AN4286
应用笔记

SPI 自举程序中使用的 I2C 协议

前言

本应用笔记说明了 STM32 微控制器自举程序中使用的 SPI 协议。它详细说明了每个支持的指令。

若需器件自举程序 SPI 硬件资源和要求的更多信息，请参考应用笔记 “STM32 微控制器系统存储器自举模式 ”（AN2606）。

本文档适用于*表 1* 中所列产品。

表 1. 适用产品

产品系列	产品系列
微控制器	STM32L0 系列： – STM32L051xx、 STM32L052xx、 STM32L053xx – STM32L062xx、 STM32L063xx STM32F4 系列： – STM32F401xx、 STM32F411xx – STM32F405xx、 STM32F407xx – STM32F415xx、 STM32F417xx – STM32F429xx、 STM32F439xx

目录

表格索引

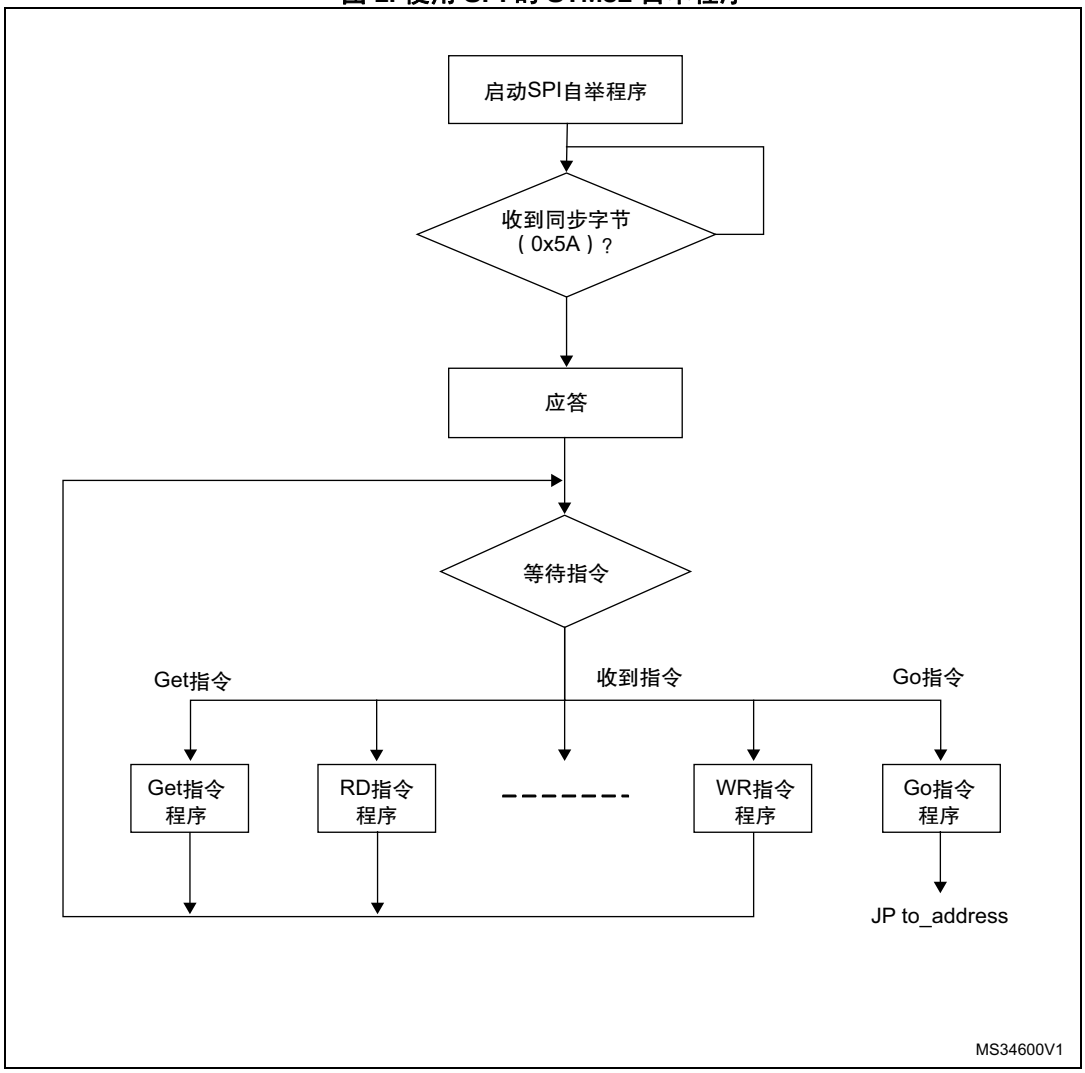
图片索引

1 SPI 自举程序代码序列

STM32 自举程序为 SPI 从机。

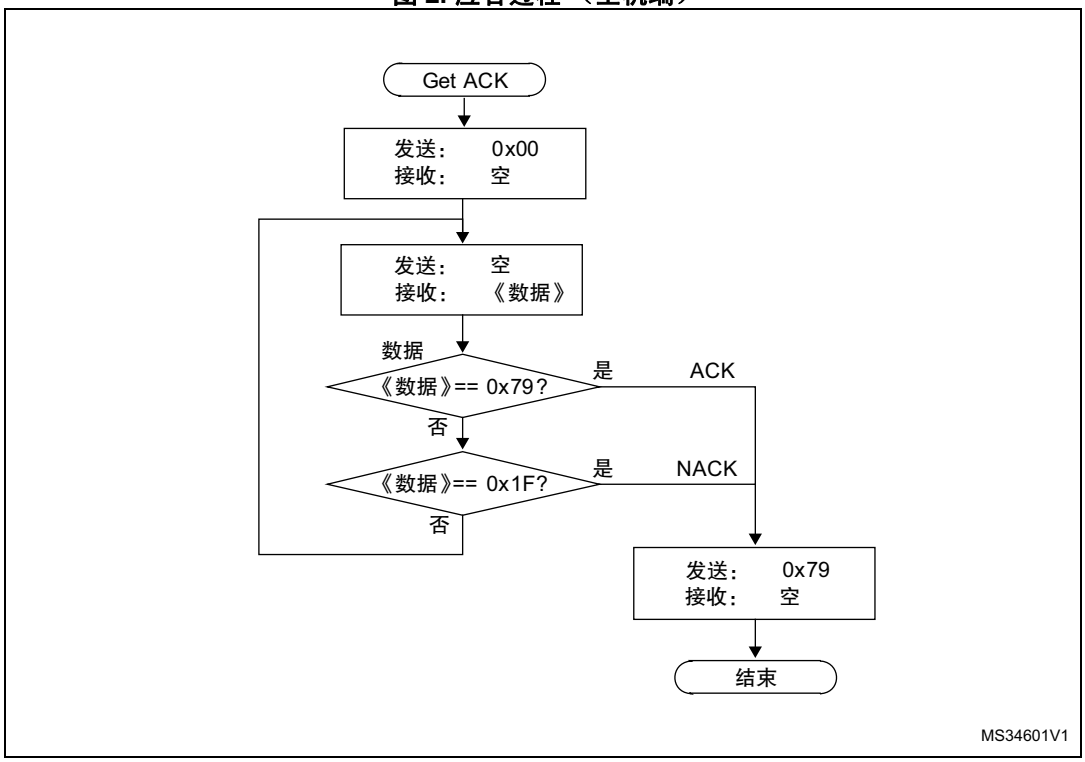
对于所有 SPI 自举程序操作， NSS 引脚 （片选）必须连至低电平。若 NSS 引脚连至高电平，则 STM32 从机将忽略 SPI 总线上的通信。

图 1. 使用 SPI 的 STM32 自举程序



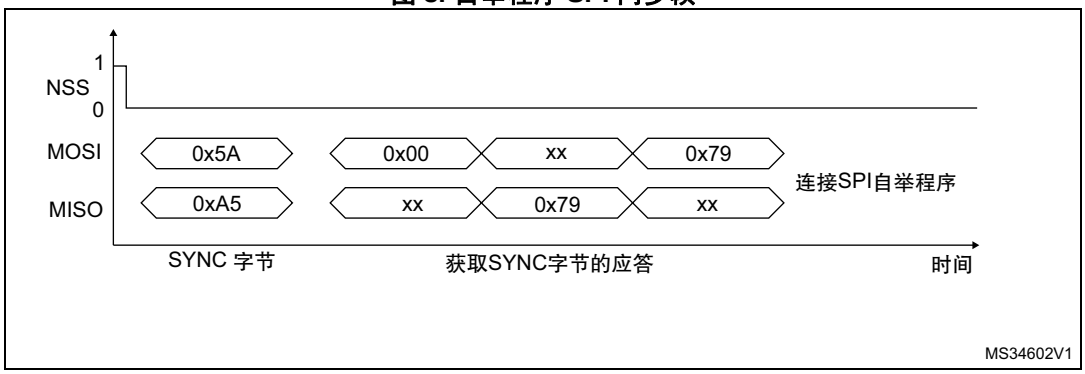
进入系统存储器自举模式后，若 STM32 微控制器已配置好 （若需更详细信息，请参考您的 STM32 系统存储器自举模式应用笔记），自举程序代码开始扫描 SPI_MOSI 线引脚，等待检测总线上的同步字节 （0x5A）。当检测到时， SPI 自举程序固件会等待接收应答过程 （请参考*图 2*），然后开始接收主机指令。

图 2. 应答过程 （主机端）



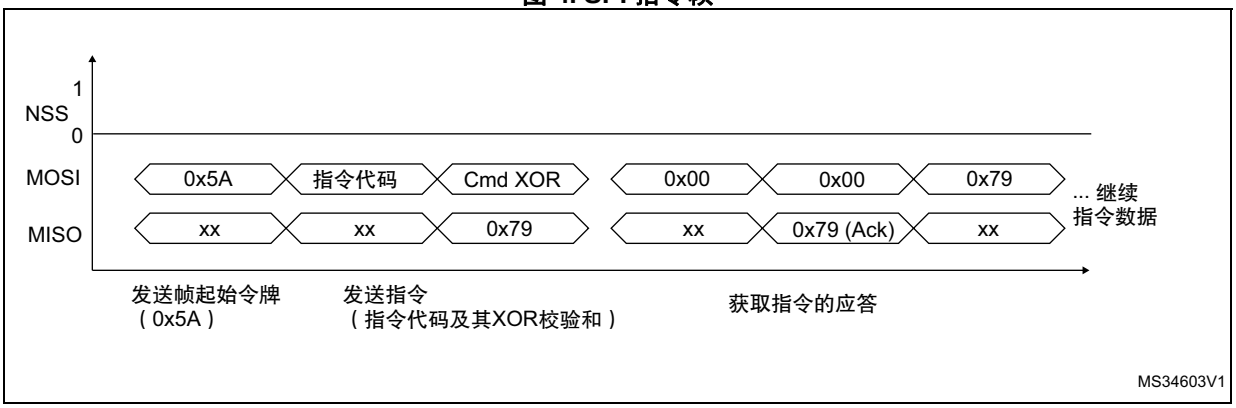
为开始与自举程序通信，主机必须首先发送一个同步字节 （0x5A），然后等待接收应答（ACK）。

图 3. 自举程序 SPI 同步帧



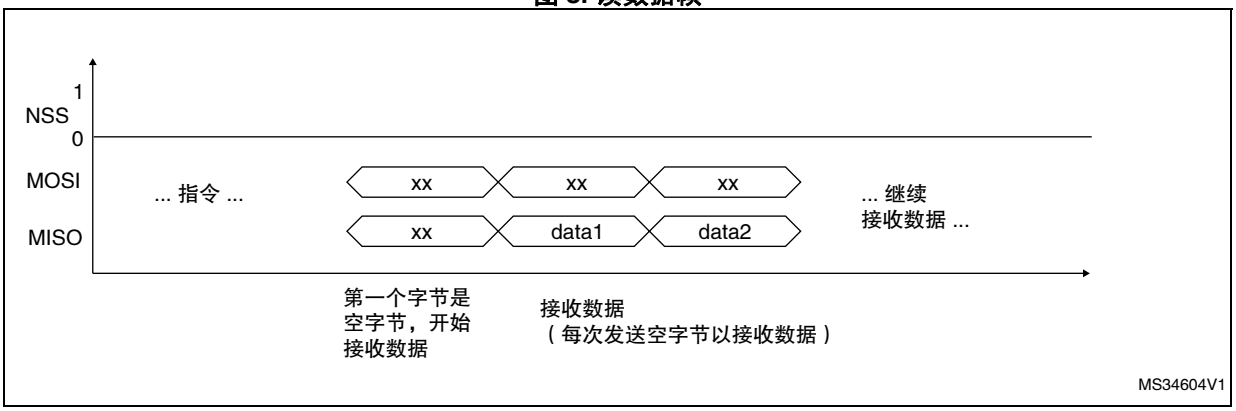
其中 xx 表示空字节。

图 4. SPI 指令帧



若需读取任何数据，主机应在开始读取从机发送的数据之前发送一个空字节。当需要读取时，所有指令都应如此。

图 5. 读数据帧



2 自举程序指令集

表 2 列出了所支持的指令。每个指令都将在本章进一步讲述。

表 2. SPI 自举程序指令

指令(1)	*指令代码*	*指令说明*
Get(2)	0x00	获取版本和当前版本所支持的指令。
Get Version(2)	0x01	获取自举程序版本。
Get ID(2)	0x02	获取芯片 ID。
Read Memory(3)	0x11	从存储器读取最多 256 字节，由应用指定起始地址。
Go(3)	0x21	跳转到内部 Flash 中的用户应用代码。
Write Memory(3)	0x31	向存储器写入最多 256 字节，由应用指定起始地址。
Erase(3)	0x44	使用双字节寻址模式擦除一至所有 Flash 页面或扇区。
Write Protect	0x63	对一些扇区启用写保护。
Write Unprotect	0x73	对所有 Flash 扇区禁用写保护。
Readout Protect	0x82	启用读保护。
Readout Unprotect(2)	0x92	禁用读保护。

