Renesas Synergy™ Platform

System Specifications for Standard Boot Firmware

R01AN3280EU0110
Rev.1.10
Oct 13, 2017

Introduction

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

Target Device

This document is for the S7G2, S3A7, and S124 microcontroller groups. Information for other products is tentative and for reference use only. This document will be revised to include other products as needed.

Contents

1. Definition of Terms

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

1.1 Flash Memory

The ROM area where program code is written is called the code flash. The ROM area where data is written is called the data flash. Both code flash and data flash are called flash memory. The area used by user is called the User area. The area that stores configuration data is called the Configuration area.

Example: Figure 1.1 shows an example of the Synergy S7 Series MCU flash memory structure.

Note: Memory structure differs from device to device.

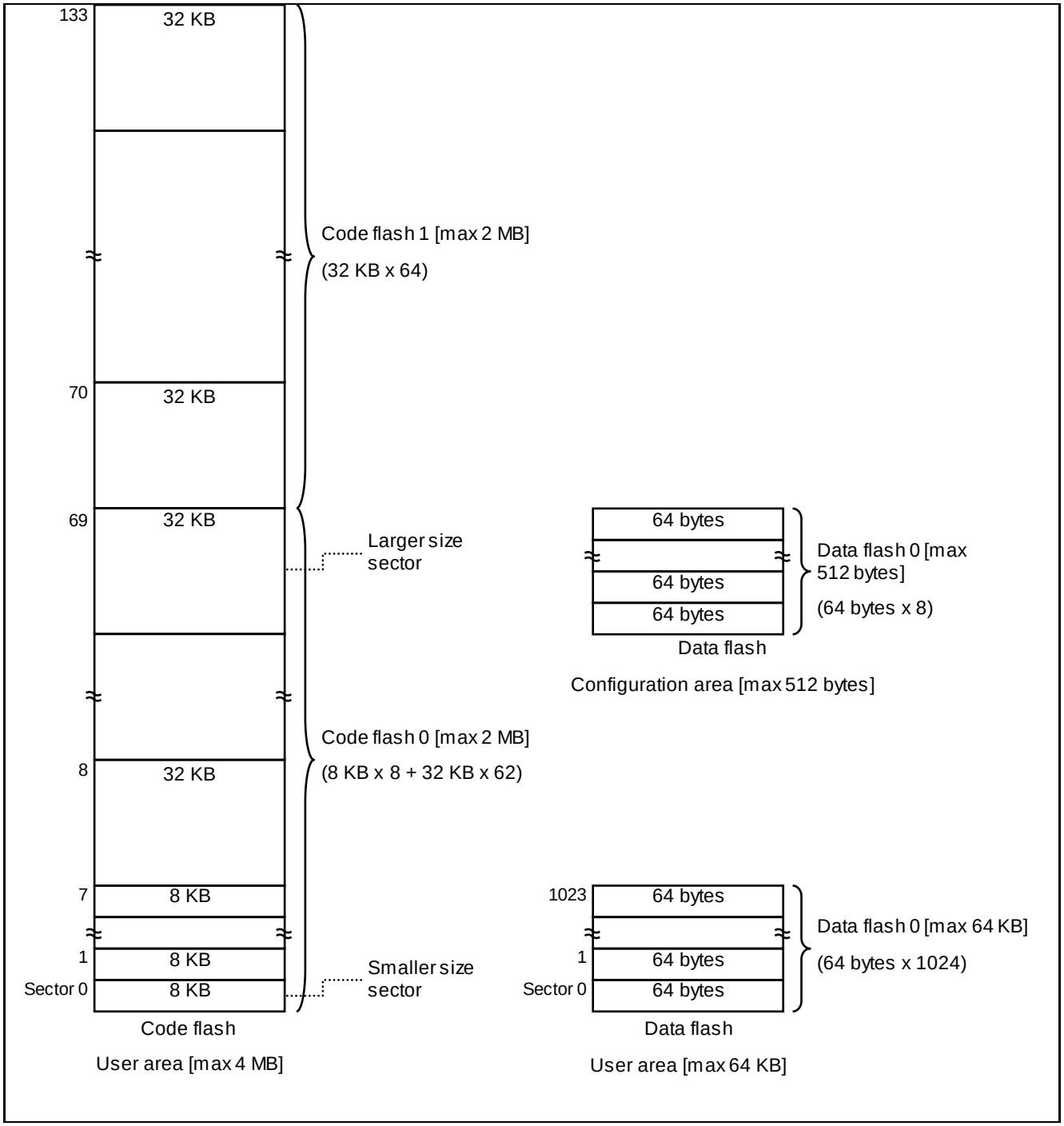


Figure 1.1 Example flash memory structure of the S7G2 MCU Group

1.2 Boot Firmware

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

1.3 Serial Communication Interface (SCI)

The SCI is an interface module that performs serial communication. The boot firmware uses the common SCI configured as the Universal Asynchronous Receiver/Transmitter (UART).

1.4 Access Window (AW)

The AW is a function that assigns accessible sectors to erase or write in the code flash. This function allows access from the start block address to the end block address, and prohibits access to other areas. Users set a value associated with the start block address to the FAWS, and set a value associated with the end block address + 1 to the FAWE. If FSPR is 0, FAWS and FAWE cannot be changed. For details, see the Renesas Synergy Flash Memory User's Manual.

Table 1.1 Example of the Synergy S5/S7 Series MCU flash memory structure

Block number	Base address	Size	Setting value
0	00000000h	8 KB	000h
1	00002000h	8 KB	001h
2	00004000h	8 KB	002h
3	00006000h	8 KB	003h
4	00008000h	8 KB	004h
5	0000A000h	8 KB	005h
6	0000C000h	8 KB	006h
7	0000E000h	8 KB	007h
8	00010000h	32 KB	008h
9	00018000h	32 KB	00Ch
10	00020000h	32 KB	010h
11	00028000h	32 KB	014h
:	:	:	:
132	003F0000h	32 KB	1F8h
133	003F8000h	32KB	1FCh

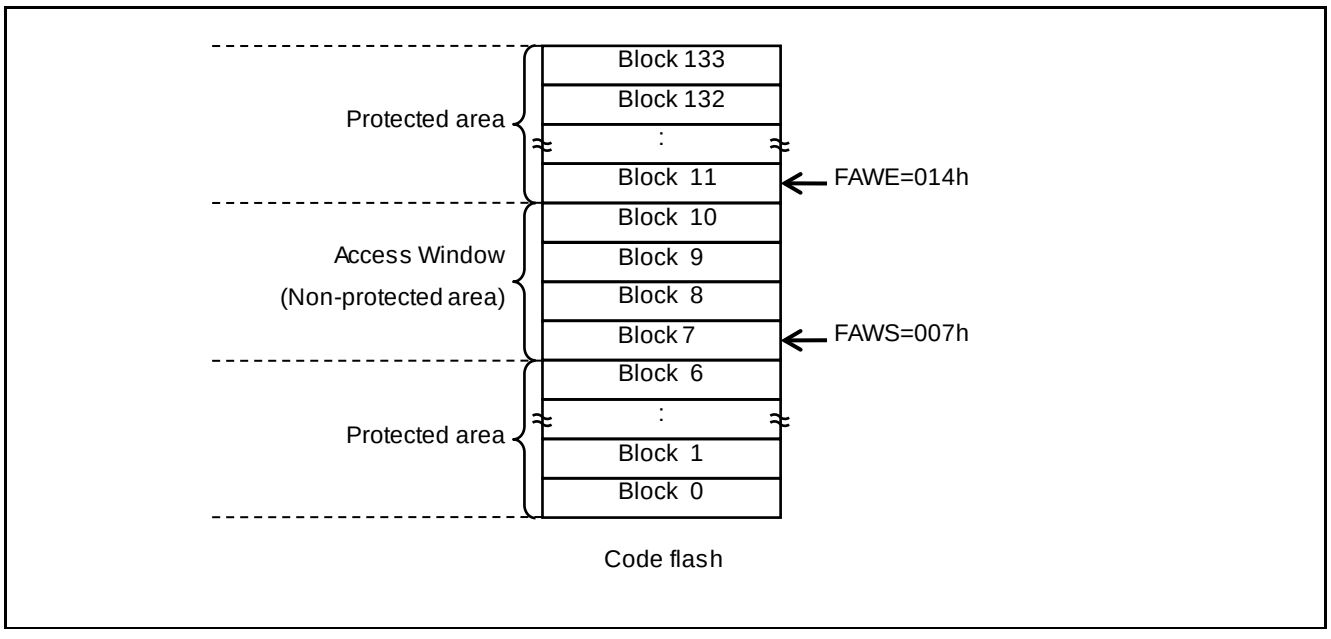


Figure 1.2 Example of access window with FAWS = 007h, FAWE = 014h

Table 1.2 Example of Synergy S1/S3 Series MCU flash memory structure [Connection1]

Block number	Base address	Size	Setting value
0	00000000h	2 KB	000h
1	00000800h	2 KB	002h
2	00001000h	2 KB	004h
3	00001800h	2 KB	006h
4	00002000h	2 KB	008h
5	00002800h	2 KB	00Ah
6	00003000h	2 KB	00Ch
7	00003800h	2 KB	00Eh
8	00004000h	2 KB	010h
9	00004800h	2 KB	012h
10	00005000h	2 KB	014h
11	00005800h	2 KB	016h
:	:	:	:
510	000FF000h	2 KB	3FCh
511	000FF800h	2 KB	3FEh

Table 1.3 Example of Synergy S1/S3 Series MCU flash memory structure [Connection3]

Block number	Base address	Size	Setting value
0	00000000h	1 KB	000h
1	00000400h	1 KB	001h
2	00000800h	1 KB	002h
3	00000C00h	1 KB	003h
4	00001000h	1 KB	004h
5	00001400h	1 KB	005h
6	00001800h	1 KB	006h
7	00001C00h	1 KB	007h
8	00002000h	1 KB	008h
9	00002400h	1 KB	009h
10	00002800h	1 KB	00Ah
11	00002C00h	1 KB	00Bh
:	:	:	:
1022	000FF800h	1 KB	3FEh
1023	000FFC00h	1 KB	3FFh

2. System Architecture

The 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, the boot firmware erases all of the User area and Configuration area. The boot firmware is embedded in the device.

2.1 Renesas Synergy™ S7 Series MCU

Table 2.1 Serial programming of the S7 Series MCU

Operating mode	High-speed mode
Supported communication	2-wire UART, USB
Main oscillator input	When using USB: required When not using USB: not required
Operating voltage condition	VCC: 2.7 V - 3.6 V. Same as device specification.

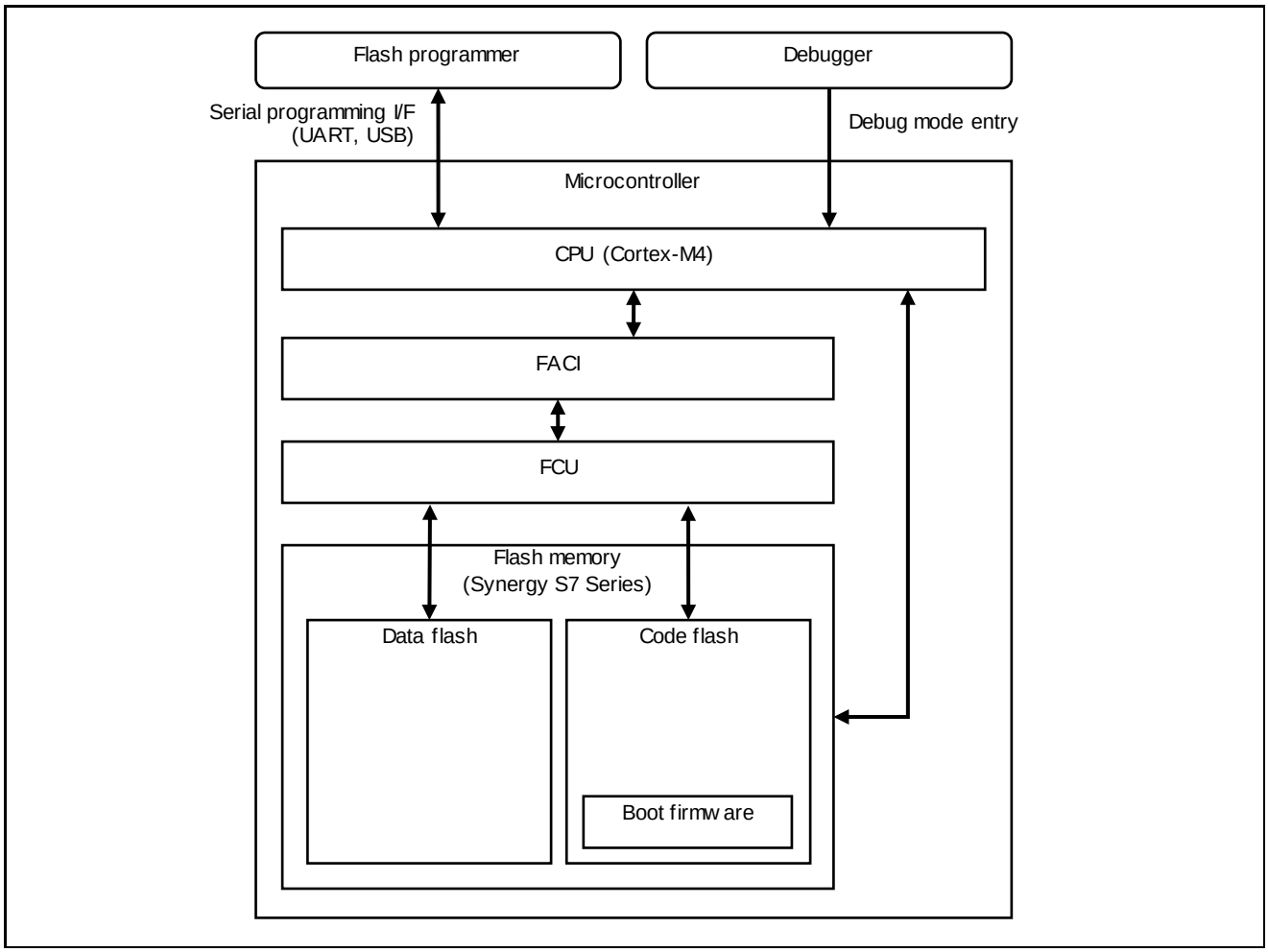


Figure 2.1 Flash system architecture of the S7 Series MCU

2.2 Renesas Synergy™ S3 Series MCU

Table 2.2 Serial programming of the S3 Series MCU

Operating mode	When VCC is higher than 2.4 V: High-speed mode When VCC is lower than 2.4 V: Middle-speed mode
Supported communication	2-wire UART, USB*1
Main oscillator input	When using USB: required*2 When not using USB: not required
Operating voltage condition	VCC = 1.8 V - 5.5 V*3

Note 1: For devices that have a UCKSEL register, that is, devices where HOCO is selectable as the USB clock source, the boot firmware generates the USB clock from HOCO. Therefore, USB communication is not guaranteed in an environment where the accuracy of HOCO oscillation does not meet the standard of the USB clock. For the accuracy of HOCO oscillation, refer to the electrical characteristics of the device.

