

# 智嵌 以太网实现远程更新固件例程使用手册

版本号：A

拟制人：赵工

时 间：2015 年 2 月 3 日



## 目 录

1	本文档编写目的 .....	3
2	基础知识(该部分摘自互联网).....	3
2.1	IAP .....	3
2.2	APP .....	6
3	IAP的实现 .....	6
3.1	实验步骤 .....	6
3.2	IAP和APP程序的注意点 .....	8
4	BIN文件的生成.....	9

## 1 本文档编写目的

本使用手册是针对“以太网实现远程更新固件例程”而编写的。

## 2 基础知识(该部分摘自互联网)

### 2.1 IAP

IAP, 全称是“In-Application-Programming”, 中文解释为“在程序中编程”。IAP 是一种通过微控制器的对外接口（如 USART, IIC, CAN, USB, 以太网接口甚至是无线射频通道），对正在运行程序的微控制器进行内部程序的更新的技术（注意这完全有别于 ICP 或者 ISP 技术）。ICP (In-Circuit Programming) 技术即通过在线仿真器对单片机进行程序烧写，而 ISP 技术则是通过单片机内置的 bootloader 程序引导的烧写技术。无论是 ICP 技术还是 ISP 技术，都需要有机械性的操作如连接下载线，设置跳线帽等。若产品的电路板已经层层密封在外壳中，要对其进行程序更新无疑困难重重，若产品安装于狭窄空间等难以触及的地方，更是一场灾难。但若引入了 IAP 技术，则完全可以避免上述尴尬情况，而且若使用远距离或无线的数据传输方案，甚至可以实现远程编程和无线编程。这绝对是 ICP 或 ISP 技术无法做到的。某种微控制器支持 IAP 技术的首要前提是其必须是基于可重复编程闪存的微控制器。STM32 微控制器带有可编程的内置闪存，同时 STM32 拥有在数量上和种类上都非常丰富的外设通信接口，因此在 STM32 上实现 IAP 技术是完全可行的。

实现 IAP 技术的核心是一段预先烧写在单片机内部的 IAP 程序。这段程序主要负责与外部的上位机软件进行握手同步，然后通过外设通信接口将来自于上位机软件的（APP）程序数据接收后写入单片机内部指定的闪存区域，然后再跳转执行新写入的程序，即 APP 程序，最终就达到了程序更新的目的。

在 STM32 微控制器上实现 IAP 程序之前首先要回顾一下 STM32 的内部闪存组织架构和其启动过程。STM32 的内部闪存地址起始于 0x8000000，一般情况下，程序文件从此地址开始写入。此外 STM32 是基于 Cortex-M3 内核的微控制器，其内部通过一张“中断向量表”来响应中断，程序启动后，将首先从“中断向量表”取出复位中断向量执行复位中断程序完成启动。而这张“中断向量表”的起始地址是 0x8000004，当中断来临，STM32 的内部硬件机制亦会自动将 PC 指针定位到“中断向量表”处，并根据中断源取出对应的中断向量执行中断服务程序。最后还需要知道关键的一点，通过修改 STM32 工程的链接脚本可以修改程序文件写入闪存的起始地址。

在 STM32 微控制器上实现 IAP 方案，除了常规的串口接收数据以及闪存数据写入等常规操作外，还需注意 STM32 的启动过程和中断响应方式。图 1 显示了 STM32 常规的运行流程。

