Renesas RA Family

System Specifications for Standard Boot Firmware

Introduction

This document describes the specification of standard boot firmware for Renesas RA microcontrollers. This document assumes that the reader has an understanding of specifications for Renesas RA products, FPSYS/FACI, and FCB.

Target Device

RA2L1, RA2E1, and RA2E2 Groups

Contents

1. Definition of Terms

The terminology used in this document is defined in this section.

1.1 Flash Memory

The ROM area where program code is written is called **Code Flash (FLI)**. The ROM area where data is written is called **Data Flash (FLD)**. Both are called Flash memory. The area used by the user is called User area. The area to store configuration data is called Config area.

An example of flash memory structure is shown in the following graphic. Memory structure will differ from device to device.

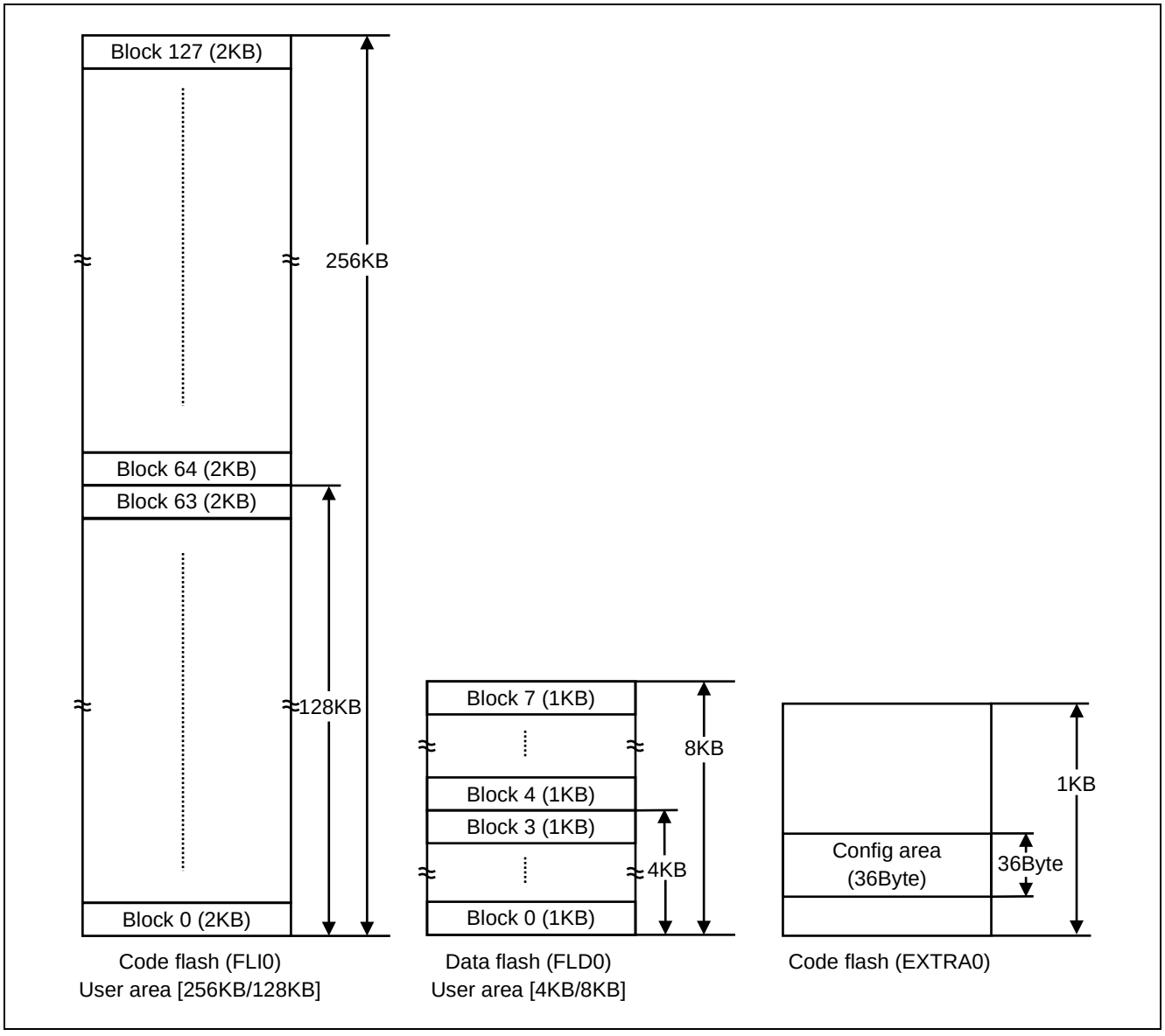


Figure 1. Flash Memory Structure Example

1.2 Boot Firmware

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

1.3 SCI: Serial Communication Interface

The interface module which performs serial communication is called the **Serial Communication Interface (SCI)**. This boot firmware uses the common SCI configured as UART (Universal Asynchronous Receiver/Transmitter).

1.4 AW: Access Window

The function that assigns accessible sectors to erase or write in code flash is called the **Access Window** (**AW)**. This function allows access from start sector to end sector and disallows access to other areas. Users set a value corresponding to the “start sector” to the FAWS (bit of Access Window Start Address) and set a value corresponding to “end sector + 1” to the FAWE (bit of Access Window End Address). If FSPR (bit of Access Window Protection Flag) is 0, FAWS and FAWE cannot be changed. For details, please refer to the *Renesas RA Flash Memory Programming Application Note*.

Table 1. Example of Flash Memory Structure

Sector No.	Base address	Size	Setting value
0	00000000h	2KB	000h
1	00000800h	2KB	001h
2	00001000h	2KB	002h
3	00001800h	2KB	003h
4	00002000h	2KB	004h
5	00002800h	2KB	005h
6	00003000h	2KB	006h
7	00003800h	2KB	007h
8	00004000h	2KB	008h
9	00004800h	2KB	009h
10	00005000h	2KB	00Ah
11	00005800h	2KB	00Bh
:	:	:	:
126	0003F000h	2KB	07Eh
127	0003F800h	2KB	07Fh

In case of FAWS=007h, FAWE=014h:

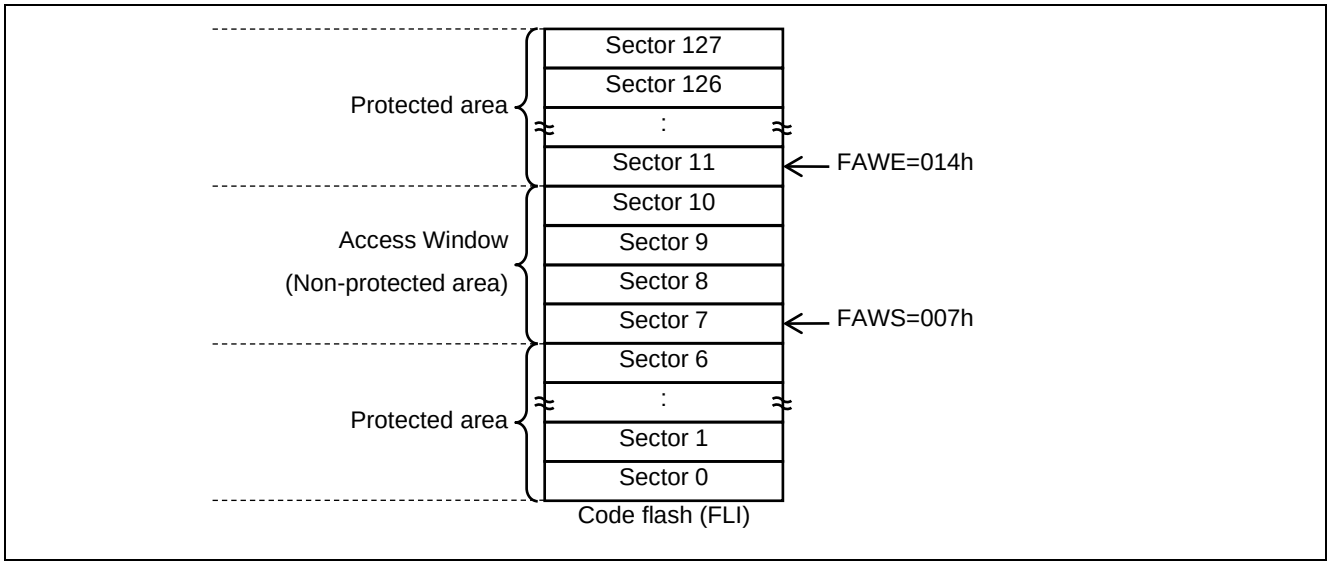


Figure 2. Access Window Example: FAWS = 007h, FAWE = 014h

2. System Architecture

Boot firmware has a serial programming interface to send and receive flash control commands between the microcontroller and the flash programmer in Serial programming mode. In Debug mode, boot firmware only erases the User area and Config area. Boot firmware is embedded in the device.

2.1 Renesas RA2L1/RA2E1/RA2E2 Groups

Table 2. Serial Programming

Operating mode	VCC=2.7 - 5.5V	High-speed mode
	VCC=1.8 - 2.7V	Middle-speed mode
	VCC=1.6 - 1.8V	Low-speed mode
Supported communication	two-wire UART	
Main-OSC input	Unnecessary	
Operating voltage condition	VCC = 1.6V – 5.5V	

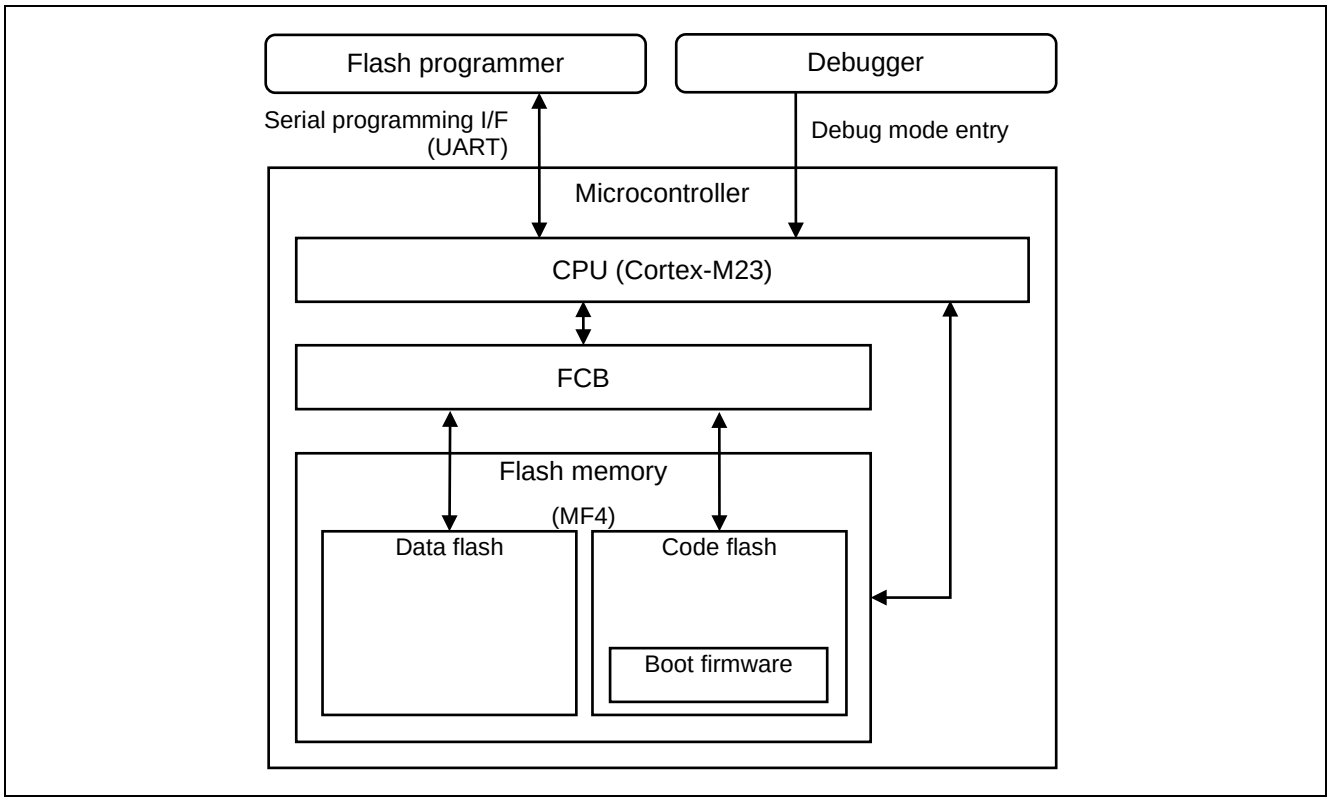


Figure 3. Flash System Architecture

2.2 Mode Entry

Boot firmware determines if the operating mode is Serial programming mode or Debug mode using the MD pin level reset timing. If operating mode is Serial programming mode, boot firmware transits to **Communication setting phase**. If operating mode is not Serial programming mode, boot firmware erases all User area and Config area, and then goes into an infinite loop. If Security MPU is valid, boot firmware goes into an infinite loop without transition to either mode.

