



应用笔记

CW32 自举程序中使用的 ISP 协议

版本号: Rev 1.0



前言

CW32 微控制器片上 FLASH 存储器有一部分区域用于存储 BootLoader 启动程序，在芯片出厂时已编程，用户可利用 BootLoader 启动程序提供的 ISP 模式，通过 UART 串口方便地实现对 CW32 微控制器片上 FLASH 主存储器的擦除和烧写。

本应用笔记将介绍如何进入 CW32 微控制器 ISP 模式，以及所使用的 ISP 协议，并详细介绍支持的每个命令。

目录

前言 1

1 目标芯片进入 ISP 模式的方法 3

2 ISP 模式工作流程 4

3 ISP 通信协议格式 5

 3.1 数据传输要求 5

 3.2 数据格式 5

 3.3 应答标志 6

 3.4 收发数据示例 6

4 ISP 模式命令集 7

5 版本信息 9

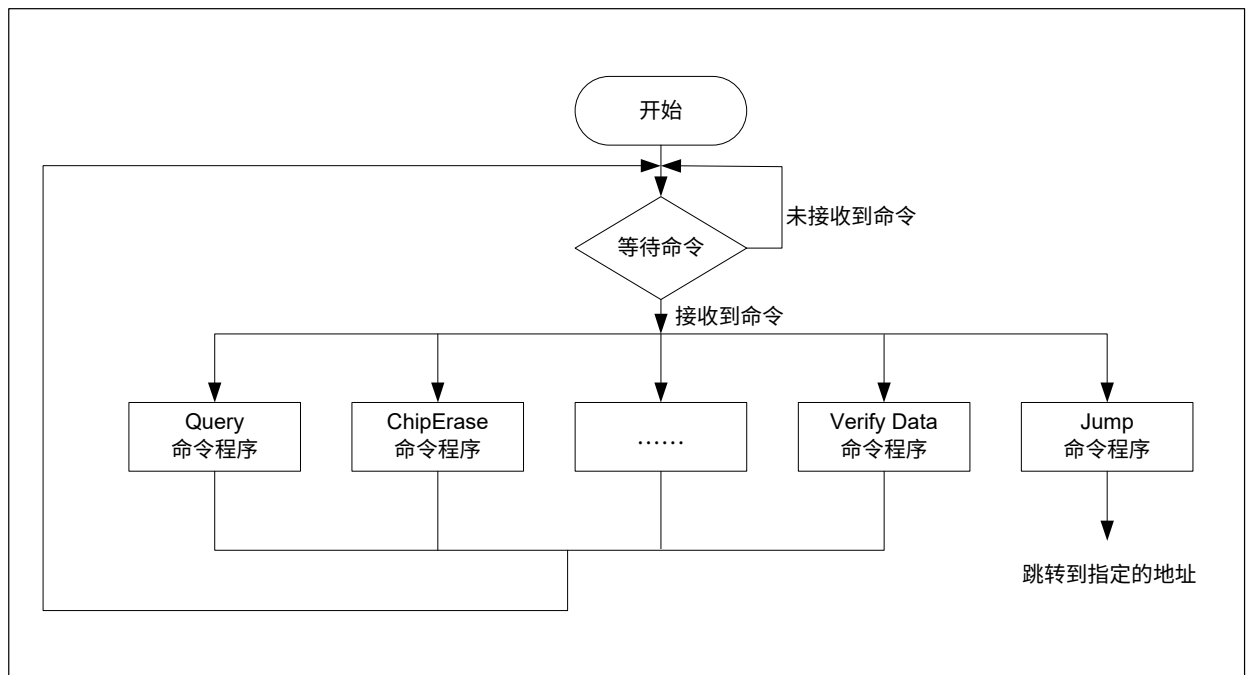


1 目标芯片进入 ISP 模式的方法

- 芯片有 BOOT 脚的流程
 - 使芯片处于 RESET 状态
 - 向芯片的 BOOT 引脚提供高电平
 - 释放芯片的 RESET 状态
 - 芯片进入 ISP 模式
- 芯片无 BOOT 脚的流程
 - 使芯片处于 RESET 状态
 - 向芯片的 RXD（SWDIO）提供 50KHz 的方波
 - 释放芯片的 RESET 状态并延时 5ms
 - 芯片进入 ISP 模式