1. 若收到了拒绝指令，或指令执行期间发生了错误，则自举程序会发送 NACK 字节，然后返回指令检查。
2. 读保护 - 当 RDP （读保护）选项激活时，仅能使用此有限子集的指令。所有其它指令都会被 NACK，对从机没有作用。取消 RDP 之后，其它指令变为激活。
3. 若需了解哪些存储器空间可执行该指令，请参考 STM32 产品数据手册和 AN2606：STM32 微控制器系统存储器自举模式。

因为 SPI 配置为全双工，所以每次主机在 MOSI 线上发送数据时，它都会同时从 MISO 线上接收数据。因为从机的应答不是立即的，所以当主机发送时，它接收的数据将被忽略 （空）（此数据不被主机使用）。

当从机需要发送数据时，主机会发送它的时钟，因此它必须在 MOSI 线上发送数据，才能从 MISO 线上接收从机数据。在此情况下，主机应一直发送 0x00 （从机不使用此数据）。

2.1 通信安全

从编程主机到从机的所有通信都经过以下方式验证。

- 校验和：接收的数据字节块都经过异或计算。所有字节异或计算后算出一个字节，加到每次通信的末尾 （校验和字节）。对所有收到的字节 —— 数据 + 校验和 —— 做异或计算，最后结果必须为 0x00。
- 若接收的数据为一个字节，则它的校验和为该值的按位取反 （0x02 的校验和为 0xFD）。
- 对于每个指令，主机都发送三个字节：一个帧开始 （SOF = 0x5A）字节、一个表示指令值的字节及其补码 （指令与其补码的异或 = 0x00）。
- 每个包都被接受 （ACK 应答）或丢弃 （NACK 应答）。
  - ACK = 0x79
  - NACK = 0x1F

注： *主机的帧可为下列之一：*

发送指令帧：主机作为主发送端发起通信，向从机发送两字节：命令代码 + Xor。

等待 ACK/NACK 帧：主机作为主接收端发起 SPI 通信，从从机接收一个字节：ACK 或 NACK。

接收数据帧：主机作为主接收端发起 SPI 通信，从从机收到响应。收到的字节数取决于指令。

发送数据帧：主机作为主发送端发起 SPI 通信，向从机发送需要的字节。发送的字节数取决于指令。

2.2 *Get* 指令

Get 指令可帮您得到自举程序版本及所支持的指令。当自举程序收到 Get 指令时，它将自举程序版本和所支持的指令代码发送给主机，如*图 6* 中所示。

图 6. Get 指令：主机端

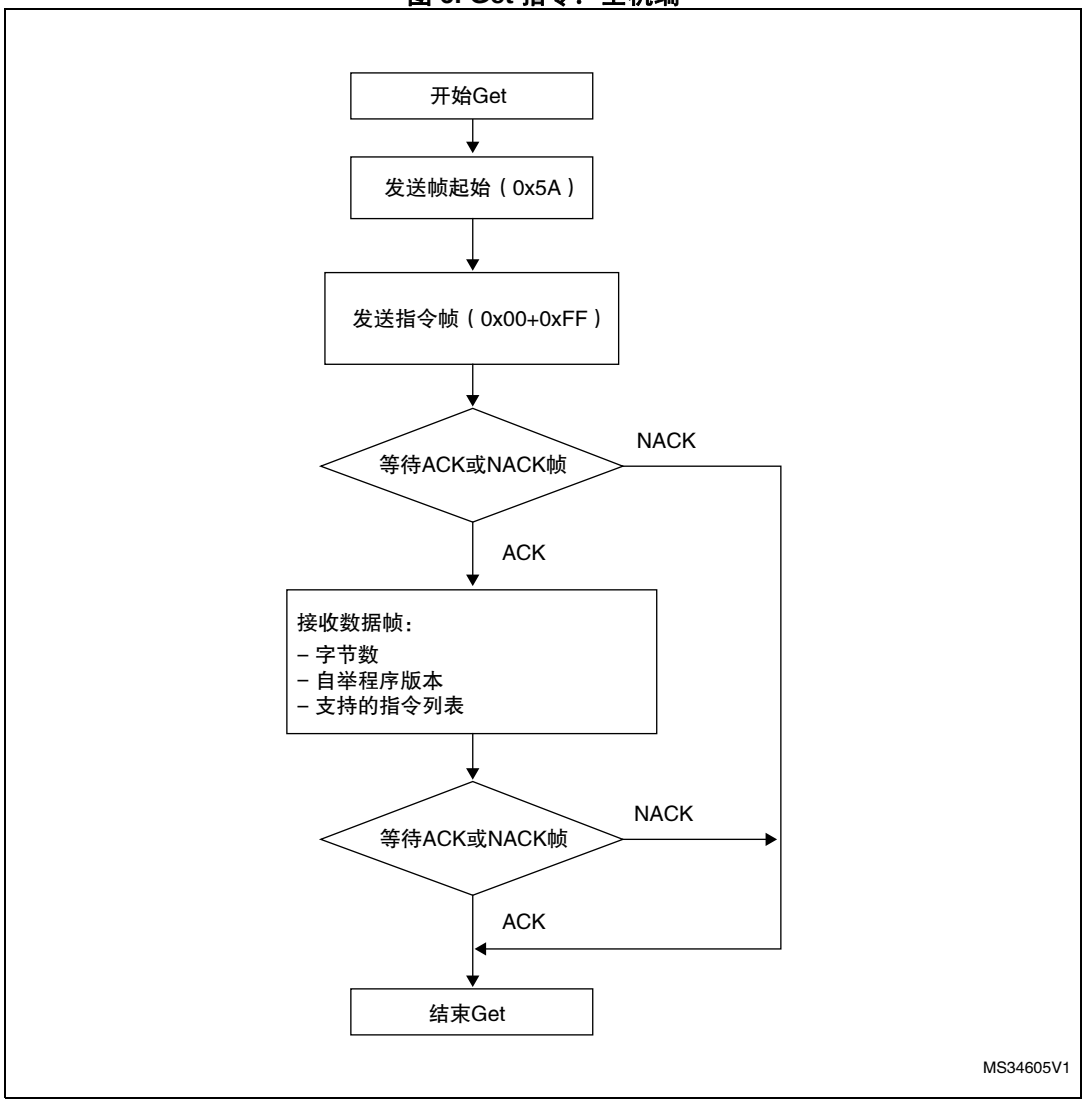
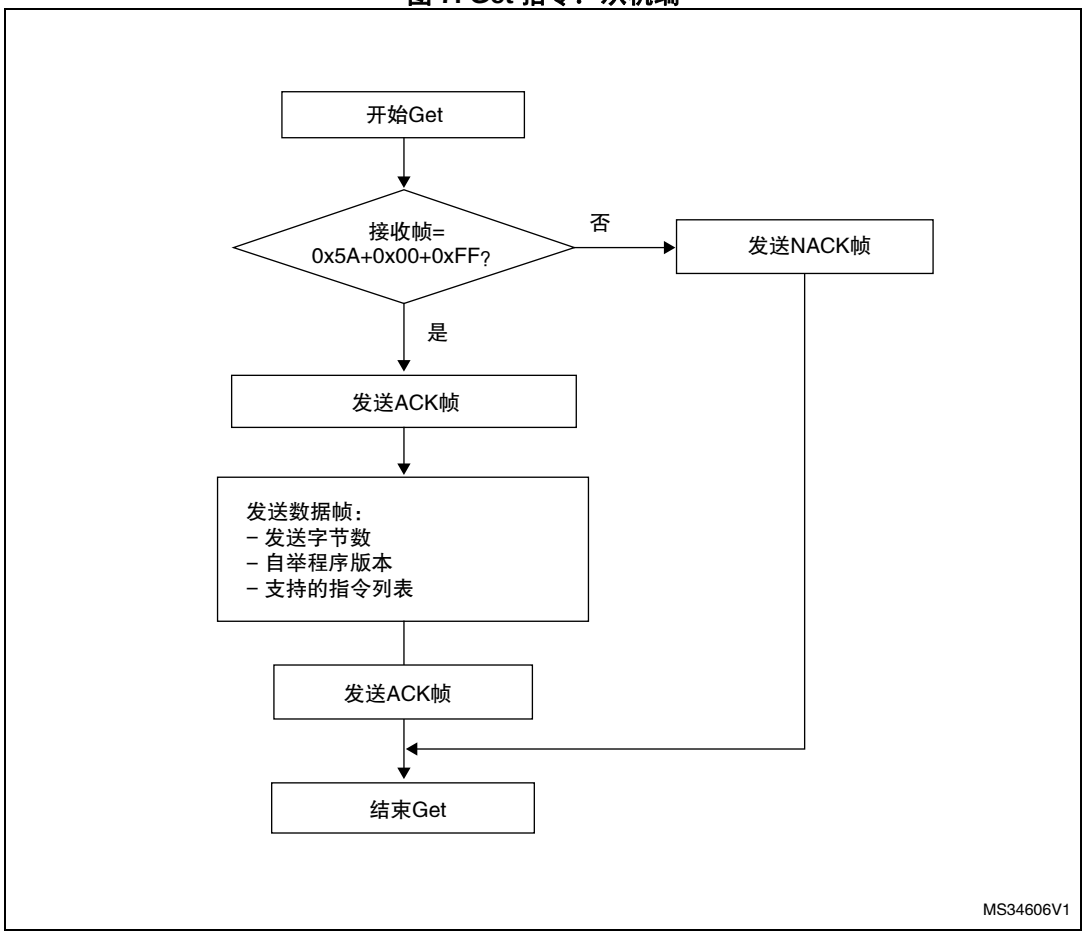


图 7. Get 指令：从机端



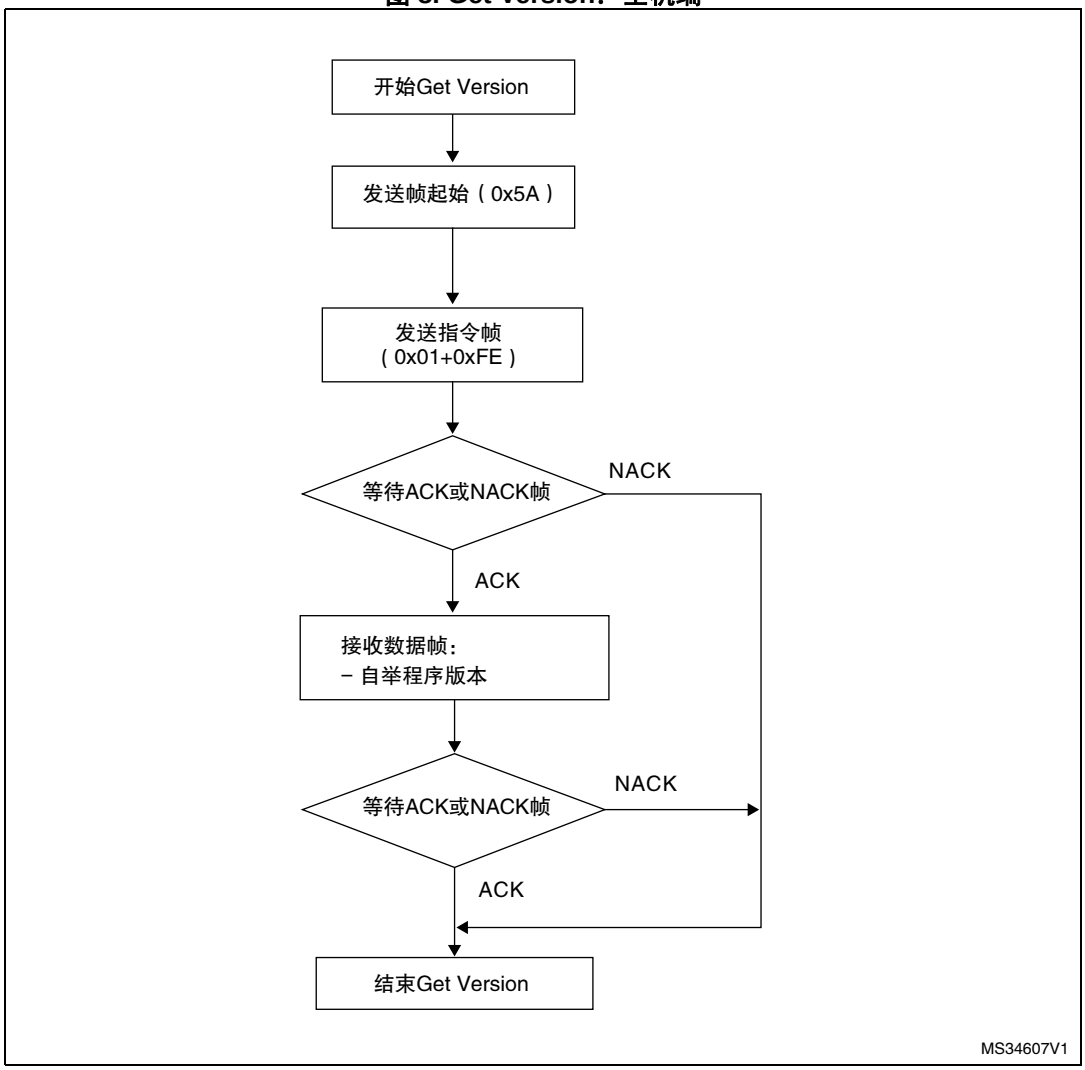
STM32 发送的字节如下。

- 字节 1：ACK
- 字节 2：N = 11 = 后续字节数 - 1，不包括当前字节和 ACK。
- 字节 3：自举程序版本 （0 < 版本 < 255），例如：0x10 = 1.0 版本
- 字节 4：0x00 (Get 指令)
- 字节 5：0x01 (Get Version)
- 字节 6：0x02 (Get ID)
- 字节 7：0x11 (Read Memory 指令)
- 字节 8：0x21 (Go 指令)
- 字节 9：0x31 (Write Memory 指令)
- 字节 10：0x44 (Erase 指令)
- 字节 11：0x63 (Write Protect 指令)
- 字节 12：0x73 (Write Unprotect 指令)
- 字节 13：0x82 (Readout Protect 指令)
- 字节 14：0x92 (Readout Unprotect 指令)

