

摘要

本文档旨在帮助用户基于芯海科技 CSU3AF10 开发设计的产品中加入 IAP 功能，从而通过 USB Type-C 接口的 CC 引脚（利用 PD 协议）或 DP & DM 引脚（利用 UART 协议）实现产品固件升级和更新。介绍和说明 IAP 功能需要用到的芯片资源、IAP 程序设计说明、IAP 升级上位机和升级小板的使用说明、IAP 升级操作流程说明、使用的 PD 和 UART 两种协议的 IAP 升级方式的差异、IAP 设计需要注意的事项等。

版本

历史版本	修改内容	日期
V1.0	初版生成	2022-10-11

目 录

1 概述	4
2 IAP 需要用到的芯片资源	4
2.1 ROM 资源.....	4
2.2 RAM 资源.....	4
2.3 NVR 用户数据区域资源.....	5
3 IAP 程序设计说明	6
3.1 Boot Loader Code 程序设计说明.....	6
3.1.1 Boot Loader Code 提供方式.....	6
3.1.2 IAP 升级密钥设定方式.....	6
3.2 AP Code 程序设计说明.....	8
3.2.1 添加 IAP 升级文件.....	8
3.2.2 调用 IAP 初始化函数.....	8
3.2.3 IAP 的 RAM 变量定义.....	10
3.2.4 IAP 的 ROM 变量定义.....	11
3.2.5 AP Code 必须打开 WDT.....	12
3.2.6 AP Code 偏移.....	13
3.2.7 AP Code 跳转 Boot Loader Code 方式.....	14
4 IAP 升级上位机和升级小板的使用说明	15
4.1 IAP 升级上位机的使用说明.....	15
4.2 IAP 升级小板的使用说明.....	16
4.2.1 IAP 升级小板选取.....	16
4.2.2 安装芯海科技专用 IAP 升级小板的驱动程序.....	17
4.2.3 IAP 专用升级小板的连接说明.....	17
5 IAP 升级操作流程说明	18
5.1 使用烧录器 CS Writer 烧录带 IAP 功能的 Hex 到芯片.....	18
5.2 使用 IAP Tool 升级 AP Code 固件.....	20
6 使用 PD 和 UART 两种协议的 IAP 升级方式的差异和注意事项	21
6.1 IAP 使用的接口和引脚不相同.....	21
6.1.1 利用 PD 协议升级使用的接口和引脚.....	21
6.1.2 利用 UART 协议升级使用的接口和引脚.....	21
6.2 IAP 使用的 Boot Loader Code 档案不同.....	22
6.2.1 利用 PD 协议升级使用的 Boot Loader Code Hex 档案.....	22
6.2.2 利用 UART 协议升级使用的 Boot Loader Code Hex 档案.....	22
6.3 IAP 使用的上位机的升级方式选择不同.....	22
6.3.1 利用 PD 协议升级使用的上位机的升级方式.....	22
6.3.2 利用 UART 协议升级使用的上位机的升级方式.....	22
6.4 IAP 使用的升级小板不同.....	23
6.4.1 利用 PD 协议升级使用的升级小板.....	23
6.4.2 利用 UART 协议升级使用的升级小板.....	23

6.5 触发 IAP 的命令方式不同.....	23
6.5.1 利用 PD 协议升级使用的触发 IAP 命令方式.....	23
6.5.2 利用 UART 协议升级使用的触发 IAP 命令方式.....	23
7 IAP 设计需要注意的事项.....	24
7.1 IAP 使用场景说明.....	24
7.2 NVR 用户数据区域.....	24
7.3 升级芯片 VDD 电压要求.....	24
7.4 必须打开 WDT 功能.....	24
7.5 烧录时不可以加密.....	24
7.6 烧录方式标志位设置.....	24
7.7 中断标志位置位.....	25

1 概述

本说明文档是基于芯海科技开发设计的 CSU3AF10 移动电源 SDK 中加入 IAP 功能的介绍和说明，介绍和说明 IAP 需要用到的芯片资源、IAP 程序设计说明、IAP 升级上位机和升级小板的使用说明、IAP 升级操作流程说明、使用的 PD 和 UART 两种协议的 IAP 升级方式的差异、IAP 设计需要注意的事项。

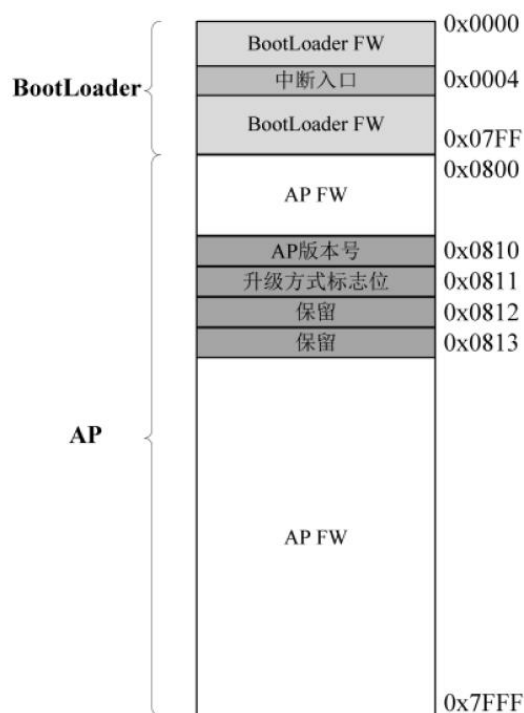
本说明文档介绍的 IAP 功能是指通过产品的 USB Type-C 接口的 CC 引脚（利用 PD 协议）或 DP & DM 引脚（利用 UART 协议）进行固件升级的方式。

2 IAP 需要用到的芯片资源

2.1 ROM 资源

CSU3AF10 内置 32Kx16 位程序存储器（ROM），CSU3AF10 的 IAP 功能需要通过 Boot Loader Code 来实现，由于 CSU3AF10 没有设计专门的 Boot Loader 区域，所以 Boot Load Code（IAP 升级程序）放置于前面 2K word ROM 空间(地址范围 0x0000~0x07FF)，AP Code（用户项目程序）放置于后面 30K word ROM 空间（地址范围 0x0800~0x7FFF）。

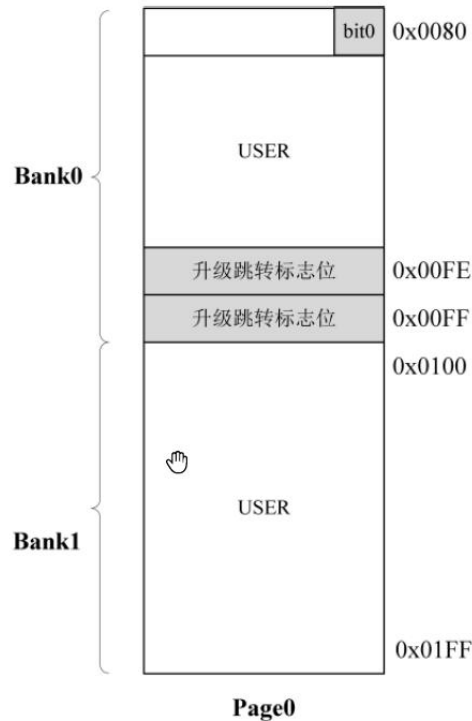
Boot Loader 区域的地址 0x0004 为 MCU 中断入口地址（Boot Load Code & AP Code 共同使用此中断入口地址），AP 区域的地址 0x0810~0x813 为预留空间，预留用于存储 AP Code 版本号、升级方式标志等信息，用户不能够使用和破坏这些预留地址空间的数据。



2.2 RAM 资源

CSU3AF10 内置 2Kx8 位数据存储器 SRAM。

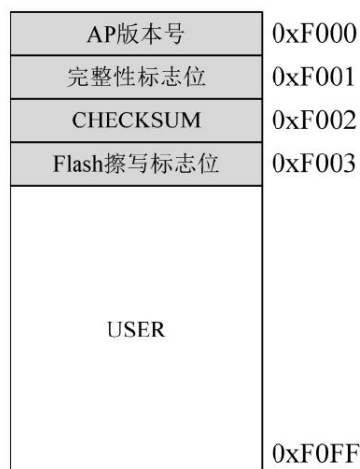
其中，SRAM 中 Page0 的 0x0080 地址的 bit0 以及 0x00FE、0x00FF 为预留空间，用户不能够使用和破坏这些预留地址空间的数据。0x0080 地址的 bit0 表示中断发生标志位，0x00FE、0x00FF 用于存储升级中转标志位。



2.3 NVR 用户数据区域资源

CSU3AF10 内置 256x16 位 Data 用户数据区 NVR0。

其中，NVR0 区域的 0xF000~0xF003 为预留空间，用户不可改变这些地址空间的值，0xF000 用于存储 AP 版本号，0xF001 用于存储完整性标志位，0xF002 用于存储 AP Code CHECKSUM，0xF003 用于存储 FLASH 擦写标志位，在执行 IAP 升级后整个 NVR0 区域 0xF000~0xF0FF 将被用作缓存区，数据将会被擦除破坏。



