

应用笔记

N32A003系列IAP应用笔记

简介

本应用笔记提供基于 N32A003 系列芯片的二级 BOOT 及 APP 工程, 并介绍如何通过 NSING 下载工具访问 BOOT。

目 录

1	概述.....	3
2	软件实现流程.....	4
3	NSING 下载工具使用	6
3.1	环境准备及代码下载.....	6
3.2	下载工具操作展示	7
4	BOOT 流程及命令处理	10
4.1	数据结构.....	10
4.2	命令说明.....	11
4.3	返回状态字说明	22
5	历史版本.....	24
6	声明.....	25

1 概述

N32A003 系列产品不支持系统 BOOT，但芯片出厂会默认在 FLASH 用户区前 3K 放置二级 BOOT，可用 SWD 接口直接擦除，由用户选择是否保留。本应用笔记提供二级 BOOT 及 APP 代码工程，可通过 NSING 下载工具访问二级 BOOT 完成以下功能：

- ◆ 支持 UART（PA2-TX、PA3-RX，初始波特率默认为 9600，支持波特率指令设置，支持的波特率 4800、9600、14400、19200、38400、57600、115200、128000、256000、576000、912600、923076）；
- ◆ 支持下载数据 CRC16($X^{16} + X^{15} + X^2 + 1$)校验；
- ◆ 支持明文下载；
- ◆ 支持芯片信息读取
- ◆ 支持 FLASH 数据读取
- ◆ 选项字节读取及修改
- ◆ 支持软件复位芯片操作

本文档详细介绍了工程使用、下载工具使用及 BOOT 指令等内容。

2 软件实现流程

以 N32A003F5Q7 芯片为例，FLASH 存储区最大为 29.5KB，其中前 3KB 作为 BOOT 存储区，其他区域作为 APP 区，详细划分如下：

功能	地址范围	SIZE(byte)	备注
BOOT 区	0x08000000~0x08000BFF	3K	提供下载、选项字节读写等功能
BOOT CRC 标志	0x08000BFC~0x08000BFF	4	在 BOOT 区中
APP 区	0x08000C00~0x080073FF	26K	在 APP 中闪烁 LED 灯
跳转标志区	0x08007400~0x080075FF	0.5K	最后一页，用于修改跳转标志
跳转标志	0x080075F8~0x080075FB	4	在跳转标志区中
跳转标志反码	0x080075FC~0x080075FF	4	在跳转标志区中

