# GMF03x 自举程序说明

版本：**1.0**

发布日期：**2020-04-28**

深圳市汇顶科技股份有限公司

## 商标声明

GOODIX 和其他汇顶商标均为深圳市汇顶科技股份有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人持有。

## 免责声明

本文档中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。

深圳市汇顶科技股份有限公司（以下简称“GOODIX”）对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。GOODIX 对因这些信息及使用这些信息而引起的后果不承担任何责任。

未经 GOODIX 书面批准，不得将 GOODIX 的产品用作生命维持系统中的关键组件。在 GOODIX 知识产权保护下，不得暗中或以其他方式转让任何许可证。

## 深圳市汇顶科技股份有限公司

总部地址：深圳市福田保税区腾飞工业大厦 B 座 2 层、13 层

电话：+86-755-33338828　　传真：+86-755-33338830

网址：http://www.goodix.com

# 前言

## 编写目的

本文详细介绍了 GMF03x 系列芯片自举程序的激活配置、执行流程、命令说明以及时间参数，旨在帮助用户使用自举程序实现简单的芯片调试功能，如：内存读写、固件升级等。

## 读者对象

本文适用于以下读者：

- GMF03x 应用开发者
- 项目或产品经理
- 开发爱好者

## 版本说明

本文档为第 1 次发布。

## 修订记录

| 版本 | 日期 | 修订内容 |
| --- | --- | --- |
| 1.0 | 2020-04-28 | 首次发布。 |

# 目录



 

# 1 简介

GMF03x 自举程序是固化在系统存储区（System Memory）的可执行程序，可不依赖调试接口实现用户选项字节（User Option Bytes）配置、SRAM 访问、Flash 读写/擦除、固件升级等功能。

# 2 自举程序的激活与硬件连接

激活自举程序需配置芯片从系统存储区启动，即用户选项字节 CFG 的 nBOOT1 位为 1（出厂默认值）并在芯片启动时拉高 BOOT0 引脚。

**提示：**

若用户选项字节的 RPROT 为 0xCC（保护等级 2），芯片只能从 Flash 启动。

激活自举程序的硬件连接示意图，如图 2-1 所示。

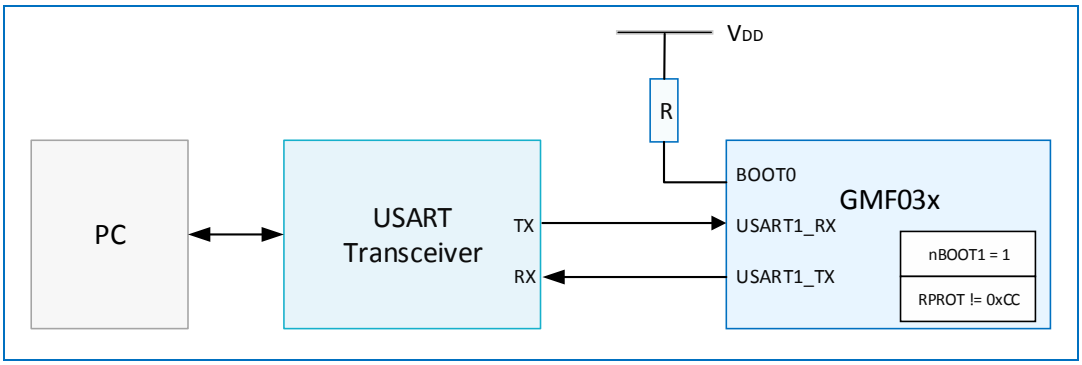


图 2-1 激活自举程序的配置及硬件连接

其中：

- PC 为主机端，用于下发命令，收发数据。
- USART Transceiver 通常为 USB 转串口设备，作为传输信道。
- GMF03x 为芯片端，执行主机端的指令。

自举程序支持两种硬件连接方案，均占用 3 个引脚，具体配置说明参见表 2-1。

表 2-1 自举程序连接方案

| 连接方案 | 占用的引脚资源 | 引脚配置 | 说明 |
|---|---|---|---|
| 方案 1 | BOOT0, PA14, PA15 | • BOOT0 连接 VDD<br>• PA14 复用为 USART1_TX<br>• PA15 复用为 USART1_RX | 将导致调试接口不可用（SWCLK 依赖的 PA14 被 USART1 占用）。 |
| 方案 2 | BOOT0, PA9, PA10 | • BOOT0 连接 $V_{DD}$<br>• PA9 复用为 USART1_TX<br>• PA10 复用为 USART1_RX | 调试接口仍可用。 |