3 IAP 程序设计说明

3.1 Boot Loader Code 程序设计说明

3.1.1 Boot Loader Code 提供方式

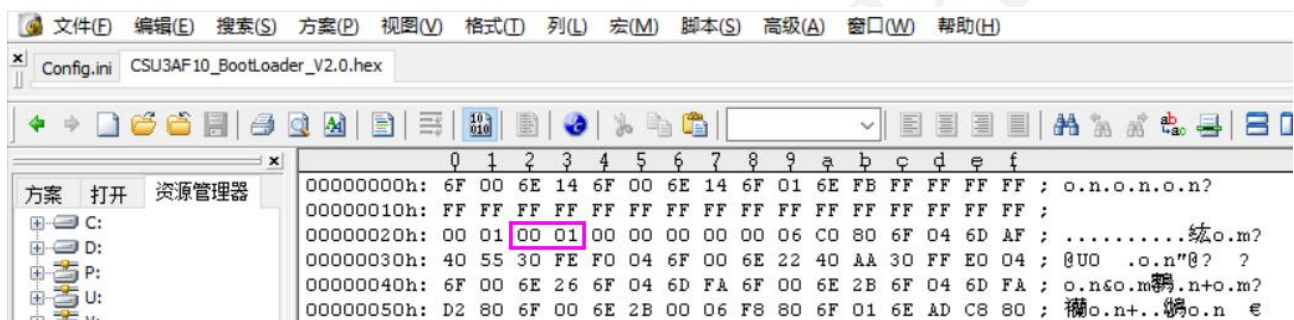
芯海科技以 Hex 档案形式提供 IAP 功能设计时需要的 Boot Loader Code，使用不同协议（PD 协议或 UART 协议）进行升级的 Boot Loader Code 的 Hex 档案不相同。

◇ 注意：为了保证 IAP 功能正常，除了升级密钥外，用户不可以修改 Boot Loader Code Hex 档案的任何其他内容。

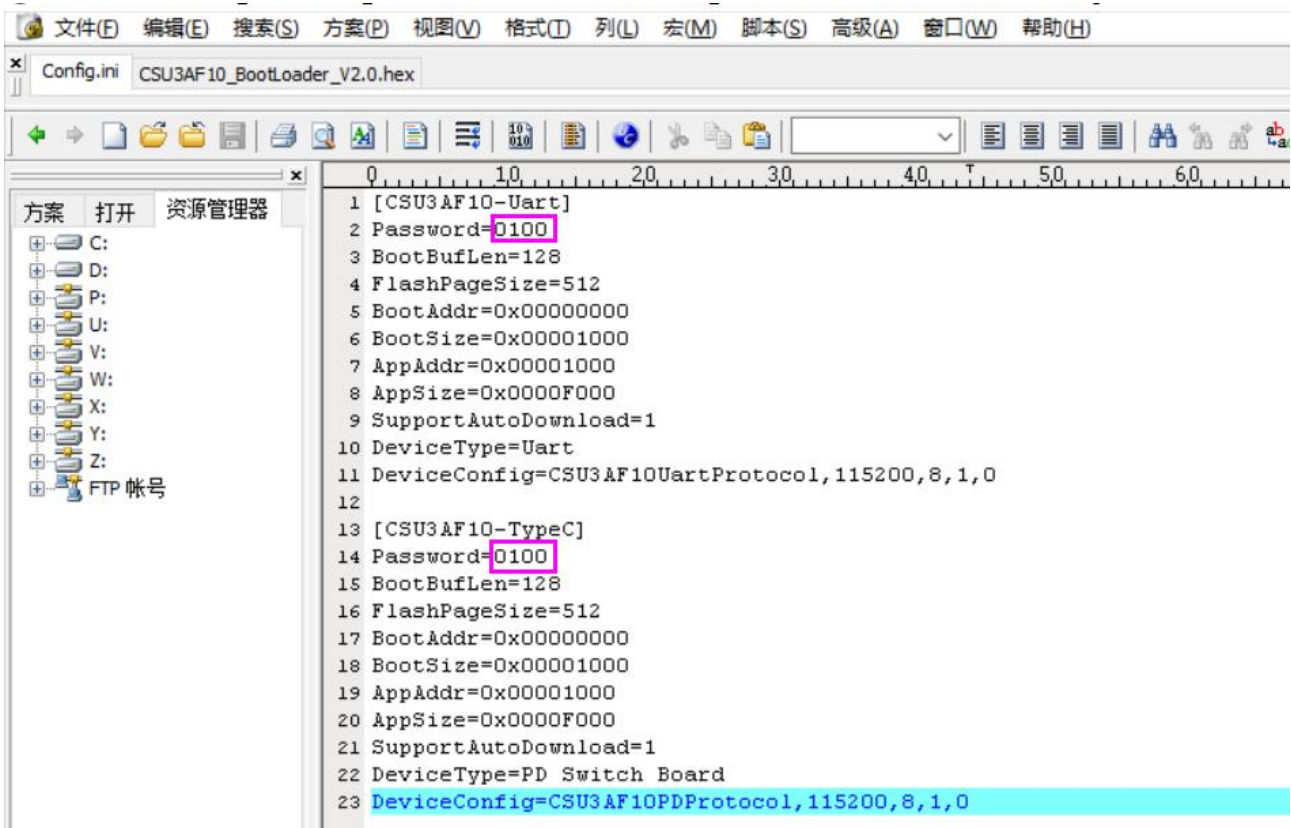
3.1.2 IAP 升级密钥设定方式

为了防止固件被篡改，IAP 支持用户自定义升级密钥，密钥采用 16 位长度，密钥自定义方法如下：

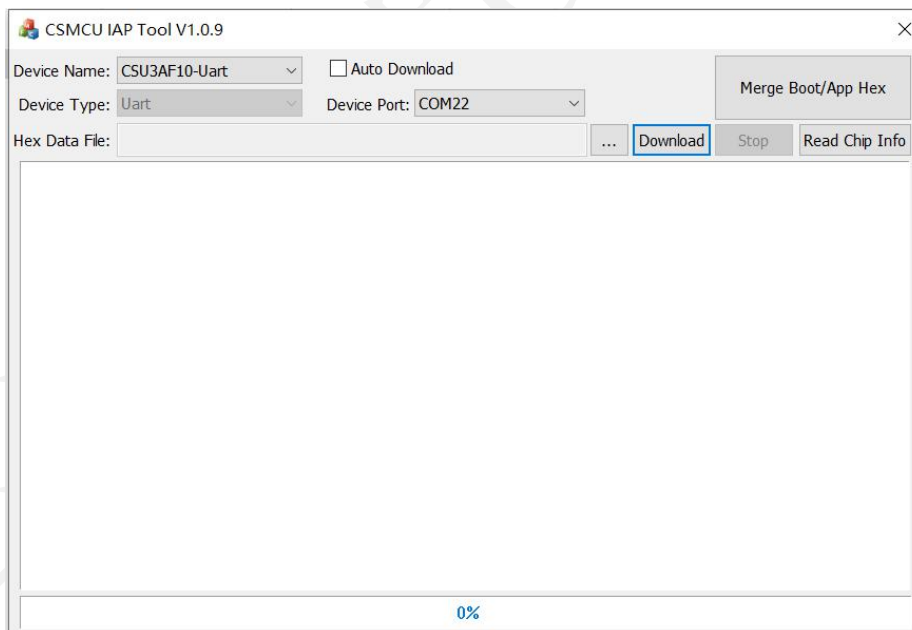
步骤 1：在 Boot Loader Code Hex 文件的地址 0x22，0x23 设定用户自定义的密钥，0x22 的内容表示高 8 位密钥，0x23 的内容表示低 8 位密钥，例如下图紫色方框所示设定的密钥为 0x0001。



步骤 2：将 IAP 上位机“CSMCU IAP Tool V1.0.9”解压缩包内的“Config.ini”文件的 Password 符号设定与 Boot Loader Code 中设定一致的密钥，Password 定义方法如下：密钥低 8 位在前，密钥高 8 位在后，例如下图紫色方框所示设定的密钥为 0x0001。



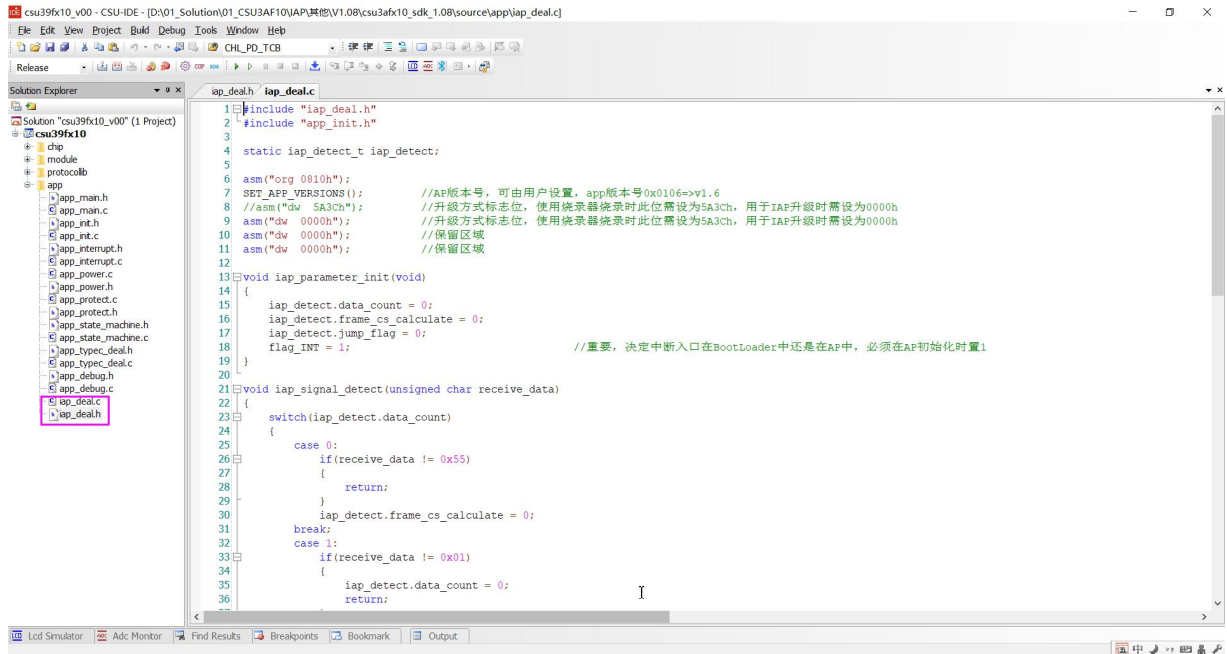
步骤 3: 完成以上两个步骤保存后, 打开 IAP 上位机才能正常进行 IAP 升级固件。



