AN3154
应用笔记

STM32™ 自举程序中使用的 CAN 协议

前言

本应用笔记将介绍 CAN 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 CAN 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	适用的产品
微控制器	STM32F105xx、STM32F107xx、STM32F20xx、STM32F21xx、STM32F40xx、STM32F41xx

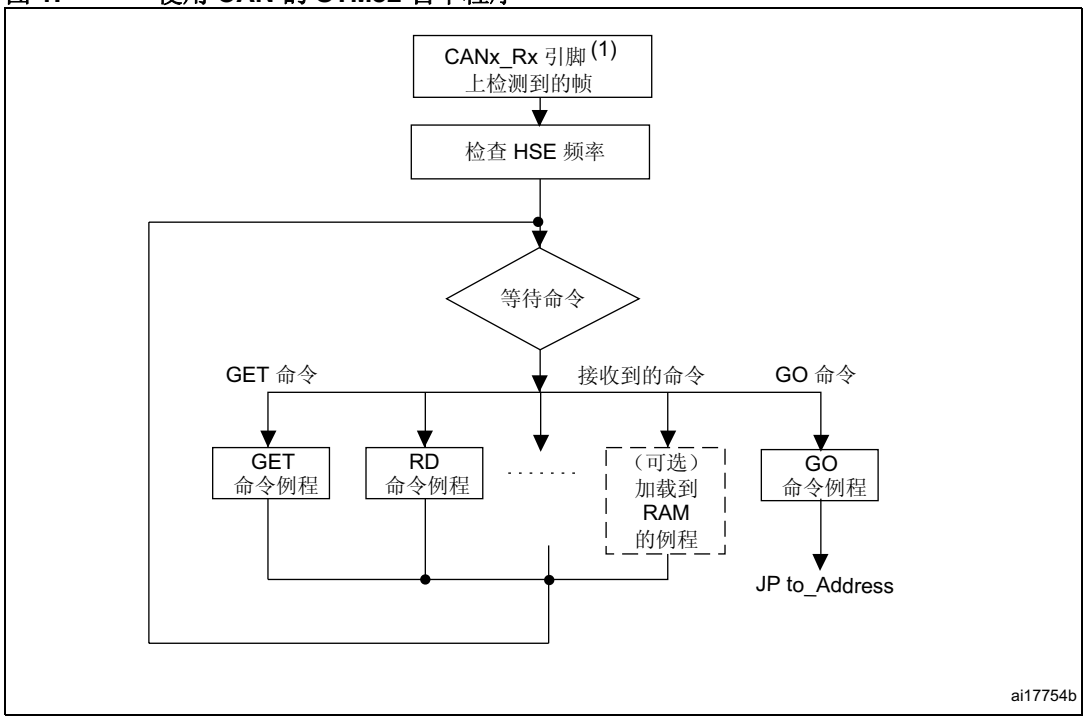
目录

表格索引

图片索引

1 自举程序代码序列

图 1. 使用 CAN 的 STM32 自举程序

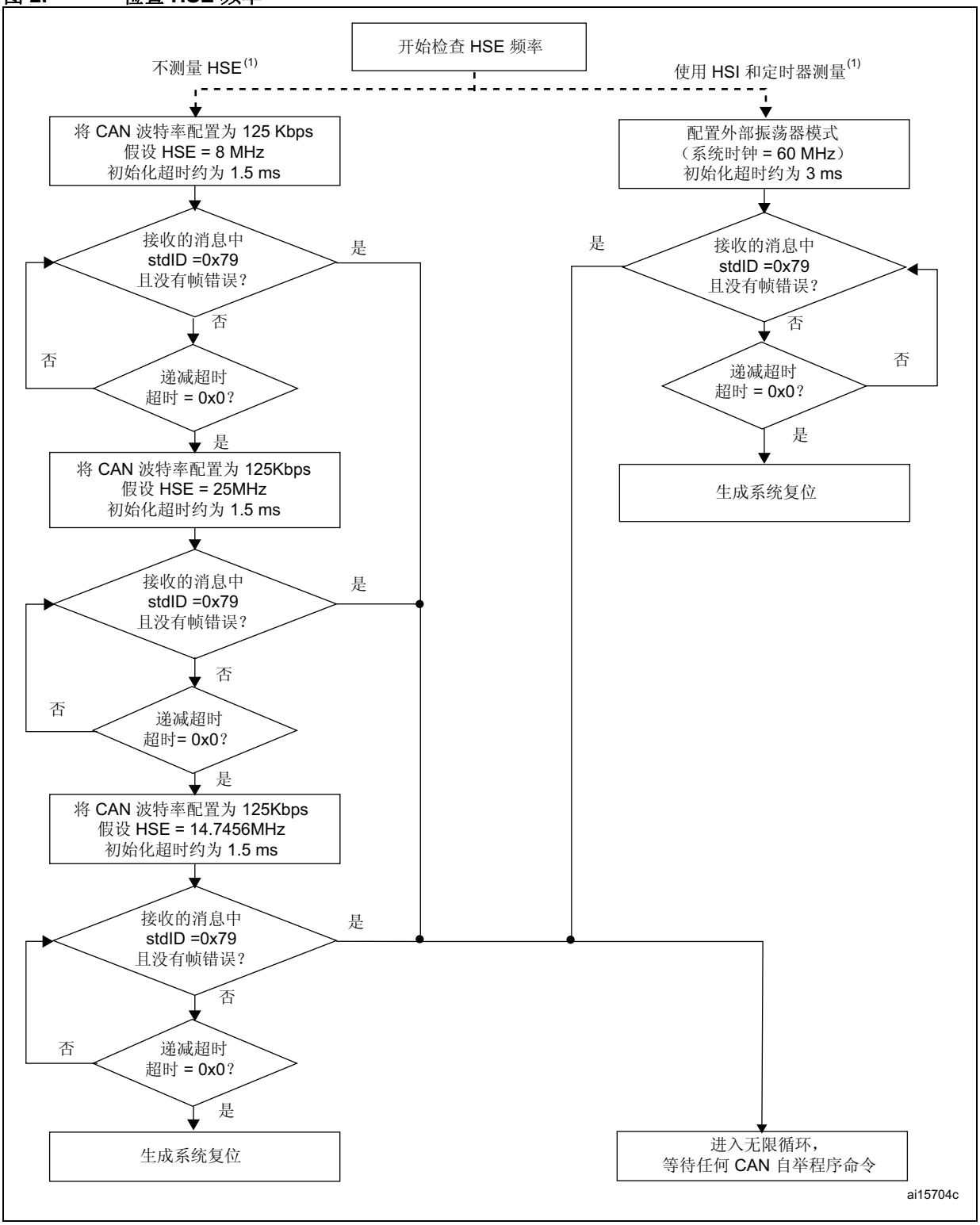


1. 发送帧时推荐使用标准 ID = 0x79。

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”），自举程序代码将等待 CANx_Rx 引脚上的帧。进行检测时，CAN 自举程序固件将检查外部时钟频率。

图 2 显示频率检查的流程图。

图 2. 检查 HSE 频率



1. 有些器件使用连接到定时器的 HSI 振荡器计算 HSE 频率，其它器件则不支持这种测量。不测量 HSE 频率的器件仅执行左侧显示的流程，而测量 HSE 频率的器件仅执行右侧的流程。要确定使用的器件涉及哪种工作流，请参见 AN2606。

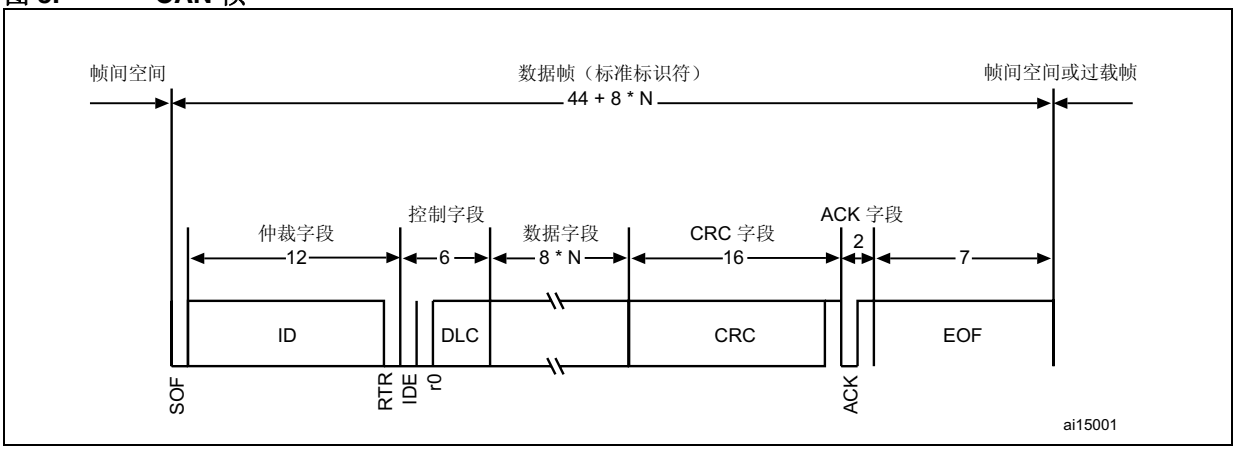
随后，代码将相应初始化串行接口。通过计算出的波特率，发送确认字节 *(0x79)* 返回主机，表示 STM32 已准备好接收命令。

2 CAN 设置

STM32 CAN 符合 2.0A 和 2.0B（主动）规范，比特率最高达 1 Mb/s。 它可接收和发送包含 11 位标识符的标准帧和包含 29 位标识符的扩展帧。

图 3 显示仅使用标准标识符的 CAN 帧。

图 3. CAN 帧



在此应用中，CAN 设置为：

- 标准标识符（非扩展）
- 比特率：开始时为 125 kbps；运行期间可通过速度命令进行更改，最高可达 1 Mbps。

发送设置（从 STM32 到主机）为：

- Tx mailbox0：开启
- Tx mailbox1 和 Tx mailbox2：关闭
- Tx 标识符：（0x00、0x01、0x02、v03、0x11、0x21、0x31、0x43、0x63、0x73、0x82、0x92）

接收设置（从主机到 STM32）为：

- 同步字节 0x79 位于 RX 标识符中，而不是数据字段中。
- RX 标识符取决于命令（0x00、0x01、0x02、0x03、0x11、0x21、0x31、0x43、0x63、0x73、0x82、0x92）。
- 错误检查：如果错误字段（CAN_ESR 寄存器中的位 [6:4]）不为 000b，则丢弃消息并向主机发送 NACK。
- 符合 FIFO 上溢条件时，丢弃消息并向主机发送 NACK。
- 传入消息可包含 1 到 8 个数据字节。

注： CAN 自举程序固件每次仅支持一个节点。也就是说固件不支持 CAN 网络管理。

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. CAN 自举程序命令

命令	命令代码	命令说明
Get(1)	0x00	获取当前自举程序版本及允许使用的命令
Get Version & Read Protection Status(1)	0x01	获取自举程序版本及 Flash 的读保护状态
Get ID(1)	0x02	获取芯片 ID
Speed	0x03	使用 Speed 命令可以更改 CAN 运行期间的波特率。
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	0x43	擦除至少一个到全部 Flash 扇区
Write Protect(2)	0x63	使能某些扇区的写保护
Write Unprotect(2)	0x73	禁止所有 Flash 扇区的写保护
Readout Protect(1)	0x82	使能读保护
Readout Unprotect(1)	0x92	禁止读保护