# 3 自举程序执行流程

自举程序执行流程包括初始化、同步和命令三个阶段，如图 3-1 所示。

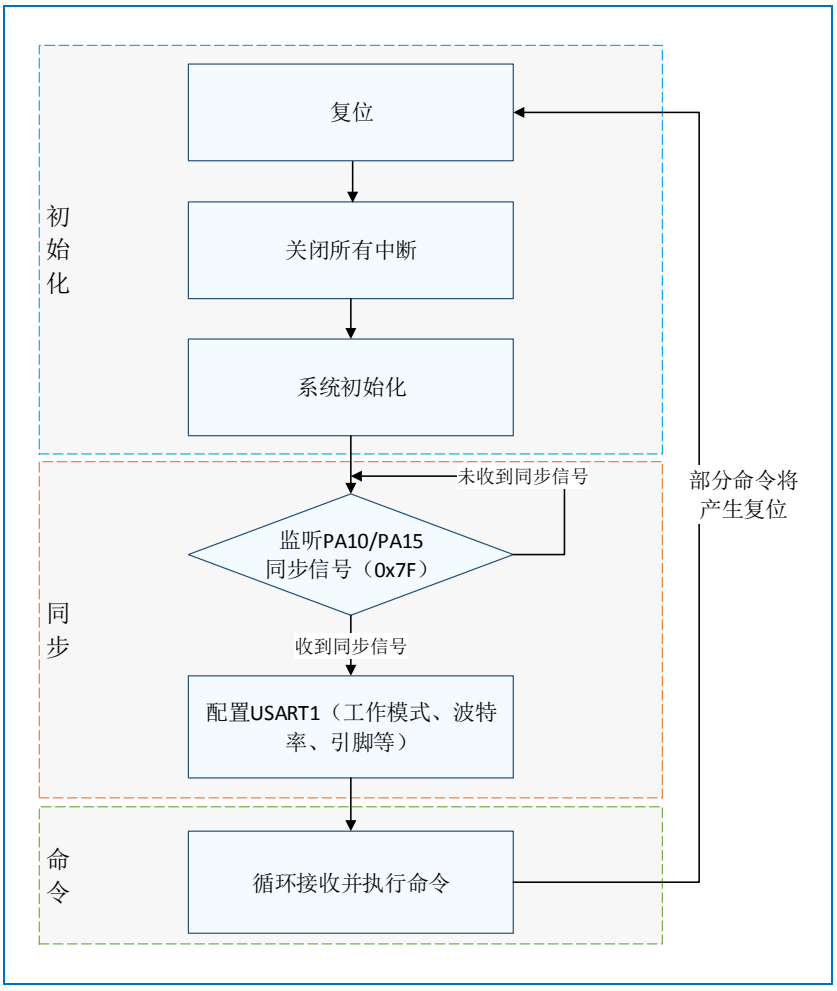


图 3-1 自举程序执行流程

## 3.1 初始化阶段

上电或系统复位后，自举程序将关闭所有中断，并对系统进行初始化配置参见表 3-1。

表 3-1 自举程序初始化配置

| 外设 | 配置 |
|---|---|
| 复位与时钟控制（RCC） | PLL 输入源选择 RC8M。<br>系统时钟设置为 24 MHz。 |
| SRAM | 0x20000000 起始的 2K 字节为自举程序占用 SRAM，不可通过命令访问。<br>0x20000800 以后的 SRAM 支持通过命令访问。 |
| 独立看门狗（FWDT） | 若用户选项字节 CFG 的 SWFWDT 位为 0，自举程序将独立看门狗（FWDT）时钟周期配置为最大值。<br>自举程序将定期刷新 FWDT 以防止芯片复位。 |
| GPIOA | PA10 和 PA15 为通用 IO，配置上拉输入状态。<br>PA13 与 PA14 为调试接口的 SWDIO 及 SWCLK。 |
| CPU | 关闭所有中断。 |
| 其它外设 | 时钟关闭。 |

## 3.2 同步阶段

主机端：主机需发送同步信号“0x7F”（LSB First）。信号包含 1 位低电平（Start 位）、7 位高电平（均为 1）、1 位低电平（最高有效位）以及高电平（校验码与 Stop 位）。

芯片端：自举程序将监测 PA10/PA15 引脚是否被拉低。当任意引脚电平被拉低时，自举程序检测到高电平时开始计数，并再次检测到低电平时停止计数，如图 3-2 所示。

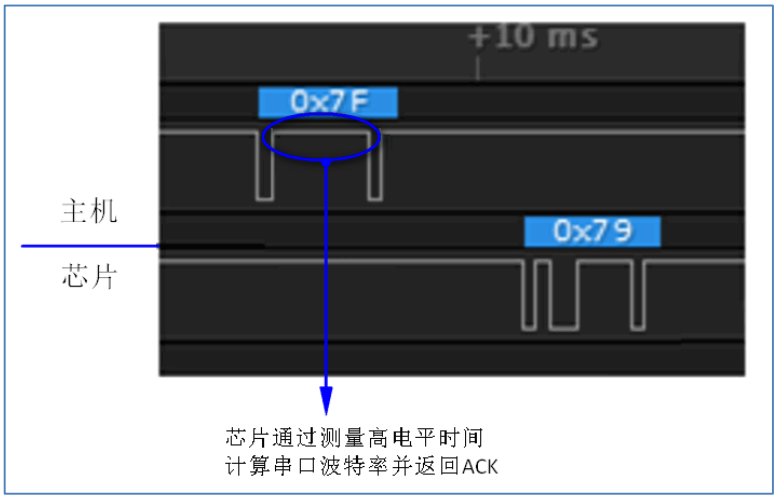


图 3-2 自举程序同步结束后将回复主机端“0x79”（ACK）

通过计算高电平的持续时间，确定 USART1 通信波特率后，芯片将按表 3-2 所示配置 USART1 并回复“0x79”（ACK）。

表 3-2 USART1 的配置

| USART1 配置项 | 说明 |
|---|---|
| 时钟 | 系统时钟 |
| 引脚 | 若 PA10 被拉低，则将 PA10 和 PA9 分别配置为 USART1_RX 和 USART1_TX。<br>若 PA15 被拉低，则将 PA15 和 PA14 分别配置为 USART1_RX 和 USART1_TX。 |
| 波特率 | 芯片根据同步信号 0x7F 计算波特率。<br>支持波特率 1200 ~ 115200 bps。 |
| 数据长度 | 8 比特 |
| 校验方式 | 偶校验 |
| 停止位 | 1 比特 |
| 发送数据 | 0x79（ACK） |

## 3.3 命令阶段

同步流程完成后芯片将进入等待命令状态，接收主机端的命令并执行。若芯片未产生复位，将一直停留在该阶段。自举程序支持的命令参见表 3-3。

表 3-3 自举程序支持的命令

| 序号 | 命令 | 主机操作 | | | 芯片端操作 |
|---|---|---|---|---|---|
| | | 主机端发送命令 | 第二步，主机端下发数据 | 第三步，主机端下发数据 | |
| 1 | 命令列表请求 | 0x00 | 无 | 无 | 发送支持的命令列表。 |
| 2 | 版本号请求 | 0x01 | 无 | 无 | 发送自举程序版本号。 |

| 序号 | 命令 | 主机操作 | | | 芯片端操作 |
|---|---|---|---|---|---|
| | | 主机端发送命令 | 第二步，主机端下发数据 | 第三步，主机端下发数据 | |
| 3 | 芯片 ID 请求 | 0x02 | 无 | 无 | 发送芯片 ID。 |
| 4 | 读内存 | 0x11 | 起始地址 | 数据长度 | 发送指定 Flash 或 SRAM 的数据。 |
| 5 | 运行程序 | 0x21 | 被执行程序的起始地址 | 无 | 从下发的起始地址执行固件（将该地址的第一个字赋给主栈指针 MSP，第二个字赋给 PC 并执行）。<br>**注意：**<br>Cortex®-M0 不支持中断向量表重定位，采用该命令执行依赖中断的固件将导致不可预知的结果。 |
| 6 | 写内存 | 0x31 | 起始地址 | 长度及数据 | 将主机端下发的数据写入 Flash 或 SRAM。 |
| 7 | 擦除内存 | 0x44 | 需要擦除的页数及页号（全局擦除发送 0xFF FF） | 无 | 擦除指定的 Flash 页（页长度为 1K 字节）或全局擦除。 |
| 8 | 使能写保护 | 0x63 | 需要保护的扇区号 | 无 | 修改用户选项字节（User Option Bytes），保护 Flash 指定的扇区。 |
| 9 | 取消写保护 | 0x73 | 无 | 无 | 修改用户选项字节，取消写保护。 |
| 10 | 使能读保护 | 0x82 | 无 | 无 | 修改用户选项字节，使能读保护（level 1）。 |
| 11 | 取消读保护 | 0x92 | 无 | 无 | 修改用户选项字节，取消读保护。 |

