

TK8610 芯片固件烧录协议

使用说明书

V1.0



造生物联
TAOLINK TECHNOLOGIES

修订记录

修订时间	修订版本	修订描述
2023-3-28	V1.0	初始版本

重要声明

版权所有 © 上海道生物联技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得对此文档的全部或部分内容进行使用、复制、修改、抄录，并不得以任何形式传播。

TurMass™ 为上海道生物联技术有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

上海道生物联技术有限公司保留随时变更、订正、增强、修改和改良此文档的权利，本文档内容可能会在未提前知会的情况下不定期进行更新。

除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议都依赖于具体的操作环境，并且不构成任何明示或暗示的担保。

联系方式

地址：上海嘉定皇庆路 333 号上海智能传感器产业园区 4 幢 5 层

邮编：201899

电话：021-61519850

邮箱：info@taolink-tech.com

网址：www.taolink-tech.com

目录

1 概述	1
2 功能和规格	2
3 运行环境	3
3.1 硬件环境	3
3.2 软件环境	3
4 使用说明	4
4.1 软件的部署	4
4.2 配置和操作	4
4.2.1 UART 参数配置	4
4.3 协议介绍	4
4.3.1 数据帧格式	4
4.3.2 协议命令集	4
4.4 实现细节	8
4.4.1 握手交互部分	9
4.4.2 烧录交互部分	9
4.5 注意事项	15

1 概述

TK8610 芯片固件烧录协议是道生物联自主开发的一套通讯协议，用于上位机工具对芯片的 FLASH 进行擦除、固件烧录和固件读取。

2 功能和规格

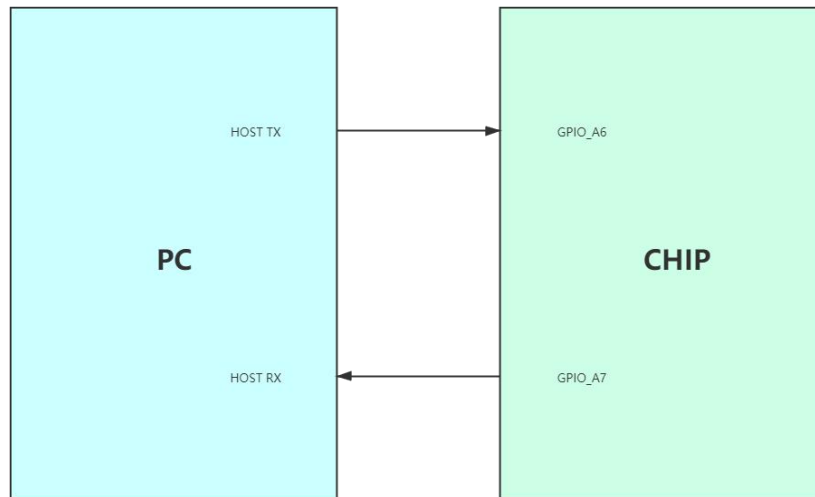
功能	说明
握手	通过握手判断通讯稳定性和正确性
波特率自适应	TK8610 芯片支持动态修改波特率 115200bps/921600bps, 默认为 115200bps, 芯片根据上位机的波特率自动适配。
Flash 操作	支持擦除、写入和读取
校验	烧录过程带校验