1. 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。
2. 请参见下面的*第 3.1 节*。

通信安全

每个数据包或者被接受（ACK 应答）或者被丢弃（NACK 应答）：

- ACK 消息 = 0x79
- NACK 消息 = 0x1F

3.1 器件相关的自举程序参数

虽然所有 STM32 器件的 CAN 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。对于一些命令，某些参数值可能取决于所使用的器件。相关参数如下：

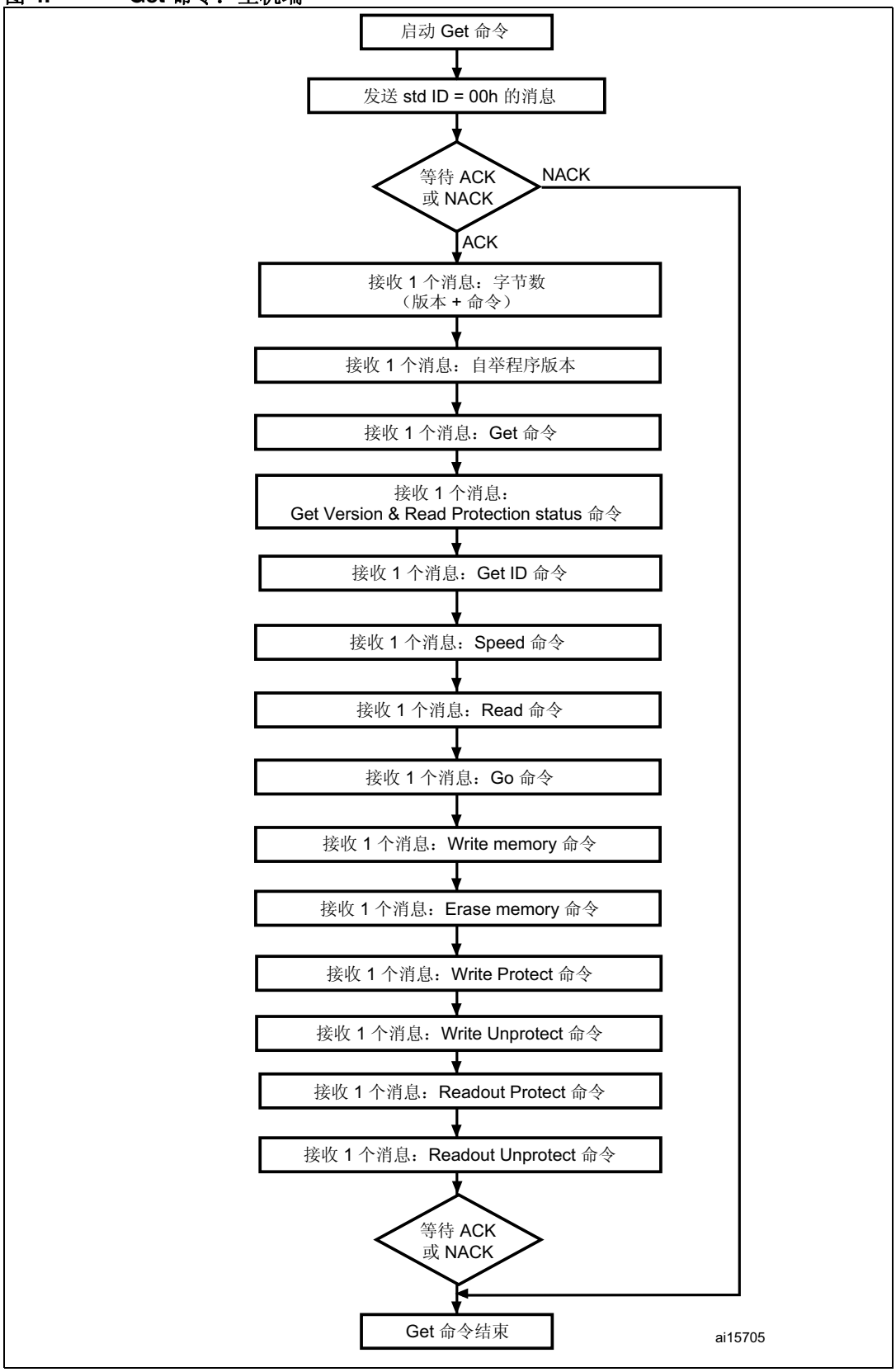
- PID（产品 ID），该参数因器件而异
- 执行 Read Memory、Go 和 Write Memory 命令时，自举程序允许的有效存储器地址（RAM、Flash、系统存储器、选项字节区域）
- 执行 Write Protect 命令时使用的 Flash 扇区的大小

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

主机通过 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的命令代码发送给主机。

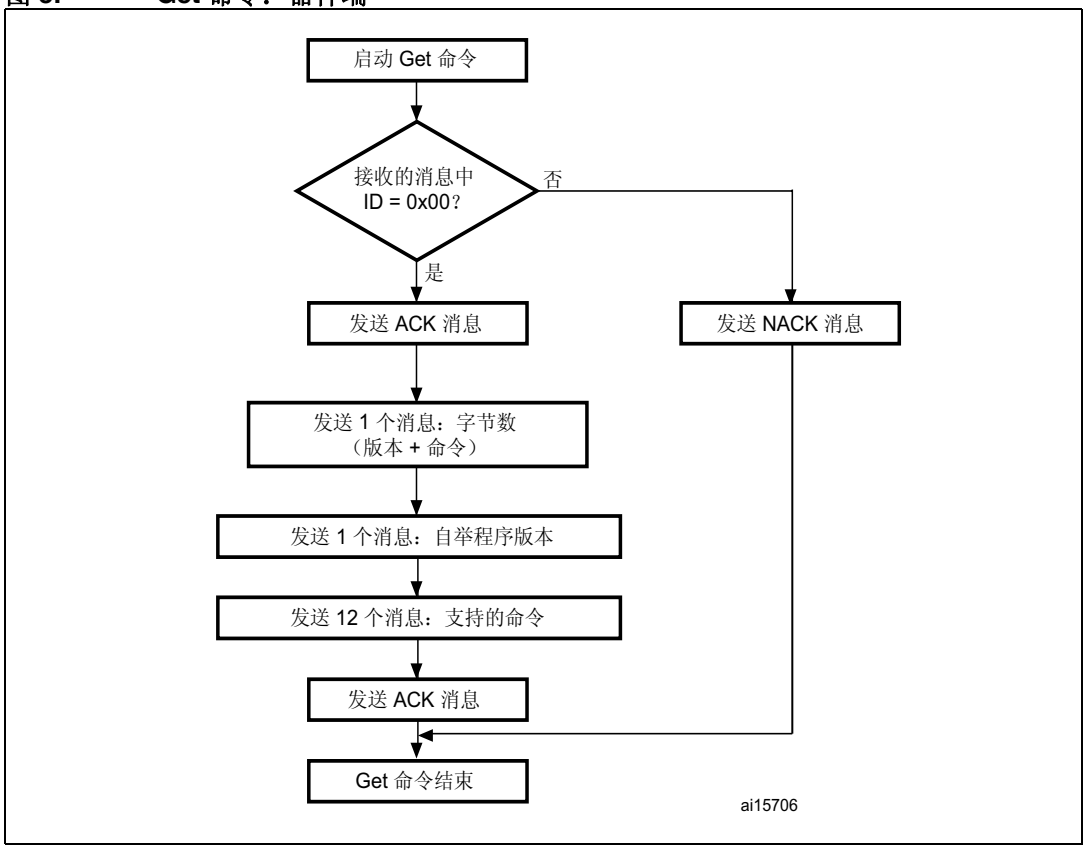
图 4. Get 命令：主机端


启动 Get 命令
发送 std ID = 00h 的消息
等待 ACK
或 NACK
NACK
ACK
接收 1 个消息：字节数
（版本 + 命令）
接收 1 个消息：自举程序版本
接收 1 个消息：Get 命令
接收 1 个消息：
Get Version & Read Protection status 命令
接收 1 个消息：Get ID 命令
接收 1 个消息：Speed 命令
接收 1 个消息：Read 命令
接收 1 个消息：Go 命令
接收 1 个消息：Write memory 命令
接收 1 个消息：Erase memory 命令
接收 1 个消息：Write Protect 命令
接收 1 个消息：Write Unprotect 命令
接收 1 个消息：Readout Protect 命令
接收 1 个消息：Readout Unprotect 命令
等待 ACK
或 NACK
Get 命令结束
ai15705

主机发送如下消息：

命令消息：Std ID = 0x00，数据长度代码 (DLC) =“不重要”。

图 5. Get 命令：器件端



STM32 发送如下消息：

消息 1： Std ID = 0x00，DLC = 1，数据 = 0x79 - ACK

消息 2： Std ID = 0x00，DLC = 1 数据 = N = 12 = 要发送的字节数 -1
(1 ≤ N +1 ≤ 256)

