

# 零死角玩转STM32

与野火同行 乐意惬意无边



原创教程，完全开源。



由浅入深，结合实操。



通俗易懂，详尽解读。



配套板子，全面玩转。



强强联合，不断更新。



野火团队 Wild Fire Team



## 0、友情提示

《零死角玩转 STM32》系列教程由初级篇、中级篇、高级篇、系统篇、四个部分组成，根据野火 STM32 开发板旧版教程升级而来，且经过重新深入编写，重新排版，更适合初学者，步步为营，从入门到精通，从裸奔到系统，让您零死角玩转 STM32。M3 的世界，与野火同行，乐意惬意无边。

另外，野火团队历时一年精心打造的《STM32 库开发实战指南》将于今年 10 月份由机械工业出版社出版，该书的排版更适于纸质书本阅读以及更有利于查阅资料。内容上会给你带来更多的惊喜。是一本学习 STM32 必备的工具书。敬请期待！



## 3、如何新建工程模板

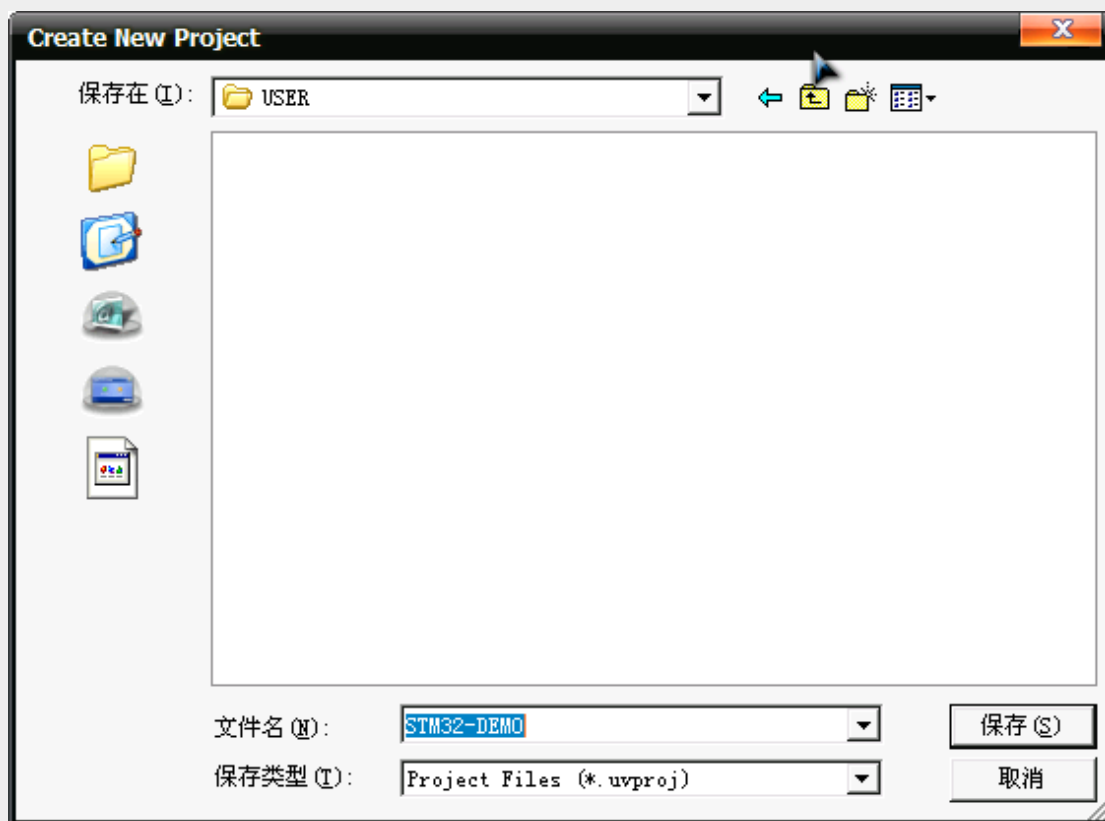
### 3.1 获取 ST 库源码

在新建工程模板之前，我们首先需要获取到 st 库的源码，源码可从 st 的官方网站下载到，也可在野火 M3 光盘目录下：**\2-程序+教程\第一部分-库开发初级篇** 找到，里面有 v3.0.0 和 v3.5.0 版本的库，这两个库的版本区别很小，几乎可以兼容。在这里我们以 v3.5.0 来新建我们的工程模板。

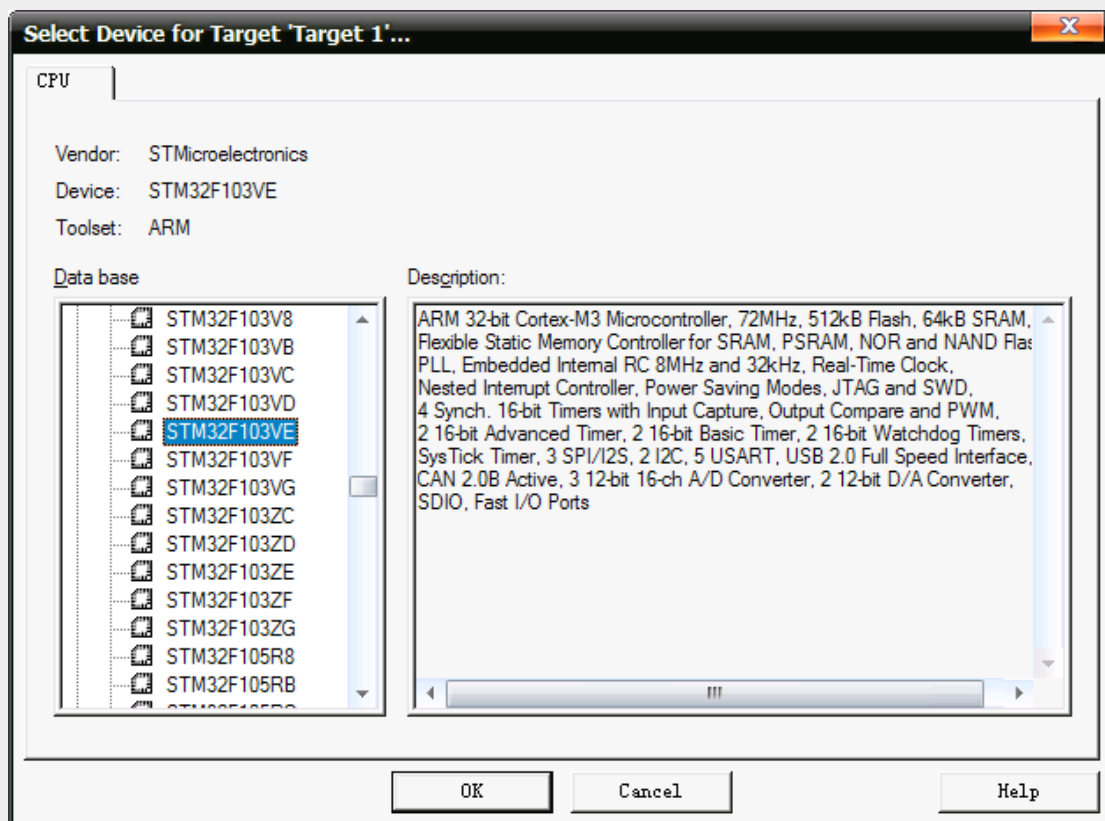
### 3.2 开始新建工程

- 点击桌面 UVision4 图标，启动软件。如果是第一次使用的话会打开一个自带的工程文件，我们可以通过工具栏 Project->Close Project 选项把它关掉。
- 在工具栏 Project->New  $\mu$ Vision Project...新建我们的工程文件，我们将新建的工程文件保存在桌面的 STM32-Template\USER 文件夹下（先在电脑桌面上新建一个 STM32-Template 文件夹，在 STM32-Template 里面新建一个 USER 文件夹），文件名取为 STM32-DEMO（英文 DEMO 的意思是例子），名字可以随便取，点击保存。