3 运行环境

支持 UART 接口的任何设备均可以根据本协议实现对 TK8610 芯片的固件烧写。

3.1 硬件环境

TK8610 终端芯片开发板或等同的模组等。如用 PC 机开发本协议，硬件接口部分如下图所示：



3.2 软件环境

略。

4 使用说明

4.1 软件的部署

本协议可被部署在任何支持 UART 接口的设备上, 实现对 TurMass TK8610 芯片的固件烧录。

4.2 配置和操作

UART 接口须支持波特率 115200bps 或 921600bps。

4.2.1 UART 参数配置

波特率	115200bps (默认), 支持配置命令
数据位	8
停止位	1
校验位	无
硬件流控	无

4.3 协议介绍

4.3.1 数据帧格式

命令类型	参数(数据遵循小端模式 (Little-endian))
1 Byte	6 Bytes

4.3.2 协议命令集

命令类型	值	描述
OPCODE_WRITE	0x02	写 flash
OPCODE_WRITE_ACK	0x03	写 flash 响应
OPCODE_WRITE_RAM	0x04	写 RAM 操作
OPCODE_WRITE_RAM_ACK	0x05	写 RAM 反馈
OPCODE_READ	0x08	读 flash
OPCODE_READ_ACK	0x09	读 flash 响应
OPCODE_ERASE	0x0C	擦除 flash(sector/block)
OPCODE_ERASE_ACK	0x0D	擦除 flash(sector/block)响应
OPCODE_DISCONNECT	0x10	断开 host 连接
OPCODE_DISCONNECT_ACK	0x11	断开 host 连接响应
OPCODE_CHANGE_BAUDRATE	0x12	修改波特率
OPCODE_CHANGE_BAUDRATE_ACK	0x13	修改波特率响应
OPCODE_EXECUTE_CODE	0x15	执行 code 命令

OPCODE_EXECUTE_CODE_END	0x17	执行 code 响应
OPCODE_CALC_CRC32	0x19	CRC32 校验命令
OPCODE_CALC_CRC32_ACK	0x1a	CRC32 校验回复
OPCODE_UNLOCK_FLASH	0x1f	解锁 flash (只能在 patch 中执行)
OPCODE_UNLOCK_FLASH_ACK	0x20	解锁 flash 回复 (只能在 patch 中执行)
OPCODE_LOCK_FLASH	0x21	锁定 flash (只能在 patch 中执行)
OPCODE_LOCK_FLASH_ACK	0x22	锁定 flash 回复 (只能在 patch 中执行)

4.3.2.1 OPCODE_WRITE

字段	大小	值	描述
命令	1	0x02	
地址	4	address	Flash 地址
长度	2	length	数据长度 (最大 16K)
数据	length		数据内容 (最大 16K)

4.3.2.2 OPCODE_WRITE_ACK

字段	大小	值	描述
命令	1	0x03	
地址	4	address	Flash 地址
长度	2	length	数据长度

4.3.2.3 OPCODE_READ

字段	大小	值	描述
命令	1	0x08	
地址	4	address	Flash 地址
长度	2	length	数据长度

4.3.2.4 OPCODE_READ_ACK

字段	大小	值	描述
命令	1	0x09	
地址	4	address	Flash 地址
长度	2	length	数据长度
数据	length		数据内容

4.3.2.5 OPCODE_DISCONNECT

字段	大小	值	描述
命令	1	0x10	

参数	1	0x01	后续操作 0 – 不退出交互模式 1 – 退出交互模式
填充	5	0	填充数据 0

4.3.2.6 OPCODE_DISCONNECT_ACK

字段	大小	值	描述
命令	1	0x11	
填充	6	Dummy	No care

4.3.2.7 OPCODE_CHANGE_BAUDRATE

字段	大小	值	描述
命令	1	0x12	
波特率	1	0x0B	波特率 bps 0x00 – 1200 0x01 – 2400 0x02 – 4800 0x03 – 9600 0x04 – 14400 0x05 – 19200 0x06 – 38400 0x07 – 57600 0x08 – 115200 0x09 – 230400 0x0A – 460800 0x0B – 921600
填充	5	0	填充数据 0

4.3.2.8 OPCODE_CHANGE_BAUDRATE_ACK

字段	大小	值	描述
命令	1	0x13	
填充	6	Dummy	No care

4.3.2.9 OPCODE_ERASE

字段	大小	值	描述
命令	1	0x0C	
地址	4	address	Flash Erase 起始地址 (flash 每次自起始地址自动擦除 4K 大小,

			下次擦除起始地址需+4K)
长度	2	length	填充数据 0

4.3.2.10 OPCODE_ERASE_ACK

字段	大小	值	描述
命令	1	0x0D	
地址	4	address	Flash Erase 起始地址
长度	2	length	No care

4.3.2.11 OPCODE_CALC_CRC32

字段	大小	值	描述
命令	1	0x19	
地址	4	address	校验起始地址 (传送地址值需要为绝对地址即: address 0xc2000000)
长度	2	length	填充数据 0x08
数据	8	校验长度	前四字节填需要校验的长度, 后四字节填 0