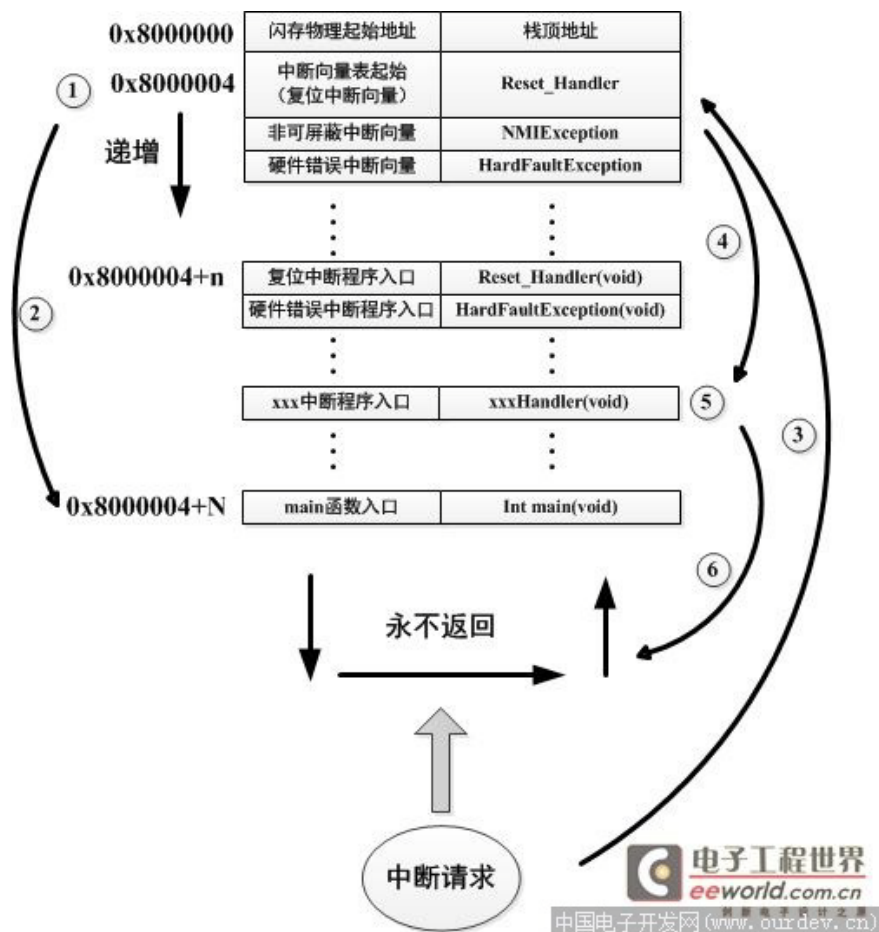


图 1 STM32 常规程序运行流程

对图 1 解读如下：

- 1、STM32 复位后，会从地址为 0x8000004 处取出复位中断向量的地址，并跳转执行复位中断服务程序，如图 1 中标①所示。
  - 2、复位中断服务程序执行的最终结果是跳转至 C 程序的 main 函数，如图 1 中标号②所示，而 main 函数应该是一个死循环，是一个永不返回的函数。
  - 3、在 main 函数执行的过程中，发生了一个中断请求，此时 STM32 的硬件机制会将 PC 指针强制指回中断向量表处，如图 1 中标号③所示。
  - 4、根据中断源进入相应的中断服务程序，如图 1 中标号⑤所示。
  - 5、中断服务程序执行完毕后，程序再度返回至 main 函数中执行，如图 1 中标号⑥所示。
- 若在 STM32 中加入了 IAP 程序，则情况会如图 2 所示：