3.2 AP Code 程序设计说明

3.2.1 添加 IAP 升级文件

将包含有 IAP 升级功能需要使用的功能程序和变量定义的文件“iap_deal.c”和“iap_deal.h”添加到 CSU3AF10 工程 APP 文件夹下，如下图紫色方框所示：



3.2.2 调用 IAP 初始化函数

在“app_main.c”文件中添加语句：`#include "iap_deal.h"`。

```

50 //iap
51 #include "iap_deal.h"

```

在从 Boot Loader Code 跳转进入 AP Code 的 main()函数后，调用 iap_parameter_init(void)函数初始化 IAP 使用到的相关变量，如下图紫色方框所示。

✧ 注意：为了保证 IAP 功能正常，此 iap_parameter_init(void)函数必须在进入 AP Code 的 main()函数后第一时间调用。


```

61     do{}while(sys_timer.time_ruler<= _ms);
62 }
63
64 static void branch_100ms(void);
65 static void branch_ls(void);
66 static void qc_break(void);
67 static void key_proc(void);
68 static void sleep_enter_sleep(void);
69 static void low_bat_close_discharge(void);
70 static void trigger_iap_upgrade(void);
71
72
73 uint16_t temp_pd_lib_version;
74 uint16_t temp_multi_lib_version;
75 void main(void)
76 {
77     iap_parameter_init();
78     app_peripheral_init();
79     app_param_init();
80     temp_pd_lib_version= pd_lib_version_number();
81     temp_multi_lib_version= multi_lib_version_number();
82     while(1)
83     {
84         /** PD interrupt polling**/
85         if(plug_state_context.protocol_flag.b_pd_en)
86         {
87             pd_interrupt_polling();
88         }
89     }

```

调用的 `iap_parameter_init(void)`函数在“`iap_deal.c`”文件中，内容如下：

```

1 #include "iap_deal.h"
2 #include "app_init.h"
3
4 static iap_detect_t iap_detect;
5
6 asm("org 0810h");
7 SET_APP_VERSIONS(); //AP版本号, 可由用户设置, app版本号0x0106=>v1.6
8 //asm("dw 5A3Ch"); //升级方式标志位, 使用烧录器烧录时此位需设为5A3Ch, 用于IAP升级时需设为0000h
9 asm("dw 0000h"); //升级方式标志位, 使用烧录器烧录时此位需设为5A3Ch, 用于IAP升级时需设为0000h
10 asm("dw 0000h"); //保留区域
11 asm("dw 0000h"); //保留区域
12
13 void iap_parameter_init(void)
14 {
15     iap_detect.data_count = 0;
16     iap_detect.frame_cs_calculate = 0;
17     iap_detect.jump_flag = 0;
18     flag_INT = 1; //重要, 决定中断入口在BootLoader中还是在AP中, 必须在AP初始化时置1
19 }
20
21 void iap_signal_detect(unsigned char receive_data)
22 {
23     switch(iap_detect.data_count)
24     {
25         //...

```

3.2.3 IAP 的 RAM 变量定义

在“sys_register.c”文件中增加 IAP 中使用到的 RAM 变量定义，如下图方框所示：

```

1 #include "base_address.h"
2
3 volatile unsigned char bank0 APToBoot0 @ 0xfe;
4 volatile unsigned char bank0 APToBoot1 @ 0xff;
5 volatile sbit bank0 flag_INT @ 0x80*8+0;
6
7 // =====
8 //page0 registers
9 volatile unsigned char bank0 IND0 @SFR_BA+ 0x00;
10 volatile unsigned char bank0 IND1 @SFR_BA+ 0x01;
11 volatile unsigned char bank0 FSR0 @SFR_BA+ 0x02;
12 volatile unsigned char bank0 FSR1 @SFR_BA+ 0x03;
13 volatile unsigned char bank0 STATUS @SFR_BA+ 0x04;
14 volatile unsigned char bank0 WORK @SFR_BA+ 0x05;
15 volatile unsigned char bank0 INTE @SFR_BA+ 0x06;
16 volatile unsigned char bank0 ... @SFR_BA+ 0x07;

```

在“csu39fx10.h”文件中增加 IAP 中使用到的 RAM 变量的 extern 声明，如下图方框所示：

```

1
2 #ifndef _CSU39FX10_H
3 #define _CSU39FX10_H
4
5 #include "base_address.h"
6
7 extern volatile unsigned char bank0 APToBoot0 @ 0xfe;
8 extern volatile unsigned char bank0 APToBoot1 @ 0xff;
9 extern volatile sbit bank0 flag_INT @ 0x80*8+0;
10
11
12 // =====
13 //page0 registers
14 extern volatile unsigned char bank0 IND0 @SFR_BA+ 0x00;
15 extern volatile unsigned char bank0 IND1 @SFR_BA+ 0x01;
16 extern volatile unsigned char bank0 FSR0 @SFR_BA+ 0x02;

```

IAP 的 RAM 变量说明如下：

- APToBoot0、APTtoBoot1：用作指示从 AP code 跳转到 boot loader code 的标志位，在收到升级命令从 AP code 跳转到 Boot loader code 前分别将其设置为 0x55，0xAA。

```

86 void pd_firmware_update_cb(void)
87 {
88     GIE = 0;
89     APToBoot0 = 0x55;
90     APTtoBoot1 = 0xAA;
91     while(1);
92 }

```

- Flag_INT：中断发生标志位，用作指示是 AP code 的中断还是 Boot loader code 的中断，进入 AP code 后将其设置为 ‘1’，进入 Boot loader code 后将其设置为 ‘0’。

```


13 void iap_parameter_init(void)
14 {
15     iap_detect.data_count = 0;
16     iap_detect.frame_cs_calculate = 0;
17     iap_detect.jump_flag = 0;
18     flag_INT = 1;
19 }

```

//重要，决定中断入口在BootLoader中还是在AP中，必须在AP初始化时置1

3.2.4 IAP 的 ROM 变量定义

在“iap_deal.c”文件中增加 IAP 中使用到的 ROM 变量定义：



```

1 #include "iap_deal.h"
2 #include "app_init.h"
3
4 static iap_detect_t iap_detect;
5
6 asm("org 0810h");
7 SET_APP_VERSIONS(); //AP版本号, 可由用户设置, app版本号0x0106=>v1.6
8 //asm("dw 5A3Ch"); //升级方式标志位, 使用烧录器烧录时此位需设为5A3Ch, 用于IAP升级时需设为0000h
9 asm("dw 0000h"); //升级方式标志位, 使用烧录器烧录时此位需设为5A3Ch, 用于IAP升级时需设为0000h
10 asm("dw 0000h"); //保留区域
11 asm("dw 0000h"); //保留区域
12
13 void iap_parameter_init(void)
14 {
15     iap_detect.data_count = 0;
16     iap_detect.frame_cs_calculate = 0;
17     iap_detect.jump_flag = 0;
18     flag_INT = 1; //重要, 决定中断入口在BootLoader中还是在AP中, 必须在P
19 }
20
21 void iap_signal_detect(unsigned char receive_data)
22 {
23     switch(iap_detect.data_count)
24     {
25         case 0:
    
```

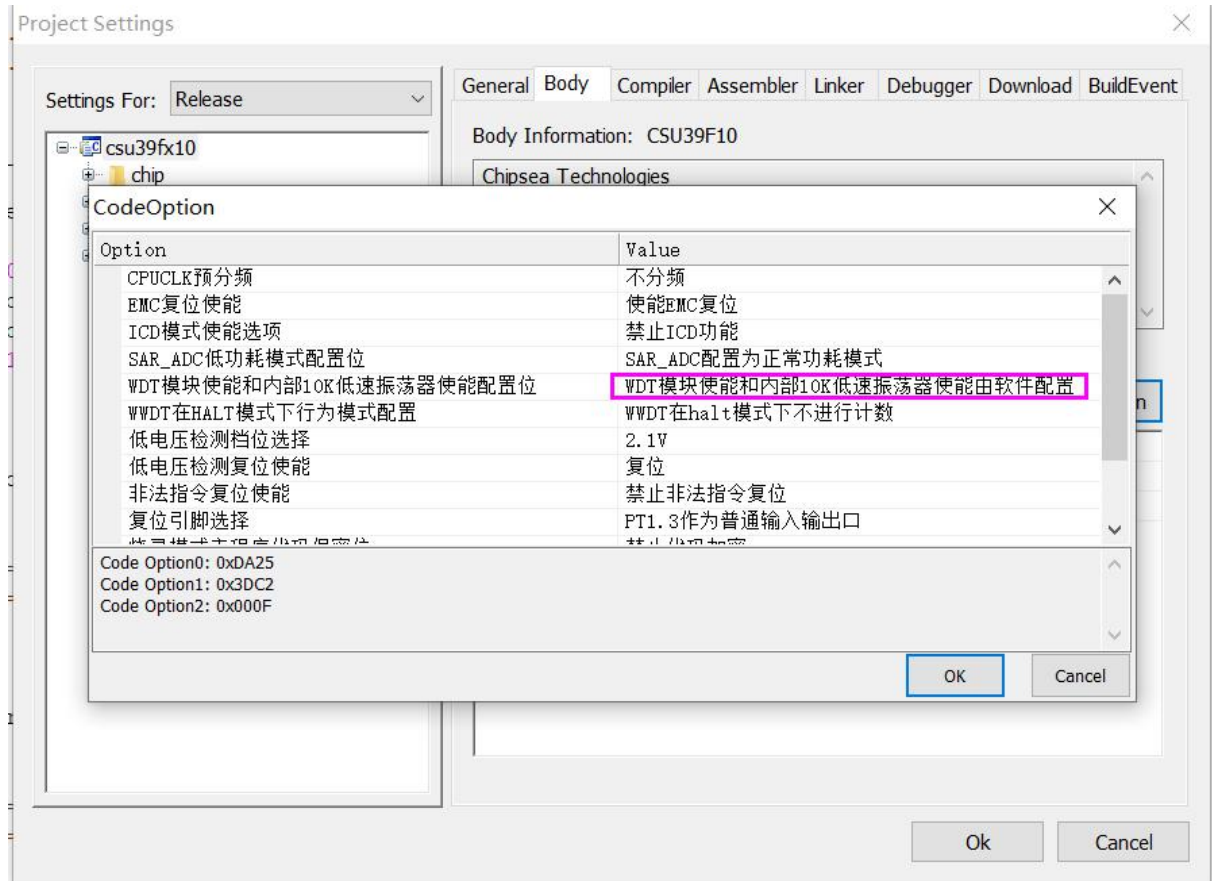
AP code 区域中 0810h~0813h 保留为 IAP 功能使用，说明如下：

