

基于 keil4 的 stm32 工程创建步骤

WWM 2017.07

写在前面：鉴于很多人学习 stm32 的时候，一开始工程不会建，所以在此编写这个 PDF 教程。本工程使用的是官方 stm32 固件库的 V3.5 版本，编译环境是 Keil for ram，如下图（图一）所示

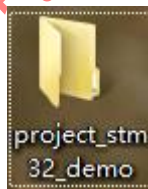
htmresc	2016/9/26 16:16	文件夹	
Libraries	2016/9/26 16:16	文件夹	
Project	2016/9/26 16:16	文件夹	
Utilities	2016/9/26 16:16	文件夹	
Release_Notes.html	2015/12/4 15:57	搜狗高速浏览器H...	111 KB
stm32f10x_stdperiph_lib_um.chm	2015/12/4 15:57	编译的 HTML 帮...	19,189 KB

图一

其中在建立工程的时候我们只需要 Libraries 和 Project 两个文件夹里的内容即可。Libraries 里面包含的有底层外设驱动函数以及内核函数。Project 里面有一些例子，后续只需复制 Project 里面的几个文件即可。

第一步：

准备工作①：新建一个文件夹，随意命名即可（图二）。接着在工程文件夹下新建一个 user 文件夹放置用户的主函数文件以及其他文件，再新建一个 output 文件夹放置工程编译过程的一些中间文件，再新建一个 app 文件夹放置用户所编写的具体模块功能实现函数。本例子这样设置文件夹，用户也可以根据自己的习惯去创建命名文件夹等。



图二

准备工作②：将 Libraries 整个文件夹复制粘贴到我们建的文件工程夹 project_stm32_demo 里，然后打开固件库 V3.5 里/project/STM32F10x_StdPeriph_Template，在此目录下复制 main.c 文件，stm32f10x_it 的 C 文件和 H 文件，以及 stm32f10x_conf.h 文件、system_stm32f10x.h 文件，再讲这些文件粘贴到 project_stm32_demo 工程文件夹下的 user 文件夹内。复制完这些文件后，我们的工程下的文件如图三(a,b,c,d)所示

project_stm32_demo				
	名称	修改日期	类型	大小
	Libraries	2017/7/22 22:24	文件夹	
	output	2017/7/22 22:35	文件夹	
	user	2017/7/22 22:35	文件夹	
	app	2017/7/22 22:35	文件夹	

图三 a

project_stm32_demo > user				
名称	修改日期	类型	大小	
main.c	2015/12/4 15:57	C 文件	8 KB	
stm32f10x_conf.h	2015/12/4 15:57	H 文件	4 KB	
stm32f10x_it.c	2015/12/4 15:57	C 文件	5 KB	
stm32f10x_it.h	2015/12/4 15:57	H 文件	3 KB	
system_stm32f10x.c	2015/12/4 15:57	C 文件	36 KB	

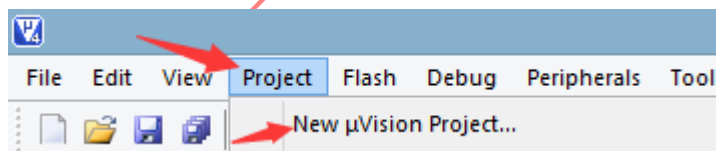
图三 b: user 文件夹下内容

project_stm32_demo > Libraries				
名称	修改日期	类型	大小	
CMSIS	2017/7/22 22:24	文件夹		
STM32F10x_StdPeriph_Driver	2017/7/22 22:24	文件夹		

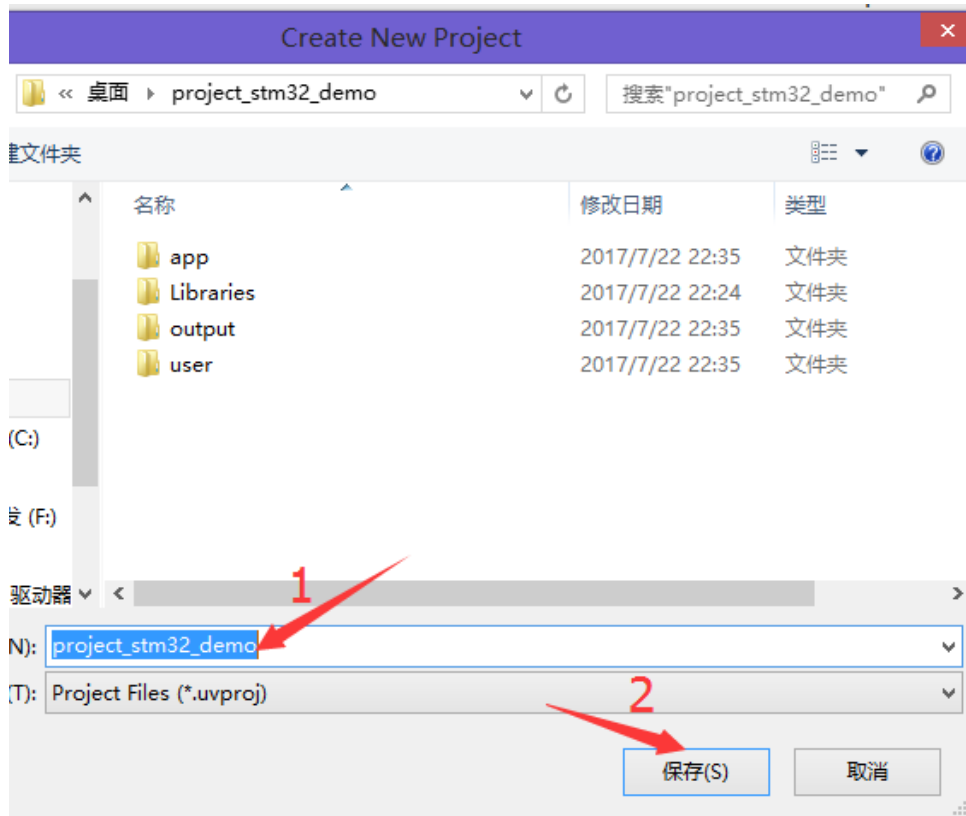
图三 c: Libraries 文件夹下内容

第二步:

打开 keil4.6 软件(确定是 keil for ram, 初学者有的把 keil for 51 当做是 keil for ram)。然后点击 Project->New UVision Project 新建一个工程, 工程保存在工程文件夹下即可, 命名随意,如图四(a,b)。

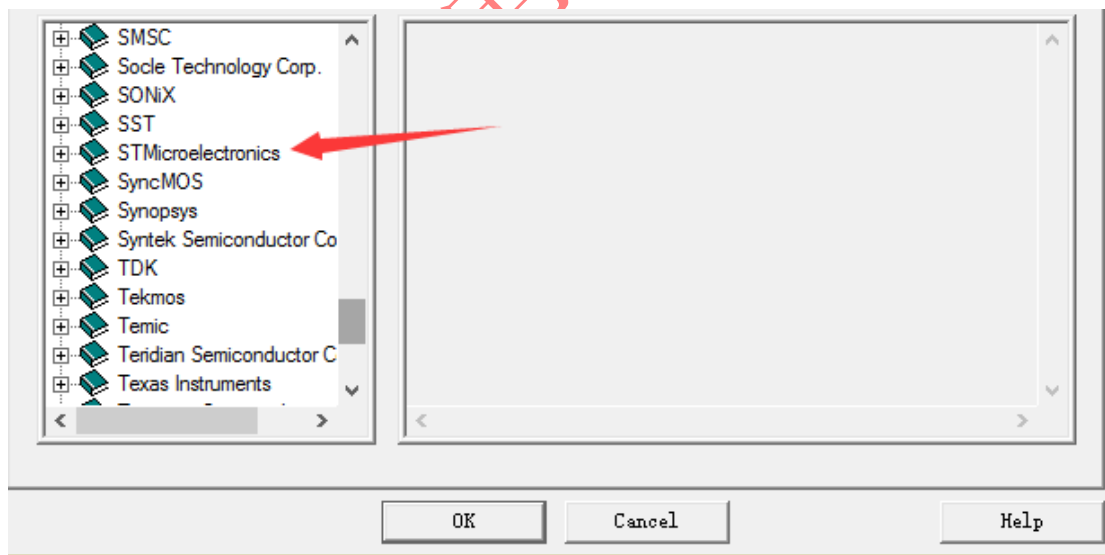


图四 a



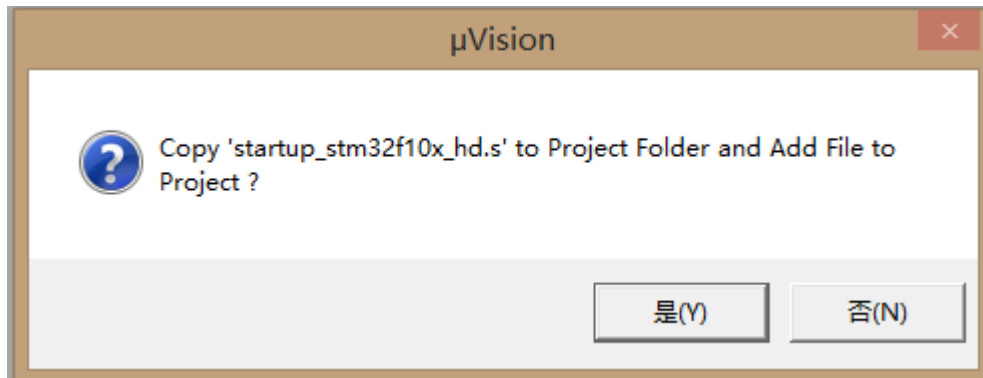
图四 b

点击保存即可。找到 STMicroelectronics,在里面选择自己所要用到目标单片机的型号，根据实际板子型号进行选择，选择好之后点击 OK 即可，如图五(a,b,c)。



