

User guide

UG_N32G430 Series MCU BOOT User Guide

Introduction

The usage guide mainly describes the BOOT interface instructions of N32G430 series MCU, which is easy to download and develop by using the NSING technology BOOT loader.

Content

1	BOOT brief introduction	3
1.1	BOOT function definition	3
2	BOOT process and command processing	4
2.1	Commands and data structures.....	4
2.1.1	The list of commands.....	4
2.1.2	The data structure	4
2.2	Command description	5
2.2.1	CMD_SET_BR.....	5
2.2.2	CMD_GET_INF	6
2.2.3	CMD_KEY_RNG.....	7
2.2.4	CMD_KEY_UPDATE.....	8
2.2.5	CMD_FLASH_ERASE	10
2.2.6	CMD_FLASH_DWNLD.....	11
2.2.7	CMD_DATA_CRC_CHECK	13
2.2.8	CMD_OPT_RW	15
2.2.9	CMD_USERX_OP	16
2.2.10	CMD_SYS_RESET.....	19
2.3	Returns the status word description	19
2.3.1	Returns the success status word.....	19
2.3.2	Returns the failure status word	20
2.3.3	Return other status words	20
3	BOOT Instructions.....	21
3.1	Upper computer control process	21
3.1.1	Erase command control flow chart	22
3.1.2	Download the command control flow chart.....	22
3.1.3	Update the key command control flow chart	24
3.1.4	Partition operation command control flow chart	24
3.1.5	Option byte read/write command control flow chart	25
4	Version history	26
5	Notice	27

1 BOOT brief introduction

The firmware program of the chip, namely BOOT, mainly provides user program download, API and other functions.

This document describes in detail the function, implementation and introduction of BOOT of N32G430 series chips. The maximum FLASH storage area of N32G430 series chips is 64KB.

1.1 BOOT function definition

◆ User program download function

- Support USART (USART1, use GPIO as PA9-TX, PA10-RX, baud rate negotiation);
- Support download data CRC32 verification;
- Support encrypted download (AES-128 ECB);
- Support key authentication during FLASH partition and partition erasure download;
- Support partition key update;
- Supports power-on BOOT self-verification.
- Support software reset chip operation.

2 BOOT process and command processing

The firmware program BOOT of N32G430 series chips supports downloading user programs and data through the USART interface. The following describes the related command processing flow.

2.1 Commands and data structures

2.1.1 The list of commands

Table 2-1 Command definition

Name of the command	The key value	Description
CMD_SET_BR	0x01	Set the baud rate of the serial port (Valid only when serial ports are used)
CMD_GET_INF	0x10	Read chip model index, BOOT version number, chip ID
CMD_GET_RNG	0x20	Get random number
CMD_KEY_UPDATE	0x21	Update the encryption download key or partition authentication key
CMD_FLASH_ERASE	0x30	Erase FLASH
CMD_FLASH_DWNLD	0x31	Download user programs to FLASH
CMD_DATA_CRC_CHECK	0x32	CRC verification download user program
CMD_OPT_RW	0x40	Read/configure option bytes (including read protection level, FLASH page write protection, Data0/1 configuration, USER configuration)
CMD_USERX_OP	0x41	Get the partition USERX size and set the partition USERX size
CMD_SYS_RESET	0x50	The system reset

2.1.2 The data structure

Here are some conventions explained below, where "<>" represents fields that must be included, and "()" represents fields that are included according to different parameters.

Upper and lower instruction data structures

- Upper instruction structure:

<CMD_H + CMD_L + LEN + Par> + (DAT).

CMD_H represents the first-level command field, CMD_L represents the second-level command field; LEN represents the length of the transmitted data; Par represents the 4-byte command parameter; DAT represents the specific data sent by the upper layer command to the lower layer;

- Lower level response structure:

< CMD_H + CMD_L + LEN > + (DAT) + <CR1+CR2>.

CMD_H represents the first-level command field, CMD_L represents the second-level command field, the lower-level command field is the same as the corresponding upper-level command field; LEN represents the length of the transmitted data; DAT represents the specific data that the lower layer responds to the upper layer; CR1+CR2

represents the return to the upper layer Command execution result, if the first-level and second-level command fields of the command sent by the upper layer do not belong to any command, BOOT replies with CR1=0xBB and CR2 = 0xCC.

Command data structures supported by the serial port:

1. The upper computer sends the upper-layer command:

STA1 + STA2 + {upper instruction structure} + XOR.

STA1 and STA2 are the starting bytes of the command sent by the serial port, STA1=0xAA, STA2=0x55. Used for the chip to identify the host computer to send the serial data stream.

XOR represents the XOR value of the previous command byte (STA1 + STA2 + {upper instruction structure}).

2. The upper computer receives the lower-layer response:

STA1 + STA2 + {lower response structure} + XOR.

STA1 and STA2 are the starting bytes of the command sent by the serial port, STA1=0xAA, STA2=0x55. Used for the host computer to identify the chip to send serial data stream

XOR represents the XOR value of the previous command byte (STA1 + STA2 + {lower response structure}).

2.2 Command description

2.2.1 CMD_SET_BR

This command is used to modify the serial port baud rate.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x01 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3(LEN)	Length of data sent: 0x00,0x00							
4~7(Par)	Par[0~3] : Set baud rate parameters							
(DAT)	None							

- Par[0~3], the serial port baud rate negotiation setting value can be set to the maximum, and the setting range is 2.4Kbps ~ 4Mbps, the default baud rate is 9600bps;
- Reserved value: 0x00;

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x01 Level-1 command field							

1(CMD_L)	0x00 Level-2 command field
2~3(LEN)	Length of data sent: 0x00,0x00
(DAT)	None
4(CR1)	Status byte 1
5(CR2)	Status byte 2

- Status bytes (CR1 and CR2) are divided into the following types according to command execution:

- 1) Return success: status flag bit (0xA0, 0x00).
- 2) Return failure: status flag bits (0xB0, 0x00).

