

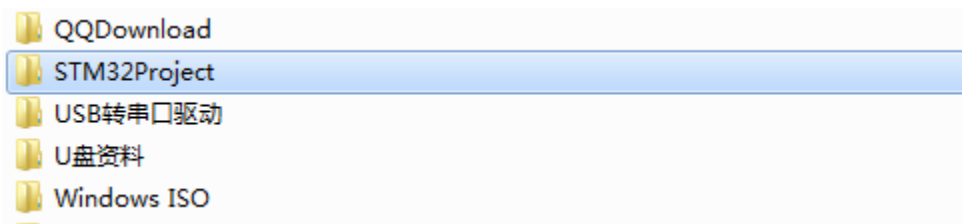
# Keil uVision5 中建 Stm32 工程教程

本教程中，介绍在 Keil uVision5 中建 STM32 工程的两种方法：一种是利用固件库建立工程，另一种是通过建工程时软件给的引导直接完成。

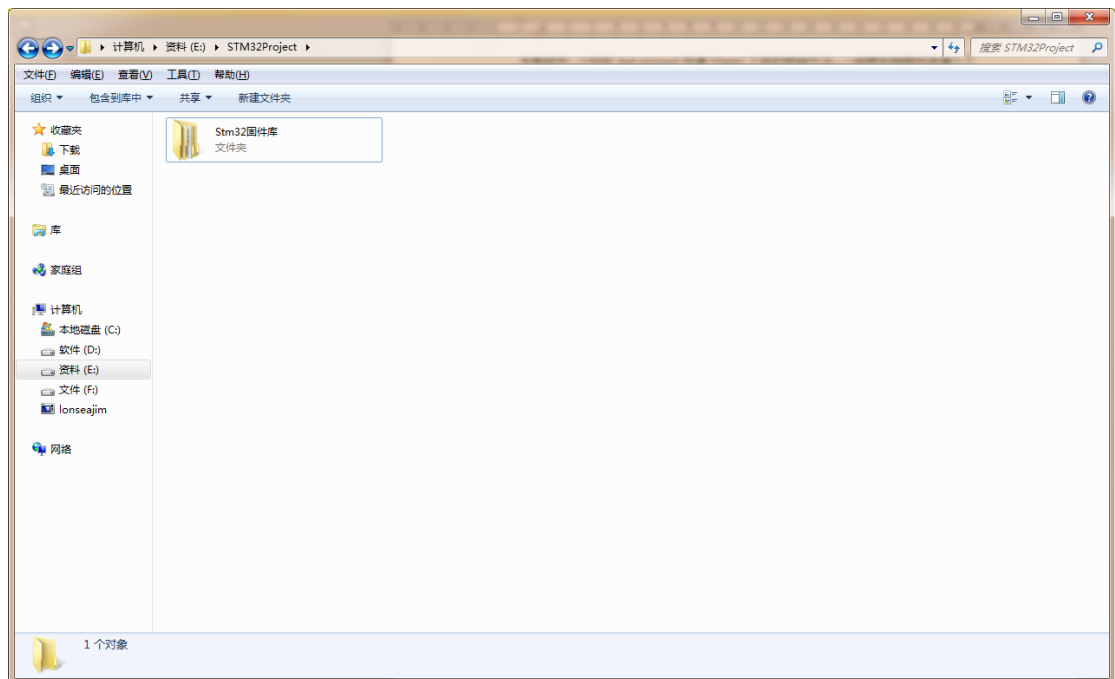
软件的安装和破解网上教程很多，这里就不赘述了，直接从建立工程开始。

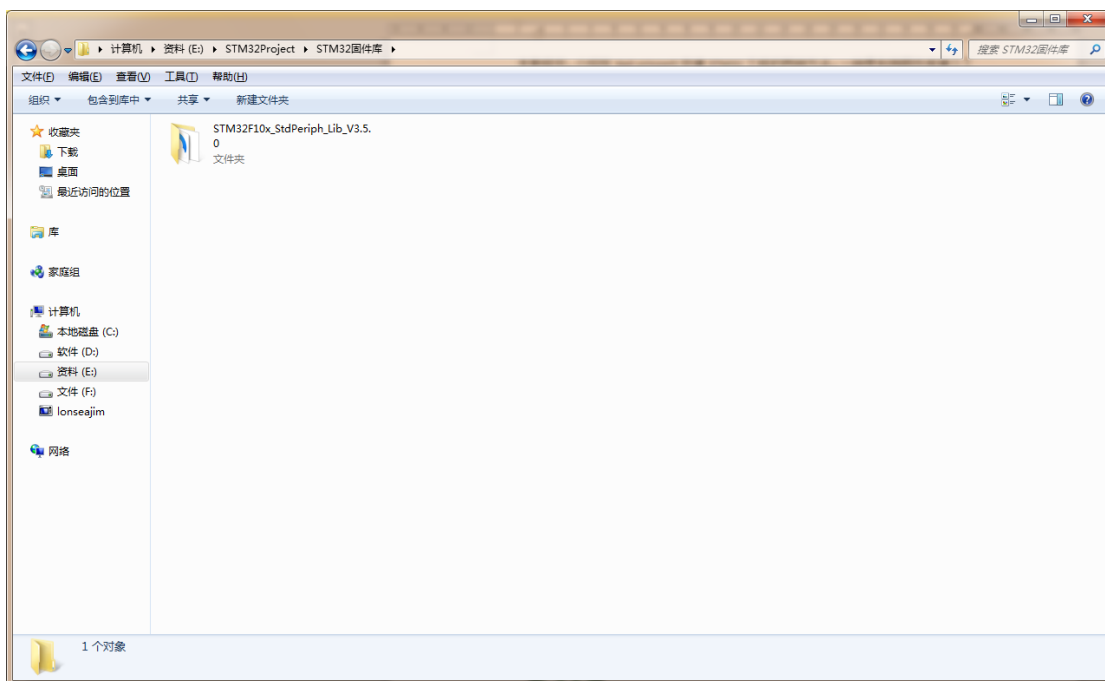
首先介绍第一种：利用固件库建立 STM32 工程

1. 在你电脑中想要建立工程的地方先新建一个文件夹。这里我在 E 盘下新建了一个文件夹，命名为 STM32Project。

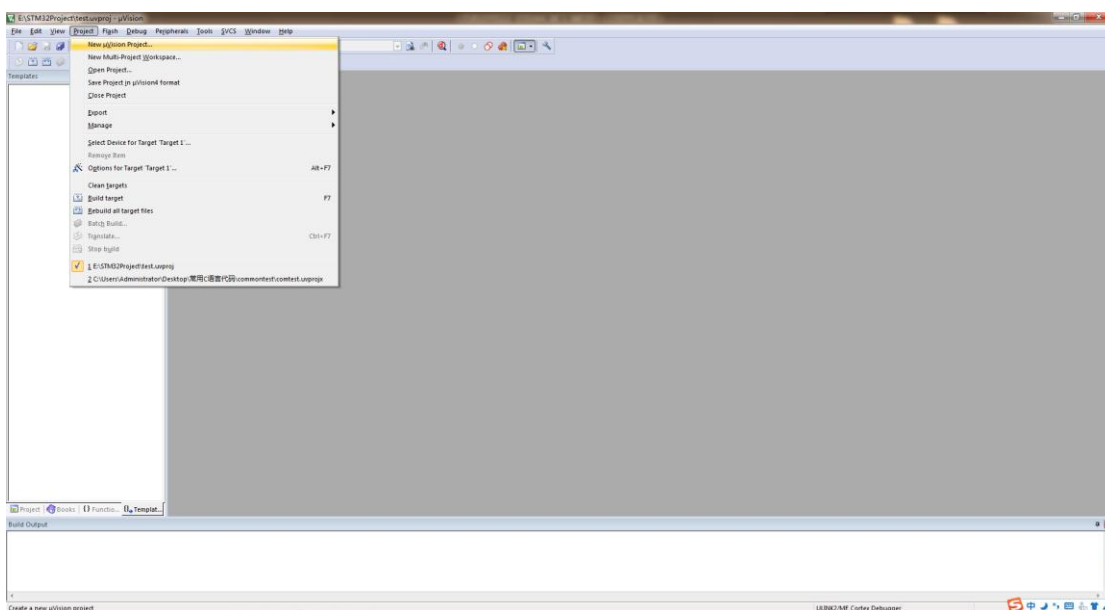


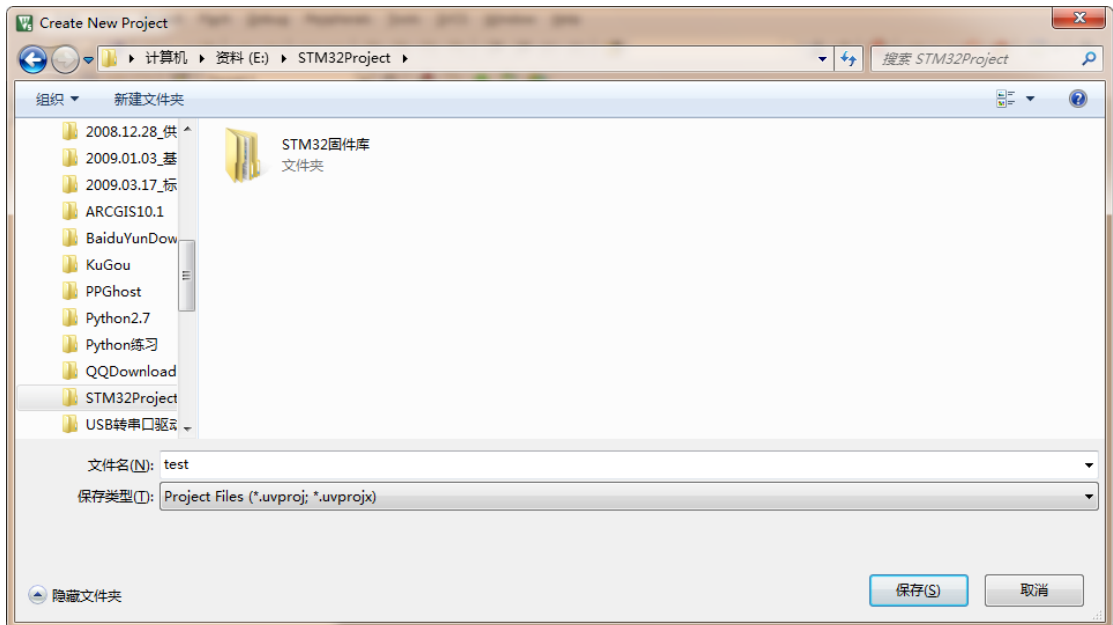
2. 在新建的 STM32Project 文件夹下再新建一个文件夹，命名为 STM32 固件库，然后把事先下载的固件库放在文件夹中，方便查找进行复制。