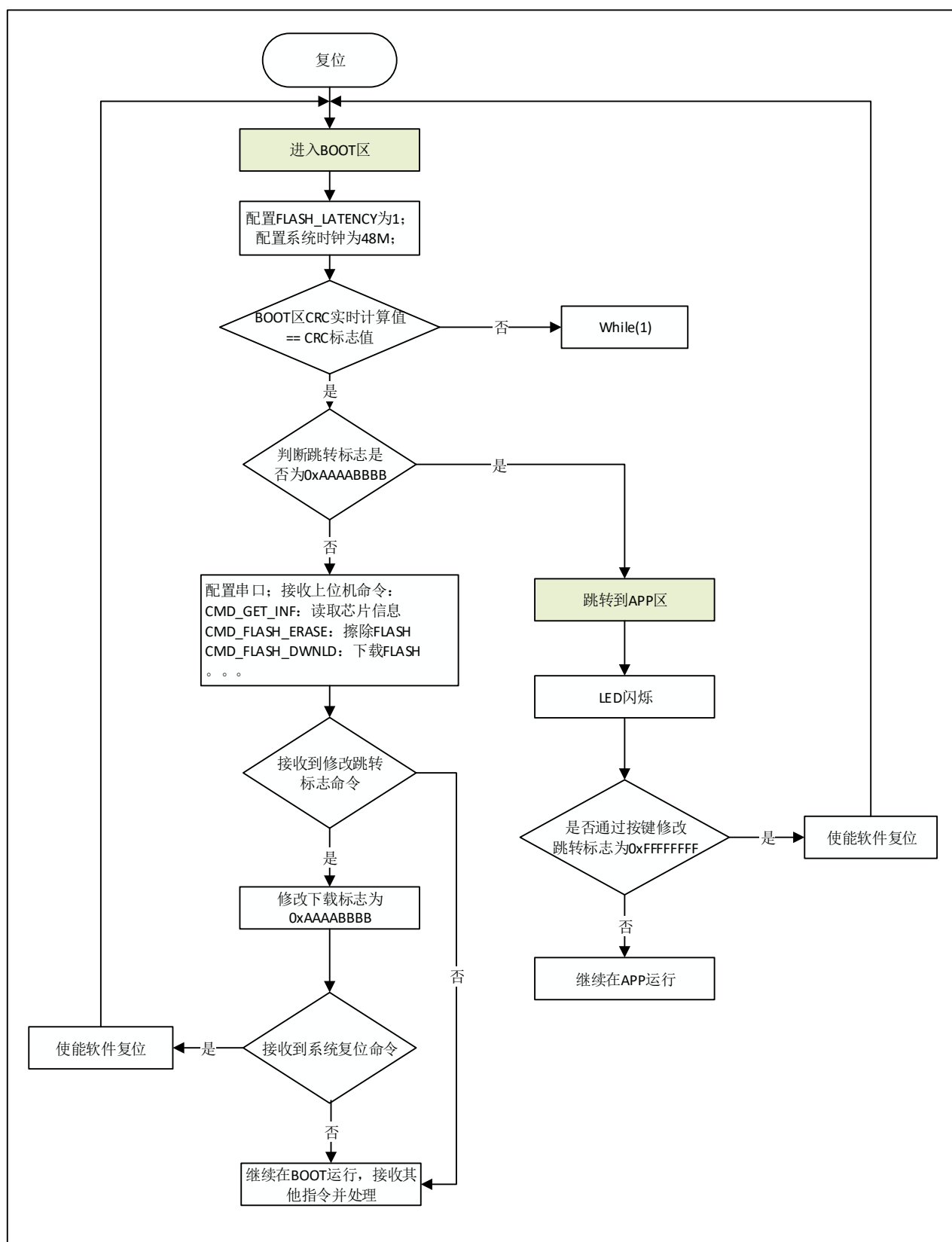
注：BOOT 大小只有 3K，部分函数已优化，若用户需要自行添加代码请注意不要超出范围

注：由于修改 FLASH 中的跳转标志需要页擦，因此 APP 最后一页用户不可用

BOOT 及 APP 代码实现流程如下：

1. 复位后先进 BOOT 区，对时钟进行初始化，对 BOOT 区进行 CRC 计算并将结果与 CRC 标志对比，一致才继续后面的流程
2. 判断跳转标志位，若为 0xAAAABBBB 则直接跳转到 APP 用户程序，进入步骤 3；若不为 0xAAAABBBB 则继续留在 BOOT，进入步骤 4
3. LED1 灯闪烁表示 APP 程序正常运行，当用户按下 KEY1(PA11)则修改跳转标志位为 0xFFFFFFFF，复位后重新从步骤 1 开始
4. 通过 UART1 接收指令，完成波特率修改、芯片信息获取、FLASH 擦除下载、复位等功能，若接收到修改跳转标志指令则将跳转标志修改为 0xAAAABBBB，复位后重新从步骤 1 开始

流程图如下：



3 NSING 下载工具使用

下面介绍如何通过 NSING 下载工具操作 BOOT:

3.1 环境准备及代码下载

1. 提前准备好以下环境:

◆ BOOT 及 APP 工程

APP	2024/7/31 16:21	文件夹
Uart_bootloader	2024/7/31 17:27	文件夹

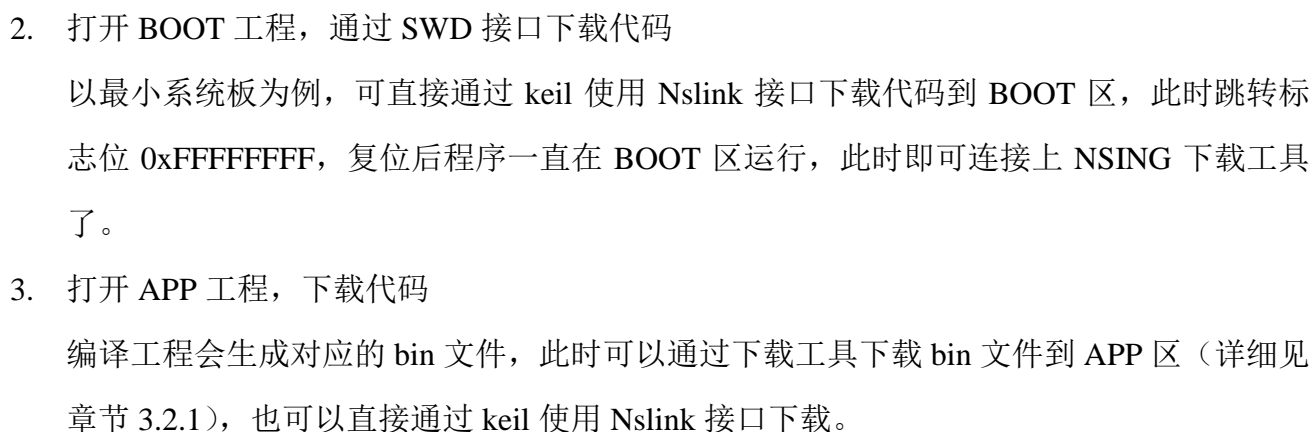
◆ V1.3.2 或以上版本的 NSING 下载工具

Devices	2022/12/15 14:07	文件夹	
doc	2022/12/15 14:07	文件夹	
driver	2022/12/15 14:07	文件夹	
ErrorLog	2023/1/10 14:59	文件夹	
FLM	2023/1/10 14:59	文件夹	
Languages	2023/1/11 17:44	文件夹	
TestLog	2022/12/15 14:08	文件夹	
CMSIS_DAP.dll	2021/5/12 16:01	应用程序扩展	260 KB
JLinkARM.dll	2021/6/25 15:24	应用程序扩展	16,258 KB
JLinkDevices.xml	2023/1/13 17:32	XML 源文件	247 KB
Language.ini	2022/7/6 11:33	配置设置	1 KB
LanguageLib.dll	2021/6/25 15:24	应用程序扩展	224 KB
NZDownloadTool.exe	2023/3/6 15:13	应用程序	2,403 KB
Series CFG Table.ini	2023/1/5 18:45	配置设置	21 KB
SkinH.dll	2010/8/15 16:08	应用程序扩展	87 KB
skin.hshe	2021/6/25 15:24	SHE 文件	21 KB
SysConfig.ini	2024/8/27 16:50	配置设置	1 KB

◆ 串口工具

◆ N32A003 最小系统板

根据下图连接串口工具（PA2 - UART_TX、PA3 - UART_RX）到 PC，并连接 RESET 按键、KEY1 按键、LED1、SWD、电源等跳线帽，连接 Nslink 接口



3.2 下载工具操作展示

3.2.1 工具连接

打开“NZDownloadTool.exe”，选择“USART”接口及波特率，选择串口对应的 COM 好，点击连

接设备，成功连接后会显示芯片信息（举例测试使用的芯片为非正式版本）：



3.2.1 APP 代码下载

填写起始地址（APP 起始地址固定为 0x08000C00），选择 APP 工程编译出的 bin 文件，勾选下载完成后写下载标志（将跳转标志写为 0xAAAABBBB），点击下载。



3.2.1 选项字节操作

选择常用操作，选择配置选项字节，可以看到所有选项字节内容，用户可根据需求修改。其中我们提供了 BOOT_LOCK 功能，勾选了以后可保护 FLASH 前 3K（0x08000000~0x08000BFF）代码不被修改，全擦时会跳过前 3K。



