



Fremont Micro Devices

AN-32010

I2C 协议的 ISP 使用说明

Rev1.00

www.fremontmicro.com

文档修改历史

日期	版本	描述
2021-10-29	1.00	初版

目录

1	文档简介	5
2	硬件配置	5
3	I2C 协议的 ISP 编码序列	6
4	BootLoader 指令集	7
4.1	GET 指令 (0x00)	8
4.2	Get Version (0x01)	9
4.3	Get ID 指令 (0x02)	11
4.4	Read Memory 指令 (0x11)	12
4.5	Go 指令 (0x21)	14
4.6	Write Memory 指令 (0x31)	16
4.7	Erase 指令 (0x44)	18
4.8	Write Protect 指令 (0x63)	22
4.9	Write Unprotect 指令 (0x73)	24
4.10	Readout Protect 指令 (0x82)	25
4.11	Readout Unprotect 指令 (0x92)	26
	联系信息	28

表目录 / List of Figures

表 4-1	BootLoader 指令集	7
表 4-2	Read Memory 有效存储区.....	12
表 4-3	Write Memory 有效存储区.....	16

图目录 / List of Tables

图 3-1	编码序列	6
图 4-1	GET 指令流程.....	8
图 4-2	Get version 指令流程.....	10
图 4-3	Get ID 指令流程.....	11
图 4-4	Read Memory 指令流程	13
图 4-5	Go 指令流程	15
图 4-6	Write Memory 指令流程.....	17
图 4-7	Erase 指令流程.....	20
图 4-8	Write Protect 指令流程	23
图 4-9	Write Unprotect 指令流程	25
图 4-10	Readout Protect 指令流程	26
图 4-11	Readout Unprotect 指令流程.....	27

1 文档简介

本文档主要描述了基于 FT32F0XX 的 I2C 编程 (ISP) 协议，是编写 I2C 协议 ISP 烧录主机的参考文档，其内容包含 ISP 硬件配置、I2C 通信协议以及详细的编程指令说明。

2 硬件配置

进行 ISP 烧录操作时，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：I2C1-SDA (PB7)、I2C1-SCL (PB6)。
- I2C 从机地址：0x3B
- BOOT0 引脚：上拉为高电平。
- 系统选项字中：nBOOT1 = 1。
- 通讯速率： 100KHz~1MHz。

3 I2C 协议的 ISP 编码序列

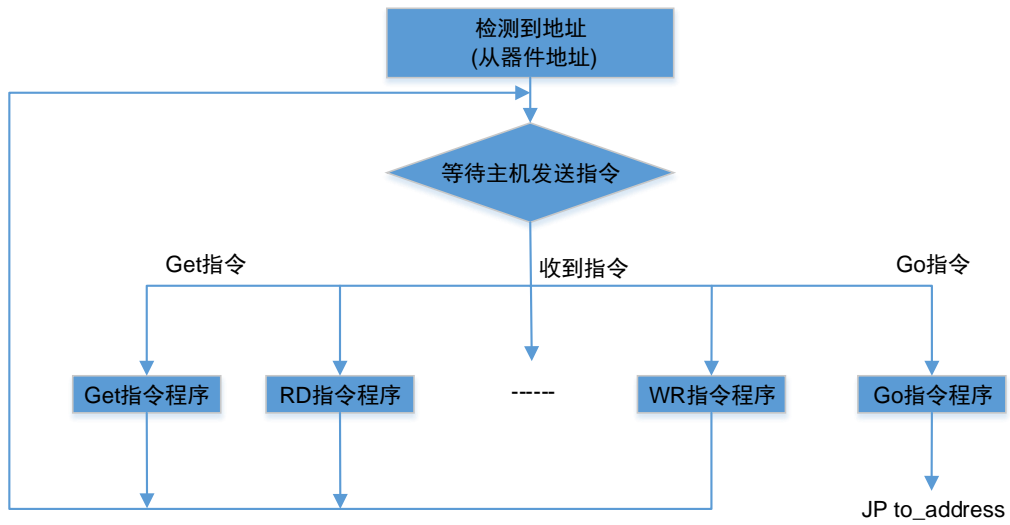


图 3-1 编码序列

当从系统存储区（0x1FFFE800~0x1FFFF7FF）启动时，存储于此引导区的代码将会不断地扫描 I2C1-SDA 和 I2C1-SCL，等待检测到 I2C 从机地址 (0x3B)，当主机发送的地址与从机地址匹配时，从机给主机一个应答信号，当主机收到应答信号之后，主机便可以发起相应的命令与从机通信。