- 0810h: 用于存放 AP code 版本号信息；
- 0811h: 用于存放升级方式标志位；
 - 当制作准备用于 IAP 合成的 AP code 时，0811h 必须填写 5A3Ch；
 - 当制作准备用于 IAP 升级的 AP code 时，0811h 必须填写 0000h；
- 0812h: 保留区域，用户不能修改；
- 0813h: 保留区域，用户不能修改；

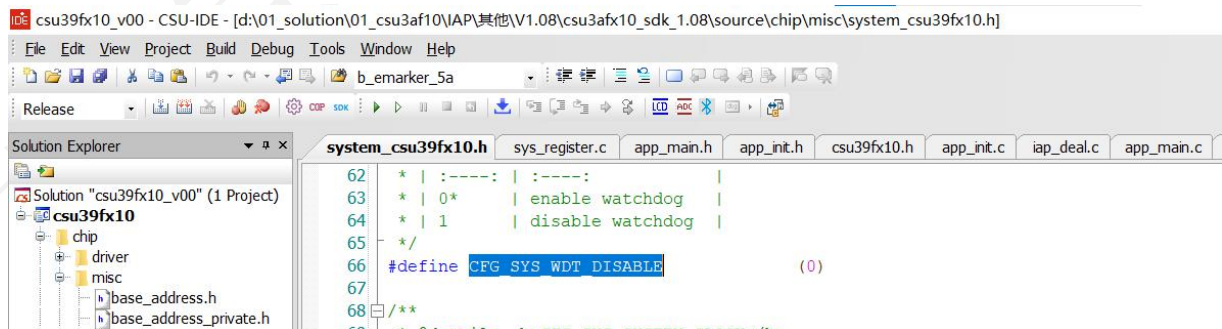
3.2.5 AP Code 必须打开 WDT

为了保证收到 IAP 升级命令后，AP Code 能够正常跳转到 Boot loader code 实现 IAP 功能，AP code 必须打开 WDT 功能，打开方法如下：

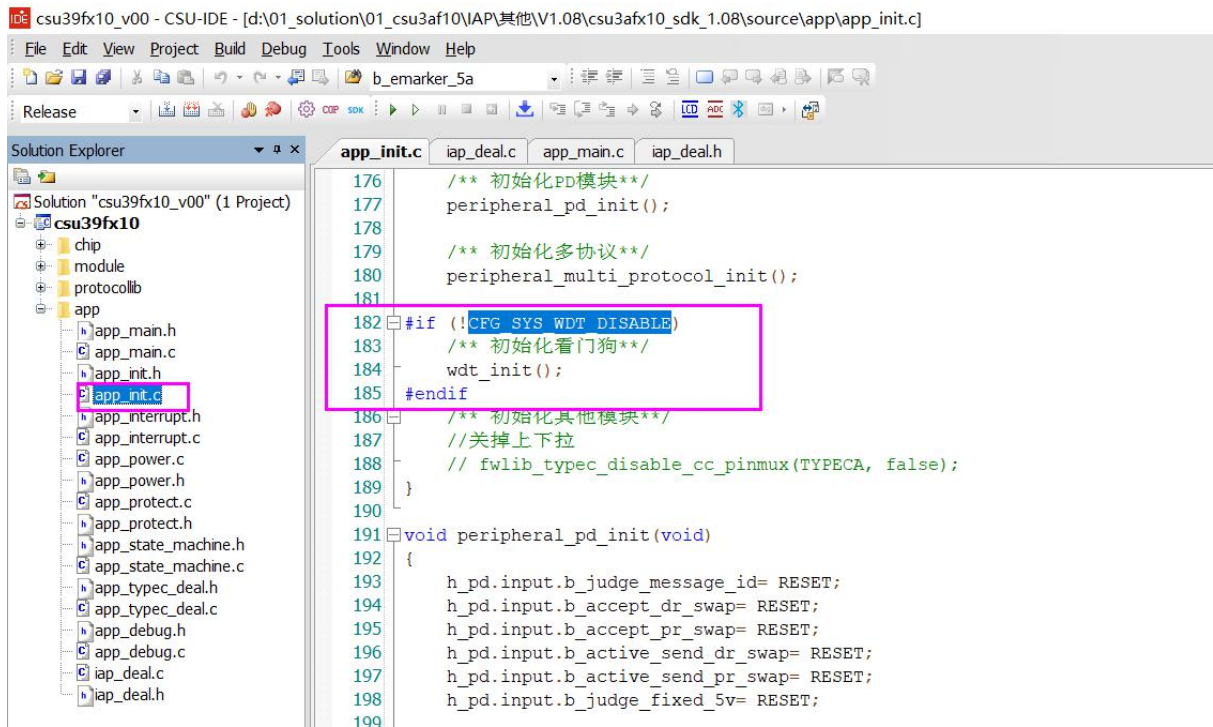
步骤 1：在工程设置窗口，按下图方框所示配置 WDT：



步骤 2：在 AP code 的 “system_csu39fx10.h” 文件中，将 “CFG_SYS_WDT_DISABLE” 定义为 ‘0’ 打开 WDT 功能：



步骤 3：在 “app_init.c” 文件的 “app_peripheral_init(void)” 函数中调用 “wdt_init()” 函数初始化打开 WDT：



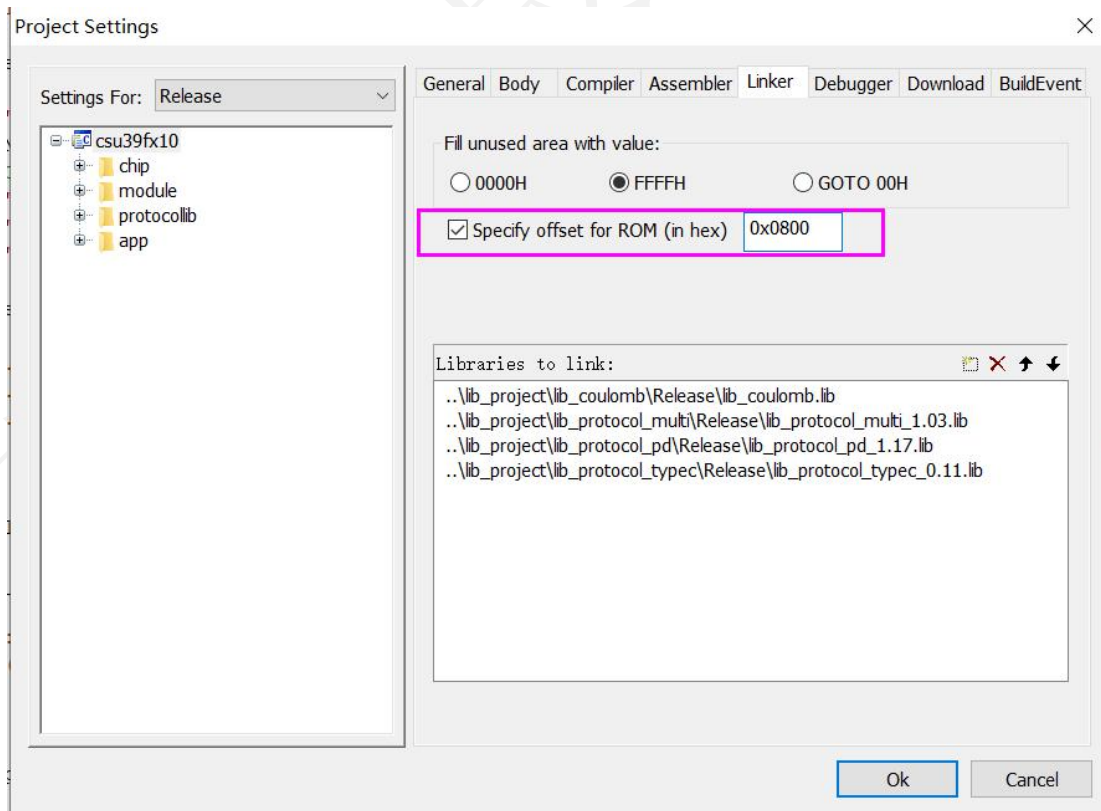
```

176  /** 初始化PD模块**/
177  peripheral_pd_init();
178
179  /** 初始化多协议**/
180  peripheral_multi_protocol_init();
181
182  #if (!CFG_SYS_WDT_DISABLE)
183  /** 初始化看门狗**/
184  wdt_init();
185  #endif
186  /** 初始化其他模块**/
187  //关掉上下拉
188  // fwlib_typec_disable_cc_pinmux(TYPECA, false);
189  }
190
191  void peripheral_pd_init(void)
192  {
193      h_pd.input.b_judge_message_id= RESET;
194      h_pd.input.b_accept_dr_swap= RESET;
195      h_pd.input.b_accept_pr_swap= RESET;
196      h_pd.input.b_active_send_dr_swap= RESET;
197      h_pd.input.b_active_send_pr_swap= RESET;
198      h_pd.input.b_judge_fixed_5v= RESET;
199
200
    
```

