AN-32016

SPI 协议的 ISP 使用说明

Rev1.00
http://www.touchmcu.com

文档修改历史

日期	版本	描述
2022-01-28	1.00	初版

目录

表目录 / List of Figures

图目录 / List of Tables

1. 文档简介

本文档主要描述了基于 FT32F0XX 的 SPI 编程（ISP）协议，是编写 SPI 协议 ISP 烧录主机的参考文档。其内容包含从机 ISP 硬件配置，SPI 通信协议、详细的编程指令说明。

2. 硬件配置

进行 ISP 烧录操作时，从机需保证以下的硬件配置，保证芯片复位上电后，系统能够从芯片内部的系统存储区启动，进入 ISP 引导烧录模式。

- ISP 烧录口：NSS（PA4）、CLK（PA5）、MISO（PA6）、MOSI（PA7）。
- SPI 模式：模式 1
- BOOT0 引脚：上拉为高电平。
- 系统选项字中：nBOOT1 = 1。
- 通讯速率：最大 8M。
- BootLoader 使用的SRAM: 6 KByte。

3. SPI 协议的 ISP 编码序列

当从机从系统存储区启动时，存储于此引导区的代码将会不断地扫描CLK 和MOSI，等待检测到 SPI 的同步指令，当同步指令正确时，回复 ACK，然后从机进入等待主机 SPI 指令模式。

