# 1、说明

该文档仅适用CH554

IAP功能使用：

整个单片机中代码由4个部分组成：

|  |  |
| --- | --- |
| 地址 | 内容 |
| 跳转区 0x0000~0x0003 | 跳转IAP程序startup，大小固定空间，不可更改 |
| 应用区 0x0x0004~用户代码完整性标志之前 | 跑用户自己的应用代码的空间，需要用户将你自己功能代码的起始地址改到0x0004 |
| IAP起始地址前2个字节 | IAP代码会对这个位置进行修改，完成IAP代码下载会在对应位置置1 |
| 芯片flash大小减去IAP程序大小 | IAP升级代码存放的位置，后续都是利用这段代码烧录HEX，向前凑整 |

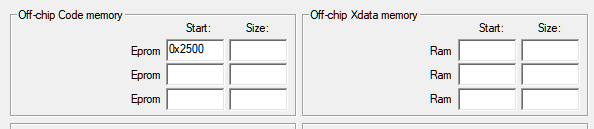
文件夹中的工程默认是上面的配置，

# 2、开发IAP程序

IAP程序禁止使用中断。

IAP程序禁止操作代码空间0x0000,0x0001,0x00020x0004 四个字节的数据，跳过。

文件夹下的工程即为IAP升级代码工程，IAP程序存放在代码空间的最后面。这个起始地址可以修改，需要满足条件是起始地址加上IAP代码大小不超过14K。先正常开发iap程序，假设最终编译结果程序大小3.5K字节，那可以预留4K给iap，那就将IAP工程的起始地址设置为10K处（在keil魔术棒里面配置）。