2.3 *Get Version* 指令

Get Version 指令用于得到 SPI 协议的版本。当自举程序收到该指令时，它会向主机发送如下信息 （自举程序版本）。

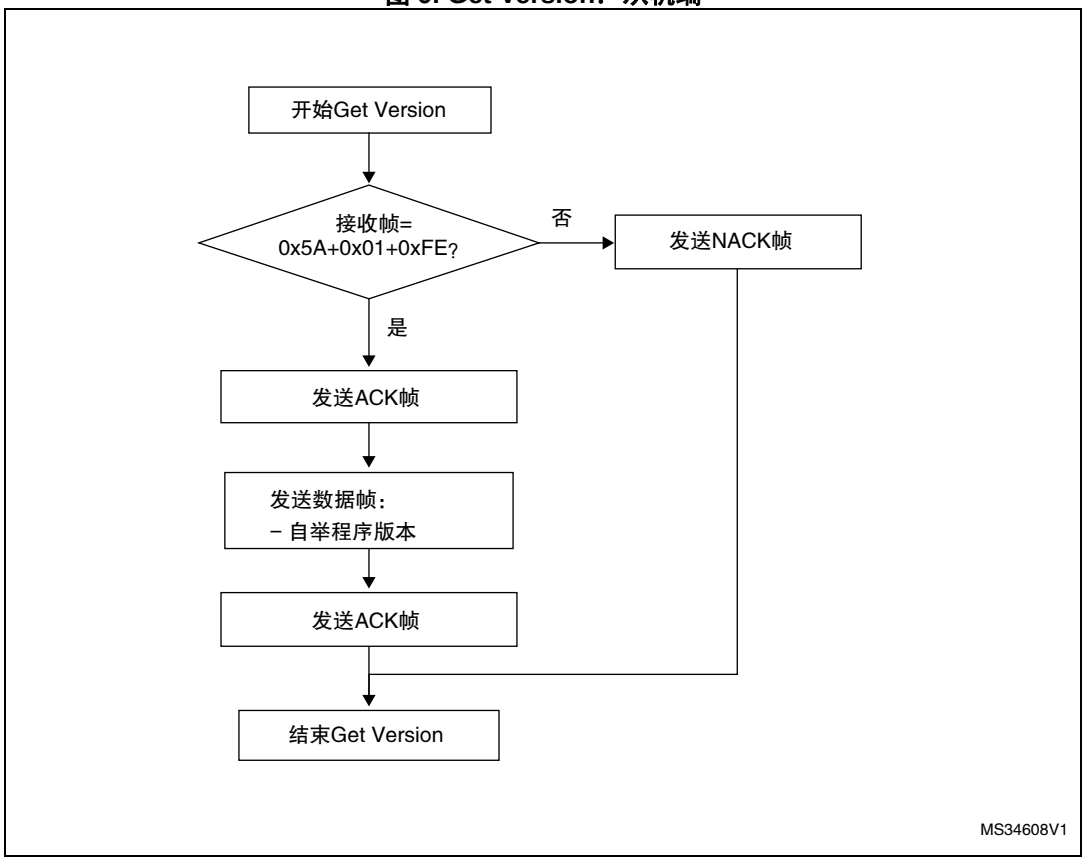
图 8. Get Version：主机端



STM32 发送的字节如下：

- 字节 1：ACK
- 字节 2：自举程序版本 （0 < 版本 ≤ 255），例如：0x10 = 1.0 版本
- 字节 3：ACK

图 9. Get Version：从机端



2.4 *Get ID* 指令

Get ID 指令用于得到芯片 ID （标识）的版本。当自举程序收到该指令时，它会向主机发送产品 ID。

STM32 从机发送的字节如下。

- 字节 1：ACK
- 字节 2：N = 字节数 - 1 （N = 1），不包括当前字节和 ACK。
- 字节 3-4：PID
  - 字节 3 = MSB
  - 字节 4 = LSB
- 字节 5：ACK

图 10. Get ID 指令：主机端

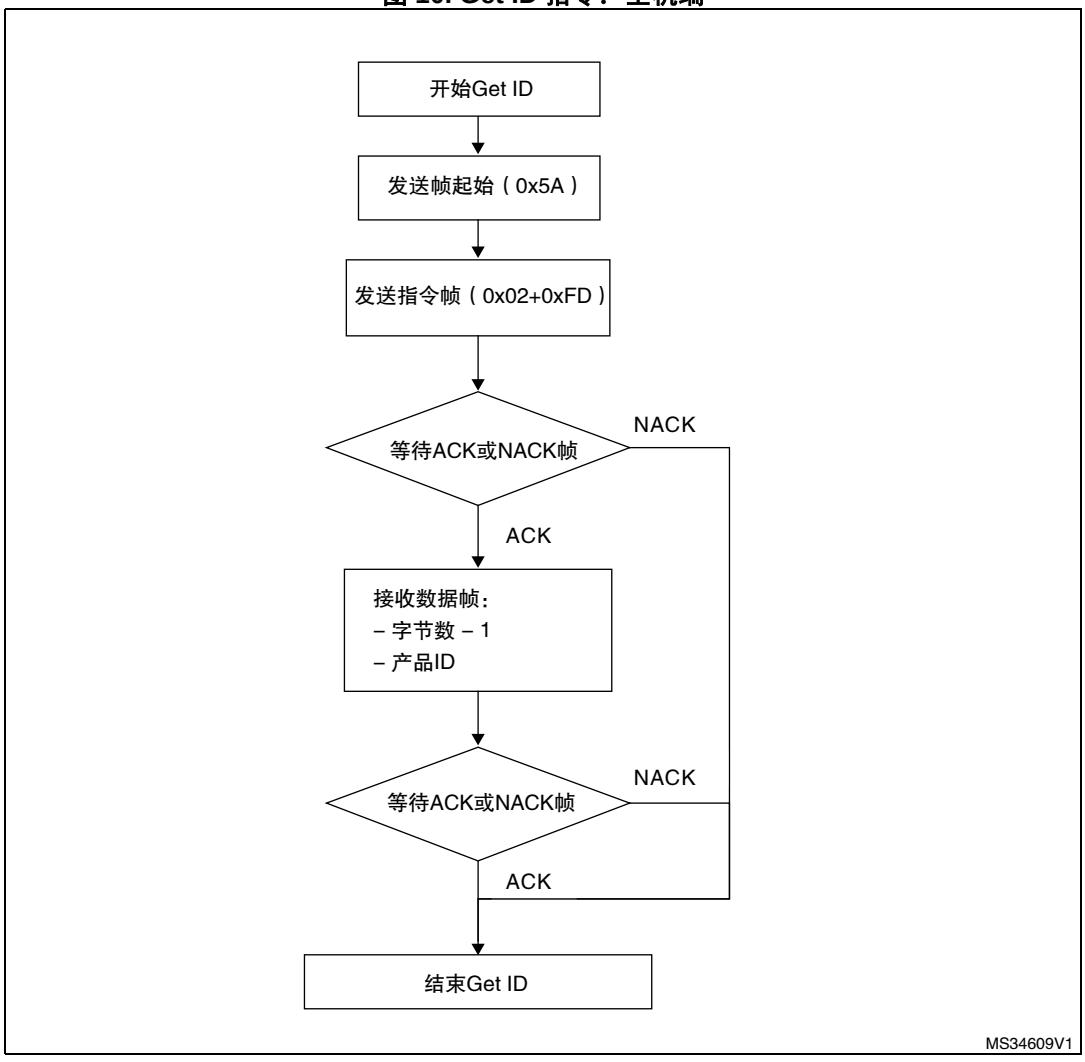
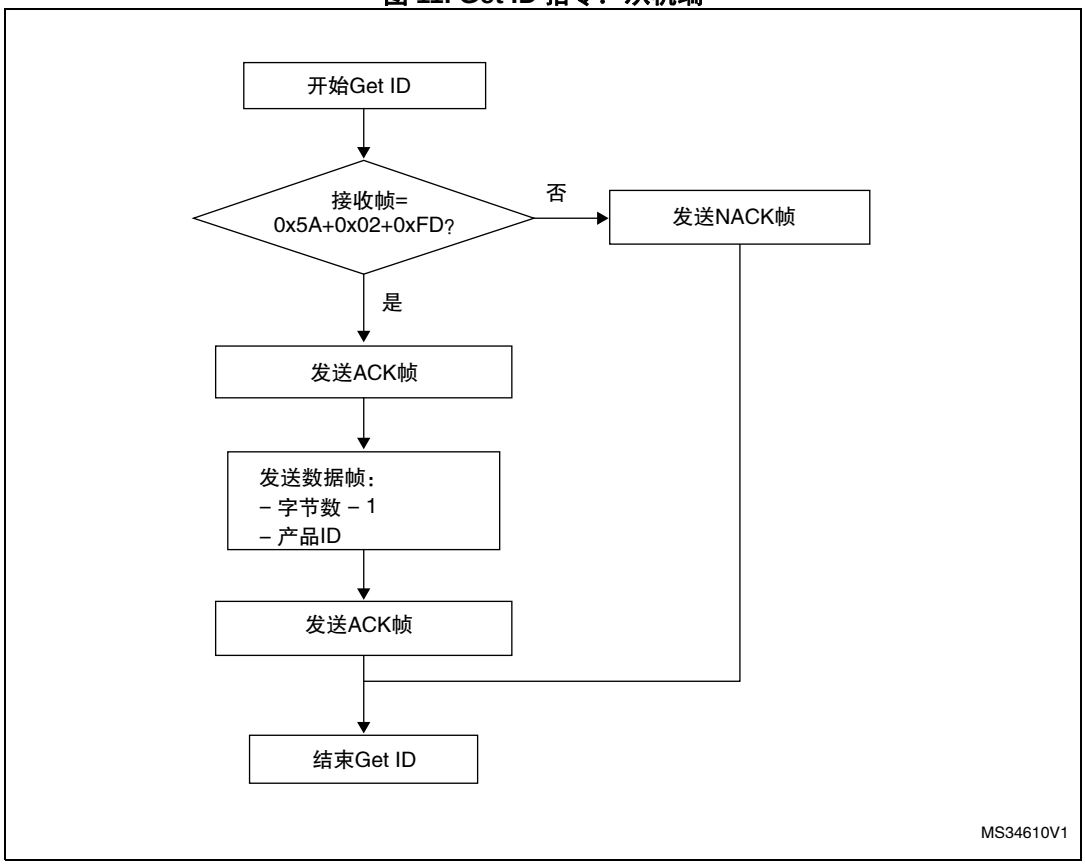


图 11. Get ID 指令：从机端



2.5 *Read Memory* 指令

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

当自举程序收到 Read Memory 指令时，它会向应用发送 ACK 字节。发送 ACK 字节后，自举程序等待一个地址 （4 字节，字节 1 为地址 MSB，字节 4 为 LSB）及校验和字节，之后它会检查收到的地址。若地址有效且校验和正确，则自举程序发送 ACK 字节；否则它发送 NACK 字节并终止该指令。

若地址有效且校验和正确，则自举程序等待要发送的字节数 （N 字节）及其补码字节 （校验和）。若校验和正确，则它从收到的地址开始向应用发送需要的数据。若校验和不正确，则它会在终止指令之前发送 NACK。

主机向 STM32 发送的字节如下。

- 帧起始：0x5A
- 字节 1-2：0x11+0xEE
- 等待 ACK （如第 1 章中所述）
- 字节 3 至 6：起始地址 （字节 3：MSB，字节 6：LSB）
- 字节 7：校验和：XOR （字节 3、字节 4、字节 5、字节 6）
- 等待 ACK （如第 1 章中所述）
- 字节 8：要读取的字节数 - 1 （0 < N ≤ 255）；
- 字节 9：校验和：XOR 字节 8 （字节 8 的补码）