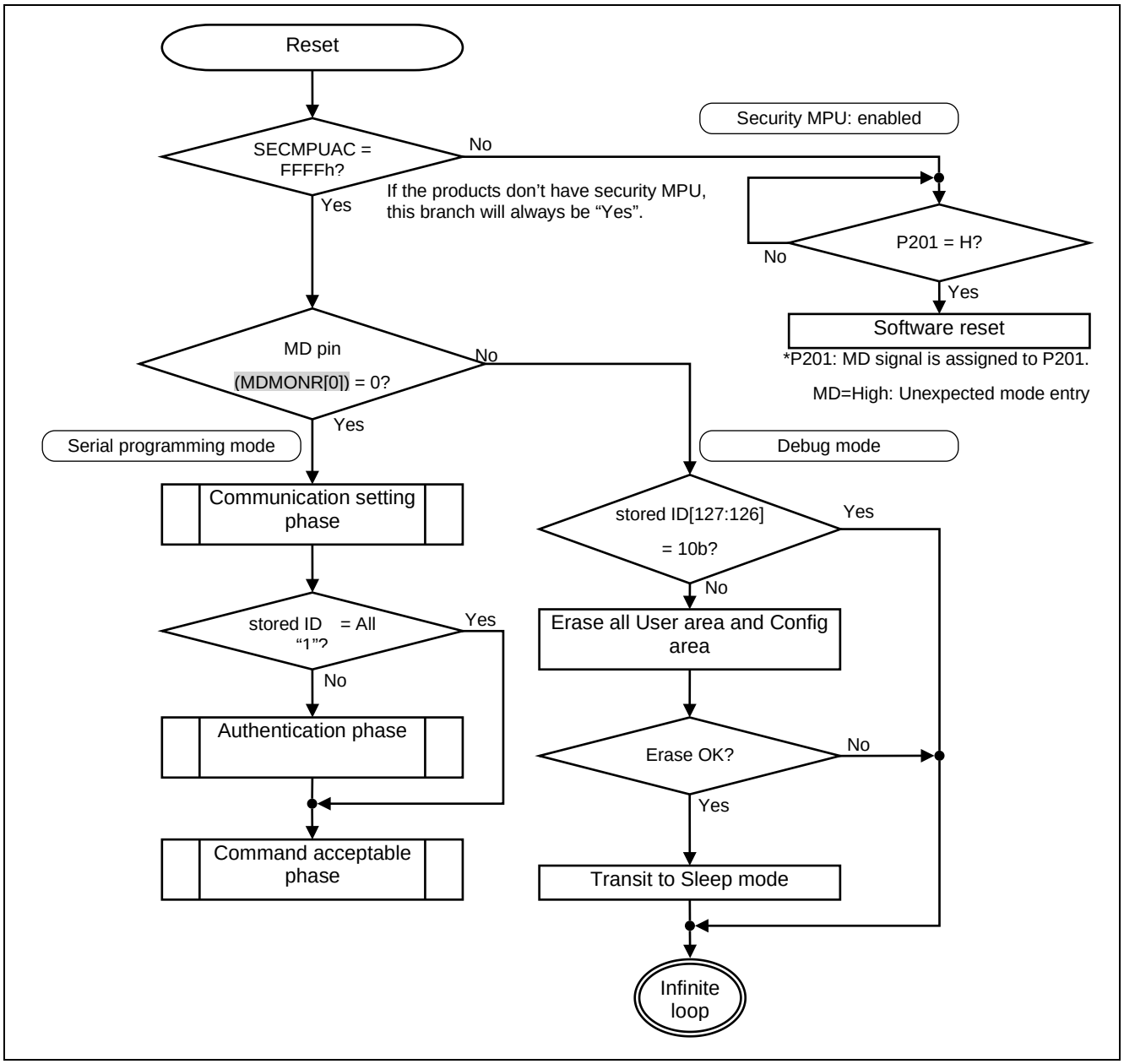


Figure 4. Flash Modes

3. Serial Programming Interface

3.1 Communication Mode

Boot firmware has an interface for the following communication mode. The flowchart of the mode decision is shown in section 3.2.1,Communication Setting Phase.

3.1.1 Two-wire UART Communication

Boot firmware supports two-wire UART communication.

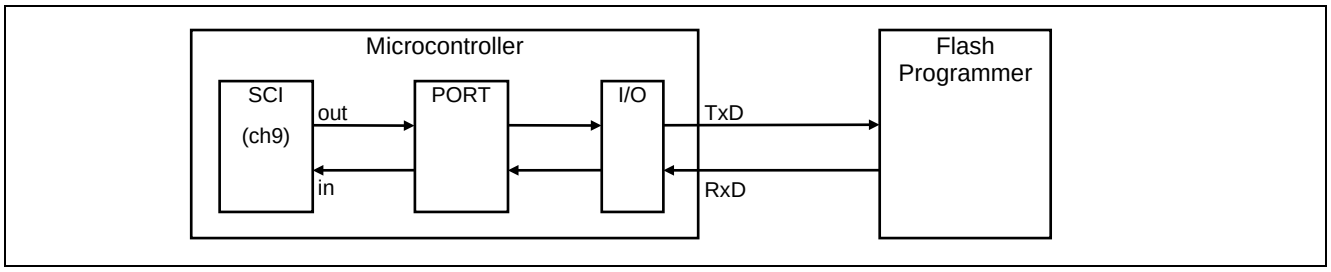


Figure 5. Two-wire UART Communication

When the operating mode is Serial programming mode, boot firmware sets RxD to port-in for communication mode detection. Boot firmware selects the UART communication and initializes the SCI by the detecting the falling edge of RxD.

Table 3. UART Settings

Interface	SCI ch9
RxD	Reception and Transmission mode
TxD	Reception and Transmission mode
Baud rate	9600 bps
Data length	8 bit (LSB first)
Parity bit	None
Stop bit	1 bit

Communication speed is 9600 bps until the baud rate setting command is completed. The communication speed is changed to the intended rate after successful completion of the baud rate setting command as shown in the following graphic.

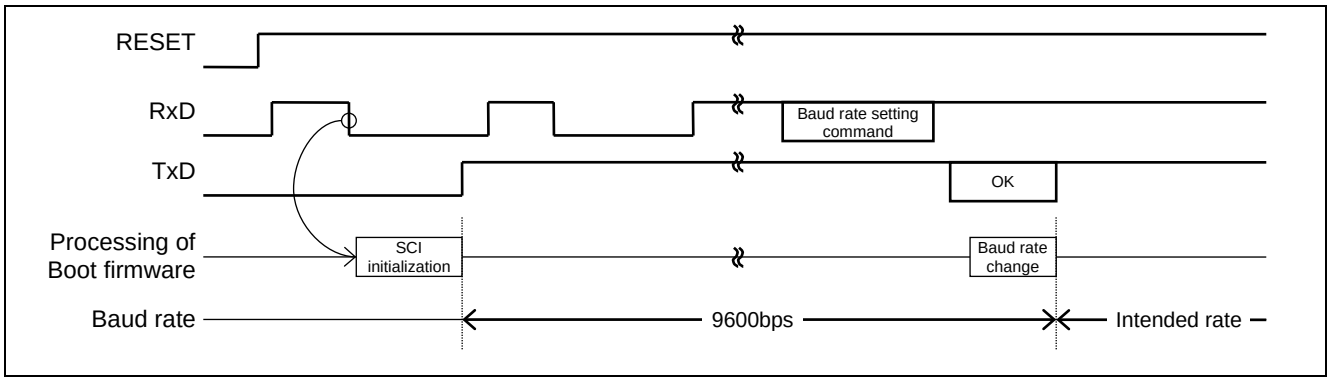


Figure 6. Baud Rate Setting

Note: If the UART communication cable is unplugged during transmission, there is no guarantee of operations that follow.

3.2 Operating Procedures

3.2.1 Communication Setting Phase

After reset release, the boot firmware determines the communication mode (two-wire UART) as shown in the following flowchart. After the communication setting and the receipt of Generic code through the selected communication method, boot firmware transitions to the **Authentication phase**. However, if ID-code stored in the device is all “1”, the boot firmware transitions to the **Command acceptable phase** directly.

3.2.1.1 Selection of Communication Mode

Table 4. Communication Mode Selection

Condition	Communication mode
Detection of High level in MD (P201) : Single chip mode	Unexpected mode entry (-> Software reset)
Detection of a falling edge in RxD	UART mode

*P201: MD signal is assigned to P201

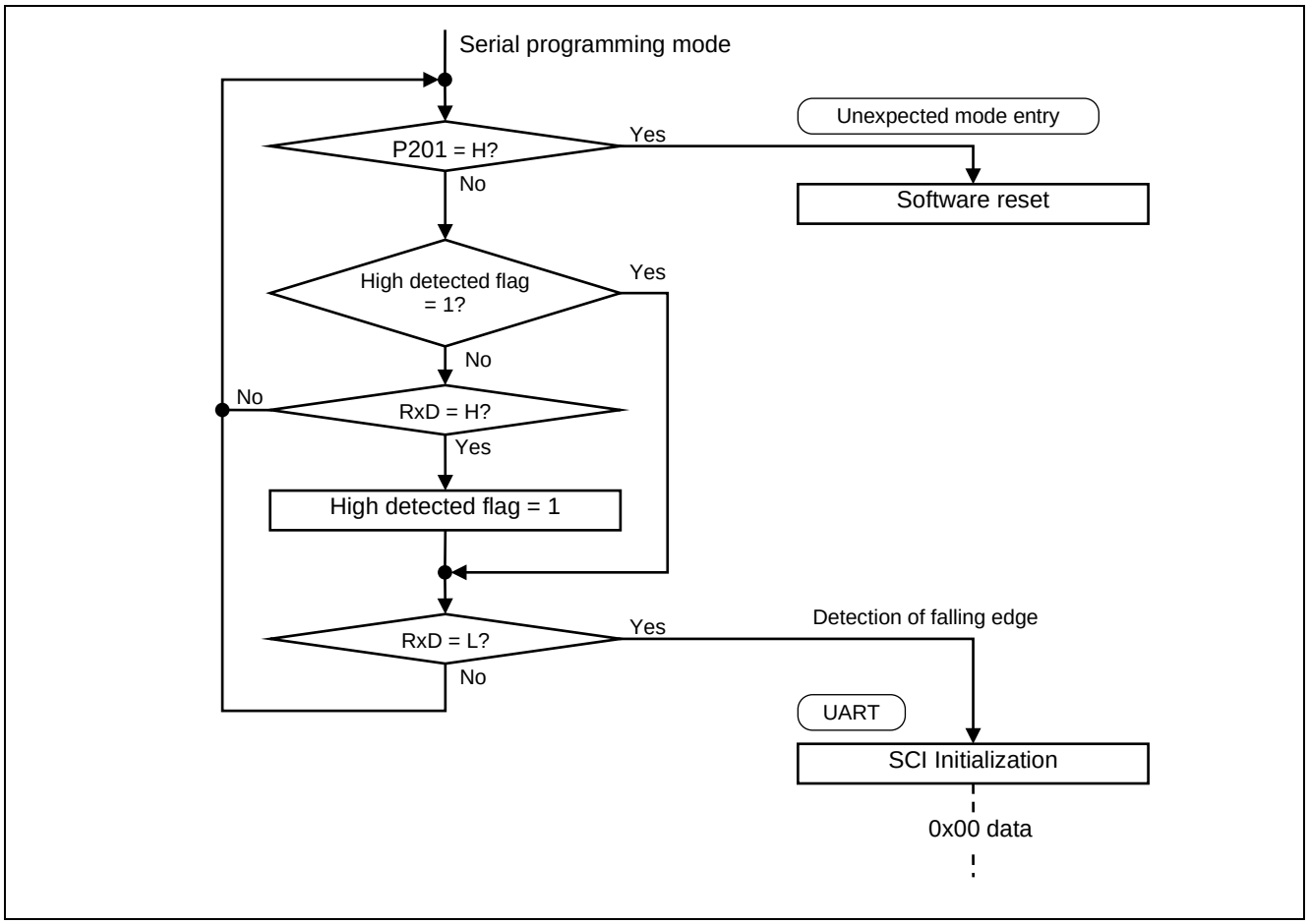


Figure 7. Selecting Communication Mode

3.2.1.2 Setting Up Two-wire UART Communication

To use the UART communication, flash programmer sends 0x00 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 selects the UART communication and initializes the SCI on detection of falling edge in RxD. After that, the boot firmware receives the 2nd Low pulse as 0x00 data in SCI, then returns the ACK, receives the Generic code, then returns the Boot code. This completes communication setting.