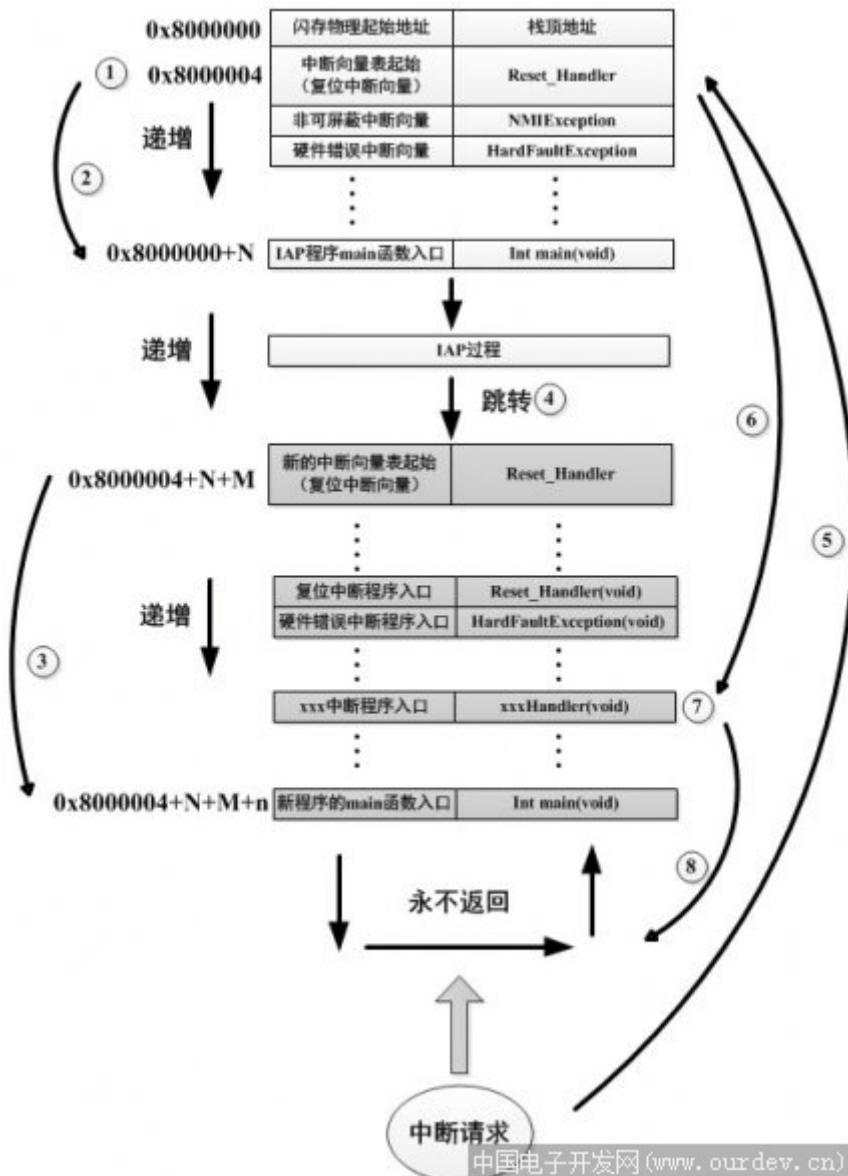


图 2 加入 IAP 后的程序运行流程图

对图 2 的解读如下:

- 1、STM32 复位后，从地址为  $0x8000004$  处取出复位中断向量的地址，并跳转执行复位中断服务程序，随后跳转至 IAP 程序的 main 函数，如图 2 中标号①、②所示。这个过程和图 1 相应部分是一致的。
- 2、执行完 IAP 过程后（STM32 内部多出了新写入的程序，图 2 中以灰色底纹方格表示，地址始于  $0x8000004+N+M$ ）跳转至新写入程序的复位向量表，取出 APP 程序的复位中断向量的地址，并跳转执行 APP 程序的复位中断服务程序，随后跳转至新程序的 main 函数，其过程如图 2 的标号③所示。APP 程序的 main 函数应该也具有永不返回的特性。同时应该注意在 STM32 的内部存储空间在不同的位置上出现了 2 个中断向量表。
- 3、在新程序 main 函数执行的过程中，一个中断请求来临，PC 指针仍会回转至地址为  $0x8000004$  中断向量表处，而并不是 APP 程序的中断向量表，如图 2 中标号⑤所示。注意到这是由 STM32 的硬件机制决定的。
- 4、根据中断源跳转至对应的中断服务，如图 2 中标号⑥所示。注意此时是跳转至了 APP 程序的中断服务程序中。



5、中断服务执行完毕后，返回 `main` 函数。如图 2 中标号⑧所示。

从上述两个过程的分析可以得知，对将使用 IAP 过程写入的程序要满足 2 个要求：

- 1、APP 程序必须从 IAP 程序之后的某个偏移量为 `x` 的地址开始；
- 2、必须将 APP 程序的中断向量表相应的移动，移动的偏移量为 `x`；

将中断向量表移动的方法是在程序中加入函数：

```
void NVIC_SetVectorTable(u32 NVIC_VectTab, u32 Offset);
```