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	将此命令后的相应数量及地址所对应的flash page进行擦除，1次命令可指定擦除1至127个page（每个page 0.5kB）。
Write Protect	0x63	使能指定flash扇区的写保护功能，F072共16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
Readout Protect	0x82	使能读保护功能
Readout Unprotect	0x92	失能芯片的读保护功能

表 4-1 BootLoader 指令集

注意：

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，然后返回到检查指令状态。
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，对器件没有作用。取消 RDP 之后，其它指令变为激活。
3. I2C 通信实现了超时机制，这是 Bootloader 程序指令正确执行所必需考虑的。此超时在同一指令的两个 I2C 帧间实现。例如，对于 Write memory 指令，在指令发送帧和地址存储器发送帧之间有超时机制。此外也将在同一 I2C 帧中的两个连续数据接收或发送实例之间插入同一超时周期。如果超时周期已过，则生成系统复位以避免自举程序崩溃。

为保证 I2C 通信安全传输，所有由主机发送的数据均要进行保护校验，校验方式如下：

1. checksum：接收到的数据块通过异或和进行校验，有效数据块后面会跟随 1Byte 的校验和，其值为此数据块的异或和。
2. 主机发送的每个指令均为 2Byte：1Byte 的指令、1Byte 的指令取反数（两数异或结果应为 0xFF）。
3. 每个数据包发送后，Bootloader 会回应接受指令应答 ACK(0x79)或拒绝指令应答 NACK(0x1F)。

另外说明，主机的帧可为下列之一：

1. 发送指令帧：主机作为主发送端发起通信，向 FT32F0XX 发送 2 个字节：命令码 + XOR(命令码的补码) 。2 个字节异或为 0xFF。
2. 等待 ACK/NACK 帧：主机作为主接收端发起 I2C 通信，从 FT32F0XX 接收 1 个字节：ACK 或 NACK。
3. 接收数据帧：主机作为主接收端发起 I2C 通信，从 FT32F0XX 读取相应字节数的数据。读取的字节数取决于指令。
4. 发送数据帧：主机作为主发送端发起 I2C 通信，向 FT32F0XX 发送需要的字节。发送的字节数取决于指令。

4.1 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 13bytes (接收字节 + BootLoader 版本 + 支持的指令列表。主机端 Get 指令处理如下：

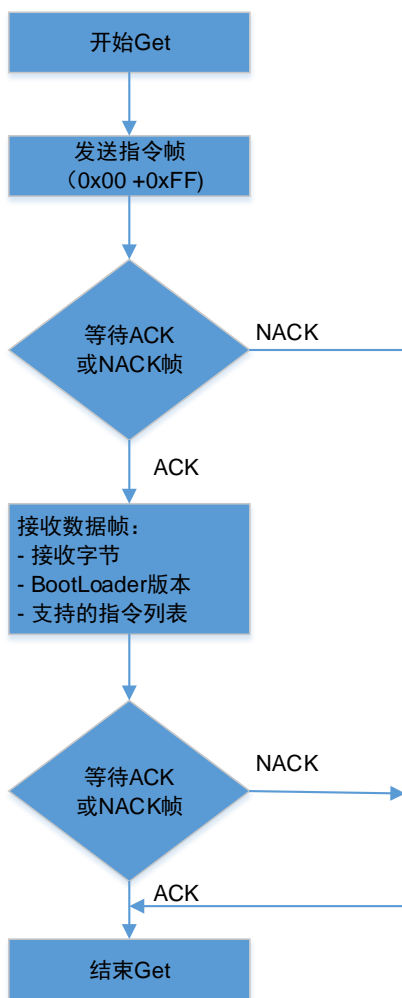


图 4-1 GET 指令流程

Get 指令 FT32F0XX 接收回复流程示例：

示例流程开始

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x00 0xFF (Get 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x0B (后续字节数是 11+1 = 12 个，即 Byte3~Byte14)