Note 2: For devices that have a UCKSEL register, that is, devices where HOCO is selectable as the USB clock source), the boot firmware can operate by HOCO even when using USB.

Note 3: USB communication is not available where VCC < 2.4 V.

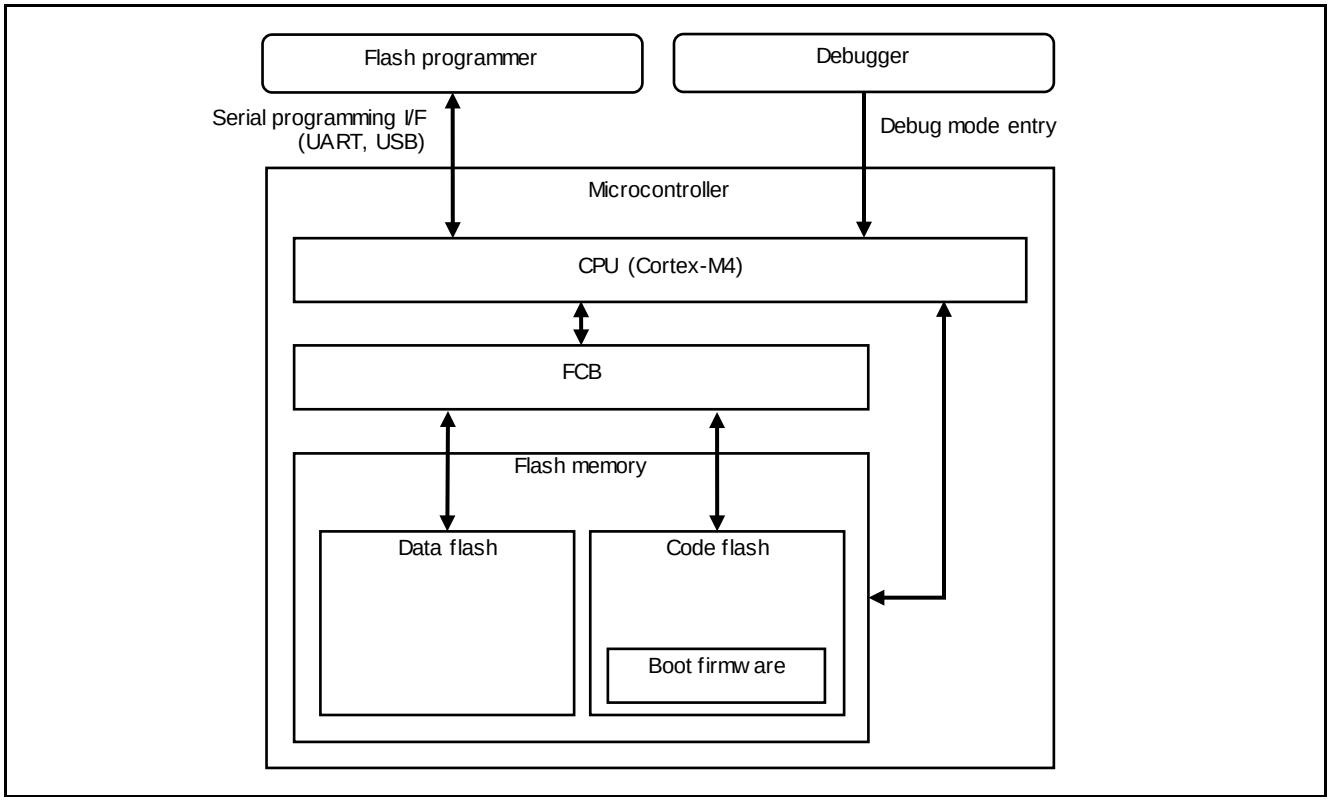


Figure 2.2 Flash system architecture of the S3 Series MCU

2.3 Renesas Synergy™ S1 Series MCU

Table 2.3 Serial programming of the S1 Series MCU

Operating mode	When VCC is higher than 2.4 V: High-speed mode When VCC is lower than 2.4 V: Middle-speed mode
Supported communication	2-wire UART
Main oscillator input	Not required
Operating voltage condition	VCC = 1.8 V - 5.5 V

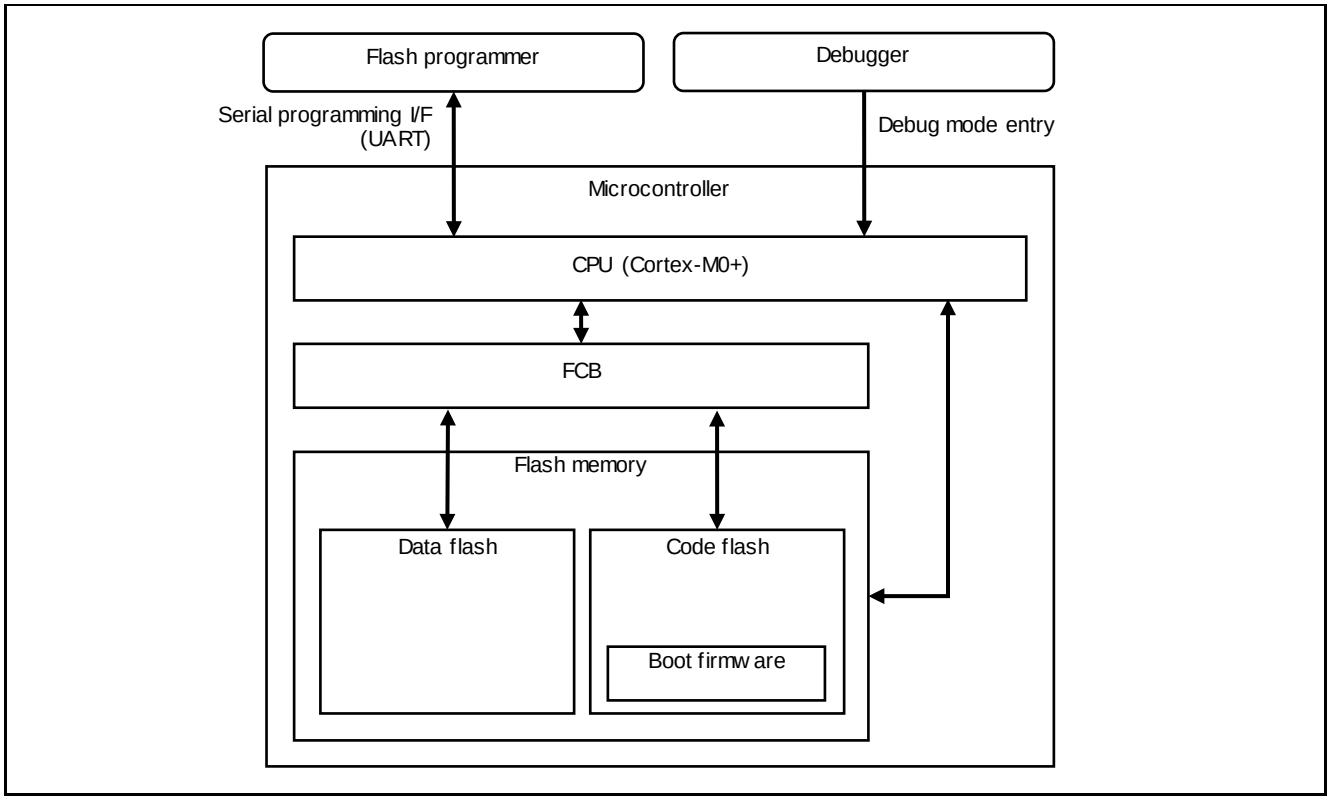


Figure 2.3 Flash system architecture of the S1 Series MCU

2.4 Renesas Synergy™ S5 Series MCU

Table 2.4 Serial programming of the S5 Series MCU

Operating mode	High-speed mode
Supported communication	2-wire UART, USB
Main oscillator input	When using USB: required When not using USB: not required
Operating voltage condition	VCC = 2.7 V - 3.6 V. Same as device specification.

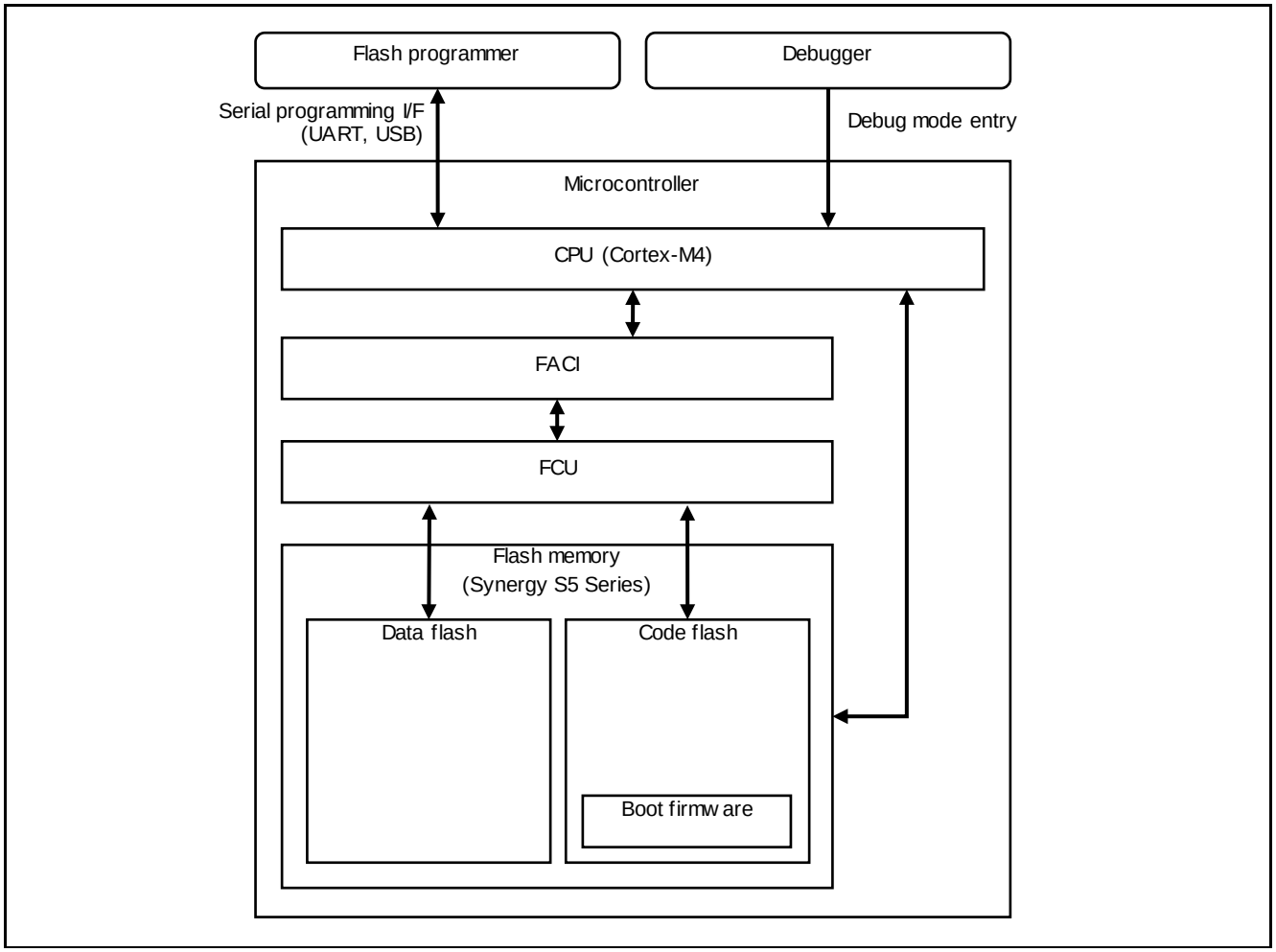


Figure 2.4 Flash system architecture of the S5 Series MCU

2.5 Mode Entry

The boot firmware selects serial programming mode or debug mode based on the MD pin level at reset timing. If the operating mode is serial programming mode, the boot firmware transits to the communication setting phase. If the operating mode is not serial programming mode, the boot firmware erases all of the User area and Configuration area, and then goes into an infinite loop. If security MPU is supported, the boot firmware goes into an infinite loop without transition to either mode.