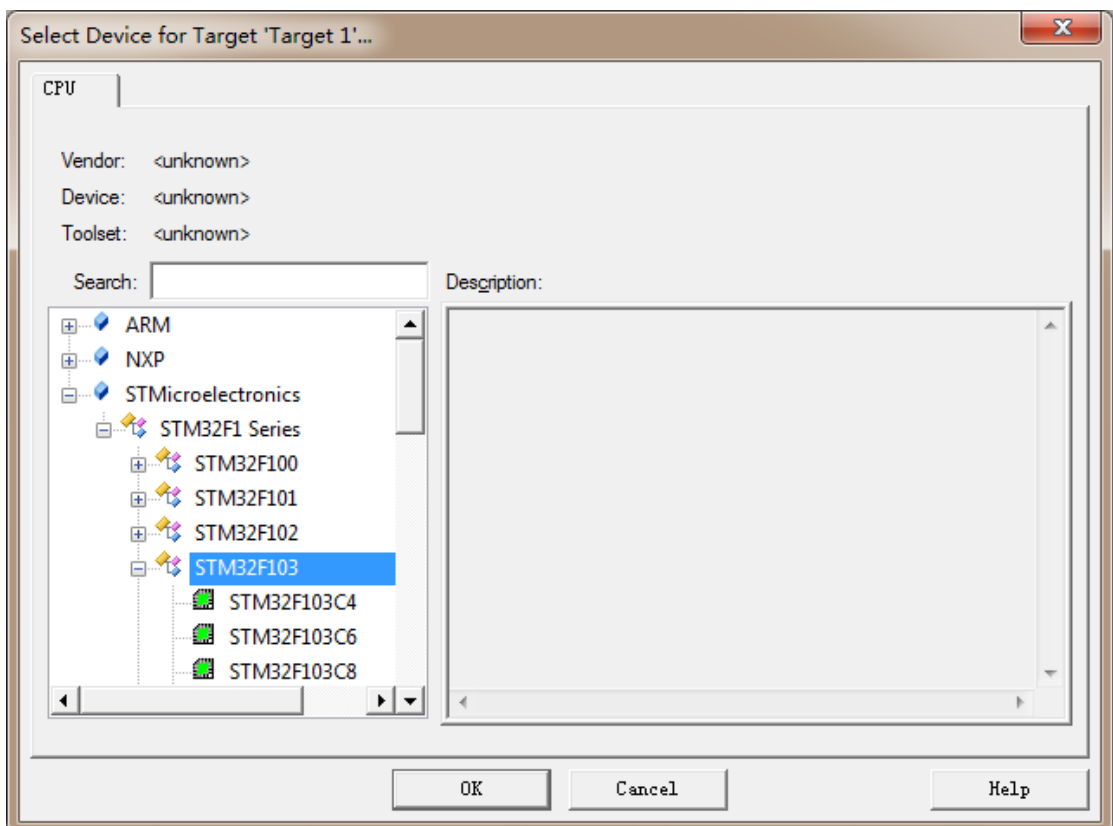


3. 打开 Keil uVision5 软件，在软件的菜单栏中找到 Project——>New uVision Project..., 单击，找到上述新建的文件夹，对新建的工程进行命名，这里命名为 test，点击保存。

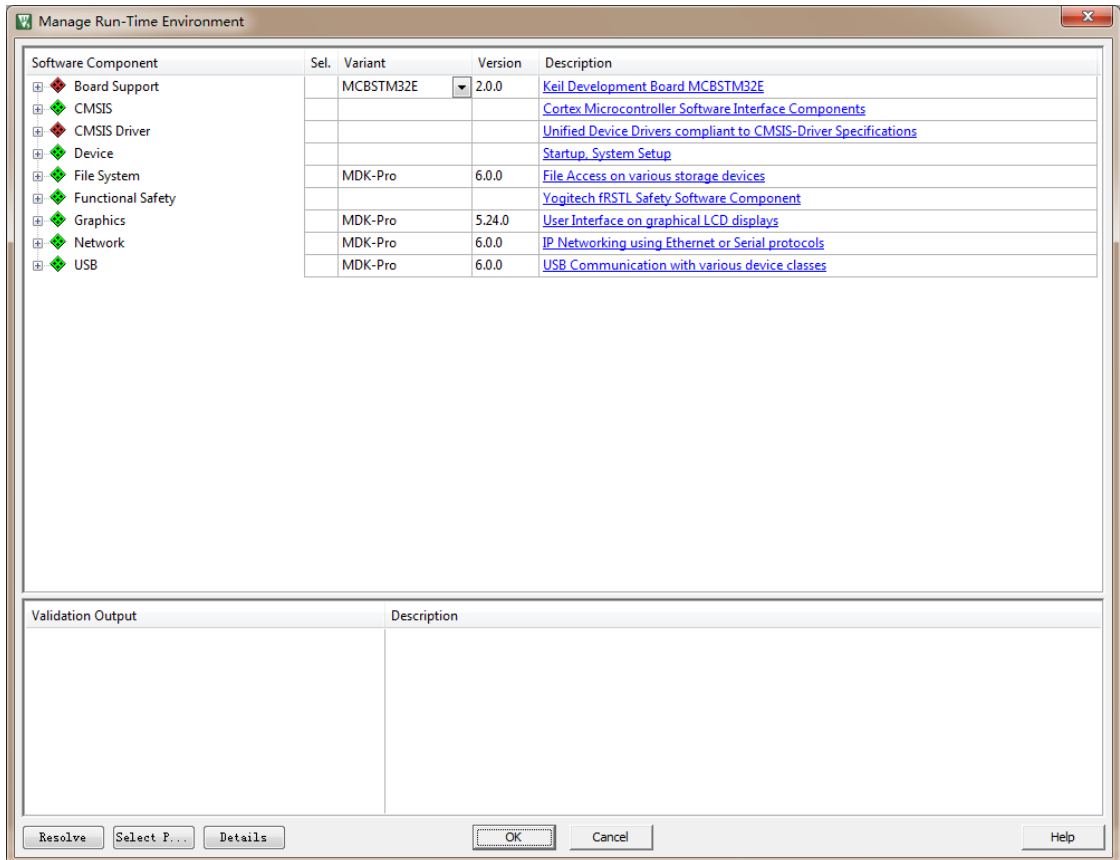




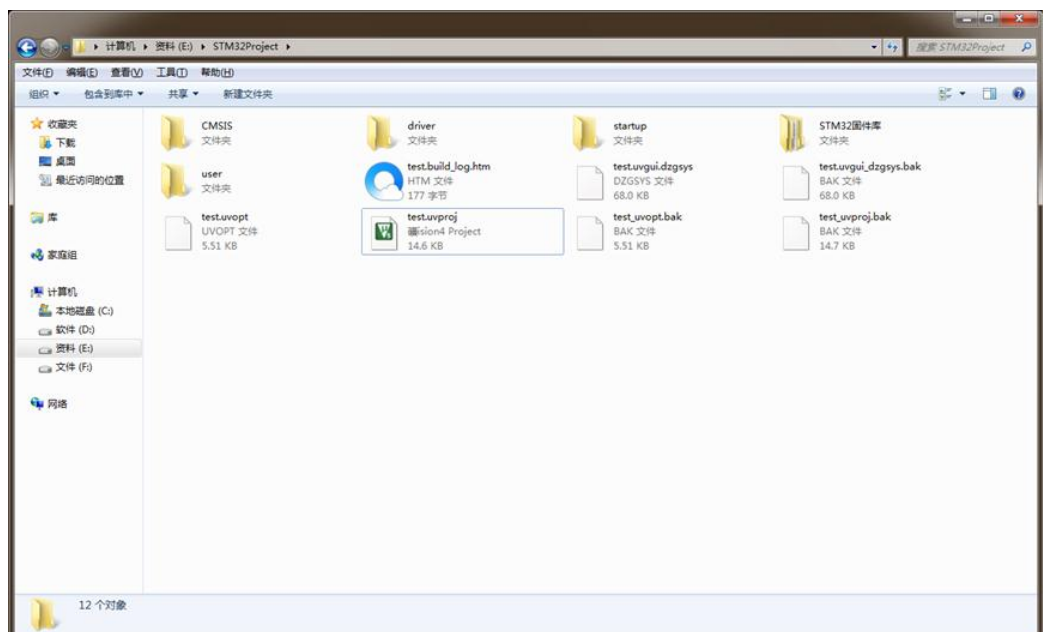
4. 此时软件弹出一个选择芯片的窗口：“Select Device for Target ‘Target 1’...”，在左侧的芯片库中找到你要使用的 STM32 系列的芯片，这里选择的是 STMicroelectronics — — >STM32F1 Series — — >STM32F103 — — >STM32F103RC，单击 OK。