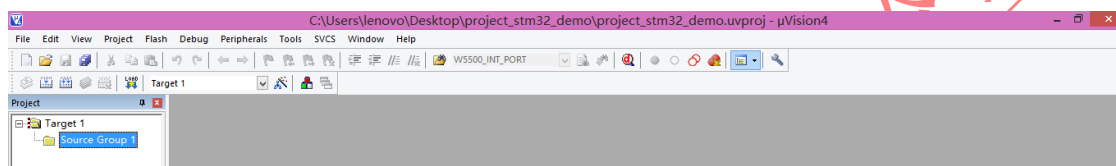
图五 a

在弹出的对话框里我们选择否即可。



图五 b

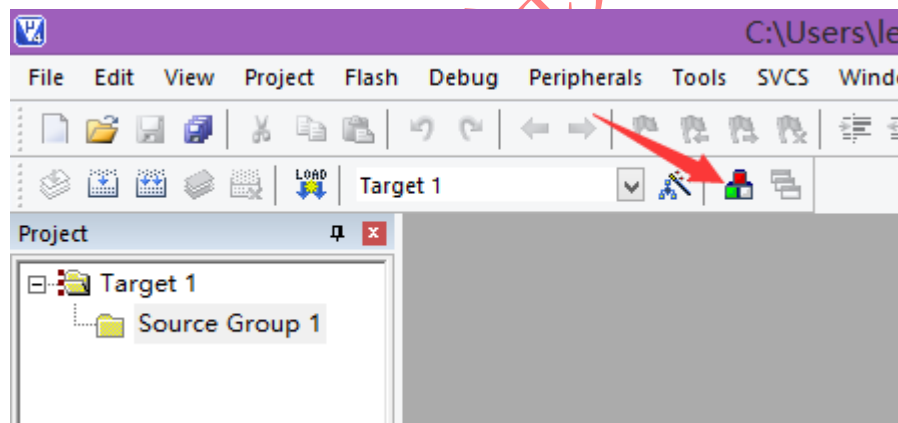
这样就建立了一个基本的工程框架。



图五 c

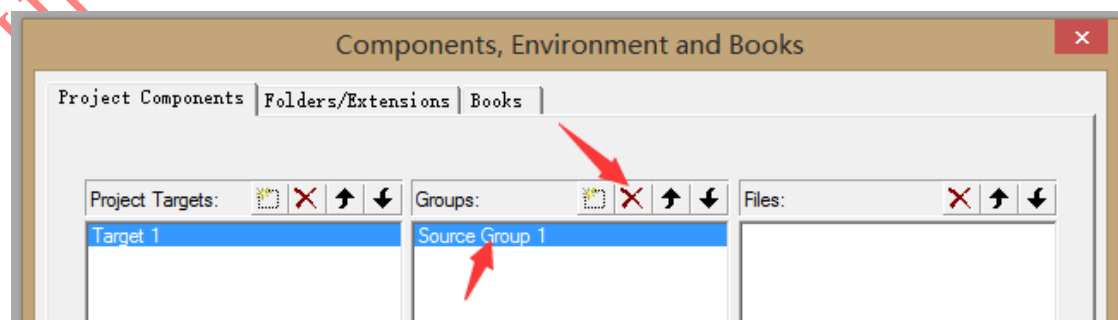
第三步：

点击图六中的按钮选项。



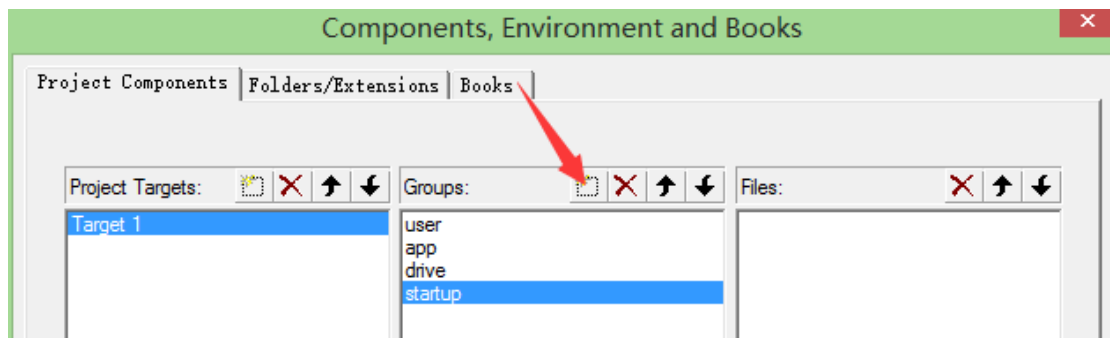
可将对话框内这个工程带的这个组删除

图六 a



