



Fremont Micro Devices

AN-22017

8bit MCU IAP 使用说明

Rev1.00

www.fremontmicro.com

文档修改历史

日期	版本	描述
2021-12-07	V1.00	初版

目录

1	串口升级通信协议	5
1.1	适用型号	5
1.2	标准模式	5
1.3	一对多模式	5
1.4	简单模式	5
2	程序说明	7
2.1	IAP DEMO 特性	7
2.2	闪存映射	7
2.3	IAP 程序设置流程	8
2.4	IAP 程序注意事项	10
2.5	IAP 在线升级示例	14
2.6	IAP 及应用程序组合烧录流程	15
2.7	简单版 IAP 介绍	17
2.8	一对多版 IAP 介绍	17
2.9	示例版本差异对比	17
	联系信息	18

表目录 / List of Figures

表 2-1	通信方式差异	17
表 2-2	所占容量大小差异	17
表 2-3	烧录时间差异	17

图目录 / List of Tables

图 2-1	FLASH 闪存区域映射	8
图 2-2	标准 IAP 程序流程	9
图 2-3	一对多 IAP 流程图	10
图 2-4	变量位置声明	12
图 2-5	工程设置	12
图 2-6	IAP 地址配置.....	13
图 2-7	应用程序地址配置.....	13
图 2-8	IAP 在线升级示例	14
图 2-9	在线编程成功	15
图 2-10	烧录器导入程序	16
图 2-11	EEPROM 存放的 CRC.....	16

1 串口升级通信协议

1.1 适用型号

目前支持的 8bit MCU 型号如下:

FT61F08X, FT62F08X, FT61F14X, FT61F0AX, FT64F0AX, FT67F0AX.

1.2 标准模式

a. 发送启动命令,

命令格式: 0x46,0x4D,0x44,0x49,0x41,0x50,0x55,0xAA

启动成功回复: 0xAA

注: 失败则重试 2 次

b. 写数据,每次写一页,

数据包格式: 0x05, Addr_H, Addr_L, Data0 Datan, CRC_H, CRC_L

成功回复: 0x00, 失败回复: 0x44(失败重复写 2 次)

c. 发送复位命令: 0x06

成功回复: 00;

1.3 一对多模式

a. 写数据,每次写一页, 每包数据重复写 3 次, 不需要回复

数据包格式: 命令头+05+地址+数据+CRC,

命令头: 0x46,0x4D,0x44,0x49,0x41,0x50,0x55,0xAA

命令头, 0x05, Addr_H,Addr_L, Data0 Datan, CRC_H,CRC_L

注: 该模式下, CRC 不计算命令头部分

b. 发送复位命令: 0x06, 不需要回复

1.4 简单模式

a. 发送启动命令,

命令格式: 0x46,0x4D,0x44,0x49,0x41,0x50,0x55,0xAA

启动成功回复: 0xAA

注：失败则重试 2 次

b. 擦除，

命令格式：0x04, Addr_H, Addr_L

成功回复：00；

c. 写数据,每次写 64Byte,

数据包格式：0x05, Addr_H, Addr_L, Data0 Data64, CRC_H, CRC_L

成功回复：0x00, 失败回复：0x44(失败重复写 2 次)

d. 读数据验证，

命令格式：0x03, Addr_H, Addr_L

回复：64 字节数据

e. 发送复位命令：0x06

成功回复：00；

2 程序说明

2.1 IAP DEMO 特性

- 通过 USART 接收上位机命令和数据；
- 串口通信波特率 115200(可根据实际情况自行修改)，8 位数据位，1 位起始位，1 位停止位；
- 数据发送和接收引脚分别为 PA6 和 PA7(或者重映射到 PB6 和 PA2)；
- 进入升级 APP 程序需接收特定格式的命令：0x46、0x4D、0x44、0x49、0x41、0x50、0x55、0xAA，共 8Byte。

2.2 闪存映射

DEMO 中 IAP Bootloader 程序位于闪存区域的前 1024 * 14bit 区域，上电复位的入口地址和中断地址被 IAP 程序占用，为了使用户的程序能够正常工作，需要重映射用户的复位入口地址和中断入口地址，如下图所示。

IAP 实现时，如果选择进入用户程序，则直接从 0x00 地址跳转到 0x400 地址去执行用户程序；当中断程序到来时，则直接跳转到 0x404 区域执行用户的中断程序。