The following are the baud rate values supported by baud rate negotiation (✓ means supported, / means not supported):

The clock parameters (MHz)		Baud rate															
		2400	4800	9600	14400	19200	38400	57600	115200	128000	256000	576000	923076	1000000	2000000	3000000	4000000
HSE	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	/
	8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	24	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	/
	32	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HSI		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	/	/	/	/

2.2.2 CMD_GET_INF

The command reads the BOOT version number, chip model index, chip ID, and chip serialization information.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x10 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3 (LEN)	Length of data sent							
4~7(Par)	Reserved							
(DAT)	None							

- Reserved value: 0x00.
- LEN send data length: 0x00(LEN[0]), 0x00(LEN[1]), LEN = LEN[0] +(LEN[1]<<8).

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x10 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3 (LEN)	The length of the data							
4~54(DAT)	BOOT version, chip model index, and chip ID							
55(CR1)	Status byte 1							
56(CR2)	Status byte 2							

- The procedure byte (CMD_H) corresponds to the upper instruction (CMD_H).
- LEN is the data length: $0x33(\text{LEN}[0])$, $0x00(\text{LEN}[1])$, $\text{LEN} = \text{LEN}[0] + (\text{LEN}[1] \ll 8)$.
- DAT[0] : 0x05, chip model index
- DAT[1]: 0xXY, BOOT version number (BCD code)
 - 0x10: indicates the command set version used by BOOT, indicating that the command set version of V1.0 is used.
- DAT[2] : BOOT command set version
- DAT[3~50] 48Byte
 - 1) DAT[3~18] : 16Byte UCID (for details about the UCID, see the user manual).
 - 2) DAT[19~30] : 12Byte Chip ID(UID) (for details, see the user manual).
 - 3) DAT[31~34] : 4Byte DBGMCU_IDCODE (for details about DBGMCU_IDCODE, see the user manual).
 - 4) DAT[35~50] : 16Byte chip model.
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 - 1) Return success: status flag bit (0xA0, 0x00).
 - 2) Return failure: status flag bits (0xB0, 0x00).

2.2.3 CMD_KEY_RNG

Gets the random number of the key that the user needs to verify.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x20 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3(LEN)	Length of data sent							
4~7(Par)	Reserved							

(DAT)	None
-------	------

- Reserved value: 0x00;
- LEN Send data length: 0x00(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x20 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3(LEN)	Length of data sent							
4~19(DAT)	16Bytes pseudo-random number							
20(CR1)	Status byte 1							
21(CR2)	Status byte 2							

- LEN Send data length: 0x10(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- 16Byte pseudo-random number is generated by software algorithm.
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 - 1) Return success: status flag bit (0xA0, 0x00).
 - 2) Return failure: status flag bits (0xB0, 0x00).

2.2.4 CMD_KEY_UPDATE

Users can update the encrypted download key and partition authentication key. Before updating, they need to use CMD_KEY_RNG to obtain a random number. The random number is used to generate the old key authentication value of 16Bytes by the host computer, and then send it to BOOT through commands such as CMD_KEY_UPDATE. The authentication value verifies whether the old key is correct, thereby confirming whether to update the key. The new key needs to be decrypted with the old key.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x21 Level-1 Command field							
1(CMD_L)	Level-2 command field: KEY index ID							
2~3(LEN)	Length of data sent							
4~7(Par)	Reserved value: 0x00							
8~55(DAT)	DAT[0~15] : 16Bytes old key authentication value							
	DAT[16~31] : 16Bytes new key authentication value							

	DAT[32 to 47] : CRC32 check encrypted value 4Bytes (old key value + new key value)CRC32 check value + 12Bytes fill the value 0x00 Then encrypt 16Bytes of data with the old key
--	--

- **CMD_L**: indicates the key index ID that needs to be updated
ID(0x00-0x01) : key index ID, 0x00 indicates partition 1, 0x01 indicates partition 3.
- **LEN** Send data length: 0x30(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- Reserved value: 0x00.
- **DAT[32~47]** : CRC32 check value.
- **DAT[0~15]** : the 16-bit random number obtained by the host computer with **CMD_KEY_RNG** and the authentication value generated by the old key.
- **DAT[16-31]** : New key encrypted with old key, BOOT decrypted with old key and then save new key.

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x21 Level-1 Command field							
1(CMD_L)	Level-2 command field: key ID							
2~3(LEN)	Length of data sent							
(DAT)	1byte: the maximum number of updates is 12							
4(CR1)	Status byte 1							
5(CR2)	Status byte 2							

- **LEN** Send data length: 0x01(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- **DAT[0]** : 1byte, the maximum number of updates is 12 times, the number of times of sharing between two partitions, when 0x0D is returned, it cannot be updated anymore, the first time is to return 0x02.
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 1. Return success: status flag bit (0xA0, 0x00)
 2. Return failure: status flag bits (CR1, CR2)
 - (1) (0xB0, 0x00) : Return failure;
 - (2) (0xB0, 0x10) : Key index ID range error
 - (3) (0xB0, 0x11) : New key CRC check error;
 - (4) (0xB0, 0x20) : Old key authentication failed;
 - (5) (0xB0, 0x3F) : Failed to update the management information;

2.2.5 CMD_FLASH_ERASE

BOOT provides the function of erasing FLASH in units of pages. CMD_KEY_RNG needs to be used to obtain a random number before erasing authentication. The page address number and page number to be erased are provided by the user. The erased FLASH space cannot exceed the entire FLASH space, and at least erase Divide by 1 page (512Byte).

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x30 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3(LEN)	Length of data sent (0)							
4~7(Par)	Page address number 2 bytes: 0~255 Page number 2 bytes :1~256							
8~23(DAT)	DAT[0:15] : 16 bytes Key authentication value for USER1/3 partition authentication							

- CMD_L: erases the partition number
 - 1) 0x00=USER1
 - 2) 0x02=USER3
- LEN Send data length: 0x10(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- The erase address and range consist of four bytes in the Par field
 - Par[0~1] : page address number 2 bytes (0~255)

Page address number = $Par[0] + Par[1] \ll 8$;
 - Par[2~3] : Page number 2 bytes (1~256)

Page number = $Par[2] + Par[3] \ll 8$;
 - The header address of page 0 is 0x0800_0000. The number of subsequent pages is incremented by 1, and the header address is incremented by 0x800.