消息 3： Std ID = 0x00，DLC = 1，数据 = 自举程序版本（0 < 版本 ≤ 255）

消息 4： Std ID = 0x00，DLC = 1，数据 = 0x00 - Get 命令

消息 5： Std ID = 0x00，DLC = 1，数据 = 0x01 - Get Version & Read Protection Status 命令

消息 6： Std ID = 0x00，DLC = 1，数据 = 0x02 - Get ID 命令

消息 7： Std ID = 0x00，DLC = 1，数据 = 0x03 - Speed 命令

消息 8： Std ID = 0x00，DLC = 1，数据 = 0x11 - Read memory 命令

消息 9： Std ID = 0x00，DLC = 1，数据 = 0x21 - Go 命令

消息 10： Std ID = 0x00，DLC = 1，数据 = 0x31 - Write memory 命令

消息 11： Std ID = 0x00，DLC = 1，数据 = 0x43 - Erase memory 命令

消息 12： Std ID = 0x00，DLC = 1，数据 = 0x63 - Write Protect 命令

消息 13： Std ID = 0x00，DLC = 1，数据 = 0x73 - Write Unprotect 命令

消息 14： Std ID = 0x00，DLC = 1，数据 = 82h - Readout Protect 命令

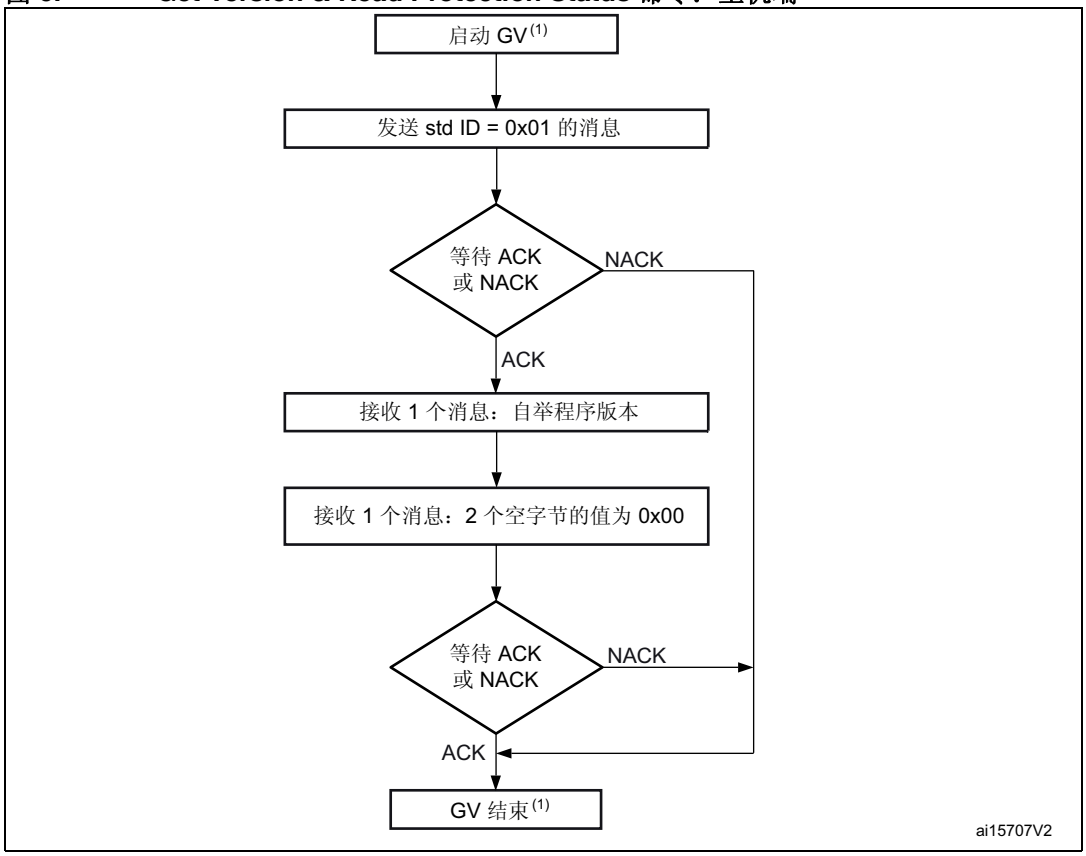
消息 15： Std ID = 0x00，DLC = 1，数据 = 92h - Readout Unprotect 命令

消息 1： Std ID = 0x00，DLC = 1，数据 = 0x79 - 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（版本和值为 0x00 的 2 个空字节）发送给主机。

图 6. Get Version & Read Protection Status 命令：主机端



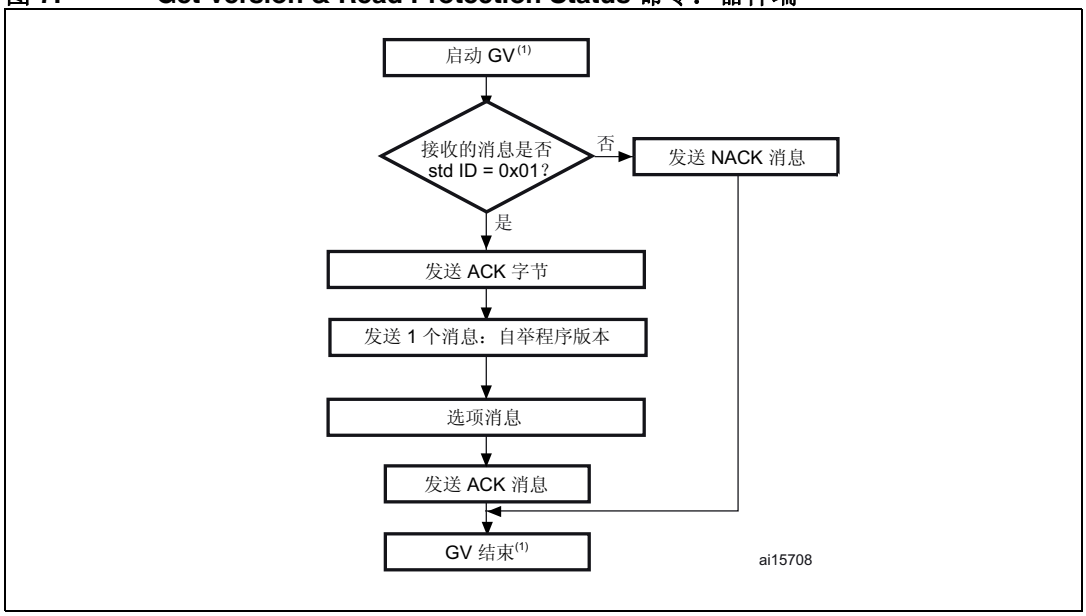
1. GV = Get Version & Read Protection Status。

主机发送如下消息：

命令消息：Std ID = 0x01，数据长度代码 (DLC) =“不重要”。

ACK 消息包含：Std ID = 0x01，DLC = 1，数据 = 0x79 - ACK

图 7. Get Version & Read Protection Status 命令：器件端



1. GV = Get Version & Read Protection Status。

STM32 发送如下消息：

消息 1： Std ID = 0x01，DLC = 1，数据 = ACK

消息 2： Std ID = 0x01，DLC = 1，数据[0] = 自举程序版本（0 < 版本 ≤ 255），如：0x10 = 版本 1.0

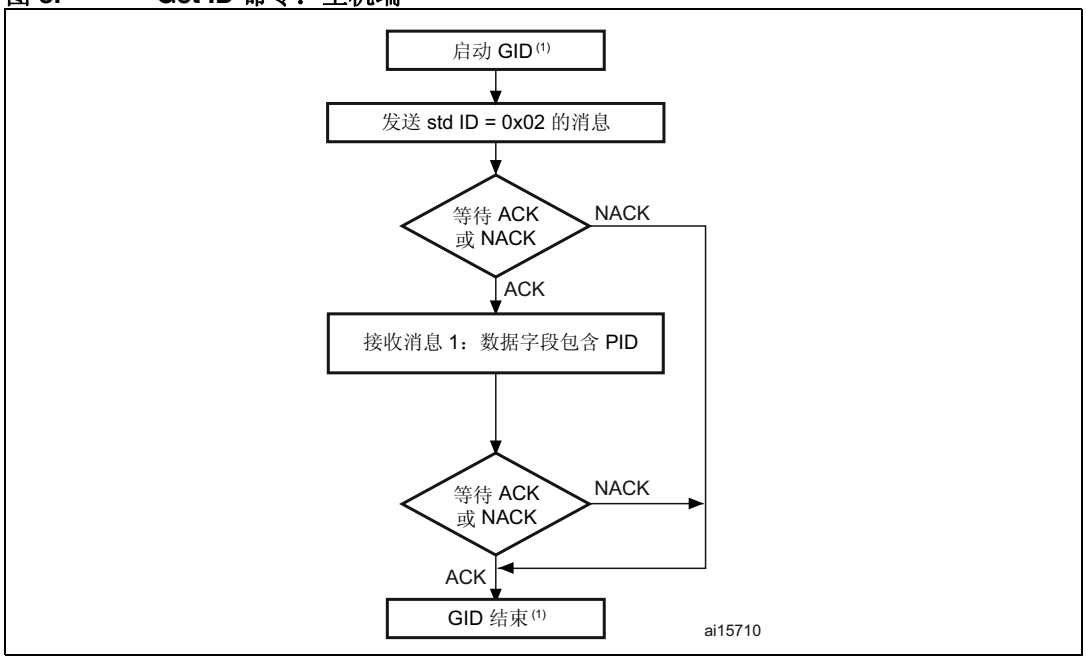
消息 3： 选项消息 1：Std ID = 0x01，DLC = 2，数据 = 0x00（字节 1 和字节 2）

消息 4： Std ID = 0x01，DLC = 1，数据 = 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。

图 8. Get ID 命令：主机端



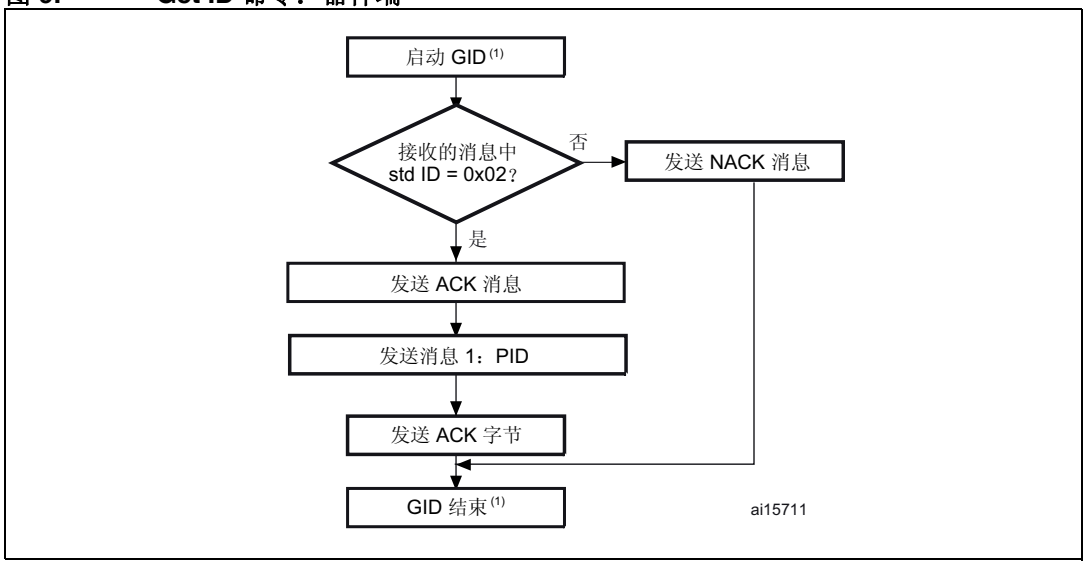
1. GID = Get ID。
2. PID 表示产品 ID。字节 1 是地址的 MSB，字节 2 是地址的 LSB。有关所用器件 PID 的详细信息，请参见 *第 3.1 节：器件相关的自举程序参数*。

主机发送如下消息：

命令消息：Std ID = 0x02，数据长度代码 (DLC) =“不重要”。

ACK 消息包含：Std ID = 0x02，DLC = 1，数据 = 0x79 - ACK

图 9. Get ID 命令：器件端



1. GID = Get ID。
2. PID 表示产品 ID。 字节 1 是地址的 MSB，字节 2 是地址的 LSB。

STM32 发送如下字节：

消息 1： Std ID = 0x02，DLC = 1，数据 = ACK，含 DLC（除当前消息和 ACK 外）。

消息 2： Std ID = 0x02，DLC = N（字节数 - 1。对于 STM32，N = 1），数据 = PID，其中字节 0 为产品 ID 的 MSB，字节 N 为产品 ID 的 LSB