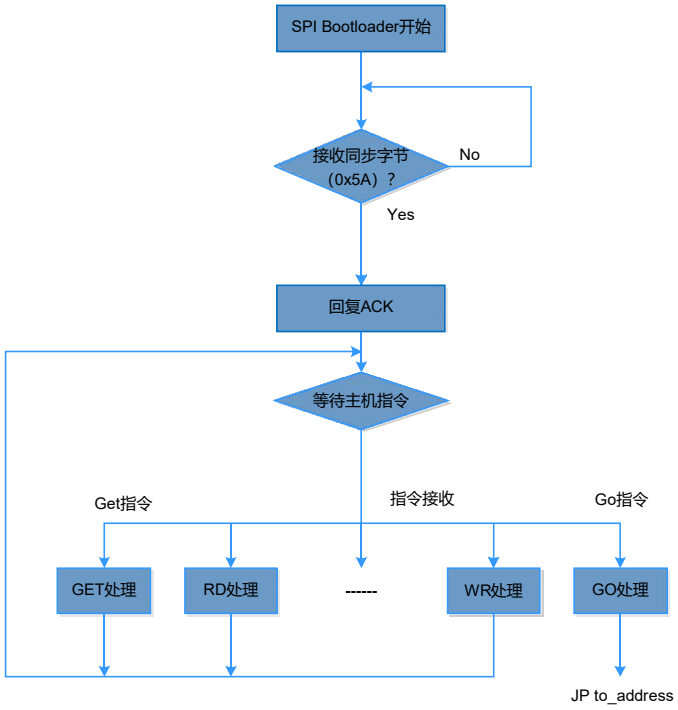

图 3-1 编码序列

如下图所示，主机先发送同步指令 0x5A，主机收到 0xA5 后，主机等待收取ACK。序列正确后，FT32 进入到SPI 指令模式。

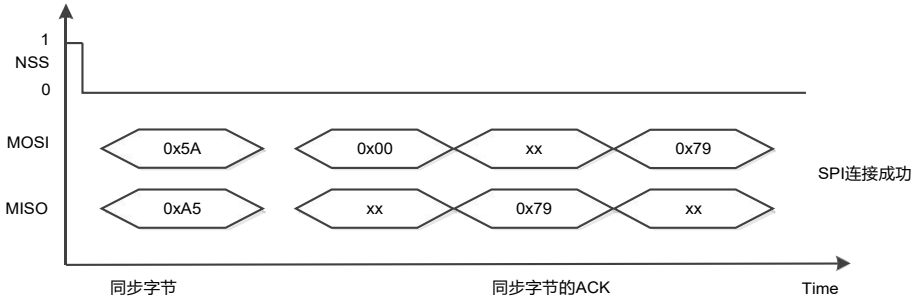

图 3-2 获取 ACK 流程 1

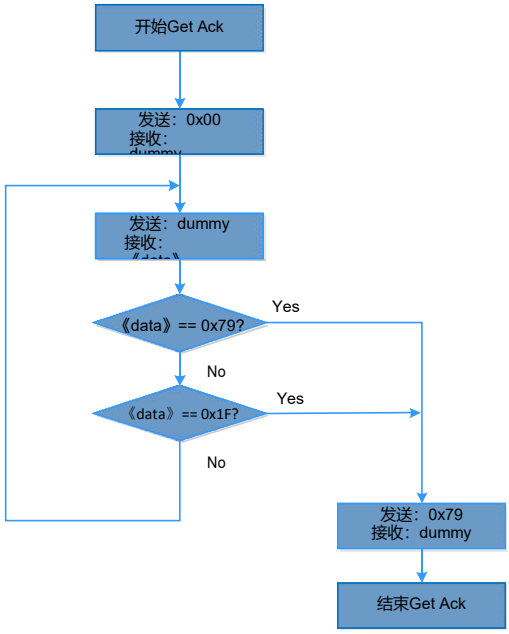

图 3-3 获取 ACK 流程 2

指令模式下，主机需要先发送起始指令 0x5A，然后紧接着发送指令代码及其 XOR 码，从机同时回复数据，流程如下。

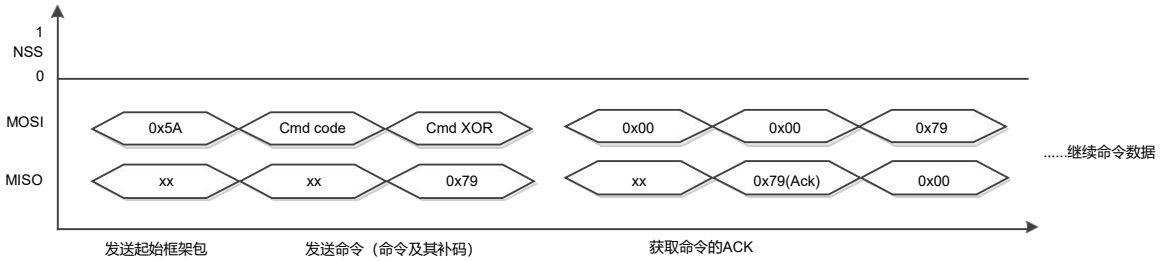

图 3-4 指令序列流程

主机读取从机发送的任意数据之前，需要发送一个 dummy 字节，然后开始接收从机发送的数据，流程如下。

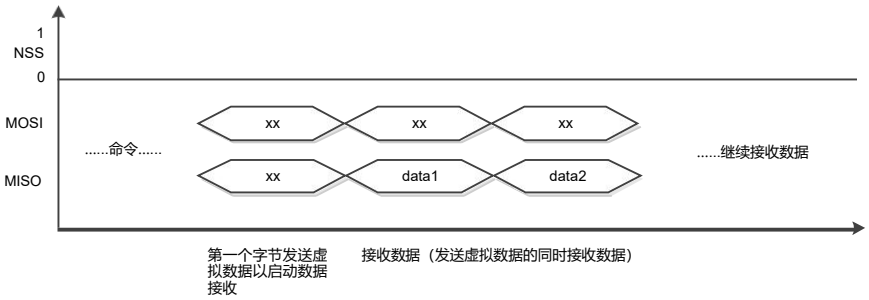

图 3-5 读数据序列流程

4. BootLoader 指令集

指令名称	指令码	指令描述
Get	0x00	读取Bootloader版本及其当前版本所支持的指令集
Get Version	0x01	读取Bootloader版本
Get ID	0x02	读取芯片识别PID
Read Memory	0x11	读取存储区指定地址的内容，最大支持256bytes
Go	0X21	跳转至位于main flash或SRAM中指定的地址区域，并从此处开始运行存储的用户代码
Write Memory	0x31	向flash或sram中指定的地址区域写入数据，最大支持256bytes
Erase	0x44	将此指令后的相应数量及地址所对应的flash page进行擦除，1次指令可指定擦除1至127个page（每个page 1kB）。
Write Protect	0x63	使能指定flash扇区的写保护功能，F072共32个扇区，每扇区4k大小
Write Unprotect	0x73	失能芯片所有扇区的写保护功能
Read Protect	0x82	使能读保护功能
Read Unprotect	0x92	失能芯片的读保护功能
Get Checksum	0xA1	在一个大小为4字节倍数的给定内存区域上计算CRC值

表 4-1 BootLoader 指令集

注意：

- 如果收到一个被拒绝的指令或在指令执行过程中发生错误，从机发送一个 NACK 字节并回到指令检查。

- 当 RDP(读保护)选项被激活时，只有 Get、Get Version、Get ID、Read Unprotect 支持， 解除 RDP 之后，其他指令重新激活。