Such as:

The header address of page 1 is $0x0800_0000 + 1 * 0x800 = 0x0800_0800$

The header address of page 2 is $0x0800_0000 + 2 * 0x800 = 0x0800_1000$
 - The entire address range erased

For example: the page address number is 0x01, and the page number is 0x02

Erasing address range:

$(0x0800_0000 + 1 * 0x800) \sim (0x0800_0000 + 1 * 0x800 + 2 * 0x800)$. That is (the header address of the page address number) ~ (the header address of the page address number + the number of pages * the size of the

page)

- DAT[0:15], 16 byte partition authentication key authentication value:
 - If partition authentication is not enabled, you can enter any value. The BOOT program will not use this value.

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x30 Level-1 command field							
1(CMD_L)	Level-2 command field: Erase area							
2~3(LEN)	Length of data sent							
(DAT)	None							
4(CR1)	Status byte 1							
5(CR2)	Status byte 2							

- LEN Send data length: 0x00(LEN[0]), 0x00(LEN[1]), LEN = LEN[0] + (LEN[1]<<8).
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 1. Return success: status flag bit (0xA0, 0x00).
 2. Return failure: status flag bits (CR1, CR2).
 - (1) (0xB0, 0x00) : Return failure;
 - (2) (0xB0, 0x20) : Key authentication fails.
 - (3) (0xB0, 0x30) : The erased FLASH page is protected by RDP;
 - (4) (0xB0, 0x31) : The erased FLASH page is protected by WRP;
 - (5) (0xB0, 0x32) : Erase FLASH page is protected by partition;
 - (6) (0xB0, 0x33) : Erase FLASH page range across partitions;
 - (7) (0xB0, 0x34) : The FLASH address range is out of bounds (that is, it exceeds the size of the entire FLASH);
 - (8) (0xB0, 0x37) : Failed to erase the FLASH.
 - (9) (0xB0, 0x3F) : Failed to update the management information;

2.2.6 CMD_FLASH_DWNLD

This command allows users to download codes into the specified FLASH. Before authentication, encryption, or decryption, CMD_KEY_RNG is used to obtain random numbers. Data length must be 16 bytes aligned (less than 0x00 automatically added by the upper computer), all provided by upper-layer commands. For partition authentication and encryption download, you need to provide the partition number. To encrypt the downloaded data, decrypt the data into plaintext by encrypting the download key (that is, the key used for partition authentication) and write the data into the FLASH.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x31 Level-1 command field							
1(CMD_L)	Level-2 command field: Download partition number							
2~3(LEN)	Length of data sent							
4~7(Par)	Start address for downloading the FLASH							
8~23(DAT)	DAT[0:15] : 16 bytes Key authentication value for USER1/3 partition authentication							
24~(24+N)(DAT)	DAT[16~16+N] : Specific data to be downloaded							
(24+N+1)~(24+N+4)(DAT)	DAT[16+N+1~16+N+4] : specifies the 4 byte CRC32 check value of data							

- CMD_L: download partition number
 - 1) 0x00 = USER1;
 - 2) 0x02 = USER3;
- LEN send data length: 0xXX(LEN[0]), 0xXX(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$
- Par [0 ~ 3] : download the starting address of the FLASH, synthetic rules to Address = $Par[0] \mid Par[1] \ll 8 \mid Par[2] \ll 16 \mid Par[3] \ll 24$.
- DAT[0~15] : 16-byte partition authentication key authentication value, encryption download key and partition authentication key is the same!:
 - 1) If partition authentication is not enabled, it can be entered as all 0x00.
- DAT[16~16+N] : download specific data, the total number of data is N+1
 - 1) USART: up to 128 bytes, $16 \leq N+1 \leq 144$. N+1 must be a multiple of 16.
- DAT[16+N+1~16+N+4] : 4Byte CRC32 check value of unencrypted data

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x31 Level-1 command field							
1(CMD_L)	Level-2 command field: Download partition number							
2(LEN)	Length of data sent							
(DAT)	None							
3(CR1)	Status byte 1							
4(CR2)	Status byte 2							
5(XOR)	XOR operation result							

- LEN Send data length: 0x00(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 1. Download success: status flag bit (0xA0, 0x00).
 2. Download failed: status flag bit (CR1, CR2).
 - (1) (0xB0, 0x00) : Return failure;
 - (2) (0xB0, 0x20) : Key authentication fails.
 - (3) (0xB0, 0x21) : The number of key authentication failures exceeds the limit.
 - (4) (0xB0, 0x30) : The downloaded FLASH address is protected by RDP;
 - (5) (0xB0, 0x31) : The downloaded FLASH address is protected by WRP;
 - (6) (0xB0, 0x32) : The downloaded FLASH address is protected by partition;
 - (7) (0xB0, 0x33) : Download FLASH address range across partitions;
 - (8) (0xB0, 0x34) : Download FLASH address range is out of bounds (refers to the size of the entire FLASH);
 - (9) (0xB0, 0x35) : Download FLASH start address is not 16 bytes aligned;
 - (10) (0xB0, 0x36) : The downloaded FLASH data length is not a multiple of 16;
 - (11) (0xB0, 0x37) : Programming FLASH fails;
 - (12) (0xB0, 0x3F) : Failed to update the management information;

2.2.7 CMD_DATA_CRC_CHECK

This command is used to check whether the downloaded data is correct. Considering the download speed and low probability of download failure, the CRC check is performed after the downloaded data is complete. The upper-layer command must provide the CRC value, start address, and check length of the downloaded data. Before CRC verification, CMD_KEY_RNG is used to obtain a random number.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x32 Level-1 command field							
1(CMD_L)	Level-2 command field: Parity partition number							
2~3(LEN)	Length of data sent							
4~7(Par)	32-bit CRC check value							
8~23(DAT)	DAT[0~15] : 16 bytes Key authentication value of USER1/3 partition authentication							
24~27(DAT)	DAT[16 to 19] : Check start address							
28~31(DAT)	DAT[20~23] : Check length (unit: byte, minimum length 2KB)							