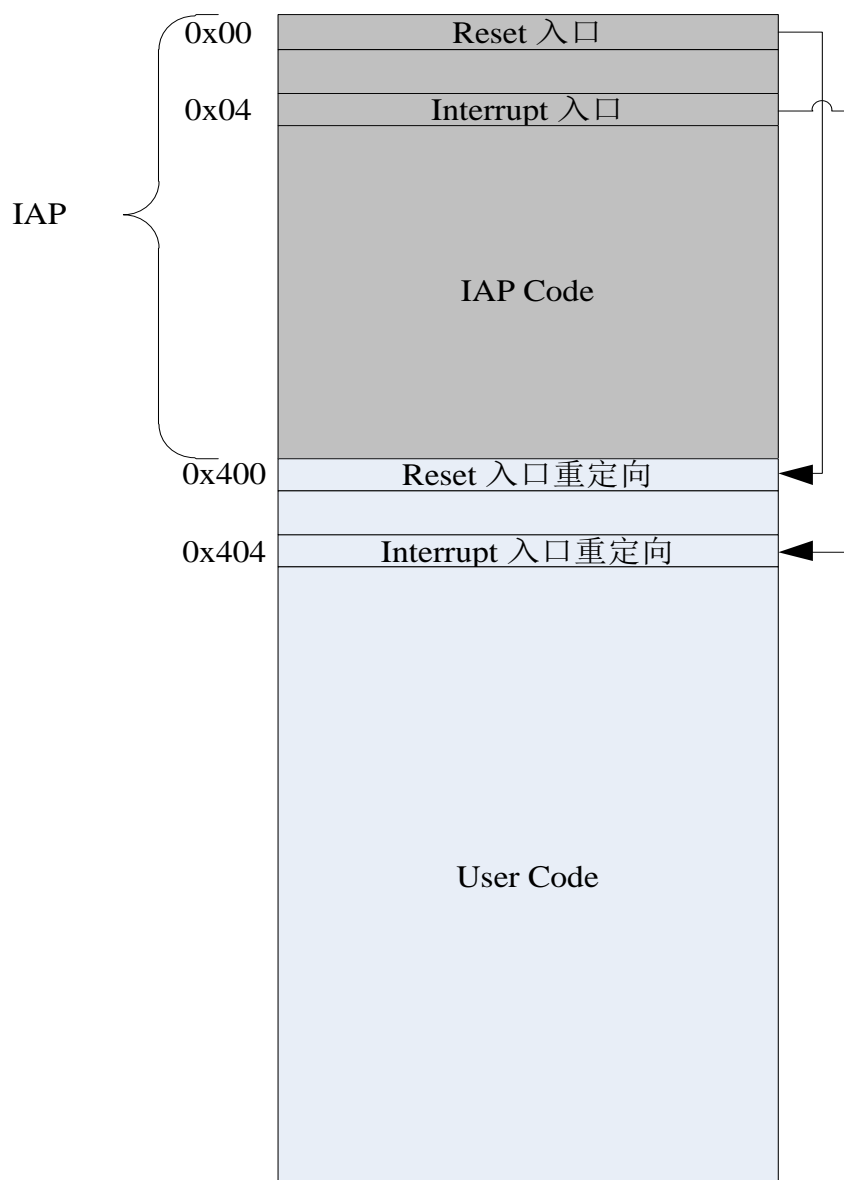


图 2-1 FLASH 闪存区域映射

2.3 IAP 程序设置流程

a. 标准 IAP 程序流程

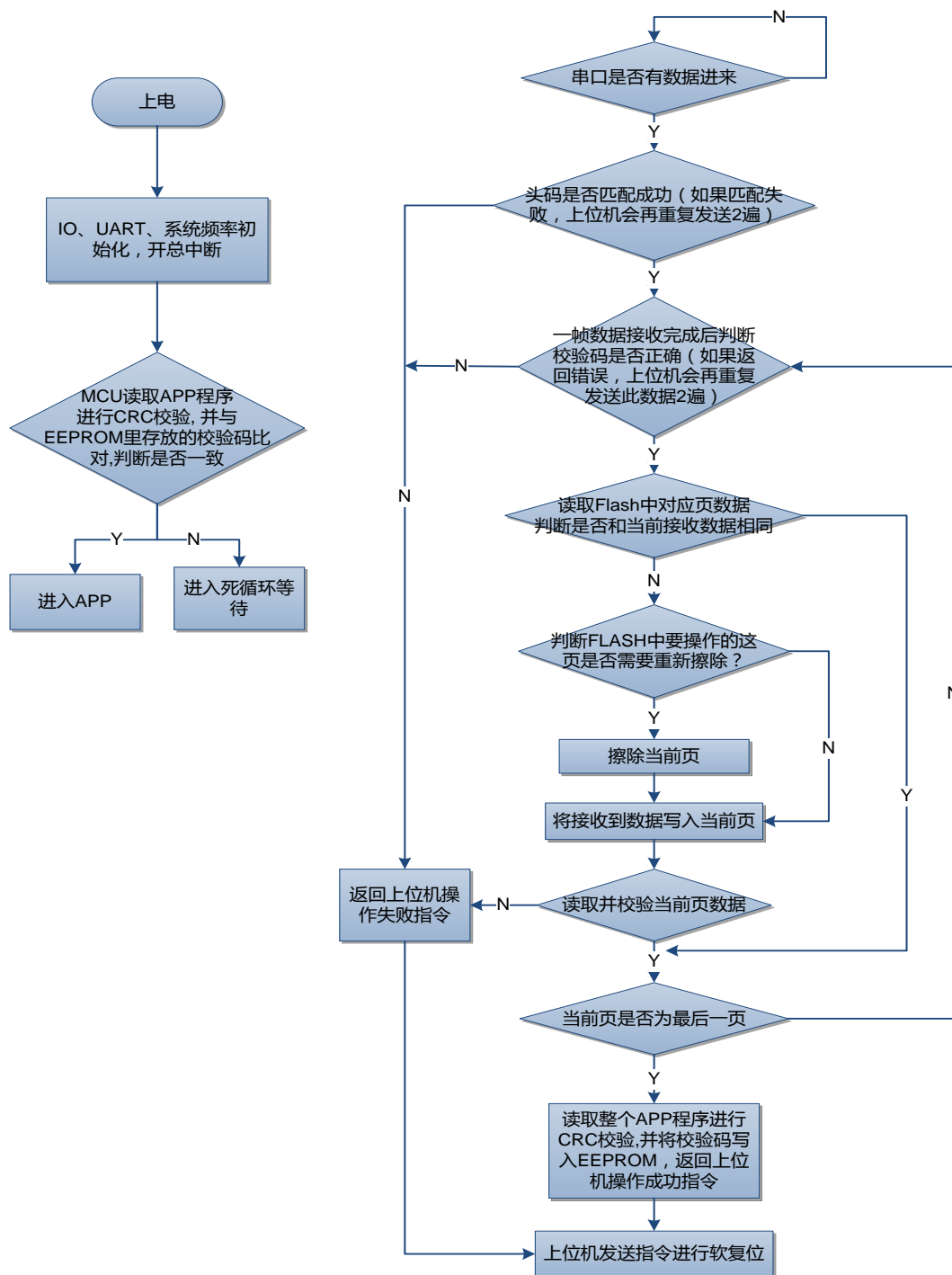


图 2-2 标准 IAP 程序流程

b. 一对多 IAP 流程图

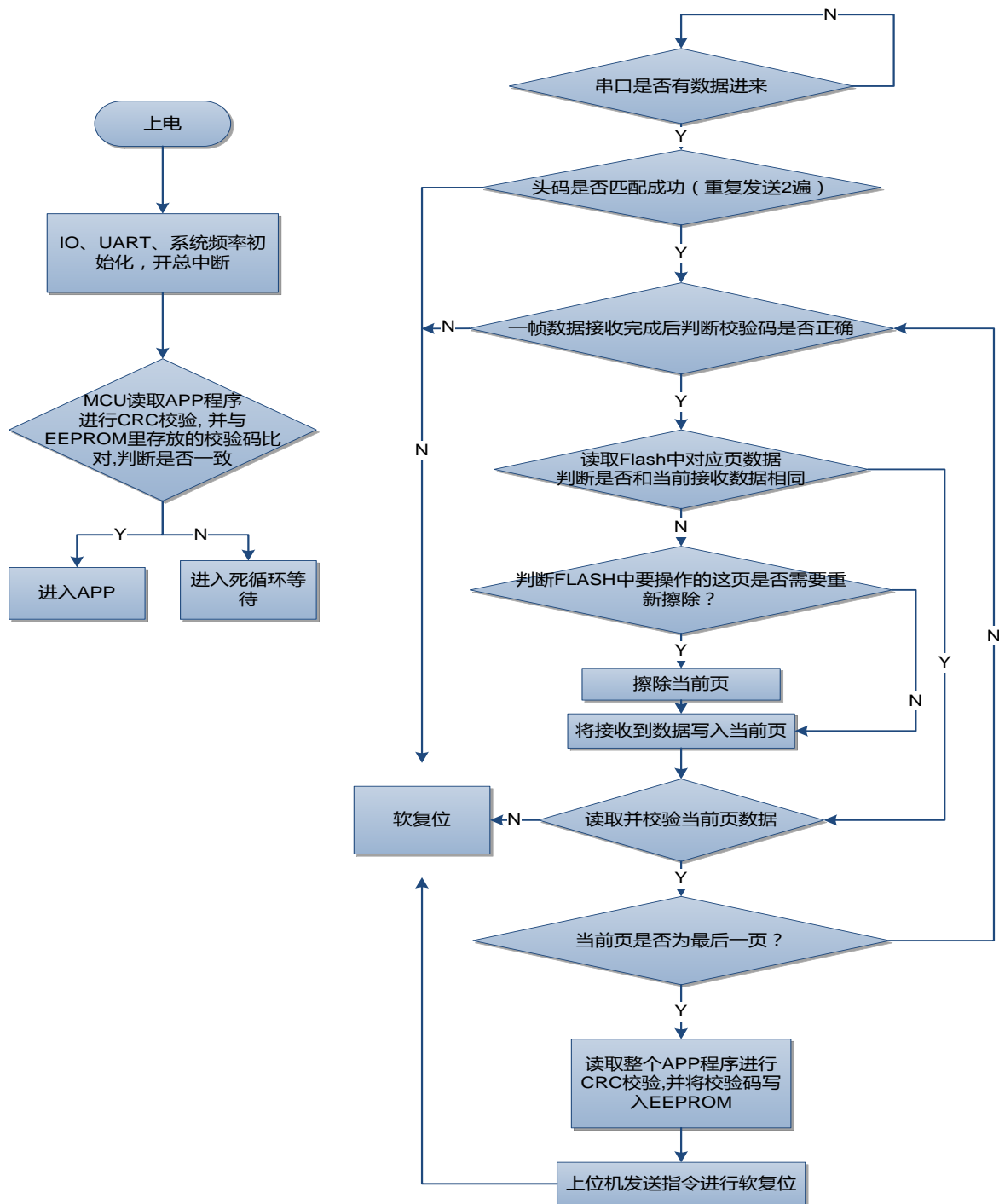


图 2-3 一对多 IAP 流程图

2.4 IAP 程序注意事项

a. IAP 程序跳转到应用程序的注意事项

- IAP 程序中有两个跳转到应用程序的函数, 分别为 main() 和中断函数。

Main()函数中跳转的语句为: `asm("goto 0x400");`

中断函数中跳转的语句为：asm("goto 0x404");

- 当 IAP 程序空间超过 2K 时，跳转应用程序的语句前需设置“PCLATH=XX ”的赋值语句，以实现跳转过程中的程序翻页处理，如下例程：

```
PCLATH = 0x06;           //设置 PC 指针高位
```

```
asm("goto 0x604");       //跳转语句
```

- IAP 程序定义的全局变量需固定地址，并且在应用程序中也需要同样声明，如：

```
unsigned char flag@0x20;
