Fremont Micro Devices

AN-32017

USB 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
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 USB 编程（ISP）协议，是编写 USB 协议 ISP 烧录主机的参考文档，其内容包含 ISP 硬件配置、USB 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：USB (PA11:USB-DM, PA12:USB-DP)。
- BOOT0 引脚：上拉为高电平。
- 系统选项字：nBOOT1 = 1。
- 通讯速率： 12Mbit/s。
- BootLoader 使用的 SRAM: 6 KByte。

3. USB 协议的 ISP 编码序列

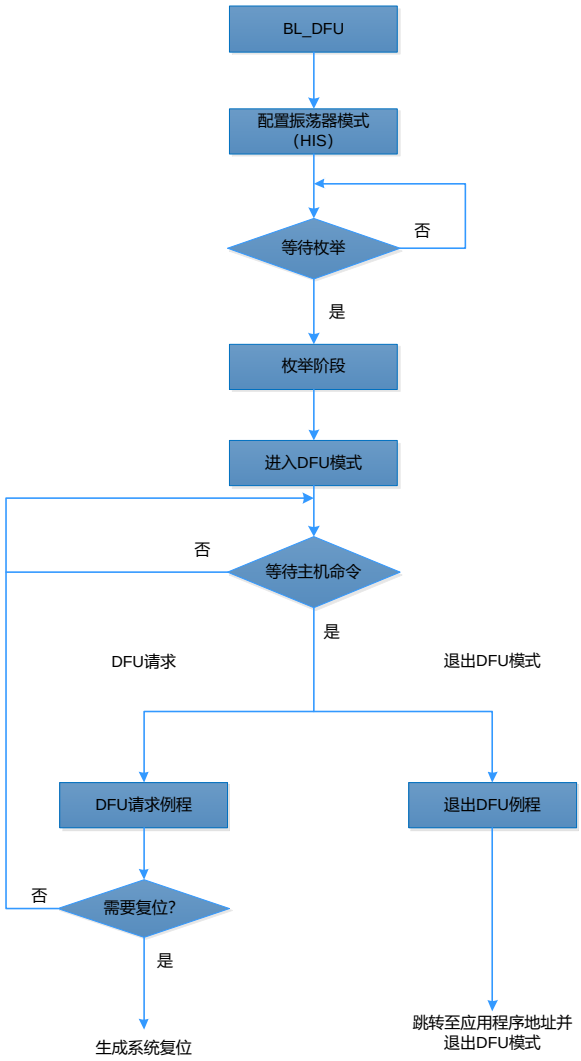

图 3-1 编码序列

当从系统存储区（0x1FFFC800~0x1FFFF7FF）启动时，存储于此引导区的代码将会不断地扫描 USB，当从机接入电脑之后，从机开始枚举，枚举完成之后，从机进入 DFU 模式，等待主机命令。

注：

1. 系统复位后，器件可能会返回到 BL_DFU 循环，也可能执行 Flash 存储器中的代码，具体视连接状态和 BOOT0 引脚的状态而定。

2. 先发出 0 Data Download 请求，然后再发出 GetStatus 请求和 Device Reset 请求，即可退出 DFU 模式。

4. USB DFU BootLoader 请求

请求	请求代码	请求说明
DFU_DETACH	0x00	请求器件退出 DFU 模式并进入应用程序。
DFU_DNLOAD	0x01	请求将数据从主机传输到器件，以便将数据加载到器件的内部 Flash 存储器中。还包括擦写指令。
DFU_UPLOAD	0x02	请求将数据从器件传输到主机，以便将器件内部 Flash 存储器的内容加载到主机文件中。
DFU_GETSTATUS	0x03	请求器件向主机发送状态报告（包括执行上一请求后得出的状态以及执行该请求后器件将立即进入的状态）。
DFU_CLRSTATUS	0x04	请求器件清除错误状态并转至下一步。
DFU_GETSTATE	0x05	请求器件仅发送将在该请求后立即进入到的状态。
DFU_ABORT	0x06	请求器件退出当前状态/操作并立即进入空闲状态。

表 4-1 DFU 类请求

BmRequest	bRequest	wValue	wIndex	wLength	Data
00100001b	DFU_DETACH	wTimeout	Interface	Zero	None
00100001b	DFU_DNLOAD	wBlockNum	Interface	Length	Firmware
10100001b	DFU_UPLOAD	Zero	Interface	Length	Firmware
00100001b	DFU_GETSTATUS	Zero	Interface	6	Status
00100001b	DFU_CLRSTATUS	Zero	Interface	Zero	None
00100001b	DFU_GETSTATE	Zero	Interface	1	Status
00100001b	DFU_ABORT	Zero	Interface	Zero	None