- 由于 SPI 被配置为全双工，每次主机在 MOSI 上传输数据，它同时接收 MISO 线上的数据。 因为从机的应答不是立即的，当主机发送时，接收到的数据立即被忽略(dummy，这些数据不被主机使用)。

- 当从机发送数据时，主机发送它的时钟，所以它必须发送数据，使 MOSI 线能够接收 MISO 线上的从数据。在这种情况下，主机必须总是发送 0x00 (从机未使用的数据)。

4.1. 通信安全

从编程主程序到从程序的所有通信都用下面的方法进行校验。

- Checksum:接收的数据字节块被异或处理。所有字节的异或被添加到每次通信的末尾(校验和字节)。通过异或处理所有接收到的字节，数据加上校验和，数据包的最后结果必须是 0x00。

- 如果接收到的数据是一个字节，那么它的校验和是该值的异或(例如，0x02 的校验和为 0xFD)。
- 对于每个命令，主机发送三个字节:一个字节表示帧的开始(SOF = 0x5A)，一个字节表示命令和一个字节表示命令的补码(命令和命令的 XOR = 0x00)。
- 每个数据包要么被接受(ACK 回答)，要么被丢弃(NACK 回答)。

—ACK = 0x79

—NACK = 0x1F

主机框架可以是以下内容之一。

- 发送命令帧:主机作为主发送器发起通信向从机发送两个字节:命令代码加异或。
- 等待 ACK/NACK 帧:主机作为主发送器发起一个SPI 通信从从机接收一个字节:ACK 或 NACK。
- 接收数据帧: 主机作为主接收器发起 SPI 通信从从机接收响应。接收的字节数取决于命令。
- 发送数据帧: 主机作为主发送器发起 SPI 通信向从机发送所需的字节。发送的字节数取决于命令。

4.2. GET 指令（0x00）

Get 指令用于用户读取BootLoader 版本及其所支持的指令集，当BootLoader 接收到 Get 指令，它将BootLoader 程序版本和所支持的指令代码发送给主机。BootLoader 返回的数据长度是 14bytes (接收字节 + BootLoader 版本 + 支持的指令列表。主机端 Get 指令处理如下：

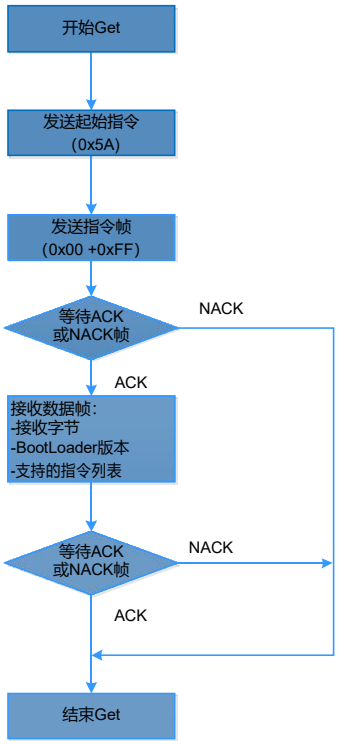

图 4-1 GET 指令流程

主机接收到的数据帧：

Byte1: 0x0C

Byte2: 0x10

Byte3~14: 0x00 0x01 0x02 0x11 0x21 0x31 0x44 0x63 0x73 0x82 0x92 0xA1

4.3. Get Version（0x01）

Get GV 指令用于用户读取 BootLoader 版本，当BootLoader 接收到此指令，它会向主机回发此信息项。BootLoader 返回的数据长度为 1 个字节 (BootLoader 版本)。主机端指令处理如下。

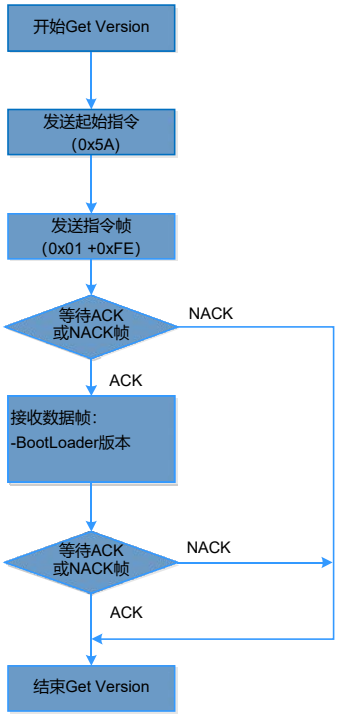

图 4-2 Get version 指令流程

主机接收到的数据帧:

Byte1: 0x10

4.4. Get ID 指令（0x02）

Get ID 指令用于用户读取芯片的内置 ID，当 BootLoader 接收到此指令，它会向主机回发此信息项。BootLoader 返回的数据长度为 3 个字节 (N + PID，N = 字节数-1，PID 由两个字节组成，MSB 在前，LSB 在后)。主机端指令处理如下：

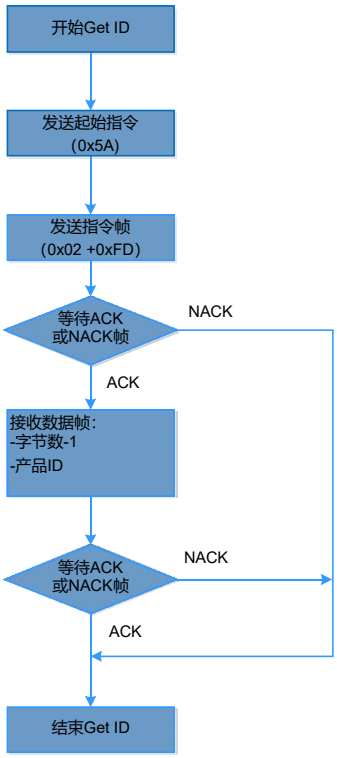

图 4-3 Get ID 指令流程

主机接收到的数据帧:

Byte1: 0x01

Byte2~3: 0x04 0x48

4.5. Read Memory 指令（0x11）

Read Memory 指令用于用户读取芯片内指定的有效存储区，包括main flash，选项字区以及SRAM。如下所示：

名称	地址范围	大小
Main Flash	0x08000000~0x0801FFFF	128kbytes
User Option	0x1FFFF800~0x1FFFF813	20bytes
SRAM	0x20000000~0x20005FFF	24kbytes

表 4-2 Read Memory 有效存储区

当 Bootloader 接收到此指令后，会回复主机 ACK，并进入等待主机发送待读取地址状态，地址为4字节，大端存储，地址后面跟随 1 字节的地址异或校验和 (即地址格式为 4Bytes Address + 1Bytes XOR checksum = 5Bytes)。当 Bootloader 接收到地址信息后，进行校验及地址有效性分析，分析后，回复主机ACK（地址校验通过且合理）或 NACK（地址校验未通过或不合理）。

当地址校验通过且合理，Bootloader 进入等待主机发送待读取字节数（N-1），以及字节数的取反校验数，1Bytes N-1 + 1Bytes N-1`s Checksum = 2Bytes （两者异或和应为 0xFF），当

Bootloader 校验 2 字节数据合理后，Bootloader 开始发送指定地址、指定数量（N）的内容到主机，若校验未通过，回复主机NACK 后结束此指令流程。

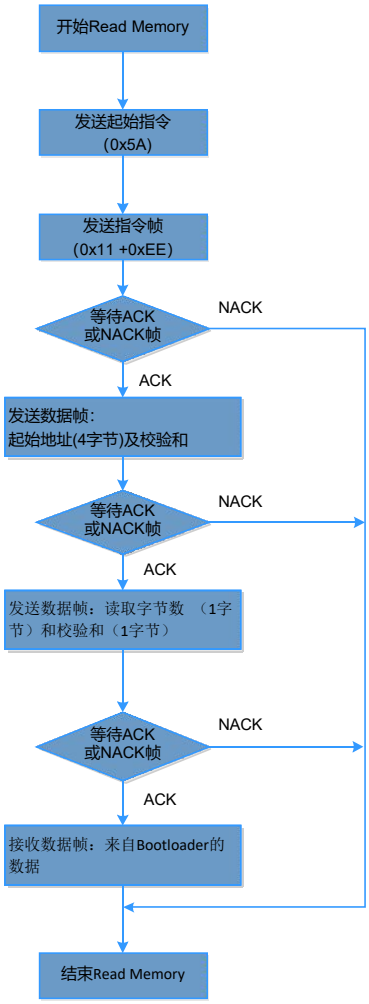

图 4-4 Read Memory 指令流程

4.6. Go 指令（0x21）

Go 指令用于跳转执行已经烧录的代码，或者执行指定地址区域的代码。（包括main flash 区域 0x08000000~0x0801FFFF 及 SRAM 区域 0x20000000~0x20005FFF）。

当 Bootloader 接收到此指令后，会回复主机 ACK，并进入等待主机发送待跳转地址状态，地址为4字节，大端存储，地址后面跟随1 字节的地址异或校验和（即地址格式为4Bytes Address + 1Bytes XOR checksum = 5Bytes）。