```

```
bit read_flag          @ ((unsigned)&flag*8)+0;
```

```
bit iap_err_flag       @ ((unsigned)&flag*8)+1;
```

```
bit iap_ok_flag        @ ((unsigned)&flag*8)+2;
```

```
bit uart_rxf          @ ((unsigned)&flag*8)+3;//应用程序所需要的串口数据标志
```

```
unsigned char count @0x21;
```

```
unsigned char iap_recdata @0x22;
```

```
unsigned int CRC_16@0x23;
```

```
unsigned char CRC_L@0x23;
```

```
unsigned char CRC_H@0x24;
```

- 应用程序的初始化，不能改变串口引脚的状态，以及中断功能；
- IAP 为保证在线升级的正确性，升级成功后会对整个应用程序空间进行 CRC 校验，并将校验和存储到 EEPROM（例程写在第 0 位和第 1 位）中，因此应用程序不能修改此位置数据；
- 由于 IAP 程序进入中断时，编译器会在某些位置自动存储一些信息，退出中断时自动释放，但其位置不是固定的，需要从 IAP 程序的汇编调试窗口查看，为避免影响应用程序的正常运行，需要在应用程序中声明这些位置，例如下图中的 0x7f 位置以及 0x7A 位置：

```

3 // Memory: Flash 10KX14b, EEPROM 128X8b, SRAM 2KX8b
4 //=====
5 //
6 #include "SYSCFG.h";
7 #include "usart_drv.h"
8 #include "mem_drv.h";
9 //=====
10 //Variable definition
11 //=====
12
13 unsigned char flag@0x20;
14 bit read_flag @ ((unsigned)&flag*8)+0;
15 bit iap_err_flag @ ((unsigned)&flag*8)+1;
16 bit iap_ok_flag @ ((unsigned)&flag*8)+2;
17
18 unsigned char count @0x21;
19 unsigned char iap_recdata @0x22;
20 unsigned int CRC_16@0x23;
21 unsigned char CRC_L@0x23;
22 unsigned char CRC_H@0x24;
23 unsigned char iap_databuf[133];

