

## 简介

IAP (In Application Programming) 即在应用程序编程，是一种自举程序。由于产品固化后不容易采用传统下载器更新固件使得许多产品中内置Bootloader程序用于远程更新固件，有的应用产品在产品固化后，只预留了SPI、UART等通信接口，所以如果需要固件更新，只能考虑使用预留的通信接口进行更新固件操作，在一些上网设备或者无线设备上，也会使用到该功能，例如通过BLE、WIFI等无线通信进行固件更新，我们称为OTA (over the air technology) 空中下载技术，也是利用到了IAP升级的原理。

HolyChip Flash系列支持第二复位向量的MCU的功能均支持此功能。

实现 IAP 功能时，即用户程序需要在运行中对自身的功能程序进行更新操作，需要在设计固件程序时编写两个项目代码，第二个项目程序不执行正常的功能操作，而只是通过某种通信方式（如UART、SPI、IIC等通信接口）接收程序或数据，执行对第一部分代码的更新，称之为 Bootloader 程序；第一个项目代码才是真正的功能代码，称之为 APP 程序。这两部分项目代码都同时烧录在Flash的不同地址范围，并且可以通过软件复位的方式随意切换。本应用笔记将以HC88F0671为范例介绍此功能。

- 本应用笔记适用芯片：HC88F0671、HC88F0661、HC88F0651、HC89S003F4、HC89S105C6、HC89S105C8、HC89S105K6、HC89S105K8、HC89S105AC8、HC89S105AK8、HC89F0421、HC89F0431、HC89F0531、HC89F0541、HC89F0630、HC89F301、HC89F302、HC89F303、HC89F3421、HC89F3531、HC89F3541、HC89F3650。（固化官方 ISP 程序的 MCU 不支持第二复位向量功能）
- 本应用笔记对应范例：HC88F06x1 Self-update V1.0.0.0。
- 相关数据手册、工具及技术文档下载网址：<http://www.holychip.cn/>。

# 目录

1、原理及流程图 .....	3
1.1 原理 .....	3
1.2 APP 程序流程图 .....	3
1.3 Bootloader 程序流程图 .....	4
2、函数介绍 .....	5
2.1 APP 函数介绍 .....	5
2.2 Bootloader 函数介绍 .....	6
3、软件使用 .....	8
3.1 CRC 校验 .....	8
3.2 Hex To Bin 使用 .....	9
4、程序协议定义及范例相关指令 .....	12
5、范例应用流程 .....	13
6、KEIL 工程配置及注意事项 .....	15
6.1 APP 程序工程配置 .....	15
6.2 Bootloader 程序工程配置 .....	17
6.3 注意事项 .....	错误！未定义书签。
7、版本说明 .....	19

# 1、 原理及流程图

## 1.1 原理

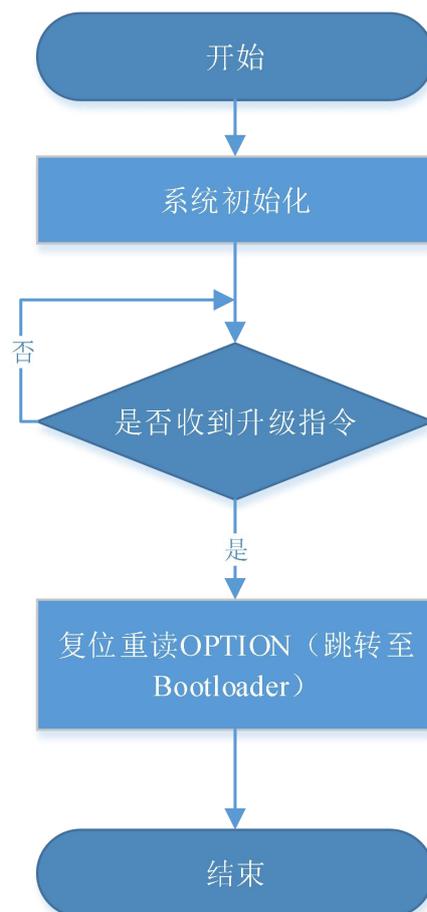
MCU 在运行 APP 程序时，等待上位发送的升级指令，接收指令后通过复位重读代码选项指令转至 Bootloader 程序区，随后 Bootloader 程序接收更新程序数据，通过 IAP 的操作方式对 APP 程序区更新。

