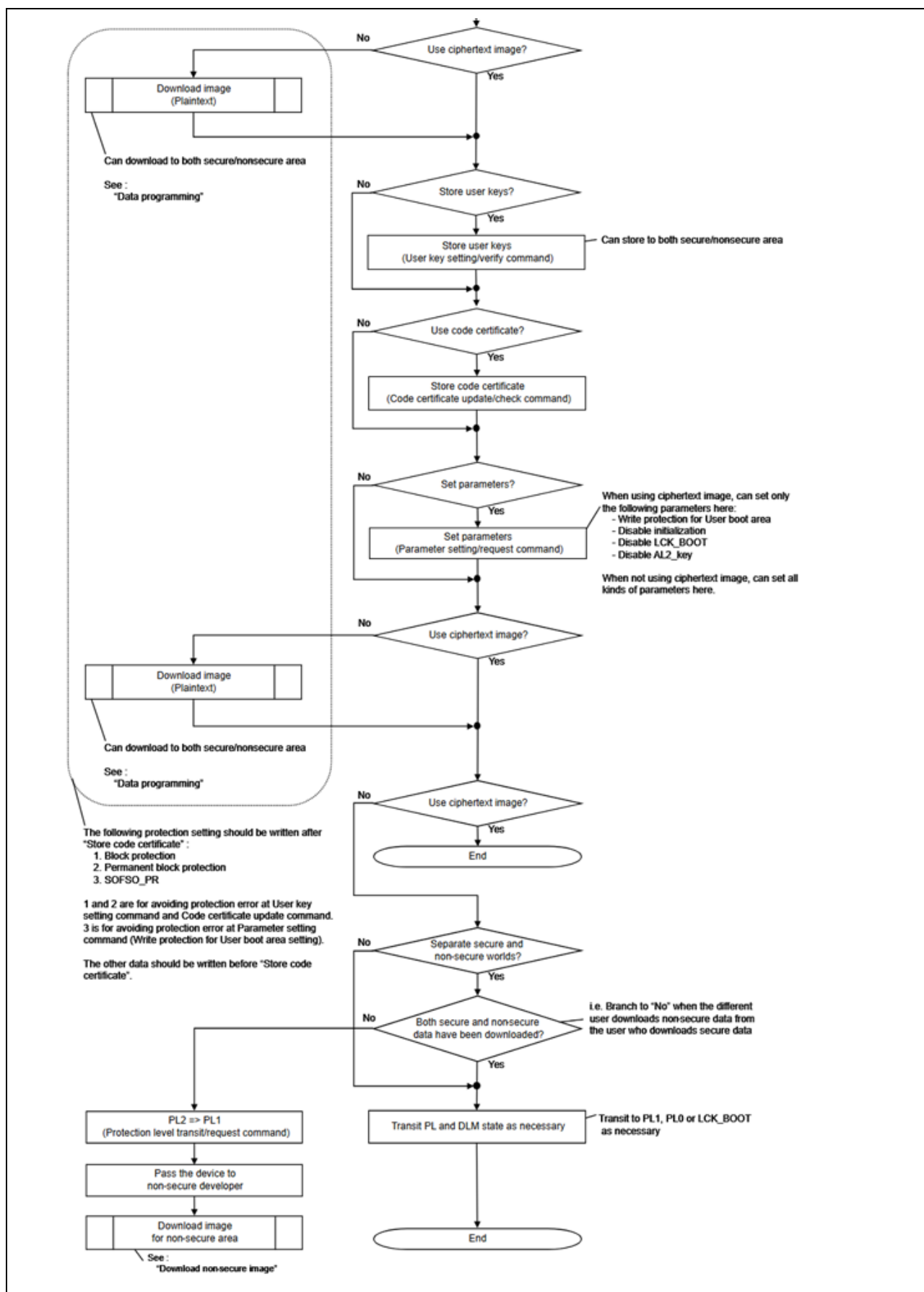


Figure 61. Downloading Whole Image (Part 2)