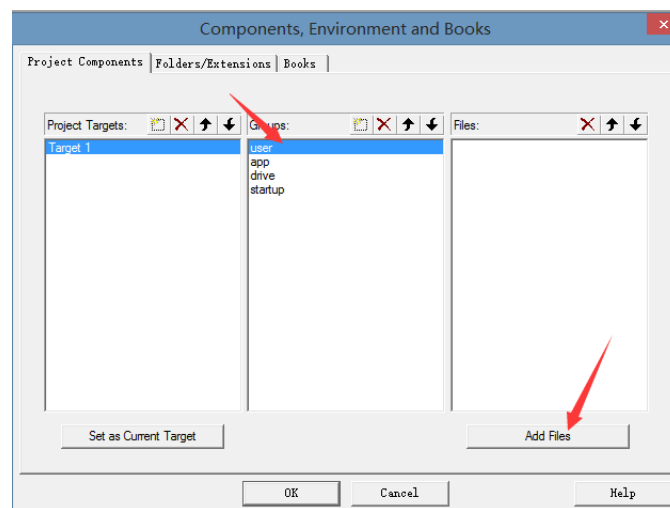
图六 b

然后点击这个选项按钮新建四个组，分别为 user，drive，app，startup，如下图七所示

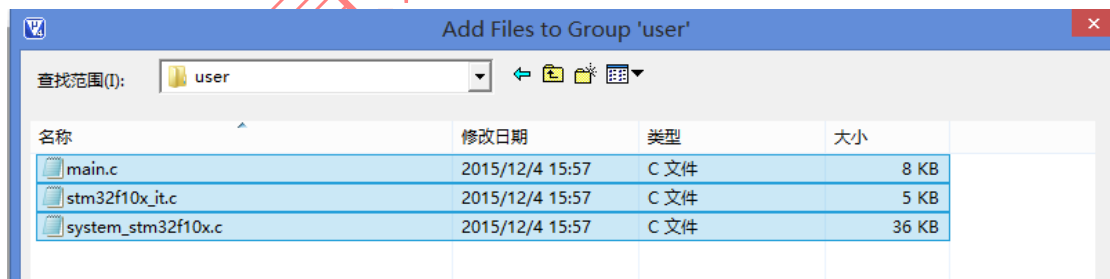


图七

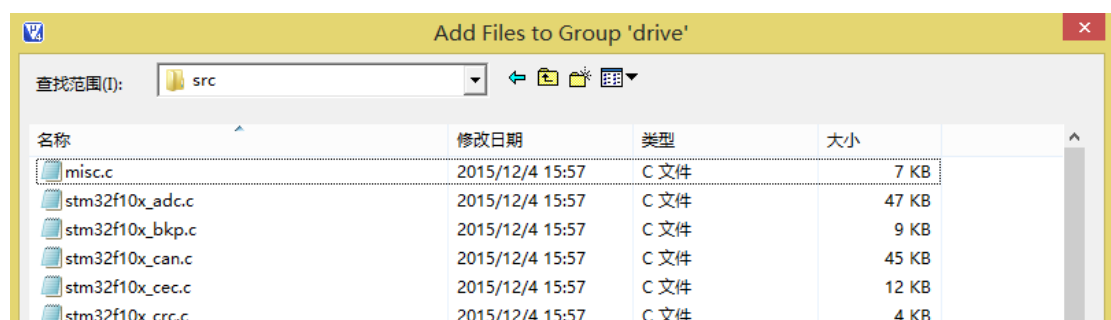
然后就是为每个组添加文件。



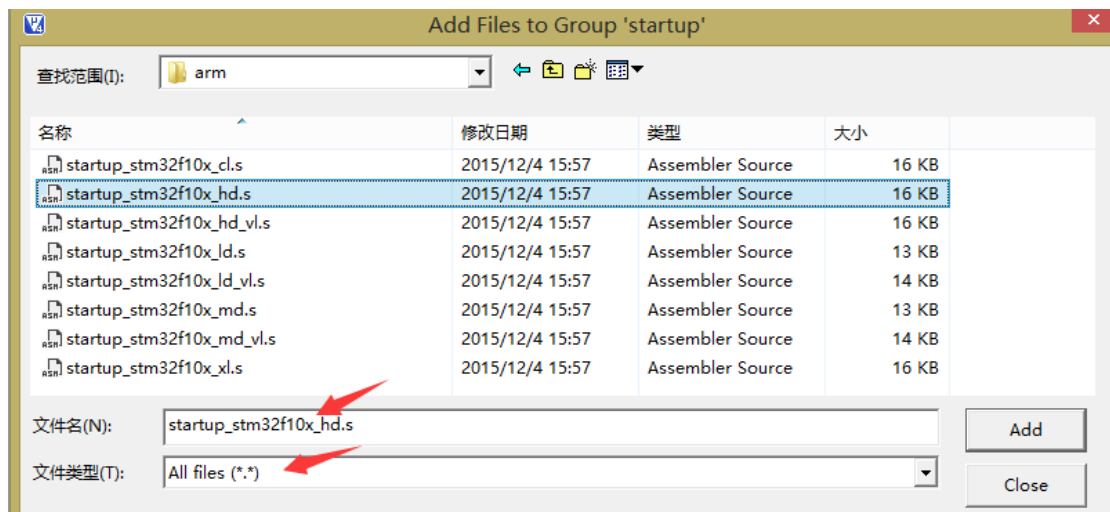
其中 user 选项里将工程下的 user 文件夹里的 C 文件添加进来。