- CMD_L: Check partition number

- 1) 0x00 = USER1;
 - 2) 0x02 = USER3;
- LEN Send data length: 0x18(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$;
 - Par [0~3] : 32 bit CRC check value, the synthetic rules for $CRC32 = Par[0] \mid Par[1] \ll 8 \mid Par[2] \ll 16 \mid Par[3] \ll 24$;
 - DAT[0:15] : key authentication value for partition authentication;
 - DAT [] 16 ~ 19: check the starting address, the synthesis rules to $Address = DAT[16] \mid DAT[17] \ll 8 \mid DAT[18] \ll 16 \mid DAT[19] \ll 24$, the Address is only within the scope of the FLASH;
 - DAT [20 to 23] : check length, its synthesis rules for $CRC_LEN = DAT[20] \mid DAT[21] \ll 8 \mid DAT[22] \ll 16 \mid DAT[23] \ll 24$, CRC_LEN is only within the effective range, length is larger than 2 KB, and is a multiple of 16;

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x32 Level-1 command field							
1(CMD_L)	Level-2 command field: Parity partition number							
2~3(LEN)	Length of data sent							
(DAT)	None							
4(CR1)	Status byte 1							
5(CR2)	Status byte 2							

- LEN Send data length: 0x00(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 1. Check succeeded: status flag bit (0xA0, 0x00).
 2. Check failure: status flag bits (CR1, CR2)
 - (1) (0xB0, 0x00) : Return failure;
 - (2) (0xB0, 0x20) : CRC check key authentication fails.
 - (3) (0xB0, 0x21) : The number of CRC check key authentication failures exceeds the limit.
 - (4) (0xB0, 0x32) : CRC check addresses are protected by partitions;
 - (5) (0xB0, 0x33) : CRC check address range is across partitions;
 - (6) (0xB0, 0x34) : CRC check address range is out of bounds (Refers to exceeding the entire FLASH size);
 - (7) (0xB0, 0x35) : CRC check address is not 16-byte alignment;
 - (8) (0xB0, 0x36) : the CRC check length is not a multiple of 16, or the CRC check length is less than 2KB.
 - (9) (0xB0, 0x38) : CRC check fails;

(10) (0xB0, 0x3F) : Failed to update the management information;

2.2.8 CMD_OPT_RW

This command is used for option byte read and write (including read protection level, FLASH page write protection, Data0/1 configuration, and USER configuration). When a partition is configured, BOOT does not allow you to change the read protection level from L1 to L0. This will cause mass erase in the user area.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x40 Level-1 command field							
1(CMD_L)	Level-2 command field							
2~3(LEN)	Length of data sent							
4~7(Par)								
8~23(DAT)	Option byte configures 16 bytes							

- CMD_L Level-2 command field:
 - 1) 0x00: Gets option bytes.
 - 2) 0x01: Configuration option byte.
 - 3) 0x02: Configuration option byte, reset again.
- LEN Send data length: 0x10(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- DAT[0~15] : Option bytes configures 16 bytes
 - RDP, nRDP, USER, nUSER, Data0, nData0, Data1, nData1, WRP0, nWRP0, WRP1, nWRP1, RDP2, nRDP2, USER2, nUSER2;
 - 1) CMD_L = 0x00: all values are 0x00.
 - 2) CMD_L = 0x01/0x02: Configuration option bytes are the values to be written.

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x40 Level-1 command field							
1(CMD_L)	Level-2 command field							
2~3(LEN)	Length of data sent							
4~19(DAT)	Option byte configures 16 bytes							
20(CR1)	Status byte 1							
21(CR2)	Status byte 2							

- LEN Send data length: 0x10(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- DAT[0~15] : The current option byte configuration is 16 bytes
 - RDP, nRDP, USER, nUSER, Data0, nData0, Data1, nData1, WRP0, nWRP0, WRP1, nWRP1, RDP2, nRDP2, USER2, nUSER2;
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 1. Return success: status flag bit (0xA0, 0x00).
 2. Check failure: status flag bits (CR1, CR2)
 - 1) (0xB0, 0x00) : return failure;
 - 2) (0xB0, 0x39) : Partitions have been configured and the read protection level cannot be reduced from L1 to L0.

2.2.9 CMD_USERX_OP

This command is used to read or configure the USER1/3 partition size. After the partition configuration is complete, the corresponding partition is automatically enabled and sealed. The USER1/3 partition size can be configured only once.

The recommended configuration process is as follows:

1. If you need to divide two areas, configure USER3 (automatic sealing is complete). If you want to also seal USER1, configure USER1 again. The size of USER1 + USER3 must be the size of the entire FLASH;

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x41 Level-1 command field							
1(CMD_L)	Level-2 command field							
2~3(LEN)	Length of data sent							
4~7(Par)	Par[0] : Partition USER1/3							
	Par [1] : Partition USER1/3 size							
	Par [2] : Partition authentication key index ID							
	Par [3] : Partition authentication and encryption download enable configuration							
DAT	None							

- CMD_L Level-2 command field:
 - 1) 0x00: Read partition USER1/3 size configuration.
 - 2) 0x01: Configure partition USER1/3 size, key ID, partition authentication/encrypted download enable.
- LEN Send data length: 0x00(LEN[0]), 0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$.
- Par[0] : Partition number

1) 0x00: partition USER1.

2) 0x02: partition USER3.

● Par [1] :

1) CMD_L = 0x00:0x00.

2) CMD_L = 0x01: partition USER1/3 size configuration

Input range for partition size: 0x1(2KB)... 0x07(14KB), 0x20(64KB), USER1+ USER3 = 64KB; The user area USER1/3 size is automatically sealed after configuration.

Partition size and address determined

The start address of the partition is 0x0800_0000, and the end address of the partition is the start address plus the total FLASH capacity (for example, if the FLASH capacity is 64K, the end address is 0x0800_0000 + 64/2*0x800 = 0x0800_FFFF).

If USER1 is partitioned, the partition address of USER1 ranges from 0x0800_0000 ~ (0x0800_0000 + USER1_Size*0x800).

If USER3 is partitioned, the partition address of USER3 ranges from (0x0801_0000 – USER3_Size*0x800) ~ 0x0800_FFFF (For example, the end address of FLASH is 0x0800_FFFF).

● Par [2] :

1) CMD_L = 0x00:0xFF.

2) CMD_L = 0x01:0x00~0x01 Encrypted download/Partition authentication key index ID, 0xFF indicates that the index ID is not configured. If the corresponding USERX is not configured with an ID, the value of Par[3] is not judged.

