

STM32F103C8T6 开发板

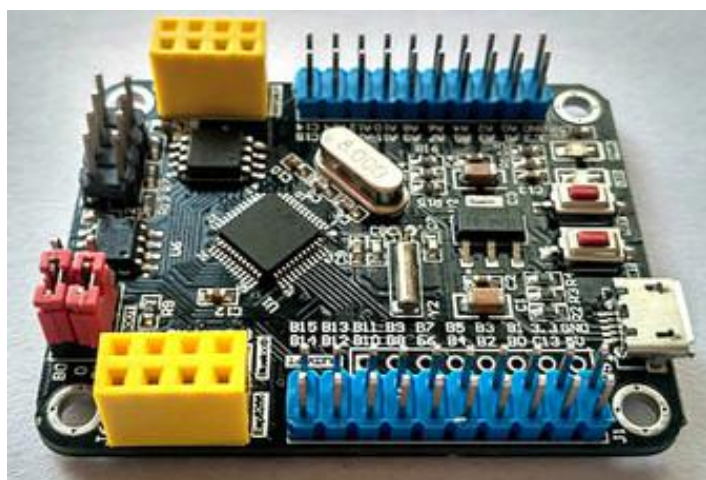
感谢您购买我们的 STM32F103C8T6 开发板

引言：

STM32 单片机作为 32 位的 MCU 已经被广大开发者、电子爱好者所慢慢的认可，也慢慢的成为了学习首先的主流单片机。想必 51 学习到一定程度的朋友都会慢慢的无法接受其速度，而开始寻找更高速的 MCU 来学习，那么相信你选择的这款开发板会开启新的旅程的。

在程序方面 为了能上更多玩 51 的朋友很好的适应，所以我们直接应用 V3.5 版本固件库。其实只要您能把固件库利用好 那么写程序上是和 51 非常接近的，也非常容易弄懂 STM32，至于具体的库函数安装方法在本手册的下面会有详细的讲解。

那么下面来一张靓照



STM32F103C8T6 芯片自身配置介绍：

STM32F103C8T6 48 脚 闪存 64k SRAM 20k 定时器 3

SPI 2 I2C 2 USART 3 USB 1 CAN 1 IO 37 ADC 2x10

下面为其他型号对比：（仅供参考）

STM32F103RBT6 64 脚 闪存 128k SRAM 20k 定时器 3

SPI 2 I2C 2 USART 3 USB 1 CAN 1 IO 51 ADC 2x16

STM32F103RCT6 64 脚 闪存 256k SRAM 48k 定时器 4

SPI 2 I2C 2 USART 5 USB 1 CAN 1 IO 51 ADC 3x16

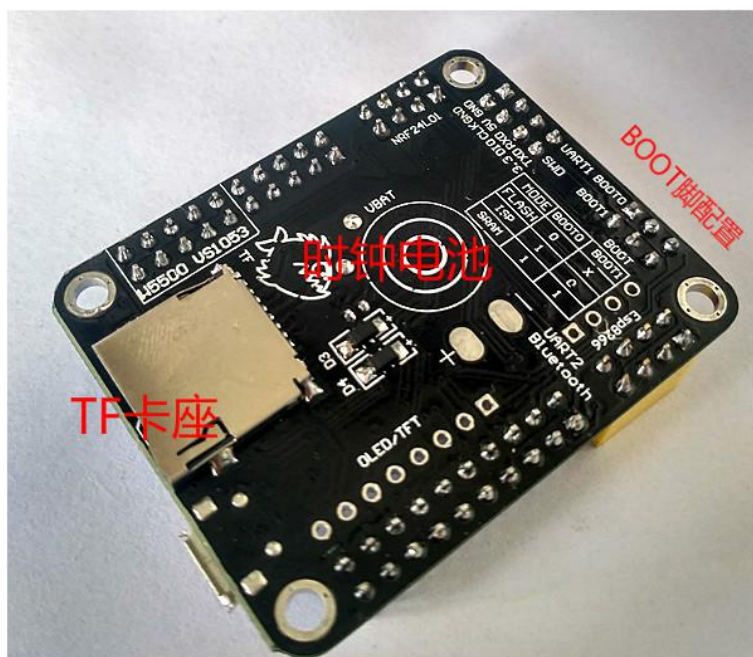
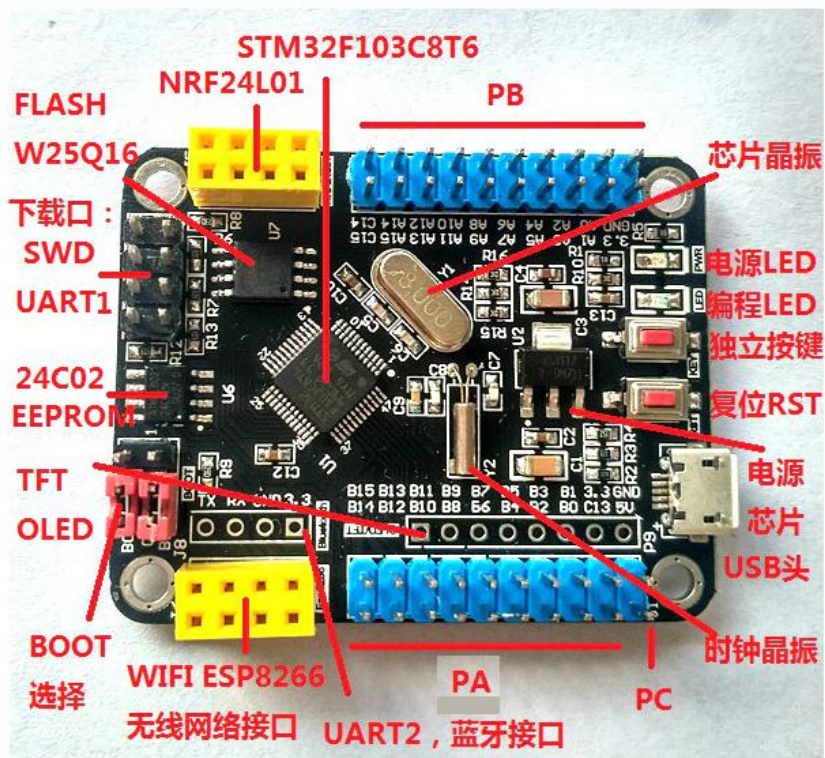
在项目开发的时候，要想好哪个芯片更适合你，这里涉及到的不只是我们店里那一点的差价，更涉及你后期量产的材料成本。

电子—STM32 功能简介

C8 开发板精简手册

这里怎么解释都不如直接拿网页上的图来的快！

硬件资源：



1 串口下载，C8 的串口下载是要通过下载器才可以下载的。在接口处同时有 SWD 接口和串口 1 接口，其中串口 1 就是用来做串口下载和串口通信用。如下图：



2 支持 SWD 在线仿真功能，SWD 在线仿真是要用的 Jlink OB 或者 JTAG 跳线方式下载、仿真的。在使用仿真器的时候要注意保持一路电源做开发板供电用，比如用 Jlink OB 供电板子就不要在插 USB 线了；如果用 USB 线来供电，那么就要把掉 Jlink OB 的供电线，即此时 Jlink OB 有三根线给开发板供电



Jlink OB 仿真器

3 采用的时钟晶振用的是比较优质的贴片式晶振，对应的电池焊接位置在板子的背面，我们采用的是直接电池焊接。

4 板上可扩展了EEPROM 芯片，芯片是128个字节的ATMEL24C02，对于一些小数据的存储还是比较实用的。

5 另一可扩展芯片为2M的FLASH芯片，它可以存储一些比较大的数据，当然也可以用更大空间的FLASH芯片。

6 支持小尺寸TFT屏及OLED屏等功能

7 可以扩展无线NRF模块、最近网上比较红的ESP8266模块、蓝牙模块等等。更多功能在程序里已经有所体现了。

8 其他本地外围模块，还可以扩展本店的W5500网络模块和VS1053音乐解码模块模块

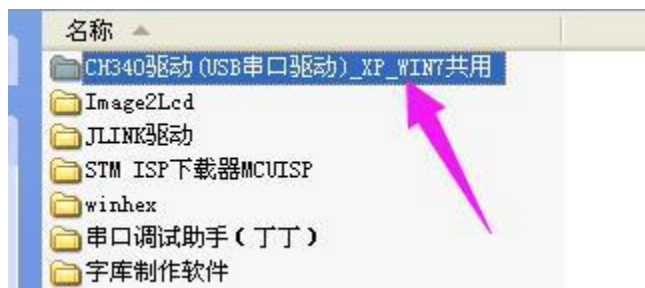
下载方法

下载一般分为两种 一种是用仿真器下载 另一种是用下载芯片下载

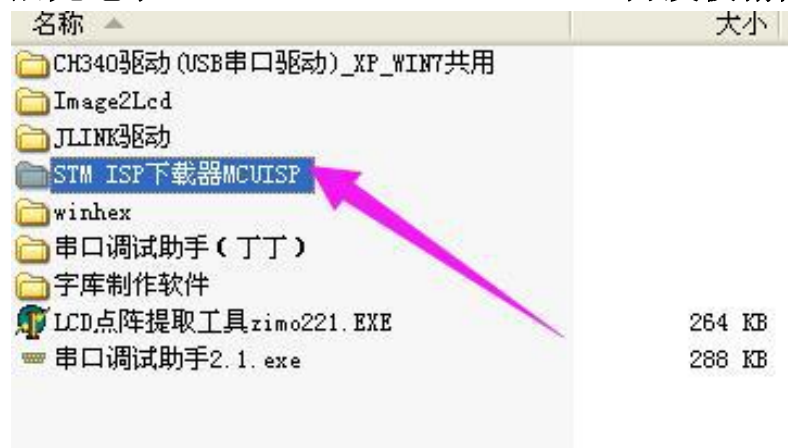
在程序方面 我们的资料里的程序都是在 keil4 的基础上编写的 所以没有安装 keil4 的朋友要先安装资料里的 keil4 并按步骤注册

USB 下载

用 USB 下载 则首先要考虑你手里的下载芯片是否在电脑上已经安装好驱动了，我们在资料里给大家带了比较常用的 CH340 驱动
驱动在 资料 应用软件



驱动安装后 就可以直接用下载软件给板子下载 HEX 文件了
下载软件在资料



里

下载前 要先用下载线和板子串口 1 接口连接好

如果驱动成功 则电脑会读出 驱动后的 COM 口

然后打开下载软件



如果驱动成功 则会



这里选择 COM4

波特率 bps 可以选择最大 这样可以最快下载

强调一下 BO



图中有两个跳帽来调整 BOOT1 和 BOOT2 的状态。跳帽跳到上边是选择 1 也就是高电平 跳到下边是选择 0 也就是低电平
如下：

BOOT0 和 BOOT1 是用于设置 STM32 的启动方式的

BOOT0	BOOT1	启动模式	说明
0	X	用户闪存存储器	用户闪存存储器，也就是FLASH启动
1	0	系统存储器	系统存储器启动，用于串口下载
1	1	SRAM启动	SRAM启动，用于在SRAM中调试代码

这里是用 USB 下载，也就是串口下载 所以选择表中的第二个方式
也就是 BOOT1 选择 0 BOOT0 选择 1

在软件上点击开始编程后，要按一下板子上的 **RESET** 键进行

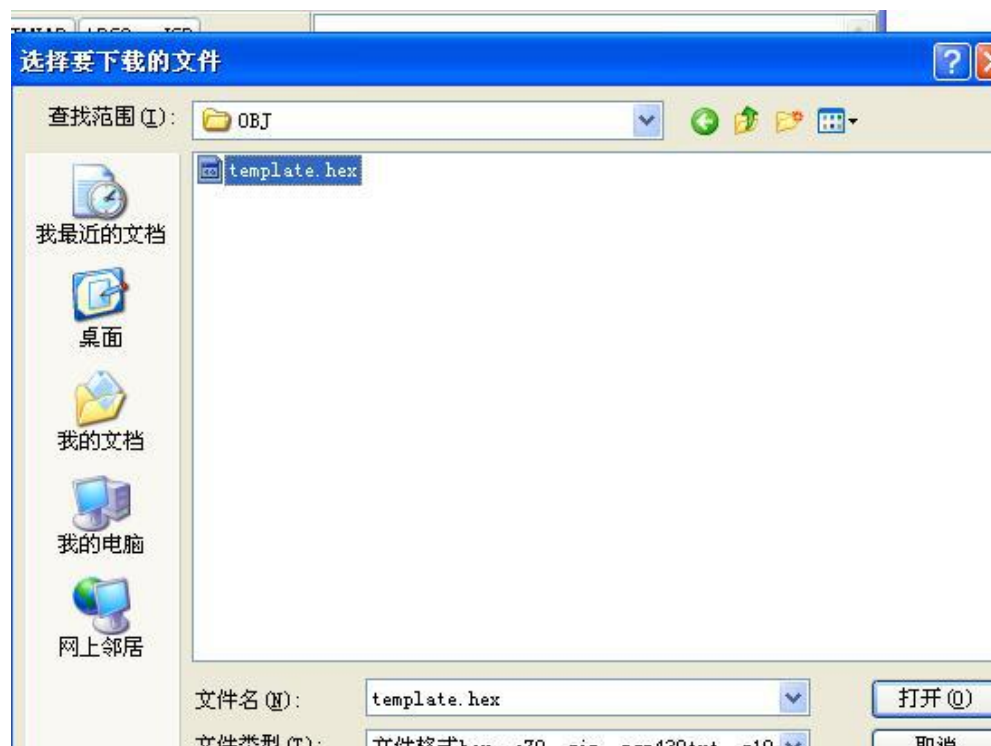
一键下载



这里我们以第一个实验 **LED 灯实验** 为例



HEX 文件 在每个实验的 **OBJ** 文件夹里



选好以后 直接点击编程即可

开始编程(P)