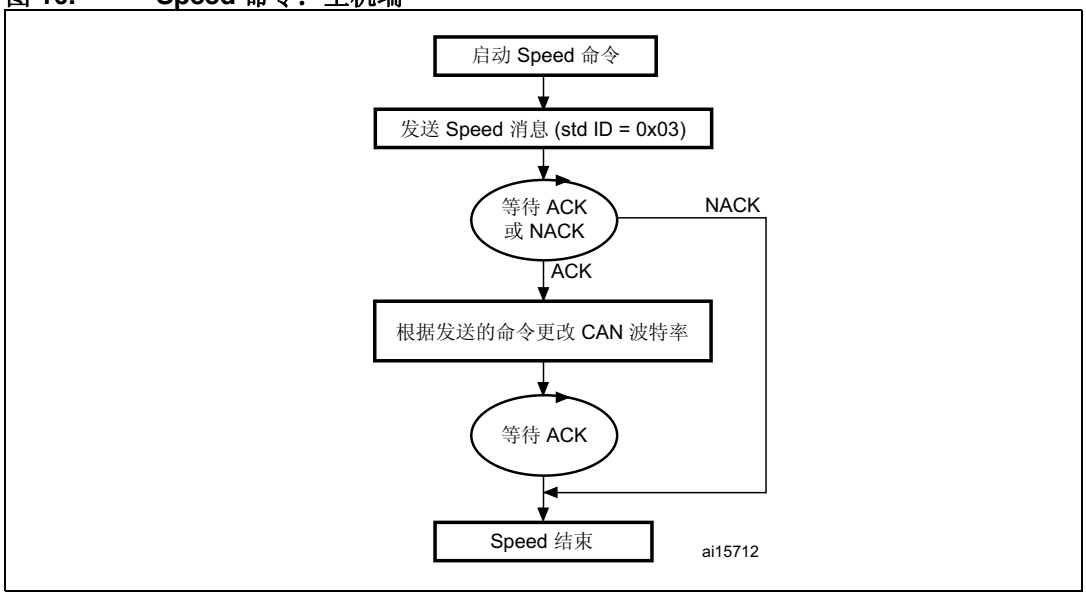
消息 3： Std ID = 0x02，DLC = 1，数据 = ACK = 0x79

3.5 Speed 命令

使用 Speed 命令可以更改 CAN 运行期间的波特率。只有 CAN 用作外设时才可使用此命令。

如果 CAN 接收到正确的消息但设置新波特率的操作失败，则会产生系统复位，防止其进入或离开初始化模式。

图 10. Speed 命令：主机端



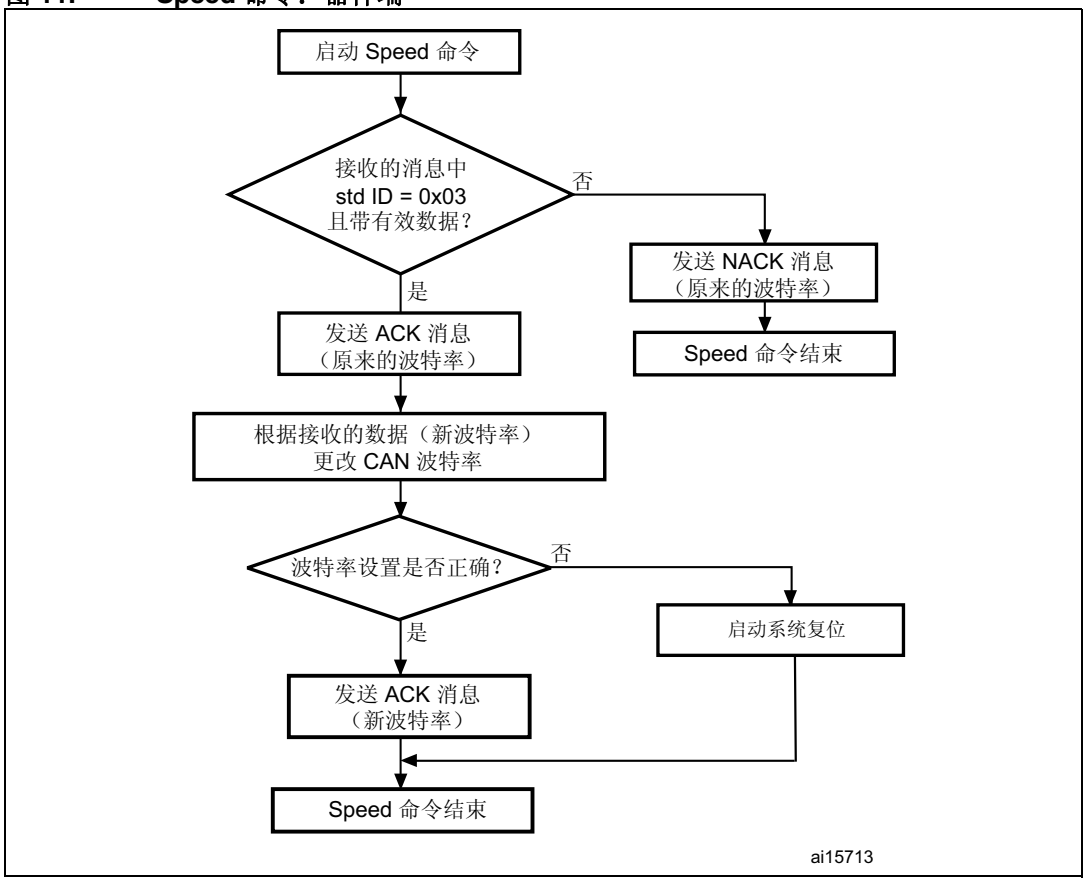
1. 成功设置新波特率后，自举程序会发送 ACK 消息。因此，主机在设置其波特率的同时会等待 ACK。

主机发送如下消息：

命令消息：Std ID = 0x03，DLC = 0x01，数据 [0] = XXh ；根据要设置的波特率，XXh 取如下值：

- 0x01 -> 波特率 = 125 kbps
- 0x02 -> 波特率 = 250 kbps
- 0x03 -> 波特率 = 500 kbps
- 0x04 -> 波特率 = 1 Mbps

图 11. Speed 命令：器件端



STM32 发送如下字节：

消息 1： Std ID = 0x03，DLC = 1，数据[0] = ACK= 0x79：如果接收到正确的消息，则使用原来的波特率，否则数据[0] = NACK= 0x1F

消息 2： Std ID = 0x03，DLC = 1，数据[0] = ACK = 0x79 且使用新波特率

3.6 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： *如需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。*

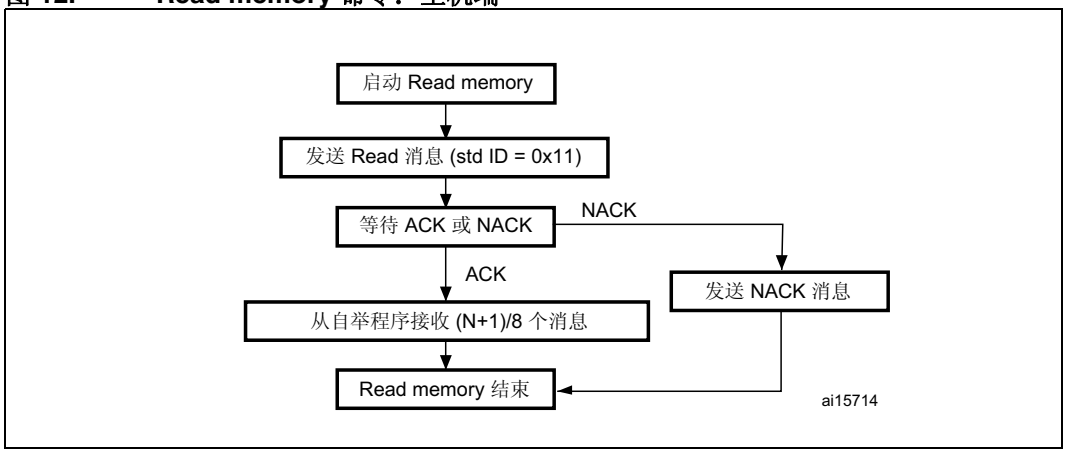
自举程序接收到 Read Memory 命令后，会开始验证消息的内容：

- 命令 ID 是否正确
- ReadOutProtection 已使能还是禁止
- 要读取的地址是否有效

如果消息内容正确，则发送 ACK 消息，否则发送 NACK 消息。

发送 ACK 消息后，从接收的地址开始向应用程序发送所需数据（(N + 1) 字节），共占用 (N+1) / 8 个消息（由于每个消息包含 8 个字节）。

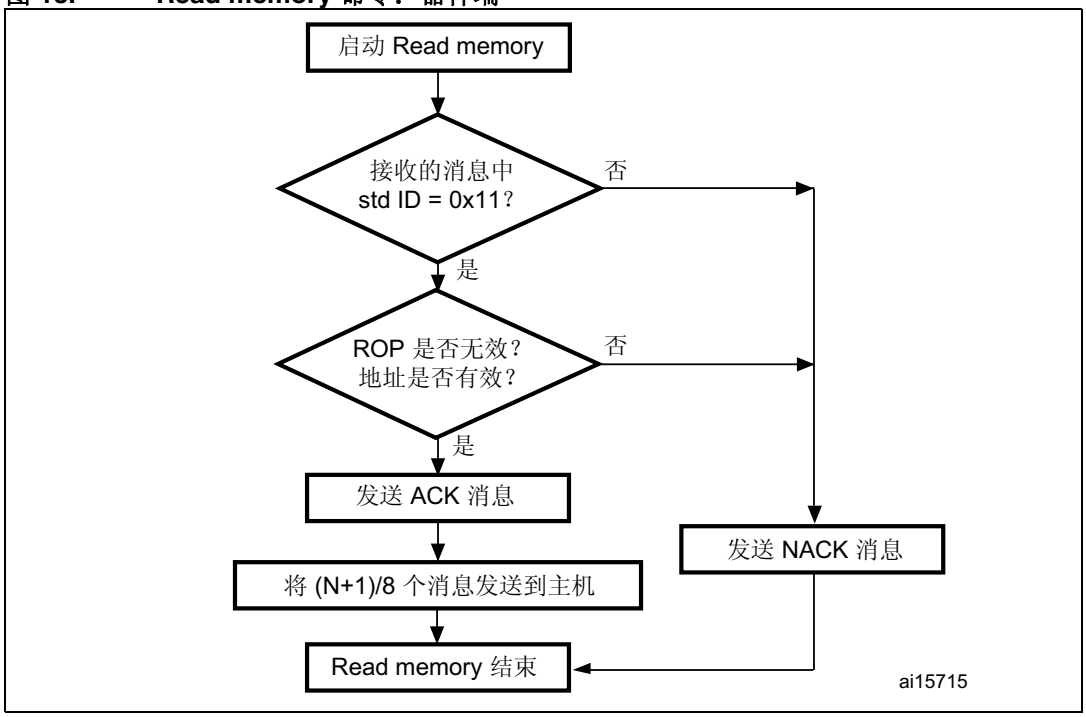
图 12. Read memory 命令：主机端



主机发送如下消息：

命令消息：
Std ID = 0x11，DLC = 0x05，数据[0] = 0xXX：地址的 MSB... 数据[3] = 0xYY：地址的 LSB，数据[4] = N：要读取的字节数 (0 < N ≤ 255)。

