RISC-V MCUs

Standard Boot Firmware for R9A02G021 RISC-V MCUs

Introduction

This application note describes the communication protocol, command set, and usage of the boot firmware provided with Renesas R9A02G021 RISC-V MCUs.

Target Device

R9A02G021 RISC-V MCUs

Contents

1. Terminology

This section defines the terminology used in this document.

1.1 Boot Firmware

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

1.2 Flash Memory

The following areas are collectively called flash memory:

- Code Flash: The ROM area where program code and configuration data is written.
  - User area: Code Flash area where the user's program code is written.
  - Config area: Code Flash area where configuration data is written.
- Data Flash: The ROM area where data is written.
  - Data area: Data Flash area where the user's data is written.

The boot firmware rewrites and reads the User area, Data area, and Config area according to commands given by the user.

Figure 1 shows an example of flash memory structure. Memory structure differs from device to device.

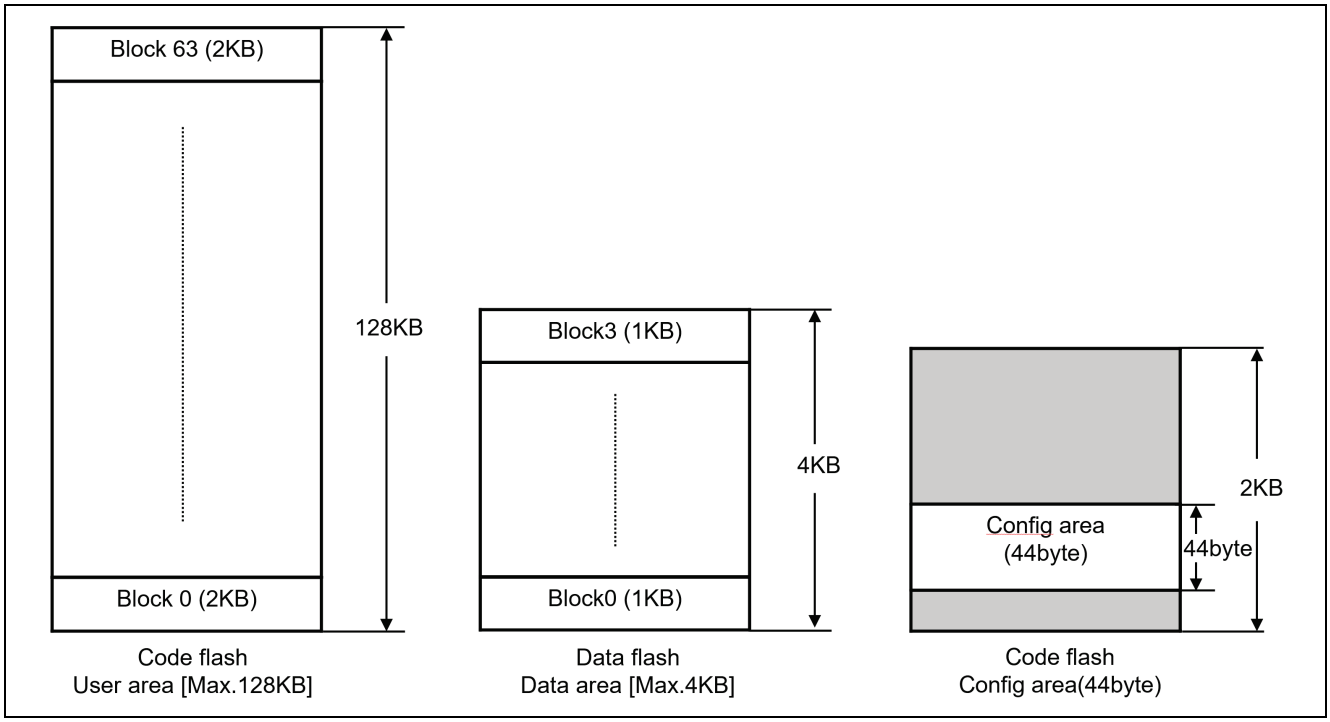

Figure 1. Flash Memory Structure Example 1

1.3 Access Window

The function for assignment of accessible sectors to erase, write or read in code flash is called Access Window (AW). This function allows access from start sector to end sector, and disallows access to other area. User set a value corresponding to “start sector” to the FAWS, also set a value corresponding to “end sector + 1” to the FAWE. If FAPR is 0, FAWS and FAWE cannot be changed. For details, please refer to the Flash Memory User's Manual.

Figure 2 and Figure 3 show an example of flash memory structure.

Sector No.	Base address	Size	Setting value
0	00000000h	2KB	000h
1	00000800h	2KB	001h
:	:	:	:
22	0000B000h	2KB	016h
23	0000B800h	2KB	017h

Figure 2. Flash Memory Structure Example 2A

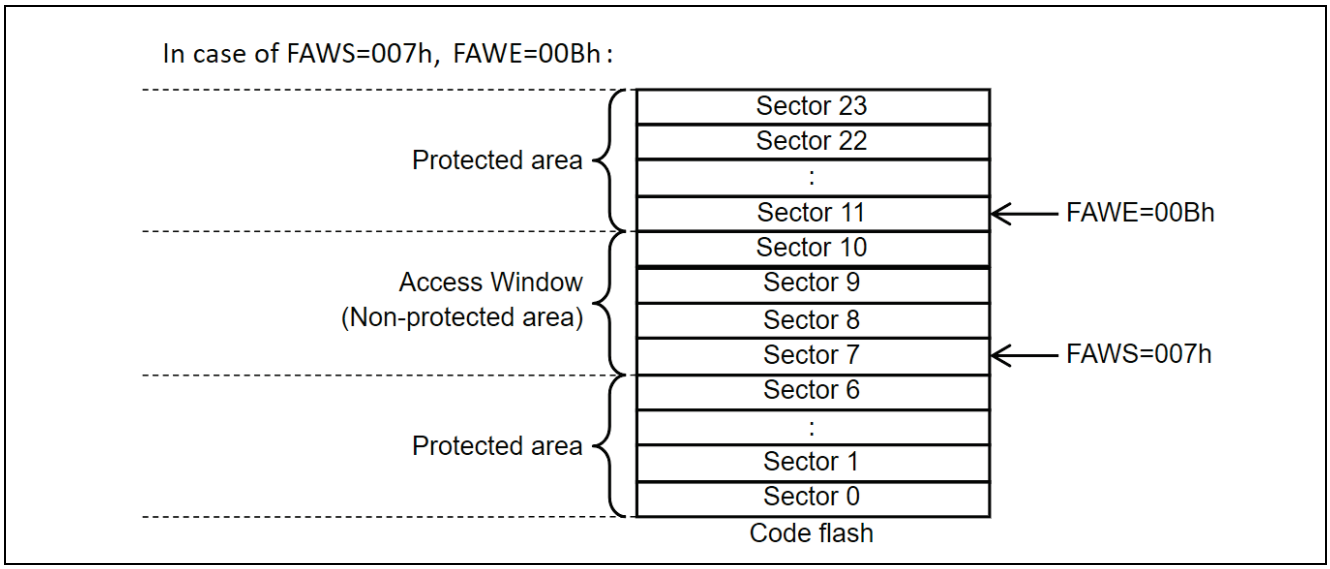

Figure 3. Flash Memory Structure Example 2B

2. System Architecture

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

2.1 R9A02G021 RISC-V MCU

This chapter describes the system architecture of R9A02G021 RISC-V MCU group regarding the flash memory control.

Table 1. Operating Environment

CPU core	Renesas RISC-V
Max CPU Operating Frequency	48 MHz Boot firmware operating frequency: • High-speed mode: 24 MHz • Middle-speed mode：3 MHz
Main Oscillator	Unnecessary
Operating Voltage	VCC = 1.6 to 5.5 V
Operating Mode	SAU boot mode • 5.5 V ≧ VCC ＞ 1.8 V：High-speed mode • 1.8 V ≧ VCC ≧ 1.6 V：Middle-speed mode
Flash Memory	Code Flash / User area / Max. 128 KB Code Flash / User boot area / None Data Flash / Data area / Max. 4 KB Data Flash / Extended data area / None
RAM	SRAM with ECC: 4 KB SRAM without ECC: 12 KB（used by the Boot firmware : within 8 KB）
Communication Method	[2-wire UART communication] (Initial / Min) 9600 bps (Max) 1.5 Mbps

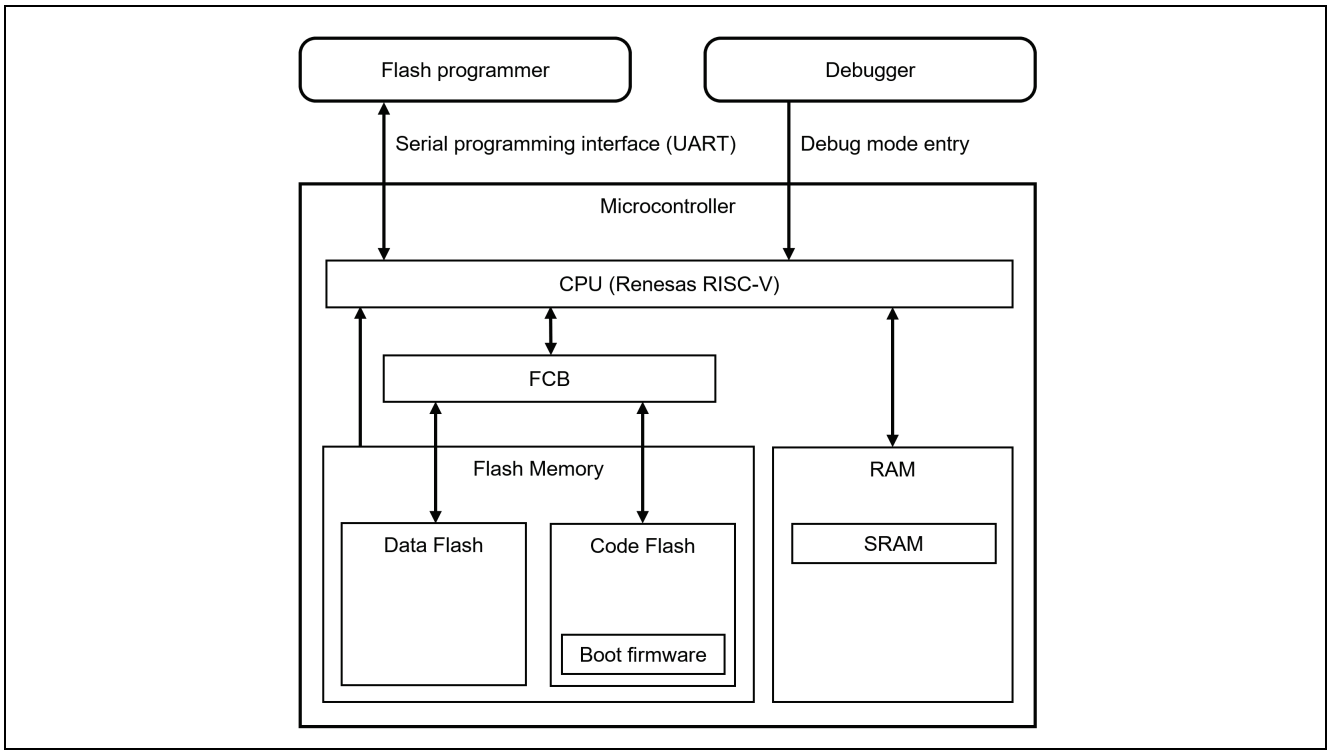

Figure 4. Block Diagram

3. Communication Methods

Boot firmware has interfaces for following the communication methods.

3.1 2-wire UART Communication

Boot firmware supports the 2-wire UART communication.

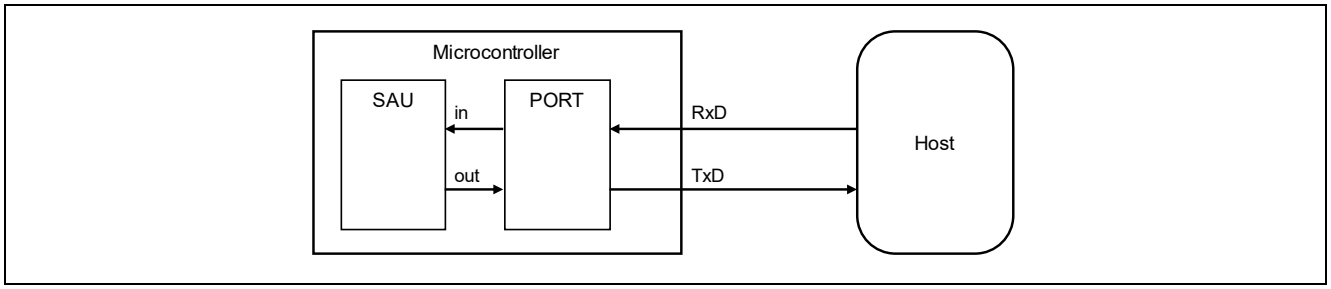

Figure 5. 2-Wire UART Communication

Table 2. General Settings

Interface	SAU Unit: SAU0 Channel: ch0 (Send only), ch1 (Receive only)	
RxD	P303, Input mode	
TxD	P302, Output mode	
Bit rate	(Initial / Min) 9600 bps (Max) 1.5 Mbps	
Data length	8-bit	(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 normally, communication is performed at the desired baud rate. The maximum bit rate that can be communicated with the device is returned by "RMB" of the signature request command.