3.2.1 复位

下载完 APP 代码后通过复位跳转至 APP；配置完选项字节后通过复位使修改生效。复位操作可通过下载工具完成，也可直接按 RESET 按键。

4 BOOT 流程及命令处理

下面阐述相关命令数据结构及处理流程。

4.1 数据结构

这里介绍下文阐述中的一些约定，其中，“<>”代表必须包含的字段，“()”代表根据参数不同包含的字段。

串口支持的命令数据结构：

1、上位机下发上层指令：

$\text{STA1} + \text{STA2} + \{\text{上层指令结构}\} + \text{XOR}$ 。

STA1 和 STA2 是串口发送命令的起始字节，STA1=0xAA，STA2=0x55。用于芯片识别上位机发送串口数据流。

XOR 代表之前命令字节的异或运算值 ($\text{STA1} + \text{STA2} + \{\text{上层指令结构}\}$)。

2、上位机接收下层应答：

$\text{STA1} + \text{STA2} + \{\text{下层应答结构}\} + \text{XOR}$ 。

STA1 和 STA2 是串口发送命令的起始字节，STA1=0xAA，STA2=0x55。用于上位机识别芯片发送串口数据流

XOR 代表之前命令字节的异或运算值 ($\text{STA1} + \text{STA2} + \{\text{下层应答结构}\}$)。

上下层指令数据结构：

1、上层指令结构：

$\langle \text{CMD_H} + \text{CMD_L} + \text{LEN} + \text{Par} \rangle + (\text{DAT})$ 。

CMD_H 代表一级命令字段，CMD_L 代表二级命令字段；LEN 代表发送数据长度；Par 代表 4 个字节命令参数；DAT 代表上层指令往下层发送的具体数据；

2、下层应答结构：

$\langle \text{CMD_H} + \text{CMD_L} + \text{LEN} \rangle + (\text{DAT}) + \langle \text{CR1} + \text{CR2} \rangle$ 。

CMD_H 代表一级命令字段，CMD_L 代表二级命令字段，下层的命令字段和对应上层的命令字段相同；LEN 代表发送数据长度；DAT 代表下层向上层应答的具体数据；CR1+CR2 代表向上层返回的指令执行结果，若上层发送命令一级、二级命令字段不属于任何命令，BOOT

回复 CR1=0xBB, CR2 = 0xCC。

举例：

若需配置 CMD_SYS_RESET 命令，则工具下发：

AA 55 50 00 00 00 00 00 00 AF

芯片应答：

AA 55 50 00 00 00 A0 00 0F

4.2 命令说明

BOOT 总共包含以下命令：

命令名称	键值	说明
CMD_SET_BR	0x01	设置串口波特率（仅使用串口时有效）
CMD_GET_INF	0x10	读取芯片型号索引、BOOT 版本号、芯片 ID
CMD_FLASH_ERASE	0x30	擦除 FLASH
CMD_FLASH_DWNLD	0x31	下载用户程序到 FLASH，以及写跳转标志
CMD_DATA_CRC_CHECK	0x32	CRC 校验下载用户程序
CMD_DATA_READ	0x33	从 FLASH 读取数据
CMD_OPT_RW	0x40	读取/配置选项字节（包含了读保护等级 Data0/1 配置、USER 配置）
CMD_SYS_RESET	0x50	系统复位
CMD_APP_GO	0x51	跳转到用户区执行程序

下面对各个命令的发送及接收进行详细说明。

4.2.1 CMD_SET_BR

该命令用于修改串口波特率。

上层指令：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x01 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度：0x00,0x00							
4~7(Par)	Par[0~3]：设置波特率参数							

(DAT)	无
-------	---

- Par[0~3], 串口波特率协商设置值:

Par[0~3]	切换指定的波特率(bps)
0x000E15C4	923076
0x000DECD8	912600
0x0008CA00	576000
0x0003E800	256000
0x0001F400	128000
0x0001C200	115200
0x0000E100	57600
0x00009600	38400
0x00004B00	19200
0x00003840	14400
0x00002580	9600
0x000012C0	4800

