

工程建立说明

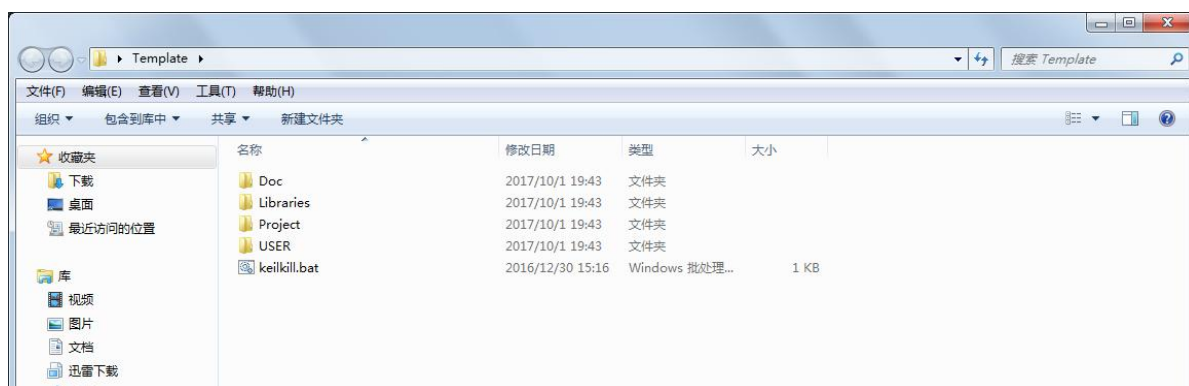
说明：这是一个基于 **F103** 建立一个标准库函数版本的工程模板的完整过程。

• 空白工程建立完成后，可以作为一个工程模板，之后写新工程时可以在这个基础上作一些配置上的微调，然后直接开始代码的编写。不必每次都如此麻烦。

建立一个空白工程

0. 首先，我们需要做一些准备工作

(0)、新建一个文件夹，为它命名，在文件夹下建立如下几个文件夹（有一个明晰的结构对一个整洁的工程非常重要!）:

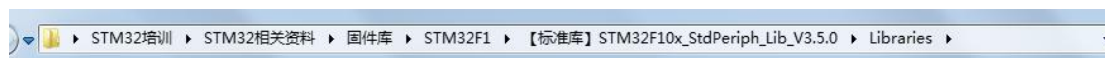


下面一一介绍主文件夹的内容：

(1)、**DOC**——里面放一个介绍文档，主要用来做版本记录和工程说明

(2)、**Libraries**——主要放置库函数的源文件与头文件，下面介绍如何从标准库中选取我们需要的文件：

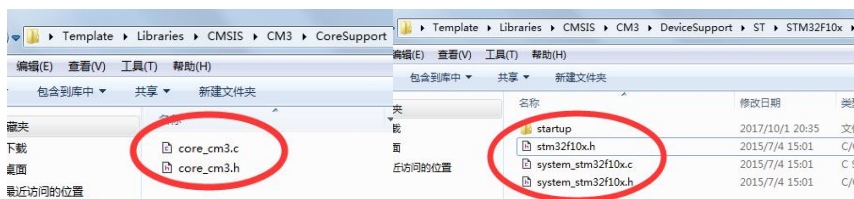
①：按照路径打开标准库，打开 **Libraries** 文件夹



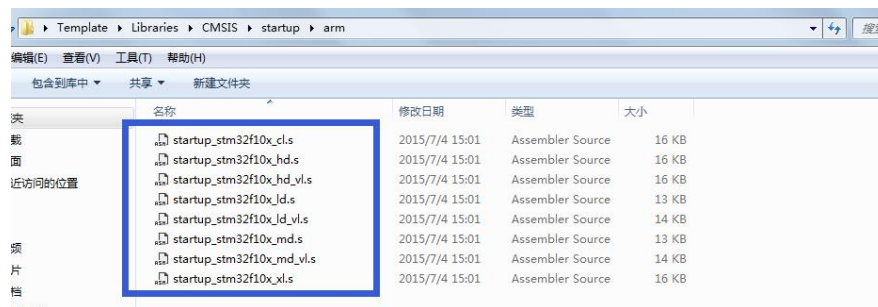
把里面的两个文件复制粘贴到我们的工程文件夹的 **Libraries** 文件夹中

②：整理复制到工程文件夹中的文件（简化文件夹的结构，删除无用文件）:

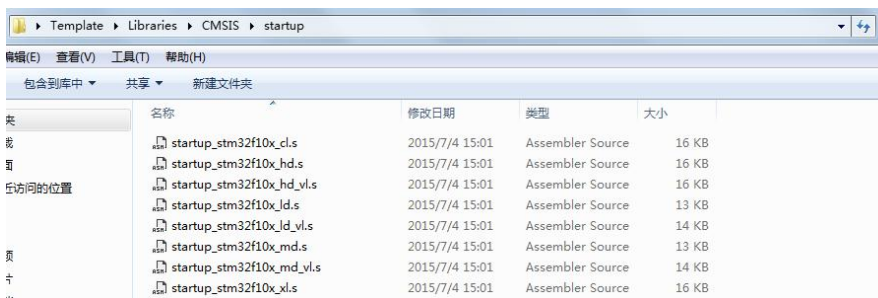
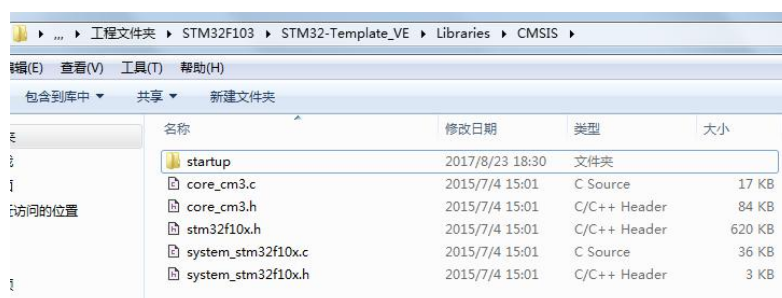
• **CMSIS** 文件夹下，按路径把如下文件直接复制粘贴到 **CMSIS** 文件夹中，其他的東西全部删除：



