

摘要

本应用笔记旨在介绍 CS8M320 UART 通信方式 IAP 升级设计原理，指导用户快速完成 IAP 功能开发和调试。

适用范围

类型	适用产品型号或系列	说明
	CS8M320F3V6Nx	

版本

历史版本	修改内容	日期
Rev 0.1	初版	2023.09.27
Rev0.2	删除数据加密、密钥功能，修改 ROM、RAM 图	2023.10.11
Rev0.3	修改上电流程框图、增加 AP 完整性校验说明	2023.10.15
Rev0.4	增加 ROM 用户信息存储区，增加升级流程交互图示	2023.10.18
Rev0.5	修改 AP 程序注意事项、IAP 升级示例	2023.11.07
Rev0.6	修改文档格式，修改 AP 程序注意事项	2023.11.29

目 录

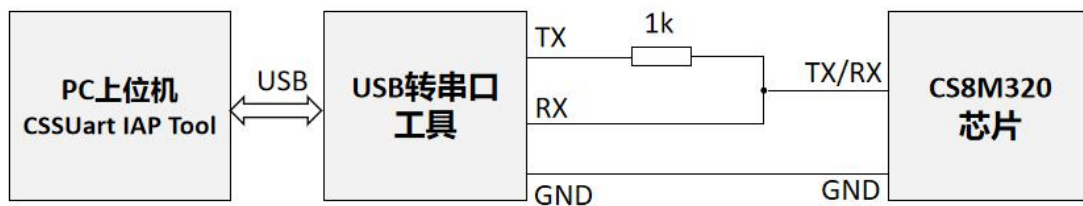
1 概述.....	4
2 升级工具介绍.....	4
3 芯片空间分配.....	5
3.1 ROM.....	5
3.2 SRAM.....	7
4 芯片工作流程.....	8
4.1 上电流程.....	8
4.2 升级流程.....	8
5 通信协议.....	11
5.1 协议帧格式.....	11
5.2 协议指令.....	11
5.2.1 命令码.....	11
5.2.2 响应状态.....	11
5.2.3 执行状态.....	12
5.3 通信数据.....	12
5.3.1 Get Version 命令.....	12
5.3.2 Write Memory 命令.....	13
5.3.3 Read Memory 命令.....	13
5.3.4 Erase Chip 命令.....	13
5.3.5 ROM Check 命令.....	14
5.3.6 异常回复.....	14
6 BOOTLOADER 注意事项.....	15
7 AP 程序注意事项.....	15
8 IAP 升级示例.....	17
8.1 烧录器烧录.....	17
8.2 IAP 串口升级.....	18

1 概述

为满足 CS8M320 芯片固件刷新的应用需求，本文档设计了一种 IAP 升级示例方案，采用芯海科技私有协议通过 UART 单线通信方式对用户程序进行升级刷新。该文档对示例方案的升级流程、通信协议以及设计要点进行了说明，可供用户或开发者参考。

