AN3155
应用笔记

STM32™ 自举程序中使用的 USART 协议

前言

本应用笔记将介绍 USART 协议在 STM32 微控制器自举程序中的应用，还将详细介绍支持的每个命令。要详细了解器件自举程序的 USART 硬件资源和要求，请参见“STM32 系统存储器自举模式”（应用笔记 AN2606）。

相关文档

可从 www.st.com 下载：

AN2606“STM32 系统存储器自举模式”

表 1. 适用的产品

类型	料号
微控制器	– STM32F050x4、STM32F050x6、STM32F051x4、STM32F051x6、STM32F051x8 – STM32F1 主流产品 – STM32F2 高性能系列 – STM32F302xx、STM32F303xx、STM32F313xx、STM32F372xx、STM32F373xx、STM32F383xx – STM32F405xx、STM32F407xx、STM32F415xx、STM32F417xx、STM32F427xx、STM32F437xx – STM32L1 系列

目录

1 USART 自举程序代码序列 5

2 选择 USARTx 波特率 6

2.1 最小波特率 6

2.2 最大波特率 6

3 自举程序命令集 7

3.1 器件相关的自举程序参数 8

3.2 Get 命令 8

3.3 Get Version & Read Protection Status 命令 10

3.4 Get ID 命令 11

3.5 Read Memory 命令 13

3.6 Go 命令 16

3.7 Write Memory 命令 18

3.8 Erase Memory 命令 21

3.9 Extended Erase Memory 命令 24

3.10 Write Protect 命令 27

3.11 Write Unprotect 命令 30

3.12 Readout Protect 命令 31

3.13 Readout Unprotect 命令 33

4 自举程序协议版本演化 35

5 版本历史 36

表格索引

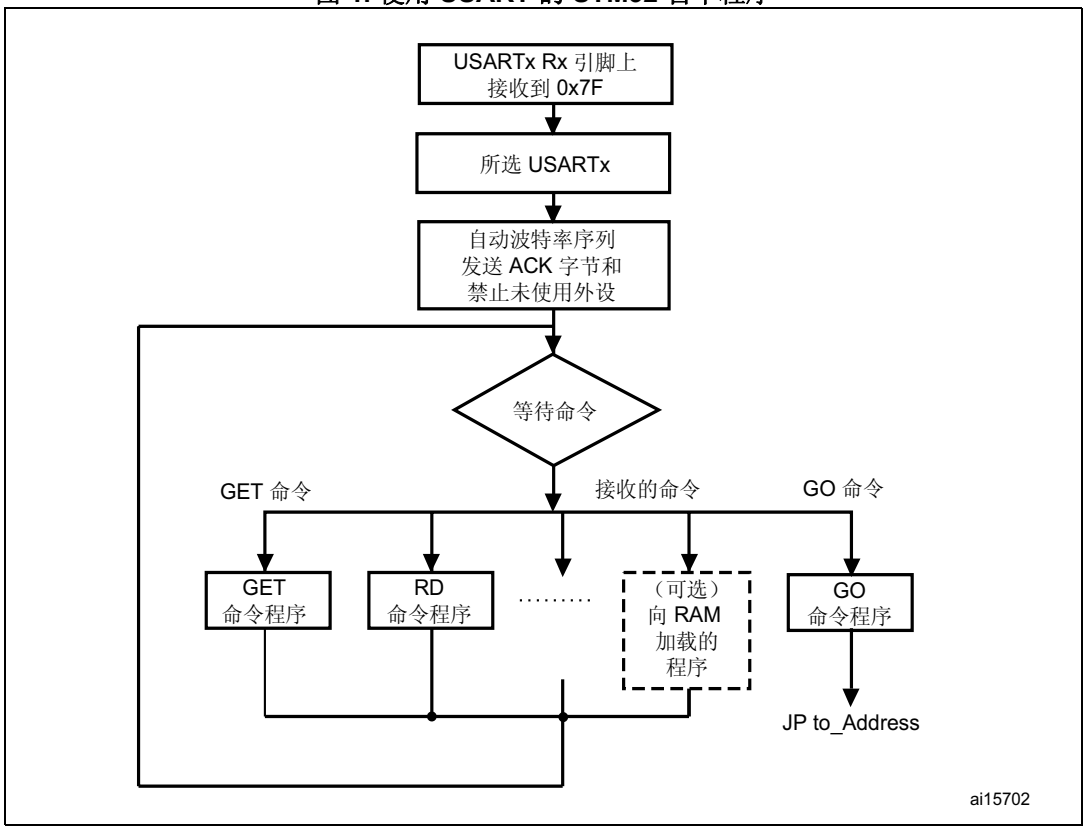
表 1. 适用的产品 1
表 2. USART 自举程序命令 7
表 3. 自举程序协议版本 35
表 4. 文档版本历史 36

图片索引

图 1. 使用 USART 的 STM32 自举程序 5
图 2. Get 命令：主机端 8
图 3. Get 命令：器件端 9
图 4. Get Version & Read Protection Status 命令：主机端 10
图 5. Get Version & Read Protection Status 命令：器件端 11
图 6. Get ID 命令：主机端 12
图 7. Get ID 命令：器件端 12
图 8. Read Memory 命令：主机端 14
图 9. Read Memory 命令：器件端 15
图 10. Go 命令：主机端 16
图 11. Go 命令：器件端 17
图 12. Write Memory 命令：主机端 19
图 13. Write Memory 命令：器件端 20
图 14. Erase Memory 命令：主机端 22
图 15. Erase Memory 命令：器件端 23
图 16. Extended Erase Memory 命令：主机端 25
图 17. Extended Erase Memory 命令：器件端 26
图 18. Write Protect 命令：主机端 28
图 19. Write Protect 命令：器件端 29
图 20. Write Unprotect 命令：主机端 30
图 21. Write Unprotect 命令：器件端 31
图 22. Readout Protect 命令：主机端 32
图 23. Readout Protect 命令：器件端 32
图 24. Readout Unprotect 命令：主机端 33
图 25. Readout Unprotect 命令：器件端 34

1 USART 自举程序代码序列

图 1. 使用 USART 的 STM32 自举程序



当配置 STM32 微控制器为自举启动，系统将进入自举程序模式（有关详细信息，请参见应用笔记 AN2606 “STM32 系统存储器自举模式”），自举程序代码将立即扫描 USARTx_RX 引脚，等待接收 0x7F 数据帧：一个起始位，0x7F 数据位，偶校验位和一个停止位。

此数据帧的持续时间由 SysTick 定时器测量。之后，该定时器的计数值用于计算关于当前系统时钟的相应波特率因子。

随后，代码将相应初始化串行接口。通过计算出的波特率，发送确认字节 *(0x79)* 返回主机，表示 STM32 已准备好接收命令。

2 选择 USARTx 波特率

USARTx 串口波特率根据接收到的首字节长度进行计算，便于在很大波特率范围下运行自举程序。不过，为了确保数据传输正常进行，波特率必须确保在对应范围的上限和下限内。

为了确保从主机到微控制器的数据传输正常进行，USARTx 内部初始化波特率与主机实际波特率之间的最大偏差应小于 2.5%。可使用如下公式计算主机波特率与微控制器波特率之间的偏差（f_B，用百分比表示）：

$$f_B = \left| \frac{\text{STM32 baud rate} - \text{Host baud rate}}{\text{STM32 baud rate}} \right| \times 100\% \quad \text{，其中 } f_B \leq 2.5\%。$$

此波特率偏差为非线性函数，其结果取决于 CPU 时钟和主机波特率。函数 (f_B) 的最大值随主机波特率增大。原因是，波特率预分频系数越小，隐含的量化误差越大。

2.1 最小波特率

测试所得的最小波特率 (B_{Low}) 为 1200。波特率低于 B_{Low} 会导致 SysTick 定时器溢出。此时 USARTx 将无法正确初始化。

2.2 最大波特率

B_{High} 为偏差不超过限值的最高波特率。B_{Low} 和 B_{High} 之间的所有波特率均低于偏差限值。测试所得的最高波特率 (B_{High}) 为 115 200。

3 自举程序命令集

下面的*表 2* 中列出了支持的命令。本部分将详细说明其中的每一个命令。

表 2. USART 自举程序命令

命令(1)	命令代码	命令说明
Get(2)	0x00	获取当前自举程序版本及允许使用的命令
Get Version & Read Protection Status(2)	0x01	获取自举程序版本及 Flash 的读保护状态
Get ID(2)	0x02	获取芯片 ID
Read Memory	0x11	从应用程序指定的地址开始读取最多 256 个字节的存储器空间
Go	0x21	跳转到内部 Flash 或 SRAM 内的应用程序代码
Write Memory	0x31	从应用程序指定的地址开始将最多 256 个字节的数据写入 RAM 或 Flash
Erase(3)	0x43	擦除一个到全部 Flash 页面
Extended Erase(3)	0x44	使用双字节寻址模式擦除一个到全部 Flash 页面（仅用于 v3.0 usart 自举程序版本及以上版本）。
Write Protect(4)	0x63	使能某些扇区的写保护
Write Unprotect(4)	0x73	禁止所有 Flash 扇区的写保护
Readout Protect	0x82	使能读保护
Readout Unprotect(2)	0x92	禁止读保护

1. 如果接收到拒绝命令或在执行命令期间出现错误，自举程序则会发送 NACK 字节并返回检查命令状态。
2. 读保护 - 激活 RDP（读保护）选项后，只能使用这一有限的命令子集。其它命令都会收到 NACK 应答，并且不会对器件起作用。取消 RDP 即可激活其它命令。
3. Erase (x043) 和 Extended Erase (0x44) 均为独占命令。一个器件可支持 Erase 命令或 Extended Erase 命令，但不能同时支持这两个命令。
4. 请参见*第 3.1 节：器件相关的自举程序参数*。

通信安全

编程工具 (PC) 到器件的所有通信均通过如下方式验证：

1. 校验和：接收到的数据字节块进行异或运算。每个通信结尾增加一个字节（校验和字节），包含前面所有字节异或运算的结果。异或运算所有接收到的字节，即数据包加上校验和字节，结果必须为 0x00
2. 针对每条命令，主机都会发送一个字节及其补码（异或结果 = 0x00）
3. UART：激活奇偶校验（偶校验）

每个数据包或者被接受（ACK 应答）或者被丢弃（NACK 应答）：

- ACK = 0x79
- NACK = 0x1F

3.1 器件相关的自举程序参数

虽然所有 STM32 器件的 USART 自举程序协议命令集和序列均相同，但某些参数与具体器件相关。对一些命令，某些参数值可能取决于所使用的器件。相关参数如下：

- PID（产品 ID），该参数因器件而异
- 执行 Read Memory、Go 和 Write Memory 命令时，自举程序允许的有效存储器地址（RAM、Flash、系统存储器、选项字节区域）。
- 执行 Write Protect 命令时使用的 Flash 扇区的大小。