4.3.2.12 OPCODE_CALC_CRC32_ACK

字段	大小	值	描述
命令	1	0x1a	
地址	4	校验值	Crc32 校验值
长度	2	length	No care

4.3.2.13 OPCODE_UNLOCK_FLASH

字段	大小	值	描述
命令	1	0x1f	
填充	6	Dummy	No care

4.3.2.14 OPCODE_UNLOCK_FLASH_ACK

字段	大小	值	描述
命令	1	0x20	
填充	6	Dummy	No care

4.3.2.15 OPCODE_LOCK_FLASH (锁定地址 0 – 0x5ffff)

字段	大小	值	描述

命令	1	0x21	
填充	6	Dummy	No care

4.3.2.16 OPCODE_LOCK_FLASH_ACK

字段	大小	值	描述
命令	1	0x22	
填充	6	Dummy	No care

4.3.2.17 OPCODE_WRITE_RAM

字段	大小	值	描述
命令	1	0x04	
地址	4	address	RAM 地址
长度	2	length	数据长度
数据	length		数据内容

4.3.2.18 OPCODE_WRITE_RAM_ACK

字段	大小	值	描述
命令	1	0x05	
地址	4	address	RAM 地址
长度	2	length	数据长度

4.3.2.19 OPCODE_EXECUTE_CODE

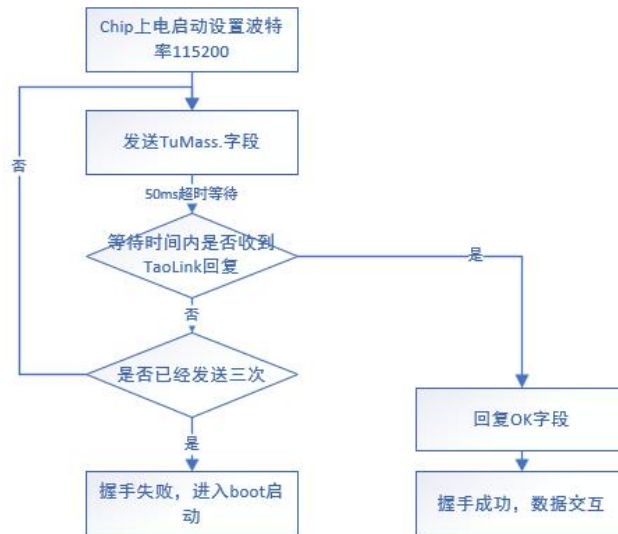
字段	大小	值	描述
命令	1	0x15	
地址	4		程序段入口
填充	2	0	填充数据 0