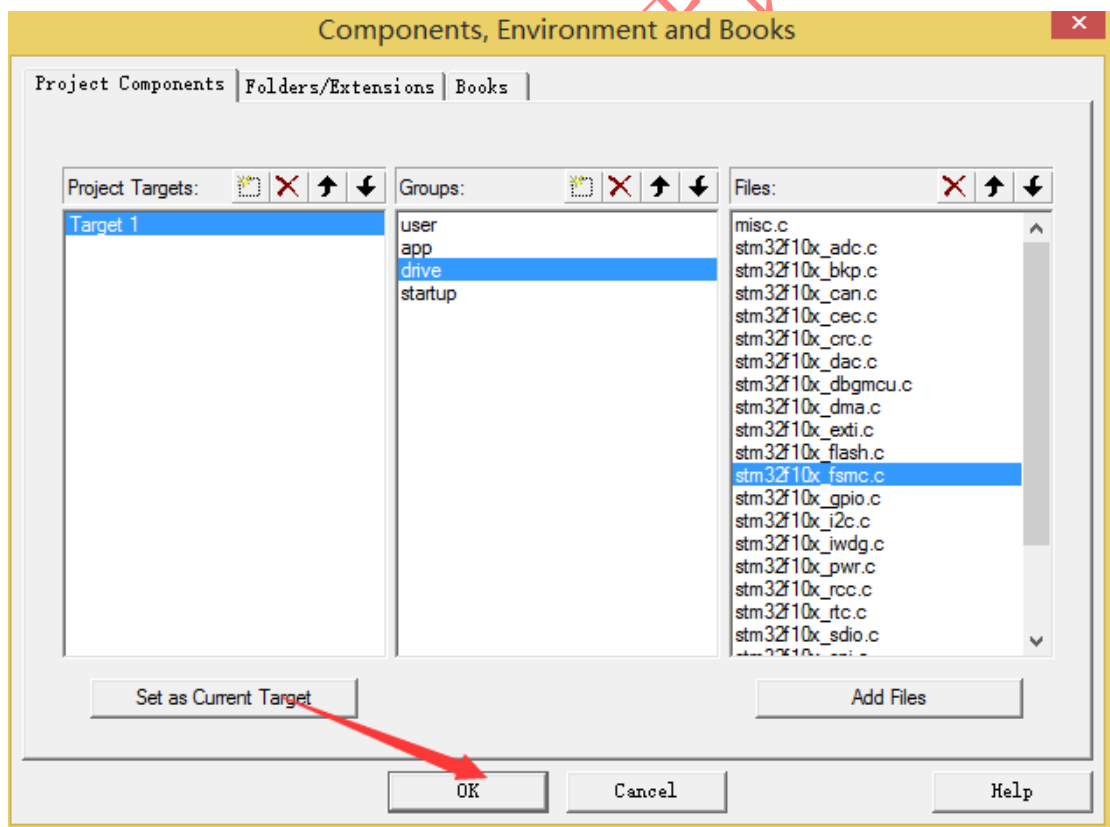
App 里面因为暂时没有文件可不添加，drive 里面添加驱动文件，文件目录是



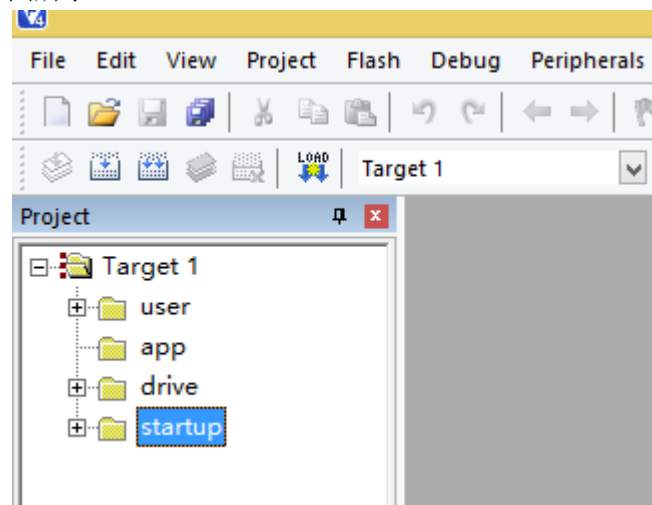
Libraries\STM32F10x_StdPeriph_Driver\src，将里面的 C 文件添加进来，可以根据需要添加。Startup 选项里面添加启动文件，文件路径为 Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm，启动文件根据具体 stm32 芯片进行选择。