同时右边将会有相应显示 以提示下载是否成功



仿真器下载

仿真器 用的是 J-Link OB 仿真器

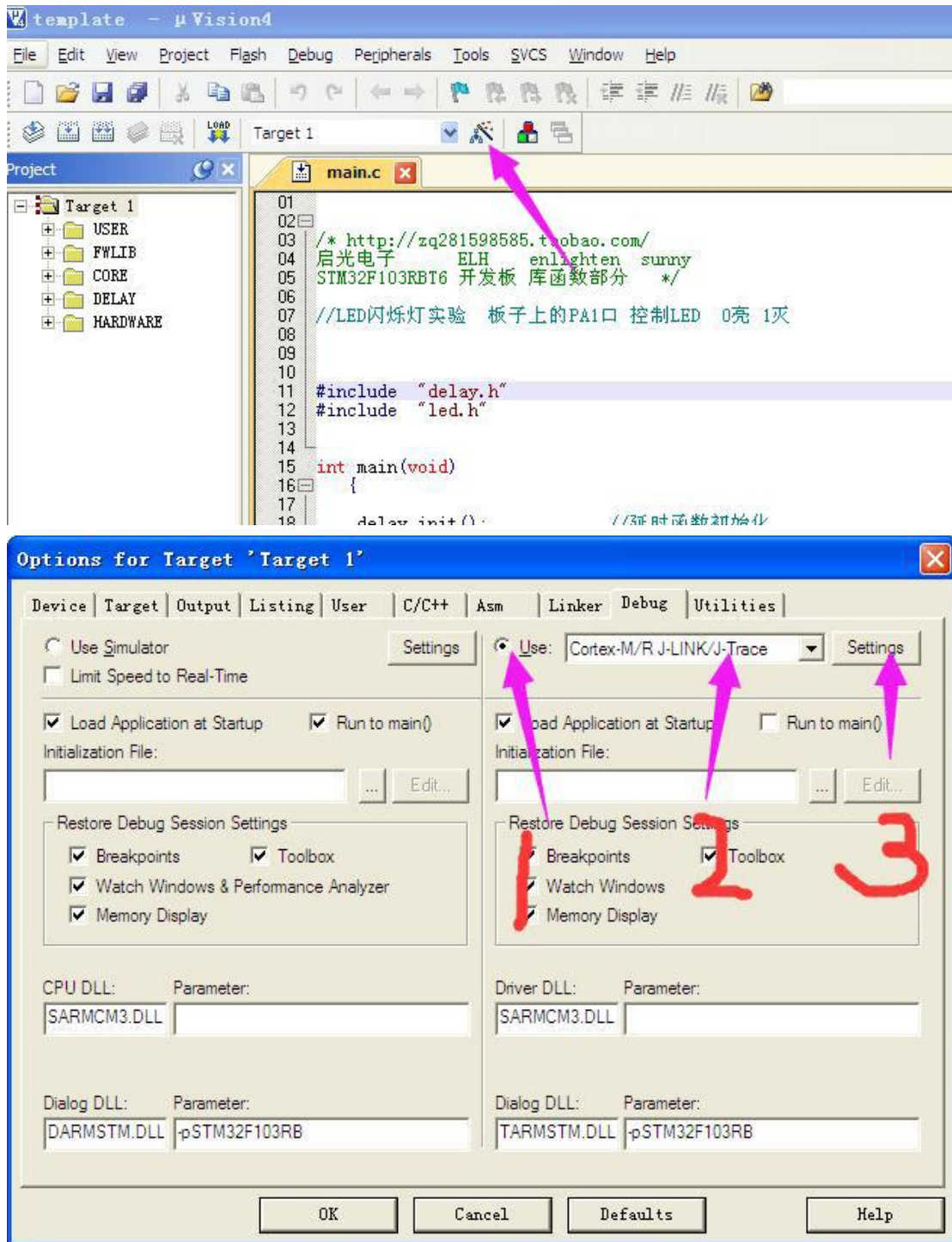
仿真器方式 在程序里用的是 SWD

具体的软件设置如下

安装仿真器 驱动

在 应用软件  中

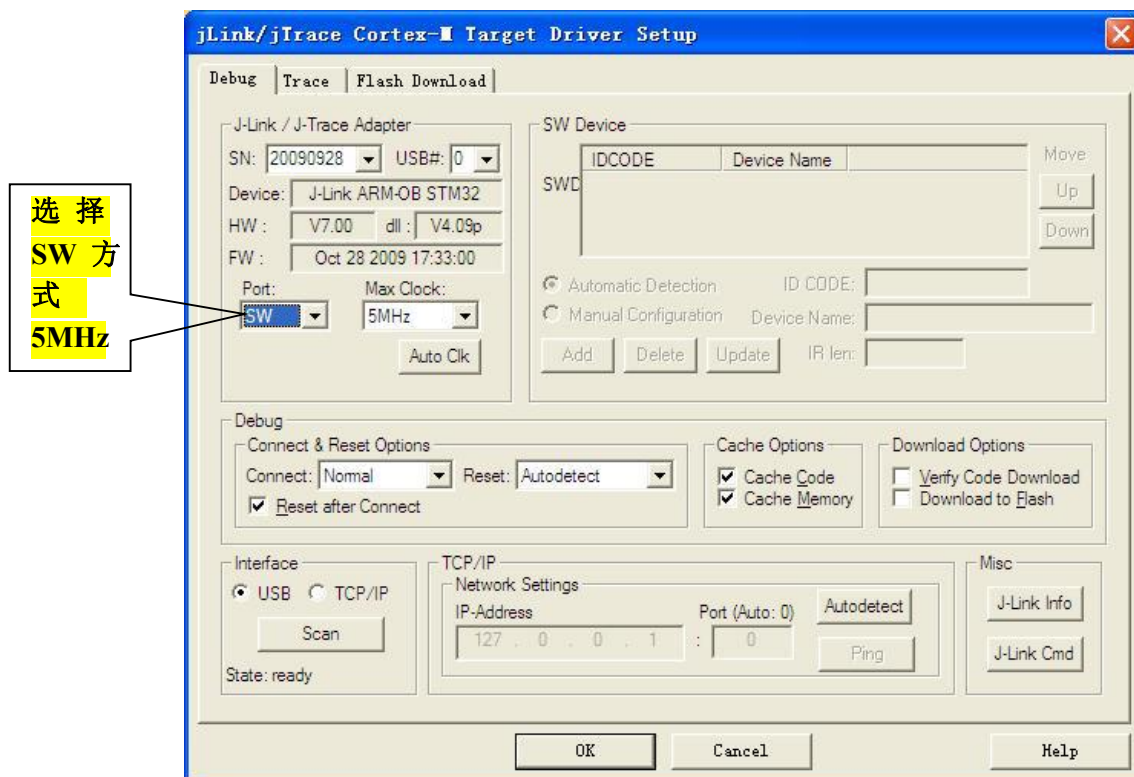
安装好后打开 KEIL4 点击



选择 Debug 选项卡

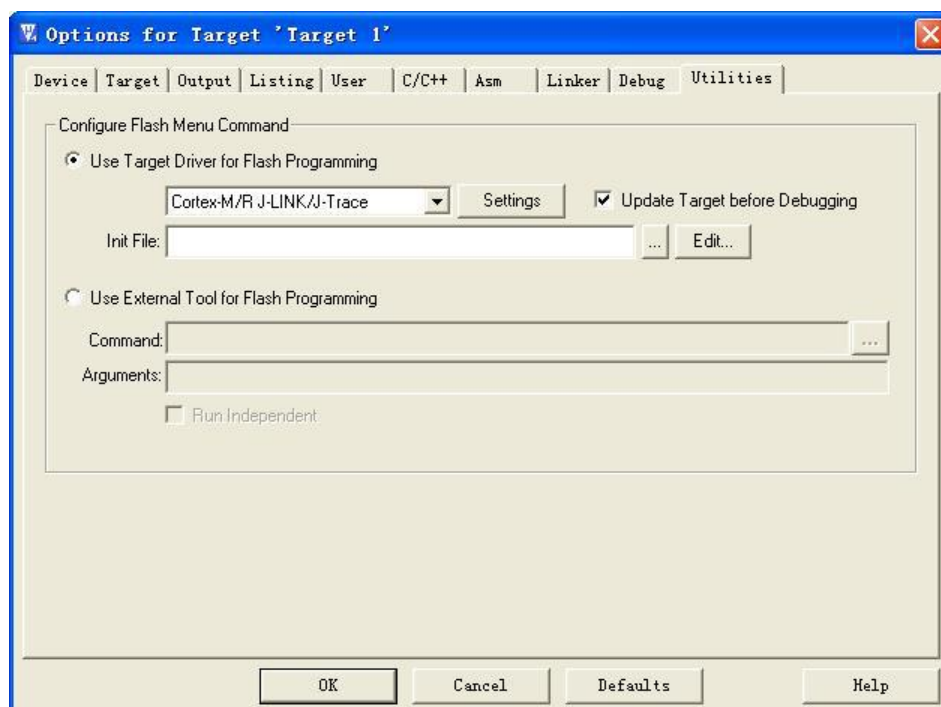
1 点击右侧的 Use 并选择如图 2

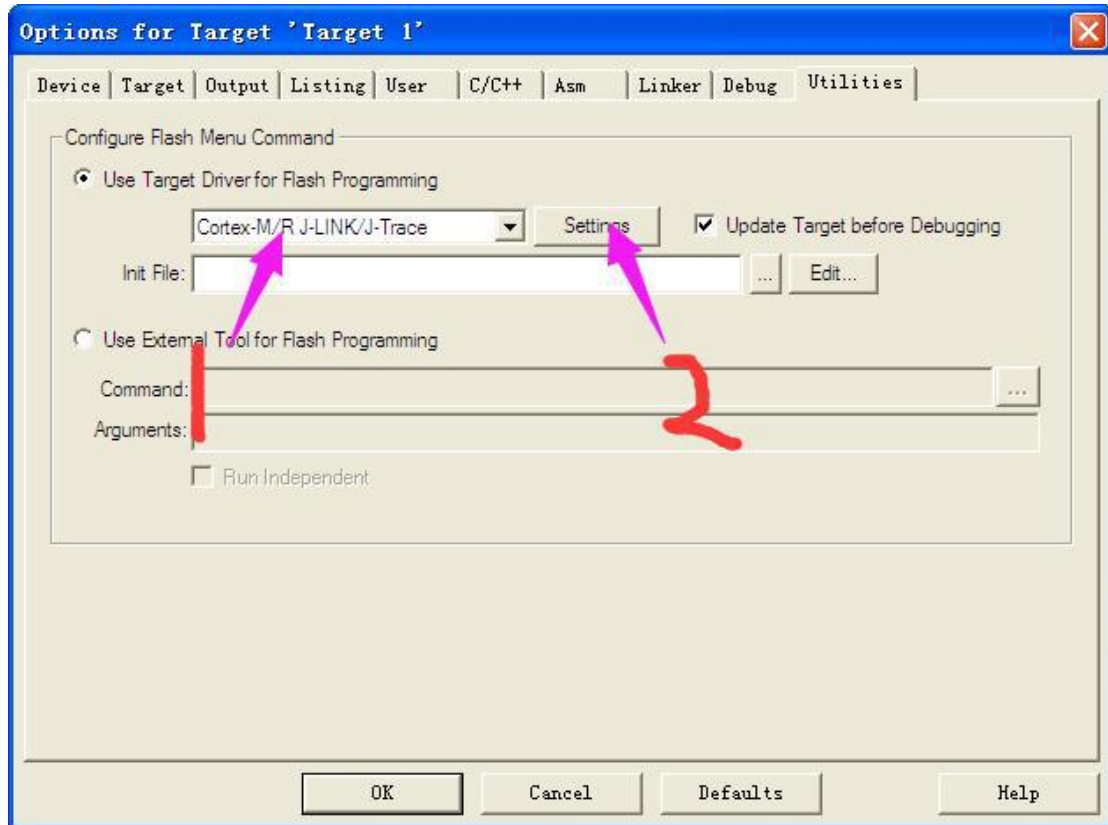
然后点击旁边的 3 Settings



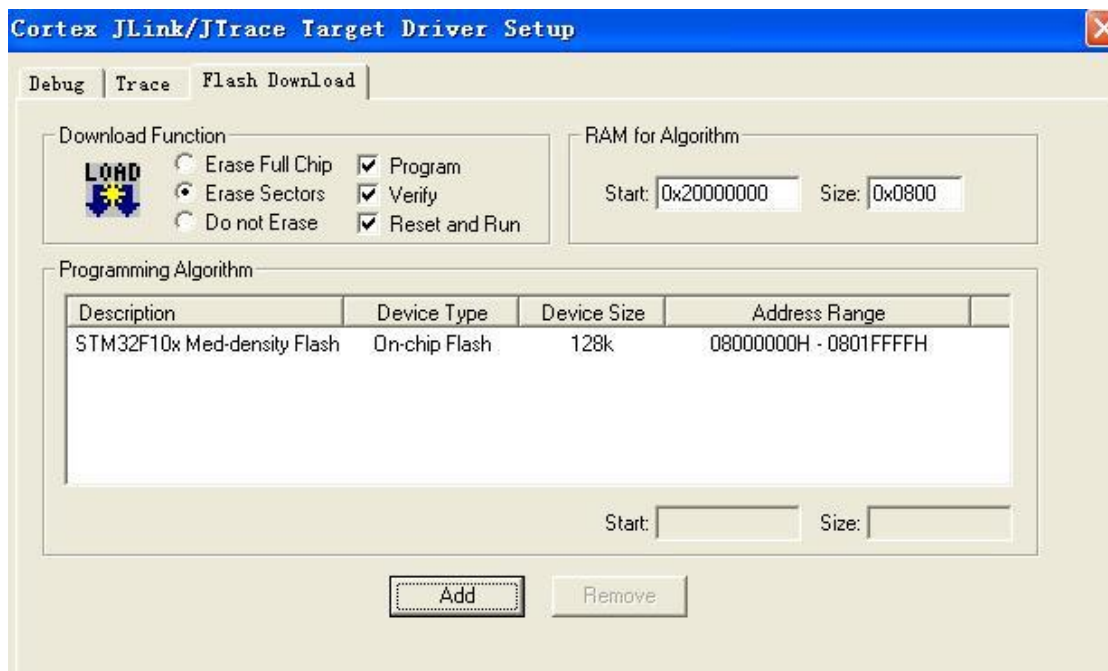
选择如图 选择后 点击 **OK** 即可

然选择 **Utilities** 选项卡 如图





选择好 1 然后点击 2



电子—STM32

C8 开发板精简手册


进入后 点击 Add 选择框中 选项 因为 STM32F103C8T6 为 64k 即
中型 128k 芯片

一切选择好以后 点击 OK 即可

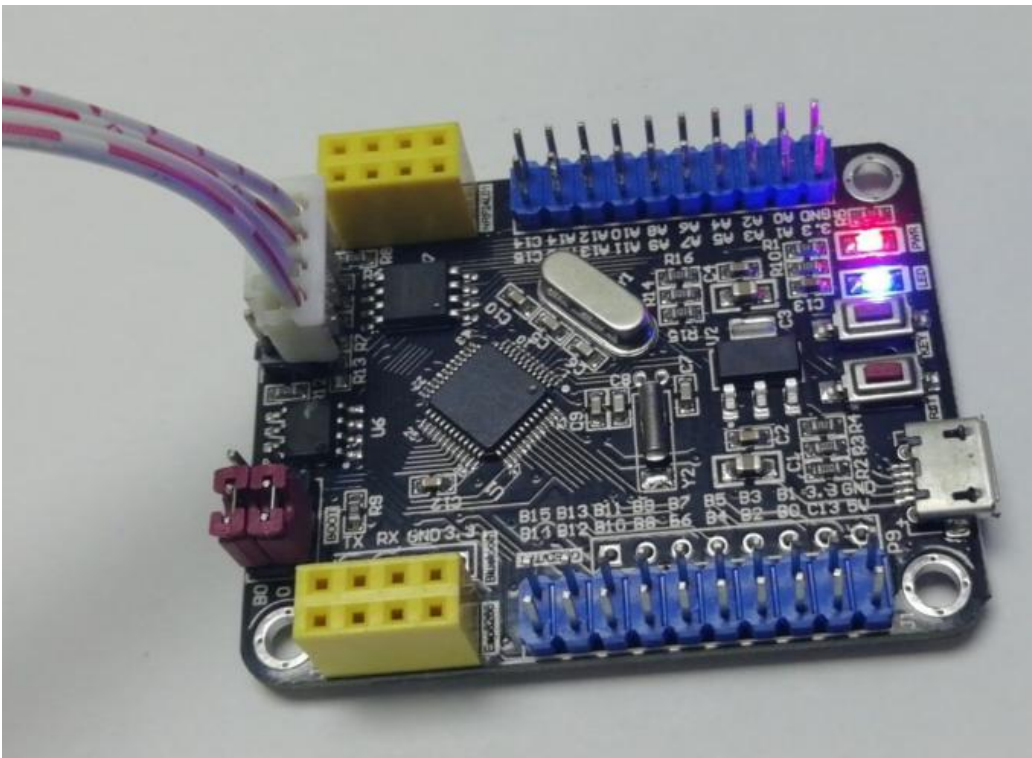
这样 对仿真器的设置就完成了

如果有看不明白的地方 可以直接打开我们资料里的程序 一一对应

如果用仿真器下载 点击  选项即可（当然要编译后）

如果要用仿真器单步仿真 选择  即可

LED 灯效果图片



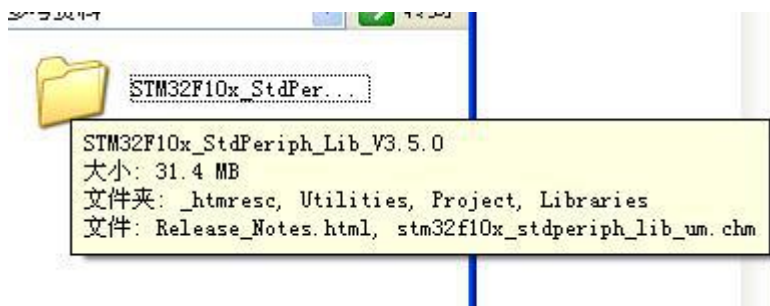
3.5 版固件库 创建篇

此篇主要讲解如何利用 官方的 3.5 固件库 创建自己的工程
主要两个步骤

- 1 利用固件库文件 创建文件 为建立工程做准备
- 2 创建工程

下面分步讲解

第一步骤



如图 为官方 3.5 版固件库



把它拽到桌面



然后在旁边创建文件夹 名字自定

打开刚创建的文件夹（STM32 工程）并在里面建立 4 个文件夹