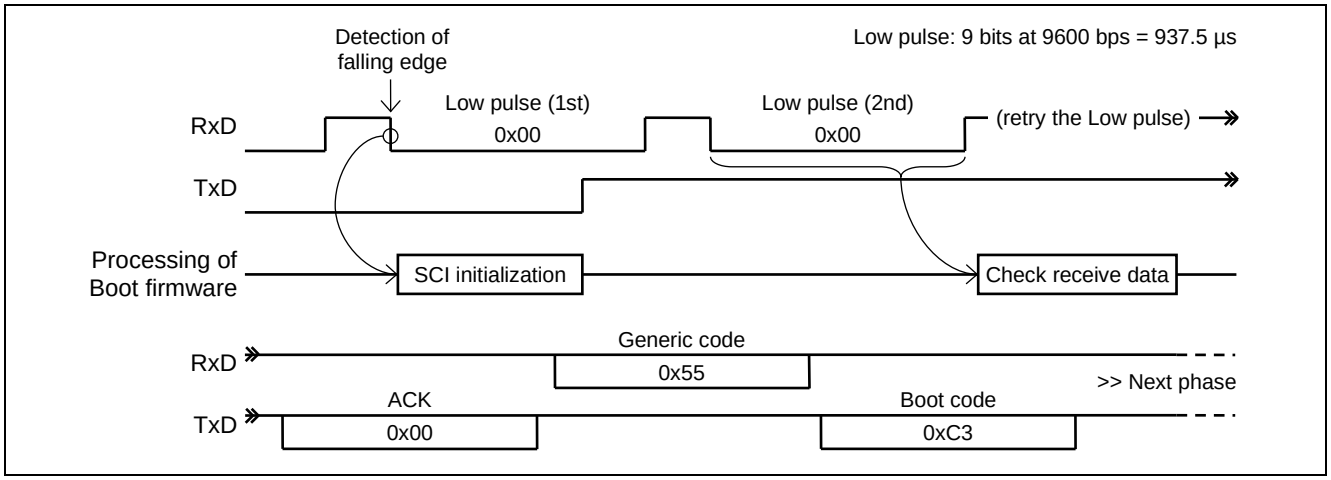


Figure 8. Two-wire UART Set Up

3.2.2 Authentication Phase

Boot firmware authenticates ID code in this phase. This phase can only accept the Authentication command. If the Authentication command passes successfully, boot firmware transits to the **Command acceptance phase**.

3.2.3 Command Acceptance Phase

This phase can accept all commands except the Authentication command. The flash programmer can judge if the current phase is **Command acceptance phase** or **Authentication phase** by the result of an Inquiry command.

3.2.4 State Transitions

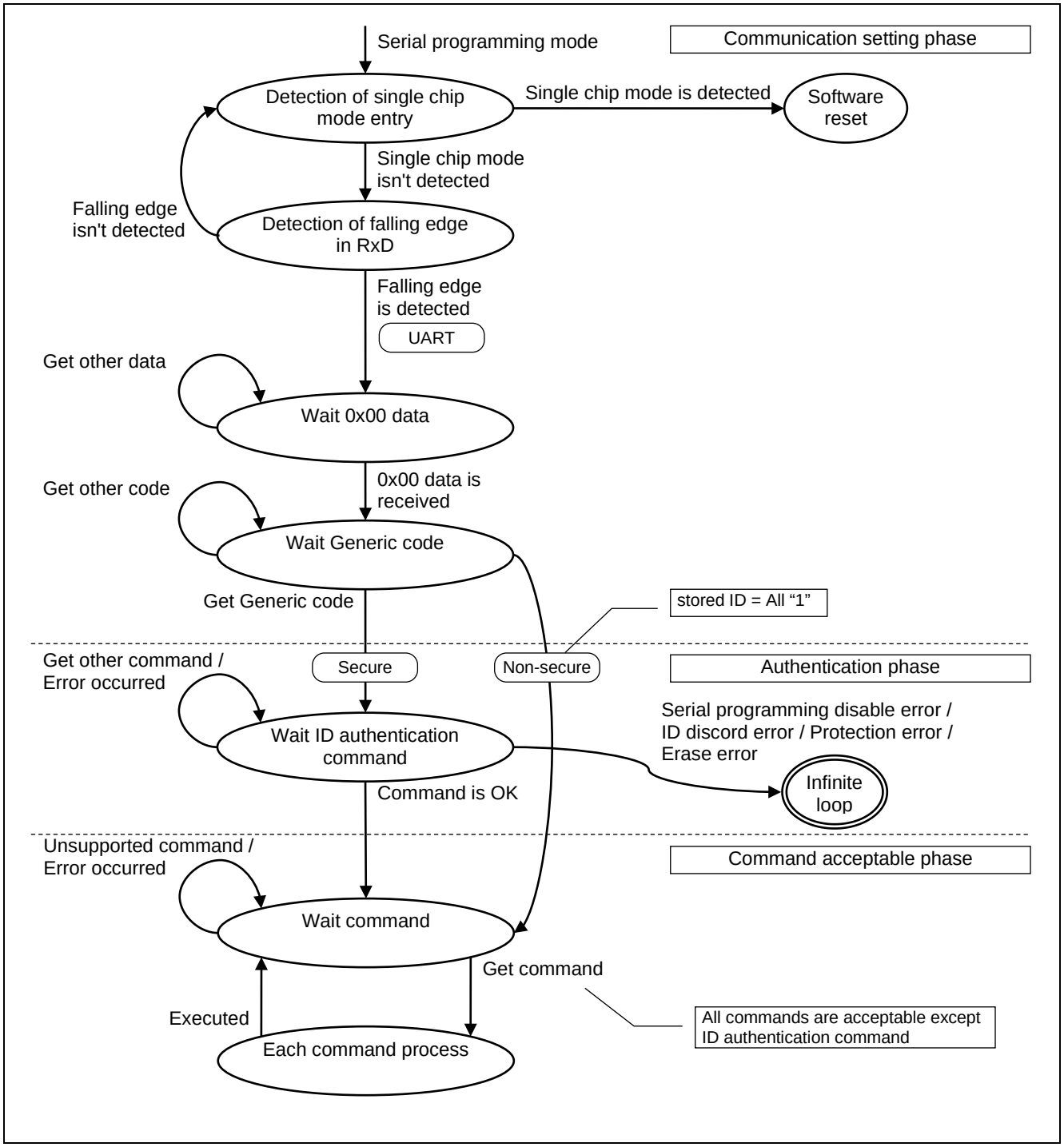


Figure 9. State Transitions

3.2.5 Beginning Communication

3.2.5.1 If ID Code is Already Stored [Secure]

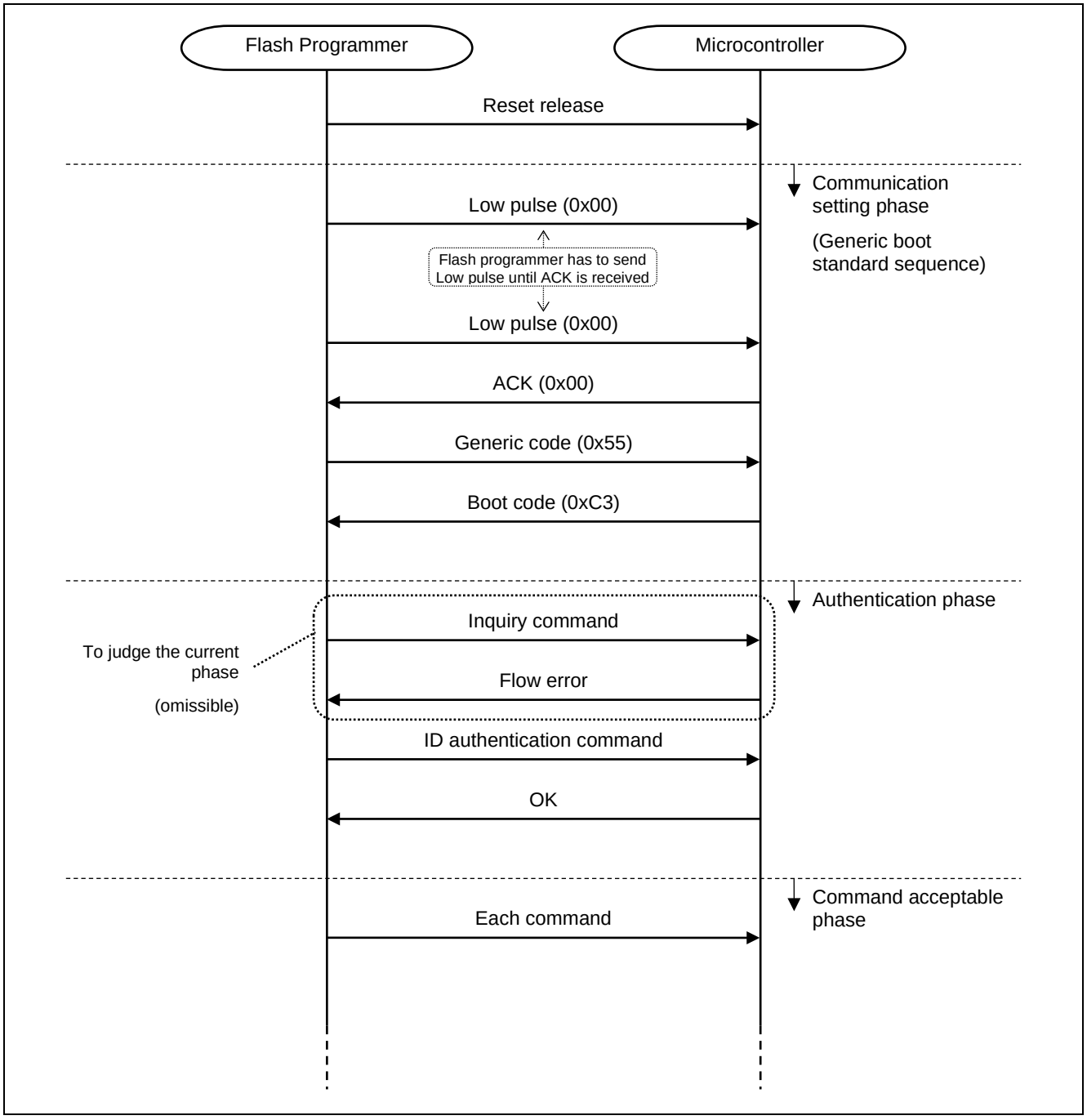


Figure 10. Beginning Communication (Secure)