4.3.2.20 OPCODE_EXECUTE_CODE_END

字段	大小	值	描述
命令	1	0x17	
填充	6	Dummy	No care

4.4 实现细节

4.4.1 握手交互部分



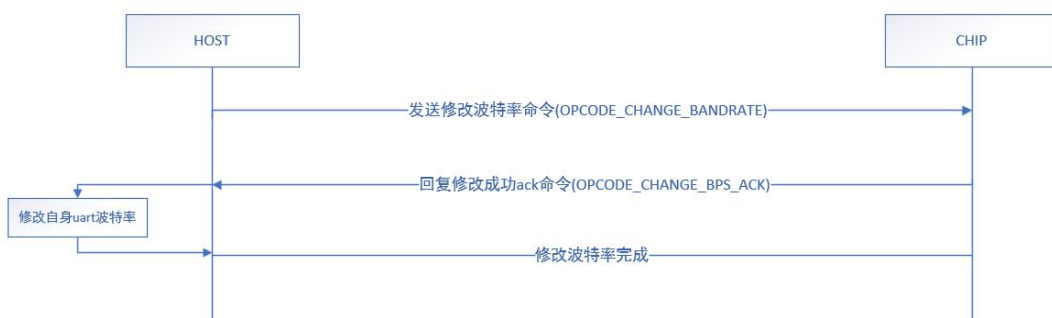
4.4.2 烧录交互部分

定义参数交互结构体

```

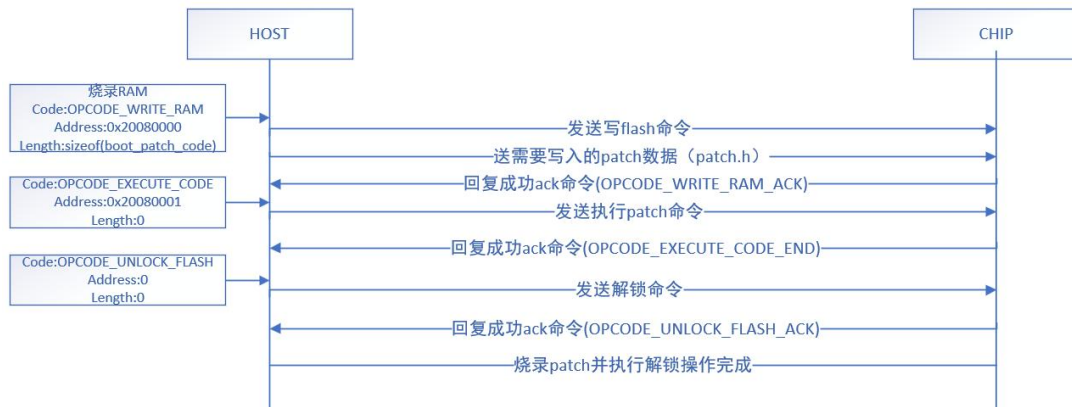
struct boot_param_header_s {
    uint8_t code;
    uint32_t address;
    uint16_t length;
}__attribute__((packed));
  