7.13 Downloading Non-secure Image

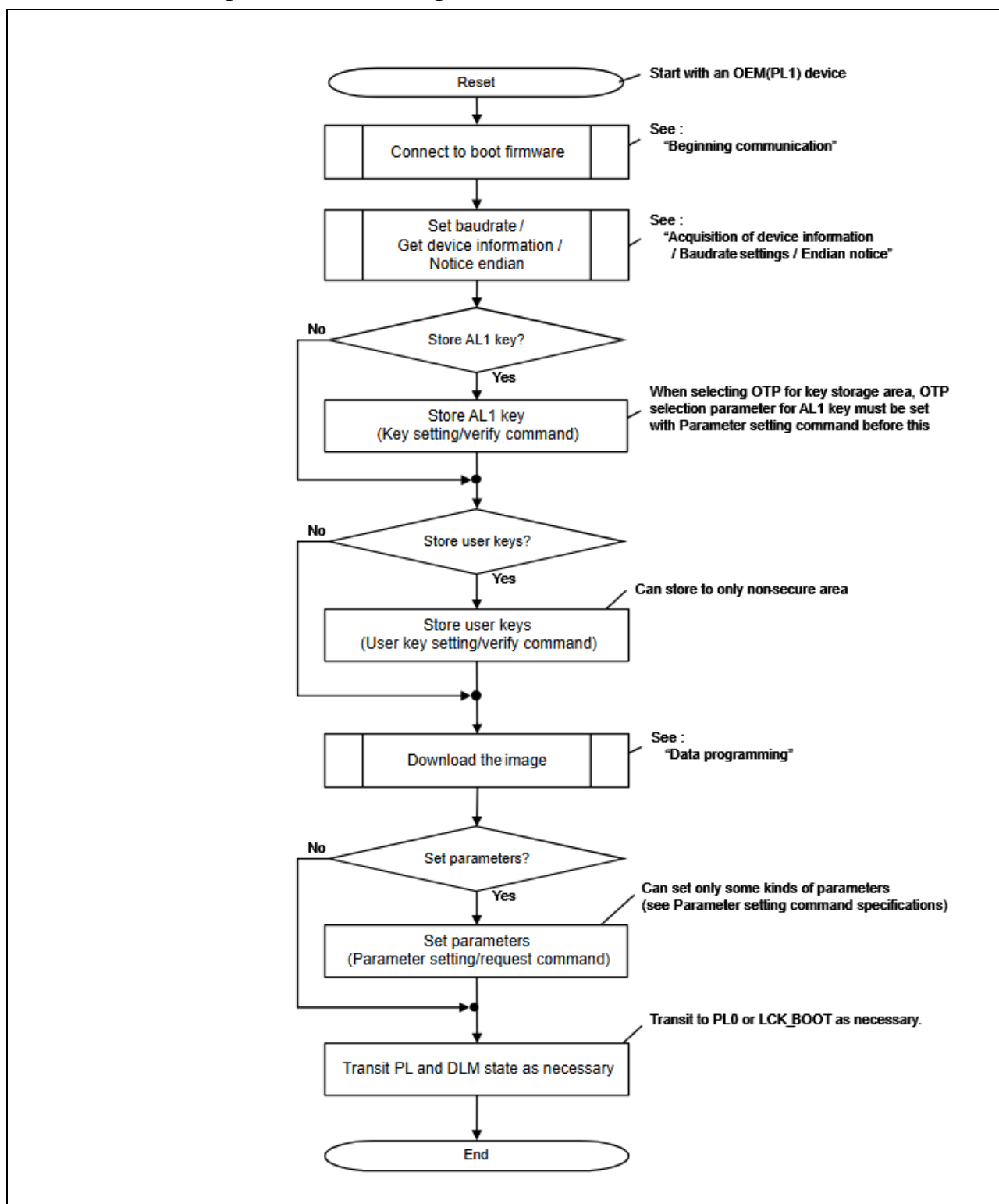


Figure 62. Downloading Non-secure Image

7.14 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.

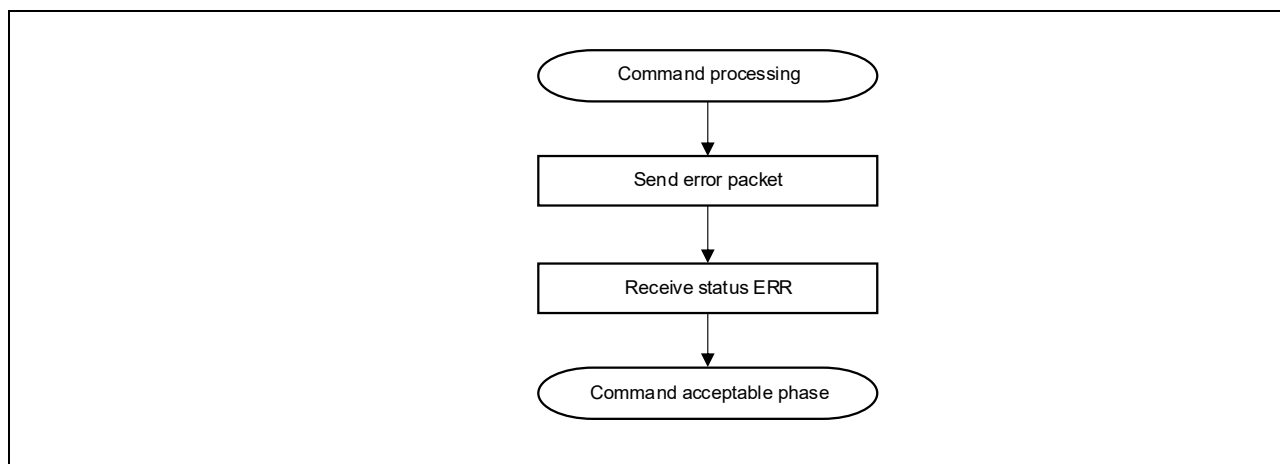


Figure 63. Command Cancel

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

Command	When to send error packets	Example of the error packet		
Authentication command	Data packet [Response value or Authentication code]	SOD	(1 byte)	81h
Key setting command	Data packet [key data]	LNH	(1 byte)	00h
User key setting command	Data packet [key data]	LNL	(1 byte)	01h
Write command	Data packet [write data]	RES	(1 byte)	FFh (ERR)
Read command	Data packet [status OK]	SUM	(1 byte)	00h
Code certificate update command	Data packet [Key/Code certificate data]	ETX	(1 byte)	03h
Encrypted data write command	Data packet [Key/encrypted user data]			

8. AC Characteristics

8.1.1 Communication Setting Phase

Parameter	Symbol	Min	Typ	Max	Unit
Initial response time for SCI boot	tIRT_SCI	-	-	3030	ms
Initial response time for USB boot	tIRT_USB	-	-	3000	ms

8.1.2 DLM State Transit Command

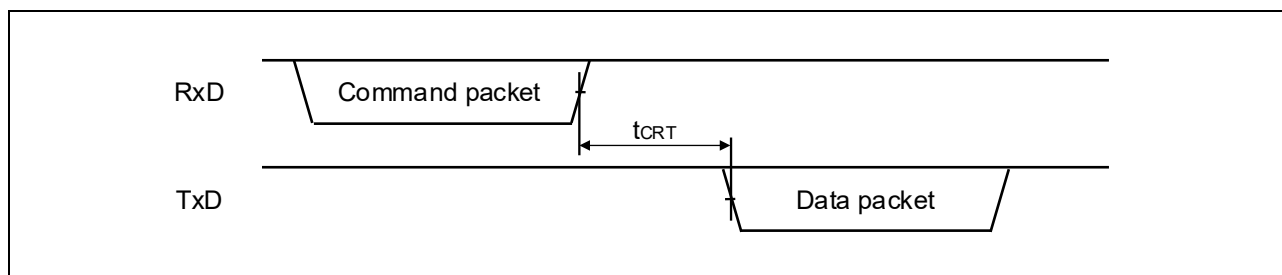


Figure 64. DLM State Transit Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s

8.1.3 DLM State Request Command

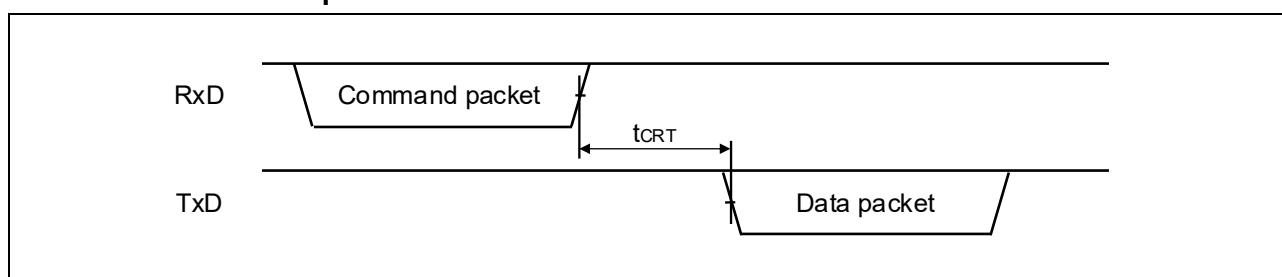


Figure 65. DLM State Request Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s