电子—STM32

C8 开发板精简手册



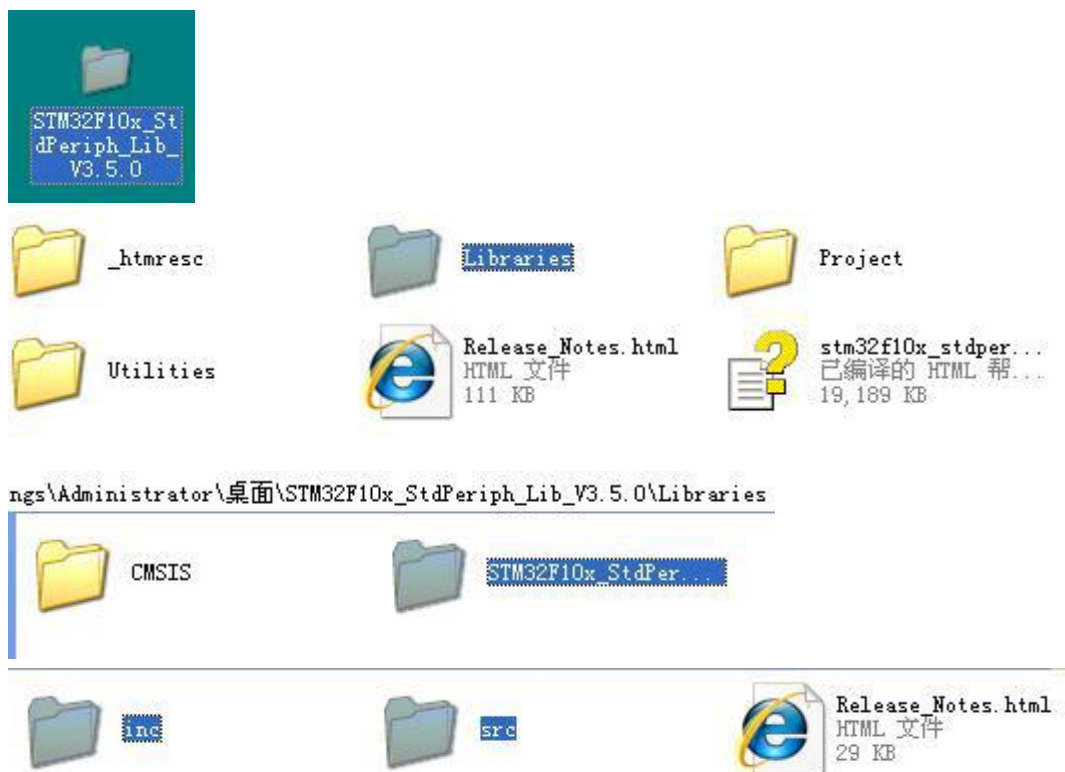
USER 使用者文件夹 主要存放打开工程的文件等

CORE 存放 启动文件 核心文件

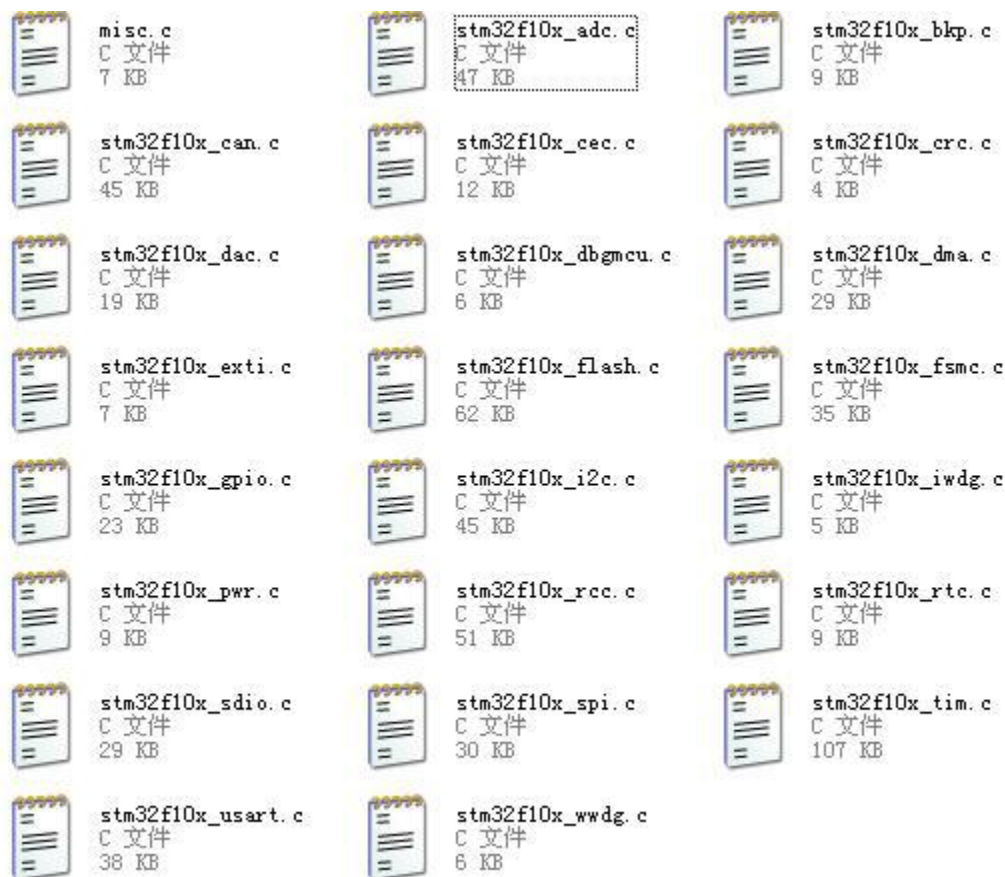
FWLIB 源码 STM32 各种外设的底层程序 如 gpio

OBJ 过程文件 如 HEX 等

A 在文件夹 **FWLIB** 存入源码



到这里 **src** 为底层源码.c 文件

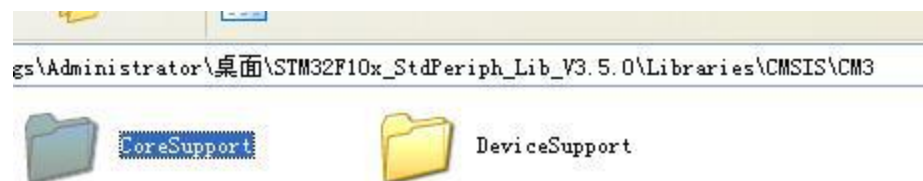
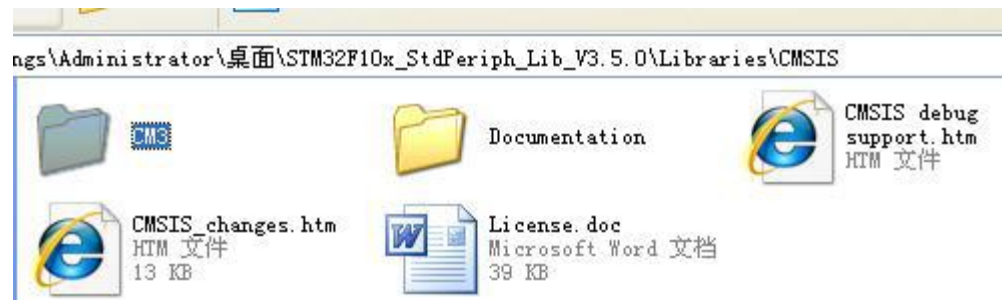
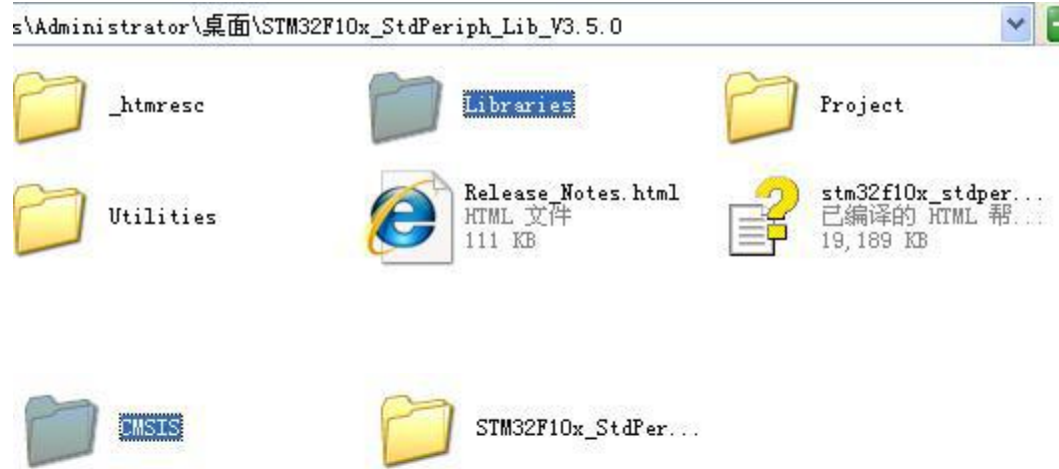


那么 inc 就是底层源码.h 文件

直接将这里两个文件夹复制到 FWLIB 文件夹中



B CORE 启动文件及核心文件



此文件为**核心文件** 复制到 CORE 中



然后固件库向上 回到



到这里 此文件为**启动文件** 复制里面后缀为 **md.S** 的文件

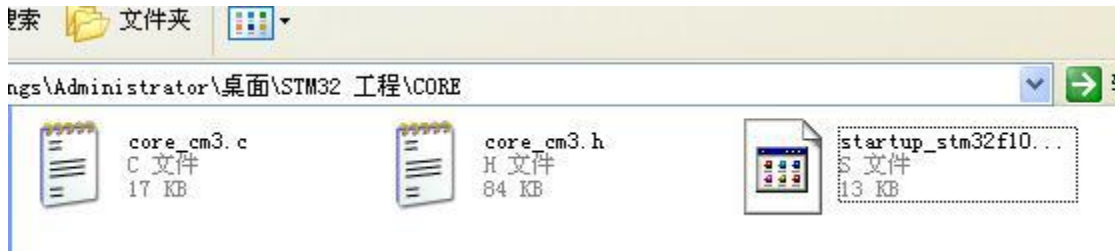
这里说一下

startup_stm32f10x_ld.s: 适用于小容量 产品
startup_stm32f10x_md.s : 适用于中等容量产品
startup_stm32f10x_hd.s: 适用于大容量产品
这里的容量是指 FLASH 的大小.判断方法如下:
小容量: $\text{FLASH} \leq 32\text{K}$
中容量: $64\text{K} \leq \text{FLASH} \leq 128\text{K}$
大容量: $256\text{K} \leq \text{FLASH}$

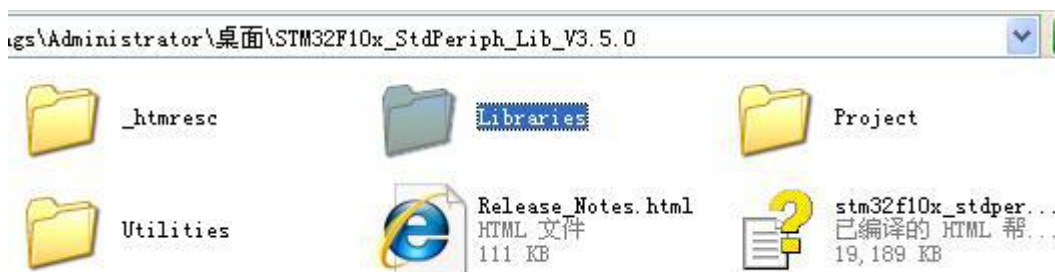
我们这里用的是 **STM32F103C8T6** 为 **64K** 所以为中容量