2 升级工具介绍

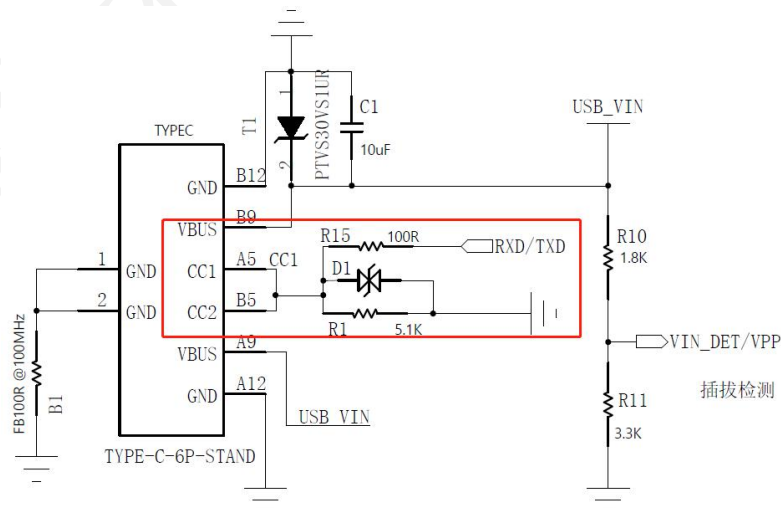
IAP 升级用到的软硬件工具如下：



CS8M320 的 UART0 模块具备 TX/RX 引脚交换功能，据此实现与上位机或其他主控 MCU 的 UART 单线通信。

USB to TTL 串口工具的 TX 脚需外接 1k 电阻再与自身 RX 脚和芯片短接，模拟单线半双工通信，否则会出现通信异常。

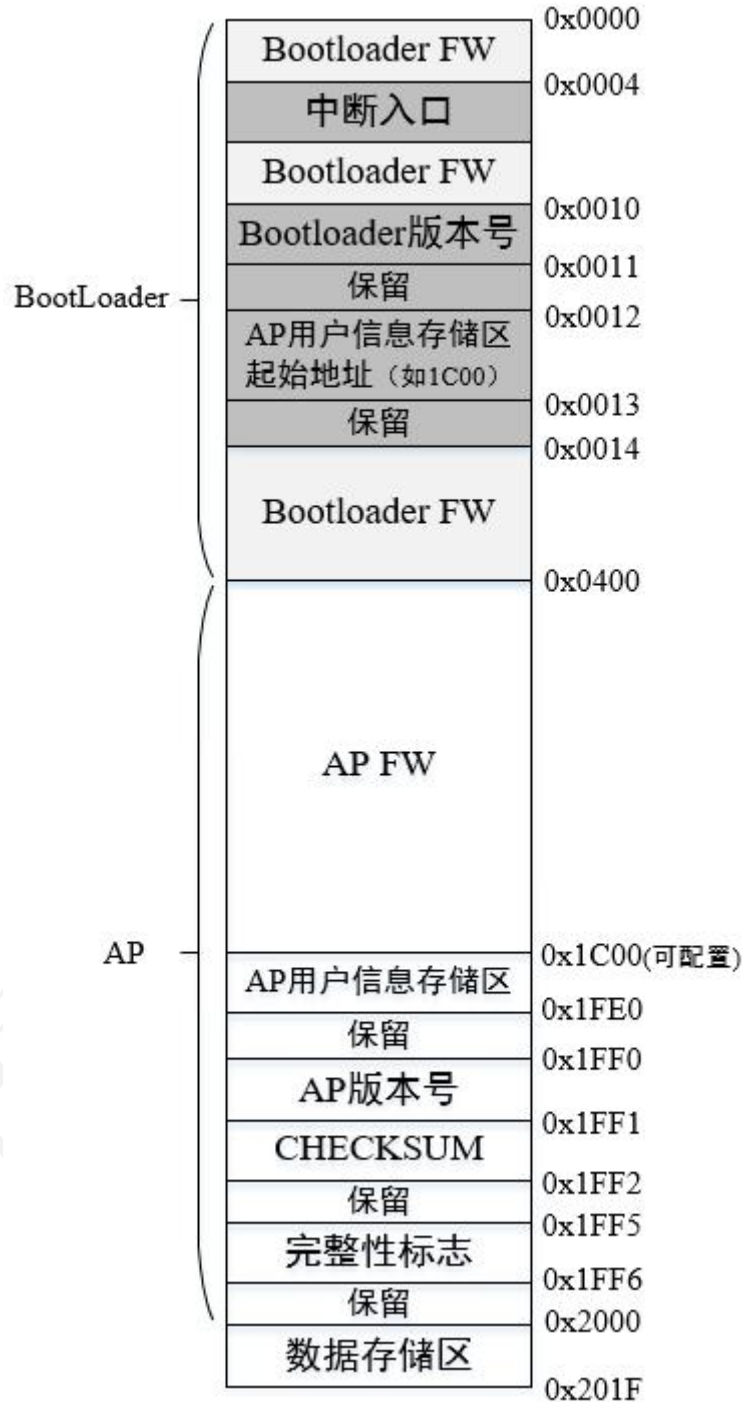
为了实现 USB 带电升级，可参照下图进行硬件设计，将芯片的串口引脚 PT3.1 与 TYPEC 母座的 CC1、CC2 短接。产品外部再借助 TYPEC 公头连接芯片与串口工具：串口工具引出的通信线与 TypeC 公头的 CC1 或 CC2 线连接。



3 芯片空间分配

3.1 ROM

CS8M320 内置 8K×16bit Flash，根据 IAP 需求可将 ROM 空间划分如下：



BootLoader 程序占据 ROM 前 1k Word 空间 (0x0000~0x03FF)，用户 AP 使用后 7k Word 空间 (0x0400~0x1FFF)。芯片中断入口地址固定为 ROM 的 0x0004，位于 BootLoader 区域。

此外，将 ROM 的 0x0010~0x0013 地址设为保留区域，其中 0x0010 地址处存放 BootLoader 版本号，0x0012 地址处存放用户信息存储区的起始地址；将 AP 区域最后一页（ROM 的 0x1FE0~0x1FFF 地址处）设为保留区域，其中 0x1FF0 地址用于存放 AP 版本号，0x1FF1 地址用于存放 Checksum 信息，0x1FF5 用于存放芯片 AP 程序的完整性标志。

注：

❖ Checksum 是由上位机对准备升级的 bin 数据进行累加和计算得到，芯片端不进行复验，只作为信息保存。注意，CSU-IDE 中生成的 Checksum 采用 CRC16 的计算方式，为了节省资源，该 IAP 上位机采用累积和方式，故两种 Checksum 数据不相同。

❖ 完整性标志是 BootLoader 判断 AP 程序完整的必要条件。升级过程中，BootLoader 顺利写到 Flash 最后一页 0x1FF5 地址处时，会往该地址填充 0x5A00。升级起始会对 Flash 的整个 AP 区域进行擦除，若升级过程中出错，Flash 的 0x1FF5 地址处并非 0x5A00。由 BootLoader 在跳入 AP 程序前，会对校验完整性标志，若 Flash 的 0x1FF5 地址处非 0x5A00，则不会进行跳转，继续等待指令帧。