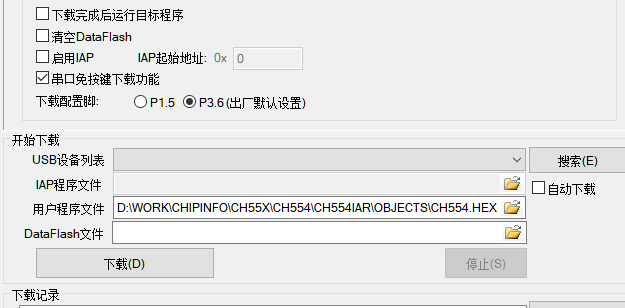


# 3、使用

全流程不使用WCHISPTool的IAP程序文件功能



使用前面编译好的IAP程序，直接烧录进芯片即可实现IAP功能，只需要利用ISP工具下载一次。



后续烧录用户代码需要用文件夹中的IAP升级工具。

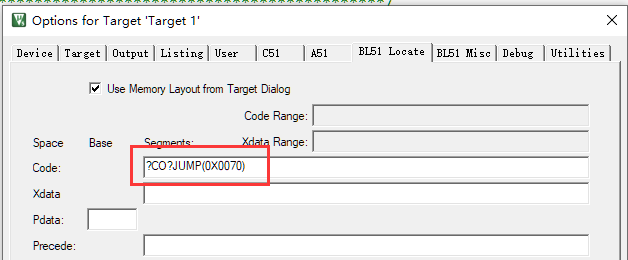
# 4、开发用户程序

1、在CH554APP工程模板上开发。

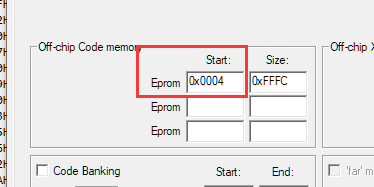
初期调试可以直接用WCHISPTool烧录固件，可以不借助IAP功能。

不要动JUMP.c的命名，不要在里面添加额外的东西

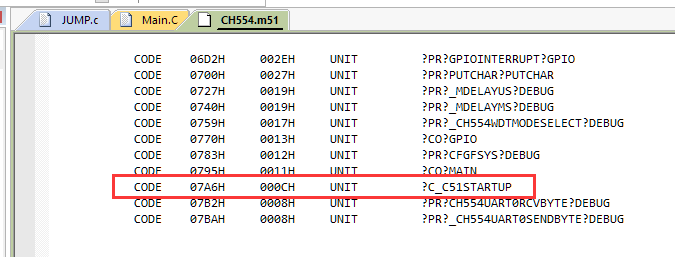
不要动工程中的这个配置



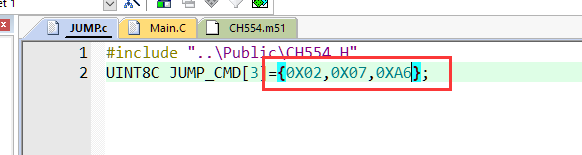
配置工程起始地址为0x0004



2、调试开发功能都完成后打开工程map文件，找一下startup的地址



把那个07A6，修改到JUMP.c的数组里面



这个07A6自己的代码工程是多少就是多少，如实填。

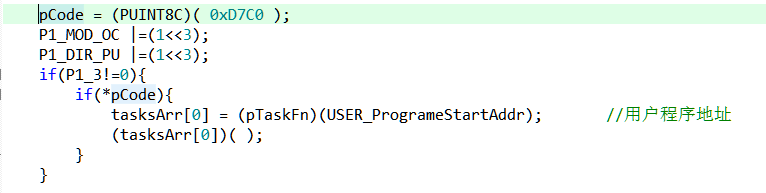
在编译一次即可。没有其余修改。

3、用户代码下载

IAP下载需要使用文件夹中对应的上位机软件，需要电脑安装CH372/375驱动



执行IAP升级代码的两种情况



1. P1.4拉低
2. \*pCode的用户代码完整标志位没有置1

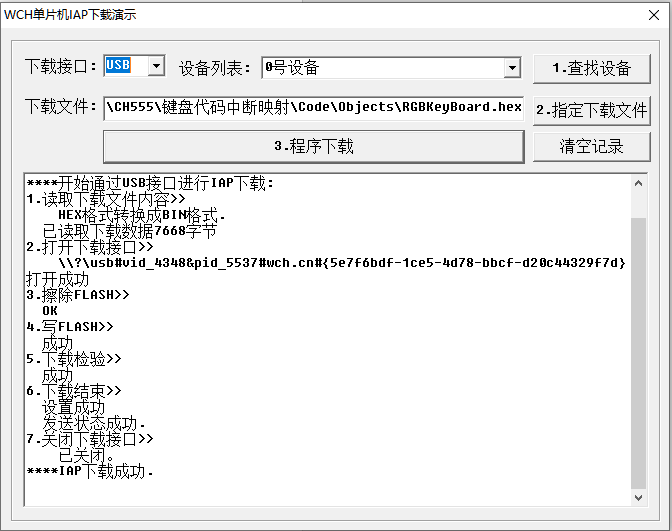
所以下载完IAP程序之后第一次上电肯定会运行IAP升级代码，后续升级代码需要将P1.4接地。

具体什么条件执行IAP升级，这完全取决软件开发人员的想法，不是绝对，这里做成这样没有任何理由，单纯随意，这里是一个内部条件+一个外部条件，一般就满足应用

运行IAP升级工具，如下图，显示已经查找到USB设备，下载文件选择你自己的用户程序，点击程序下载，等待下载完成就可以了。



下载成功显示如下



断电重启就能正常运行你的代码了，IAP代码中增加了将P1.0接地就复位的功能。

串口下载选择对应的串口号，单片机连接到UART0,默认波特率57600，和USB下载一样的操作



原理简单说明：

上电直接跳IAP，IAP程序判断用户代码完整性标志位

完整就：运行用户程序

不完整：执行IAP升级程序

单片机中的IAP代码会忽略USB

IAP代码在传输固件过程中会将用户程序的入口放到0x100地址去，升级成功将代码完整性标志的地方写1

IAP固定跳至0x100实现用户代码的启动