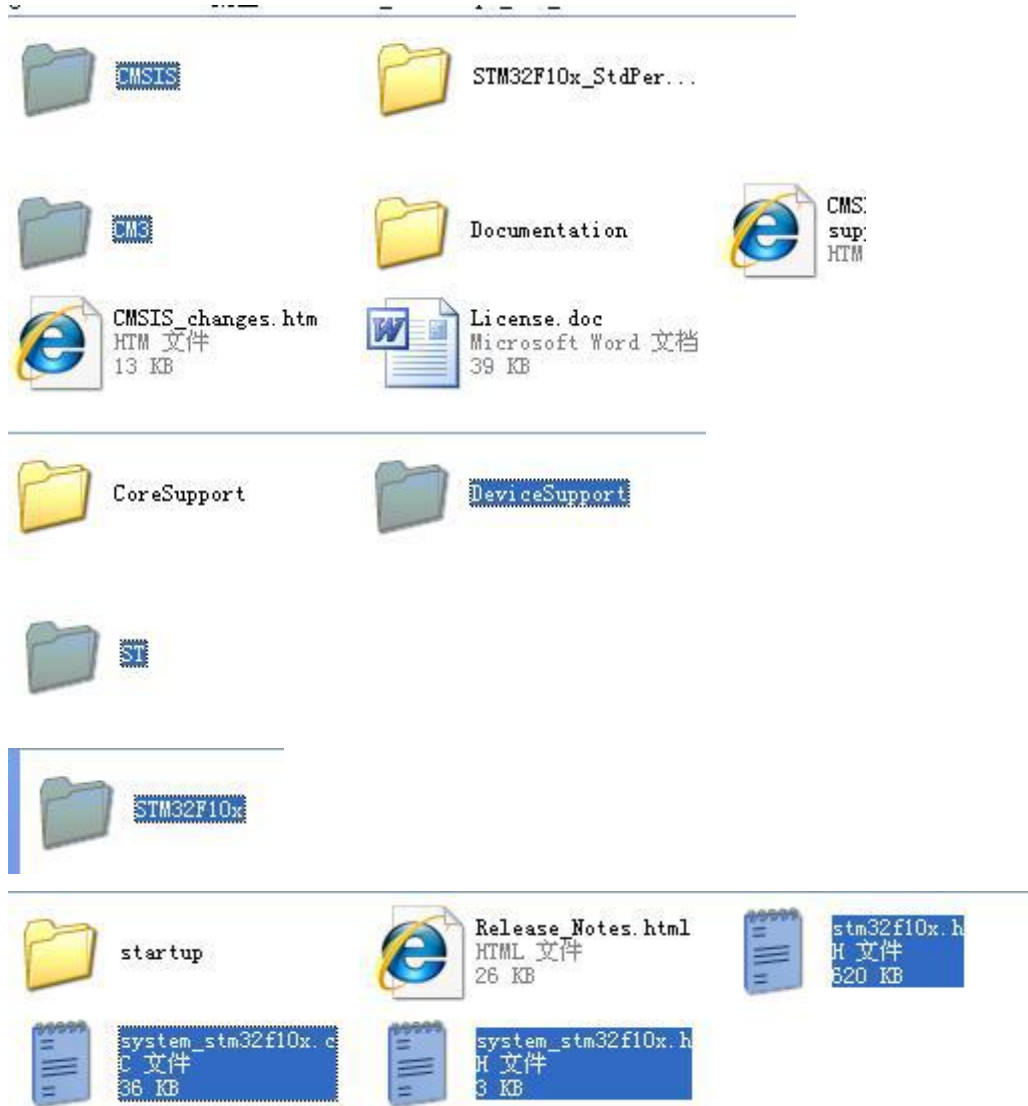
选择 **startup_stm32f10x_md.s**

最后 **CORE** 文件为



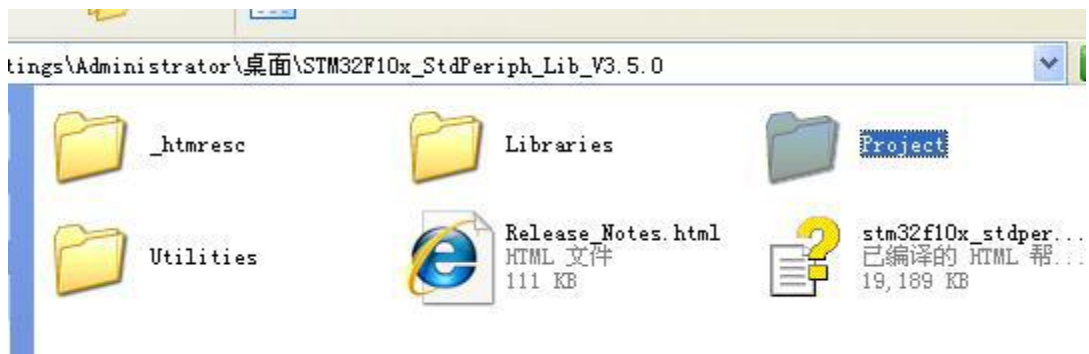
C USER 使用文件

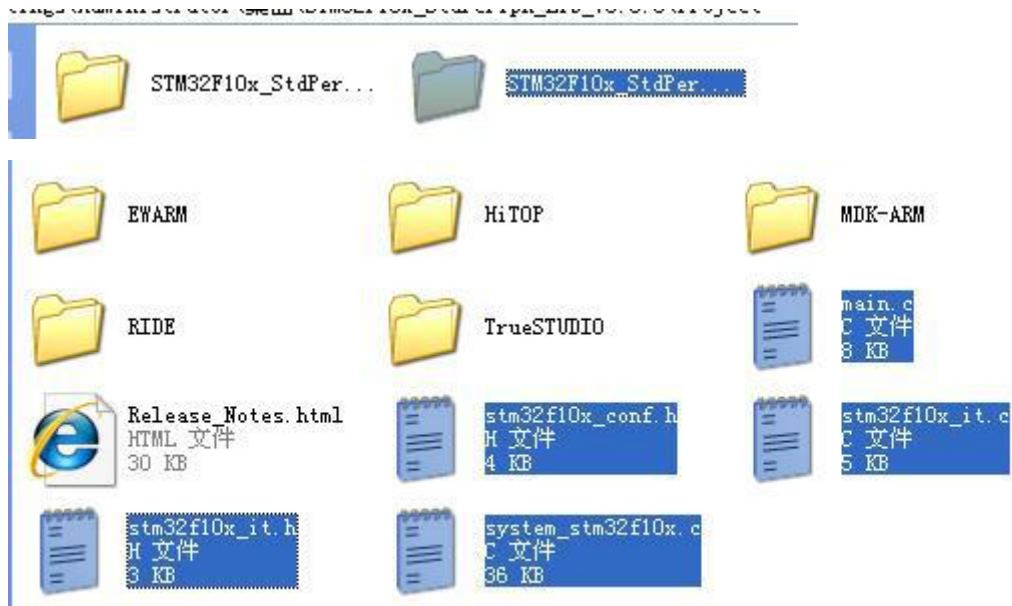




到这里 选择 选中的文件 复制到 **USER** 文件夹中

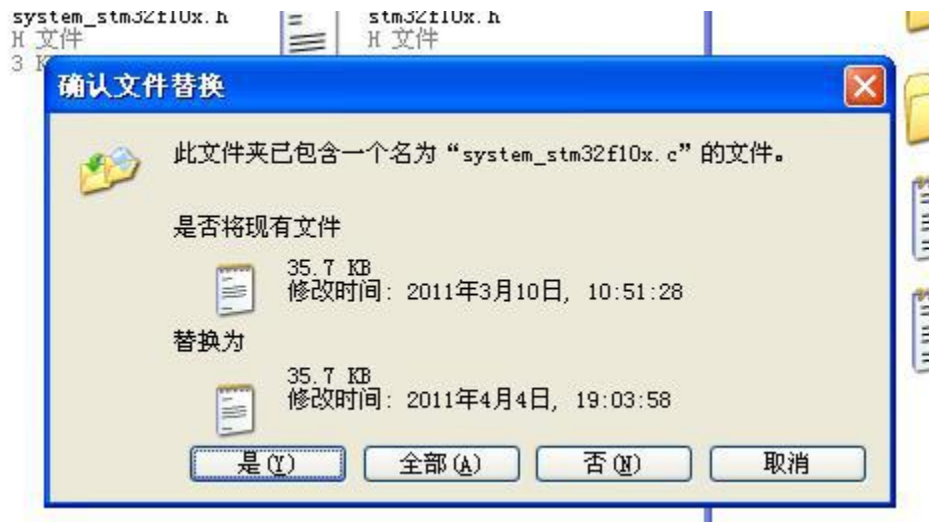
然后回到最上面





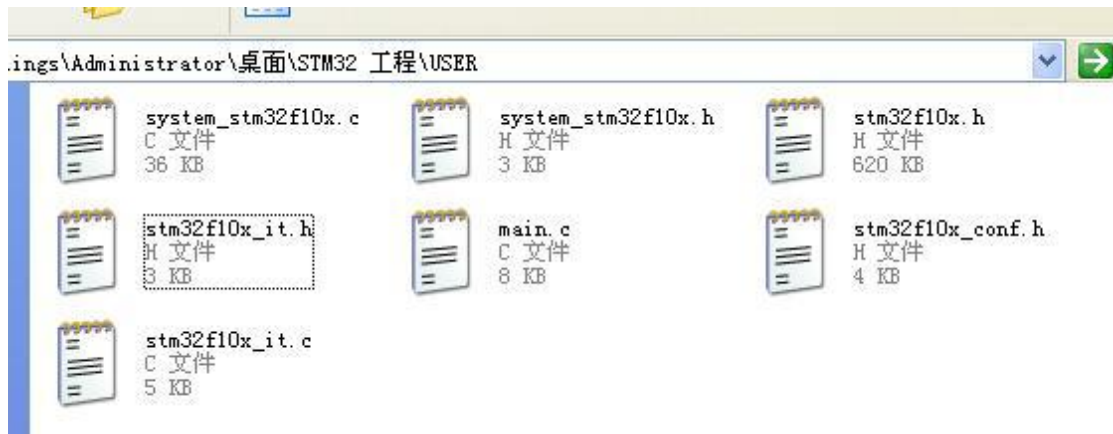
复制选中的 同样到 **USER** 文件夹中

这是出现



选择 是 或 否 都一样 因为两个文件是一样的

这样 **USER** 就搞定了

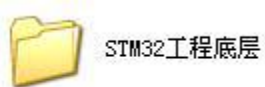


第一步完成



在这个基础就可以创建 工程了

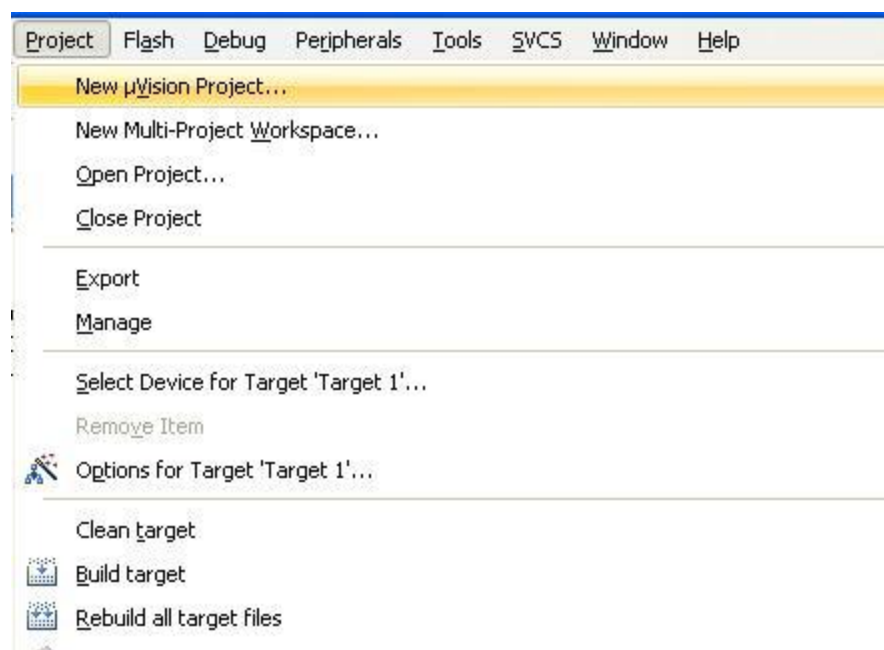
这个底层的工程 在此文件夹已经做好



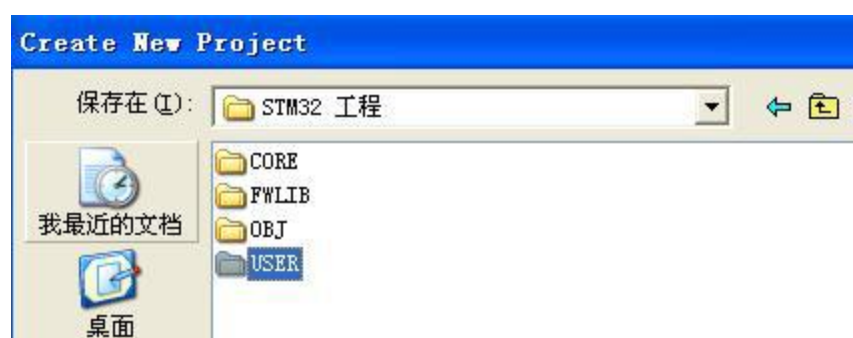
可以直接复制过去 在基础上打开 KEIL 创建工程
(直接进入 第二步骤)