用过可通过复位指令在 APP 以及 Bootloader 程序中相互切换：

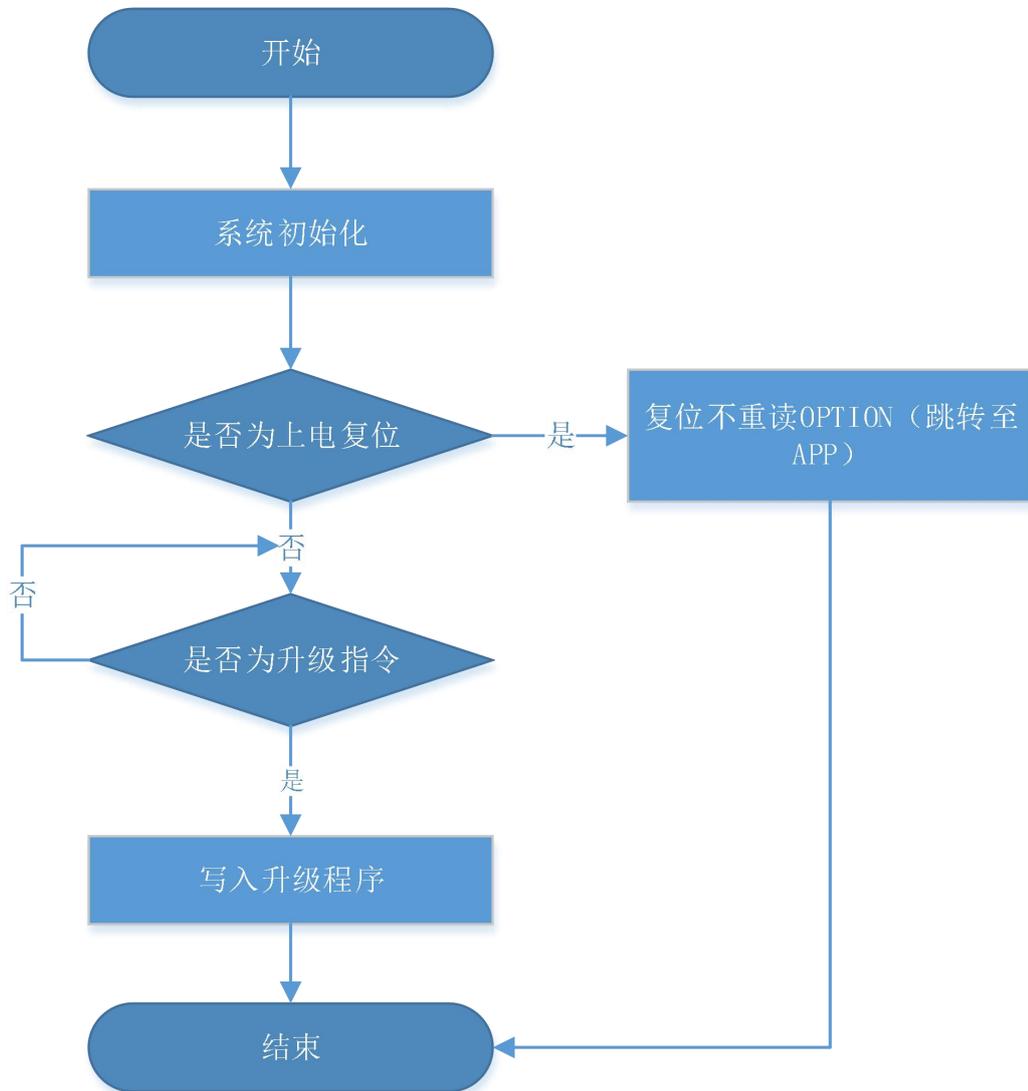
APP → Bootloader: 复位重读代码选项

Bootloader → APP: 复位不重读代码选项

## 1.2 APP 程序流程图



### 1.3 Bootloader 程序流程图



## 2、 函数介绍

### 2.1 APP 函数介绍

#### 2.1.1 SystemInit

**描述:** 系统初始化函数  
**C 语言原型:** void SystemInit(void);  
**输入参数:** 无  
**返回值:** 无

#### 2.1.2 CRC\_CalcCRC

**描述:** CRC 校验函数  
**C 语言原型:** unsigned int CRC\_CalcCRC(unsigned char \*fucp\_CheckArr,unsigned int fui\_CheckLen);  
**输入参数:** \*fucp\_CheckArr : CRC 校验数据首地址  
              fui\_CheckLen : CRC 校验数据长度  
**返回值:** CRC 校验结果, 低位在前, 高位在后

#### 2.1.3 ResetReadOption

**描述:** 复位重读代码选项函数  
**C 语言原型:** void ResetReadOption (void);  
**输入参数:** 无  
**返回值:** 无

#### 2.1.4 UART\_Send\_Str

**描述:** Uart 写字符串函数  
**C 语言原型:** void UART\_Send\_Str(unsigned char \*fucp\_Str);  
**输入参数:** \*fucp\_Str : 写入的数据的首地址  
**返回值:** 无

## 2.2 Bootloader 函数介绍

### 2.2.1 SystemInit

**描述:** 系统初始化函数  
**C 语言原型:** void SystemInit(void);  
**输入参数:** 无  
**返回值:** 无

### 2.2.2 CRC\_CalcCRC

**描述:** CRC 校验函数  
**C 语言原型:** unsigned int CRC\_CalcCRC(unsigned char \*fucp\_CheckArr,unsigned int fui\_CheckLen);  
**输入参数:** \*fucp\_CheckArr : CRC 校验数据首地址  
              fui\_CheckLen : CRC 校验数据长度  
**返回值:** CRC 校验结果, 低位在前, 高位在后

### 2.2.3 ResetNoReadOption

**描述:** 复位不重读代码选项函数  
**C 语言原型:** void ResetNoReadOption (void);  
**输入参数:** 无  
**返回值:** 无

### 2.2.4 UART\_Send\_Str

**描述:** Uart 写字符串函数  
**C 语言原型:** void UART\_Send\_Str(unsigned char \*fucp\_Str);  
**输入参数:** \*fucp\_Str : 写入的数据的首地址  
**返回值:** 无

### 2.2.5 Flash\_EraseBlock

**描述:** 扇区擦除, 约消耗 5ms 的时间  
**C 语言原型:** void Flash\_EraseBlock(unsigned int fui\_Address);  
**输入参数:** fui\_Address : 被擦出扇区的任意一个地址  
**返回值:** 无