```

4.4.2.1 修改波特率, 提升读写速度



4.4.2.2 下载 patch 并解锁 flash



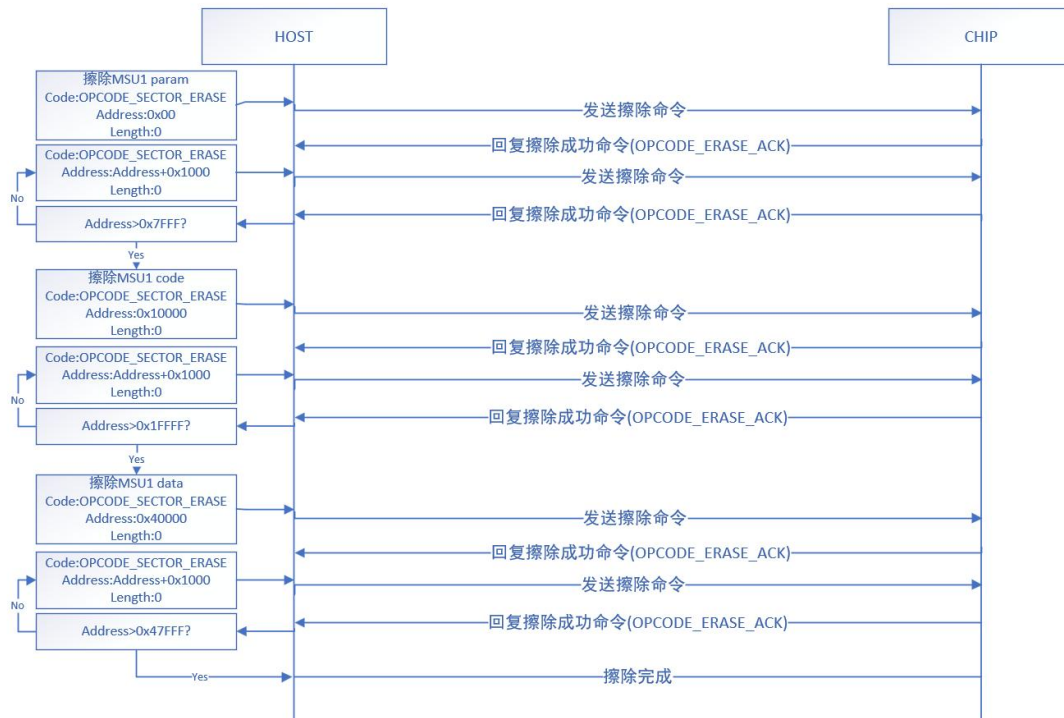


4.4.2.3 擦除操作

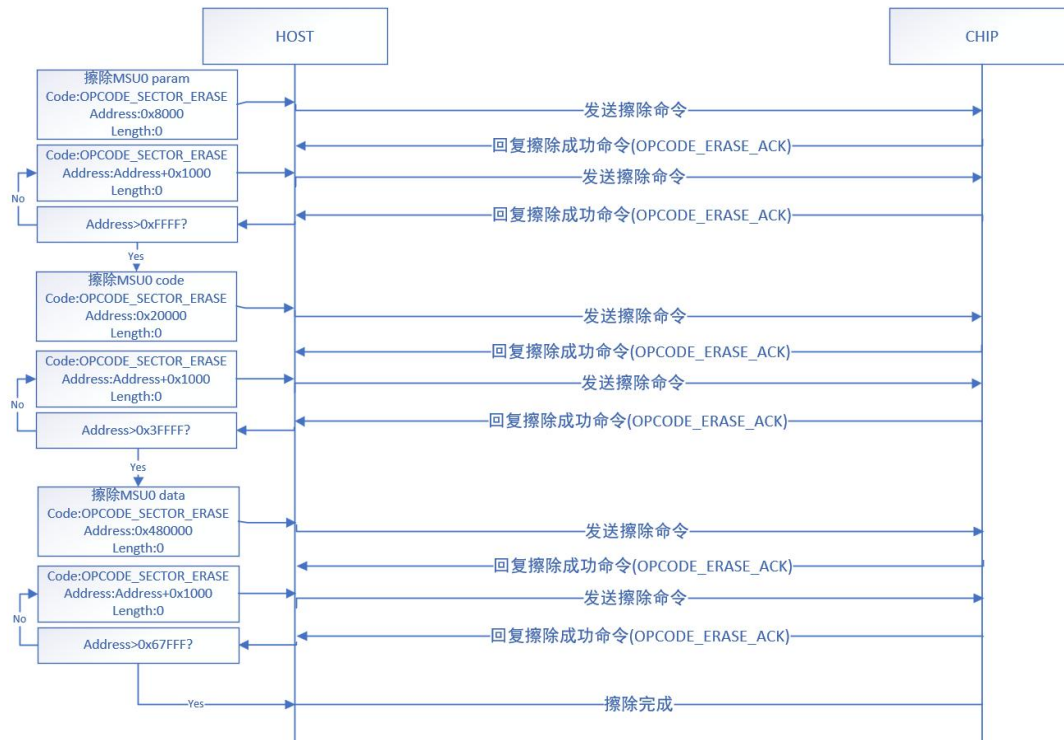
	MSU1			MSU0		
	Param flash 地址	Code flash 地址	Data flash 地址	Param flash 地址	Code flash 地址	Data flash 地址
起始	0x00	0x10000	0x40000	0x8000	0x20000	0x48000
结束	0x7FFF	0x1FFFF	0x47FFF	0xFFFF	0x3FFFF	0x67FFF
Size	32K	64K	32K	32K	128K	128K

按照分区规划，每次以 4K 为单位，循环向被烧录端送擦除命令及起始地址。

4.4.2.3.1 擦除 MSU1 相关需要烧录区域



4.4.2.3.2 擦除 MSU0 相关需要烧录区域(不更新 MSU0 则不执行)