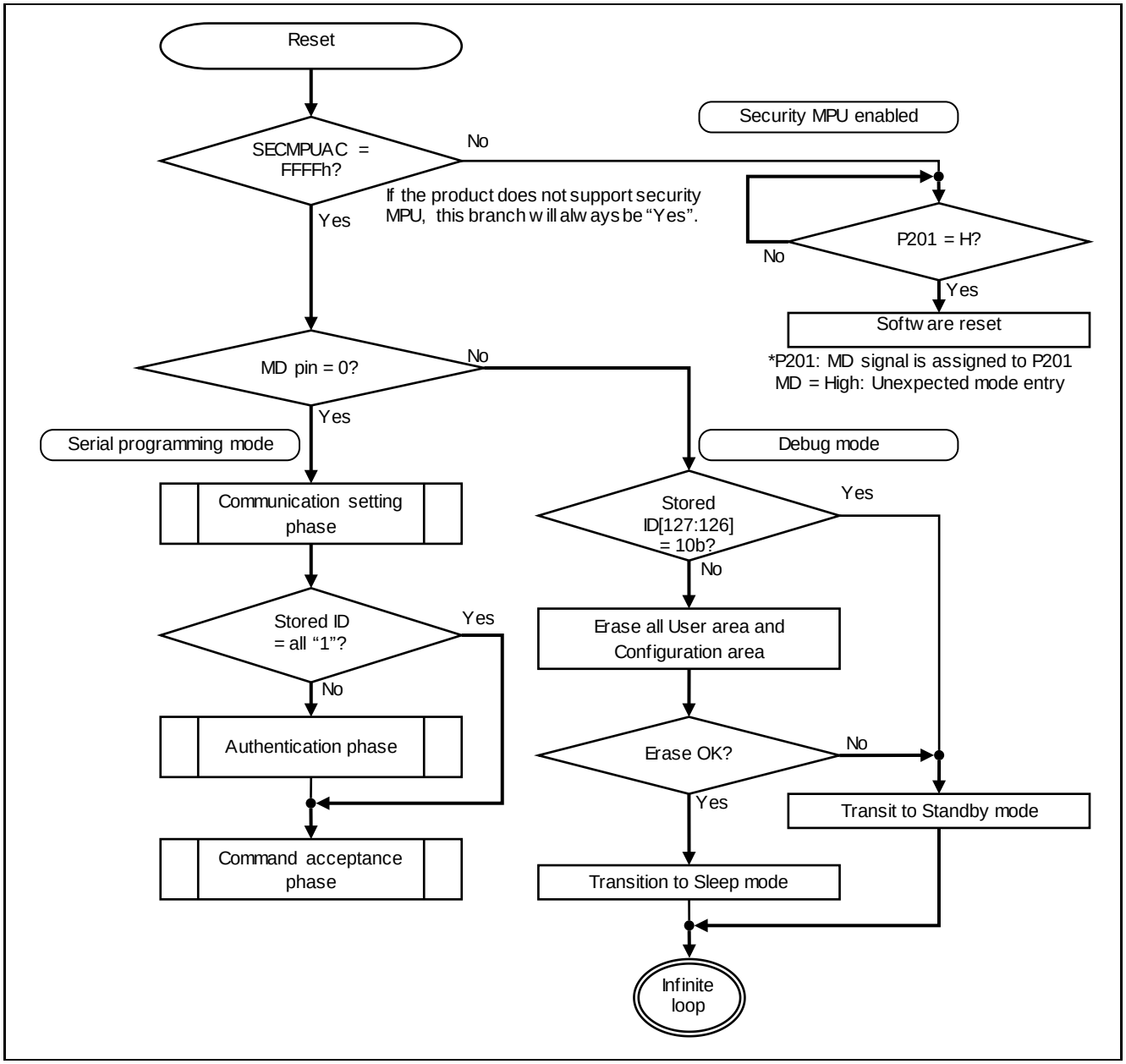


Figure 2.5 Flash modes

3. Serial Programming Interface

3.1 Communication Mode

The boot firmware includes interfaces for the following communication modes. Section 3.2.1 includes a flowchart for selecting the mode.

- 2-wire UART communication
- USB communication.

3.1.1 2-wire UART Communication

The boot firmware supports 2-wire UART communication, see Figure 3.1.

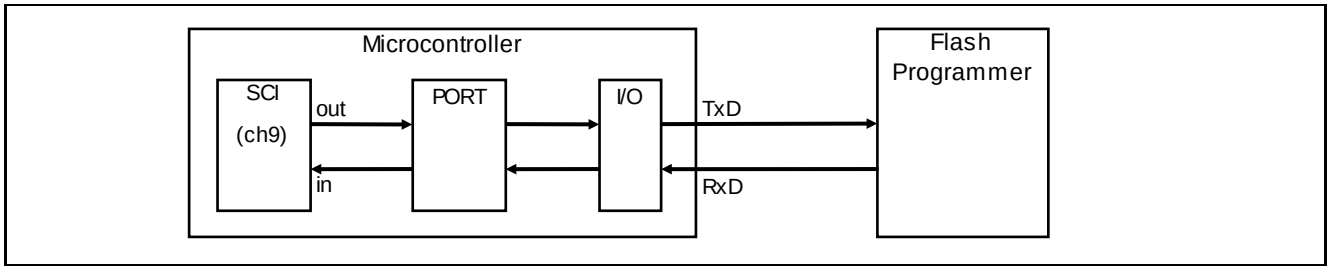


Figure 3.1 2-wire UART communication

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

Table 3.1 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 complete. The communication speed is changed to the intended rate after successful completion of the baud rate setting command, see Figure 3.2.

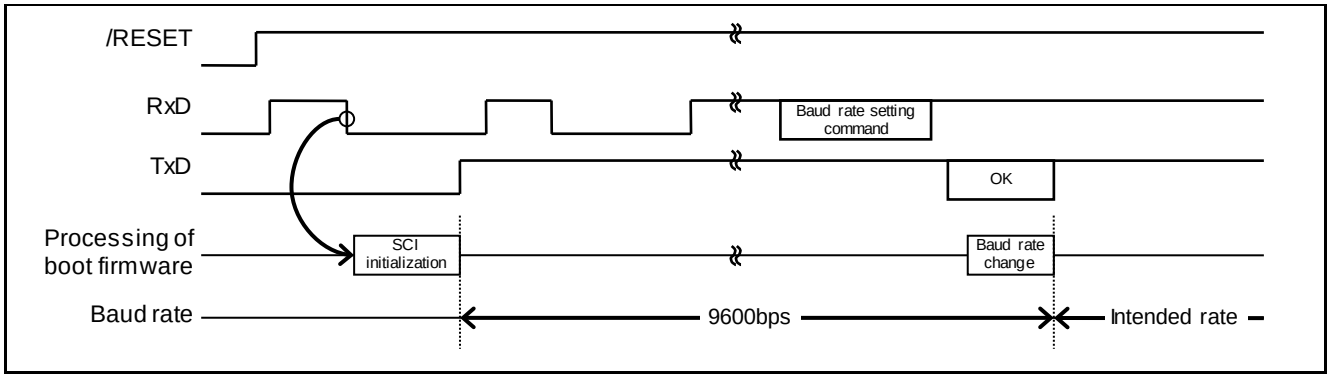


Figure 3.2 Baud rate setting

When using UART communication, do not connect to flash programmer with the USB cable. If the UART communication cable is unplugged during transmission, the result is undefined.

3.1.2 USB Communication

Boot firmware supports USB communication, see Figure 3.3.

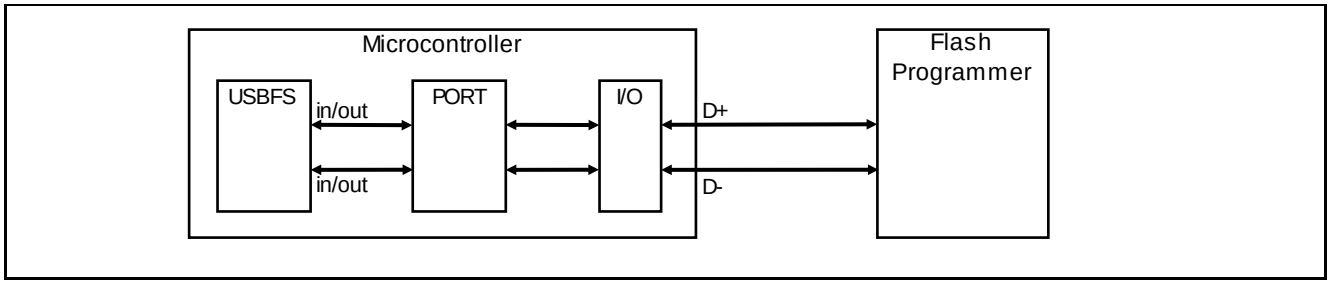


Figure 3.3 USB communication

In serial programming mode, boot firmware sets the USB state to enumeration. Boot firmware selects USB communication and initiates communication by detecting the configured state.

Table 3.2 USB settings

Interface	USBFS
VBUS	Input mode
D+	Input-Output mode
D-	Input-Output mode
Transfer rate	12 Mbps (USB 2.0 Full Speed)
Device class	Communication Device Class (CDC) Sub class: Abstract Control Model (ACM) Protocol: Common AT commands
Vendor ID	045Bh (Renesas)
Product ID	0238h (Synergy USB Boot)
Transfer mode	Control (in/out) Bulk (in, out) Interrupt (in)
End points	EP0: Default control pipe, control transfers (in/out) EP1: TxD pipe, bulk transfers (in) 64 bytes EP2: RxD pipe, bulk transfers (out) 64 bytes EP6: Control pipe, interrupt transfers (in)

Clock input from the main clock oscillator with the frequencies listed in Table 3.3 is required to generate the USB communication clock.

Table 3.3 Clock frequencies for USB communication

MCU Series	Frequency
S7/S5	8, 10, 12, 15, 16, 20 or 24 MHz
S3 w/o UCKSEL register	4, 6, 8 or 12 MHz
S3 w/ UCKSEL register	N/A (USB clock is generated from the HOCO and not the main oscillator)

When using USB communication, release the reset after connecting to the flash programmer with the USB cable. If the USB cable is unplugged during transmission, the result is undefined.

When using USB communication, boot firmware notifies the host that the power mode is self-powered.

3.2 Operating Procedures

3.2.1 Communication Setting Phase

After a reset release, boot firmware selects the communication mode (2-wire UART or USB) as shown in the flowchart in Figure 3.4. After specifying the communication setting and receiving generic code from the selected communication method, boot firmware transitions to the authentication phase. However, if the ID code stored in the device is all “1”, the boot firmware transitions directly to the command acceptance phase.

(1) **Selection of communication mode**

Table 3.4 Communication mode selection

Condition	Communication mode
Detection of high level on MD (P201): Single chip mode	Unexpected mode entry (-> Software reset)
Detection of configured state (Enumeration is complete)	USB mode
Detection of falling edge on RxD	UART mode

* P201: MD signal is assigned to P201.

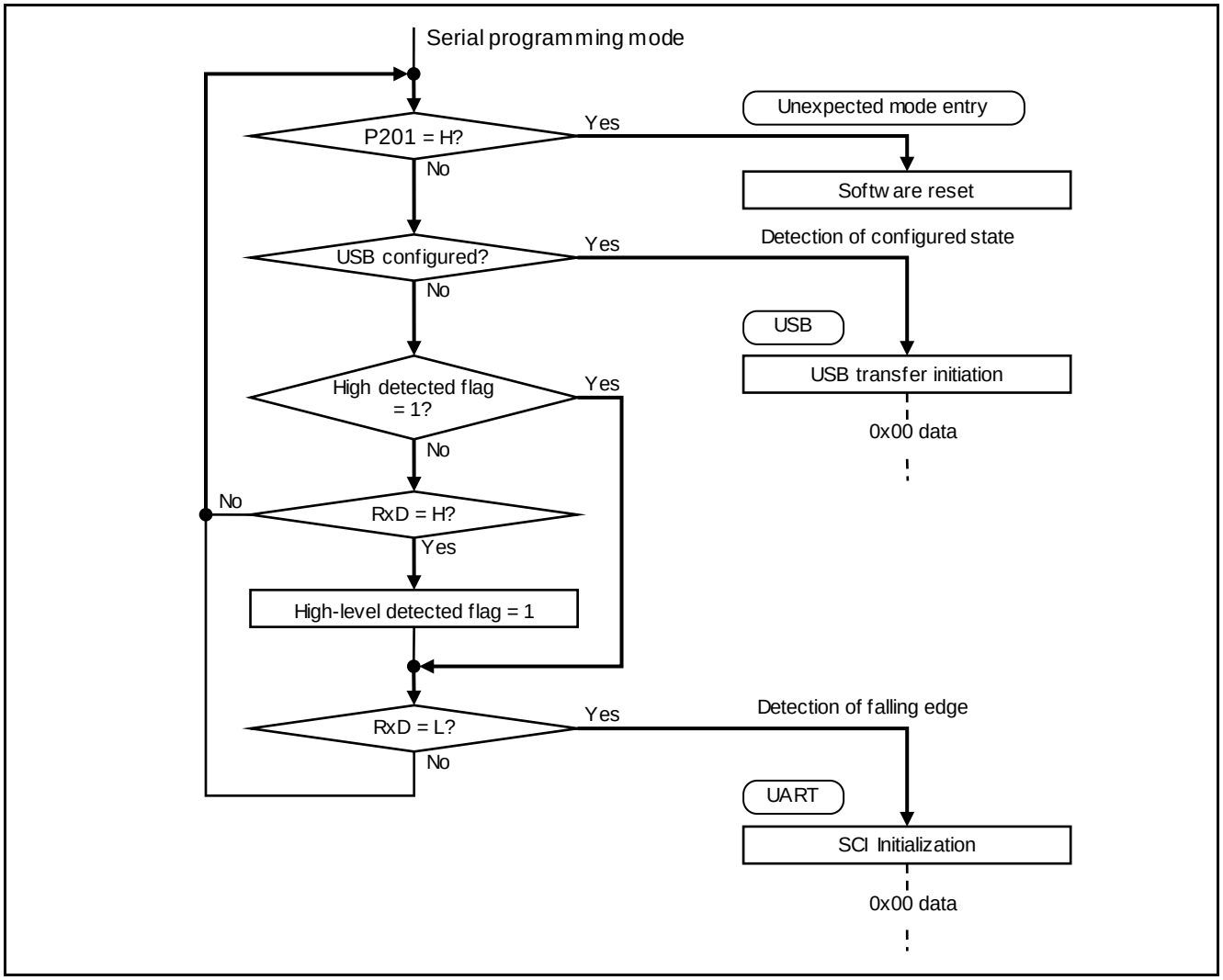


Figure 3.4 Selecting communication mode

(2) **Setting up 2-wire UART communication**

To use UART communication, the flash programmer sends 0x00 data (low pulse) at 9600 bps at least two times. If the environment is noisy, retry sending the low pulse until ACK is received. Boot firmware selects UART communication and initializes the SCI when it detects the falling edge on RxD. Thereafter, boot firmware receives the second low pulse as 0x00 data in SCI, returns the ACK, receives the generic code, then returns the boot code. This completes communication setting.