## 2.2.6 FLASH\_WriteData

**描述：** 写入一个字节到 Flash 里面

**C 语言原型：** void FLASH\_WriteData(unsigned char fuc\_SaveData, unsigned int fui\_Address);

**输入参数：** fui\_Address : Flash 地址

fuc\_SaveData : 写入的数据

**返回值：** 无

## 2.2.7 Flash\_WriteArr

**描述：** 写入任意长度的数据到 Flash 里面

**C 语言原型：** void Flash\_WriteArr(unsigned int fui\_Address, unsigned char fuc\_Length, unsigned char \*fucp\_SaveArr);

**输入参数：** fui\_Address : Flash 地址

fuc\_Length : 写入数据长度

\*fucp\_SaveArr : 写入的数据存放区域的首地址

**返回值：** 无

## 3、 软件使用

压缩包解压后有一个工具文件夹，其中有本次例程中需要使用的常用工具。

### 3.1 CRC 校验

#### 1、 打开 GCRC.exe



#### 2、 选择算法



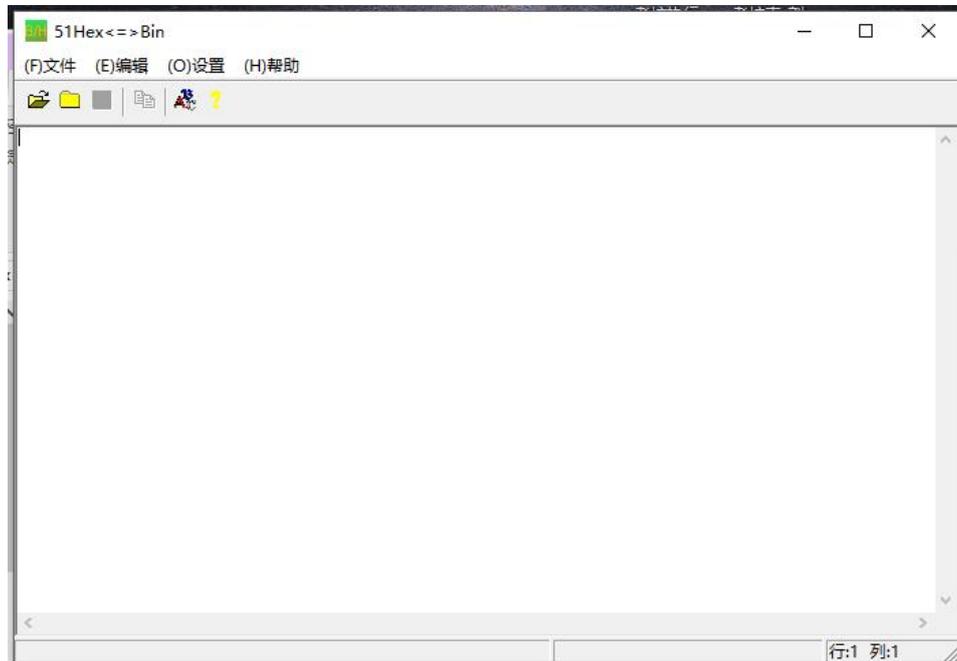
3、输入待计算数据，点击计算获得计算结果



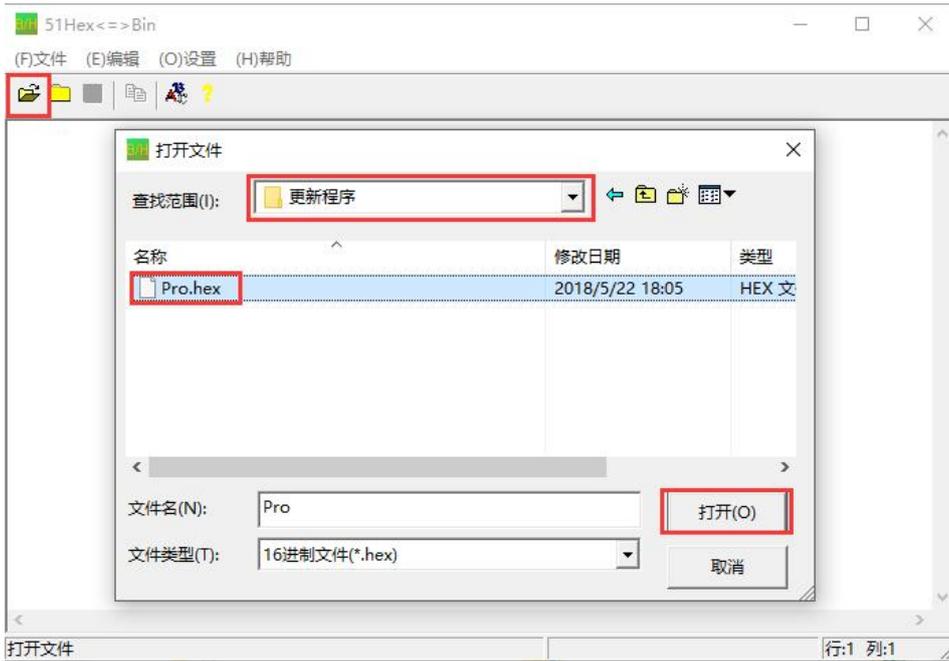
4、其他计算不一举例，计算方式与此相同。

## 3.2 Hex To Bin 使用

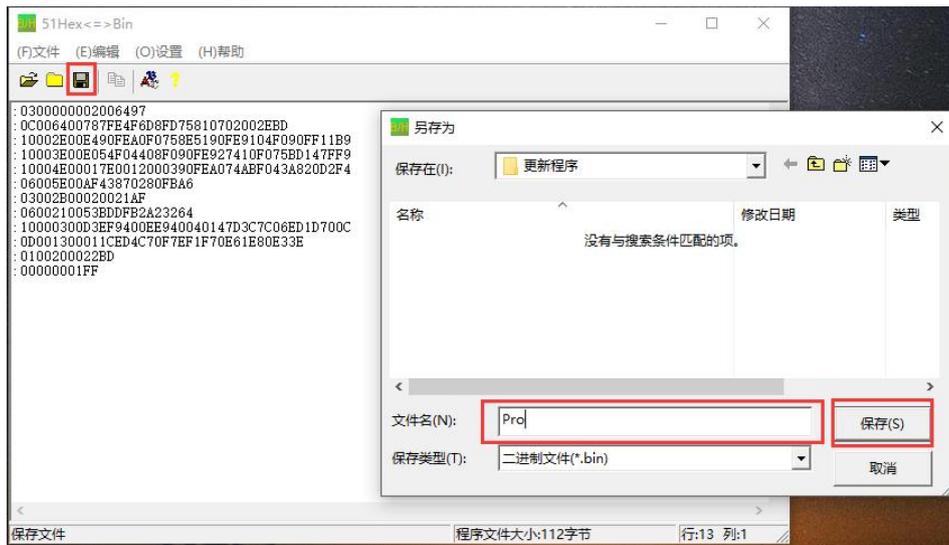
1、打开 Bin\_Hex.exe



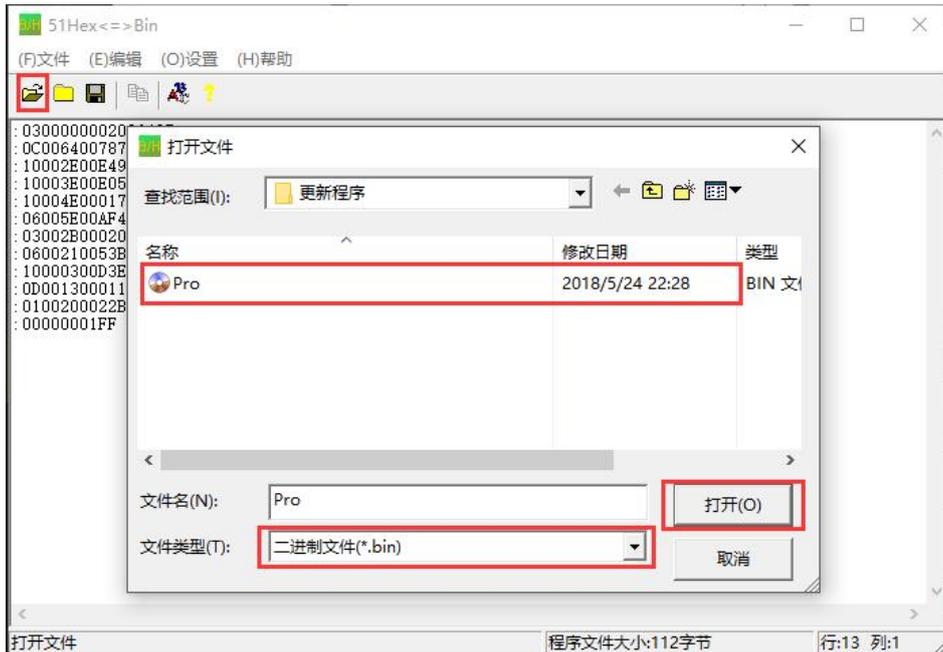
2、打开压缩包中的更新程序文件夹选择其中的 hex 文件