当 Bootloader 接收到地址信息后，进行校验及地址有效性分析，分析后，回复主机 ACK（地址校验通过且合理）或 NACK（地址校验未通过或不合理），当地址校验通过且合理，Bootloader 复位其所用的外设资源，并重置堆栈，跳转至指定程序地址运行程序。

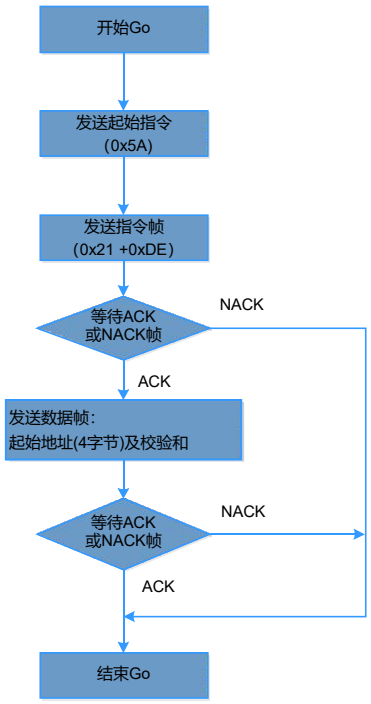

图 4-5 Go 指令流程

4.7. Write Memory 指令（0x31）

Write Memory 指令用于用户向芯片内指定的有效存储区（包括 main flash，选项字区以及 SRAM）写入数据。

名称	地址范围	大小
Main Flash	0x08000000~0x0801FFFF	128kbytes
User Option	0x1FFFF800~0x1FFFF813	20bytes
SRAM	0x20000000~0x20005FFF	24kbytes

表 4-3 Write Memory 有效存储区

注：对 Main Flash 及 UserOption 操作时，地址需按字对齐（即地址是 4 的倍数），写入数量应为 4 的倍数字节。

当 Bootloader 接收到此指令后，会回复主机 ACK，并进入等待主机发送待读取地址状态，地址为4字节，大端存储，地址后面跟随1 字节的地址异或校验和（即地址格式为4Bytes Address + 1Bytes XOR checksum = 5Bytes）。当 Bootloader 接收到地址信息后，进行校验及地址有效性分析，分析后，回复主机ACK（地址校验通过且合理）或 NACK（地址校验未通过或不合理）

当地址校验通过且合理，Bootloader 进入等待主机发送待写入的字节数（1byte 表示 N-1,一次烧录支持的最大字节数是 256），以及 N 个 Bytes 待写入数据，后面跟随 1byte 的异或校验和：

checksum = （N-1）^ data1 ^ data2…. dataN

当 Bootloader 校验烧录字节数及字节数据合理后，Bootloader 开始向指定地址区域写入数据，写入成功后回复主机ACK（烧写成功），若写入未成功，回复主机NACK 后结束此指令流程。

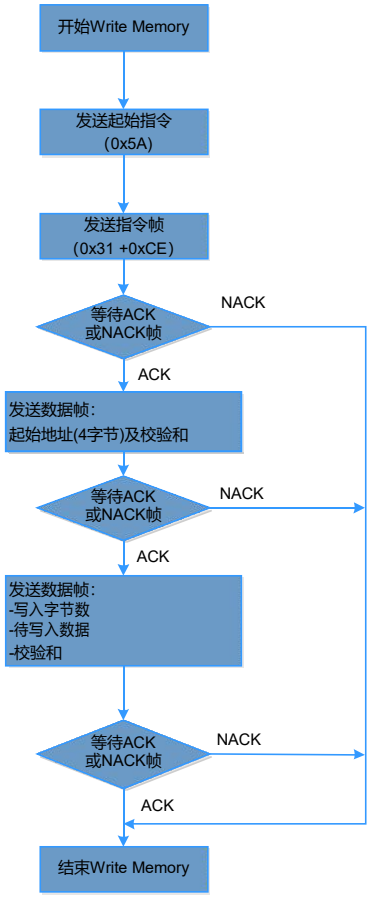

图 4-6 Write Memory 指令流程

4.8. Erase 指令（0x44）

Erase Memory 指令用于主机擦除芯片内全部或指定的有效存储区（main flash），其擦除最小单位为 1page（FT32F072XB 1page 为 1k bytes，总计 127 个 page）。

当 Bootloader 接收到此指令后，会回复主机 ACK，并进入等待主机发送待擦除的 page 数量状态，擦除的 page 数量(N-1)为 2 字节，大端存储，Bootloader 会根据接收的（N-1）page 数量进行如下流程。