4.4.2.4 MSU1、MSU0 参数烧录

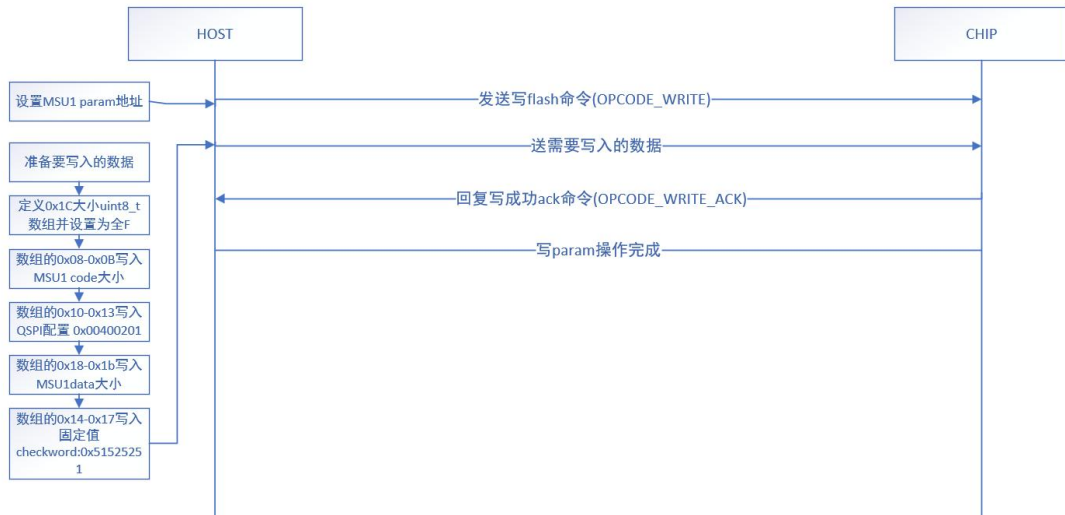
4.4.2.4.1 MSU1 参数烧录

此部分为固定参数，直接写入。

填入参数说明：MSU1 编译生成的 hex 镜像前 64K 为 CODE，后 32K 为 DATA。

MSU1 code 大小：hex 镜像中前 64K 的最后 4 个 byte；

MSU1 data 大小：hex 镜像中后 32K 的最后 4 个 byte。



4.4.2.4.2 MSU0 参数烧录（不更新 MSU0 则不执行）

此部分为固定参数，直接写入。

填入参数说明：MSU0 编译生成的 hex 镜像前 128K 为 CODE，后 128K 为 DATA。

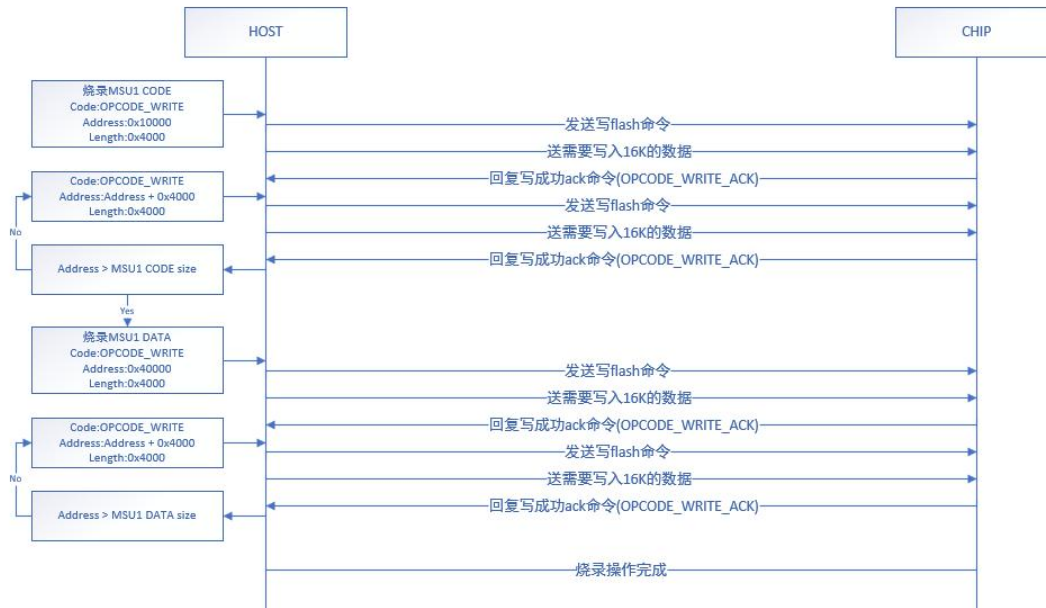
MSU0 code 大小：hex 镜像中前 128K 的最后 4 个 bytes；