• **Startup** 文件夹，把如下文件复制到 **Startup** 文件夹下，其他文件全部删除（那些是在其他编译器下使用的启动代码，我们仅保留 MDK 中需要使用的代码）：



获得如下效果：

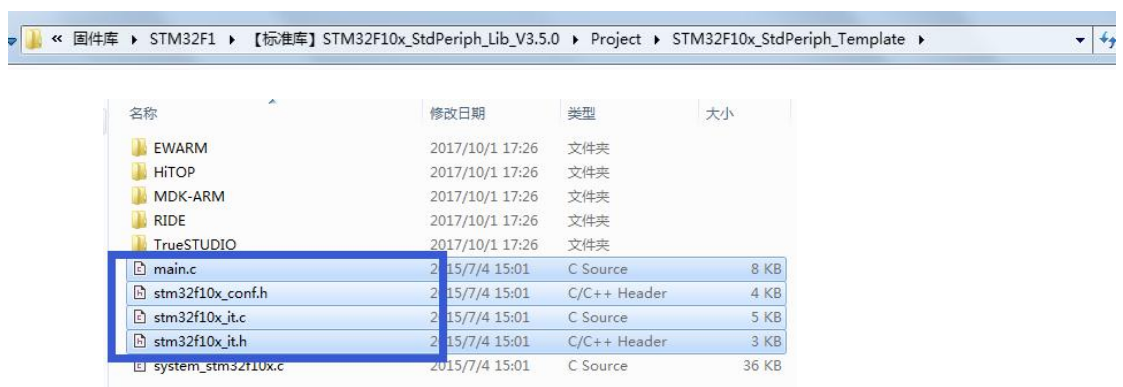


• **Libraries** 文件夹不需要改动。

(3)、**Project**——我们建立工程的目标文件夹

(4)、**USER**——放置主函数体、中断相关的源文件、特定的头文件等等（是固定的几个文件，一般不变）：

• 可以从标准库给出的模板工程中 **COPY** 出这些文件：



之所以不选择最后一个，是因为在 **CMSIS** 文件夹中我们已经有了那个文件了。各文件功能如下表：

stm32f10x_conf.h:	用来配置库的头文件
stm32f10x_it.h	
stm32f10x_it.c:	中断相关的函数都在这个文件编写，暂时为空
main.c:	main 函数文件

(5)、**APPS**——主要放置一些与具体功能相关的代码（可以更名，可以扩展），下设两个文件夹，如图：



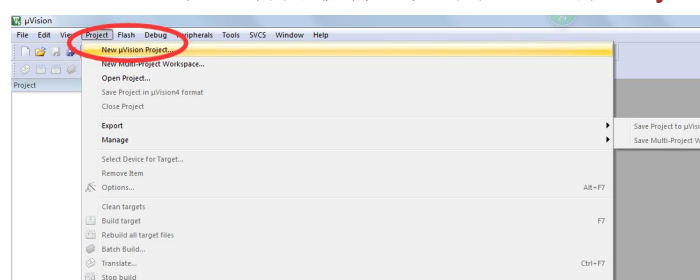
• **Inc** 文件夹下是我们要写的代码的头文件，**Src** 文件夹下是我们要写的代码的源文件

(6)、**keilkill.bat**——这是一个 **Windows** 批处理文件，双击运行时可以删除我们工程在编译后生成的链接文件。这在我们传递代码的时候是一个非常有用的东西，一个比较大的工程在编译后会有 **20M** 到 **50M**，而删除掉那些文件后整个工程便仅剩下代码和配置文件，大概几百 **KB** 的样子，上传、下载或者保存的成本将会大大减少（它附在例程文件夹里）。