其中参数 `NVIC_VectTab` 为中断向量表起始位置，而参数 `Offset` 则为地址偏移量，如将中断向量表移至 `0x8002000` 处，则应调用该函数如下：

```
void NVIC_SetVectorTable(0x8000000, 0x2000);
```

同时有必要提醒读者注意的是，此函数只会修改 STM32 程序中用于存储中断向量的结构体变量，而不会实质地改变中断向量表在闪存中的物理位置，详情请研究该程序原型。

有了以上准备后就可以着手设计一个 IAP 方案了，如下：

- 1、STM32 复位后，利用一个按键（也可以不用）的状态进行同步，当按键按下时表示将要进行 IAP 过程；
- 2、在 IAP 过程中，通过上位机软件向 STM32 的外设通讯接口（比如 USART1、网口等）发送所要更新的 APP 程序文件，STM32 接收到数据后转而从 `0x8002000` 地址开始写入收到的数据；
- 3、再次复位后，跳转 `0x8002004` 地址开始运行 APP 程序；

注意事项：

- （1）利用 IAP 写入的 APP 程序最好是 .bin 格式的文件，但不能是 .hex 格式的文件；
- （2）向 STM32 发送 APP 程序文件时尽量慢一些，因为 STM32 对 FLASH 的写入速度往往跟不上通讯外设接口的速度；
- （3）建议在 STM32 和上位机之间设计一套握手机制和出错管理机制，这样可以大幅提高 IAP 的成功率；

## 2.2 APP

2.1 节所述的 IAP 程序必须通过其它手段，如 JTAG 或 ISP 烧入，我们可以把 IAP 程序称为 Bootloader 程序。APP 程序即是我们通过 IAP 程序下载的应用程序。IAP 和 APP 程序分别存放在 STM32 FLASH 的不同地址范围，一般从最低地址区开始存放 IAP，紧跟其后的就是 APP 程序（注意，如果 FLASH 容量足够，是可以设计很多 APP 程序的）。

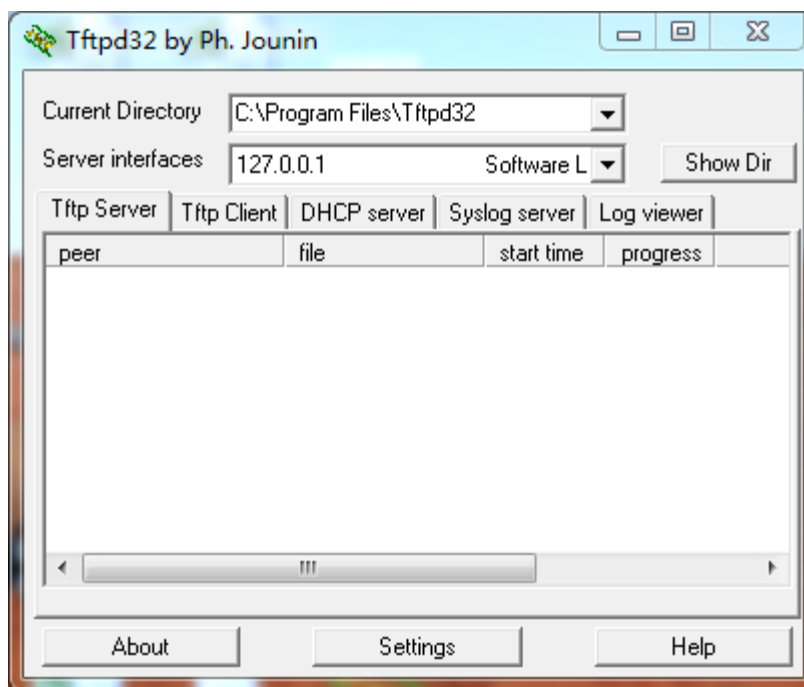
## 3 IAP 的实现

IAP 程序内建 TFTP 服务器，IP: 192.168.1.253，端口号：69。PC 机通过 TFTP 软件，如“Tftpd32”，该软件需要先安装。向该服务器传输文件，写入 STM32 的片内 FLASH 中。

