PM0005
Programming Manual

AT32 Bootloader UART Protocol

Introduction

This user manual describes the UART protocol used in the AT32 MCU bootloader, detailing the supported commands and their operating procedure. For more information about the UART hardware resources and requirements for your device bootloader, refer to the document PM0007_AT32_Bootloader_Program_Manual.

Contents

1 UARTx bootloader code sequence 8

2 UARTx baud rate 9

3 Bootloader command set 10

4 Bootloader command description 11

4.1 Set ISP command 11

4.1.1 Flow chart on host and device sides 12

4.1.2 Data transmit on host side 13

4.2 Get Command 14

4.2.1 Flow chart on host and device sides 14

4.2.2 Data transmit on host side 15

4.3 Get Version command 16

4.3.1 Flowchart on host and device sides 16

4.3.2 Data transmit on host and device sides 17

4.4 Get Device ID command 18

4.4.1 Flowchart on host and device sides 18

4.4.2 Data transfer on host side 19

4.5 Read Memory command 20

4.5.1 Flowchart on host and device sides 20

4.5.2 Data transmit on host side 22

4.6 Jump command 22

4.6.1 Flowchart on host and device sides 23

4.6.2 Data transmit on host side 25

4.7 Write Memory command 25

4.7.1 Flowchart on host and device sides 26

4.7.2 Data transmit on host side 28

4.8 Erase command 28

4.8.1 Flowchart on host and device sides 30

4.8.2 Data transmit on host side 32

4.9 Erase and Program Protect command 33

4.9.1 Flowchart on host and device sides 33
4.9.2 Data transmit on host side 34
4.10 Erase and Program Unprotect 35
4.10.1 Flowchart on host and device sides 35
4.10.2 Data transmit on host side 36
4.11 Access Protect command 37
4.11.1 Flowchart on host and device sides 37
4.11.2 Data transmit on host side 38
4.12 Access Unprotect command 39
4.12.1 Flowchart on host and device sides 39
4.12.2 Data transmit on host side 40
4.13 Firmware CRC command 41
4.13.1 Flowchart on host and device sides 41
4.13.2 Data transmit on host side 43
4.14 Enable SPIM command 43
4.14.1 Flowchart on host and device sides 44
4.14.2 Data transmit on host side 45
4.15 Enable sLib 46
4.15.1 Flowchart on host and device sides 47
4.15.2 Data transmit on host side 49
4.16 Disable sLib 49
4.16.1 Flowchart on host and device sides 50
4.16.2 Data transmit on host side 52
4.17 Get sLib status 52
4.17.1 Flowchart on host and device sides 53
4.17.2 Data transmit on host side 55
4.18 SPIM Remap 56
4.18.1 Flowchart on host and device sides 56
4.18.2 Data transmit on host side 57
4.19 Reset Device 58
4.19.1 Flowchart on host and device sides 58
4.19.2 Data transmit on host side 59
4.20 Advanced Access Protect 60
4.20.1 Flowchart on host and device sides 60

4.20.2 Data transmit on host side 61

5 Revision history 62

List of tables

Table 1 List of erase type 29
Table 2 Document revision history 62

List of figures

Figure 1 Bootloader Main Loop 8
Figure 2 Set ISP: host side 12
Figure 3 Set ISP: device side 13
Figure 4 Get Commands: host side 14
Figure 5 Get Commands: device side 15
Figure 6 Get Version: host side 16
Figure 7 Get Version: device side 17
Figure 8 Get Device ID: host side 18
Figure 9 Get Device ID: device side 19
Figure 10 Read Memory: host side 20
Figure 11 Read Memory: device side 21
Figure 12 Jump: host side 23
Figure 13 Jump: device side 24
Figure 14 Write Memory: host side 26
Figure 15 Write Memory: device side 27
Figure 16 Erase: host side 30
Figure 17 Erase: device side 31
Figure 18 Erase and Program Protect: host side 33
Figure 19 Erase and Program Protect: device side 34
Figure 20 Erase and Program Unprotect: host side 35
Figure 21 Erase and Program Unprotect: device side 36
Figure 22 Access Protect: host side 37
Figure 23 Access Protect: device side 38
Figure 24 Access Unprotect: host side 39
Figure 25 Access Unprotect: device side 40
Figure 26 Firmware CRC: host side 41
Figure 27 Firmware CRC: device side 42
Figure 28 Enable SPIM: host side 44
Figure 29 Enable SPIM: device side 45
Figure 30 Enable sLib: host side 47
Figure 31 Enable sLib: device side 48
Figure 32 Disable sLib: host side 50
Figure 33 Disable sLib: device side 51
Figure 34 Get sLib status: host side 53

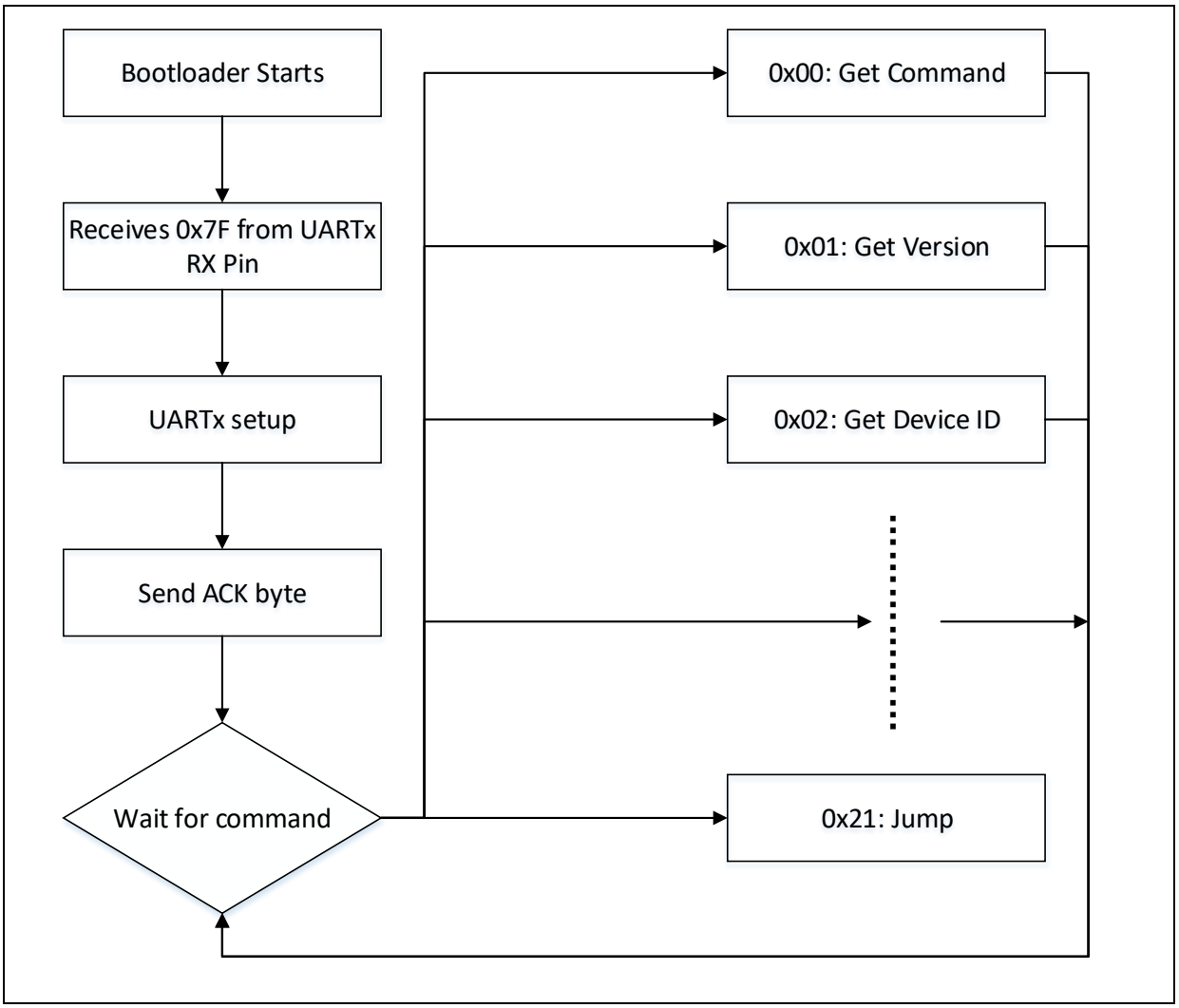
Figure 35 Get sLib status: device side 54
Figure 36 SPIM Remap: host side 56
Figure 37 SPIM Remap: device side 57
Figure 38 Reset Device: host side 58
Figure 39 Reset Device: device side 59
Figure 40 Advanced Access Protect: host side 60
Figure 41 Advanced Access Protect: device side 61

1 UARTx bootloader code sequence

When the UART programming mode is enabled, the bootloader begins to scan the UARTx Rx pin, waiting to receive the 0x7F data frame: a start bit, 0x7F data bits, parity check bit and a stop bit.

The duration of the data frame is measured using the Systick timer, and then the baud rate of the UARTx is calculated. An acknowledge byte (0x79) is returned to the host, signaling that the device is ready to receive commands.

Figure 1 Bootloader Main Loop



2 UARTx baud rate

Setting the baud rate:

AT32 Bootloader supports automatic baud rate detection. The deviation between the device baud rate and the real baud rate of the host must be below 2.5%. The minimum tested baud rate is 1200, and the maximum tested baud rate is 256000.

Deviation = $\left|\frac{Device\ baud\ rate - Host\ baud\ rate}{Device\ baud\ rate}\right| \times 100\%$. The deviation must be below 2.5%.

Bootloader baud rate detection method:

Once the MCU gets started, a 0x7F data is transmitted from the host to the device. This data includes a start bit, an 8-bit 0x7F data, even parity check bit and a stop bit.

When the data is detected by the device, a UARTx baud rate is configured accordingly, and then an acknowledge byte (0x79) is transmitted to the host. After the 0x79 is received by the host, it represents a successful connection, and the host continues to send commands.

ACK = 0x79

NACK = 0x1F

3 Bootloader command set

The number of the supported commands depends on the product series. However, for the same command set such as read, write or erase, they are entirely compatible across all the AT32 MCU series.

Bootloader commands

Command	Code	Description	Supported AT32 MCU series
Set ISP	0xFA	Get the version of the host.	AT32F413xx，AT32F415xx，AT32F403Axx，AT32F407xx，AT32F421xx，AT32A403Axx
Get Commands(1)	0x00	Get the allowed commands supported by the device	ALL
Get Version	0x01	Get the protocol version of the bootloader	ALL
Get Device ID(1)	0x02	Get the device ID.	ALL
Read memory	0x11	Read data of the memory starting from an address specified by the application	ALL
Jump	0x21	Jump to memory address executable application	ALL
Write memory	0x31	Write data to memory starting from an address specified by the application	ALL
Erase	0x44	Erase memory	ALL
Erase and program protect	0x63	Enable erase/program protection	ALL
Erase and program unprotect	0x73	Disable erase/program protection	ALL
Access Protect	0x82	Enable access protection	ALL
Access Unprotect	0x92	Disable access protection	ALL
Firmware CRC	0xAC	Compute a CRC value on a given memory area	ALL
Enable SPIM(bank3)	0xB3	Enable external memory	MCU with SPIM feature
Enable sLib	0xD0	Enable sLib	MCU with sLib feature
Disable sLib	0xD1	Disable sLib	MCU with sLib feature
Get sLib Status	0xD2	Get sLib status	MCU with sLib feature
SPIM Remap	0xD3	Enable SPIM IO pin remapping	MCU with SPIM feature
Reset Device	0xD4	Reset device	AT32F403xx does not support
Advanced Access Protect	0xD6	Enable advanced access protection	MCU with advanced access protection

Note: A “Set ISP” command must be sent first before starting the “Get Commands” and “Get Device ID” commands for the AT32F413,， AT32F415, AT32F403A, AT32F407, AT32F421 and AT32A403A series. If a “Set ISP” command is sent to a device which does not support it, an “NACK” byte is transmitted from the device side, which can be discarded by the host. The host continues its operations.

4 Bootloader command description

This chapter describes how to use each of the bootloader commands from the point of the host side and the device side.

4.1 Set ISP command

This command allows the user to distinguish between the hosts. For the MCUs that support this command, this command must be sent first in order for “Get Commands” and “Get Device ID” to be activated. For those that do not support it, whether or not to send this command has no impact on the subsequent commands.

In order to ensure compatibility between different MCUs in terms of operation process, this command is divided into two parts on the host side.

- The host sends a “Set ISP” command, and the device transmits an “ACK” byte. Then the host continues to send 4-byte ID code: 0x02, 0x03, 0x54, 0x41, and a 1-byte checksum. After this, the device sends an “ACK” byte to the host, signaling the end of the command and that host is ready to send the next command.
- The host sends a “Set ISP” command, and the device transmits an “NACK” byte. After receiving this NACK byte by the host, it signals the completion of the command, and the host is ready to send next command.

This command is allowed and executed when the access protection is enabled.

4.1.1 Flow chart on host and device sides

Figure 2 Set ISP: host side

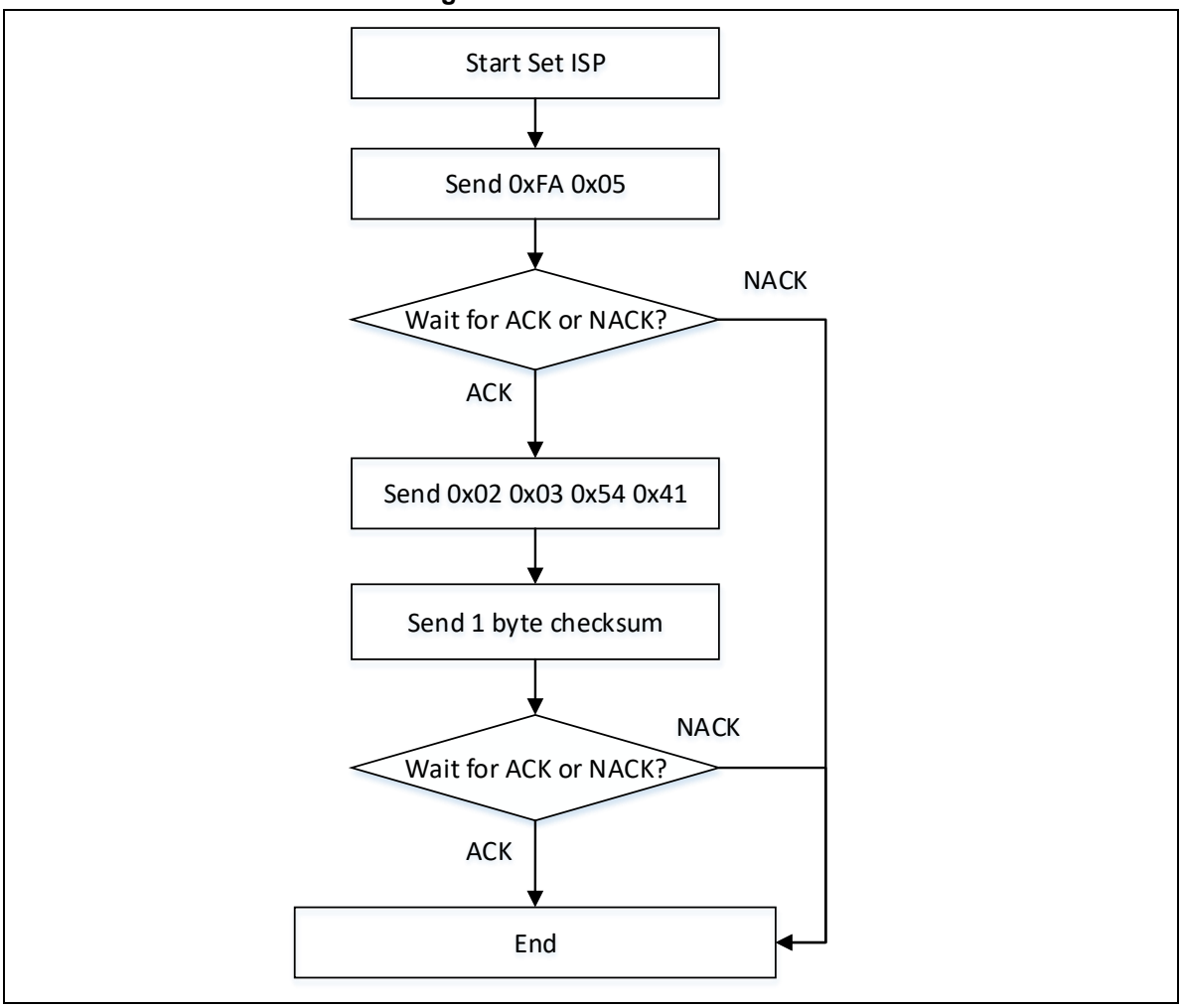
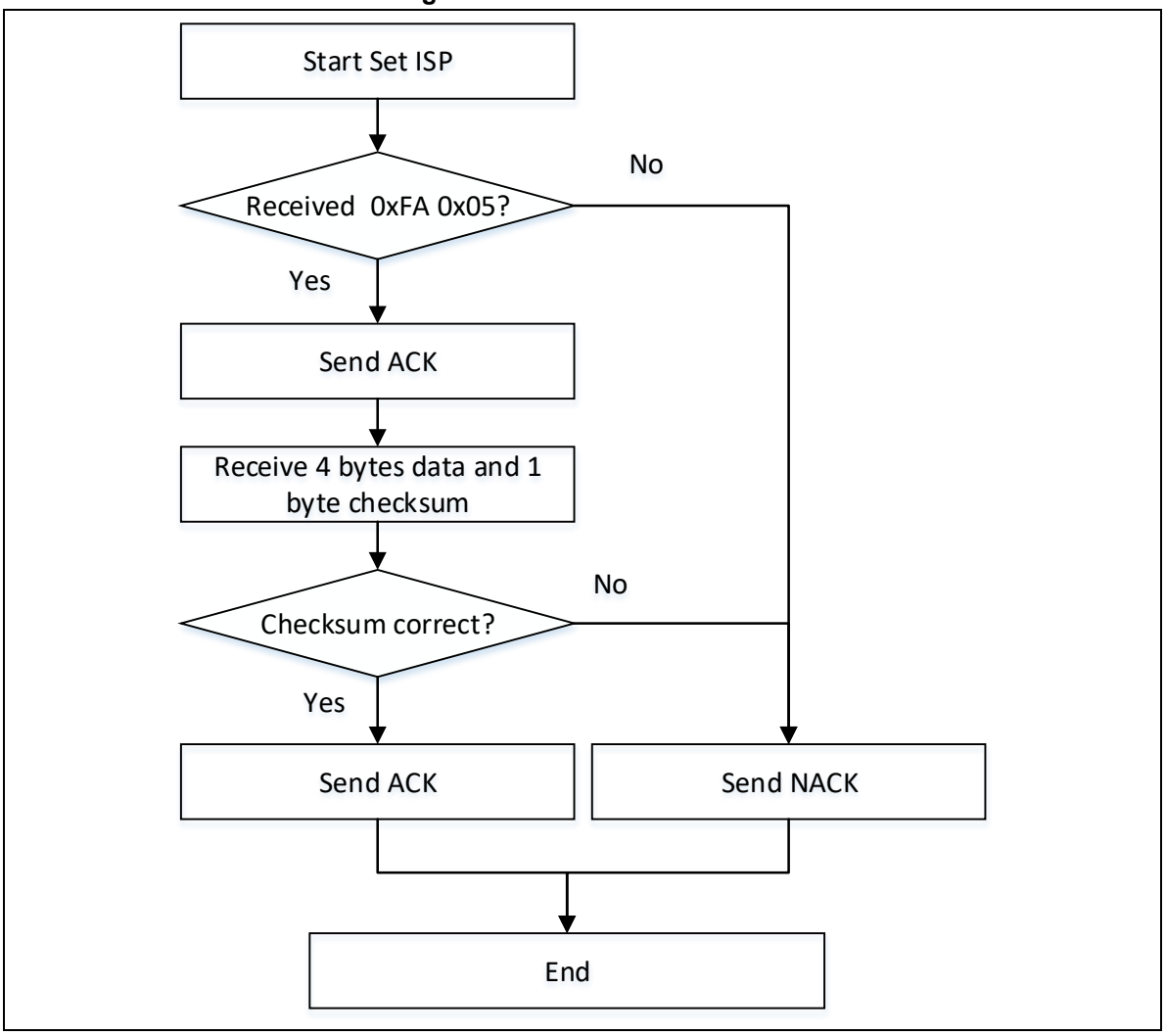


Figure 3 Set ISP: device side



4.1.2 Data transmit on host side

Send	Receive	Data	Description
1		0xFA	
2		0x05	
	1	ACK/NACK	When the NACK byte is received, it represents the end of the command
3		0x02	
4		0x03	
5		0x54	
6		0x41	
7		*	Checksum(Byte3~Byte6)
	2	ACK	End of the command

4.2 Get Command

This command is used to get the protocol version and the supported commands. The number of the supported commands depends on the MCU series.

When the bootloader receives the command, it transmit an “ACK” byte to the host. Then the bootloader sends the number of bytes (one byte) to the host, the protocol version and the supported commands and after that, transmits an ACK byte.

This command is allowed and executed when the access protection is enabled.

Note: A “Set ISP” command must be sent first before sending “Get Commands” for the AT32F413, AT32F415, AT32F403A, AT32F407, AT32F421 and AT32A403A series.