从机端回复 Byte3： 0x10 (Bootloader 版本 V1.0)

从机端回复 Byte4~14： 0x00 0x01 0x02 0x11 0x21 0x31 0x44 0x63 0x73 0x82 0x92 (每字节均表示 1 条支持的指令：Get、Get version.....)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte15： 0x79 (ACK)

示例流程结束。

4.2 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。

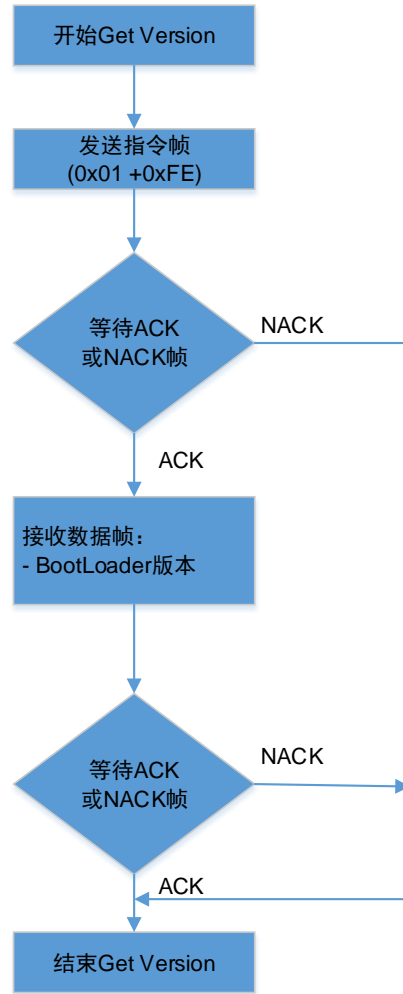


图 4-2 Get version 指令流程

Get Version 指令 FT32F0XX 接收回复流程示例:

示例流程开始

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte1~2: 0x01 0xFE (Get Version 指令)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte2: 0x10 (Bootloader 版本 V1.0)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte3: 0x79 (ACK)

示例流程结束。

4.3 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：

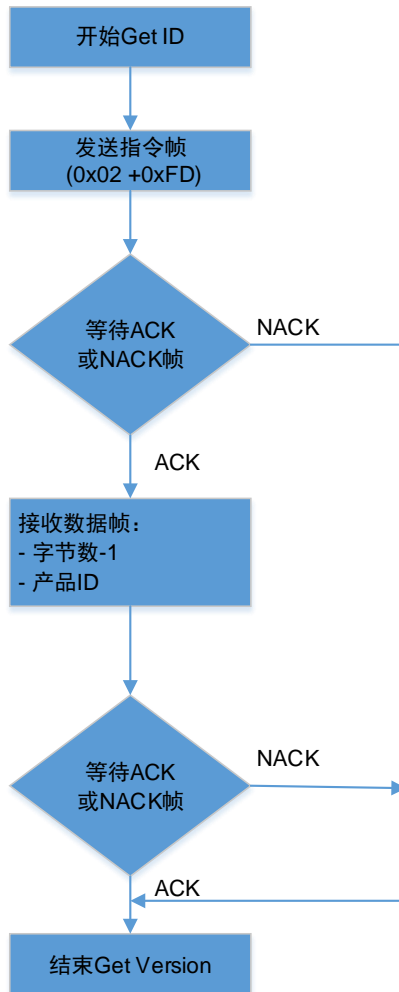


图 4-3 Get ID 指令流程

Get ID 指令 FT32F0XX 接收回复流程示例：

示例流程开始

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x02 0xFD (Get ID 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x01（后续字节数是 $1+1 = 2$ 个，即 Byte3~4）

从机端回复 Byte3： 0x04（芯片 PID_H）

从机端回复 Byte4： 0x48（芯片 PID_L）

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte5： 0x79 (ACK)