8.1.4 Protection Level Transit Command

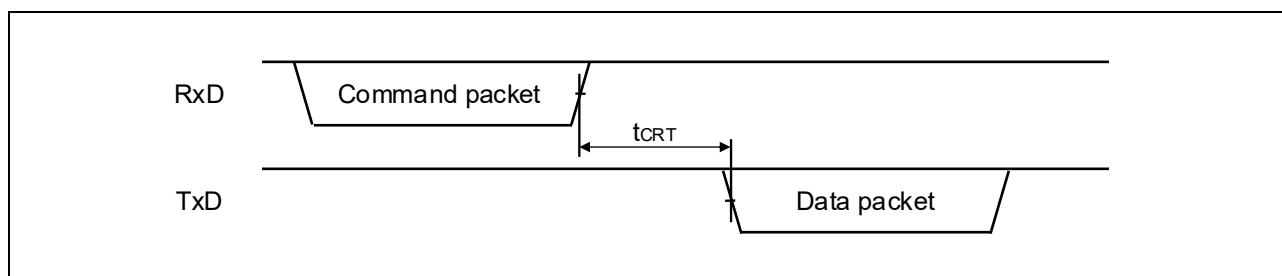


Figure 66. Protection Level Transit Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.5 Protection Level Request Command

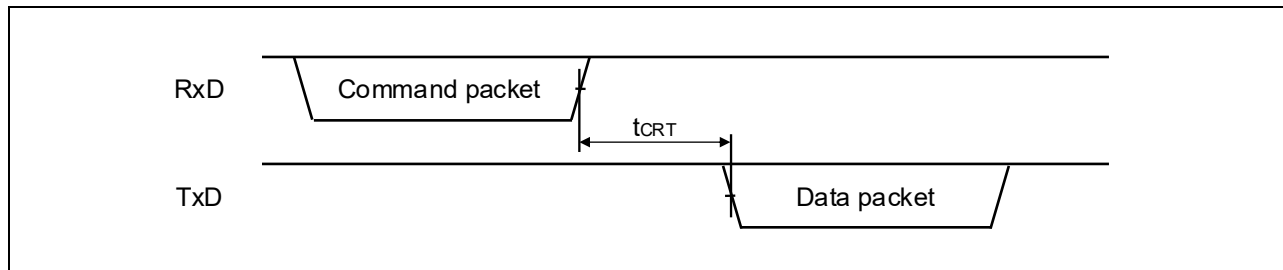


Figure 67. Protection Level Request Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.6 Authentication Level Request Command

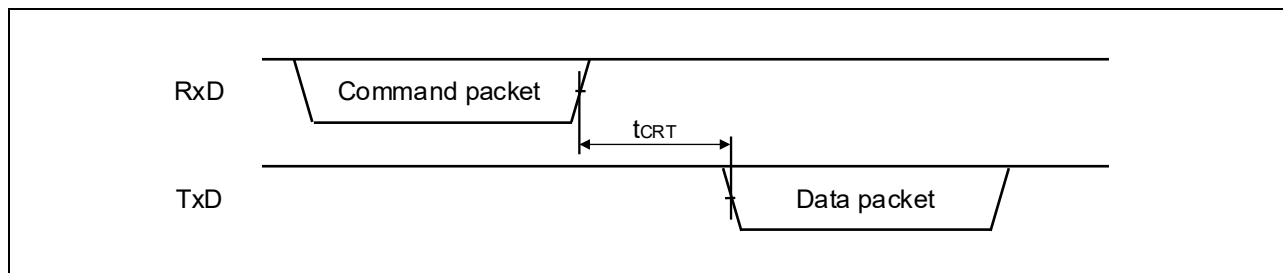


Figure 68. Authentication Level Request Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.7 Authentication Command

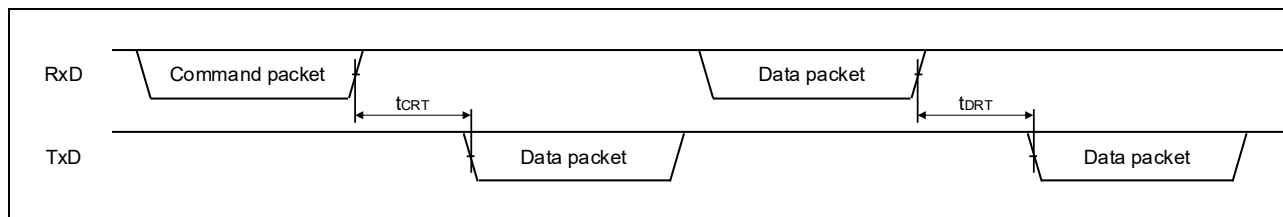


Figure 69. Authentication Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s
Data response time	tDRT	-	-	60	s

8.1.8 Key Setting Command

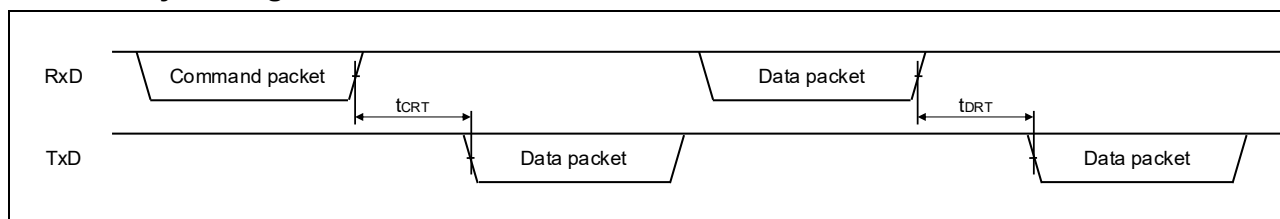


Figure 70. Key Setting Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s
Data response time	t_{DRT}	-	-	3	s

8.1.9 User Key Setting Command

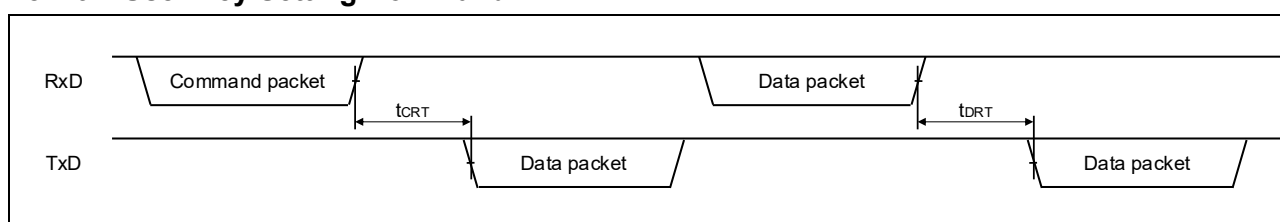


Figure 71. User Key Setting Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s
Data response time	t_{DRT}	-	-	60	s