5. 此时弹出一个窗口：“Manage Run-Time Environment”，这个窗口在第二个建工程的方法中会使用到，这里先直接点击 OK，跳过。

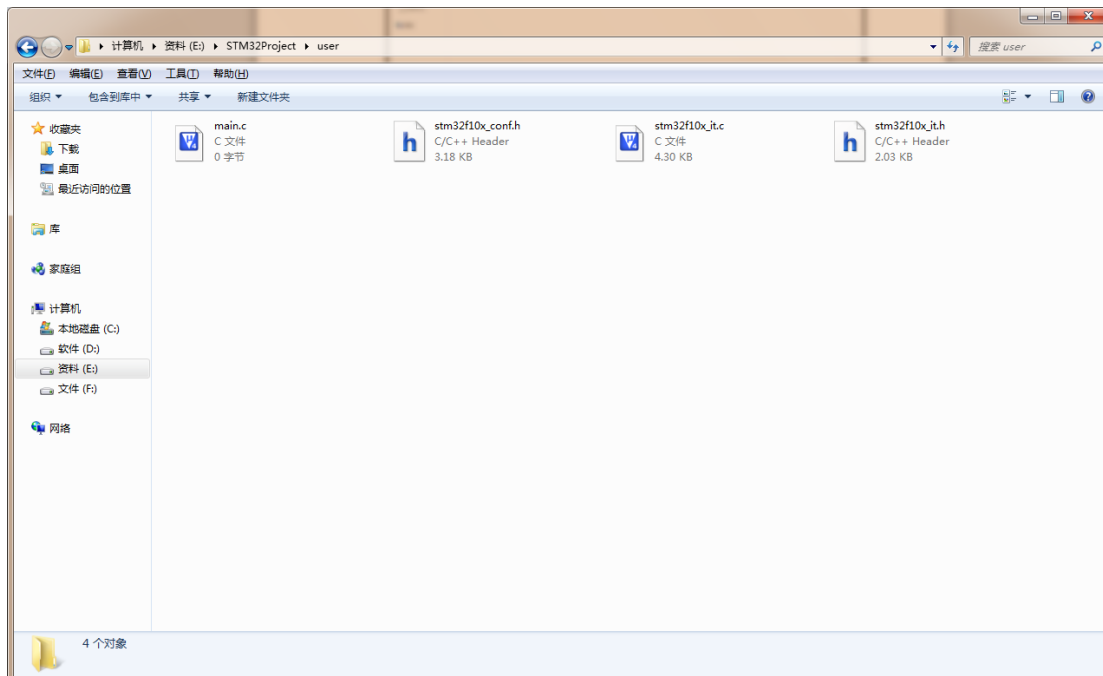


6. 现在开始添加相关文件。
- 在 STM32Project 文件夹中新建以下四个文件夹：
- user：放置用户自己编写的相关文件
  - CMSIS：放置与内核相关的文件
  - startup：放置启动单片机的汇编文件
  - driver：放置外设操作的驱动文件

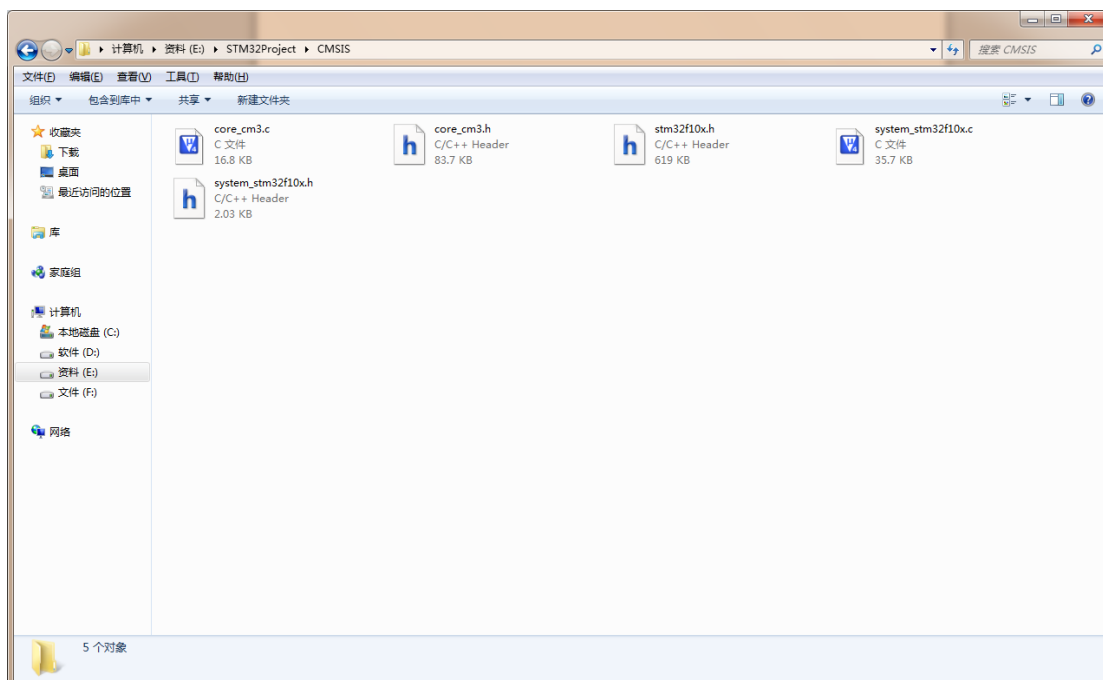


再向这四个文件夹中添加固件库中的相关文件：

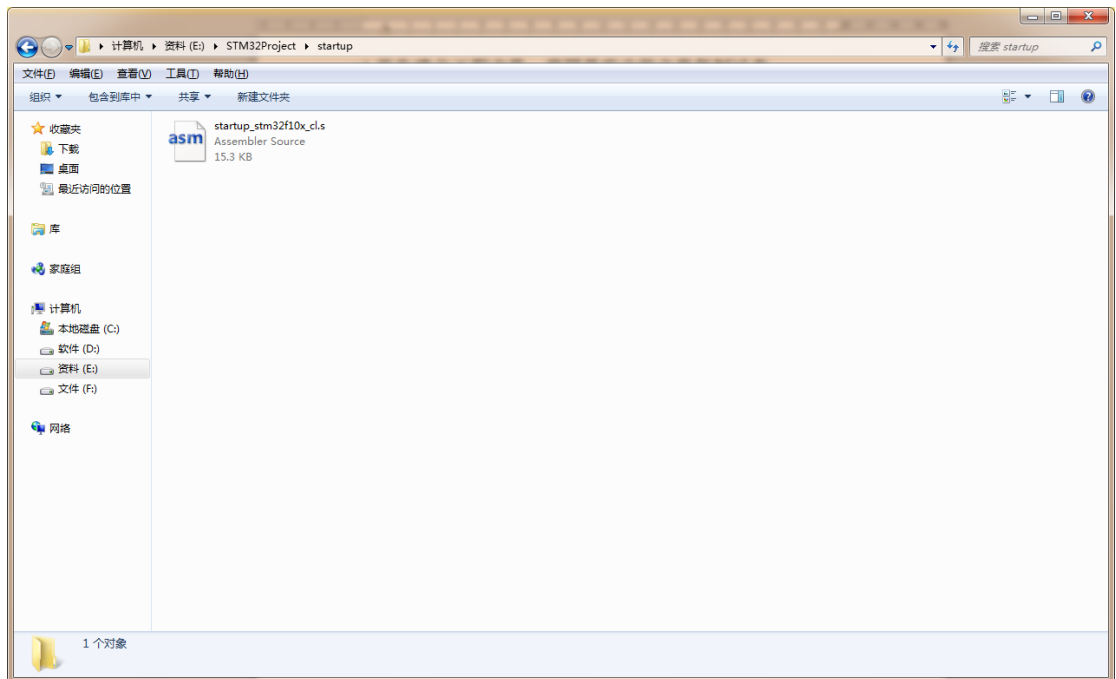
user：STM32Project\STM32 固件库\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Template 中的 stm32f10x\_conf.h、stm32f10x\_it.c、stm32f10x\_it.h 一共 3 个文件复制粘贴到 user 文件夹中，同时新建一个文本文档，重命名为 main.c。



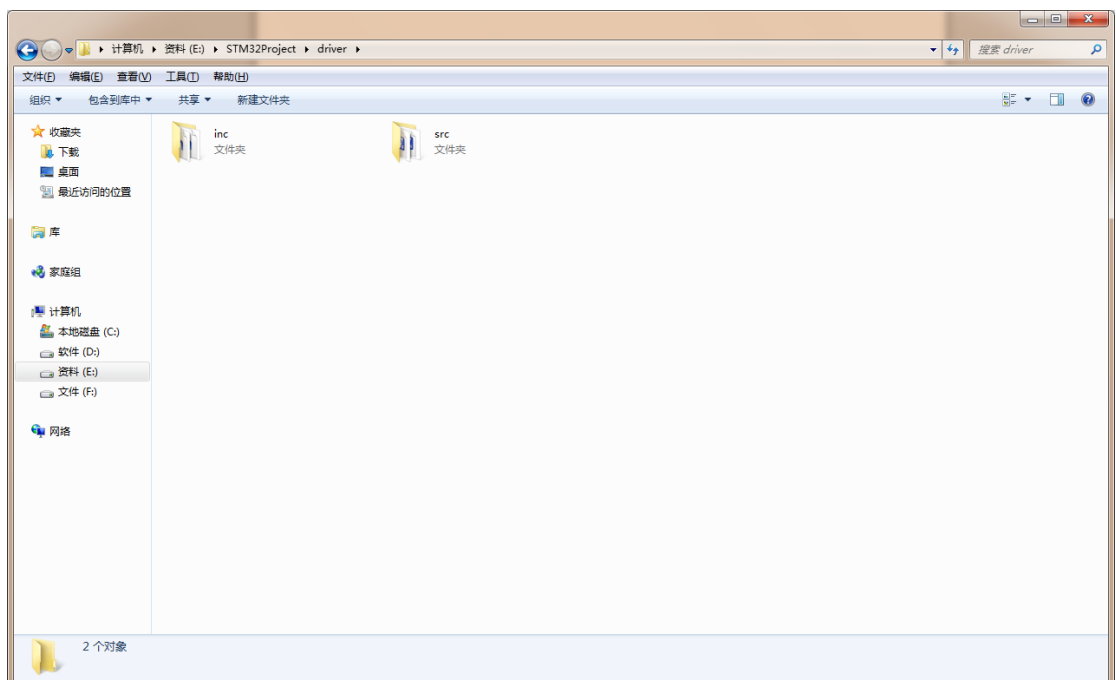
CMSIS：STM32Project\STM32 固件库\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\CoreSupport 中的 core\_cm3.c、core\_cm3.h 文件，以及 STM32Project\STM32 固件库\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x 中的 stm32f10x.h、system\_stm32f10x.c、system\_stm32f10x.h 一共 5 个文件复制粘贴到 CMSIS 文件夹中。