示例流程结束。

4.4 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	0x08000000~0x0800FFFF	64kbytes
User Option	0x1FFFF800~0x1FFFF813	20bytes
SRAM	0x20000000~0x20001FFF	8kbytes

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。

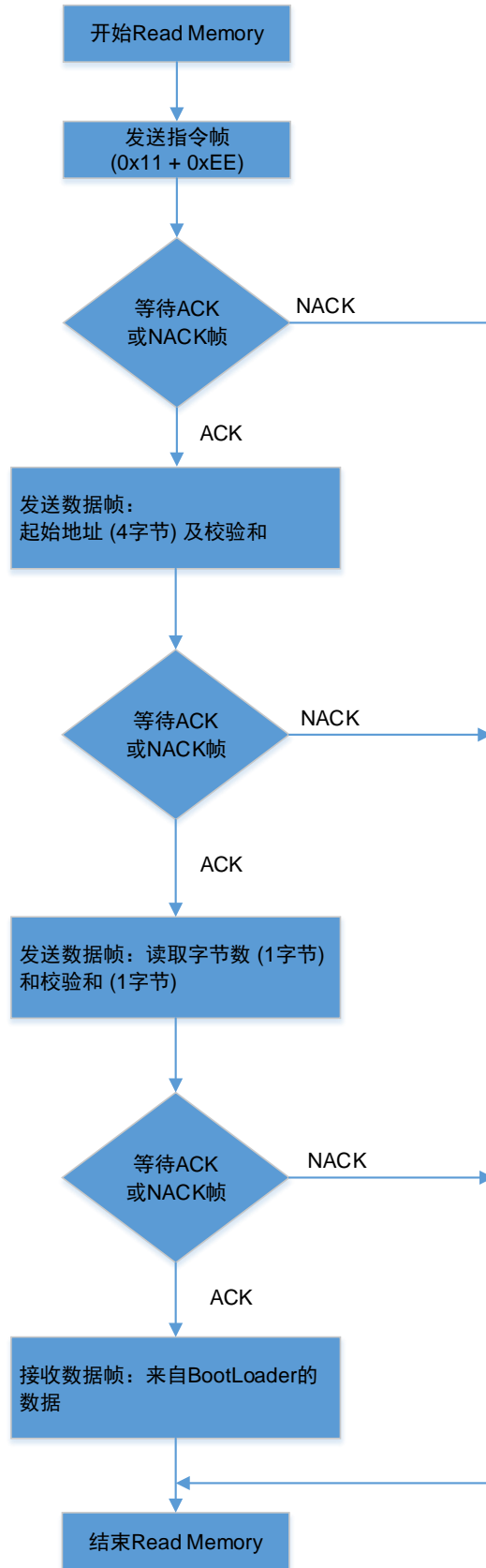


图 4-4 Read Memory 指令流程

Read Memory 指令 FT32F0XX 接收回复流程示例（从地址 0x08000000 读取 64 字节数据）：

示例流程开始

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x11 0xEE (Get ID 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte3~6： 0x08 0x00 0x00 0x00 (从 0x8000000 地址读取)

主机端发送 Byte7： 0x08 (Byte3~6 的 XOR 码)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x79 (ACK)

主机端发送写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte8： 0x3F (读取 0x3F +1 字节数据)

主机端发送 Byte9： 0xC0 (Byte8 的补码)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte3： 0x79 (ACK)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte4~Byte67： 0xFF 0xFF ... 0xFF

示例流程结束。

4.5 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 0x08000000~0x0800FFFF 及 SRAM 区域 0x20000000~0x20001FFF）。