AP 区域的用户信息存储区用于存放应用程序的重要数据，在升级过程中不会被 BootLoader 擦除和改写。位域 BootLoader 区域的 ROM 的 0x0012 地址处定义了用户信息存储区的起始地址。以起始地址 ROM 0x1C00 为例，ROM 的 0x1C00~0x1FDF 区域为用户信息存储区。

ROM 区域的 0x2000~0x201F 是额外的数据存储区。升级过程中，BootLoader 在接收到写 flash 最后一页（0x1FE0~0x1FFF）指令时，会暂时将接收数据缓存至该数据存储区。待 BootLoader 接收到校验指令时，会对 AP 区域 0x0400~0x1FFF 数据进行校验（与上位机发送的校验数据比对），校验成功后再执行 Flash 最后一页的写操作：将 0x2000~0x200F 地址处的数据复制到 0x1FE0~0x1FEF；将 0x2010~0x2011 地址处缓存的 AP 版本号和 Checksum 信息复制到 0x1FF0~0x1FF1。前述过程都执行成功后，再往 0x1FF5 处填充完整性标志 0x5A00。

注：

❖ 校验指令中 ROM 区校验值计算方法：

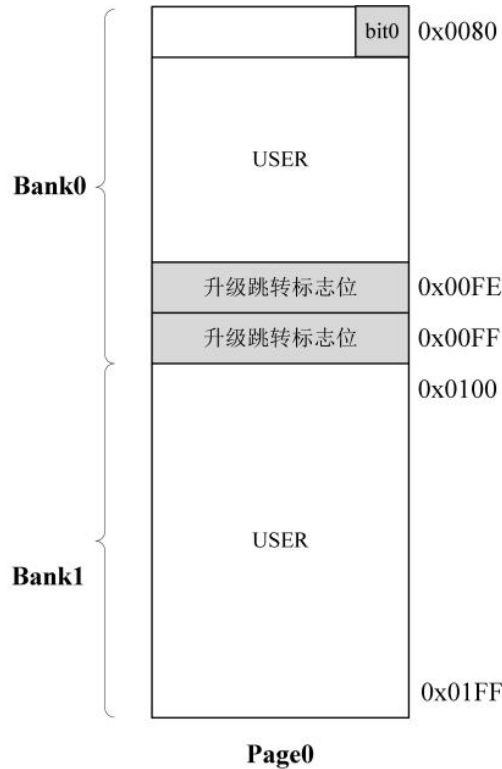
◇ 上位机：对待升级的 bin 0x0400~0x1FDF 地址处数据进行累加，一个字节存储累加和，不计溢出。

◇ BootLoader：对 ROM 区 0x0400~0x1BFF 地址处数据进行累加，0x1C00~0x1FDF

地址处强制使用 0xFFFF 参加累加(上位机计算校验时, AP 程序的该区域数据为默认的全 F), 一个字节存储累加和, 不计溢出。

3.2 SRAM

在 AP 程序运行和 IAP 升级过程中还需要使用到部分 SRAM 空间, 相关位置如下:



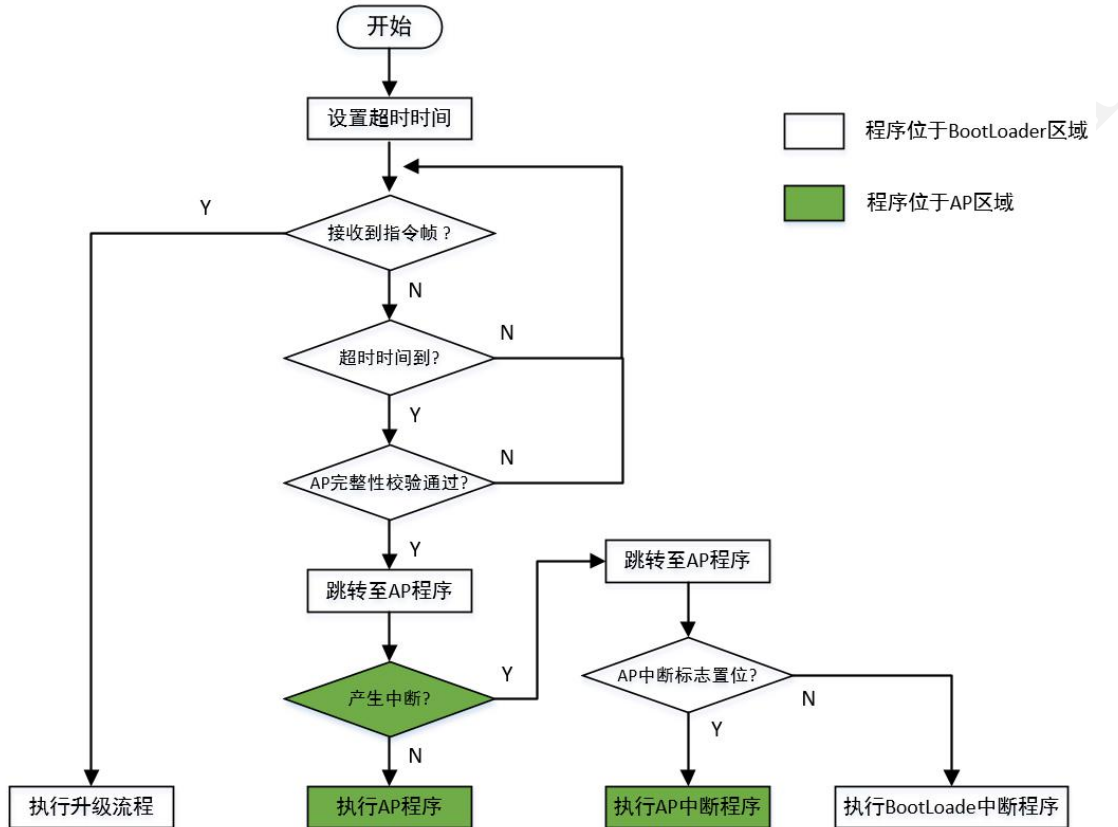
其中, SRAM 的 0x0080 地址处 bit0 用于指示中断入口标志。实际上芯片只有一个硬件中断入口, 地址是 ROM 的 0x0004, 位于 BootLoader 区域中, 在 BootLoader 中断服务函数里将会对该中断入口标志进行判断, 以决定是否跳转至用户中断程序执行;