3.2.5.2 If ID Code Is Not stored [Non-secure]

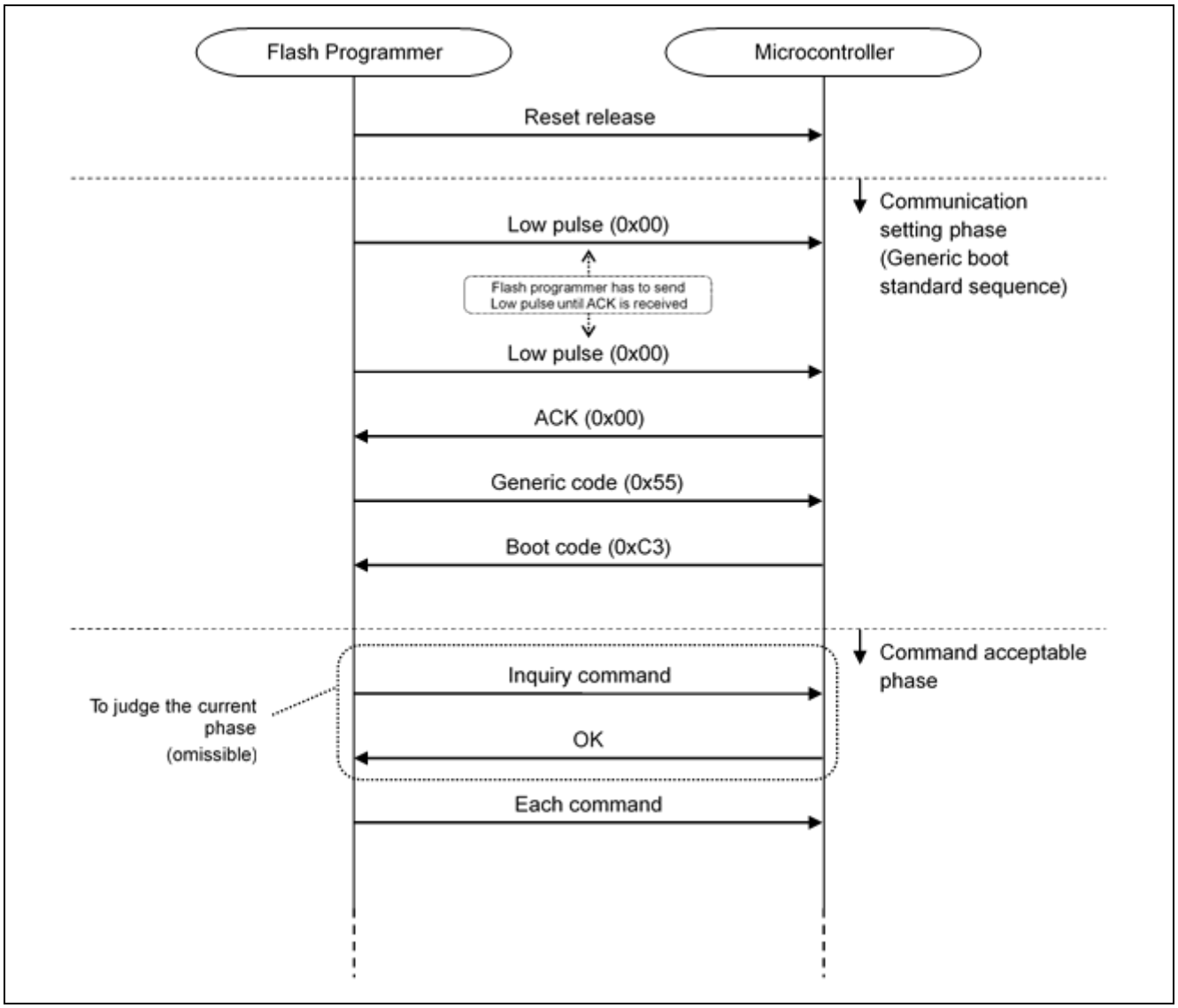


Figure 11. Beginning Communications (Non-secure)

3.3 Packet Format

Be sure to follow this format. If the boot firmware receives a packet of more than 1024 bytes, the boot firmware may not be able to reply.

3.3.1 Command Packet

The flash programmer sends data of a command packet to the microcontroller in the following format.

SOH	LNH	LNL	COM	Command information (flexible length [max:255 bytes])	SUM	ETX

Symbol	Size	Code	Description
SOH	1 byte	0x01	Start of command packet
LNH	1 byte	-	Packet length (Length = “COM + Command information”) [High]
LNL	1 byte	-	Packet length (Length = “COM + Command information”) [Low]
COM	1 byte	-	Command code (refer to section 3.4.1)
Command information	-	-	Command information [Max: 255 bytes] Example: For write command: Write destination address For erase command: Erase target address
SUM	1 byte	-	Sum data of “LNH + LNL + COM + Command information” (expressed as two's complement) Example: LNH + LNL + COM + Command information(1) + Command information(2) + ... + Command information(n) + SUM = 0x00
ETX	1 byte	0x03	End of packet

If the packet length is 0 or over 1024, the value of RES will not be defined.

3.3.2 Data Packet

The flash programmer sends the data packet to the microcontroller in the following format. The boot firmware uses the same format when it sends the data packet to the programmer.

SOD	LNH	LNL	RES	Data (flexible length [max:1024byte])	SUM	ETX

Symbol	Size	Code	Description
SOD	1byte	0x81	Start of data packet
LNH	1byte	-	Packet length (Length of “RES + Data”) [High]
LNL	1byte	-	Packet length (Length of “RES + Data”) [Low]
RES	1byte	-	Response code (refer to 0)
Data	-	-	Transmit data [Max : 1024byte] e.g.) In case of Write command : Write data
SUM	1byte	-	Sum data of “LNH + LNL + RES + Data” (express as two's complement) e.g.) LNH + LNL + RES + Data(1) + Data(2) + ... + Data(n) + SUM = 0x00
ETX	1byte	0x03	End of packet
In case the packet length is 0 or over 1025, RES will be an indefinite value.			

3.4 Communication Command

3.4.1 List of Command Codes

Table 5. Command Codes

Command code	Command name	Description
0x00	Inquiry command	Return ACK (to determine the current phase)
0x12	Erase command	Erase data on target area
0x13	Write command	Write data on target area
0x15	Read command	Read data on target area
0x30	ID authentication command	Authenticate ID for connection with the device
0x34	Baud rate setting command	Set baud rate for UART
0x3A	Signature request command	Get signature information
0x3B	Area information request command	Get area information

3.4.2 List of Status Codes

Table 6. Status Codes

Status code	Description
0x00 | Command code	OK (ongoing normally): used in Response code [RES]
0x80 | Command code	ERR (occurrence of an error): used in Response code [RES]
0x00	OK (successful completion)
0xC0	Unsupported command error
0xC1	Packet error (Illegal length, Missing ETX, and so forth)
0xC2	Checksum error
0xC3	Flow error
0xD0	Address error
0xD4	Baud rate margin error
0xDA	Protection error
0xDB	ID mismatch error
0xDC	Serial programming disable error
0xE1	Erase error (*1)
0xE2	Write error (*1)
0xE7	Sequencer error (*1)

*1: For Erase command or Write command, boot firmware checks status registers of flash sequencer (FCB) and returns following status.

Status	Flash process	Condition
Erase error	RA2L1/RA2E1/RA2E2 Group	If ERERR is detected
Write error	RA2L1/RA2E1/RA2E2 Group	If PRGERR or PRGERR01 are detected
Sequencer error	RA2L1/RA2E1/RA2E2 Group	If ILGLERR or EILGLERR are detected

3.4.3 Executable Command in Each Phase

Table 7. Executable Commands

Command	Communication setting phase	Authentication phase	Command acceptance phase
Inquiry command	NG (*1)	Flow error	OK
Erase command	NG (*1)	Flow error	OK
Write command	NG (*1)	Flow error	OK
Read command	NG (*1)	Flow error	OK
ID authentication command	NG (*1)	OK	Flow error
Baud rate setting command	NG (*1)	Flow error	OK
Signature request command	NG (*1)	Flow error	OK
Area information request command	NG (*1)	Flow error	OK

*1: Boot firmware does not return any data packet.

3.4.4 Unsupported Command

If boot firmware receives the command packet of a command code that is not defined in section 3.4.1, it returns the unsupported command error, then goes back to the “Wait for command” state.

3.4.4.1 Command Processing Procedure

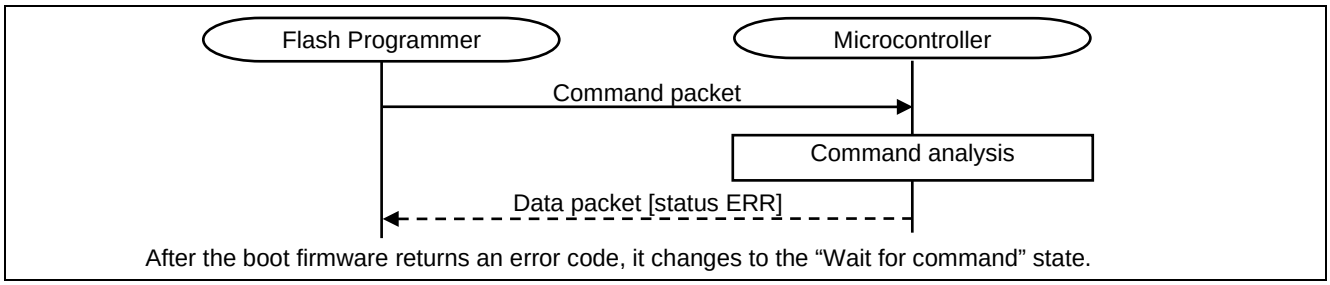


Figure 12. Unsupported Command Processing

3.4.4.2 Command Packet

SOH	LNH	LNL	COM	SUM	ETX

SOH	(1 byte)	0x01
LNH	(1 byte)	Length high
LNL	(1 byte)	Length low
COM	(1 byte)	Command code not defined in section 3.4.1
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.4.3 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x80 | COM (ERR)
STS	(1 byte)	Status code 0xC0 (Unsupported command error) 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.4.4 Status Code from Microcontroller [Priority high: 1 -> low: 10]

Table 8. Unsupported Command Status Codes

Condition	Status	Priority	Code
If COM in the received packet is undefined code.	Unsupported command error	4	0xC0
If LNH and LNL in the received packet are different from defined values.	Packet error	3	0xC1
If the received packet does not have ETX.		1	
If SUM in the received packet is different from the value calculated by the boot firmware.	Checksum error	2	0xC2

3.4.5 Inquiry Command

The Inquiry command is used to check if boot firmware is in the Command acceptance phase or not. This command can be performed only in the Command acceptance phase.

3.4.5.1 Command Processing Procedure

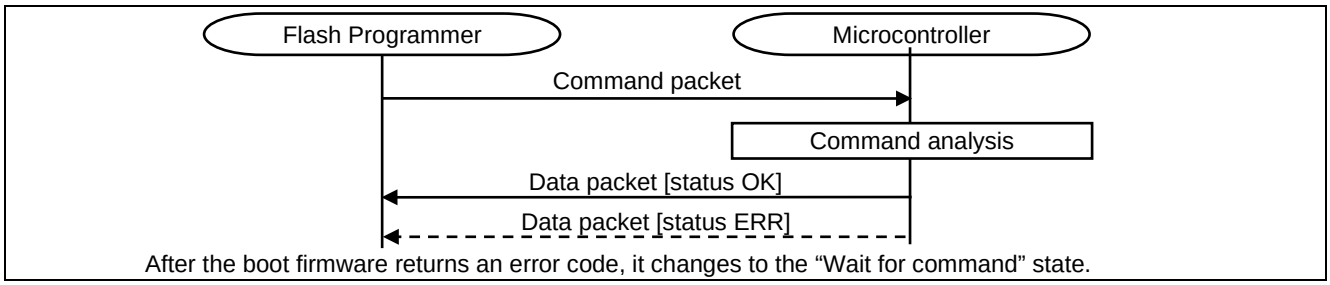


Figure 13. Inquiry Command Processing

3.4.5.2 Command Packet

SOH	LNH	LNL	COM	SUM	ETX

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

3.4.5.3 Data Packet [Status OK]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x00 (OK)
STS	(1 byte)	Status code 0x00 (OK)
SUM	(1 byte)	0xFE
ETX	(1 byte)	0x03

3.4.5.4 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x80 (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.5.5 Status Code from Microcontroller [Priority High: 1 -> low: 10]

Table 9. Inquiry Status Codes

Condition	Status	Priority	Code
If the state is **Command acceptance phase**	OK	5	0x00
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is **Authentication phase**	Flow error	4	0xC3

3.4.6 Erase Command

The Erase command erases data in the designated area of the flash memory. The alignment of the target addresses follows 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 can be performed only in **Command acceptance phase**.

