1.0.8及以上版本较1.0.5版本有以下变化：

1. UI界面更新；
2. 针对M9F951及M9F680添加了ROM编译范围设置；
3. M9F951的配置字变更；

您可以选择继续使用1.0.5版本，但后续将不会对该版本进行维护。

## IDE基本配置：

1. 旧工程先重新选择一次芯片(菜单栏-配置-选择芯片)，否则可能造成配置字不一致问题。
2. 检查新的配置字选项，此次配置字界面与之前版本不同如下图红框：



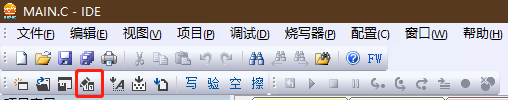
**复位向量**即上电程序运行位置，与Boot程序密切相关，Boot置于Flash最前面则选择0X0000，置于Flash最后面则选择0X7FF8；

**写保护范围**进行了变更，主要针对Boot程序位置进行了调整。

**1.0.8及以上版本强烈建议将Boot写在Flash最前方，并对0X0000-0X0FFF进行写保护，这段区域被保护将同时保护配置字信息，可防止配置字被误擦写。**

1. 配置当前工程ROM编译范围：

点击图中红框处：



弹出如下窗口：



先填写Rom区域，如图中1000-7FFF则程序编译在0X1000-0X7FFF范围内，然后填写偏移量，此处填写Rom区域的起始地址，图中即1000，**注意如果ROM区域以0起始时，偏移量留空不要填写，否则会产生一个警告。**

1. 对于boot置于Flash前面的用户程序，此时用户程序的编译起始地址不为0，而在仿真或者用户程序验证阶段，可能并没有将boot放进Flash前面，此时需要添加以下信息，才可保证正常运转，而添加了boot程序后以下内容可删除：

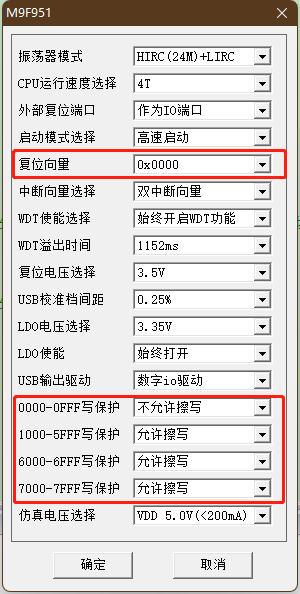


您也可以直接在usbhid\_lib或升级示例的userupdate中直接获取本文件添加到工程，文件名:HWCONFIG.H

## Boot前置写法（推荐）

Flash的0000-0FFF区域进行写保护操作时，会同时对芯片配置字进行写保护，此时即便出现误擦写也不会修改到配置字，所以Boot放于这个区域并进行写保护，既能保护Boot程序也可以保护配置字。

1. 修改配置字，如下图：



注意红框处即可，其余配置字根据需求设定。

1. 配置boot工程ROM编译范围为0000-0FFF，如图：



1. boot程序中添加对中断入口重映射代码，如下图：



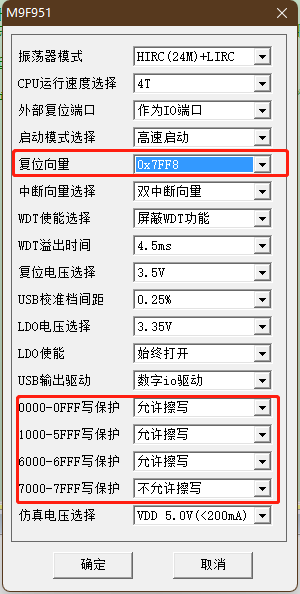
相较IDE基本配置条目里，此处删除了跳转用户程序的数组，实际如何跳转由boot代码决定。

1. 编写代码。可参考开发包-USB升级示例-bootloard例程。

## Boot后置写法（不推荐）

**不建议将boot后置，后置无法对芯片配置字进行保护；此处仅针对于兼容以前旧产品说明，步骤基本与1.0.5版本编写方式相同：**

1、修改配置字，如下图：



注意红框处即可，其余配置字根据需求设定。

2、配置ROM编译范围：



1. 程序中添加常量数组：