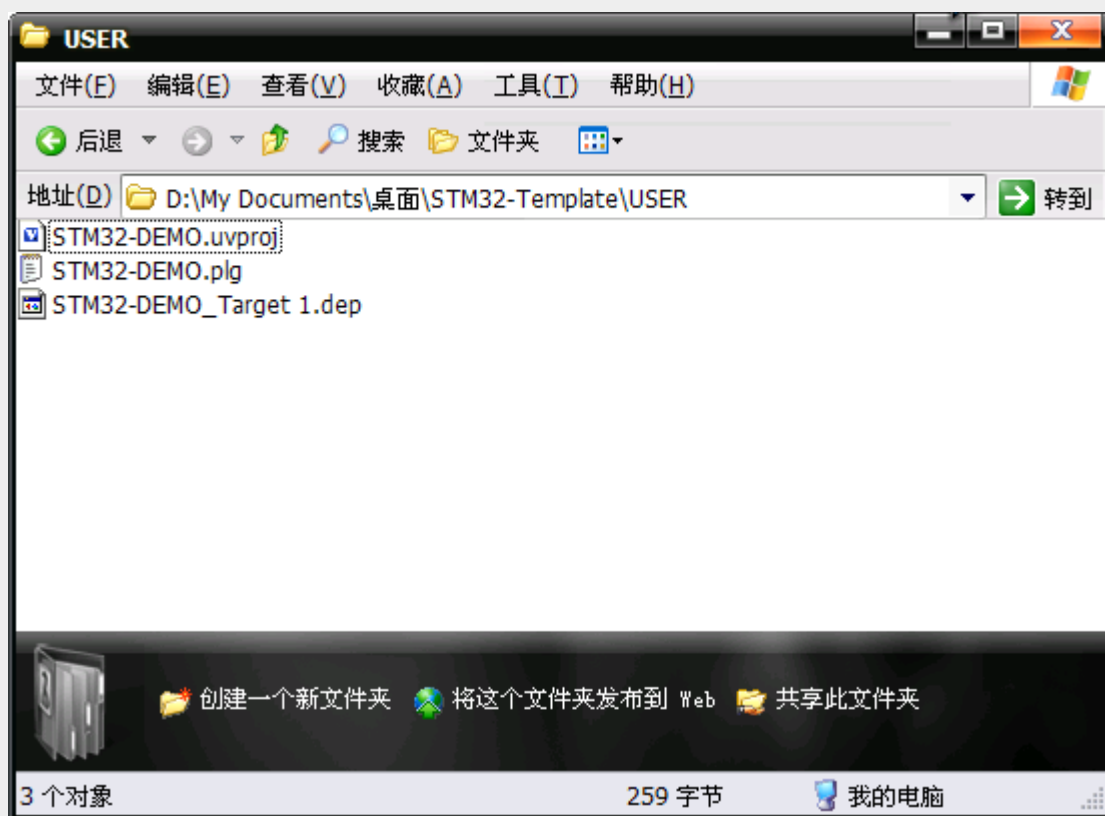
- 接下来的窗口是让我们选择公司跟芯片的型号，我们用的芯片是 ST 公司的 STM32F103VET6，有 64K SRAM, 512K Flash，属于高集成度的芯片。按如下选择即可。



- 接下来的窗口问我们是否需要拷贝 STM32 的启动代码到工程文件中，这份启动代码在 M3 系列中都是适用的，一般情况下我们都点击是，但我们这里用的是 ST 的库，库文件里面也自带了这一份启动代码，所以为了保持库的完整性，我们就不需要开发环境为我们自带的启动代码了，稍后我们自己手动添加，这里我们点击否。