```

```

3 //-----Variable END-----
4 ORG 0000H
5 MOVLP 1H //0000 0181
6 LJUMP 178H //0001 3978
7 ORG 0004H
8 MOVLP 0H //0004 0180
9 MOVLP 0H //0005 1020
10 LDR 7FH, 0H //0006 187F
11 STR 7AH //0007 10FA
12
13 //;IAP_FW.C: 137: if(UR1RXNE && UR1R
14 MOVLP 9H //0008 1029
15 BTSC EH, 0H //0009 280E
16 BTSS 12H, 0H //000A 2C12
17 LJUMP 173H //000B 3973
18 ORG 000CH
19
20 //;IAP_FW.C: 138: {
21 //;IAP_FW.C: 139: if(UR1FEF==1)
22 BTSC 12H, 3H //000C 2992
23

```

图 2-4 变量位置声明

```
unsigned char information1 @0x7a;
```

```
unsigned char information2 @0x7f;
```

b. 工程设置

在仿真器界面，右键点击 Device -> Project Options... 如下图：

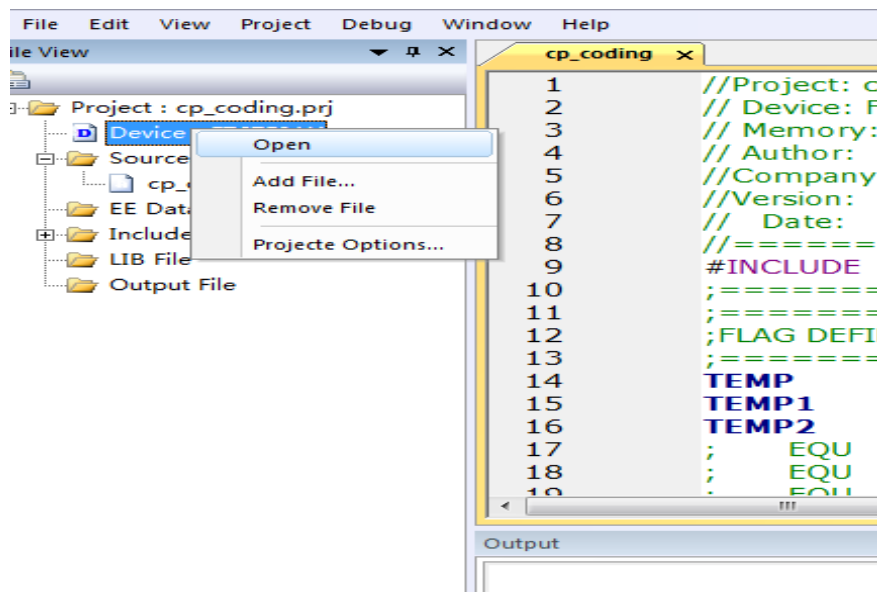


图 2-5 工程设置

- IAP 程序：设置开始地址(首地址)和 IAP 程序所占空间的结束地址，如下图：

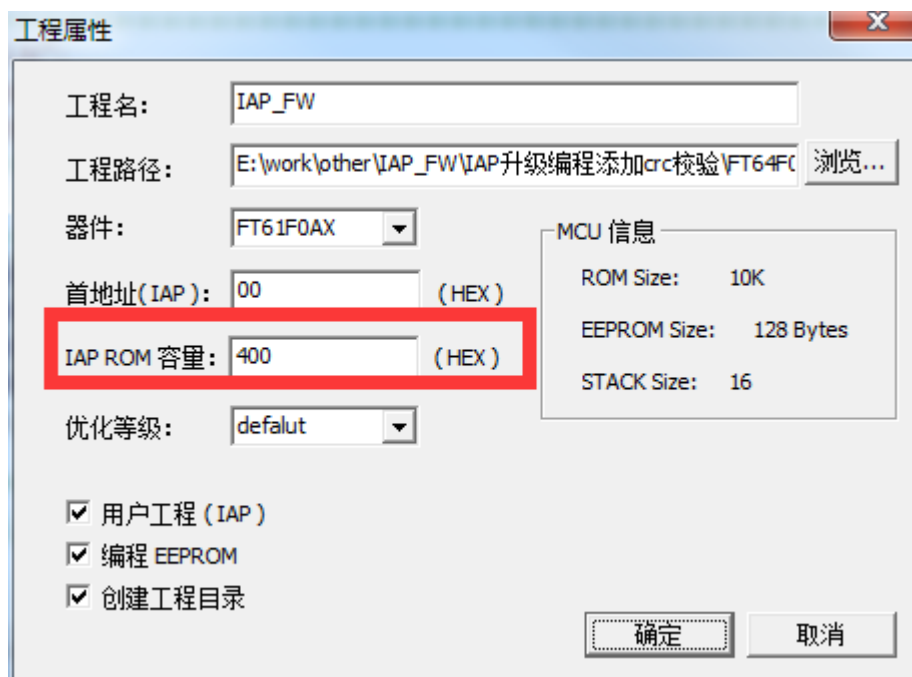


图 2-6 IAP 地址配置

- 应用程序：设置偏移地址(首地址)和 IAP 程序所占空间的结束地址，如下图：

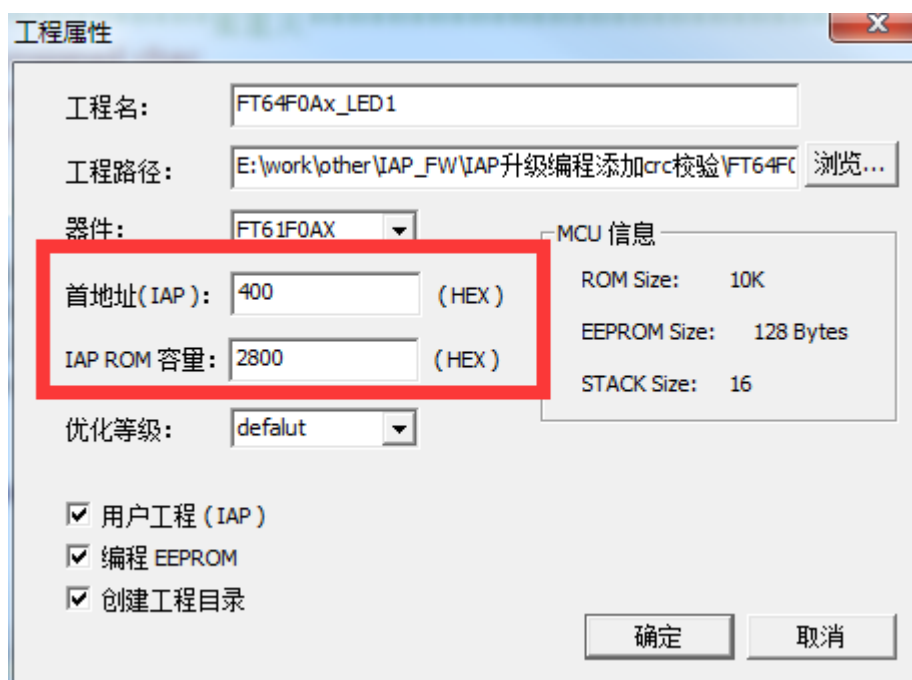


图 2-7 应用程序地址配置

如果应用程序用汇编语言编写，则程序中复位向量入口和中断入口都需要进行偏移设置，设置如下：

```
ORG      0x0400          ; 单片机复位向量入口
```

```
LJUMP    RESTART
```

ORG 0x0404 ; 中断复位向量入口

LJUMP INT_PROGRAM

注意：无论用 C 或汇编语言编写应用程序，应用程序中必须有中断函数。若 IAP 串口接收到的不是进入编程的命令数据，则会跳到应用程序的中断函数，但若应用程序中没有中断函数，则后续就跳不回 IAP 中断函数，程序会跑乱。