8.1.10 Key Verify Command

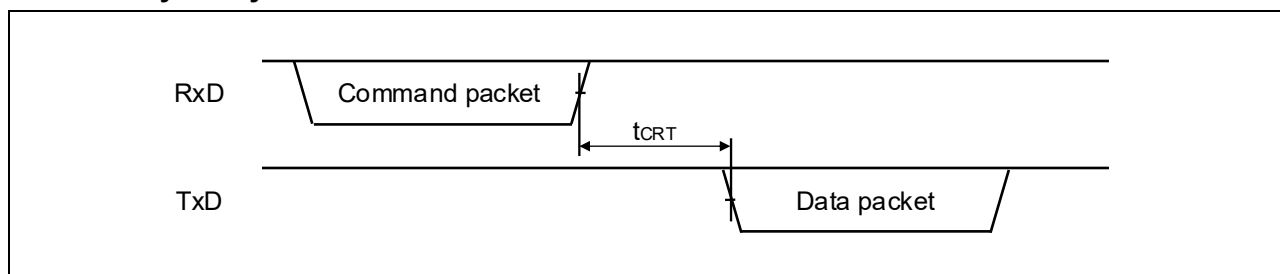


Figure 72. Key Verify Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s

8.1.11 User Key Verify Command

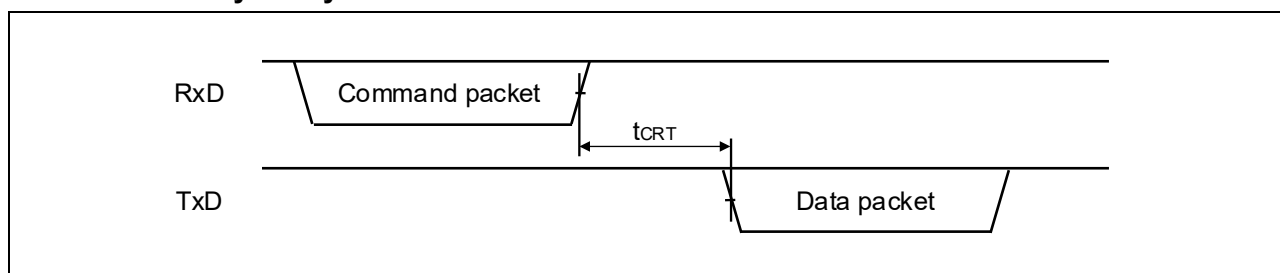


Figure 73. User Key Verify Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.12 Initialize Command

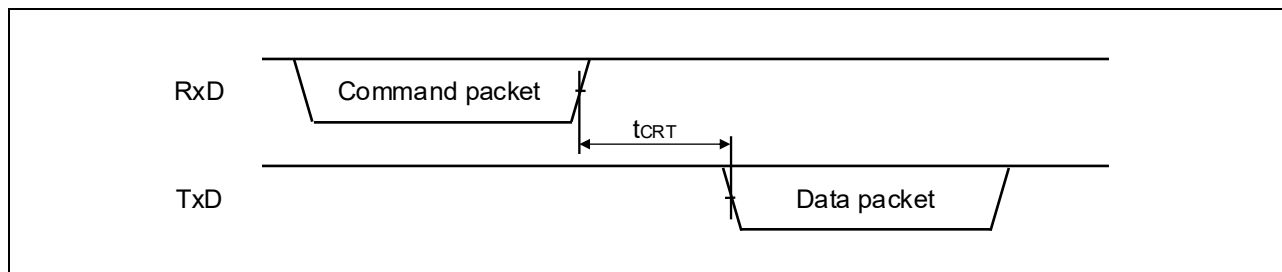


Figure 74. Initialize Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	60	s

8.1.13 Boundary Setting Command

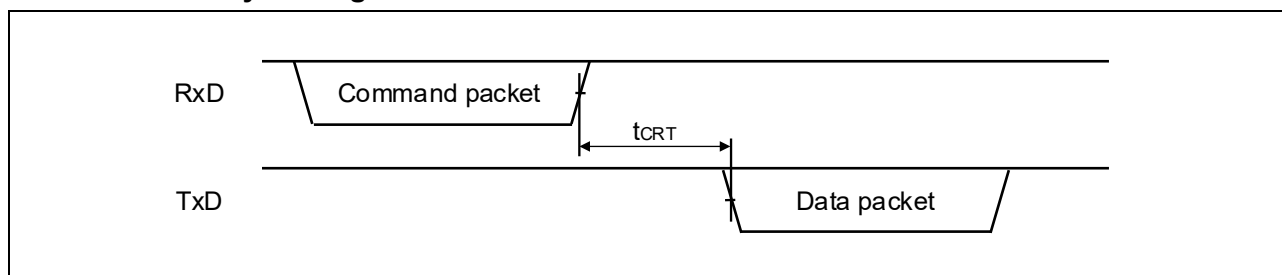
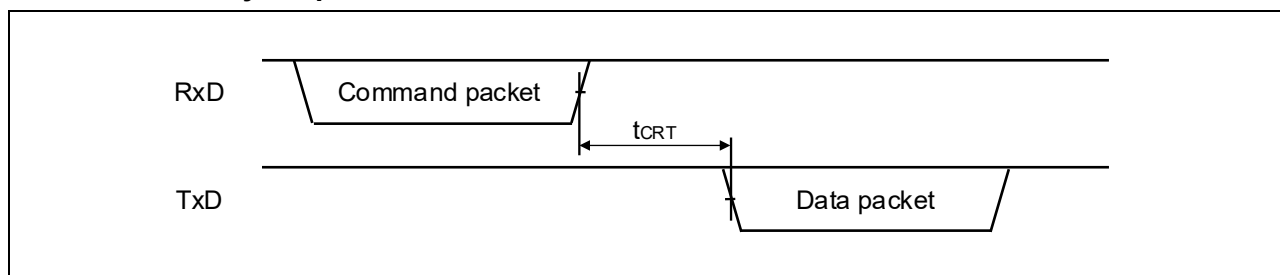
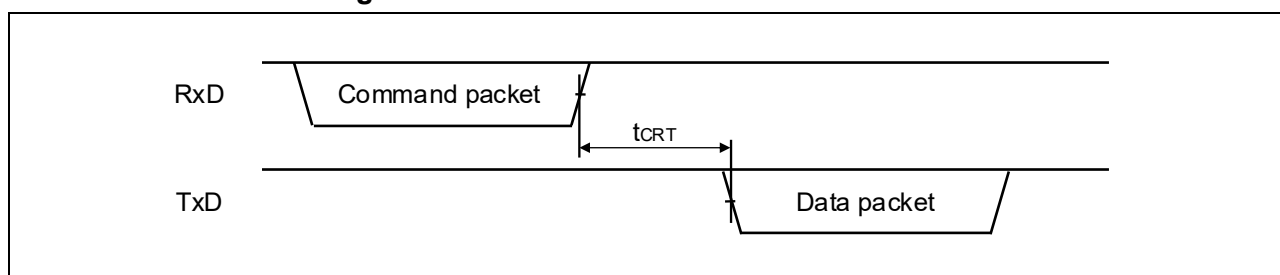