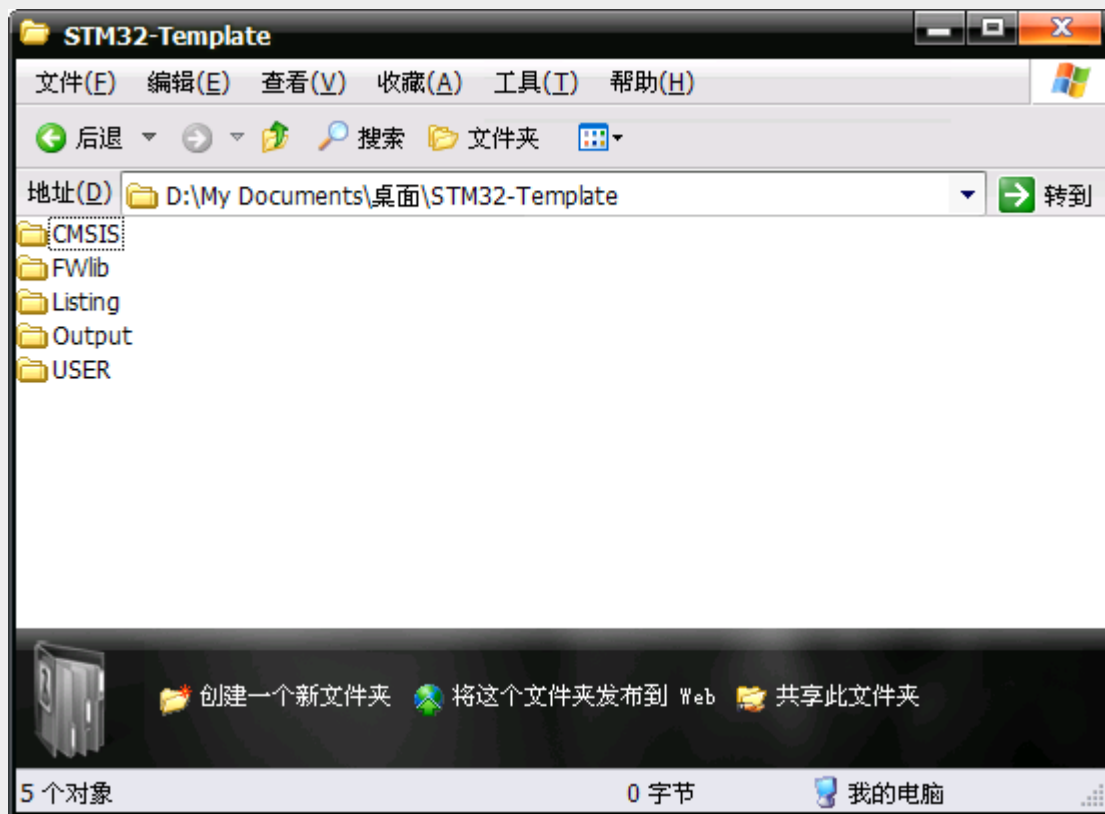


- 此时我们的工程新建成功，如下图所示。但我们的工程中还没有任何文件，接下来我们需要在我们的工程中添加所需文件。

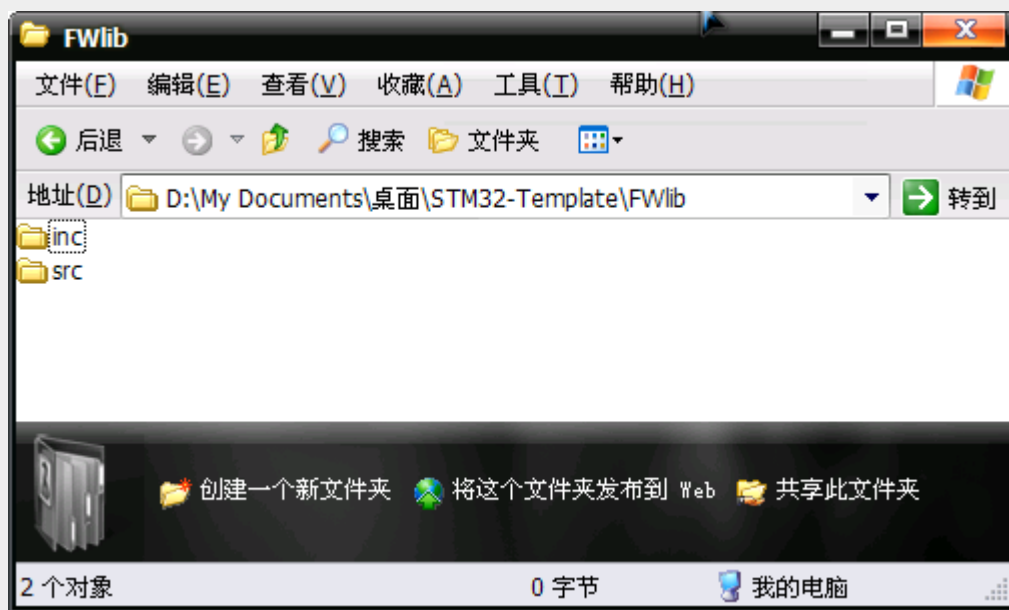


- 在 STM32-Template 文件夹下，我们新建四个文件夹，分别为 FWlib、CMSIS、Uotput、Listing。原先新建的 USER 用来存放工程文件和用户代码，包括主函数 main.c。FWlib 用来存放 STM32 库里面的 inc 和 src 这两个文件，这两个文件包含了芯片上的所有驱动。CMSIS 用来存放库为我们自带的启动文件和一些 M3

系列通用的文件。CMSIS 里面存放的文件适合任何 M3 内核的单片机。CMSIS 的缩写为：Cortex Microcontroller Software Interface Standard，是 ARM Cortex 微控制器软件接口标准，是 ARM 公司为芯片厂商提供的一套通用的且独立于芯片厂商的处理器软件接口。Output 用来保存软件编译后输出的文件，Listing 用来存放一些编译过程中产生的文件，具体可不用了解。



- 把野火 M3 光盘目录下：\3-ST 库 3.5.0 源码  
\\3.5.0\\3.5.0\\STM32F10x\_StdPeriph\_Lib\_V3.5.0\\Libraries\\STM32F10x\_StdPeriph\_Driver 的 inc 跟 src 这两个文件夹拷贝到 STM32-Template\\FWlib 文件夹中。

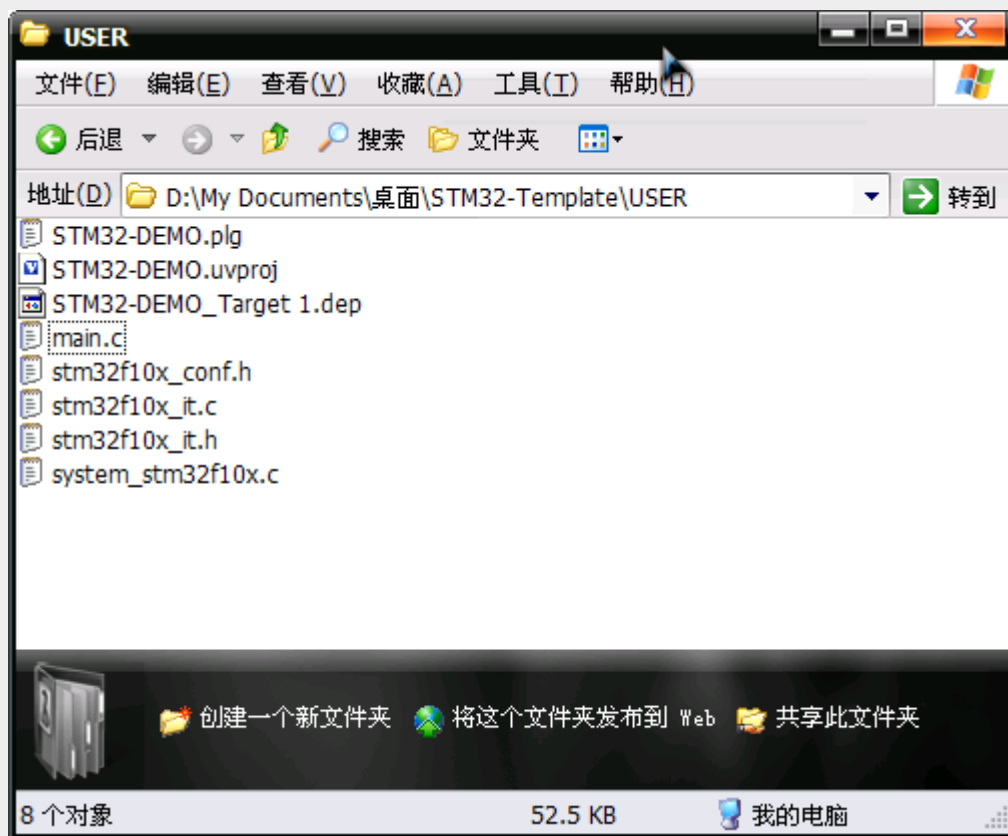


- 把野火 M3 光盘目录下：\2-程序+教程\第一部分-库开发初级篇\3-ST 库 3.5.0 源码

\3.5.0\3.5.0\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Template 下的 main.c、stm32f10x\_conf.h、stm32f10x\_it.h、stm32f10x\_it.c、system\_stm32f10x.c 拷贝到 STM32-Template\USER 目录下。stm32f10x\_it.h、和 stm32f10x\_it.c 这两个文件里面是中断函数，里面为空，并没有写任何的中断服务程序。stm32f10x\_conf.h 是用户需要配置的头文件，当我们需要用到芯片中的某部分外设的驱动时，我们只需要在该文件下将该驱动的头文件包含进来即可，片上外设的驱动在 src 文件夹中，inc 文件夹里面是它们的头文件。这三个文件是用户在编程时需要修改的文件，其他库文件一般不需要修改。system\_stm32f10x.c 是 ARM 公司提供的符合 CMSIS 标准的库文件，等下我们把这个文件移动到 STM32-Template\CMSIS 这个文件夹中。







- (1) 把野火 M3 光盘目录下: \2-程序+教程\第一部分-库开发初级篇\3-ST 库 3.5.0 源码

\3.5.0\3.5.0\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm 的全部文件拷贝到 STM32-