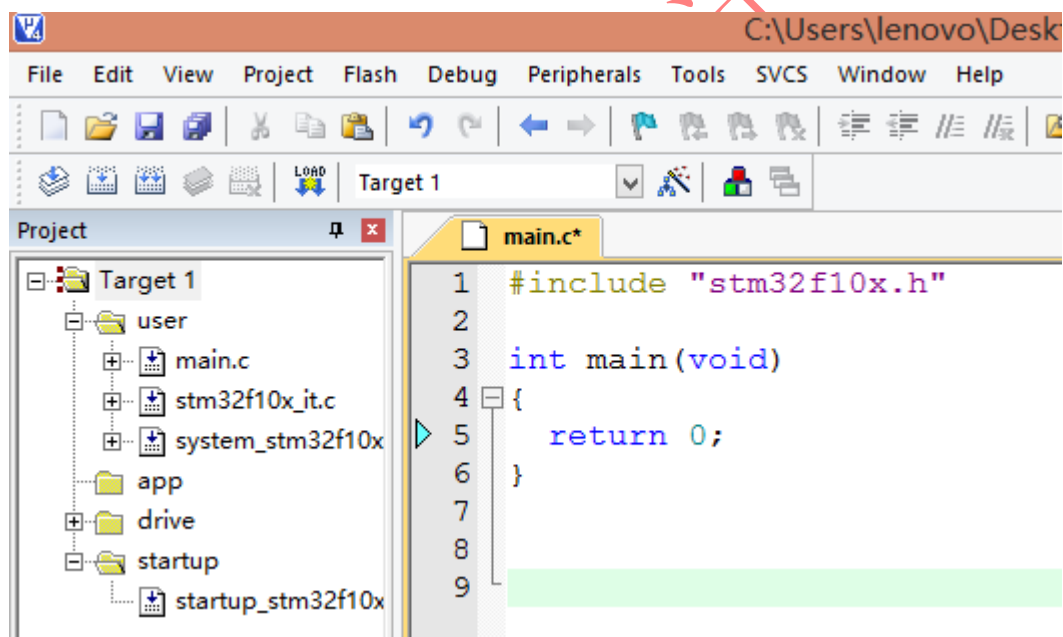
然后点击 OK 即可。



此时文件目录如下图所示

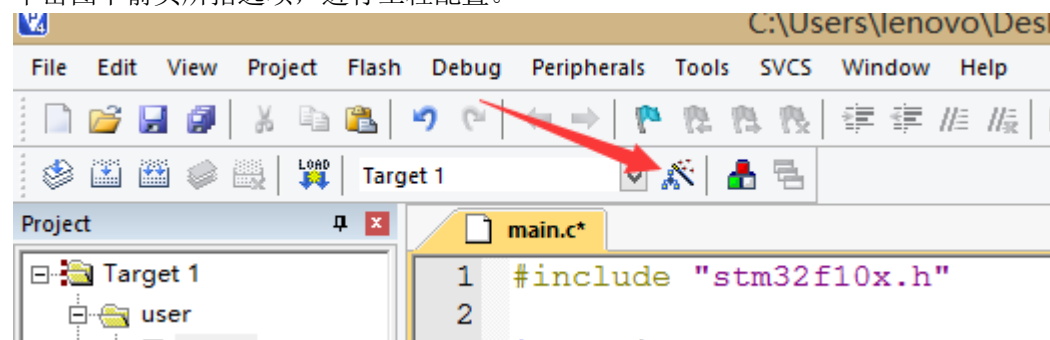


打开 main 文件，将里面内容删除，然后写一个 main 函数即可。至此工程就建立好了，接下来就是对工程进行简单配置了。

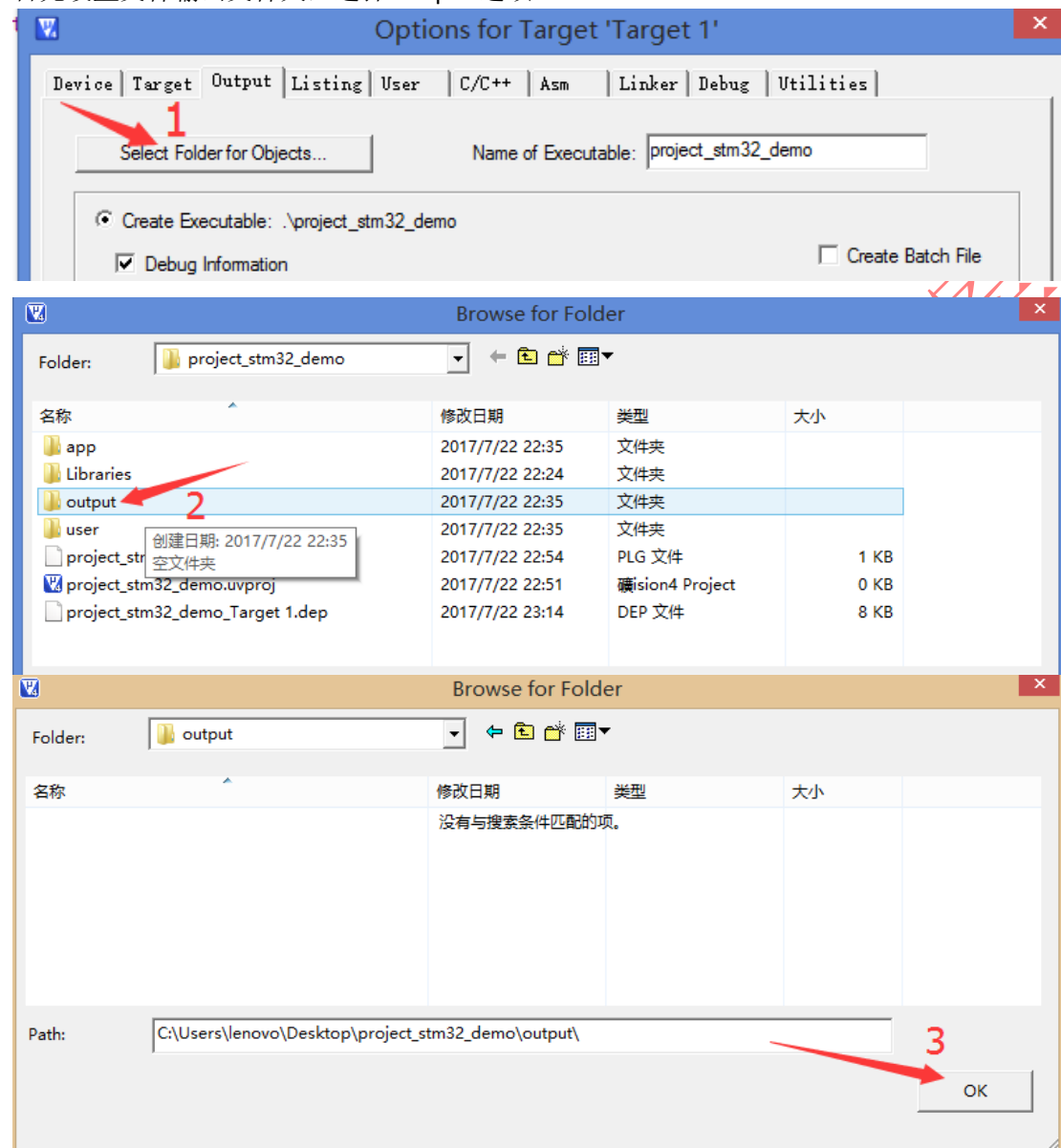


第四步：

单击图中箭头所指选项，进行工程配置。



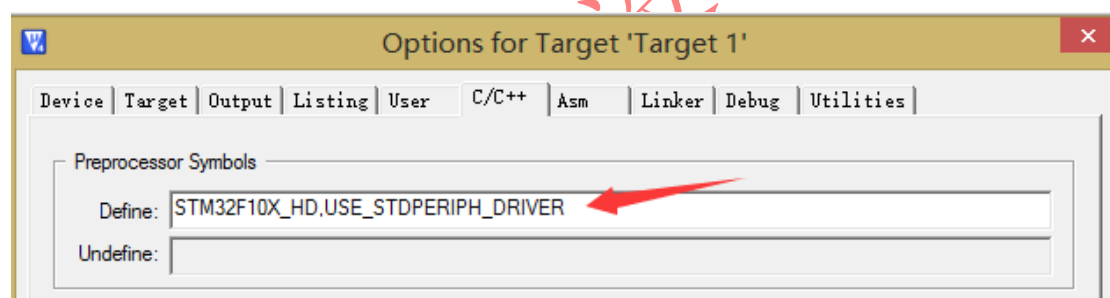