SRAM 的 0x00FE、0x00FF 地址处数据用以指示是否为 AP 跳转至 BootLoader, AP 跳回 BootLoader 前, 会将 SRAM 的 0x00FE 地址处填充 0x55、SRAM 的 0x00FF 地址处填充 0xAA。BootLoader 程序将据此延迟超时时间来保证可以正确接收到上位机指令。

4 芯片工作流程

4.1 上电流程

芯片正常上电或复位后，将由 BootLoader 引导启动，引导流程如下：



当由 AP 跳回 BootLoader 或 BootLoader 接收过上位机数据，BootLoader 超时时间为 4s，否则 BootLoader 超时时间为 30ms。

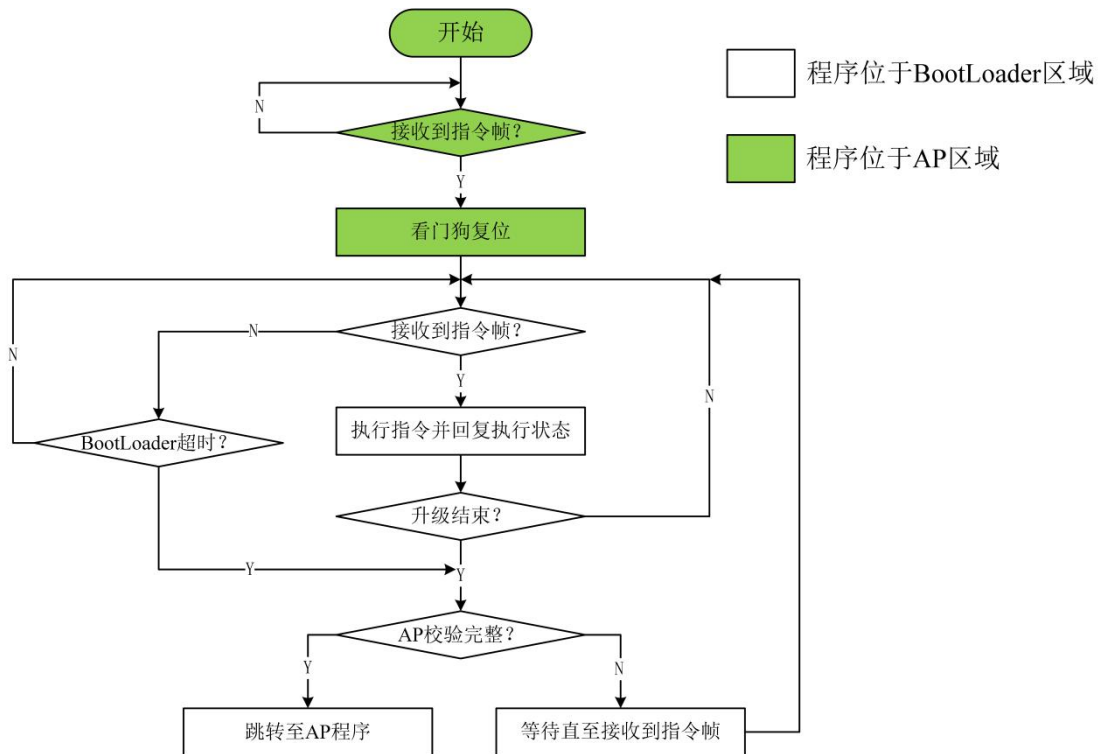
若无升级指令下发且 AP 完整则跳转至用户 AP 执行，AP 完整性校验失败后将不会再次进行校验，直至再次接收到指令帧。