2.5 IAP 在线升级示例

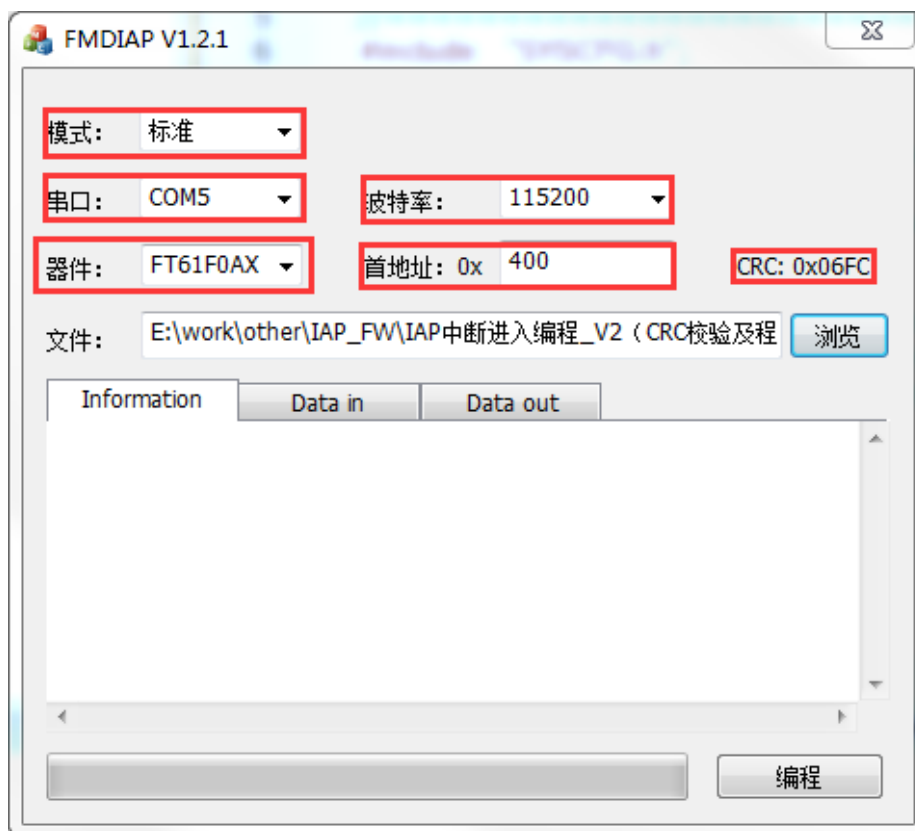


图 2-8 IAP 在线升级示例

- 模式：选择 IAP 所对应的工作模式（标准、简单、一对多）；
- 串口：配置串口对应的电脑端口（COM5）；
- 器件：选择所用的 FMD 芯片；
- 波特率：设置 IAP 程序所配置的波特率(有 4 种常用波特率可供选择)；
- 首地址：升级的 APP 程序开始的地址（示例中 IAP 程序所占用的空间为 0x000-3FF，因此需要更新程序的首地址为 0x400）；
- CRC：点击浏览导入所需要升级的 HEX，会自动显示所导入程序的校验和；

点击编程：



图 2-9 在线编程成功

烧录完成后提示如上图所示（其中 Data in 是所导入的 hex 数据）。

2.6 IAP 及应用程序组合烧录流程

打开 FMD 烧录器软件界面(FMD Link 或 FMD Writer 工具均可)，点击器件/设置，如下图，点击器件 list 中所对应的芯片型号，导入 IAP 程序和对应的应用程序，以及 EEPROM 文件（第 0 和 1 位需为指定数据）点击确定：