Template\CMSIS\startup (需先在 CMSIS 新建好 startup 文件夹) 文件夹下。

这些是用汇编写的启动文件。野火 M3 开发板用的 CPU 是 STM32F103VET6, 有 512K Flash, 属于大容量的, 所以等下我们把 startup\_stm32f10x\_hd.s 添加到我们的工程中。根据 ST 的官方资料: Flash 在 16 ~32 Kbytes 为小容量, 64 ~128 Kbytes 为中容量, 256 ~512 Kbytes 为大容量, 不同大小的 Flash 对应的启动文件不一样, 这点要注意。(2) 把野火 M3 光盘目录下: \2-程序+教程\第一部分-库开发初级篇\3-ST 库 3.5.0 源码

\3.5.0\3.5.0\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\CoreSupport 的 core\_cm3.c 和 core\_cm3.h 也拷贝到 STM32-Template\CMSIS 文件夹下。

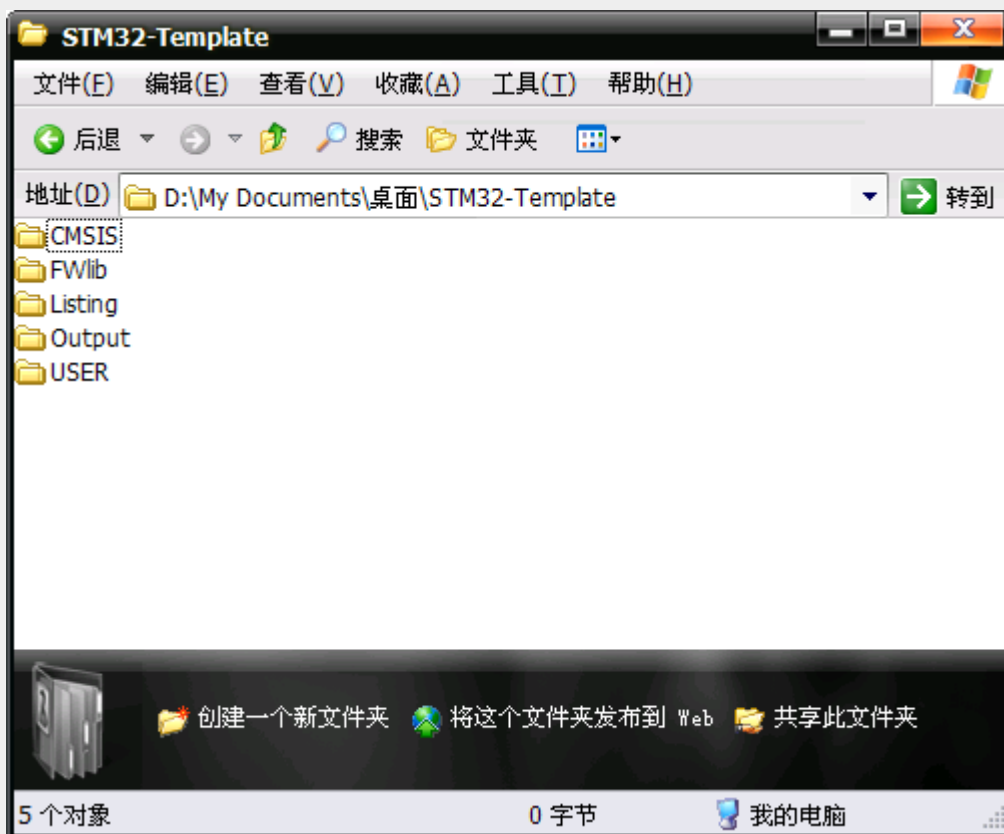
- (3) 把野火 M3 光盘目录下: \2-程序+教程\第一部分-库开发初级篇\3-ST 库 3.5.0 源码



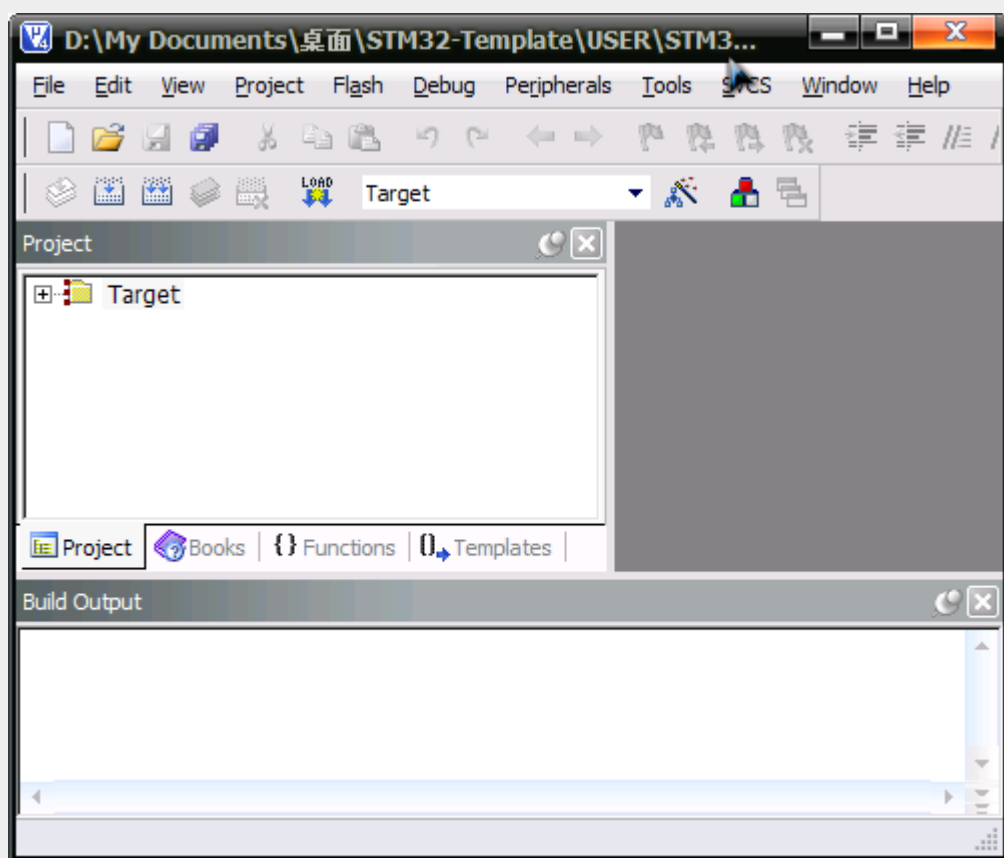
\\3.5.0\\3.5.0\\STM32F10x\_StdPeriph\_Lib\_V3.5.0\\Libraries\\CMSIS\\CM3\\DeviceSupport\\ST\\STM32F10x 的 stm32f10x.h、system\_stm32f10x.c、system\_stm32f10x.h 拷贝到 STM32-Template\\CMSIS 文件夹下。