3.2.6 AP Code 偏移

由于 AP Code 放置于后面 30K word ROM 空间 0x0800~0x7FFFF，AP Code 需要偏移，设置方法如下：

打开工程设置窗口，选择 Linker 标签页设置 AP code 偏移地址为：0x0800。如下图所示：



3.2.7 AP Code 跳转 Boot Loader Code 方式

在 AP code 中收到 IAP 升级命令（PD 升级封包或按键）后，调用函数“void pd_firmware_update_cb(void)”执行以下程序后跳转到 Boot Loader 进行 IAP 升级，执行程序如下：

- 关闭总中断；
- 将 APTtoBoot0、APTtoBoot1 分别设置为 0x55，0xAA；
- 进入 while(1)不清 WDT 等待 WDT 溢出复位 MCU 进入 Boot loader code 进行 IAP 升级。

```

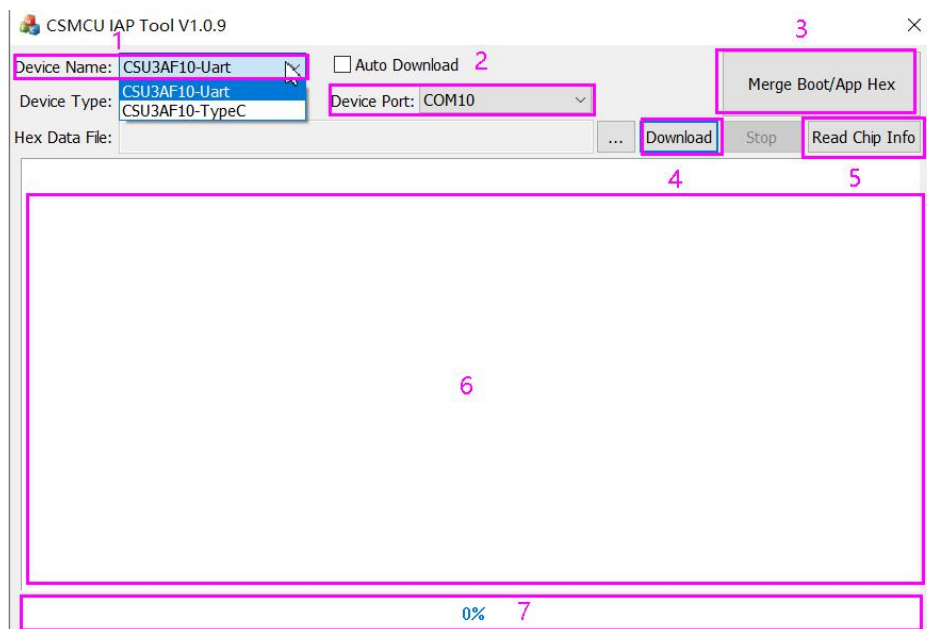
61         return;
62     }
63     iap_detect.data_count++;
64     iap_detect.frame_cs_calculate += receive_data;
65 }
66
67 void iap_reset_judge(void)
68 {
69     if(iap_detect.jump_flag)
70     {
71         GIE = 0;
72         APTtoBoot0 = 0x55;
73         APTtoBoot1 = 0xAA;
74         while(1);
75     }
76 }
77
78 void iap_upgrade_app(void)
79 {
80     GIE = 0;
81     APTtoBoot0 = 0x55;
82     APTtoBoot1 = 0xAA;
83     while(1);
84 }
85
86 void pd_firmware_update_cb(void)
87 {
88     GIE = 0;
89     APTtoBoot0 = 0x55;
90     APTtoBoot1 = 0xAA;
91     while(1);
92 }
93

```

4 IAP 升级上位机和升级小板的使用说明

4.1 IAP 升级上位机的使用说明

无论是选取 PD 协议方式还是 UART 协议方式的 IAP 功能升级固件，必须选择 CSMCU IAP Tool 作为 IAP 升级上位机，IAP Tool 介面和各个菜单功能说明如下：



1. Device Name:

如果使用 DP & DM 引脚（UART 协议）升级方式，请选择 CSU3AF10-Uart。

如果使用 CC 引脚（PD 协议）升级方式，请选择 CSU3AF10-TypeC。

2. Device Port:

升级小板插入电脑后，电脑会自动分配一个串口号给升级小板，根据电脑分配的串口号（设备管理器显示的串口号），正确选择升级工具的串口号。

3. Merge Boot/App Hex:

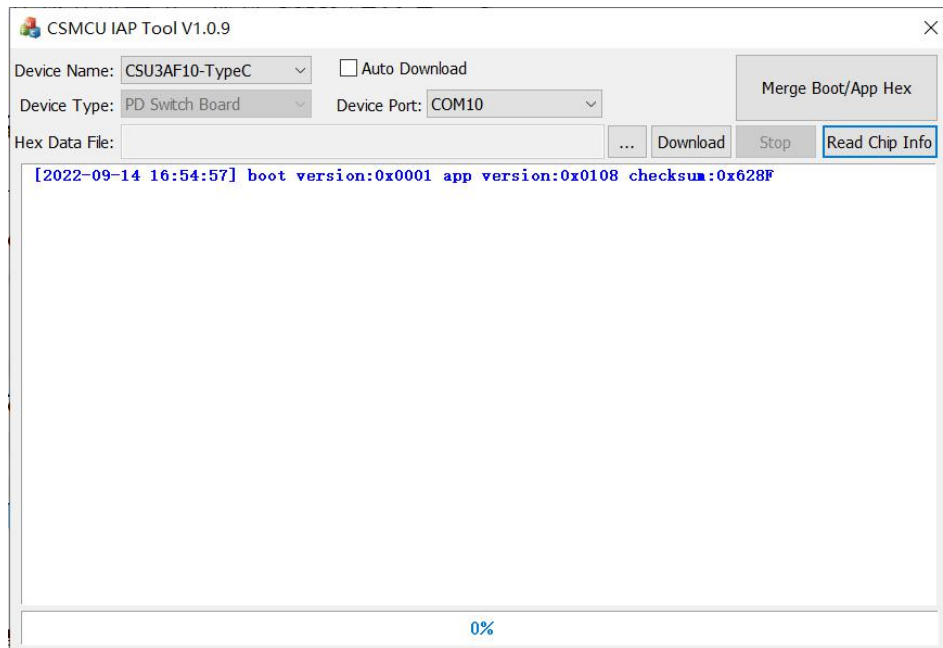
用于合成 Boot loader code 和 AP code，合成输出带 IAP 的 HEX 档案。

4. Download:

通过 IAP 方式升级待更新的 AP code 到芯片；

5. Read Chip Info:

读取芯片信息：boot loader code 版本，AP code 版本，AP Code checksum。



6. 升级信息输出窗口：

输出显示升级过程的状态信息。

7. 升级状态栏：

显示升级进度的百分比信息。

4.2 IAP 升级小板的使用说明

4.2.1 IAP 升级小板选取

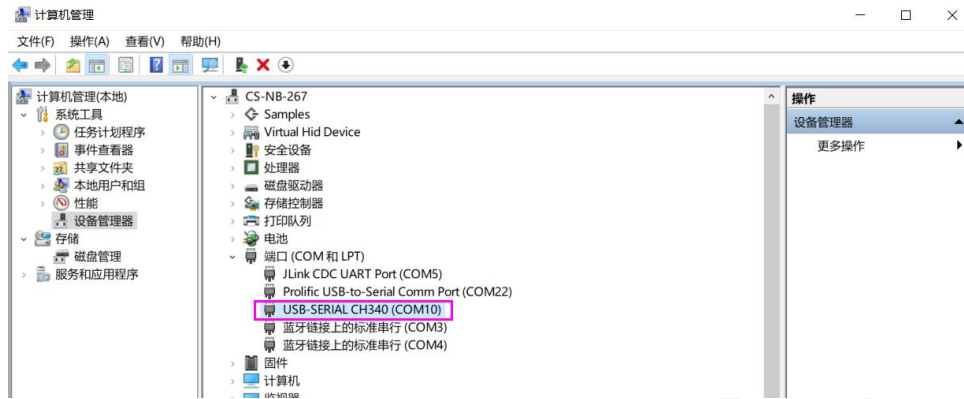
如果利用 CC 引脚（PD 协议）升级固件，必须使用芯海科技提供的专用的 IAP 升级小板进行 IAP 升级固件。专用的 IAP 升级小板的板名为：CS32G050_UpgradeTool_V1.0。