表 4-2 DFU 类特有的请求汇总

5. DFU BootLoader 指令

DFU_DNLOAD 和 DFU_UPLOAD 请求主要用于执行简单的存储器读写操作。这两个请求也用于发出集成 BootLoader 指令（write、read unprotect、erase、set address 等）。

DFU_GETSTATUS 指令会触发这些指令被真正执行。

在 DFU 下载请求中，指令是通过 USB 请求结构中的 wValue 参数选择的。如果 wValue = 0 主机在发送请求之后发出的数据就是 BootLoader 指令代码。第一个字节是指令代码，其它字节（如果存在）是与该指令相关的数据。

在 DFU 上传请求中，指令是通过 USB 请求结构中的 wValue 参数选择的。如果 wValue = 0，则会选择 Get 指令并执行。

DFU 请求	BootLoader 指令	写保护禁用 读保护禁用	写保护启用 读保护禁用	读保护启用
DFU_UPLOAD	Read Memory	允许	允许	不允许
	Get	允许	允许	允许
DFU_DNLOAD	Write Memory	允许	允许	不允许
	Erase	允许	允许	不允许
	Read Unprotect	无意义	无意义	允许
	Set Address Pointer	允许	允许	允许
	Leave DFU mode	允许	允许	允许

表 5-1 DFU 请求执行条件

如果执行 Read Unprotect 操作，同时存储器未受保护，那么整个 RAM 存储器会被 BootLoader 固件清空，而 Flash 存储器不会被擦除（由于 Flash 存储器之前未受读保护）。

没有针对 Write Protect、Write Unprotect 和 Read Protect 操作的指令。这些操作应通过用于选项字节区域的 Write Memory 和 Read Memory 指令来执行。

6. DFU_UPLOAD 请求指令

上传请求允许执行不同指令。指令是通过 USB 请求结构中 wValue 的参数值来选择的。

6.1. Read Memory 指令

当 wValue > 1 时，会选择 Read memory 操作。

主机会请求器件从内部 Flash 存储器、嵌入式 RAM、系统存储器的有效存储器地址、或者从选项字节发送指定数目的数据字节 (wLength)。

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

表 6-1 Read Memory 有效存储区

允许读取的字节数取决于存储器目标：

- 对于内部 Flash 存储器、嵌入式 RAM 和系统存储器：读取字节的大小为 2 到 2048 字节
- 对于选项字节：读取字节的大小应等于选项字节块的大小

主机请求从哪一地址开始读取数据是使用 wBlockNumber (wValue) 的值以及地址指针在下列公式中计算得出的：

地址 = ((wBlockNum – 2) × wTransferSize) + Address_Pointer，其中： – wTransferSize 是请求的数据缓冲区的长度。

地址指针应事先通过 Set Address Pointer 指令（使用 DFU_DNLOAD 请求）指定。否则（如果未事先指定地址），器件会假定起始地址为内部 Flash 起始地址 (0x08000000)。

如果启用 Flash Read Protection，无论读取目标是内部 Flash 存储器、嵌入式 RAM、系统存储器还是选项字节，都不会执行 Read 操作，返回的器件状态为 Status = dfuERROR、State = errVENDOR。