要了解所使用器件中这些参数值的详细信息，请参见“STM32 系统存储器自举模式”（应用笔记 AN2606）中的“器件相关的自举程序参数”部分。

3.2 Get 命令

用户通过 Get 命令可获取自举程序的版本及支持的命令。自举程序接收到 Get 命令后，会将自举程序版本和支持命令的代码发送给主机，如*图 2* 所示。

图 2. Get 命令：主机端

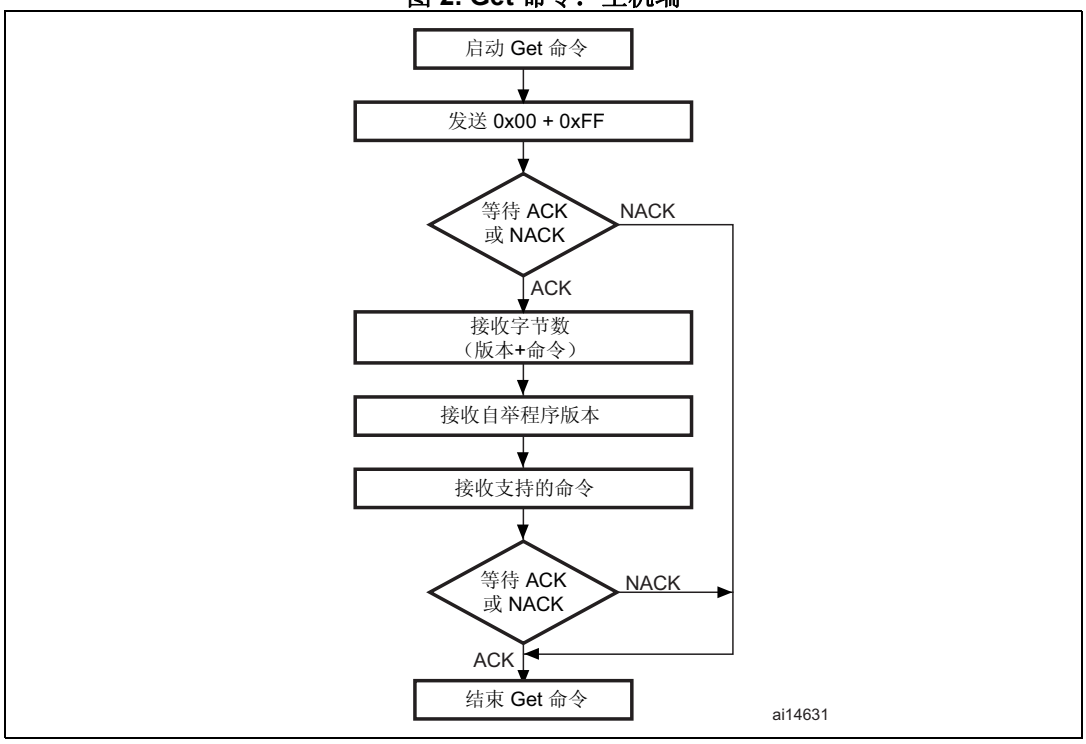
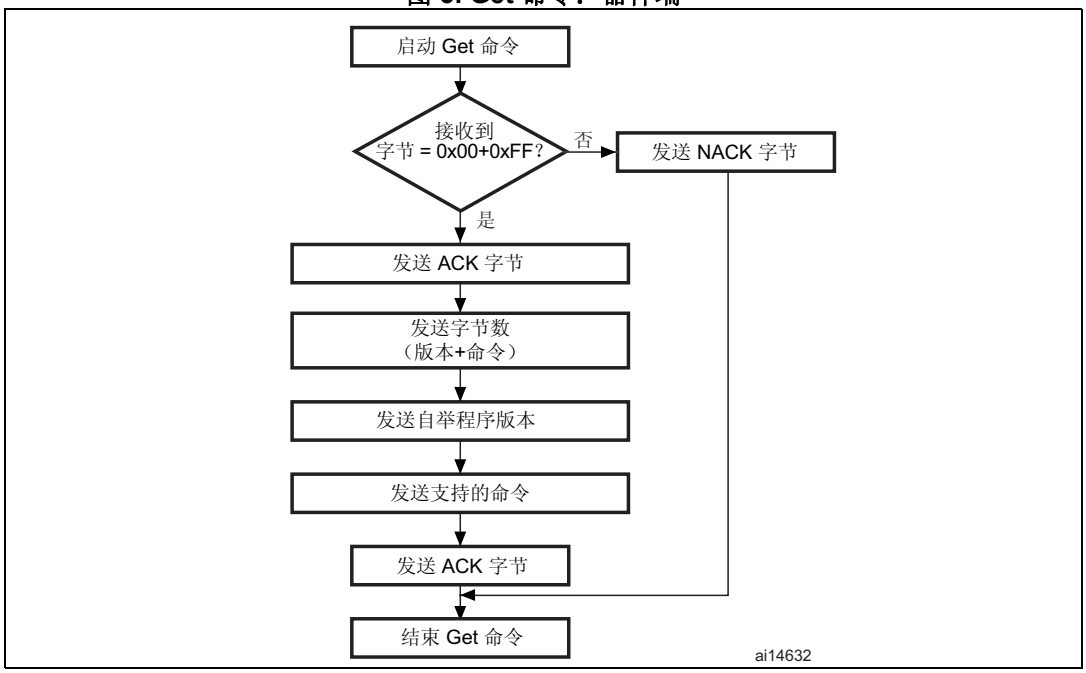


图 3. Get 命令：器件端



STM32 发送如下字节：

字节 1： ACK

字节 2： N = 11 = 后接字节数 – 1 个除当前字节和 ACK 之外的字节。

字节 3： 自举程序版本（0 < 版本 < 255），示例：0x10 = 版本 1.0

字节 4： 0x00 – Get 命令

字节 5： 0x01 – Get Version and Read Protection Status

字节 6： 0x02 – Get ID

字节 7： 0x11 – Read Memory 命令

字节 8： 0x21 – Go 命令

字节 9： 0x31 – Write Memory 命令

字节 10： 0x43 或 0x44 – Erase 命令或 Extended Erase 命令（这些命令均为独占命令）

字节 11： 0x63 – Write Protect 命令

字节 12： 0x73 – Write Unprotect 命令

字节 13： 0x82 – Readout Protect 命令

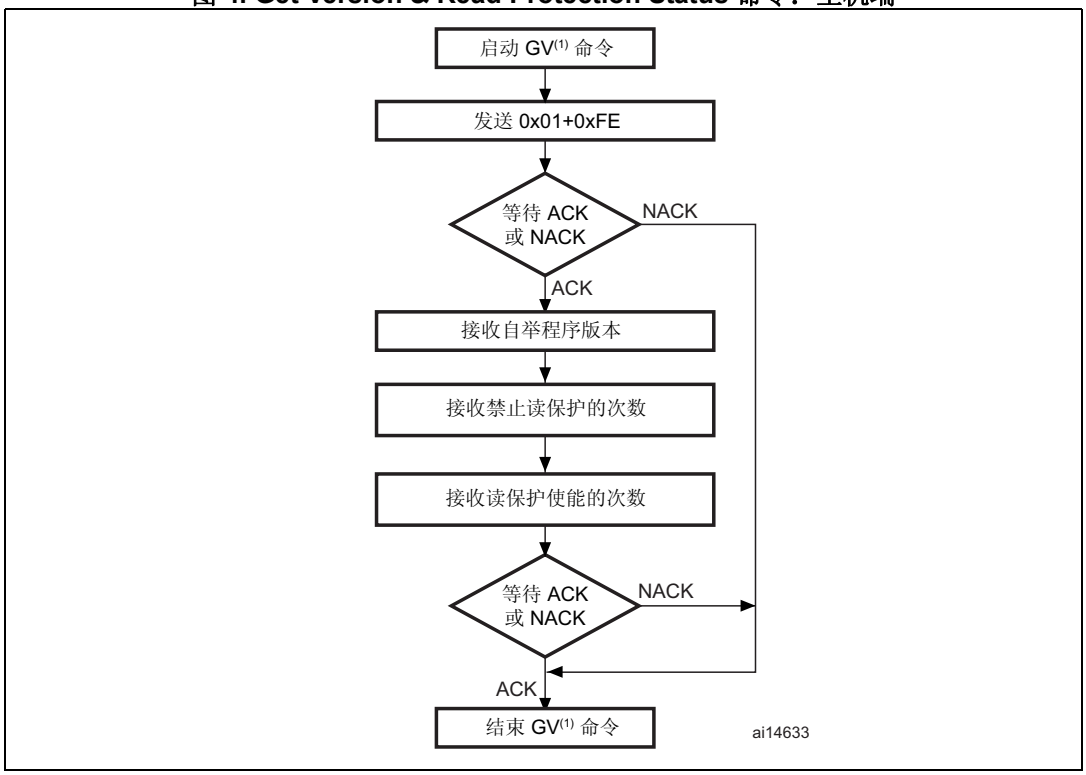
字节 14： 0x92 – Readout Unprotect 命令

末字节 (15)： ACK

3.3 Get Version & Read Protection Status 命令

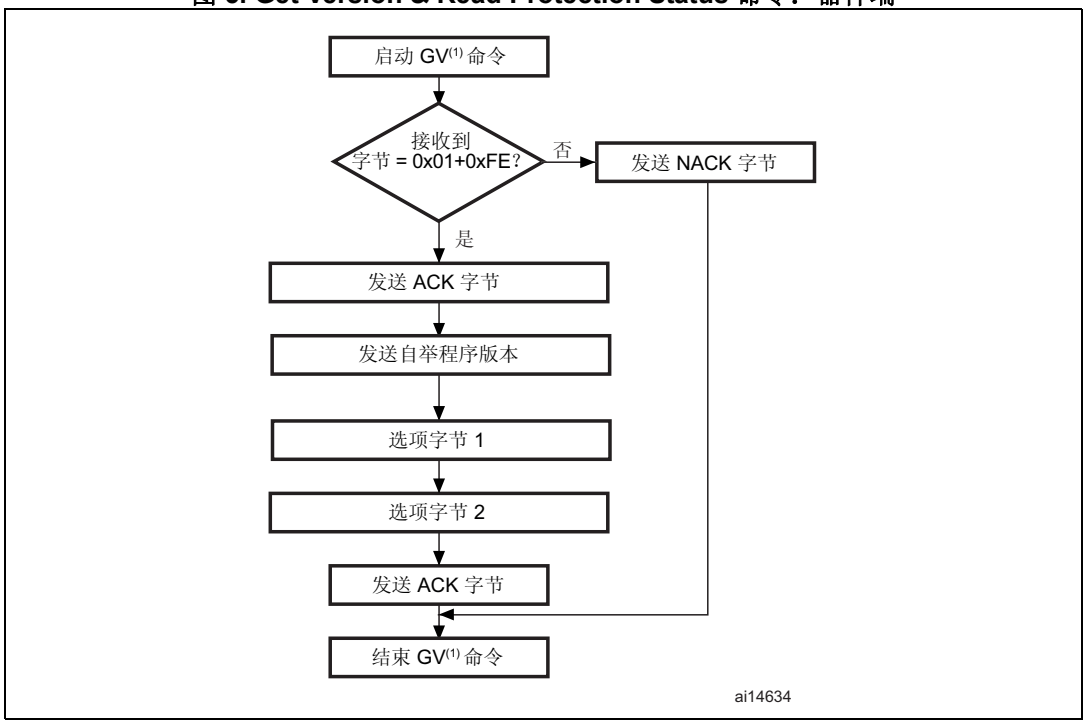
Get Version & Read Protection Status 命令用于获取自举程序版本及读保护状态。自举程序接收到此命令后，会将如下信息（版本、使能和禁止读保护的次数）发送给主机。

图 4. Get Version & Read Protection Status 命令：主机端



1. GV = Get Version & Read Protection Status。

图 5. Get Version & Read Protection Status 命令：器件端



1. GV = Get Version & Read Protection Status。

STM32 发送如下字节：

字节 1： ACK

字节 2： 自举程序版本（0 < 版本 ≤ 255），示例：0x10 = 版本 1.0

字节 3： 选项字节 1：0x00，保持与通用自举程序协议的兼容性

字节 4： 选项字节 2：0x00，保持与通用自举程序协议的兼容性

字节 5： ACK

3.4 Get ID 命令

Get ID 命令用于获取芯片 ID（标识）的版本。自举程序接收到此命令后，会将产品 ID 发送给主机。

STM32 器件发送如下字节：

字节 1： ACK

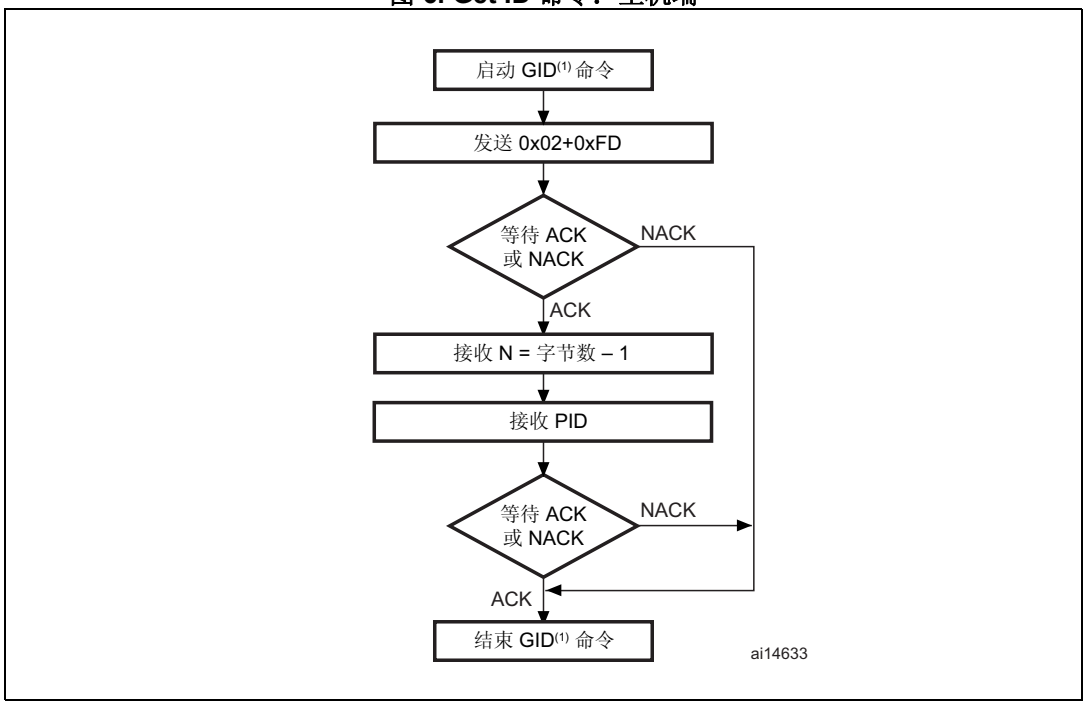
字节 2： N = 字节数 – 1（对 STM32 N = 1），除当前字节和 ACK 之外。