4.2.1 Flow chart on host and device sides

Figure 4 Get Commands: host side

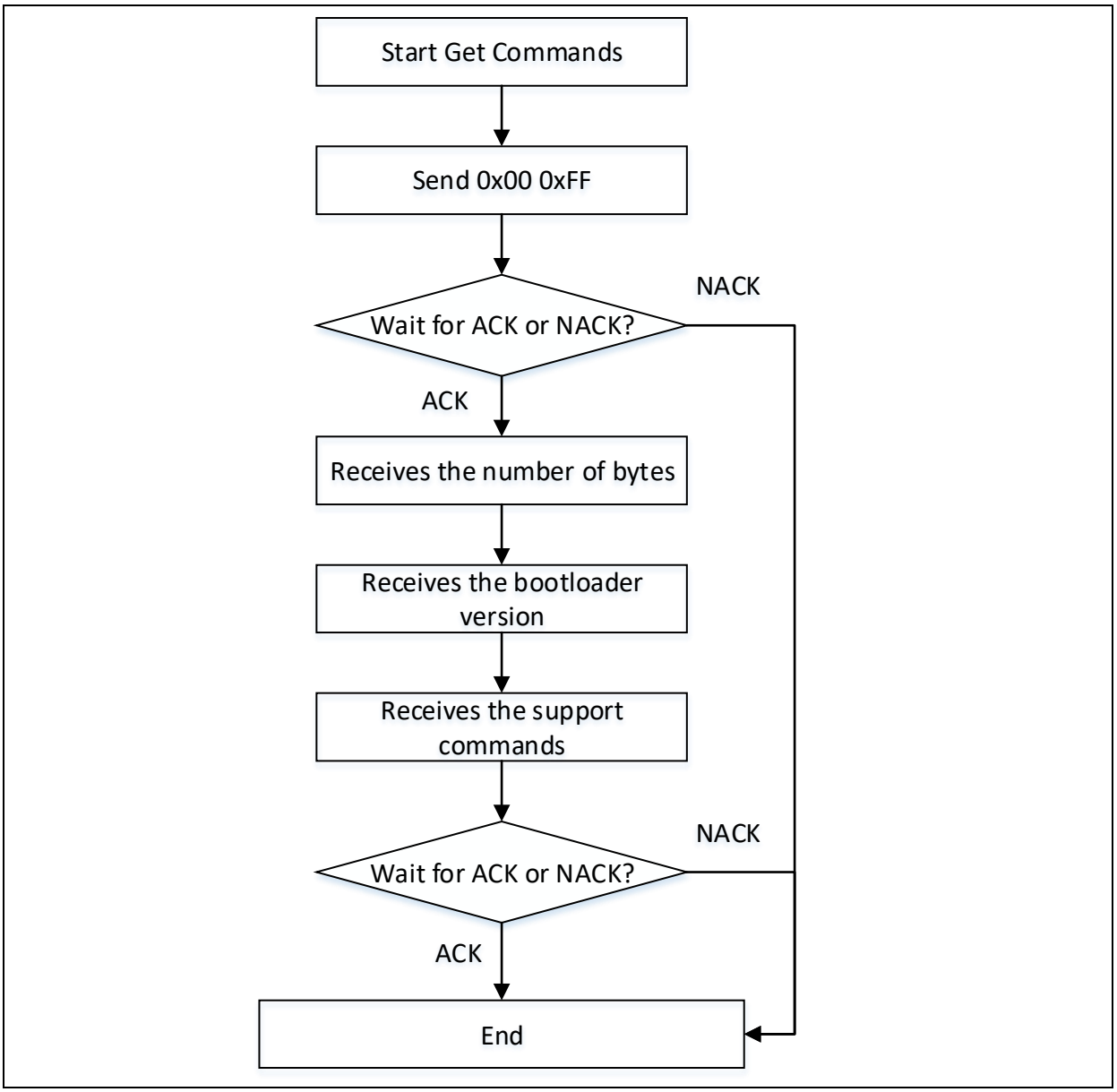
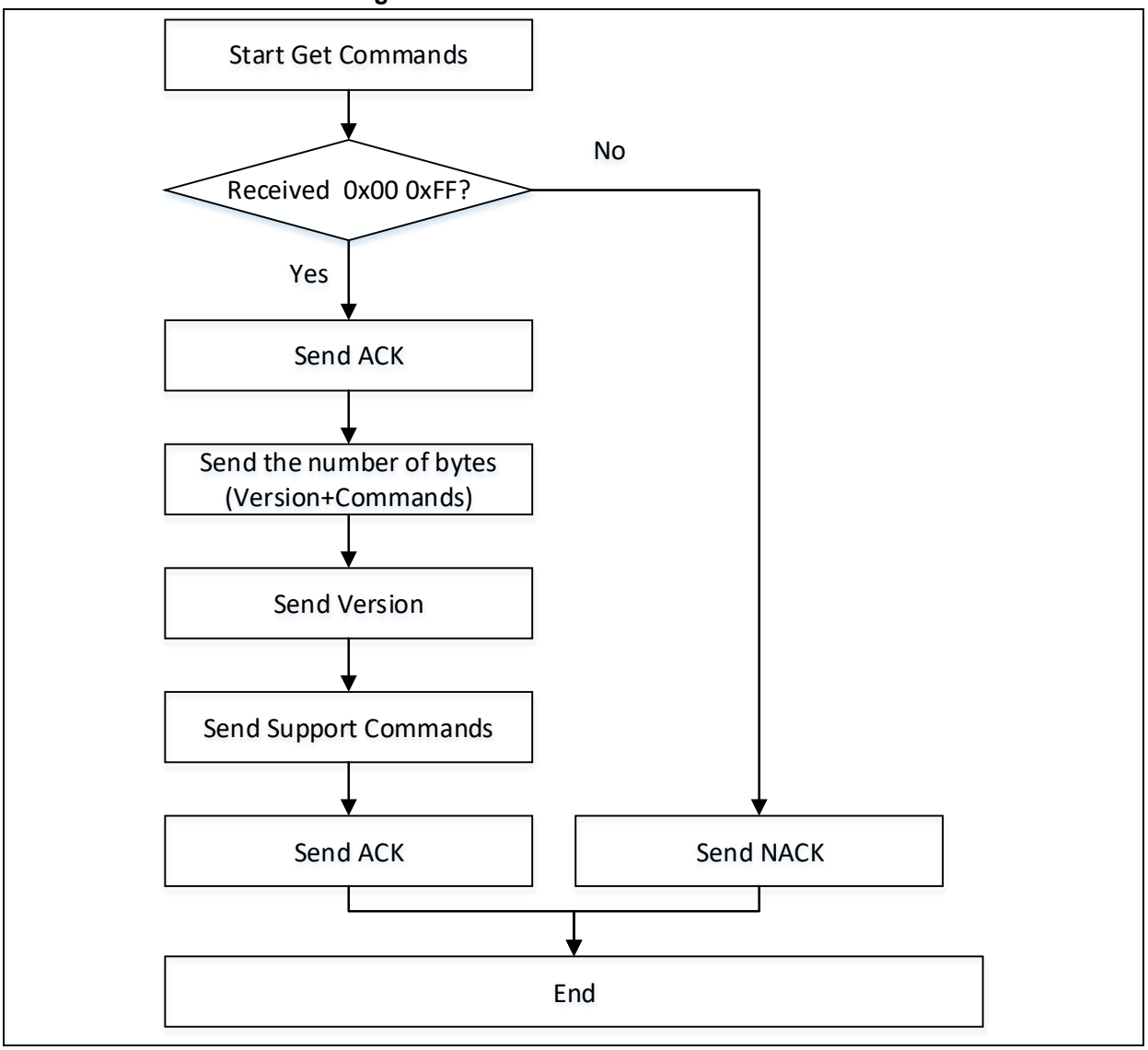


Figure 5 Get Commands: device side



4.2.2 Data transmit on host side

Send	Receive	Data	Description
1		0x00	
2		0xFF	
	1	ACK/NACK	When an NACK is received, it represents the end of the command.
	2	n	It represents the number of data to be received.
	3	*	The protocol version of the bootloader
	4	0x00	First command: Get Commands
	5	0x01	Second command: Get Version
	…	…	…
	n+2	*	N command
	n+3	ACK	End of command

4.3 Get Version command

This command is used to get the version of the bootloader. When the device receives the command, it transmits an ACK byte to the host. After the transmission of the ACK byte, the device sends the bootloader protocol version (1 byte) and the bootloader version code (BID, two bytes). After this operation, the device transmits the ACK byte to the host signaling the end of the command.

This command is allowed and executed when the access protection is enabled.

4.3.1 Flowchart on host and device sides

Figure 6 Get Version: host side

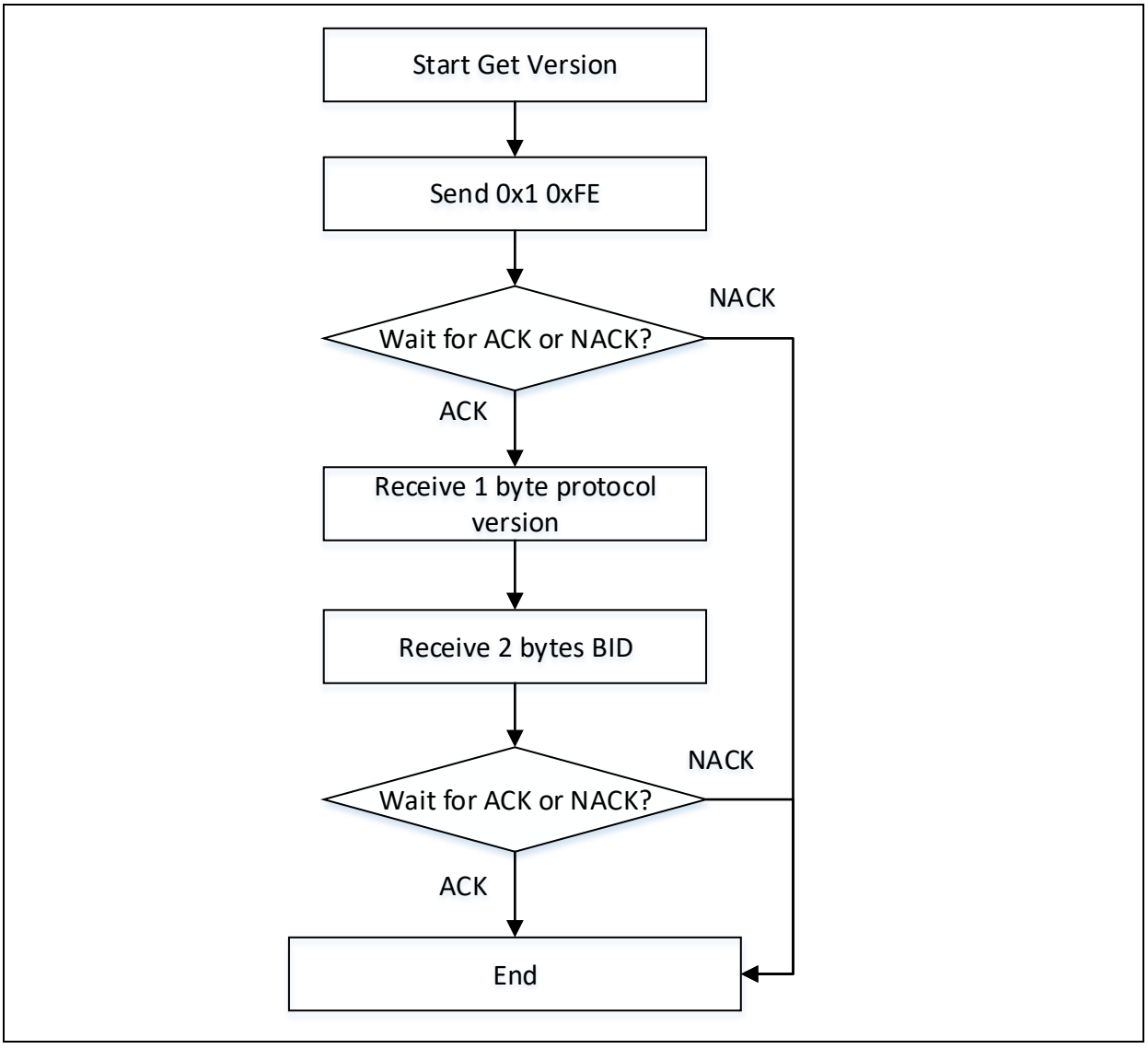
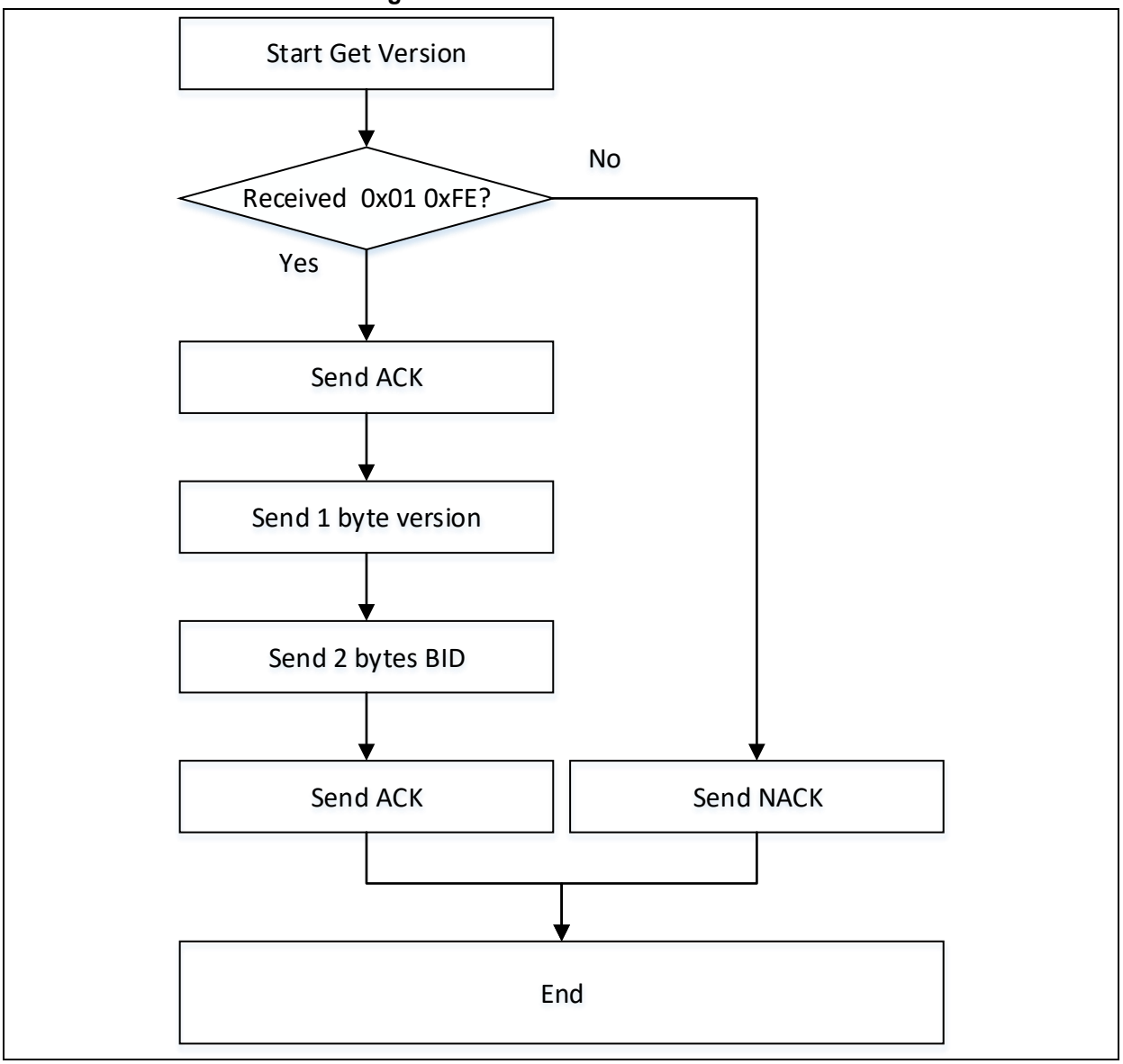


Figure 7 Get Version: device side



4.3.2 Data transmit on host side

Send	Receive	Data	Description
1		0x01	
2		0xFE	
	1	ACK/NACK	When an NACK is received, it represents the end of the command.
	2	*	Bootloader protocol version
	3	*	Bootloader version ID (BID）
	4	*	Bootloader version ID（BID）
	5	ACK	End of command

4.4 Get Device ID command

This command is used to get the identification code of the device. This ID data includes a 4-byte product ID and a 1-byte project ID. The “project ID” is used to identify which MCU series is being used, and the “product ID” is used to get the exact part number of the current MCU series in use. When the device receives this command, it transmits an “ACK” byte to the host. After the transmission of the ACK byte, the device sends a 1-byte data length indicating the number of data to be transmitted-- the value is the transmitted length minus 1 (for example, the transmitted data size is 5, then the value is 4), then sends another 4-byte product ID and 1-byte project ID. After this operation, the device transmits an ACK byte to the host.
This command is allowed and executed when the access protection is enabled.

4.4.1 Flowchart on host and device sides

Figure 8 Get Device ID: host side

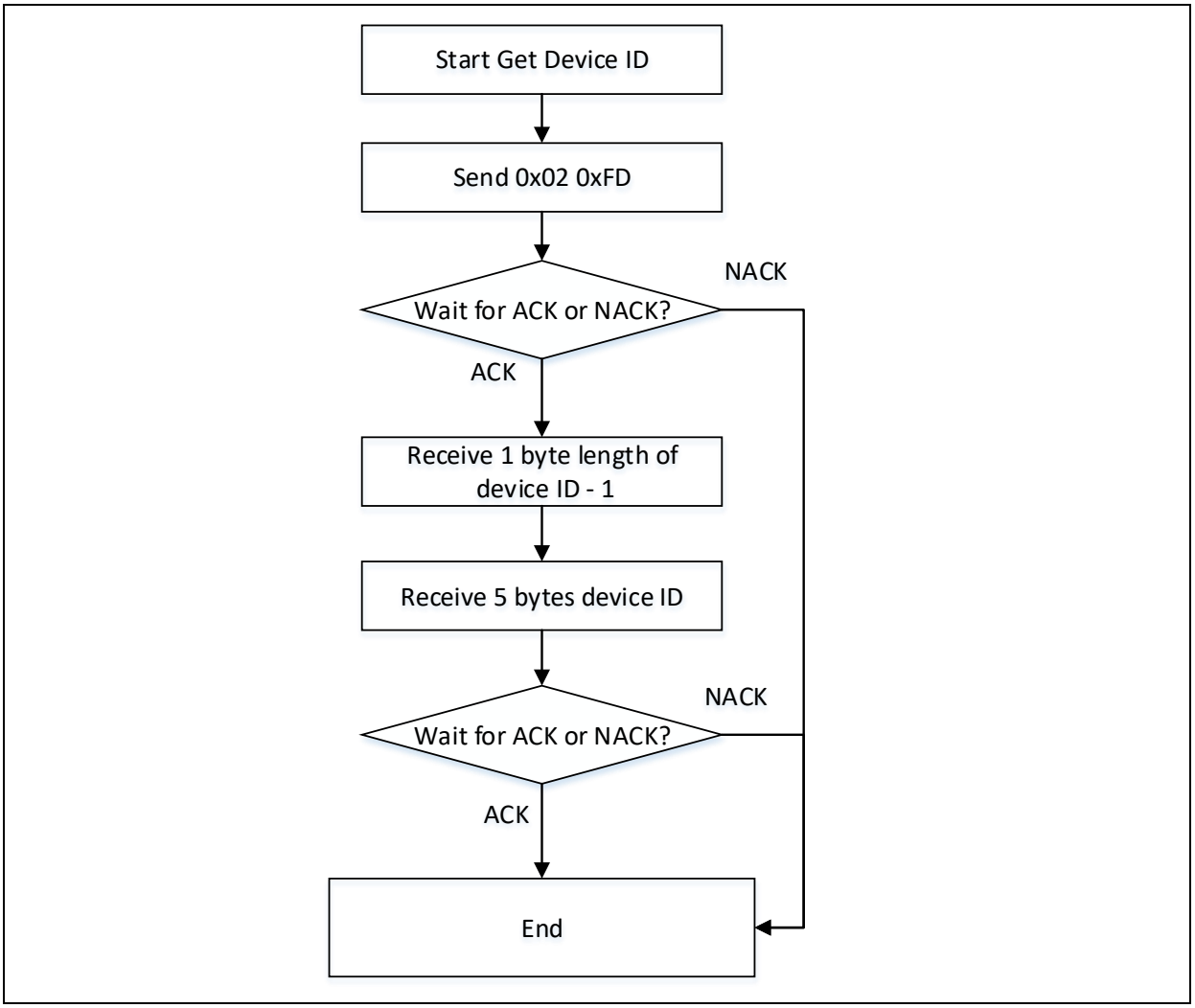
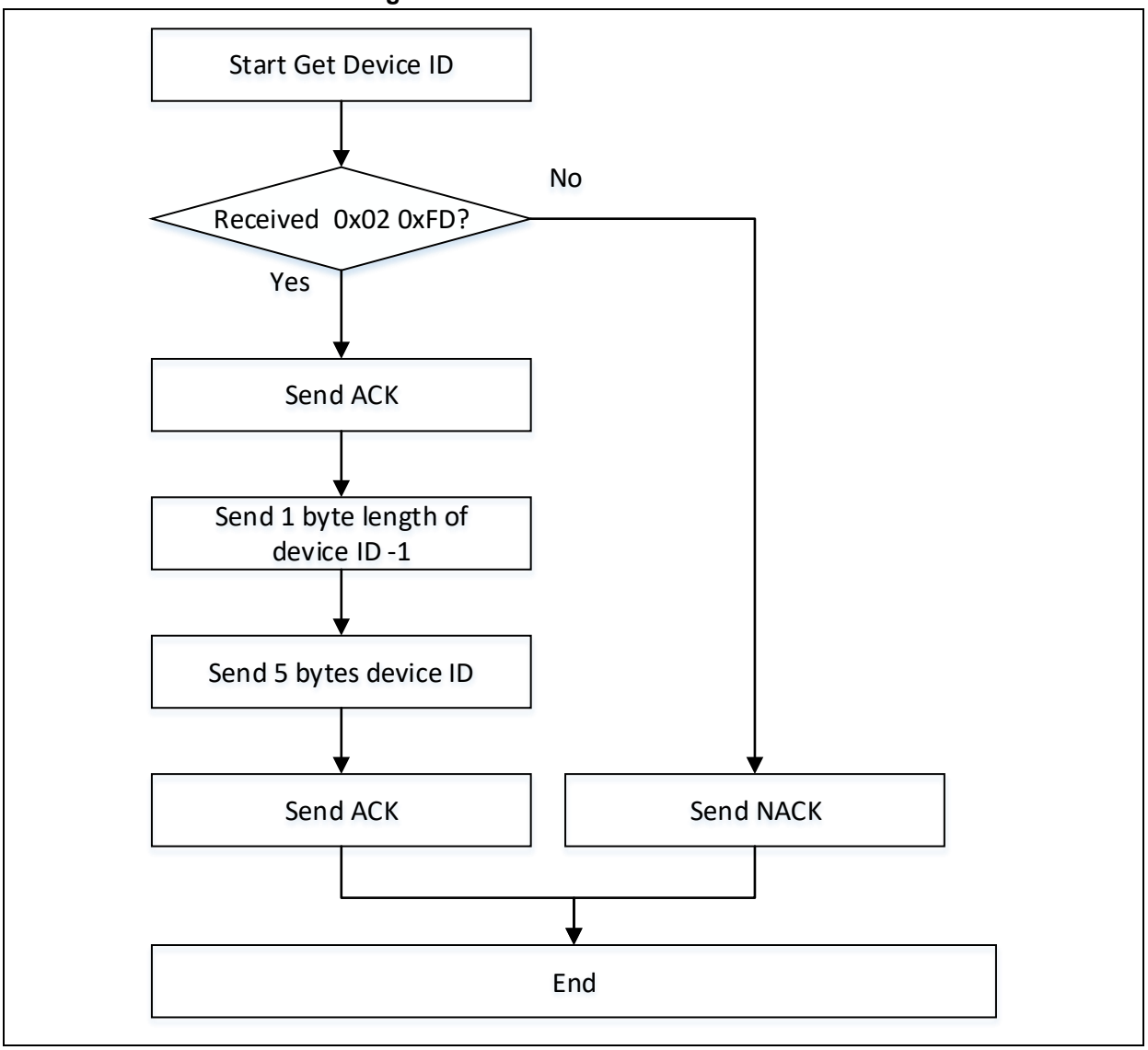


Figure 9 Get Device ID: device side



4.4.2 Data transmit on host side

Send	Receive	Data	Description
1		0x02	
2		0xFD	
	1	ACK/NACK	When an NACK is received, it represents the end of the command.
	2	0x04	Device ID length minus 1
	3	*	Product ID [8-15]
	4	*	Product ID [0-7]
	5	*	Product ID [24-31]
	6	*	Product ID [16-23]
	7	*	Project ID
	8	ACK	End of command

4.5 Read Memory command

This command is used to read data from any valid memory address (refer to the product datasheets) in SRAM, boot memory and in the user system data area. For those MCUs with SPIM feature, SPIM must be enabled first (send Enable SPIM command) to read data from it.
When the device receives the command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device waits for an address (four bytes) and a checksum byte. If the Checksum is correct and the valid is valid, the devices transmits an ACK byte, and then waits for the number of bytes to be read (1 byte) and for a checksum byte. The number of byte to read is the actual read data length -1(for example, to read 10 bytes, the value here is 9). If the checksum is correct, the devices sends an ACK before starting to send the needed data.
This command is not allowed and executed when the access protection is enabled.

