# 一、说明

IAP功能使用：

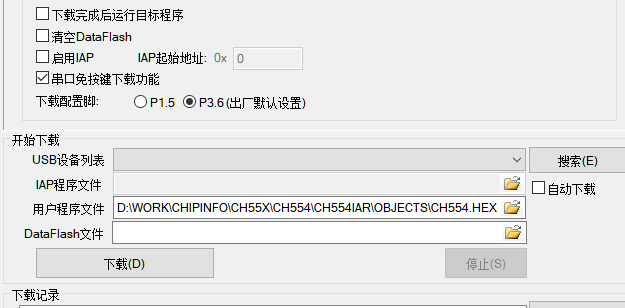
整个单片机中代码由4个部分组成：

|  |  |
| --- | --- |
| 地址 | 内容 |
| 禁止修改区 0x0000~0x0078 | IAP功能负责中断跳转和上电跳转的部分 |
| app区 0x100~ | 跑用户自己的应用代码的空间，需要用户将app代码的起始地址改到0x0104 |
| 用户代码完整性标志0x6000-64的一个page | 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 |
| IAP区 0X6000~ | IAP升级代码存放的位置，后续都是利用这段代码烧录HEX |

文件夹中的工程默认是上面的配置，

# 二、使用

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。

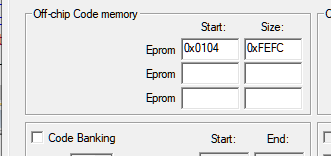


后续烧录用户代码需要用文件夹中的IAP升级工具。

# App工程的修改

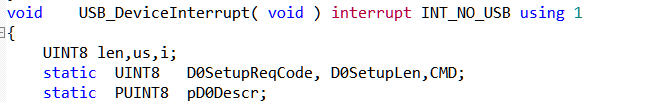
使用这个bin文件的的限制条件：

1、用户代码起始位置需要修改成0x104

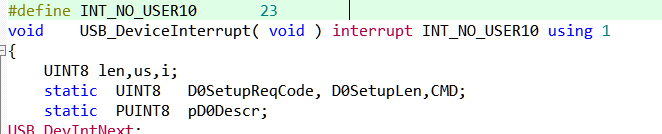


2、用户代码中断的实现

我们要实现USB中断，正常代码是下面的写法



因为IAP的固件做了二次的跳转，所以要根据下面的表，重新写用户代码中中断函数的中断号，只需要将原本的USB中断号7替换成23即可，中断函数内部保持不变。



利用二次跳转实现了所有中断。

中断号偏移：

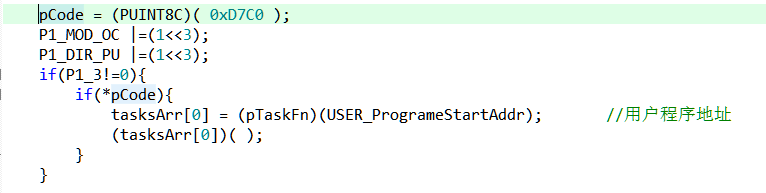
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中断名称 | 中断号 | 入口地址 | 对应中断号 | 入口地址 |  |
| INT\_NO\_INT0 | 0 | 0X0003 | 16 | 0X0083 |  |
| INT\_NO\_TMR0 | 1 | 0X000B | 17 | 0X008B |  |
| INT\_NO\_INT1 | 2 | 0X0013 | 18 | 0X0093 |  |
| INT\_NO\_TMR1 | 3 | 0X001B | 19 | 0X009B |  |
| INT\_NO\_UART0 | 4 | 0X0023 | 20 | 0X00A3 |  |
| INT\_NO\_TMR2 | 5 | 0X002B | 21 | 0X00AB |  |
| INT\_NO\_SPI0 | 6 | 0X0033 | 22 | 0X00B3 |  |
| INT\_NO\_TKEY | 7 | 0X003B | 23 | 0X00BB |  |
| INT\_NO\_USB | 8 | 0X0043 | 24 | 0X00C3 |  |
| INT\_NO\_ADC | 9 | 0X004B | 25 | 0X00CB |  |
| INT\_NO\_UART1 | 10 | 0X0053 | 26 | 0X00D3 |  |
| INT\_NO\_PWMX | 11 | 0X005B | 27 | 0X00DB |  |
| INT\_NO\_GPIO | 12 | 0X0063 | 28 | 0X00E3 |  |
| INT\_NO\_WDOG | 13 | 0X006B | 29 | 0X00EB |  |

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. P1.3拉低
2. \*pCode的用户代码完整标志位没有置1

所以下载完这个文件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。

这部分逻辑按需自行设计

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