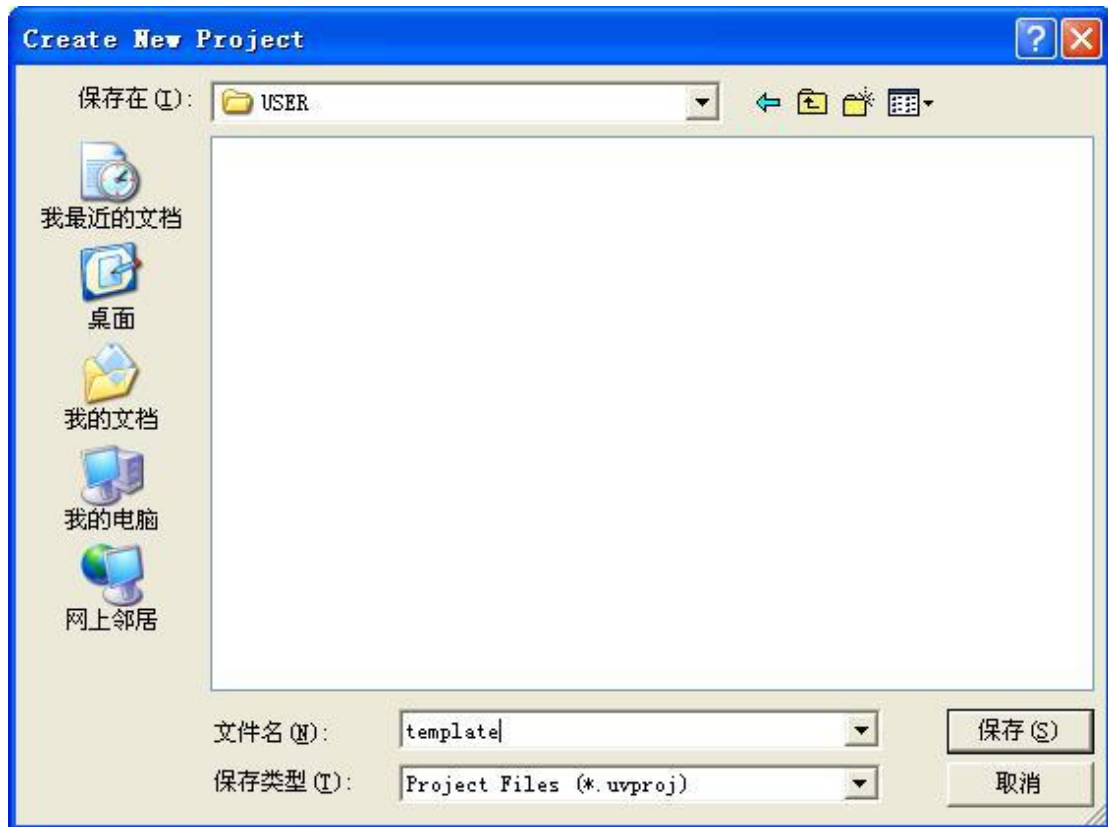
第二步骤

打开已安装好的 **KEIL4** 软件



创建工程

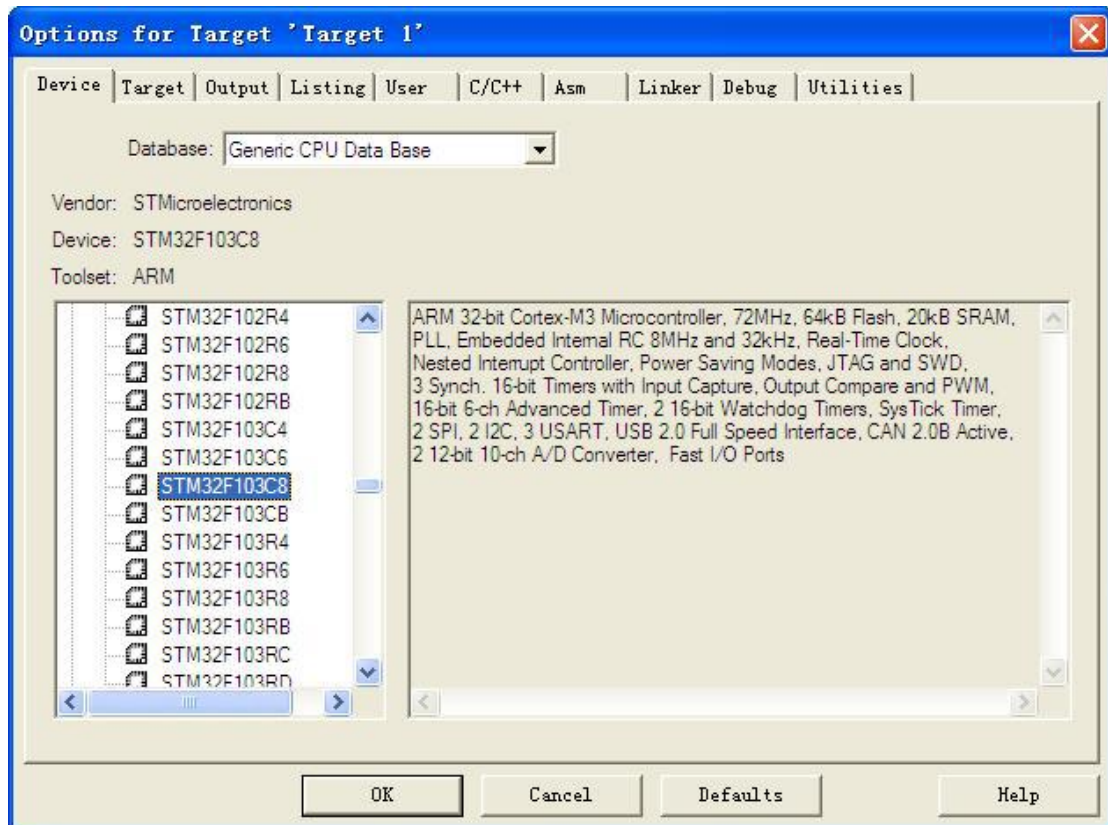




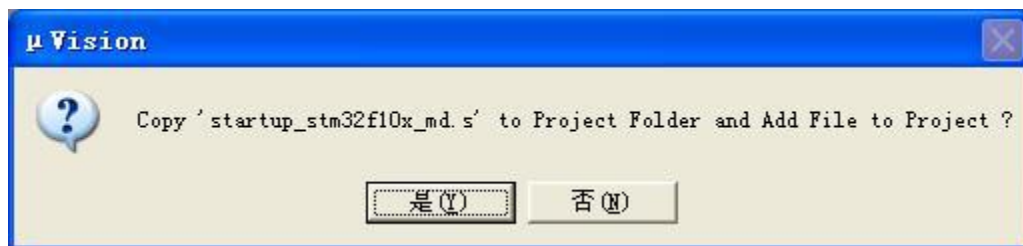
工程名随意 确定后点保存



找到 **STMicroelectronics**



选择 **STM32F103C8** 点 OK

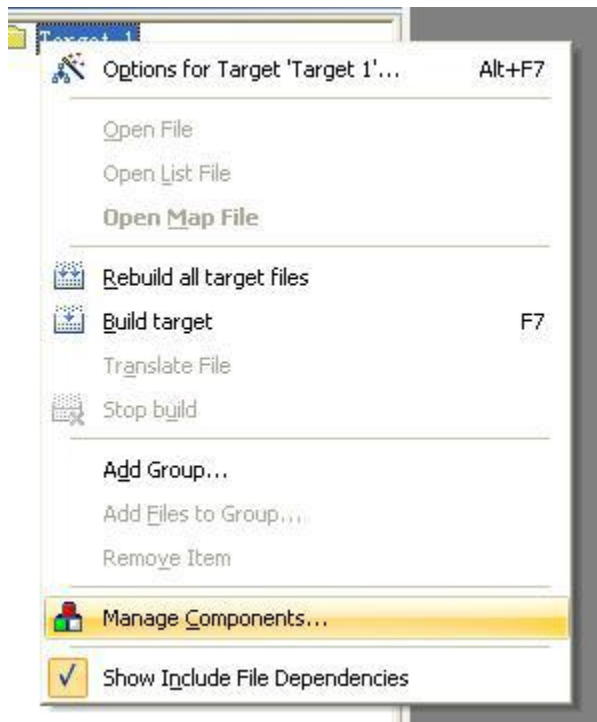


点否

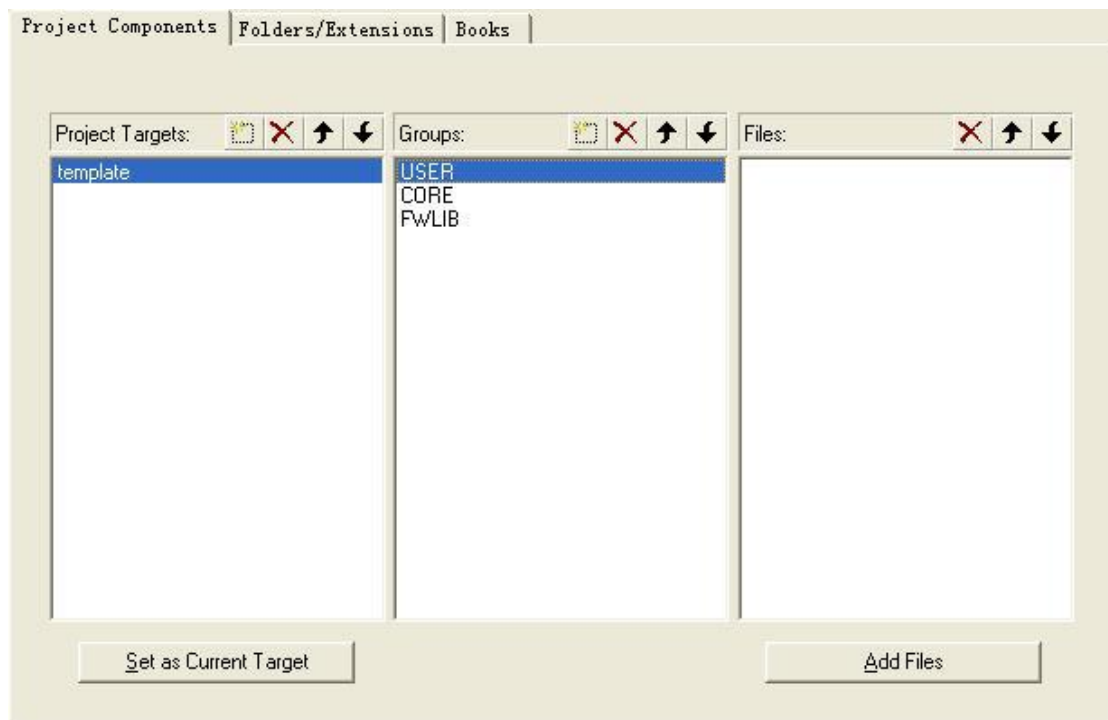
这样工程建立完成 下面建立结构并导入源码.C



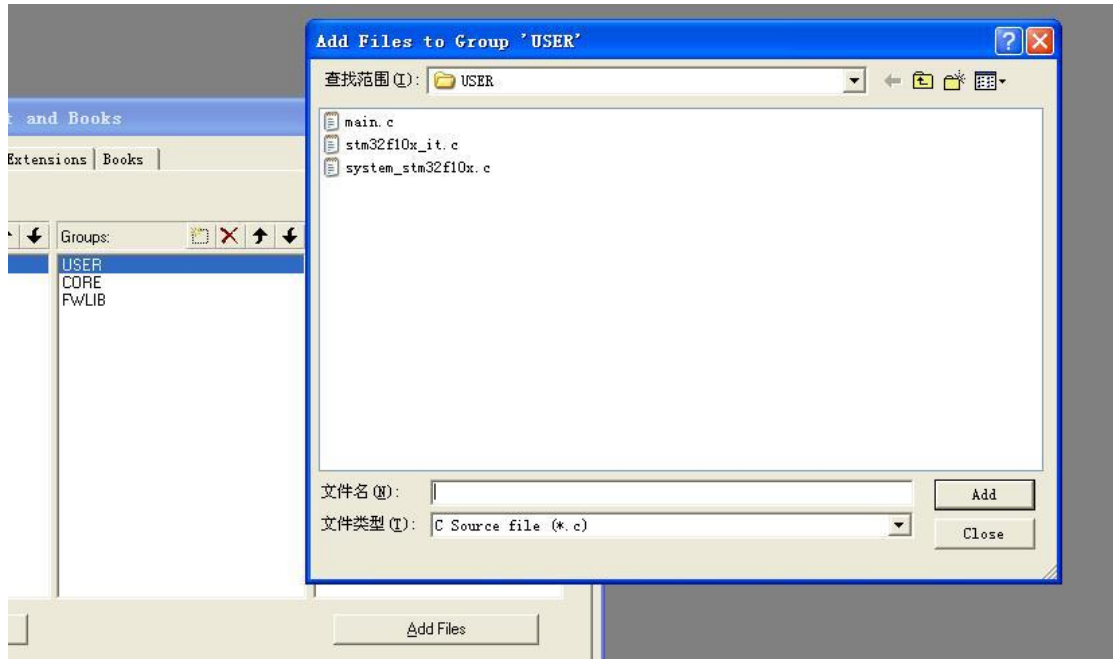
右键



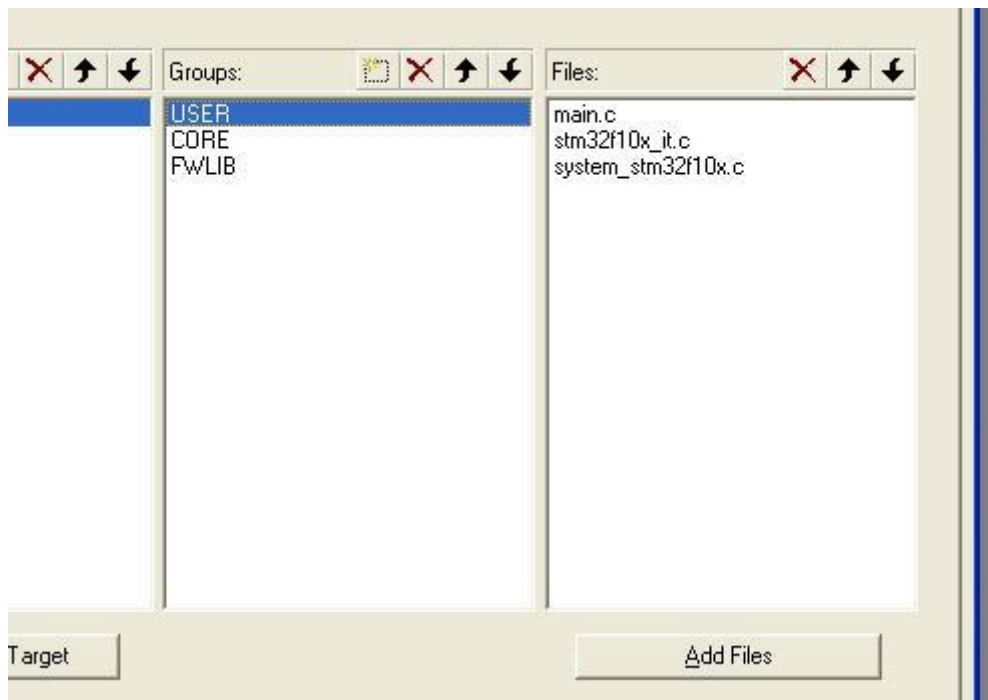
选择 **Manage Components...**