如果利用 DP & DM（UART 协议）升级固件，可以使用通常使用的 USB 转串口调试板，此部分不作过多说明。

4.2.2 安装芯海科技专用 IAP 升级小板的驱动程序

上网下载或与芯海科技索取 IAP 专用升级小板的驱动程序“USB 转串口 CH340-341”并安装到电脑上，安装驱动程序后，将 IAP 专用升级小板插入电脑后，电脑会自动识别到芯海科技提供的 IAP 专用升级小板并显示其串口端号，如下图所示：

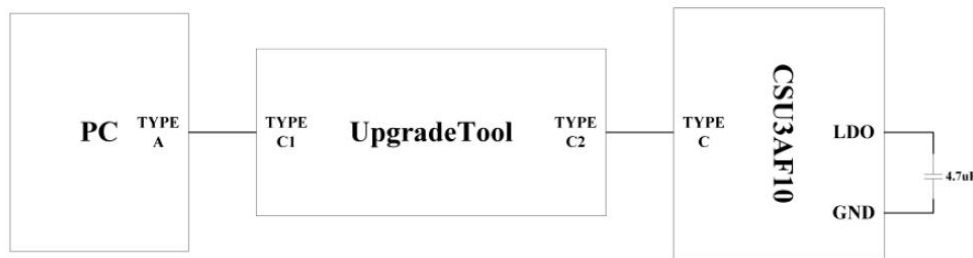


4.2.3 IAP 专用升级小板的连接说明

将 IAP 专用升级小板的 USB-Type C1 口通过 A to C 线缆连接至电脑端的 USB-A 接口；

将 IAP 专用升级小板的 USB-Type C2 口通过 C to C 线缆连接至待升级的产品的 USB-TypeC 接口；

连接示意图如下：



5 IAP 升级操作流程说明

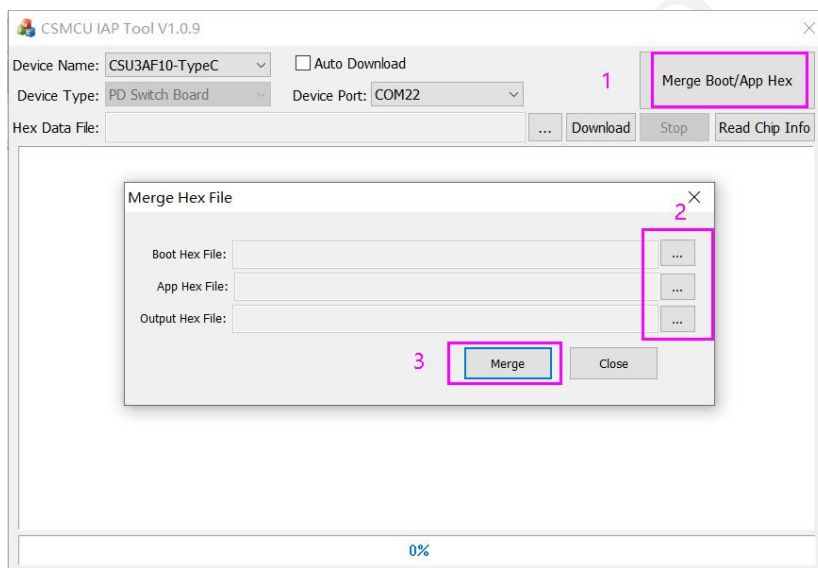
5.1 使用烧录器 CS Writer 烧录带 IAP 功能的 Hex 到芯片

步骤 1: 准备好待合成的 Boot Loader Code Hex 档案和 AP Code Hex 档案（**注意：AP Code 的 0811h 处需要填入 5A3Ch**）准备合成带 IAP 功能的 Hex；

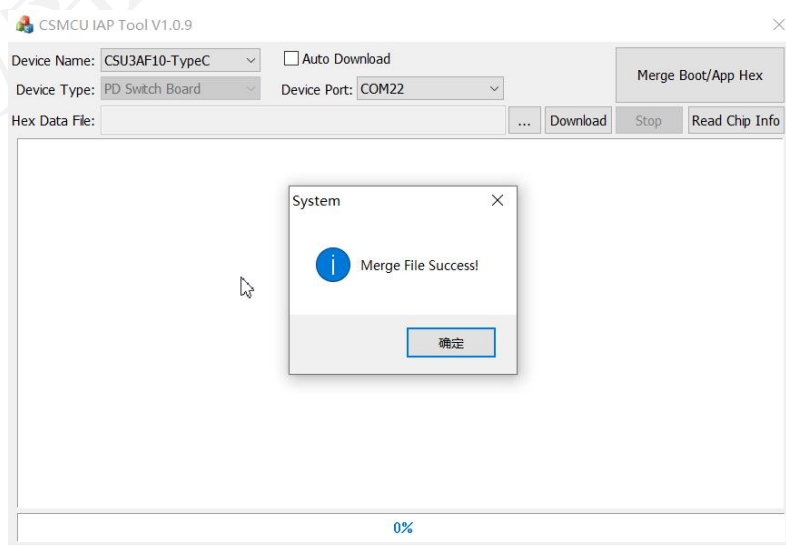
Boot Loader Code Hex 档案制作请参考：“Boot Loader Code 程序设计说明”章节说明；

AP code hex 档案制作请参考：“AP Code 程序设计说明”章节说明；

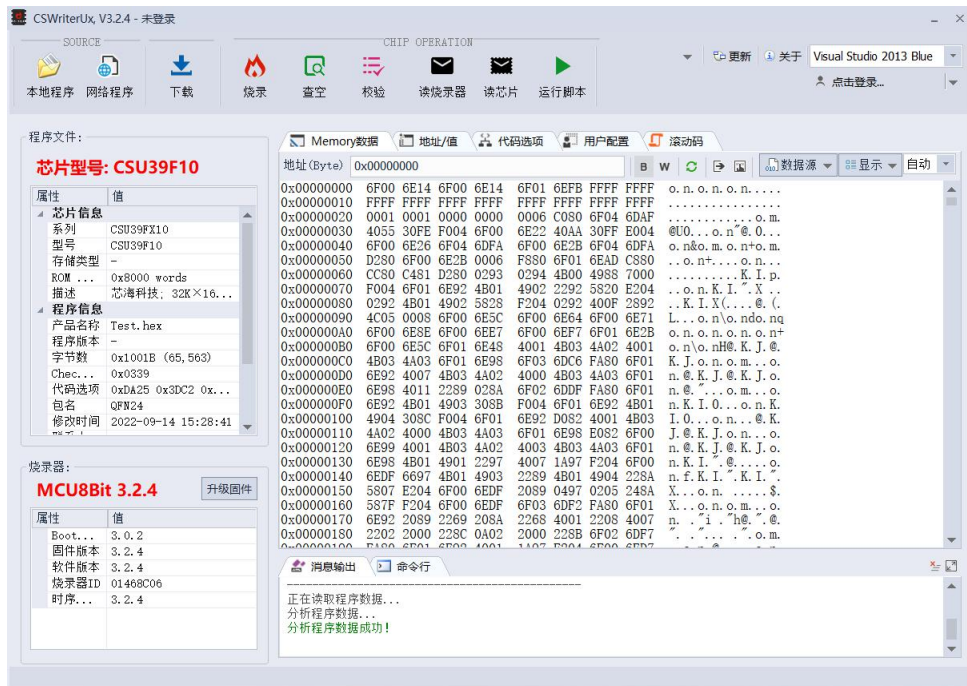
步骤 2: 打开“CSMCU IAP Tool V1.0.9.exe”，点击“Merge Boot/APP Hex”按钮，弹出“Merge Hex File”窗口，在“Boot Hex File:”和“APP Hex File”分别加载 Boot Loader Code Hex 档案和 AP Code Hex 档案，在“Output Hex File”确定拟合成的 Hex 的输出路径和文件名，点击“Merge”按钮完成合成。



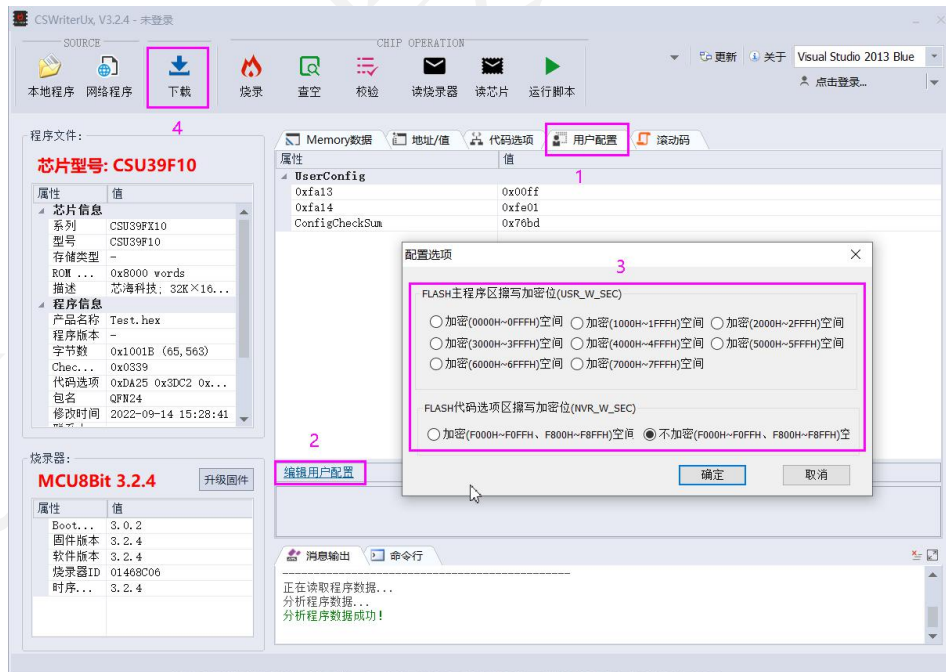
合成 Hex 成功显示界面如下：



步骤 3: 使用烧录器 CS writer 将步骤 2 合成的 Hex 烧录至 CSU3AF10 芯片。



注意：在下载至烧录器烧录到芯片前，必须按下图所示的标示的数字的四个步骤在“用户配置”选项中关闭 FLASH 的“主程序区域”和“代码选项区域”的加密功能（如步骤 3 图片配置），否则 IAP 功能会不正常。

