

基于 STM32 V3.5 库建立 MDK 工程的方法

一、准备文件

创建相关的文件夹，并放置相关的文件。库文件为 **STM32F10x_StdPeriph_Lib_V3.5.0**
在项目工程文件根目录下创建四个文件夹，分别为 Core,Driver,MDK,User.具体如下图所示：

名称	修改日期	类型	大小
Core	2018/1/9,星期二 ...	文件夹	
Driver	2018/1/9,星期二 ...	文件夹	
MDK	2018/1/9,星期二 ...	文件夹	
User	2018/1/9,星期二 ...	文件夹	

各个文件夹下的创建过程如下所示

(1) Core：存放内核文件和启动文件

startup	2018/1/9,星期二 ...	文件夹	
core_cm3.c	2010/6/7,星期一 ...	C Source File	17 KB
core_cm3.h	2011/2/9,星期三 ...	C Header File	84 KB
stm32f10x.h	2011/3/10,星期...	C Header File	620 KB
system_stm32f10x.c	2011/3/10,星期...	C Source File	36 KB
system_stm32f10x.h	2011/3/10,星期...	C Header File	3 KB

文件来源如下：

startup: ...\\STM32F10x_StdPeriph_Lib_V3.5.0\\Libraries\\CMSIS\\CM3\\DeviceSupport\\ST\\STM32F10x 中的 startup 文件夹 **完全拷贝**
stm32f10x.h\\system_stm32f10x.c\\system_stm32f10x.h : ...\\STM32F10x_StdPeriph_Lib_V3.5.0\\Libraries\\CMSIS\\CM3\\DeviceSupport\\ST\\STM32F10x
core_cm3.c\\core_cm3.h : ...\\STM32F10x_StdPeriph_Lib_V3.5.0\\Libraries\\CMSIS\\CM3\\CoreSupport

(2) Driver：存放标准的驱动文件

名称	修改日期	类型	大小
inc	2018/1/9,星期二 ...	文件夹	
src	2018/1/9,星期二 ...	文件夹	

文件来源如下：

inc\\src : ...\\STM32F10x_StdPeriph_Lib_V3.5.0\\Libraries\\STM32F10x_StdPeriph_Driver

(3) MDK：存放工程文件、hex 文件、编译的中间文件

新建 output, list

output: 存放 hex 文件

list: 存放编译的中间文件

(4) User: 存放用户文件

名称	修改日期	类型	大小
 main.c	2018/1/9,星期二 ...	C Source File	1 KB
 stm32f10x_conf.h	2011/4/4,星期一 ...	C Header File	4 KB
 stm32f10x_it.c	2011/4/4,星期一 ...	C Source File	5 KB
 stm32f10x_it.h	2011/4/4,星期一 ...	C Header File	3 KB
 system_stm32f10x.c	2011/4/4,星期一 ...	C Source File	36 KB

文件来源如下:

main.c\stm32f10x_conf.h\stm32f10x_it.c\stm32f10x_it.h\system_stm32f10x.c : ... \STM32F10x_StdPeriph_Lib_V3.5.0\Project\STM32F10x_StdPeriph_Template

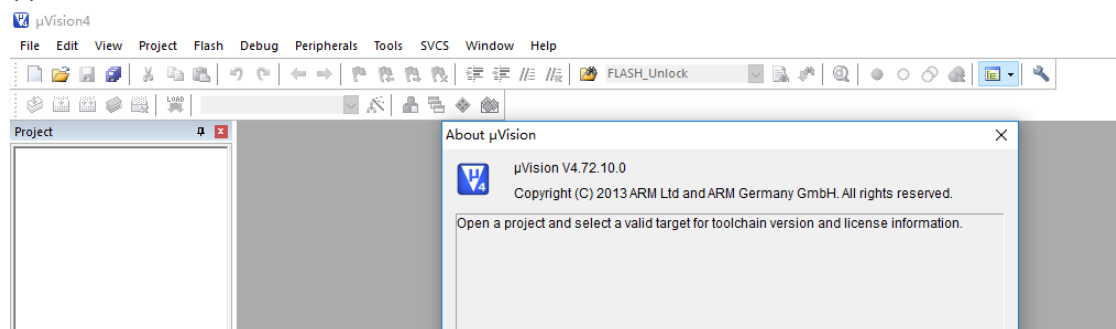
主要需要将 main.c 中的代码删除, 仅保留以下代码:

```
int main(void)
{
    while(1)
    {

    }
}
```