● Par [3] :

Enable configuration of partition authentication and encrypted download, 0xXY

■ X = 0 - Disable partition authentication, can be configured to 1.

■ X = 1 - Enable partition authentication, cannot be set to 0.

■ Y = 0 - Disable encrypted download, can be configured to 1.

■ Y = 1 - Enable encrypted download, cannot be set to 0.

1) CMD_L = 0x00: read status, retain value 0x00;

2) CMD_L = 0x01: configuration status, configuration value 0xXY;

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x41 Level-1 command field							
1(CMD_L)	Level-2 command field							
2~3(LEN)	Length of data sent							

4~7(DAT)	DAT[0] : partition USER1/3
	DAT[1] : partition USER1/3 size
	DAT [2] : partition authentication key index ID configuration status
	DAT [3] : Read partition authentication and encryption download enable configuration
8(CR1)	Status byte 1
9(CR2)	Status byte 2

- LEN Send data length: 0x02(LEN[0]), 0x00(LEN[1]), LEN = LEN[0] + (LEN[1]<<8).
- DAT[0] : Partition number
 - 1) 0x00: partition USER1.
 - 2) 0x02: partition USER3.
- DAT[1] : Read the current partition USER1/3 size
 Partition size output range: 0x0(0KB), 0x1(2KB)... 0x07 (14 KB), 0 x20 (64 KB).
 0x0 indicates that the partition size is not configured. USER1 + USER3 = 64KB.
- DAT [2].
 - 1) 0x00, the ID has been configured.
 - 2) 0xFF, the ID is not configured
- DAT [3] :
 Read partition authentication and encryption download enable configuration, 0xXY
 - X = 0 - Disable partition authentication, can be configured to 1.
 - X = 1 - Enable partition authentication, cannot be set to 0.
 - Y = 0 - Disable encrypted download, can be configured to 1.
 - Y = 1 - Enable encrypted download, cannot be set to 0.
- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 - 1) Return success: status flag bit (0xA0, 0x00).
 - 2) Return failure: status flag bit (0x70, 0x00)
 - (1) (0xB0, 0x00) : Return failure;
 - (2) (0xB0, 0x10) : The key index ID range is incorrect;
 - (3) (0xB0, 0x3A) : The partition size has been configured and cannot be configured again.
 - (4) (0xB0, 0x3B) : The partition size is incorrectly configured, USER1 + USER3 = FLASH capacity, and the minimum configuration of USER1/3 is 0x01 (2KB).
 - (5) (0xB0, 0x3D) : The partition key index ID fails to be configured or has been configured.
 - (6) (0xB0, 0x3E) : The configuration of partition authentication and encryption download fails or has

been configured.

(7) (0xB0, 0x3F) : Failed to update the management information;

2.2.10 CMD_SYS_RESET

This command is used to reset the BOOT program.

Upper-level instructions:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x50 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3(LEN)	Length of data sent							
4~7(Par)	Reserved							
(DAT)	None							

- Reserved value: 0x00;

Lower layer response:

Byte \ Bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x50 Level-1 command field							
1(CMD_L)	0x00 Level-2 command field							
2~3(LEN)	Length of data sent							
(DAT)	None							
4(CR1)	Status byte 1							
5(CR2)	Status byte 2							

- Status bytes (CR1 and CR2) are divided into the following types according to command execution:
 - 1) Return success: status flag bit (0xA0, 0x00).
 - 2) Return failure: status flag bits (0xB0, 0x00).

2.3 Returns the status word description

2.3.1 Returns the success status word

Return success: status flag bit (0xA0, 0x00). Indicates that the command issued by the upper layer is executed successfully, and returns a success status word.

Contains the success return value of the read, update, configure, and other commands.

2.3.2 Returns the failure status word

Return failure: status flag bits (0xB0, 0x00). Indicates that the command issued by the upper layer fails to execute due to other reasons (command acceptance format error or timeout, etc.), and the failure status word is returned.

2.3.3 Return other status words

The following return status words also return failure. The second byte status word indicates a different error type.

- (1) (0xB0, 0x10) : Key index id range error;
- (2) (0xB0, 0x11) : New key CRC check error;
- (3) (0xB0, 0x20) : Key authentication fails.
- (4) (0xB0, 0x21) : The number of key authentication failures exceeds the limit (The maximum number of key authentication failures is 16, and the number of times the two partitions are shared).
- (5) (0xB0, 0x30) : Erase/download FLASH page protected by RDP;
- (6) (0xB0, 0x31) : Erased/downloaded FLASH page is protected by WRP;
- (7) (0xB0, 0x32) : Erase/download /CRC check address is protected by partition;
- (8) (0xB0, 0x33) : erase/download /CRC check address range across partitions;
- (9) (0xB0, 0x34) : The address range of erase/download /CRC is out of bounds (refers to the size of the entire flash);
- (10) (0xB0, 0x35) : The start address of erase/download /CRC is not 16 bytes aligned;
- (11) (0xB0, 0x36) : The length of the downloaded /CRC data is not a multiple of 16. Data length indicates the length of erasing flash, or the length of downloading code to FLASH, or the length of checking FLASH CRC values;
- (12) (0xB0, 0x37) : Failed to erase/download FLASH programming;
- (13) (0xB0, 0x38) : CRC check failed.
- (14) (0xB0, 0x39) : Partitions have been configured and the read protection level cannot be changed from L1 to L0.
- (15) (0xB0, 0x3A) : The partition has been configured and cannot be configured again.
- (16) (0xB0, 0x3B) : Partition size configuration error, must satisfy $USER1 + USER3 = \text{FLASH capacity}$.
- (17) (0xB0, 0x3E) : The configuration of partition authentication and encryption download fails or has been configured.
- (18) (0xB0, 0x3F) : Failed to update the management information;
- (19) (0xBB, 0xCC) : Upper-layer send commands Level-1 and level-2 command fields do not belong to any command.

3 BOOT Instructions

3.1 Upper computer control process

Upper computer support user erasing FLASH area, user code download, download code integrity check. By reading partition information, the upper computer automatically identifies the address range of erasing, downloading and checking entered by the user and requires authentication.