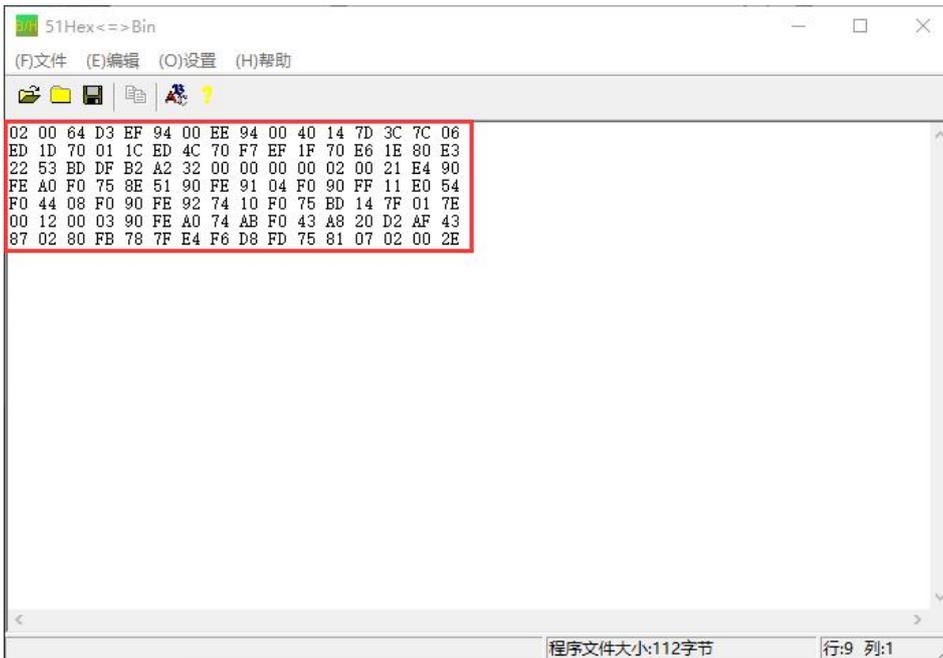
3、将文件保存在相同文件夹下



4、再次通过 Hex To Bin 打开刚刚保存的 Bin 文件



5、从而得到转换好的 Bin 文件



## 4、程序协议定义及范例相关指令

命名	A	B1	C1	D1D2	Ex	F1F2
序号	1	2	3	4	5	6
举例	0xAA	05	01	-	-	0x2A 0xAF
序号	名称	说明				
1	帧头	数据帧头，若帧头错误返回相应提示				
2	数据长度	数据总长度				
3	指令	0x01: 进入 Bootloader 程序 0x02: 开始 APP 程序更新				
4	待更新 APP 程序地址	如此指令非开始 APP 程序更新指令，可忽略				
5	待更新 APP 程序数据	如此指令非开始 APP 程序更新指令，可忽略 待更新 APP 程序数据最大 128				
6	CRC 校验值	校验值低位在前，高位在后				

### HC89S003F4 Self-update V1.0.0.0 例程升级指令

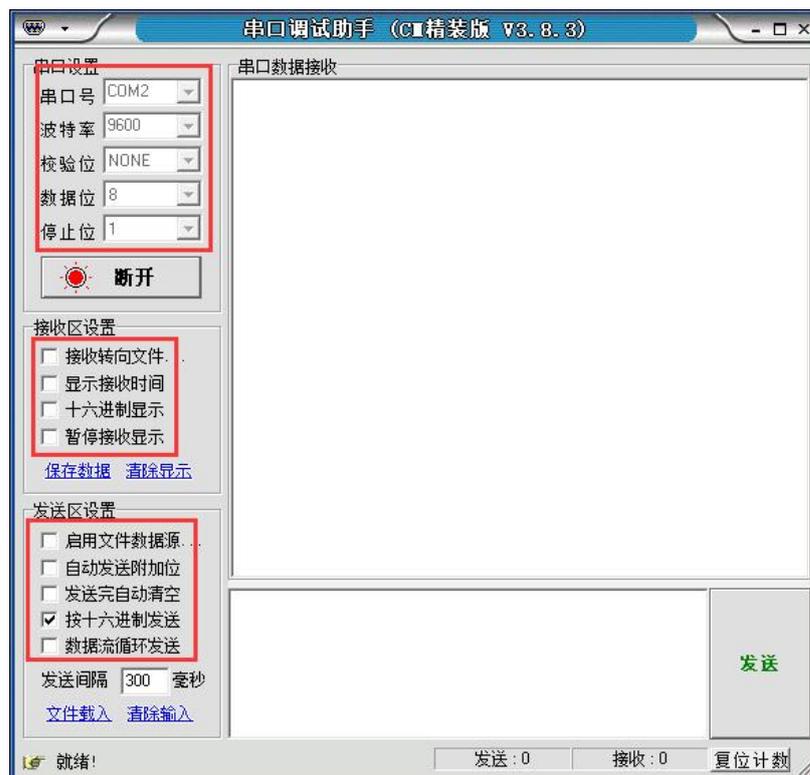
进入 Bootloader	AA 05 01 2A AF
写入更新程序	AA 77 02 00 00 02 00 64 D3 EF 94 00 EE 94 00 40 14 7D 3C 7C 06 ED 1D 70 01 1C ED 4C 70 F7 EF 1F 70 E6 1E 80 E3 22 53 BD DF B2 A2 32 00 00 00 00 02 00 21 E4 90 FE A0 F0 75 8E 51 90 FE 91 04 F0 90 FF 11 E0 54 F0 44 08 F0 90 FE 92 74 10 F0 75 BD 14 7F 01 7E 00 12 00 03 90 FE A0 74 AB F0 43 A8 20 D2 AF 43 87 02 80 FB 78 7F E4 F6 D8 FD 75 81 07 02 00 2E CF 90