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

4. General Procedure

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

1. Initialization phase
2. Communication setting phase
3. Authentication phase
4. Command acceptable phase
5. Command packet reception

4.1.1 Sequence Diagram (if ID code is already stored [Secure])

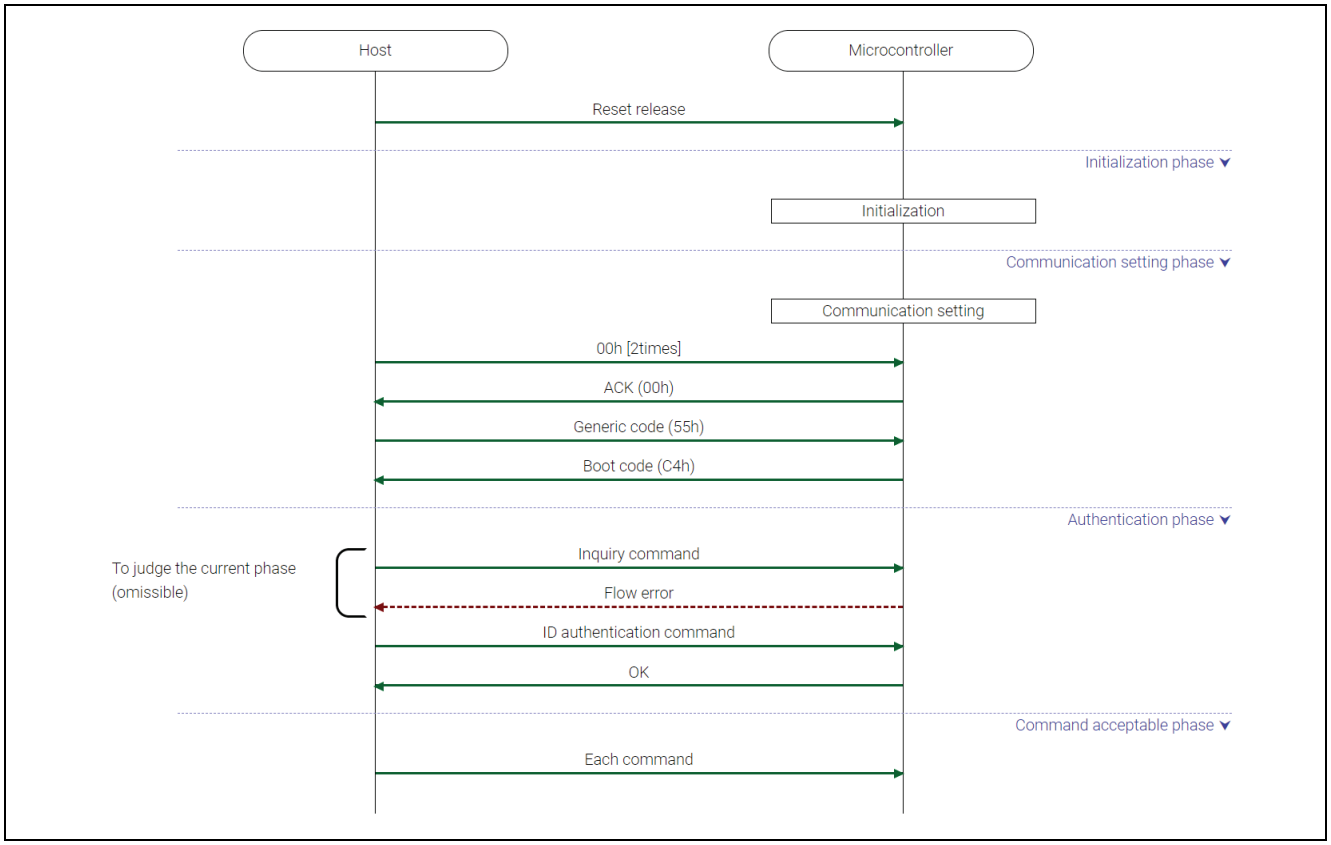

Figure 6. Sequence Diagram (if ID code is already stored [Secure])

4.1.2 Sequence Diagram (if ID code is not stored [Non-secure])

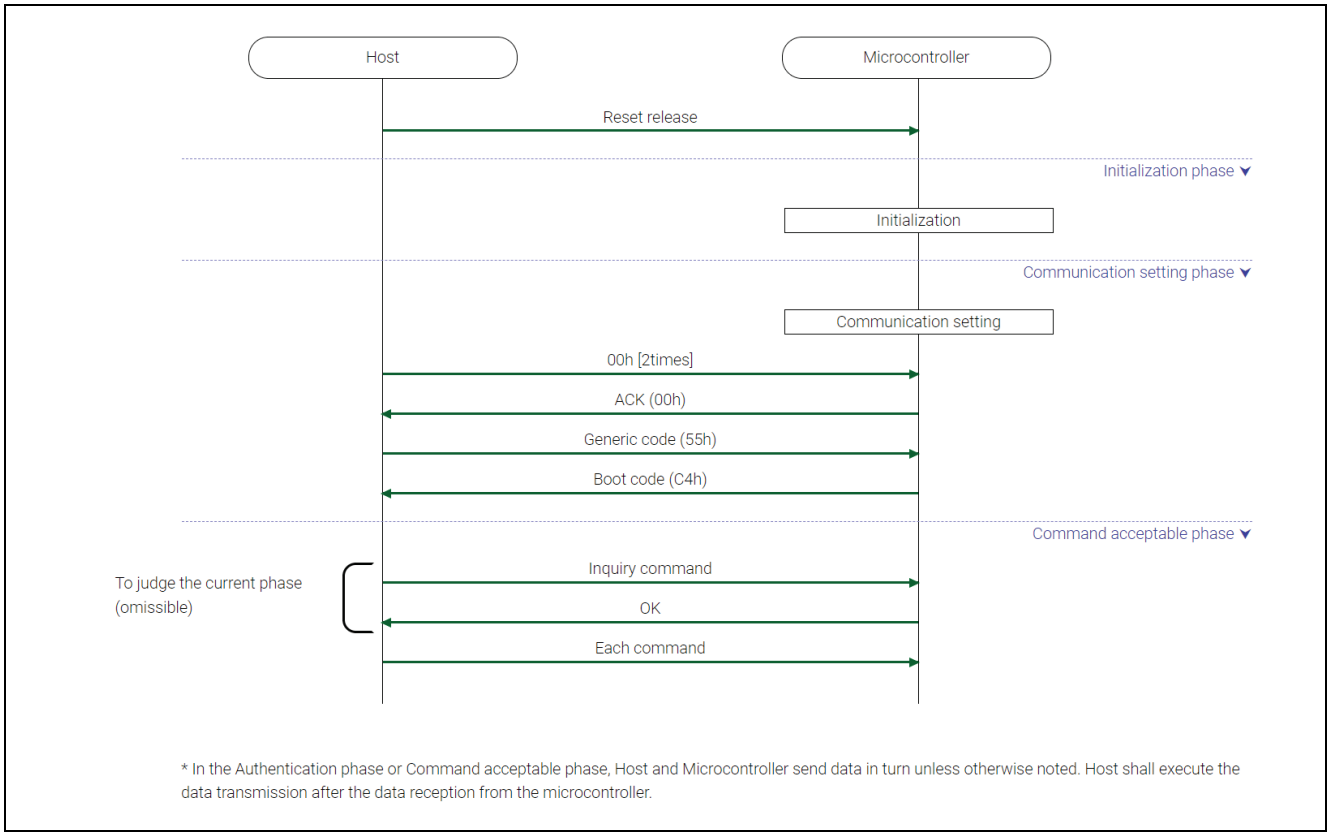

Figure 7. Sequence Diagram (if ID-code is not already stored [Non-secure])

4.1.3 State Transition Diagram (Generic state transition)

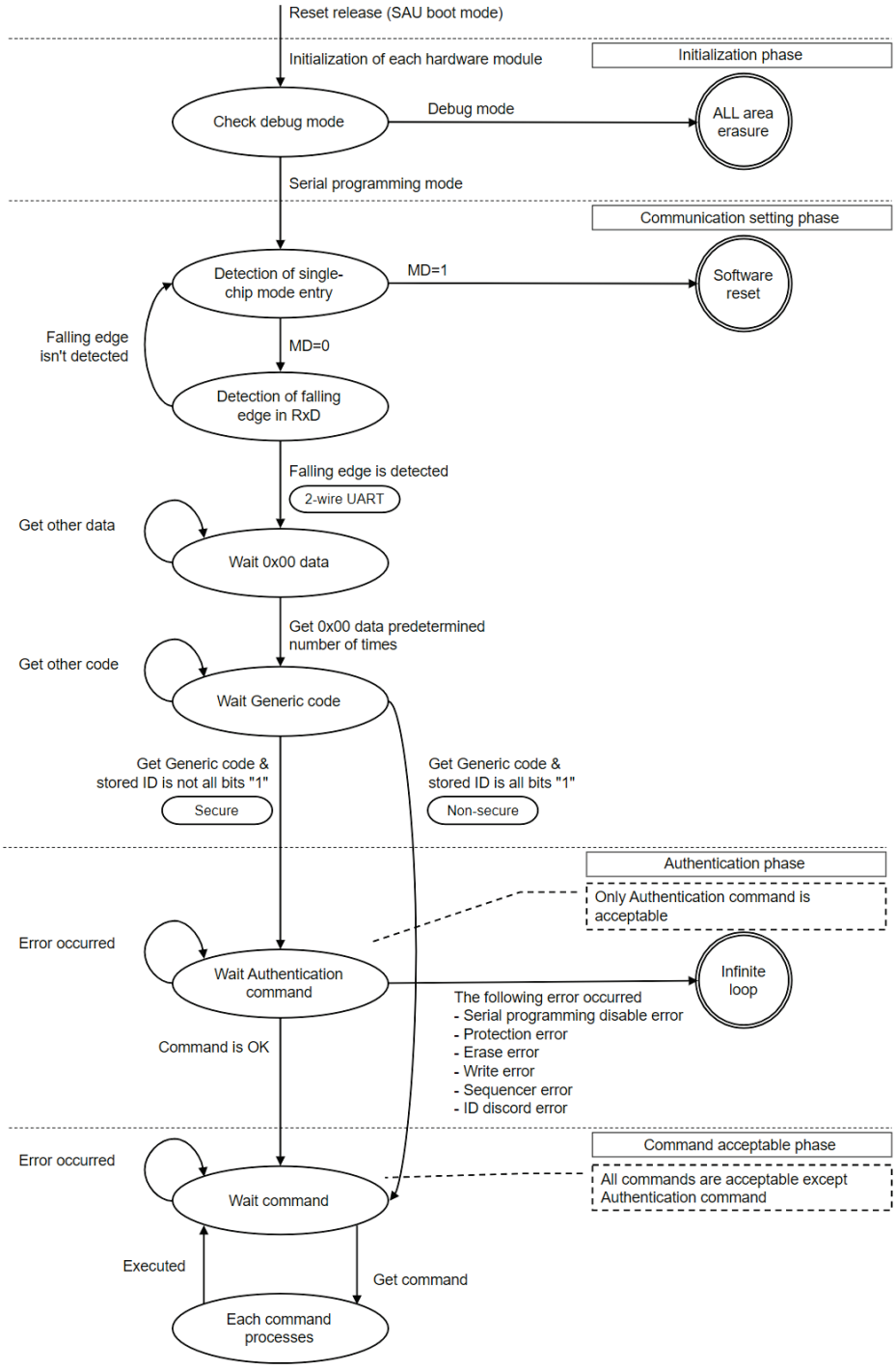

Figure 8. State Transition Diagram (Generic State Transition)

Check Debug Mode

Condition	Mode Decision
MD level is high at reset release (MD = 1)	Debug mode (Execute all area erasure)
MD level is high at reset release (MD = 0)	Serial programming mode

Note: If SECS.OCDDIS bit is cleared to 0b, no debugger connection is accepted. Make sure to execute an all erasure by Authentication command to write OCDDIS back to 1b. Please refer to the device user manual for more details on the OCDDIS functionality.

4.1.4 Cause for Operation Stop

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

- If stored ID[127] = 0 in the authentication command, an infinite loop will occur after returning a Serial programming disable error.
- If the received ID and stored ID do not match in the authentication command, an infinite loop will occur after returning an ID discord error.
- If all area erasure fails with the authentication command, an infinite loop will occur after returning an error code.

4.1.5 Cause for Software Reset

Boot firmware performs software reset in the following cases.

- If MD signal (P203) is High in the Communication setting phase, perform a software reset.

4.2 Initialization Phase

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

4.2.1 Processing Procedure

Boot firmware initializes after reset release:

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

  If the FAPR in the config area is set 0, boot firmware sets to Standby mode without executing all erasure, enters an infinite loop.
  * Flash memory status does not change before command reception.

4.2.2 Processing ALeRASE

Process	Description	Time
Boot Processing	Processing time excluding ALeRASE	29 ms
ALeRASE Execution	Clear Access Window + Chip erase - code flash(* flash macro) + - data flash(* flash macro) + - config area	-

4.3 Communication Setting Phase

After the reset release, boot firmware decides the communication mode (2-wired UART) by the following flowchart. After the communication setting and the receipt of Generic code via the selected communication method, boot firmware transits to the "Authentication phase". However, if ID code stored in the device is all "1", boot firmware transits to the "Command acceptable phase" directly.

4.3.1 Processing Procedure

Boot firmware performs communication settings:

- When MD = H, the boot firmware performs a software reset.
- When the 1st 00h is received in 2-wire UART communication, communication is decided to 2-wire UART communication.
- When the 2nd 00h is received, boot firmware transmit ACK (00h).
  When data which is not 00h is received, boot firmware discards the received data and continues waiting for 00h.

- When Generic code is received, boot firmware transmit Boot code.
  When data which is not Generic code is received, boot firmware discards the received data and continues waiting for Generic code.
- Boot firmware transmits to the following phase after sending Boot code:
  - When ID code stored in the device is all-1: Command acceptable phase.
  - When ID code stored in the device is not all-1: Authentication phase.