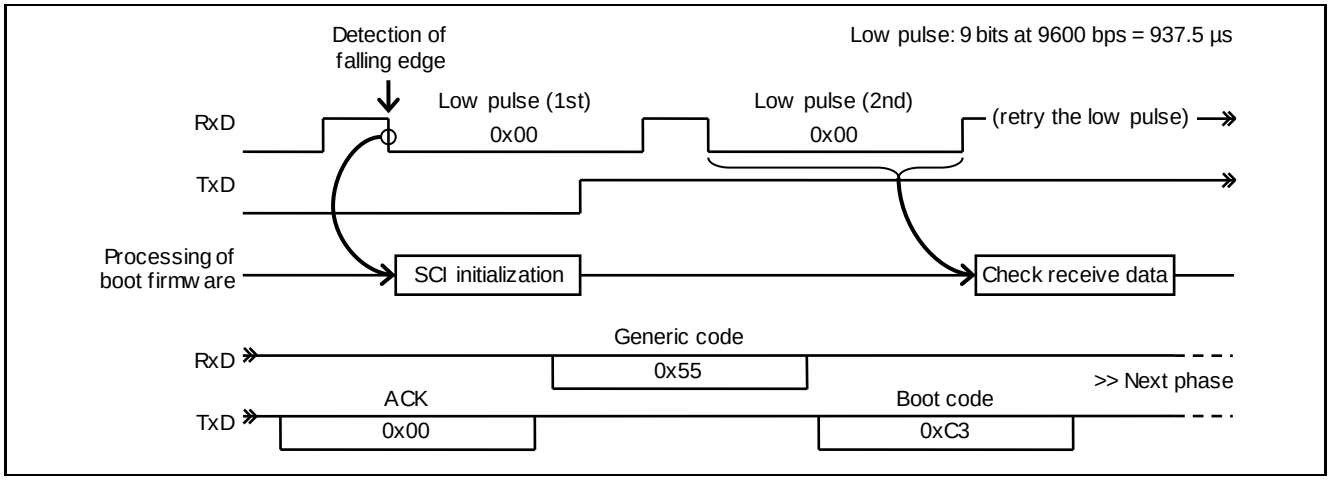


Figure 3.5 2-wire UART set up

(3) **Setting up USB communication**

Boot firmware selects USB communication when it detects the configured state. Thereafter, boot firmware receives 0x00 data from the RxD pipe, returns the ACK, receives the generic code, then returns the boot code. This completes communication settings.

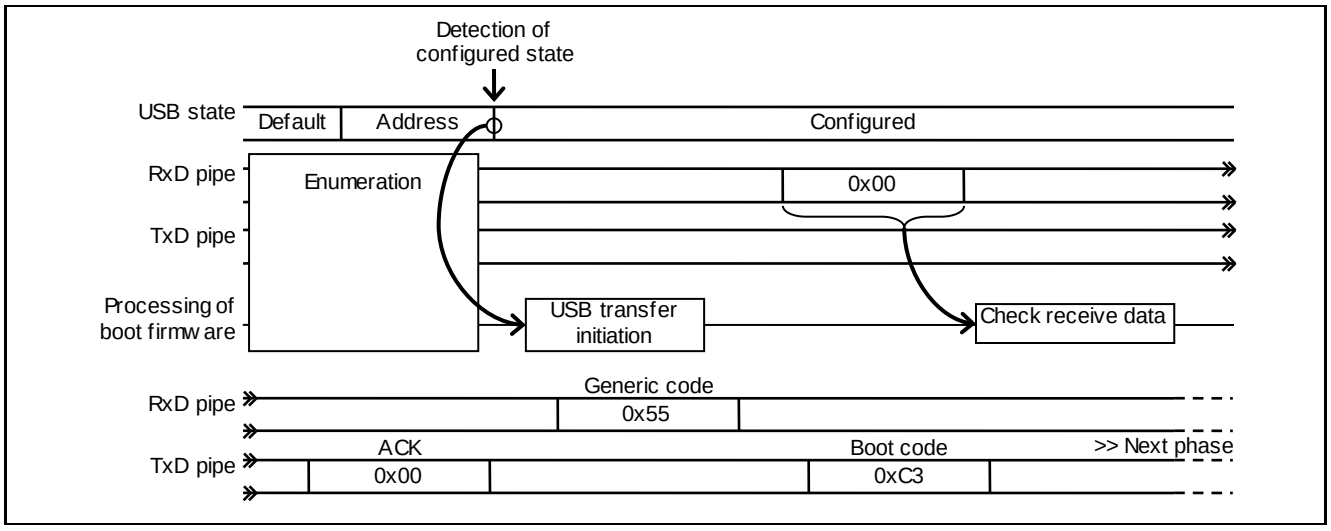


Figure 3.6 USB communication set up

3.2.2 Authentication Phase

Boot firmware authenticates the ID code in this phase. This phase can only accept the Authentication command. If the Authentication command passes successfully, boot firmware transitions 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 determine if the current phase is the command acceptance phase or authentication phase by the result of an Inquiry command.

3.2.4 State Transitions

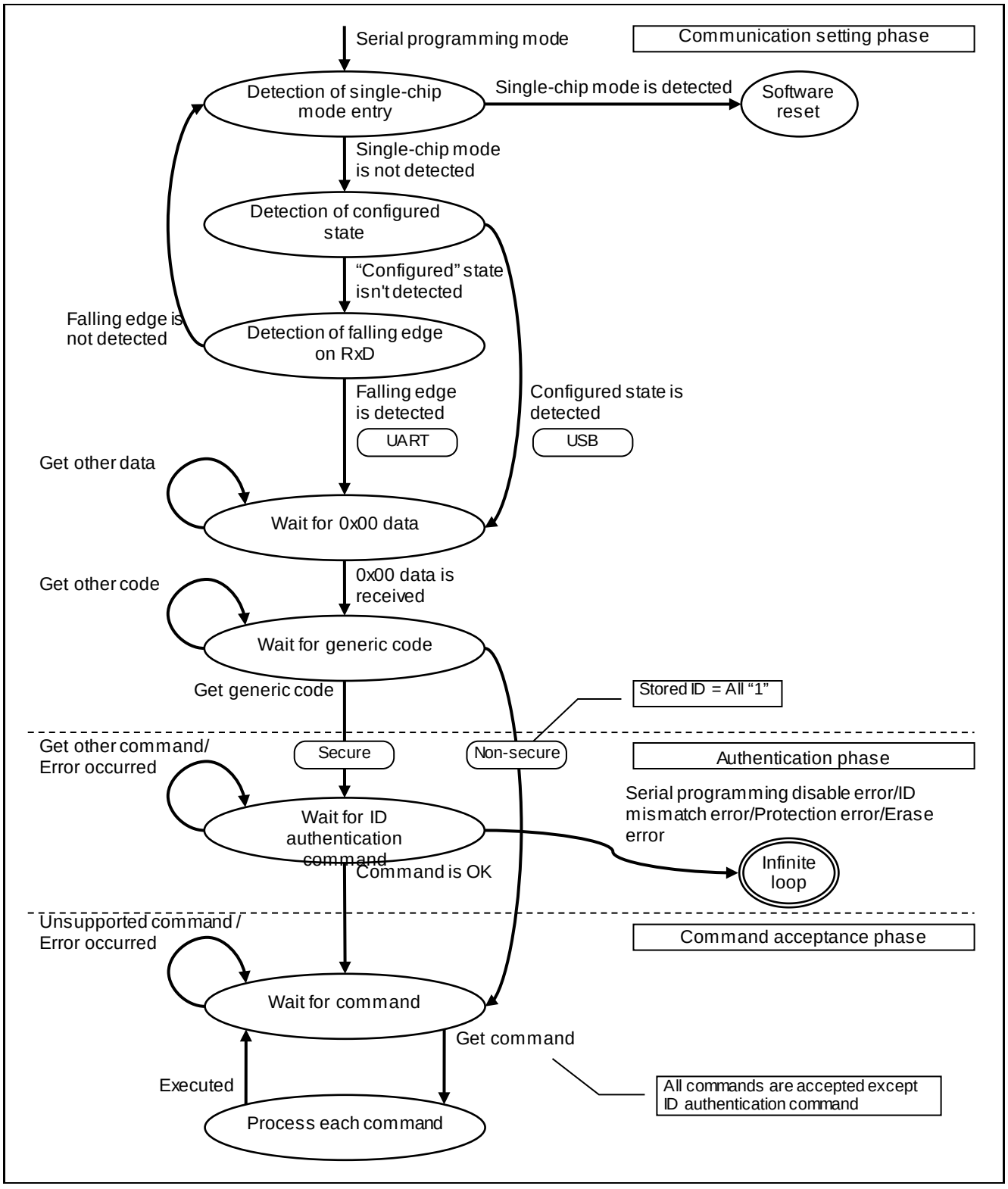


Figure 3.7 State transitions

3.2.5 Starting Communication

(1) **If ID code is already stored [Secure]**

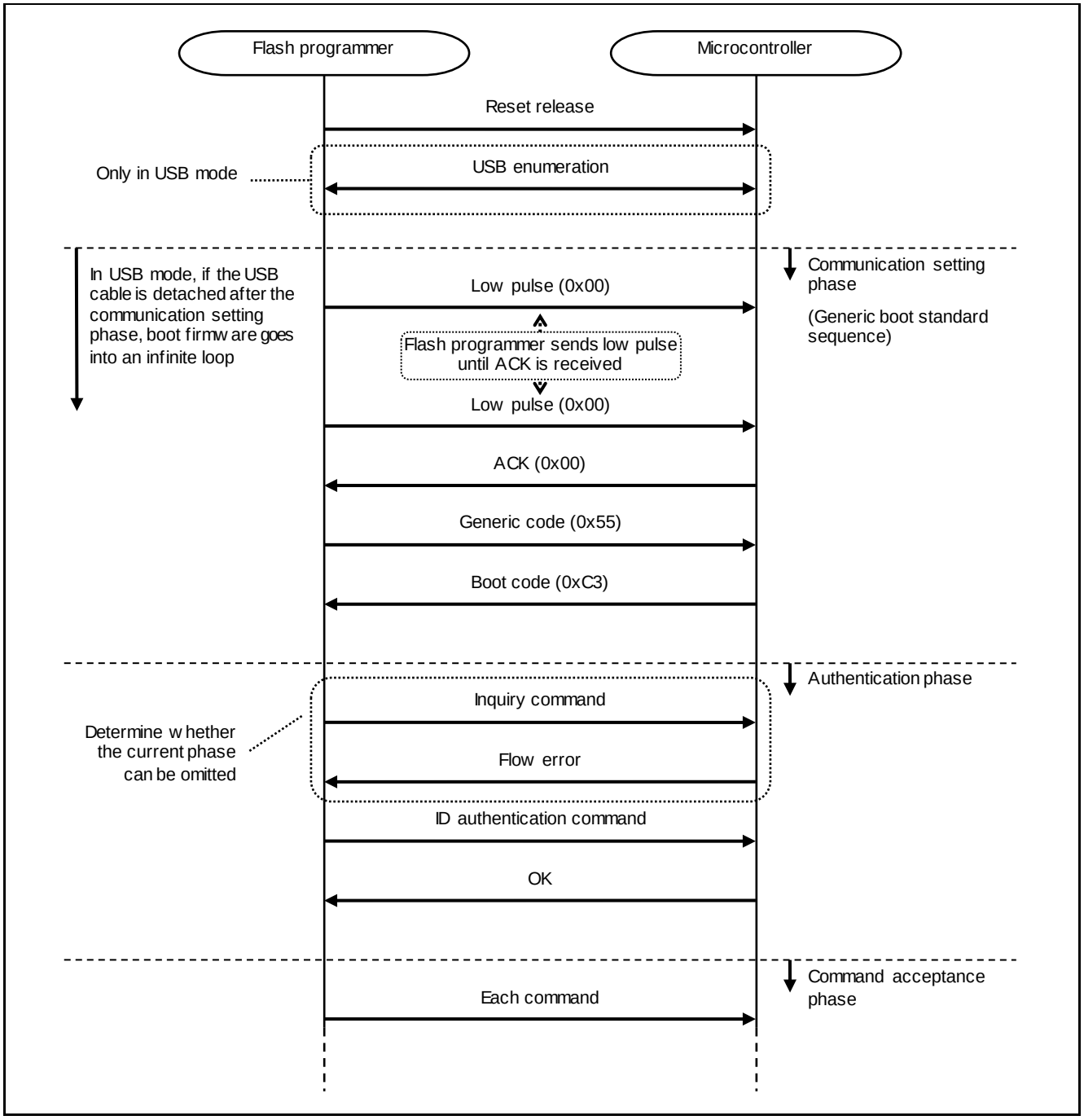


Figure 3.8 Beginning communication (secure)