## 5、 范例应用流程

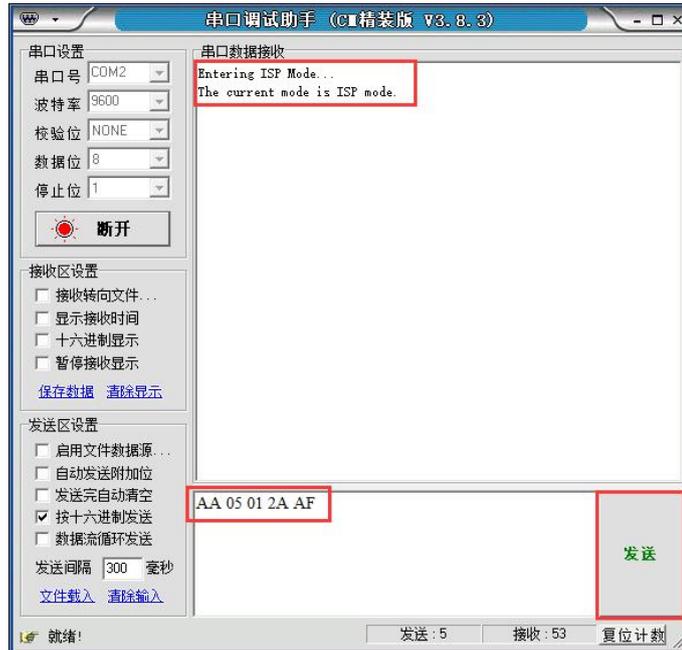
- 1、 打开 HC88F06x1 Self-update V1.0.0.0 文件夹中的 APP 文件夹中的 Keil 工程文件编译后下载入 DEBoard。
- 2、 打开 HC88F06x1 Self-update V1.0.0.0 文件夹中的 Bootloader 文件夹中的 Keil 工程文件编译后下载入 DEBoard。（DEBoard 的使用方法详见《UM0100\_HCFlash MCU DEBoard 使用说明》）。
- 3、 接上 DEBoard 左端的 USB 线此时可以看到 LED2 灯常亮，然后打开工具文件夹中的 UartAssist.exe 并做好相关配置。

串口波特率 9600、无校验位、8 位数据位、1 停止位

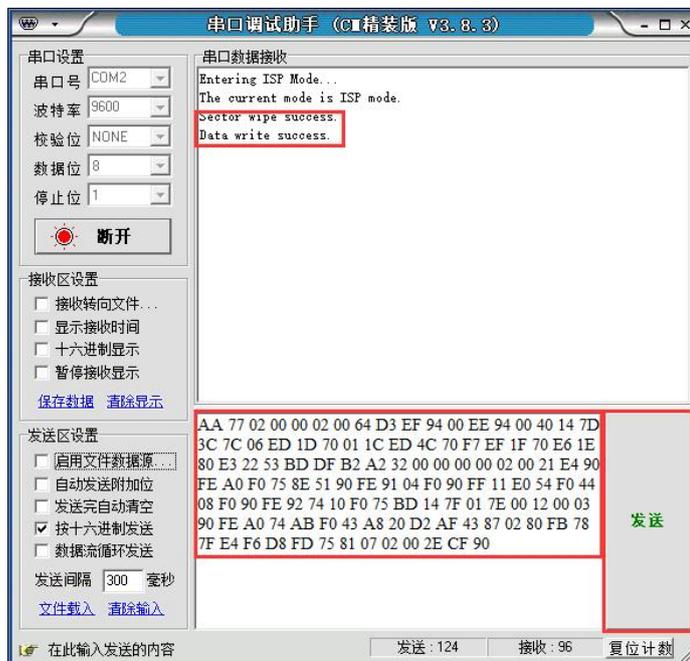
接收区选择非十六进制显示、发送区选择按十六进制发送



- 4、在发送窗口输入 AA 05 01 2A AF 指令，并点击发送。指令定义详见[程序协议定义及范例相关指令](#)章节。指令可直接在升级指令.TXT 中获取复制。



- 5、此时可以看到接收端口返回相应的指令并且 DEBoard 上的 LED2 熄灭，LED1 点亮。然后再在发送窗口输入 AA 77 02 00 00 02 00 64 D3 EF 94 00 EE 94 00 40 14 7D 3C 7C 06 ED 1D 70 01 1C ED 4C 70 F7 EF 1F 70 E6 1E 80 E3 22 53 BD DF B2 A2 32 00 00 00 00 02 00 21 E4 90 FE A0 F0 75 8E 51 90 FE 91 04 F0 90 FF 11 E0 54 F0 44 08 F0 90 FE 92 74 10 F0 75 BD 14 7F 01 7E 00 12 00 03 90 FE A0 74 AB F0 43 A8 20 D2 AF 43 87 02 80 FB 78 7F E4 F6 D8 FD 75 81 07 02 00 2E CF 90 指令并点击发送。