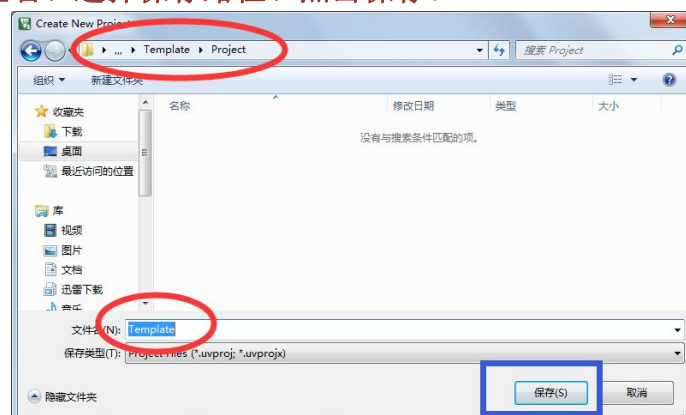
至此，我们的准备工作就结束了。

/***** 我是分割线 *****/

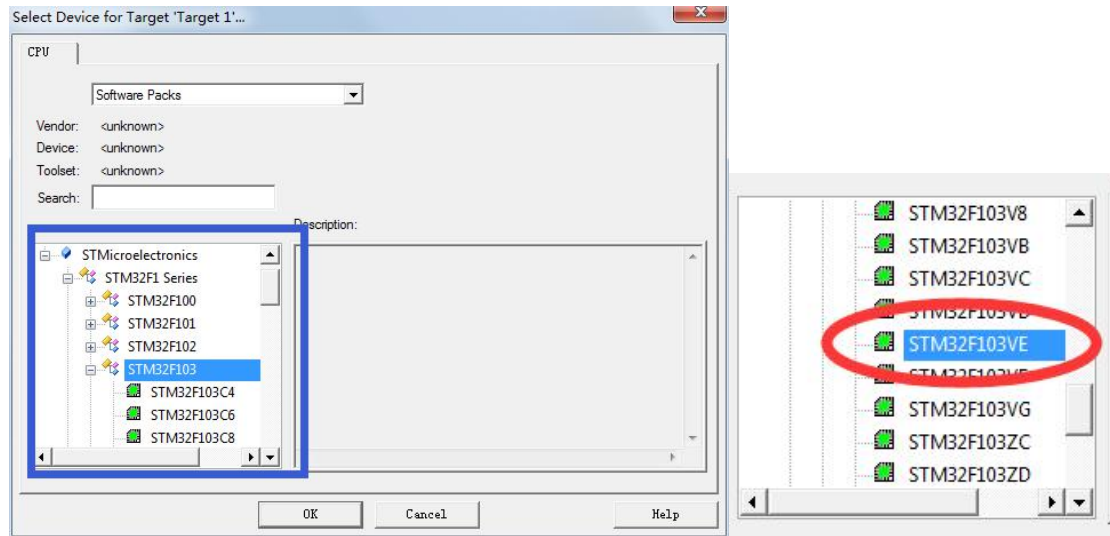
1. 安装好 **Keil5 (MDK)** 后，打开软件，如下图，选择 **Project**, 新建工程：



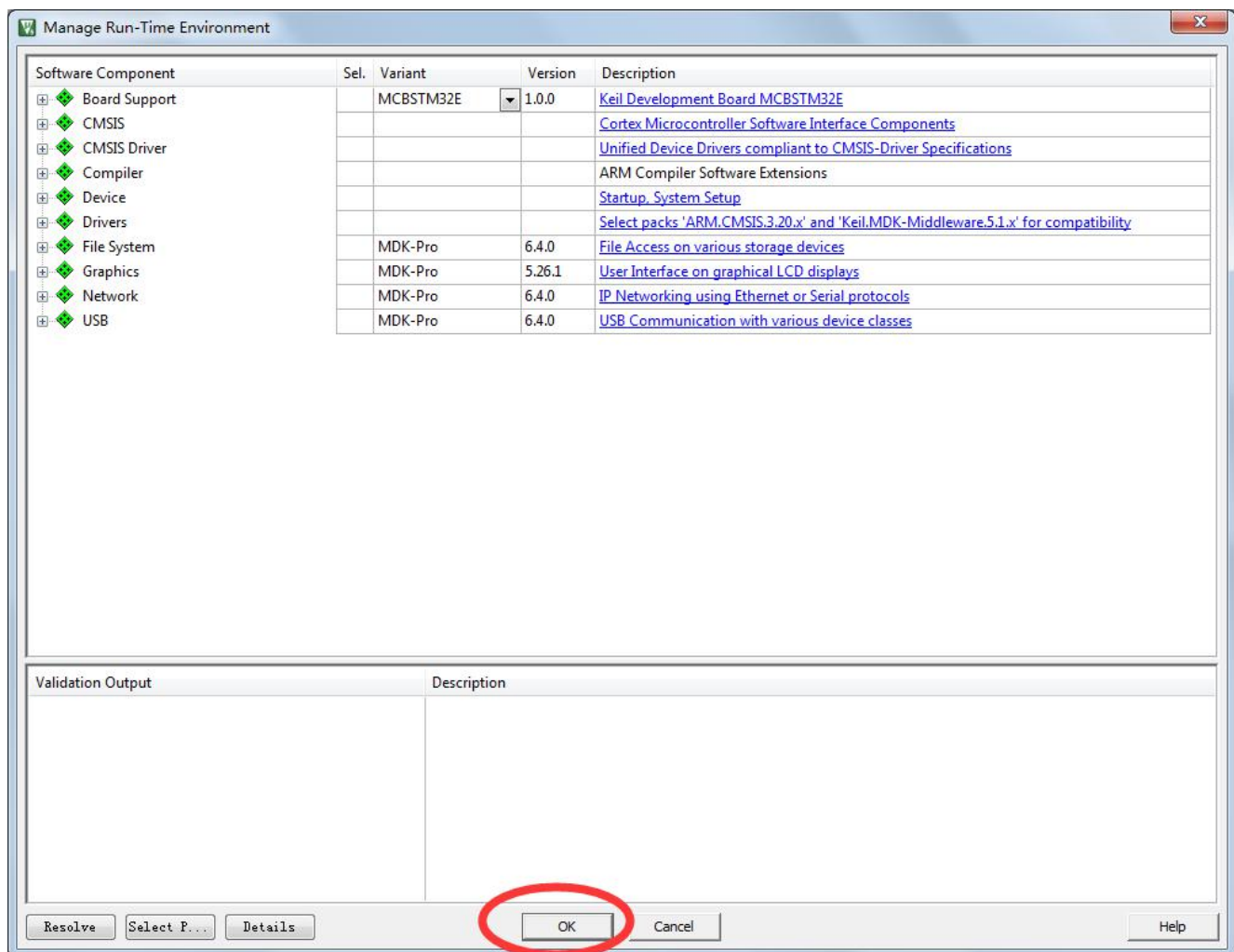
2. 输入工程名、选择保存路径、点击保存。



3.在框内寻找到我们开发板上所使用的芯片型号 **STM32F103VET6**,点击“OK”:

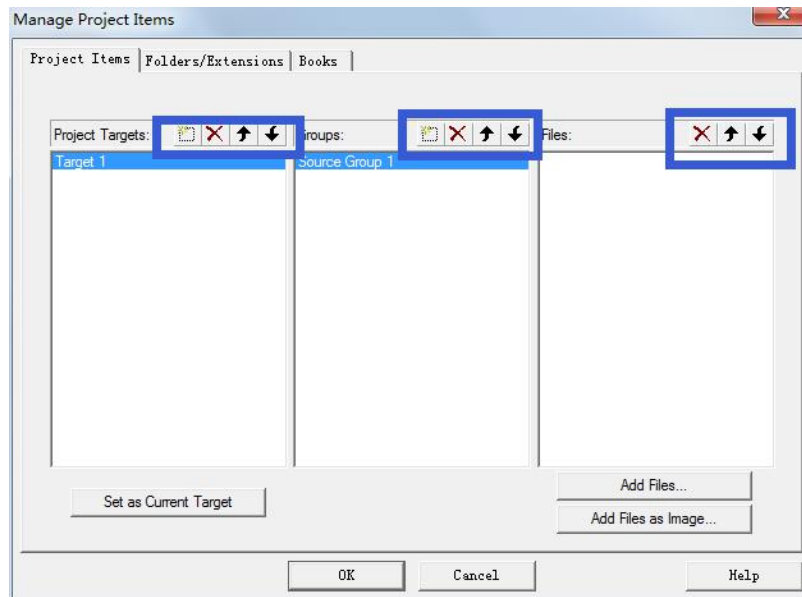


4.点击“OK”

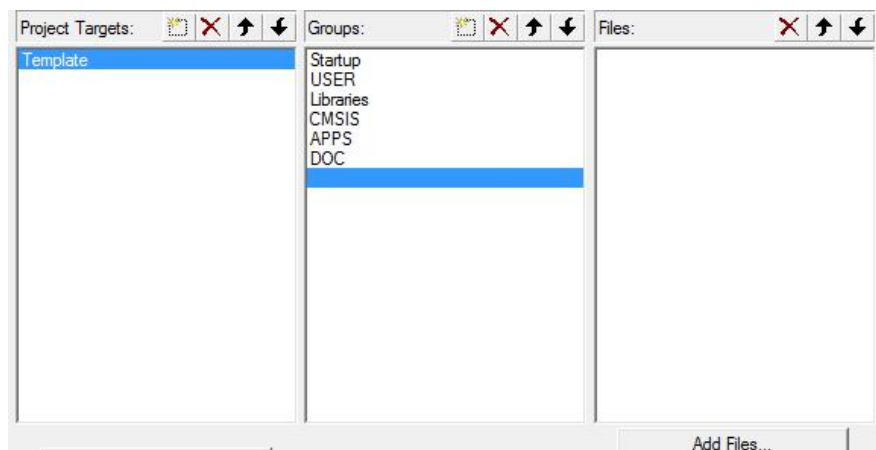


5.配置工程结构:

找到工具栏中这个品字形的按钮，开始配置:

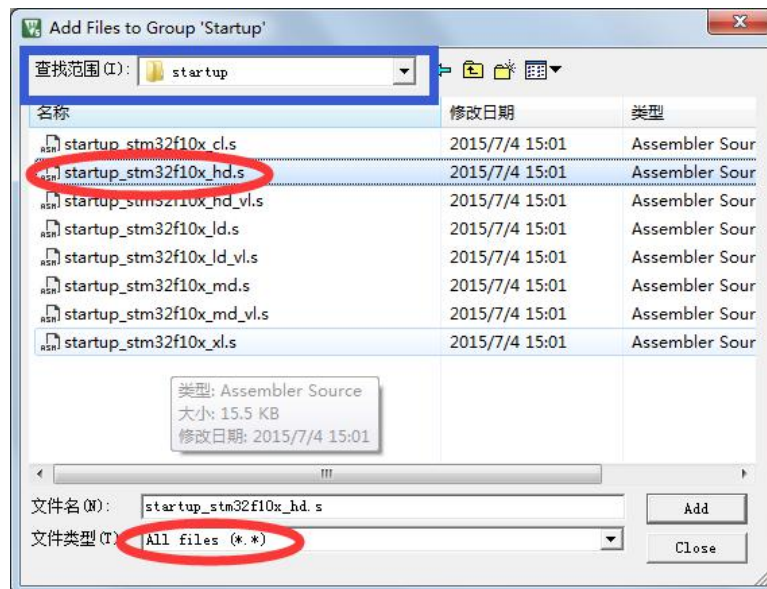


- 一共有三个可以编辑的框，从左到右依次是工程编译的目标、目标下的分组、分组中的文件。
- 被方框框住的是一些功能按钮，从左往依次是新建、删除和上移、下移。
- 双击名字，可以修改目标和分组的名字。
- 点击 **Add Files...**按钮，可以添加你想添加的源文件到选定分组中。
- 参照我们文件夹的结构，配置成如下的样子:

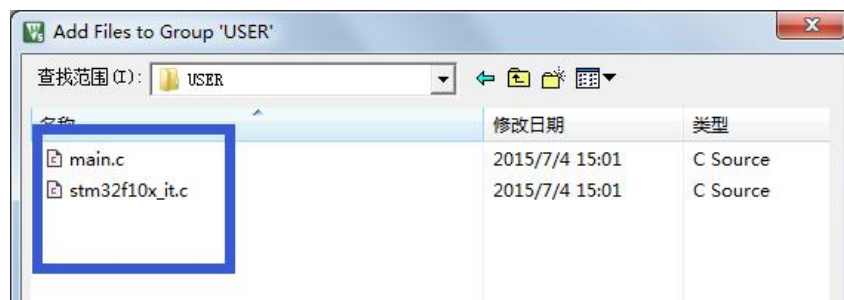


- 现在开始添加文件（点击 Add Files...按钮）:

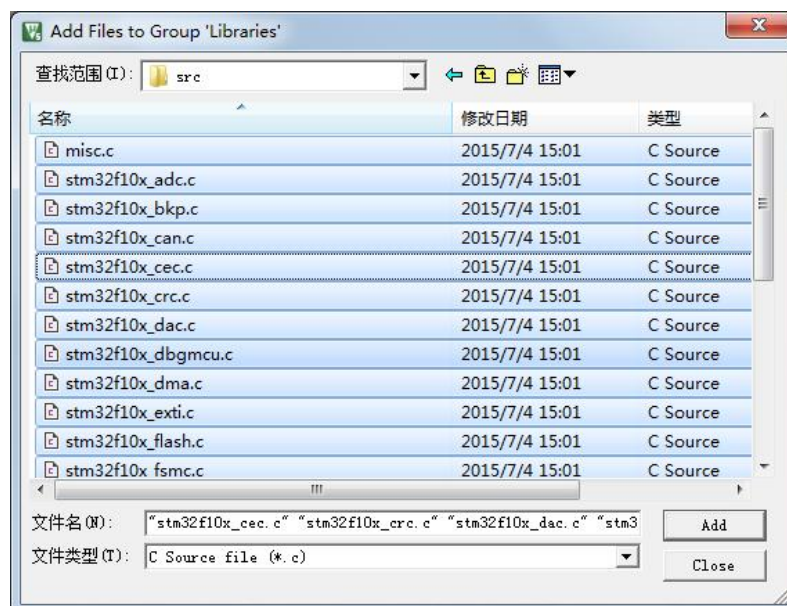
Startup: 在文件夹中找到 **startup**, 选择 **_hd.s** 的文件（我们芯片所对应的启动代码）



USER: 在文件夹中找到 **USER**, 选中并加入 (Tips:Ctrl+鼠标 Click 为多选)

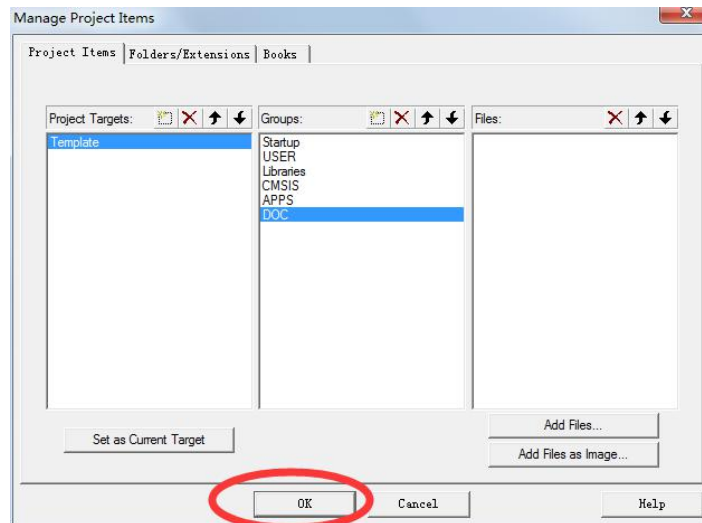


Libraries: 在文件夹中找到 **STM32F10x_StdPeriph_Driver** 中的 **src**, 选中所有并加入(Tips:Ctrl+A 全选)

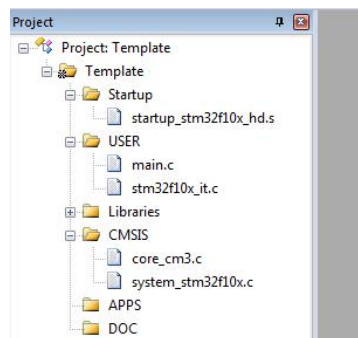


CMSIS:在文件夹中找到 **CMSIS**，选中里面的两个源文件并加入

配置完成，点击“OK”