4.3.2 Decision of the Communication Mode

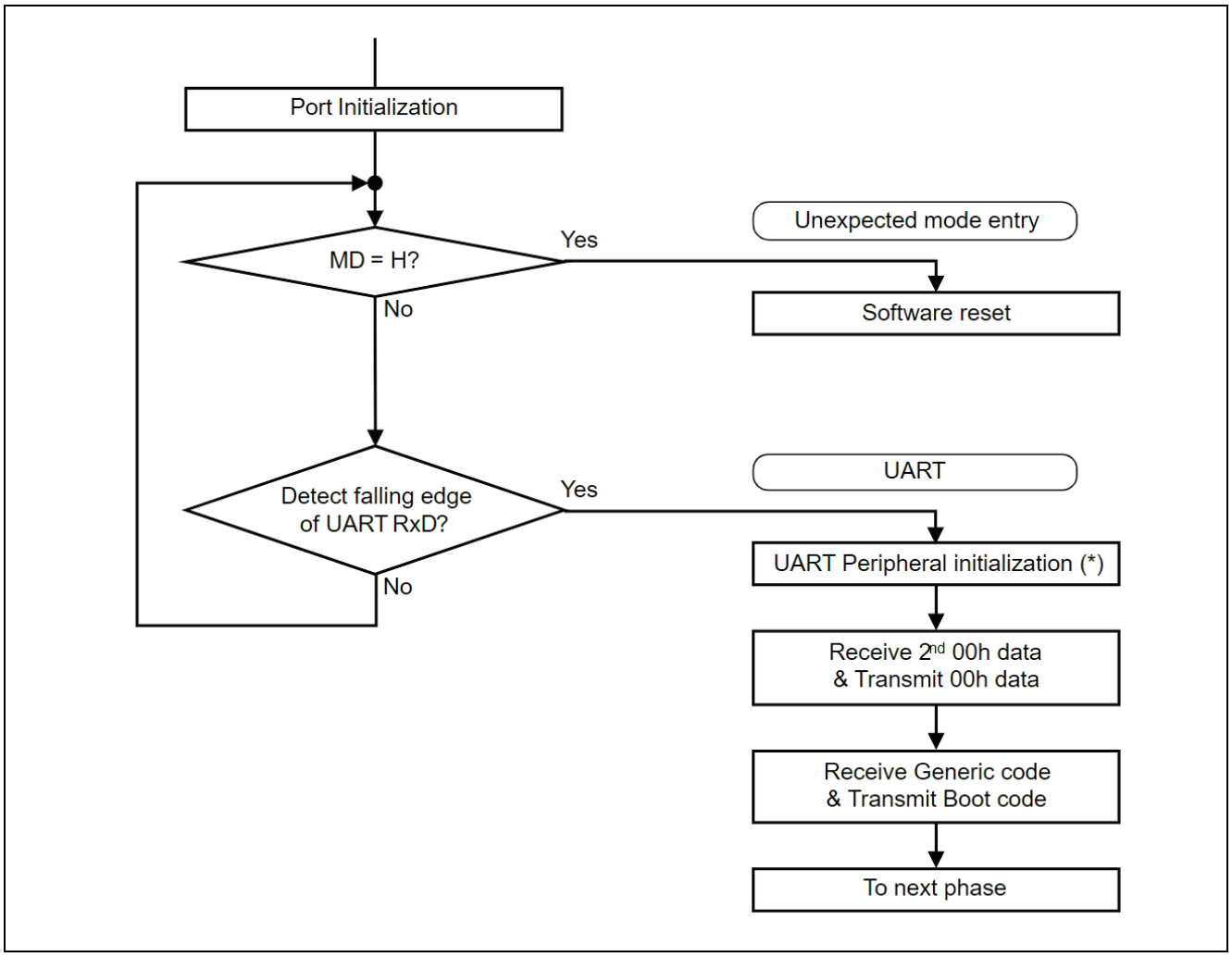

Figure 9. Communication Mode Decision

* UART Peripheral is SAU for R9A02G021 RISC-V MCUs.

4.3.3 Settings of the 2-wire UART Communication

To use the UART communication, flash programmer shall send 00h data (low pulse) at 9600 bps at least 2 times. (If the user's environment is noisy, it is recommended to retry sending the low pulse until ACK is received.) Boot firmware sets UART communication by the detection of falling edge in RxD. After that, boot firmware receives the 2nd low pulse as 00h data in SCI, then returns the ACK, receives the Generic code, then returns the Boot code. Communication setting is completed by these handlings.

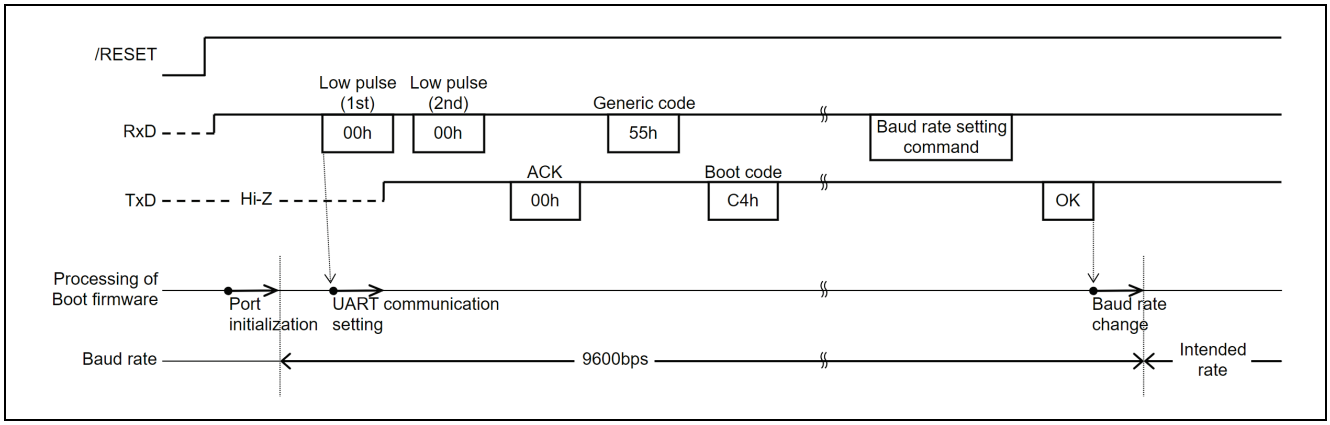

Figure 10. 2-wire UART Communication

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

1. Receive 2 bytes of 00h data (9600 bps) from the host/ (perform 00h data transmission until ACK is received in step 2.)
2. Send 00h data (ACK) from boot firmware.
3. Receive 55h data (Generic code) from the host.
4. Send C4h data (Boot code) from boot firmware.

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

4.4 Authentication Phase

Boot firmware authenticates ID code in this phase. This phase can accept only Authentication command. If Authentication command passed successfully, boot firmware transits to "Command acceptable phase".

4.4.1 Processing Procedure

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

(Refer to Command packet reception for details)

When the packet analysis is successfully completed, boot firmware executes Authentication command.

(Refer to Authentication command for details)

- Boot firmware enters an infinite loop when Authentication command abnormally finishes.
- Boot firmware transits to Command acceptable phase when Authentication command normally finishes.

4.5 Command Acceptable Phase

This phase can accept all of the commands except Authentication command. Flash programmer can judge if the current phase is "Command acceptable phase" or "Authentication phase" by the result of the Inquiry command.

4.5.1 Processing Procedure

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

(Refer to Command packet reception for details).

When the packet analysis is successfully completed, boot firmware executes received command.

(Refer to Command List for details).

- Boot firmware stays on Command acceptable phase after a command is finished.

4.6 Command Packet Reception

Boot firmware analyzes received command packet.

4.6.1 Processing Procedure

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

- The boot firmware recognizes the start of the command packet by receiving SOH.
  If the boot firmware receives something other than SOH, it waits until SOH is received.
- If ETX is not added to the received command packet, the boot firmware sends a "Packet error".
- If the SUM of the received command packet is different from the sum value, the boot firmware sends a "Checksum error".
- If the received command packet's LNH and LNL are different from the values specified in the packet format, the boot firmware sends a "Packet error".
- If the CMD in the received command packet is an undefined code, the boot firmware sends an "Unsupported command error".
- If the received command packet's LNH and LNL are different from the values specified in each command, the boot firmware sends a "Packet error".
- If a command other than Authentication command is received in Authentication phase, boot firmware sends a "Flow error".
- If Authentication command is received in Command acceptable phase, boot firmware sends a "Flow error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.

5. Packet Format

Use the following packet types:

1. Command packet
2. Data packet

Elements in the Packet

- CMD : Command code
- RES : Response code
- STS : Status code

5.1 Command Packet

Host sends data of a command packet to microcontroller by the following 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 byte	-	Command information for example, for write command: Start/End address
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

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

5.2 Data Packet

Host and boot firmware sends data to each other by the following 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	1~1024 byte	-	Transmit data for example, For Write data transmission: Write data for example, For status transmission: Status code (STS)
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

*1: If the host sends a packet whose length is 0 byte or over 1025 bytes, the microcontroller will return a packet with indefinite RES value.

*2: If the host sends data that exceeds 1030 bytes, subsequent operations are not guaranteed.

5.3 CMD: Command Code

Value	Name	Description
00h	Inquiry command	Return ACK (to determine the current phase)
12h	Erase command	Erase data on target area
13h	Write command	Write data on target area
15h	Read command	Read data on target area
18h	CRC command	Cyclic Redundancy Check of target area
30h	ID authentication command	Authenticate ID for connection with the device
34h	Baud rate setting command	Set baud rate for UART
3Ah	Signature request command	Get signature information
3Bh	Area information request command	Get area information

5.4 RES: Response Code

Value	Name	Description
00h | CMD	OK (ongoing normally)	-
80h | CMD	ERR (occurrence of an error)	-

5.5 STS: Status Code

Value	Name	Description
00h	Communication is normal [OK]	-
C0h	Unsupported command error	(*1), Received an unsupported command.
C1h	Packet error	(*1), Abnormality of packet format.
C2h	Checksum error	(*1), Abnormality of packet's checksum value.
C3h	Flow error	(*1), The phase in which the command cannot be executed.
D0h	Address error	(*1), Abnormality of packet's address.
D4h	Baud rate margin error	(*1), Abnormality of baud rate value.
DAh	Protection error	(*1), Erase, write or read protected area.
DBh	ID discord error	(*1), ID authentication failed.
DCh	Serial programming disable error	(*1), If serial programming is disabled. (stored ID[127] = 0).
E1h	Erase error	(*1), If Erase Error is detected.
E2h	Write error	(*1), If Program Error is detected.
E7h	Sequencer error	(*1), If Other Flash errors are detected.

*1: When this error occurs, the response code (RES) will be ERR.

6. Command List

Name	Communication Method	Phase				Prerequisite Command
		Initialization	Communication Setting	Authentication	Command Acceptable	
Inquiry Command	2-wire UART	NG	NG	Flow error	OK	-
Erase command	2-wire UART	NG	NG	Flow error	OK	-
Write command	2-wire UART	NG	NG	Flow error	OK	-
Read command	2-wire UART	NG	NG	Flow error	OK	-
Authentication command	2-wire UART	NG	NG	OK	Flow error	-
Baud rate setting command	2-wire UART	NG	NG	Flow error	OK	-
Signature request command	2-wire UART	NG	NG	Flow error	OK	-
Area information request command	2-wire UART	NG	NG	Flow error	OK	-
CRC command	2-wire UART	NG	NG	Flow error	OK	-

Communication method: This command is available in the communication. (Even if an unavailable command is sent, boot firmware does not return an error.)
NG: This command is not available in the phase. (Boot firmware does not return any data packet.)

6.1 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.

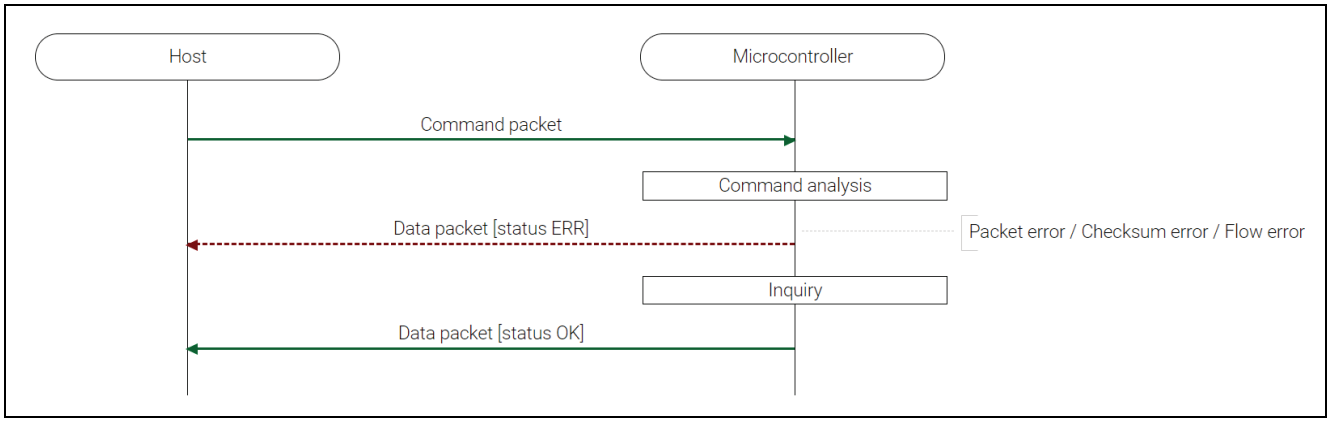

Figure 11. Inquiry Command Sequence Diagram

6.1.1 Packets

6.1.1.1 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

6.1.1.2 Data Packet [status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	00h (OK)
STS	(1 byte)	00h (OK)
SUM	(1 byte)	FEh
ETX	(1 byte)	03h

6.1.1.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	80h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.1.2 Processing Procedure

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

(Refer to Command packet reception for details)

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

- The boot firmware sends "OK".

* Flash memory status does not change before command reception.

6.1.3 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX.	Packet error
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Authentication phase".	Flow error
The process has ended normally.	OK

6.2 Erase Command

This command erases data in the specified area of the flash memory. The alignment of the target addresses shall follow the area information returned by the Area information request command. Erasures are executed in order from the start address to the end address by the erase access unit.

This command require adherence to conditions described in Command List.