图 12. Read Memory 指令：主机端

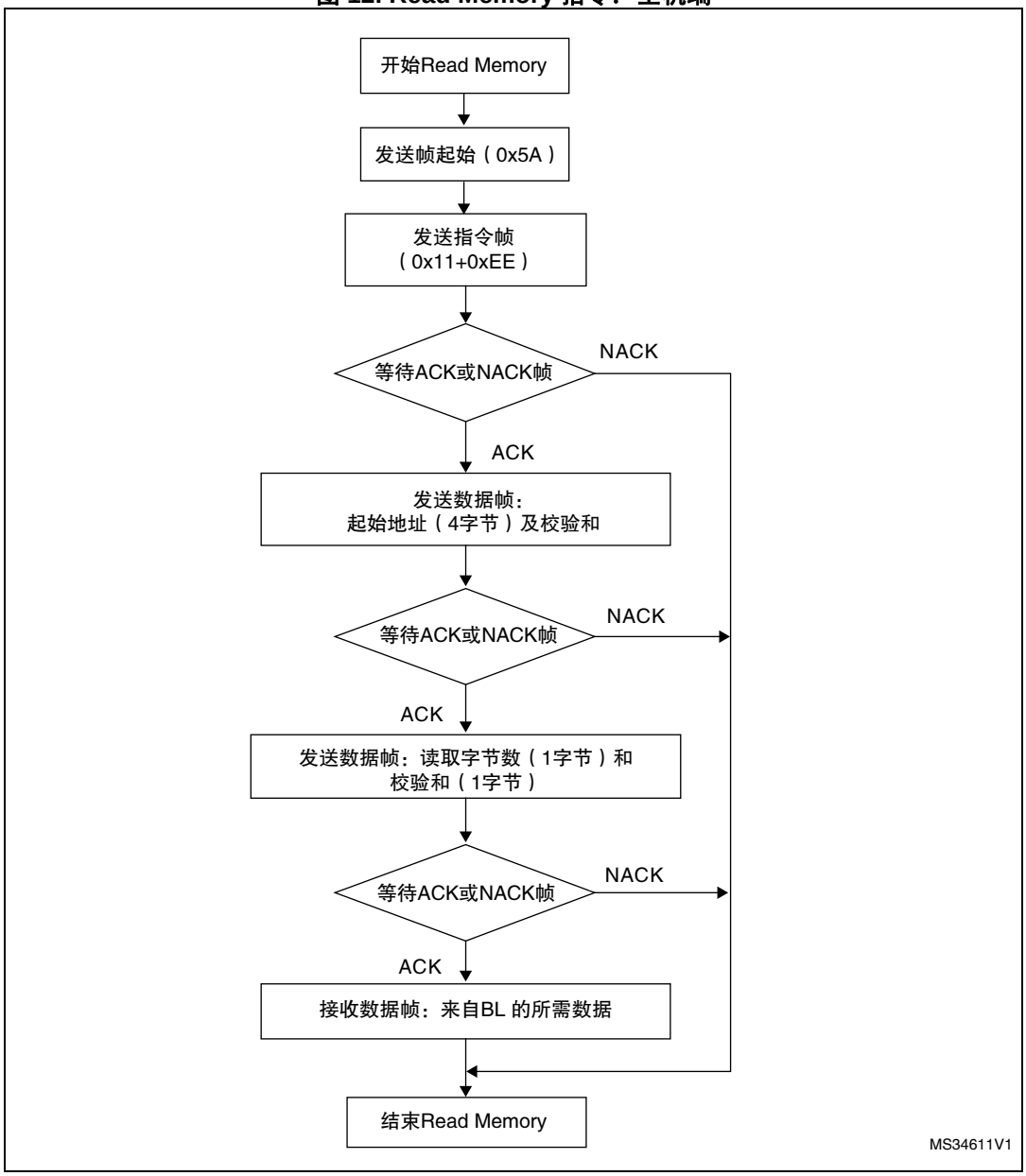
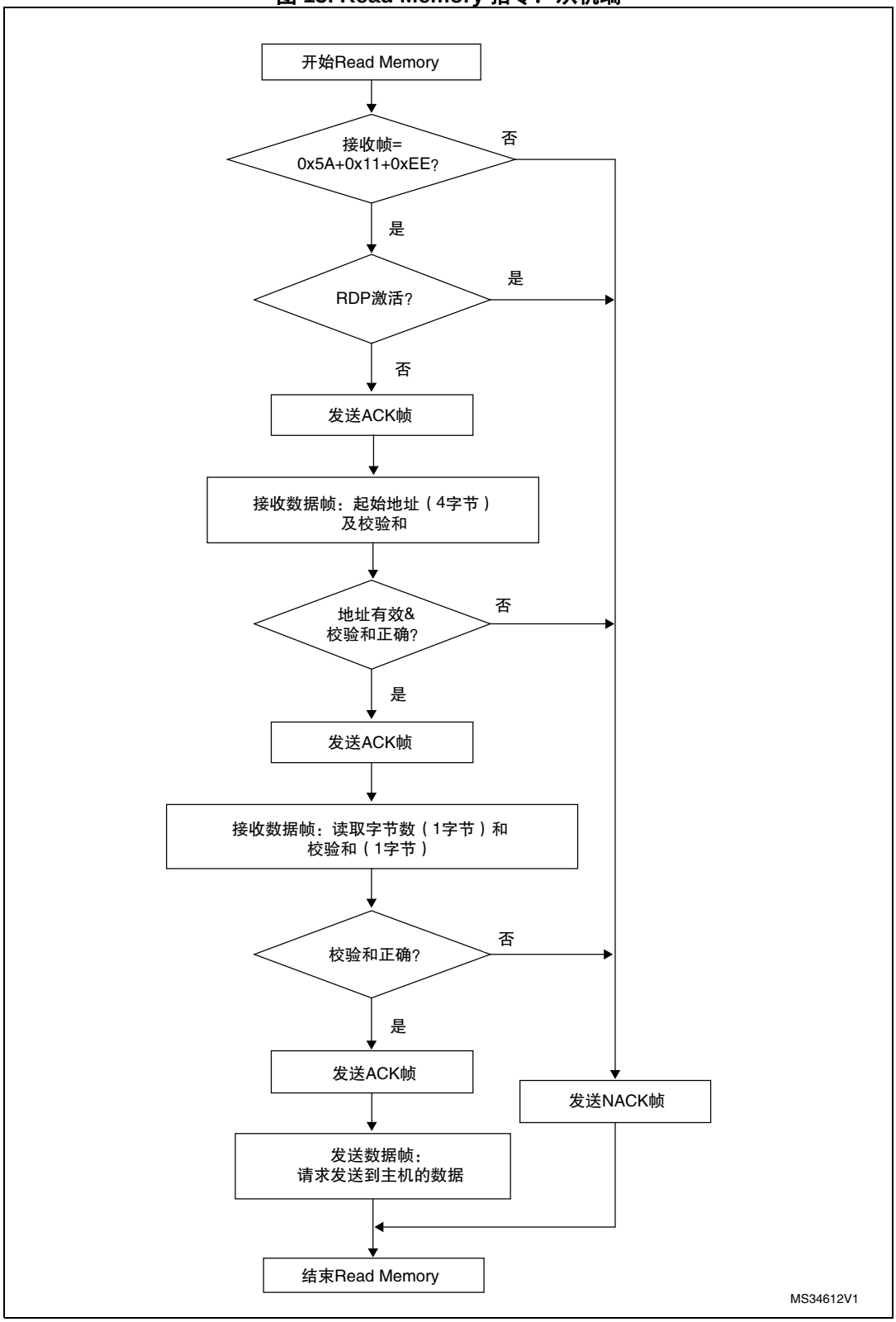


图 13. Read Memory 指令：从机端



2.6 *Go* 指令

Go 指令用于执行下载的代码或由应用指定跳转地址的任何其它代码。当自举程序收到 Go 指令时，它会向应用发送 ACK 字节。发送 ACK 字节后，自举程序等待一个地址 （4 字节，字节 1 为地址 MSB，字节 4 为 LSB）及校验和字节，之后它会检查收到的地址。若地址有效且校验和正确，则自举程序发送 ACK 字节；否则它发送 NACK 字节并终止该指令。

若地址有效且校验和正确，则自举程序固件会执行如下操作。

- 将自举程序所用外设的寄存器初始化至其默认复位值。
- 初始化用户应用的主堆栈指针。
- 跳转至收到的 ' 地址 + 4' 所编程的存储器位置（对应于应用复位向量的地址）。例如，若收到的地址为 0x08000000，则自举程序跳转至编程为 0x08000004 地址的存储器位置。

总之，主机发送基址，应用编程跳转。

注： *仅当用户应用正确设置了指向应用地址的向量表时，跳转到应用才能工作。*

主机向 STM32 发送的字节如下。

- 帧起始：0x5A
- 字节 1：0x21
- 字节 2：0xDE
- 等待 ACK （如第 1 章中所述）
- 字节 3 至字节 6：起始地址
  - 字节 3：MSB
  - 字节 6：LSB
- 字节 7：校验和：XOR （字节 3、字节 4、字节 5、字节 6）

图 14. Go 指令：主机端

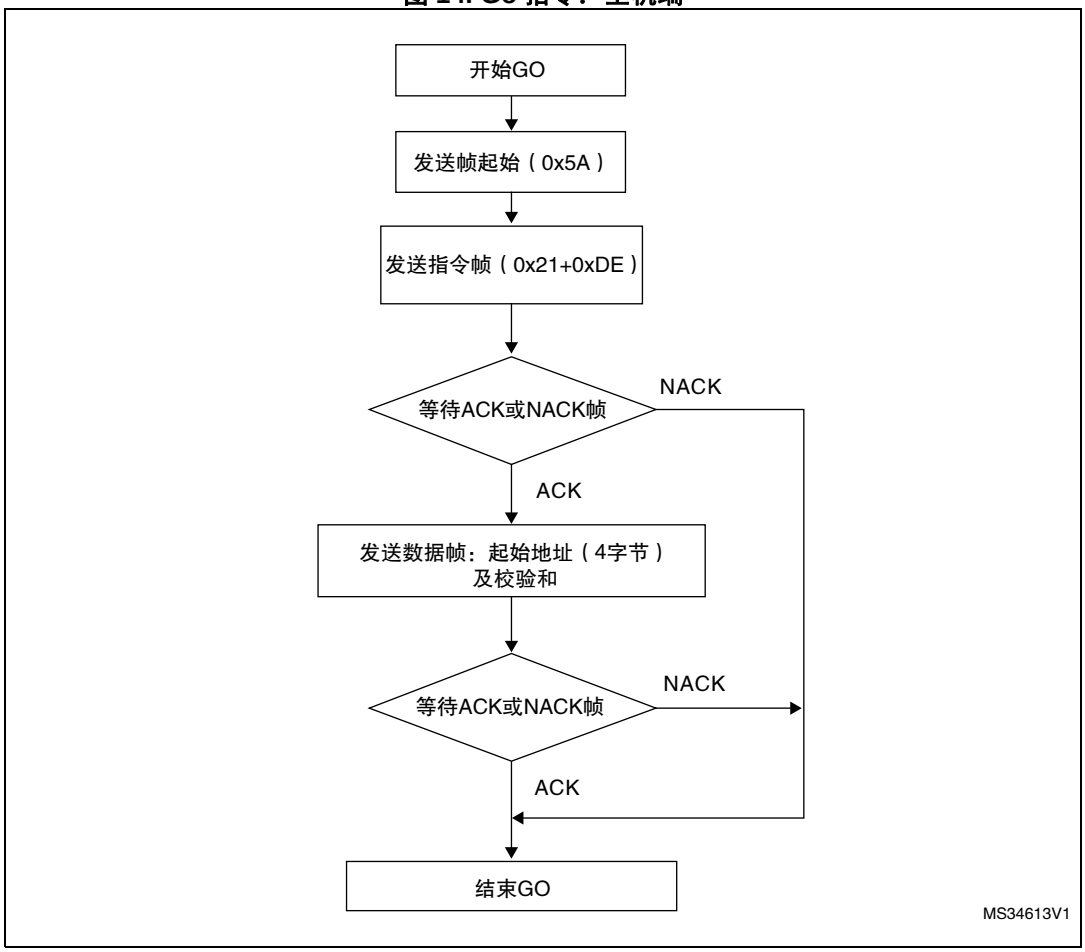
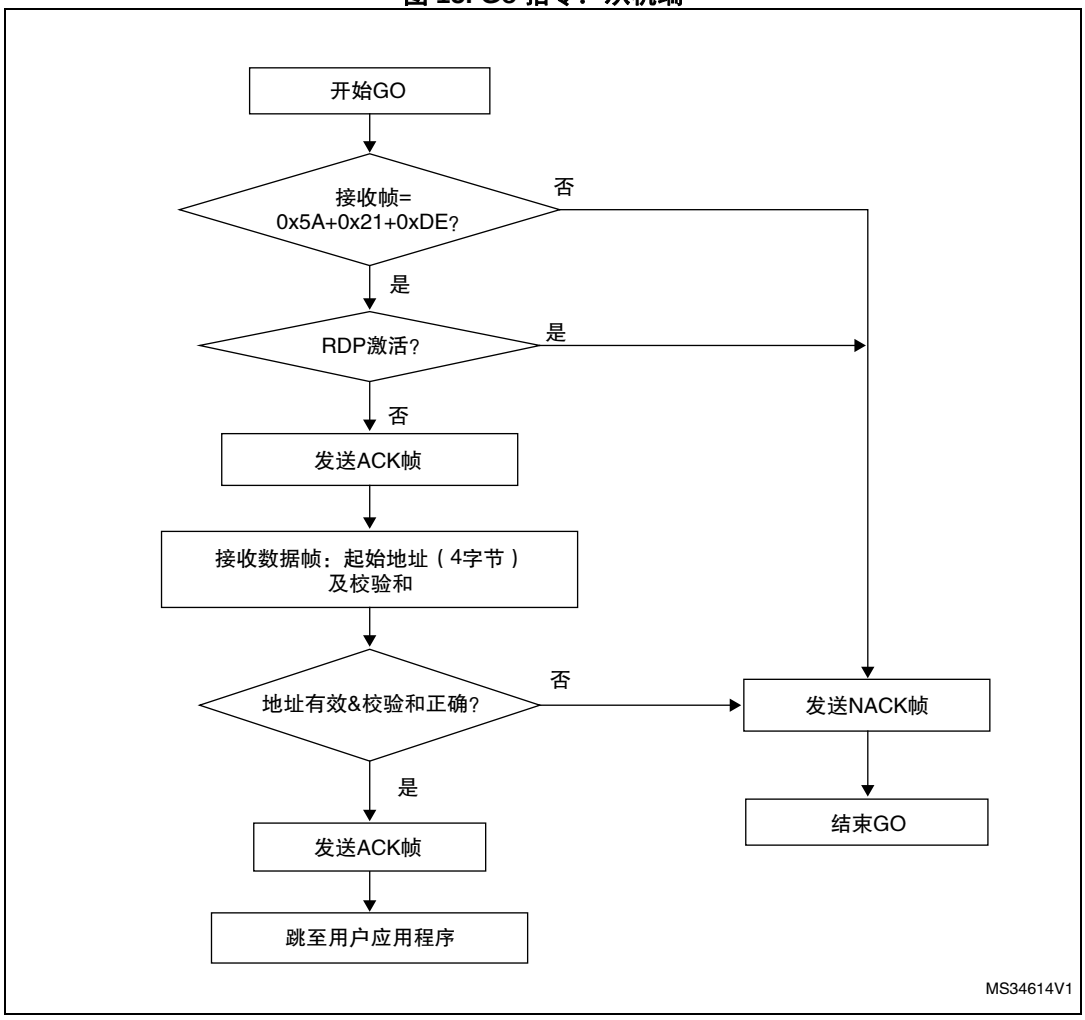