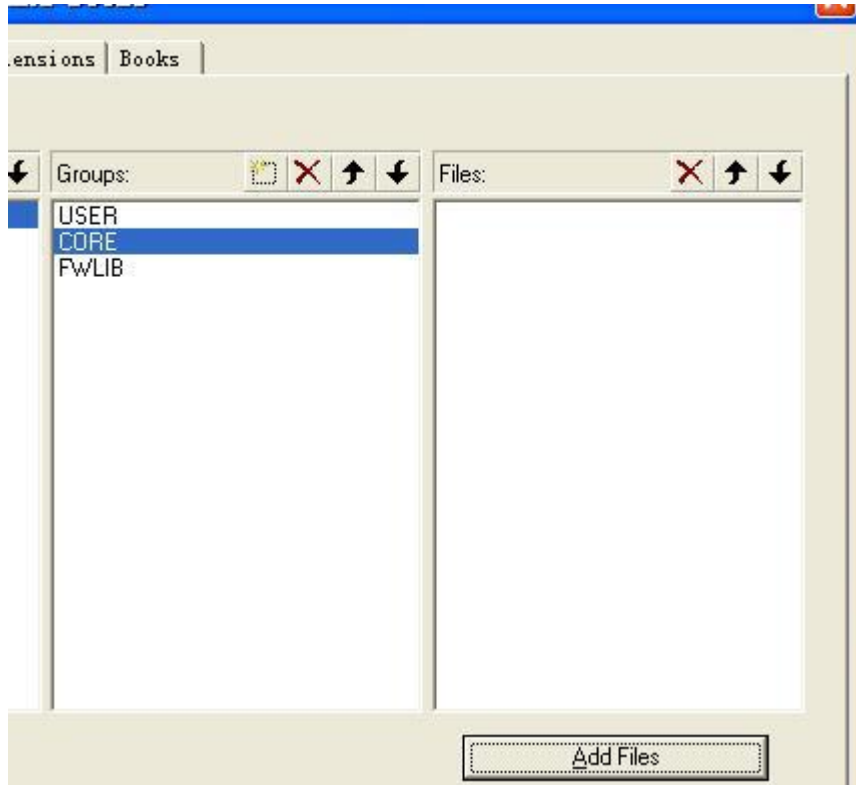


创建如图文件 **USER** **CORE**（启动 核心） **FWLIB**（底层源码）

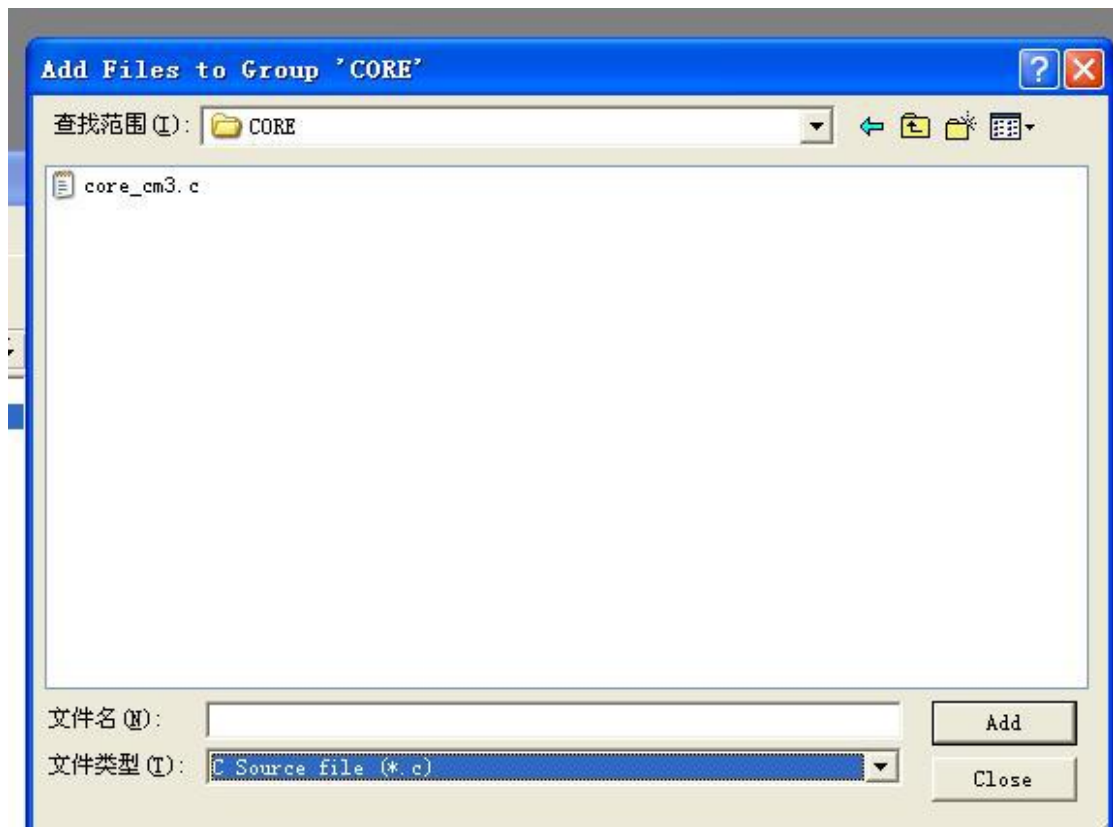


USER 下 Add Files 然后在弹出的对话框中全选 点 Add

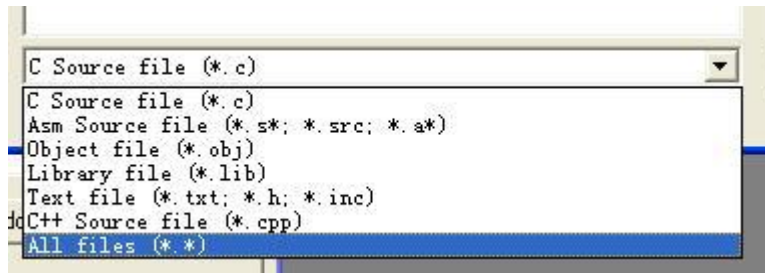




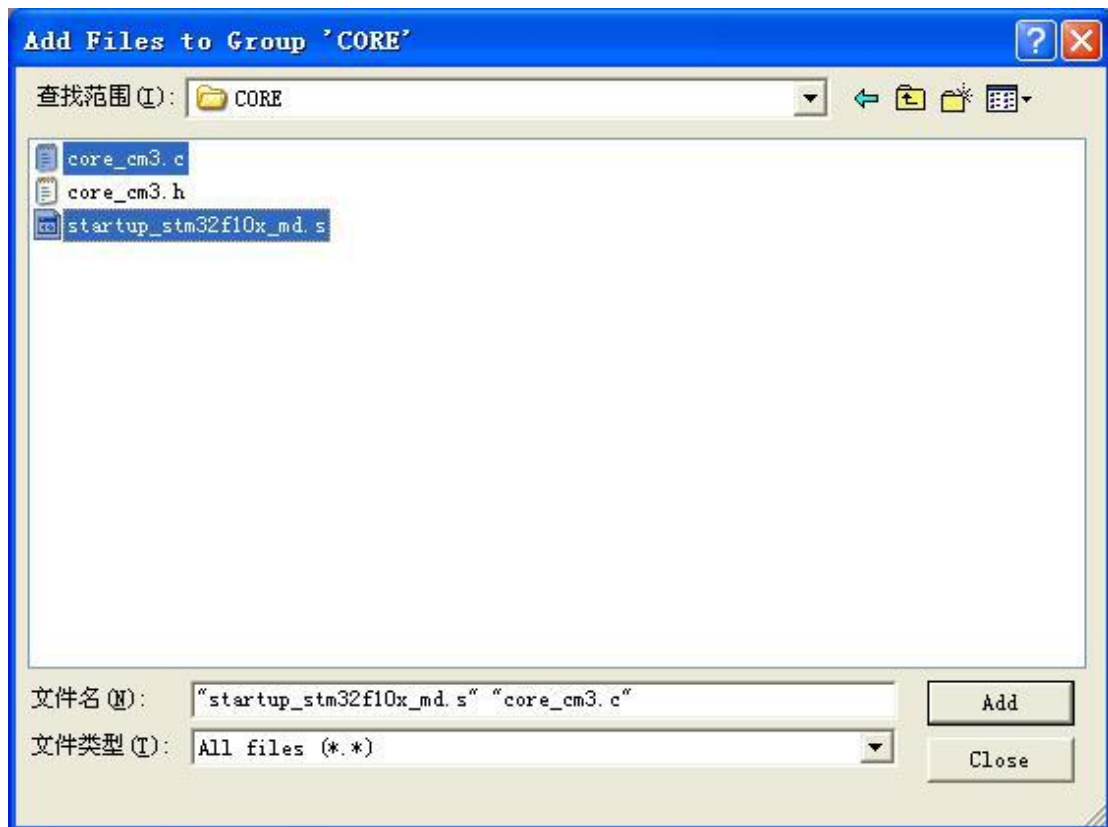
然后 CORE 下 Add Files



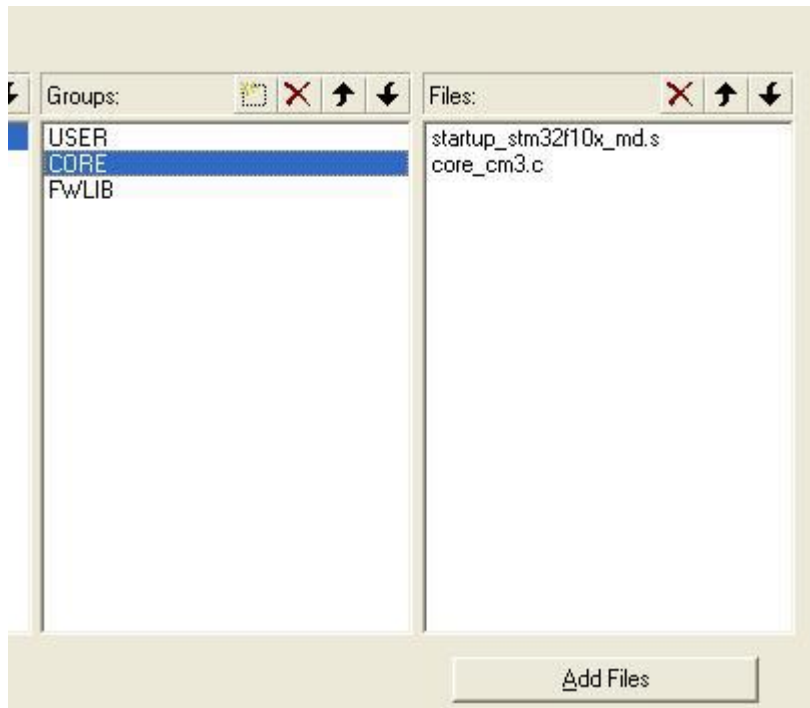
这里只显示了 核心文件 没有启动文件 是因为文件类型只为.C



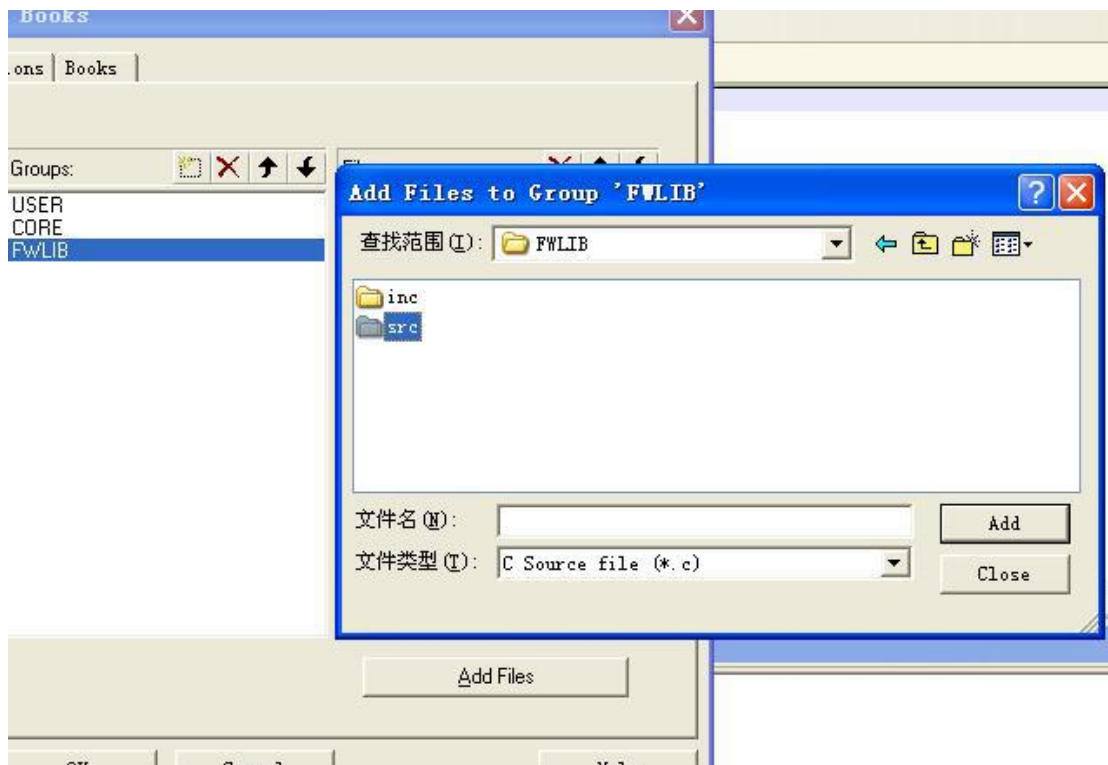
在文件类型选择 All files



然后选择图中选中的两个文件 点击 Add

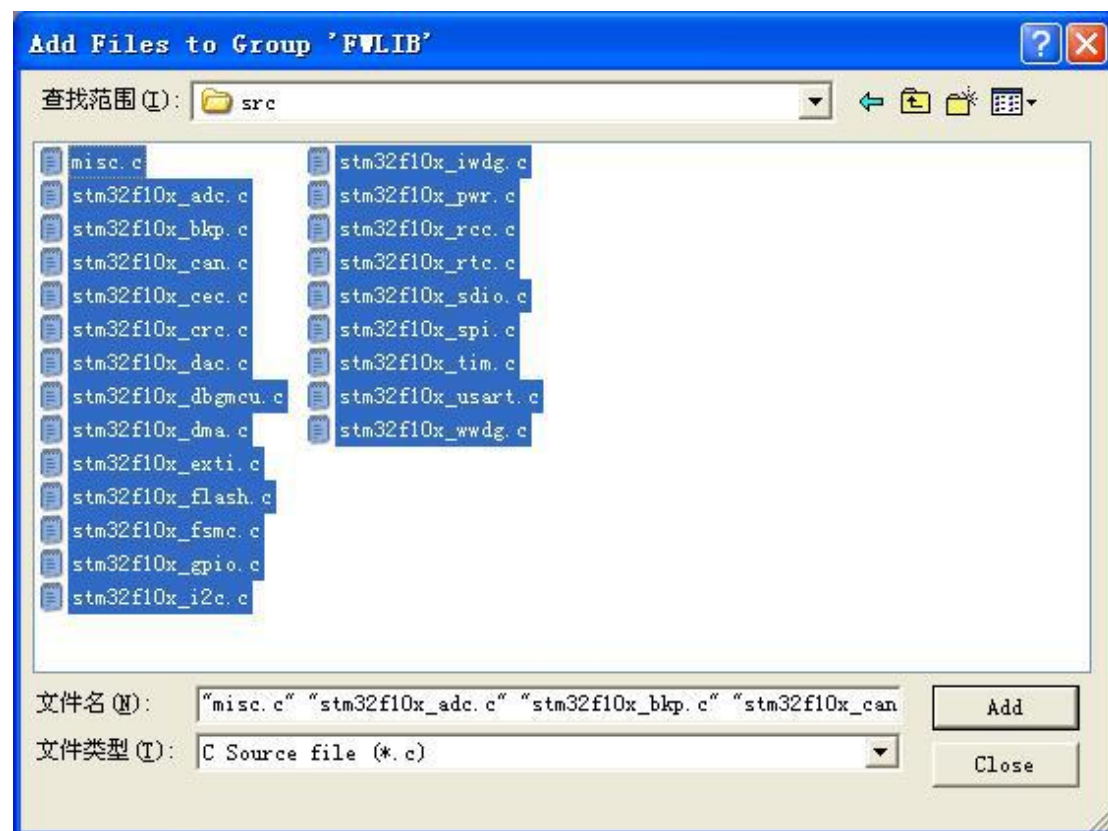