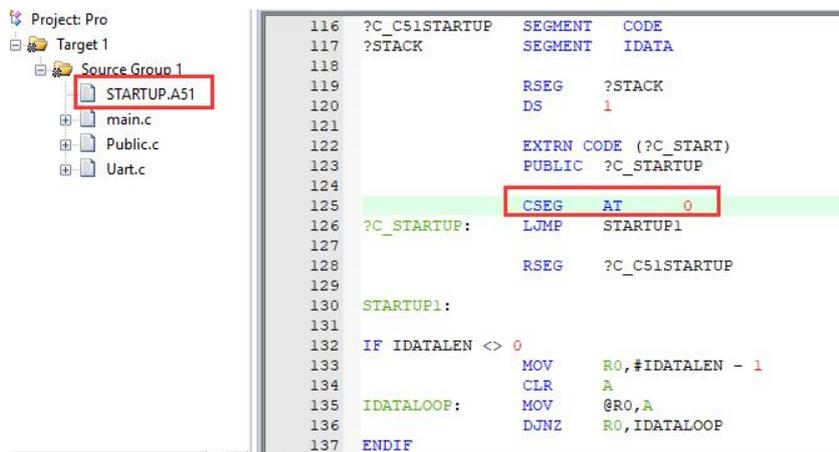


- 6、数据接收窗口提示数据写入成功后，拨下 USB 接头重新给 DEBoard 上电，此时 LED1 闪烁，程序更新成功。

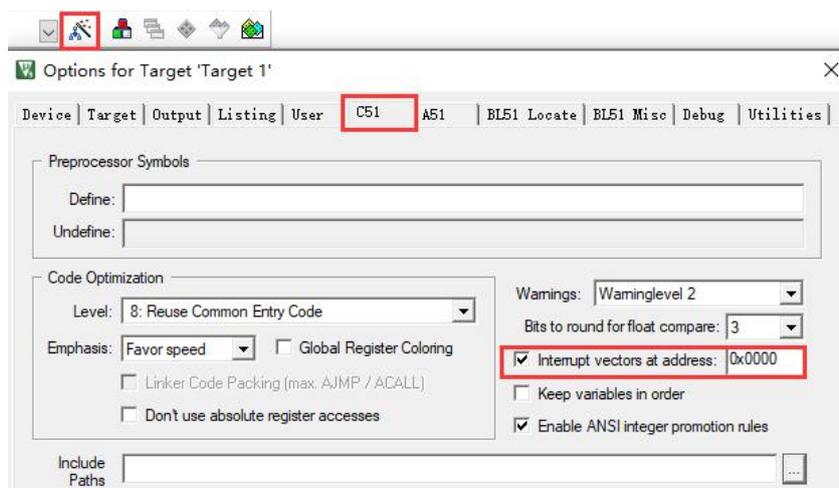
## 6、 KEIL 工程配置及注意事项

### 6.1 APP 程序工程配置

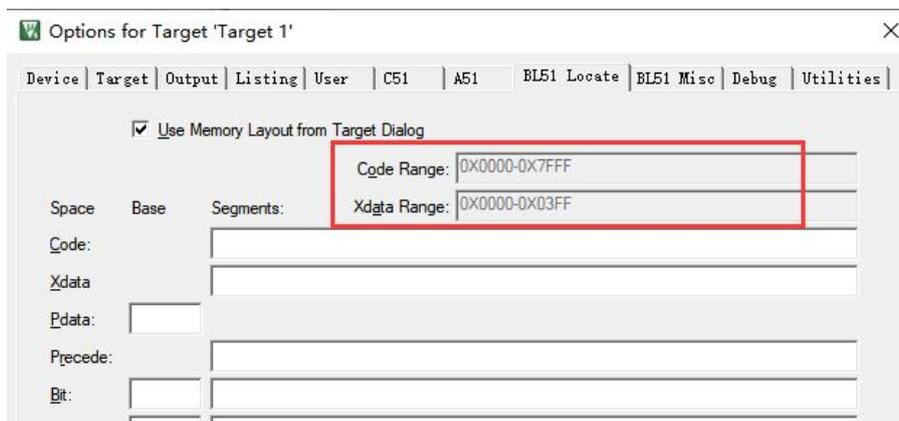
1、 STARTUP.A51 文件中的起始地址需配置为 0。



2、 OPTION 中的 C51 选项卡需要将中断地址配置为 0x0000



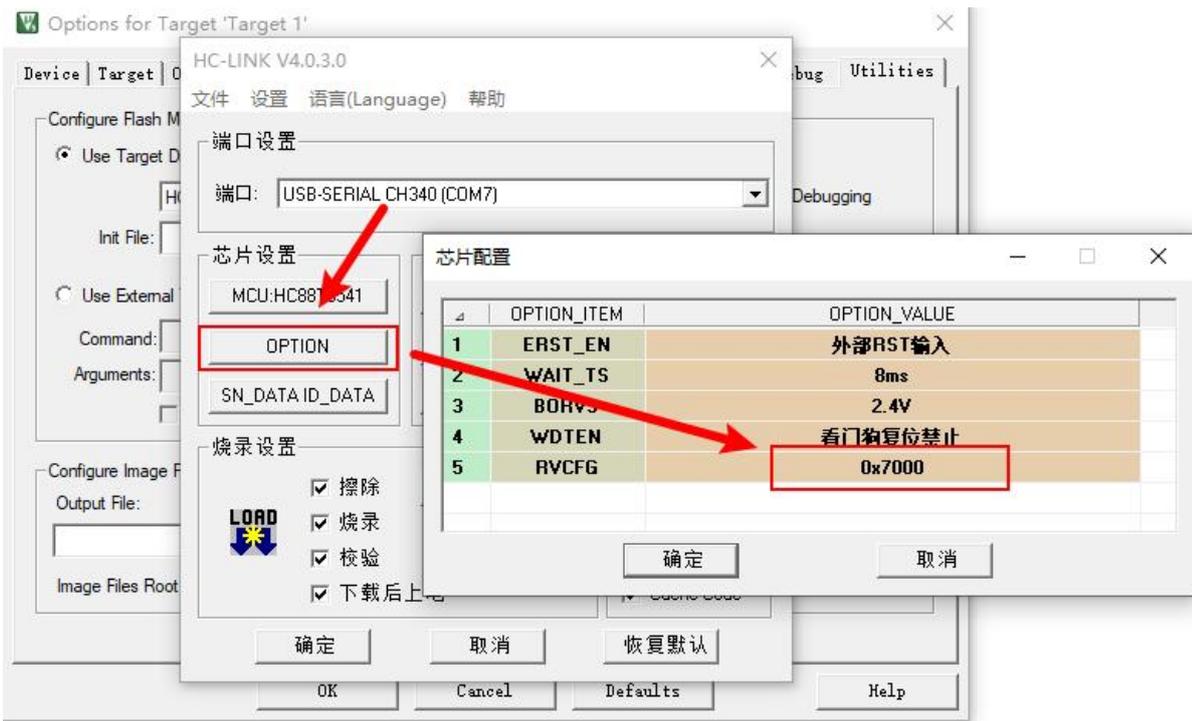
3、 内存区间可设置为默认区间