2 ISP 模式工作流程

图 2-1 ISP 模式工作流程图



当 CW32 微控制器进入 ISP 模式后，系统将等待串口接收命令。系统接收到命令后，将根据命令类型执行相应的程序操作。

3 ISP 通信协议格式

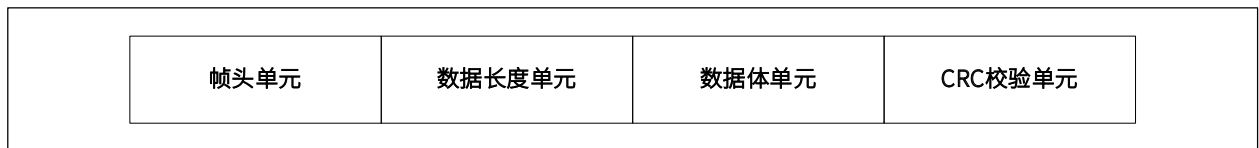
3.1 数据传输要求

采用异步半双工通信方式，8 个数据位、1 个停止位、无校验位，初始速率为 115200BPS。

3.2 数据格式

协议以数据帧的形式进行交互，一个完整的数据帧由帧头单元、数据长度单元、数据体单元、CRC 校验单元 4 部分组成，如下图所示：

图 3-1 数据帧格式



- 帧头单元
1 字节长度，表示一个数据帧的开始，固定为 16 进制数 0x65。
- 数据长度单元
1 字节长度，表示数据体单元有多少个字节，取值范围为 0-255。
- 数据体单元
长度不固定，为实际的应用层数据 / 指令。
- 校验单元
2 字节长度，为帧头单元、数据长度单元及数据体单元所有数据的校验值。采用 CCITT 推荐的 16 位的 CRC-16/X25 算法 $x^{16}+x^{12}+x^5+1(0x1021)$ ，生成 2 字节的 CRC 校验和（低字节收发在前，高字节收发在后）。发送方必须根据要发送的数据生成 2 字节的 CRC 校验和，接收方收到完整的数据帧后，根据接收的数据生成新的 CRC 校验和，如果新的 CRC 校验和与收到的校验和相等则表明该数据帧有效，否则向发送方回送“校验错”的应答。

3.3 应答标志

本协议采用半双工方式通讯，作为命令的主动发起方，需要收到被动接收方返回的应答标志后，才可进行后续的操作。接收方发送应答的数据体单元的第一个字节为应答标志。应答标志编码如下表所示：

表 3-1 应答标志编码定义

编码	含义	备注
0x00	成功	
0x80	校验错	需要重发该命令
0x90	指令不支持	
0x91	参数不支持	
0x92	没有读权限	
0x93	没有写权限	
0x94	没有擦权限	
0x95	没有 Verify 权限	
0x96	没有跳转权限	
0x98	写数据到 Flash 失败	
0x99	BlankCheck 失败	