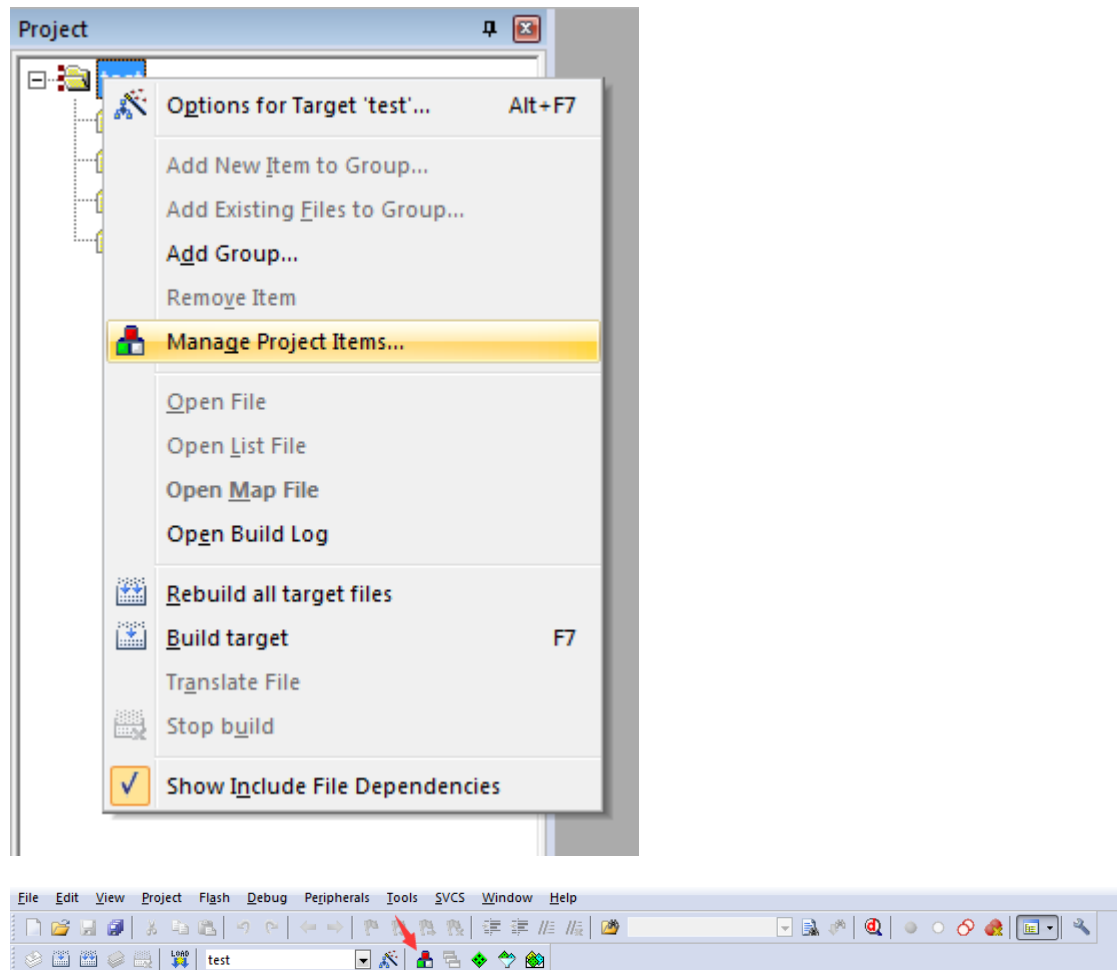
startup：STM32Project\STM32 固件库\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\_DeviceSupport\ST\STM32F10x\startup\arm 中的 startup\_stm32f10x\_cl.s（互联型启动文件）文件复制粘贴到 startup 文件夹中。



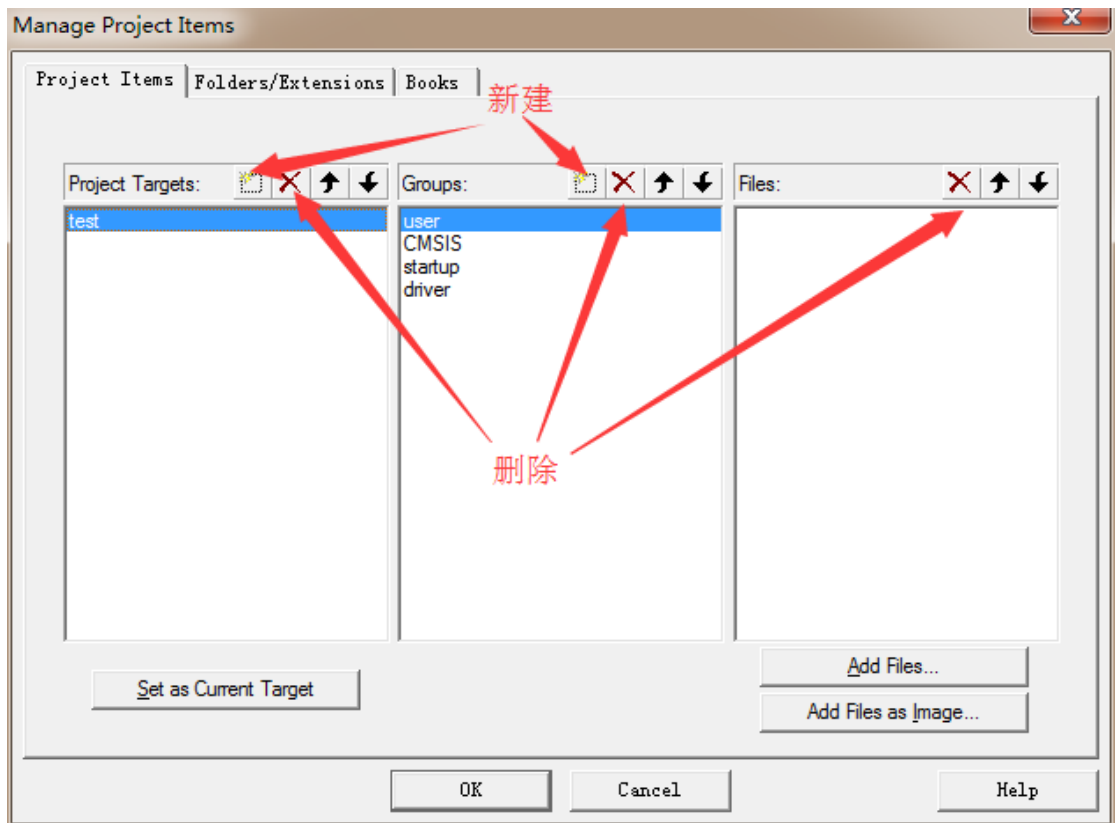
driver：STM32Project\STM32 固件库\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\STM32F10x\_StdPeriph\_Driver 中的 src 和 inc 两个文件夹全部复制粘贴到 driver 文件夹中。



7. 文件复制结束后，现在回到 Keil 软件界面，对新建的工程进行管理。  
对新建的工程右键出现菜单栏，选择 **Manage Project Items**，或者在软件上方的工具栏中找到有个彩色品字形图标，单击。

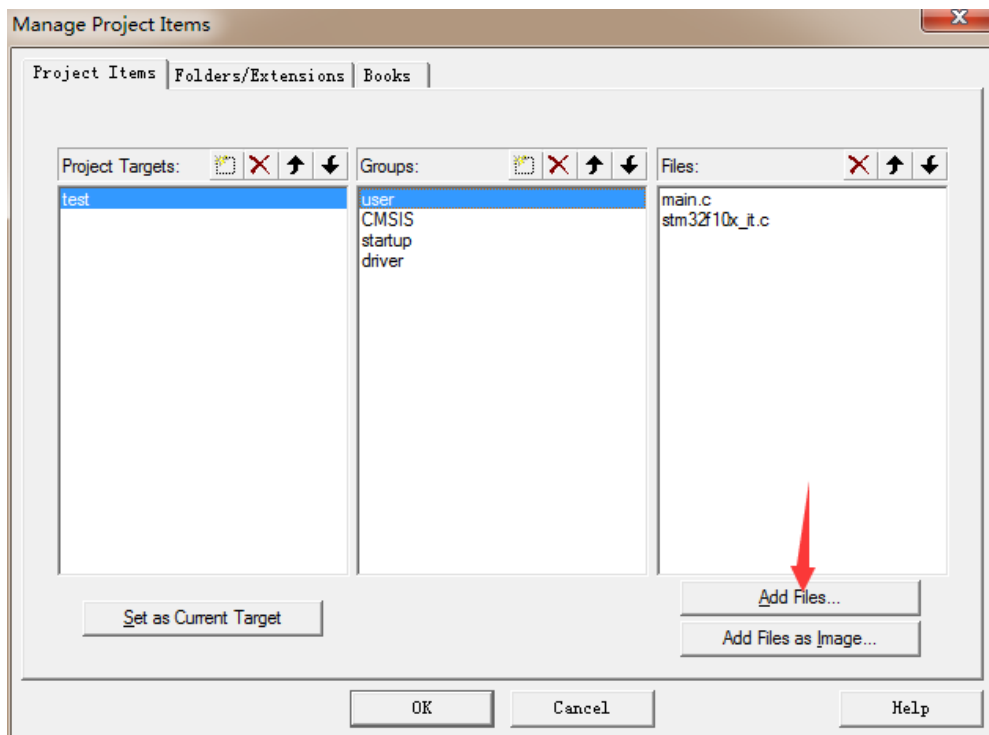


8. 进行完第 7 步操作后进入到 **Manage Project Items** 界面。在这个界面里，我们可以对该工程进行管理，通过 **Project Target** 的删除和新建，可以对该工程进行重命名，这里命名为 **test**。删除 **Groups** 中原有的一个文件，新建第 6 步操作中在工程文件夹中添加的对应的 4 个文件名，分别为 **user**、**CMSIS**、**startup** 和 **driver**。