MSU0 data 大小：hex 镜像中前 128K 的最后 4 个 bytes。



4.4.2.5 MSU0、MSU1 hex 镜像烧录

4.4.2.5.1 烧录 MSU1 CODE、MSU1 DATA

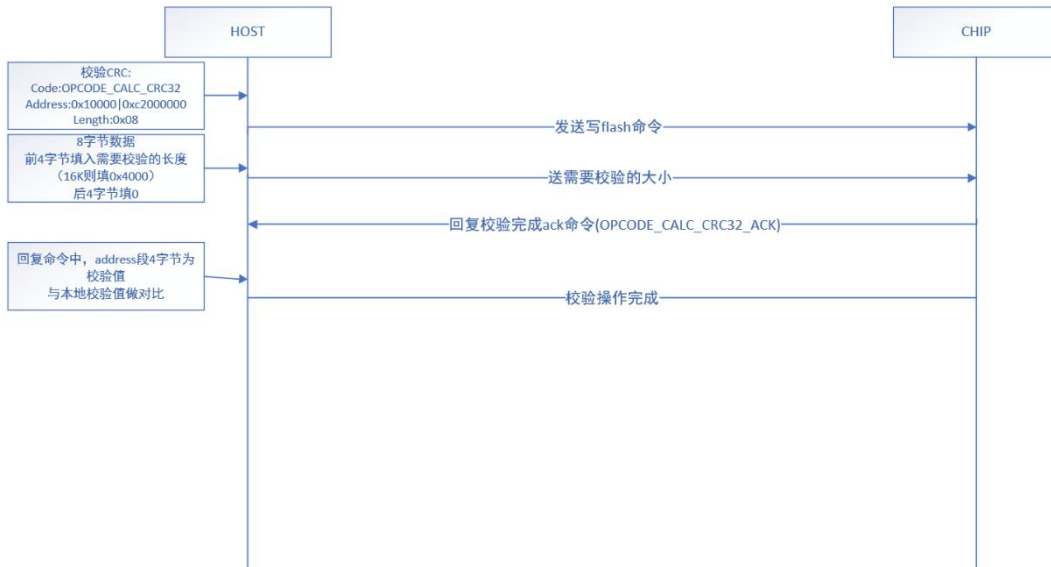


4.4.2.5.2 烧录 MSU0 CODE、MSU0DATA（不更新 MSU0 则不执行）

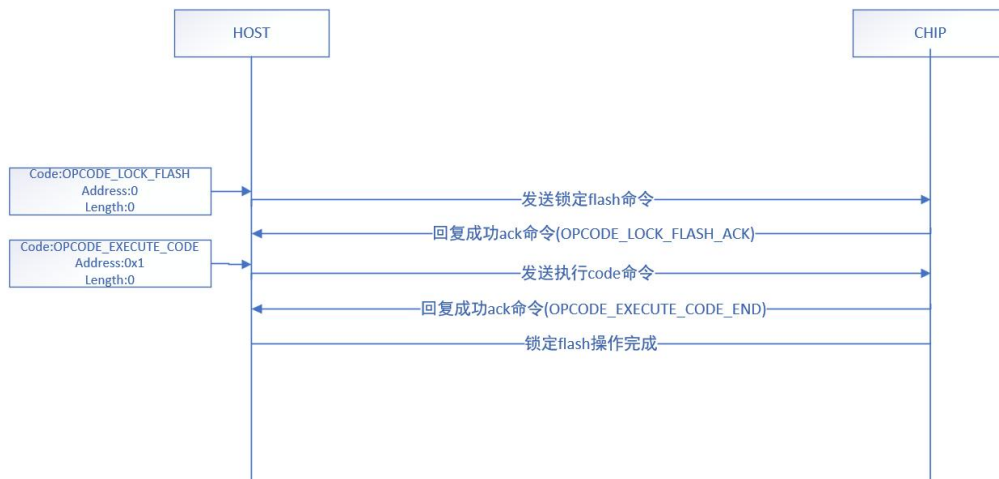


4.4.2.6 CRC 校验（可在每段烧录完成以后进行烧录校验）

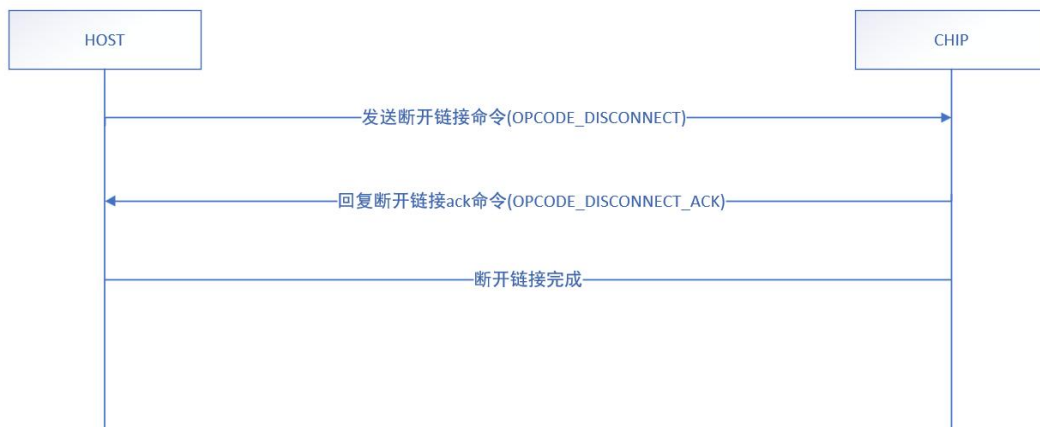
下面以烧录 MSU1 code 校验为例进行说明。



4.4.2.7 锁定 flash 操作



4.4.2.8 断开链接



4.5 注意事项

无