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。

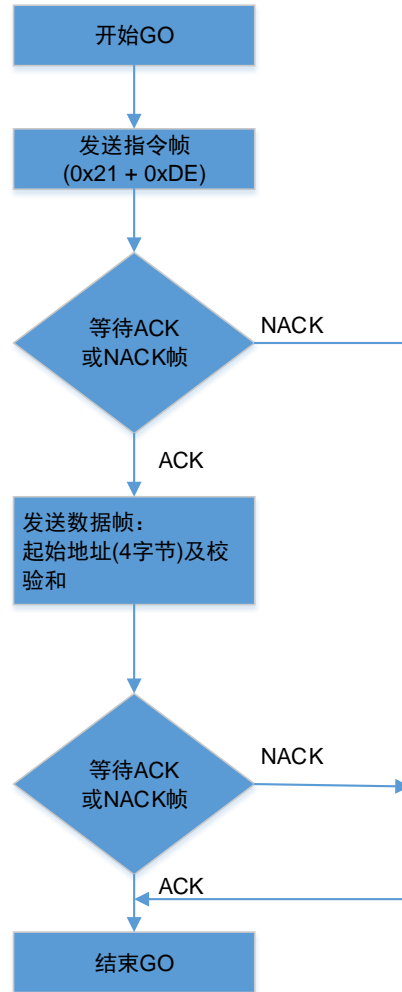


图 4-5 Go 指令流程

Go 指令 FT32F0XX 接收回复流程示例（跳转至 main flash 区的 0x08000000 执行程序）：

示例流程开始

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x21 0xDE (Go 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送 Byte3~6： 0x08 0x00 0x00 0x00 (待跳转地址为 0x08000000)

主机端发送 Byte7： 0x08 (Byte3 ~ Byte6 校验结果为 0x08)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x79 (ACK)

示例流程结束。

4.6 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	0x08000000~0x0800FFFF	64kbytes
User Option	0x1FFFF800~0x1FFFF813	20bytes
SRAM	0x20000000~0x20001FFF	8kbytes

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：

$$\text{checksum} = (N-1) \wedge \text{data1} \wedge \text{data2} \dots \text{dataN}$$

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。

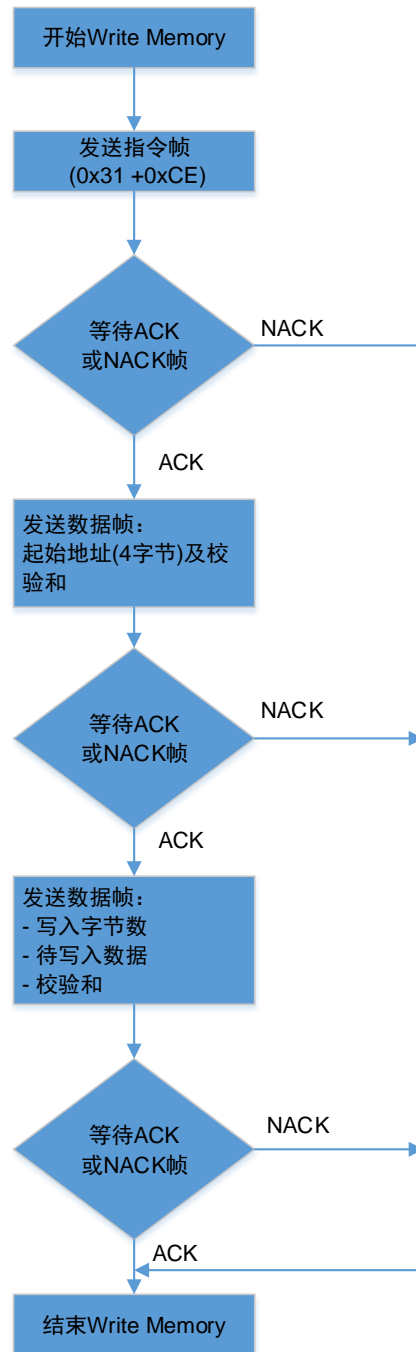


图 4-6 Write Memory 指令流程

Write Memory 指令 FT32F0XX 接收回复流程示例（向地址 0x08000000 写入 64 字节数据）：

示例流程开始

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x31 0xCE (Write Memory 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte3~6: 0x08 0x00 0x00 0x00 (Write Memory 指令)

主机端发送 Byte7: 0x08 (Byte3~6 校验结果为 0x08)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte2: 0x79 (ACK)

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte8: 0x3F (待写入数据长度-1, 即实际写入数据长度为 $0x3F + 1 = 0x40$ bytes)

主机端发送 Byte9~72: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07

0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F

0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17

0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F

0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27

0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F

0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37

0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F

主机端发送 Byte73: 0x3F (Byte8~72 的异或校验结果)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte3: 0x79 (ACK)