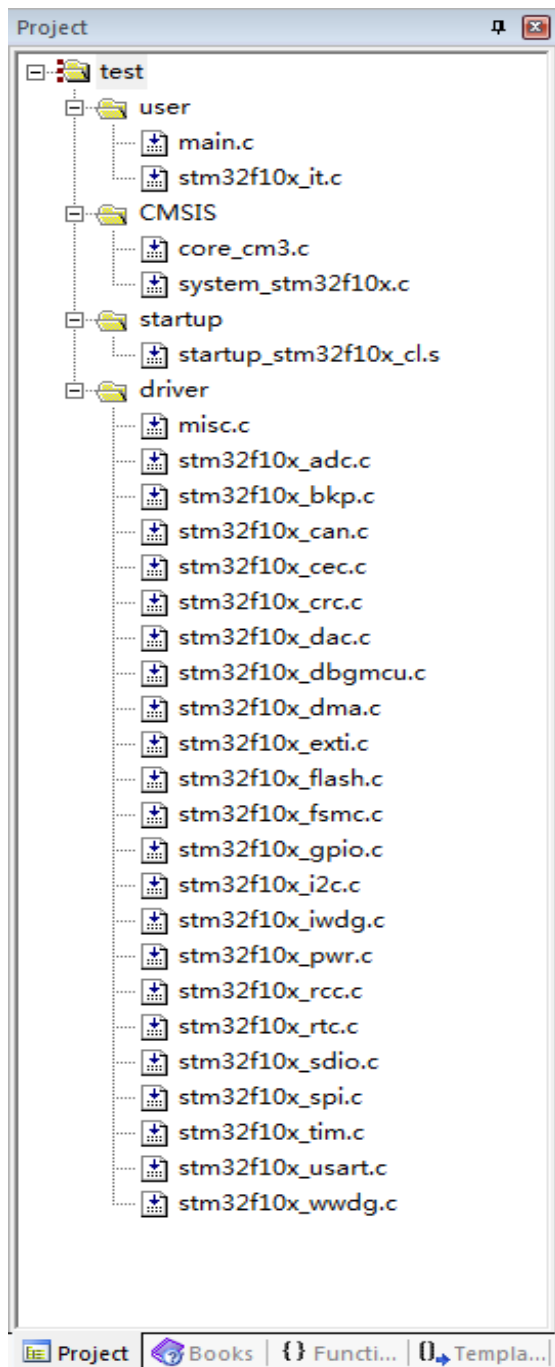


9. 分别对上述的 user、CMSIS、startup 和 driver 4 个文件进行 Add Files 操作。所要添加的文件分别为第 6 步操作中对应文件夹中的相关对应文件，点击 OK。

注意：添加文件时只添加.c 后缀的文件，.h 后缀的文件不需要添加，只有 startup 中添加对应的.s 后缀文件，driver 中添加的文件在对应文件夹 driver\src 中。



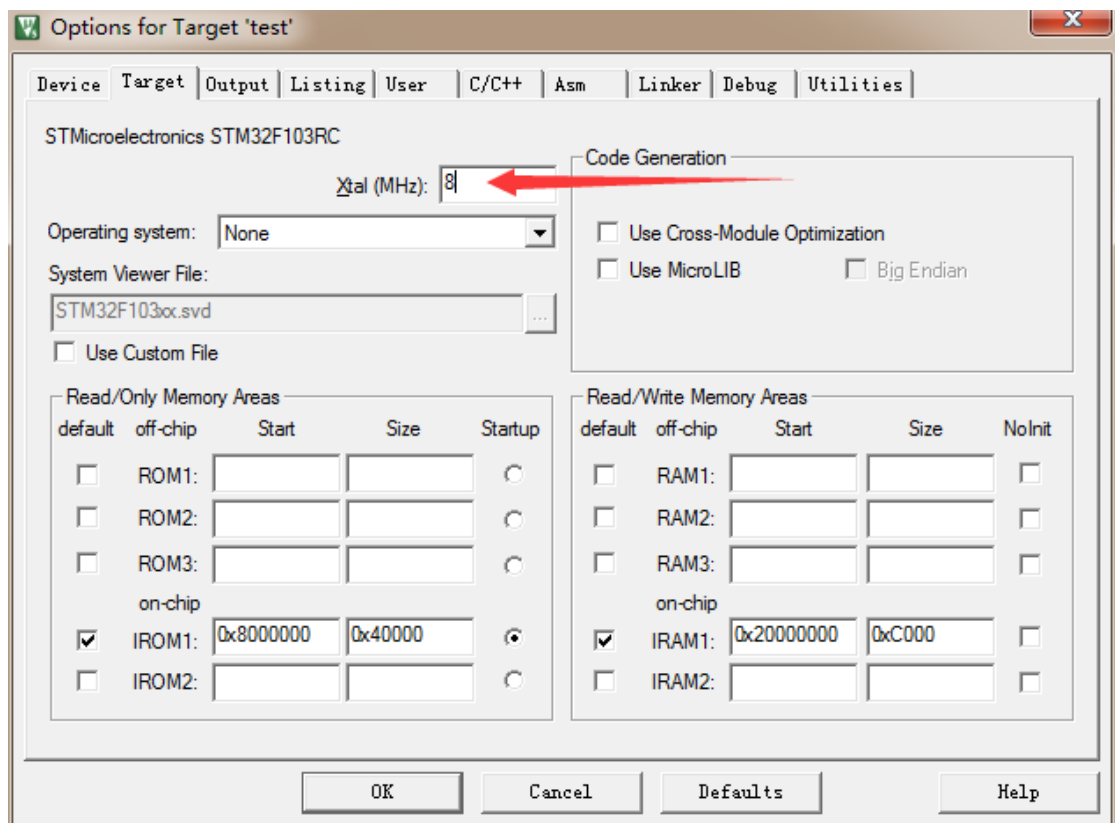




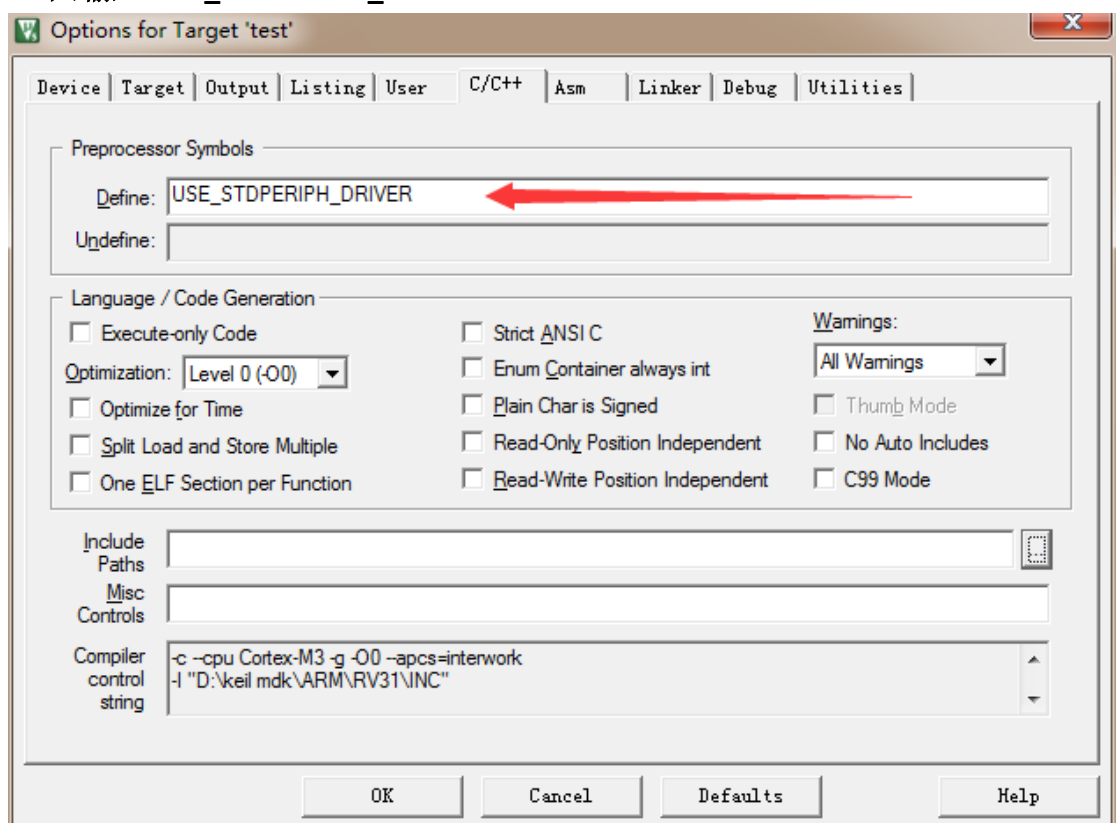
10. 配置编译相关的设置选项。

- (1) 单击魔法棒形图标，出现 Options for Target 'test' 界面，先将 Target 标签下的晶振频率设置成和你电路板上一样的值，这里是 8MHz。