4、HC-LINK 下载选择程序所有存储区设置全擦除

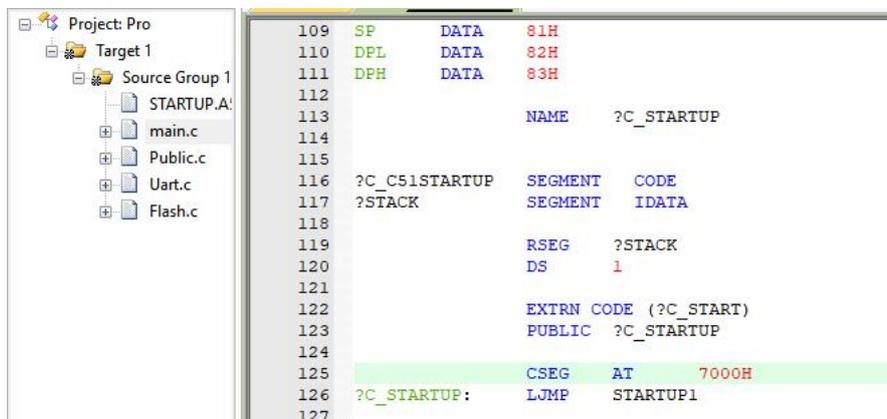


5、代码选项中需要使能第二复位向量并且将第二复位向量地址设置为 0x7000，如下图所示

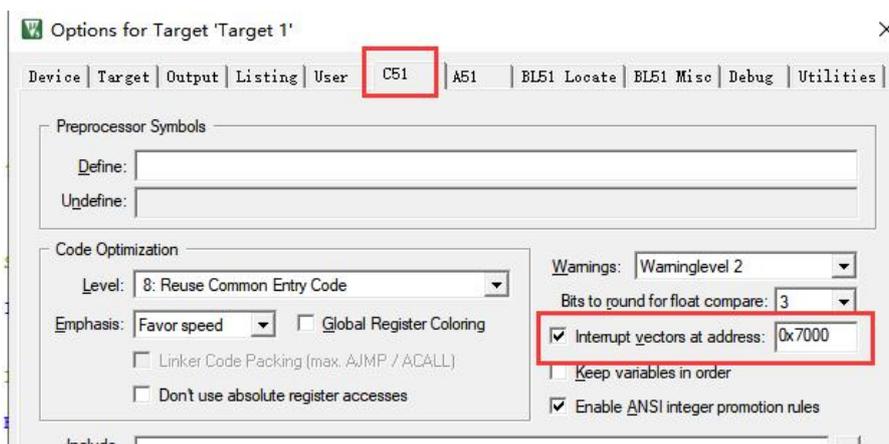


## 6.2 Bootloader 程序工程配置

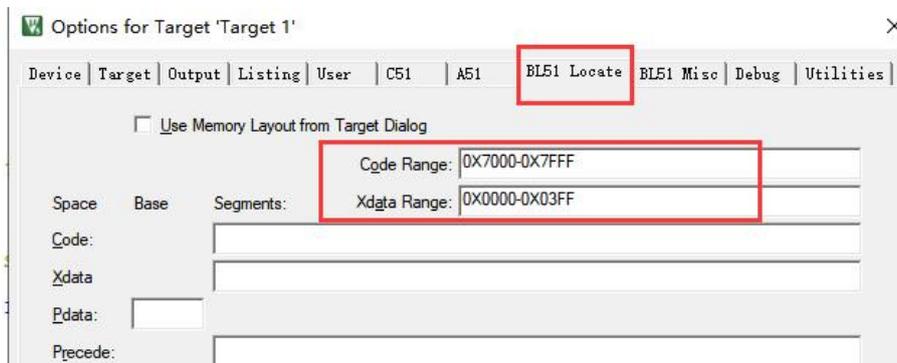
1、STARTUP.A51 文件中的起始地址需配置为 7000H。



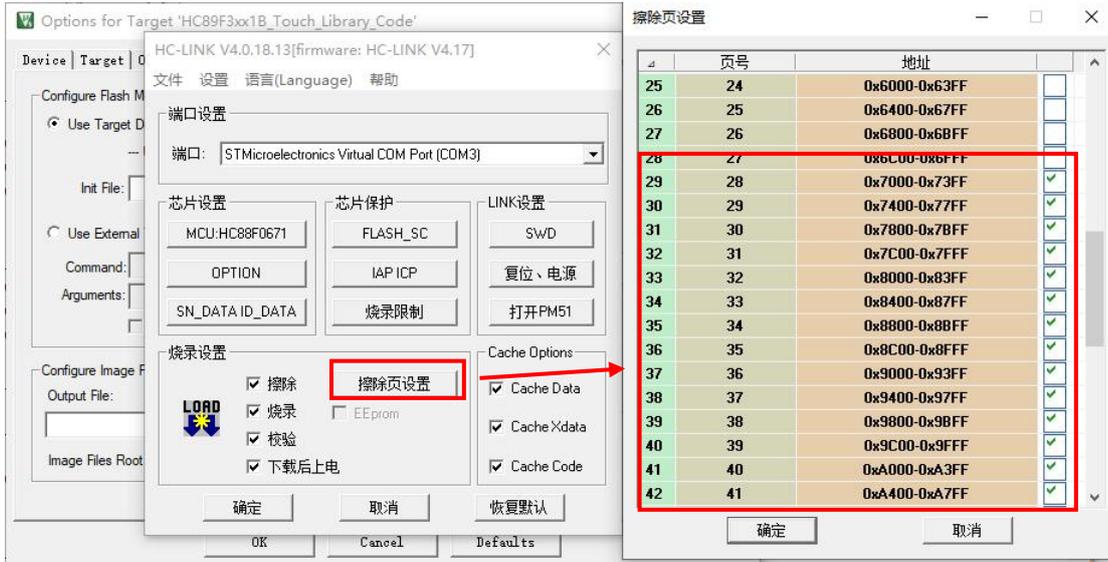
2、OPTION 中的 C51 选项卡需要将中断地址配置为 0x7000。如下图所示