3.4 收发数据示例

收发数据示例：

发送方：0x65 0x01 0x10 0x65 0xF3

接收方：0x65 0x09 0x00 0x18 0x00 0x08 0x00 0x01 0x01 0x06 0x00 0xBA 0x2B

其中黄色为帧头单元，蓝色为数据长度单元，绿色为数据体单元，灰色为 CRC 校验单元。接收方的数据体单元的第二个字节为应答标志。



4 ISP 模式命令集

下面的表 4-1 列出了支持的 ISP 命令：

表 4-1 ISP 命令集

指令名称	指令格式	指令说明
Query	0x10	查询芯片的型号、自举程序的版本、UCLK 的时钟频率。 返回：< 应答标志 + UCLK + BootLoaderId + ChipName>。 UCLK 为 2 个字节，低字节先发，以 MHz 为单位。 BootLoaderId 为 2 个字节，低字节先发。 ChipName 为芯片商业型号的 ASCII 码，不定长。
PPS	0x11 DIVN	更改芯片的波特率为 UCLK / DIVN。 返回：< 应答标志 >。 DIVN 为 2 个字节，低字节先发。 例：当 UCLK 的时钟频率为 6MHz 时，要设置波特率为 115200bps，则 DIVN=6000000/115200=52，DIVN 为 0x34 0x00
Set BaseAddr	0x20 0x00 0x00 BasedAddr	设置读写操作的基地址 (32bit)。 返回：< 应答标志 >。 BasedAddr 为 4 个字节，低字节先发。 0x000xxxxx 代表代码存储区。 0x2000xxxx 代表 RAM 区。
FLASH_Blank Check	0x22	检查 FLASH 区域是否全为 0xFF。 返回：< 应答标志 >。
FLASH_ChipErase	0x24	擦除整个 FLASH 区域。 返回：< 应答标志 >。
FLASH_ SectorErase EE_SectorErase	0x26 Offset	擦除 BaseAddr + Offset 地址所在的页面。 返回：< 应答标志 >。 Offset 为 2 个字节，低字节先发。
Write Data	0x28 Offset Data1-DataN	将 N 个字节的数据写入 BaseAddr + Offset 处，并验证读出与写入的数据是否相同 (N = 1 - 248)。 返回：< 应答标志 >。 Offset 为 2 个字节，低字节先发。 注 1：根据地址范围将数据写入 Flash 或 RAM 或 EE 或 OTP。 注 2：当 BaseAddr+Offset 与待编程字节数均为 Word 对齐时，调用 Word Program 操作，否则调用 Byte Program 操作。Word Program 仅需 Byte Program 一半的时间。 注 3：当向 EE 写入数据时，数据必须与页面的起始地址对齐且数据长度必须是页面大小的整数倍。向 EE 写入数据前需要先擦除相应区域。
Read Data	0x29 Offset Cnt	从 BaseAddr + Offset 处，读出 CNT 个字节。 返回：< 应答标志 + CNT 个字节的数据 >。 Offset 为 2 个字节，低字节先发。 Cnt 为 1 个字节，取值范围为 0-255。 注：读取特定地址可以获取商业编号、Flash 的容量、Ram 的容量、FlashPageSize、引脚数量。详见器件用户手册的数字签名章节。

指令名称	指令格式	指令说明
Verify Data	0x2A Offset ByteCnt	计算从 BaseAddr + Offset 开始的 ByteCnt 个数据的 CRC 值。 返回：< 应答标志 +CRC 校验值 >。 Offset、ByteCnt、CRC 均为 2 个字节，低字节先发。 ByteCnt 取值范围为 8 - 65535。
GetPreprocess Message	0x2C	获取 FLASH 预处理信息，该指令由客户编写的预处理 HEX 响应。 返回：< 应答标志 + Addr + Data1-DataN> 或 仅返回应答标志。 Addr 为 4 个字节，低字节先发。 Data1-DataN 最多为 64 字节。 编程器需要将 Addr 及 Data1-DataN 保存在内存中。 注：本命令适用于编程器，芯片不支持。
WritePreprocess Message	0x2D	编程器需要将内存中的 Addr 及 Data1-DataN 写入 Flash。 返回：< 应答标志 >。 Addr 及 Data1-DataN 由 GetPreprocessMessage 指令获取。 注：本命令适用于编程器，芯片不支持。
SetFlashReadOut Level	0x30 RdLevel	设置芯片的读保护等级。 返回：< 应答标志 + RdLevel>。 RdLevel 为 0x00 代表 Level0； RdLevel 为 0x01 代表 Level1； RdLevel 为 0x02 代表 Level2； RdLevel 为 0x03 代表 Level3； RdLevel 为 0x55 则仅返回当前的保护等级。 Level0：SWD 可读写 FLASH，ISP 可读写 FLASH； Level1：SWD 可降级，ISP 可降级； Level2：SWD 端口物理断开，ISP 可降级； Level3：SWD 端口物理断开，ISP 端口物理断开，无法再次烧录。 注：部分芯片仅支持 Level0 和 Level2。
Jump	0x40 0x00 0x00 Addr	跳转到 Addr 所指定的地址执行程序。 Addr 为 4 个字节，低字节先发。 注：Addr 只能是 0x2000xxxx 或 0x00000000。

5 版本信息

表 5-1 文档修订信息

日期	版本	变更信息
2021-10-14	Rev 1.0	初始发布