字节 3-4： PID(1) 字节 3 = 0x04，字节 4 = 0xXX

字节 5： ACK

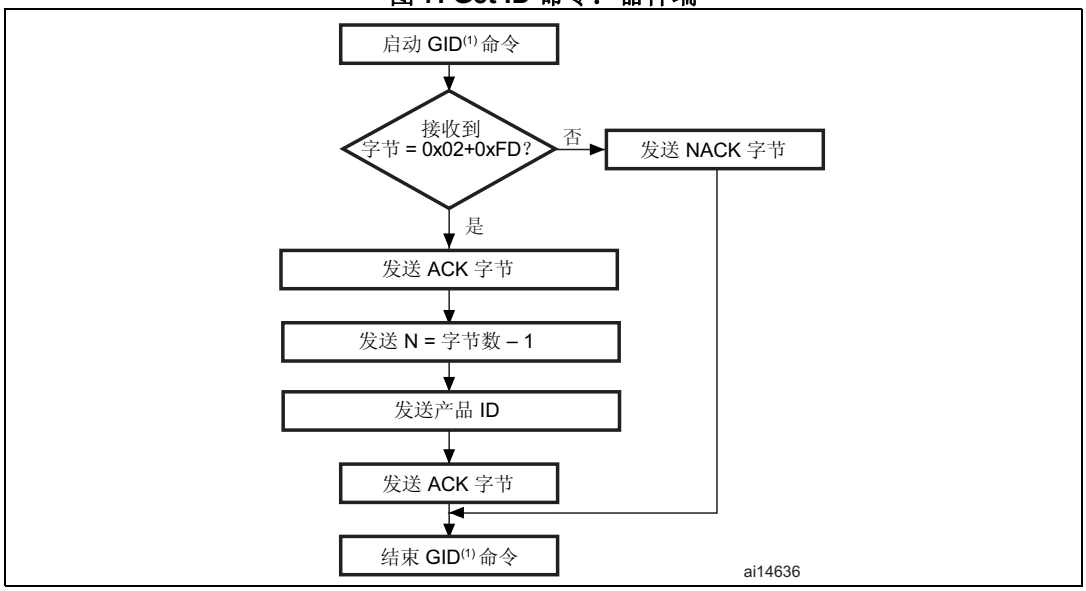
1. PID 表示产品 ID。字节 1 表示 MSB，字节 2 表示 ID 的 LSB。有关所使用器件的 PID 详细信息，请参见 *第 3.1 节：器件相关的自举程序参数*。

图 6. Get ID 命令：主机端



1. GID = Get ID。

图 7. Get ID 命令：器件端



1. GID = Get ID。

3.5 Read Memory 命令

Read Memory 命令用于从 RAM、Flash 和信息块（系统存储器或选项字节区域）中的任何有效存储器地址（参见注释）读取数据。

注： *有关所用器件有效存储器地址的详细信息，请参见第 3.1 节：器件相关的自举程序参数。*

自举程序接收到 Read Memory 命令后，会将 ACK 字节发送到应用程序。发送 ACK 字节后，自举程序会等待一个地址（4 个字节，字节 1 表示地址 MSB，字节 4 表示 LSB）和一个校验和字节，然后检查接收到的地址。如果接收到的地址有效且校验和正确，则自举程序将发送一个 ACK 字节，否则将发送一个 NACK 字节并中止此命令。
如果接收到的地址有效且校验和正确，则自举程序将等待要发送的字节数 - 1（N 字节）及其补充字节（校验和）。如果校验和正确，则自举程序将从接收的地址开始向应用程序发送所需数据（(N + 1) 字节）。如果校验和错误，则会在中止命令之前发送 NACK。

主机将如下字节发送到 STM32：

字节 1-2： 0x11+0xEE

等待 ACK

字节 3 到 6： 起始地址

- 字节 3：MSB
- 字节 6：LSB

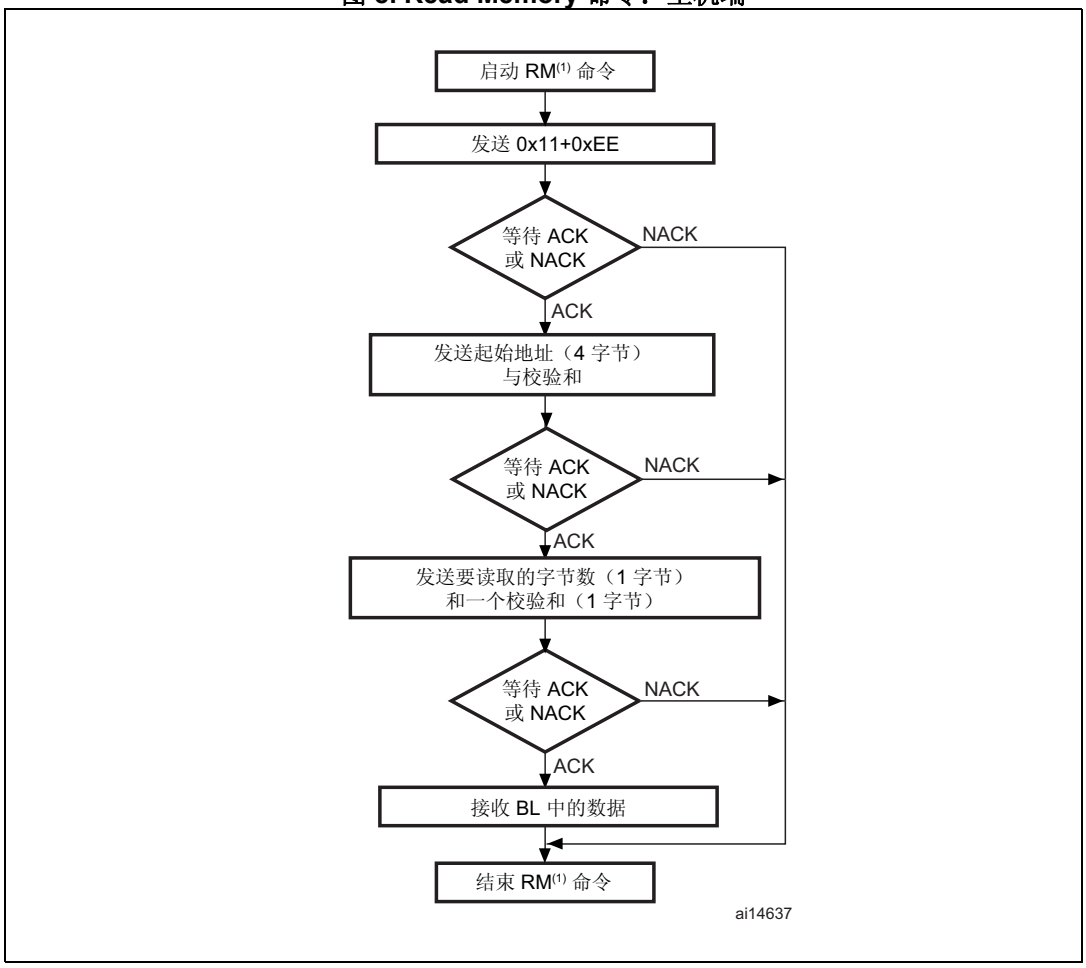
字节 7： 校验和：异或（字节 3、字节 4、字节 5、字节 6）

等待 ACK

字节 8： 要读取的字节数 – 1 (0 < N ≤ 255)；

字节 9： 校验和：异或字节 8（字节 8 的补码）

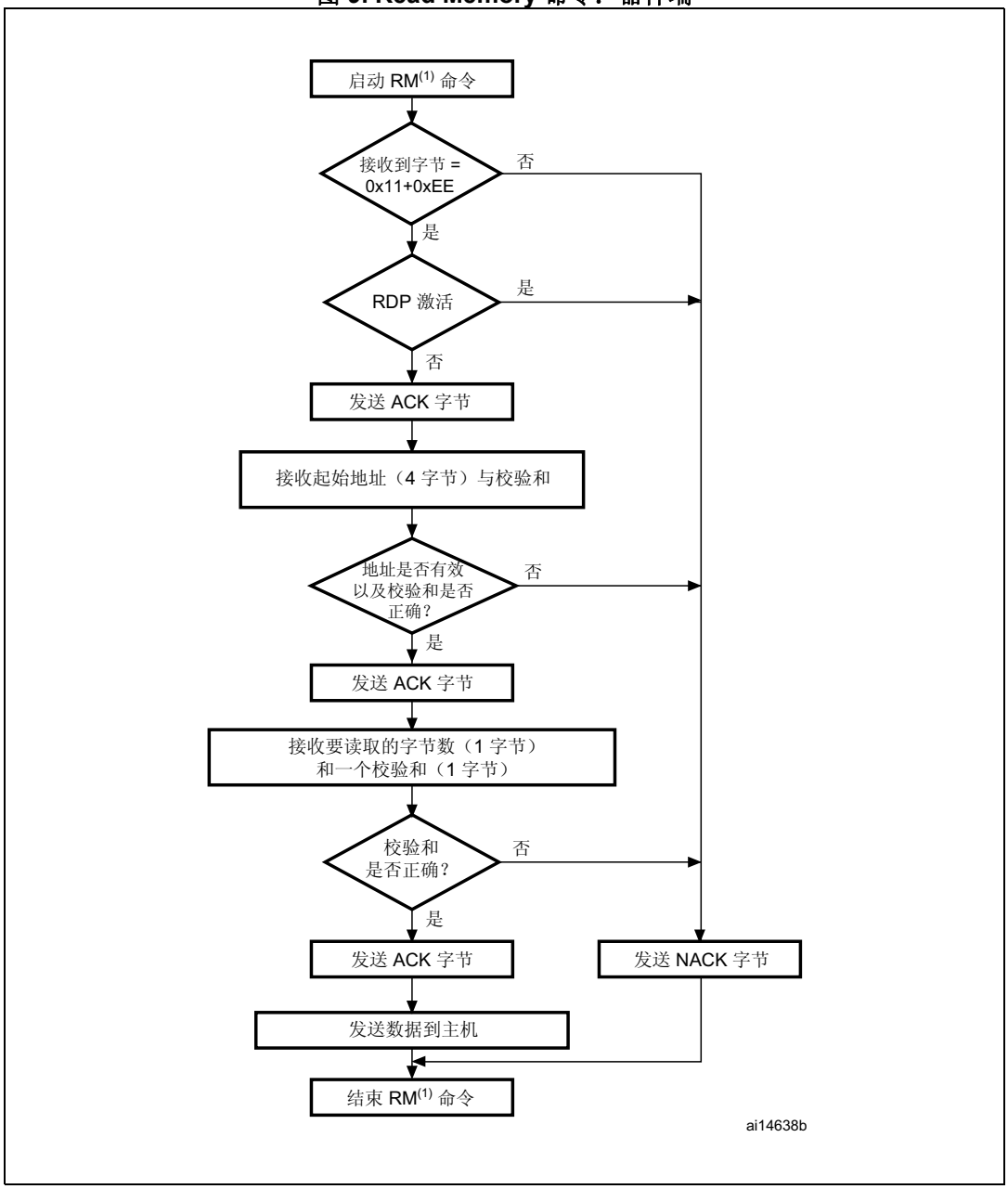
图 8. Read Memory 命令：主机端



1. RM = Read Memory。

注： *激活读保护 (RDP) 后（或激活读保护等级 1），某些产品会返回两个 NACK，而非仅返回一个 NACK。如需了解指定产品在这种情况下仅返回一个 NACK 还是返回两个 NACK，请参见 AN2606 中与该产品相关的已知限制部分。*