其中未执行过升级流程时 AP 完整性校验的方法为：校验 ROM 区 0x1FF5 地址处数据是否为 0x5A00。

4.2 升级流程

所有的升级流程都由 BootLoader 完成，用户可采用重新上电、硬件复位、AP 看门狗复

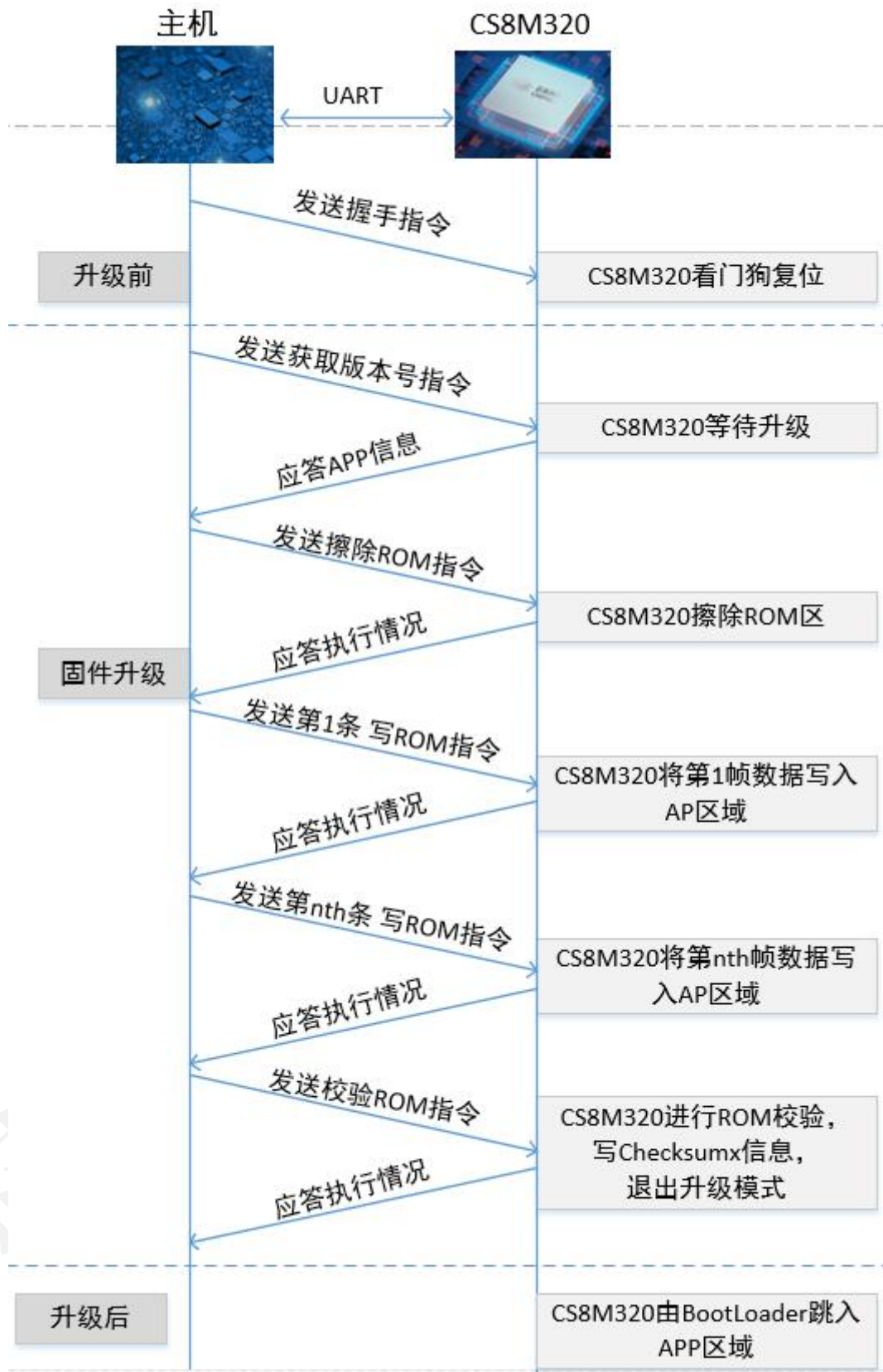
位等方式进入 BootLoader，下图为通过 AP 看门狗复位方式进入 BootLoader 并执行升级流程的流程图：



其中 AP 完整性校验的手段包括：

- ❖ 执行上位机校验命令过程中，当 ROM 区数据校验失败，则 AP 完整性校验失败；
- ❖ 执行上位机校验命令过程中，当烧录 ROM 最后一页出错，则 AP 完整性校验失败；
- ❖ BootLoader 超时后，判断 ROM 区 0x1FF5 地址处的完整性标志数据是否为 0x5A00，若不是，则 AP 完整性校验失败。