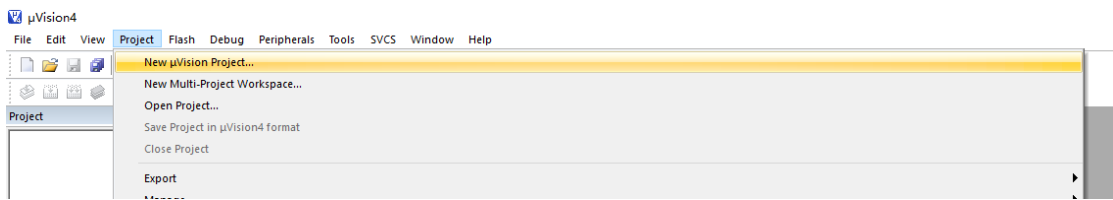
二、新建工程

(1) 启动 keil, 界面如下,版本为 4.72

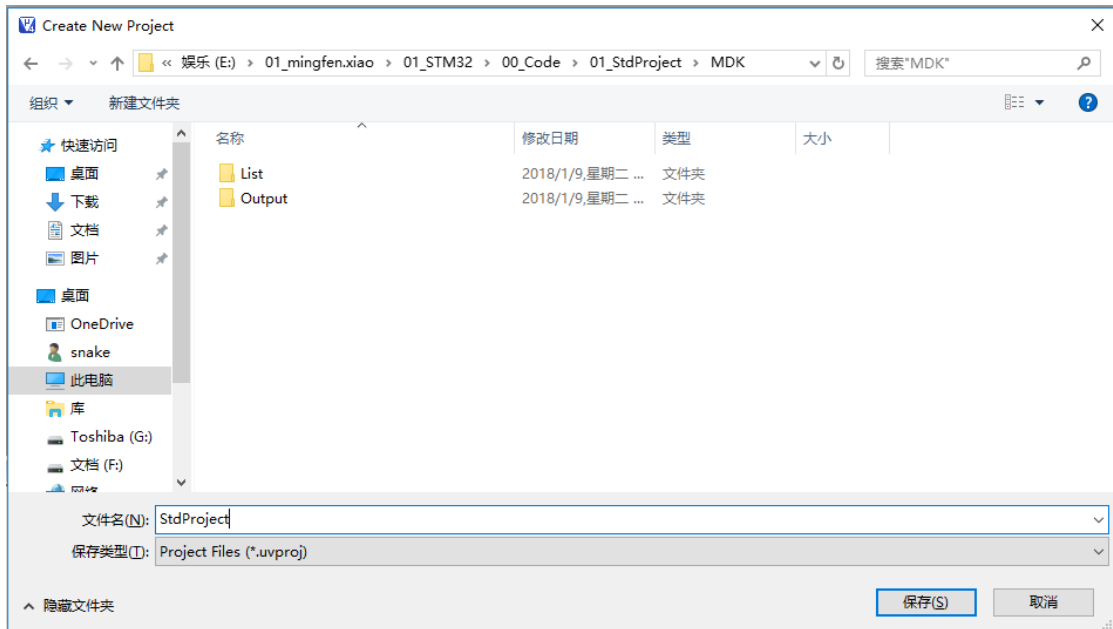


(2) 新建工程

Project->New...