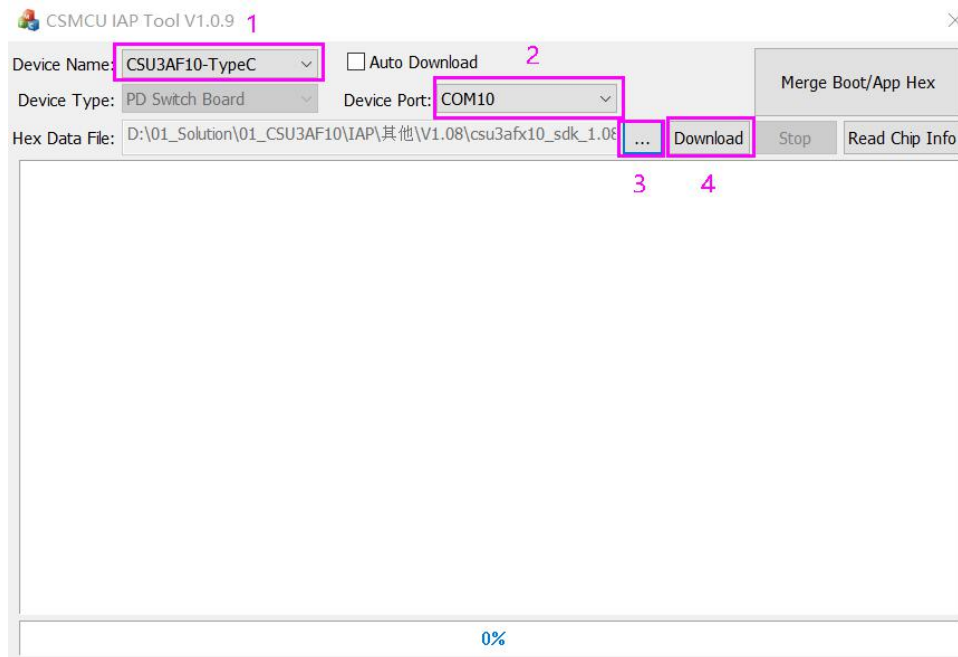


5.2 使用 IAP Tool 升级 AP Code 固件

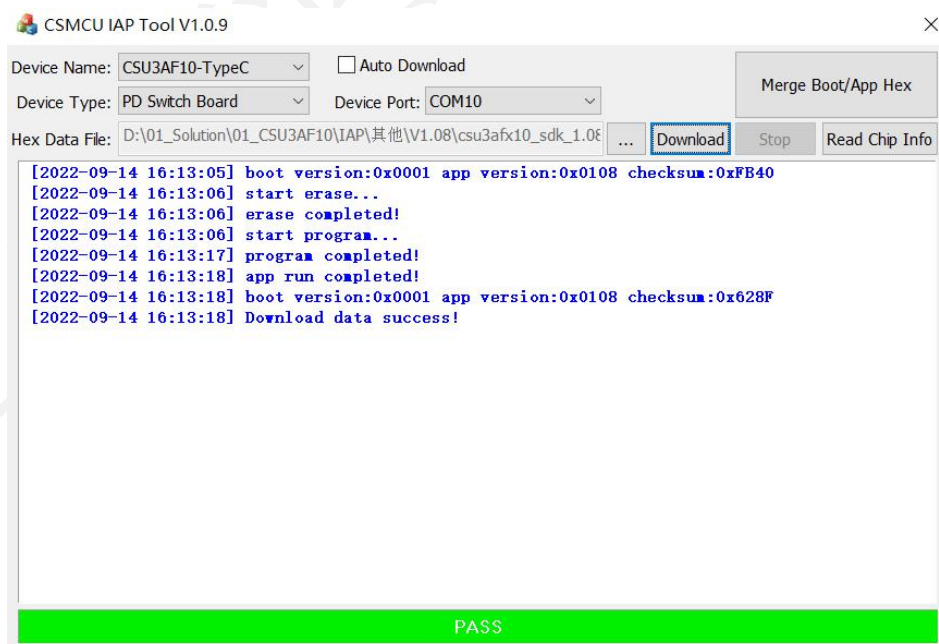
步骤 1: 准备好待升级的 AP Code Hex 档案（注意：AP Code 的 0811h 处需要填入 0000h）；

AP code hex 档案制作请参考：“AP Code 程序设计说明”章节说明；

步骤 2: 打开“CSMCU IAP Tool V1.0.9.exe”，第 1 步，根据升级协议正确选择升级方式，第 2 步，根据电脑设备管理器正确选择串口号，第 3 步，加载待升级的 AP Code Hex 档案，第 4 步，点击 Download 按钮开始升级。



升级成功后，显示如下信息：



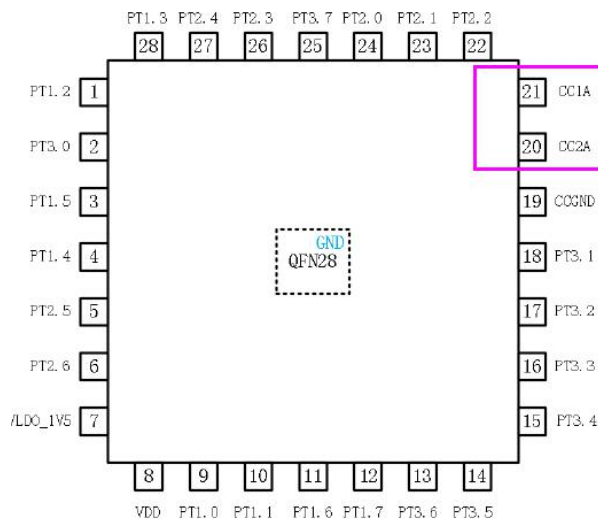
6 使用 PD 和 UART 两种协议的 IAP 升级方式的差异和注意事项

目前 CSU3AF10 使用的 IAP 升级功能支持下面两种协议进行传输数据：PD 协议或 UART 协议。用户选取的传输数据的协议方式不同，IAP 使用的接口和引脚、Boot Loader Code 档案、升级小板是不相同的，下面章节将详细介绍两种协议的 IAP 的差异。

6.1 IAP 使用的接口和引脚不相同

6.1.1 利用 PD 协议升级使用的接口和引脚

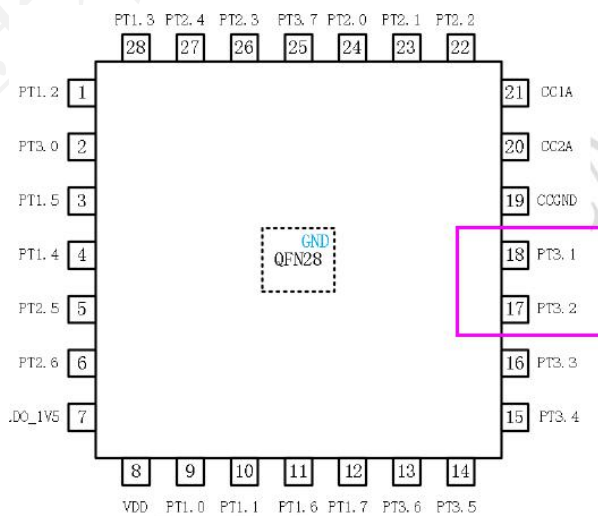
如果利用 PD 协议升级固件，固定使用 USB-TypeCA 接口的 CC1A 或 CC2A 引脚（具体使用哪只引脚取决于正反插）。



6.1.2 利用 UART 协议升级使用的接口和引脚

如果利用 UART 协议升级固件，固定使用 PT3.1（TX）、PT3.2（RX）引脚，UART 波特率为 115200。

在 CSU3AF10_EVB_V1.1 上，PT3.1、PT3.2 分别作为 DP1，DM1 引脚。



6.2 IAP 使用的 Boot Loader Code 档案不同

6.2.1 利用 PD 协议升级使用的 Boot Loader Code Hex 档案

如果利用 PD 协议升级固件，使用 Hex 档案名称为：CSU3AF10_BootLoader_V20.HEX。

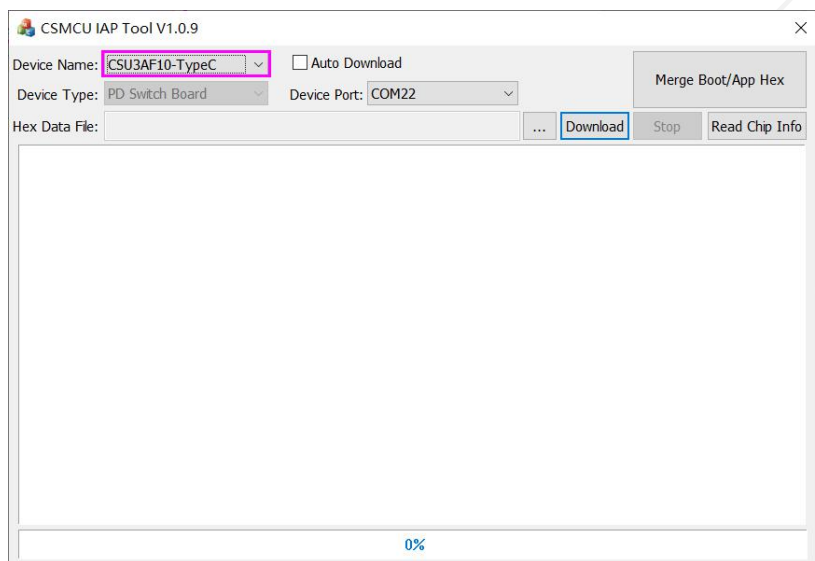
6.2.2 利用 UART 协议升级使用的 Boot Loader Code Hex 档案