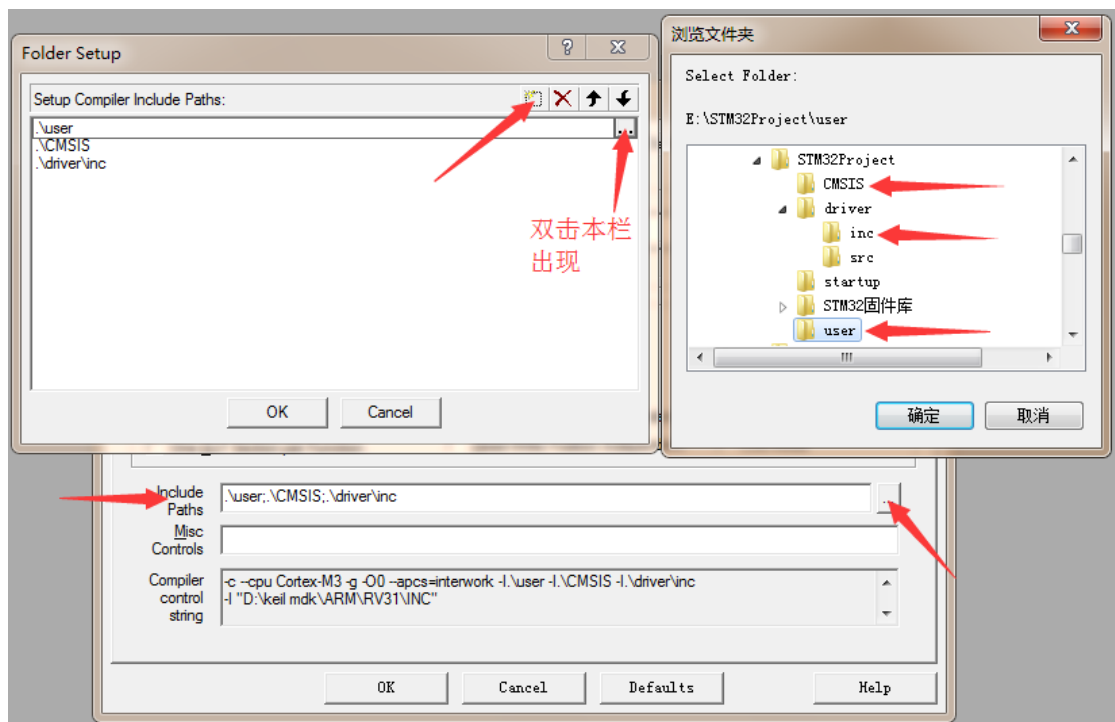




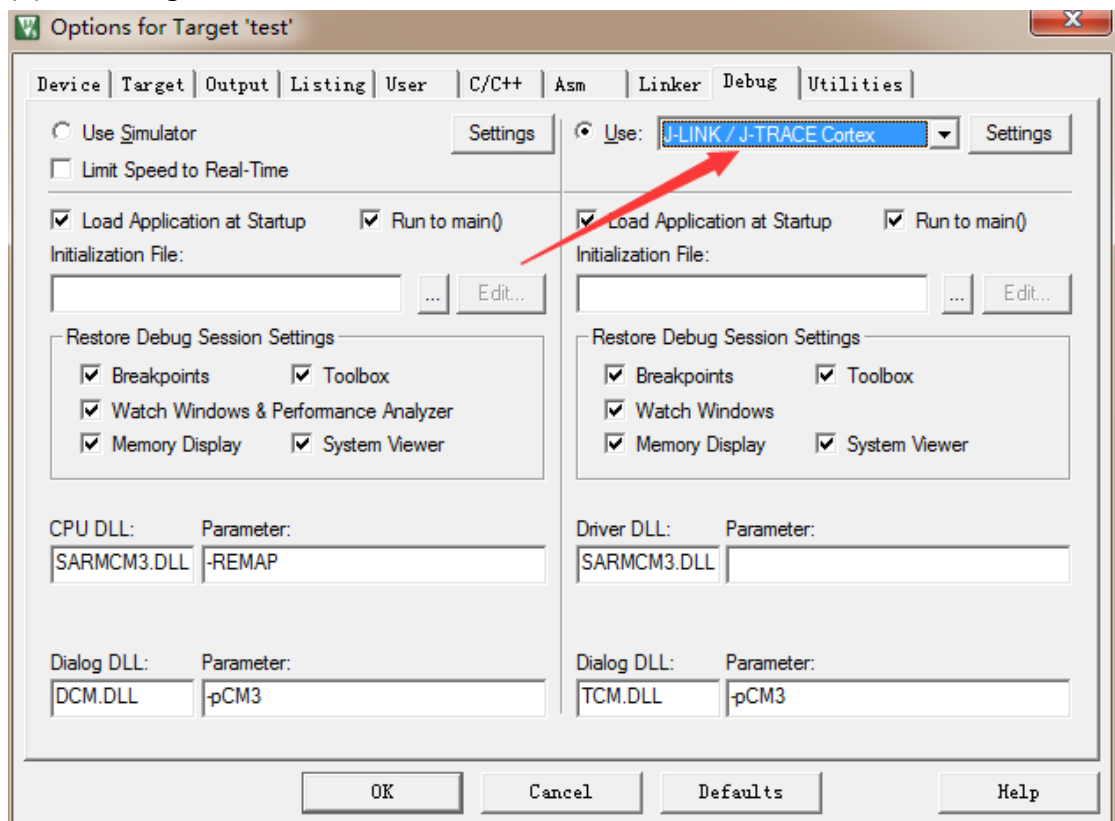
- (2) 选择 C/C++ 标签，在 Define 中输入：USE\_STDPERIPH\_DRIVER  
 注意：有的教程中在 USE\_STDPERIPH\_DRIVER 的后面还输入了,STM32F10X\_CL，实测在 Keil uVision5 中编译时会出错，因此这里只输入 USE\_STDPERIPH\_DRIVER。



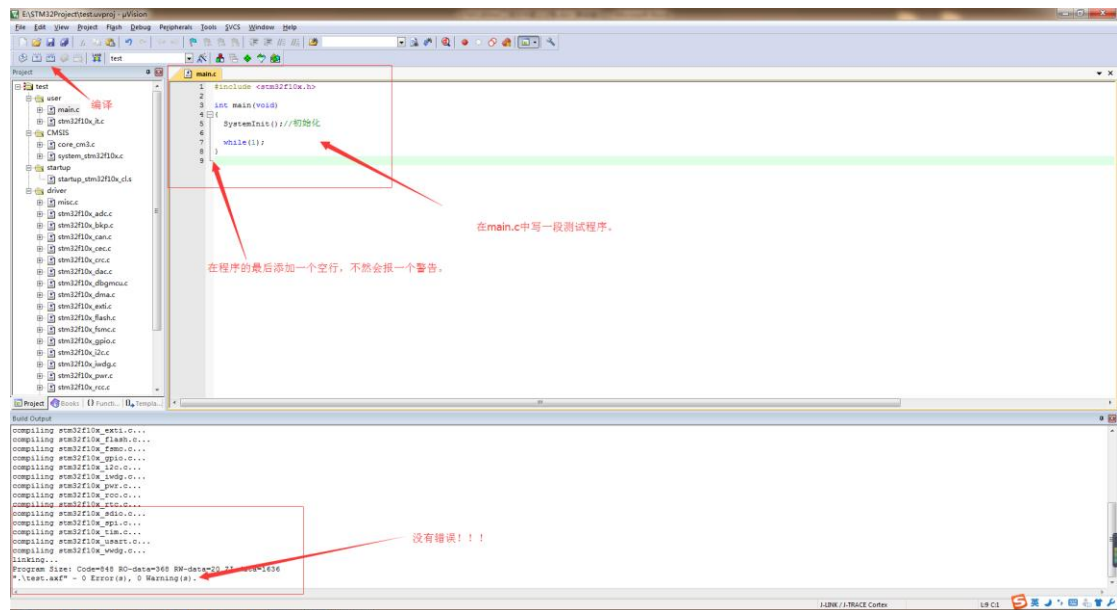
- (3) 继续在 C/C++ 标签的标签下操作。在 Include Paths 中添加所有 .h 文件的根目录。



- (4) 在 Debug 标签中选择 Use: J-LINK/J-TRACE Cortex，点击 OK，完成。



## 11. 工程建完，成果展示。



代码：

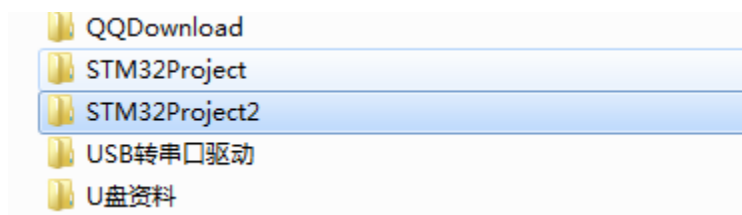
```
#include <stm32f10x.h>

int main(void)
{
    SystemInit();//初始化

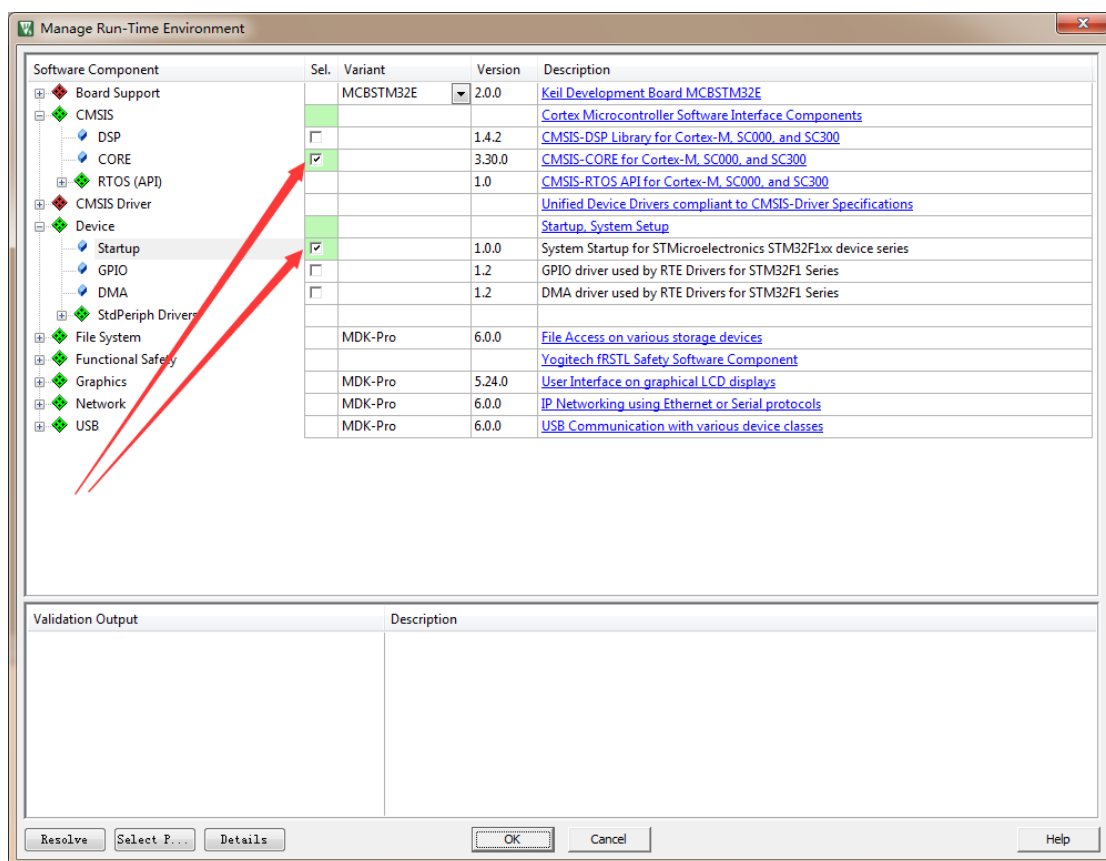
    while(1);
}
```