图 9. Read Memory 命令：器件端



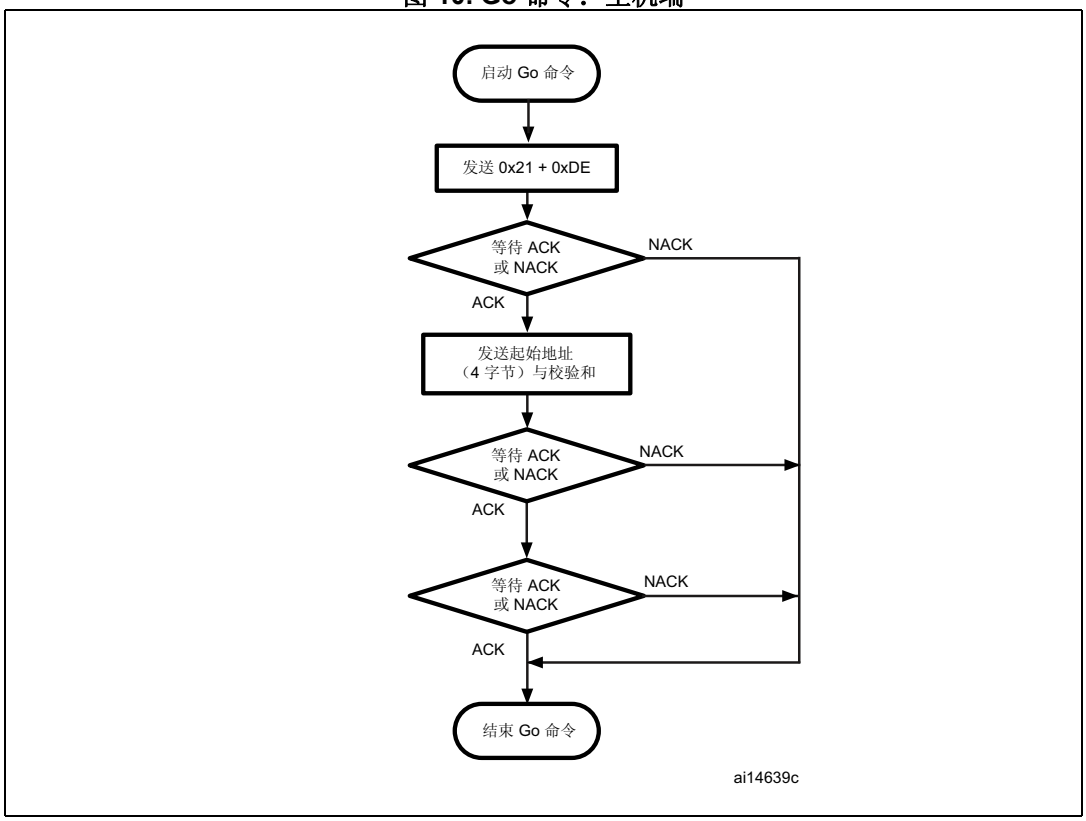
1. RM = Read Memory。

3.6 Go 命令

Go 命令用于从应用程序指定的地址开始执行已下载的代码或其它任何代码。自举程序接收到 Go 命令后，会将 ACK 字节发送到应用程序。发送 ACK 字节后，自举程序会等待一个地址（4 个字节，字节 1 表示地址 MSB，字节 4 表示 LSB）和一个校验和字节，然后检查接收到的地址。如果接收到的地址有效且校验和正确，则自举程序将发送一个 ACK 字节，否则将发送一个 NACK 字节并中止此命令。
若接收的地址有效且校验和正确，则自举程序固件将执行以下操作：

- 初始化自举程序使用的外设寄存器，将其设置为默认复位值
- 初始化用户应用程序的主栈指针
- 跳转到接收的“地址 + 4”（与应用程序中复位处理程序的地址相对应）中指定的存储器位置。
  例如，如果接收的地址为 0x0800 0000，则自举程序将跳转到地址为 0x0800 0004 的存储器位置。
  一般来说，主机应发送基准地址，在该地址指定应用程序的跳转目标位置

图 10. Go 命令：主机端



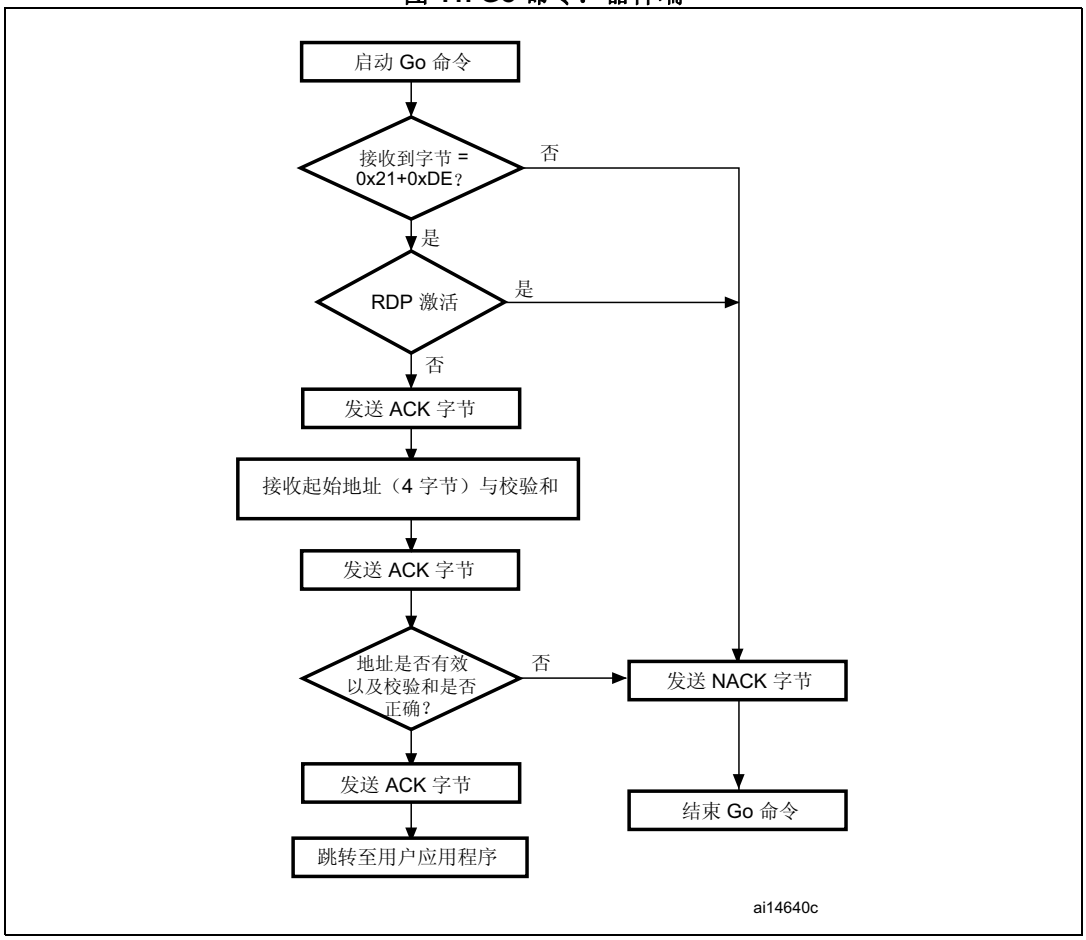
注： 1 *Go 命令的有效地址存储于 RAM 或 Flash 中（要详细了解所使用器件的有效存储器地址，请参见第 3.1 节：器件相关的自举程序参数）。其它地址均无效，并由器件发出 NACK 应答。*

2 *如果将应用程序加载到 RAM 后执行跳转，则程序运行时必须配置相应的偏移来避免与自举程序固件使用的 RAM 存储空间发生重叠（要详细了解所使用器件的 RAM 偏移，请参见第 3.1 节：器件相关的自举程序参数）。*

3 *只有用户应用程序正确设置向量表，使其指向应用程序地址时，应用程序的跳转才可正常动作。*

4 *激活读保护 (RDP) 后（或激活读保护等级 1），某些产品会返回两个 NACK，而非仅返回一个 NACK。如需了解指定产品在这种情况下仅返回一个 NACK 还是返回两个 NACK，请参见 AN2606 中与该产品相关的已知限制部分。*

图 11. Go 命令：器件端



主机将如下字节发送到 STM32：

字节 1： 0x21

字节 2： 0xDE

等待 ACK

字节 3 到字节 6：起始地址

字节 3：MSB

字节 6：LSB

字节 7： 校验和：异或（字节 3、字节 4、字节 5、字节 6）

3.7 Write Memory 命令

Write Memory 命令用于将数据写入 RAM、Flash 或选项字节区域的任意有效存储器地址（参见下文注释）。

自举程序接收到 Write Memory 命令后，会将 ACK 字节发送到应用程序。发送 ACK 字节后，自举程序会等待一个地址（4 个字节，字节 1 表示地址 MSB，字节 4 表示 LSB）和一个校验和字节，然后检查接收到的地址。对于选项字节区域，起始地址必须为选项字节区域（参见注释）的基准地址，以避免在此区域中不当写入数据。

注： 1 *对 Flash/SRAM 的写入操作必须按字（32 位）对齐且写入数据应为 4 字节的倍数。如果写入数据少于分配的存储空间，则应以 0xFF 填充剩余字节。*

2 *有关所用器件有效存储器地址的详细信息，请参见第 3.1 节：器件相关的自举程序参数。*

如果接收到的地址有效且校验和正确，则自举程序将发送一个 ACK 字节，否则将发送一个 NACK 字节并中止此命令。若接收的地址有效且校验和正确，则自举程序将执行以下操作：

- 获取字节 N，其中包含待接收的数据字节数
- 接收用户数据（(N + 1) 字节）以及校验和（N 以及所有数据字节的异或运算结果）
- 从接收的地址开始针对存储器进行用户数据编程
- 在命令结束时，如果写操作成功完成，则自举程序将发送 ACK 字节；否则将 NACK 字节发送给应用程序并中止命令

对 STM32 执行写操作时，允许写入的最大数据块长度为 256 个字节。

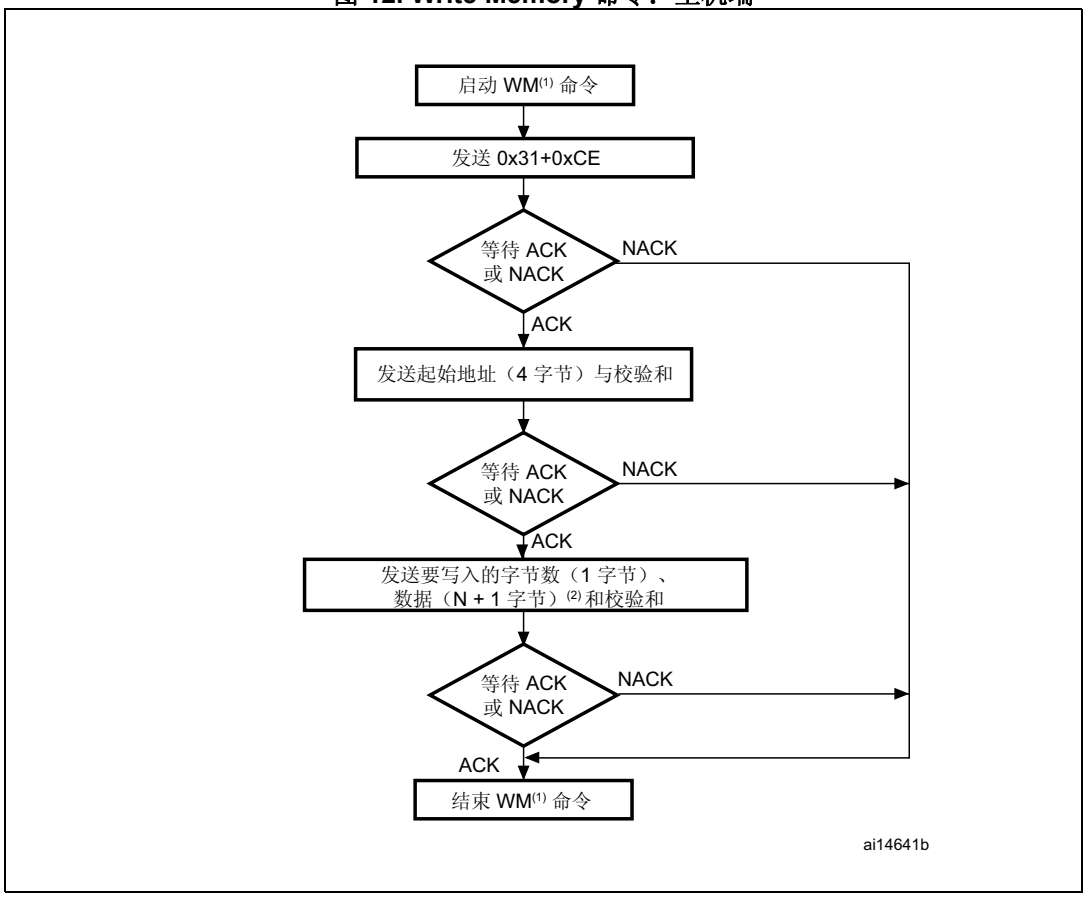
如果将 Write Memory 命令发送到选项字节区域，则在写入新值前会清除所有的选项，并由自举程序在命令结束时启动系统复位来实施新的选项字节配置。