3.4.6.1 Command Processing Procedure

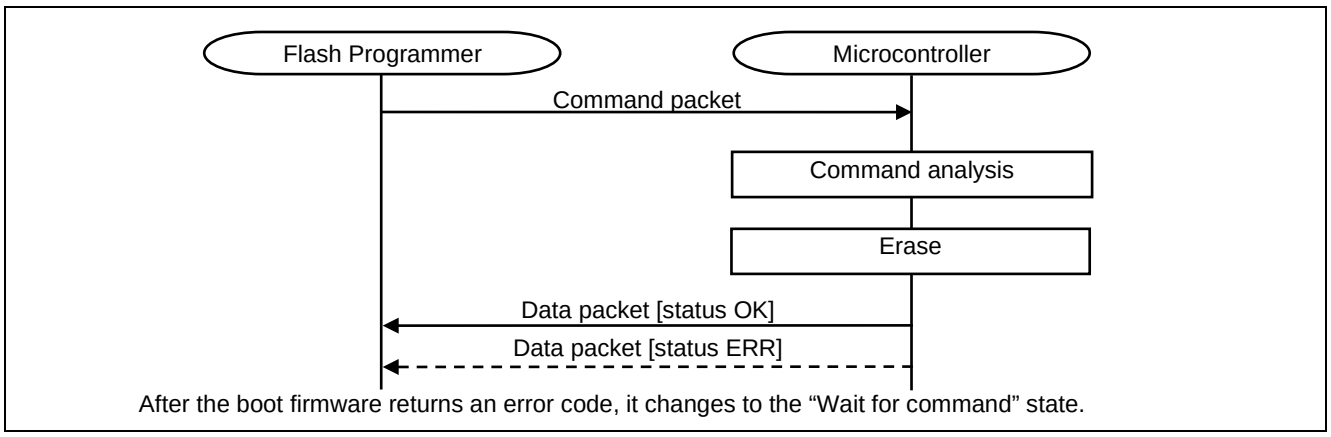


Figure 14. Erase Command Processing

3.4.6.2 Command Packet

SOH	LNH	LNL	COM	SAD	EAD	SUM	ETX

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x09
COM	(1 byte)	0x12 (Erase command)
SAD	(4 bytes)	Start address Example: 0000_4000h -> 0x00, 0x00, 0x40, 0x00
EAD	(4 bytes)	End address Example: 003F_FFFFh -> 0x00, 0x3F, 0xFF, 0xFF
SUM	(1 bytes)	Sum data
ETX	(1 bytes)	0x03

3.4.6.3 Data Packet [Status OK]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x12 (OK)
STS	(1 byte)	Status code 0x00 (OK)
SUM	(1 byte)	0xEC
ETX	(1 byte)	0x03

3.4.6.4 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x92 (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error) 0xD0 (Address error) 0xDA (Protection error) 0xE1 (Erase error) 0xE2 (Write error) 0xE7 (Sequencer error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.6.5 Status Code from Microcontroller [Priority High: 1 -> Low: 10]

Table 10. Erase Status Codes

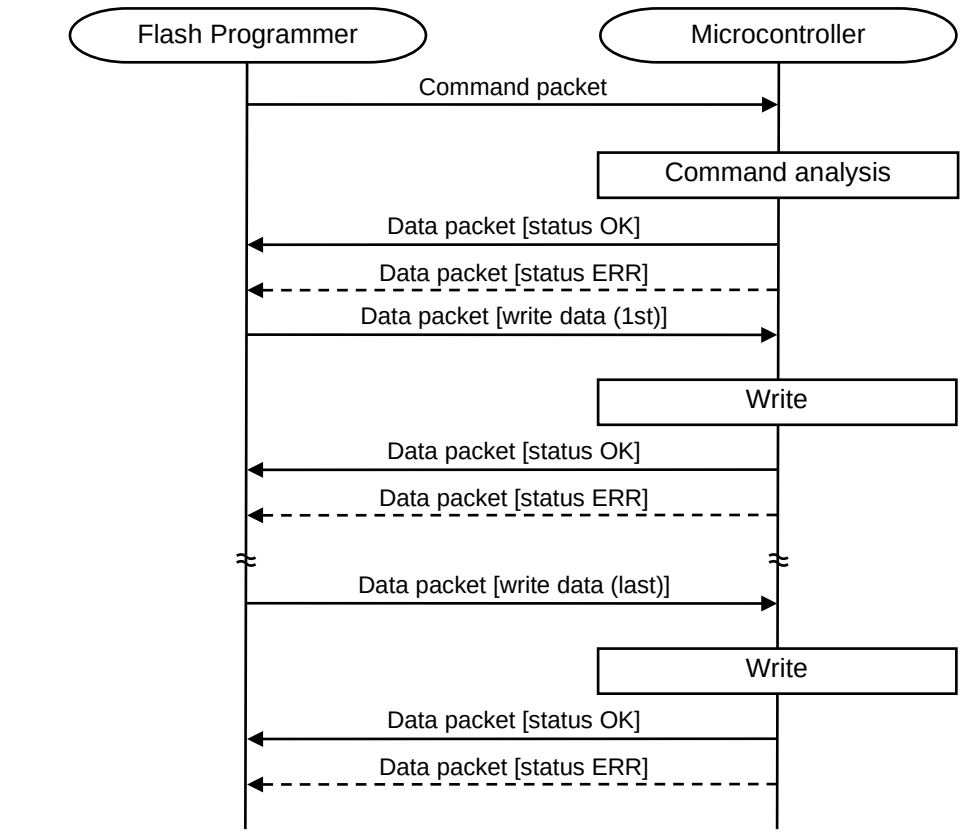
Condition	Status	Priority	Code
Successful completion	OK	8	0x00
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is “Authentication phase”	Flow error	4	0xC3
If the start address or the end address does not belong to any areas *1	Address error	5	0xD0
The start address is in a different area from the end address *1		5	
This command isn't available for this area		5	
The start address is bigger than the end address		5	
If the start address or the end address does not match the alignment of the area		5	
If the target area is not entirely within the Access Window in the case of User area in code flash	Protection error	6	0xDA
If the target area is Config area, and the FSPR bit is 0		6	
If the erase error occurs	Erase error	7	0xE1
If the write error occurs	Write error	7	0xE2
If the sequencer error occurs	Sequencer error	7	0xE7

*1: Scope of each area is subject to area information request command

3.4.7 Write Command

The Write command receives write data from the flash programmer and writes those data to the flash memory. The alignment of the target address follows the area information returned by the Area information request command. Writes are executed in order from start address to end address by the write access unit. This command can be performed only in the **Command acceptance phase**.

3.4.7.1 Command Processing Procedure



After the boot firmware returns an error code, it changes to the Wait for command state.

Figure 15. Write Command Processing

3.4.7.2 Command Packet

SOH	LNH	LNL	COM	SAD	EAD	SUM	ETX

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x09
COM	(1 byte)	0x13 (Write command)
SAD	(4 bytes)	Start address Example: 0000_4000h -> 0x00, 0x00, 0x40, 0x00
EAD	(4 bytes)	End address Example: 003F_FFFFh -> 0x00, 0x3F, 0xFF, 0xFF
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.7.3 Data Packet [Write Data]

SOD	LNH	LNL	RES	DAT	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	N + 1 (High)
LNL	(1 byte)	N + 1 (Low)
RES	(1 byte)	0x13 (OK)
DAT	(N bytes)	Write data
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

N = 1–1024

3.4.7.4 Data Packet [Status OK]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x13 (OK)
STS	(1 byte)	Status code 0x00 (OK)
SUM	(1 byte)	0xEB
ETX	(1 byte)	0x03

3.4.7.5 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x93 (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error) 0xD0 (Address error) 0xDA (Protection error) 0xE1 (Erase error) 0xE2 (Write error) 0xE7 (Sequencer error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.7.6 Status Code from Microcontroller [Priority High: 1 -> low: 10]

Table 11. Write Status Codes

Condition	Status	Priority	Code
Successful completion	OK	9	0x00
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If the amount of data of all data packets received from flash programmer is beyond the number of write data designated by the command packet		7	
If the write length does not match the write access unit of the area		7	
If the RES of received data packet isn't OK		7	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is “Authentication phase”	Flow error	4	0xC3
If the start address or the end address does not belong to any areas *1	Address error	5	0xD0
If the start address is in a different area from the end address *1		5	
If this command isn't available for this area		5	
If the start address is bigger than the end address		5	
If the start address or the end address does not match to the alignment of the area		5	
If the target area is not entirely within the Access Window in the case of User area in code flash	Protection error	6	0xDA
If the target area contain FAWS or FAWE or BTFLG in Config area, and the FSPR bit is 0		6	
If the erase error occurs	Erase error	8	0xE1
If the write error occurs	Write error	8	0xE2
If the sequencer error occurs	Sequencer error	8	0xE7

*1: Scope of each area is subject to area information request command

3.4.7.7 Basic Precautions for Write Command

If you write any value except 0xFFFF to SECMPUAC, boot firmware will hang after the next reset release. See section 2.2.

If you write 0 to ID[127] in Config area, the device cannot perform ID authentication. ID authentication command will return the serial programming disable error. See section 3.4.9.

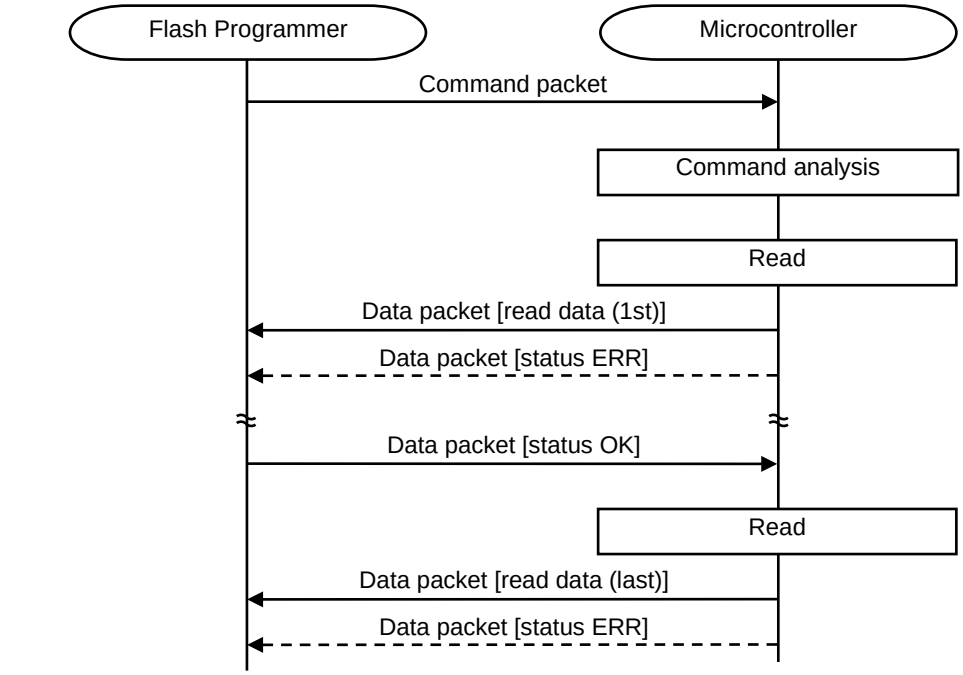
If you write 0 to ID[126] in Config area, the device cannot perform total area erasure by ID authentication command. See section 3.4.9.

In addition, device configuration data may be assigned in the Option-Setting Memory area and the Config area. The result of writing to an unassigned address depends on each device. Therefore, these areas should be rewritten according to the device documents.

3.4.8 Read Command

The Read command reads data from a designated area in the flash memory and sends that data to the flash programmer. The target address can be designated by 1-byte units. Reads are executed in order from start address to end address by 1 byte. This command can be performed only in **Command acceptance phase**.

3.4.8.1 Command Processing Procedure



After the boot firmware returns an error code, it changes to the **Wait for command** state.

Figure 16. Read Command Processing

3.4.8.2 Command Packet

SOH	LNH	LNL	COM	SAD	EAD	SUM	ETX

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x09
COM	(1 byte)	0x15 (Read command)
SAD	(4 bytes)	Start address Example: 0000_4000h -> 0x00, 0x00, 0x40, 0x00
EAD	(4 bytes)	End address Example: 003F_FFFFh -> 0x00, 0x3F, 0xFF, 0xFF
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.8.3 Data Packet [Read Data]

SOD	LNH	LNL	RES	DAT	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	N + 1 (High)
LNL	(1 byte)	N + 1 (Low)
RES	(1 byte)	0x15 (OK)
DAT	(N bytes)	Read data
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

N = 1-1024

3.4.8.4 Data Packet [Status OK]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x15 (OK)
STS	(1 byte)	Status code 0x00 (OK)
SUM	(1 byte)	0xE9
ETX	(1 byte)	0x03

3.4.8.5 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x95 (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error) 0xD0 (Address error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.8.6 Status Code from Microcontroller [Priority High: 1 -> low: 10]

Table 12. Read Status Codes

Condition	Status	Priority	Code
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If the RES of received data packet is not OK		6	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is **Authentication phase**	Flow error	4	0xC3
If the start address or the end address does not belong to any areas *1	Address error	5	0xD0
If the start address is a different kind of area from the end address *1		5	
If the start address is bigger than the end address		5	

Note: *1. Scope of each area is subject to area information request command.

3.4.9 ID Authentication Command

The ID authentication command compares the ID code stored in the device with the ID code received from the flash programmer. The result is sent to the flash programmer. This command can be performed only in **Authentication phase**.

Table 13. ID Authentication

Condition		**Result**
Stored ID[127] = 0		Go into an infinite loop
Stored ID[127:126] = 10b		Compare the received ID and the stored ID
Stored ID[127:126] = 11b	Received ID != “ALeRASE”	Compare the received ID and the stored ID
	Received ID = “ALeRASE”	Erase all User area and Config area [Total area erasure] -> Transition to “Command acceptance phase”
Compare	Received ID != stored ID	Go into an infinite loop
	Received ID = stored ID	Transition to “Command acceptance phase”

Note: * “ALeRASE”: 0x41, 0x4C, 0x65, 0x52, 0x41, 0x53, 0x45, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF.