6.2. Get 指令

wValue = 0 时，会选择该指令。

主机会请求读取 BootLoader 支持的指令。收到该指令后，器件会返回 N 个代表指令代码的字节。BootLoader 会发送以下字节 (N = 4)：

- 字节 1： 0x00 - Get 指令
- 字节 2： 0x21 - Set Address Pointer
- 字节 3： 0x41 - Erase

- 字节 4： 0x92 - Read Unprotect

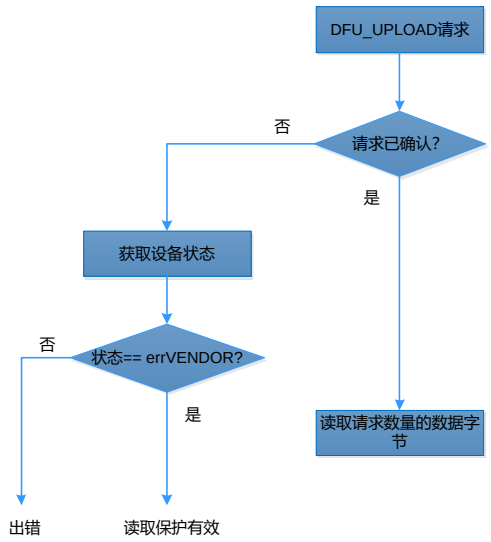

图 6-1 DFU_UPLOAD 请求

注：发出 Upload 请求之前，主机必须检查器件是否处于正确状态（dfuIDLE 或 dfuUPLOAD-IDLE 状态），并且要检测状态中是否报错。如果器件并未处于要求的状态，主机需要清除错误（DFU_CLRSTATUS 请求）并获取新状态，直至器件恢复到 dfuIDLE 状态。

7. DFU_DNLOAD 请求指令

下载请求用于执行不同的指令。指令是通过 USB 请求结构中 wValue 的参数值来选择的。

支持的操作如下：

- Write Memory (wValue > 1)
- Set Address Pointer （wValue = 0，并且第一个字节= 0x21）
- Erase （wValue = 0，并且第一个字节= 0x41）
- Read Unprotect （wValue = 0，并且第一个字节 = 0x92）
- Leave DFU （退出 DFU 模式并跳转至应用程序）

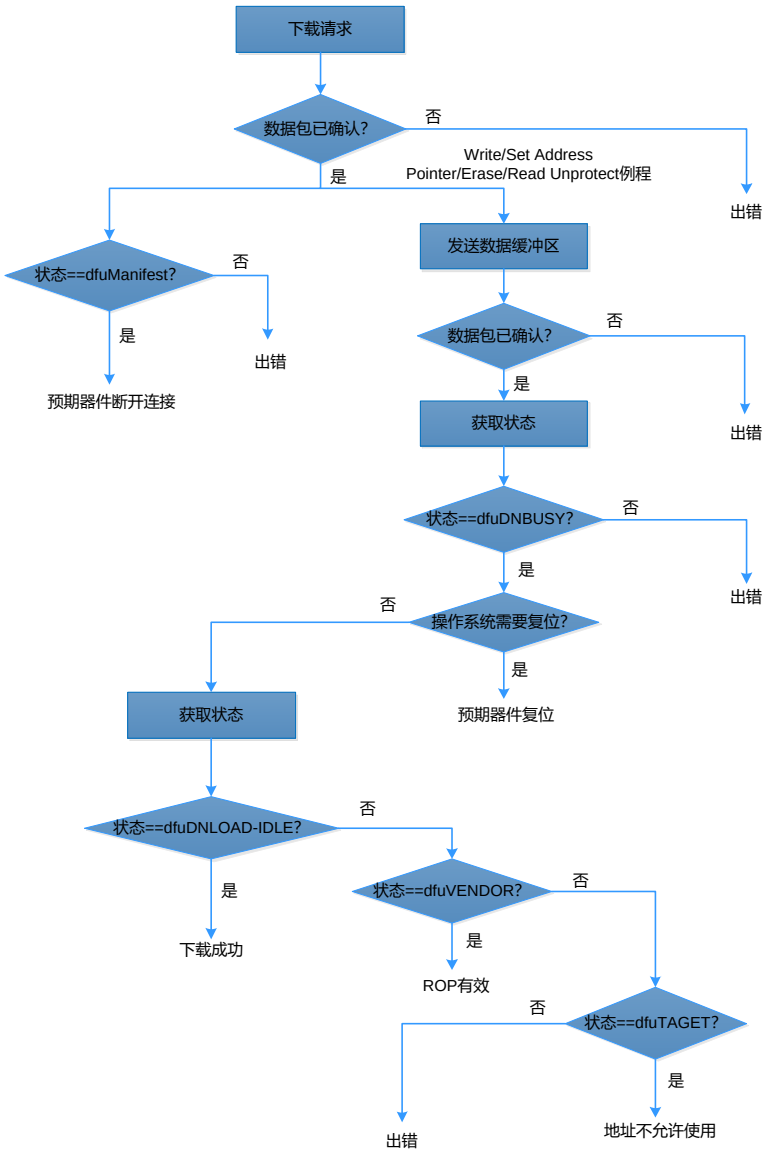

图 7-1 DFU_DNLOAD 请求

注：

1. 需要系统复位的操作包括：对选项字节执行的 Read Unprotect 指令和 Write 操作。

2. 发出 Download 请求之前，主机必须检查器件是否处于正确状态（dfuIDLE 或 dfuDNLOD-IDLE 状态），并且要检测状态中是否报错。如果器件并未处于要求的状态，主机需要清除其错误（DFU_CLRSTATUS 请求）并再次获取状态，直至器件恢复到 dfuIDLE 状态。