注： 1 *写入 RAM 时，应注意避免与自举程序固件使用的第一个 RAM 存储器发生重叠。*

2 *在具有写保护的扇区中执行写操作时不会返回错误信息。*

3 *起始地址无效时也不会返回错误信息。*

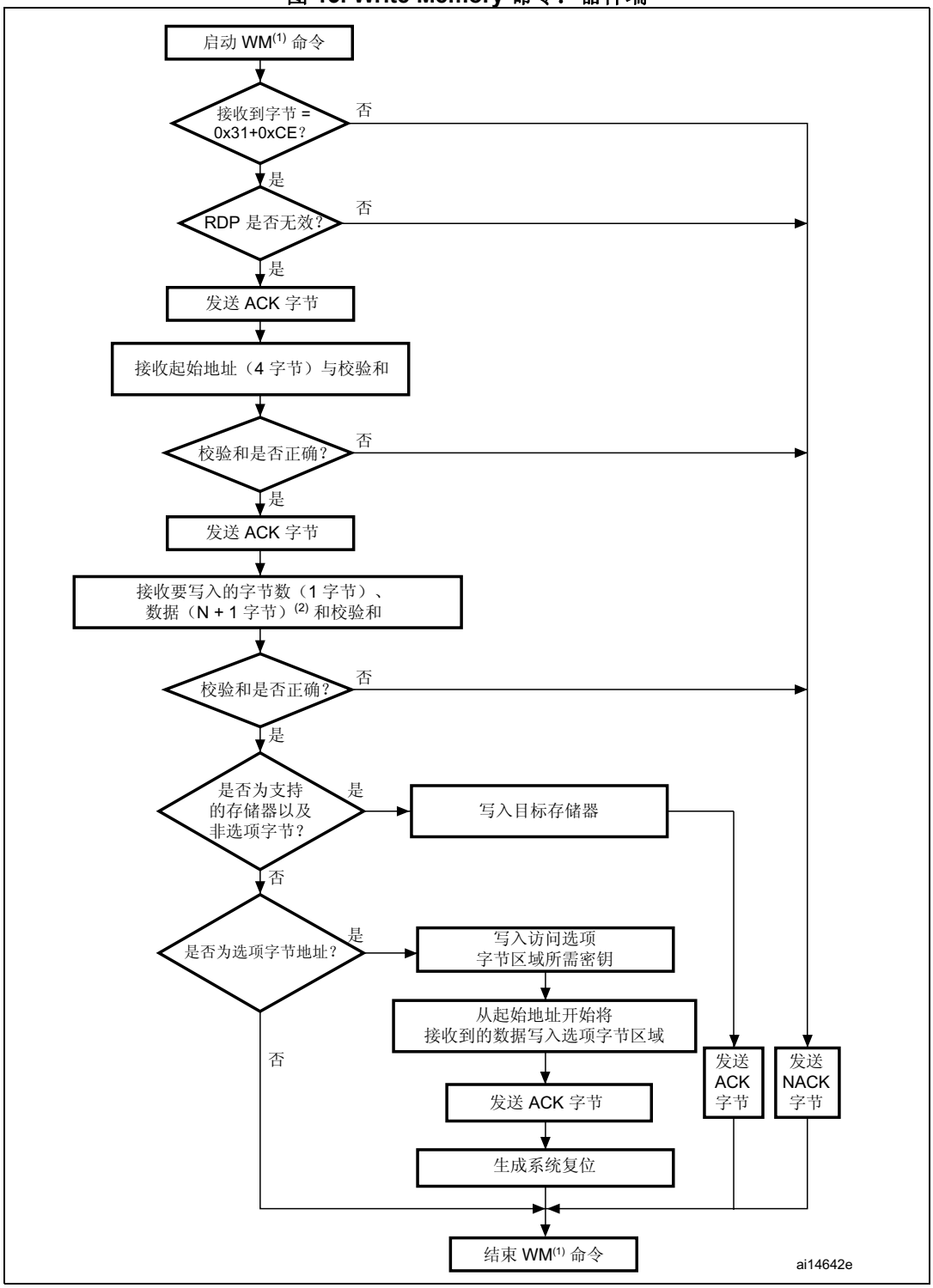
图 12. Write Memory 命令：主机端



1. WM = Write Memory。
2. N+1 应始终为 4 的倍数。

注： *激活读保护 (RDP) 后（或激活读保护等级 1），某些产品会返回两个 NACK，而非仅返回一个 NACK。如需了解指定产品在这种情况下仅返回一个 NACK 还是返回两个 NACK，请参见 AN2606 中与该产品相关的已知限制部分。*

图 13. Write Memory 命令：器件端



1. WM = Write Memory。
2. N+1 应始终为 4 的倍数。

主机将如下字节发送到 STM32：

字节 1： 0x31

字节 2： 0xCE

等待 ACK

字节 3 到字节 6： 起始地址

字节 3：MSB

字节 6：LSB

字节 7： 校验和：异或运算结果（字节 3、字节 4、字节 5、字节 6）

等待 ACK

字节 8： 要接收的字节数 (0 < N ≤ 255)

N +1 数据字节： （最多 256 字节）

校验和字节： 异或运算结果（N、N +1 数据字节）

3.8 Erase Memory 命令

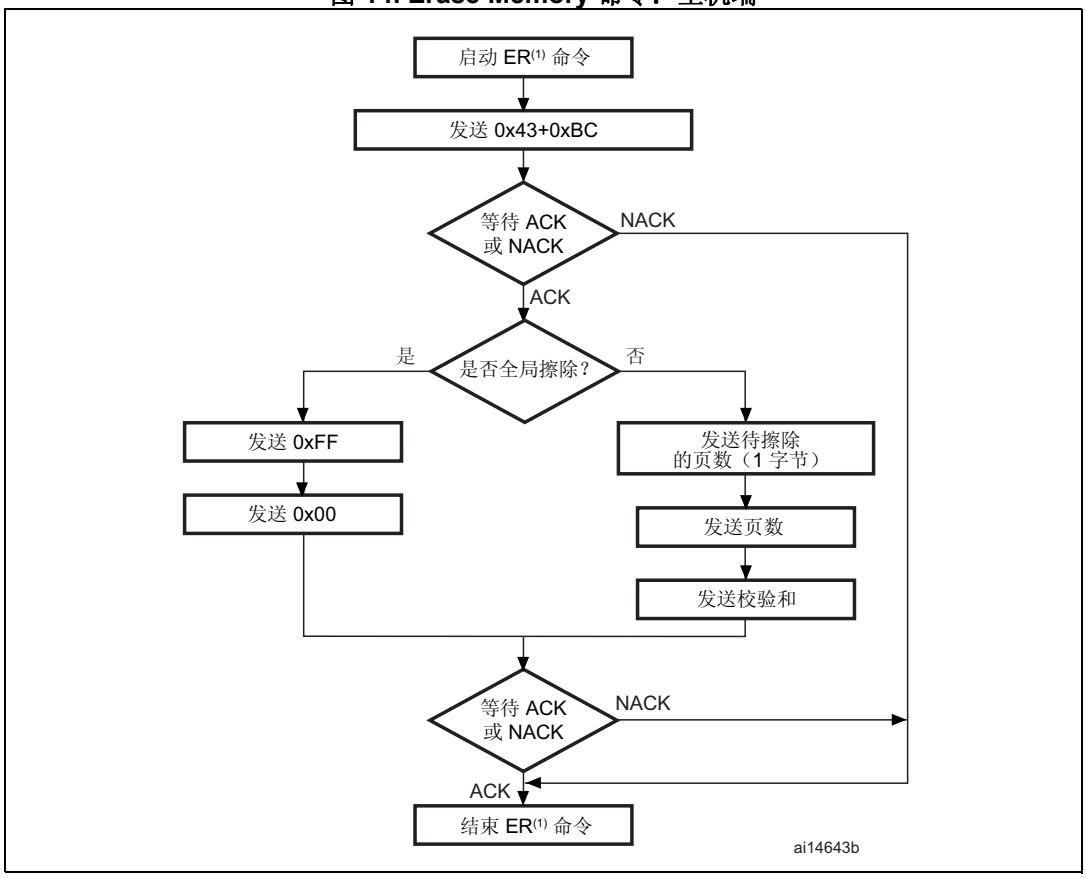
主机可通过 Erase Memory 命令擦除 Flash 页面。自举程序接收到 Erase Memory 命令后，会将 ACK 字节发送到主机。发送 ACK 字节后，自举程序将接收一个字节（待擦除的页面数）、Flash 页面代码以及一个校验和字节； 如果校验和正确，则自举程序将擦除该存储器并向主机发送一个 ACK 字节，否则将向主机发送一个 NACK 字节并中止此命令。

Erase Memory 命令规范：

1. 自举程序接收一个包含 N（要擦除的页数 - 1）的字节。
   保留 N = 255 用于全局擦除请求。0 ≤ N ≤ 254 时，擦除 N + 1 页。
2. 自举程序接收到 (N + 1) 个字节，每个字节都包含一个页数

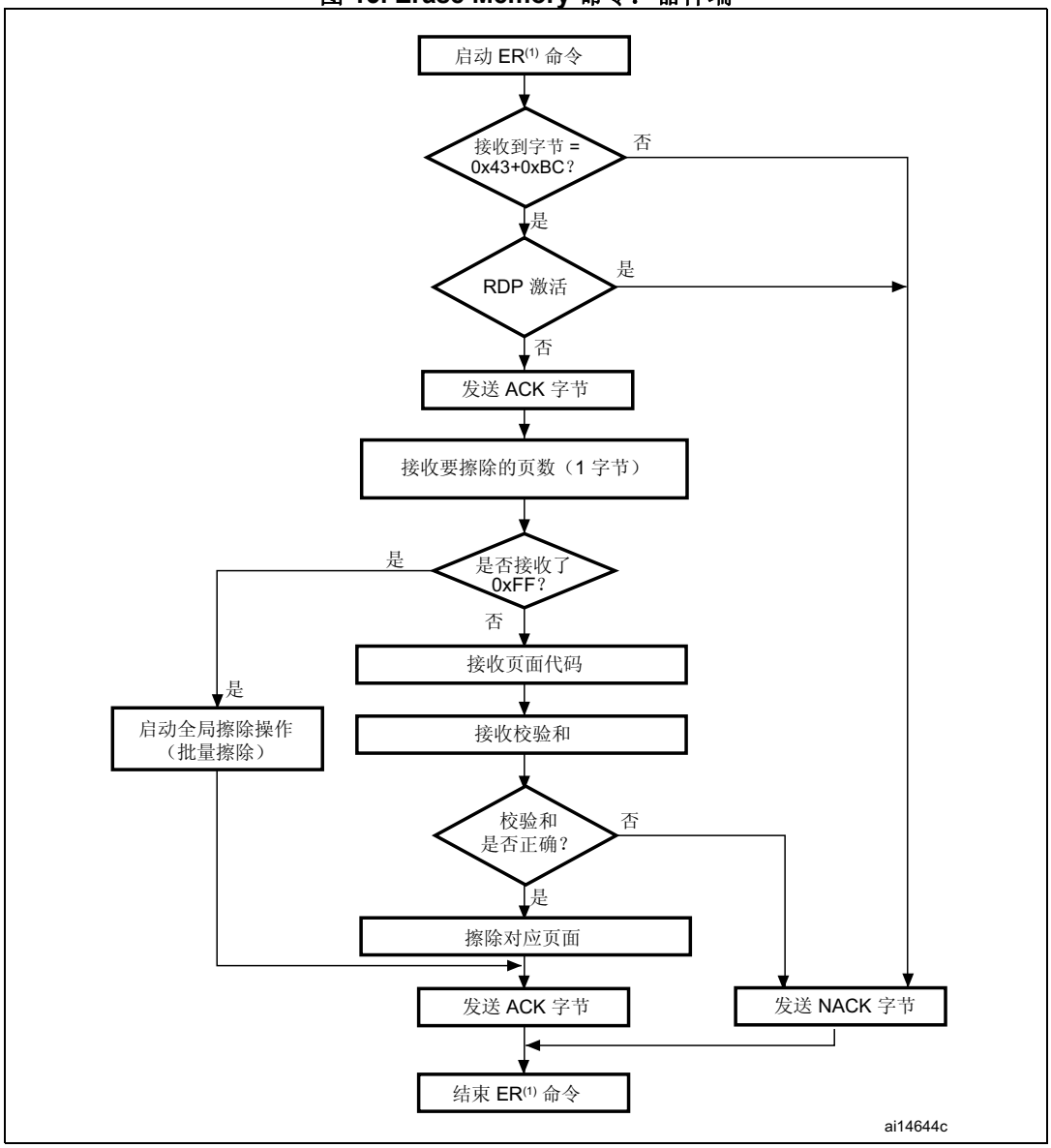
注： 在具有写保护的扇区中执行擦除操作时不会返回错误信息。

图 14. Erase Memory 命令：主机端



1. ER = Erase Memory。

图 15. Erase Memory 命令：器件端



1. ER = Erase Memory。

注： *主机在发送擦除存储器命令及其检验和后，如果发送了 0xFF 并且后面的数据非 0x00，则不会执行批量擦除操作，但相关器件会发送一个 ACK。*

主机将如下字节发送到 STM32：

字节 1： 0x43

字节 2： 0xBC

等待 ACK

字节 3： 0xFF 或待擦除的页数 - 1（0 ≤ N ≤ 最大页数）

字节 4： 0x00（针对全局擦除）或（(N + 1) 字节（页数）以及校验和异或运算结果（N、N+1 字节））

3.9 Extended Erase Memory 命令

主机可通过 Extended Erase Memory 命令使用双字节寻址模式擦除 Flash 页面。自举程序接收到 Extended Erase Memory 命令后，会将 ACK 字节发送到主机。发送 ACK 字节之后，自举程序会接收两个字节（待擦除的页数）、Flash 页面代码（采用双字节编码，MSB 在前）和一个校验和字节（已发送字节的异或运算结果）；若校验和正确，自举程序将擦除存储器并向主机发送一个 ACK 字节。否则将发送一个 NACK 字节到主机并中止命令。