4.5.1 Flowchart on host and device sides

Figure 10 Read Memory: host side

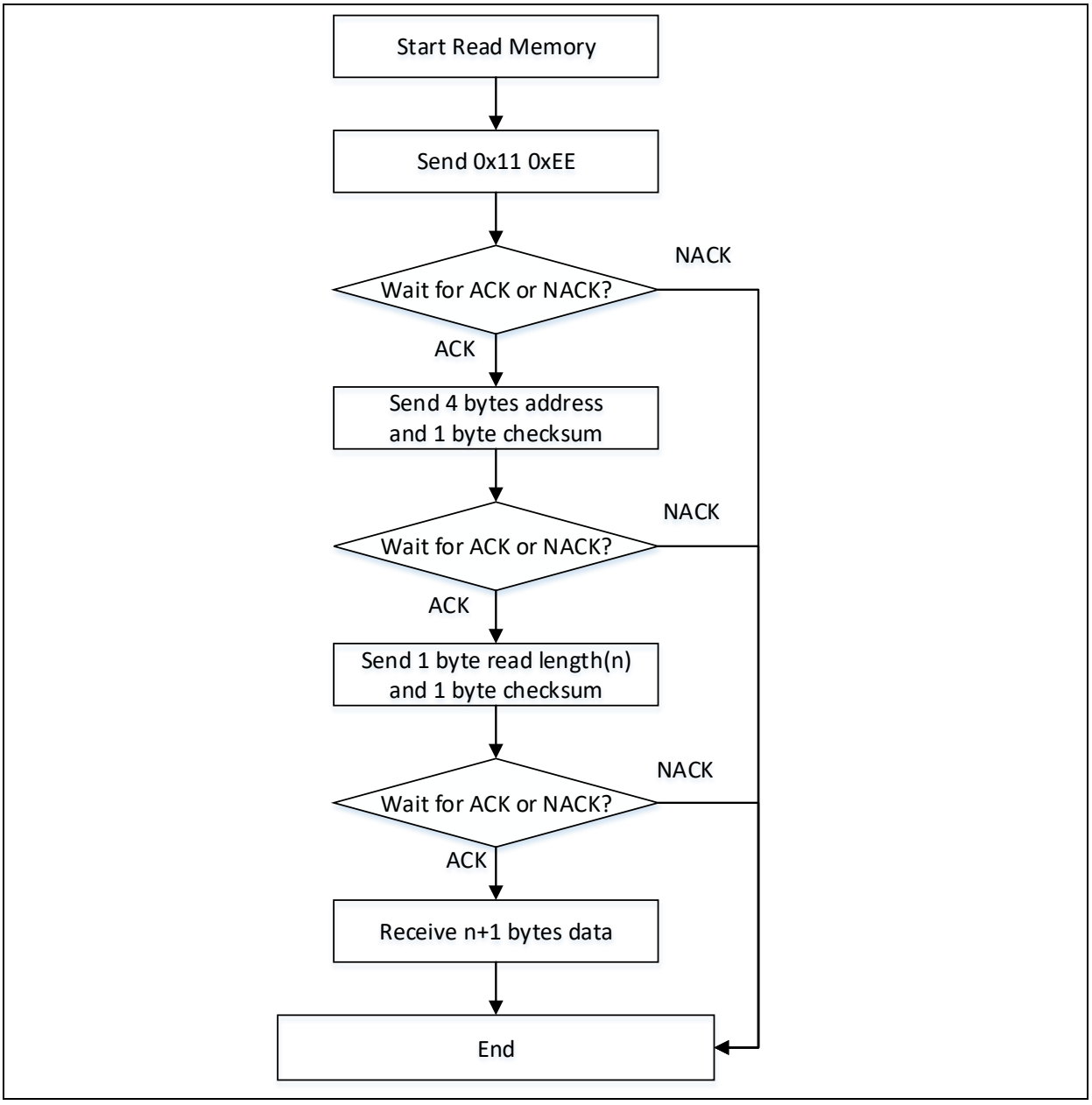
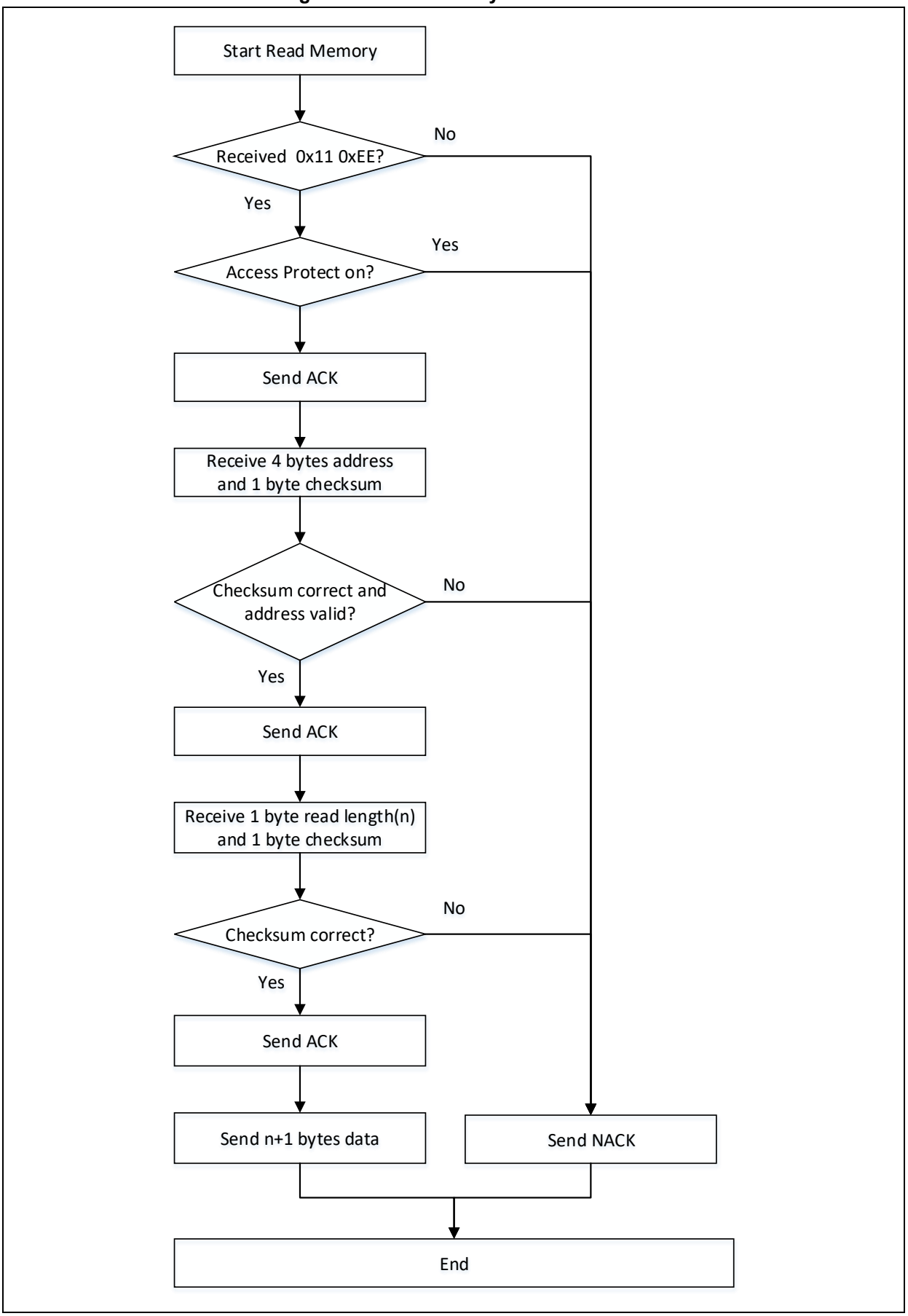


Figure 11 Read Memory: device side



4.5.2 Data transmit on host side

Send	Receive	Data	Description
1		0x11	
2		0xEE	
	1	ACK/NACK	When an NACK is received, it represents that the access protection is enabled, and the end of the command.
3		*	Address MSB
4		*	
5		*	
6		*	Address LSB
7		*	Checksum：XOR（address byte3~byte6）
	2	ACK/NACK	When an NACK is received, it represents the end of the command.
8		*	Read data length -1 (n)
9		*	Checksum：0xFF XOR byte8
	3	ACK/NACK	When an NACK is received, it represents the end of the command
	4	*	Destination address data
	…	…	Destination address data
	4+n+1	*	Destination address data

4.6 Jump command

This command is used to execute the code in a specified address. It can jump to main memory or SRAM to execute code. The address must be valid, which depends on the MCU series.
When the device receives the command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device waits for an address (four bytes) and a checksum byte, then it checks the received address. If the address is valid and the checksum is correct, the devices transmits an ACK byte, and jumps to the memory location.
This command is not allowed and executed when the access protection is enabled.

Note: the Jump command may not work for the AT32F421 series.

4.6.1 Flowchart on host and device sides

Figure 12 Jump: host side

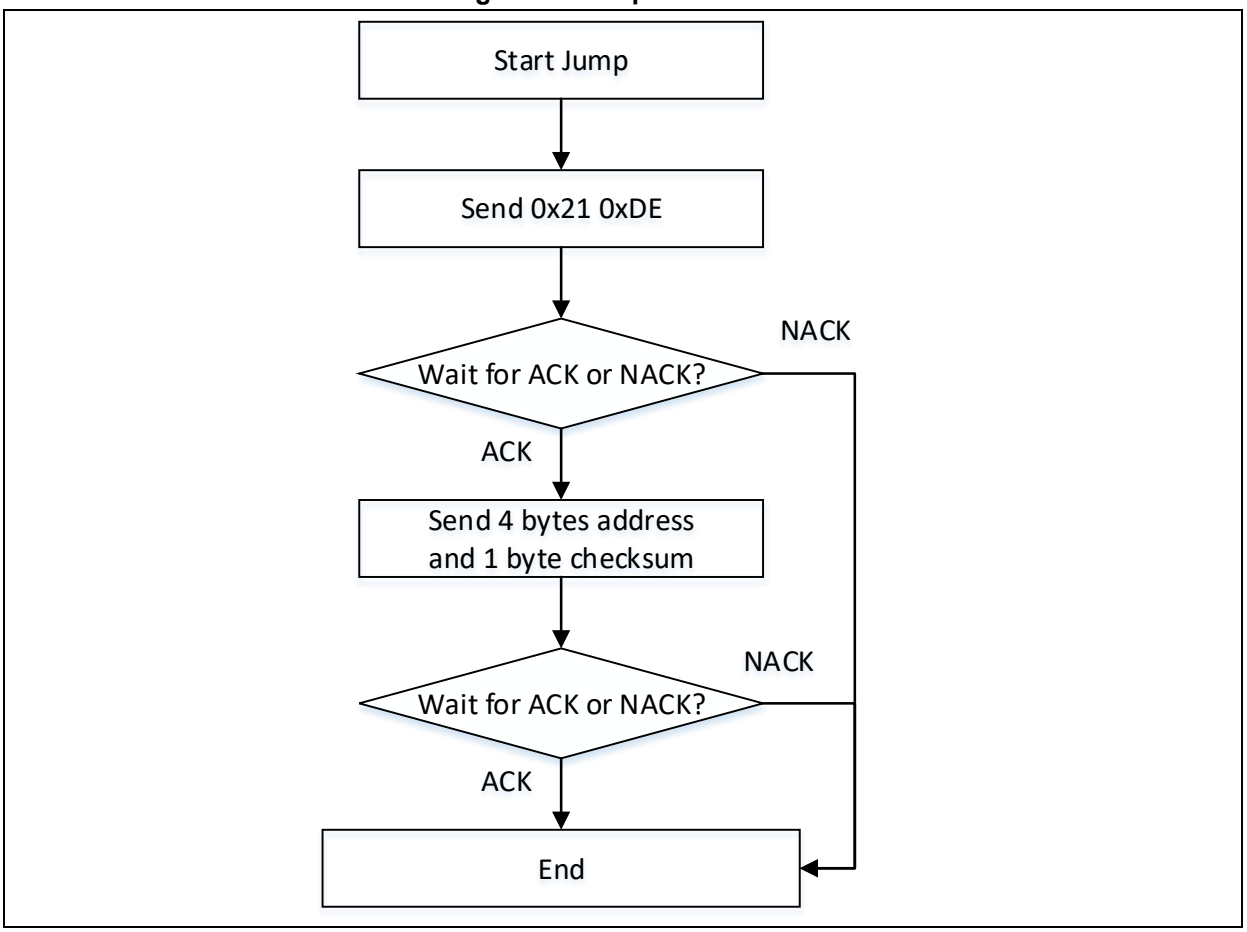
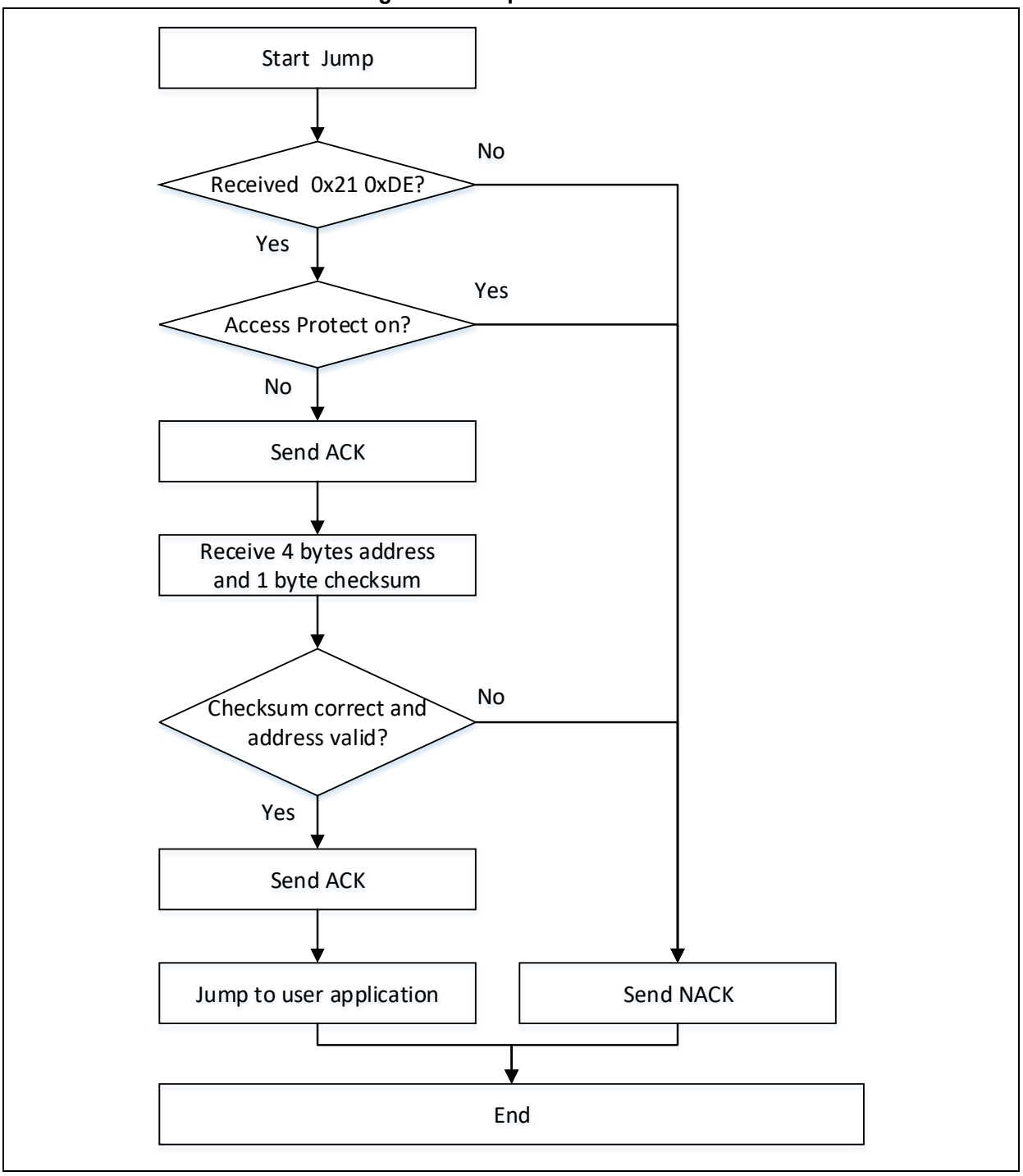


Figure 13 Jump: device side



4.6.2 Data transmit on host side

Send	Receive	Data	Description
1		0x21	Jump
2		0xDE	Jump
	1	ACK/NACK	When an NACK byte is received, it indicates that the access protection is enabled, and the end of the command.
3		*	Address MSB
4		*	
5		*	
6		*	Address LSB
7		*	Checksum：XOR（address byte3~byte6）
	2	ACK/NACK	

4.7 Write Memory command

This command is used to write data to any valid memory address (depending on the product series) such as Flash memory, SRAM, or user system data area.

If the Write Memory command is issued to the Flash memory, all bytes are erased before writing the new values.

The size of the user system data area depends on the product series:

- User system data area smaller than 256 bytes
  Only a single Write Memory command is issued by the host to write data to the user system data area. When the device receives the command, it erases all bytes in the user system data area before writing to it and then performing a system reset.
- User system data area greater than 256 bytes
  Multiple Write Memory commands are issued by the host to write data to the user system data area.
  When the device receives the command, if the received address to write is the start address of the user system data area, the address is erased before written.
  If it is not the start address of the user system data area, no erase operation is performed and data are directly written.
  After the write operation (after all of data are written), a system reset is performed automatically.

Note: For the AT32F435/AT32F437 series, no system reset is performed following the write operation, instead it asks the host to send a Reset command.

When the device receive the “Write Memory” command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device waits for an address (four bytes) and a checksum byte. If the received address is valid and the checksum is correct, the device transmits an ACK byte. After the transmission of the ACK byte, the device waits for an address (one byte) and a checksum byte. The value is the actual written length minus 1 (for example, to write 10 bytes, the value here is 9). If the checksum is correct, the device writes data to the received address and then transmits an ACK byte to the host.

This command is not allowed and executed when the access protection is enabled.

4.7.1 Flowchart on host and device sides

Figure 14 Write Memory: host side

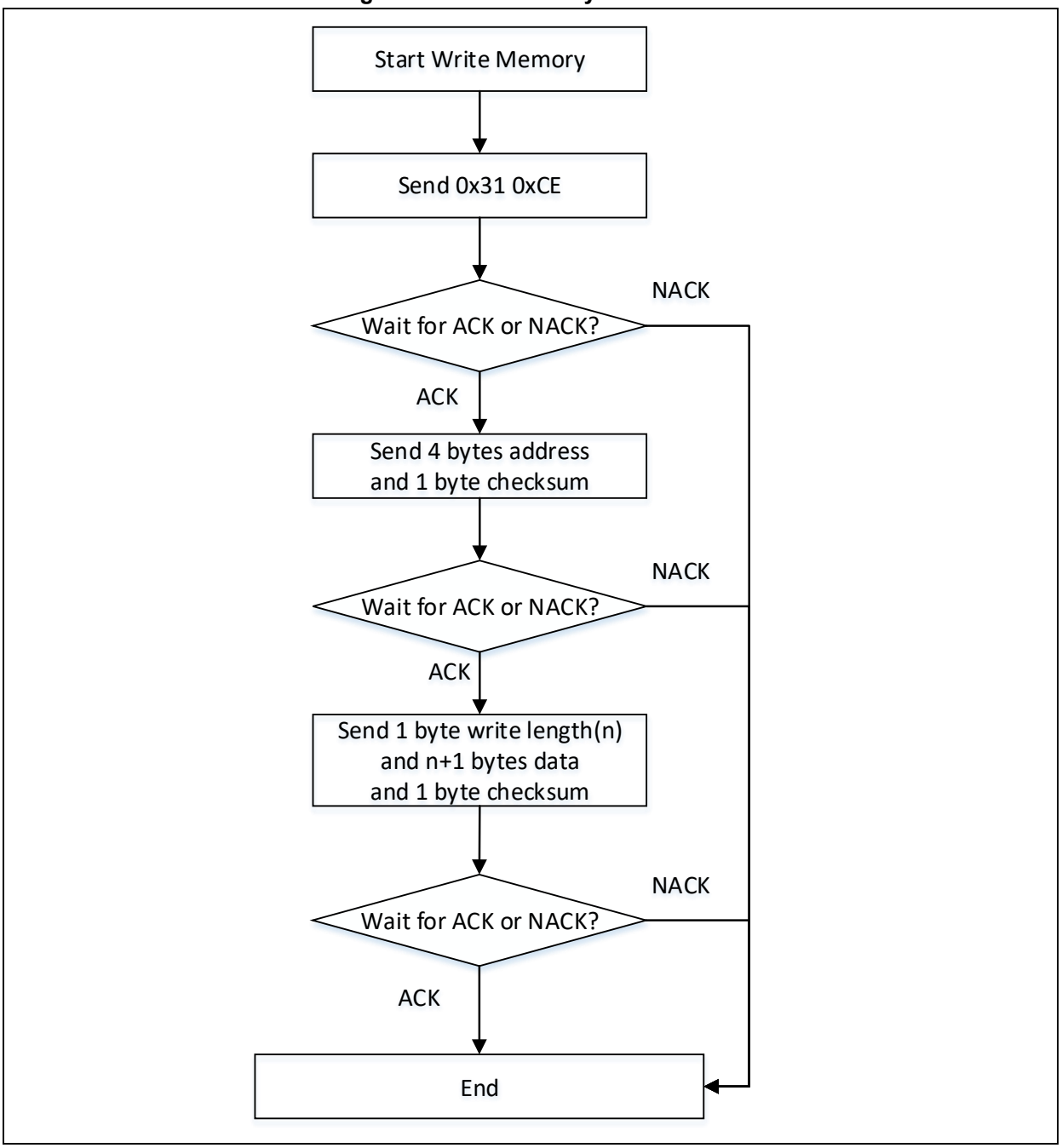
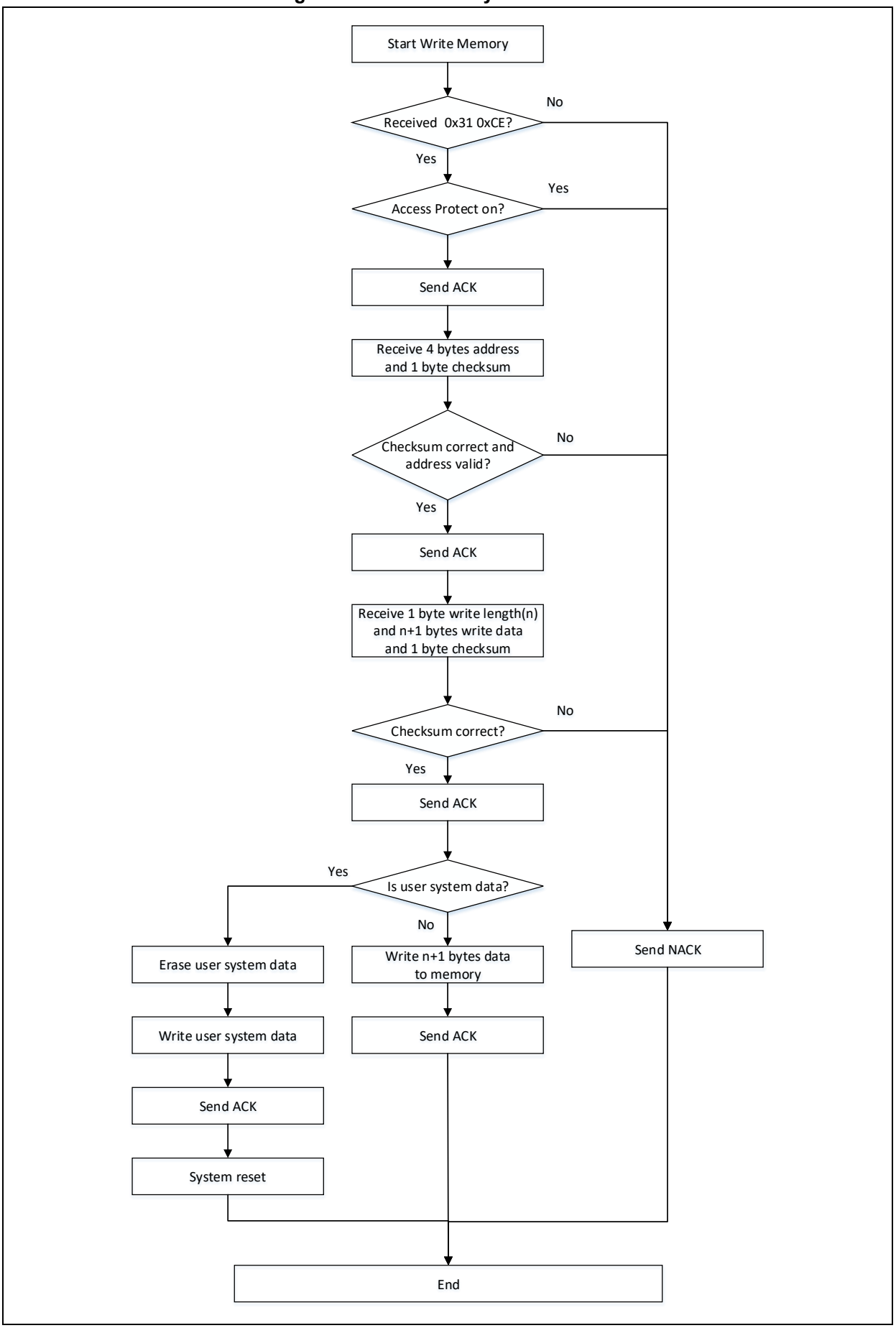