图 13. Read memory 命令：器件端



STM32 发送如下消息：

ACK 消息：Std ID = 0x11，DLC = 1，如果命令的内容正确，数据[0] = ACK，否则数据[0] = NACK

数据消息 (N+1) / 8：Std ID = 0x11，DLC = 字节数，数据[0] = 0xXX... 数据[字节数 - 1] = 0xYY

ACK 消息：Std ID = 0x11，DLC = 1，数据[0] = ACK

3.7 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，会在消息中包含以下有效信息时开始执行代码：

- 命令 ID 是否正确
- ReadOutProtection 已使能还是禁止
- 分支目标地址是否有效（数据[0] 是地址的 MSB，数据[4] 是地址的 LSB）

如果消息内容正确，则发送 ACK 消息，否则发送 NACK 消息。

将 ACK 消息发送给应用程序后，自举程序固件将执行以下操作：

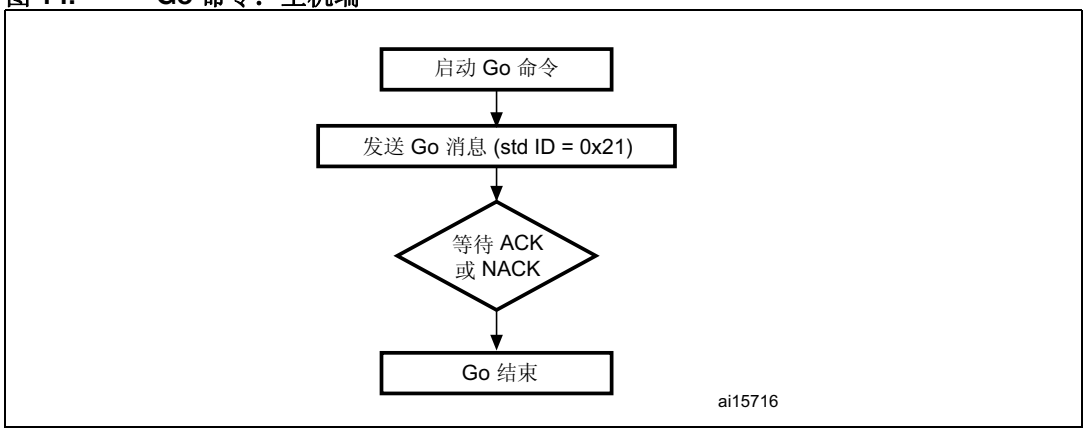
- 初始化自举程序使用的外设寄存器，将其设置为默认复位值
- 初始化用户应用程序的主栈指针
- 跳转到接收的“地址 + 4”（与应用程序中复位处理程序的地址相对应）中指定的存储器位置。
  例如，如果接收的地址为 0x0800 0000，则自举程序将跳转到地址为 0x0800 0004 的存储器位置。
  一般来说，主机应发送基准地址，在该地址指定应用程序的跳转目标位置

注： *1 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。*

2 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。

3 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）。

图 14. Go 命令：主机端

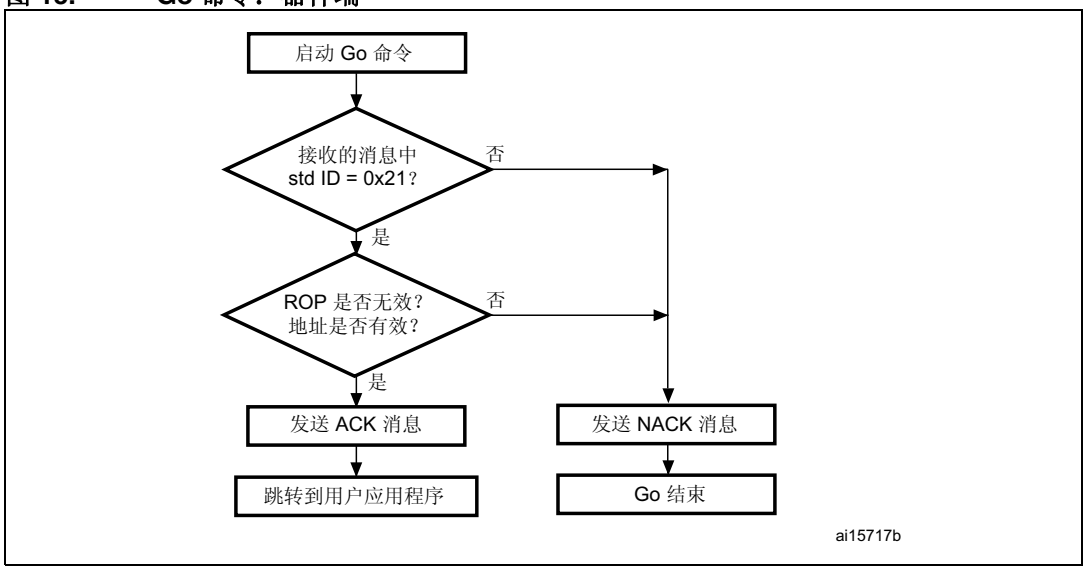


1. 有效地址请参见产品数据手册。

主机发送如下字节

Go 命令消息：Std ID = 0x21，DLC = 0x04，数据[0] = 0xXX：MSB 地址，...数据[3] = 0xYY LSB 地址。

图 15. Go 命令：器件端



STM32 发送如下消息：

ACK 消息：Std ID = 0x21，DLC = 1，如果命令内容正确，则数据[0] = ACK，否则数据[0] = NACK

3.8 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（参见注释）。自举程序接收到 Write Memory 命令后（数据长度为 5 个字节的消息，数据 [0] 是地址的 MSB，数据 [3] 是地址的 LSB，数据 [4] 是要接收的数据字节数），会检查接收的地址。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： *如需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。*

如果接收的地址有效，则自举程序将发送 ACK 消息，否则发送 NACK 消息并中止此命令。地址有效时，自举程序将：

- 接收用户数据（N 个字节），器件会接收到 N/8 个消息（每个消息包含 8 个数据字节）
- 从接收的地址开始将用户数据编程到存储器中
- 在命令结束时，如果写操作成功完成，则自举程序将发送 ACK 消息；否则将 NACK 消息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。

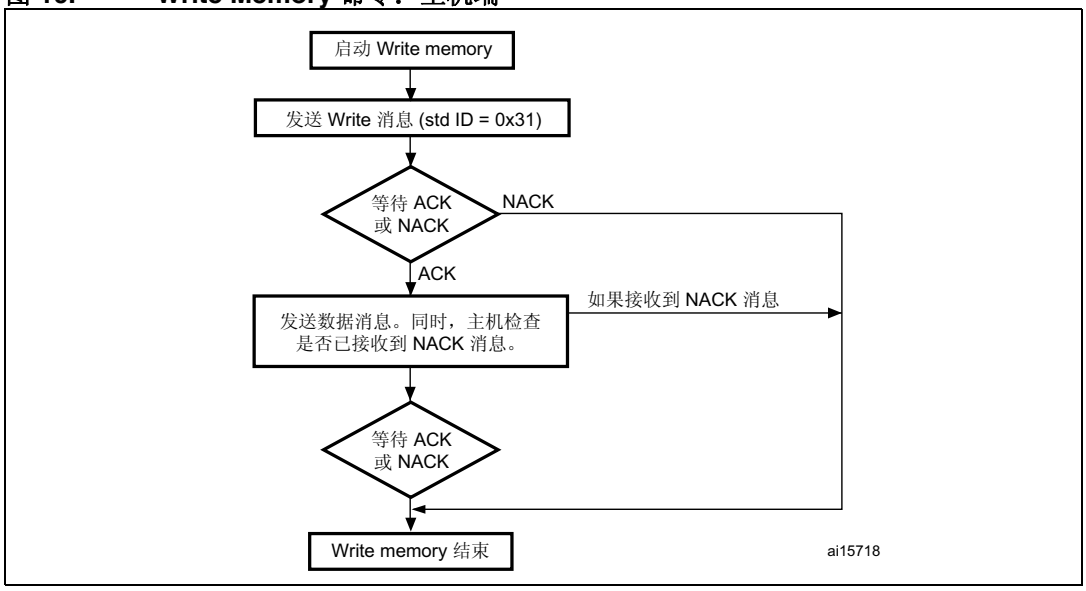
如果将 Write Memory 命令发送到选项字节区域，则在写入新值前会清除所有的选项，并由自举程序在命令结束时启动系统复位来实施新的选项字节配置。

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。*

对 Flash/SRAM 的写入操作必须按字对齐。如果写入数据少于分配的存储空间，则以 0xFF 填充剩余字节。

图 16. Write Memory 命令：主机端



注： *如果起始地址无效，则命令将收到器件发出的 NACK 应答。*

主机发送如下消息：

命令消息：Std ID = 0x31，DLC = 0x05，数据[0] = 0xXX：MSB 地址，...，数据[3] = 0xYY：LSB 地址，数据[4] = N-1 （要写入的字节数 - 1， 0 < N ≤ 255）。

随后主机发送 N/8 个消息

数据消息：Std ID = 0x31，DLC_1 = 1 到 8，数据 = 字节_11, … 字节_18…

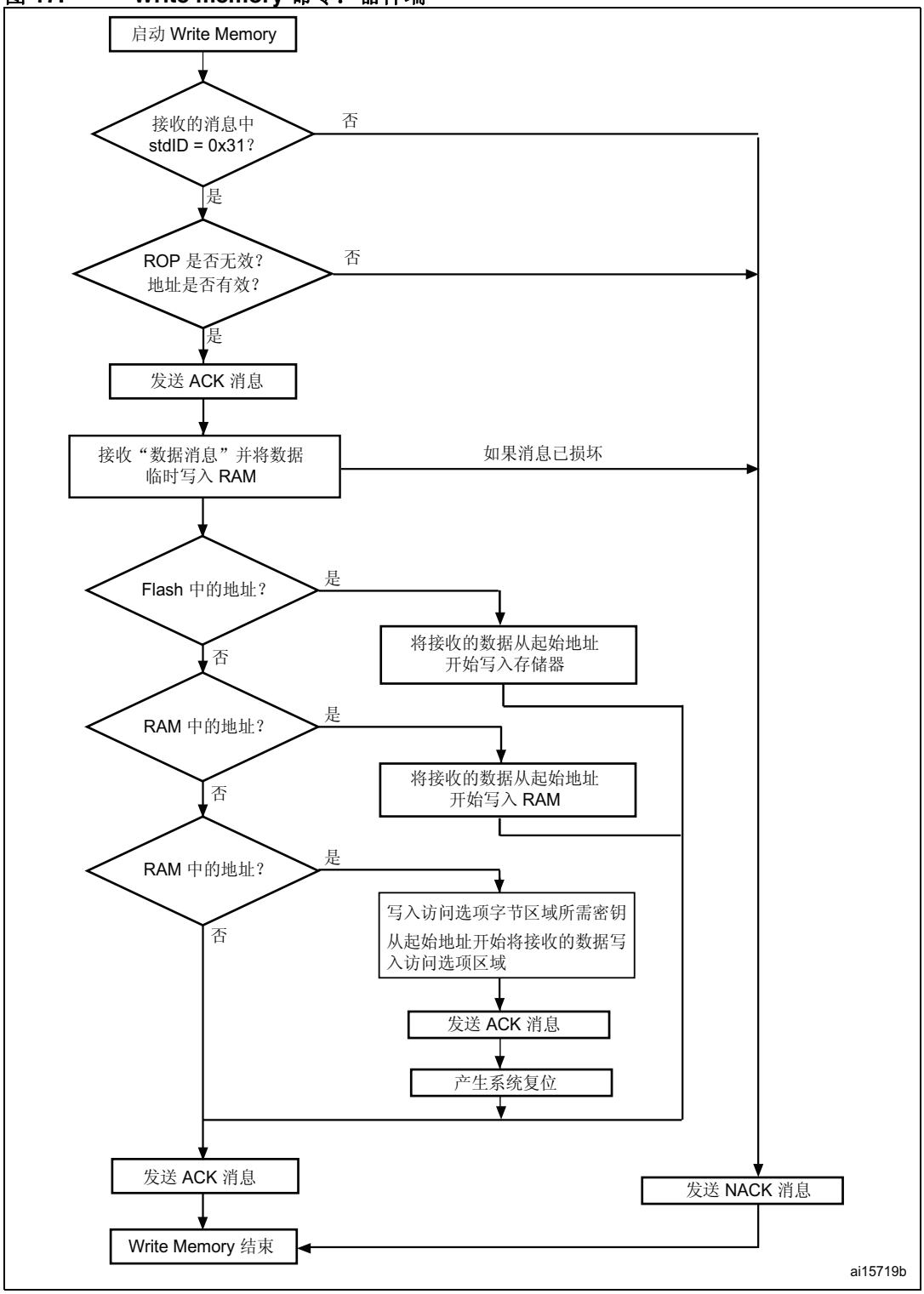
数据消息_M：Std ID = 0x04，DLC_M = 1 到 8，数据 = 字节_m1，…，字节_M8