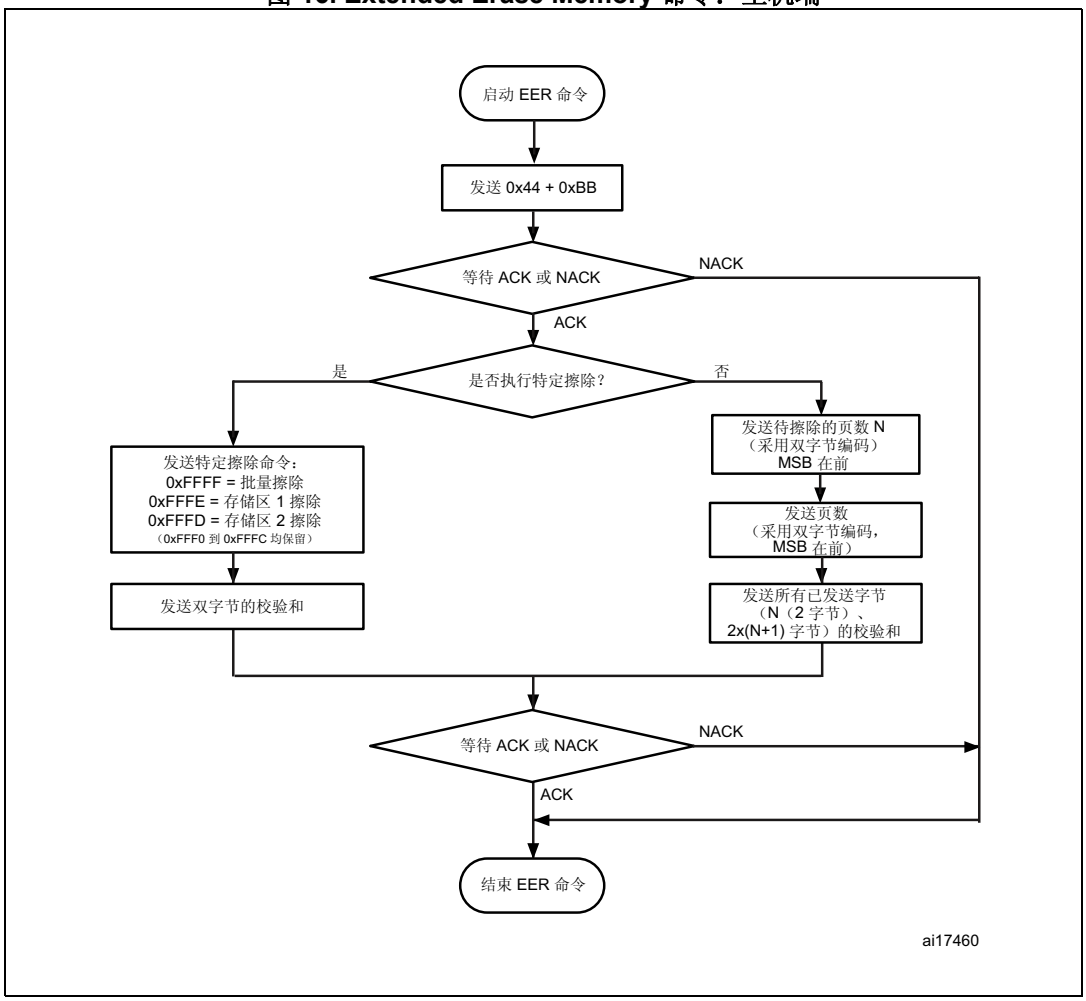
Extended Erase Memory 命令规范：

1. 自举程序接收一个包含 N（要擦除的页数 - 1）的半字（两个字节）：
   a) N = 0xFFFY（Y 范围为 0 到 F）时，执行特定擦除操作：
      - 0xFFFF，执行全局批量擦除
      - 0xFFFE，执行存储区 1 批量擦除
      - 0xFFFD，执行存储区 2 批量擦除
      - 0xFFFC 到 0xFFF0 的代码保留不变
   b) 针对 0 ≤ N < 最大页数之间的其它值：擦除 N + 1 页。
2. 自举程序：
   a) 执行特定擦除时，接收一个字节：之前字节的校验和：
      - 0xFFFF 对应 0x00
      - 0xFFFE 对应 0x01
      - 0xFFFD 对应 0x02
   a) 执行 N+1 页擦除时，自举程序接收 (2 x (N + 1)) 字节，每个半字中包含一个页数（采用双字节编码，MSB 在前）。之后将接收所有之前字节的校验和（一个字节）。

注： *在具有写保护的扇区中执行擦除操作时不会返回错误信息。*

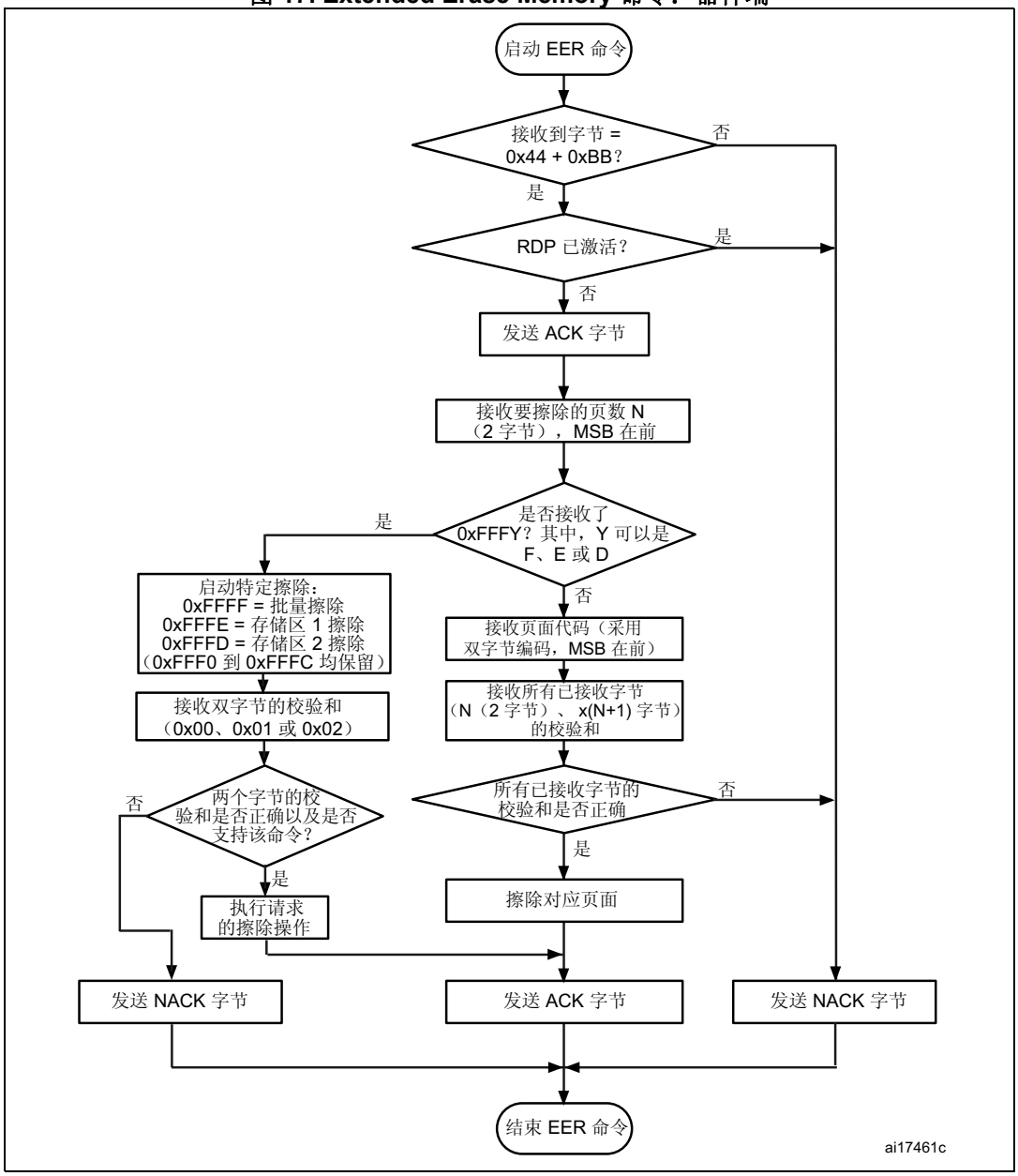
最大页数与产品相关，因此必须遵循。

图 16. Extended Erase Memory 命令：主机端



1. EER = Extended Erase Memory

图 17. Extended Erase Memory 命令：器件端



1. EER = Extended Erase Memory。

主机将如下字节发送到 STM32F1xxx：

字节 1： 0x44

字节 2： 0xBB

等待 ACK

字节 3-4： - 特定擦除（0xFFFF、0xFFFE 或 0xFFFD）

或者

- 待擦除的页数（N+1，其中：0 ≤ N < 最大页数）。

剩余字节 - 执行特定擦除时字节 3-4 的校验和（0xFFFF 对应 0x00、0xFFFE 对应 0x01 或者 0xFFFD 对应 0x02）。

或者

- (2 x (N + 1)) 字节（以双字节编码的页数，MSB 在前），之后接收字节 3-4 及所有后续字节的校验和）

3.10 Write Protect 命令

Write Protect 命令用于为一部分或所有 Flash 扇区使能写保护。自举程序接收到 Write Protect 命令后，会将 ACK 字节发送到主机。发送 ACK 字节后，自举程序将等待要接收的字节数（受保护的扇区），之后从应用程序接收 Flash 扇区代码。