The upper computer supports users to choose whether to enable encryption download to protect user code.

The upper computer supports the user to read and configure the partition USER1/3 size. The partition size cannot be changed after being configured.

The upper computer supports users to update the security key (used for partition authentication and encryption download).

The upper computer supports user update option byte reading and modification.

Enter BOOT: Enter BOOT, you can interact with PC TOOL through USART1 interface at this time.

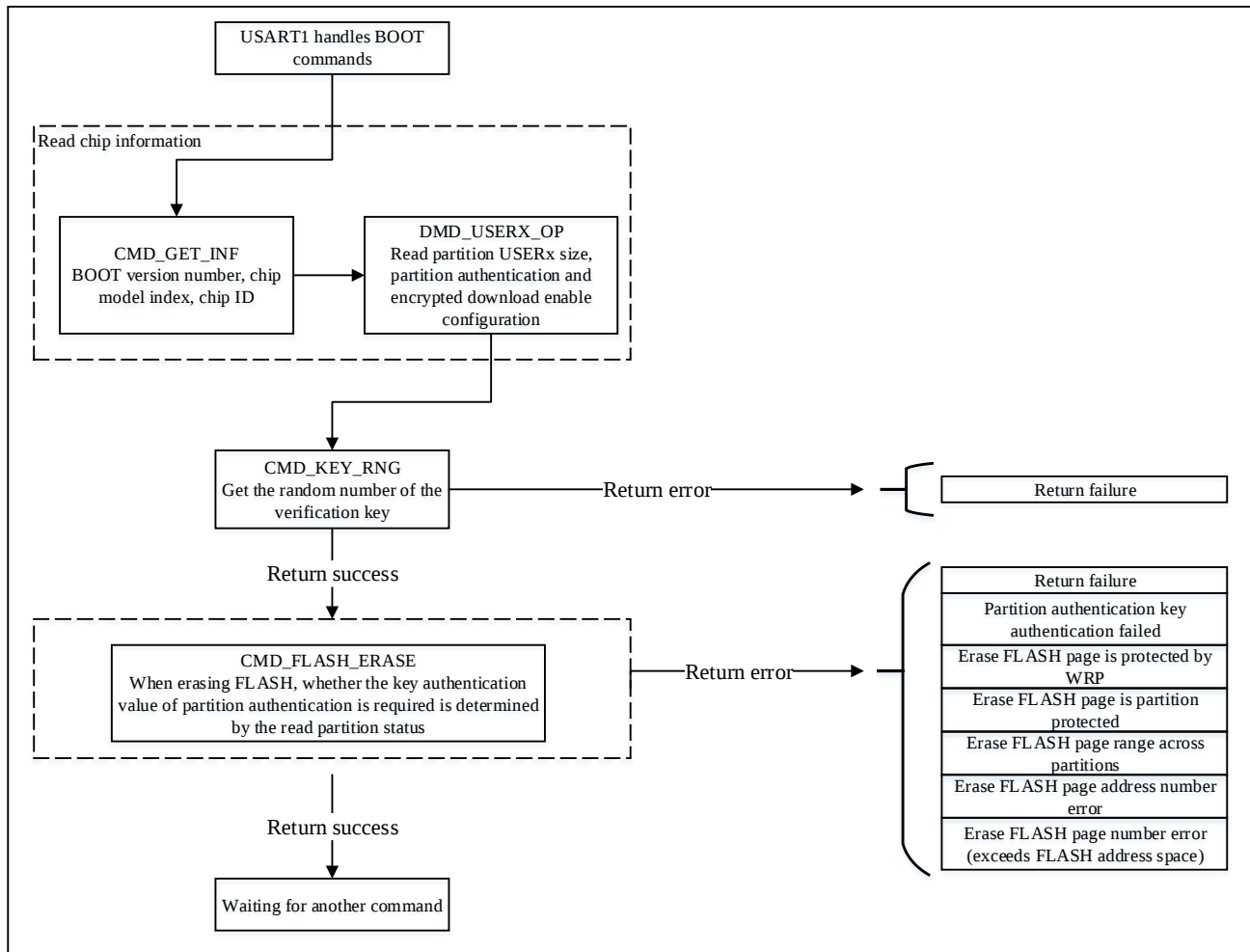
Chip firmware integrity check: If you choose to start from the system storage area, BOOT automatically performs integrity self-checking. If the check fails, it will enter an infinite loop, and subsequent functions cannot be used..

Command set interaction: The PC TOOL sends different commands based on the command set supported by the BOOT to use corresponding functions.

- 1) Read BOOT version number, chip model index, chip ID;
- 2) Get 16byte random number;
- 3) Update the security key (for partition authentication and encrypted download);
- 4) Erase FLASH;
- 5) Download user programs to FLASH;
- 6) CRC check downloaded user program;
- 7) Read/configure option bytes (including read protection level, FLASH page write protection, Data0/1 configuration, USER configuration);
- 8) Get partition USERX size, set partition USERX size;
- 9) System reset, you can reset the BOOT program to run again;