- 若 N-1 = 0xFFFF，且紧随其后的 1Byte 校验码为 0x00， Bootloader 则会进行main flash 的整片擦除；

- 若 N-1 小于等于 127，Bootloader 会继续接收 N x 2 个字节的 page 号，其每两个字节表示待指定擦除的 page，而后 Bootloader 会再接收 1 字节的异或和校验码 checksum。

Checksum = 2byte（待擦除数量 N-1）^ N x 2 bytes （所有待擦除 page 号）

当 Bootloader 校验擦除 page 数及各个待擦除 page 号合理后，Bootloader 开始擦除指定数量指定号码的 page，擦除成功回复主机 ACK（擦除成功），未成功，回复主机NACK 后结束此指令流程。

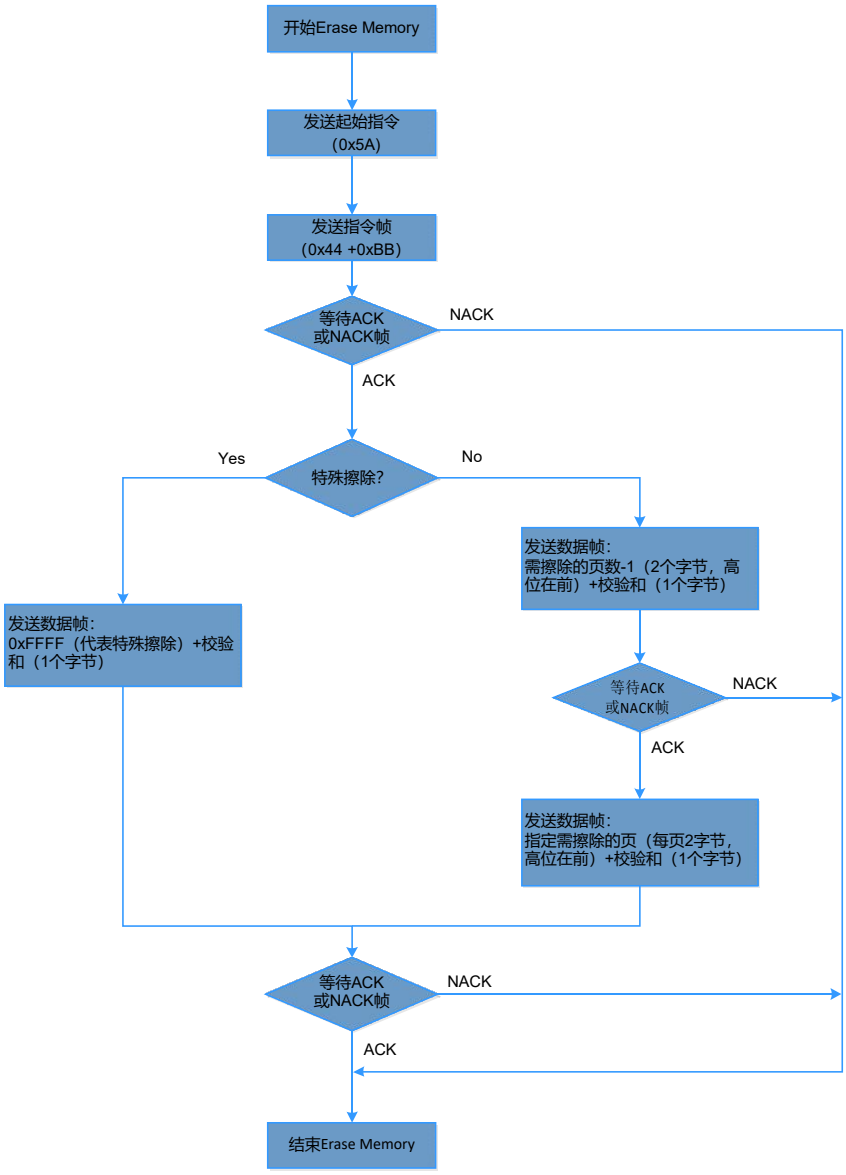

图 4-7 Erase 指令流程

4.9. Write Protect 指令（0x63）

Write Protect 指令用于用户对main flash 的某些指定扇区或整片进行写保护处理。FT32F0XX 每个扇区（sector）组织为 4k bytes。

当 Bootloader 接收到此指令后，会回复主机ACK，并进入等待主机发送待写保护扇区数量,主机应发送 1byte 待保护扇区数量N-1（实际发送待保护扇区数量 -1），后面跟随N 字节待保护的各个扇区号（每个扇区号 1 字节），最后发送扇区数量及各个扇区号的异或校验和（即格式为 1Bytes sector 数量(-1) + NBytes 扇区号 + 1Bytes XOR checksum = N+2 Bytes）。当 Bootloader 接收到信息后，进行校验，并回复主机ACK（校验通过）或 NACK（校验未通过）。