为保障通信的可靠性，主机端发送的所有信息都必须附带校验码（1 字节）。

**提示：**

芯片端回复的信息一律不带校验码。

校验码的计算方法为：

- 若发送数据只有 1 个字节，则校验码为该数据的反码。例如，发送“写内存（0x31）”命令，则主机端需发送“0x31 0xCE”。
- 若发送数据超过 1 个字节，则校验码为所有字节的连续异或。例如，发送“写内存”的起始地址 0x08001000，则主机端需发送 5 个字节：0x08 0x00 0x10 0x00 0x18，其中前 4 个字节为地址信息，第 5 个字节为校验码。

芯片端接收到数据后，一律回复主机端：

- 正确接收或命令执行完毕回复“0x79”（ACK）。例如，执行“使能读保护”命令，芯片端接收到“0x82 0x7D”后，将立即回复“0x79”（ACK）表示正确接收命令；“使能读保护”完成后再次回复“0x79”（ACK），表示命令执行完毕。
- 校验码或其它错误回复“0x1F”（NACK）。

# 4 自举程序命令说明

本节主要描述自举程序各命令的功能、代码以及操作交互流程等。

## 4.1 命令列表请求（0x00）

“命令列表请求”命令，用于获取自举程序的协议版本号及所有支持的命令。

**命令代码：**0x00

**校验码：**0xFF

**操作交互流程：**

主机端发送“0x00 0xFF”（命令及校验码），芯片端回复 15 字节数据。各字节的具体描述参见表 4-1。

表 4-1 命令列表请求执行结果及含义

| 字节序号 | 值 | 含义 |
|---|---|---|
| 1 | 0x79 | ACK |
| 2 | 0x0B | 后续数据部分长度（不含 ACK）为 12 个字节（即 0x0B + 1） |
| 3 | 0x31 | 协议版本号 |
| 4 | 0x00 | 命令列表请求 |
| 5 | 0x01 | 版本号请求 |
| 6 | 0x02 | 芯片 ID 请求 |
| 7 | 0x11 | 读内存 |
| 8 | 0x21 | 运行程序 |
| 9 | 0x31 | 写内存 |
| 10 | 0x44 | 擦除内存 |
| 11 | 0x63 | 使能写保护 |
| 12 | 0x73 | 取消写保护 |
| 13 | 0x82 | 使能读保护 |
| 14 | 0x92 | 取消读保护 |
| 15 | 0x79 | ACK |

自举程序执行“命令列表请求”命令的详细流程，如图 4-1 所示。

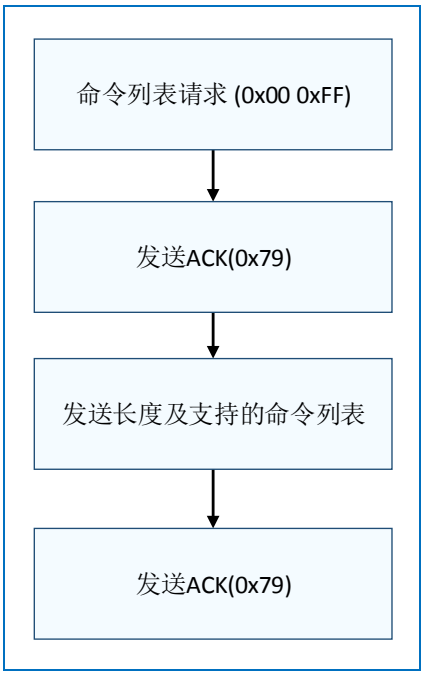


图 4-1 自举程序命令列表请求命令执行流程

## 4.2 版本号请求（0x01）

“版本号请求”命令，用于获取自举程序的协议版本号以及软件版本号。

**命令代码：**0x01

**校验码：**0xFE

**操作交互流程：**

主机端发送“0x01 0xFE”（命令及校验码）；芯片端给主机端回复 5 字节数据。各字节的具体描述参见表 4-2。

表 4-2 版本号请求执行结果及含义

| 字节序号 | 值 | 含义 |
|---|---|---|
| 1 | 0x79 | ACK |
| 2-4 | 0x31 0xXX 0xXX | 协议版本号 3.1，软件版本号 XX.XX |
| 5 | 0x79 | ACK |

自举程序执行“版本号请求”命令的详细流程，如图 4-2 所示。

 

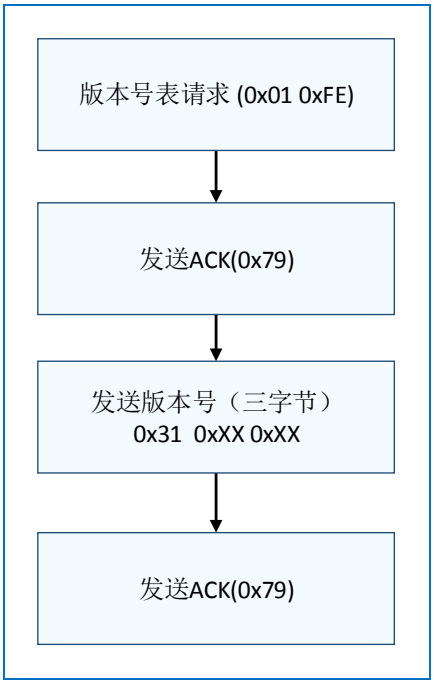


图 4-2 自举程序版本号请求命令执行流程

## 4.3 芯片 ID 请求（0x02）

“芯片 ID 请求”命令，用于获取芯片的 ID。

**命令代码：**0x02

**校验码：**0xFD

**操作交互流程：**

主机端发送“0x02 0xFD”（命令及校验码）；芯片端回复 5 字节数据。各字节的具体描述参见表 4-3。

表 4-3 芯片 ID 请求执行结果及含义

| 字节序号 | 值 | 含义 |
|---|---|---|
| 1 | 0x79 | ACK |
| 2 | 0x01 | 长度为两个字节（1+1） |
| 3-4 | 0xXX 0xXX | 芯片 ID，长度为两个字节 |
| 5 | 0x79 | ACK |

自举程序执行“芯片 ID 请求”命令的详细流程，如图 4-3 所示。

 