- 保留值: 0x00;

底层应答:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x01 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度: 0x00,0x00							
(DAT)	无							
4(CR1)	状态字节 1							
5(CR2)	状态字节 2							

- 状态字节(CR1、CR2)根据命令执行情况分为:
 1. 返回成功: 状态标志位(0xA0、0x00)。
 2. 返回失败: 状态标志位(0xB0、0x00)。

4.2.2 CMD_GET_INF

该命令提供的功能是读取 BOOT 版本号信息。该命令提供的功能是读取芯片型号、命令集版本号、BOOT 版本号、UID/UCID/DBG_MCUID 共 6 种信息。上层指令：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x10 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3 (LEN)	发送数据长度							
4~7(Par)	保留为 0x00, 0x00, 0x00, 0x00.							
(DAT)	无							

- 保留值：0x00。
- LEN 发送数据长度：0x00(LEN[0])、0x00(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$ 。

底层应答：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x10 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3 (LEN)	数据长度							
4~38(DAT)	芯片型号、命令集版本号、BOOT 版本号、UID/UCID/DBG_MCUID							
39(CR1)	状态字节 1							
40(CR2)	状态字节 2							

- 过程字节(CMD_H)和上层指令中的(CMD_H)对应。
- LEN 是数据长度：0x33(LEN[0])、0x00(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$ 。
- DAT[0]: 0x07, 芯片型号，指示芯片型号 N32A003 系列
- DAT[1]: 0x12, 命令集版本号，指示命令集版本为 V1.2
- DAT[2]: 0x10, BOOT 版本号，指示 BOOT 版本为 V1.0。
- DAT[3~50]: 48 字节

其中: DAT[3~18]: 16-Byte UCID（详细内容参考用户手册）

DAT[19~30]: 12-Byte UID（详细内容参考用户手册）

DAT[31~34]: 4-Byte DBG_MCUID (详细内容参考用户手册)

- 状态字节(CR1、CR2)根据命令执行情况分为:

1. 返回成功: 状态标志位(0xA0、0x00)。
2. 返回失败: 状态标志位(0xB0、0x00)。

4.2.3 CMD_FLASH_ERASE

BOOT 提供以页为单位擦除 FLASH 的功能, 擦除页地址编号和和页数由用户提供, 擦除的 FLASH 空间不能超过整个 FLASH 空间, 且至少擦除 1 个页(512Byte)。

上层指令:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x30 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度(0)							
4~7(Par)	页地址编号 2 字节: 0~255 页数 2 字节:1~256							
8~23(DAT)	无							

- CMD_L: 0x00
- LEN 发送数据长度: 0x10(LEN[0])、0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$ 。
- 擦除地址和范围由 Par 字段中的 4 个字节构成

Par[0~1]: 页地址编号 2 字节(0~255)

页地址编号 = Par [0] + Par [1] << 8;

Par[2~3]: 页数 2 字节(1~256)

页数 = Par [2] + Par [3] << 8;

0 号页首地址为 0x0800_0C00, 以后的页地址编号加 1, 首地址累加 0x200。

比如:

1 号页首地址为 $0x0800_0C00 + 1 * 0x200 = 0x0800_0E00$

2 号页首地址为 $0x0800_0C00 + 2 * 0x200 = 0x0800_1000$

则当: 页地址编号为 m, 页数为 n

那么擦除的地址范围：

$$(0x0800_0C00 + m*0x200) \sim (0x0800_0C00 + m*0x200 + n*0x200)$$

即（页地址编号的首地址） ~ （页地址编号的首地址 + 页数*页的大小）

底层应答：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x30 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
(DAT)	无							
4(CR1)	状态字节 1							
5(CR2)	状态字节 2							