然后 FWLIB 下 Add Files

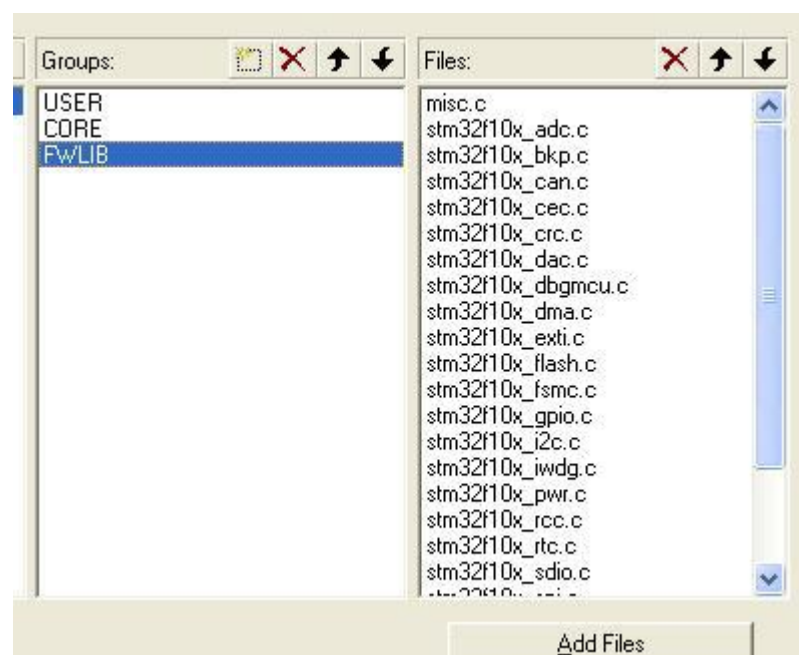


找到 FWLIB 文件夹 后 选择 src 文件夹 此文件里为底层源

码.c 文件

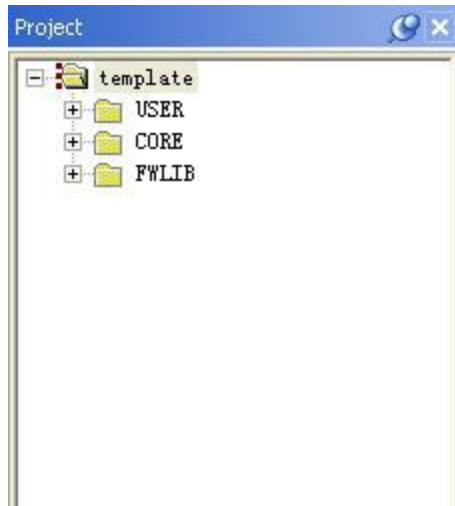


全选 点 Add

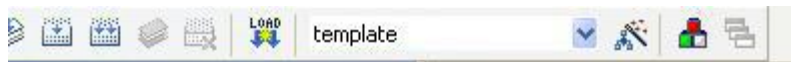


到这里 三个文件夹的源码导入完毕

点 **OK** 即可

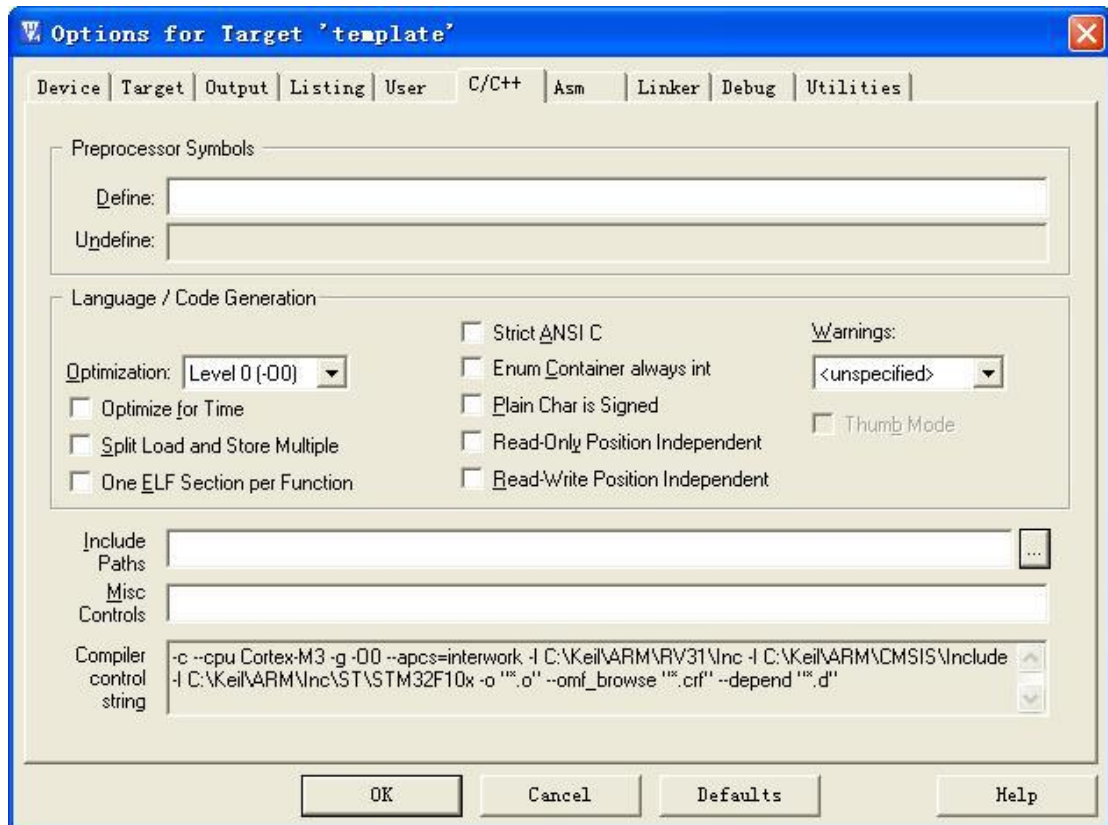


接下来 导入头文件及其他设置



选择



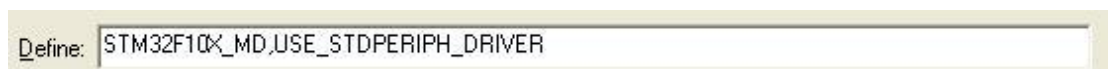


选择 C/C++选项卡

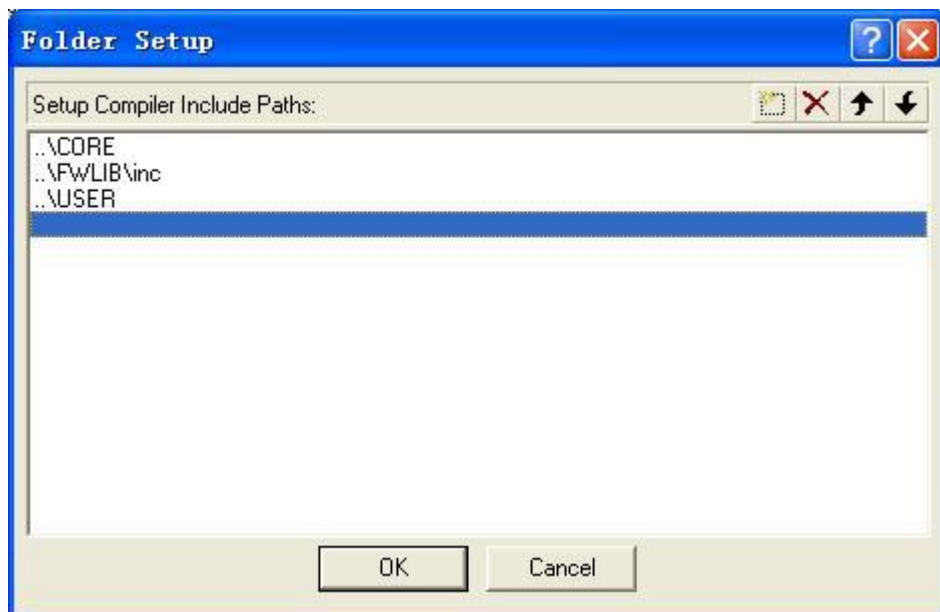
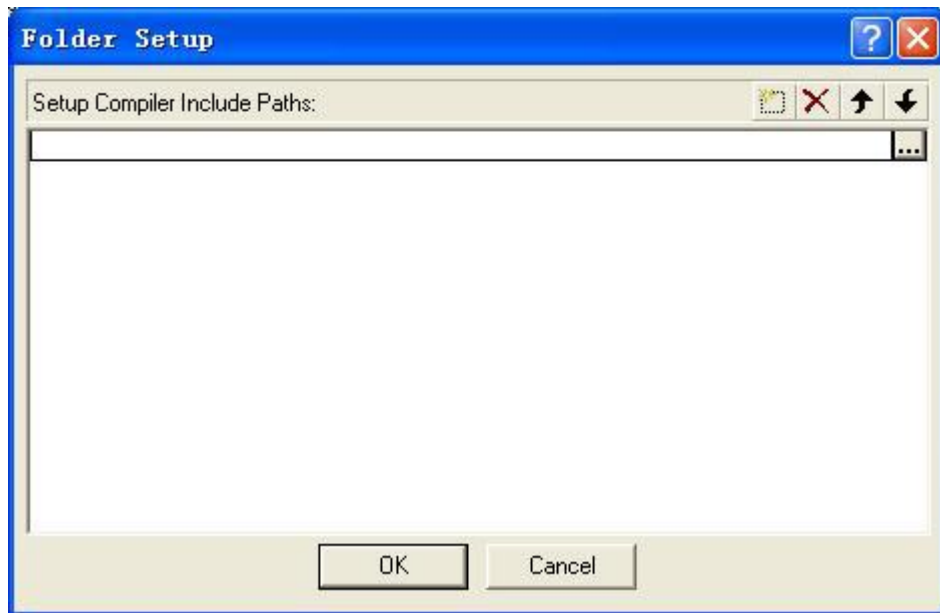