Figure 15 Write Memory: device side



4.7.2 Data transmit on host side

Send	Transmit	Data	Description
1		0x31	Write Memory
2		0xCE	Write Memory
	1	ACK/NACK	When an NACK byte is received, it indicates the access protection is enabled, and the end of the command.
3		*	Address MSB
4		*	
5		*	
6		*	Address LSB
7		*	Checksum：XOR（address byte3~byte6）
	2	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
8		*	The number of data to be written n – 1
9		*	Write data
…		…	Write data
9+n		*	Write data
			Checksum: XOR byte8 - byte9+n
	3	ACK/NACK	End of command

4.8 Erase command

This command allows the host to erase Flash memory pages. It supports sector erase (sector size depends on the product series) and mass erase.

For those products that contain bank 1 and bank 2, this command can be used to erase bank 1 and bank 2.

For those with SPIM feature, it support erase operation on Bank 3.

Note that this command is used to erase any valid memory address, depending on the product series.

In addition, for AT32F435 / AT32F437 series, it supports block erase operation.

When the device receives the command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the devices receives two bytes data and check which erase operation is performed based on the received data type.

- If it is Sector erase, the received data indicates the number of the sectors to be erased. Then the device receives the index of N sectors to be erased. After receiving the index, the device starts erasing the corresponding sector and then transmits an ACK byte to the host.
- If it is mass erase or bank erase, the device directly performs erase operation and then transmits the ACK byte to the host after the erase operation.
- If it is block erase, the device receives Block start address (four bytes) and a checksum byte. If the checksum is correct, the device starts erasing block. After the erase operation, the device then transmits the ACK byte to the host. For the AT32F435 and AT32F437 series that support block erase, each block size is 64KB.

This command is not allowed and executed when the access protection is enabled.

Table 1 List of erase type

Erase type	Erase index	Erase area
Sector erase	0x00 0x00	Sector0
	0x00 0x01	Sector1
	…	…
	0x80 00	Bank3 Sector0
	0x80 01	Bank3 Sector1
	…	…
	0x8F 0xFF	Bank3 Sector4096
Mass erase	0xFF 0xFF	All Flash
Bank erase	0xFF 0xFE	Bank1 Erase
	0xFF 0xFD	Bank2 Erase
	0xFF 0xFC	Bank3 Erase
Block erase	0xFF 0xFB	Block Erase

Note: the sector size is dependent on the product series.

4.8.1 Flowchart on host and device sides

Figure 16 Erase: host side

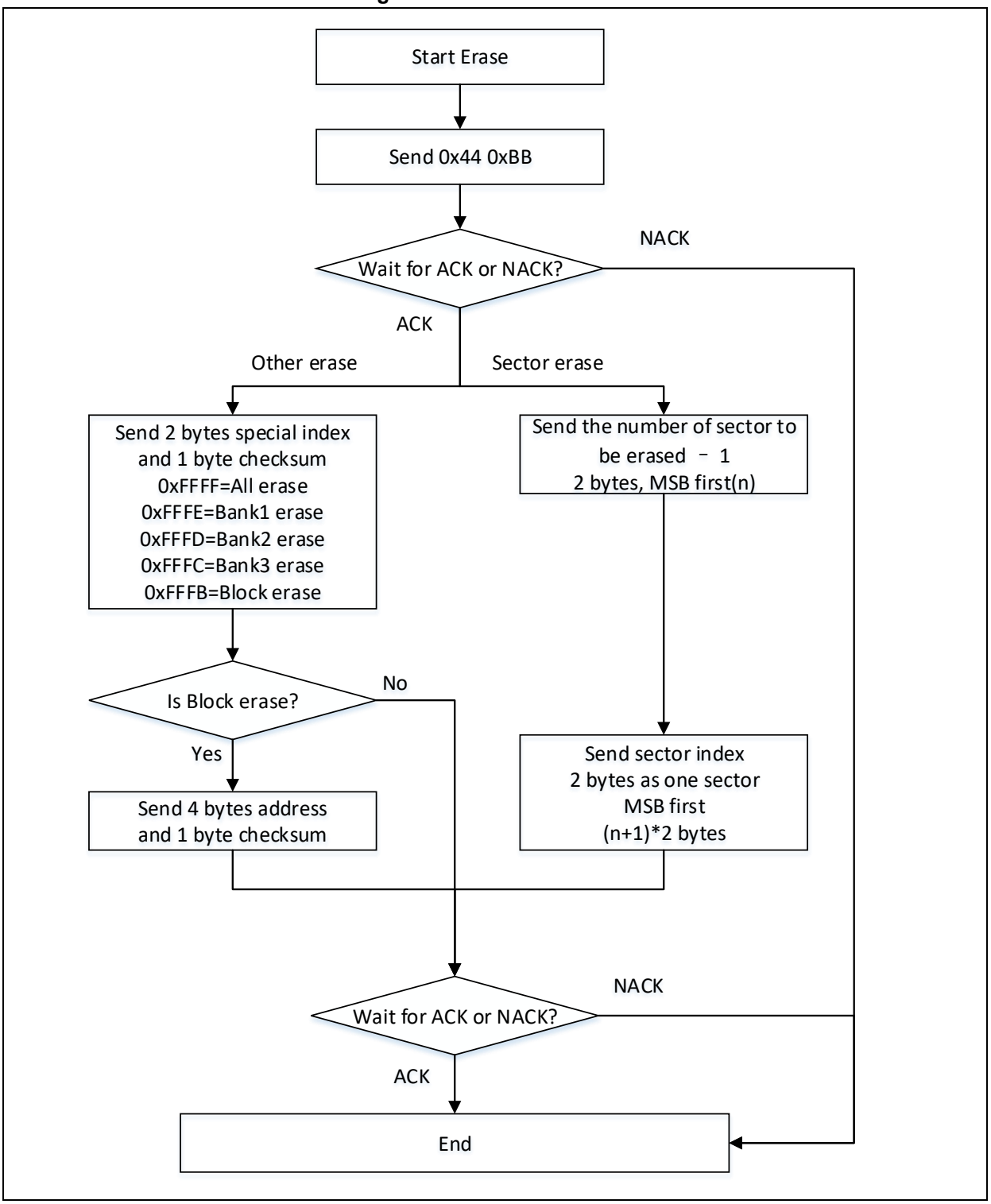
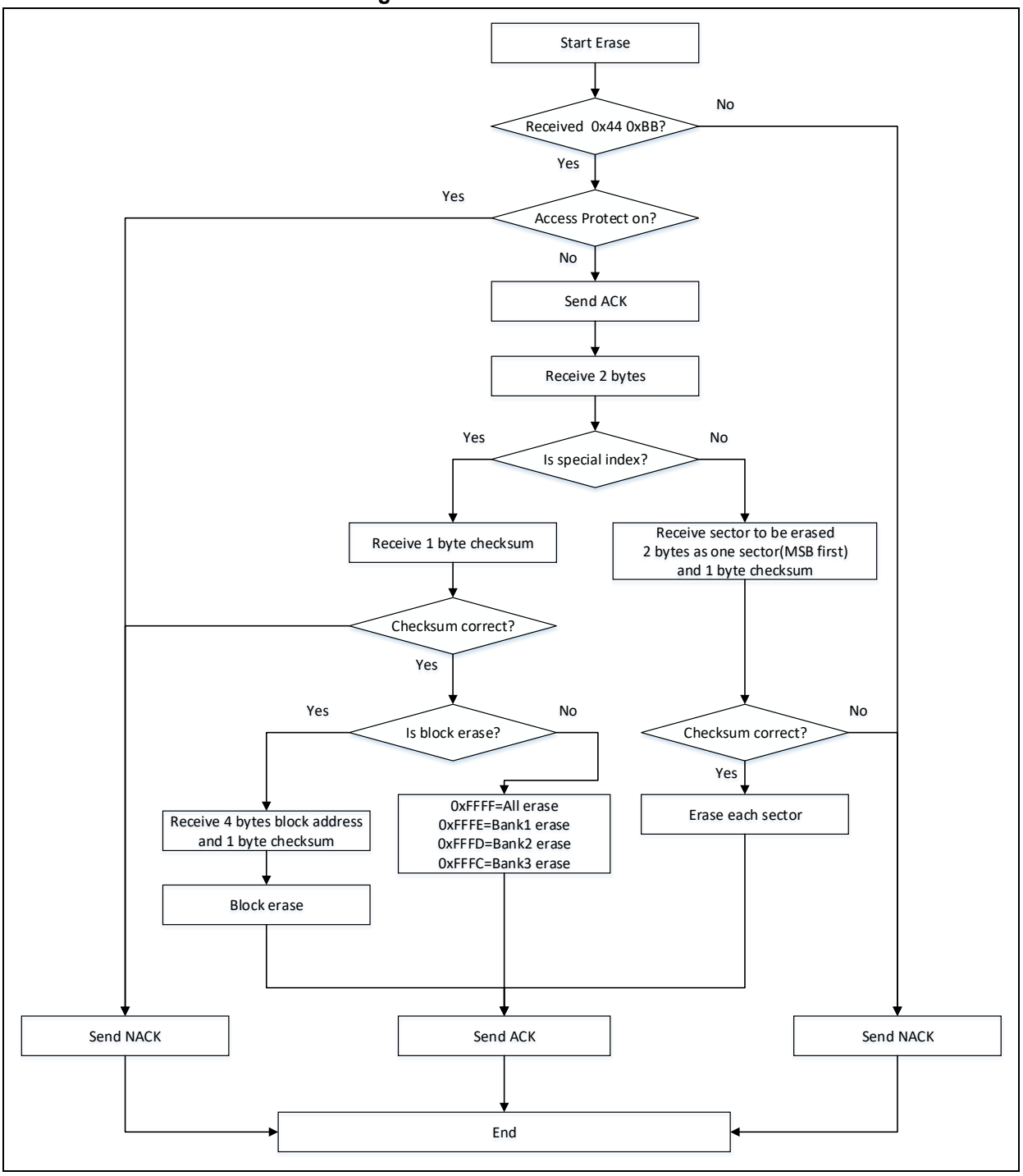


Figure 17 Erase: device side



4.8.2 Data transmit on host side

Sector erase

Send	Receive	Data	Description
1		0x44	Erase
2		0xBB	Erase
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Erase sector number – 1 (n) MSB
4		*	Erase sector number– 1 (n) LSB
5		*	First Sector index (MSB)
6		*	First Sector index (LSB)
…		*	X Sector index (MSB)
…		*	X Sector index (LSB)
7+2(n+1)		*	Checksum：XOR（byte 3~6+2(n+1)）
	2	ACK	

Block erase

Send	Receive	Data	Description
1		0x44	Erase
2		0xBB	Erase
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		0xFF	Block erase
4		0xFB	Block erase
5		*	Checksum：XOR (byte 3~4)
6		*	Block address (MSB)
7		*	Block address
8		*	Block address
9		*	Block address (LSB)
10		*	Checksum: XOR (byte 6~9)
	2	ACK	

Bank or mass erase

Send	Receive	Data	Description
1		0x44	Erase
2		0xBB	Erase
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		0xFF	Mass erase or Bank erase index (MSB)
4		*	Mass erase (0xFF) or Bank erase (0xFE, 0xFD, 0xFC) index (LSB)
5		*	Checksum: XOR (byte 3~4)
	2	ACK	

4.9 Erase and Program Protect command

This command is used to enable erase/program protection for the specified sectors.

When the device receives the command, if the access protection is not enabled, it transmits an ACK byte to the host. After the transmission of the ACK byte, it waits for one-byte length-1 (n), and then receives the index bit corresponding to the erase/program protection (n+1 bytes) ---refer to the corresponding product datasheet for more details about the bit definition --- and a checksum byte. If the checksum is correct, the device erases the user system data area (those data other than erase/program protect bytes are reserved), and writes the settings for erase/program protection, and transmits the ACK byte to the host before performing a system reset.

Note: The index bit (0, 1, 2…n) for erase/program protection corresponds to the bit (0-N) of the erase/program protection byte of the user system data area. See the corresponding product sheet for more details about the user system data area.

This command is not allowed and executed when the access protection is enabled.

4.9.1 Flowchart on host and device sides

Figure 18 Erase and Program Protect: host side

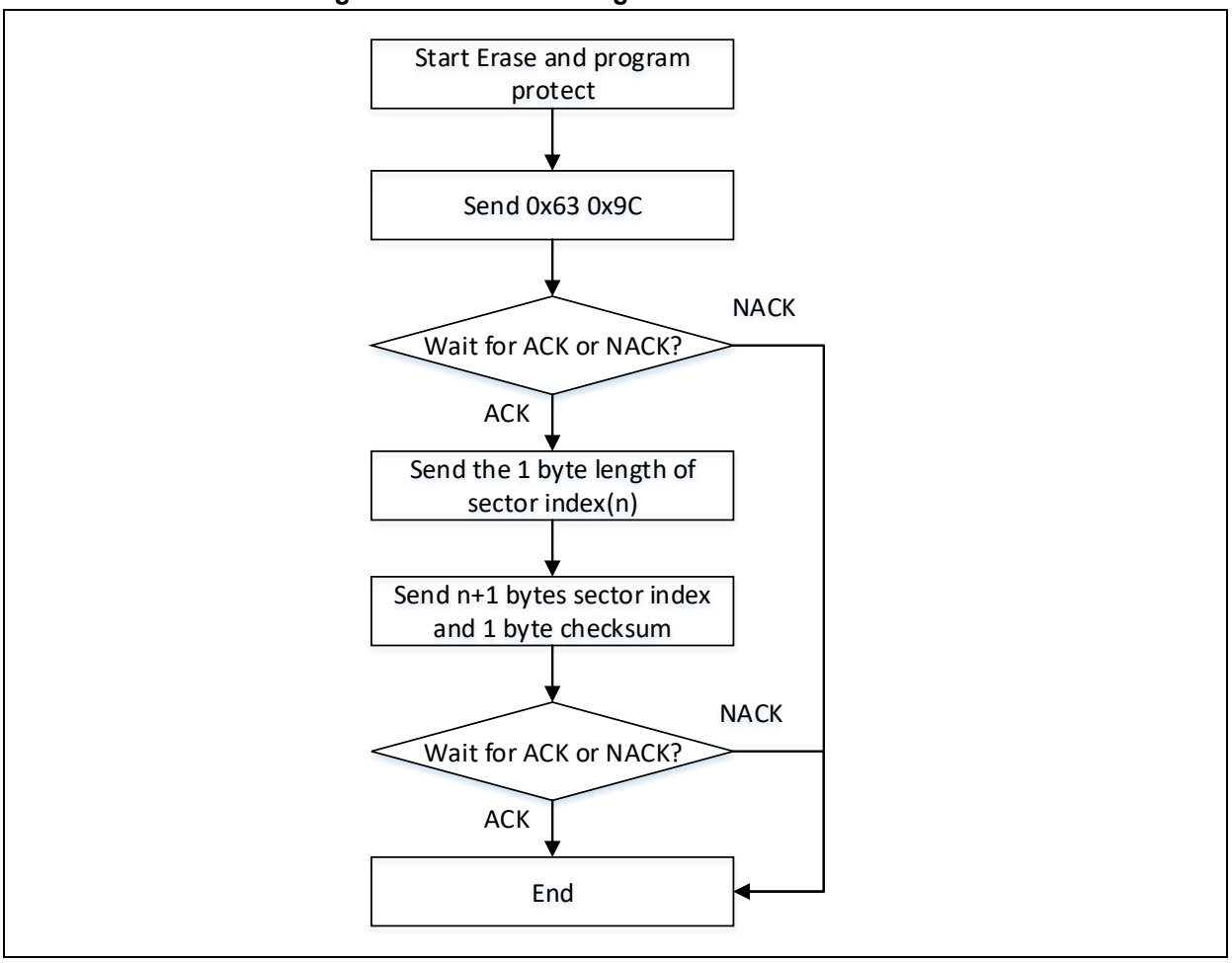
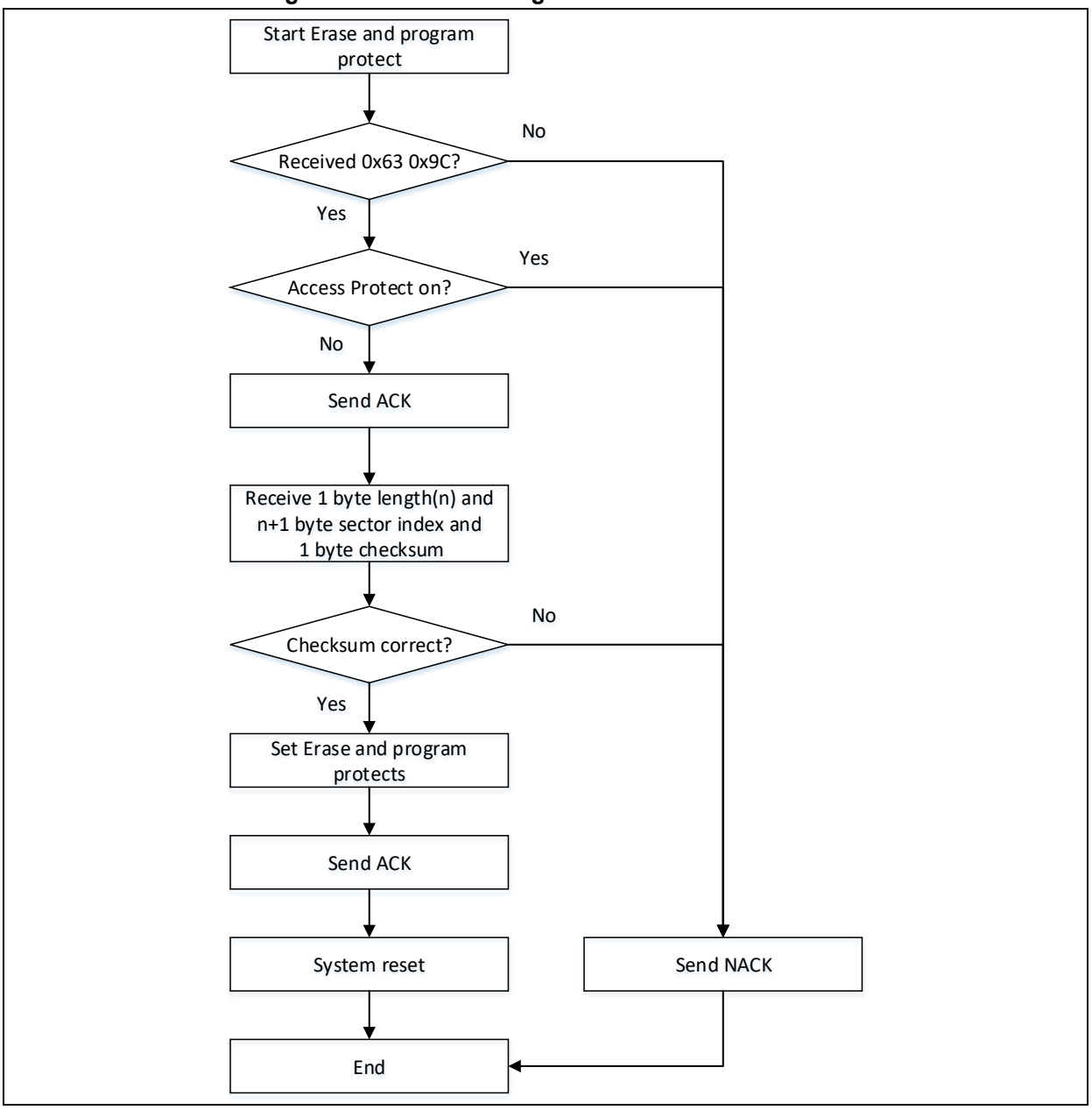


Figure 19 Erase and Program Protect: device side



4.9.2 Data transmit on host side

Send	Receive	Data	Description
1		0x63	Erase and program protect
2		0x9C	Erase and program protect
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Protection byte-1 (n)
…		…	The index to be protected
5+n+1		*	Checksum：XOR（byte 3~4+n+1）
	2	ACK	

4.10 Erase and Program Unprotect

This command is used to disable the erase/program protection of memory.
When the device receives the command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device disables the erase/program protection of all the flash memory sectors. After the unprotection operation, the device transmits the ACK byte and performs a system reset.
This command is not allowed and executed when the access protection is enabled.