6.2.1 Sequence Diagram

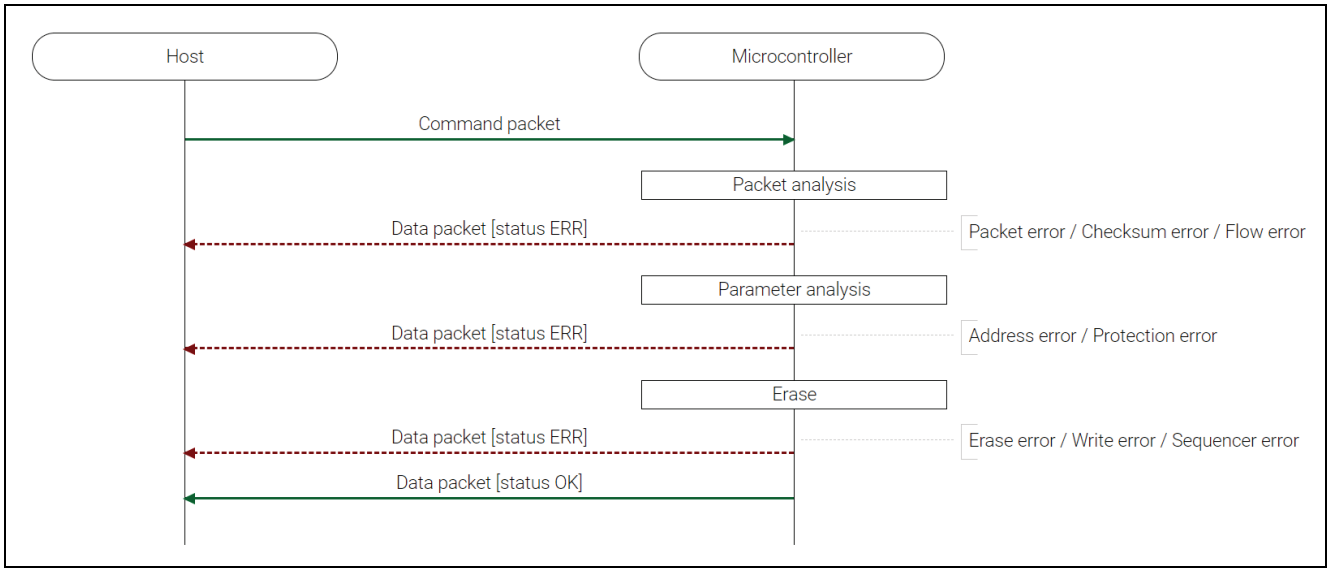

Figure 12. Erase Command Sequence Diagram

6.2.2 Packets

6.2.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	12h (Erase command)
SAD	(4 byte)	Start address for example, 00004000h -> 00h, 00h, 40h, 00h
EAD	(4 byte)	End address for example, 003FFFFFh -> 00h, 3Fh, FFh, FFh
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.2.2.2 Data Packet [status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	12h (OK)
STS	(1 byte)	00h (OK)
SUM	(1 byte)	ECh
ETX	(1 byte)	03h

6.2.2.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	92h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.2.3 Processing Procedure

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

(Refer to Command packet reception for details)

Boot firmware analyzes the received parameters after packet analysis.

- If the SAD is greater than EAD, the boot firmware sends an "Address error".
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends an "Address error".
- If SAD and EAD belong to different KOA, boot firmware will send an "Address error".
- If the EAU for the specified area is 0, the boot firmware sends an "Address error".
- If SAD and EAD are not specified in the EAU of the area, the boot firmware sends an "Address error".
- If the specified area is the user area in Code Flash and also includes area outside of Access window, Boot firmware sends a "Protection error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis is successfully completed, boot firmware executes the erase processing.

- Erase processing is performed.
- When the erase processing is normally finished, boot firmware returns "OK" and waits for the next command.
  * Specified area on flash memory are erased state.
- When the erase processing is abnormally finished, boot firmware returns "Erase error", "Write error" or "Sequencer error", then waits for the next command.
  * Refer to Status code for conditions of the errors.

  * Specified flash memory area are undefined.

6.2.4 Status Information from the Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX	Packet error
Sum data in the received packet is different from the value calculated by the boot firmware.	Checksum error

Condition	STS
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Authentication phase".	Flow error
SAD is bigger than EAD.	Address error
SAD or EAD is outside the scope of accessible area specified in area information.	Address error
SAD and EAD belong to different KOA.	Address error
The access unit "EAU" of the specified area is 0.	Address error
SAD or EAD does not comply with EAU of the area.	Address error
If the target area contains outside of Access window in case of User area in code flash.	Protection error
If the erase error occurs.	Erase error
If the write error occurs.	Write error
If the sequencer error occurs.	Sequence error
Successful completion	OK

6.3 Write Command

This command receives data from the host and writes those data to the flash memory. The alignment of the target address shall follow the area information returned by the Area information request command. Writings are executed in order from the start address to the end address by the write access unit.

This command require adherence to conditions described in Command List.

6.3.1 Sequence Diagram

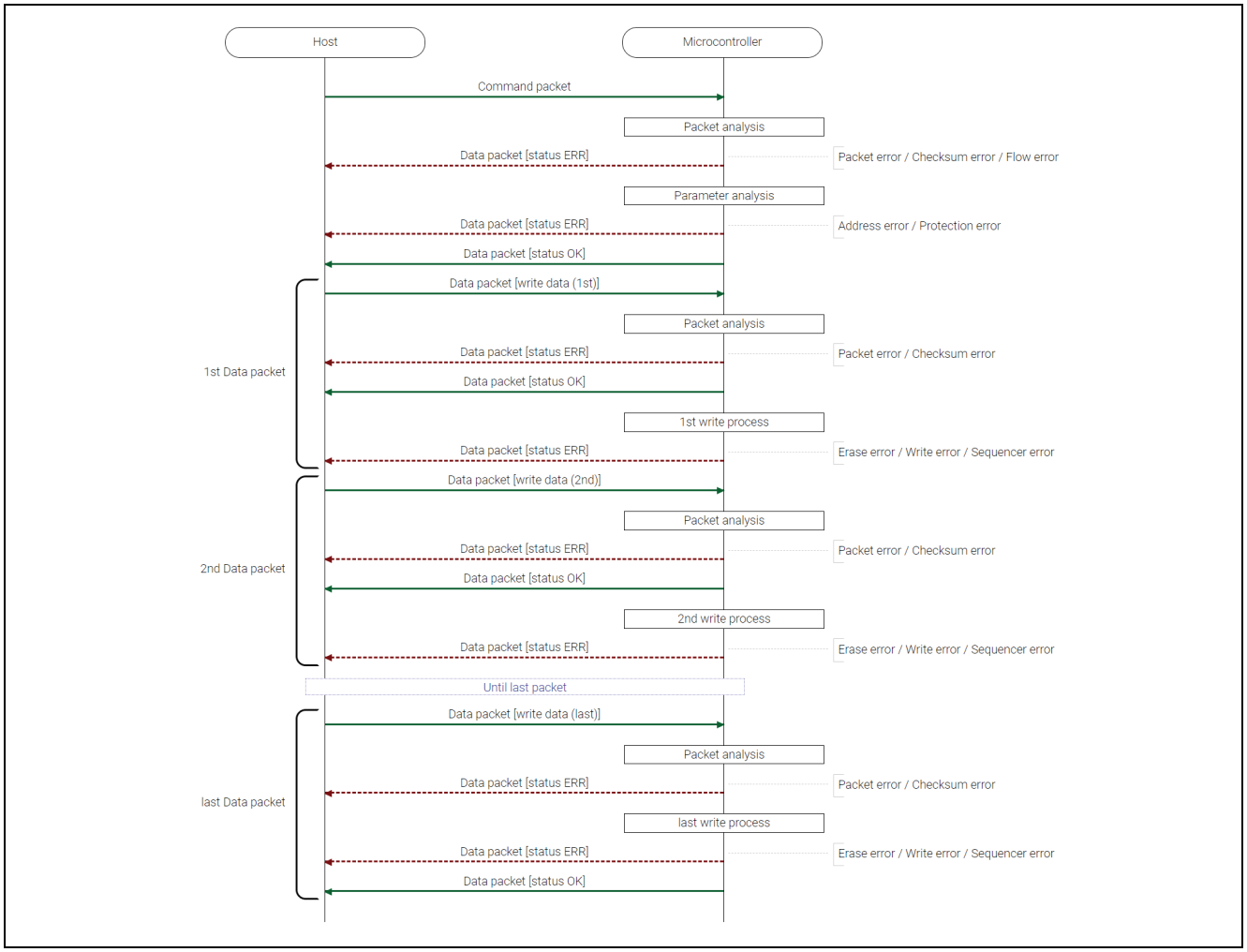

Figure 13. Write Command Sequence Diagram

6.3.2 Packets

6.3.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	13h (write command)
SAD	(4 byte)	Start address for example 00004000h -> 00h, 00h, 40h, 00h
EAD	(4 byte)	End address for example 003FFFFFh -> 00h, 3Fh, FFh, FFh
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.3.2.2 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 byte)	Write data
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

N = 1 ~ 1024

6.3.2.3 Data Packet [Status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	13h (OK)
STS	(1 byte)	00h (OK)
SUM	(1 byte)	EBh
ETX	(1 byte)	03h

6.3.2.4 Data Packet [Status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	93h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.3.3 Processing Procedure

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

(Refer to Command packet reception for details).

Boot firmware analyzes the received parameters after packet analysis.

- If the SAD is greater than EAD, the boot firmware sends an "Address error".
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends an "Address error".
- If SAD and EAD belong to different KOA, boot firmware will send an "Address error".
- If the WAU for the specified area is 0, the boot firmware sends an "Address error".
- If SAD and EAD are not specified in the WAU of the area, the boot firmware sends an "Address error".

- If the specified area is user area in Code Flash and also includes area outside of Access window, boot firmware sends a "Protection error".
- If the specified area includes access window setting area and also FAPR is set (= 0), boot firmware sends a "Protection error".
- If the specified range is an area that includes the following in the Config area and the FAPR bit is set (FAPR=0), a "Protection error" is sent.
  - Customer ID
  - Customer ID key address
  - Customer ID key data
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception
- If the above error does not occur, the boot firmware sends "OK".

Boot firmware receives data packet and executes write processing after parameter analysis.

- The boot firmware recognizes the start of the data packet by receiving SOD.
  If the boot firmware receives something other than SOD, it will wait until it receives SOD.
- If ETX is not added to the received data packet, the boot firmware sends a "Packet error".
- If the SUM of the received data 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, "Packet error" is returned.
- If the size of the write data are not specified in the WAU of the area, the boot firmware sends a "Packet error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the 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 a data packet and returns "Erase error", "Write error" or "Sequencer error", then waits for the next command.
  * Refer to Status code for conditions of the errors
  * Specified flash memory area are undefined.
- When the write processing is normally finished, boot firmware receives the next data packet.

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

- Boot firmware executes the write processing.
- When the write processing is abnormally finished, boot firmware returns "Erase error", "Write error" or "Sequencer error", then waits for the next command.
  * Refer to Status code for conditions of the errors
  * Specified flash memory area are undefined.
- When the write processing is normally finished, boot firmware returns "OK" and waits for the next command.
  * Sent data are written to the specified area on flash memory.

6.3.4 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX	Packet error
Sum data in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Authentication phase".	Flow error
SAD is bigger than EAD.	Address error
SAD or EAD is outside the scope of accessible area specified in area information.	Address error
SAD and EAD belong to different KOA.	Address error
The access unit "WAU" of the specified area is 0.	Address error
SAD or EAD does not comply with WAU of the area.	Address error
If the target area contains outside of Access window in case of User area in code flash.	Protection error
If the target area contains FAWS or FAWE in Config area, and the FAPR bit is 0.	Protection error
If the target area contains the followings in Config area, and the FAPR bit is 0: • Customer ID • Customer ID key address • Customer ID key data	Protection error
The RES of the received data packet is different from the value specified by this command.	Packet error
The total length of received data of data packets exceeds the specified end address.	Packet error
The data size of the data packet doesn’t comply with writing unit of the area.	Packet error
If the erase error occurs.	Erase error
If the write error occurs.	Write error
If the sequencer error occurs.	Sequence error
Successful completion.	OK

6.3.5 Precautions

- The following parameters cannot be executed when FAPR is 0:
  - Writing to access window start/end address setting
  - Writing to Customer ID
  - Writing to Customer ID key address
  - Writing to Customer ID key data
  - All erasure by Authentication command with ALeRASE ID
  - All erasure by debug mode

  There is no way to write FAPR back to 1 once after writing 0.

- When writing 0 to FAPR and writing protected data*1 are executed by one Write command, Sequencer error occurs at writing to protected data*1 for the following reasons:
  - FAPR is placed at smaller address than the protected data*1
  - It is impossible to write FAPR and protected data*1 by one sequencer command
    - Protection by FAPR is enabled before writing protected data*1

  To avoid this, execute write command two times to write FAPR and the protected data*1:
  1st write command: Writing to the protected data*1
  2nd write command: Writing to FAPR = 0 (recommended to execute at very last step during production programming)

  Note: *1. The protected data are the following:
  - Access window setting
  - Customer ID
  - Customer ID key address
  - Customer ID key data

- OCDDIS cannot be written back from 0 to 1 by Write command (the value remains 0 even when Write command normally finishes).
  Execute all erasure by Authentication command to write OCDDIS back to 1.

6.4 Read Command

This command reads data from a specified area in the flash memory, and sends those data to host. Readings are executed in order from the start address to the end address by 1 byte unit.

This command require adherence to conditions described in Command List.