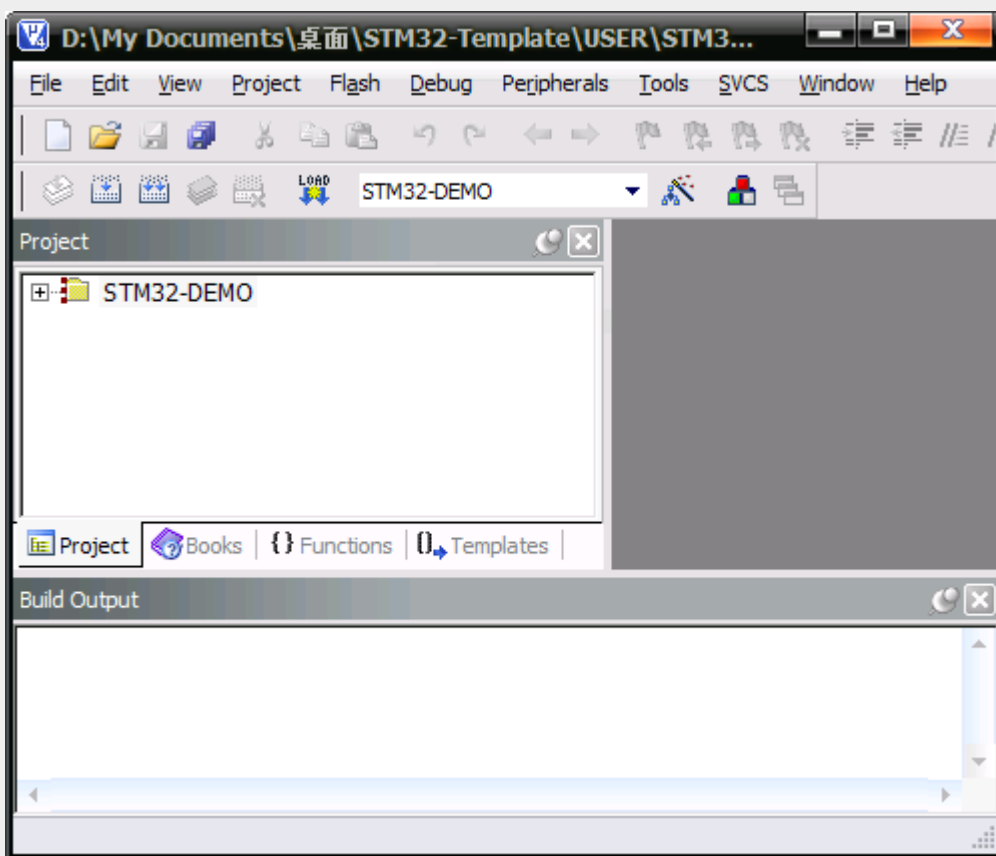


- 此时我们新进的工程目录如下所示



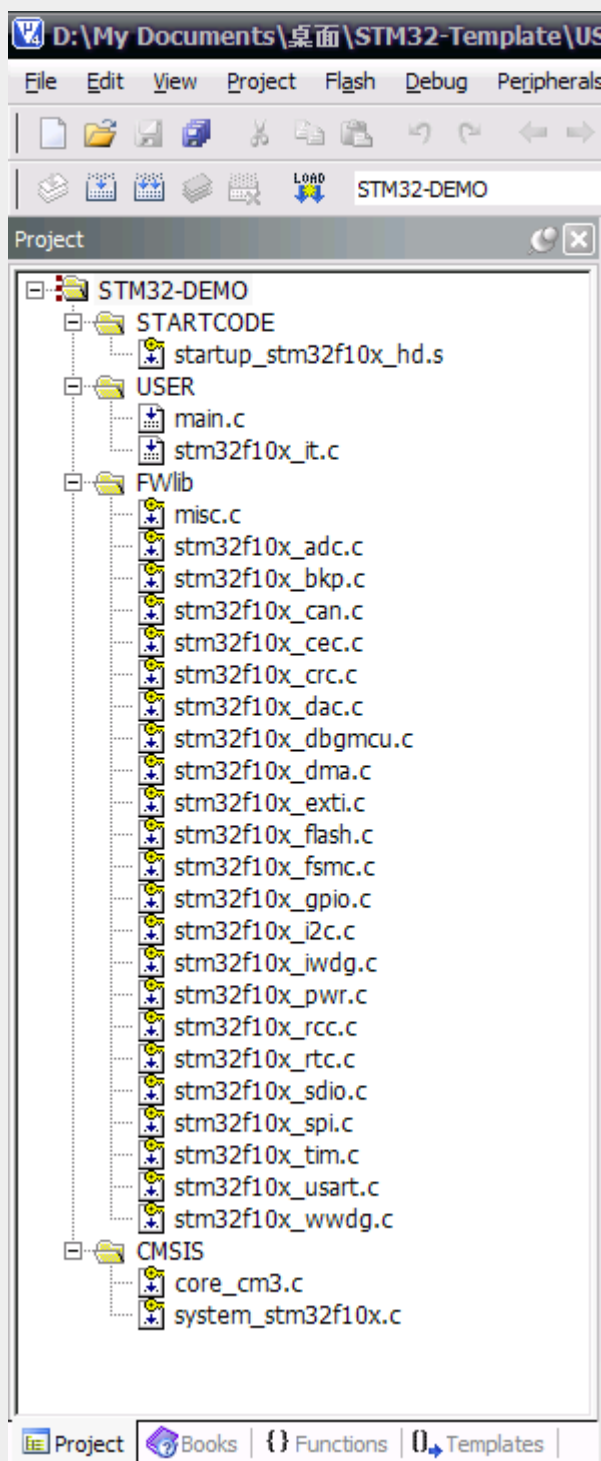
- 回到我们刚刚新建的 MDK 工程中，将 Target 改为 STM32-DEMO（不改也行）