Figure 75. Boundary Setting Command

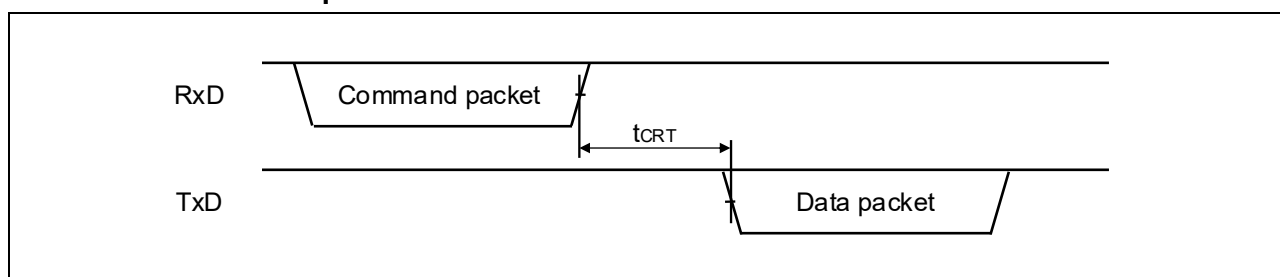
Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.14 Boundary Request Command**Figure 76. Boundary Request Command**

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s

8.1.15 Parameter Setting Command**Figure 77. Parameter Setting Command**

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s

8.1.16 Parameter Request Command**Figure 78. Parameter Request Command**

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.17 Inquiry Command

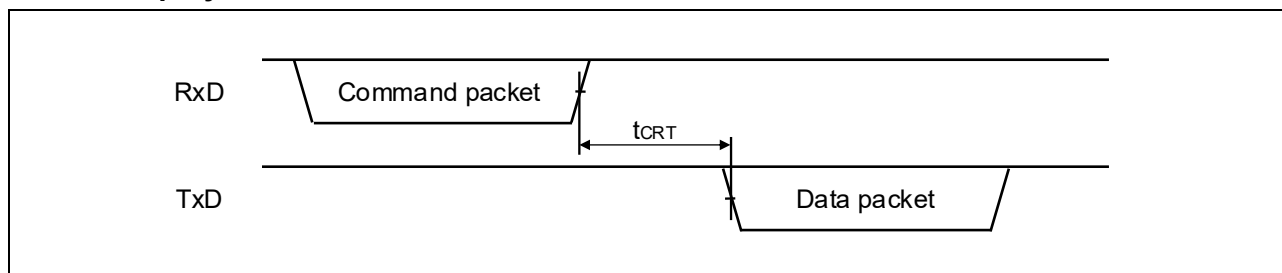


Figure 79. Inquiry Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.18 Signature Request Command

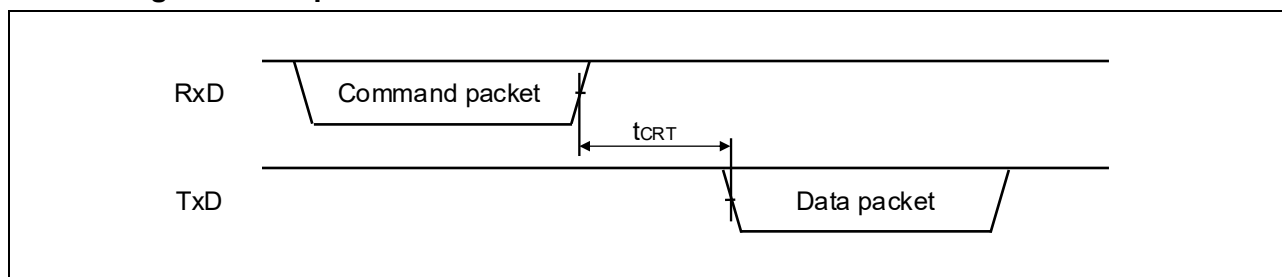


Figure 80. Signature Request Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.19 Area Information Request Command

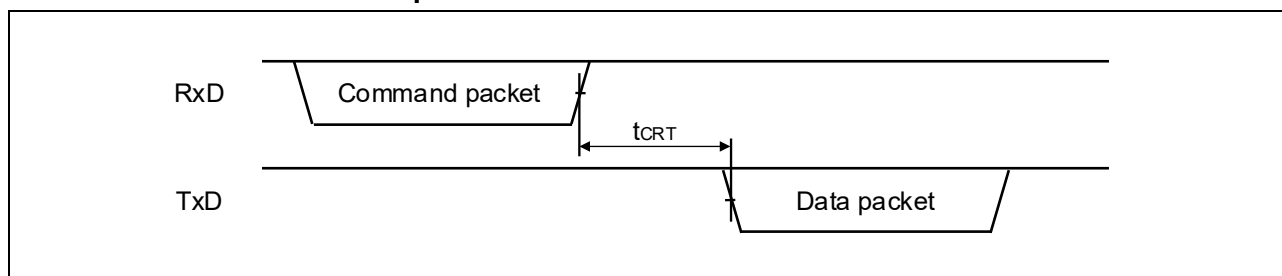


Figure 81. Area Information Request Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.20 Baudrate Setting Command