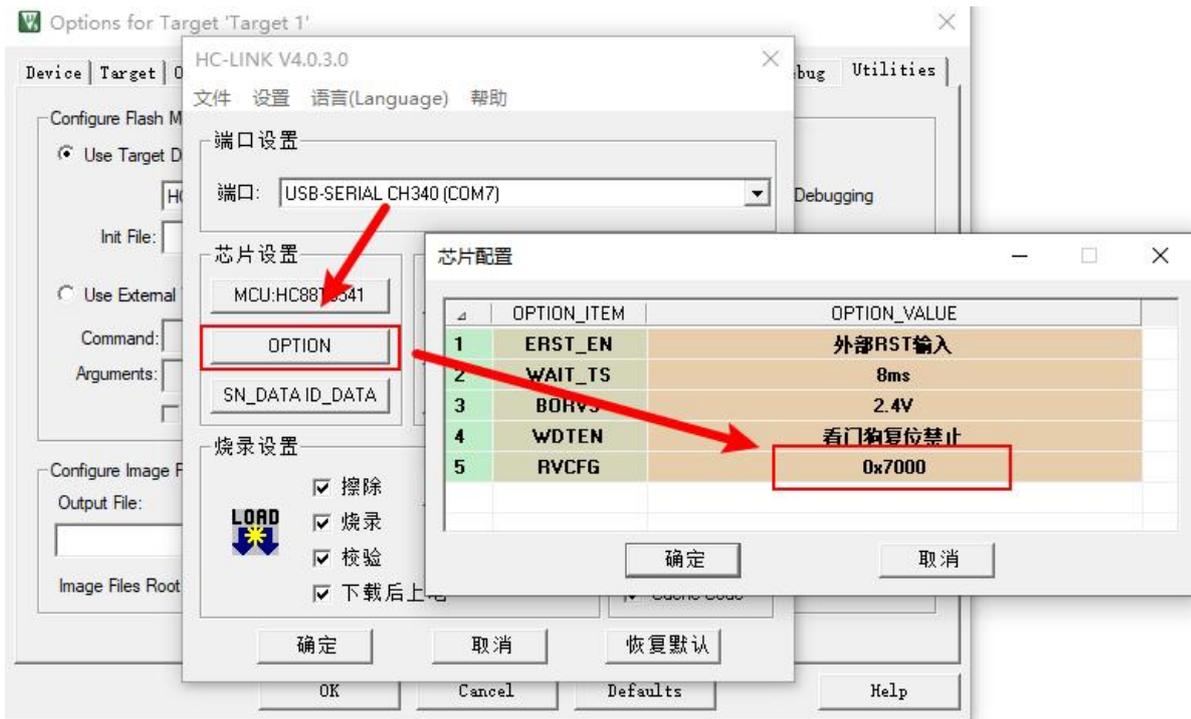
3、内存区间地址需设置为 0x7000-0x7FFF



4、页擦写选择第 28、29、30、31 页，并需要打开 Option 擦写开关。如下图所示



5、代码选项卡中需要使能第二复位向量并且将第二复位向量地址设置为 0x7000（新版 link 插件擦写页设置非全擦时，无法修改 OPTION。则可以不需要设置 OPTION）



1、6、关于起始地址、中断地址、程序空间地址、页擦写地址、第二复位向量的地址等配置规则详见各 MCU Datasheet 第二复位向量配置章节。以 HC88T3541 为例，第二复位向量空间大小只能为 1K、2K、4K、8K、16K、32K。如果将第二复位向量空间大小定义为 4K，那么 APP 程序地址为 0x0000-0x7FFF，第二复位向量地址设置为 7000。Bootloader 程序地址为 0x7000-0x7FFF。那么 Bootloader 工程中的起始地址应该设置为 7000H，中断地址需设置为

0x7000，内存区间地址需设置为 0x7000-0x7FFF，页擦写选择第 28、29、30、31 页，第二复位向量地址设置为 0x7000。

## 6.3 注意事项

- 1、Bootloader 程序中关于 IAP 的操作并不严谨，具体的操作方式请严格参照《AN0003\_HA\_通用系列 IAP 使用》。
- 2、APP 程序和 Bootloader 程序烧录时请先烧录 APP 程序再烧录 Bootloader 程序。
- 3、在使用脱机工具 HC-PM51 烧录时可将 APP 和 Bootloader 程序同时烧录进入 MCU，具体操作方式请参照《UM0101\_HC-PM51 V5.0\_工具使用手册》。
- 4、使用第二复位向量 IAP 升级只能升级 APP 程序，无法升级代码选项。

## 7、版本说明

版本	日期	描述
V1.00	2024/11/14	初版

HOLYCHIP公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。HOLYCHIP不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，HOLYCHIP的产品不是专门设计来应用于外科植入、生命维持和任何HOLYCHIP产品产生的故障会对个体造成伤害甚至死亡的领域。如果将HOLYCHIP的产品用于上述领域，即使这些是由HOLYCHIP在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证HOLYCHIP及其雇员、子公司、分支机构和销售商与上述事宜无关。

芯圣电子  
2018年5月