首先设置文件输出文件夹，选择 Output 选项



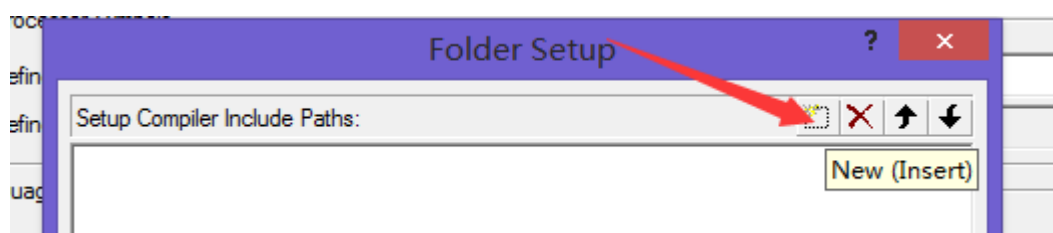
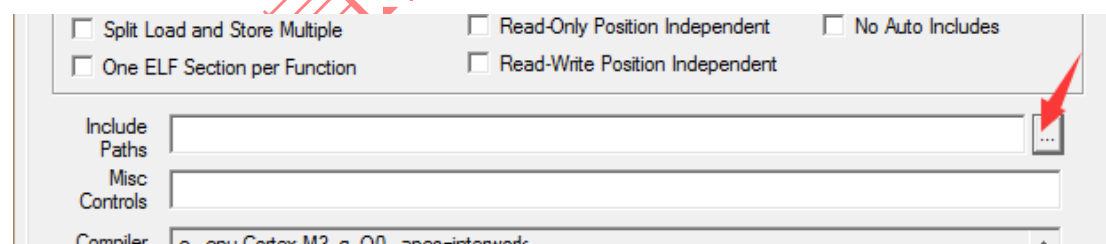
设置 Listing 选项，其实就是将编译过程中的文件都放在 output 文件夹里。



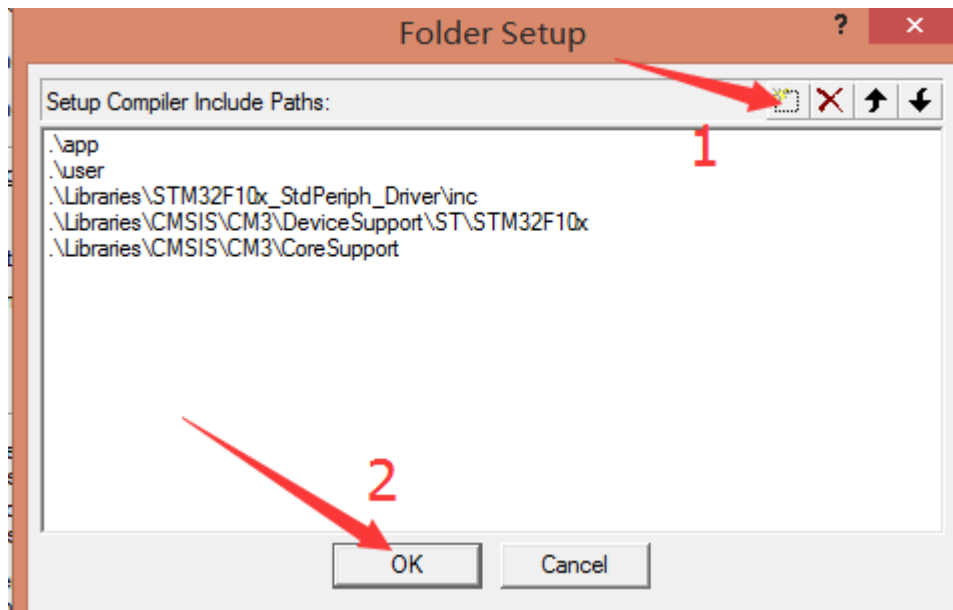
其中 C/C++选项里配置最为重要，首先在 Define 后面添加 STM32F10X_HD,USE_STDPERIPH_DRIVER 宏，如果不加编译会有错误发生。