(2) **If ID code is not stored [Non-secure]**

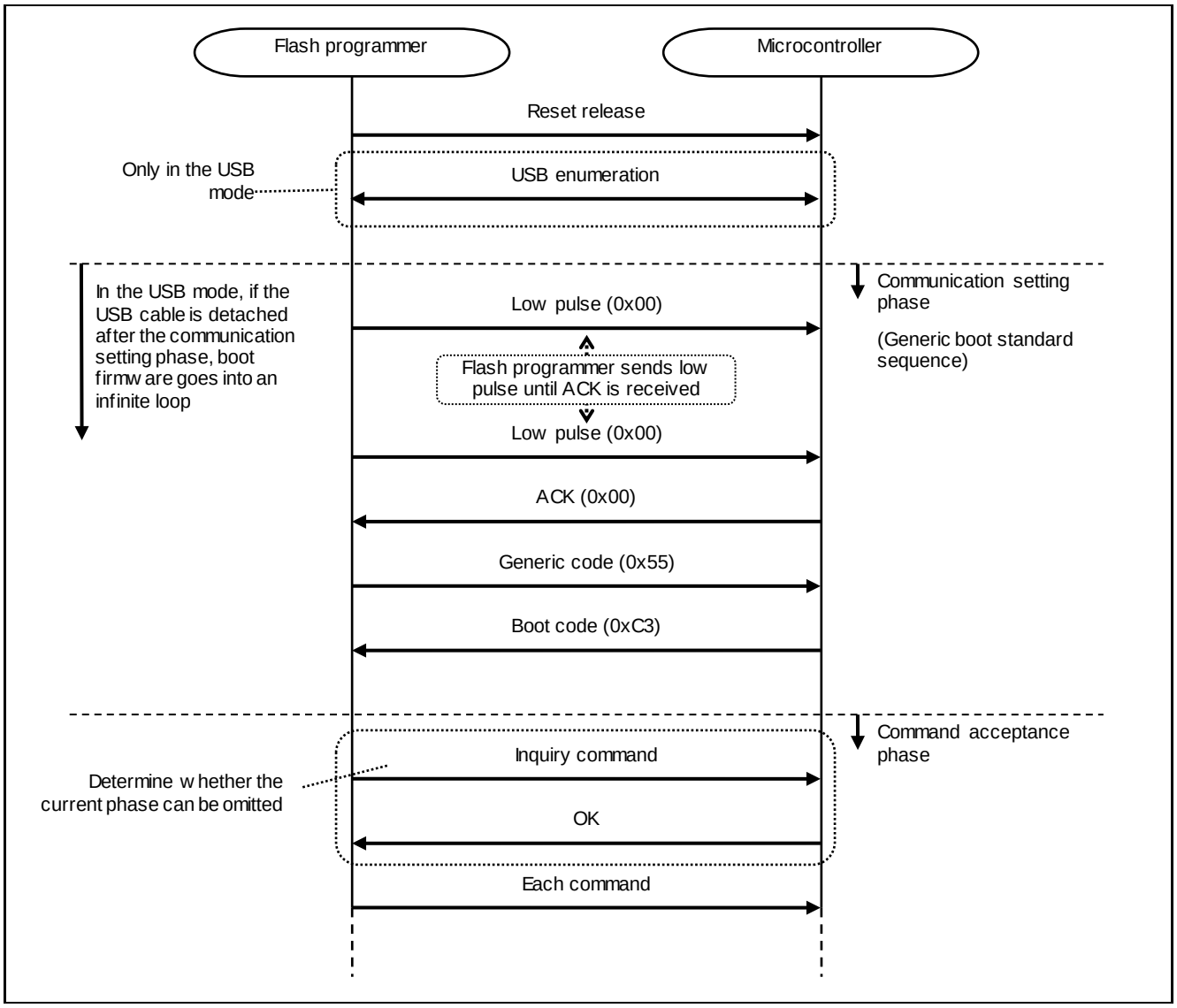


Figure 3.9 Starting communications (non-secure)

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 (adjustable 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 (see 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 2's complement) Example: LNH + LNL + COM + command information(1) + command information(2) + ... + command information(n) + SUM = 0x00
ETX	1 byte	0x03	End of packet

3.3.2 Data Packet

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

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

Symbol	Size	Code	Description
SOD	1 byte	0x81	Start of data packet
LNH	1 byte	-	Packet length (length = RES + Data) [High]
LNL	1 byte	-	Packet length (length = RES + Data) [Low]
RES	1 byte	-	Response code (see 3.4.2)
Data	-	-	Transmit data [max: 1024 bytes] Example: In case of Write command: Write data
SUM	1 byte	-	Sum of LNH + LNL + RES + Data (express as 2's complement) Example: LNH + LNL + RES + Data(1) + Data(2) + ... + Data(n) + SUM = 0x00
ETX	1 byte	0x03	End of packet

Note: If the packet length is 0 byte or over 1024 bytes, the value of RES is not defined.

3.4 Communication Command

3.4.1 List of Command Codes

Table 3.5 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 3.6 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 (such as illegal length, missing ETX)
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 the status registers of the flash sequencer (FACI or FCB) and returns the following status.

Status	Flash process	Condition
Erase error	S1/S3 Series	If ERERR is detected
	S7 Series, S5 Series	If ERSERR is detected
Write error	S1/S3 Series	If PRGERR or PRGERR01 is detected
	S7 Series, S5 Series	If PRGERR is detected
Sequencer error	S1/S3 Series	If ILGLERR or EILGLERR is detected
	S7 Series	If ILGLERR, FCUERR or FLWEERR is detected
	S5 Series	If ILGLERR or FLWEERR is detected

3.4.3 Executable Command in Each Phase

Table 3.7 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.

(1) **Command processing procedure**

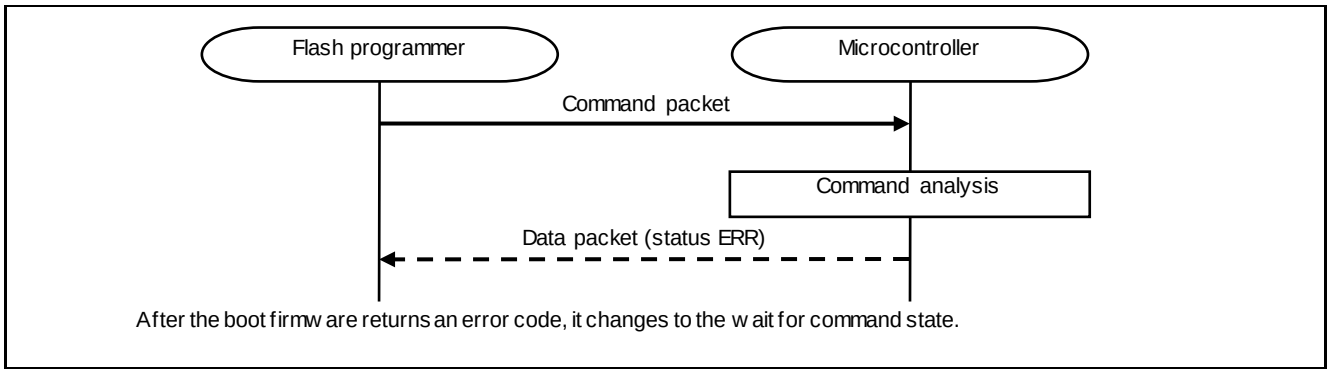


Figure 3.10 Unsupported command processing

(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) **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

(4) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.8 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.

(1) **Command processing procedure**

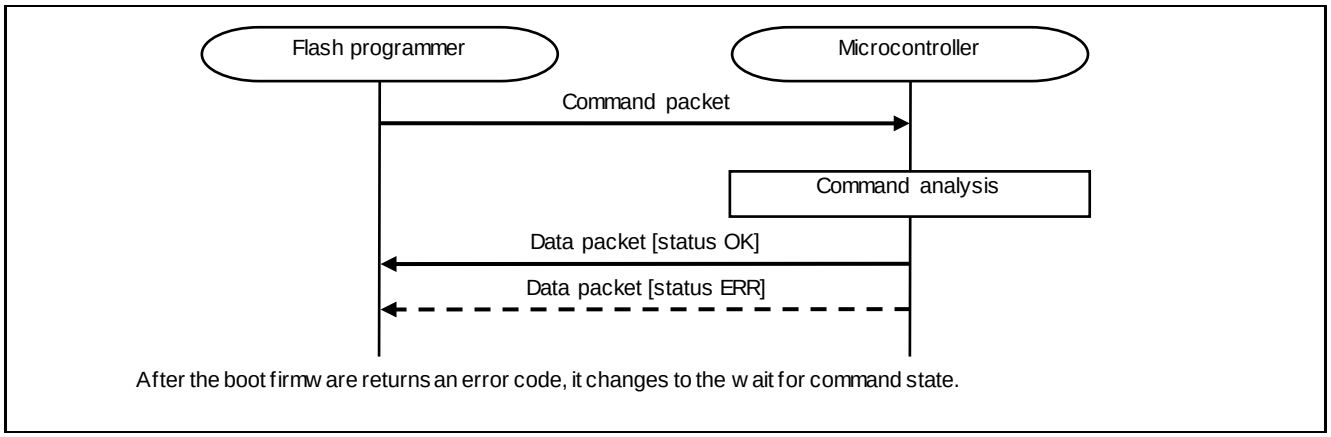


Figure 3.11 Inquiry command processing

(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) **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

(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

(5) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.9 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 only be performed in command acceptance phase.

(1) **Command processing procedure**

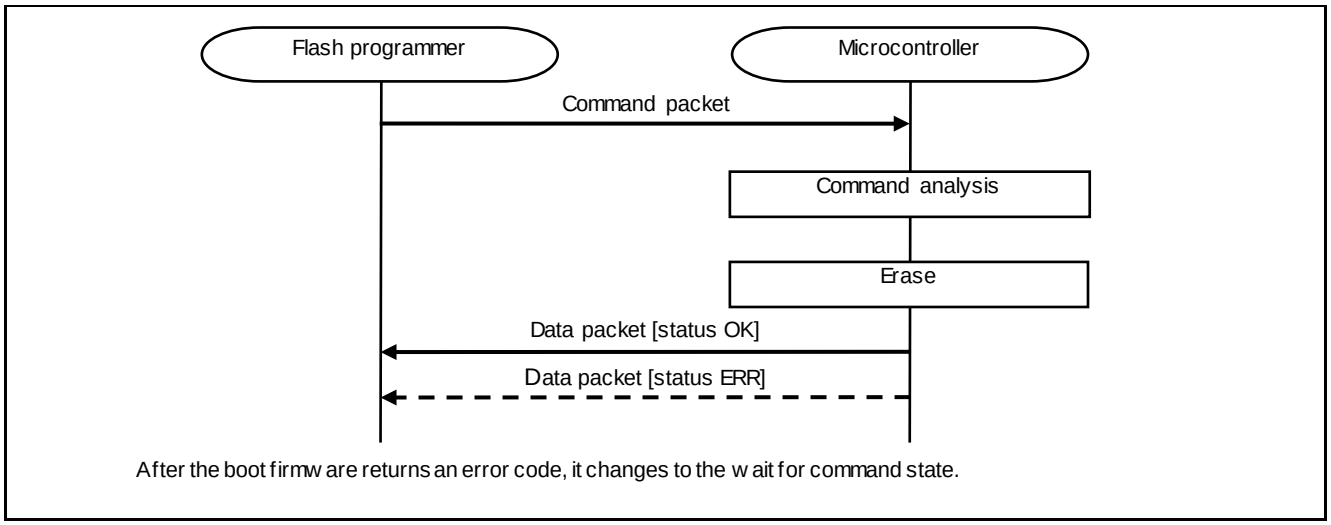


Figure 3.12 Erase command processing

(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) **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

(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

(5) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.10 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 is not available for this area		5	
The start address is larger 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 the code flash	Protection error	6	0xDA
If the target area is the configuration 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 the 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 only be performed in the command acceptance phase.