示例流程结束。

4.7 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 (FT32F0XXF072X 1page 为 0.5k 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 $N \times 2$ 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。

$$\text{Checksum} = 2\text{byte (待擦除数量 } N-1) \wedge N \times 2 \text{ bytes (所有待擦除 page 号)}$$

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。

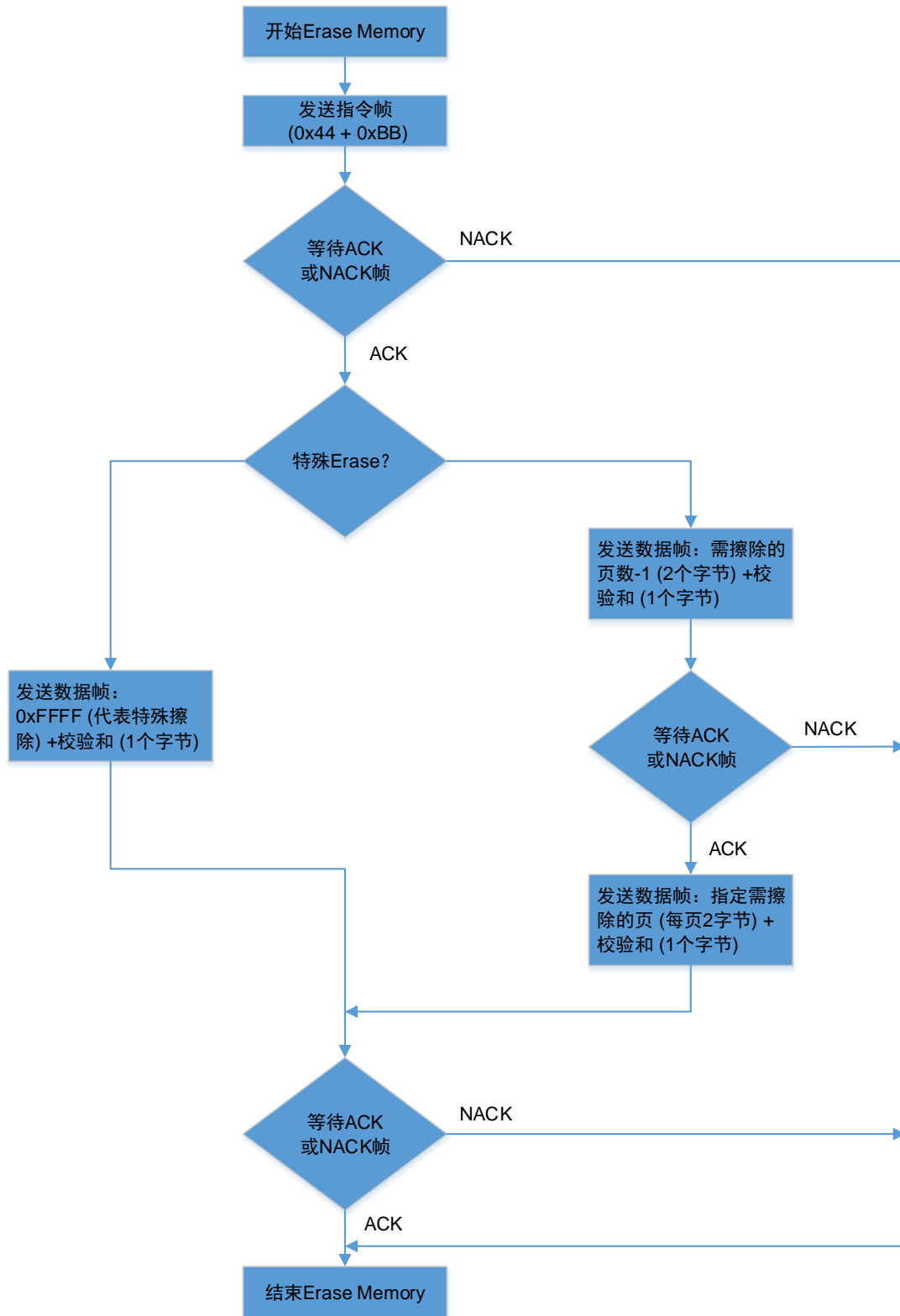


图 4-7 Erase 指令流程

Erase Memory 指令 FT32F0XX 接收回复流程示例 1（擦除整片 main flash）：

示例流程开始：

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2: 0x44 0xBB (Erase 指令)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