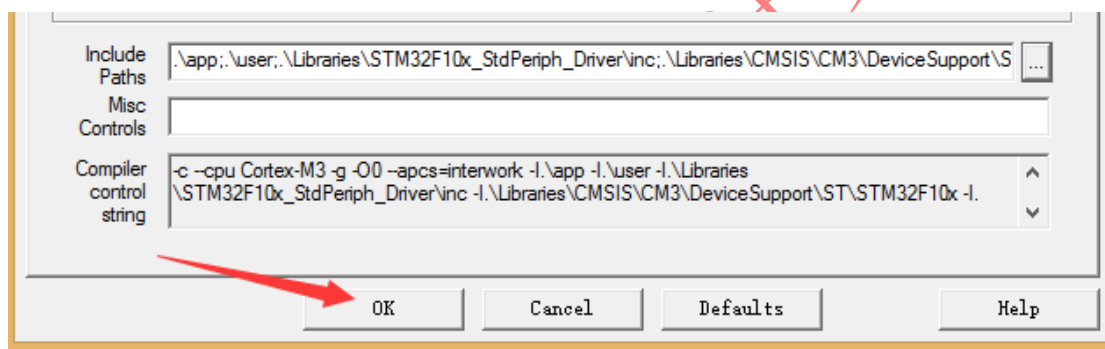
然后在 Include Paths 里添加工程中所有头文件路径。



本工程中头文件包含路径主要如下：



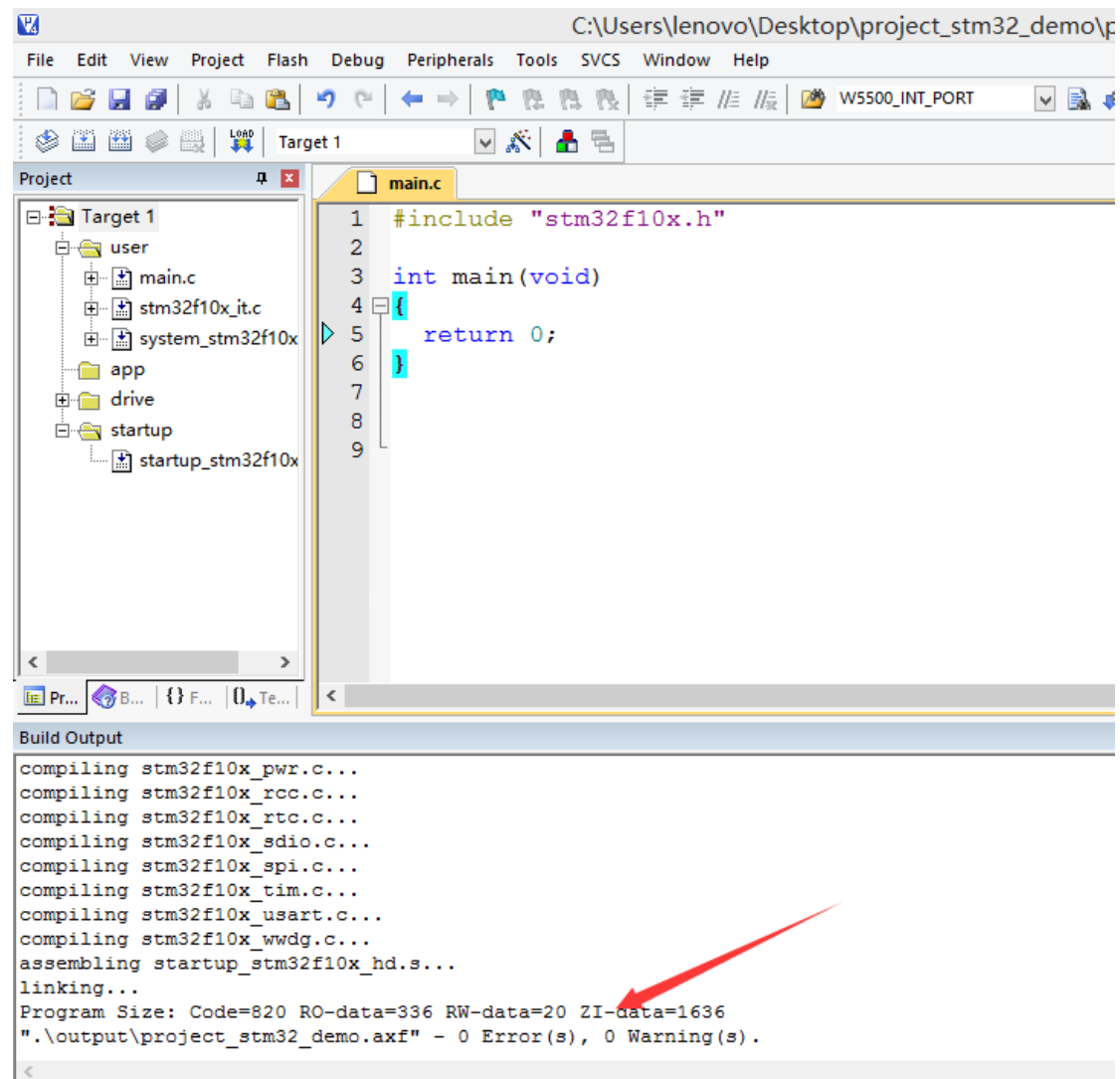
配置结束后点击 OK 即可完成基本配置



至此我们就可以进行编译了，点击编译按钮



工程编译结束，没有警告和错误。



常见问题:

- ① 工程创建时启动文件选择不正确，启动文件选择是根据你的目标板子来确定的，取决于芯片
- ② 在配置工程的时候没有加 STM32F10X_HD, USE_STDPERIPH_DRIVER 宏，关于这个宏也可以不加，如果不加的时候需要改动固件库代码也可以避免报错。
- ③ 头文件包含不正确，在添加的时候需要将.h 存在的目录添加进来，否则工程定位不到头文件位置。