4.10.1 Flowchart on host and device sides

Figure 20 Erase and Program Unprotect: host side

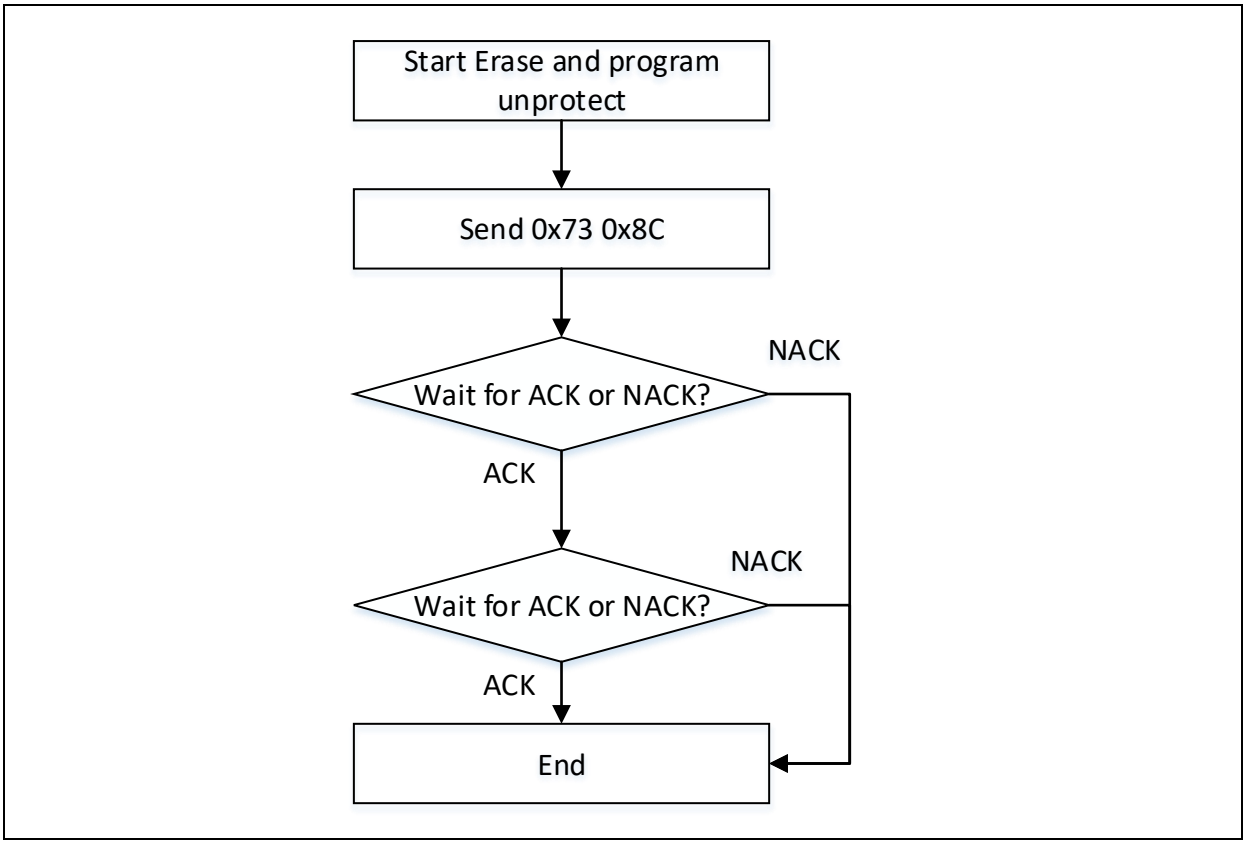
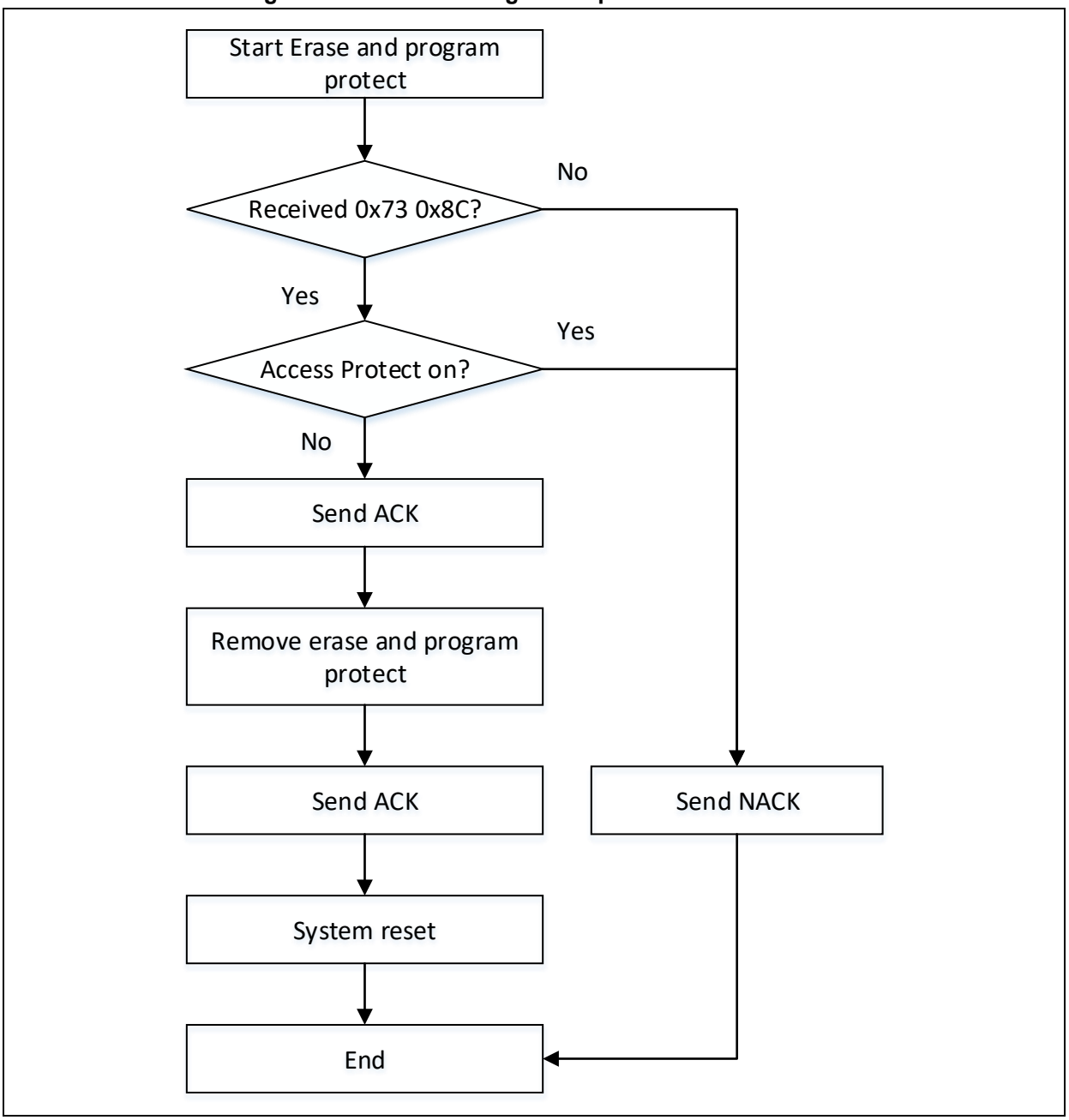


Figure 21 Erase and Program Unprotect: device side



4.10.2 Data transmit on host side

Send	Receive	Data	Description
1		0x73	Erase and program unprotect
2		0x8C	Erase and program unprotect
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	2	ACK	

4.11 Access Protect command

This command is used to enable the flash memory access protection. After this command is enabled, it is not allowed to read data from flash memory.

When the device receives the command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device enables the access protection for the flash memory. After the protection operation, the device transmits the ACK byte to the host and then performs a system reset.

This command is not allowed and executed when the access protection is enabled.

4.11.1 Flowchart on host and device sides

Figure 22 Access Protect: host side

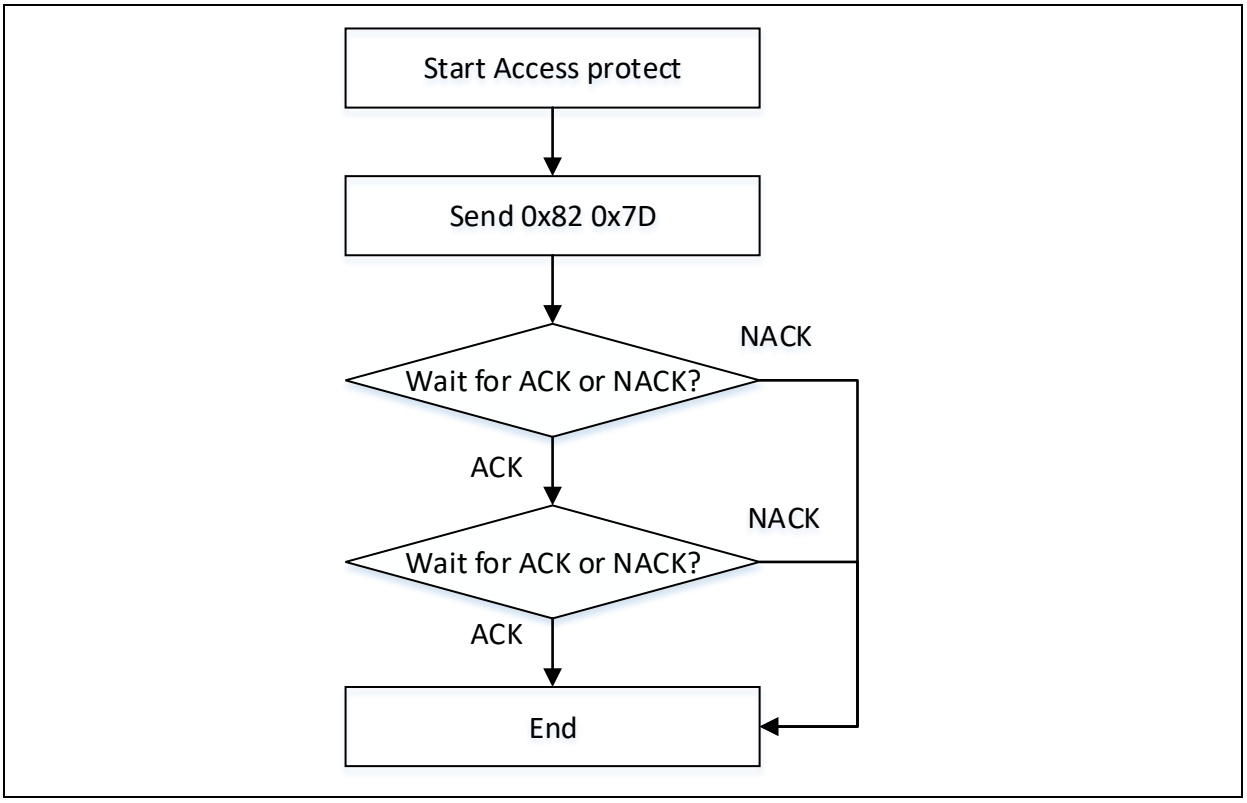
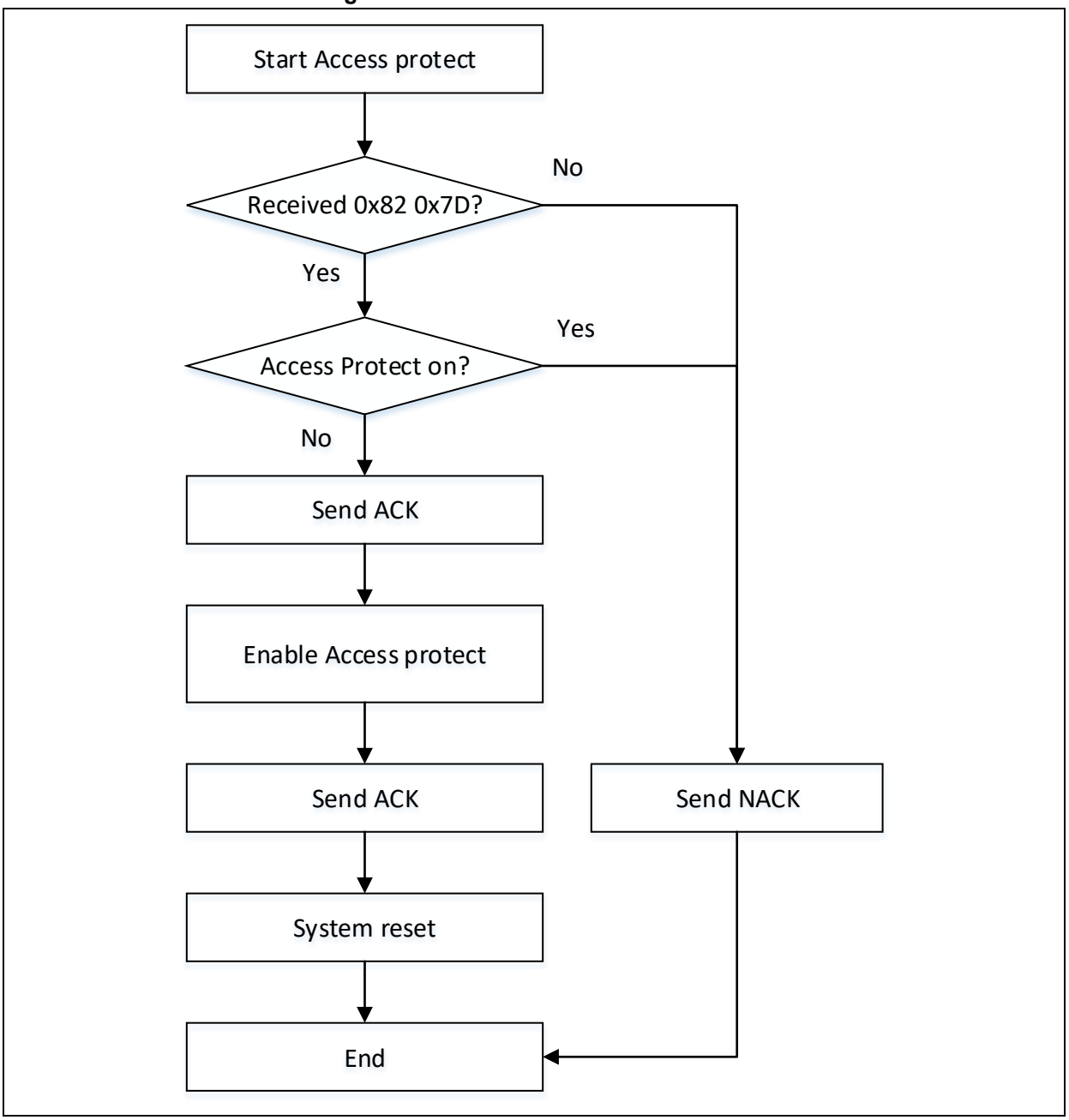


Figure 23 Access Protect: device side



4.11.2 Data transmit on host side

Send	Receive	Data	Description
1		0x82	Access protect
2		0x7D	Access protect
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	2	ACK	

4.12 Access Unprotect command

This command is used to disable the flash memory access protection. If the device is in Access protection mode, this command operation induces a mass erase of the flash memory.
When the device receives the command, it transmits the ACK byte to the host. After the transmission of the ACK byte, the devices disables the access protection of the flash memory. After the unprotection operation, the device transmits the ACK byte to the host and then performs a system reset.

4.12.1 Flowchart on host and device sides

Figure 24 Access Unprotect: host side

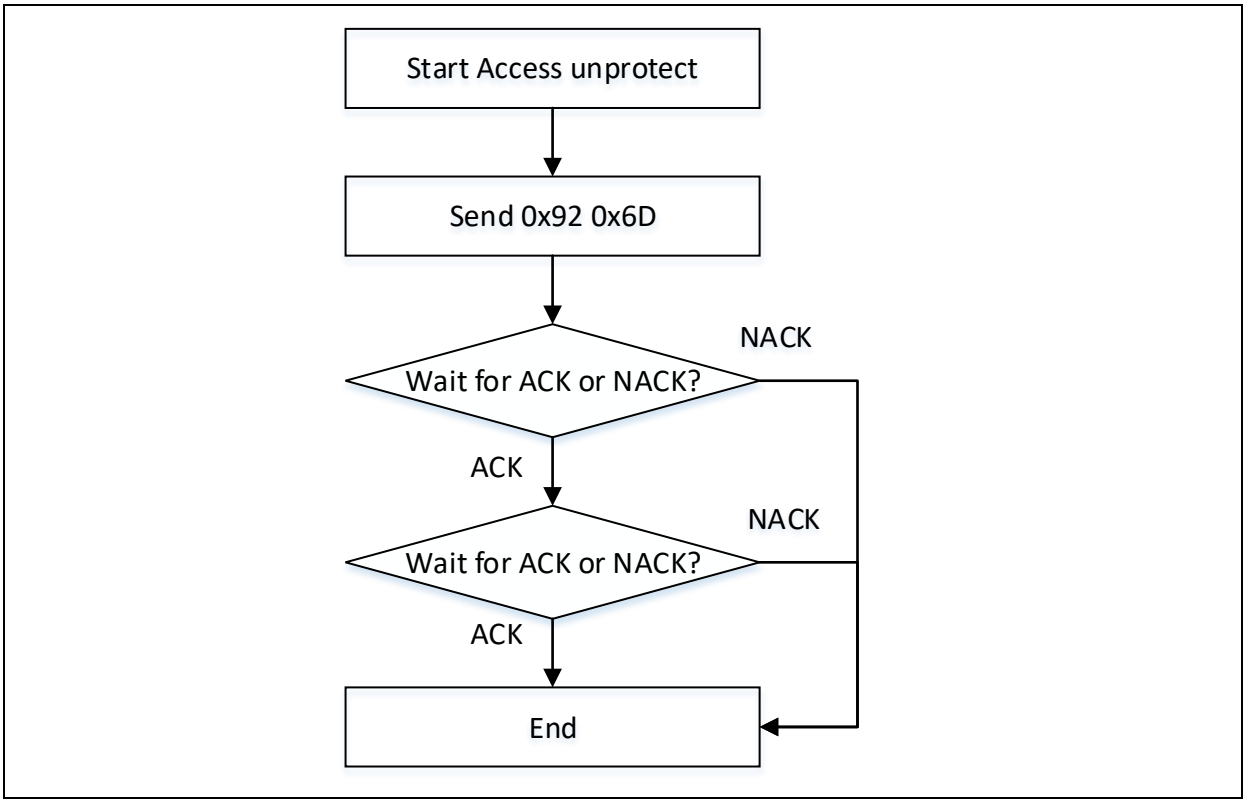
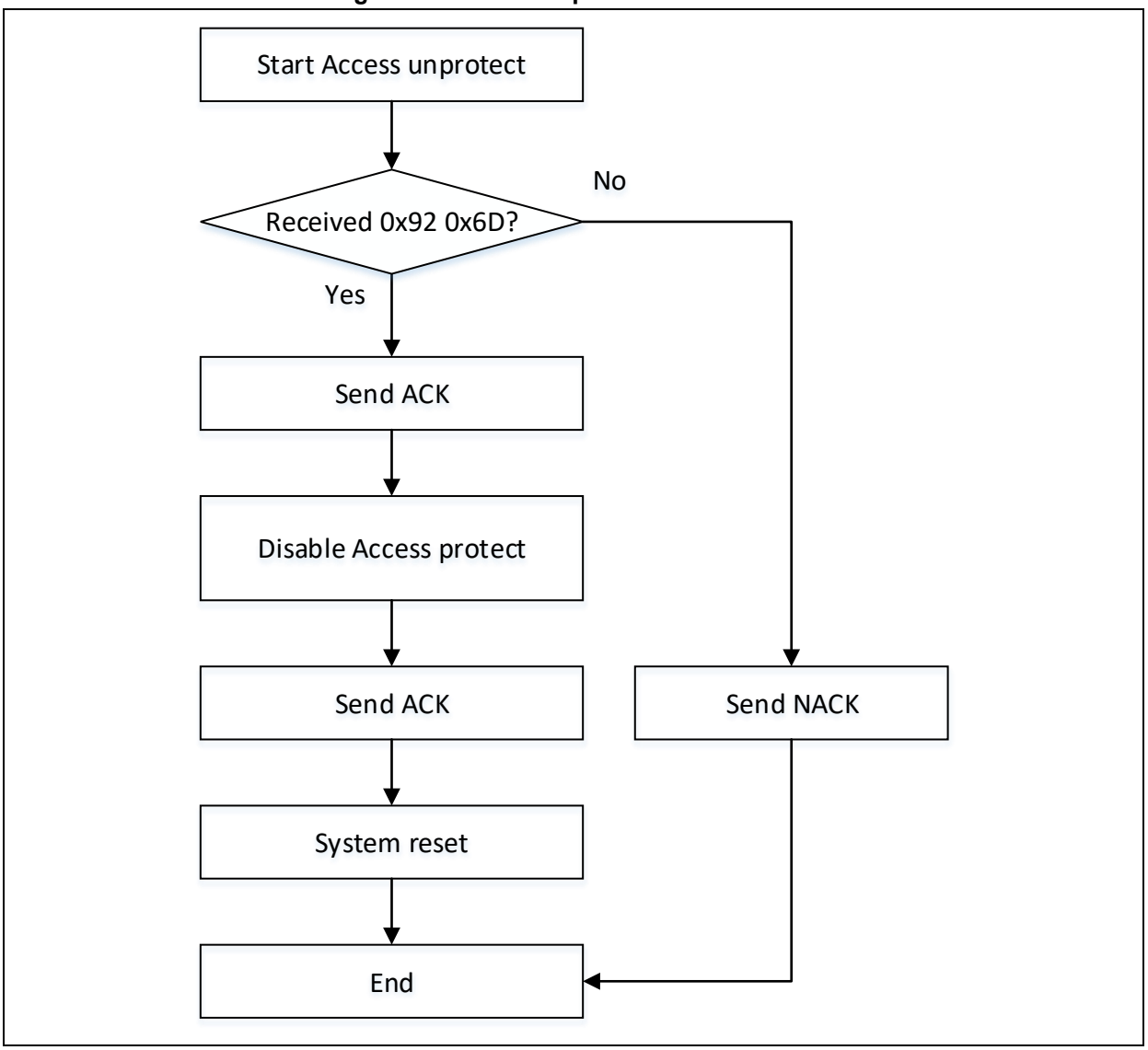


Figure 25 Access Unprotect: device side



4.12.2 Data transmit on host side

Send	Receive	Data	Description
1		0x92	Access unprotect
2		0x6D	Access unprotect
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	2	ACK	

4.13 Firmware CRC command

This command is used to check whether memory data is correct. The memory area is specified by the host but must be expressed in sector unit and its start address must be aligned with a sector. When the device receives the command, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device waist for the start address (four bytes) and a checksum byte. If the received address and the checksum are all valid, the device transmits the ACK byte to the host. Then the device waits for the number of sectors (two bytes) to be computed -1 (n) and a checksum byte. If the checksum is correct, the device transmits an ACK byte to the host and starts computing CRC value. After the computing operation, the device sends back a four-byte CRC value to the host.