主机端发送 Byte3~4: 0xFF 0xFF (主机全片擦除指令码)

主机端发送 Byte5: 0x00 (校验字节 0xFF xor 0xFF = 0x00)

从机端回复 Byte2: 0x79 (ACK)

示例流程结束。

Erase Memory 指令 FT32F0XX 接收回复流程示例 2 (擦除 main flash 内指定数量的指定 page 区) :

示例流程开始:

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte1~2: 0x44 0xBB (Erase 指令)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

主机端发送 Byte3~4: 0x00 0x02 (需擦除的扇区数: 0x0002 +1)

主机端发送 Byte5: 0x02 (Byte3 ~ Byte4 校验结果为 0x02)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte2: 0x79 (ACK)

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte6~Byte11: 0x00 0x20 0x00 0x21 0x00 0x22 (要擦除的页编号: 0x0020,0x0021,0x0022)

主机端发送 Byte12: 0x23 (Byte6~ Byte11 校验结果为 0x23)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte2: 0x79 (ACK)

示例流程结束。

4.8 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。

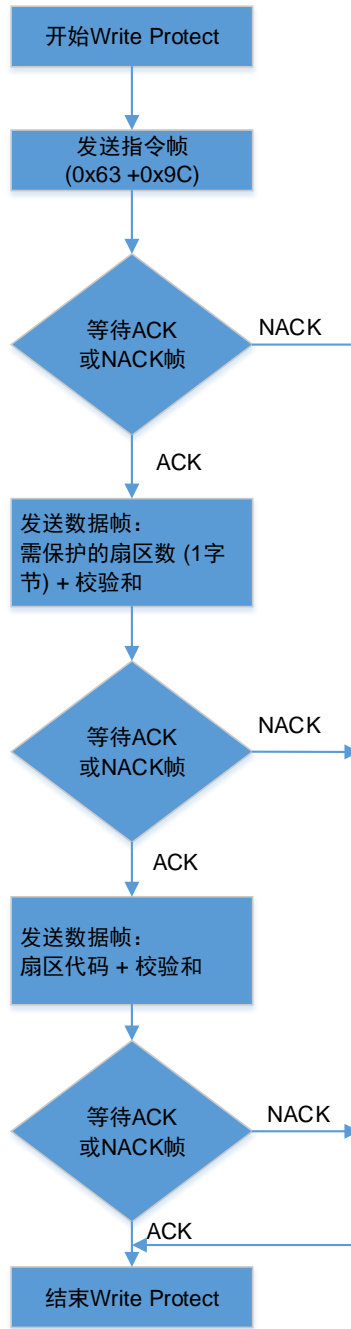


图 4-8 Write Protect 指令流程

Write Protect 指令 FT32F0XX 接收回复流程示例（写保护指定的 4 个扇区）：

示例流程开始：

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x63 0x9C (Write Protect 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte3: 0x03 (需保护的扇区数 : 0x03 +1)

主机端发送 Byte4: 0xFC (Byte3 的补码)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

主机端发起写传输地址: 0x3B (I2C_ADDR)

主机端发送 Byte5~Byte8: 0x07 0x08 0x09 0x0A (需保护的扇区编号)

主机端发送 Byte9: 0x0C (Byte4~Byte7 的异或码)

主机端发送读传输地址: 0x3B (I2C_ADDR)

从机端回复 Byte1: 0x79 (ACK)