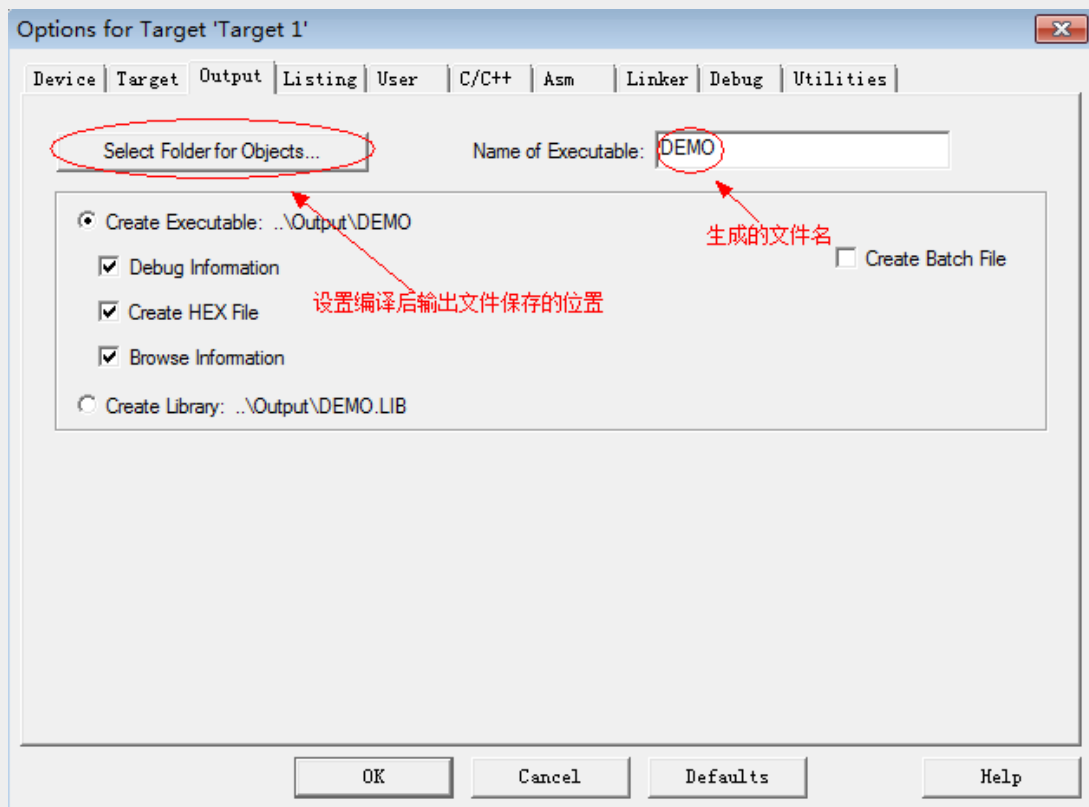
- 在 STM32-DEMO 上右键选中 Add Group...选项，新建四个组，分别命名为 STARTCODE、USER、FWlib、CMSIS。STARTCODE 从名字就可以看得出我们是用它来放我们的启动代码的，USER 用来存放用户自定义的应用程序，FWlib 用来存放库文件，CMSIS 用来存放 M3 系列单片机通用的文件。
- 接下来我们往我们这些新建的组中添加文件，**双击**哪个组就可以往哪个组里面添加文件。我们在 STARTCODE 里面添加 startup\_stm32f10x\_hd.s，在 USER 组里面添加 main.c 和 stm32f10x\_it.c 这两个文件，在 FWlib 组里面添加 src 里面的全部驱动文件，当然，src 里面的驱动文件也可以需要哪个就添加哪个。这里将它们全部添加进去是为了后续开发的方便，况且我们可以通过配置 stm32f10x\_conf.h 这个头文件来选择性添加，只有在 stm32f10x\_conf.h 文件中配置的文件才会被编译。**在 CMSIS 里面添加 core\_cm3.c 和 system\_stm32f10x.c 文件。**注意，这些组里面添加的都是汇编文件跟 c 文件，头文件是不需要添加的。最终效果如下图：



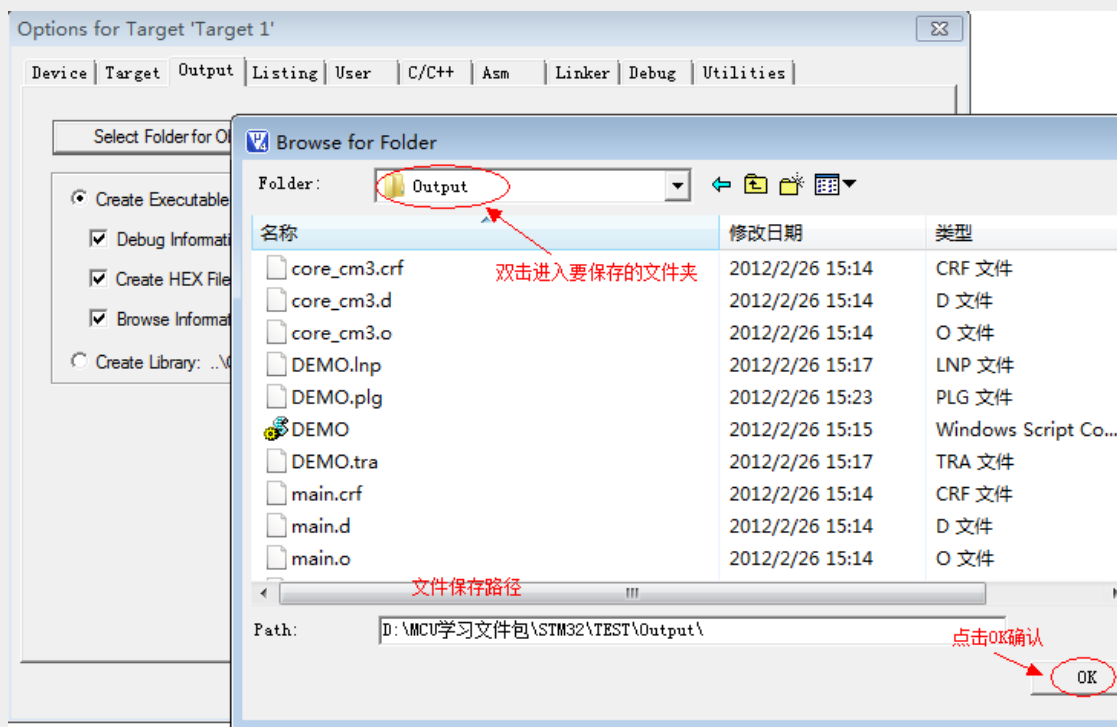


至于有些文件有个锁的图标，是因为这些都是库文件，不需要我们修改，属性为只读。

- 至此，我们的工程已经基本建好，下面来配置一下 MDK 的配置选项，点击工具栏中的魔术棒按钮 ，在弹出来的窗口中选 Output |

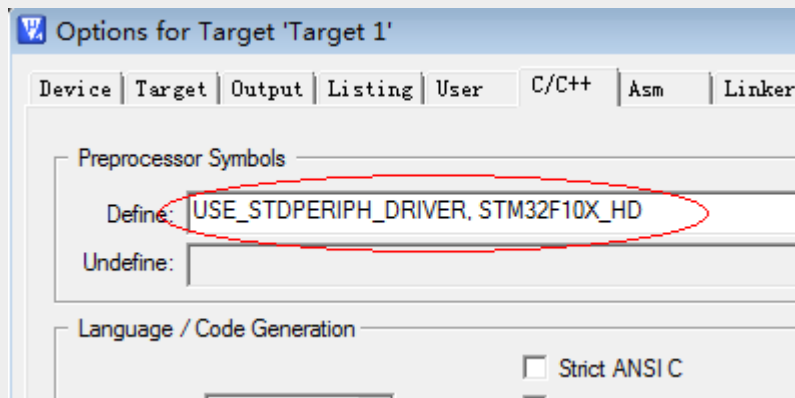


点击 Select Folder for Objects... 设置编译后输出文件保存的位置。同时把 Create HEX File 和 Browse information 这两个选项框也选上。




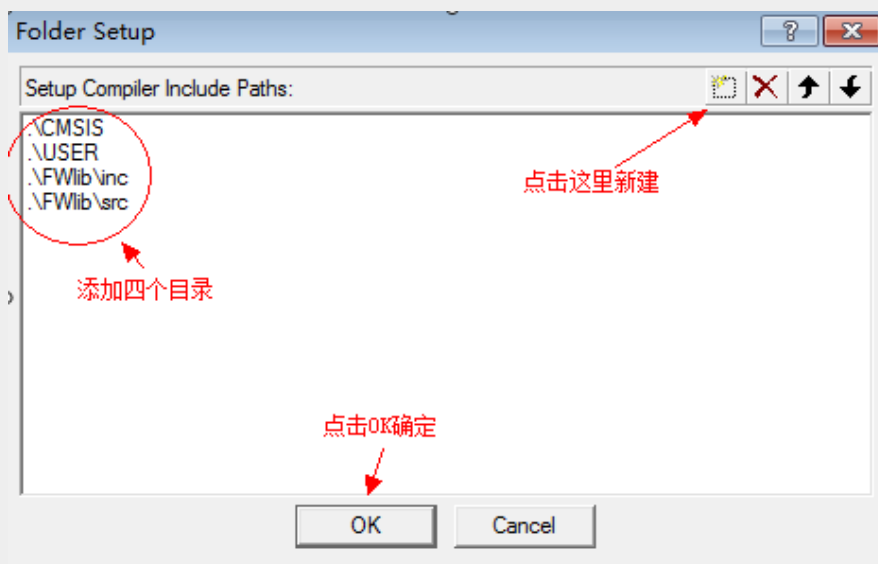
同样在 Listing 这个选项卡中，我们也点击 Select Folder listings...定位到模板中的 Listing 文件夹。