The Firmware CRC adopts the MPEG-2 CRC algorithm.

4.13.1 Flowchart on host and device sides

Figure 26 Firmware CRC: host side

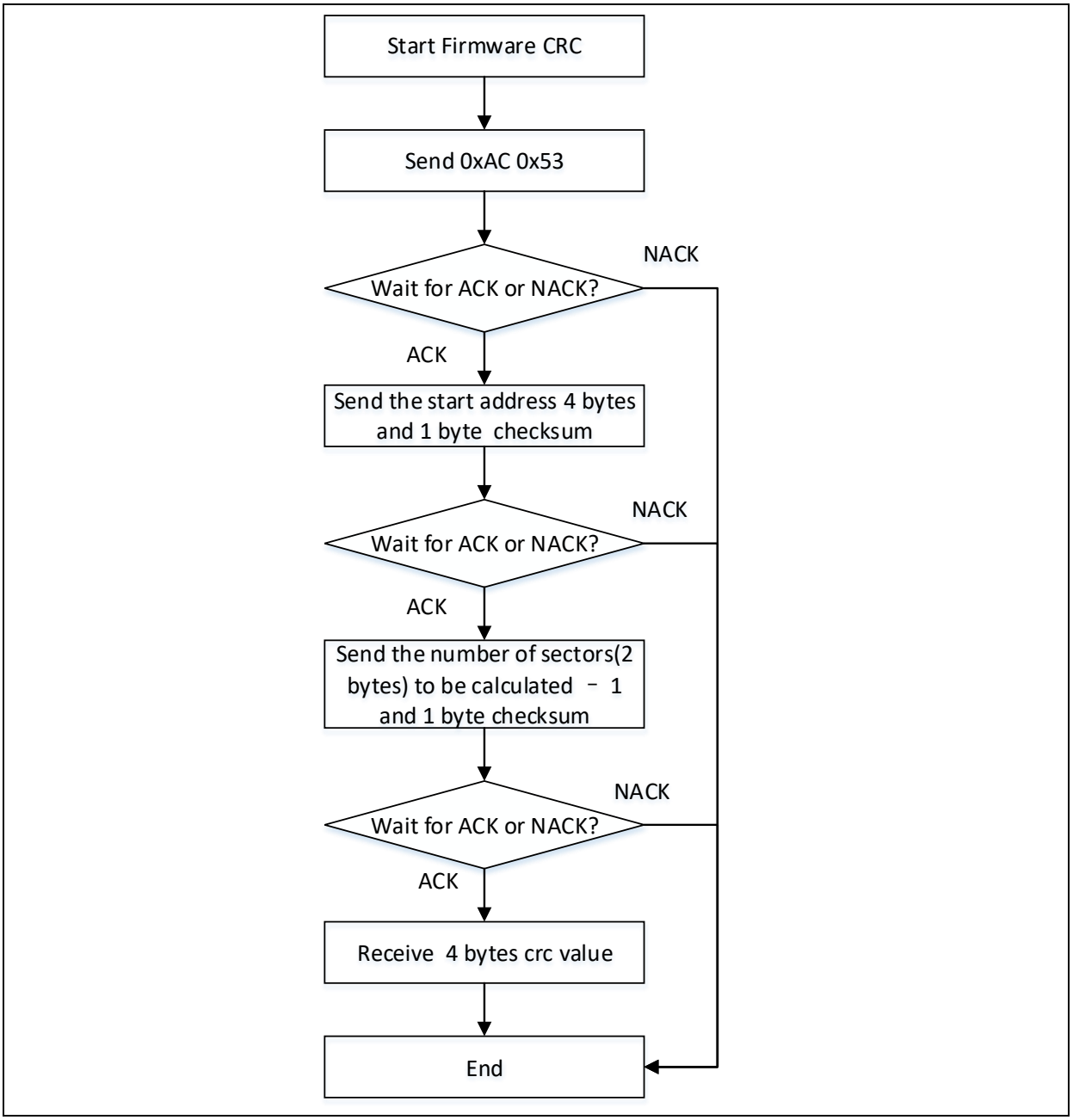
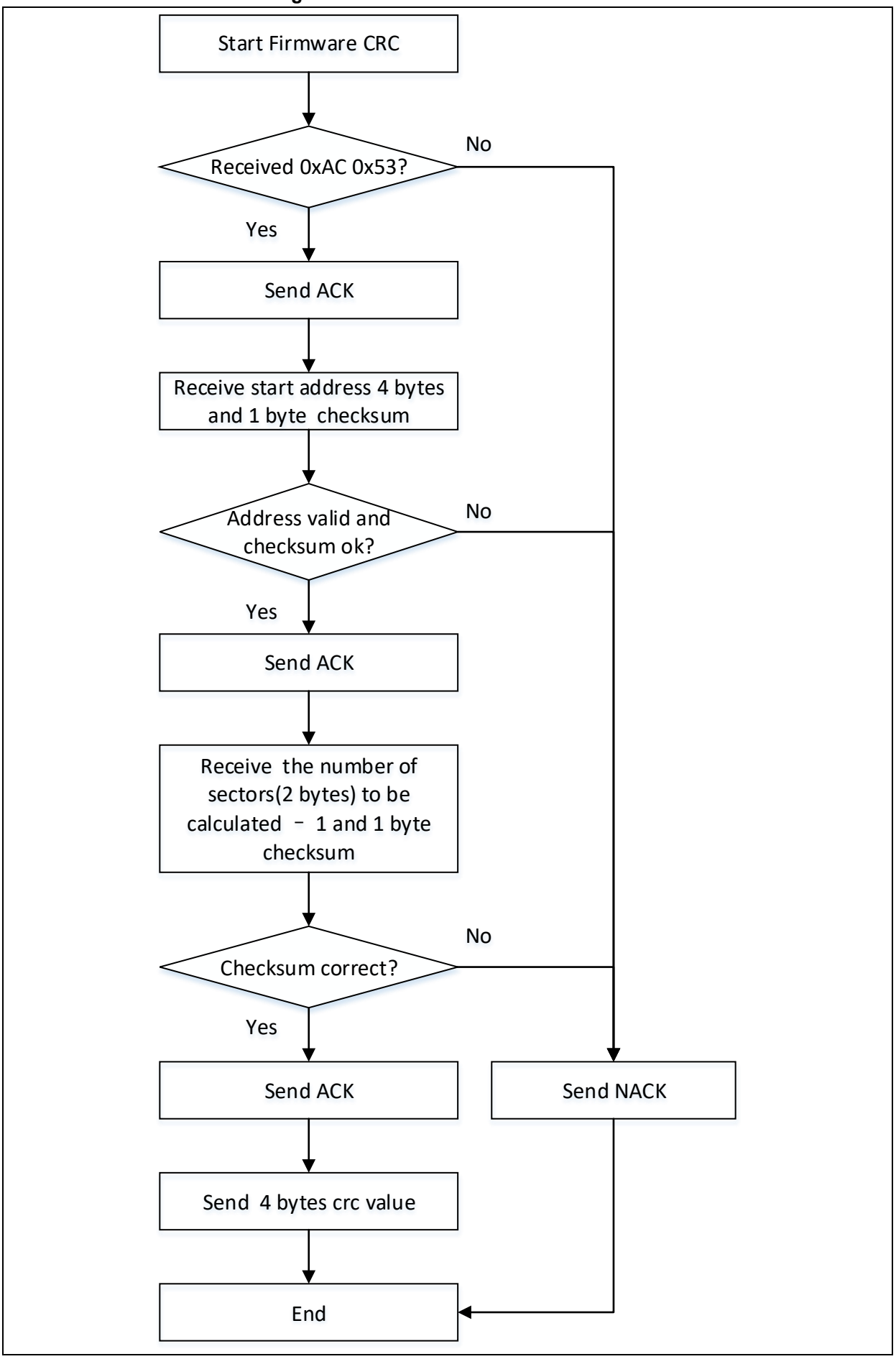


Figure 27 Firmware CRC: device side



4.13.2 Data transmit on host side

Send	Receive	Data	Description
1		0xAC	Firmware CRC
2		0x53	Firmware CRC
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Address （MSB）
4		*	Address
5		*	Address
6		*	Address （LSB）
7		*	Checksum XOR (byte 3~6)
	2	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
8		*	Number of sectors – 1 MSB（n）
9		*	Number of sectors – 1 LSB（n）
10		*	Checksum = byte8 XOR byte9 XOR 0xFF
	3	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	4	*	CRC value (MSB)
	5	*	CRC value
	6	*	CRC value
	7	*	CRC value (LSB)

4.14 Enable SPIM command

This command is used to enable bank 3. SPIM Flash type, Flash size and Flash memory encryption range shall be available for the device to use this command. Refer to the SPIM use manual for more details.

Flash type list:

- 0x90: general Flash
  Dummy cycle is 4
- 0x91: general Flash, and Quad Enable setting
  Dummy cycle is 4
  Quad Enable is activated in “Volatile” format.

The maximum allowable limit for the Flash size is 16M Bytes. The device erases, reads or writes the Flash size sent from the host.

In order to prevent SPIM Flash from being read out, SPIM is equipped with an encryption feature. In other words, it allows the Flash memory controller to write encrypted data to the SPIM, and the data are then decrypted when readout.

For MCU, read and write operation are done in a plaintext format. But those data which are stored in the external Flash memory are encrypted. The user can define a password and set encrypted range. The encryption key is located in the user system data area and must be set before write operation to SPIM (bank3).

Encryption range (also called FLASH_FDA) is configured before enabling SPIM (bank3) each time.

It is used to define how many of the bytes starting from the SPIM (bank3) initial address need to be encrypted. When a value greater than 16M Bytes is set, it indicates that all of the SPIM (bank3) must be encrypted. When the value is 0, it indicates that the whole SPIM (bank3) is not encrypted When the device receives the command, it transmits an ACK byte to the host. After the transmission of the ACK byte, the device waits for the Flash type (one byte), Flash size (four bytes), encryption address range (four bytes) and a checksum byte. If all the parameters and checksum are valid, the device transmits an ACK byte to the host.

4.14.1 Flowchart on host and device sides

Figure 28 Enable SPIM: host side

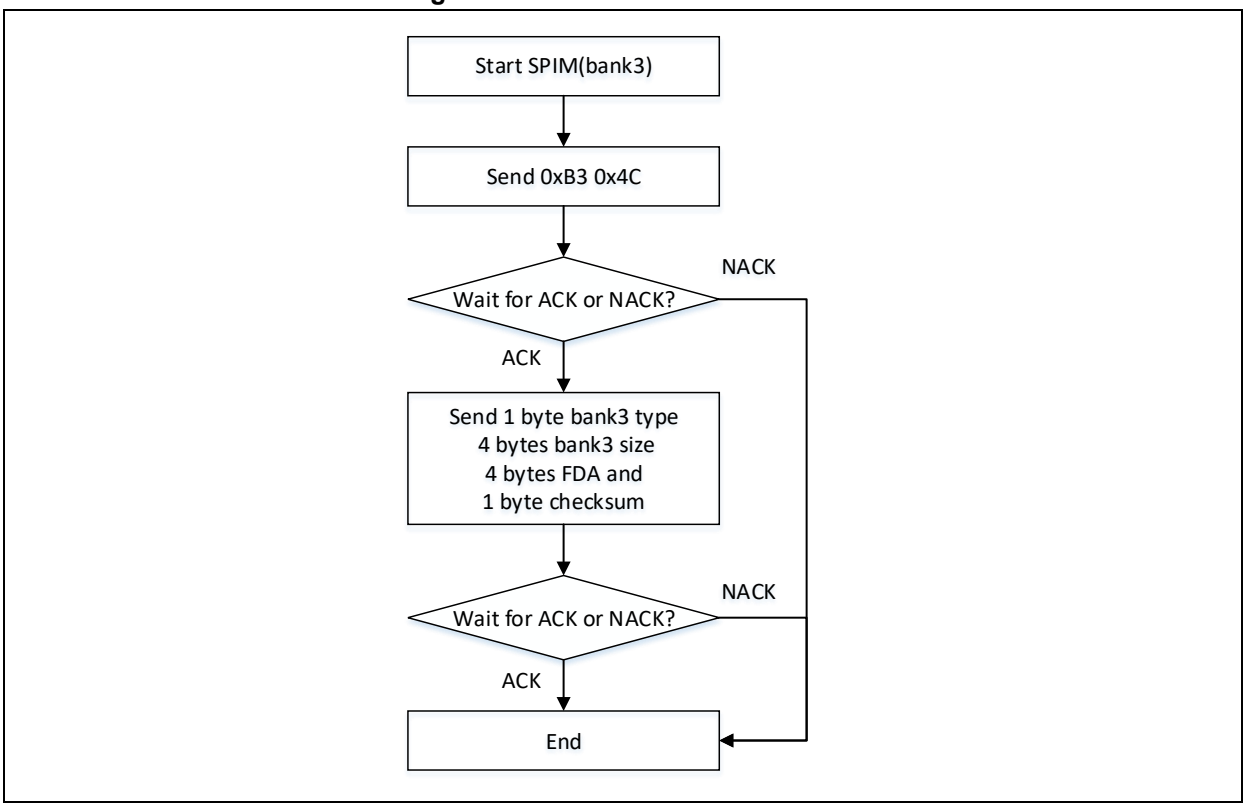
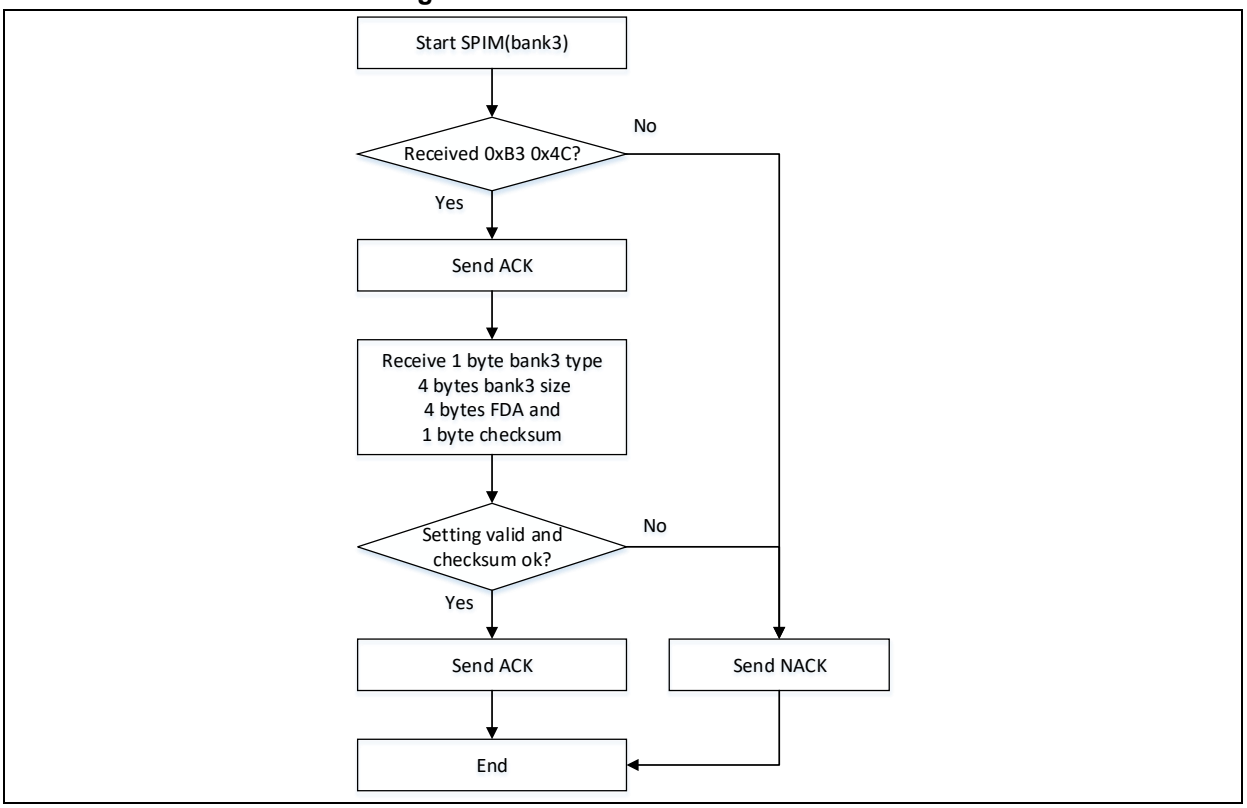


Figure 29 Enable SPIM: device side



4.14.2 Data transmit on host side

Send	Receive	Data	Description
1		0xB3	Enable SPIM
2		0x4C	Enable SPIM
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Flash type
4		*	Flash size（MSB）
5		*	Flash size
6		*	Flash size
7		*	Flash size（LSB）
8		*	Flash FDA（MSB）
9		*	Flash FDA
10		*	Flash FDA
11		*	Flash FDA（LSB）
12		*	Checksum XOR (byte 3~11)
	2	ACK	

4.15 Enable sLib

This command is used to enable security library (also called sLib). Refer to the sLib use manual for more details about it.

When the device receives the command, if the access protection is not enabled, it transmits an ACK byte to the host. After the transmission of the ACK byte, the device waits for the sLib password (four bytes), sLib start sector (two bytes), sLib data/instruction start sector (two bytes), sLib end sector (two bytes) and a checksum byte. If the checksum is valid, the device configures sLib, and then returns the setting status (one byte). After this operation, the device transmits the ACK byte to the host.

Note that a system reset has to be done for the sLib settings to take into account.

This command is not allowed and executed when the access protection is enabled.

sLib state:

- 1: sLib is enabled
- 0: sLib configuration success

4.15.1 Flowchart on host and device sides

Figure 30 Enable sLib: host side

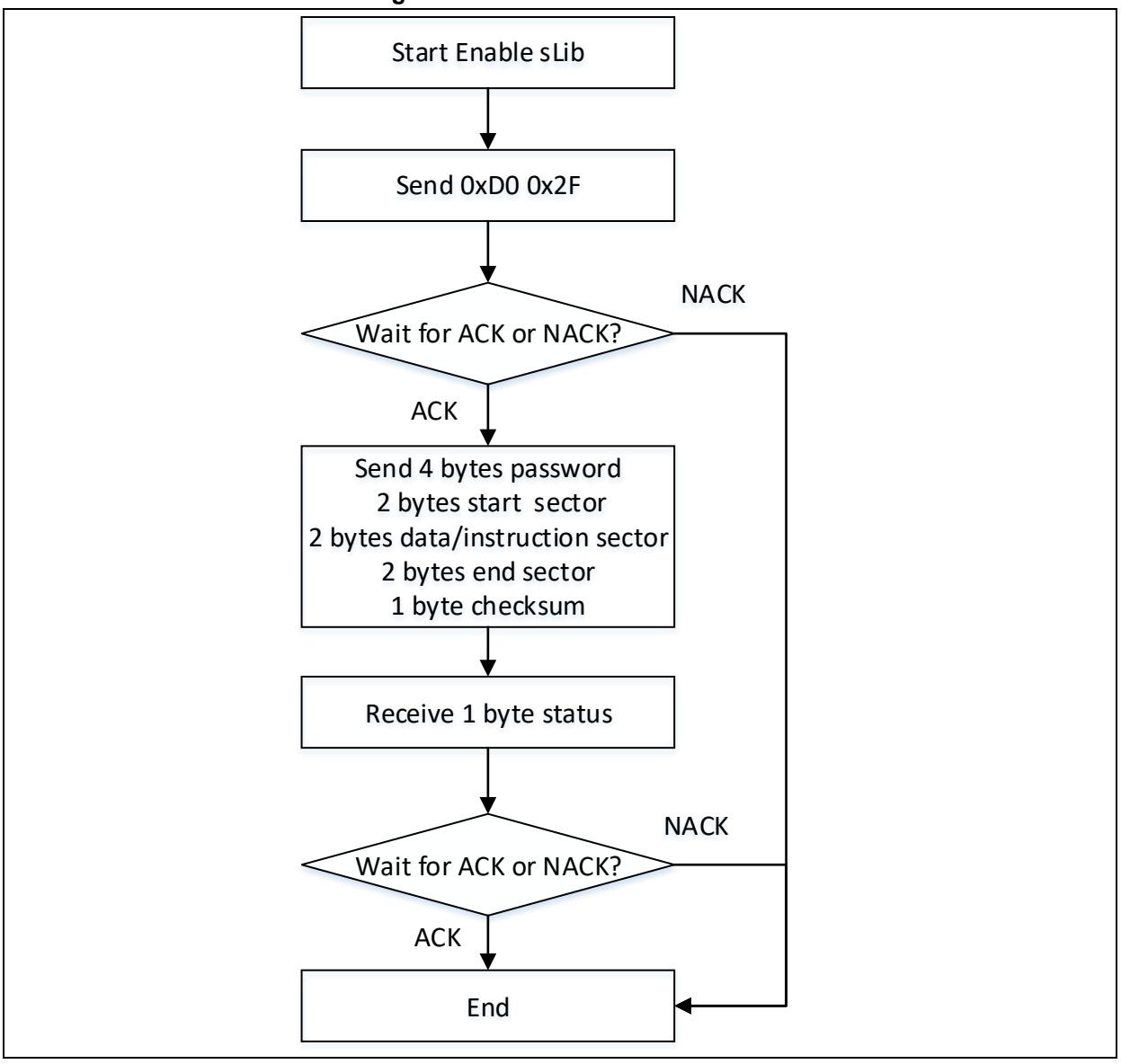
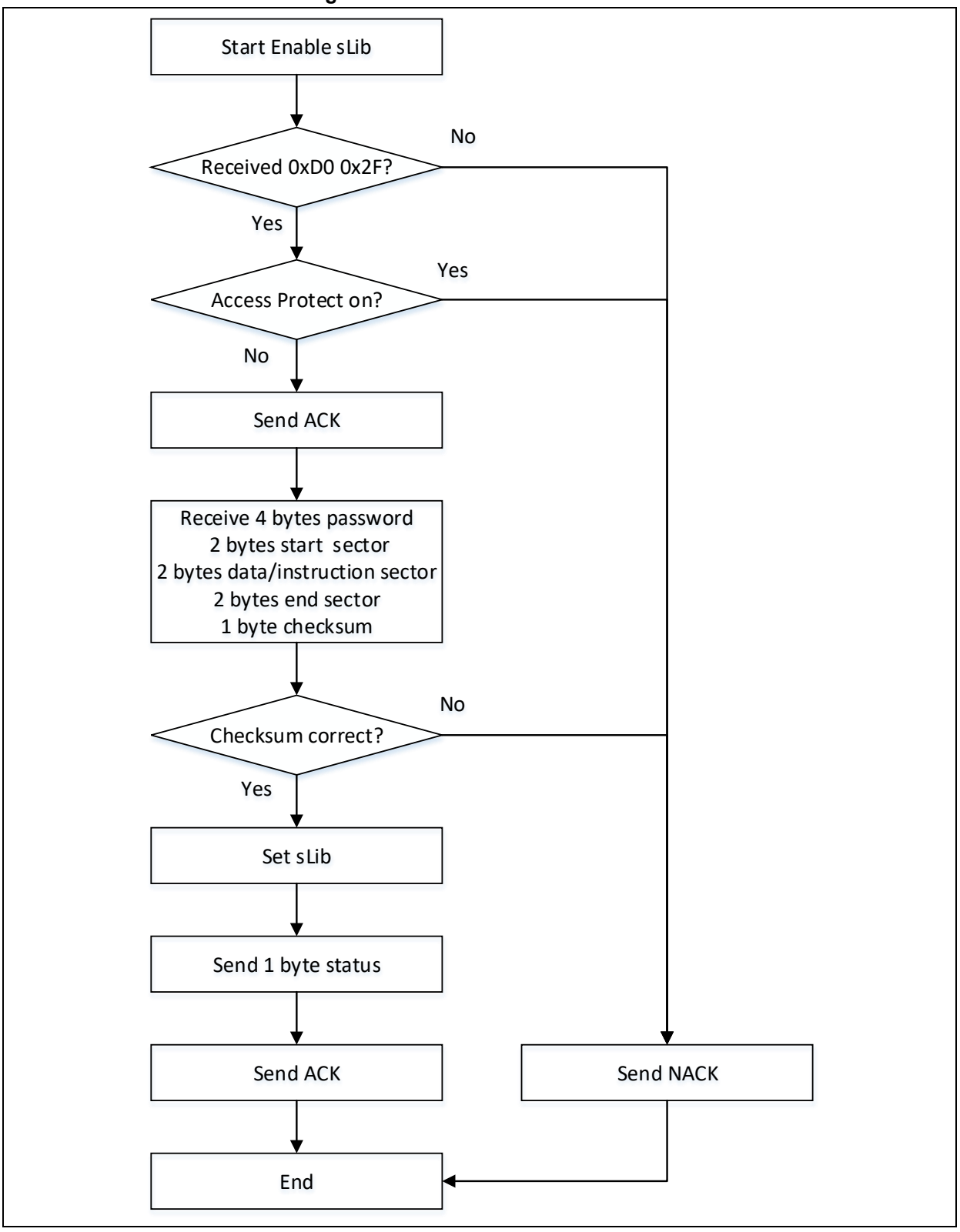


Figure 31 Enable sLib: device side



4.15.2 Data transmit on host side

Send	Receive	Data	Description
1		0xD0	Enable sLib
2		0x2F	Enable sLib
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Password（MSB）
4		*	Password
5		*	Password
6		*	Password（LSB）
7		*	sLib start sector（MSB）
8		*	sLib start sector（LSB）
9		*	sLib data/instruction start sector（MSB）
10		*	sLib data/instruction start sector（LSB）
11		*	sLib end sector（MSB）
12		*	sLib end sector（LSB）
13		*	Checksum XOR byte (3~12)
	2	0/1	sLib configuration state
	3	ACK	

4.16 Disable sLib

This command is used to disable sLib. This operation induces mass erase of the memory.
When the device receives the command, if the access protection is not enabled, it transmits an ACK byte to the host. After the transmission of the ACK byte, the device waits for the password (four bytes) and a checksum byte. If the checksum is valid, the device erases sLib, and returns the current state (one bytes). After the operation, the device transmits the ACK byte to the host.
sLib disable state:

- 1: sLib password wrong
- 0: sLib is disabled.