开始GO
发送帧起始（0x5A）
发送指令帧（0x21+0xDE）
等待ACK或NACK帧
NACK
ACK
发送数据帧：起始地址（4字节）
及校验和
等待ACK或NACK帧
NACK
ACK
结束GO
MS34613V1

图 15. Go 指令：从机端


开始GO
接收帧=
0x5A+0x21+0xDE?
否
是
RDP激活?
是
否
发送ACK帧
接收数据帧：起始地址（4字节）
及校验和
地址有效&校验和正确?
否
发送NACK帧
是
结束GO
发送ACK帧
跳至用户应用程序
MS34614V1

2.7 *Write Memory* 指令

Write Memory 指令用于向 RAM、 Flash、选项字节区域的任何有效存储器地址 （见下面的注释）写入数据。

当自举程序收到 Write Memory 指令时，它会向应用发送 ACK 字节。发送 ACK 字节后，自举程序等待一个地址 （4 字节，字节 1 为地址 MSB，字节 4 为 LSB）及校验和字节，之后它会检查收到的地址。

若收到的地址有效且校验和正确，则自举程序发送 ACK 字节；否则它发送 NACK 字节并终止该指令。若地址有效且校验和正确，则自举程序会执行如下操作。

- 得到一个字节 N，它包含要接收的数据字节数。
- 接收 （(N + 1) 字节）用户数据及其校验和 （N 和所有数据字节的异或）。
- 从收到的地址开始将用户数据编程至存储器。

在该指令末尾，若写入操作成功，则自举程序向应用发送 ACK 字节；否则它发送 NACK 字节并终止指令。

若 Write Memory 指令用于选项字节区域，则在写入新值之前会擦除所有选项。在指令末尾，自举程序会生成系统复位，以使选项字节的新配置生效。要写入选项字节区的块起始地址和最大长度必须考虑产品的选项字节地址和大小。

若 write memory 的目标为 Flash，则在轮询从机响应之前，主机必须等待足够长时间以写入发送缓冲 （请参考产品数据手册以获取时间值）。

注： *写入 RAM 或 Flash 的最大块长度为 256 字节。*

向 Flash 的写操作必须为字 （16 位）对齐，数据应为两字节的整数倍。若要写入的数据不足，应用 0xFF 填充剩余字节。

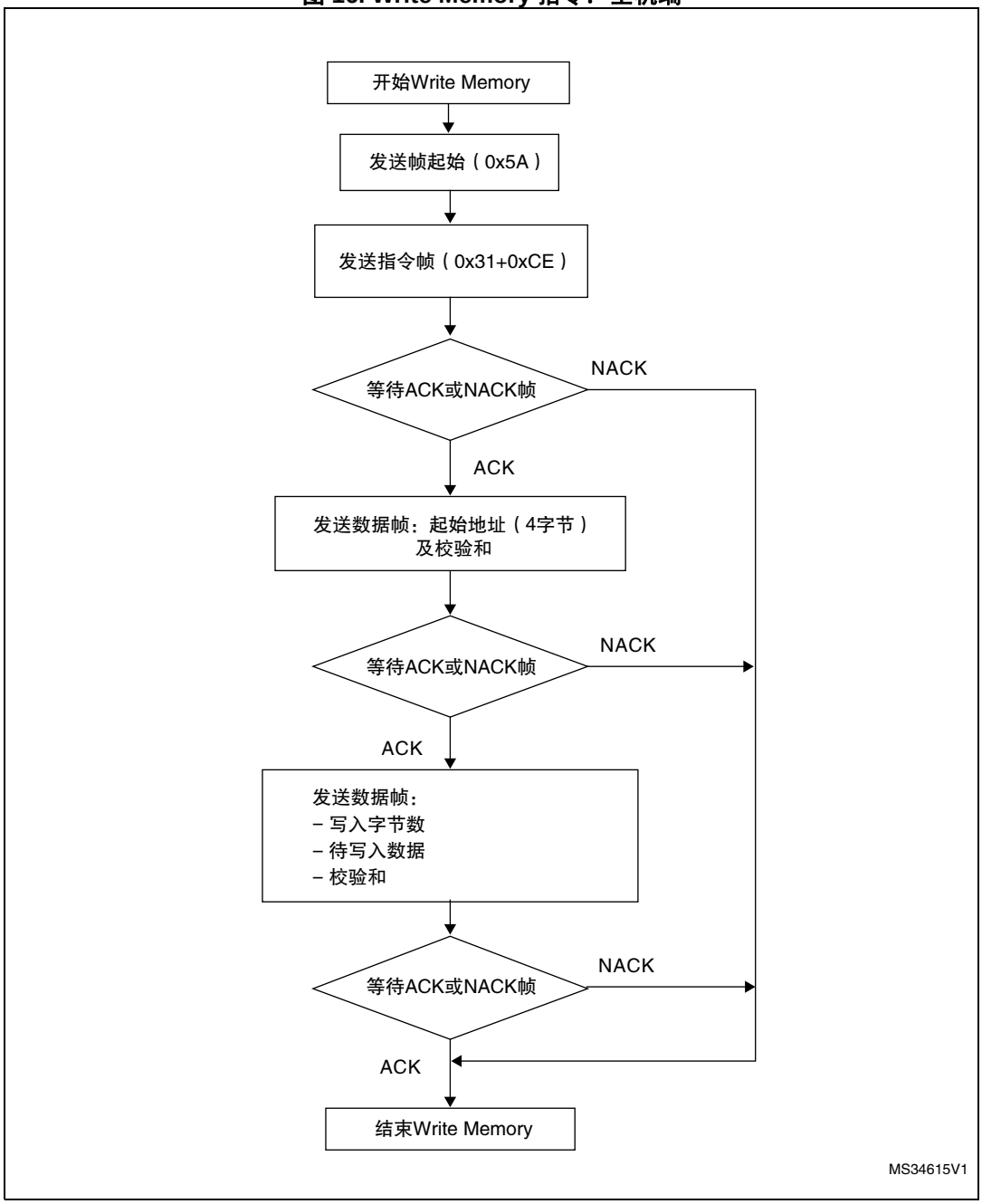
当写入 RAM 时，您应注意不要与自举程序固件使用的第一个 RAM 存储器重叠。

当向写保护的扇区执行写操作时，不会返回错误。

主机向 STM32 发送的字节如下。

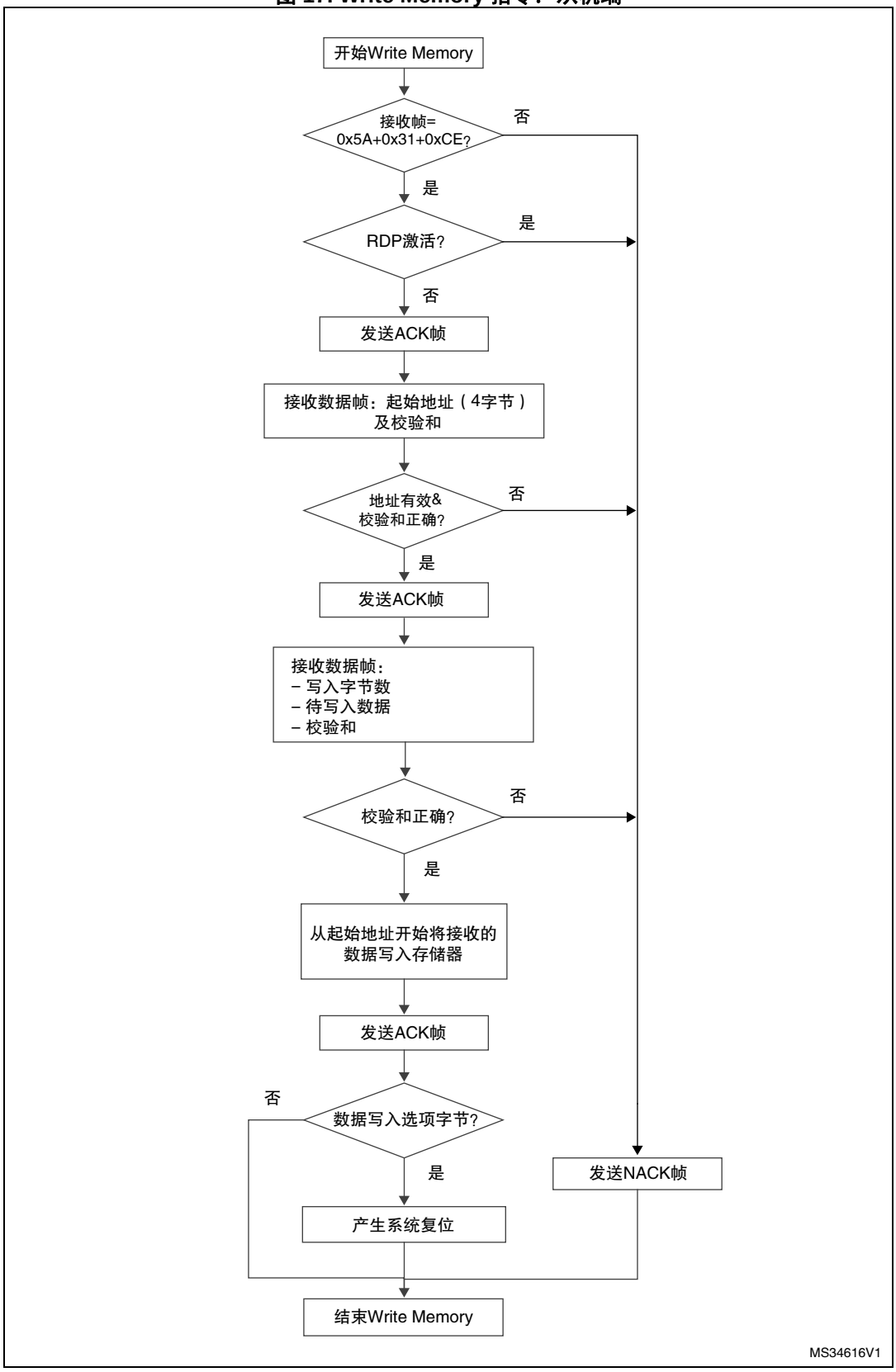
- 帧起始：0x5A
- 字节 1：0x31
- 字节 2：0xCE
- 等待 ACK （如*第 1 节*中所述）：
- 字节 3 至字节 6：起始地址
  - 字节 3：MSB
  - 字节 6：LSB
- 字节 7：校验和：XOR （Byte3、 Byte4、 Byte5、 Byte6）
- 等待 ACK （如*第 1 节*中所述）
- 字节 8：要接收的字节数 （0 < N ≤ 255）
- N +1 数据字节：（最大 256 字节）
- 校验和字节：XOR （N， N+1 个数据字节）

图 16. Write Memory 指令：主机端



注： *在一些工作条件下，在收到应答之后、发送数据帧（要写入的字节个数、要写入的数据、校验和）之前，主机必须等待 1 ms。*

图 17. Write Memory 指令：从机端



2.8 *Erase Memory* 指令

Erase Memory 指令可使主机用双字节寻址模式擦除 Flash 页面或扇区。当自举程序收到 Erase Memory 指令时，它会向主机发送 ACK 字节。发送 ACK 字节后，自举程序接收两个字节 （要擦除的页面或扇区数）、 Flash 页面或扇区码 （每个都以双字节编码， MSB 在前）、校验和字节 （所发送字节的 XOR）。若校验和正确，则自举程序擦除存储器，向主机发送 ACK 字节。否则，它向主机发送 NACK 字节，命令终止。

Erase Memory 指令规范

自举程序收到两个字节，它包含 N—— 要擦除的页面或扇区数。

- 对于 N = 0xFFFY （其中 Y 为 0 到 F），会执行特殊擦除。
  - 0xFFFF 为全局批量擦除。
  - 0xFFFE 为 Bank 1 全局擦除 （仅适用于支持此特性的产品）。
  - 0xFFFD 为 Bank 2 全局擦除 （仅适用于支持此特性的产品）。
  - 值 0xFFFC 到 0xFFF0 为预留。
- 对于 0 ≤ N < 页面或扇区最大数的其它值，会擦除 N + 1 个页面或扇区。

然后，自举程序接收如下。

- 对于特殊擦除的情况，一个字节：之前字节的校验和：（即，对于 0xFFFF 为 0x00）。
- 若擦除 N+1 个页面或扇区，则自举程序接收 （2 x (N + 1)）个字节，其中的每半个字都包含了一个双字节编码的页面号， MSB 在前。这时所有前面字节的校验和都在一个字节接收。