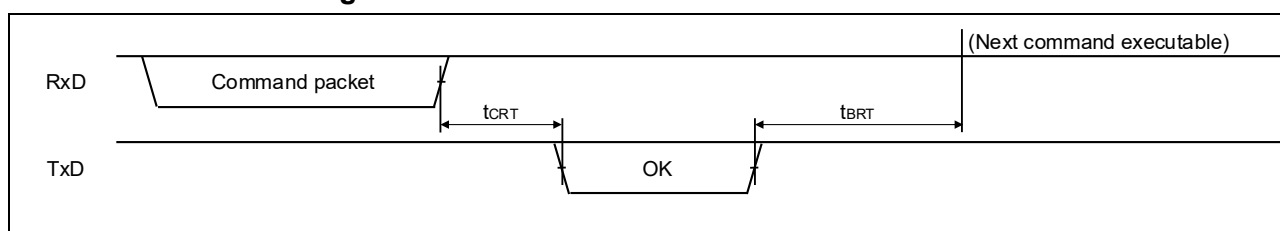


Figure 82. Baudrate Setting Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s
Baudrate setting time	t_{BRT}	-	-	1	ms

8.1.21 Endian Notice Command

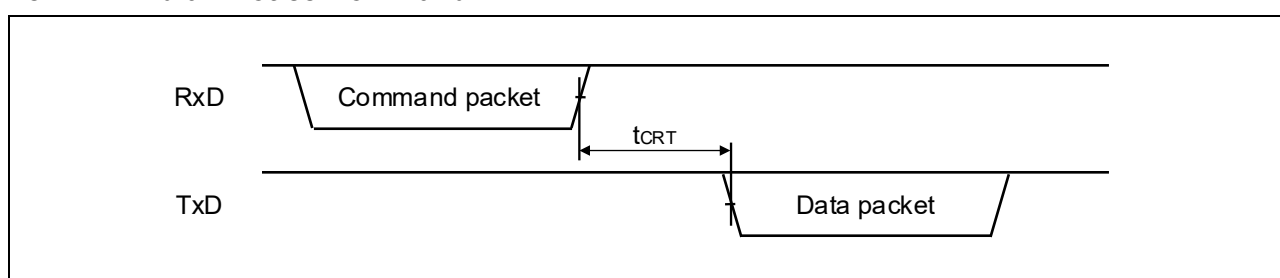


Figure 83. Area Information Request Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s

8.1.22 Write Command

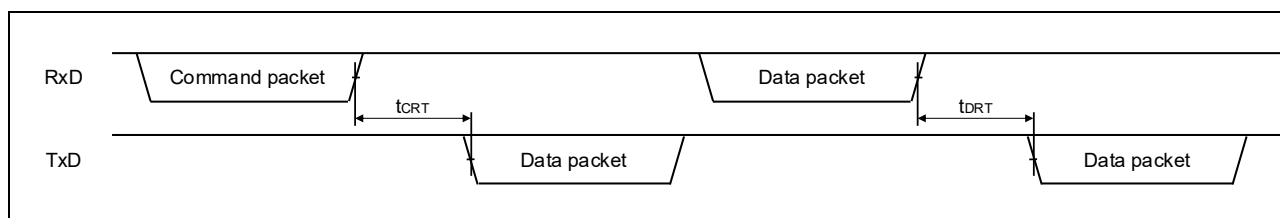


Figure 84. Write Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	t_{CRT}	-	-	3	s
Data response time	t_{DRT}	-	-	60	s

8.1.23 Read Command

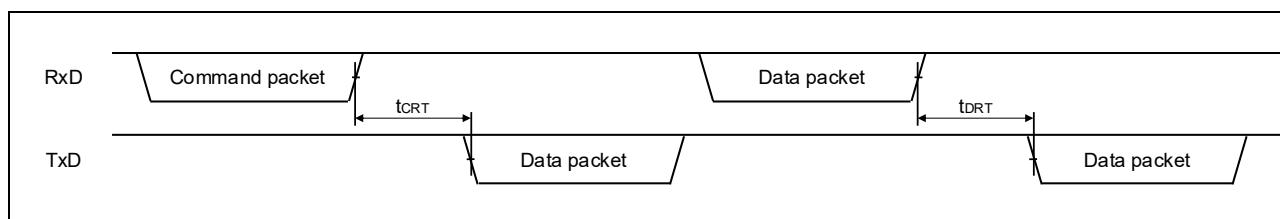


Figure 85. Read Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s
Data response time	tDRT	-	-	3	s

8.1.24 CRC Command

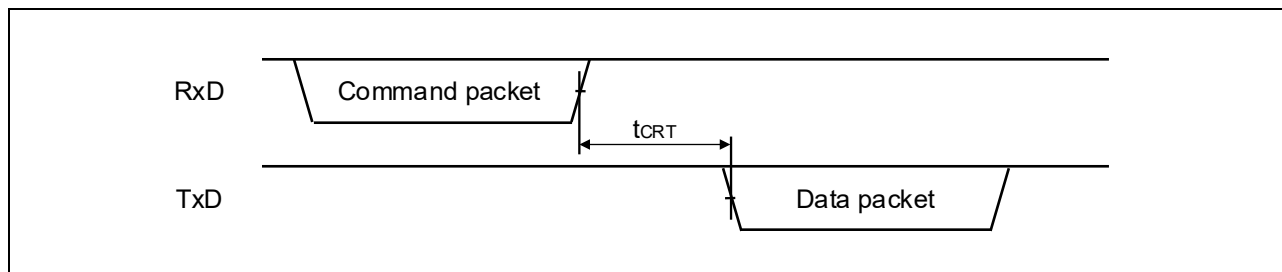


Figure 86. CRC Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.25 Code Certificate Update Command

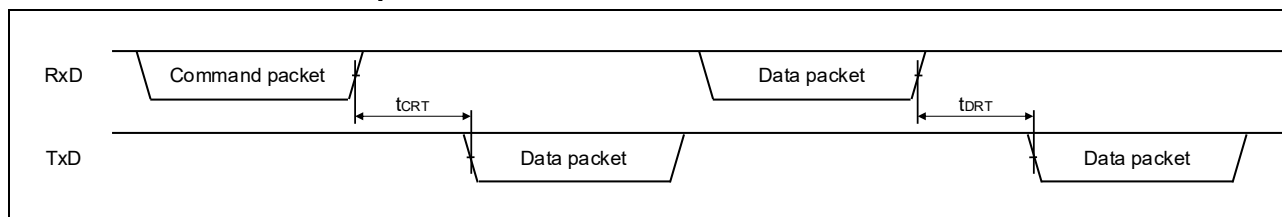


Figure 87. Code Certificate Update Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s
Data response time	tDRT	-	-	60	s