当校验通过且合理，Bootloader 对相应的扇区进行写保护开启，全部指定扇区开启写保护后，Bootloader 回复主机ACK，并通过复位对写保护状态进行加载，以便于新的写保护状态生效。

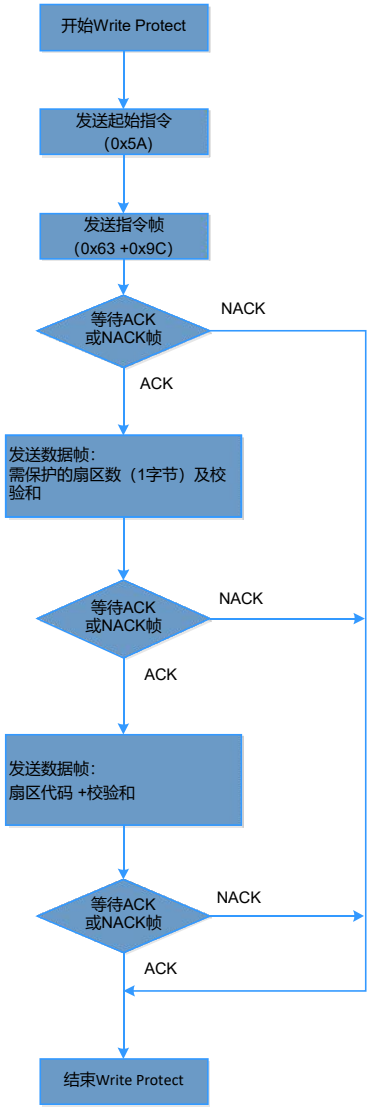

图 4-8 Write Protect 指令流程

4.10. Write Unprotect 指令（0x73）

Write Unprotect 指令用于用户对main flash 整区进行写保护解除处理。

当 Bootloader 接收到此指令后，会回复主机 ACK，并执行全片写保护解除，全部扇区解除写保护后，Bootloader 回复主机ACK，并通过复位对写保护状态进行加载，以便于写保护解除状态生效。

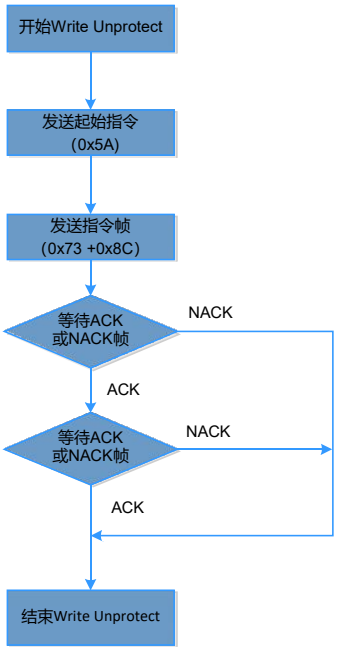

图 4-9 Write Unprotect 指令流程

4.11. Readout Protect 指令（0x82）

Readout Protect 指令用于用户对main flash 整区进行读出保护处理。

当 Bootloader 接收到此指令后，会回复主机ACK，并执行全片读出保护，完成读出保护后，Bootloader 回复主机ACK，并通过复位对新的保护状态进行加载，以便于读保护状态生效。

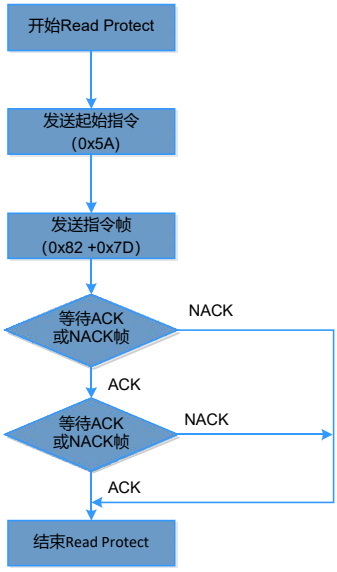

图 4-10 Readout Protect 指令流程

4.12. Readout Unprotect 指令（0x92）

Readout Unprotect 指令用于用户对main flash 的读出保护功能进行解除。

当 Bootloader 接收到此指令后，会回复主机ACK，并执行全片读出保护解除，此时会擦除main flash 全部数据，若擦除动作或保护解除动作完成异常，Bootloader 回复主机NACK；若正常完成读出保护解除动作，Bootloader 回复主机ACK，并通过复位对新的保护状态进行加载，以便于读保护解除状态生效。