工程目录的效果：



- **Tips:**在工程目录慢点双击分组或者目标名字也可以对它们的名字进行编辑。

6.配置下载器选项、包含路径和公共宏

在工具栏找到魔术棒按钮，点击进入选项卡

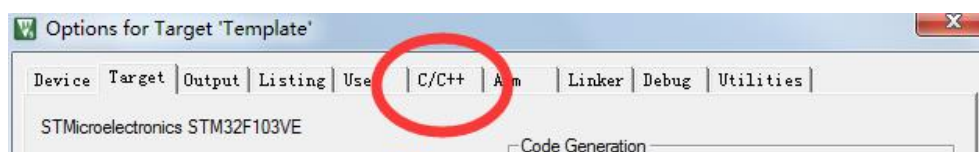


(1)、配置包含路径

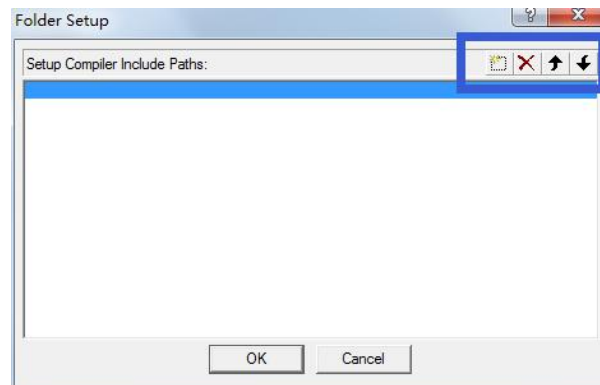
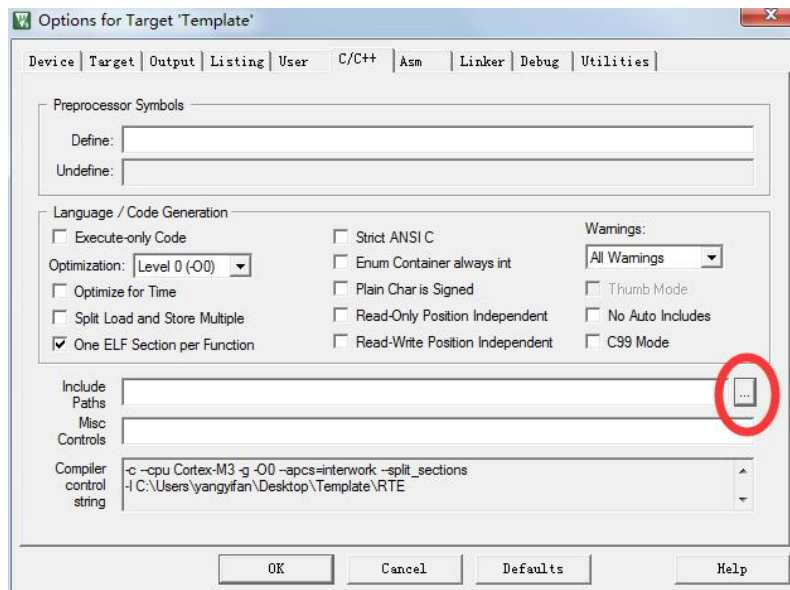
包含路径，就是告诉编译器编译时源文件所包含的头文件所在的位置。

按照如下操作：

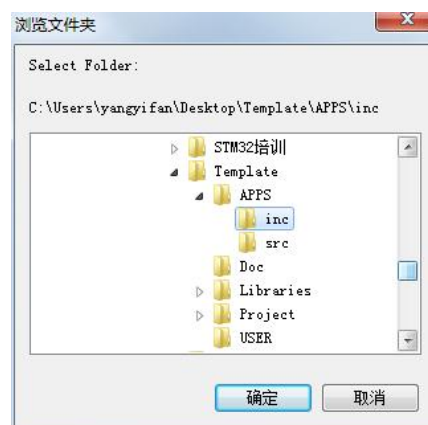
- 点击图示选项卡



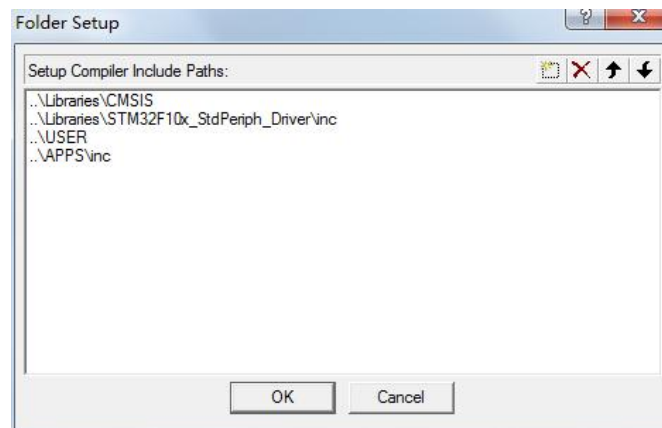
- 点击图示按钮



- 这几个功能按钮与上面工程结构配置面板中的功能相同
- 【举例】添加一个新路径，点击按钮，定位到目标路径下：

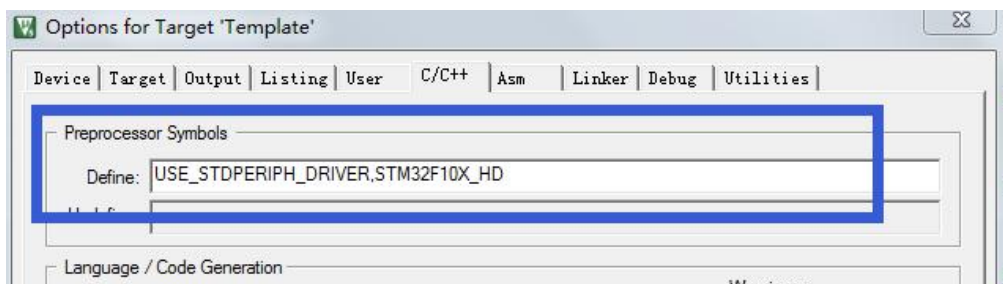


- 最终添加如下几个路径，点击 **OK**：



(2)、配置公共宏

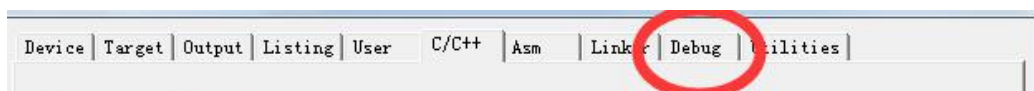
所谓公共宏，是编译器添加在每一个源文件中的宏，是对整个工程声明，我们为了使用标准库，需要在此处声明一下。同时，还要声明我们使用的芯片型号，这样一些编译选项才可以正常地被配置。（上课的时候会讲解如何去查找这些宏）



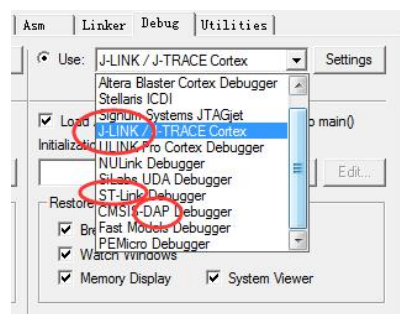
此处要声明的是两个：**USE_STDPERIPH_DRIVER,STM32F10X_HD** 注意，二者之间要以英文逗号隔开，不然编译时会报许多个错（相当于每个源文件都出现了错误）。