6.4.1 Sequence Diagram

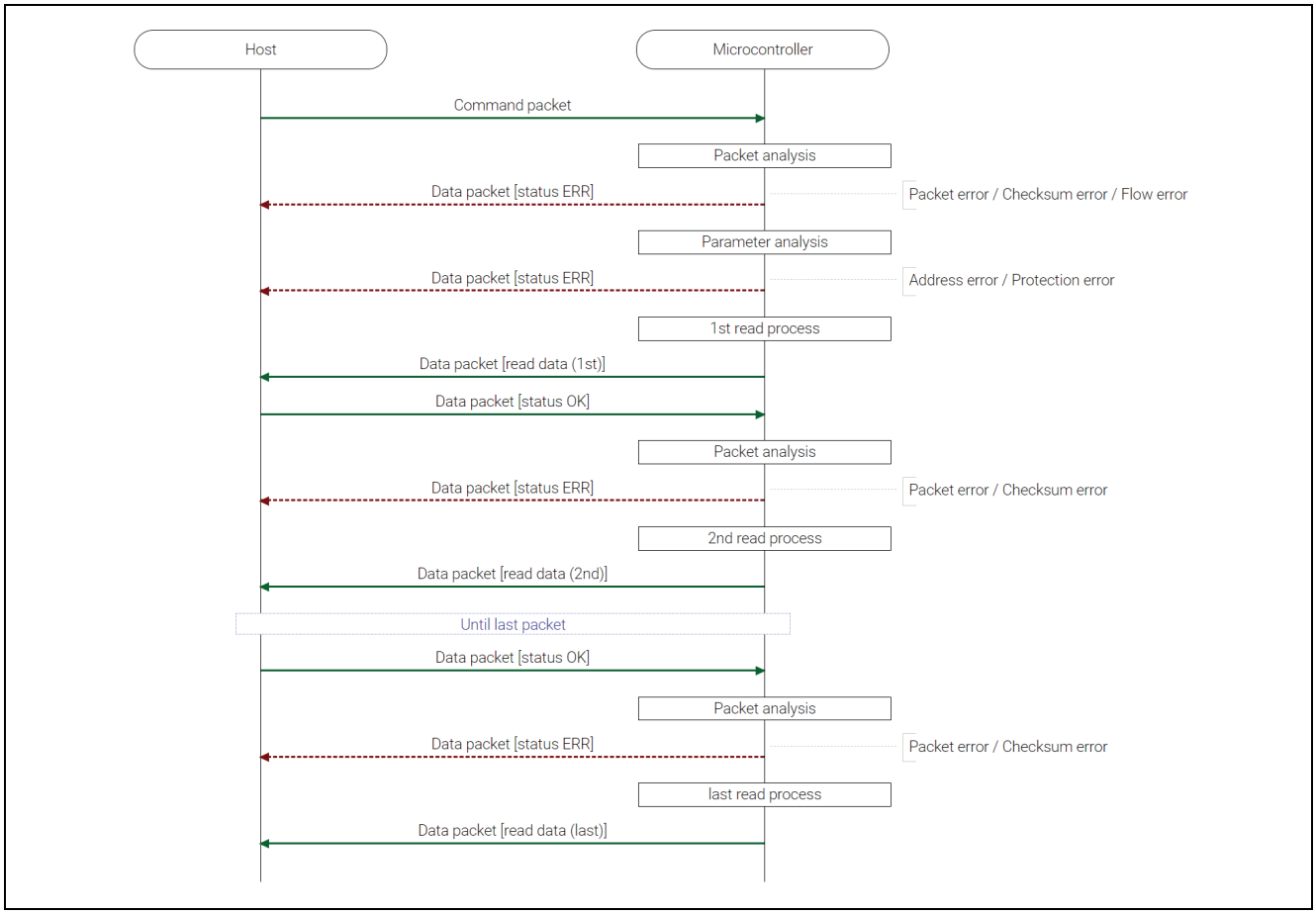

Figure 14. Read Command Sequence Diagram

6.4.2 Packets

6.4.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	15h (read command)
SAD	(4 byte)	Start address for example, 00004000h -> 00h, 00h, 40h, 00h
EAD	(4 byte)	End address for example, 003FFFFFh -> 00h, 3Fh, FFh, FFh
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)	02h
RES	(1 byte)	15h (OK)
STS	(1 byte)	00h (OK)
SUM	(1 byte)	E9h
ETX	(1 byte)	03h

6.4.2.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	95h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.4.2.4 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

N = 1~1024

6.4.3 Processing Procedure

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

(Refer to Command packet reception for details).

Boot firmware analyzes the received parameters after packet analysis:

- If the SAD is greater than EAD, the boot firmware sends an "Address error".
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends an "Address error".
- If SAD and EAD belong to different KOA, boot firmware will send an "Address error".
- If the specified area is user area in Code Flash and also includes area outside of Access window, boot firmware sends a "Protection error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

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

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

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

- The boot firmware recognizes the start of the data packet by receiving SOD.
  If the boot firmware receives something other than SOD, it will wait until it receives SOD.
- If ETX is not added to the received data packet, the boot firmware sends a "Packet error".
- If the SUM of the received data 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 the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception
- If the above error does not occur, the boot firmware continues to read and send data.

6.4.4 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX.	Packet error
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Authentication phase".	Flow error
SAD is bigger than EAD.	Address error
SAD or EAD is outside the scope of accessible area specified in area information.	Address error
SAD and EAD belongs to different KOA.	Address error

Condition	STS
If the specified area is a user area in code flash and contains outside the scope of the access window.	Protection error
The RES of the received data packet is different from the value specified by this command.	Packet error

6.5 Authentication Command

This command compares the ID-code stored in the microcontroller with the ID-code received from the flash programmer, and sends the result to the flash programmer. If the received ID code is "ALeRASE(*1)", All areas of the flash memory are erased without ID authentication.

This command require adherence to conditions described in Command List.

*1: "ALeRASE" = 41h, 4Ch, 65h, 52h, 41h, 53h, 45h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, FFh, FFh

Condition		Processing	
Stored ID[127:126] = 0xb (x means Don't care)		Serial programming disable	-> Enter an infinite loop
Stored ID[127:126] = 10b		ID authentication	when ID mismatch -> Enter an infinite loop when ID match -> Transition to "Command acceptable phase"
Stored ID[127:126] = 11b	received ID != "ALeRASE"		
	received ID = "ALeRASE"	All area erasure	when FAPR = 0 -> Enter an infinite loop when Error occurred -> Enter an infinite loop when Successful erasure -> Transition to "Command acceptable phase"

6.5.1 Sequence Diagram

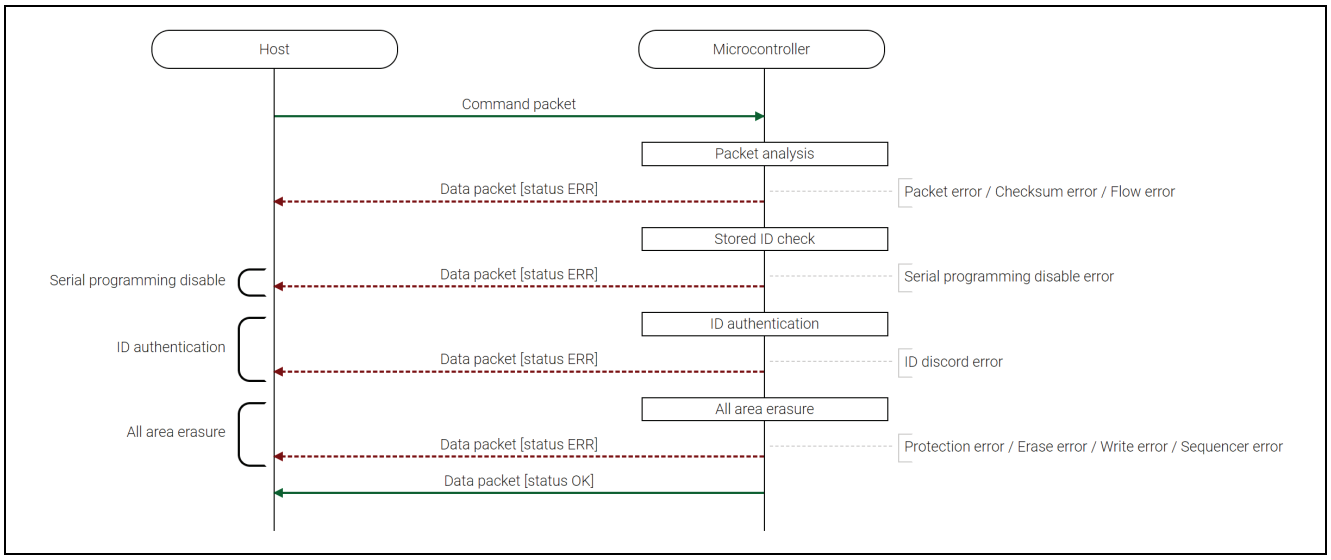

Figure 15. Authentication Command Sequence Diagram

6.5.2 Packets

6.5.2.1 Command Packet

<table>
<tr><td>SOH</td><td>(1 byte)</td><td>01h</td><td></td></tr>
<tr><td>LNH</td><td>(1 byte)</td><td>00h</td><td></td></tr>
<tr><td>LNL</td><td>(1 byte)</td><td>11h</td><td></td></tr>
<tr><td>CMD</td><td>(1 byte)</td><td>30h (authentication command)</td><td></td></tr>
<tr><td>IDC</td><td>(16 byte)</td><td>ID code</td><td>For example, when stored ID is following case, flash programmer shall send IDC in order of the lower table.
Stored ID:<table><tr><td>ID[127:96]</td><td>ID[95:64]</td></tr><tr><td>F0F1F2F3h</td><td>E0E1E2E3h</td></tr><tr><td>ID[63:32]</td><td>ID[31:0]</td></tr><tr><td>D0D1D2D3h</td><td>C0C1C2C3h</td></tr></table>Order of sending IDC for ID authentication:<table><tr><td>1st</td><td>2nd</td><td>3rd</td><td>4th</td><td>5th</td><td>6th</td><td>7th</td><td>8th</td></tr><tr><td>F0h</td><td>F1h</td><td>F2h</td><td>F3h</td><td>E0h</td><td>E1h</td><td>E2h</td><td>E3h</td></tr><tr><td>9th</td><td>10th</td><td>11th</td><td>12th</td><td>13th</td><td>14th</td><td>15th</td><td>16th</td></tr><tr><td>D0h</td><td>D1h</td><td>D2h</td><td>D3h</td><td>C0h</td><td>C1h</td><td>C2h</td><td>C3h</td></tr></table>for example, for All area erasure.
Order of sending IDC for ID authentication:<table><tr><td>1st</td><td>2nd</td><td>3rd</td><td>4th</td><td>5th</td><td>6th</td><td>7th</td><td>8th</td></tr><tr><td>41h</td><td>4Ch</td><td>65h</td><td>52h</td><td>41h</td><td>53h</td><td>45h</td><td>FFh</td></tr><tr><td>9th</td><td>10th</td><td>11th</td><td>12th</td><td>13th</td><td>14th</td><td>15th</td><td>16th</td></tr><tr><td>FFh</td><td>FFh</td><td>FFh</td><td>FFh</td><td>FFh</td><td>FFh</td><td>FFh</td><td>FFh</td></tr></table></td></tr>
<tr><td>SUM</td><td>(1 byte)</td><td>Sum data</td><td></td></tr>
<tr><td>ETX</td><td>(1 byte)</td><td>03h</td><td></td></tr>
</table>

6.5.2.2 Data Packet [status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	30h (OK)
STS	(1 byte)	00h (OK)
SUM	(1 byte)	CEh
ETX	(1 byte)	03h

6.5.2.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	B0h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.5.3 Processing Procedure

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

(Refer to Command packet reception for details)

Boot firmware checks for "Serial programming disable error" after packet analysis.

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

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

- Boot firmware execute ID code authentication when at least one of the following conditions are met.
  - Bit 126 of the ID code stored in the device is 0b.
  - Bit 126 of the ID code stored in the device is 1b but received IDC is not "ALeRASE".
- When ID code authentication fails, boot firmware send "ID discord error" and finishes command processing.
  Boot firmware does not accept any commands nor response for the commands after sending this error.
  * Flash memory status does not change before command reception.
- When ID code authentication passes, boot firmware send "OK" and finishes command processing.
  Then boot firmware transits to Command acceptable phase and wait for the next command.
  * Flash memory status does not change before command reception.

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

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

6.5.4 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX.	Packet error
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Command acceptable phase".	Flow error
If the Highest-order bit of stored ID is "0".	Serial programming disable error
If the FAPR bit is 0, when all area erasure.	Protection error
If the erase error occurs, when all area erasure.	Erase error
If the write error occurs, when all area erasure.	Write error
If the sequencer error occurs, when all area erasure.	Sequencer error
The ID authentication fails.	ID discord error
Successful completion.	OK

6.6 Baud Rate 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 requires adherence to conditions described in Command List.