- 选中 **C/C++** 选项卡，在 **Define** 里面输入添加 `USE_STDPERIPH_DRIVER`,  
`STM32F10X_HD`。



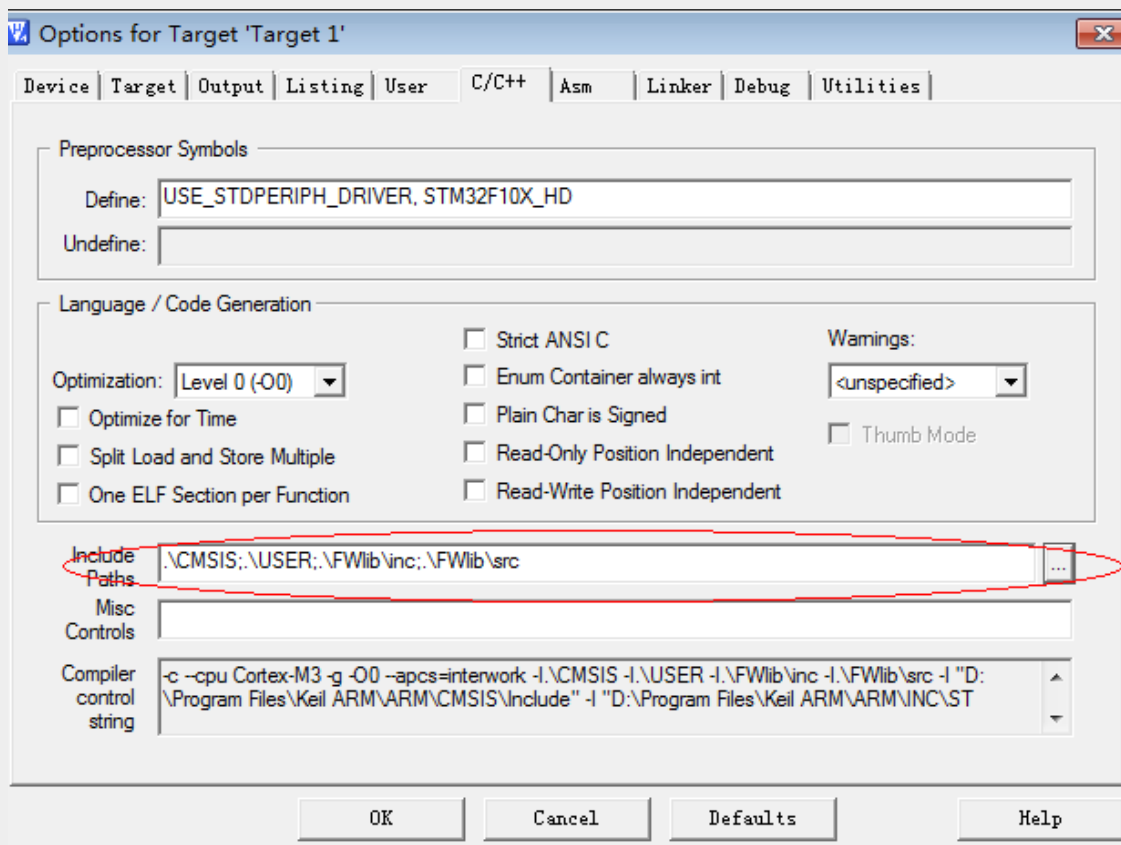
添加 `USE_STDPERIPH_DRIVER` 是为了屏蔽编译器的默认搜索路径，转而使用我们添加到工程中的 ST 的库，添加 `STM32F10X_HD` 是因为我们用的芯片是大容量的，添加了 `STM32F10X_HD` 这个宏之后，库文件里面为大容量定义的寄存器我们就可以用了。芯片是小或中容量的时候宏要换成 `STM32F10X_LD` 或者 `STM32F10X_MD`。其实不管是什么容量的，我们只要添加上 `STM32F10X_HD` 这个宏即可，当你用小或者中容量的芯片时，那些为大容量定义的寄存器我不去访问就是了，反正也访问不了。

在 **Include Paths** 栏点击 ，在这里添加库文件的搜索路径，这样就可以屏蔽掉默认的搜索路径。



但当编译器在我们指定的路径下 搜索不到的话还是会回到标准目录去搜索，就像有些 ANSI C 的库文件，如 `stdin.h`、`stdio.h`。

- 库文件路径修改成功之后如下所示：



- 修改 main.c 文件。因为刚刚我们的 main.c 文件是从官方库里面复制过来的，里面有许多东西我们是不需要的，为了简化 main.c 文件，我们将修改如下。

```
● /***** (C) COPYRIGHT 2012 WildFire Team *****/
●
● * 文件名 : main.c
● * 描述 : 用 3.5.0 版本建的工程模板。
● * 实验平台: 野火 STM32 开发板
● * 库版本 : ST3.5.0
● *
● * 作者 : wildfire team
● * 论坛 : http://www.amobbs.com/forum-1008-1.html
● * 淘宝 : http://firestm32.taobao.com
● *****/
● #include "stm32f10x.h"
●
●
●
● /*
●
● * 函数名: main
```




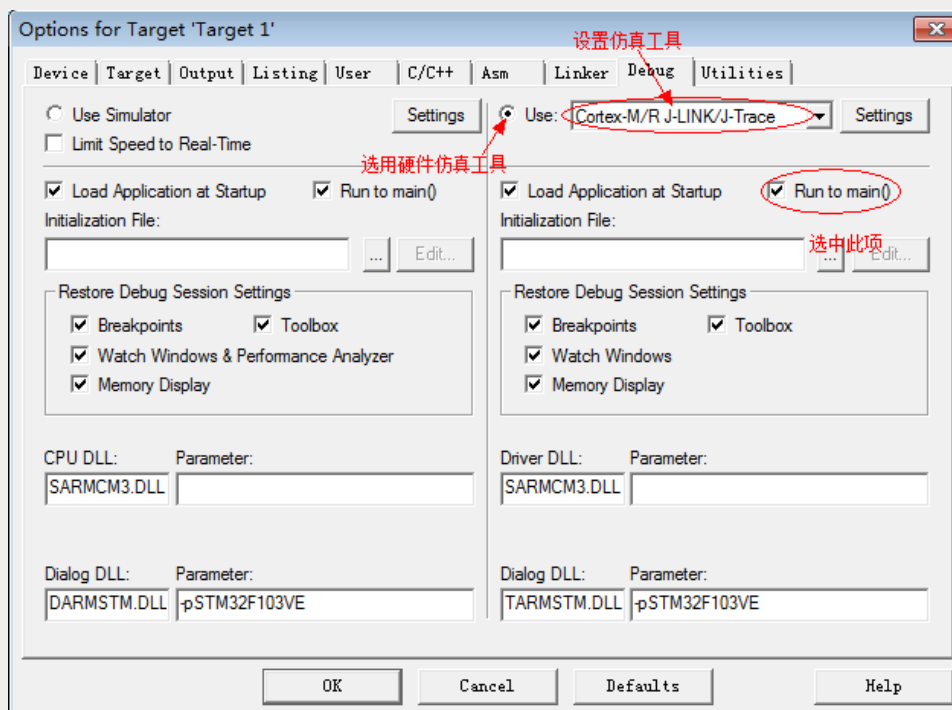
```
• * 描述 : 主函数
• * 输入 : 无
• * 输出 : 无
• */
• int main(void)
• {
•     while(1);
•     // add your code here ^_^。
• }
• /***** (C) COPYRIGHT 2012 WildFire Team *****/
```