(3)、配置调试选项和下载选项

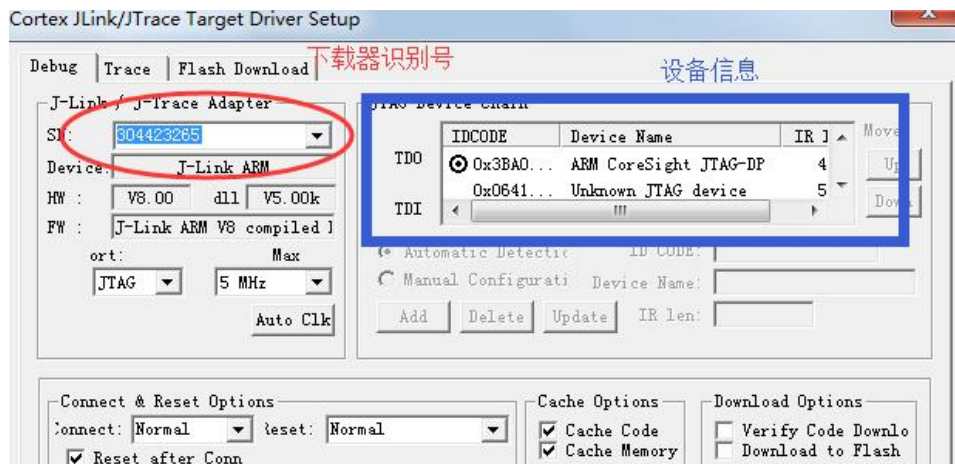
- 调试选项，选中“**Debug**”选项卡



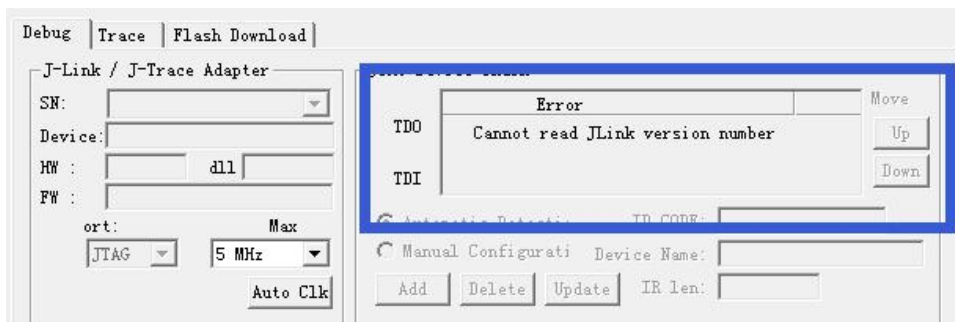
- 在这里选择你所使用的下载器类型



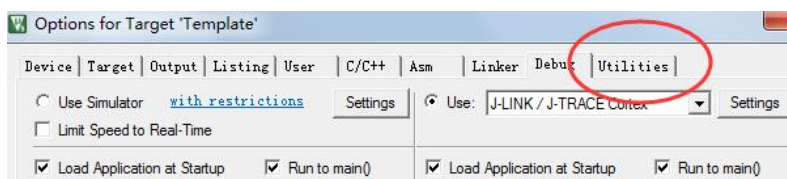
- 再点击旁边的 **Settings** 按钮，如果此时你的下载器连接着电脑且驱动安装完毕，则在如图框内出现你所使用的设备信息：