在 Write Protect 命令结束时，自举程序会发送 ACK 字节并启动系统复位，以实施新的选项字节配置。

注： *有关所用器件扇区大小的详细信息，请参见第 3.1 节：器件相关的自举程序参数。*

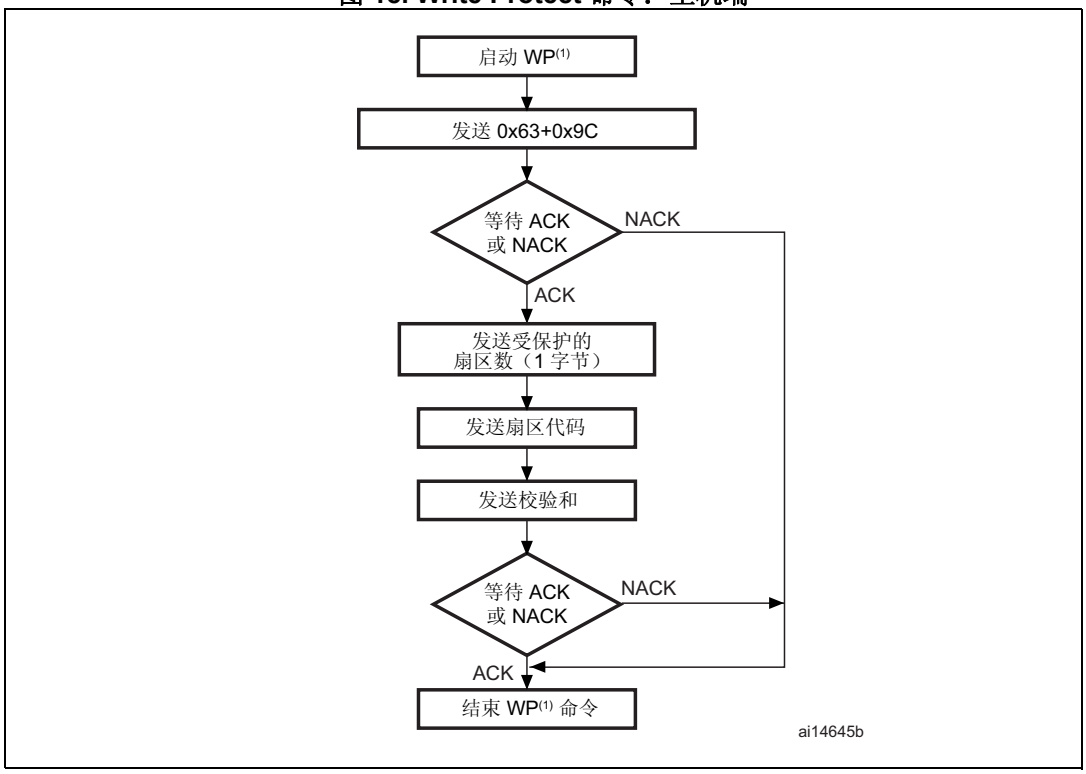
Write Protect 命令序列如下所示：

- 自举程序接收一个包含 N（受写保护的扇区数 - 1）的字节 (0 ≤ N ≤ 255)
- 自举程序接收到 (N + 1) 个字节，每个字节都包含一扇区代码

注： *1 不检查扇区的总数和要保护的扇区号，也就是说，即使给命令传递了错误的待保护扇区数或扇区号，也不会返回错误信息。*

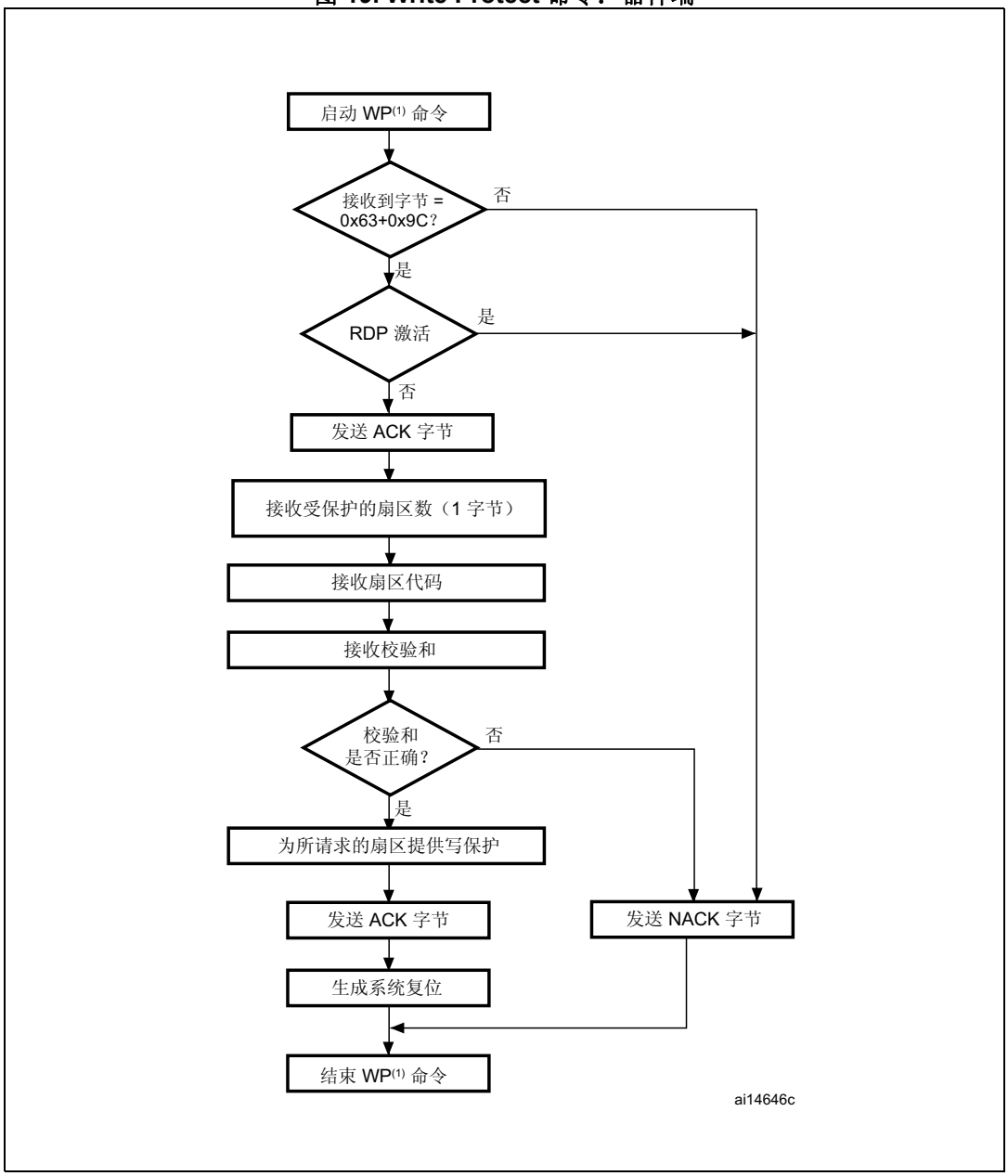
2 如果执行第二个 Write Protect 命令，则取消第一个命令对 Flash 扇区施加的写保护，仅对第二个 Write Protect 命令涉及的扇区提供写保护。

图 18. Write Protect 命令：主机端



1. WP = Write Protect。

图 19. Write Protect 命令：器件端



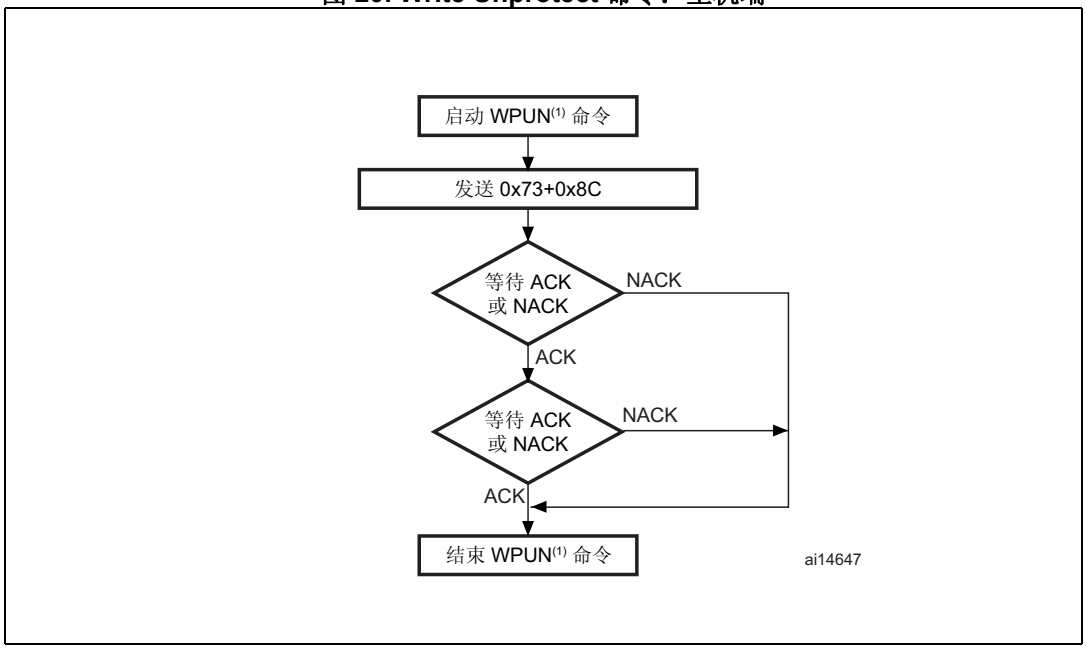
1. WP = Write Protect。

3.11 Write Unprotect 命令

Write Unprotect 命令用于禁止所有 Flash 扇区的写保护。自举程序接收到 Write Unprotect 命令后，会将 ACK 字节发送到主机。发送 ACK 字节后，自举程序会禁止所有 Flash 扇区的写保护。执行禁止保护操作后，自举程序发送 ACK 字节。

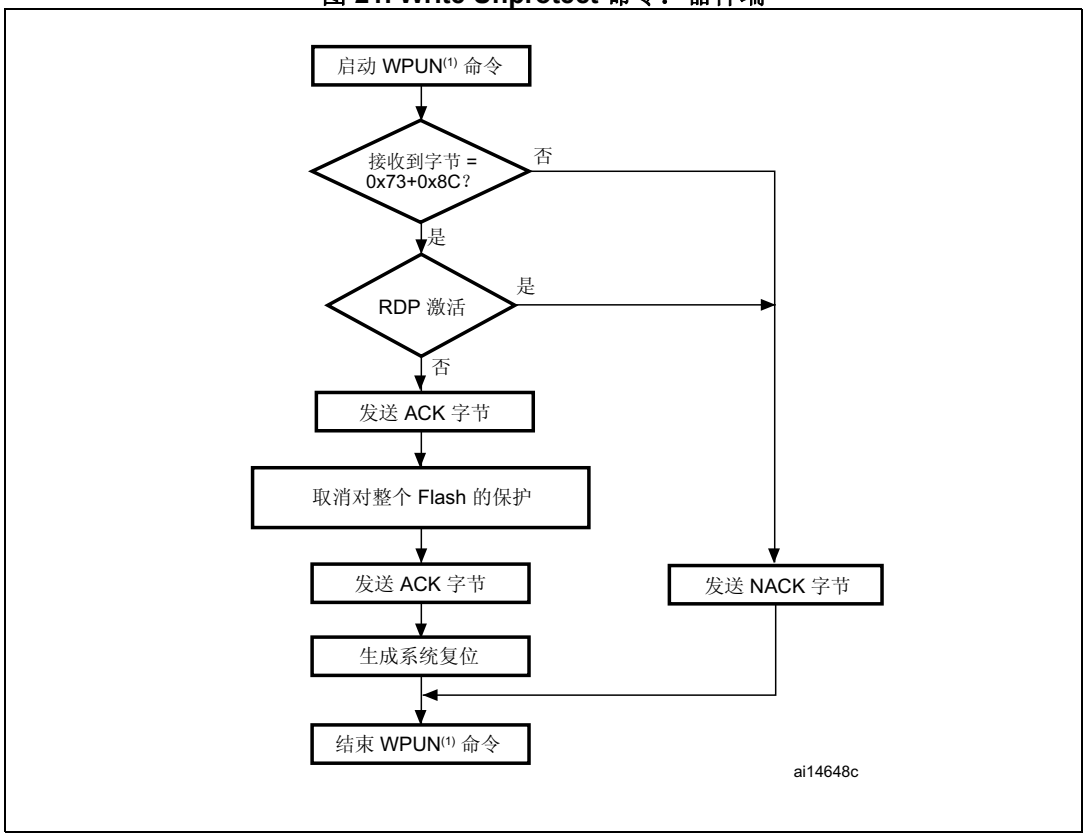
在 Write Unprotect 命令结束时，自举程序会发送 ACK 字节并启动系统复位，以实施新的选项字节配置。

图 20. Write Unprotect 命令：主机端



1. WPUN = Write Unprotect。

图 21. Write Unprotect 命令：器件端



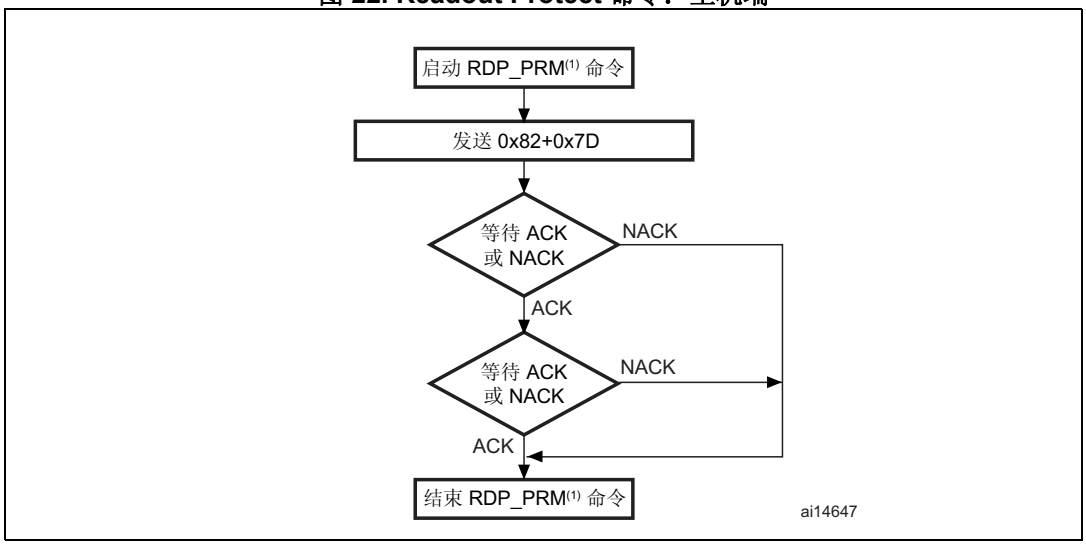
1. WPUN = Write Unprotect。

3.12 Readout Protect 命令

Readout Protect 命令用于使能 Flash 读保护。自举程序接收到 Readout Protect 命令后，会将 ACK 字节发送到主机。发送 ACK 字节后，自举程序将使能 Flash 的读保护。