注： *1* *DLC_1 + DLC_2 + ...DLC_M = 最大 256。*

2 *在每条消息后，主机都会从器件收到 ACK 或 NACK 消息。*

3 *自举程序不会检查数据的标准 ID，因此可使用从 0h 到 0xFF 的任意 ID。推荐使用 0x04。*

图 17. Write memory 命令：器件端



STM32 发送如下消息：

ACK 消息：Std ID = 0x31，DLC = 1，如果命令内容正确，则数据[0] = ACK，否则数据[0] = NACK

器件在每次接收到消息后，如果命令内容正确，则发送 ACK，否则发送 NACK。但正如 *图 17* 中所述，在接收到所有“数据消息”（N/8 个消息）并将数据临时写入 RAM 后，如果消息内容没有损坏，自举程序将在请求的地址（Flash、RAM 或选项字节）写入 N 个字节；如果写保护操作成功完成，器件会将 ACK 消息发送给主机。

也就是说，在发送 N/8 个消息后，主机将接收两个连续的 ACK；第一个由器件在正确接收全部的 N/8 个消息后发送，第二个则在向请求的地址正确写入 N/8 个消息后发送

3.9 Erase Memory 命令

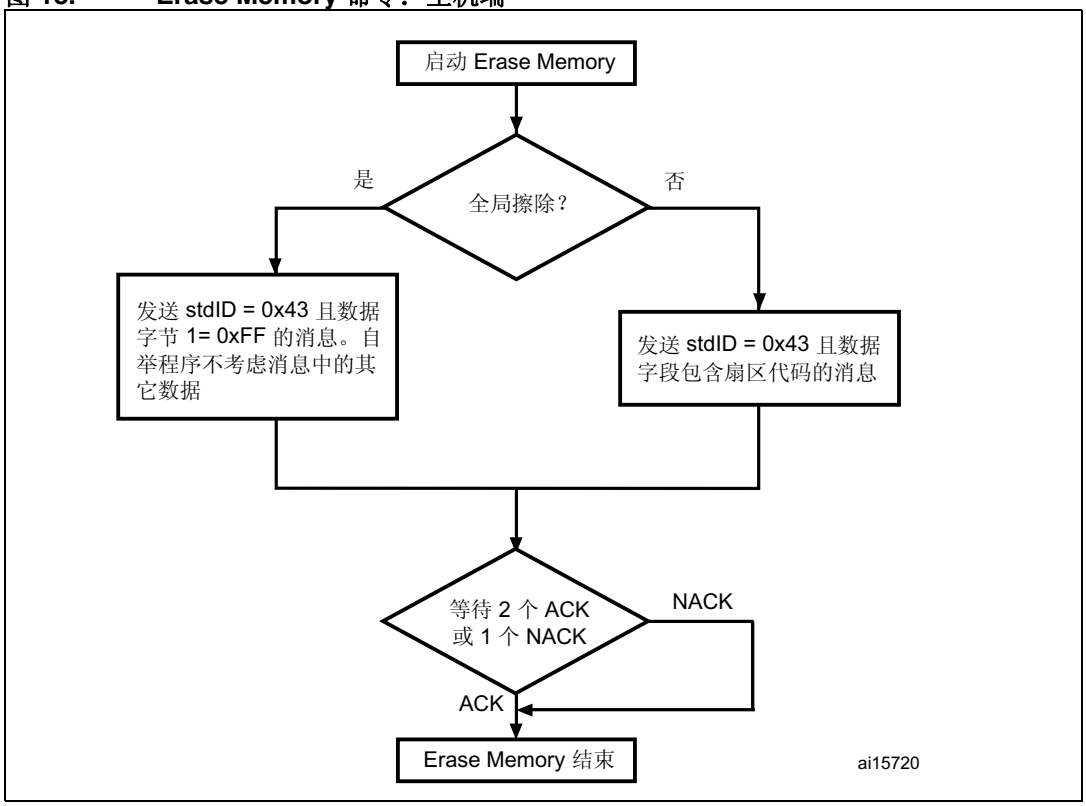
主机可通过 Erase Memory 命令擦除 Flash 页面。自举程序在接收到 Erase Memory 命令并禁止 ROP 后，会将 ACK 消息发送到主机。发送 ACK 消息后，自举程序会检查数据[0] 是否为 0xFF。如果是，则启动全局存储器擦除操作，并在完成后发送 ACK 消息。否则（即数据[0] 不是 0xFF），自举程序将根据主机的定义启动存储器页擦除操作，并在擦除每页后发送 ACK 或 NACK 消息。

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）的消息。
   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，每个字节都包含页编号

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 18. Erase Memory 命令：主机端



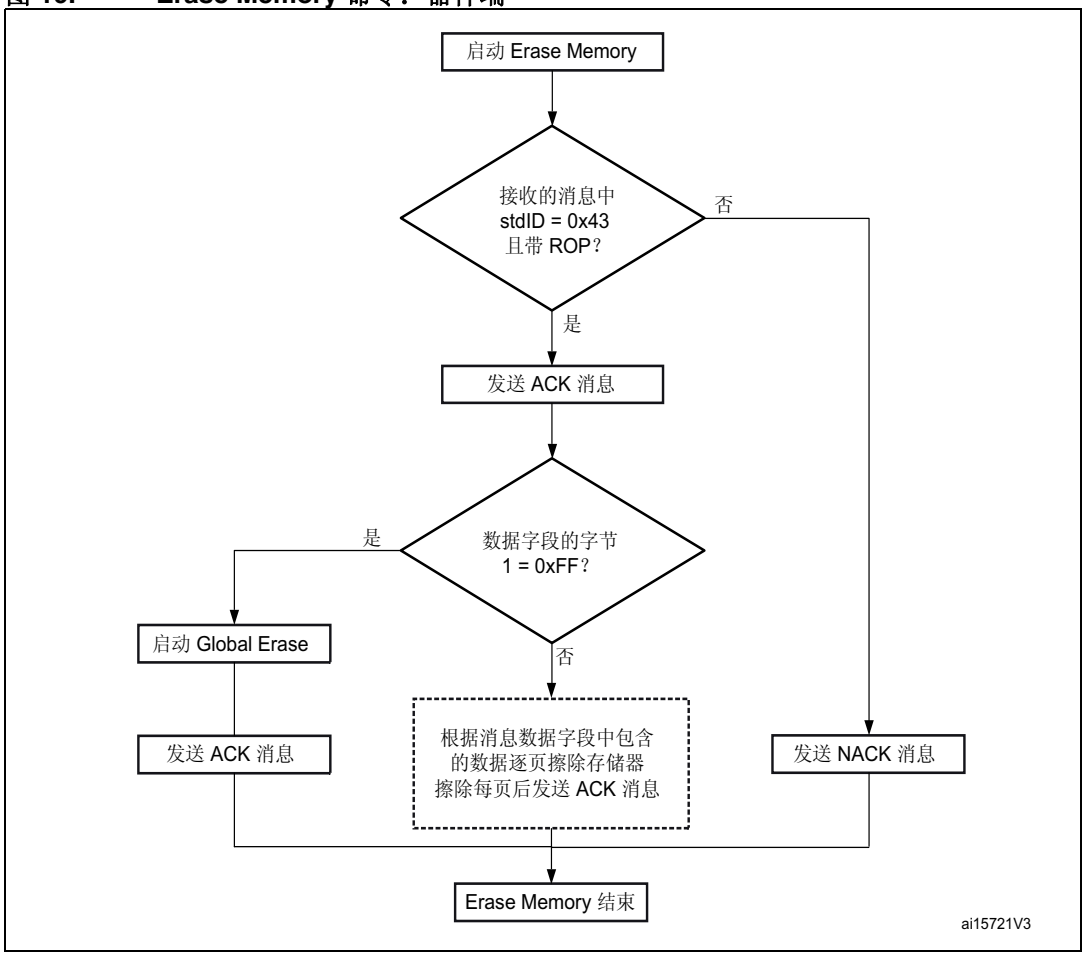
主机发送如下消息：

ID 包含命令类型 (0x43)：

- 总擦除消息：Std ID = 0x43，DLC = 0x01，数据 = 0xFF。
- 逐个擦除扇区消息：Std ID = 0x43，DLC = 0x01 到 0x08，数据 = 参见产品数据手册。

采用逐页擦除时，主机在每条消息后都会从器件接收到 ACK 或 NACK 消息。

图 19. Erase Memory 命令：器件端



STM32 发送如下消息：

ACK 消息：Std ID = 0x43，DLC = 1，如果命令内容正确且 ROP 未激活，则数据[0] = ACK，否则数据[0] = NACK

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。自举程序在接收到 Write Protect 命令后，如果已禁止 ROP，则将 ACK 消息发送到主机，否则发送 NACK。