注： *当向写保护的扇区执行擦除操作时，不会返回错误。页面或扇区的最大数目与产品有关，因此应引起注意。*

主机向 STM32 发送的字节如下。

- 帧起始：0x5A
- 字节 1：0x44
- 字节 2：0xBB
- 等待 ACK （如第 1 章中所述）
- 字节 3-4：
  - 特殊擦除：全局擦除 （0xFFFY 其中 Y={F、 E、 D}）

或

  - 要擦除的页面或扇区数 （N+1，其中：0 ≤ N < 页面或扇区的最大数目）。
- 其余字节：
  - 对于特殊擦除 （0x00、 0x01、 0x02），为字节 3-4 的校验和。

或

  - （2 x (N + 1)）字节 （双字节编码的页面号， MSB 在前），然后是字节 3-4 与其后所有字节的校验和）。

图 18. Erase Memory 指令：主机端

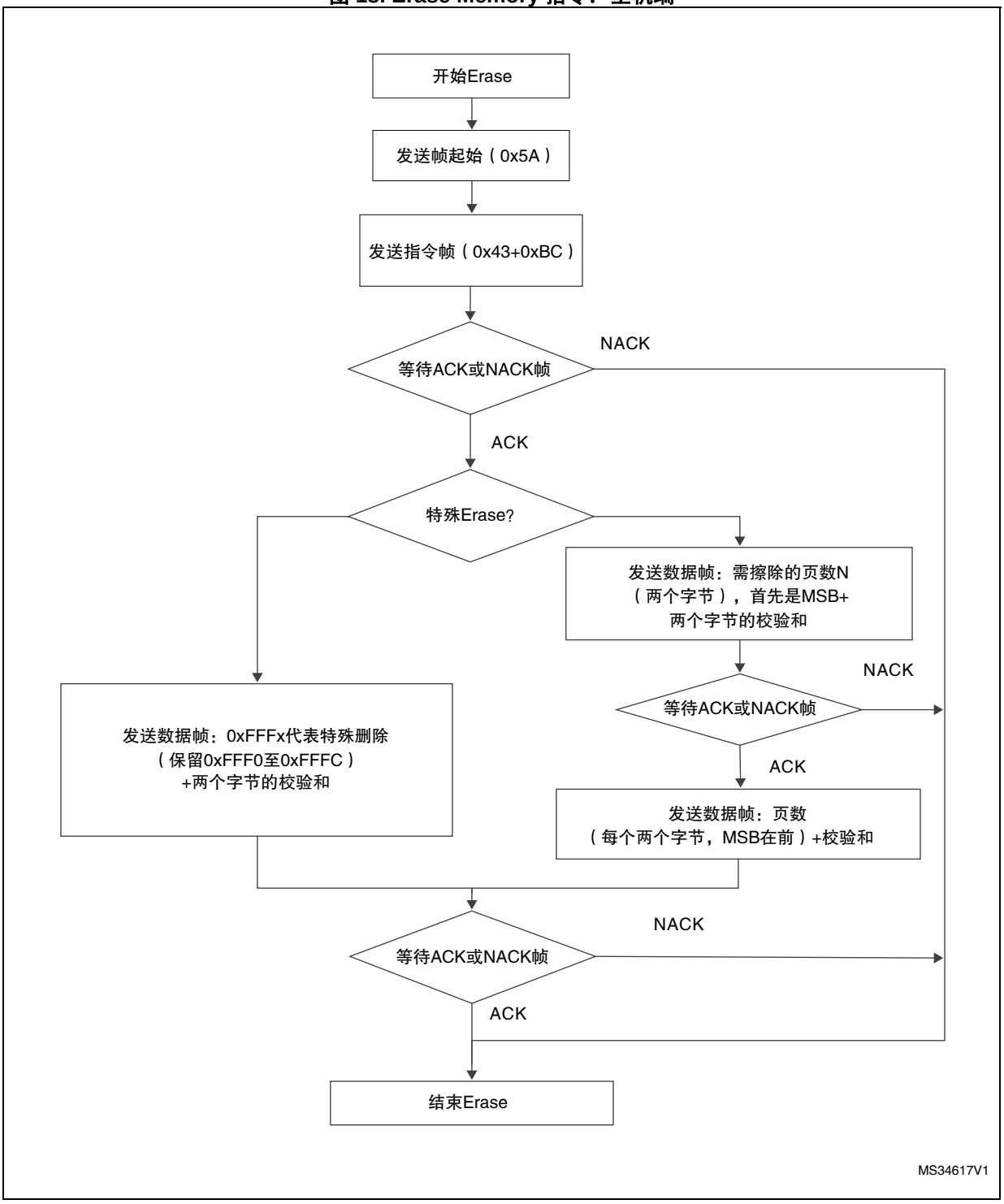
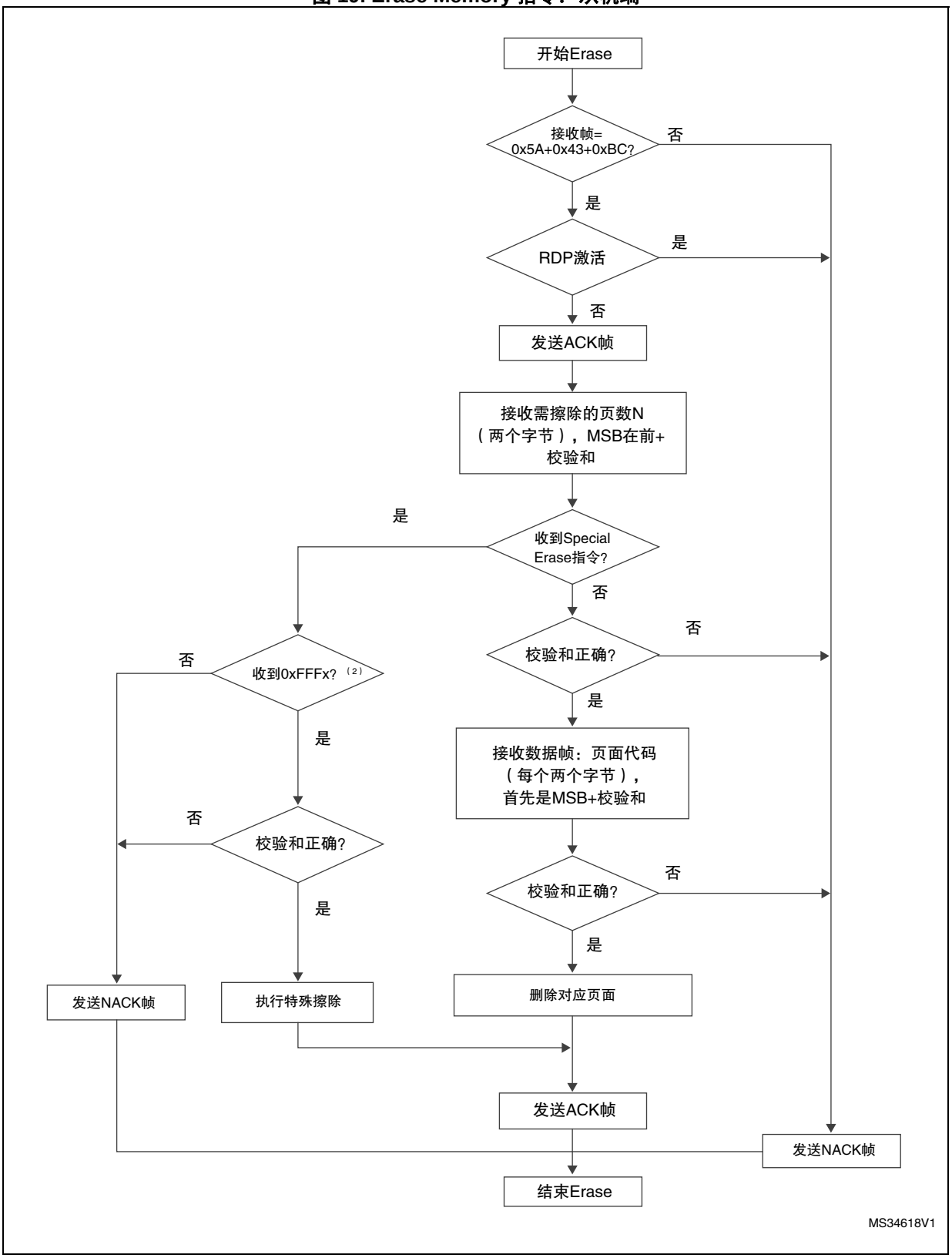


图 19. Erase Memory 指令：从机端



2.9 *Write Protect* 指令

Write Protect 指令用于对部分或全部 Flash 扇区启用写保护。当自举程序收到 Write Protect 指令时，它会向主机发送 ACK 字节。发送 ACK 字节之后，自举程序等待要接收的字节数（要保护的扇区），然后从应用收到 Flash 扇区码。

在 Write Protect 指令末尾，自举程序会发送 ACK 字节并生成系统复位，以使选项字节的新配置生效。

Write Protect 指令流程如下。

- 自举程序收到一个字节，它包含 N—— 要写保护的扇区数 - 1 （0 ⩽ N ⩽ 255）
- 自举程序收到 （N + 1）字节，其中每个字节都包含扇区码。

注： *要保护的扇区总数和扇区号不经过校验，这意味着若指令中要保护的扇区总数和扇区号错误，也不会返回错误。*

若执行了第二个 Write Protect 指令，则第一个指令已经保护的 Flash 扇区会被解除保护，只有第二个 Write Protect 指令内的扇区才会被保护。

图 20. Write Protect 指令：主机端

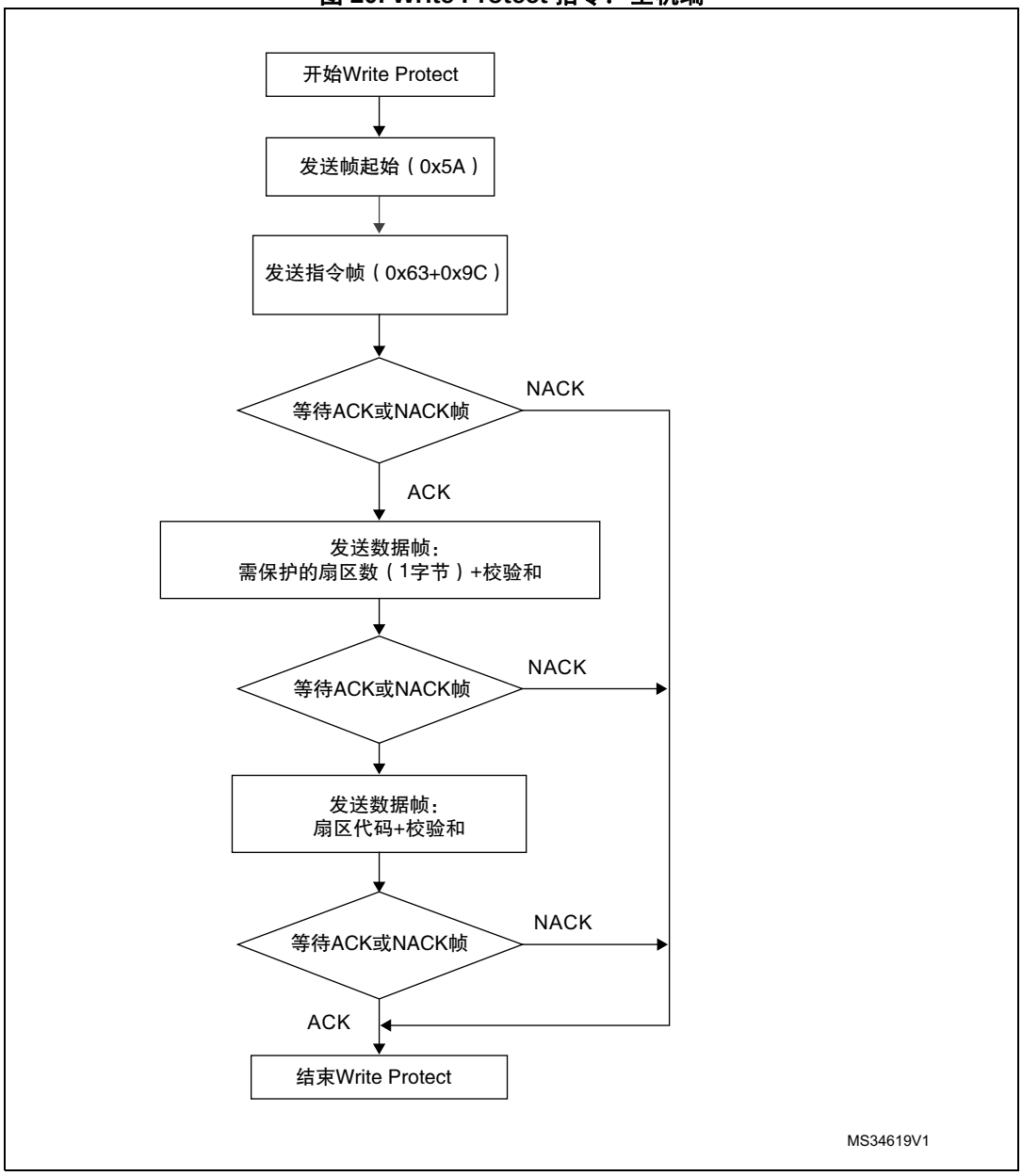
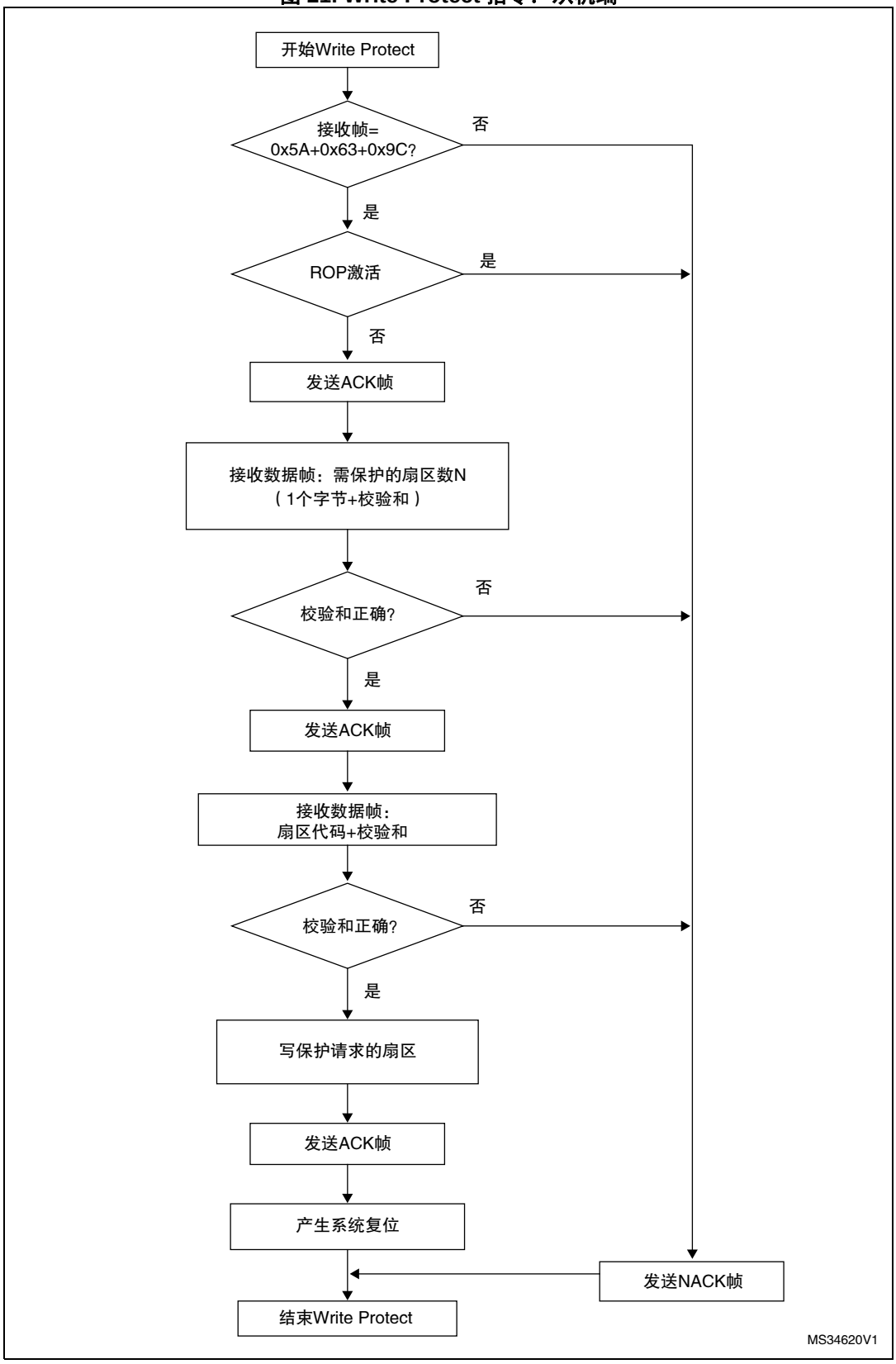