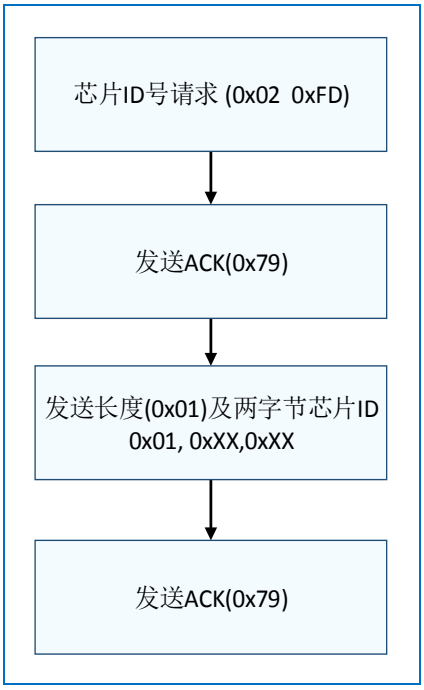


图 4-3 自举程序芯片 ID 号请求命令执行流程

## 4.4 读内存（0x11）

“读内存”命令，用于读取不超过 256 字节的内存数据。

可读取的内存区域包括：

- SRAM 区域（除去自举程序代码占用的空间）：0x2000 0800 ~ SRAM 结束地址。
- 用户选项字节：0x1FFF F800 ~ 0x1FFF F81F。
- Flash 区域：0x0800 0000 ~ Flash 结束地址。

**命令代码：**0x11

**校验码：**0xEE

**操作交互流程：**

1. 主机端发送“0x11 0xEE”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 主机端发送读取区域起始地址以及校验码（例如，若起始地址为“0x08000010”，则发送“0x08 0x00 0x00 0x10 0x18”）；芯片端返回“0x79”（ACK）。
3. 主机端发送数据长度 Length 及校验码（例如，数据长度为 256 字节，则发送 Length 为 0xFF，以此类推）；芯片端返回“0x79”（ACK）和读取的 Length+1 字节的数据。

对于异常情况，自举程序处理方式如下：

- 芯片“读保护”被激活时，自举程序将返回“0x1F”（NACK）。
- 起始地址不在自举程序可读的内存区域（如寄存器地址）时，自举程序将返回“0x1F”（NACK）。
- 起始地址或长度校验错误，自举程序将返回“0x1F”（NACK）。

 

- 起始地址有效但读取长度越界时，自举程序只返回有效地址空间的数据（例如 SRAM 结束地址为 0x20000FFF，则对于以 0x20000FF0 为起始地址读取长度为 256 字节的命令，自举程序只返回 16 字节的内容）。

自举程序执行“读内存”命令详细流程，如图 4-4 所示。

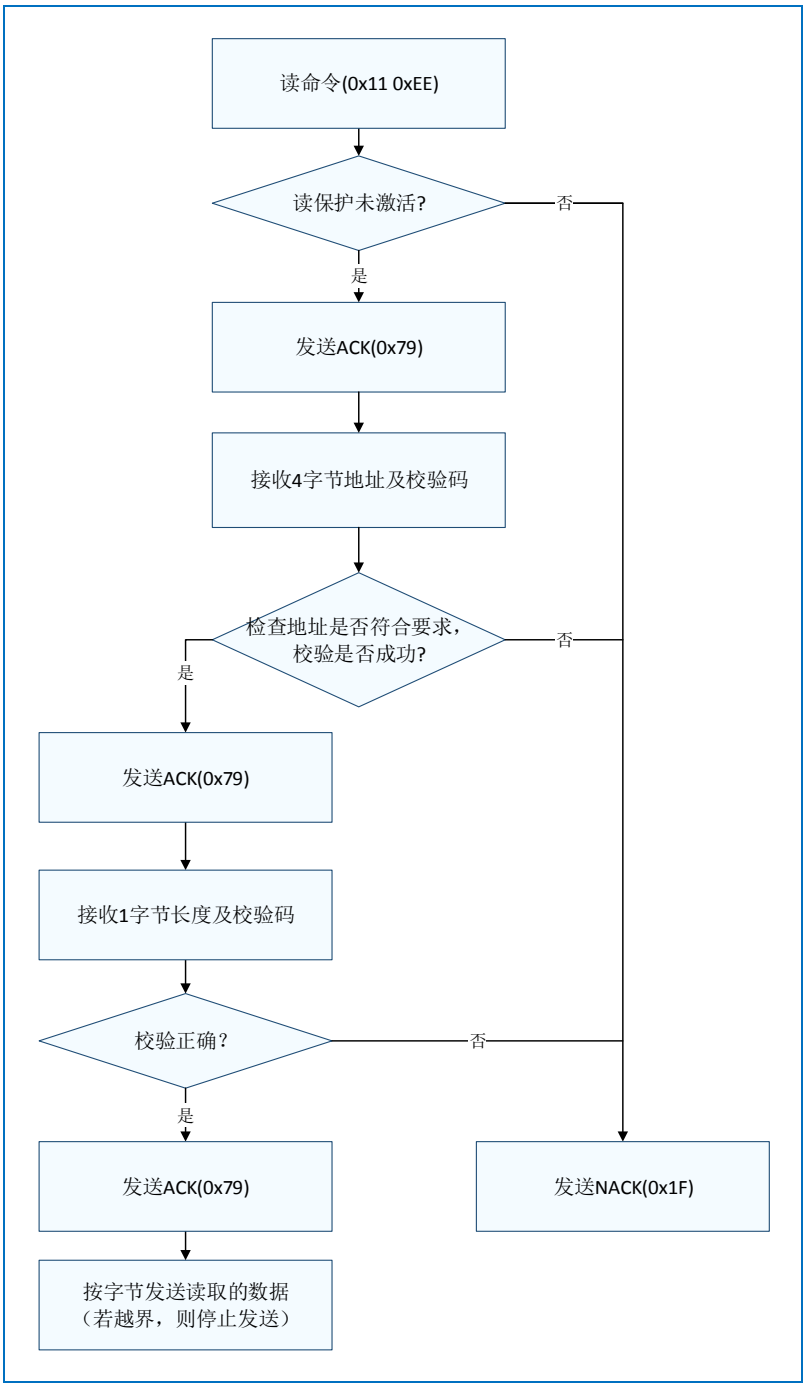


图 4-4 自举程序读内存命令执行流程

## 4.5 运行程序（0x21）

“运行程序”命令，用于执行指定地址的二进制程序。

**命令代码：**0x21

**校验码：**0xDE

**操作交互流程：**

1. 主机端发送“0x21 0xDE”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 主机端发送程序的起始地址以及校验码；芯片端返回“0x79”（ACK）。
3. 对于正确的运行程序命令，芯片端将执行以下操作（假设主机端发送的地址为 ADDR）：
   (1) 关闭所有中断。
   (2) 将 ADDR 地址的 32 位的字（WORD）拷贝至主栈指针寄存器（MSP）。
   (3) 将 ADDR+4 地址的 32 位的字（WORD）拷贝至 R15（PC）。
4. 成功执行该命令后将跳出自举程序，无法执行其它命令。