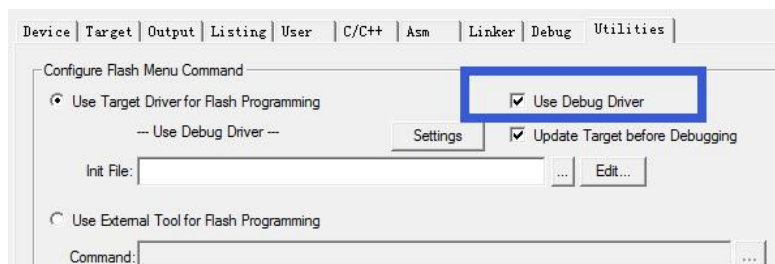
- 如果不正常（未识别设备），会是这样的：



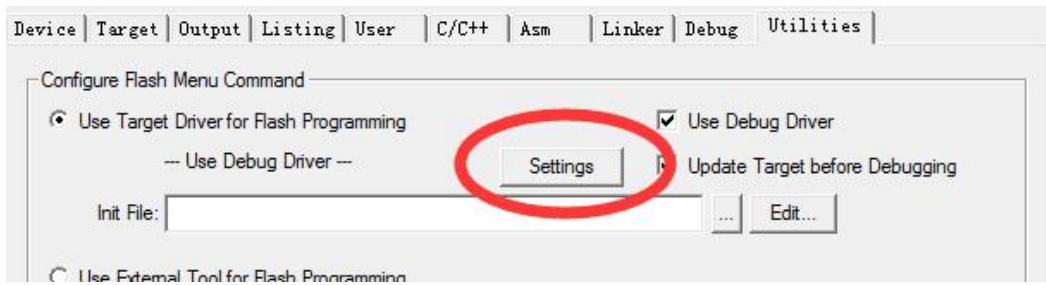
- 如果一切正常，配置下载选项。点击最后一个选项卡。



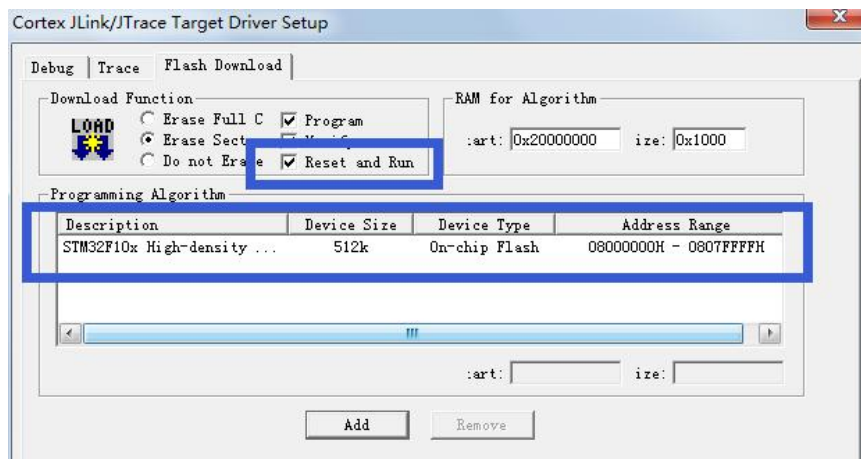
- 勾选图示选项（使用调试设备下载程序）



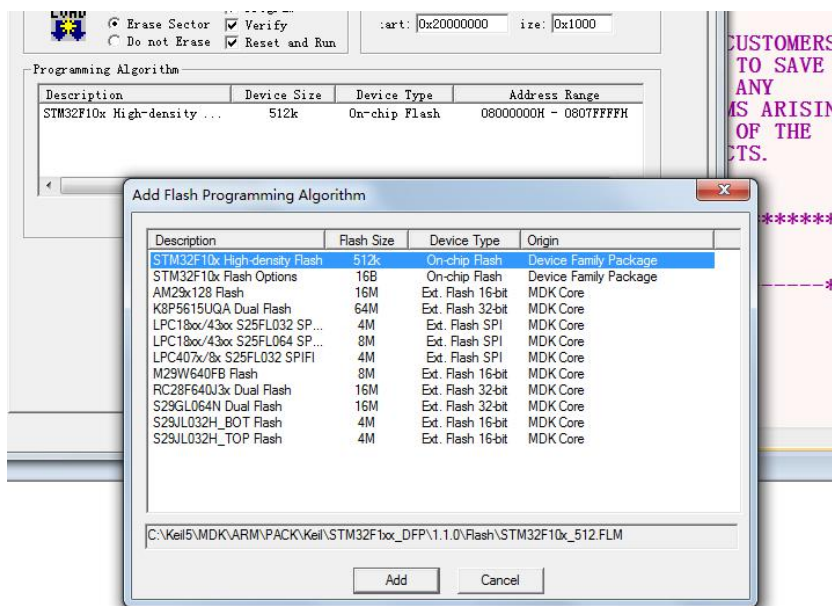
- 点击 “Settings”



- 注意下面框起来的这两个区域，上面那个需要打钩，（如果不勾选，那么下载程序后程序将不会自动运行，需要按下复位键才会运行）。下面那个地方是我们单片机 Flash 的规格。



- 如果规格那里是空的，那么可以点击下面的 “Add” 按钮添加相应的型号即可（如果不添加，会导致下载失败）。



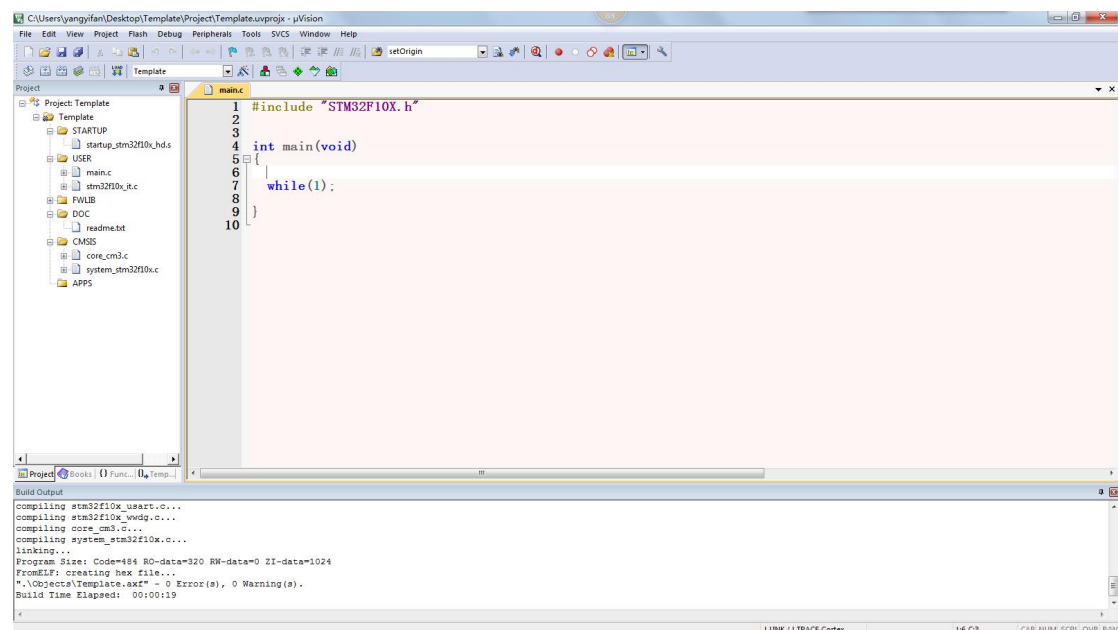
至此，我们所有的配置工作就全部结束啦！

/****** 我是分割线 *****/

```
main.c
1 #include "STM32F10X.h"
2
3
4 int main(void)
5 {
6
7     while(1);
8
9 }
10
```

把主函数中原有的代码清除干净，加上头文件，写一个简单的 main 函数框架。（死循环是必须的！）

编译完成无误后，一个空白工程模板就全部完成啦！（不要在意为什么背景是粉色的...-_-!!）



HIT-ROVER 团队
杨亦凡
2017.10.02