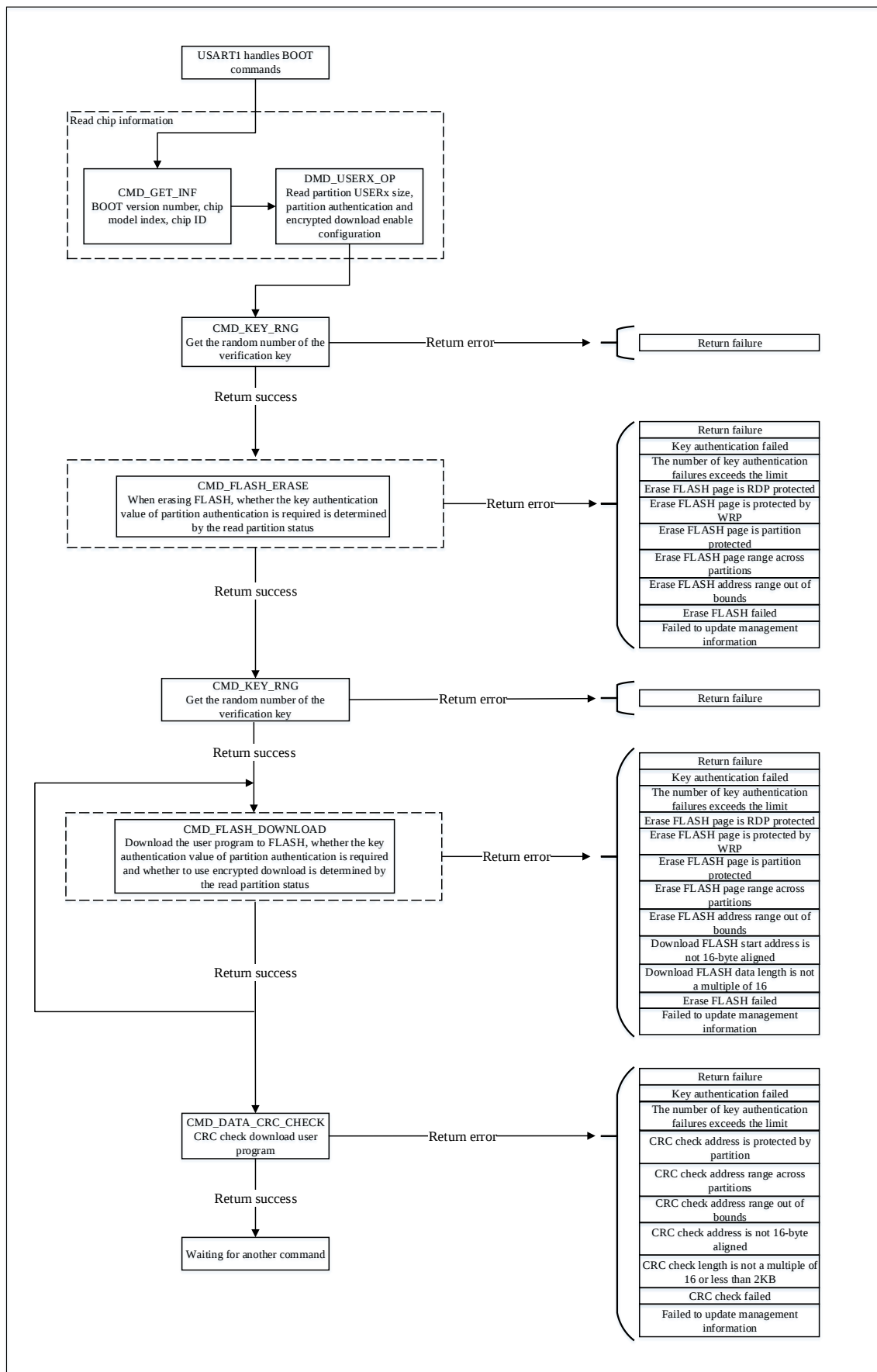
3.1.1 Erase command control flow chart

Figure 3-1 Flowchart of erasing command control



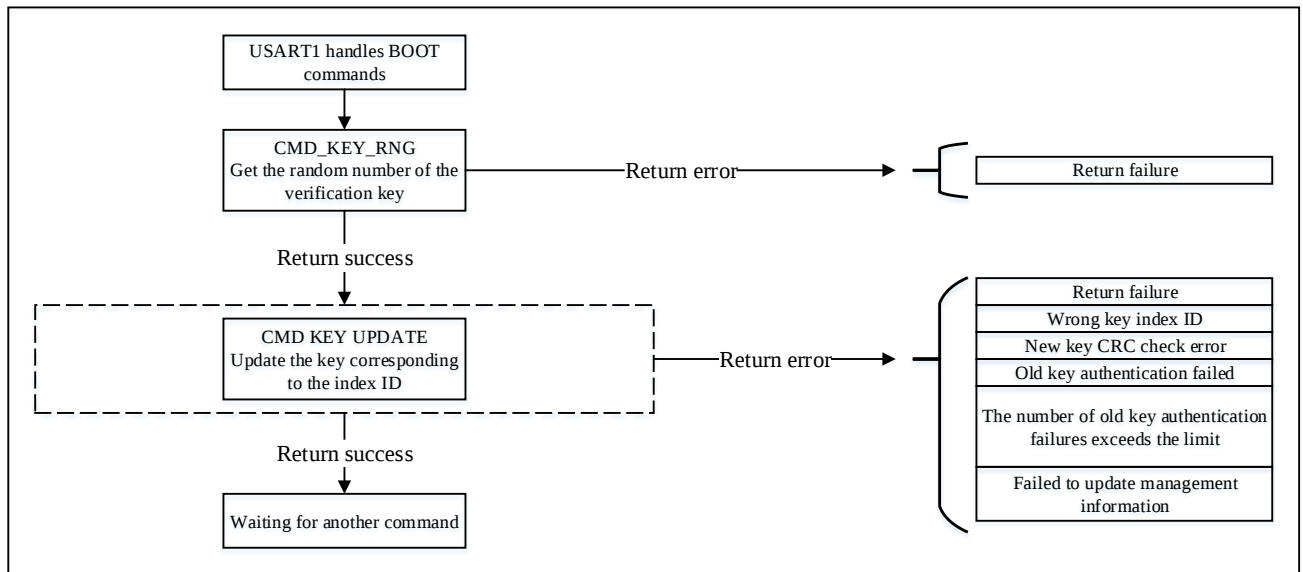
3.1.2 Download the command control flow chart

Partition authentication encryption obtains a random number before downloading, and the host computer uses this random number to generate the key authentication value of 16-byte USER1/3 partition authentication. In the case of continuous download, the random number used in the subsequent download command is generated by the random number deriving algorithm of the first time instead of obtaining a new random number.