下图展示了正常工作无错误时的升级流程：



5 通信协议

IAP 升级方案采用芯海科技私有协议进行信息传输，以下对协议内容进行详细说明。

5.1 协议帧格式

协议帧根据传输方向可分为两类，第一类为上位机下发至芯片端的指令帧，其帧格式如下表所示：

Head	Length	Command	Data	CKL	CKH
0x55	命令码+数据长度	命令码	数据	帧校验码低位	帧校验码高位

第二类协议帧为芯片端上传至上位机的响应帧，其帧格式如下表所示：

ACK	Head	Length	Execute	Data	CKL	CKH
响应状态	0x55	执行状态+数据长度	执行状态	数据	帧校验码低位	帧校验码高位

协议帧数据长度最大为 64Bytes，最小可为 1Byte，即协议帧可不包含 Data 段，采用累加和 不计溢出的方式校验指令帧，帧校验码仅高位有效，低位可作为随机数。

本方案中 ROM 数据传输采用小端模式，即先传输地址低位，再传输地址高位，协议帧数据长度最大为 69Bytes，一帧数据需在 50ms 内下发完成。

5.2 协议指令

5.2.1 命令码

IAP 支持的命令码列表如下表所示：

命令	命令码	描述	响应	备注
Get Version	0x21	获取 BootLoader、AP 版本号等信息	Yes	
Write Memory	0x23	写 ROM 区域	Yes	
Read Memory	0x24	(调试使用,发布版本剔除)读取 ROM 区域	Yes	
Erase Chip	0x26	擦除整个 ROM 区域	Yes	
ROM Check	0x28	校验整个 ROM 区域	Yes	

ROM 区校验采用按字节累加和 不计溢出的方式，同一地址处低位首先参与运算。

5.2.2 响应状态

响应状态是指目标芯片端对接收到的数据的解析结果，以向上位机反馈协议帧是否下发

正确，并对错误类型进行分类。

响应状态	描述	备注
0xF0	数据包接收正确 (ACK)	
0xE1	包头错误	
0xD2	包校验错误	
0xC3	包长度错误, 为 0	
0xB4	包长度错误, 超过最大允许长度	
0xA5	未知错误	

5.2.3 执行状态

执行状态用于指示目标芯片端对接收到的指令的执行情况，上位机可据此决定后续下发的指令。

命令状态	描述	备注
0x00	命令执行成功	
0x01	无效或不支持的命令	
0x02	命令参数错误	如擦写地址超限等
0x04	命令执行失败	如协议帧校验失败等

5.3 通信数据

5.3.1 Get Version 命令

上位机下发数据:

Byte0	Byte1	Byte2	Byte3	Byte4
Head	Length	Command	CKL	CKH
0x55	0x01	0x21		

芯片端回复数据:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11
ACK	Head	Length	Execute	BTVL	BTVH	APVL	APVH	APCSL	APCSH	CKL	CKH
0xF0	0x55	0x07	0x00								

注:

BTVL: BootLoader 版本号低位

BTVH: BootLoader 版本号高位
 APVL: AP 版本号低位
 APVH: AP 版本号高位
 APCSL: AP CHECKSUM 低位
 APCSH: AP CHECKSUM 高位
 作为握手指令。

5.3.2 Write Memory 命令

上位机下发数据:

Byte0	Byte1	Byte2	Byte3-6	Byte7-134	Byte135	Byte136
Head	Length	Command	Addr[0:3]	ROM Data	CKL	CKH
0x55	0x85	0x23				

注:

Addr: 数据起始地址, 从低位至高位依次发送, 本方案中仅低 2bytes 地址有效

ROM Data: ROM 区数据

芯片端回复数据:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
ACK	Head	Length	Execute	CKL	CKH
0xF0	0x55	0x01	0x00		

5.3.3 Read Memory 命令

上位机下发数据:

Byte0	Byte1	Byte2	Byte3-6	Byte7	Byte8
Head	Length	Command	Addr[0:3]	CKL	CKH
0x55	0x05	0x24			

注:

Addr: 数据起始地址, 从低位至高位依次发送, 本方案中仅低 2bytes 地址有效

芯片端回复数据:

Byte0	Byte1	Byte2	Byte3	Byte4-131	Byte132	Byte133
ACK	Head	Length	Execute	ROM Data	CKL	CKH
0xF0	0x55	0x81	0x00			

注:

ROM Data: ROM 区数据

5.3.4 Erase Chip 命令

上位机下发数据:

Byte0	Byte1	Byte2	Byte5	Byte6
Head	Length	Command	CKL	CKH
0x55	0x01	0x26		

芯片端回复数据:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
ACK	Head	Length	Execute	CKL	CKH
0xF0	0x55	0x01	0x00		

5.3.5 ROM Check 命令

上位机下发数据:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Head	Length	Command	ROM CK	APCSL	APCSH	CKL	CKH
0x55	0x04	0x28					

注:

ROM CK: ROM 区校验比对结果

APCSL: AP CHECKSUM 低位

APCSH: AP CHECKSUM 高位

芯片端回复数据:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
ACK	Head	Length	Execute	CKL	CKH
0xF0	0x55	0x01	0x00		

5.3.6 异常回复

在帧接收异常或执行异常后, 芯片端可按此格式回复异常情况:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
ACK	Head	Length	Execute	CKL	CKH
	0x55	0x01			