(1) **Command processing procedure**

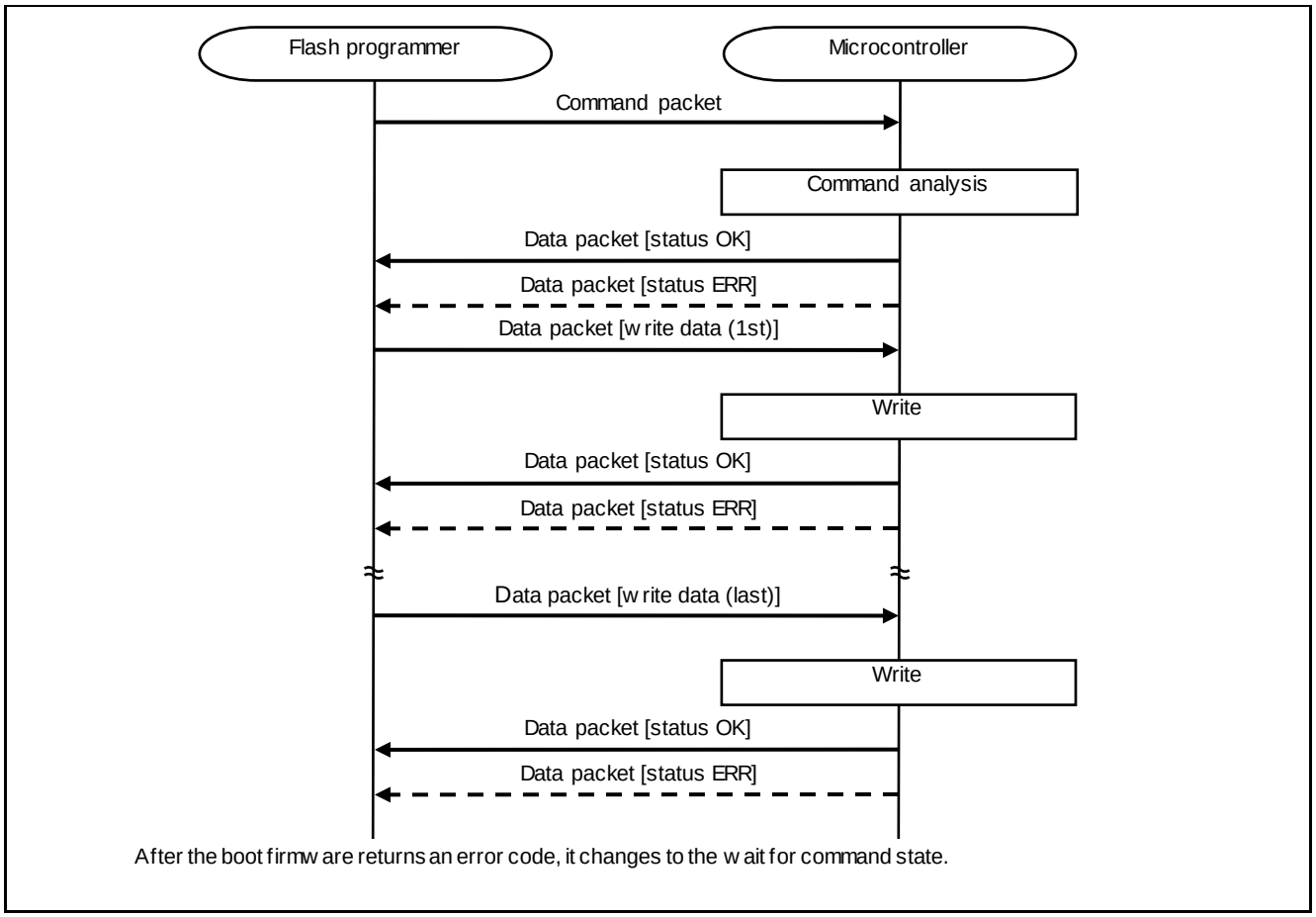


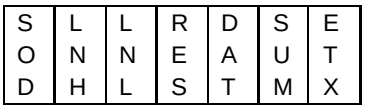
Figure 3.13 Write command processing

(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

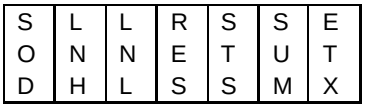
(3) **Data packet [write data]**



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

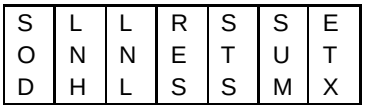
N = 1–1024

(4) **Data packet [status OK]**



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

(5) **Data packet [status ERR]**



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

(6) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.11 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 number of all data packets received from the flash programmer is above 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 is not valid		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 is not available for this area		5	
If the start address is greater 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 the code flash	Protection error	6	0xDA
If the target area contain FAWS or FAWE or BTFLG in the configuration 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

(7) **Basic precautions for Write command**

If you write any other value except 0xFFFF to SECMPUAC, boot firmware hangs after the next reset release. See section 2.5.

If you write 0 to ID[127] in the configuration area, the device cannot perform ID authentication. The ID authentication command returns a serial programming disable error. See section 3.4.9.

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

In addition, device configuration data may be assigned in the Option-Setting Memory area and the configuration 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 only be performed in the command acceptance phase.

(1) **Command processing procedure**

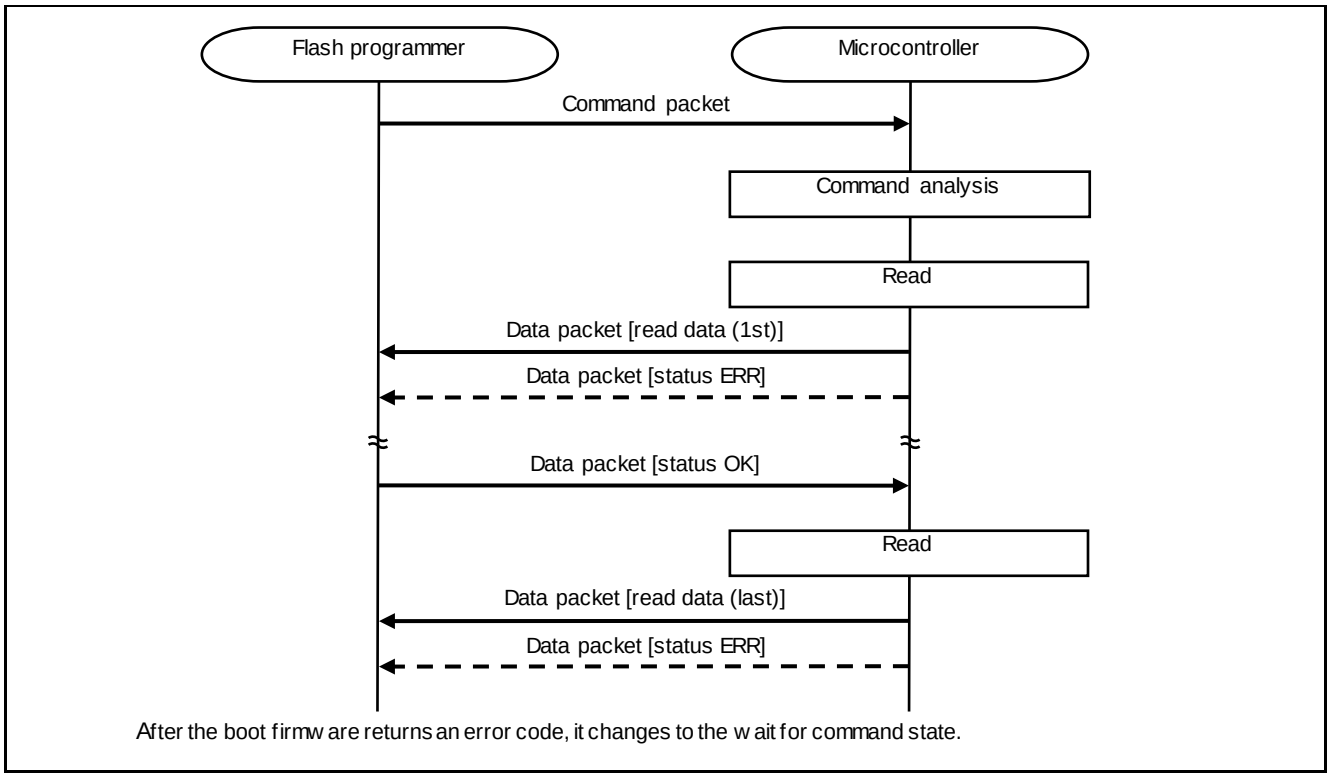


Figure 3.14 Read command processing

(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) **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		

(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

(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

(6) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.12 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 greater than the end address		5	

*1: Scope of each area is subject to the 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 only be performed in the authentication phase.

Table 3.13 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 configuration 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

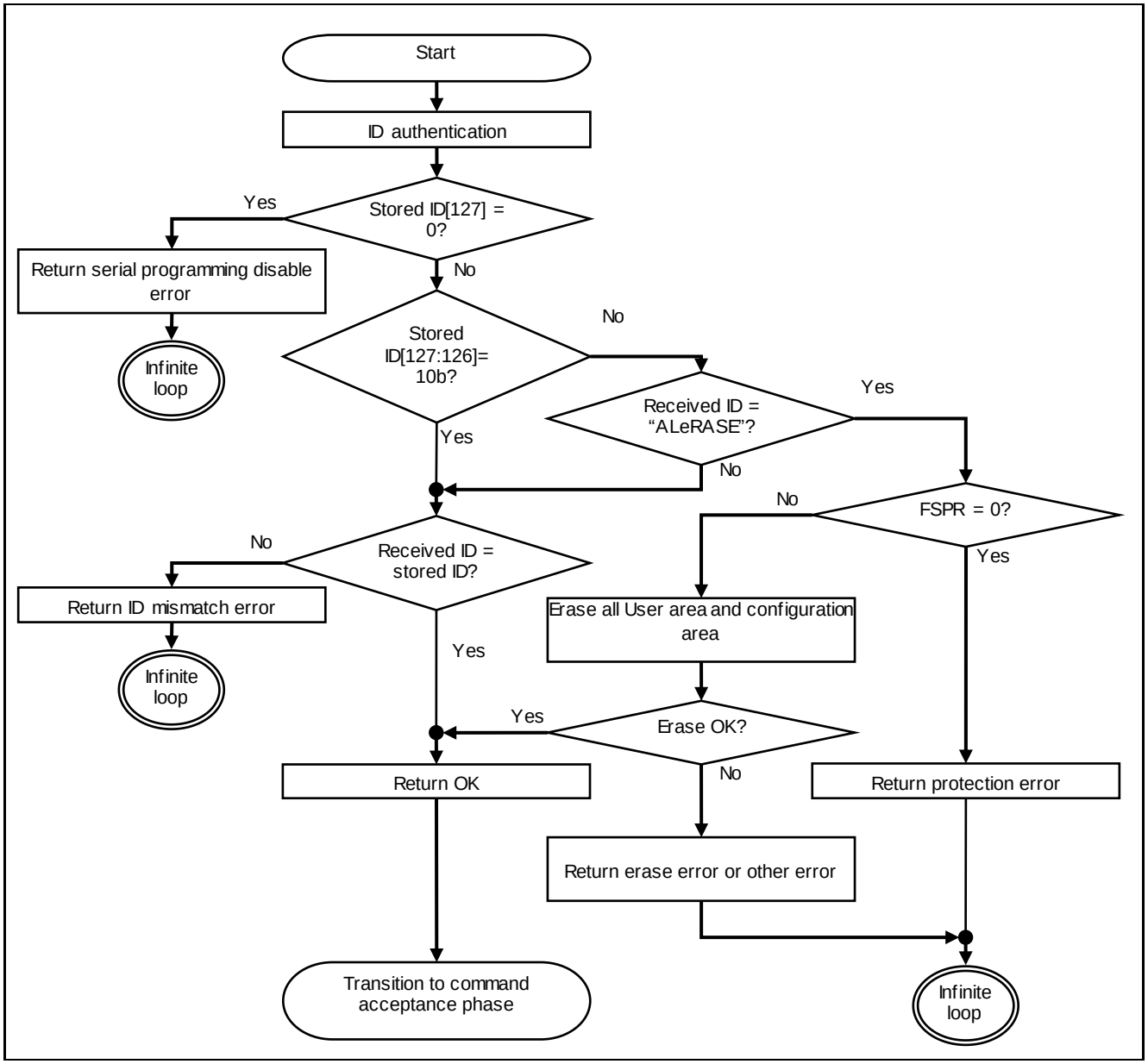


Figure 3.15 ID Authentication

- **Command processing procedure**

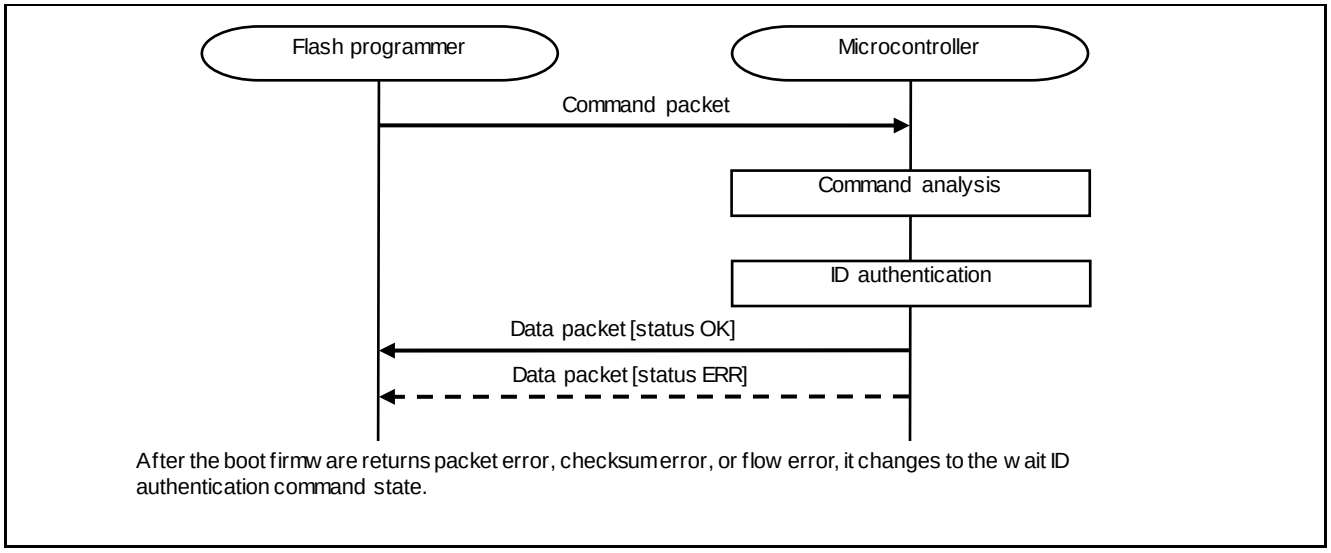


Figure 3.16 ID authentication command processing

(1) **Command packet**

SOH	LNH	LNL	COM	Authentication phase	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

(2) **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) **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

(4) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.14 ID authentication status codes

Condition	Status	Priority	Code
For “ALeRASE”, if erasure of all User area and configuration 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 the 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 upper-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 does not change. This command can only be performed in the command acceptance phase. This command does not change the communication speed of USB communication.