7.1. Write Memory 指令

wValue > 1 时，会选择 Write Memory 操作。

主机会请求器件接收指定数目 (wLength) 的数据字节，并将这些字节加载到内部 Flash 存储器、嵌入式 RAM 中的有效存储器地址（见说明）或选项字节中。

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

表 7-1 Write Memory 地址

允许写入的字节数取决于存储器目标：

- 对于内部 Flash 存储器和嵌入式 RAM：写入字节的大小为 2 到 2048 字节
- 对于选项字节：写入字节的大小应等于选项字节块的大小

注： 对于选项字节，可以写入不同于块大小的字节，但建议一次性写入整个块，以确保数据完整性。如果目标为选项字节区域，地址指针必须始终是选项字节的起始地址，否则将不会执行请求。

仅当 DFU_GETSTATUS 请求是由主机发出的情况下，Write memory 操作才能有效执行。如果器件返回的状态不是 dfuDNBUSY，说明发生了错误。

需要再发一次 DFU_GETSTATUS 请求，检查指令是否正确执行，但目标位置是选项字节区域的情况除外（在这种情况下，器件会在写入操作完成后立即复位）。如果接收到的地址不正确或不受支持，器件状态会变为 Status = dfuERROR、 State = errTARGET。

主机请求从哪一地址开始写入数据是使用 wBlockNumber (wValue) 的值以及地址指针在与上传请求相同的公式中计算得出的：

地址 = ((wBlockNum – 2) × wTransferSize) + Addres_Pointer，其中：

– wTransferSize：主机发送的数据缓冲区的长度

– wBlockNumber：wValue 参数的值

如果启用 Flash Read Protection，无论读取目标是内部 Flash 存储器、嵌入式 RAM、系统存储器还是选项字节，都不会执行 Write memory 操作，返回的器件状态为 Status = dfuERROR、State = errVENDOR。

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，BootLoader 会生成系统复位，以使选项字节的新配置生效。

注：

1. 当写入 RAM 时，您应注意不要与 BootLoader 固件使用的第一个 RAM 存储器重叠。

2. 当向写保护的扇区执行写操作时，不会返回错误。

7.2. Set Address Pointer 指令

如果 wValue = 0，并且主机发送的缓冲区的第一个字节是 0x21，则会选择 Set Address Pointer 指令。缓冲区长度应为 5（其余四个字节是地址字节，LSB 优先（32 位地址格式））。

主机会发送包含上述参数的 DFU_DNLOAD 请求，以设置计算 Read memory 和 Write memory 操作的起始地址所使用的地址指针值。

BootLoader 器件接收的字节如下：

- 字节 1： 0x21 - Set Address Pointer 指令
- 字节 2： A[7:0] - 地址指针的 LSB
- 字节 3： A[15:8] - 地址指针的第二个字节
- 字节 4： A[22:16] - 地址指针的第三个字节
- 字节 4： A[31:23] - 地址指针的 MSB

发送 Set Address Pointer 指令后，主机需要发送 DFU_GETSTATUS 请求。

仅当 DFU_GETSTATUS 请求是由主机发出的情况下，Set AddressPointer 指令才能有效执行。如果器件返回的状态不是 dfuDNBUSY，说明发生了错误。

需要再发一次 DFU_GETSTATUS 请求，检查指令是否正确执行。如果接收到的地址不正确或不受支持，器件状态会变为 Status = dfuERROR、State = errTARGET。

允许存储地址指针值的位置包括 Flash 存储器、嵌入式 RAM、系统存储器中的有效存储器地址（见说明）以及选项字节。

7.3. Erase 指令

如果 wValue = 0，并且主机发送的缓冲区的第一个字节是 0x41，则会选择 Erase 指令。对于页擦除操作，缓冲区长度是 5 个字节（其余四个字节是地址字节，LSB 优先），对于批量擦除操作，缓冲区也可以只有 1 个字节（仅包含指令字节）。