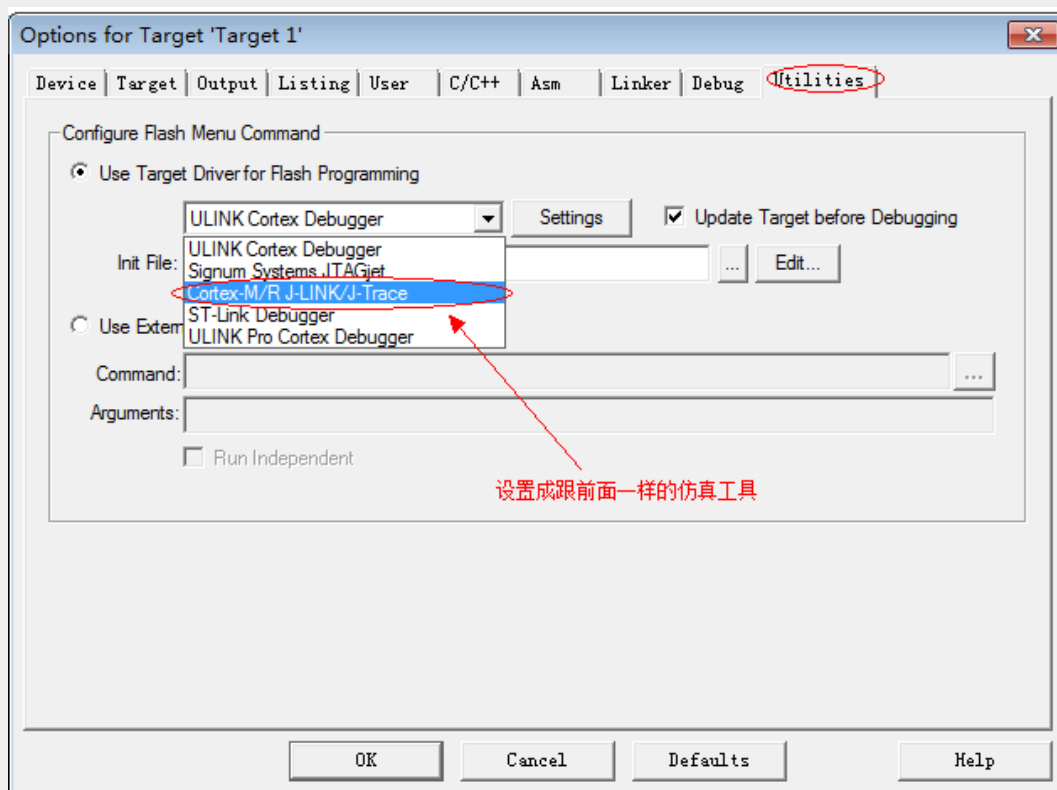
- 至此，我们的工程模板就建成了。学会新建工程，是学习 stm32 的第一步。

### 3.3 硬件调试配置

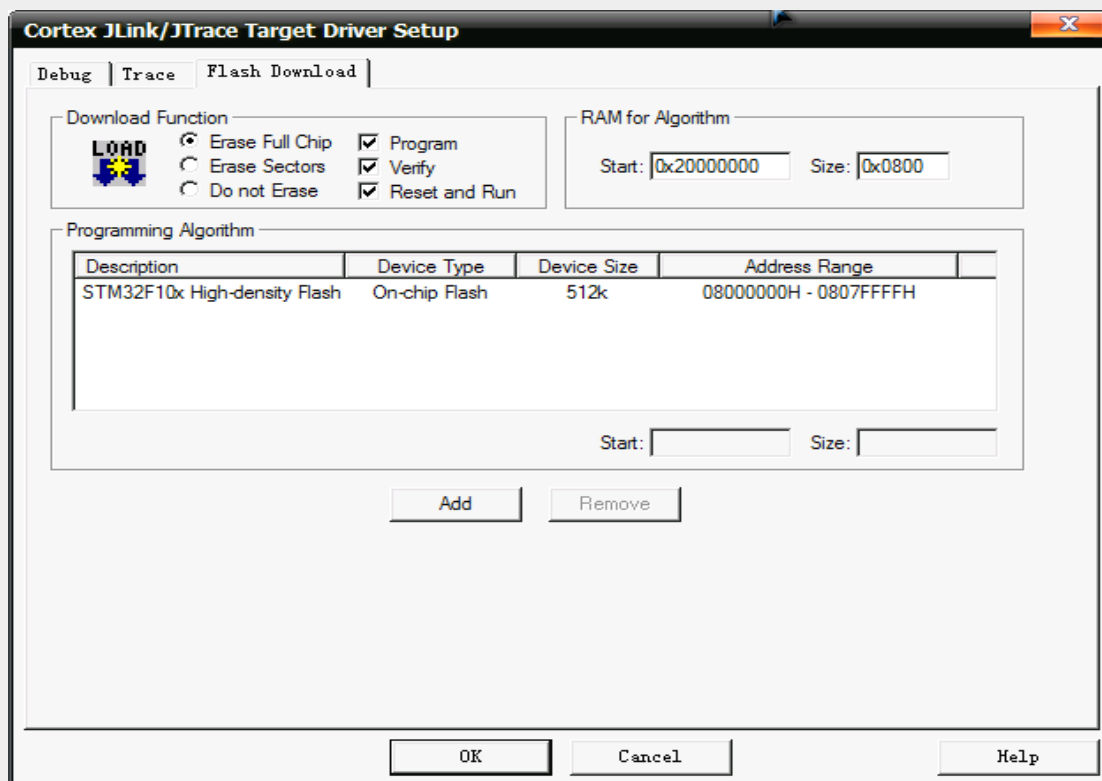
这个工程默认的是软件仿真，如果开发板要用 J-LINK 调试的话，还需要在开发环境中做如下修改。实际上，我们开发程序的时候 80% 都是在硬件上调试的。

具体配置如下图所示：点击 ，在 Debug 选项里





在选项卡 Debug\Setting\Flash download 中我们设置成如下：



- 到了这里就算是大功告成了。如果在新建工程中遇到什么问题，先不要急，可先参考野火 M3 光盘目录下提供的已经新建好的工程模板。