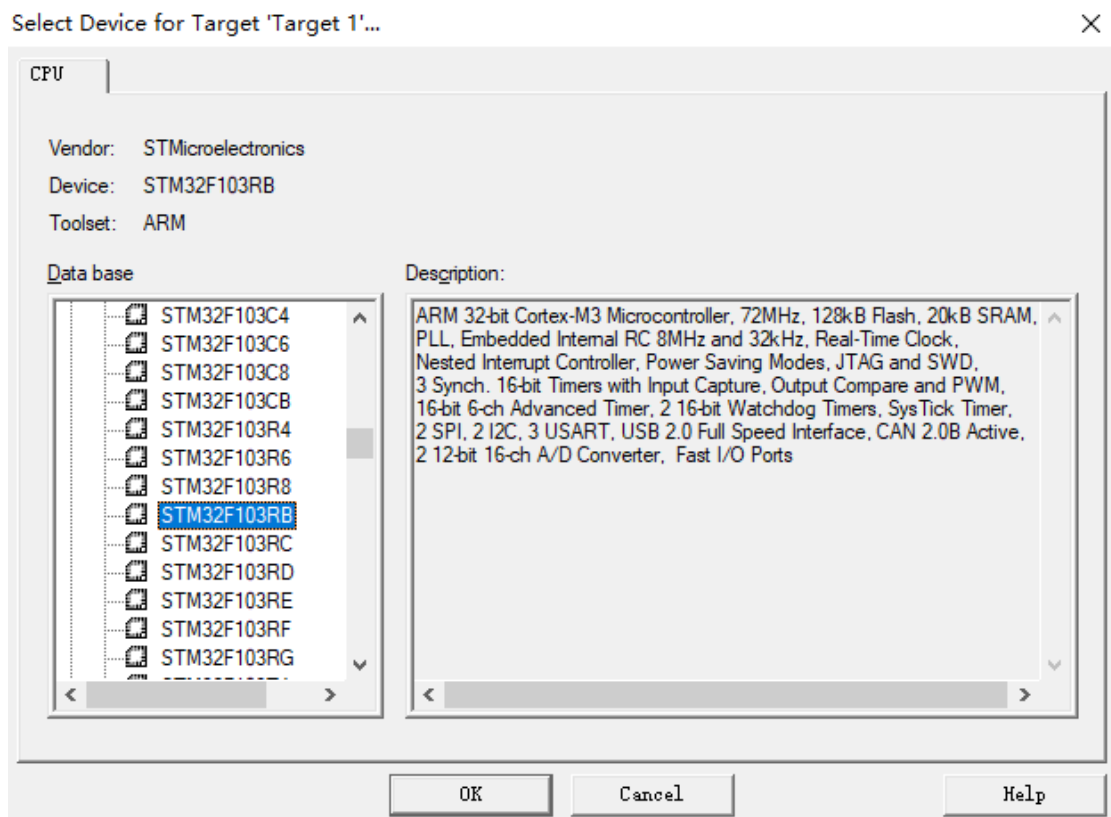


(3) 选择工程路径：MDK，输入工程名



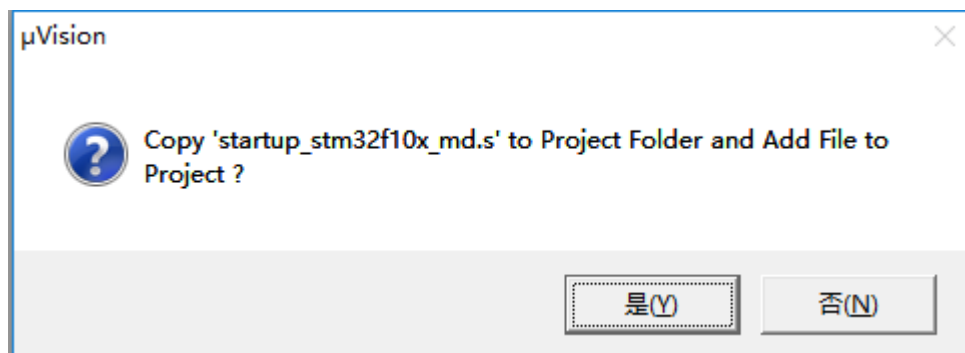
点击“保存”

(4) 选择芯片型号，根据实际的芯片型号选择，此处现在 STM32F103RB



点击“OK”