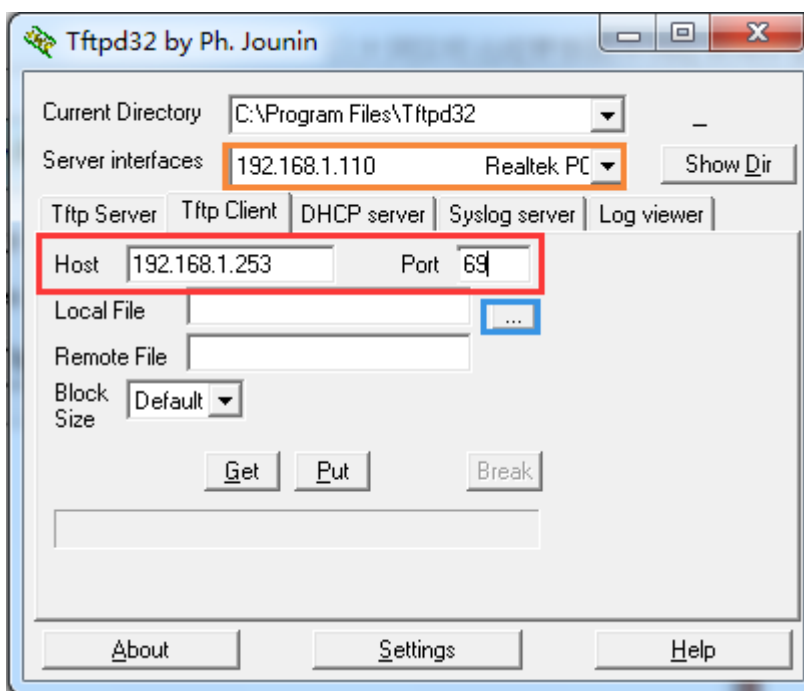
### 3.1 实验步骤

（1）用 JLINK 或串口线将“以太网远程升级例程\_IAP”下载到开发板，按下 S1~S4 任一按键（具体为什么可以参考程序），再按复位键“RESET”，此时开发板开始运行 IAP 程序，等待 PC 机 TFTP 下发文件。