图 21. Write Protect 指令：从机端



2.10 *Write Unprotect* 指令

Write Unprotect 指令用于对全部 Flash 扇区禁用写保护。当自举程序收到 Write Unprotect 指令时，它会向主机发送 ACK 字节。发送完 ACK 字节之后，自举程序禁用所有 Flash 扇区的写保护。在禁用保护的操作之后，自举程序发送 ACK 字节。

在 Write Unprotect 指令末尾，自举程序会发送 ACK 字节并生成系统复位，以使选项字节的新配置生效。

图 22. Write Unprotect 指令：主机端

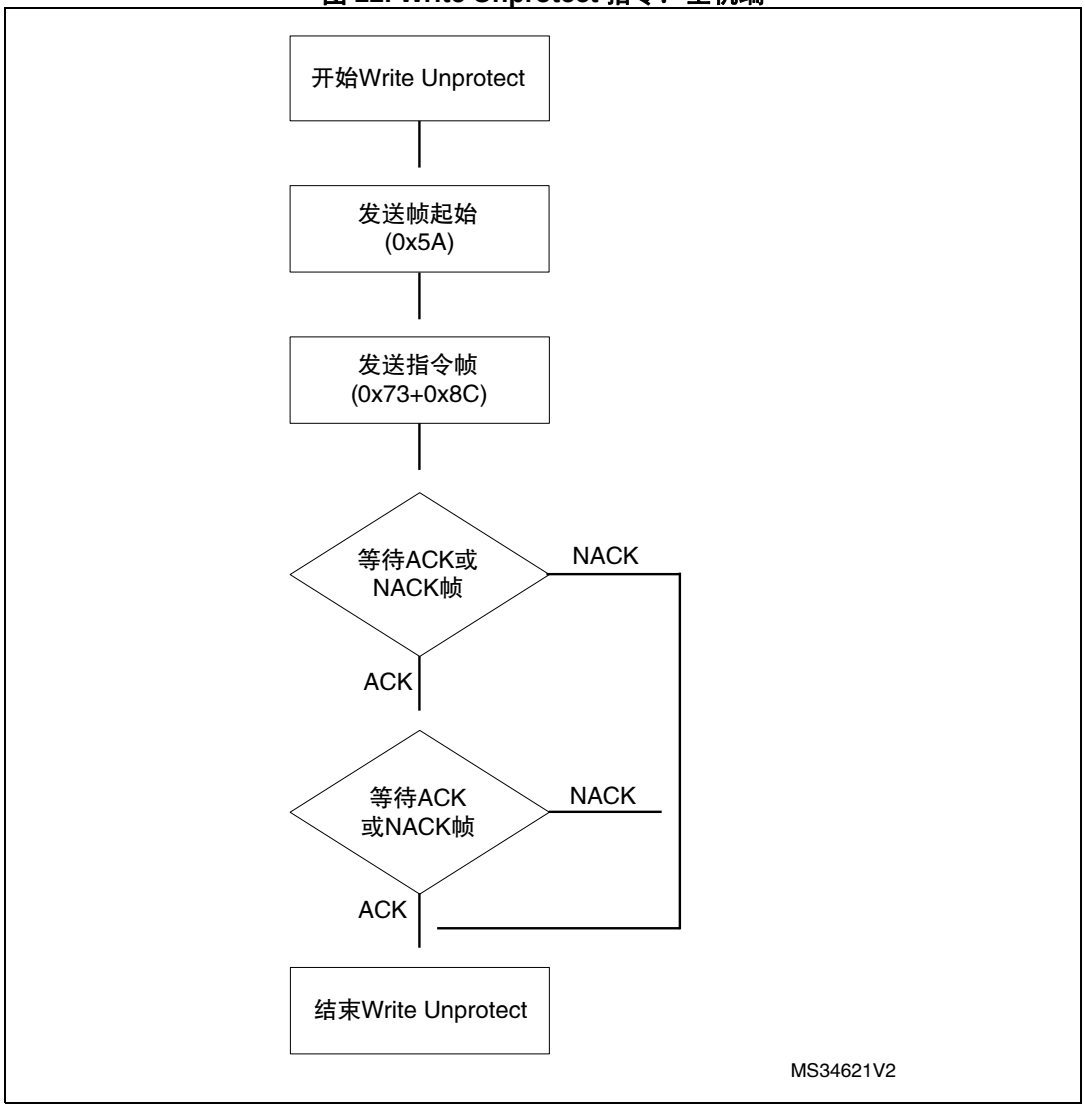
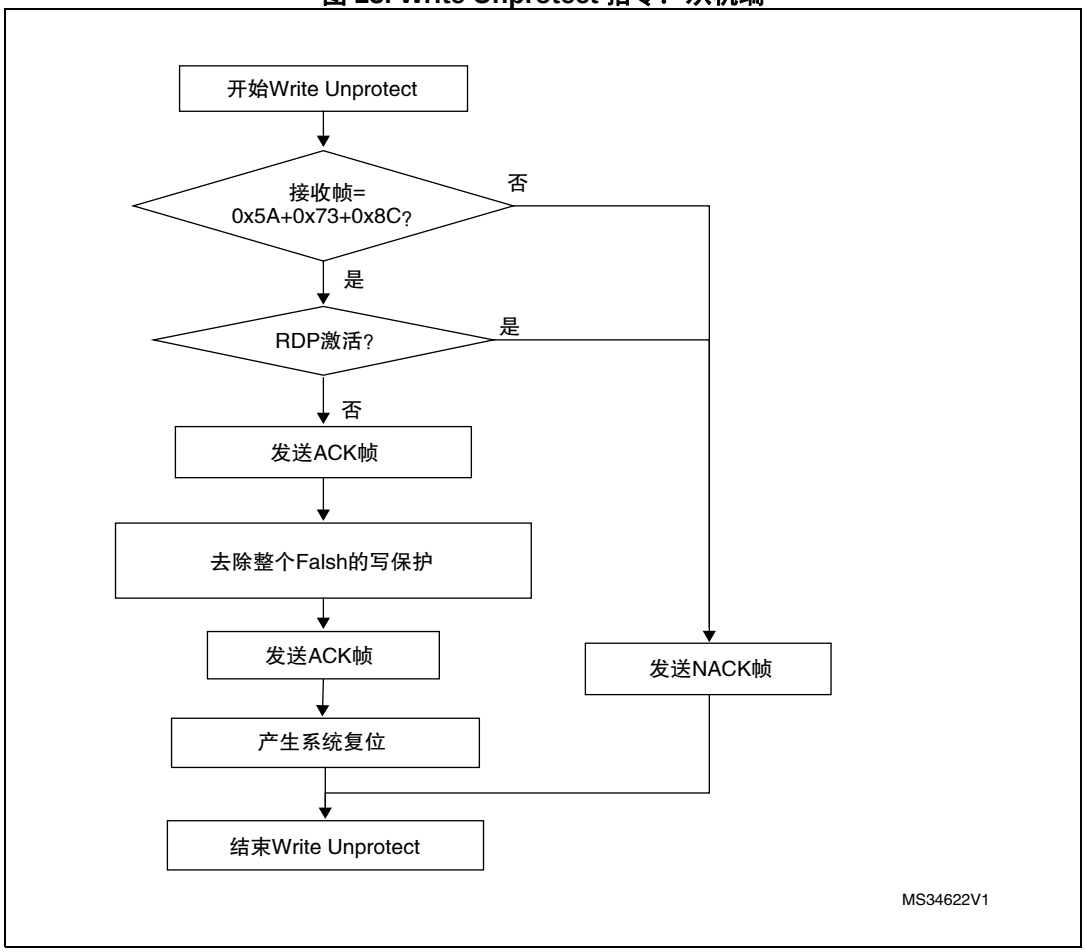


图 23. Write Unprotect 指令：从机端



2.11 *Readout Protect* 指令

Readout Protect 指令用于启用 Flash 的读保护。当自举程序收到 Readout Protect 指令时，它会向主机发送 ACK 字节。发送完 ACK 字节之后，自举程序启用 Flash 的读保护。