- LEN 发送数据长度：0x00(LEN[0])、0x00(LEN[1])，LEN = LEN[0] + (LEN[1]<<8)。
- 状态字节(CR1、CR2)根据命令执行情况分为：
 1. 返回成功：状态标志位(0xA0、0x00)。
 2. 返回失败：状态标志位(CR1, CR2)。
 - (1)、(0xB0、0x00)：返回失败；
 - (2)、(0xB0、0x30)：擦除 FLASH 页被 RDP 保护；
 - (3)、(0xB0、0x34)：擦除 FLASH 地址范围越界（指超出整个 FLASH 大小）；
 - (4)、(0xB0、0x37)：擦除 FLASH 失败。

4.2.4 CMD_FLASH_DWNLD

当二级命令为 0x00：该命令提供用户下载代码到指定 FLASH 中。数据长度必须 16 字节对齐（不足上位机自动补 0x00），都由上层命令提供。明文下载。

当二级命令为 0xF0：该命令用于设置“跳转标志”为 0xAAAABBBB。

上层指令：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x31 一级命令字段							
1(CMD_L)	二级命令字段为 0x00：表示下载 FLASH 数据							

	二级命令字段为 0XF0：表示设置跳转标志
2~3(LEN)	发送数据长度
4~7(Par)	下载 FLASH 的起始地址
8~23+N(DAT)	DAT[0~15]：保留（0） DAT[16~16+N]：下载的具体数据 DAT[16+N+1~16+N+4]：数据的 4Byte CRC16 校验值

● CMD_L:

0x00：表示下载 FLASH 数据

0xF0：表示设置跳转标志

● LEN 发送数据长度：0xXX(LEN[0])、0xXX(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$

● Par[0~3]：下载 FLASH 的起始地址，合成规则为 $Address = Par[0] | Par[1] \ll 8 | Par[2] \ll 16 | Par[3] \ll 24$ 。

● DAT[16~16+N]：下载的具体数据

UART：最大 128 个字节， $15 \leq N \leq 127$ ，N+1 必须为 16 的倍数。

DAT[16+N+1~16+N+4]：数据的 4Byte CRC16 校验值

底层应答：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x31 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
(DAT)	无							
4(CR1)	状态字节 1							
5(CR2)	状态字节 2							

● LEN 发送数据长度：0x00(LEN[0])、0x00(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$ 。

● 状态字节(CR1、CR2)根据命令执行情况分为：

1. 下载成功：状态标志位(0xA0、0x00)。

2. 下载失败：状态标志位(CR1, CR2)。

(1)、(0xB0、0x00)：返回失败；

- (2)、(0xB0、0x30): 下载 FLASH 地址被 RDP 保护;
- (3)、(0xB0、0x34): 下载 FLASH 地址范围越界 (指超出整个 FLASH 大小);
- (4)、(0xB0、0x35): 下载 FLASH 起始地址不是 16 字节对齐;
- (5)、(0xB0、0x36): 下载 FLASH 数据长度不是 16 的倍数;
- (6)、(0xB0、0x37): 编程 FLASH 失败。

4.2.5 CMD_DATA_CRC_CHECK

该命令用于校验下载数据是否正确, 考虑到下载速度的因素和下载失败概率比较小, 所以采用数据下载完成后统一进行 CRC 校验, 上层指令需提供下载数据的 CRC 值和校验起始地址以及校验长度。

上层指令:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x32 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
4~7(Par)	32bit CRC16 校验值							
8~31(DAT)	DAT[0:15]: 保留 (0) DAT[16:19]: 校验起始地址 DAT[20:23]: 校验长度(单位: 字节, 长度最小 512B)							