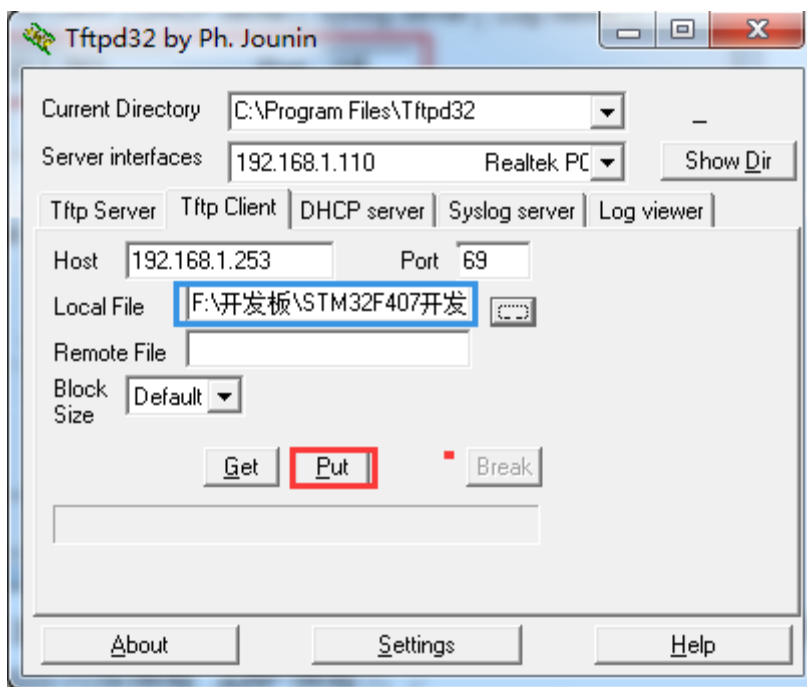
（2）打开“Tftpd32”，默认如下设置：



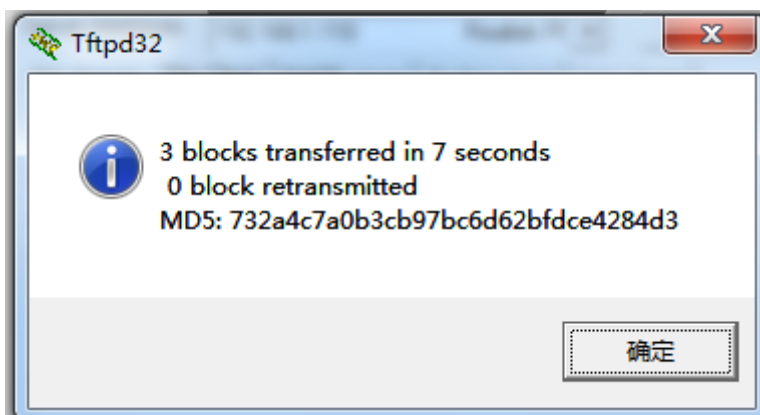
点击上图中的“Tftp Client”,并设置如下:



上图中,橘红色框为电脑 IP,红色框为开发板 IP 和端口号,点击淡蓝色按钮,选择要下载的 bin 文件(比如下载” LED 闪烁例程\_APP”例程:基于 TFTP 协议的以太网远程升级例程(裸机)\LED 闪烁例程\_APP\Project\BIN):



选择好文件后，点击“Put”按钮，稍等片刻，则会下载成功：



点击“确定”即可。

按下“RESET”，则 APP 程序开始运行。如果要再更新固件时，只需要按下 S1~S4 任一按键，再复位即可进入到 IAP 程序，此时又可以用“Tftpd32”更新固件了。

### 3.2 IAP 和 APP 程序的注意点

- 1、在 IAP 程序中，跳转到 APP 程序前要关闭总中断以及复位中断向量表和时钟，如：

```
CLI();           //关闭总中断
NVIC_DeInit();
RCC_DeInit();
```

- 2、在 IAP 程序中中断向量表的偏移值为 0x0，如：



```
10 void NVIC_Configuration(void)
11 {
12
13     NVIC_InitTypeDef    NVIC_InitStructure;
14
15     NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
16
17     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4);
18     NVIC_InitStructure.NVIC_IRQChannel = ETH_IRQn;
19     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
20     NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
21     NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
22     NVIC_Init(&NVIC_InitStructure);
23 }
```