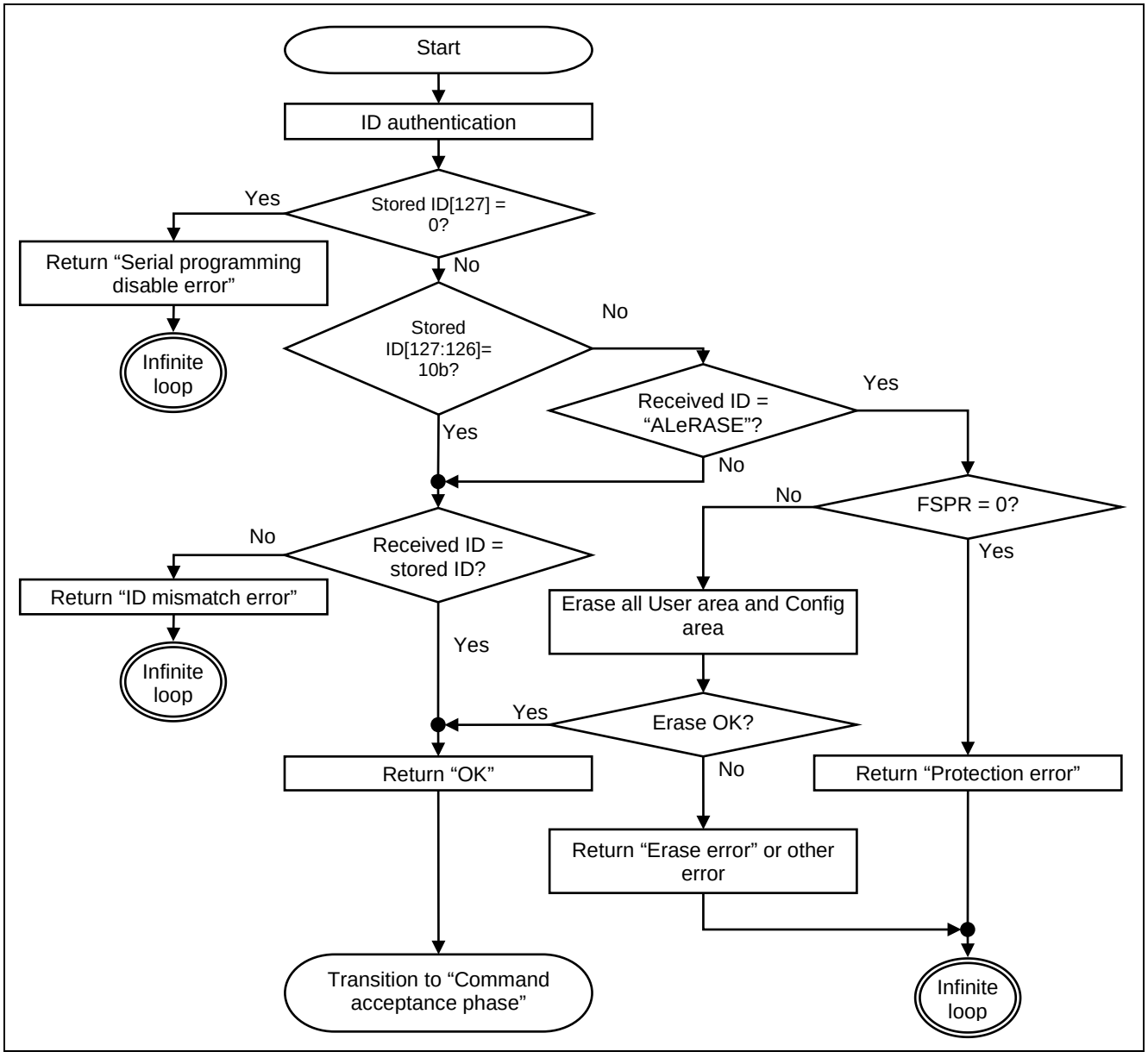
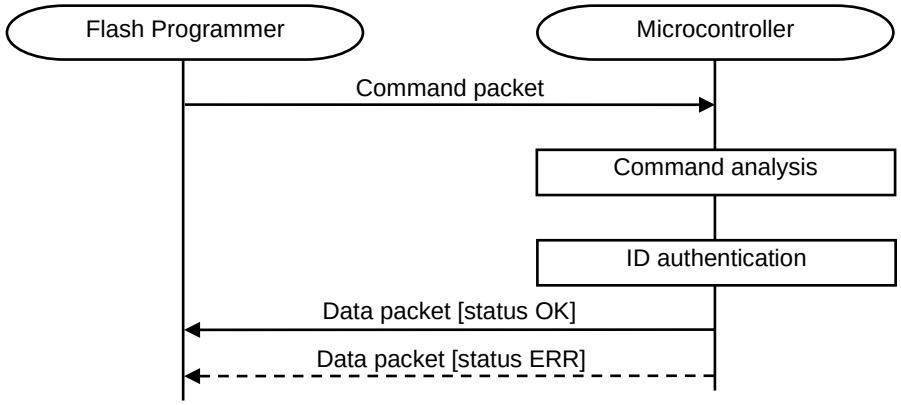


Figure 17. ID Authentication

3.4.9.1 Command Processing Procedure



After the boot firmware returns Packet error or Checksum error or Flow error, it changes to the **Wait ID authentication command** state.

Figure 18. ID Authentication Command Processing

3.4.9.2 Command Packet

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x11
COM	(1 byte)	0x30 (ID authentication command)
IDC	(16 byte)	ID code
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

Example: when stored ID = “0xF0F1F2F3_E4E5E6E7_D8D9DADB_CCCDCECF”,

Flash programmer sends IDC in order of “0xF0”, “0xF1”, “0xF2”, “0xF3”, “0xE4”, “0xE5”,

“0xE6”, “0xE7”, “0xD8”, “0xD9”, “0xDA”, “0xDB”, “0xCC”, “0xCD”, “0xCE”, “0xCF”.

Stored ID:

ID[127:96]				ID[95:64]				ID[63:32]				ID[31:0]			
F0	F1	F2	F3	E4	E5	E6	E7	D8	D9	DA	DB	CC	CD	CE	CF

Order of sending IDC for ID authentication:

1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th
F0	F1	F2	F3	E4	E5	E6	E7	D8	D9	DA	DB	CC	CD	CE	CF

Example: for Total area erasure:

Flash programmer sends IDC in order of “0x41”, “0x4C”, “0x65”, “0x52”, “0x41”, “0x53”, “0x45”, “0xFF”, “0xFF”, “0xFF”, “0xFF”, “0xFF”, “0xFF”, “0xFF”, “0xFF”, “0xFF”.

Order of sending IDC for Total area erasure:

1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th
41	4C	65	52	41	53	45	FF	FF	FF	FF	FF	FF	FF	FF	FF

3.4.9.3 Data Packet [Status OK]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x30 (OK)
STS	(1 byte)	Status code 0x00 (OK)
SUM	(1 byte)	0xCE
ETX	(1 byte)	0x03

3.4.9.4 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0xB0 (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error) 0xDA (Protection error) 0xDB (ID mismatch error) 0xDC (Serial programming disable error) 0xE1 (Erase error) 0xE2 (Write error) 0xE7 (Sequencer error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.9.5 Status Code from Microcontroller [Priority High: 1 -> Low: 10]

Table 14. ID Authentication Status Codes

Condition	Status	Priority	Code
For “ALeRASE”, if erasure of all User area and Config area is complete	OK	8	0x00
If the received ID matches the stored ID	OK	10	0x00
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is “Command acceptance phase”	Flow error	4	0xC3
For “ALeRASE”, if the FSPR bit is 0	Protection error	6	0xDA
If the received ID is different from the stored ID	ID mismatch error	9	0xDB
If the highest-order bit of stored ID is “0”	Serial programming disable error	5	0xDC
For “ALeRASE”, if an erase error occurs	Erase error	7	0xE1
For “ALeRASE”, if a write error occurs	Write error	7	0xE2
For “ALeRASE”, if a sequencer error occurs	Sequencer error	7	0xE7

3.4.10 Baud Rate Setting Command

The Baud rate setting command receives baud rate data and changes the UART baud rate of the device. If an error occurs, the baud rate is not changed. This command can be performed only in **Command acceptance phase**.

3.4.10.1 Command Processing Procedure

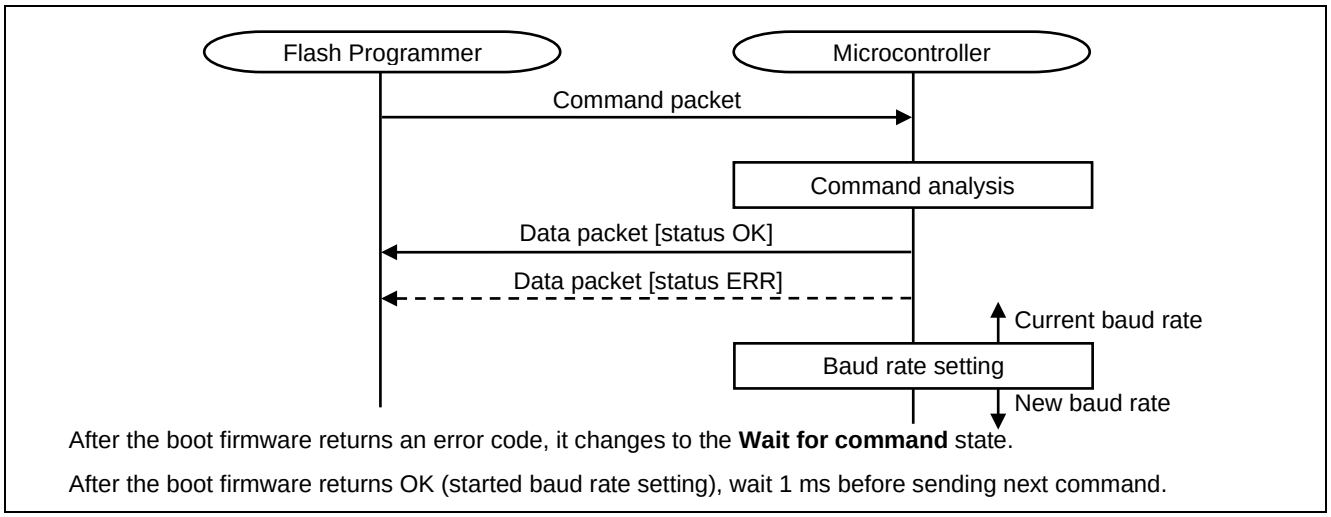


Figure 19. Baud Rate Setting Command Processing

3.4.10.2 Command Packet

SOH	LNH	LNL	COM	BRT	SUM	ETX

SOH	(1 byte)	0x01
LNH	(1 byte)	0x00
LNL	(1 byte)	0x05
COM	(1 byte)	0x34 (Baud rate setting command)
BRT	(4 bytes)	UART baud rate [bps] Example: 2 Mbps (2000000 bps) -> 0x00, 0x1E, 0x84, 0x80
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.10.3 Data Packet [Status OK]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0x34 (OK)
STS	(1 byte)	Status code 0x00 (OK)
SUM	(1 byte)	0xCA
ETX	(1 byte)	0x03

3.4.10.4 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0xB4 (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error) 0xD4 (Baud rate margin error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.10.5 Status Code from Microcontroller [Priority High: 1 -> low: 10]

Table 15. Baud Rate Setting Status Codes

Condition	Status	Priority	Code
Started the baud rate setting	OK	6	0x00
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is **Authentication phase**	Flow error	4	0xC3
If the baud rate error exceeds acceptable range (4%)	Baud rate margin error	5	0xD4
If the BRT value is 0			
If the BRT value exceeds the RMB value			