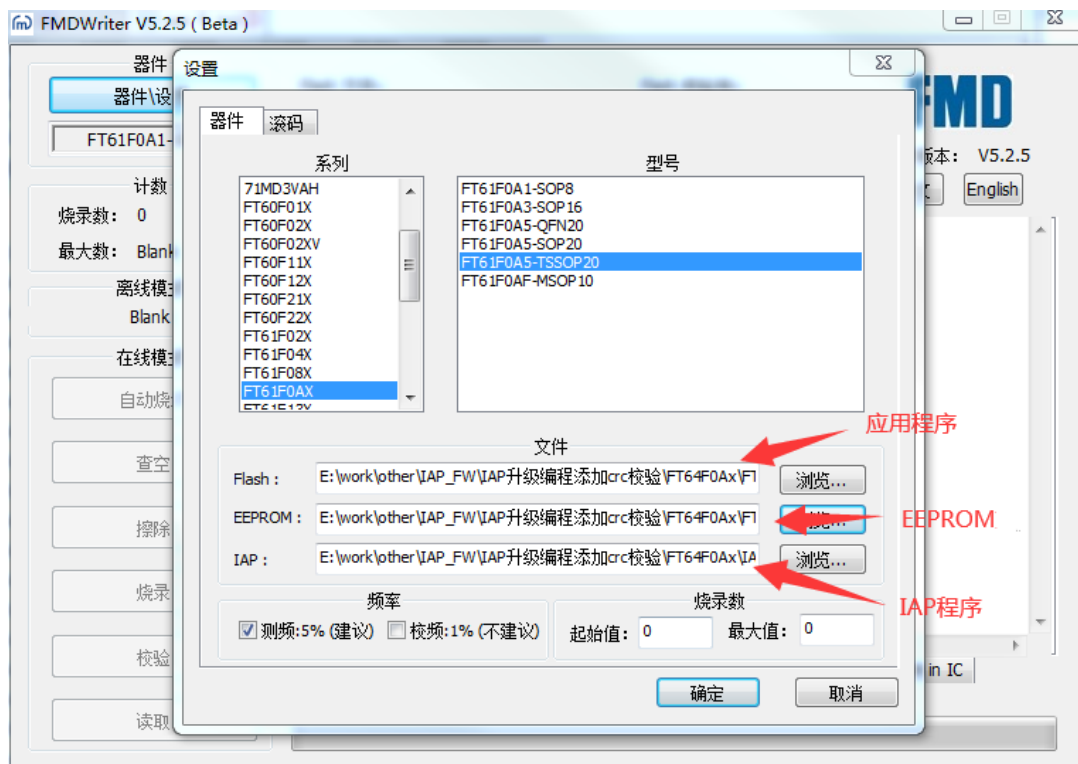


图 2-10 烧录器导入程序

为保证应用程序的正常运行，EEPROM 的指定位置必须为应用程序的校验和（3F A2 为应用程序的 CRC 校验和）；

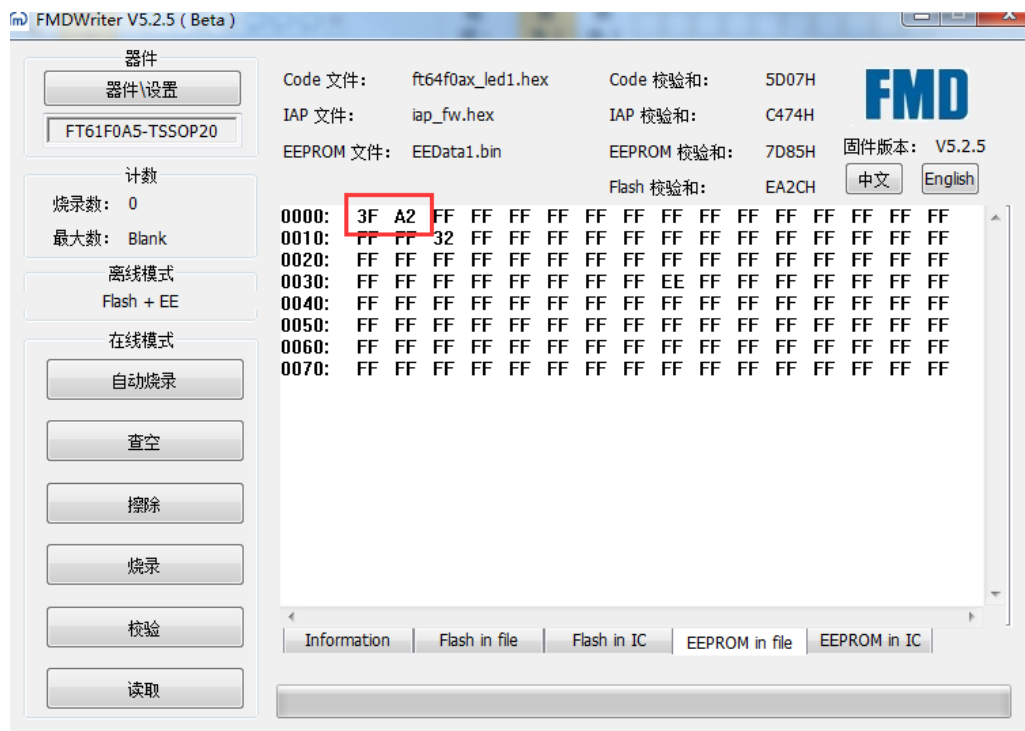


图 2-11 EEPROM 存放的 CRC

按照烧录器的操作流程，正常烧录即可。

2.7 简单版 IAP 介绍

- 程序容量：相对于标准版 IAP，简单版 IAP 程序所占内存(<512 words)较小，只在 EEPROM 第 0 位写一个标志位，校验方式相对简单，侧重于依靠上位机来完成校验，烧录时间固定不变。
- 烧录时间：简单版 IAP 采用全擦除再写入，更新时间固定；标准版 IAP 会先对比芯片里的数据与需要更新的数据，不一致的时候才会更新，因此时间不是固定的，与更新的程序内容有关。

2.8 一对多版 IAP 介绍

- 通信方式：相较于标准版 IAP 程序需要跟上位机进行双向通信，一对多版 IAP 程序则无需应答上位机，只从上位机接收数据，所以能同时给多台设备升级。相邻页数据传输时间间隔固定，故烧录时间固定不变。除了无需应答上位机，一对多版的数据擦除及更新方式与标准版无差异。

2.9 示例版本差异对比

工作模式	USART 通信	传输方向
标准	是	双向
简单	是	双向
一对多	是	单向

表 2-1 通信方式差异

波特率 115200	标准	简单	一对多
FT61F14X IAP 容量	691	499	687
FT62F08X IAP 容量	774	502	822
FT61F0AX IAP 容量	768	496	816

表 2-2 所占容量大小差异

FT61F0AX	标准		简单		一对多	
	应用程序 1K	应用程序 9K	应用程序 1K	应用程序 9K	应用程序 1K	应用程序 9K
烧录时间	3.73s	13.5s	15.5s	15.5s	20.3s	20.3s

表 2-3 烧录时间差异

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>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.