- CMD_L: 0x00;
- LEN 发送数据长度: 0x18(LEN[0])、0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$ 。
- Par[0~3]: 32bit CRC16 校验值, 其合成规则为 $CRC16 = Par[0] \mid Par[1] \ll 8 \mid Par[2] \ll 16 \mid Par[3] \ll 24$ 。
- DAT[0:15]: 保留值, 16 字节全部为 0x00。
- DAT [16~19]: 校验起始地址, 其合成规则为 $Address = DAT[16] \mid DAT[17] \ll 8 \mid DAT[18] \ll 16 \mid DAT[19] \ll 24$, Address 只能是在 FLASH 范围内。
- DAT [20~23]: 校验长度, 其合成规则为 $CRC_LEN = DAT[20] \mid DAT[21] \ll 8 \mid DAT[22] \ll 16 \mid DAT[23] \ll 24$, CRC_LEN 只能是在有效范围内, 长度大于 0.5KB, 且是 16 的倍数。

底层应答：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x32 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
(DAT)	无							
4(CR1)	状态字节 1							
5(CR2)	状态字节 2							

- LEN 发送数据长度：0x00(LEN[0])、0x00(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$ 。
- 状态字节(CR1、CR2)根据命令执行情况分为：
 1. 校验成功：状态标志位(0xA0、0x00)。
 2. 校验失败：状态标志位(CR1, CR2)
 - (1)、(0xB0、0x00)：返回失败；
 - (4)、(0xB0、0x34)：CRC 校验地址范围越界（指超出整个 FLASH 大小）；
 - (5)、(0xB0、0x35)：CRC 校验地址不是 16 字节对齐；
 - (6)、(0xB0、0x36)：CRC 校验长度不是 16 的倍数，或者长度小于 512B；
 - (7)、(0xB0、0x38)：CRC 校验失败。

4.2.6 CMD_DATA_READ

该命令从 FLASH 空间读取数据。明文读取。

上层指令：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x33 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	DAT 数据长度 0x01							
4~7(Par)	FLASH 的起始地址							
8(DAT)	需要读取的数据长度							

- CMD_L 二级命令字段：0x0
- LEN 数据长度：0x01(LEN[0])、0x00(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$ 。
- Par[0~3]: 读取FLASH的起始地址，合成规则为 $Address = Par[0] | Par[1] \ll 8 | Par[2] \ll 16 | Par[3] \ll 24$ 。

- DAT[0]: 需要读取的数据长度

USART: 最大 128 个字节，最小 0 字节。

底层应答:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x33 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	DAT 数据长度							
3~3+N(DAT)	DAT[0~N]: 读取的具体数据							
4+N~8+N	数据的 4Byte CRC16 校验值							
9+N(CR1)	状态字节 1							
10+N(CR2)	状态字节 2							

- LEN 读取数据长度：0x00(LEN[0])、0x00(LEN[1])， $LEN = LEN[0] + (LEN[1] \ll 8)$ 。

- DAT[0~N]: 读取的具体数据

USART: 最大 128 个字节， $N \leq 127$ 。

- CRC32: 数据的 4 Byte CRC32 校验值
- 状态字节(CR1、CR2)根据命令执行情况分为:

1. 成功: 状态标志位(0xA0、0x00)。

2. 失败: 状态标志位(CR1, CR2)。

(1)、(0xB0、0x00): 返回失败;

(2)、(0xB0、0x34): 读取的 FLASH 范围地址越界;

4.2.7 CMD_OPT_RW

该命令用于选项字节读写（包含了读保护等级、Data0/1 配置、USER 配置）。

上层指令:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x40 一级命令字段							
1(CMD_L)	二级命令字段							
2~3(LEN)	发送数据长度							
4~7(Par)	无							
8~27(DAT)	CMD_L = 0x00: 全部为 0x00 CMD_L = 0x01/0x02: 选项字节配置值							

● CMD_L 二级命令字段:

1. 0x00: 获取选项字节。
2. 0x01: 配置选项字节。
3. 0x02: 配置选项字节, 再复位。

● LEN 发送数据长度: 0x10(LEN[0])、0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$ 。

● DAT[0~15]: 选项字节配置 16 个字节

RDP、nRDP、USER、nUSER、Data0、nData0、Data1、nData1、USER2、nUSER2、USER3、nUSER3、RDP2、nRDP2、USER4、nUSER4;