8.1.26 Code Certificate Check Command

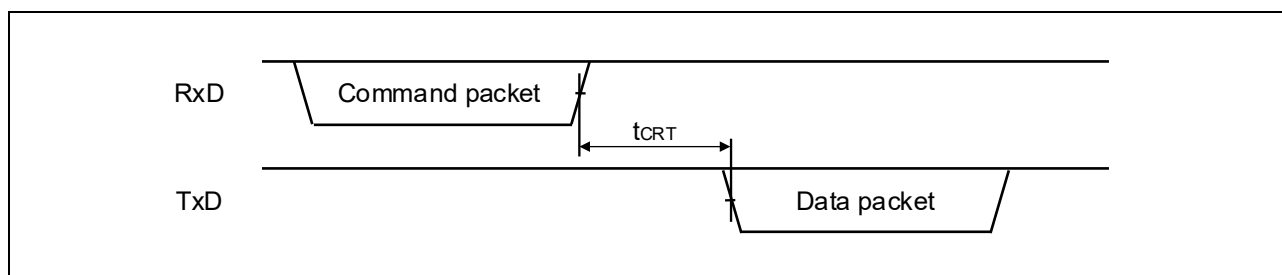


Figure 88. Code Certificate Check Command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	3	s

8.1.27 Encrypted Data Write Command

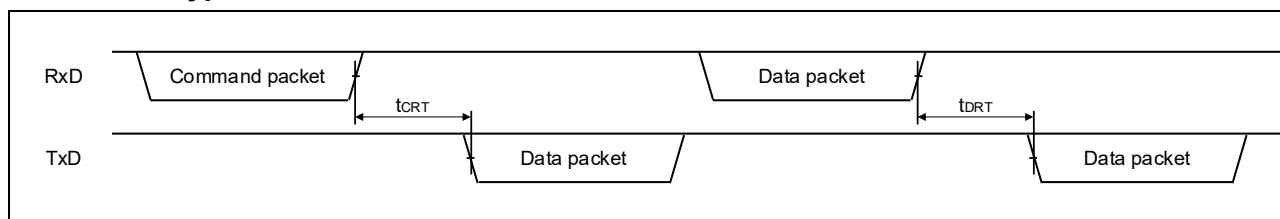


Figure 89. Encrypted Data Write Command

*) tDRT specifies the longest time among all the kinds of data packets of this command

Parameter	Symbol	Min	Typ	Max	Unit
Command response time	tCRT	-	-	60	s
Data response time	tDRT	-	-	60	s

9. Sequencer Command List

Table 23 shows the sequencer commands executed by each communication command.

Table 23. Sequencer Command List

Communication command	Sequencer command	Number of issue times
DLM state transit command	Configuration set	1 time
	Program	1 time
	Forced stop *) Use to clear error status.	1 time
Protection level transit command	Configuration set	1 time
	Forced stop *) Use to clear error status.	1 time
Authentication command	Program	Depends on the DDLM and the result of all erasure at RMA_REQ transition
	Configuration set	Depends on the DDLM and the result of all erasure at RMA_REQ transition
	Forced stop *) Use to clear error status	Depends on the DDLM and the result of all erasure at RMA_REQ transition
Key setting command	Program	3 time when KYTY = FDh 4 time when other KYTY
	Forced stop *) Use to clear error status	1 time
Initialize command	Program	[Size of User Extra MRAM / 16] + 100 times
	Configuration set	228 times
	Forced stop *) Use to clear error status	Depends on the result of all erasure
Boundary setting command	Configuration set	1 time
	Forced stop *) Use to clear error status	1 time
Parameter setting command	Program	1 time
	Forced stop *) Use to clear error status	1 time

Communication command	Sequencer command	Number of issue times
Write command	Program	Depends on designated address
	Configuration set	Depends on designated address
	Forced stop *) Use to clear error status	1 time
Code certificate update command	Program	Depends on Code certificate start address and size of Code certificate
	Read counter	Depends on current and new Anti-rollback counter value
	Increment counter	Depends on current and new Anti-rollback counter value
	Forced stop *) Use to clear error status	Depend on the result of writing "Code certificate" and "MAC value of Code certificate and OEM boot loader"
Encrypted data write command	Program	Depends on designated address
	Configuration set	Depends on designated address
	Forced stop *) Use to clear error status	1 time

10. Precaution List

10.1.1 Authentication command

1. This device has 4 storage areas each for authentication keys which are to transit to AL2, AL1 or RMA_REQ (AL2_KEYn, AL1_KEYn and RMA_KEYn where n=0~3).

Boot firmware searches a storage area that satisfies the both of followings from the 4 areas:

- Authentication key is stored in the area
- The stored authentication key is not revoked

Boot firmware performs this search in order from n=3, n=2, n=1 to n=0, and use the authentication key stored in the firstly found area for the authentication.

For example, when all the 4 keys are stored and also no one is revoked, only the key n=3 can be used for authentication.

To change the used authentication key to n=2 from this state, enable the revoke setting of key n=3 mapped in Config area.

10.1.2 Key setting command

1. Written hash value of OEM root public key cannot be rewritten because it is written to OTP area. Specify a Key storage ID which has not been written yet to change the value of OEM root public key.
2. Storage area of DLM/AL key's key index can be switched to OTP area. When the storage area has been switched to OTP area, written key index cannot be rewritten. Specify a Key storage ID which has not been written yet to change the value of DLM/AL key.

10.1.3 Initialize Command

1. The parameters that cannot be unset using Parameter setting command, cannot be unset by this command too.
2. Anti-rollback counter area is not initialized by this command.

10.1.4 Parameter setting command

1. Parameters that only 00h is specifiable as PRMT cannot be unset again once after set.
2. Parameters [Write protection for User Extra MRAM], hereafter referred to as "this parameter", can be changed to OTP data by the Memory type select setting mapped in Config area Config area (OTP

selectable area) 0, hereafter referred to as "the Memory type select".

Therefore it is possible to lock this parameter permanently by setting this parameter after setting the Memory type select.

The reverse setting order, namely setting the Memory type select while this parameter is set is prohibited.

10.1.5 Write Command

- When the following protection is set, the protected area cannot be rewritten. Therefore, rewrite the protected area before setting the following.
 - Permanent block protection
 - MOFSPSn_S
 - MOFSPS0_NS
 - SOFSPSn_S
 - SOFSPS0_NS
- The memory type select setting for User Extra MRAM is mapped in Config area.
 When changing User Extra MRAM to OTP, the memory type select setting must be written earlier than writing User Extra MRAM.
 Since there is no protection setting is mapped in Config area, it is recommended that writing to User Extra MRAM is performed after writing to all the Config areas are done.