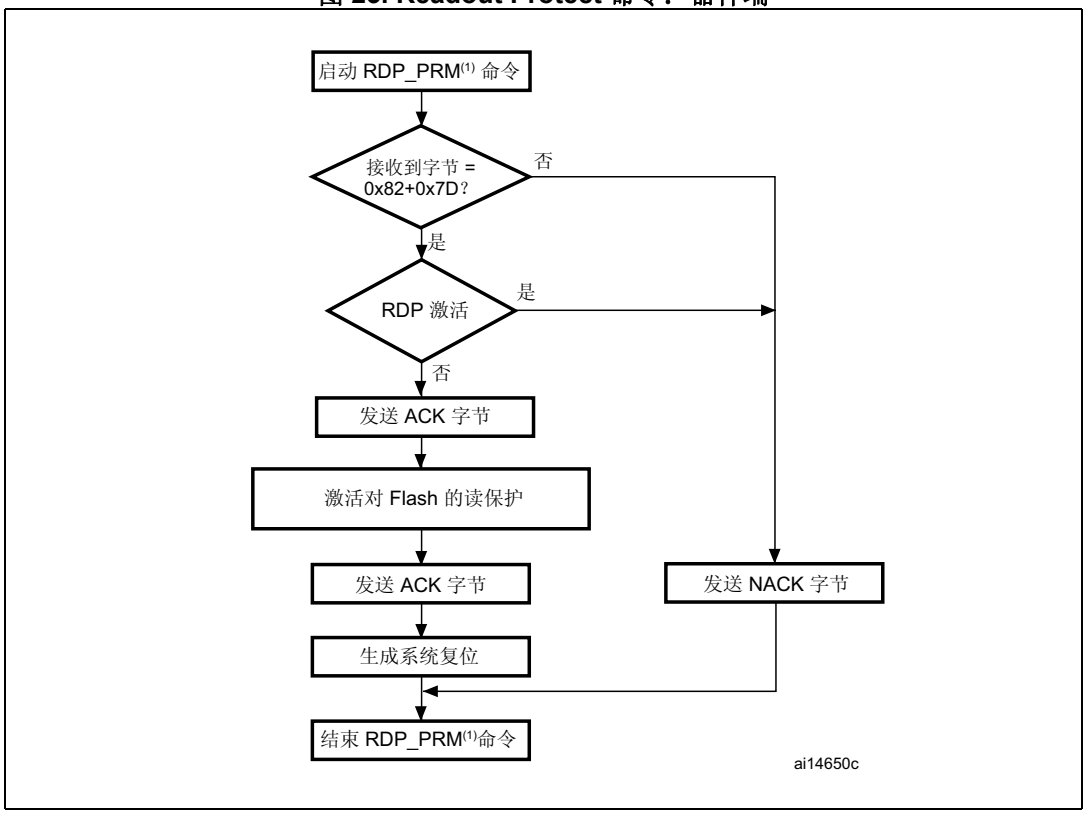
在 Readout Protect 命令结束时，自举程序会发送 ACK 字节并启动系统复位，以实施新的选项字节配置。

图 22. Readout Protect 命令：主机端



1. RDP_PRM = Readout Protect。

图 23. Readout Protect 命令：器件端



1. RDP_PRM = Readout Protect。

3.13 Readout Unprotect 命令

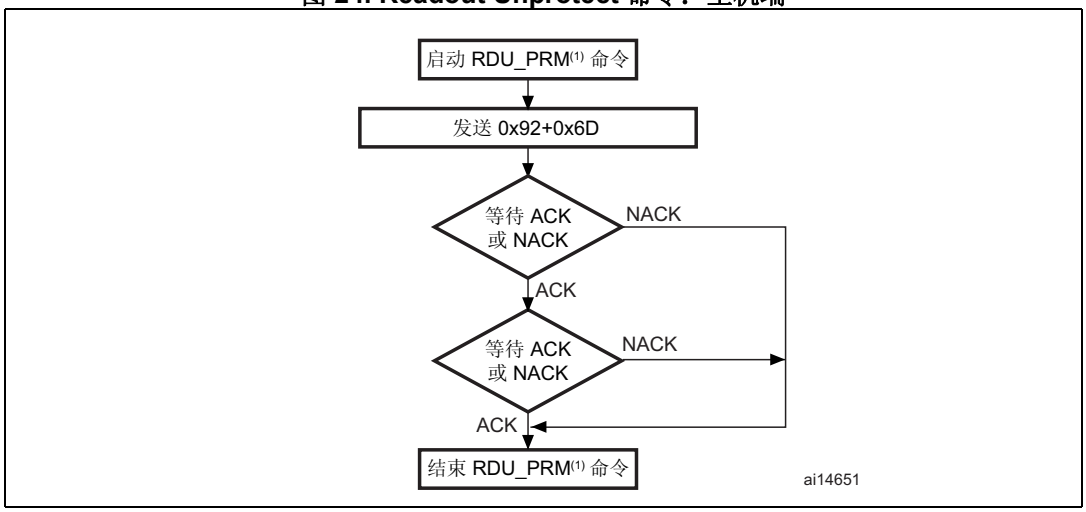
Readout Unprotect 命令用于禁止 Flash 读保护。自举程序接收到 Readout Unprotect 命令后，会将 ACK 字节发送到主机。发送了 ACK 字节后，自举程序将擦除所有 Flash 扇区并禁止整个 Flash 的读保护。如果擦除操作成功完成，自举程序将停用 RDP。

如果擦除操作失败，自举程序会发送一个 NACK 且读保护仍然有效。

在 Readout Unprotect 命令结束时，自举程序会发送 ACK 字节并启动系统复位，以实施新的选项字节配置。

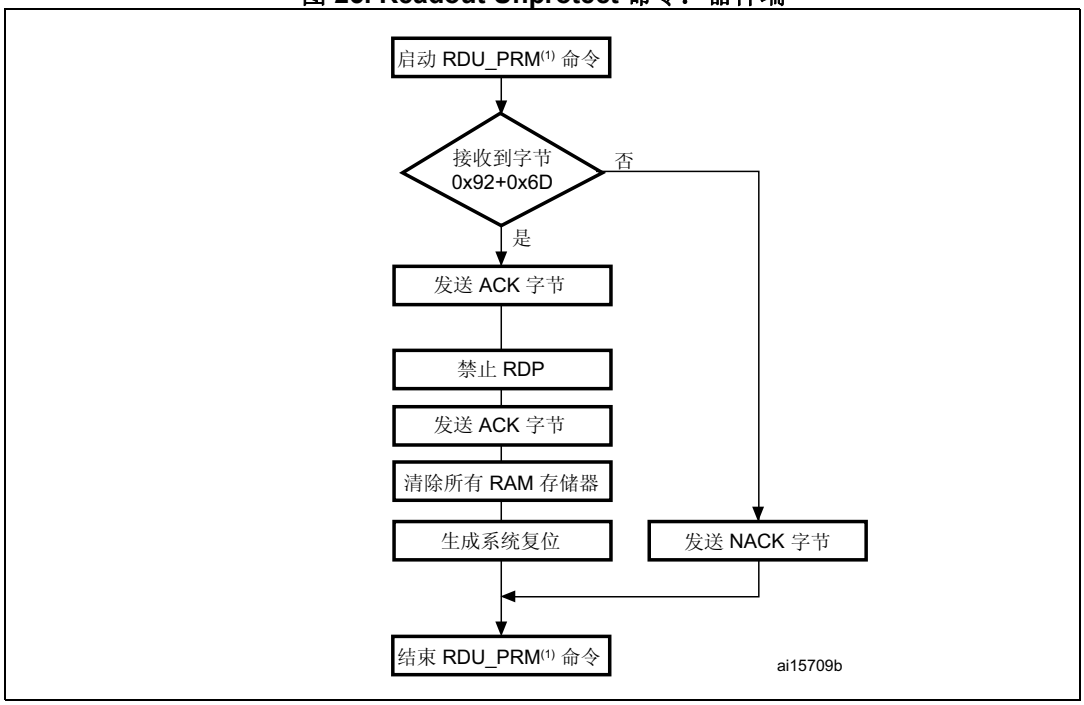
注： *对于大多数 STM32 产品，禁止读保护操作后会导致 flash 批量擦除，因此，主机在发送第二个 ACK 之后需要等待充分的时间，然后才会重启连接。如需了解此操作所需时间，请参见产品数据手册中的批量擦除时间（如相关）。*

图 24. Readout Unprotect 命令：主机端



1. RDU_PRM = Readout Unprotect。

图 25. Readout Unprotect 命令：器件端



2. RDU_PRM = Readout Unprotect。

4 自举程序协议版本演化

表 3 列出了自举程序的各个版本。

表 3. 自举程序协议版本

版本	说明
V2.0	自举程序初始版本。
V2.1	– 更新 Go 命令以初始化主栈指针 – 更新 Go 命令，以在跳转地址处于选项字节区域或系统存储器区域时返回 NACK – 更新 Get ID 命令以返回双字节格式的器件 ID – 将自举程序版本更新至 V2.1
V2.2	– 更新 Read Memory、Write Memory 和 Go 命令，以通过 NACK 响应拒绝对自举程序所用的 RAM 存储器首字节的访问 – 更新 Readout Unprotect 命令，以在禁止 RDP 之前将整个 RAM 内容初始化为 0x0
V3.0	– 新增了 Extended Erase 命令，以支持大于 256 的页数和单独的存储区批量擦除操作。 – 不过，由于新增了 Extended Erase 命令，此版本虽未对 Erase 命令进行修改但也不再支持该命令（Erase 和 Extended Erase 命令均为独占命令）。
V3.1	– 修正了以下缺陷： “当通过一个不受支持的存储器地址和一个正确的地址检验和（即，地址 0x6000 0000）发出 Read Memory 命令或 Write Memory 命令时，自举程序器件会中止这一命令，但不会向主机发送 NACK (0x1F)。因此，接下来的两个字节（即，待读/写的字节数及其校验和）会被看作一条新命令及其校验和。”(1) 产品规范无更改，产品实现已经过修正。

1. 如果待读 / 写的“数据数 - 1”(N-1) 不等于有效命令代码（0x00、0x01、0x02、0x11、0x21、0x31、0x43、0x44、0x63、0x73、0x82 或 0x92），则无法从主机发现缺陷，因为该命令始终会收到 NACK 应答（当作不受支持的新命令）。

5 版本历史

表 4. 文档版本历史

日期	版本	变更
2010 年 03 月 09 日	1	初始版本。
2010 年 04 月 20 日	2	*表 2：USART 自举程序命令*：新增了 Extended Erase 命令；删除了 Readout Protect 命令中关于读保护的脚注 2。 *通信安全*：修订了*注 1*。 *第 3.2 节：Get 命令*：更新了字节 10。 针对缺少 ACK 状态更新了*图 10：Go 命令：主机端*。 *第 3.7 节：Write Memory 命令*：新增了*注 1* 和*注 2*。 *图 12* 和*图 13*：新增了关于 N+1 的注释。 新增了*第 3.9 节：Extended Erase Memory 命令*。 *表 3：自举程序协议版本*：新增了 v3.0。
2013 年 02 月 15 日	3	新增了*注：*、*注 4* 和*注：*。 将下列图中的所有“ROP”更改为“RDP”：*图 9*、*图 11*、*图 13*、*图 15*、*图 17*、*图 19*、*图 21*、*图 23*、*图 25*。 新增了*表 1：适用的产品*。
2013 年 03 月 26 日	4	*表 3：自举程序协议版本*中新增了 3.1 版。 *第 3.4 节：Get ID 命令*中更新了“字节 4”值。 在此*注 1* 中用“ID”替换了“地址”。 在*图 10：Go 命令：主机端*中用“结束 Go 命令”替换了“结束 EER 命令”。 更新了*第 3.7 节：Write Memory 命令*中的首句。 *图 13：Write Memory 命令：器件端*中删除了“& 地址=0x1FFF F800”并用单个测试替换了“Flash 地址？”和“RAM 地址？”这两个测试。 明确了*图 17：Extended Erase Memory 命令：器件端*中第三个测试缺少的“Y”值的情况。 在*图 24：Readout Unprotect 命令：主机端*上方新增了此*注：*。
2013 年 05 月 22 日	5	在*表 1：适用的产品*中用“STM32L1 系列”替换了“STM32L151xx、STM32L152xx 和 STM32L162xx”。

请仔细阅读：

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与ST产品有关。意法半导体公司及其子公司（“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有ST产品均根据ST的销售条款出售。

买方自行负责对本文所述ST产品和服务的选择和使用，ST概不承担与选择或使用本文所述ST产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为 ST 授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在ST的销售条款中另有说明，否则，ST对ST产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且/或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML或JAN正式认证产品适用于航天应用。

经销的ST产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致ST针对本文所述ST产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大ST的任何责任。

ST和ST徽标是ST在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 – 菲律宾 – 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com