3、在 APP 程序中，要打开全局中断（如果用到中断的话）。

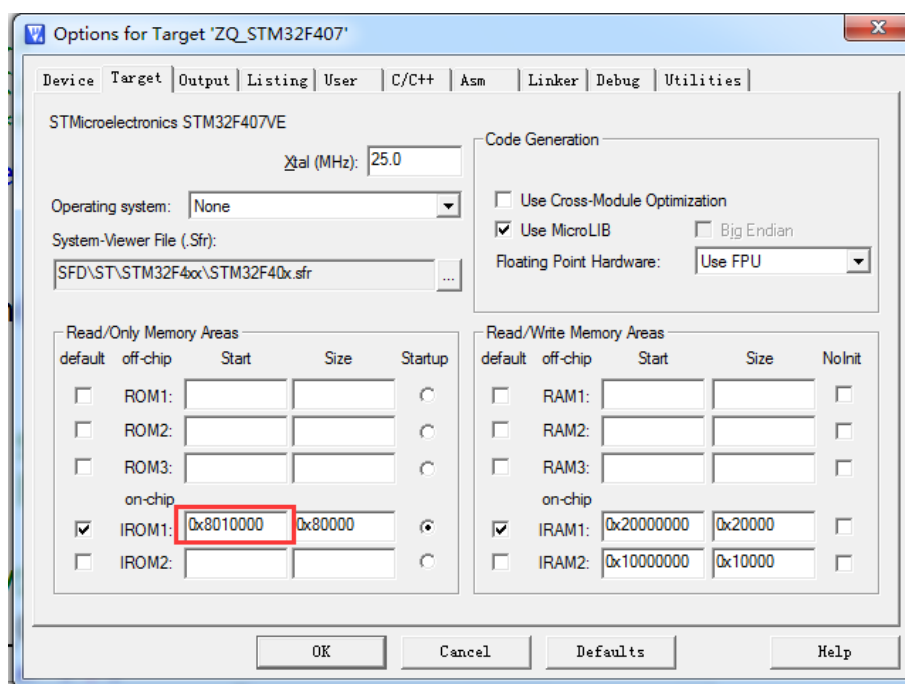
4、在 APP 程序中，中断向量表的偏移值为 0x10000（这个值要和 APP 程序的起始地址对应，本示例中的 APP 起始地址都是 0x08010000），如：

```
231 void ETH_NVIC_Config(void)
232 {
233     NVIC_InitTypeDef    NVIC_InitStructure;
234     NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x10000);
235     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4);
236
237     /* Enable the Ethernet global Interrupt */
238     NVIC_InitStructure.NVIC_IRQChannel = ETH_IRQn;
239     NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
240     NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
241     NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
242     NVIC_Init(&NVIC_InitStructure);
243 }
244
```

## 4 BIN 文件的生成

### 1、设置 APP 程序起始地址

打开示例工程(如 LED 闪烁例程\_APP), 在工程的“Option for Target...”界面中的“Target”页里将“IROM”的“Start”列改为 APP 程序起始的地址，如下图中将程序起始位置设为 0x8010000（也可以是其他地址）。