Similarly, there are memory type select settings for Config area are mapped in also Config area.

Basically there is no need to pay attention to the writing order because the memory type select setting is written first when writing in the order of the address. However, for example when writing to the area multiple times separately, please be careful not to write in reverse order.

- When writing to Data area, the block protection settings for the write target area must be disabled beforehand.
- OTP area cannot be written back to 1 once after 0 is written.

For OTP area which has ECC, also ECC bits cannot be written back from 0 to 1. Therefore note that re-writing is not possible even when the data bits are to be changed only from 1 to 0. However when the following cases re-writing is possible as an exception:

- Re-writing the same value.
 - The data already written is all "F".
- When MRAM sequencer error occurs when writing Config area and the returned ST2 value is CFGSETERR=1b, an ECC error that cannot be corrected would have been occurred.

Please resolve the ECC error using one of the following ways:

- Execute Initialize command.
- Identify the address where the error occurred by the returned ADR value and rewrite the identified address with the correct value.

10.1.6 Code Certificate Update Command

- Use this command after writing the following data in advance.

Data	Area to write	Command used for write	Remarks
User program verified by FSBL	User area, User Extra MRAM	Write command, Encrypted data write command	–
Code certificate start address	Config area		
Anti-rollback counter mode			Not necessary for CRC boot

Data	Area to write	Command used for write	Remarks
OEM root public key encrypted Hash value	Dedicated area	Key setting command	

2. Verification fails if data of received Key certificate or Code certificate does not conform to device specifications.
Refer to the user's manual of the device for certificates' specifications.
3. Key certificate is not necessary when MAC type = None. Specify KCS = 0 and do not send any data as Key certificate data in this case.

10.1.7 Encrypted Data Write Command

1. This command becomes inexecutable once permanent block protection is set.
2. This command becomes inexecutable once after write protection for User Extra MRAM is set after changed to OTP.
See specifications of Parameter setting command for details of write protection for User Extra MRAM.
3. This command becomes inexecutable once after User Boot start-up is enabled by setting UMRECR.UMREEN = 0b.
4. This command becomes inexecutable if any of the followings is set and also MOFSPSn_S or MOFSPS0_NS is set:
 - SAS2.BTFLG = 0b
 - BANKSEL_NS.BANKSWP[2:0] ≠ 111b (only for dual mode supported devices)
 - BANKSEL_S.BANKSWP[2:0] ≠ 111b (only for dual mode supported devices)
5. Do not set permanent block protection of the area where user keys are to be written when User key setting command is used.
Do not set permanent block protection of the area where Code certificate is to be written when Code certificate update command is used.
In addition, for both commands, when the area to be written is Data area, do not set not only permanent block protection but also block protection by this command.
If they are set, both commands become inexecutable due to Protection error.
6. Do not write 0b to bit56 of SOFSPSn_S when Write protection for User Extra MRAM will be enabled.
If 0b is written, Write protection for User Extra MRAM becomes unchangeable due to Protection error.
*) Write protection for User Extra MRAM can be set with Parameter setting command.
7. If the protection settings such as the followings are written before the protected area, this command abnormally finishes at writing of the protected area:
 - Permanent block protection
 - Block protection for Data area
 - MOFSPSn_S
 - MOFSPS0_NS
 - SOFSPSn_S
 - SOFSPS0_NS

To avoid this, Data packet [encrypted user data] for the protected areas must be sent earlier than ones for protection settings above.

8. The memory type select setting for User Extra MRAM is mapped in Config area.
Therefore when changing User Extra MRAM to OTP, Data packet [encrypted user data] for the memory type select setting must be sent earlier than ones for User Extra MRAM.
Since there is no protection setting is mapped in Config area, it is recommended that writing to User Extra MRAM is performed after writing to all the Config areas are done.

Similarly, there are memory type select settings for Config area are mapped in also Config area.
Basically there is no need to pay attention to the writing order because the memory type select setting is written first when writing in the order of the address.

9. OTP area cannot be written back to 1 once after 0 is written.

For OTP area which has ECC, also ECC bits cannot be written back from 0 to 1. Therefore note that re-writing is not possible even when the data bits are to be changed only from 1 to 0. However when the following cases re-writing is possible as an exception:

- Re-writing the same value.
- The data already written is all "F".

10. When MRAM sequencer error occurs when writing Config area and the returned ST2 value is CFGPRGERR=1b, an ECC error that cannot be corrected would have been occurred.
In this case, please initialize the memory area using Initialize command to resolve the ECC error.

11. Causes for Operation Stop

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

11.1 Initialization Phase

- When an exception of the CPU core occur.
- When Trusted system goes into an abnormal state.

11.2 Communication Setting Phase

- When the USB cable is disconnected after USB enumeration is completed
- When an exception of the CPU core occur.

11.3 Command Acceptable Phase

- When the USB cable is disconnected when the USB status is "Configured"
- When an exception of the CPU core occur.

11.4 DLM State Transit Command

- When transition to LCK_BOOT is complete.
- When transition to RMA_RET is complete.
- When Hardware error occurred.

11.5 Authentication Command

- When Trusted system goes into an abnormal state.
- When DLM state transition is complete.
- When Hardware error occurred.

11.6 Key Setting Command

- When Trusted system goes into an abnormal state.

11.7 User Key Setting Command

- When Trusted system goes into an abnormal state.

11.8 Key Verify Command

- When Trusted system goes into an abnormal state.

11.9 User Key Verify Command

- When Trusted system goes into an abnormal state.

11.10 Initialize Command

- When the command completes successfully.

11.11 Code certificate update command

- When Trusted system goes into an abnormal state.
- When the address of the verified program pointed to by the Code certificate is invalid.

11.12 Code Certificate Check Command

- When Trusted system goes into an abnormal state.
- When the address of the verified program pointed to by the Code certificate is invalid.
- When the value of the Code certificate start address is invalid.
- When the Code certificate check command is executed even though the Code certificate update command has not completed normally.

11.13 Encrypted Data Write Command

- When the Trusted system goes into an abnormal state.
- When Hardware error occurred.

12. Causes for Software Reset

Boot firmware performs software reset in the following cases:

12.1 Communication Setting Phase

- When all the conditions below are met:
 - MD=Hi is detected while boot firmware waits for 3-byte 00h data
 - Neither JTAG/SWD boot nor FINE boot
- When all the conditions below are met:
 - 3 seconds elapsed while boot firmware waits for Generic code
 - FINE boot

-No software reset occurs during command execution.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug.31.26	1.	First release document

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/.