主机会发送包含上述参数的 DFU_DNLOAD 指令，以擦除一页内部 Flash 存储器，或对该 Flash 存储器执行批量擦除。

BootLoader 接收到的字节如下 （页擦除）：

- 字节 1： 0x41 - Erase 指令
- 字节 2： A[7:0] - 页地址的 LSB
- 字节 3： A[15:8] - 页地址的第二个字节
- 字节 4： A[22:16] - 页地址的第三个字节
- 字节 4： A[31:23] - 页地址的 MSB

或者，如果接收到 1 个字节的指令：

FT32 接收到的字节如下（批量擦除）：

- 字节 1： 0x41 - Erase 指令

发送 Erase 指令后，主机需要发送 DFU_GETSTATUS 请求。

仅当 DFU_GETSTATUS 请求是由主机发出的情况下，Erase 指令才能有效执行。如果器件返回的状态不是 dfuDNBUSY，说明发生了错误。

需要再发一次 DFU_GETSTATUS 请求，检查指令是否正确执行。如果接收到的页地址不正确或不受支持，器件状态会变为 Status = dfuERROR， State = errTARGET。如果激活了 Flash Read Protection，器件会恢复为状态 Status = dfuERROR，State = errVENDOR，并且器件会忽略擦写操作。

允许的 Erase 页地址为内部 Flash 存储器地址。

注：

当向写保护的扇区执行 Erase 操作时，不会返回错误。

7.4. Read Unprotect 指令

如果 wValue = 0，并且主机发送的缓冲区的第一个字节是 0x92，则会选择 Read Unprotect 指令。缓冲区长度应仅为 1 个字节（仅包含指令字节）。 主机会发送包含上述参数的 DFU_DNLOAD 请求，以取消对内部 Flash 存储器的读保护。

BootLoader 接收到的字节如下：

- 字节 1： 0x92 - Read Unprotect 指令

发送 Read Unprotect 指令后，主机需要发送 DFU_GETSTATUS 请求。

仅当 DFU_GETSTATUS 请求是由主机发出的情况下，Read Unprotect 指令才能有效执行。 如果器件返回的状态不是 dfuDNBUSY，说明发生了错误。执行此操作后，器件会取消 Read Protection，进而会彻底擦除内部 Flash 存储器和嵌入式 RAM。

因此，执行完该指令之后，器件会立即断开自身连接，并会执行系统复位。在这种情况下，器件将无法响应下一个 Get Status 请求。主机必须等待器件再次被枚举。

7.5. Leave DFU mode

可以使用 DFU 下载请求退出 DFU 模式(和 BootLoader)并跳转到已加载的应用程序（内部 Flash 存储器或嵌入式 RAM 中）。

主机会发送数据长度为 0 的 DFU_DNLOAD 请求（请求后没有数据阶段），以通知器件主机需要退出 DFU 模式。如果器件当前状态为 dfuDNLOAD-IDLE 或 dfuIDLE，则会确认该请求。

仅当 DFU_GETSTATUS 请求是由主机发出的情况下，DFU Leave 操作才能有效执行。如果器件返回的状态不是 dfuMANIFEST，说明发生了错误。执行该操作后，器件会执行以下操作：

- 断开自身连接
- 将 BootLoader 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 ’ 地址指针 + 4’ 所编程的存储器位置，对应于应用复位处理程序的地址。 例如，若收到的地址为 0x08000000，则 BootLoader 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。

需要在启动 Leave DFU 例程之前设定地址指针（使用 Set Address Pointer 指令），否则，BootLoader 将跳转到默认地址（内部 Flash 存储器起始地址：0x08000000）。还可以通过上一 Write Memory 操作设置地址指针：如果执行的是下载操作，则将存储为此次下载使用的地址指针，并在后续跳转时使用该地址指针）。

注：

1. 如果地址指针指向的地址不包含可执行代码，那么器件会复位，并可能再次进入 BootLoader 模式（具体视 BOOT0 引脚的状态而定）。

2. 由于 BootLoader DFU 应用不允许进行表示，表示阶段完成后，器件将无法响应主机请求。还可以再发一次 DFU_GETSTATUS 请求（器件仍保持连接的情况下），检查指令是否正确执行。如果器件无法执行指令，则会返回错误状态（具体视错误类型而定）。

3. 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。

4. 从 BootLoader 跳转到使用 USB IP 的已加载应用程序代码时，用户应用程序需要先禁用所有待处理的 USB 中断并复位内核，然后再启用中断。否则，待处理中断（通过 BootLoader 代码发出）可能会干扰用户代码并导致函数错误。退出系统存储器自举模式后，不需要执行此步骤。

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