3.4.10.6 Baud Rate Setting Values

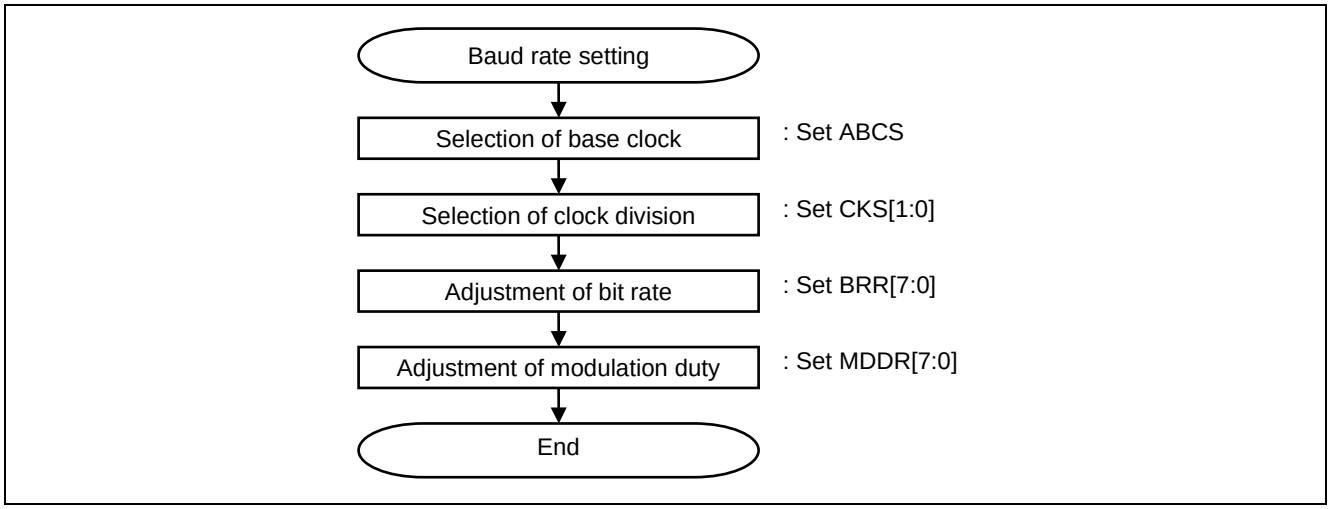


Figure 20. Baud Rate Setting Values

Table 16. Baud Rate Setting

Process	Condition	Setting value
Selection of base clock (ABCS setting)	In case of (SCI / BRT) < 32	ABCS = 1
	Otherwise	ABCS = 0
Selection of clock division (CKS setting)	-	CKS[1:0] = 00b
Adjustment of bit rate (BRR setting)	In case of (SCI / BRT) < 32	BRR[7:0] = 00h
	Otherwise	BRR[7:0] = (SCI / BRT) / 32 - 1
	* If BRR is overflow	BRR[7:0] = FFh
Adjustment of modulation duty (MDDR setting)	In case of (SCI / BRT) < 32	Baud rate = (SCI / (BRR+1)) / 16 MDDR[7:0] = 256 * BRT / baud rate
	Otherwise	Baud rate = (SCI / (BRR+1)) / 32 MDDR[7:0] = 256 * BRT / baud rate
	* If MDDR is overflow	MDDR[7:0] = (non-use) * disable the adjustment by MDDR
	* If MDDR < 128	MDDR[7:0] = 80h

SCI: SCI operating clock frequency [Hz]
BRT: Intended baud rate [bps]

Example: typical settings for baud rate.

Baud Rate Settings [when SCI = 32 MHz]

Intended baud rate	ABCS	CKS[1:0]	BRR[7:0]	MDDR[7:0]	Accuracy
9600	0	00b	67h	FFh	-0.3%
1000000	0	00b	00h	(non-use)	0.0%
1500000	1	00b	00h	C0h	0.0%
2000000	1	00b	00h	(non-use)	0.0%
>2000000	disable	disable	disable	disable	-

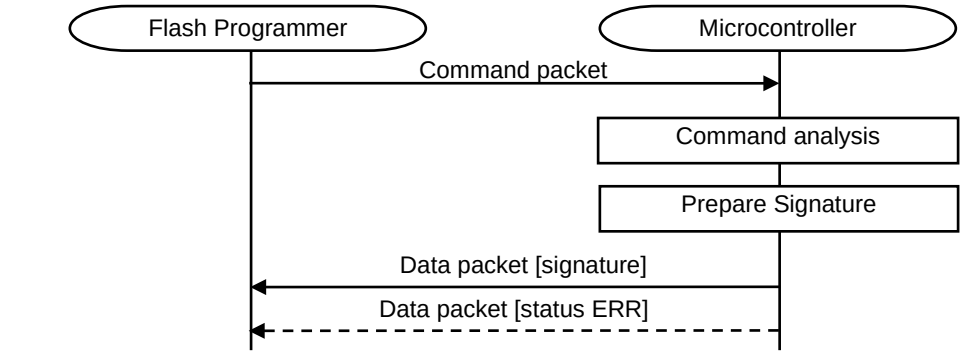
Baud Rate Settings [when SCI = 2 MHz]

Intended baud rate	ABCS	CKS[1:0]	BRR[7:0]	MDDR[7:0]	Accuracy
9600	0	00b	05h	EBh	-0.4%
125000	1	00b	00h	(non-use)	0.0%
>125000	disable	disable	disable	disable	-

3.4.11 Signature Request Command

The Signature request command sends information about the device signature to the flash programmer. This command can be performed in **Command acceptance phase**.

3.4.11.1 Command Processing Procedure



After the boot firmware returns an error code, it changes to the **Wait for command** state.

Figure 21. Signature Request Command Processing

3.4.11.2 Command Packet

SOH	LNH	LNL	COM	SUM	ETX

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

3.4.11.3 Data Packet [Signature]

SOD	LNH	LNL	RES	SCI	RMB	NOA	TYP	BFV	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x0D
RES	(1 byte)	0x3A (OK)
SCI	(4 bytes)	SCI operating clock frequency [Hz] e.g.) 32MHz (32,000,000Hz) -> 0x01, 0xE8, 0x48, 0x00
RMB	(4 bytes)	Recommended maximum UART baud rate of the device [bps] Example: 2 Mbps (2000000 bps) -> 0x00, 0x1E, 0x84, 0x80
NOA	(1 byte)	Number of recordable areas Example: If device has following areas. 0. User area in Code flash 1. User area in Data flash 2. Config area -> 0x03
TYP	(1 byte)	Type code 0x06 (PEAKS MCU + MF4)
BFV	(2 byte)	Boot firmware version Example: Ver10.8 -> 0x0A, 0x08
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.11.4 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0xBA (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.11.5 Status Code from Microcontroller [Priority High: 1 -> low: 10]

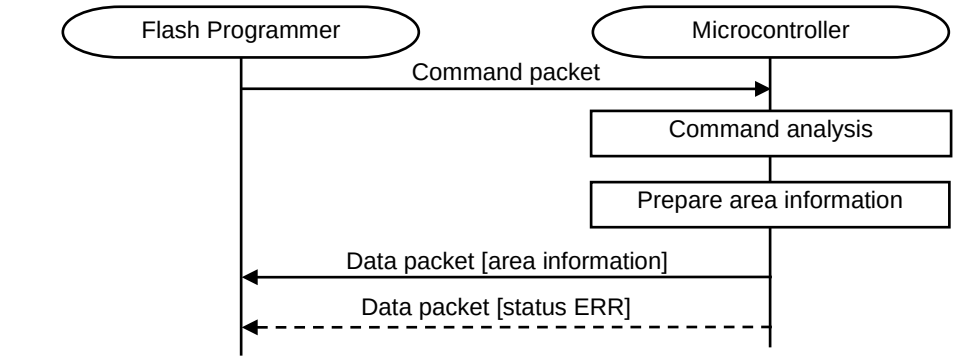
Table 17. Signature Request Status Codes

Condition	Status	Priority	Code
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is **Authentication phase**	Flow error	4	0xC3

3.4.12 Area Information Request Command

The Area information request command sends information about the designated area to the flash programmer. The alignment of the target address of Erase command and Write command will follow this area information. This command can be performed only in **Command acceptance phase**.

3.4.12.1 Command Processing Procedure



After the boot firmware returned an error code, it changes to the **Wait for command** state.

Figure 22. Area Information Request Command Processing

3.4.12.2 Command Packet

SOH	LNH	LNL	COM	NUM	SUM	ETX

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

3.4.12.3 Data Packet [Area information]

SOD	LNH	LNL	RES	KOA	SAD	EAD	EAU	WAU	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x12
RES	(1 byte)	0x3B (OK)
KOA	(1 byte)	Kind of area 0x00 (User area in Code flash) 0x01 (User area in Data flash) 0x02 (Config area)
SAD	(4 bytes)	Start address Example: 0001_0000h -> 0x00, 0x01, 0x00, 0x00
EAD	(4 bytes)	End address Example: 001F_FFFFh -> 0x00, 0x1F, 0xFF, 0xFF
EAU	(4 bytes) *1	Erase access unit (alignment) [bytes] Example: 32 KB (32768 bytes) -> 0x00, 0x00, 0x80, 0x00
WAU	(4 bytes)	Write access unit (alignment) [byte] Example: 256 byte -> 0x00, 0x00, 0x01, 0x00
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

*1: If EAU is 0x00000000, Erase command is not available for the area.

3.4.12.4 Data Packet [Status ERR]

SOD	LNH	LNL	RES	STS	SUM	ETX

SOD	(1 byte)	0x81
LNH	(1 byte)	0x00
LNL	(1 byte)	0x02
RES	(1 byte)	0xBB (ERR)
STS	(1 byte)	Status code 0xC1 (Packet error) 0xC2 (Checksum error) 0xC3 (Flow error) 0xD0 (Address error)
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

3.4.12.5 Status Code from Microcontroller [Priority High: 1 -> low: 10]

Table 18. Area Information Request Status Codes

Condition	Status	Priority	Code
If LNH and LNL in the received packet are different from defined values	Packet error	3	0xC1
If the received packet does not have ETX		1	
If SUM in the received packet is different from the value calculated by the boot firmware	Checksum error	2	0xC2
If the state is **Authentication phase**	Flow error	4	0xC3
If NUM in the received packet is a nonexistent area number	Address error	5	0xD0