发送 ACK 字节后，自举程序将等待从应用程序接收 Flash 扇区代码。

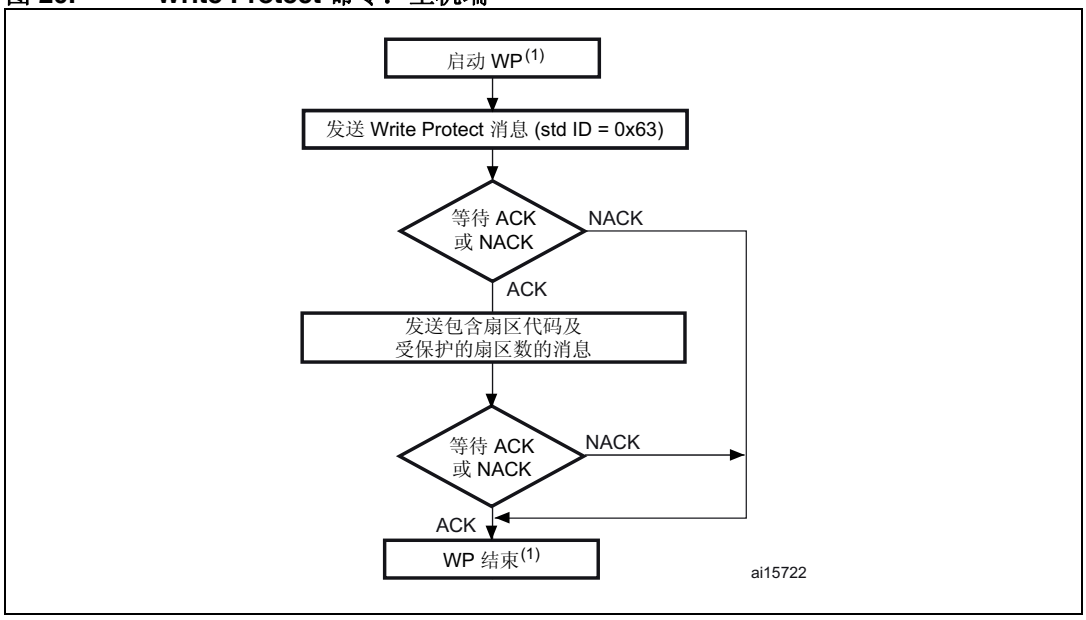
在 Write Protect 命令结束时，自举程序会发送 ACK 消息并启动系统复位，以实施新的选项字节配置。

注： *1* *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。*

2 *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。*

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 20. Write Protect 命令：主机端



1. WP = Write Protect。

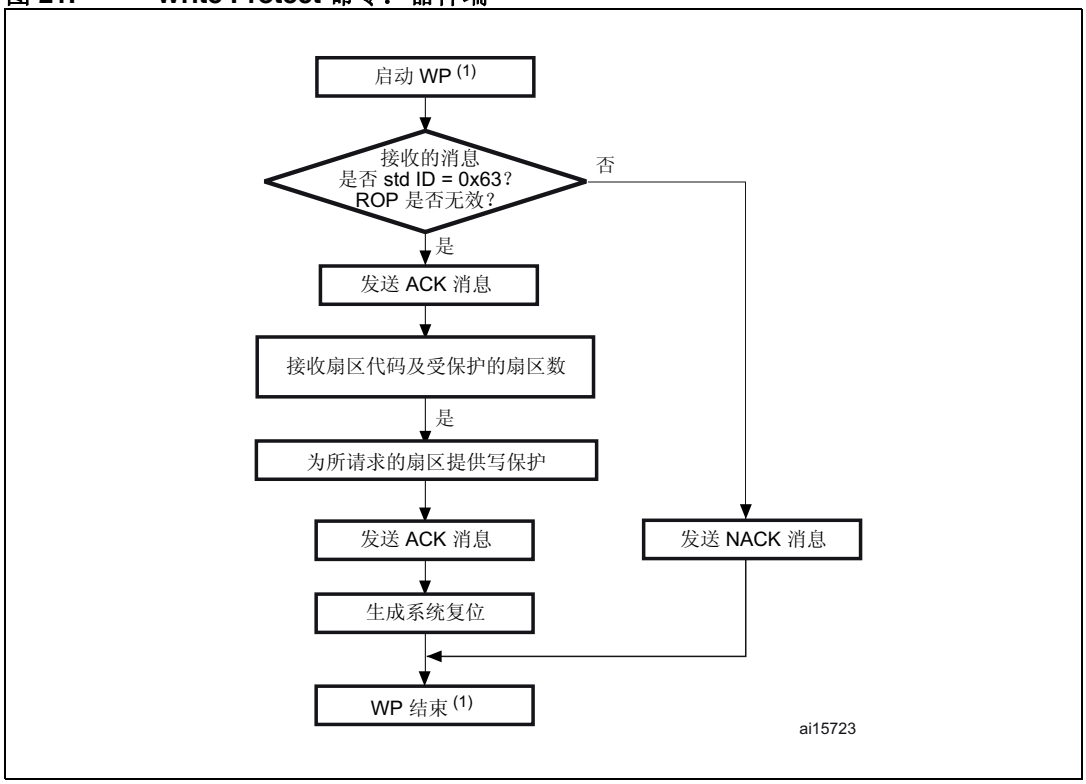
主机发送如下消息：

命令消息：Std ID = 0x63，DLC = 0x01，数据[0] = N (0 < N ≤ 255)。

命令消息：Std ID = 0x63，DLC = 0x01..08，数据[0] = N (0 < N ≤ 255)。

在每条消息后，主机都会从器件收到 ACK 或 NACK 消息。

图 21. Write Protect 命令：器件端



1. WP = Write Protect

STM32 发送如下消息：

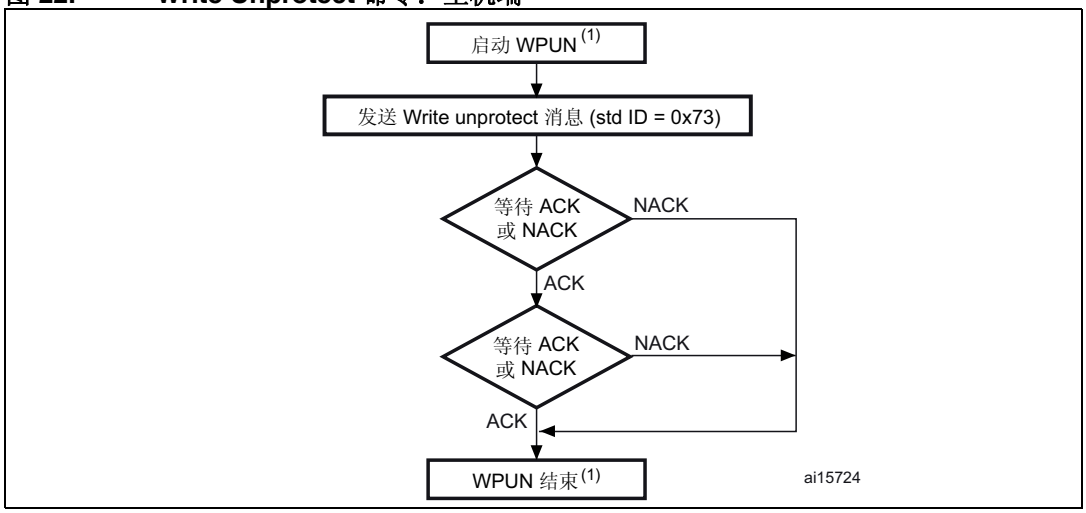
ACK 消息：Std ID = 0x63，DLC = 1，如果命令内容正确且 ROP 未激活，则数据 [0] = ACK，否则数据 [0] = NACK。

3.11 Write Unprotect 命令

Write Unprotect 命令用于禁止所有 Flash 扇区的写保护。自举程序在接收到 Write Unprotect 命令后，如果已禁止 ROP，则将 ACK 消息发送到主机，否则发送 NACK。发送 ACK 消息后，自举程序会禁止所有 Flash 扇区的写保护。

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

图 22. Write Unprotect 命令：主机端

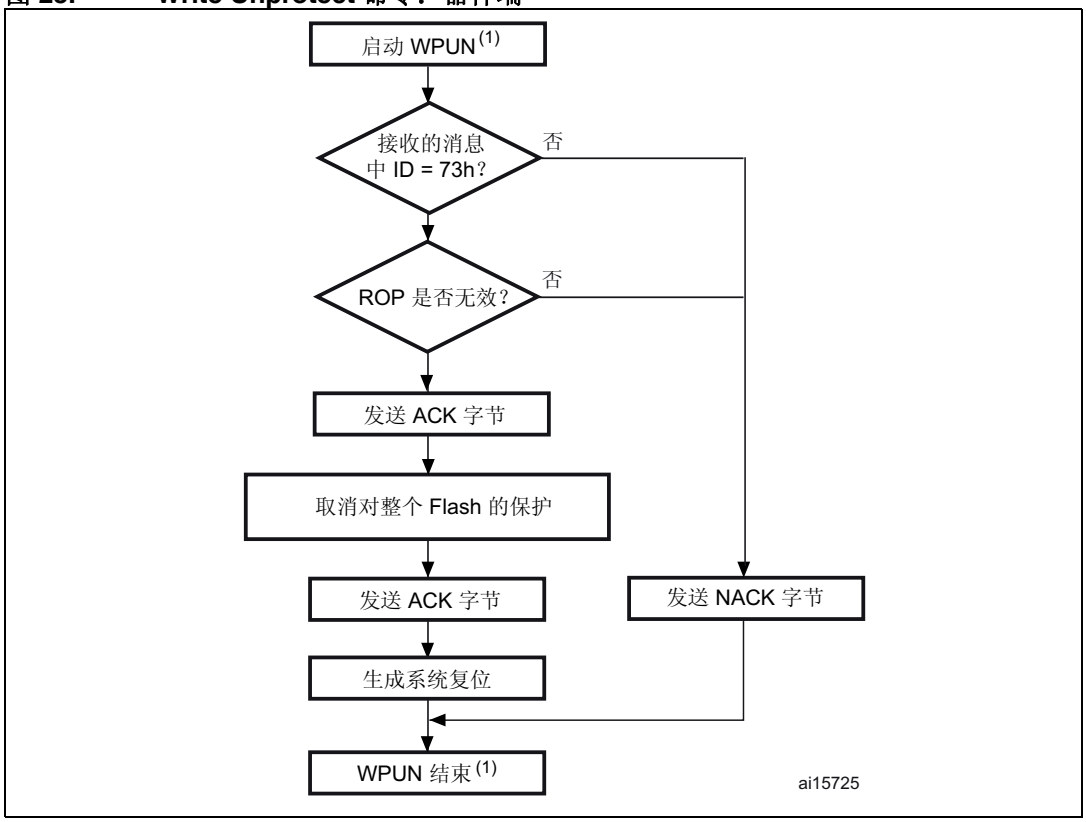


1. WPUN = Write Unprotect。

主机发送如下消息：

命令消息：Std ID = 0x73，DLC = 0x01，数据 = 00。

图 23. Write Unprotect 命令：器件端



1. WPUN = Write Unprotect。

STM32 发送如下消息：

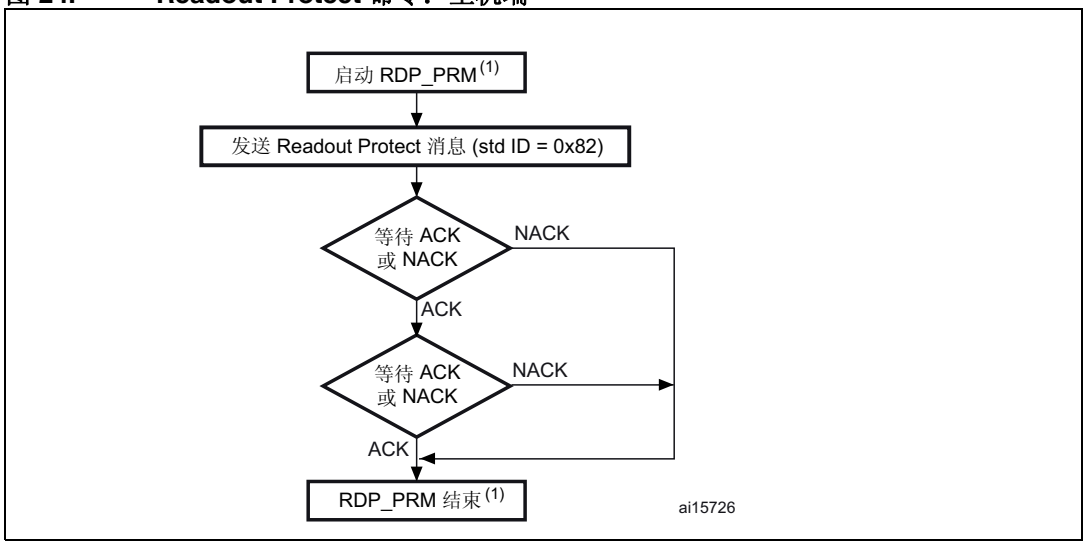
ACK 消息：Std ID = 0x73，DLC = 1，如果命令内容正确且 ROP 未激活，则数据[0] = ACK，否则数据[0] = NACK。