当 CMD_L = 0x00: 全部为 0x00。

当 CMD_L = 0x01/0x02: 配置选项字节为要写入的值。

底层应答:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x40 一级命令字段							
1(CMD_L)	二级命令字段							
2~3(LEN)	发送数据长度							
4~19(DAT)	选项字节配置 16 个字节							
24(CR1)	状态字节 1							
25(CR2)	状态字节 2							

● LEN 发送数据长度: 0x10(LEN[0])、0x00(LEN[1]), $LEN = LEN[0] + (LEN[1] \ll 8)$ 。

- DAT[0~15]: 当前选项字节配置 16 个字节

RDP、nRDP、USER、nUSER、Data0、nData0、Data1、nData1、USER2、nUSER2、USER3、nUSER3、RDP2、nRDP2、USER4、nUSER4;

- 状态字节(CR1、CR2)根据命令执行情况分为:

1. 返回成功: 状态标志位(0xA0、0x00)。

2. 校验失败: 状态标志位(CR1, CR2)

(1)、(0xB0、0x00): 返回失败;

4.2.8 CMD_SYS_RESET

该命令用于软件复位 BOOT 程序。

上层指令:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x50 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
4~7(Par)	保留							
(DAT)	无							

- 保留值: 0x00;

底层应答:

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x50 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
(DAT)	无							
4(CR1)	状态字节 1							
5(CR2)	状态字节 2							

- 状态字节(CR1、CR2)根据命令执行情况分为:

1. 返回成功: 状态标志位(0xA0、0x00)。

2. 返回失败：状态标志位(0xB0、0x00)。

4.2.9 CMD_APP_GO

该命令用于 BOOT 下载完应用程序到 FLASH 后跳转用户复位程序入口地址（默认为 0x0800_0C00）执行，注意偏移中断向量表。

上层指令：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x51 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3(LEN)	发送数据长度							
4~7(Par)	保留							
(DAT)	无							

- 保留值：0x00；

底层应答：

byte \ bit	b7	b6	b5	b4	b3	b2	b1	b0
0(CMD_H)	0x51 一级命令字段							
1(CMD_L)	0x00 二级命令字段							
2~3 (LEN)	发送数据长度							
(DAT)	无							
4(CR1)	状态字节 1							
5(CR2)	状态字节 2							

- 状态字节(CR1、CR2)根据命令执行情况分为：

1. 返回成功：状态标志位(0xA0、0x00)。
2. 返回失败：状态标志位(0xB0、0x00)。

4.3 返回状态字说明

4.3.1 返回成功状态字

返回成功：状态标志位(0xA0、0x00)。表示上层下发的命令执行成功，返回成功状态字。

包含了读取、更新、配置等命令的成功返回值。

4.3.2 返回失败状态字

返回失败：状态标志位(0xB0、0x00)。表示上层下发的命令由于其他原因（命令接受格式错误或者超时等）执行失败，返回失败状态字。

4.3.3 返回其他状态字

下列的返回状态字也是返回失败，第二字节的状态字表示不同的错误类型。

- (1)、(0xB0、0x30)：擦除/下载 FLASH 页被 RDP 保护；
- (2)、(0xB0、0x34)：擦除/下载/CRC 校验地址范围越界（指超出整个 FLASH 大小）；
- (3)、(0xB0、0x35)：擦除/下载/CRC 校验起始地址不是 16 字节对齐；
- (4)、(0xB0、0x36)：下载/CRC 校验数据长度不是 16 的倍数；数据长度表示擦除 FLASH 的长度，或者是下载代码到 FLASH 的长度，或者是校验 FLASH CRC 值的长度；
- (5)、(0xB0、0x37)：擦除/下载 FLASH 编程失败；
- (6)、(0xB0、0x38)：CRC 校验失败；
- (7)、(0xBB、0xCC)：上层发送命令一级、二级命令字段不属于任何命令。

5 历史版本

版本	日期	备注
V1.0.0	2026-03-24	创建文档

6 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用人在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证责任，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。