3.5 版本的库函数在配置和选择外设的时候是通过宏定义来选择的 所以这里要配置一个全局的宏定义变量

STM32F10X_MD,USE_STDPERIPH_DRIVER

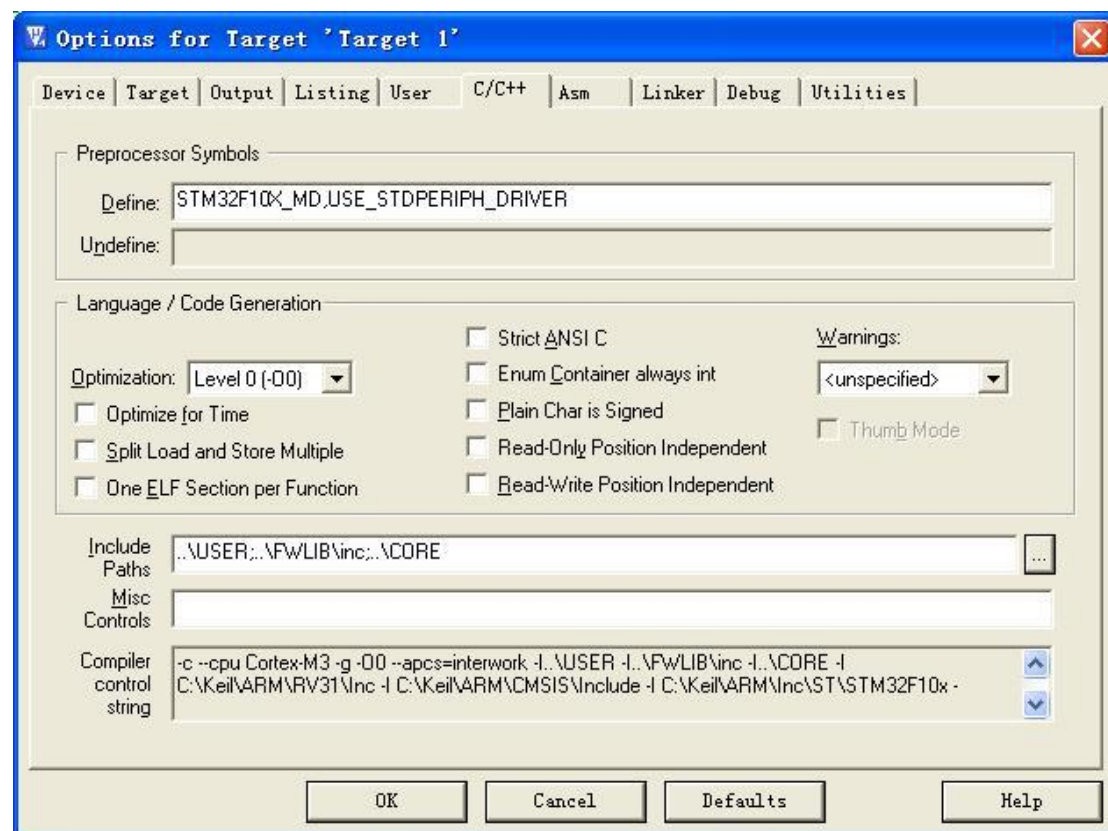


这里添加头文件



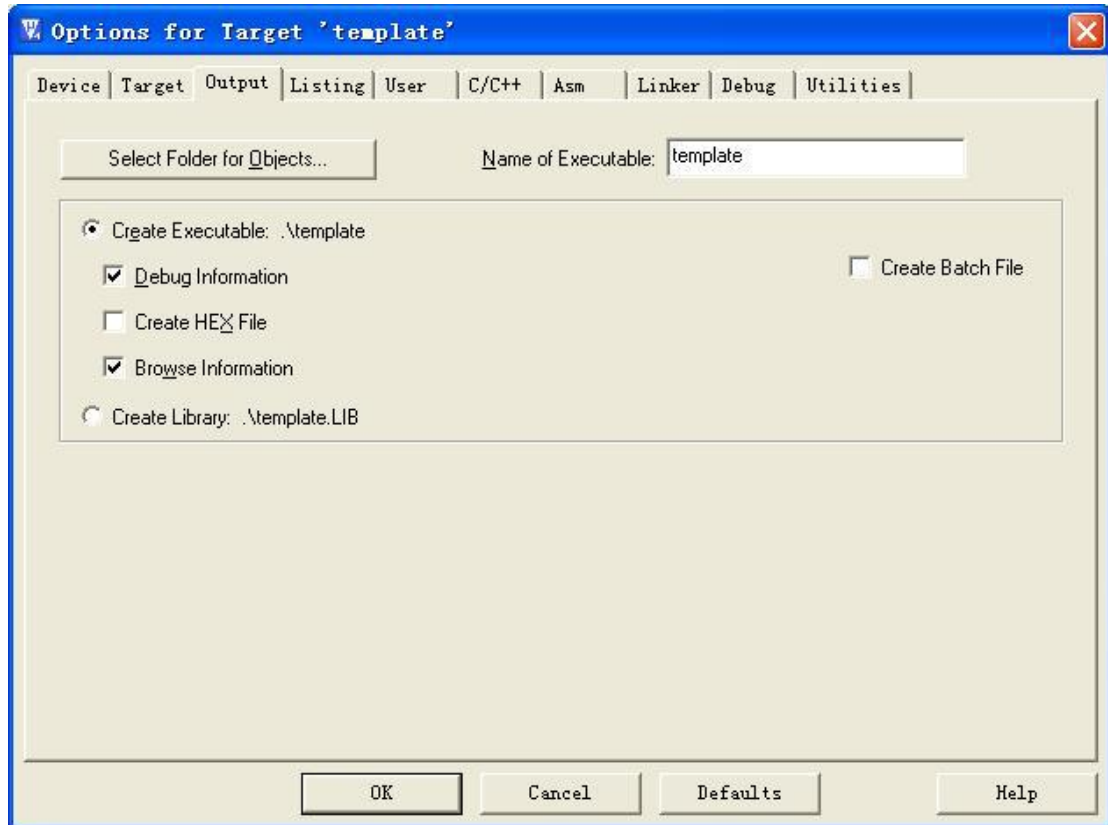
要**强调**的是 **FWLIB** 文件 要选择 **inc** 因为 **inc** 里面的所有文件才是**.h** 源码头文件

(当然 以后应用中如果 创建了新的**.h** 文件 也要在这里导入 才能编译 否则将



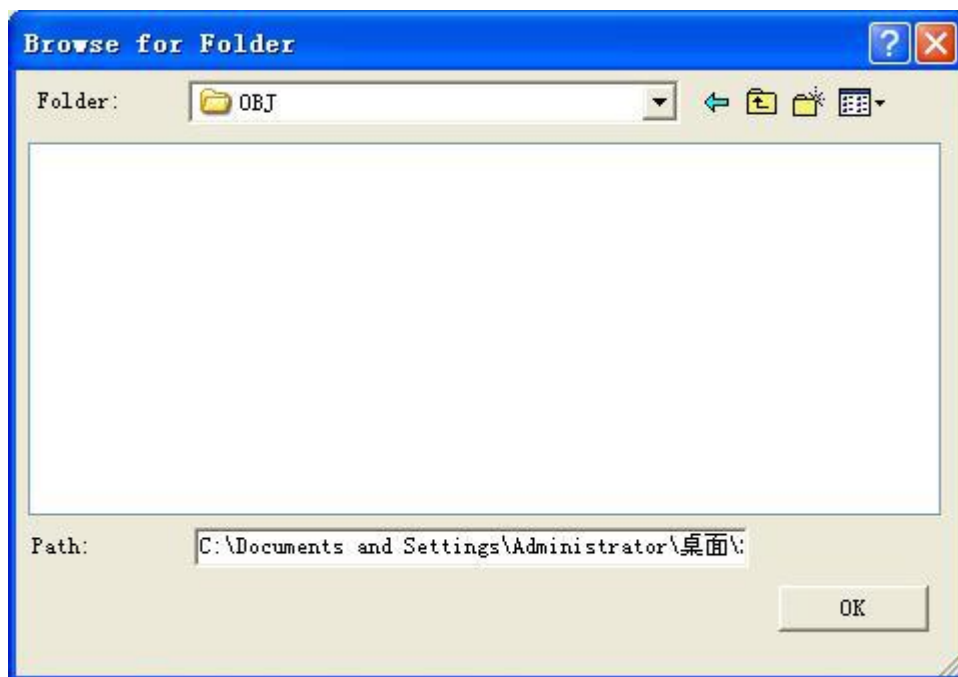
添加好后 点击 **OK**

选择 **Output** 选项卡



点击 

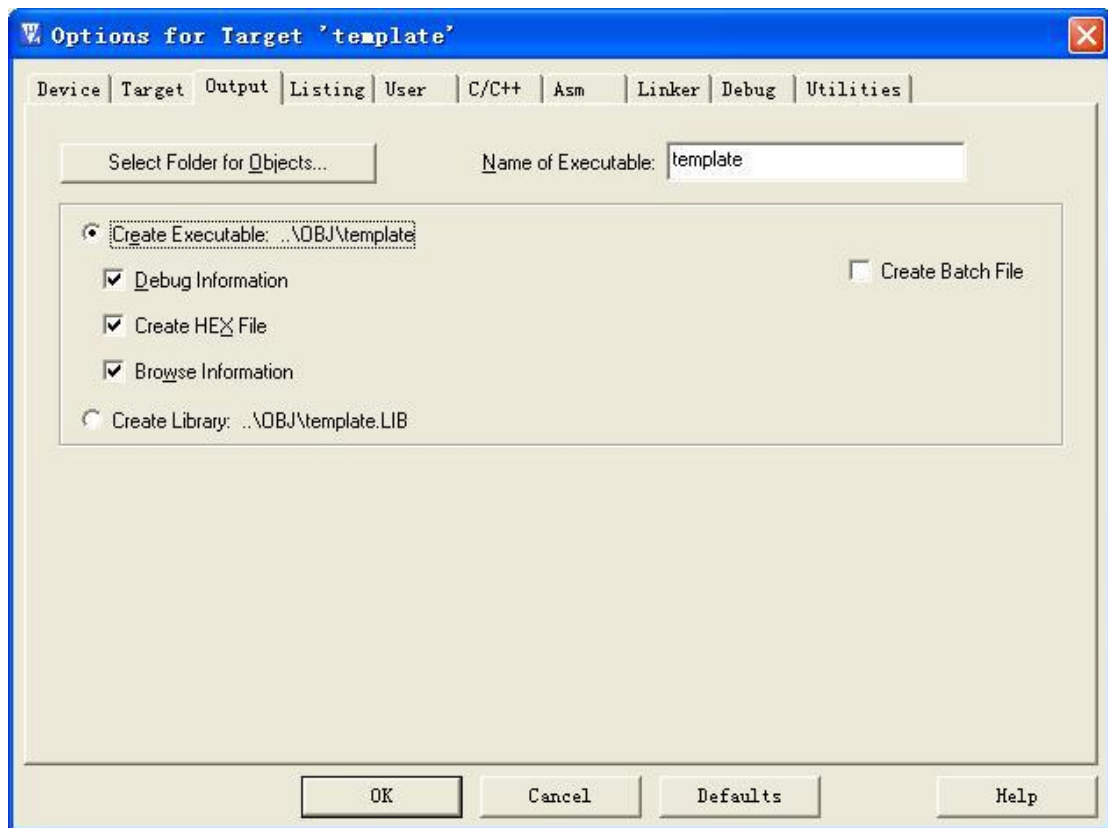
这为编译过程中生成的文件 选择我们创建好的 **OBJ** 文件夹



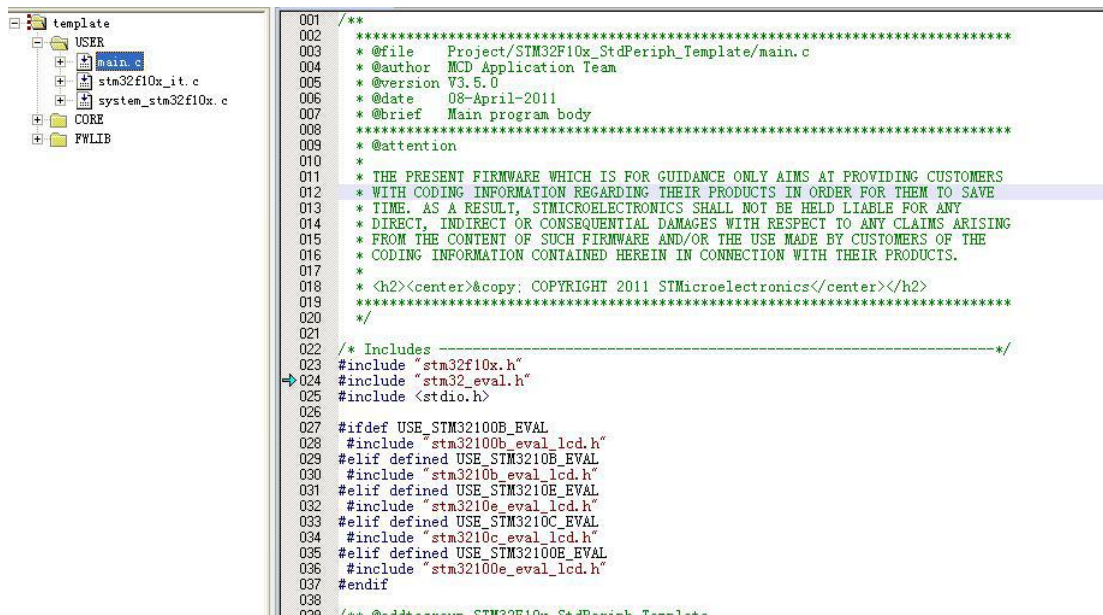
点击 **OK**



这个里为 是否生成 **HEX** 文件 在这里打调（**HEX** 为烧写文件 这个相信大家明白）



这样 整个 这里就都设置完成了 点击 **OK**



双击 **main.c** 将里面所有代码去掉

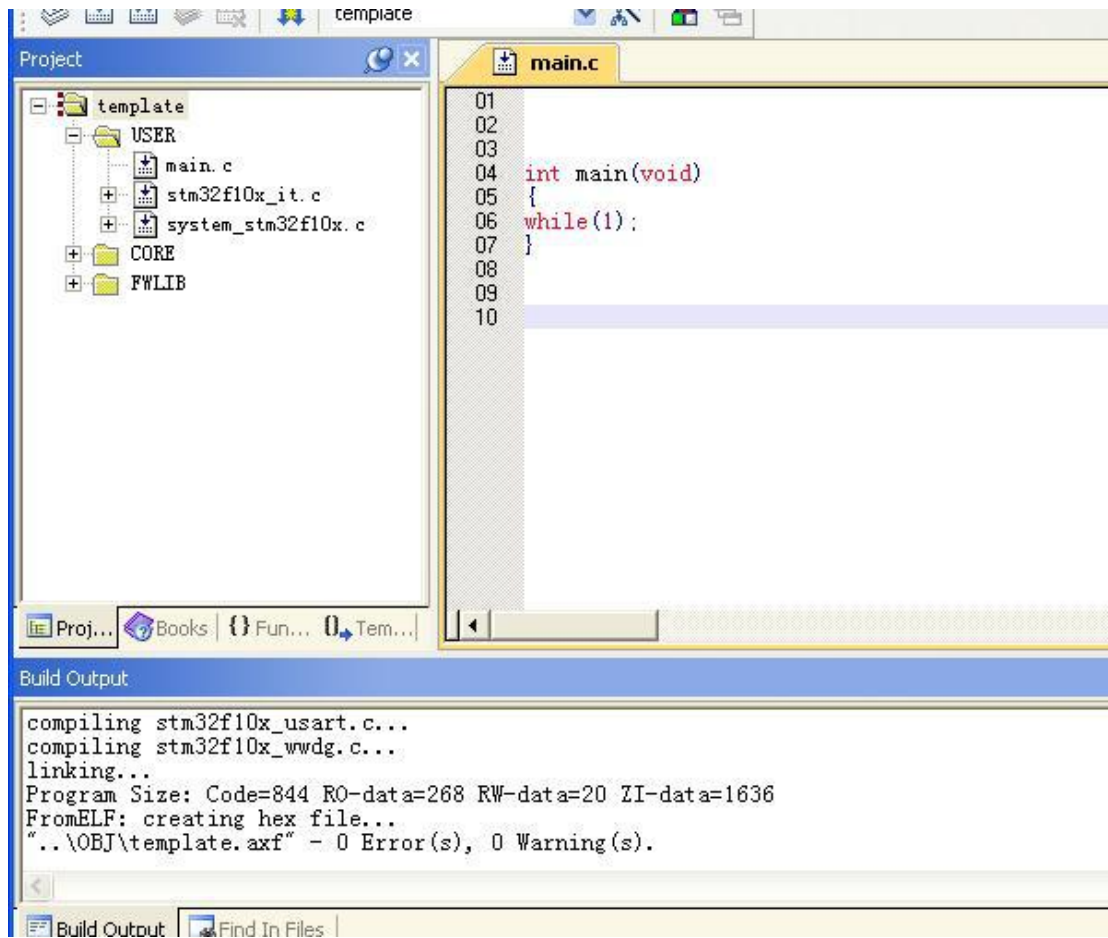
随便写上自己想要写的代码

比如



点击编译





到这里 3.5 库版本的工程 就已经完全建好了 可以在此基础上
任意调用库函数 以实现自己的程序