3.12 Readout Protect 命令

Readout Protect 命令用于使能 Flash 读保护。自举程序在接收到 Readout Protect 命令后，如果已禁止 ROP，则将 ACK 消息发送到主机，否则发送 NACK。发送 ACK 消息后，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 消息并启动系统复位，以实施新的选项字节配置。

图 24. Readout Protect 命令：主机端

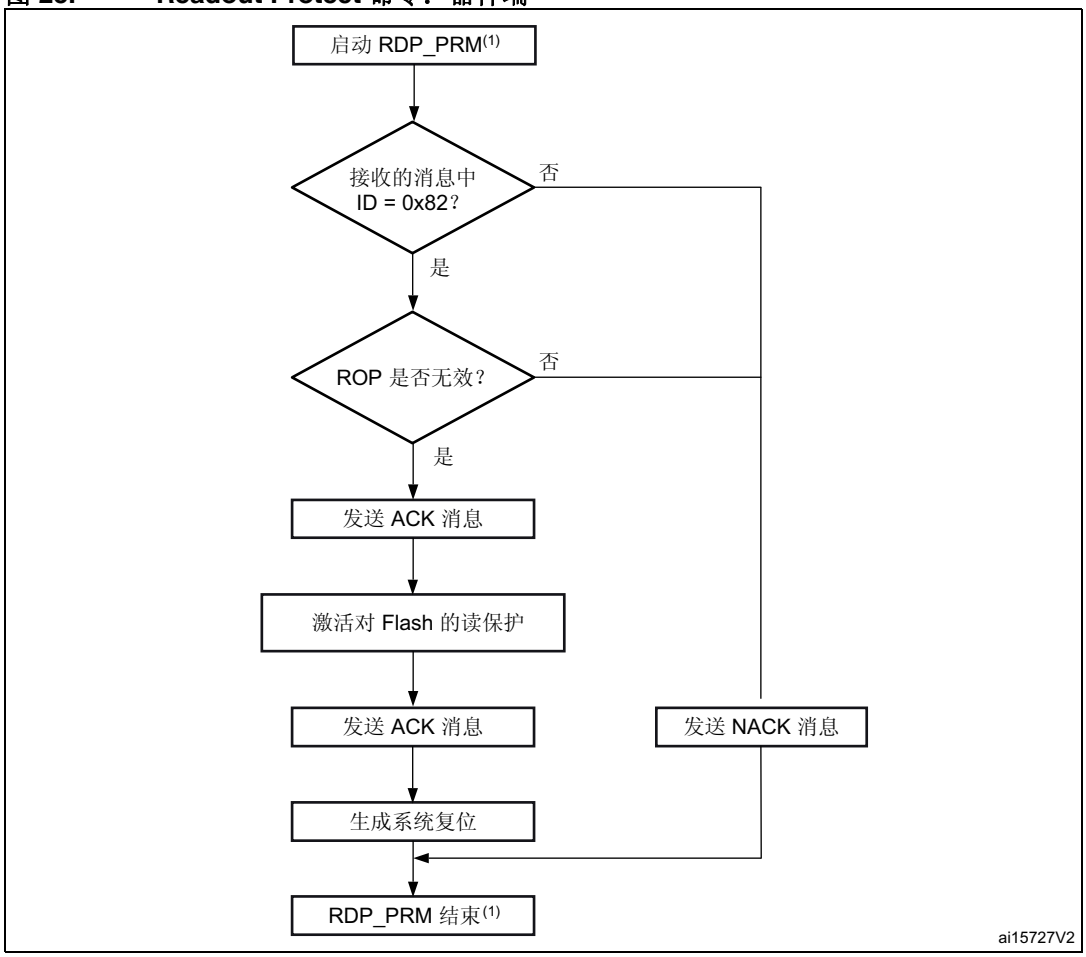


1. RDP_PRM = Readout Protect。

主机发送如下消息

命令消息：Std ID = 0x82，DLC = 0x01，数据[0] = 00。

图 25. Readout Protect 命令：器件端



1. RDP_PRM = Readout Protect

STM32 发送如下消息：

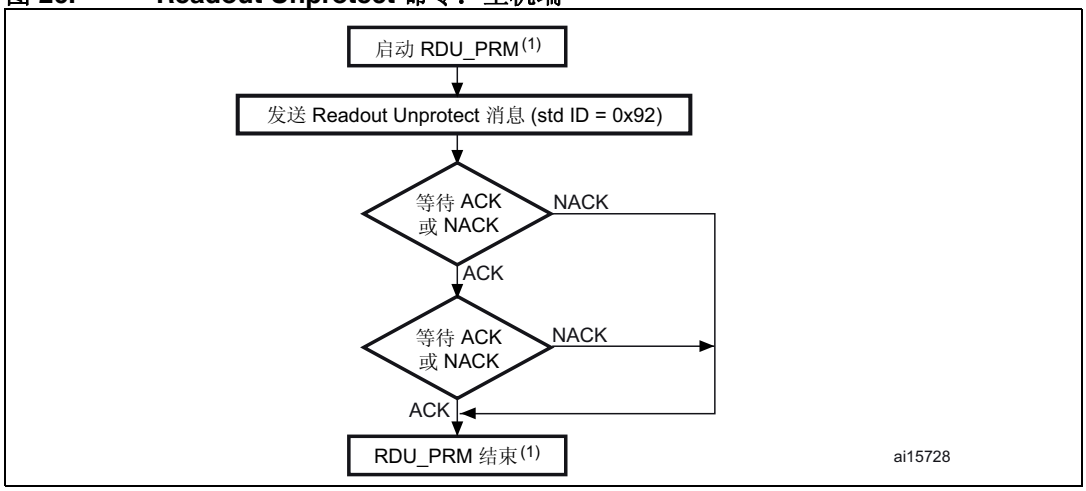
ACK 消息：Std ID = 0x82，DLC = 1，如果命令内容正确且 ROP 未激活，则数据[0] = ACK，否则数据[0] = NACK。

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 消息发送到主机。发送 ACK 消息后，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。

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

图 26. Readout Unprotect 命令：主机端

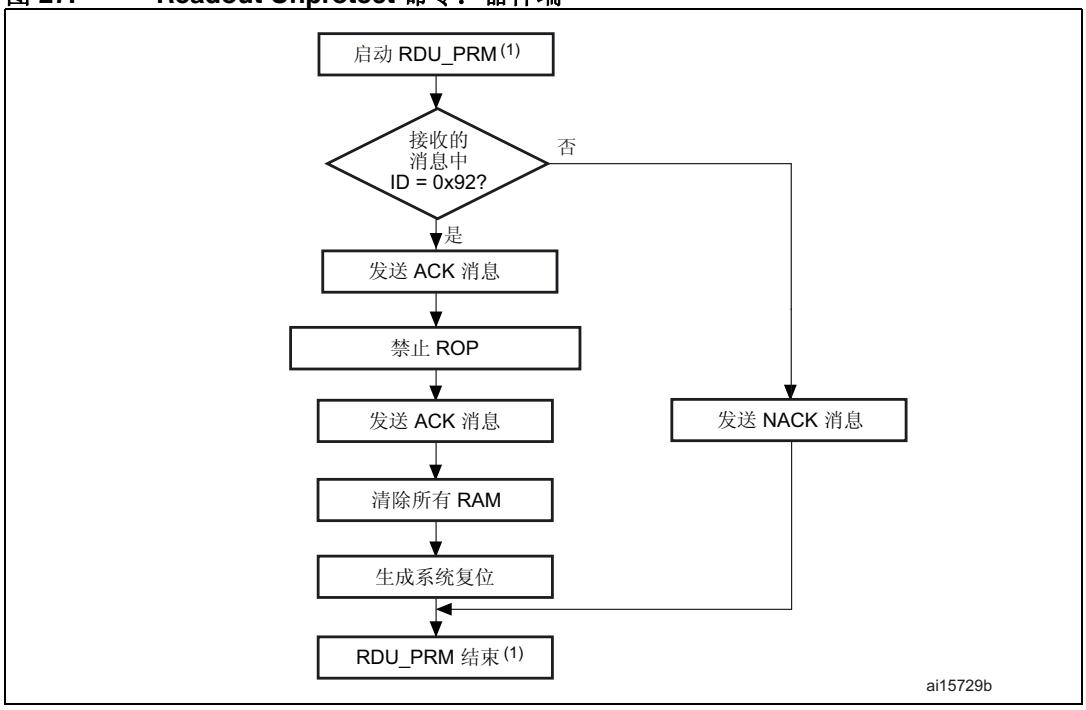


1. RDU_PRM = Readout Unprotect。

主机发送如下消息

命令消息：Std ID = 0x92，DLC = 0x01，数据 = 00。

图 27. Readout Unprotect 命令：器件端



1. RDU_PRM = Readout Unprotect。

STM32 发送如下消息：

ACK 消息：Std ID = 0x92，DLC = 1，如果命令内容正确且 ROP 未激活，则数据[0] = ACK，否则数据[0] = NACK。

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	初始自举程序版本。

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。
2011 年 04 月 15 日	2	更新了*图 2：检查 HSE 频率*并增加了*注 1*。
2011 年 04 月 22 日	3	更新了*第 3.5 节：Speed 命令*中消息 2 的 Std ID。
2012 年 10 月 24 日	4	更新了 *第 3.3 章：Get Version & Read Protection Status 命令* *图 6：Get Version & Read Protection Status 命令：主机端* *第 3.8 章：Write Memory 命令* *第 3.9 章：Erase Memory 命令* *图 19：Erase Memory 命令：器件端* *第 3.10 章：Write Protect 命令* *图 25：Readout Protect 命令：器件端*

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