6.6.1 Sequence Diagram

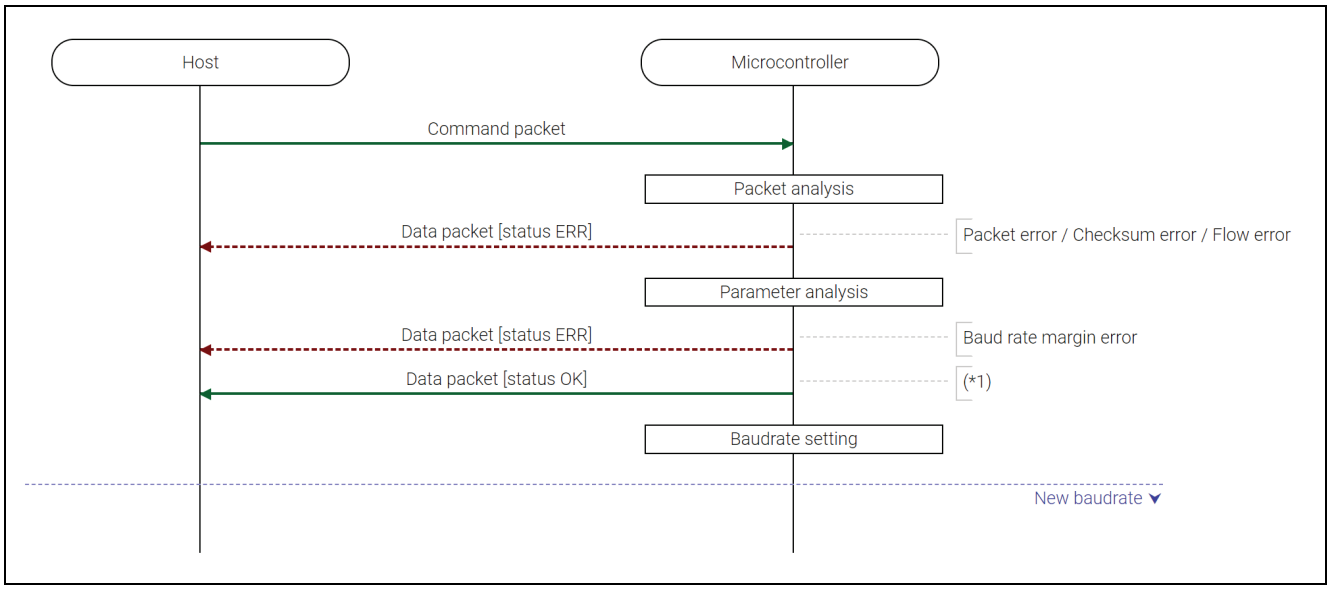

Figure 16. Baud Rate Command Sequence Diagram

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

6.6.2 Packets

6.6.2.1 Command Packet

<table>
<tr><td>SOH</td><td>(1 byte)</td><td>01h</td><td></td></tr>
<tr><td>LNH</td><td>(1 byte)</td><td>00h</td><td></td></tr>
<tr><td>LNL</td><td>(1 byte)</td><td>05h</td><td></td></tr>
<tr><td>CMD</td><td>(1 byte)</td><td>34h (baudrate setting command)</td><td></td></tr>
<tr><td>BRT</td><td>(4 byte)</td><td>UART baudrate [bps]</td><td>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>00h</td><td>00h</td><td>25h</td><td>80h</td></tr>
<tr><td>115200 bps</td><td>00h</td><td>01h</td><td>C2h</td><td>00h</td></tr>
<tr><td>500 Kbps</td><td>00h</td><td>07h</td><td>A1h</td><td>20h</td></tr>
<tr><td>1.0 Mbps</td><td>00h</td><td>0Fh</td><td>42h</td><td>40h</td></tr>
<tr><td>1.5 Mbps</td><td>00h</td><td>16h</td><td>E3h</td><td>60h</td></tr>
</table></td></tr>
<tr><td>SUM</td><td>(1 byte)</td><td>Sum data</td><td></td></tr>
<tr><td>ETX</td><td>(1 byte)</td><td>03h</td><td></td></tr>
</table>

6.6.2.2 Data Packet [status OK]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	34h (OK)
STS	(1 byte)	00h (OK)
SUM	(1 byte)	CAh
ETX	(1 byte)	03h

6.6.2.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	B4h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.6.3 Processing Procedure

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

(Refer to Command packet reception for details)

When the packet analysis is completed, the parameter analysis is performed.

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

When the parameter analysis is completed normally, the baud rate setting process is performed.

- Set the baud rate.

6.6.4 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX.	Packet error
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is “Authentication phase”	Flow error
Received BRT exceeds RMB.	Baud rate margin error
Different from the baud rate value supported by the received BRT.	Baud rate margin error
Started the baud rate setting.	OK

6.6.5 Baudrate Setting

Intended Baudrate	SPS[3:0]	SDR[15:9]	Accuracy
9600 bps	0001b	1001101b	+0.16%
115200 bps	0000b	0001100b	+0.16%

Intended Baudrate	SPS[3:0]	SDR[15:9]	Accuracy
9600 bps	0100b	1001101b	+0.16%
115200 bps	0011b	0001100b	+0.16%
500 Kbps	0000b	0010111b	0%
1.0 Mbps	0000b	0001011b	0%
1.5 Mbps	0000b	0000111b	0%

6.7 Signature Request Command

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

This command require adherence to conditions described in Command List.

6.7.1 Sequence Diagram

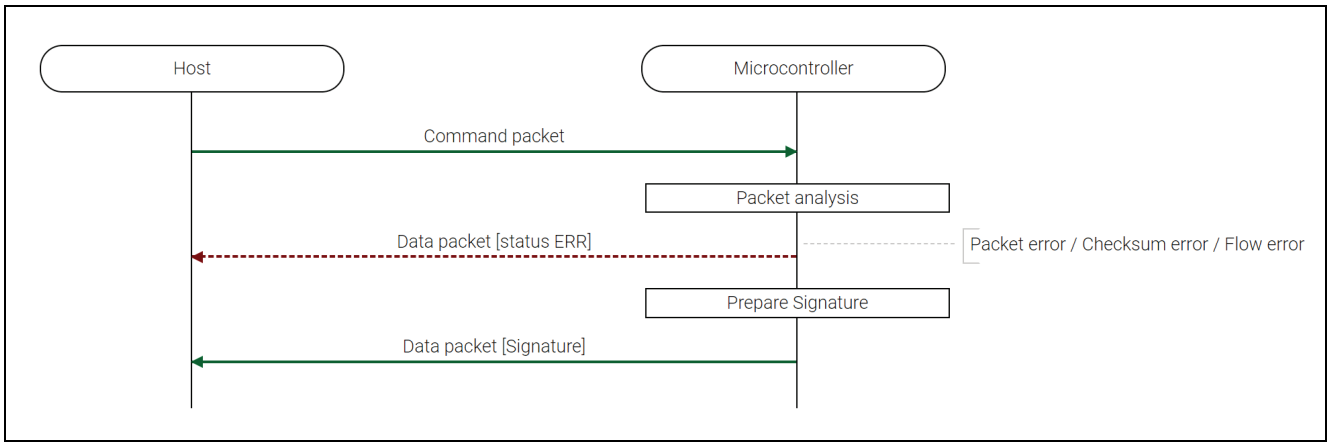

Figure 17. Signature Request Command Sequence Diagram

6.7.2 Packets

6.7.2.1 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

6.7.2.2 Data Packet [Signature]

SOD	(1 byte)	81h	
LNH	(1 byte)	00h	
LNL	(1 byte)	2Eh	
RES	(1 byte)	3Ah (OK)	
SAU	(4 byte)	SAU operating clock frequency [Hz]	For example, 24 MHz (24,000,000 Hz) -> 01h, 6Eh, 36h, 00h
RMB	(4 byte)	Recommended maximum UART baudrate [bps]	*Order of sending: High -> ... -> Low For example, 1 Mbps (1000000 bps) -> 00h, 0Fh, 42h, 40h
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)	02h: R9A02G021 RISC-V MCUs
BFV	(3 byte)	Boot firmware version	*Order of sending: Major version -> Minor version -> Build For example, v2.4.16 -> 02h, 04h, 10h
PNC	(16 byte)	P/N code	For example, when stored P/N code is following case, boot firmware will send PNC in order of the lower table. However, return FFh where P/N code is not set. Stored P/N code: "R9A02G0204GNPA01"

P/N code[127:96]	**P/N code[95:64]**
52394130h ("R9A0")	32473032h ("2G02")

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>P/N code[63:32]</td><td>P/N code[31:0]</td></tr>
<tr><td>3034474Eh ("04GN")</td><td>50413031h ("PA01")</td></tr>
</table>
Order of sending P/N code:
<table>
<tr><td>1st</td><td>2nd</td><td>3rd</td><td>4th</td><td>5th</td><td>6th</td><td>7th</td><td>8th</td></tr>
<tr><td>52h</td><td>39h</td><td>41h</td><td>30h</td><td>32h</td><td>47h</td><td>30h</td><td>32h</td></tr>
<tr><td>9th</td><td>10th</td><td>11th</td><td>12th</td><td>13th</td><td>14th</td><td>15th</td><td>16th</td></tr>
<tr><td>30h</td><td>34h</td><td>47h</td><td>4Eh</td><td>50h</td><td>41h</td><td>30h</td><td>31h</td></tr>
</table>
</td></tr>
<tr><td>UID</td><td>(16 byte)</td><td>Unique ID</td><td>
For example, when stored Unique ID is following case, boot firmware will send UID in order of the lower table.
Stored Unique ID:
<table>
<tr><td>UNIQID[127:96]</td><td>UNIQID[95:64]</td></tr>
<tr><td>00010203h</td><td>10111213h</td></tr>
<tr><td>UNIQID[63:32]</td><td>UNIQID[31:0]</td></tr>
<tr><td>20212223h</td><td>30313233h</td></tr>
</table>
Order of sending Unique ID:
<table>
<tr><td>1st</td><td>2nd</td><td>3rd</td><td>4th</td><td>5th</td><td>6th</td><td>7th</td><td>8th</td></tr>
<tr><td>00h</td><td>01h</td><td>02h</td><td>03h</td><td>10h</td><td>11h</td><td>12h</td><td>13h</td></tr>
<tr><td>9th</td><td>10th</td><td>11th</td><td>12th</td><td>13th</td><td>14th</td><td>15th</td><td>16th</td></tr>
<tr><td>20h</td><td>21h</td><td>22h</td><td>23h</td><td>30h</td><td>31h</td><td>32h</td><td>33h</td></tr>
</table>
</td></tr>
<tr><td>SUM</td><td>(1 byte)</td><td>Sum data</td><td></td></tr>
<tr><td>ETX</td><td>(1 byte)</td><td>03h</td><td></td></tr>
</table>

6.7.2.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	BAh (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.7.2.4 Processing Procedure

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

(Refer to Command packet reception for details).

When the packet analysis is successfully completed, boot firmware returns signature information.

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

6.7.2.5 Status Information from Microcontroller

(listed in descending order of priority)

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

6.8 Area Information Request Command

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

This command require adherence to conditions described in Command List.

6.8.1 Sequence Diagram

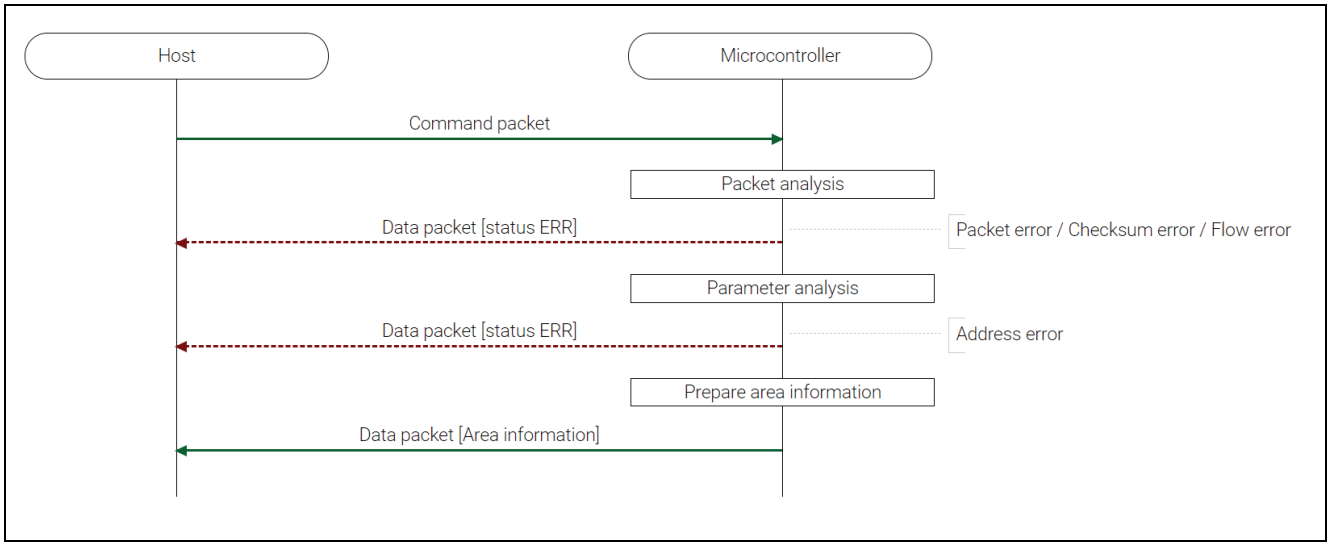

Figure 18. Area Information Request Command Sequence Diagram

6.8.2 Packets

6.8.2.1 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

6.8.2.2 Data packet [Area Information]

SOD	(1 byte)	81h	
LNH	(1 byte)	00h	
LNL	(1 byte)	12h	
RES	(1 byte)	3Bh (OK)	
KOA	(1 byte)	Kind of the area	00h: User area 01h: Data area 02h: Config area
SAD	(4 byte)	Start address	*Order of sending: High -> ... -> Low For example, 00010000h -> 00h, 01h, 00h, 00h
EAD	(4 byte)	End address	*Order of sending: High -> ... -> Low For example, 001FFFFFh -> 00h, 1Fh, FFh, FFh
EAU	(4 byte)	Erase access unit (alignment) [byte]*1	*Order of sending: High -> ... -> Low example 2 KB (2048 byte) -> 00h, 00h, 08h, 00h Target command : Erase command
WAU	(4 byte)	Write access unit (alignment) [byte] *1	*Order of sending: High -> ... -> Low example 8 byte -> 00h, 00h, 00h, 08h Target command : Write 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.

6.8.2.3 Data Packet [status ERR]

SOH	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	BBh (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.8.2.4 Processing Procedure

When the boot firmware receives a command packet, it performs packet analysis (refer to Command packet reception for details).

Boot firmware analyzes the received parameters after packet analysis:

- If the specified NUM is "NOA" returned by "signature requests command" or more, send "Address error" and return to command waiting status.
  * Flash memory status does not change before command reception.
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

When the packet analysis is successfully completed, boot firmware returns area information:

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

6.8.2.5 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX.	Packet error
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Authentication phase".	Flow error
If Area number in the received packet is non-existent area number.	Address error

6.8.2.6 Example of Area Information

(For example, R9A02G021 RISC-V MCUs)

NUM	Area	KOA	SAD	EAD	EAU	WAU
0	User area	00h	00000000h	0001FFFFh	2 KB	8B
1	Data area	01h	40100000h	40100FFFh	1 KB	1B
2	Config area	02h	01010008h	01010033h	0 (*1)	4B

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

6.9 CRC Command

This command calculates CRC data from a specified area in the flash memory, and sends it to host. The alignment of target addresses shall follow fixed 4 bytes. Calculations are executed in order from the start address to the end address by the CRC access unit.