Figure 3-2 Flowchart for downloading command control


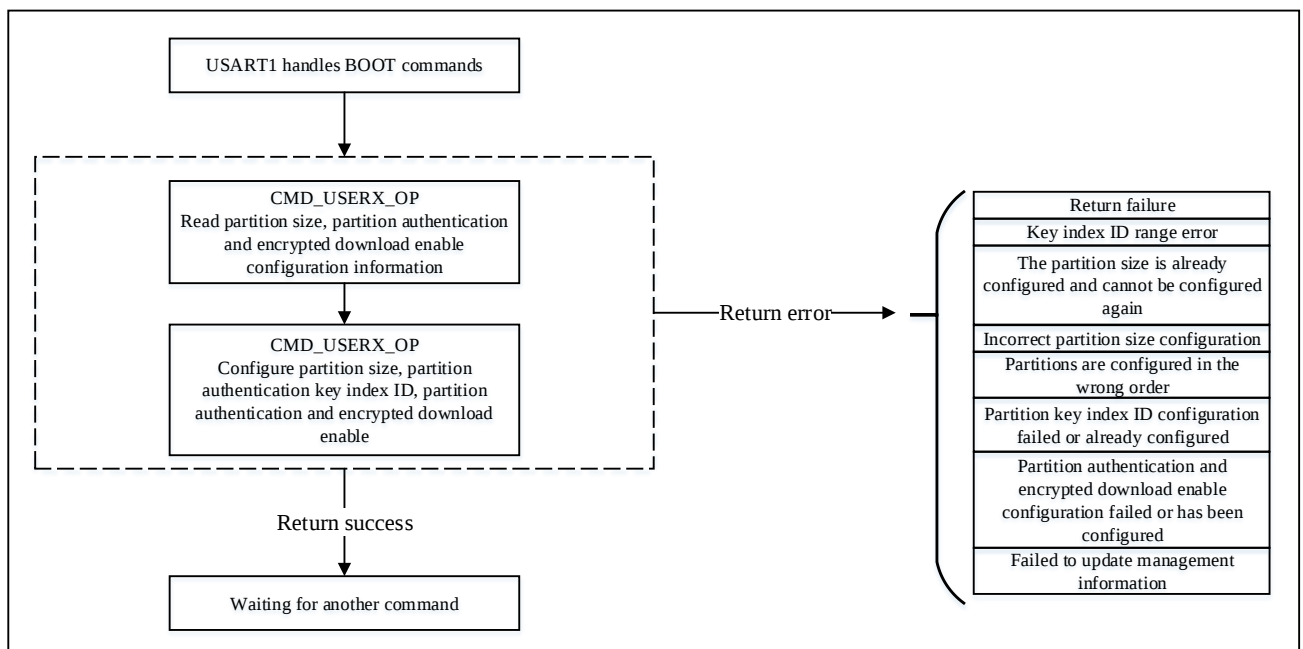
3.1.3 Update the key command control flow chart

Figure 3-3 Command control flowchart for updating a key



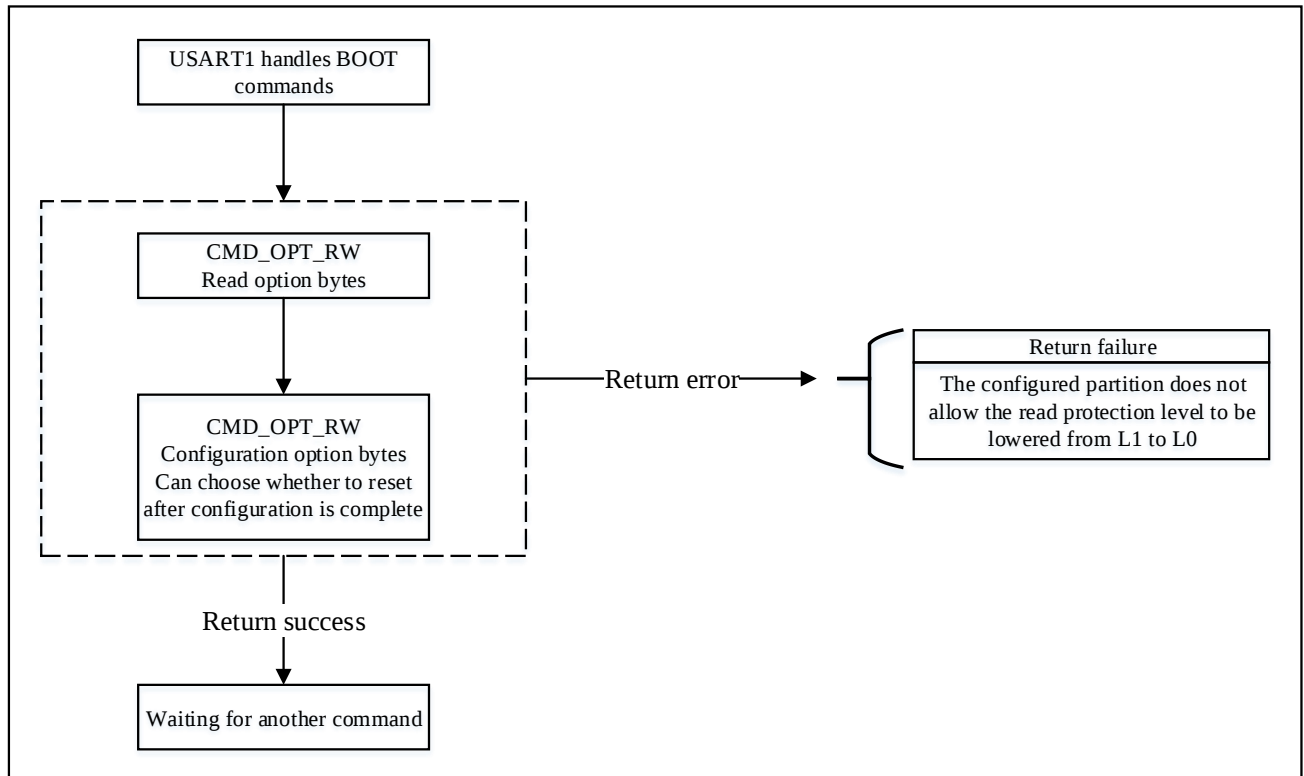
3.1.4 Partition operation command control flow chart

Figure 3-4 Partition operation command control flow chart



3.1.5 Option byte read/write command control flow chart

Figure 3-5 Option byte read/write command control flow chart



4 Version history

Version	The revision date	Remark
V1.0	2022/3/24	Initial release

5 Notice

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD. (Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to NSING Technologies Inc. and NSING Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections, enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NSING has attempted to provide accurate and reliable information, NSING assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NSING be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product. NSING Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, Insecure Usage'. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NSING and hold NSING harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NSING, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.