对于以下异常情况，自举程序将返回“0x1F”（NACK）：

- 芯片“读保护”被激活。
- 发送的地址不在有效地址范围内。
- 校验错误。

**提示：**

GMF03x 芯片不支持中断向量表重定位，若运行的程序依赖中断将会导致不可预知的结果。

自举程序执行“运行程序”命令的详细流程，如图 4-5 所示。

 

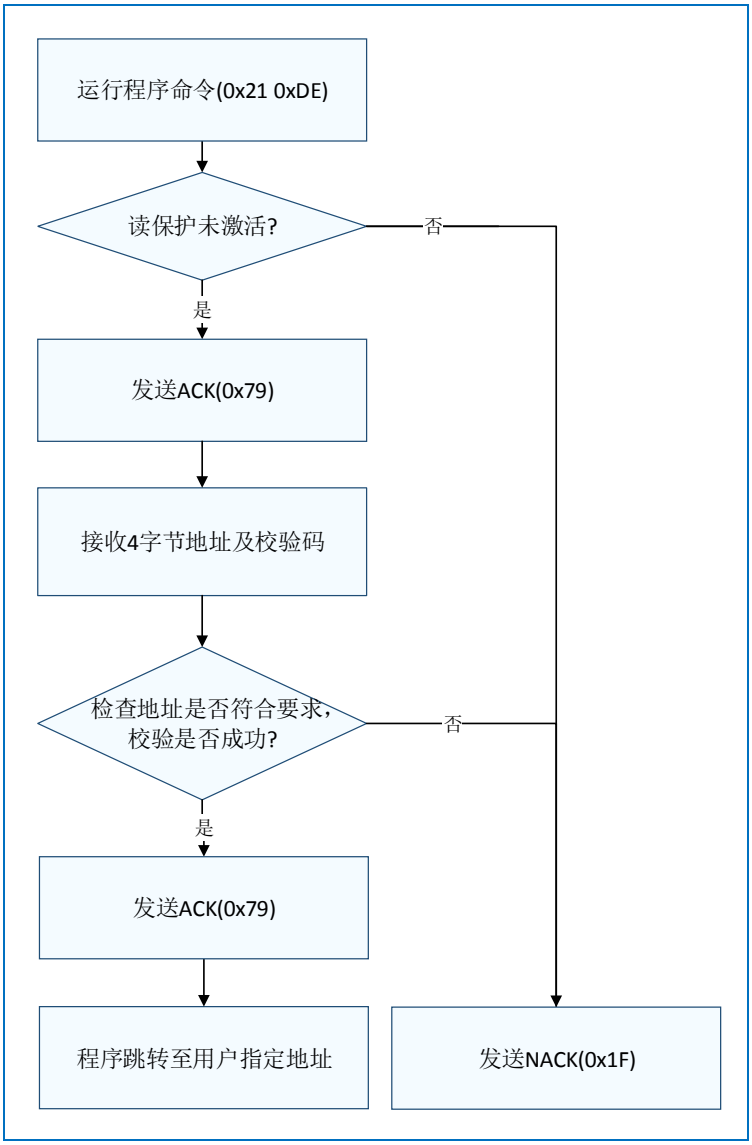


图 4-5 自举程序运行程序命令执行流程

## 4.6 写内存（0x31）

“写内存”命令，用于向内存写入不超过 256 字节的数据。可写入的区域包括：

- SRAM 区域（除去自举程序代码占用的空间）：0x2000 0800 ~ SRAM 结束地址。
- 用户选项字节：0x1FFF F800 ~ 0x1FFF F81F（成功执行“写命令”后，自举程序将自动复位，以重新加载用户选项字节）。
- Flash 区域：0x0800 0000 ~ Flash 结束地址（自举程序不会检查 Flash 区域是否处于“擦除”或“写保护”状态，也不会查检 Flash 错误标志。执行“写命令”后，无论是否成功都不会返回任何错误信息，用户需执行“读命令”去验证是否写成功）。

**提示：**

- 向用户选项字节写入数据，芯片将自动复位。主机端必须重新发送“0x7F”执行同步流程以连接芯片。
- 对于任意内存写操作，建议起始地址按字（32 位）对齐，且写入数据长度应为 4 字节的整数倍（不足应填充 0xFF），避免总线或 Flash 访问错误导致芯片挂死。

**命令代码：0x31**

校验码：0xCE

操作交互流程：

1. 主机端发送“0x31 0xCE”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 主机端发送写入区域的起始地址以及校验码(例如，起始地址为0x08000010，则发送0x08 0x00 0x00 0x10 0x18)；芯片端返回“0x79”（ACK）。
3. 主机端发送数据长度 Length、数据包（Length+1 字节）以及 1 字节的校验码（Length+2 字节的连续异或）；芯片端返回“0x79”（ACK）。

对于异常情况，自举程序处理方式如下：

- 芯片“读保护”被激活时，自举程序将返回“0x1F”（NACK）。
- 起始地址不在有效地址范围内（如寄存器地址）时，自举程序将返回“0x1F”（NACK）。
- 起始地址或写入长度校验错误时，自举程序将返回“0x1F”（NACK）。
- 起始地址有效但写入长度越界时，自举程序只执行有效地址空间的数据写操作（例如 SRAM 结束地址为 0x20000FFF，则对于以 0x20000FF0 为起始地址写入长度为 256 字节的命令，自举程序将只写 16 字节），且不再发送 ACK。

自举程序执行“写内存”命令的详细流程，如图 4-6 所示。

 