示例流程结束。

4.9 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。

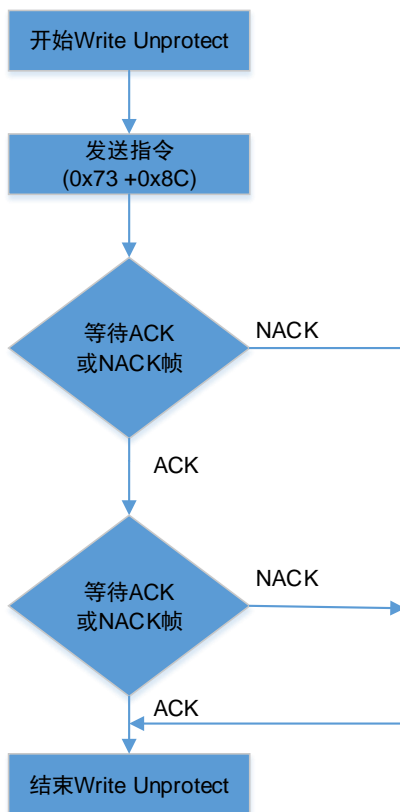


图 4-9 Write Unprotect 指令流程

Write Unprotect 指令 FT32F0XX 接收回复流程示例（main flash 全部写保护解除）：

示例流程开始：

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x73 0x8C (Write Unprotect 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x79 (ACK)

示例流程结束。

4.10 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。

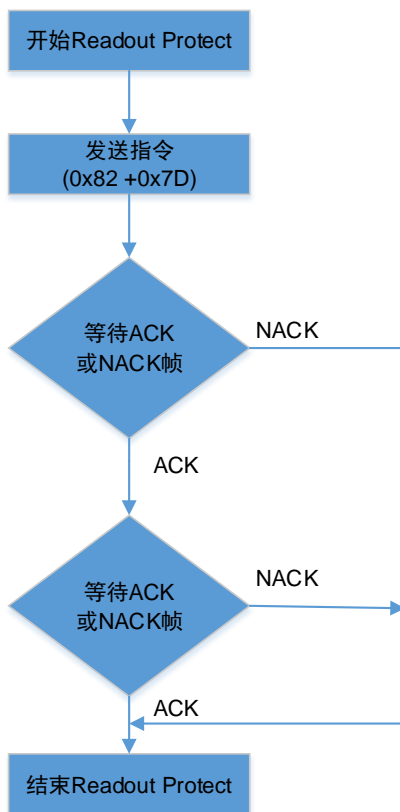


图 4-10 Readout Protect 指令流程

Readout Protect 指令 FT32F0XX 接收回复流程示例（main flash 读保护）：

示例流程开始：

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x82 0x7D (Readout Protect 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x79 (ACK)

示例流程结束。

4.11 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。

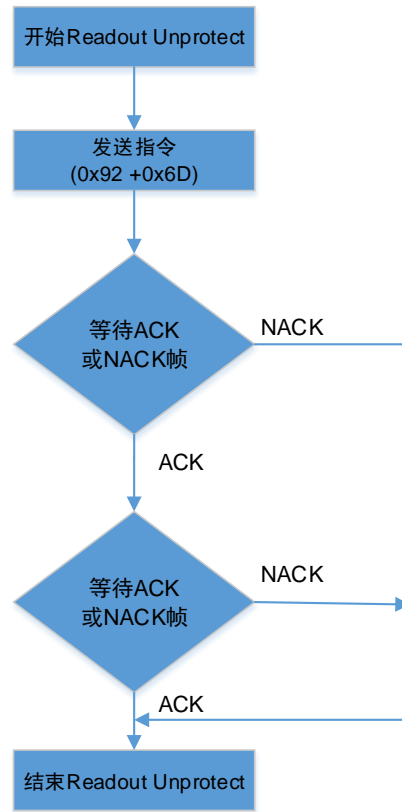


图 4-11 Readout Unprotect 指令流程

Readout Unprotect 指令 FT32F0XX 接收回复流程示例（main flash 读保护解除）：

示例流程开始：

主机端发起写传输地址：0x3B (I2C_ADDR)

主机端发送 Byte1~2： 0x92 0x6D (Readout Unprotect 指令)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte1： 0x79 (ACK)

主机端发送读传输地址：0x3B (I2C_ADDR)

从机端回复 Byte2： 0x79 (ACK)

示例流程结束。

联系信息**Fremont Micro Devices Corporation**

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.