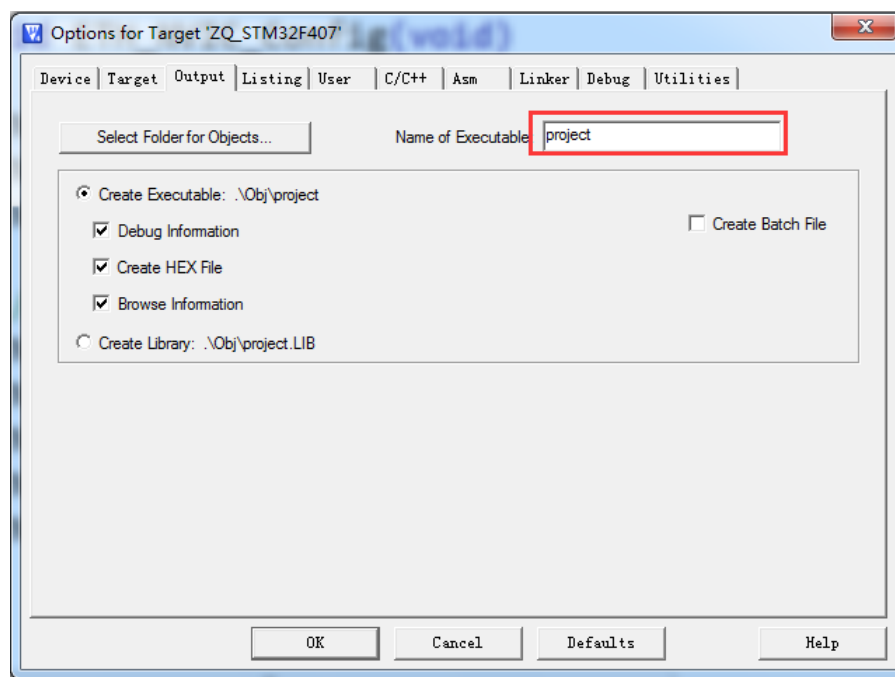


## 2、利用 KEIL4 生成 BIN

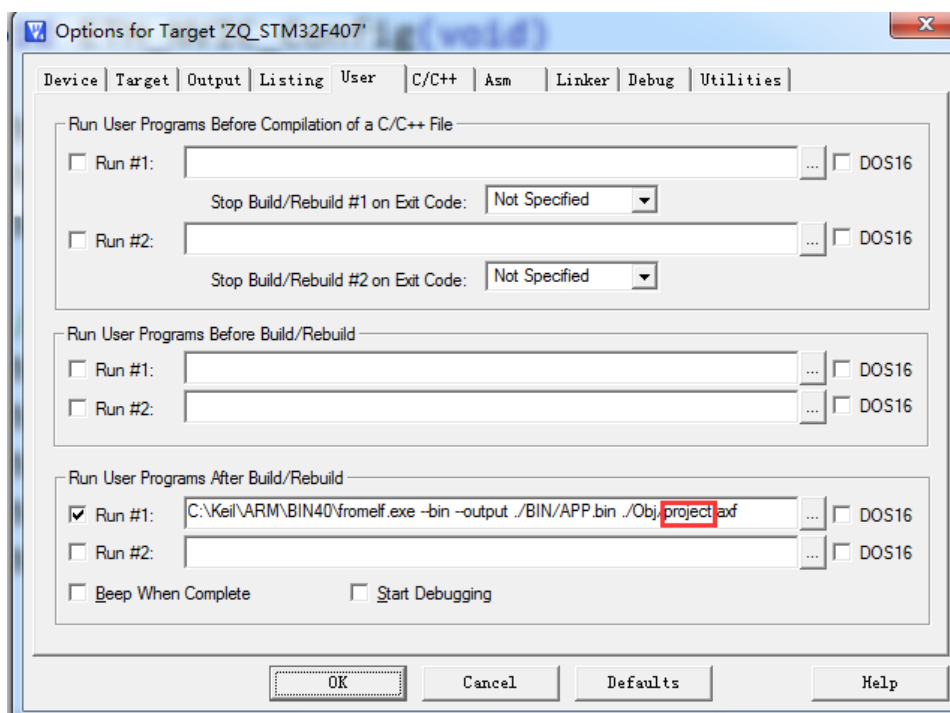
(1) 在“LED 闪烁例程\_APP\Project”路径下新建文件夹“BIN”:

BIN	2015/2/3 13:51	文件夹	
List	2015/2/3 9:51	文件夹	
Obj	2015/2/3 13:51	文件夹	
JLink Regs CM3	2014/12/2 17:23	文本文档	1 KB
JLinkLog	2015/2/3 12:23	文本文档	38 KB
JLinkSettings	2014/10/22 20:47	Configuration Se...	1 KB
project.uvgui.Administrator	2015/2/3 13:55	ADMINISTRATO...	74 KB
project.uvgui_Administrator.bak	2015/2/3 12:23	BAK 文件	74 KB
project.uvopt	2015/2/3 13:55	UVOPT 文件	17 KB
project	2015/2/3 13:55	vision4 Project	34 KB
project_Flash.dep	2014/10/22 20:44	DEP 文件	30 KB
project_uvopt.bak	2015/2/3 12:23	BAK 文件	17 KB
project_uvproj.bak	2015/2/3 12:23	BAK 文件	34 KB
project_ZQ_LED.dep	2014/10/23 0:06	DEP 文件	30 KB
project_ZQ_STM32F407.dep	2015/2/5 15:30	DEP 文件	30 KB

(2) 在“Output”选项中，设置如下:



(3) 在“User”选项中，设置如下：



将以下红色内容复制到“Run #1”中：

**C:\Keil\ARM\BIN40\fromelf.exe --bin --output ./BIN/APP.bin ./Obj/project.axf**

点击“OK”退出设置，再对工程编译，然后就可以看到新建的“BIN”文件夹中有“APP.BIN”生成。

-----以下无正文。