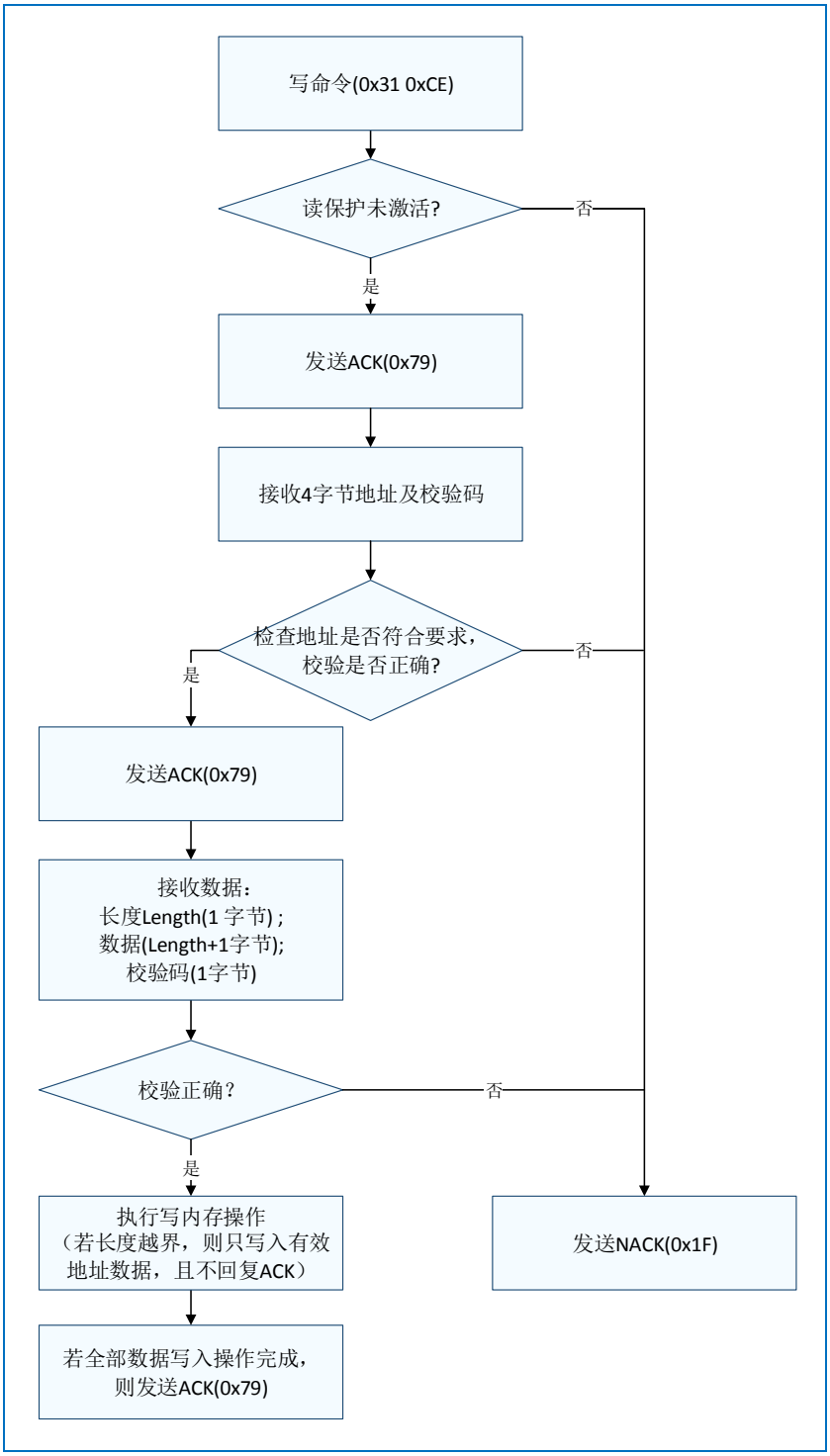


图 4-6 自举程序写内存命令执行流程

## 4.7 擦除内存（0x44）

“擦除内存”命令，用于擦除 Flash（支持全局擦除和页擦除）。

**命令代码：**0x44

**校验码：**0xBB

**操作交互流程：**

- 全局擦除

 

1. 主机端发送“0x44 0xBB”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 主机端发送“0xFF 0xFF 0x00”（其中“0xFF 0xFF”表示全局擦除，“0x00”为校验码），芯片端返回“0x79”（ACK）。
3. 擦除完成后，芯片端返回“0x79”（ACK）。

- 页擦除

1. 主机端发送“0x44 0xBB”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 主机端发送待擦除页数 N（实际擦除页数为 N+1），用两个字节表示（例如，若需擦除两页，则发送 0x00 0x01）。
3. 主机端发送页的序号，每页用两个字节表示（例如，若需擦除页 1 和页 8，则发送 0x00 0x01 0x00 0x08）。
4. 主机端发送“2”~“3”步数据的校验码（1 字节）；芯片端返回“0x79”（ACK）。
5. 擦除完成后，芯片端返回“0x79”（ACK）。

对于异常情况，自举程序处理方式如下：

- 芯片“读保护”被激活时，自举程序将返回“0x1F”（NACK）。
- 校验错误时，自举程序将返回“0x1F”（NACK）。
- 页码越界时，自举程序将输入页码与芯片总页数求余的结果作为需要擦除的页码，例如芯片的总页数为 32 页，则擦除第 32、33 页的指令将导致自举程序擦除第 0、1 页。

自举程序执行“擦除内存”命令的详细流程，如图 4-7 所示。

 