如果利用 UART 协议升级固件，使用 Hex 档案名称为：CSU3AF10_BootLoader_V10.HEX。

6.3 IAP 使用的上位机的升级方式选择不同

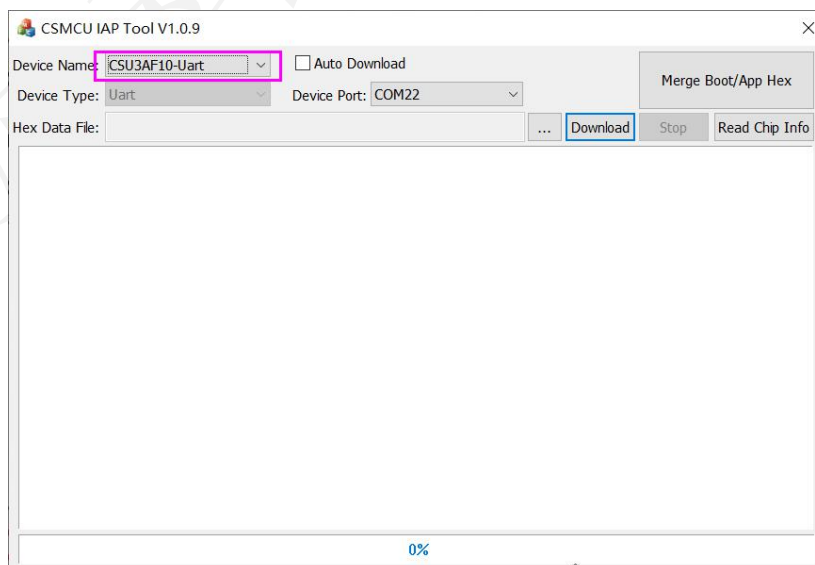
6.3.1 利用 PD 协议升级使用的上位机的升级方式

如果利用 PD 协议升级固件，选择的上位机的升级方式为：CSU3AF10-TypeC。



6.3.2 利用 UART 协议升级使用的上位机的升级方式

如果利用 UART 协议升级固件，选择的上位机的升级方式为：CSU3AF10-Uart。



6.4 IAP 使用的升级小板不同

6.4.1 利用 PD 协议升级使用的升级小板

如果利用 PD 协议升级固件，用户必须使用芯海科技提供的专用升级小板：
CS32G050_UpgradeTool_V1.0。



6.4.2 利用 UART 协议升级使用的升级小板

如果利用 UART 协议升级固件，用户可以使用通用的 USB 转串口调试工具小板：

6.5 触发 IAP 的命令方式不同

6.5.1 利用 PD 协议升级使用的触发 IAP 命令方式

如果利用 PD 协议升级固件，直接通过 USB-TypeC 口接收到固件升级 PD 封包触发 IAP 升级功能。

6.5.2 利用 UART 协议升级使用的触发 IAP 命令方式

如果利用 UART 协议升级固件，由于 PT31、PT32 分别作为 DP、DM 功能，没有办法通过直接接收固件升级 UART 封包来触发 IAP 升级固件，必须通过其他方式（例如长按键、双击按键等方式）来触发 IAP 升级功能。

7 IAP 设计需要注意的事项

7.1 IAP 使用场景说明

特别声明：IAP 更新固件方式并非芯海科技提供的官方量产烧录方式，不建议作为产品工厂批量烧录方式使用，此工具原意是用于工程调试时更新固件用途。

7.2 NVR 用户数据区域

用户数据存储区域（F000h~F0FFh）在升级过程中被用作缓存区，升级后该区域的数据将被改写。

7.3 升级芯片 VDD 电压要求

为了保证芯片 Flash 擦写正常，升级过程中芯片 VDD 电压需要高于 3.7V。

7.4 必须打开 WDT 功能

为了保证能够让 AP code 跳转到 boot loader 进行 IAP 升级，AP code 必须打开 WDT 功能。

7.5 烧录时不可以加密

由于 ROM 加密后，用户模式对应空间不能进行擦写，所以使用 CS writer 烧录合成的 HEX 档案到芯片时，应当按下图所示设置不进行加密烧录。



7.6 烧录方式标志位设置

为了保证使用 CS writer 烧录合成的 HEX 后的产品的功能正常，必须按照如下建议说明制作 AP code HEX 档案：

当制作准备用于合成的 AP code 时，0811h 必须填写 5A3Ch；

```

6  asm("org 0810h");
7  SET_APP_VERSIONS();           //AP版本号，可由用户设置，app版本号0x0106=>v1.6
8  asm("dw 5A3Ch");              //升级方式标志位，使用烧录器烧录时此位需设为5A3Ch，用于IAP升级时需设为0000h
9  //asm("dw 0000h");           //升级方式标志位，使用烧录器烧录时此位需设为5A3Ch，用于IAP升级时需设为0000h
10 asm("dw 0000h");              //保留区域
11 asm("dw 0000h");              //保留区域

```

当制作准备用于 IAP 升级的 AP code 时，0811h 必须填写 0000h；

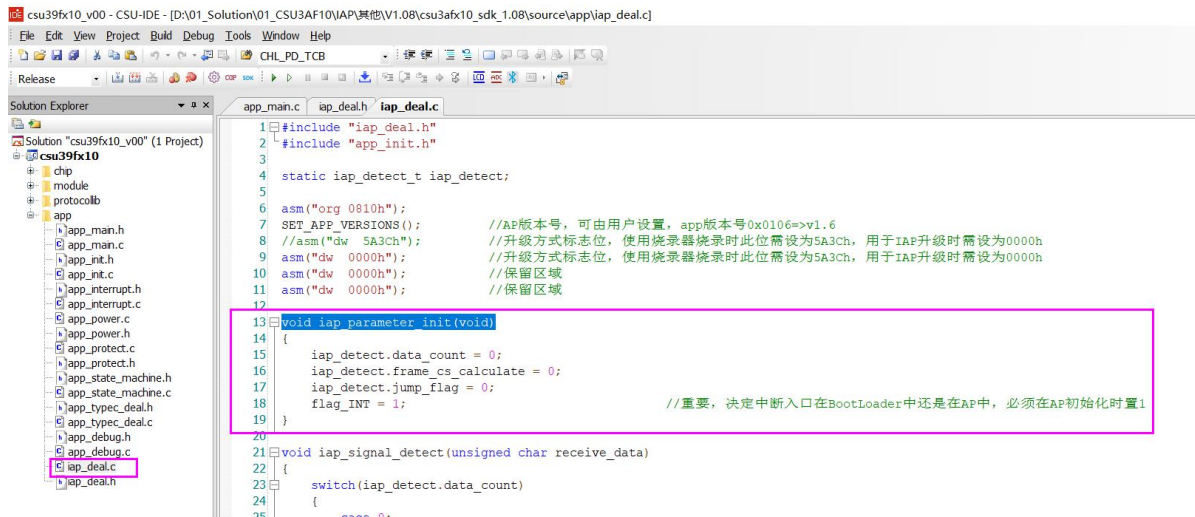
```

6  asm("org 0810h");
7  SET_APP_VERSIONS();           //AP版本号，可由用户设置，app版本号0x0106=>v1.6
8  //asm("dw 5A3Ch");           //升级方式标志位，使用烧录器烧录时此位需设为5A3Ch，用于IAP升级时需设为0000h
9  asm("dw 0000h");              //升级方式标志位，使用烧录器烧录时此位需设为5A3Ch，用于IAP升级时需设为0000h
10 asm("dw 0000h");              //保留区域
11 asm("dw 0000h");              //保留区域
12

```


7.7 中断标志位置位

为了保证在执行 AP code 时产生的中断能够正常执行，必须在进入 AP code 后第一时间调用 `iap_parameter_init()` 函数对中断标志位 ‘flag_INT’ 变量置 ‘1’。



```

1 #include "iap_deal.h"
2 #include "app_init.h"
3
4 static iap_detect_t iap_detect;
5
6 asm("org 0810h");
7 SET_APP_VERSIONS(); //AP版本号，可由用户设置，app版本号0x0106=>v1.6
8 //asm("dw 5A3Ch"); //升级方式标志位，使用烧录器烧录时此位需设为5A3Ch，用于IAP升级时需设为0000h
9 asm("dw 0000h"); //升级方式标志位，使用烧录器烧录时此位需设为5A3Ch，用于IAP升级时需设为0000h
10 asm("dw 0000h"); //保留区域
11 asm("dw 0000h"); //保留区域
12
13 void iap_parameter_init(void)
14 {
15     iap_detect.data_count = 0;
16     iap_detect.frame_cs_calculate = 0;
17     iap_detect.jump_flag = 0;
18     iap_detect.flag_INT = 1; //重要，决定中断入口在BootLoader中还是在AP中，必须在AP初始化时置1
19 }
20
21 void iap_signal_detect(unsigned char receive_data)
22 {
23     switch(iap_detect.data_count)
24     {
25         // ...
    
```

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 © 2022 芯海科技（深圳）股份有限公司，保留所有权利。


芯海科技
CHIPSEA

股票代码:688595