This command require 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) 04C11DB7h

6.9.1 Sequence Diagram

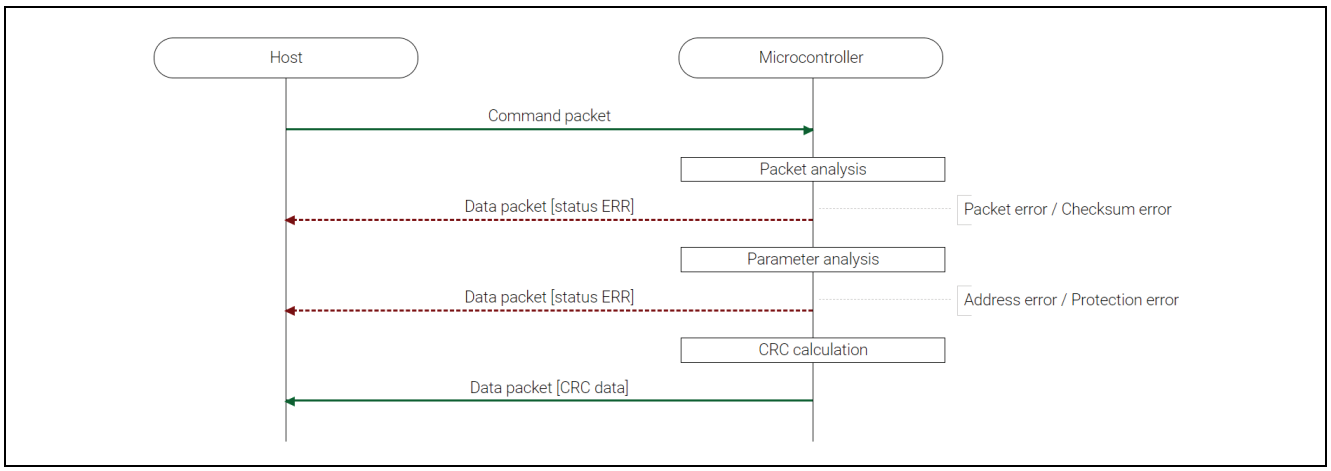

Figure 19. CRC Command Sequence Diagram

6.9.2 Packets

6.9.2.1 Command Packet

SOH	(1 byte)	01h
LNH	(1 byte)	00h
LNL	(1 byte)	09h
CMD	(1 byte)	18h (CRC command)
SAD	(4 byte)	Start address for example 00004000h -> 00h, 00h, 40h, 00h
EAD	(4 byte)	End address for example 003FFFFFh -> 00h, 3Fh, FFh, FFh
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.9.2.2 Data Packet [CRC data]

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

6.9.2.3 Data Packet [status ERR]

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	02h
RES	(1 byte)	98h (ERR)
STS	(1 byte)	Status code
SUM	(1 byte)	Sum data
ETX	(1 byte)	03h

6.9.3 Processing procedure

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

(Refer to Command packet reception for details).

Boot firmware analyzes the received parameters after packet analysis.

- If the SAD is greater than EAD, the boot firmware sends an "Address error".
- If the SAD or EAD is outside the range specified in the area information, the boot firmware sends an "Address error".
- If SAD and EAD belong to different KOA, boot firmware will send an "Address error".
- If SAD and EAD are not 4 byte alignment, the boot firmware sends an "Address error".
- If the specified area is user area in Code Flash and also includes area outside of Access window, boot firmware sends a "Protection error".
- When the above error occurs, the boot firmware does not process and returns to the command waiting state.
  * Flash memory status does not change before command reception.

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.
  * Flash memory status does not change before command reception

6.9.4 Status Information from Microcontroller

(listed in descending order of priority)

Condition	STS
The received packet does not have ETX.	Packet error
SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error
LNH and LNL in the received packet do not comply with the packet format.	Packet error
LNH and LNL in the received packet do not comply with the specifications of this command.	Packet error
The phase is "Authentication phase".	Flow error
SAD is bigger than EAD.	Address error
SAD or EAD is outside the scope of accessible area specified in area information.	Address error
SAD and EAD belongs to different KOA.	Address error
SAD and EAD are not 4 byte alignment.	Address error
If the specified area is a user area in code flash and contains outside the scope of the access window	Address error