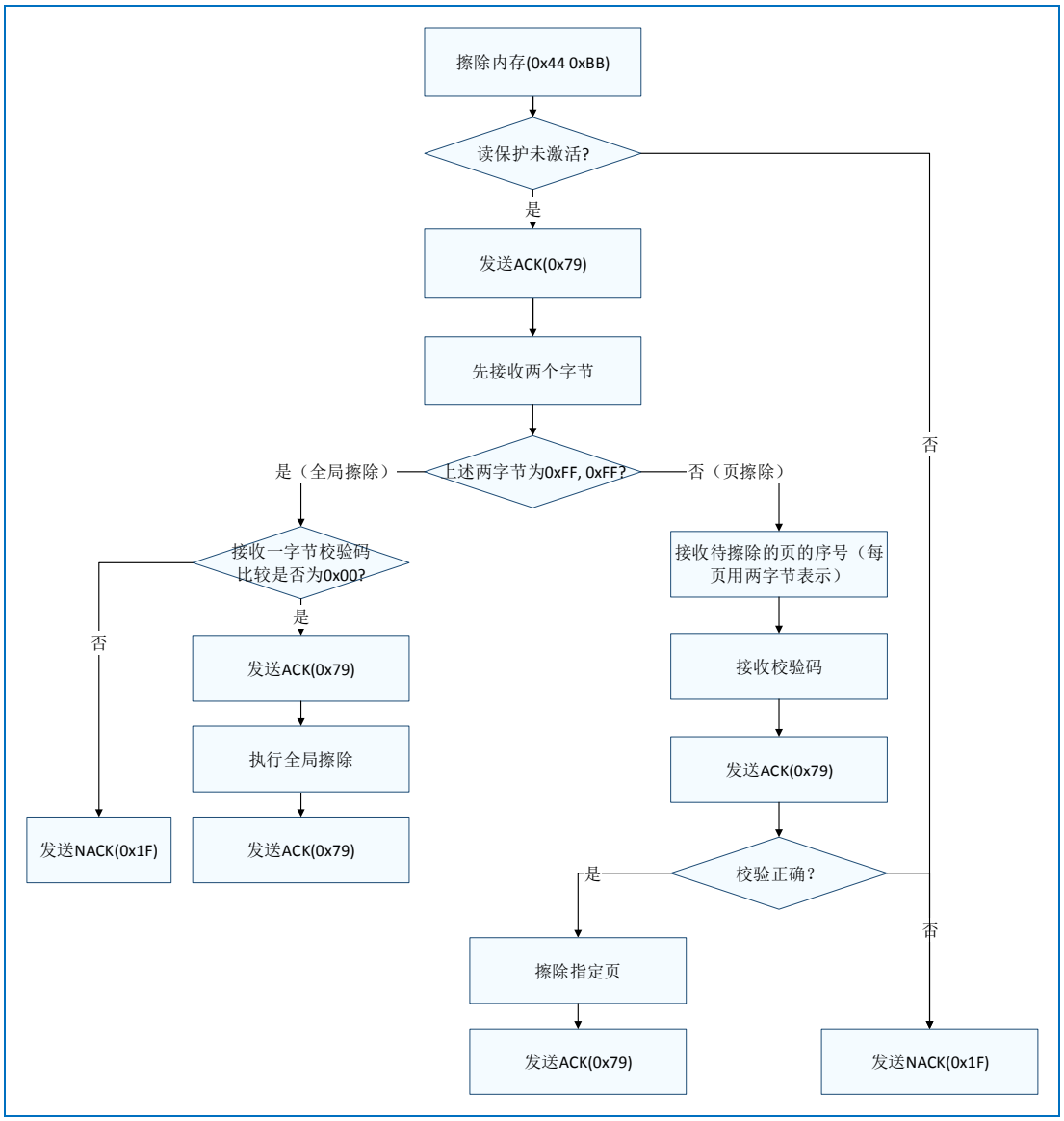


图 4-7 自举程序擦除内存命令执行流程

## 4.8 使能写保护（0x63）

“使能写保护”命令，用于按扇区（Sector）对 Flash 进行写保护操作，一个扇区为 4K 字节。

**命令代码：**0x63

**校验码：**0x9C

**操作交互流程：**

1. 主机端发送“0x63 0x9C”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 主机端发送待保护的扇区数 N（1 字节，实际保护的扇区数为 N+1）。
3. 主机端发送待保护的 N+1 个扇区的序号（每个扇区 1 个字节）。
4. 主机端发送上述 N+2 个字节的校验码（1 字节）；芯片端返回“0x79”（ACK）。
5. 使能写保护完成后，芯片端再次返回“0x79”（ACK），并复位芯片使保护生效。