下面介绍另一种建立工程的方法：通过建工程时软件给的引导直接完成。

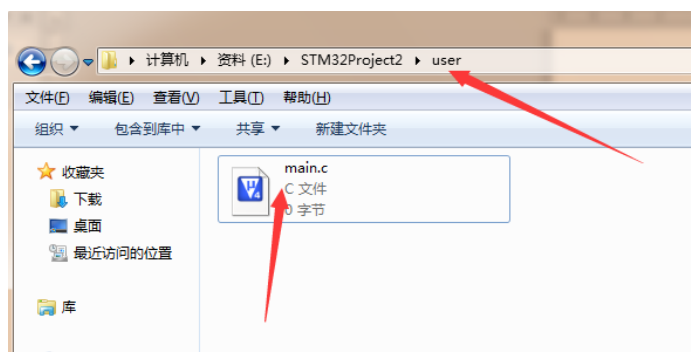
1. 依然是在你电脑中想要建立工程的地方先新建一个文件夹。这里我在 E 盘下新建了一个文件夹，命名为 STM32Project2。



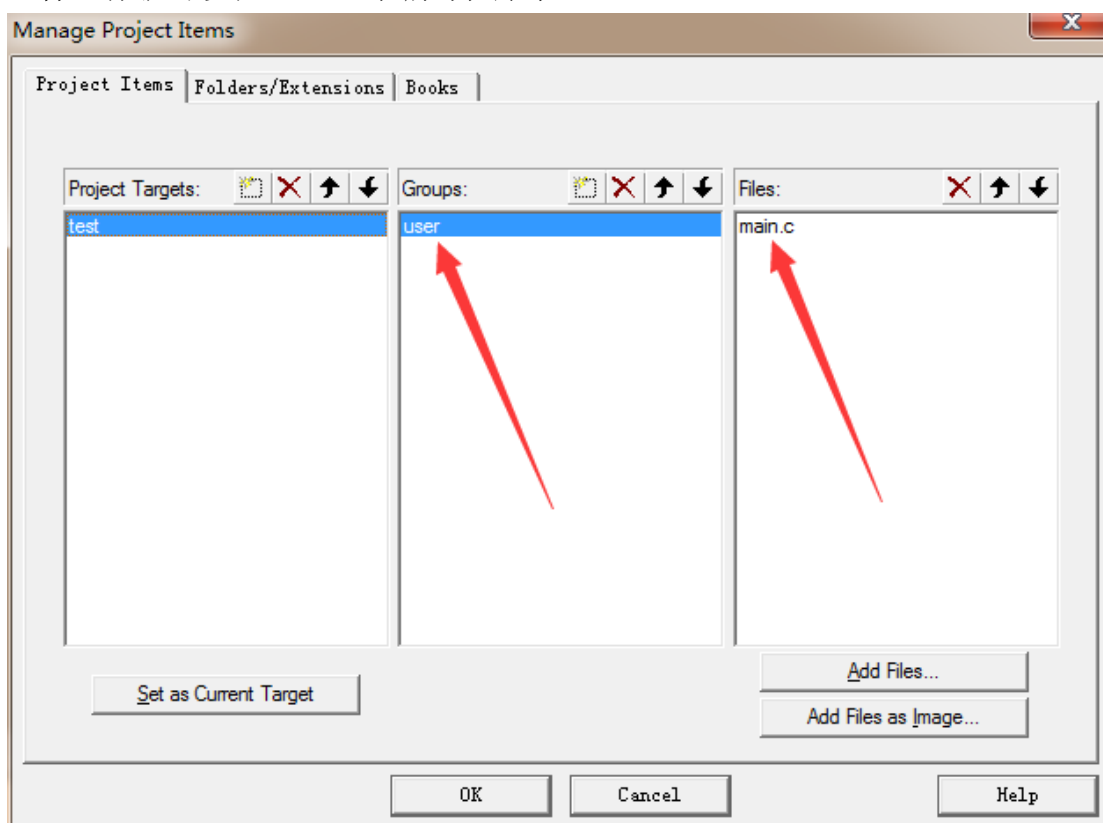
2. 之后 2-4 步和上述方法一样，直接到上述方法的第 5 步：此时弹出一个窗口：“Manage Run-Time Environment”，这里就不能直接跳过了。
3. 在 Manage Run-Time Environment 界面中，勾选两个建工程时最基本的选项，点击 OK。



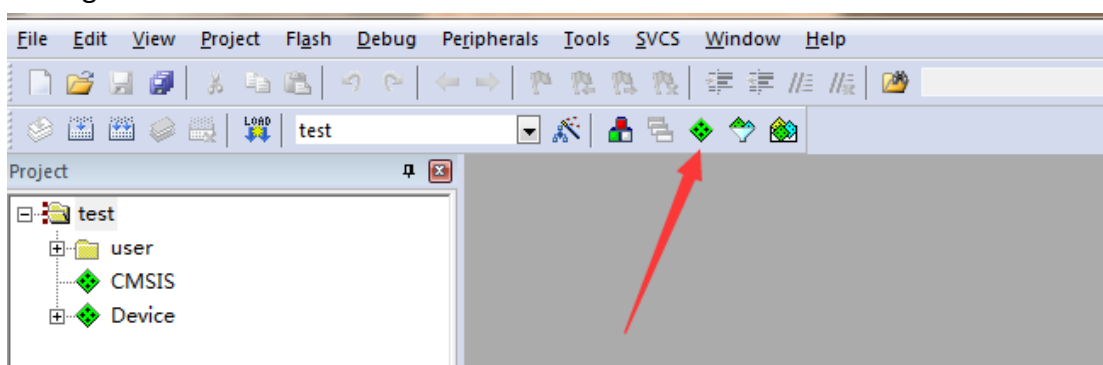
4. 在 STM32Project2 文件夹中新建名为 user 的文件夹，并在 user 文件夹下添加一个文本文档，重命名为 main.c。



5. 类似上述方法中的第 7 和第 8 步, 在工程管理中新建 `user`, 并添加 `main.c`。这样, 你就可以在 `main.c` 中编写程序了。

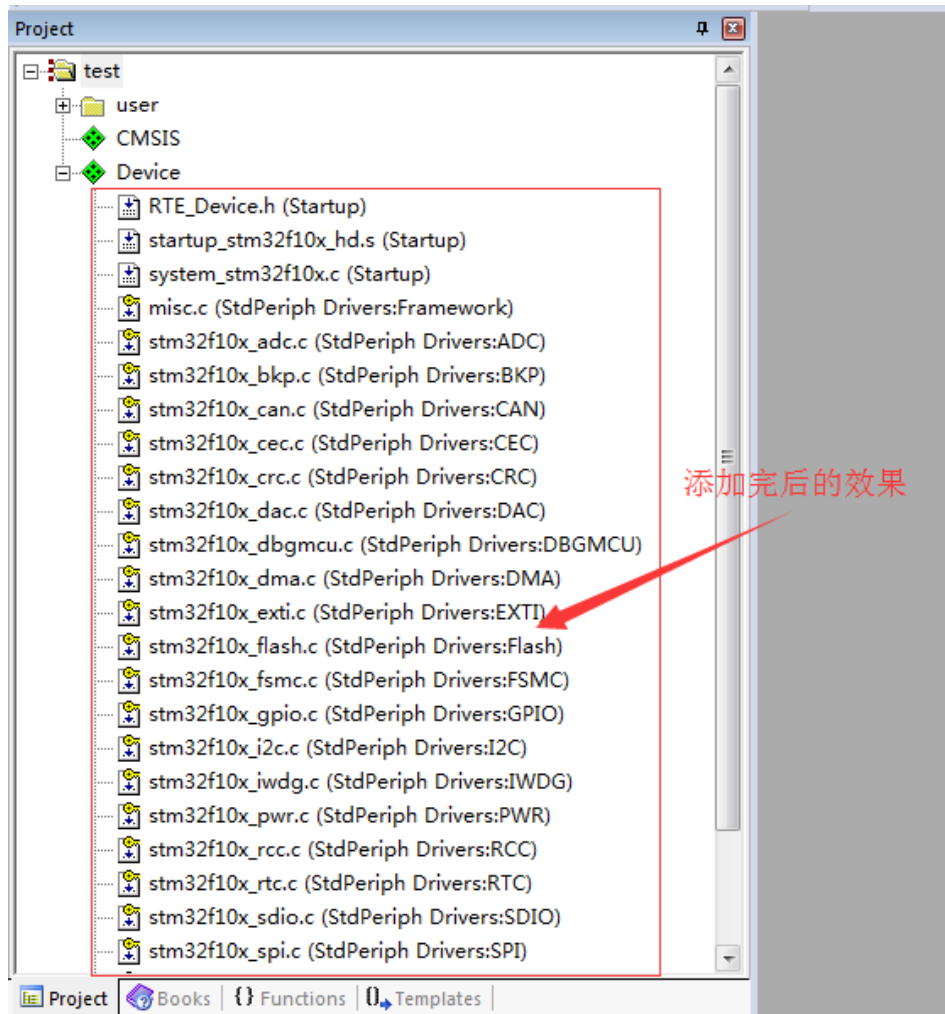
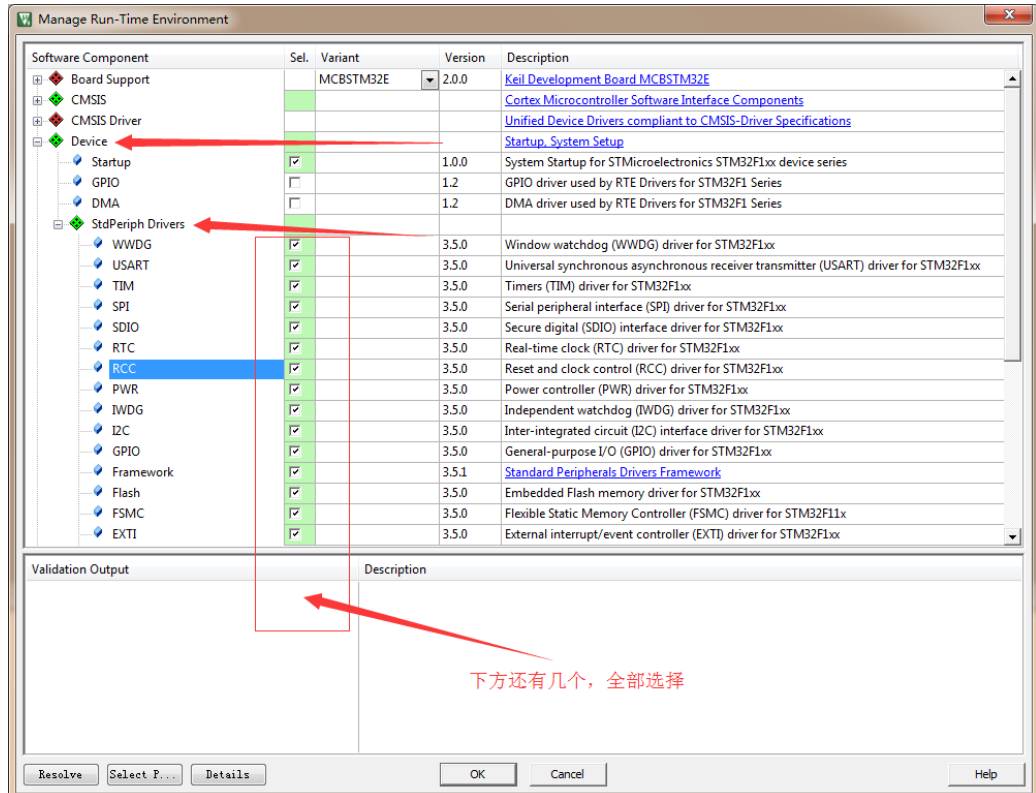