(1) **Command processing procedure**

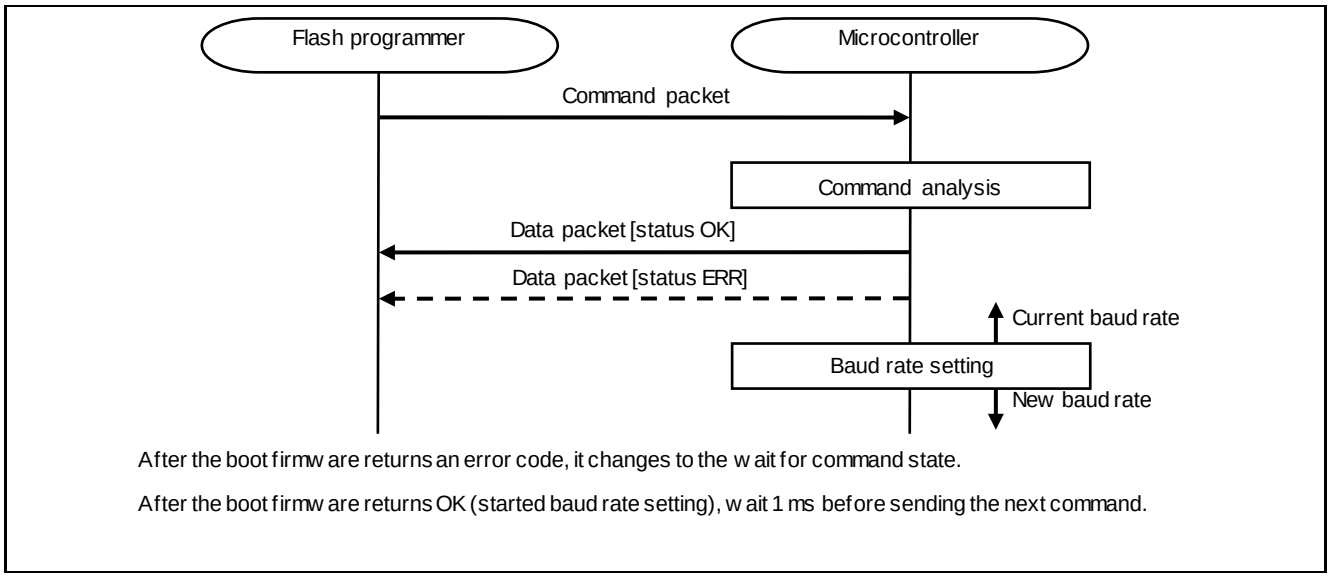


Figure 3.17 Baud rate setting command processing

(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 2000000 in decimal converted to hexadecimal is 0x1E8480
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

(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

(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

(5) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.15 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 the 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			

(6) **Baud rate setting values**

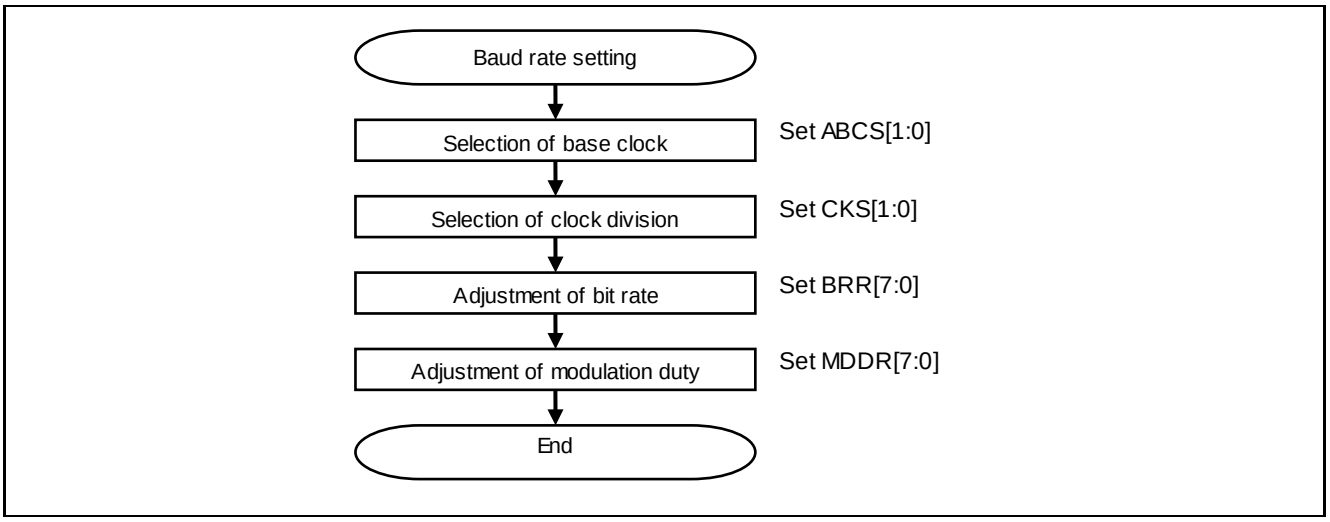


Figure 3.18 Baud rate setting values

Table 3.16 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 overflowed	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 overflowed	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]

Table 3.17 and Table 3.18 show example typical settings for baud rate.

Table 3.17 Baud rate settings for Renesas Synergy™ S7 Series MCU [when SCI = 60 MHz]

Intended baud rate	ABCS	CKS[1:0]	BRR[7:0]	MDDR[7:0]	Accuracy
9600	0	00b	C2h	FFh	-0.3%
1000000	0	00b	00h	88h	-0.4%
1500000	0	00b	00h	CCh	-0.4%
2000000	1	00b	00h	88h	-0.4%
3000000	1	00b	00h	CCh	-0.4%
3500000	1	00b	00h	EEh	-0.4%
3750000	1	00b	00h	(Not used)	0.0%

Table 3.18 Baud rate settings for Renesas Synergy™ S3 Series MCU [when SCI = 24 MHz]

Intended baud rate	ABCS	CKS[1:0]	BRR[7:0]	MDDR[7:0]	Accuracy
9600	0	00b	4Dh	FFh	-0.3%
1000000	1	00b	00h	AAh	-0.4%
1500000	1	00b	00h	(Not used)	0.0%
2000000	disabled	disabled	disabled	disabled	-

3.4.11 Signature Request Command

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

(1) **Command processing procedure**

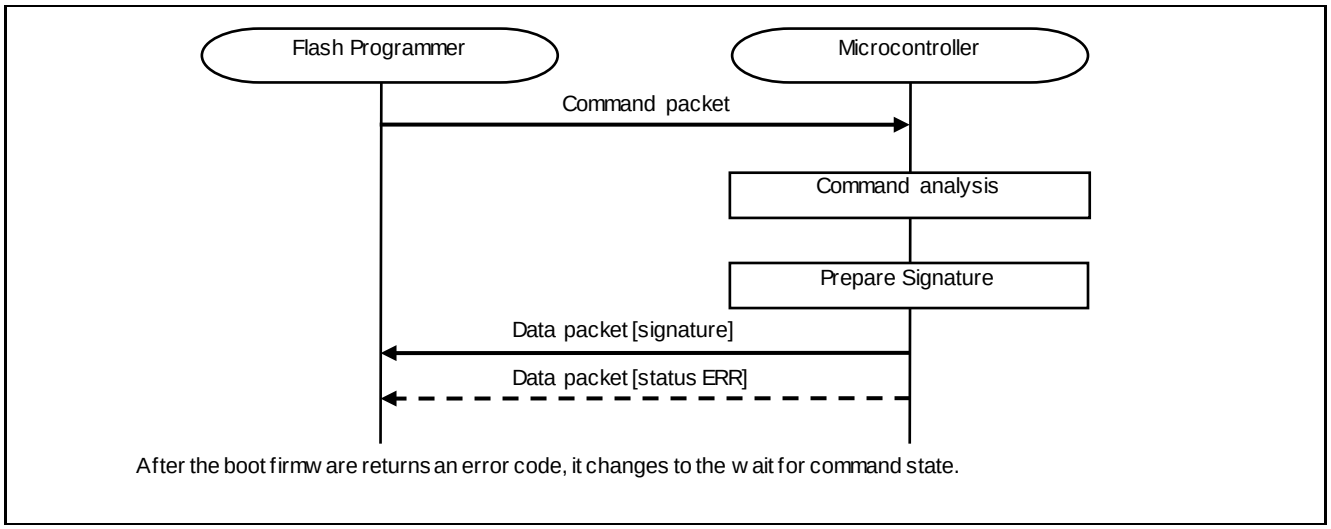


Figure 3.19 Signature request command processing

(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) **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] Example: 20 MHz (20000000 Hz) -> 0x01, 0x31, 0x2D, 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 (1st) 1. User area in Code flash (2nd) 2. User area in Data flash 3. Configuration area -> 0x04
TYP	(1 byte)	Type code 0x01 (Synergy MCU + Synergy S7 Series) 0x02 (Synergy MCU + Synergy S1/S3 Series) 0x03 (Synergy MCU + Synergy S5 Series)
BFV	(2 byte)	Boot firmware version Example: Ver10.8 -> 0x0A, 0x08
SUM	(1 byte)	Sum data
ETX	(1 byte)	0x03

(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

(5) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.19 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 the 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 the Erase command and the Write command follows this area information. This command can only be performed in the command acceptance phase.

(1) **Command processing procedure**

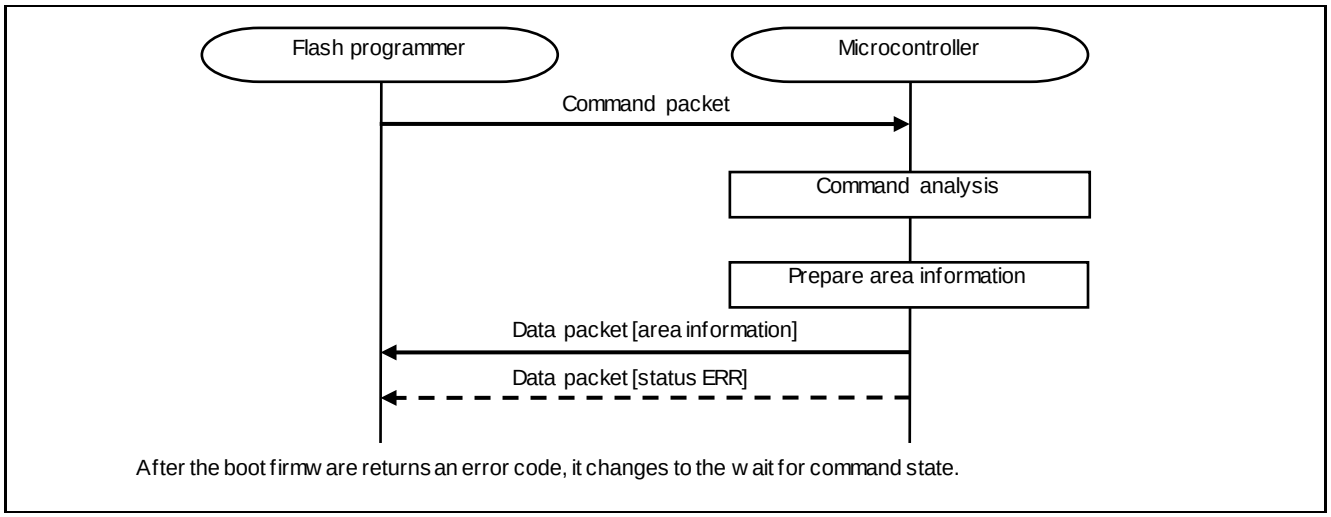


Figure 3.20 Area information request command processing

(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) **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 (Configuration 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.

(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

(5) **Status code from microcontroller [Priority high: 1 -> low: 10]**

Table 3.20 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 the 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