对于异常情况，自举程序处理方式如下：

- 芯片“读保护”被激活时，自举程序将返回“0x1F”（NACK）。

- 校验错误时，自举程序将返回“0x1F”（NACK）。
- 扇区号越界时，自举程序将取扇区号的最低 4 个 bit，修改至 WPROT0 后的保留区域。

**提示：**

执行该命令后自举程序将自动复位，主机端需重新执行同步流程以连接芯片。

自举程序执行“使能写保护”命令的详细流程，如图 4-8 所示。

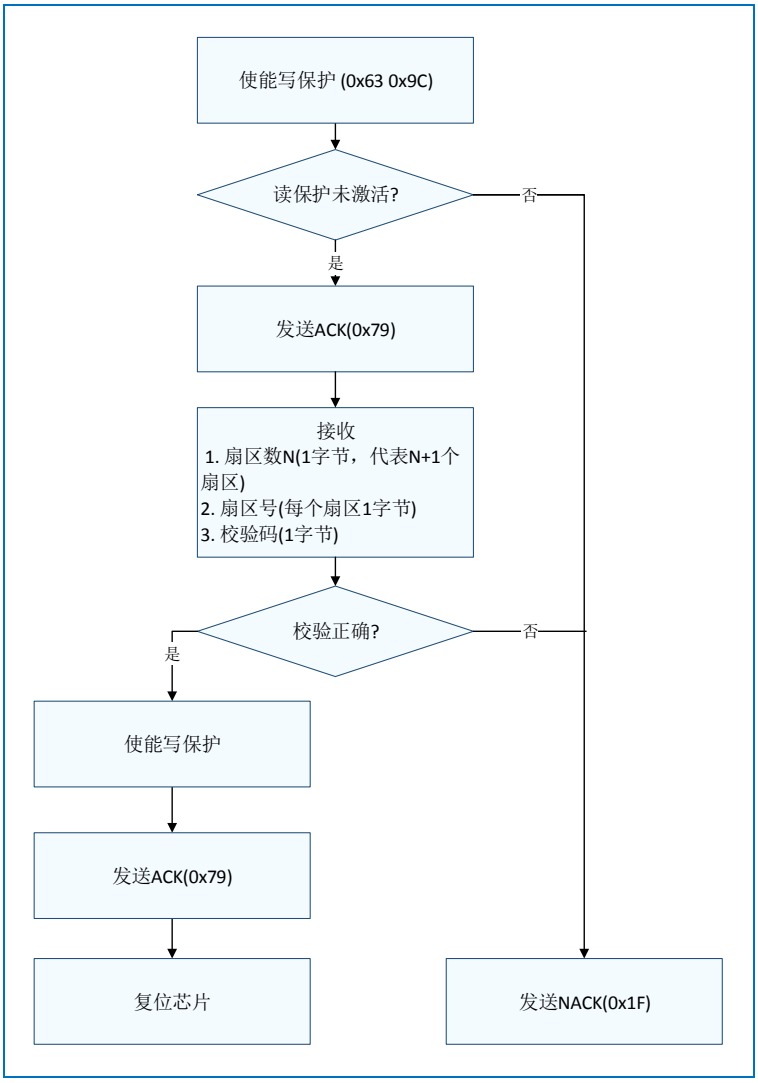


图 4-8 自举程序写保护命令执行流程

**提示：**

多次对芯片进行写保护操作，仅最后一次有效。

## 4.9 取消写保护（0x73）

“取消写保护”命令，用于取消所有扇区的“写保护”。

**命令代码：**0x73

**校验码：**0x8C

**操作交互流程：**

1. 主机端发送“0x73 0x8C”（命令及校验码）；芯片端返回“0x79”（ACK）。

2. 操作完成后，芯片端再次返回“0x79”（ACK），并复位芯片使“取消写保护”生效。

对于以下异常情况，自举程序将返回“0x1F”（NACK）：

芯片“读保护”被激活。

**提示：**

执行该命令后自举程序将自动复位，主机端需重新执行同步流程以连接芯片。

自举程序执行“取消写保护”命令的详细流程，如图 4-9 所示。

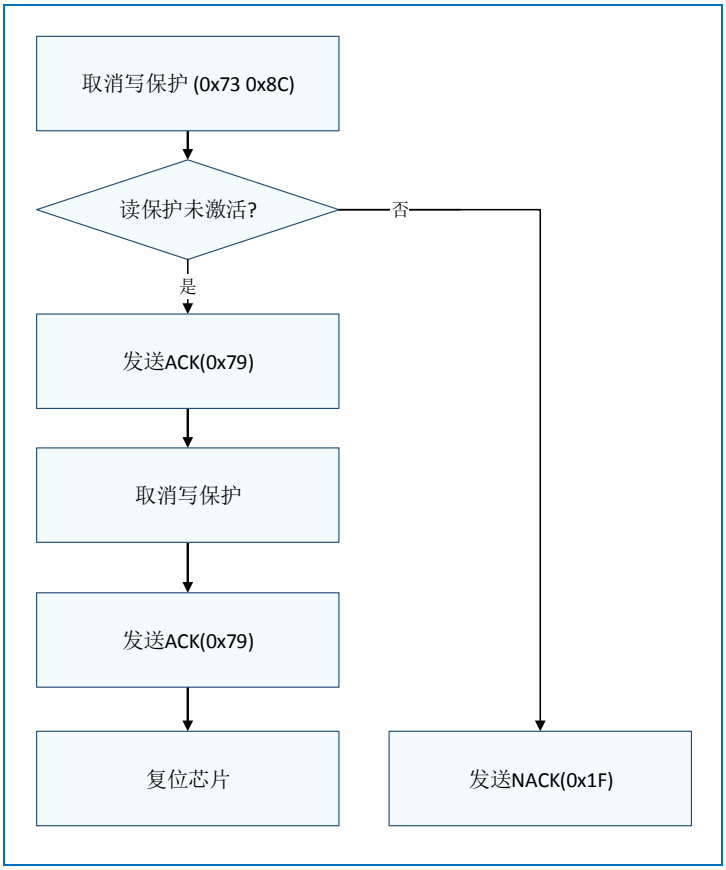


图 4-9 自举程序取消写保护命令的执行流程

## 4.10 使能读保护（0x82）

“使能读保护”命令，用于使能 Flash 的读保护功能，使主机端无法通过自举程序、调试接口从 Flash 中读取内容。

**命令代码：**0x82

**校验码：**0x7D

**操作交互流程：**

1. 主机端发送“0x82 0x7D”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 芯片端修改用户选项字节 RPROT 为“0xBB”，使能读保护（Level1）。
3. 操作完成后，芯片端再次返回“0x79”（ACK），并复位芯片使读保护生效。

对于以下异常情况，自举程序将返回“0x1F”（NACK）：

 

芯片“读保护”被激活。

**提示：**

执行该命令后自举程序将自动复位，主机端需重新执行同步流程以连接芯片。

自举程序执行“使能读保护”命令流程，如图 4-10 所示。

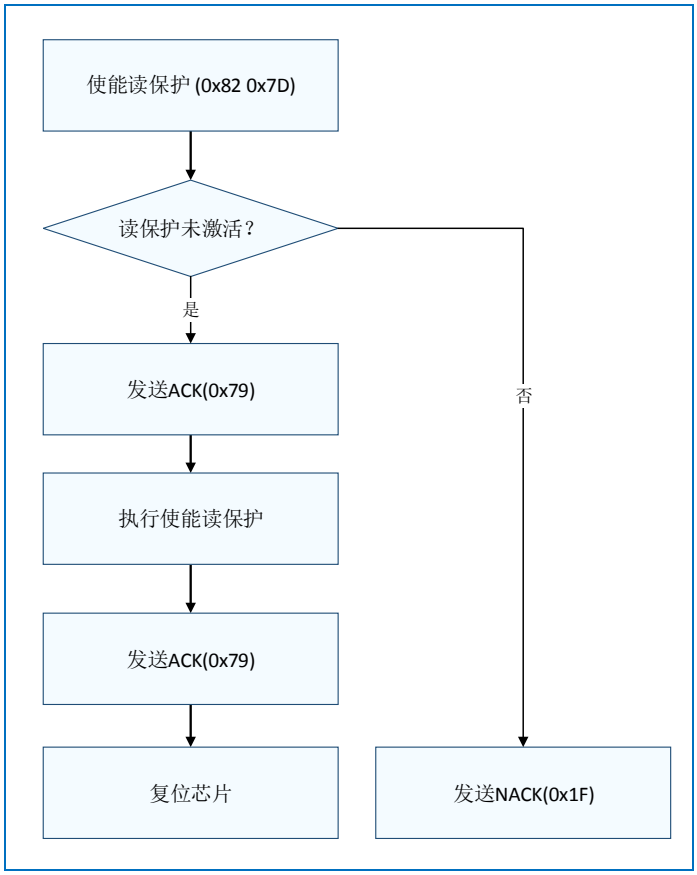


图 4-10 自举程序使能读保护命令执行流程

## 4.11 取消读保护（0x92）

“取消读保护”命令，用于取消 Flash 的读保护功能。若芯片处于“读保护”状态，命令将导致 Flash 中所有内容被擦除。

**命令代码：**0x92

**校验码：**0x6D

**操作交互流程：**

1. 主机端发送“0x92 0x6D”（命令及校验码）；芯片端返回“0x79”（ACK）。
2. 操作完成后，芯片端再次返回“0x79”（ACK），并复位芯片使“取消读保护”生效。

**提示：**

执行该命令后自举程序会自动复位，主机端需重新执行同步流程以连接芯片。

完整的取消读保护流程如图 4-11 所示。

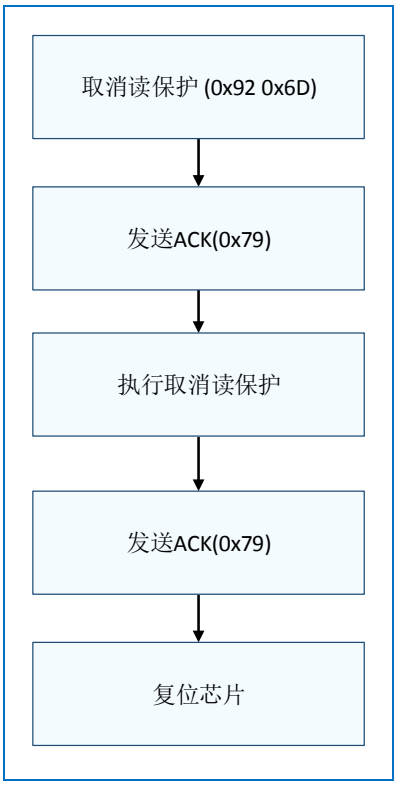


图 4-11 自举程序取消读保护命令执行流程

# 5 自举程序时间参数

自举程序时间参数包括：上电时间与连接时间。

表 5-1 自举程序时间参数

| 参数名称 | 描述 | 值 |
| --- | --- | --- |
| 上电时间 | 从芯片上电到自举程序可接收同步信号（0x7F）的时间 | 3.0 ms[1] |
| 连接时间 | 从主机完成同步信号的发送到接收 ACK 的时间 | 0.16 ms[1] |

[1] 设计参考值，未经生产测试。