(5) 启动文件的选择

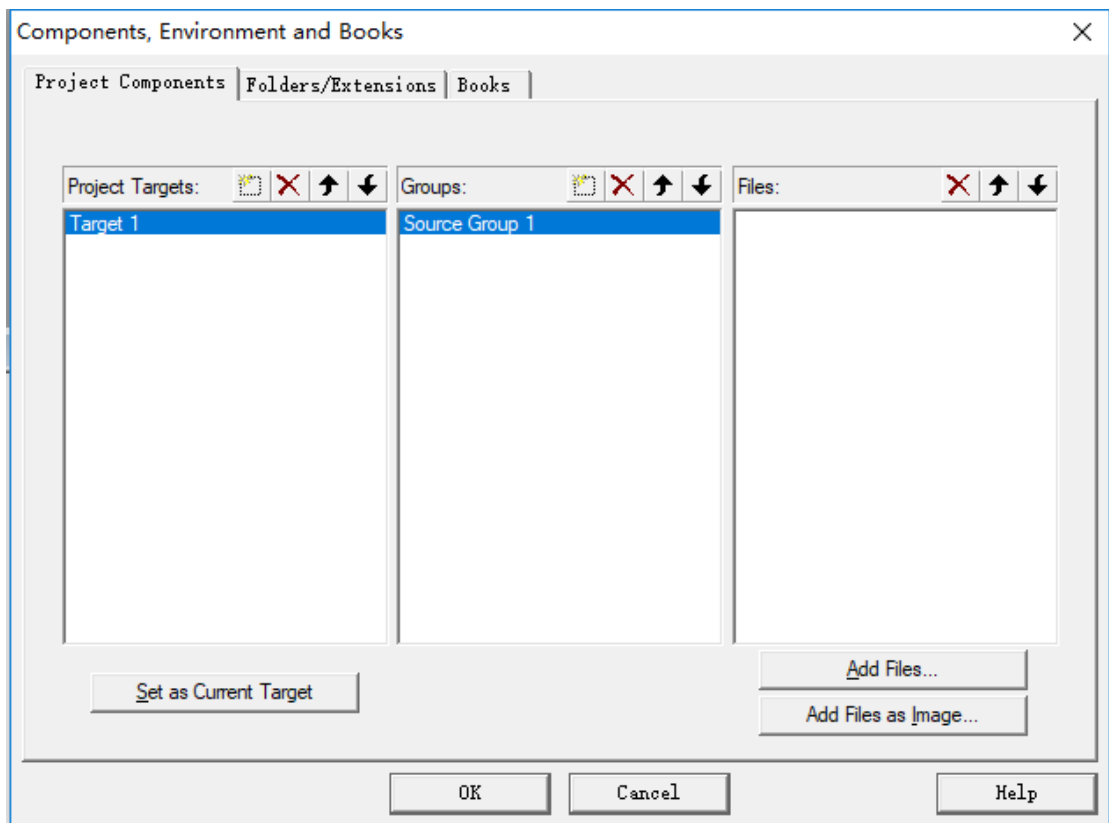


选择“否（N）”，使用标准库中的启动文件，不使用 keil 提供的启动文件

自此，工程已经新建好了。

三、添加文件