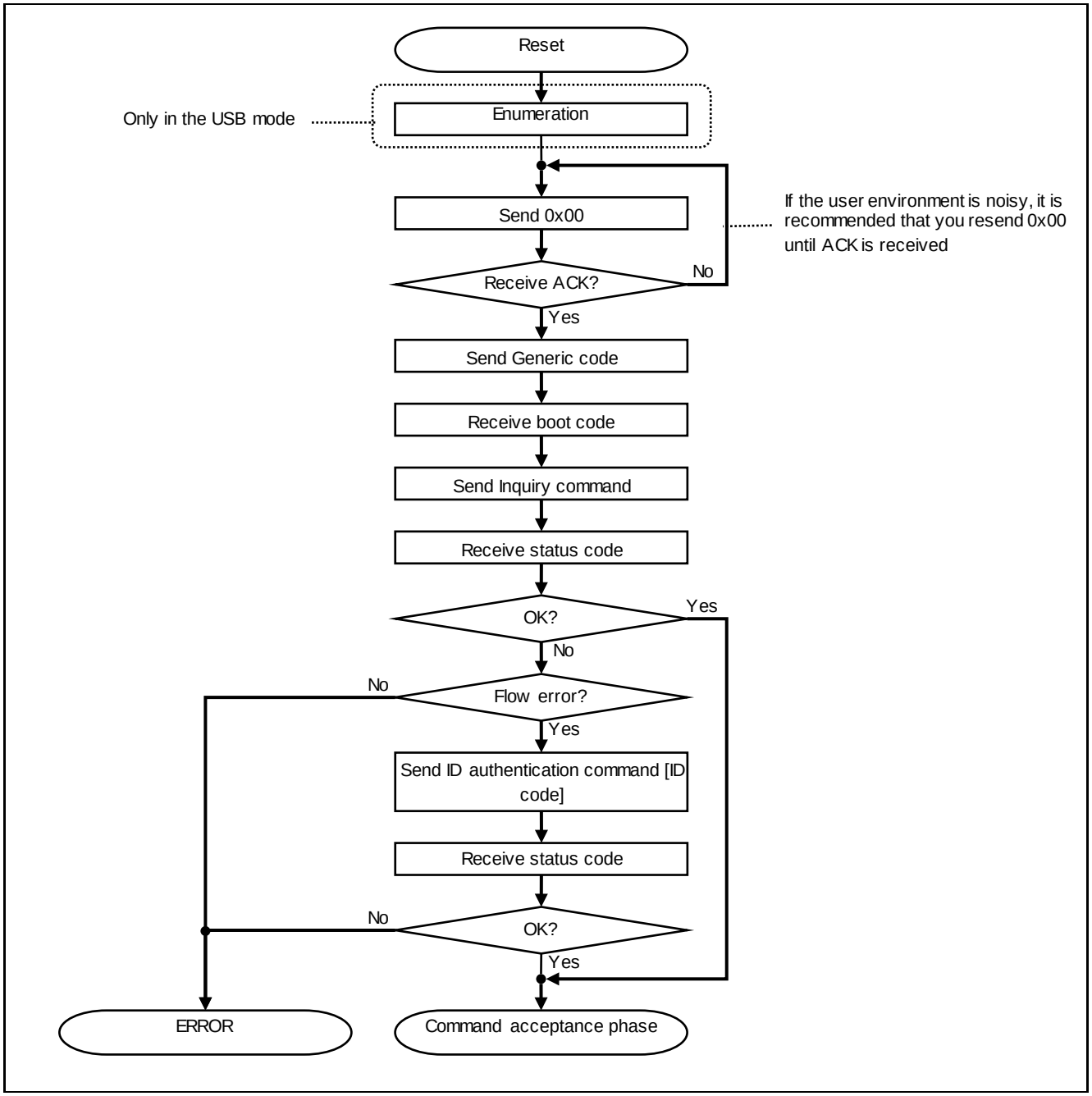


Figure 3.21 Starting communication

3.5.2 Total Area Erasure

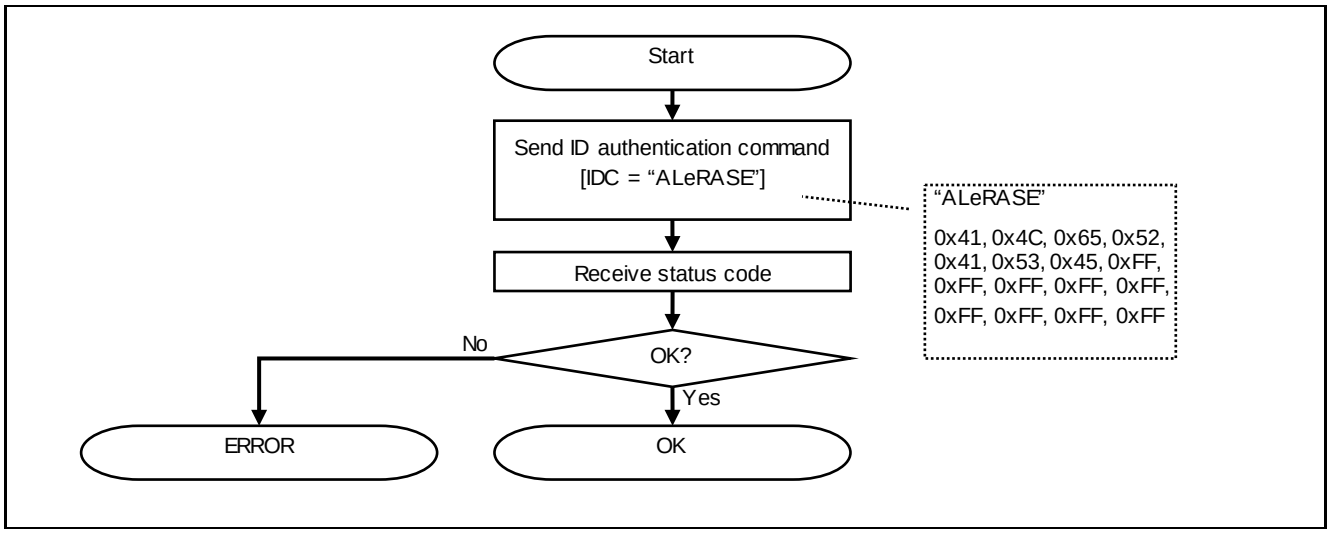


Figure 3.22 Total area erasure

3.5.3 Acquisition of Device Information

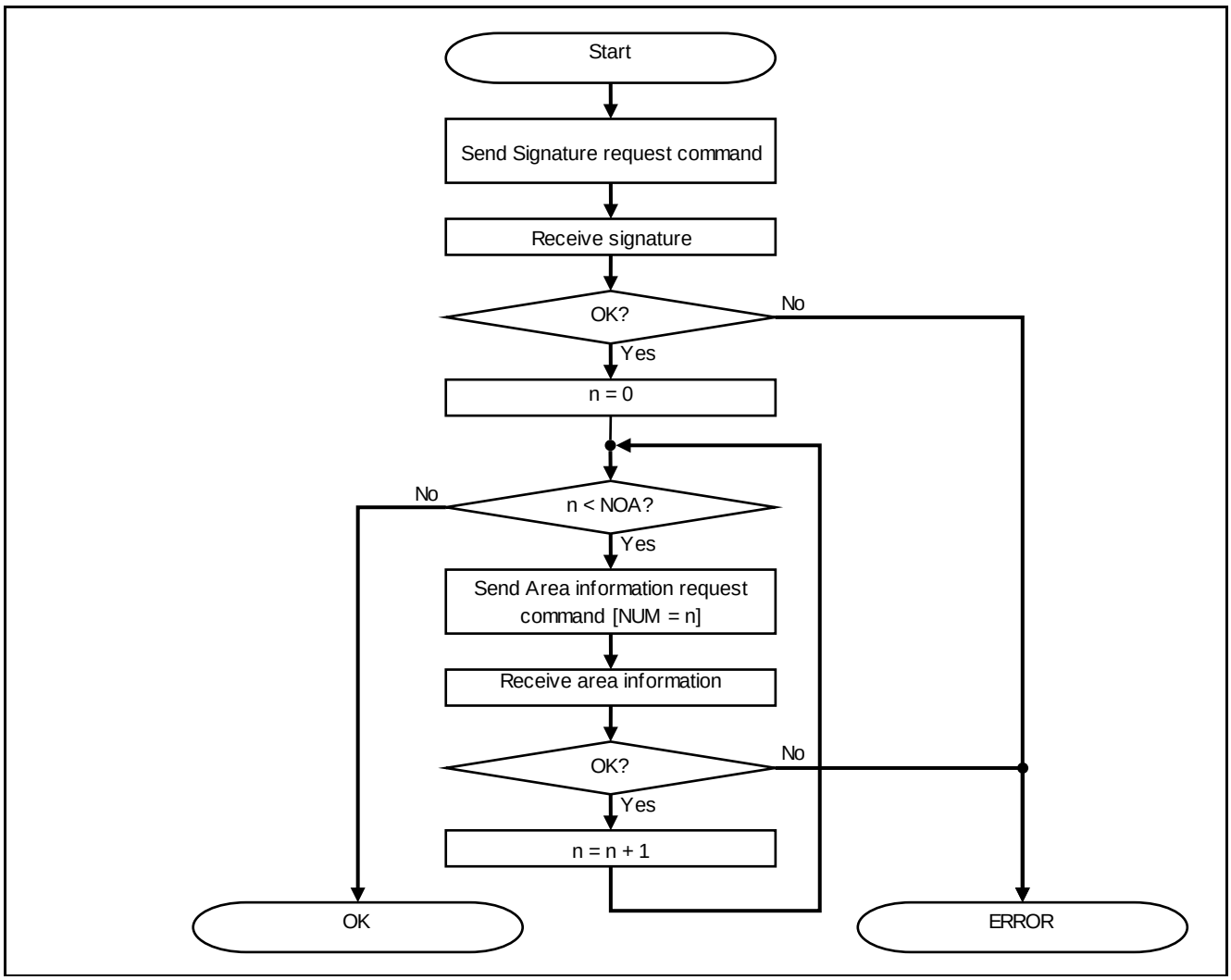


Figure 3.23 Acquisition of device information

3.5.4 Code and Data in User Area Updates

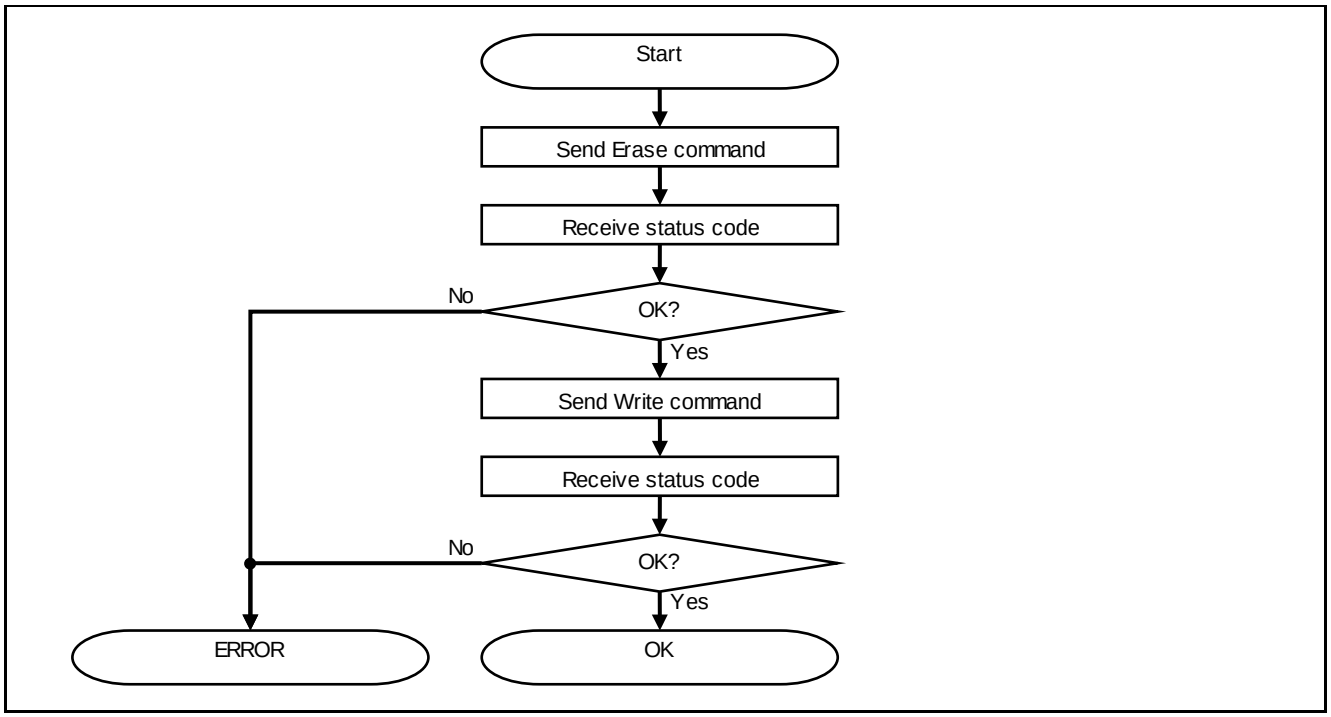


Figure 3.24 Code and data in User area updates

3.5.5 Configuration Data Updates

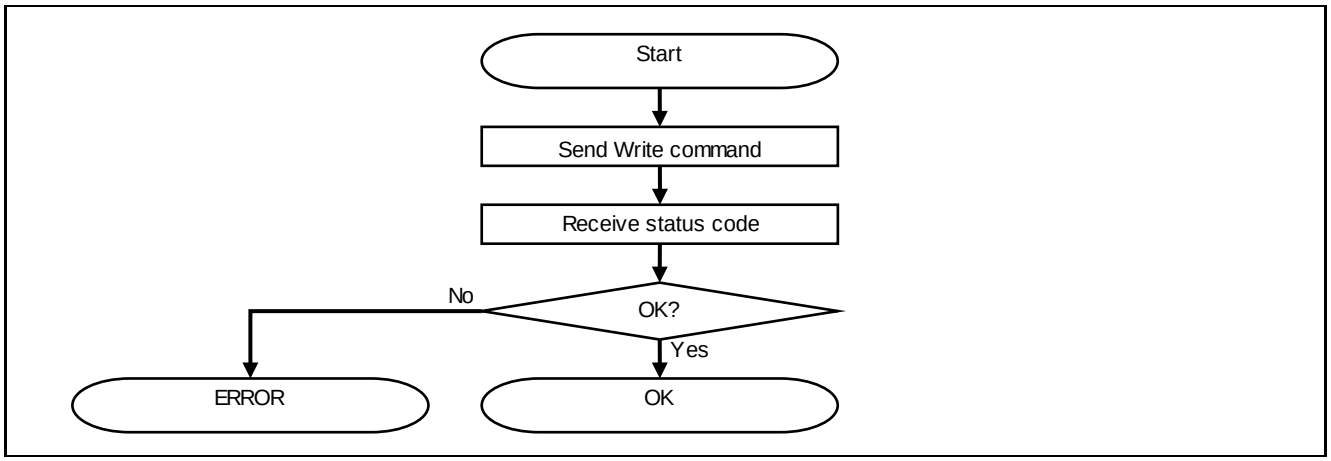


Figure 3.25 Configuration data updates

Website and Support

Support: https://synergygallery.renesas.com/support

Technical Contact Details

- America: https://www.renesas.com/en-us/support/contact.html
- Europe: https://www.renesas.com/en-eu/support/contact.html
- Japan: https://www.renesas.com/ja-jp/support/contact.html

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Apr 13, 2016	-	New document
1.10	Oct 13, 2017	-	First public release

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 disputes 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, drawing, chart, program, algorithm, 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 not alter, modify, copy, or otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
5. 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; and 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.

   Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges.
7. 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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.
9. 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0