在 Readout Protect 指令末尾，自举程序会发送 ACK 字节并生成系统复位，以使选项字节的新配置生效。

图 24. Readout Protect 指令：主机端

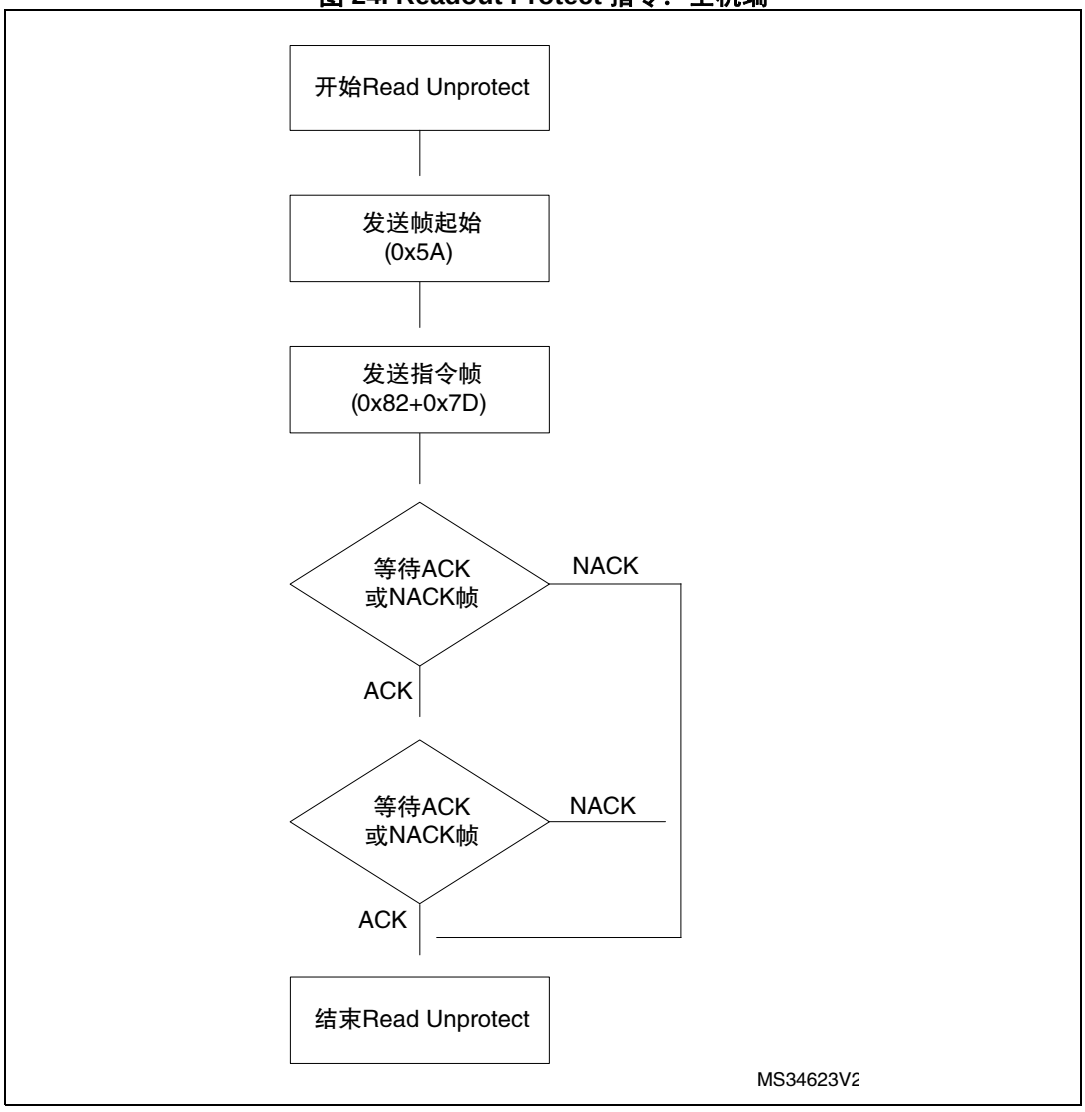
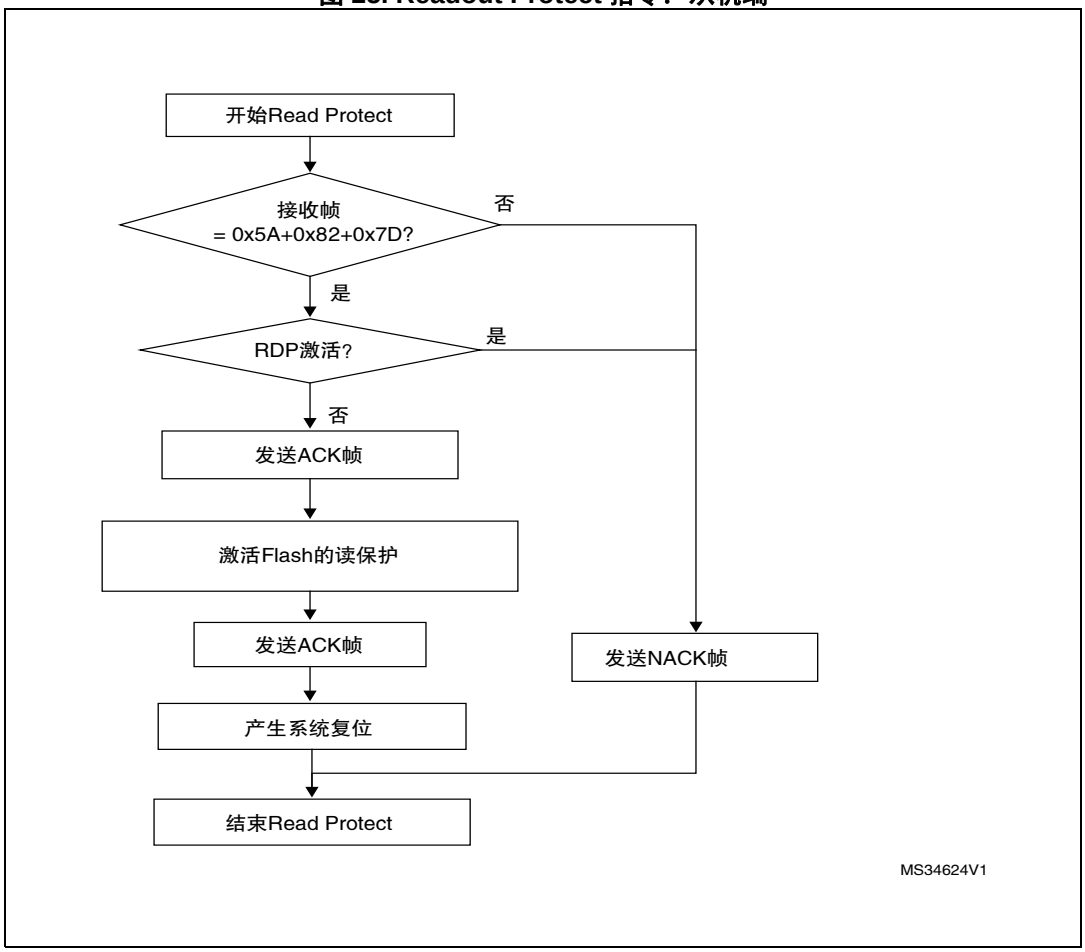


图 25. Readout Protect 指令：从机端



2.12 *Readout Unprotect* 指令

Readout Unprotect 指令用于禁用 Flash 的读保护。当自举程序收到 Readout Unprotect 指令时，它会向主机发送 ACK 字节。发送完 ACK 字节之后，自举程序擦除所有 Flash 扇区，对整个 Flash 禁用读保护。若擦除操作成功，则自举程序取消激活 RDP。

若擦除操作不成功，则自举程序会发送 NACK，读保护仍然有效。

在轮询从机的应答之前，主机必须等待足够时间来禁用读保护 （在大多数产品上，它等于批量擦除时间 —— 请参考产品数据手册以获取更多信息）。

在 Readout Unprotect 指令末尾，自举程序会发送 ACK 并生成系统复位，以使选项字节的新配置生效。

图 26. Readout Unprotect 指令：主机端

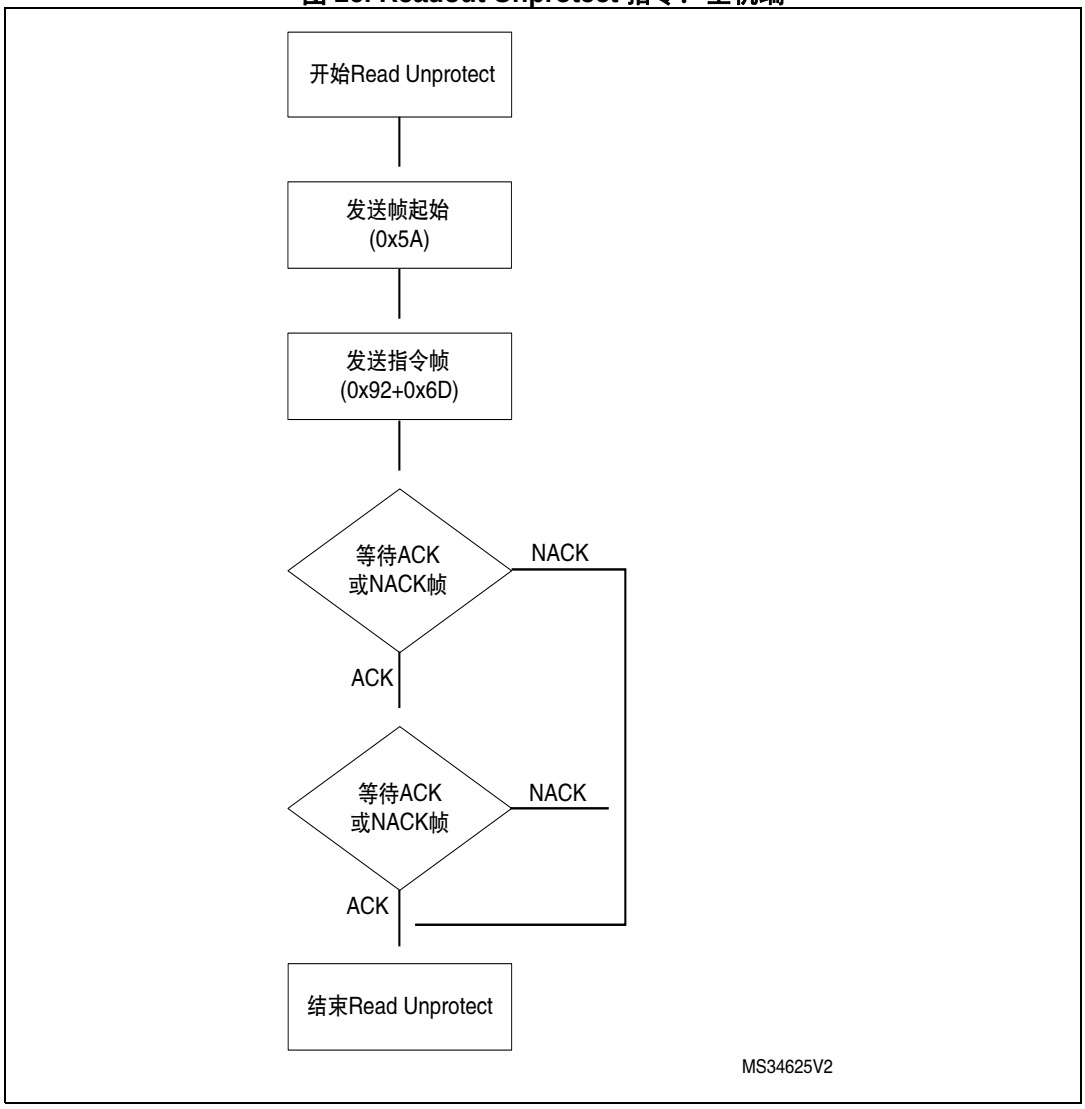
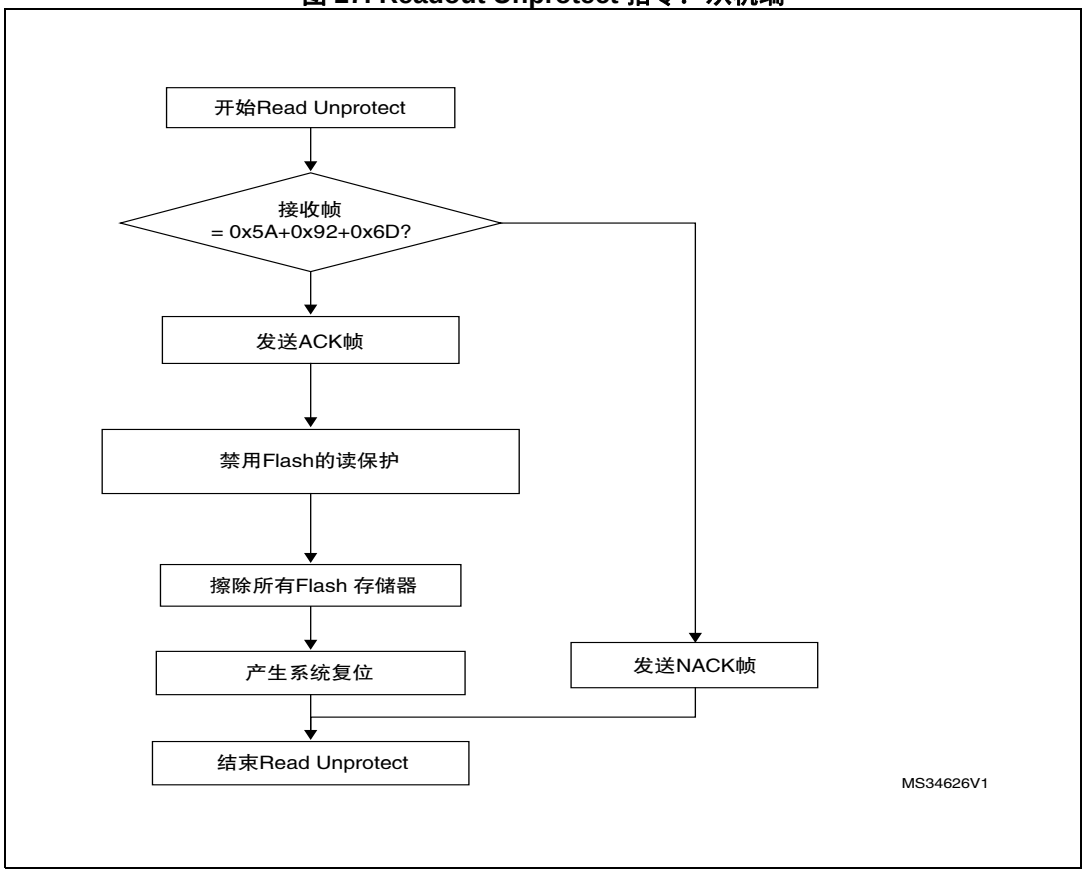


图 27. Readout Unprotect 指令：从机端



3 自举程序协议版本演进

表 3 列出了自举程序的版本。

表 3. 自举程序协议版本

版本	*说明*
V1.1	更新了应答机制。 更新了 Get、 GetID、 GetVersion 和 Read 指令。
V1.0	初始自举程序版本。

4 修订历史

表 4. 文档修订历史

日期	*修订*	*变更*
2014 年 3 月 27 日	1	初始版本。
2014 年 5 月 2 日	2	更新了*表 1: 适用产品*。 在*表 2: SPI 自举程序指令*中增加了脚注。 更新了*第 2 章节：自举程序指令集*。 更新了*图 22*、*图 24* 和*图 26*。

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