This command is not allowed and executed when the access protection is enabled.

4.16.1 Flowchart on host and device sides

Figure 32 Disable sLib: host side

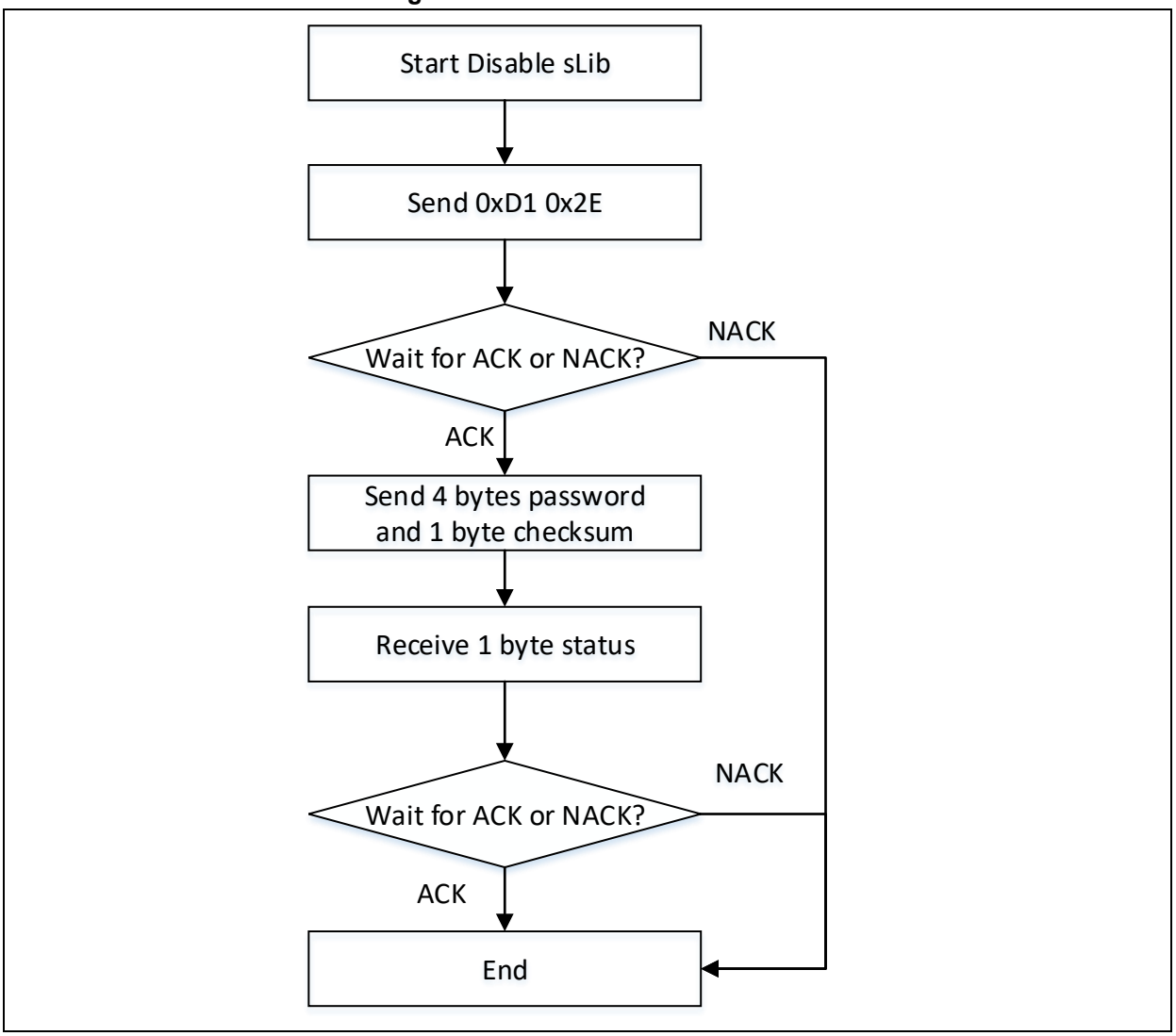
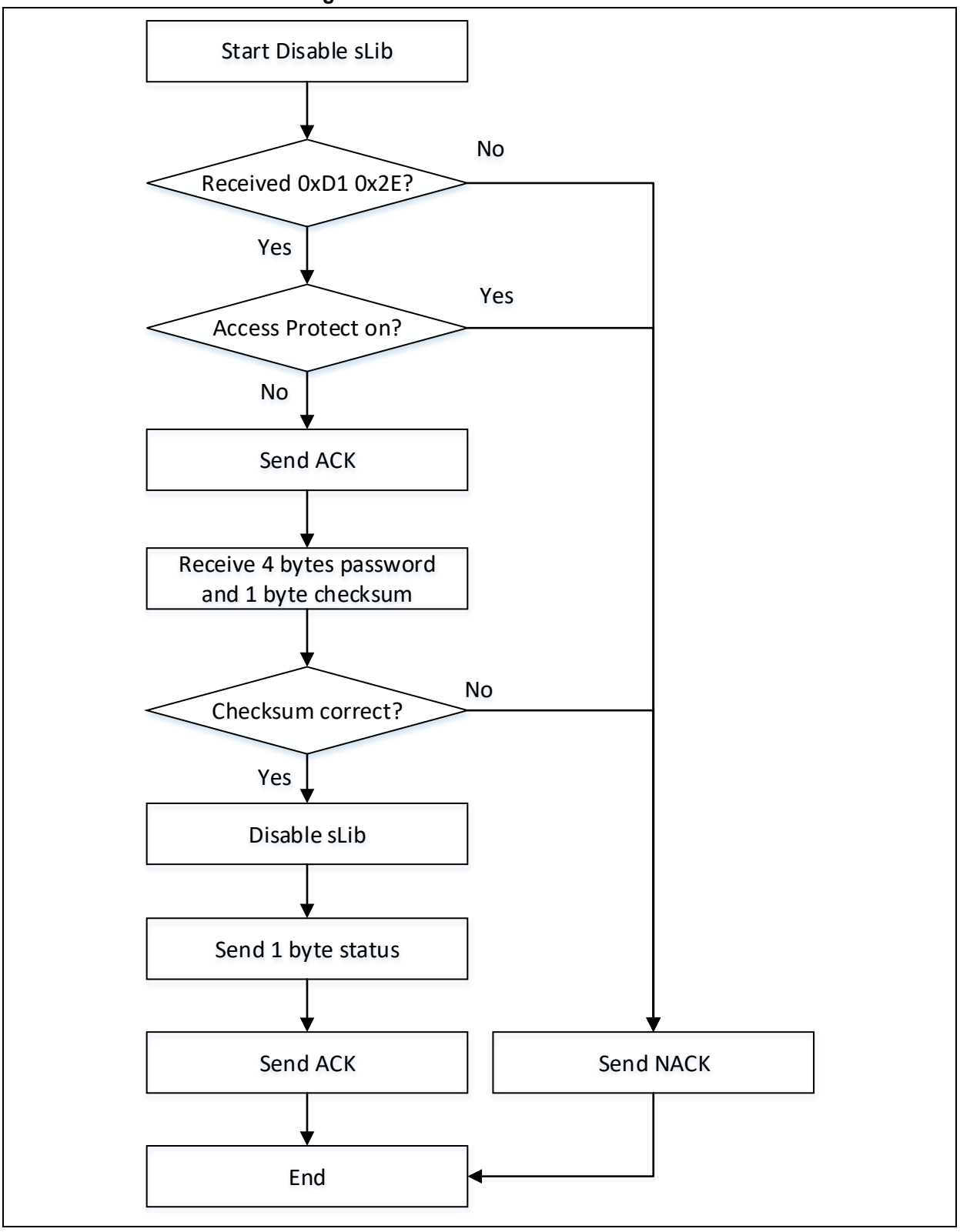


Figure 33 Disable sLib: device side



4.16.2 Data transmit on host side

Send	Receive	Data	Description
1		0xD1	Disable sLib
2		0x2E	Disable sLib
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Password（MSB）
4		*	Password
5		*	Password
6		*	Password（LSB）
	2	0/1	sLib disable state
	3	ACK	

4.17 Get sLib status

This command is used to get the current sLib state. It returns the corresponding sLib register value. For more details about the sLib, see the corresponding product datasheet.

AT32F435xx/AT32F437xx: (send 16 bytes data)
When the device receives the command, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device sends the values of the SLIB_STS0 register (four bytes), the SLIB_STS1 register (four bytes), the SLIB_STS2 register (four bytes) and SLIB_MISC_STS register (four bytes). After the operation, the device transmits the ACK byte to the host.

Other product series (send 12 bytes data)
When the device receives the command, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device sends the values of the SLIB_STS0 register (four bytes), the SLIB_STS1 register (four bytes) and the SLIB_MISC_STS register (four bytes). After the operation, the device transmits the ACK byte to the host.

Note: For AT32F403A, AT32F407 and AT32F413 series, this command is not allowed and executed when the access protection is enabled. For other product series, this command is still active when the access protection is enabled.

4.17.1 Flowchart on host and device sides

Figure 34 Get sLib status: host side

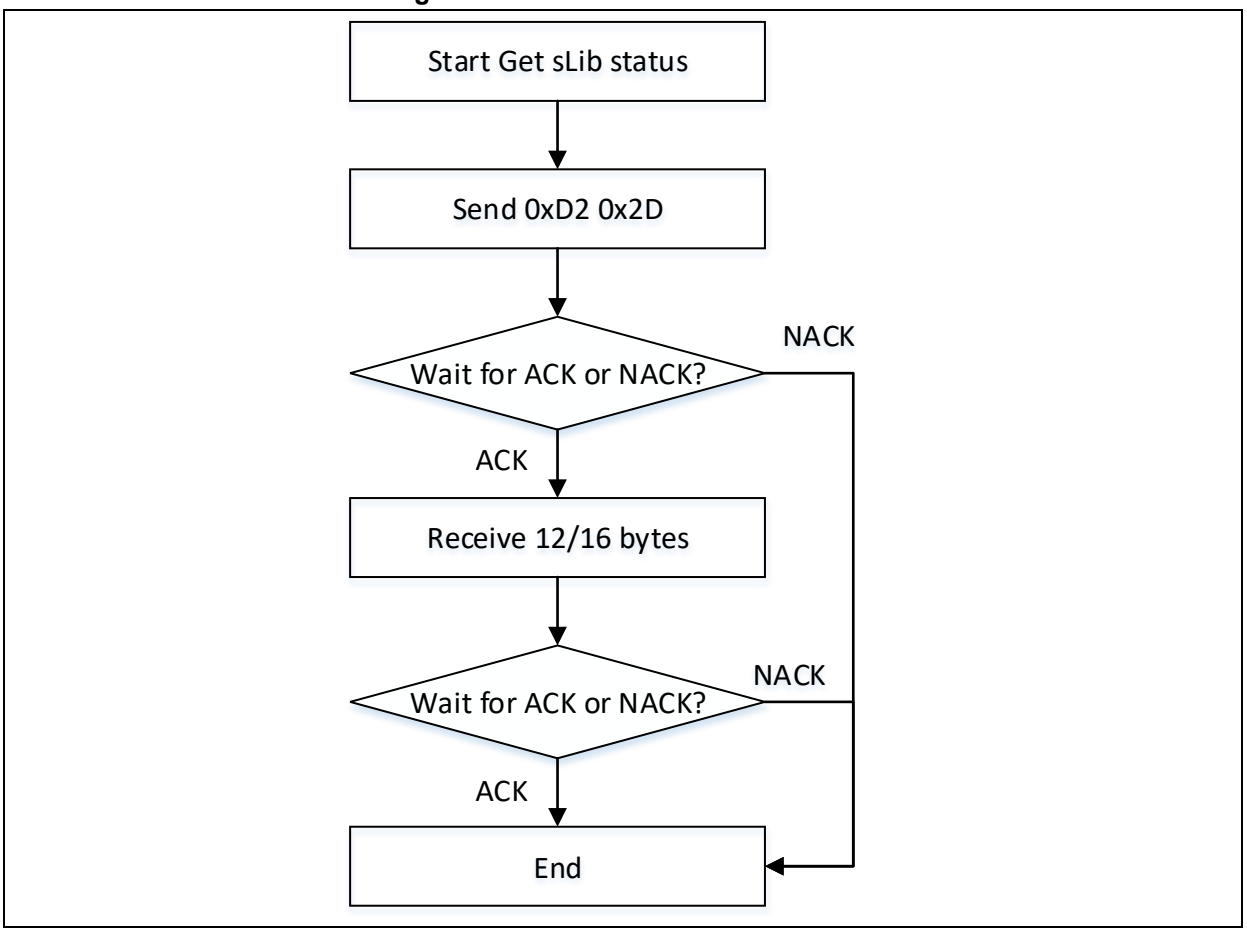
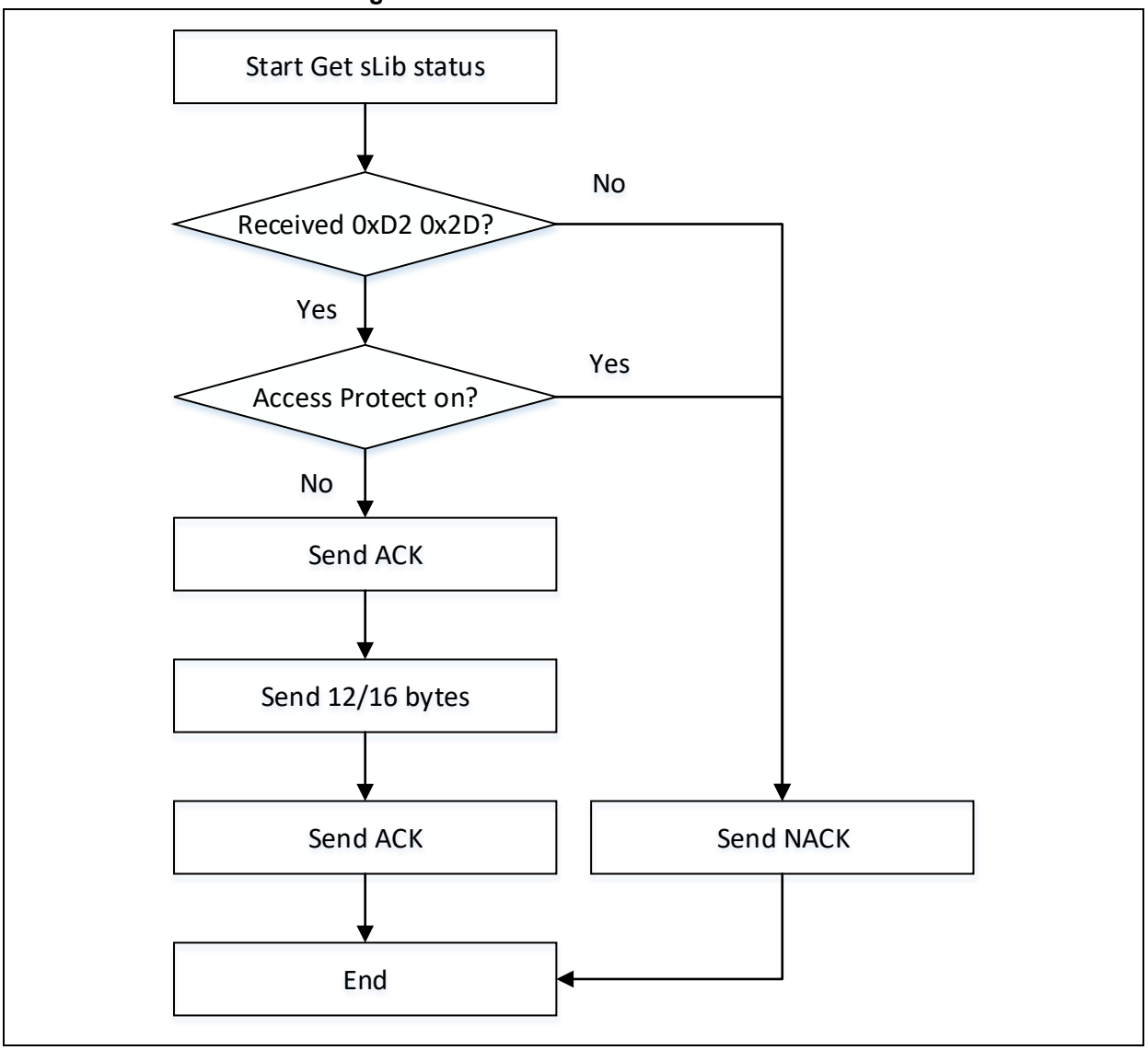


Figure 35 Get sLib status: device side



4.17.2 Data transmit on host side

AT32F435xx/AT32F437xx:

Send	Receive	Data	Description
1		0xD2	Get sLib status
2		0x2D	Get sLib status
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	2	*	SLIB_STS0（MSB）
	3	*	SLIB_STS0
	4	*	SLIB_STS0
	5	*	SLIB_STS0（LSB）
	6	*	SLIB_STS1（MSB）
	7	*	SLIB_STS1
	8	*	SLIB_STS1
	9	*	SLIB_STS1（LSB）
	10	*	SLIB_STS2（MSB）
	11	*	SLIB_STS2
	12	*	SLIB_STS2
	13	*	SLIB_STS2（LSB）
	14		SLIB_MISC_STS（MSB）
	15		SLIB_MISC_STS
	16		SLIB_MISC_STS
	17		SLIB_MISC_STS（LSB）
	18	ACK	

Other product series:

Send	Receive	Data	Description
1		0xD2	Get sLib status
2		0x2D	Get sLib status
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	2	*	SLIB_STS0（MSB）
	3	*	SLIB_STS0
	4	*	SLIB_STS0
	5	*	SLIB_STS0（LSB）
	6	*	SLIB_STS1（MSB）
	7	*	SLIB_STS1
	8	*	SLIB_STS1
	9	*	SLIB_STS1（LSB）
	10	*	SLIB_MISC_STS（MSB）
	11	*	SLIB_MISC_STS
	12	*	SLIB_MISC_STS
	13	*	SLIB_MISC_STS（LSB）
	14	ACK	

4.18 SPIM Remap

This command is used to remap the external IO of the SPIM (bank3).
When the device receives the command, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device waits for a Remap Flag byte and a checksum byte. If the received checksum is valid, the device stores Remap state and then transmits the ACK byte to the host.
Remap Flag:

- 0: SCK/PB1 CS/PA8 IO0/PA11 IO1/PA12 IO2/PB7 IO3/PB6
- 1: SCK/PB1 CS/PA8 IO0/PB10 IO1/PB11 IO2/PB7 IO3/PB6

4.18.1 Flowchart on host and device sides

Figure 36 SPIM Remap: host side

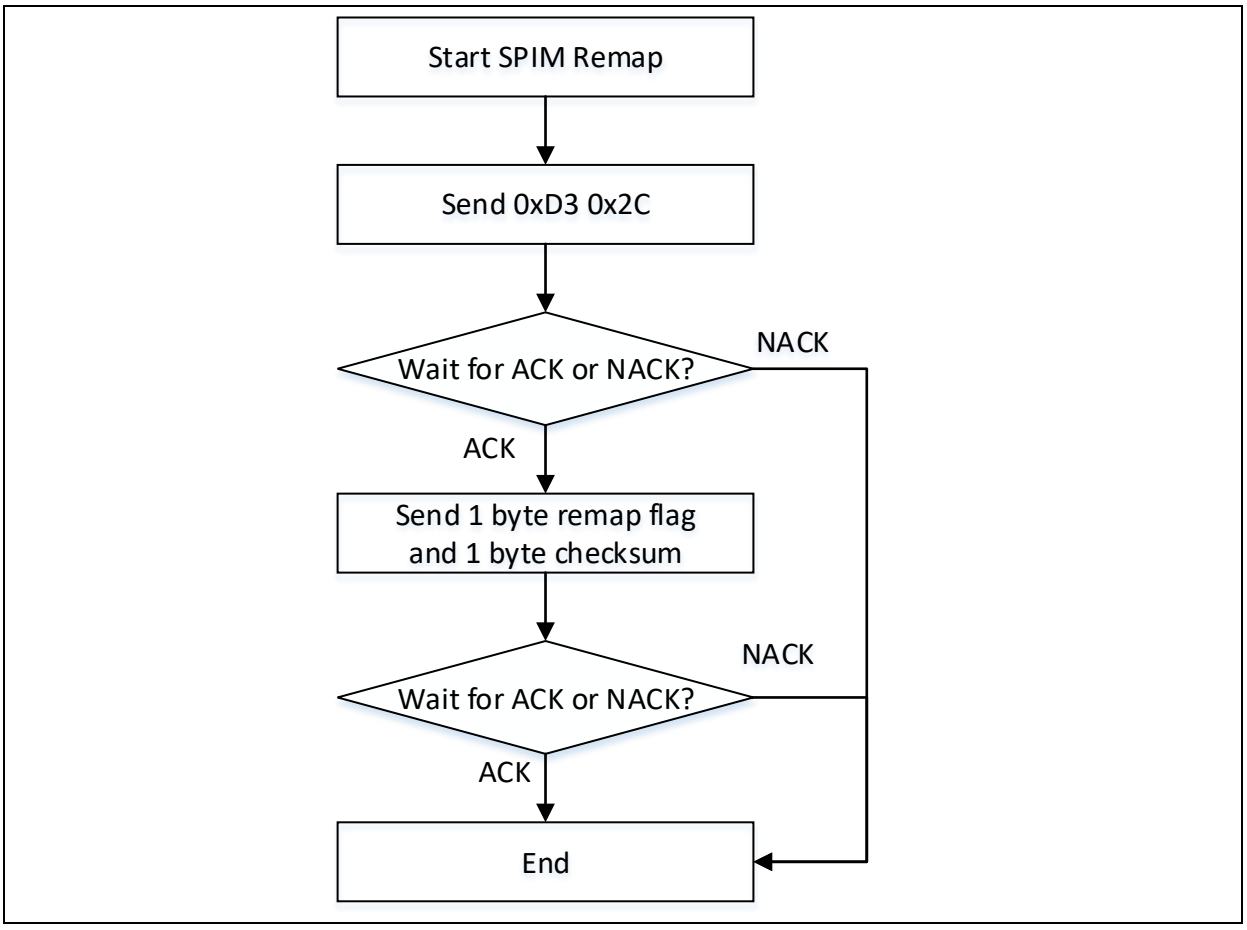
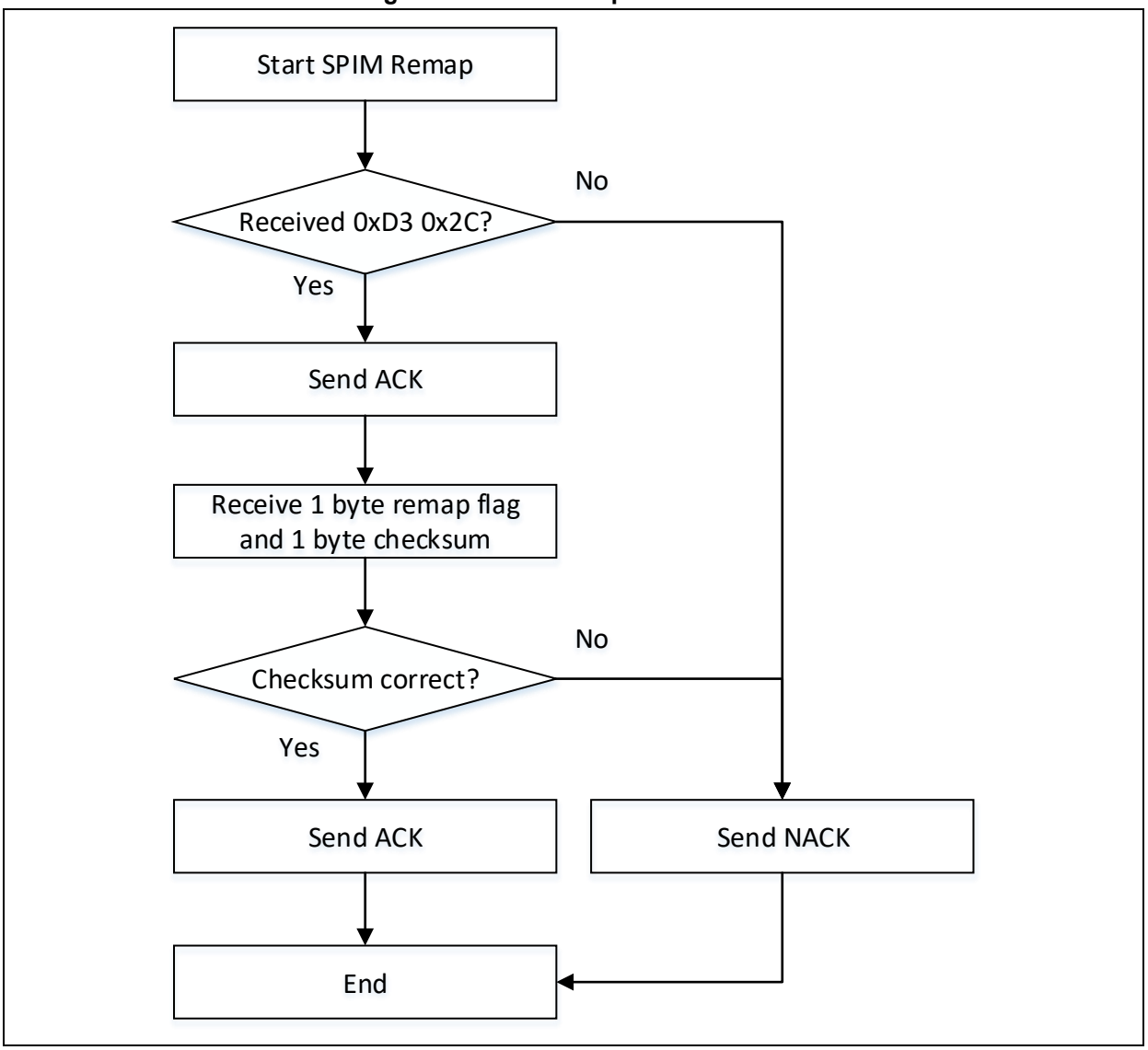


Figure 37 SPIM Remap: device side



4.18.2 Data transmit on host side

Send	Receive	Data	Description
1		0xD3	SPIM Remap
2		0x2C	SPIM Remap
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		0/1	Remap Flag
4		*	Checksum XOR byte3
	2	ACK	

4.19 Reset Device

This command is used to enable a system reset.
When the device receives the command, it transmits the ACK byte to the host in two times and then performs a system reset.

4.19.1 Flowchart on host and device sides

Figure 38 Reset Device: host side

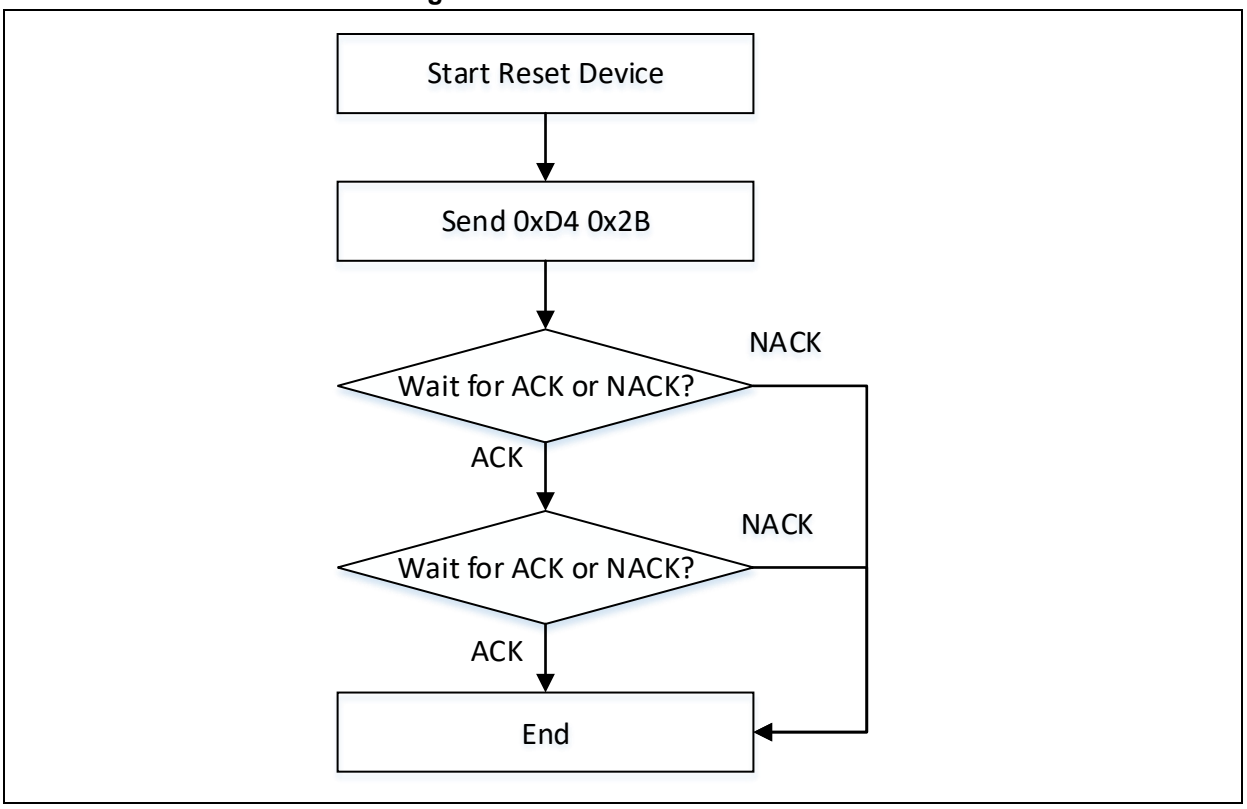
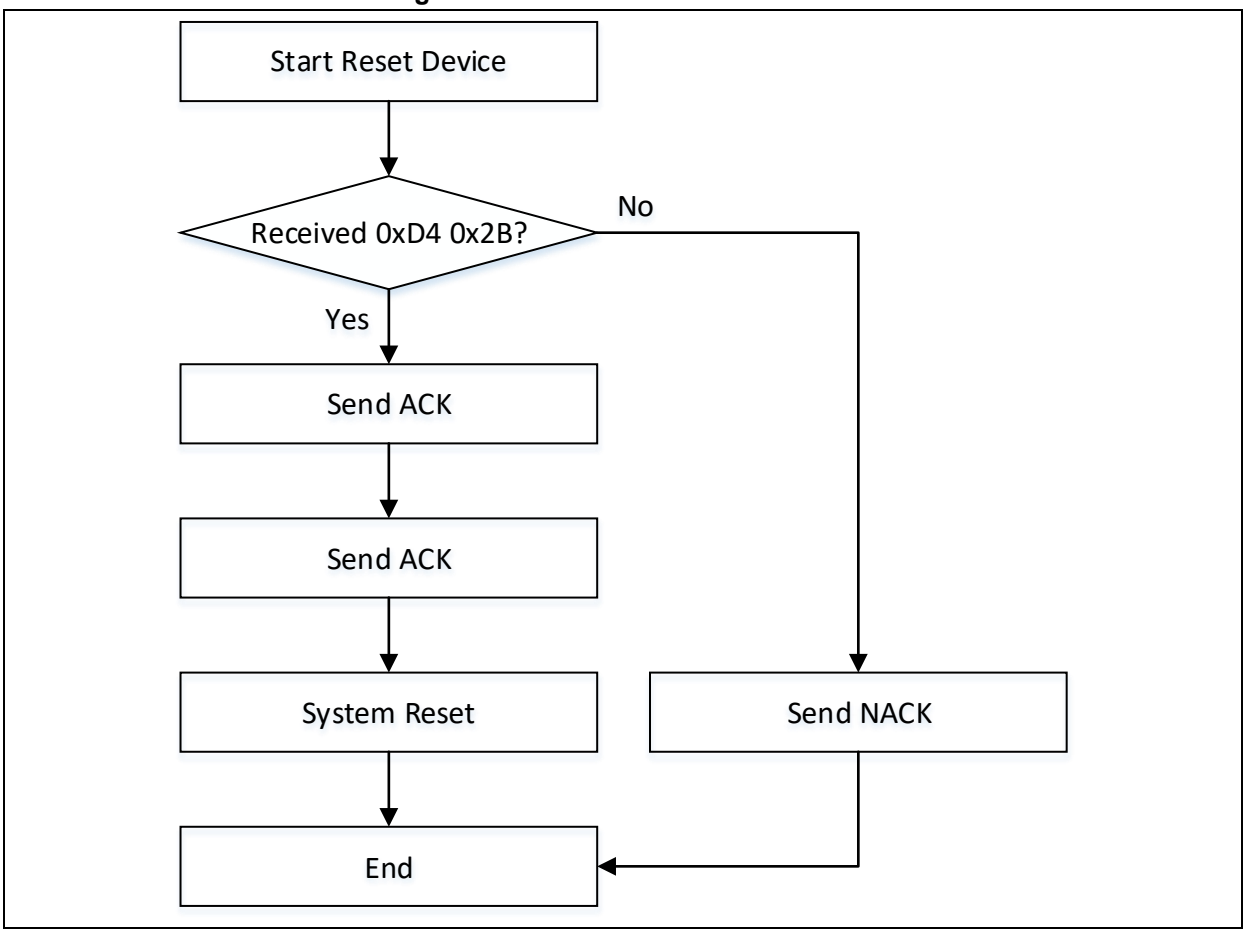


Figure 39 Reset Device: device side



4.19.2 Data transmit on host side

Send	Receive	Data	Description
1		0xD4	Reset Device
2		0x2B	Reset Device
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
	2	ACK	

4.20 Advanced Access Protect

This command is used to enable advanced access protection. Refer to the corresponding product datasheet for more details about this protection.

When the device receives the command, if the access protection is not enabled, it transmits the ACK byte to the host. After the transmission of the ACK byte, the device waits for the advanced access protection flag (two bytes, flag can be any value). Then it starts configuring the advanced access protection and transmits the ACK byte to the host before performing a system reset.

This command is not allowed and executed when the access protection is enabled.

Note: the advanced access protection cannot be disabled for some product series. For details, refer to the corresponding user manual.

4.20.1 Flowchart on host and device sides

Figure 40 Advanced Access Protect: host side

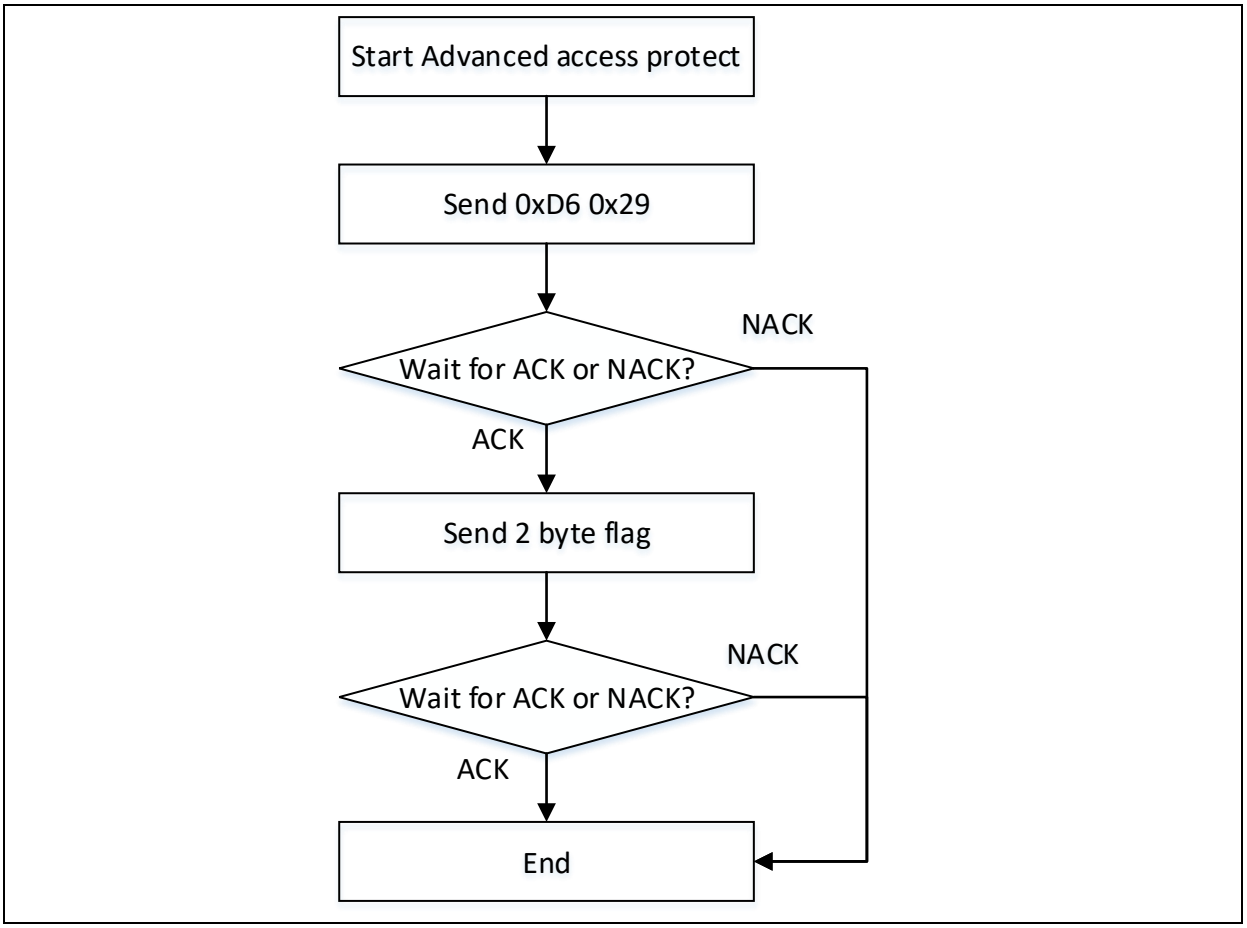
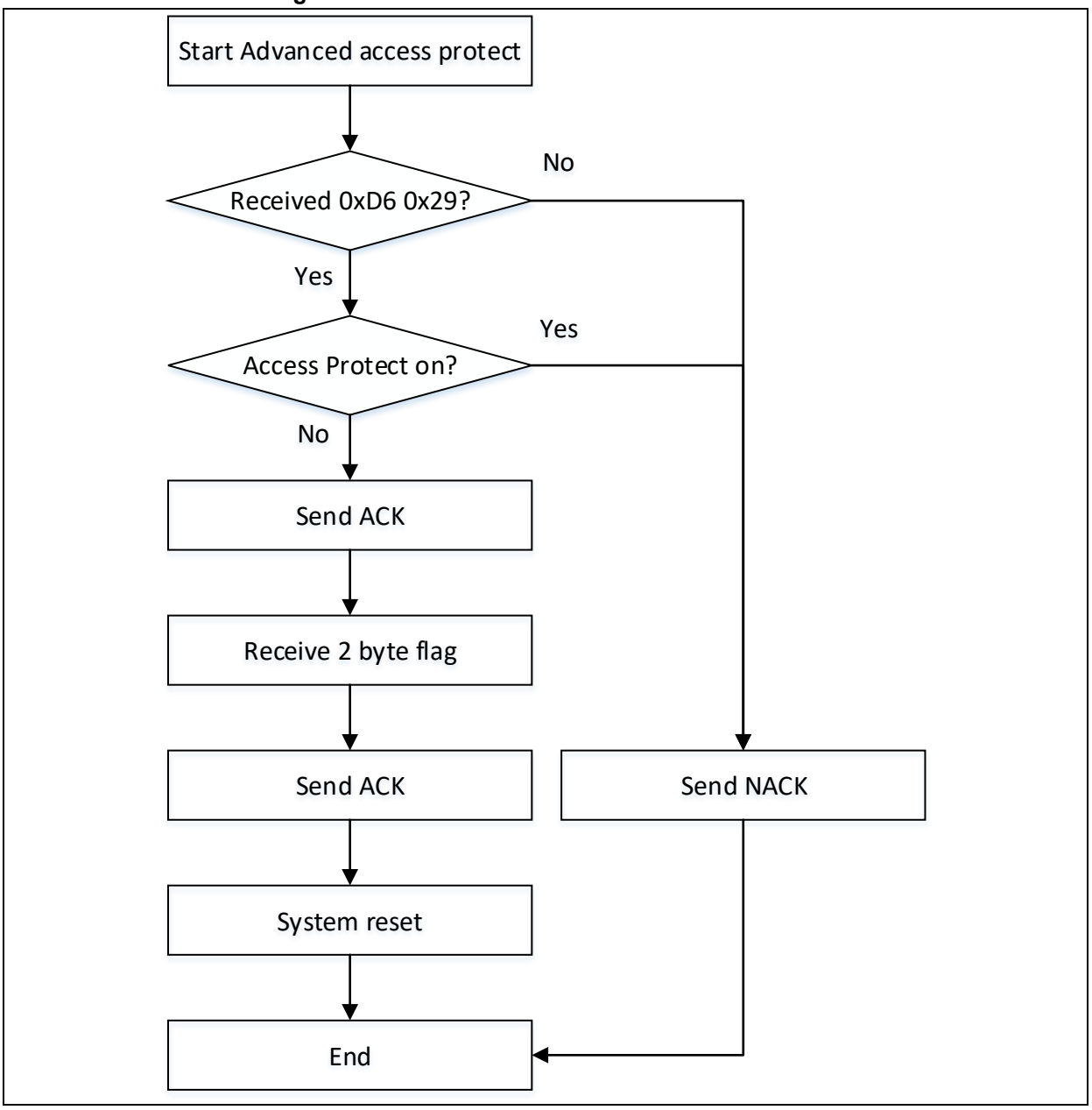


Figure 41 Advanced Access Protect: device side



4.20.2 Data transmit on host side

Send	Receive	Data	Description
1		0xD6	Advanced access protect
2		0x29	Advanced access protect
	1	ACK/NACK	When an NACK byte is received, it indicates the end of the command.
3		*	Flag
4		*	Flag
	2	ACK	

5 Revision history

Table 2 Document revision history

Date	Revision	Changes
2021.12.07	2.0.0	Initial release
2023.01.05	2.0.1	1. Add AT32F423 series 2. Add AT32L021 series 3. Update descriptions 4. Revise the description of AT32F421 to “does not support Jump command”
2023.06.30	2.0.2	1. Add AT32F405/402 series 2. Add AT32A403A series
2024.03.20	2.0.3	1. Add AT32A423 series
2024.04.28	2.0.4	1. Update supported product list 2. Update supported command list

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license granted by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement on any patent, copyright or other intellectual property right.

Purchasers hereby agree that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any aircraft application; (C) any aerospace application or environment; (D) any weapon application, and/or (E) or other uses where the failure of the device or product could result in personal injury, death, property damage. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and Purchasers are solely responsible for meeting all legal and regulatory requirements in such use.

Resale of ARTERY products with provisions different from the statements and/or technical characteristics stated in this document shall immediately void any warranty grant by ARTERY for ARTERY's products or services described herein and shall not create or expand any liability of ARTERY in any manner whatsoever.

© 2024 Artery Technology -All rights reserved