3.5 Recommended Procedure for Flash Programmer

3.5.1 Beginning Communication

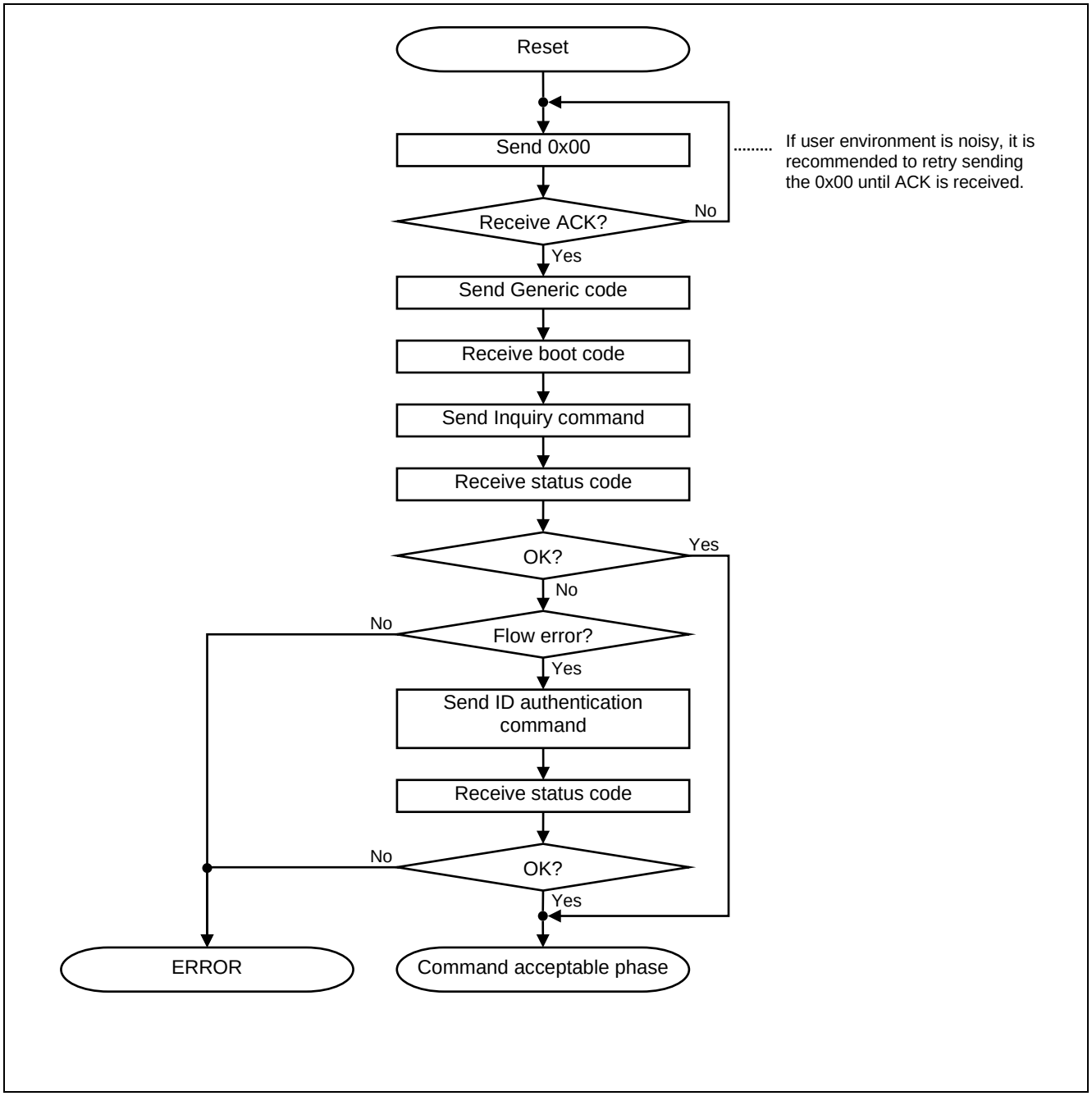


Figure 23. Beginning Communication

3.5.2 Total Area Erasure

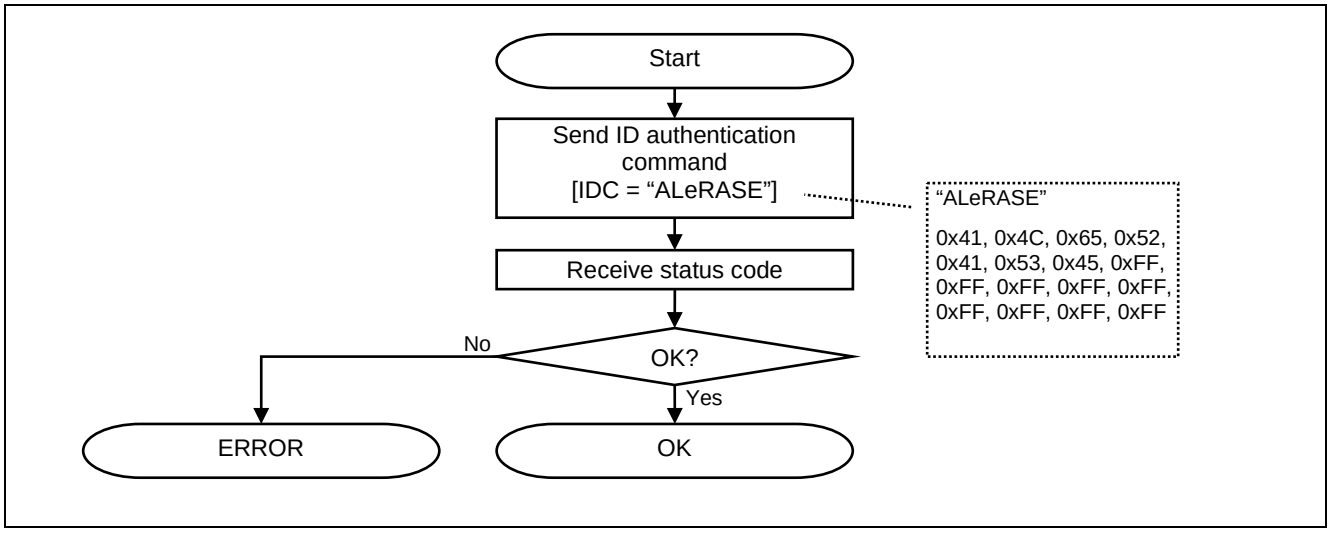


Figure 24. Total Area Erasure

3.5.3 Acquisition of Device Information

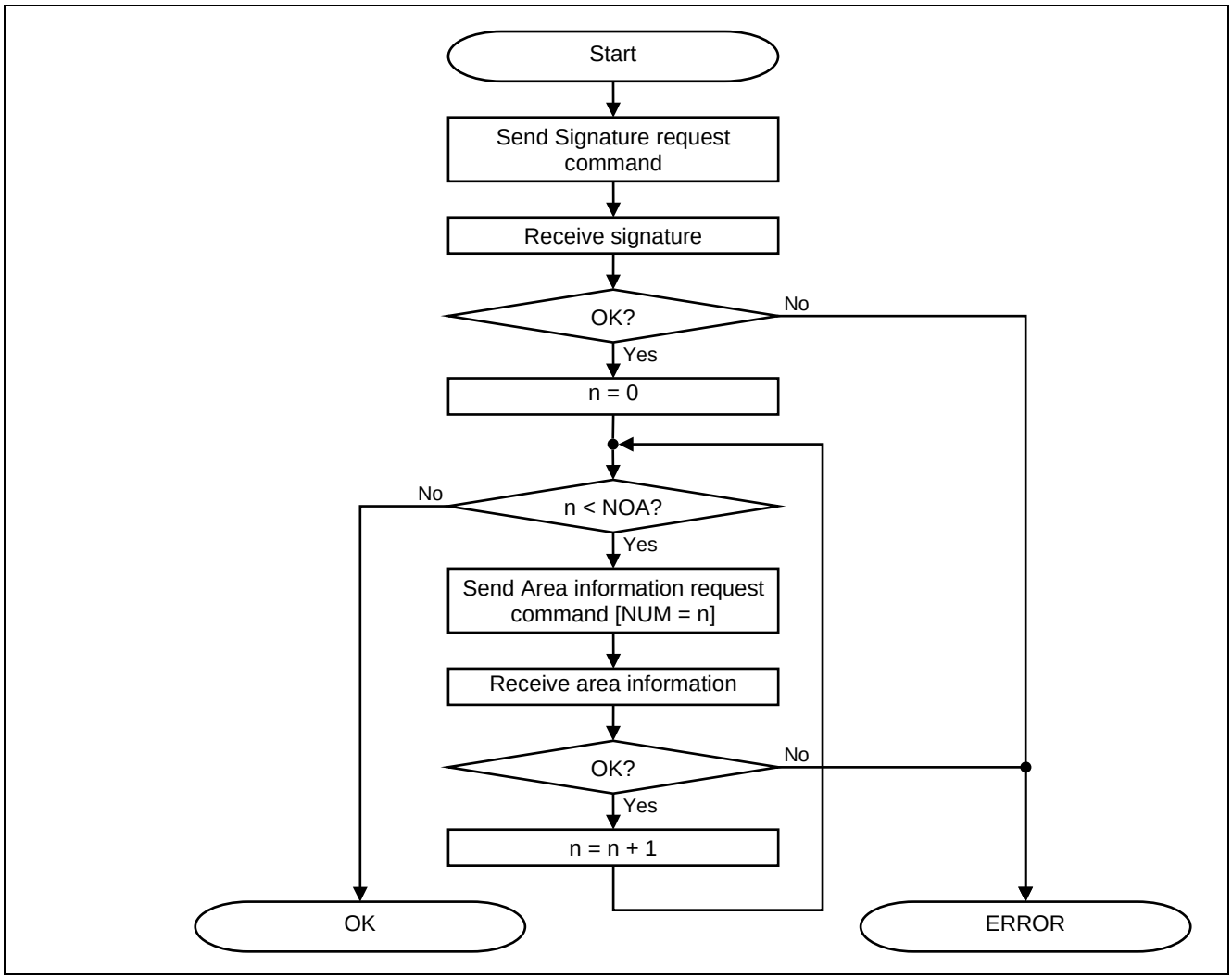


Figure 25. Acquisition of Device Information

3.5.4 Code and Data in User Area Updates

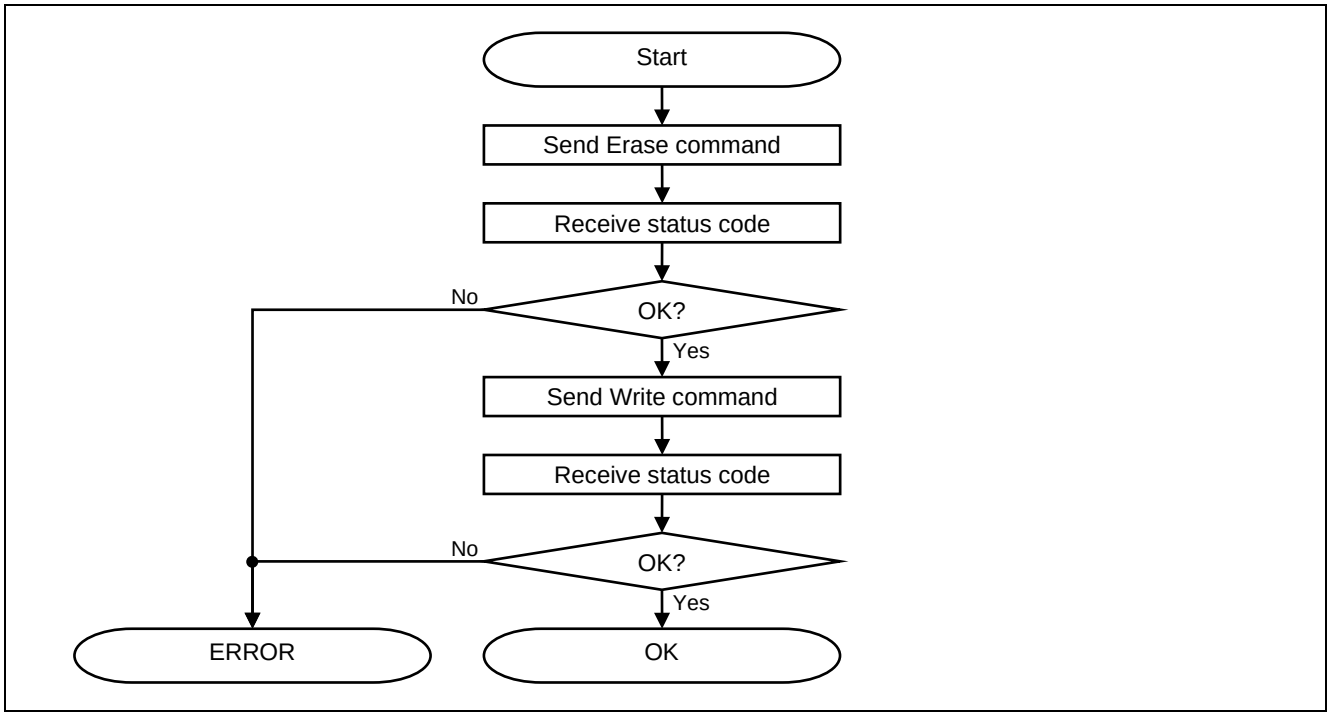


Figure 26. Code and Data in User Area Updates

3.5.5 Configuration Data Updates

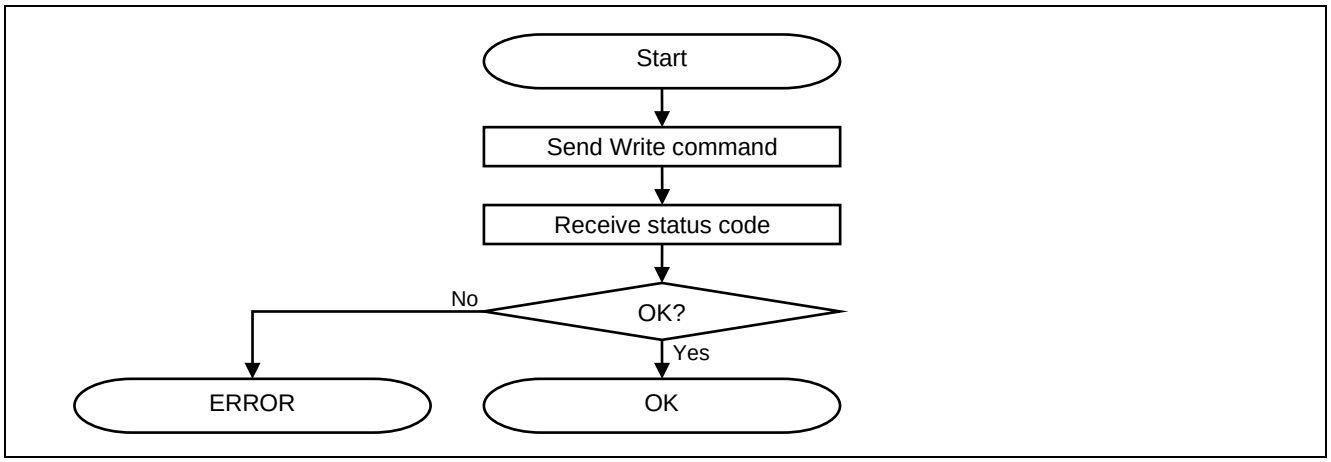


Figure 27. Configuration Data Updates

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.31.22	—	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/.