上位机在接收到异常回复后应立即停止发送数据, 并等待 20ms 后才可重新发送数据。

6 BootLoader 注意事项

BootLoader 区域的 ROM 区 0x0012 地址处存放了不被擦除和改写的用户信息存储区的起始地址。如下图所示，用户在此位置做定义。

```

29 | ;=====
30 | ; BootLoader信息
31 | ;=====
32 | CSCC_BT_INFO    .section rom,addr=0x10
33 |     dw          0x0001          ;BootLoader版本号
34 |     dw          0x0000
35 |     dw          0x1C00          ;用户数据存储区起始地址，须为0x20的整数倍
36 |     dw          0x0000
37 | .ends
    
```

7 AP 程序注意事项

AP Demo 程序包括 UART 初始化、UART 接收以及 IAP 协议执行等操作，具体要求及设计要点如下：

(1) “iap_deal.c”、“iap_deal.h” 文件为 IAP 协议执行文件，利用看门狗复位的方式从 AP 跳转至 BOOT 区，用户在工程中直接添加这两个文件即可。

(2) 系统初始化时需调用 iap_init() 函数，初始化 UART、WDT 等模块、初始化 IAP 参数，其中 UART 通信口为 PT3.1 (RX) 波特率为 19200。其余 CPUCLK、总中断等配置依用户程序。

```

40 | void iap_init(void)
41 | {
42 |     GIE = 0;
43 |     CPUCLK_4M();
44 |
45 |     wdt_init();          // IAP所需程序 ,看门狗初始化
46 |     PT3CON_1 = 0;       // IAP所需程序 ,PT3.1设为数字口, 做UART管脚
47 |     uart_init();        // IAP所需程序 ,UART初始化
48 |
49 |     PT1EN_5 = 1;        // AP Demo应用PT1.5为输出口
50 |     timer0_init();
51 |
52 |     iap_data_count = 0; // IAP所需程序 ,参数初始化
53 |     iap_frame_cs_calculate = 0; // IAP所需程序 ,参数初始化
54 |     iap_jump_flag = 0; // IAP所需程序 ,参数初始化
55 |     APToBoot0 = 0;      // IAP所需程序 ,参数初始化
56 |     APToBoot1 = 0;      // IAP所需程序 ,参数初始化
57 |     flag_INT = 1;      // IAP所需程序 ,参数初始化决定中断入口在BootLoc
58 |
59 |     GIE = 1;
60 | }
    
```

(3) 在主程序中需调用 `iap_reset_judge()`函数以判断是否进行 IAP 升级;

```

44 while(1)
45 {
46     asm("clrwdt"); //IAP所需程序
47     iap_reset_judge(); //IAP所需程序
48 }
    
```

(4) 在 UART 接收中断中需调用 `iap_signal_detect()`函数, 此函数不影响用户程序, 只用于监听是否接收到 IAP 握手指令;

```

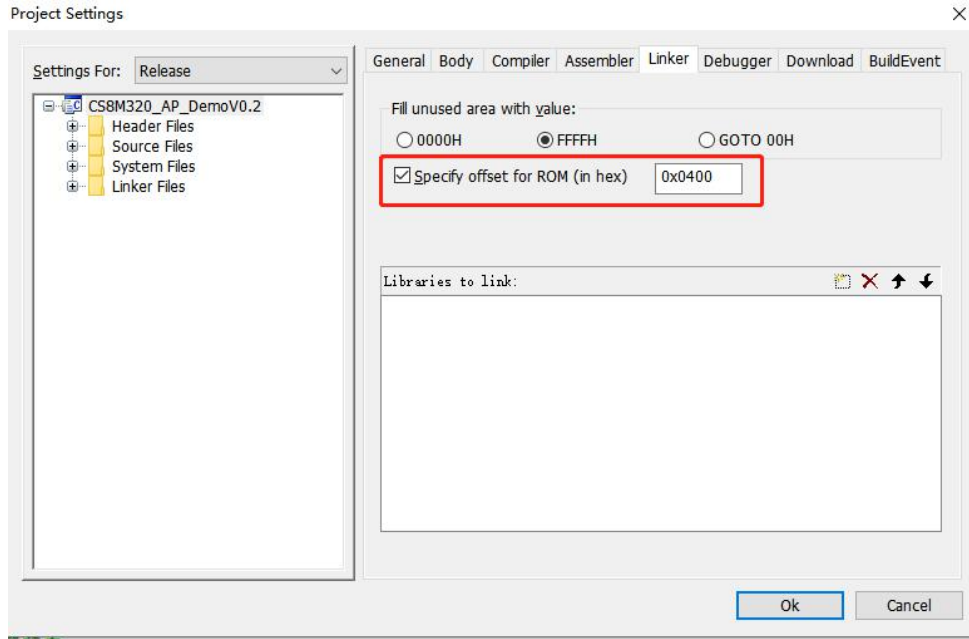
26 if(URO_RNIF & URO_RNIE)
27 {
28     URO_RNIF = 0;
29     uart_read_temp = URO_RX_REG;
30     iap_signal_detect(uart_read_temp); //用户只需在接收到数据后调用此函数即可
31 }
    
```