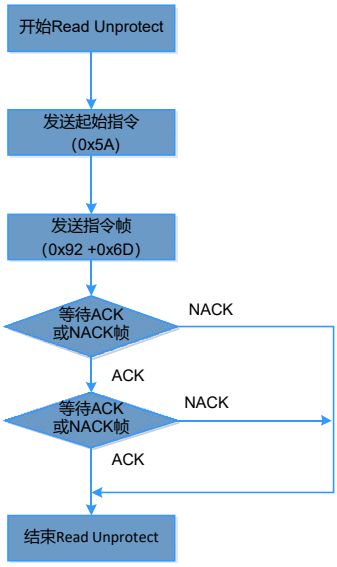

图 4-11 Readout Unprotect 指令流程

4.13. Get Checksum 指令（0xA1）

Get Checksum 指令用于计算给定地址区域的 CRC 值。地址必须是 4 字节的倍数。

当从机接收到Get Checksum 指令时，它将 ACK 字节发送给主机。

在 ACK 字节传输后，从机等待一个地址(四个字节，字节 1 是 MSB，字节 4 是 LSB)和一个校验和字节，然后从机检查收到的地址。如果地址有效且校验和正确，从机将发送一个 ACK 字节，否则它会发送一个NACK 字节并中止指令。

当地址有效且校验和正确时，从机将等待需要校验的字（4bytes）的个数以及校验和。

如果校验和不正确，从机会在终止指令之前发送一个 NACK。

如果校验和是正确的，从机检查地址是否不等于 0 和它不会超过内存的大小。如果内存大小正确，从机将 ACK 字节发送给主机。

当地址有效且校验和正确时，从机将等待 CRC 多项式值（只支持默认多项式:0x4C11DB7）及其校验和。

如果校验和是正确的，从机发送一个 ACK 字节，否则它发送一个NACK 字节并中止指令。

当校验和有效时，从机等待 CRC 初始化值及其校验和。

如果校验和是正确的，从机发送一个 ACK 字节，否则它发送一个NACK 字节并中止指令。

如果校验和是正确的，从机开始计算给定地址的 CRC，然后向主机发送一个 ACK，后面跟着 CRC 值以及其校验和。

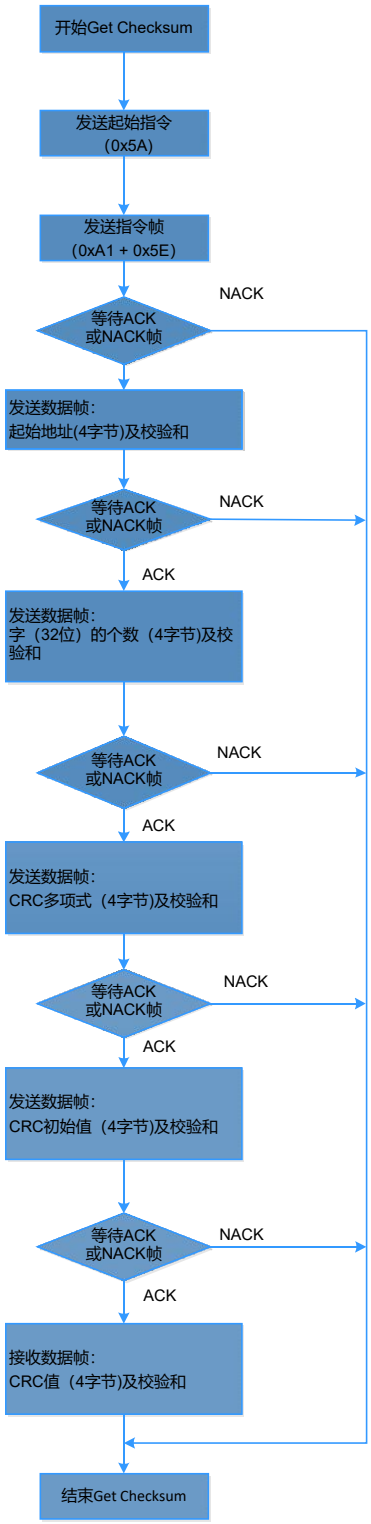

图 4-12 Get Checksum 指令流程

联系信息

4th Floor, Building 11,
Zhongke Innovation Plaza, 150 Pubin Road, Jiangbei New District,
Nanjing city, Jiangsu Province

Tel: 025-58101616
Fax: 025-58263220

http://www.touchmcu.com