6.9.5 Precautions

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

7. Flow Example

7.1 Beginning Communication

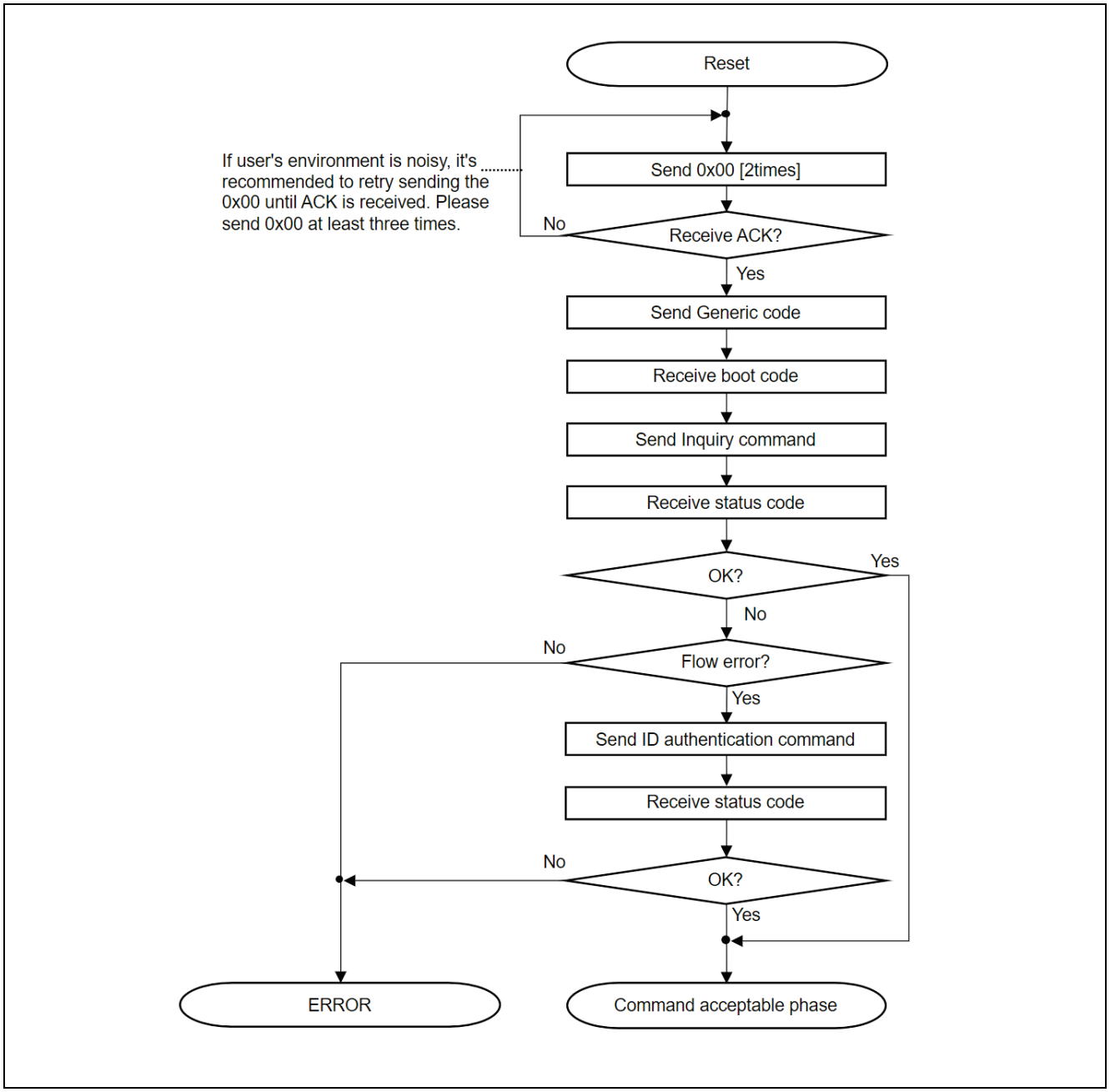

Figure 20. Beginning Communication

7.2 All Area Erasure

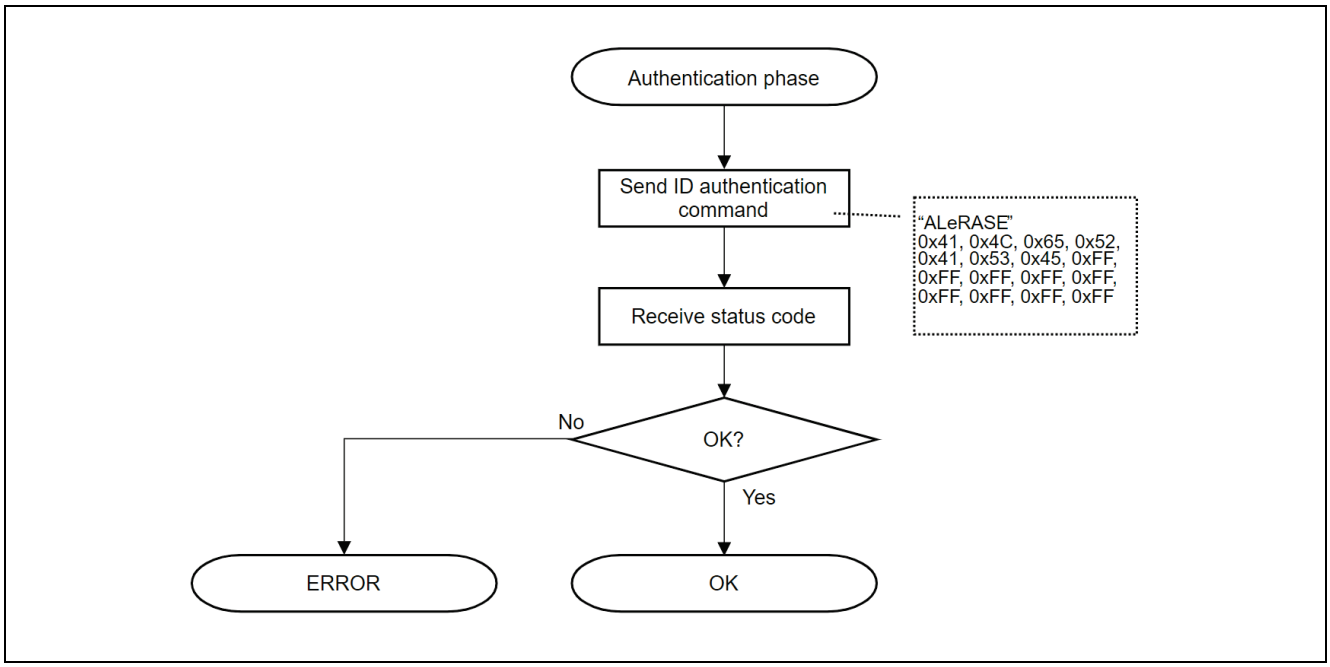

Figure 21. All Area Erasure

7.3 Acquisition of Device Information

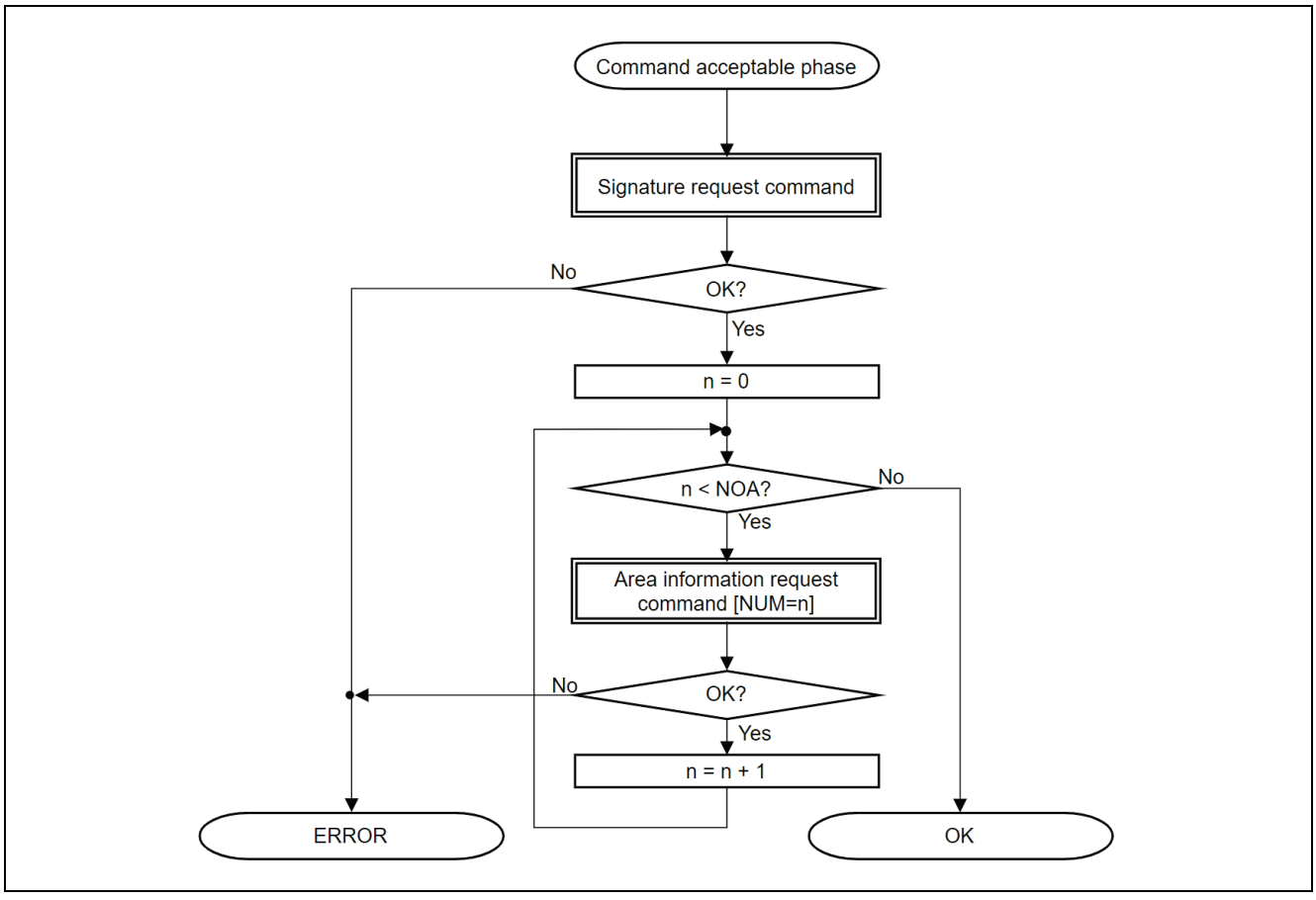

Figure 22. Acquisition of Device Information

7.4 User Area/ Data Area Updates

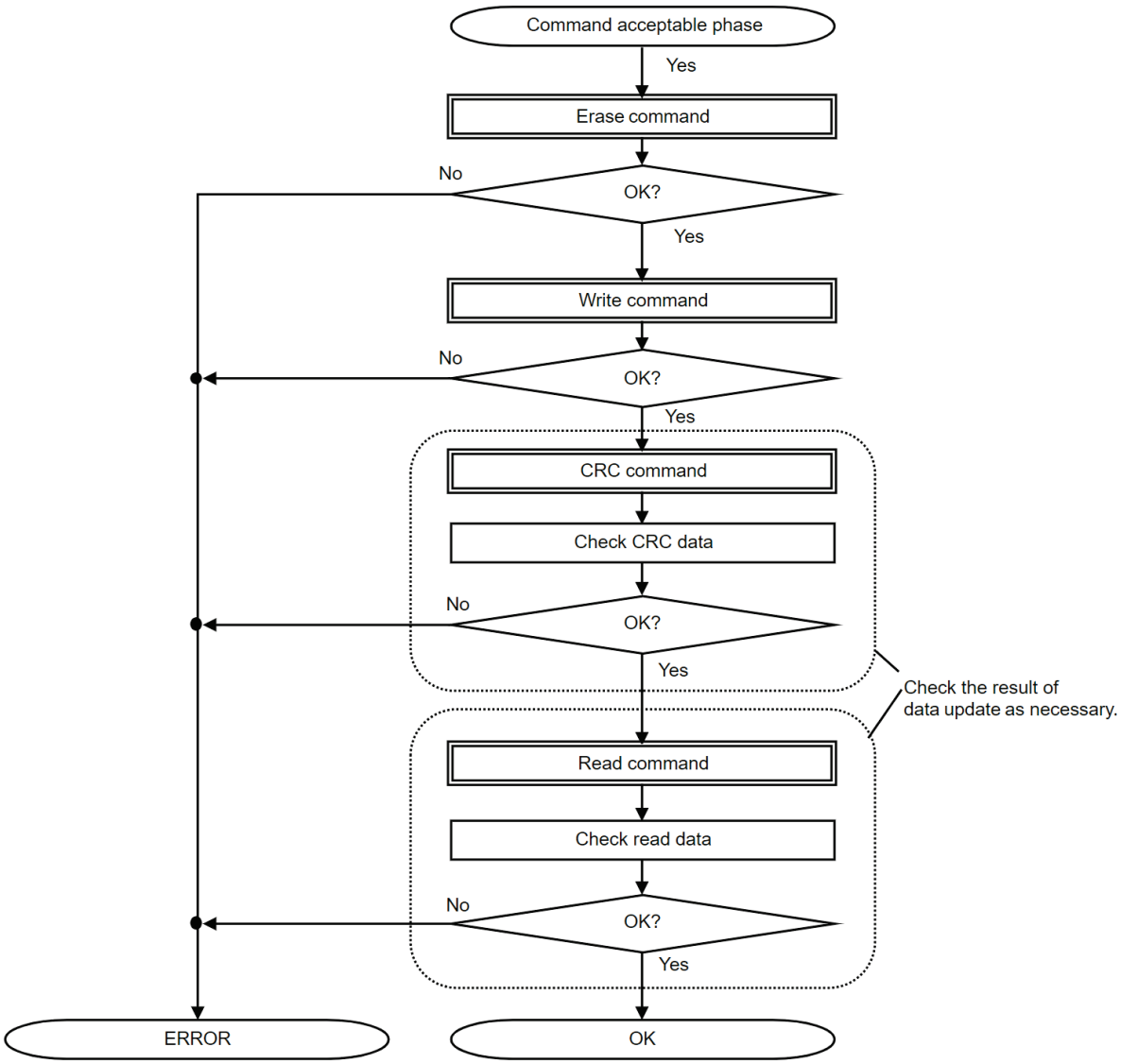

Figure 23. User Area/ Data Area Updates

7.5 Configuration Area Updates

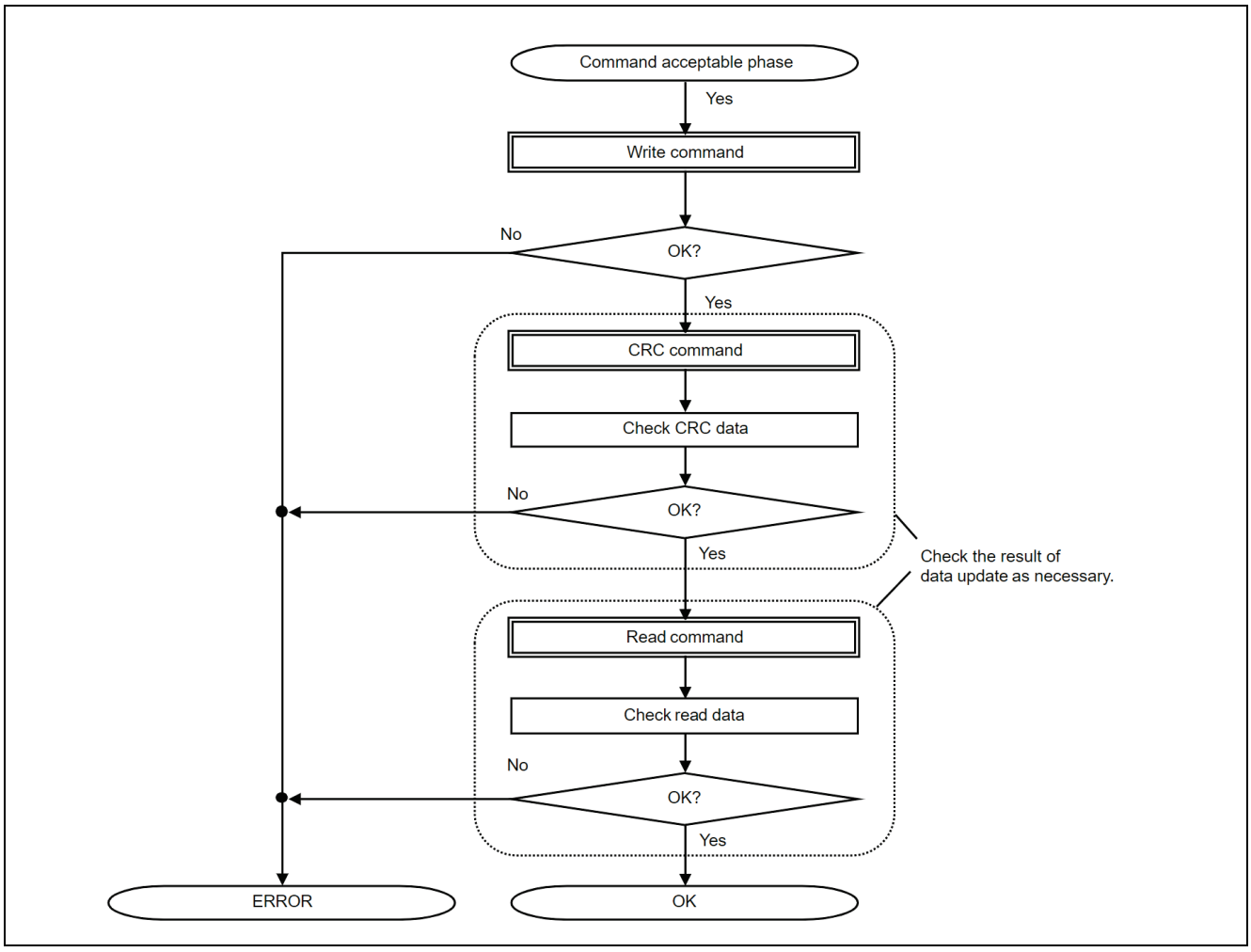

Figure 24. Configuration Area Updates

7.6 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.

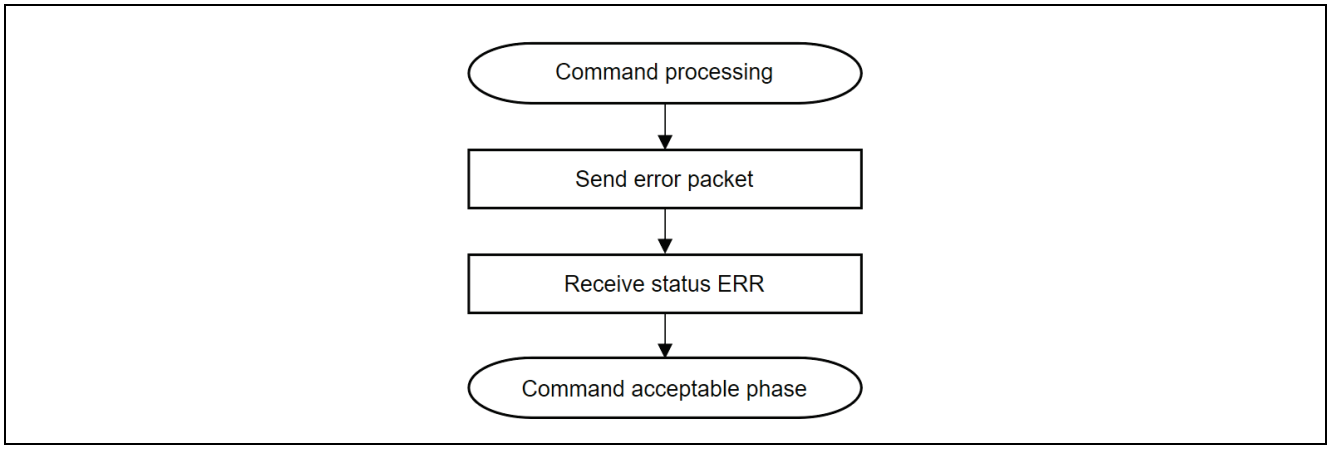

Figure 25. Command Cancel

For example, error packets to end the command.

Command	When to send error packets	Example of the error packet
Write command	Data packet [write data]	
Read command	Data packet [status OK]	

SOD	(1 byte)	81h
LNH	(1 byte)	00h
LNL	(1 byte)	01h
RES	(1 byte)	FFh (ERR)
SUM	(1 byte)	00h
ETX	(1 byte)	03h

8. AC Characteristics

8.1.1 Baud Rate Setting Command

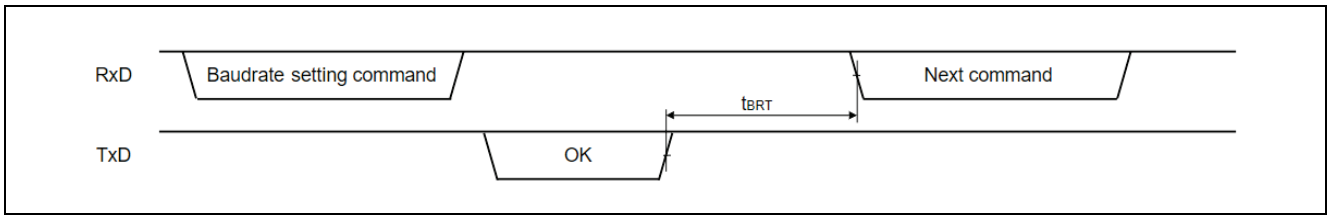

Figure 26. Baud Rate Setting Command

Parameter	Symbol	Min	Typ	Max	Unit
Baud rate setting time	tBRT	-	-	1	ms

9. Precaution List

9.1 Programming and Erasure Restrictions

1. The following cannot be executed when FAPR is 0:
   - Writing to access window start/end address setting.
   - Writing to Customer ID
   - Writing to Customer ID key address
   - Writing to Customer ID key data
   - All erasure by Authentication command with ALeRASE ID
   - All erasure by debug mode

   There is no way to write FAPR back to 1 once after writing 0.
2. When writing 0 to FAPR and writing protected data[*1] are executed by one Write command, Sequencer error occurs at writing to protected data[*1] for the following reasons:
   - FAPR is placed at smaller address than the protected data[*1].
   - It is impossible to write FAPR and protected data[*1] by one sequencer command.
     - Protection by FAPR is enabled before writing protected data[*1].

   To avoid this, execute write command two times to write FAPR and the protected data[*1]:
   1st write command: Writing to the protected data [*1]
   2nd write command: Writing to FAPR=0 (recommended to execute at very last step during production programming)

   Note *1: The following are protected data:
   - Access window setting
   - Customer ID
   - Customer ID key address
   - Customer ID key data

3. OCDDIS cannot be written back from 0 to 1 by Write command (the value remains 0 even when Write command normally finishes). Execute all erasure by Authentication command to write OCDDIS back to 1.

9.2 CRC Command

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

10. Website and Support

Visit the following URLs to learn about key elements of the RA family, download components and related documentation, and get support:

RA Product Information	renesas.com/ra
RA Product Support Forum	renesas.com/ra/forum
RA Flexible Software Package	renesas.com/FSP
Renesas Support	renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb.20.24	—	Initial release

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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