(5) 1FF0h~1FFFh 为保留空间, 其中 1FF0h 为 AP 版本号, 用户可使用版本号对软件版本进行有效管理; 1FF1h 保存上位机发送 bin 的 checksum 信息, 升级成功时 BootLoader 将填充该地址, 用户无需关注; 1FF5h 为升级方式标志位, 用以区分烧录器烧录和 IAP 升级, 在使用不同方式刷新固件时, 用户需修改此处数值: 利用烧录器烧录 BootLoader 与 AP 的合并 hex 时, 1FF5h 地址处写 0x5A00; 用于 IAP 升级文件时, 1FF5h 地址处写 0x0000。

```

61 /*<!-- IAP所需程序 -----*/
62 asm("org 0x1FF0"); //AP信息区
63 asm("dw 0x0001"); //AP版本号 //1FF0
64 asm("dw 0x0000"); //CHECKSUM //1FF1
65 asm("dw 0x0000"); //保留
66 asm("dw 0x0000"); //保留
67 asm("dw 0x0000"); //保留
68 asm("dw 0x0000"); //保留, 用于合并hex时, 此地址写0x5A00, 用于升级文件时, 此地址写0x00h
69 asm("dw 0x0000"); //保留
70 asm("dw 0x0000"); //保留
71 asm("dw 0x0000"); //保留
72 asm("dw 0x0000"); //保留
73 asm("dw 0x0000"); //保留
74 asm("dw 0x0000"); //保留
75 asm("dw 0x0000"); //保留
76 asm("dw 0x0000"); //保留
77 asm("dw 0x0000"); //保留
78 asm("dw 0x0000"); //保留
79 /*<!-- IAP所需程序 -----*/
    
```

(6) AP 生成 hex 文件时需进行偏移。具体设置为:



8 IAP 升级示例

8.1 烧录器烧录

该方式适用于用户批量烧录芯片。

(1) 准备好待拟合的 BootLoader.hex 和 AP .hex（注意 1FF5h 处需填入 5A00h），双击打开 38F20Merge.exe，分别加载好 BootLoader.hex 和 AP .hex，点击合成，确定拟合 hex 输出路径和文件名，点击保存；

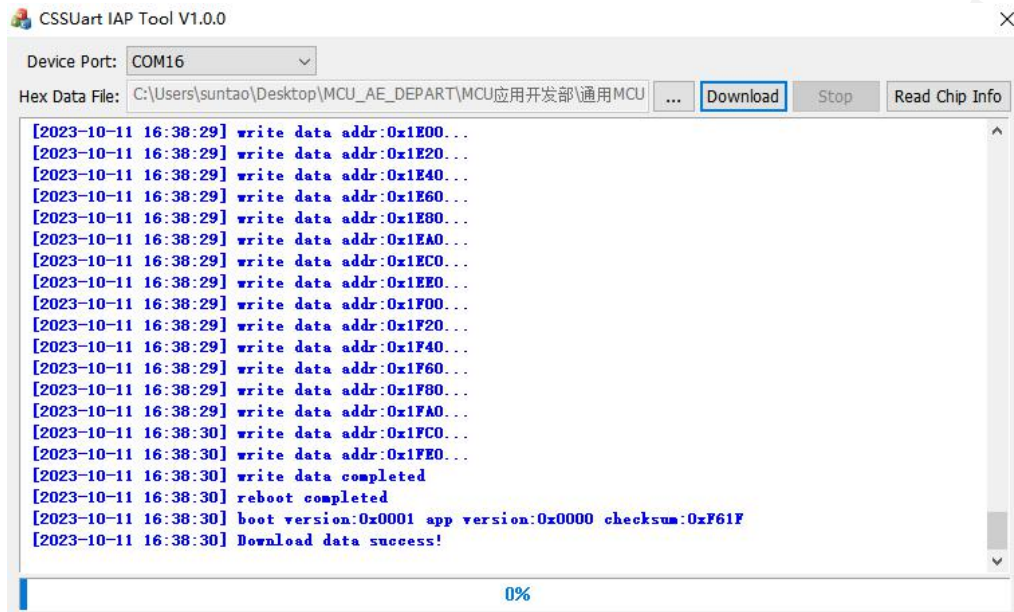


(2) 使用烧录器将拟合生成的 hex 文件烧录至芯片；

8.2 IAP 串口升级

(1) 将 BootLoader 程序利用烧录器烧录或 IDE 下载的方式烧录到芯片。

(2) 准备待升级的 AP hex(注意 1FF5h 处需填入 0000h) ,打开 CSSUart IAP Tool V1.0.0.exe, 选择响应串口号, 加载待升级的 AP hex 文件, 点击 DownLoad, 并等待升级结束。升级成功或超时失败将显示对应 Log。上位机显示升级成功后, 请等待 4s 再拔出 USB 升级工具。





芯海科技
CHIPSEA

股票代码:688595

免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2023 芯海科技（深圳）股份有限公司，保留所有权利。