6. 添加外设驱动文件。上述方法中的 `driver` 文件中放置了外设驱动文件, 并通过工程管理添加进工程, 这里无需这么做, 有更简单的方法。单击工具栏中类似二阶魔方的图标, 我们发现又进入到了第 3 步中的 **Manage Run-Time Environment** 界面。

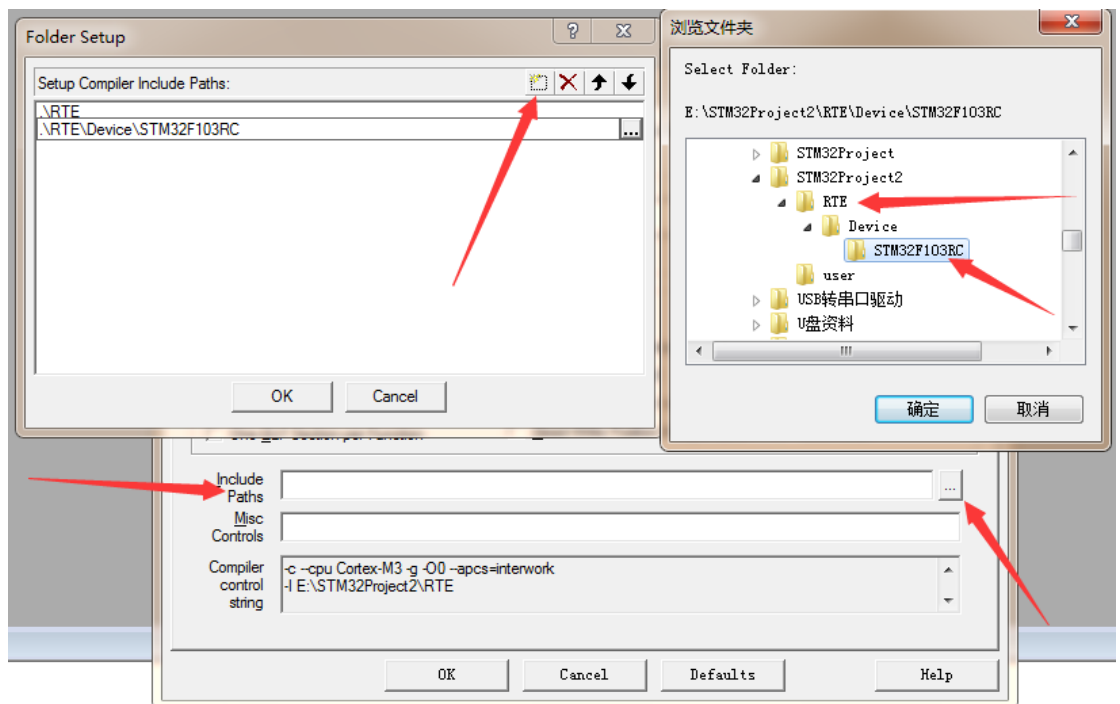
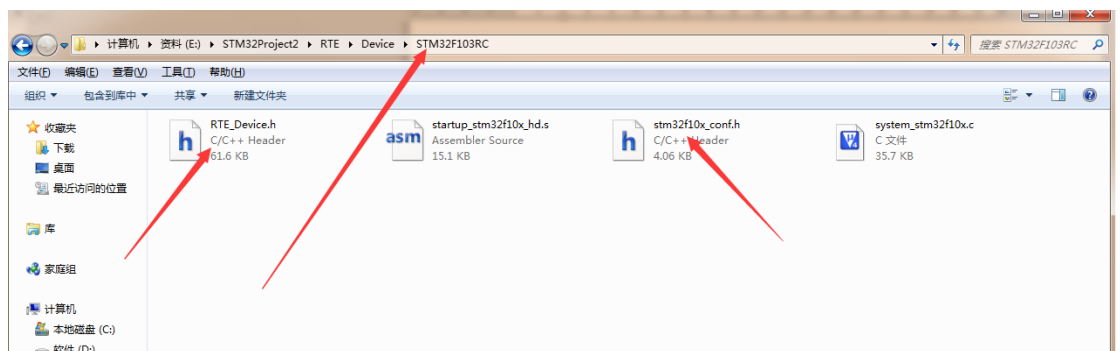
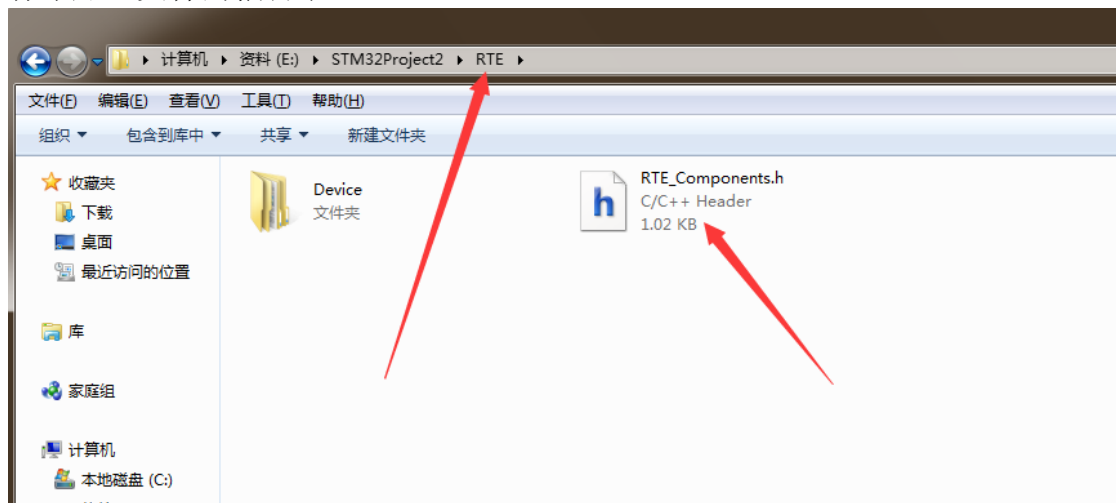


此时, 我们应该选择 `Device`——>`StdPeriph Drivers`, 就出现了一长串的外设文件选项, 全部勾选, 点击 `OK`。

注意: 我们使用 **STM32** 时不一定能够用上全部的外设文件, 可以根据自己项目的需求进行选择, 但是由于有的外设文件属于基础性外设文件, 例如 **Framework** 等, 在选择其他外设文件时必须选择, 否则会出黄色的警告, 导致编译错误。所以在你还未搞懂这些外设文件前, 建议全部勾选, 等到学习逐渐深入, 了解了这些文件间的关联性, 就可以自由选择了。

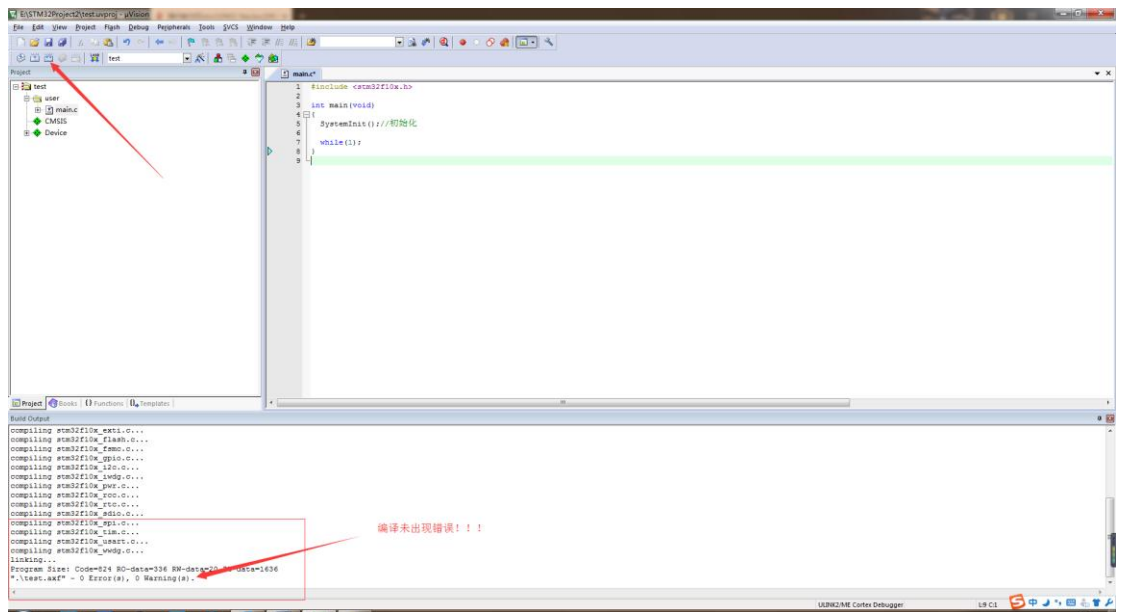


7. 配置编译相关的设置选项和上述的第 10 步类似，这里主要说明 C/C++ 标签中的 Include Paths 中添加所有 .h 文件的根目录有所不同。  
查看工程文件夹中，可以发现.h 文件的位置，然后对 Include Paths 中进行添加.h 文件的根目录。





## 8. 工程建完，成果展示。



补充：生成.hex 文件

点击上述方法中的魔棒形图标，出现 Options for Target ‘test’ 界面，选择 Output 标签，勾选 Create HEX File，单击 OK，然后对工程进行重新编译，即可生成相应的.hex 文件。