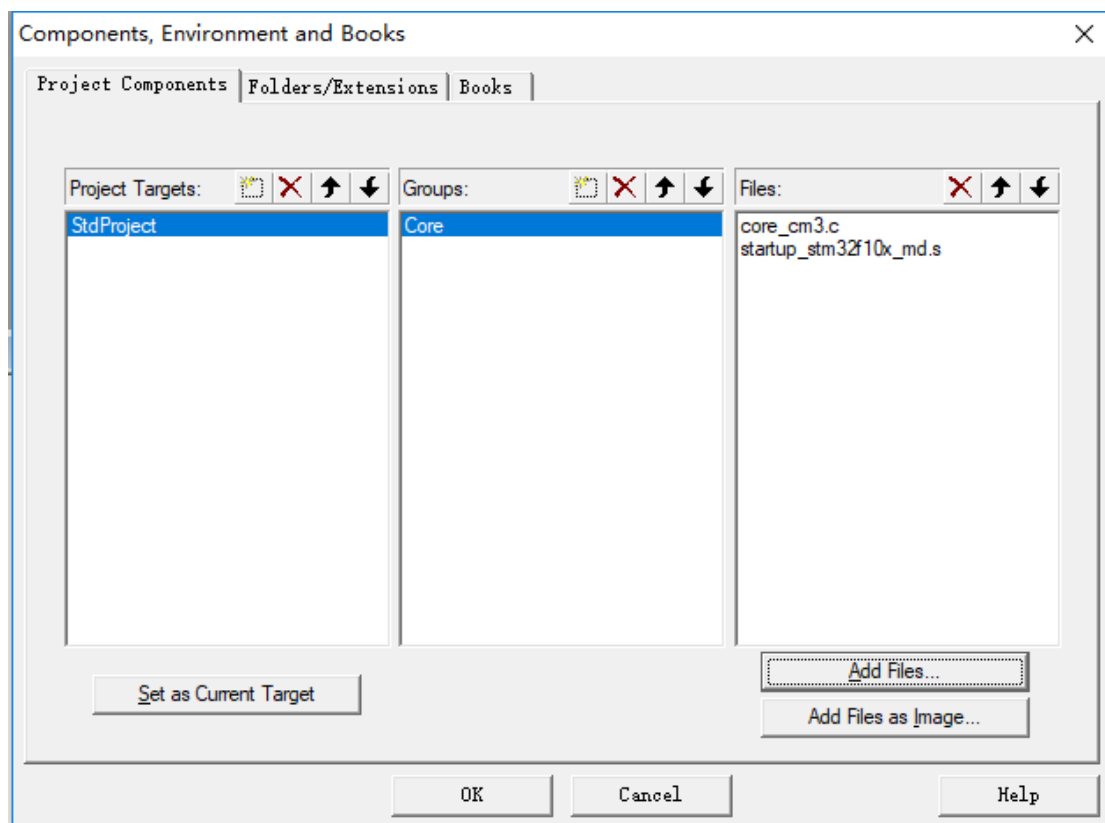
点击  图标向工程中添加文件。



需要添加三个目录，分别为 Core,Driver,User

添加成功之后的截图如下所示：

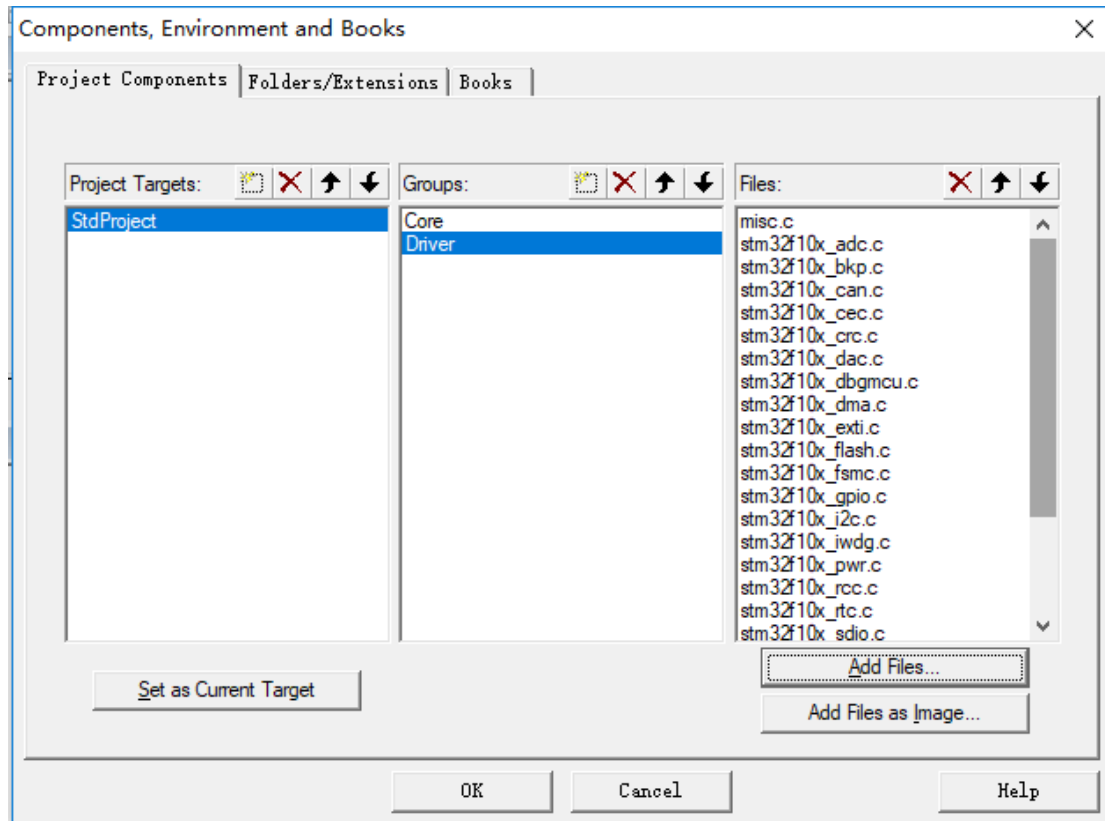
Core:



注意在添加启动文件时需要注意，根据所选的芯片来选择启动文件。

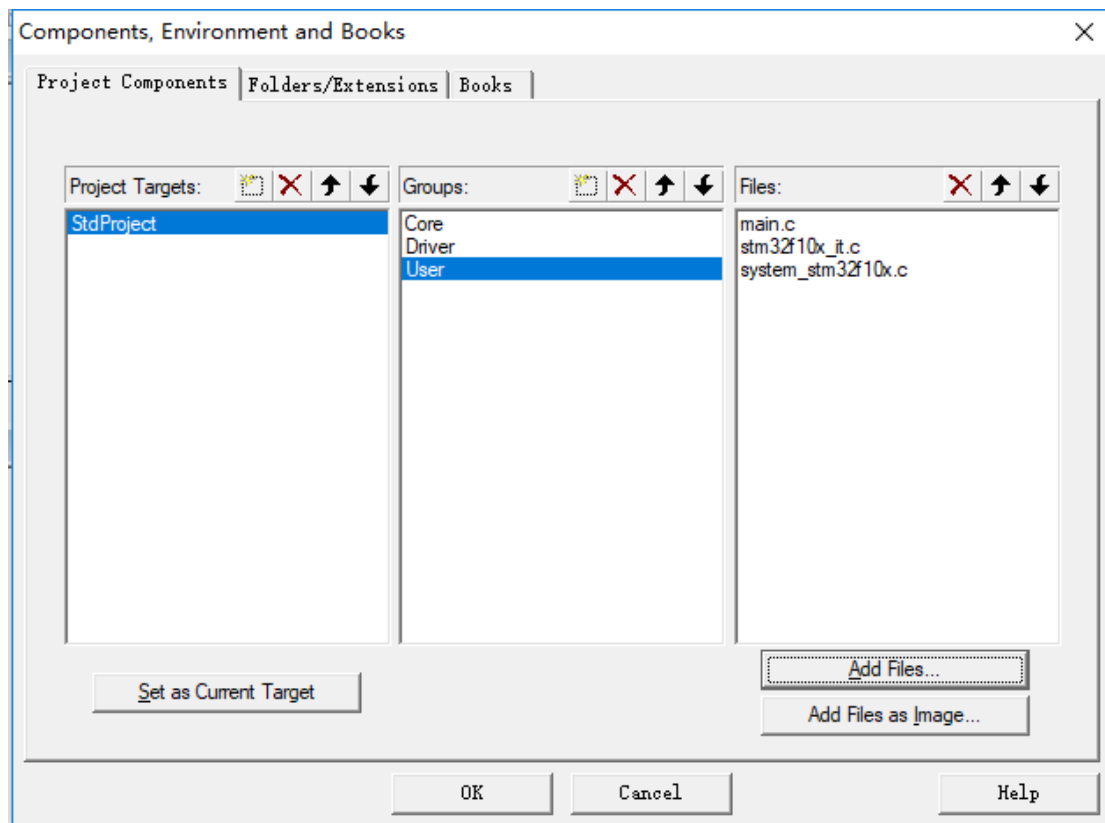
startup_stm32f10x_hd.s 为大容量芯片； startup_stm32f10x_md.s 为中容量芯片； startup_stm32f10x_ld.s 为小容量芯片。芯片的分类可以根据 STM32 给出的资料查询到。

Driver:



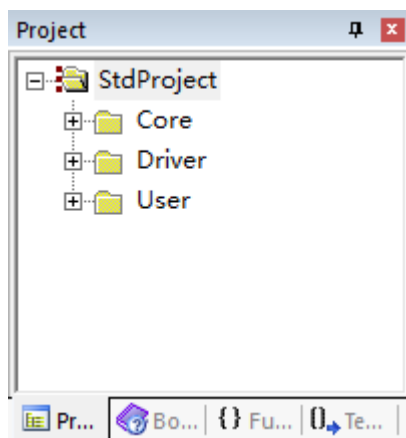
将 Driver/src 中所有的文件添加进去即可。在实际应用的过程中，可以根据所需要的驱动来进行添加，以减小代码量。

User:




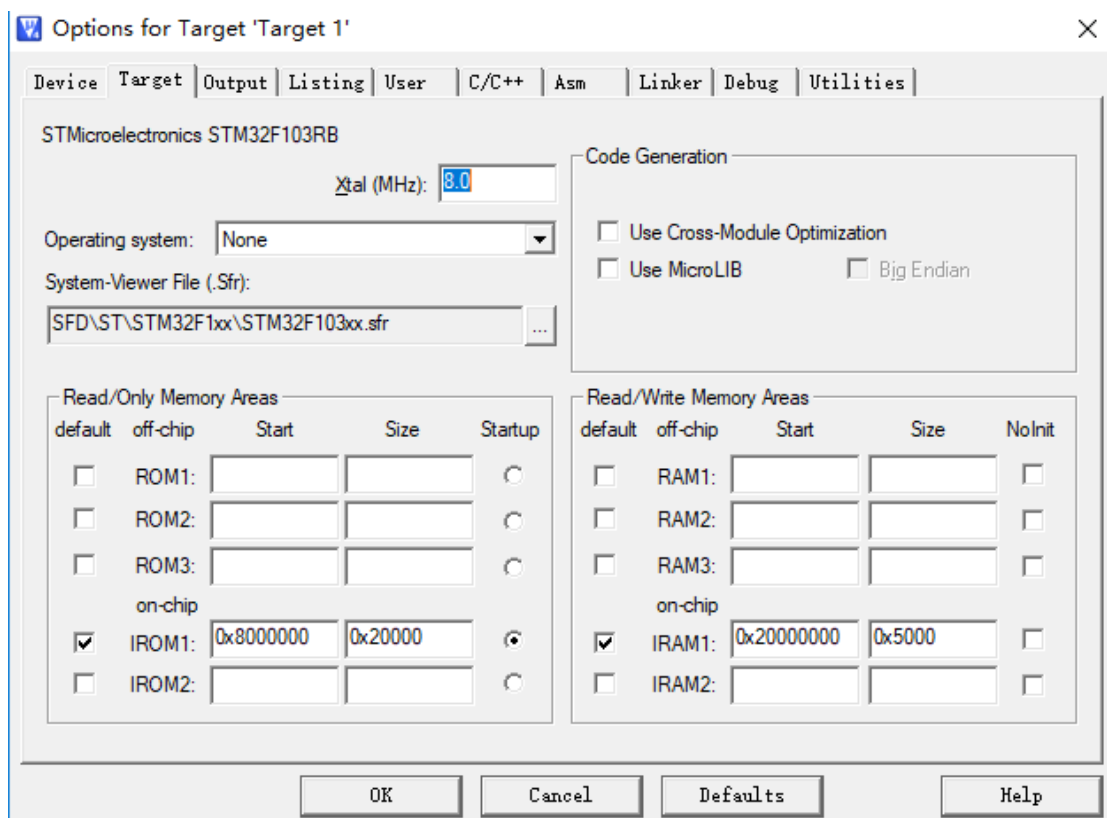
添加完成，点击“OK”。即添加成功。

添加成功，在 Project 窗口中能看到添加的分支和文件



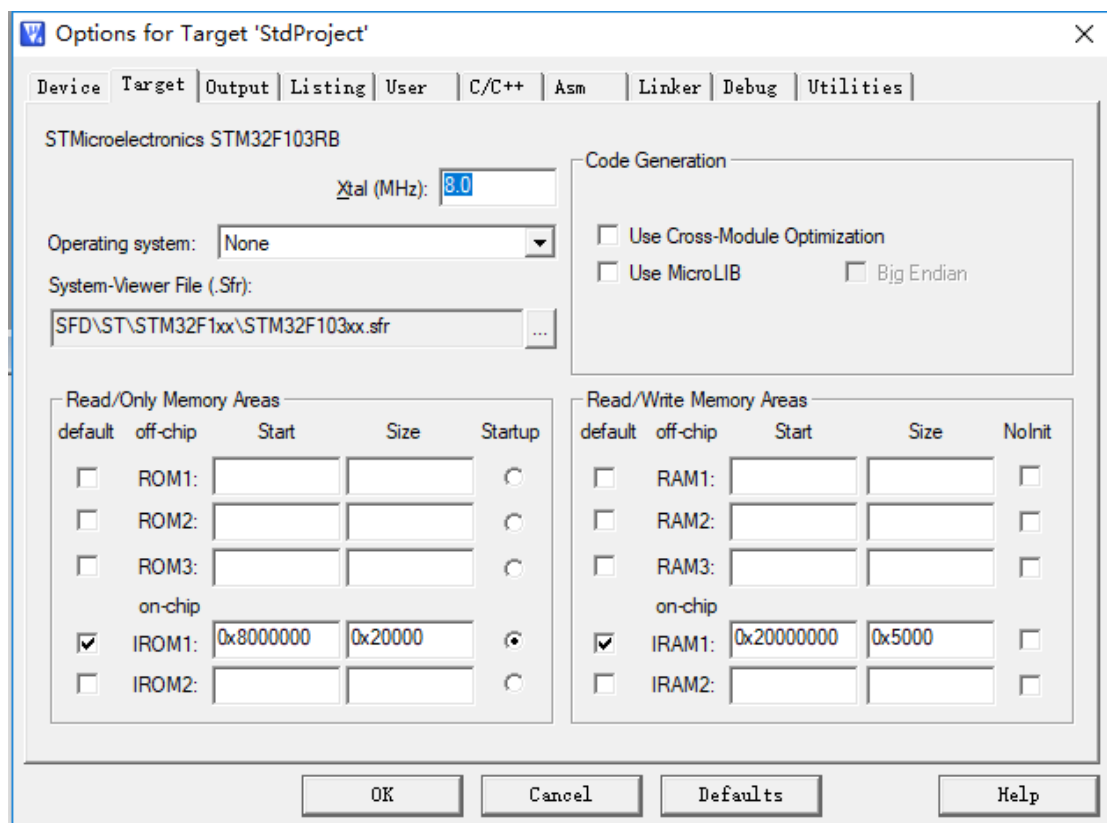
四、配置工程

- (1) 点击  图标，进行工程的配置

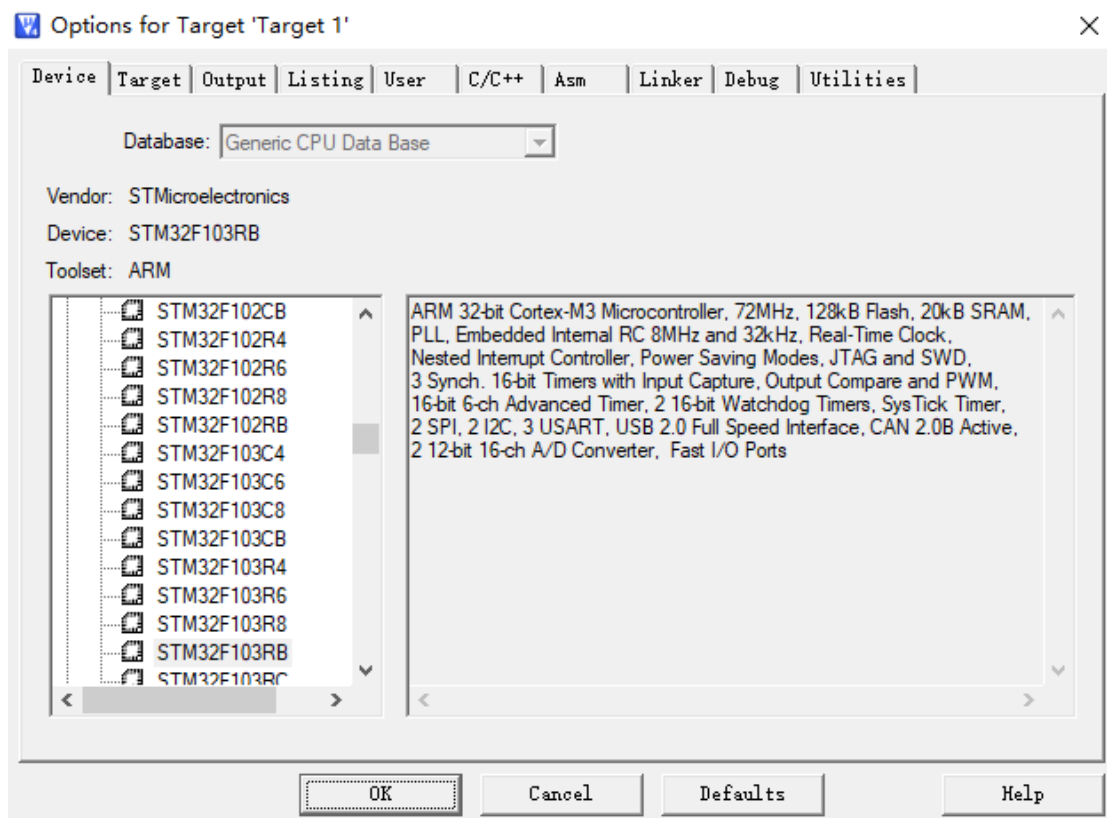


(2) Target 选项卡

主要是设定晶振，一般需要使用仿真的时候，才比较重要，如果不需要仿真的话，可以不用理会。



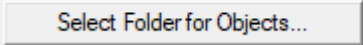
(3) Device 选项卡



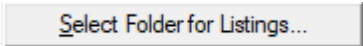
如果芯片型号选择错误，可以在此选项卡中进行配置。此处不需要进行配置。

(4) Output 选项卡

选中“Create HEX File”复选框

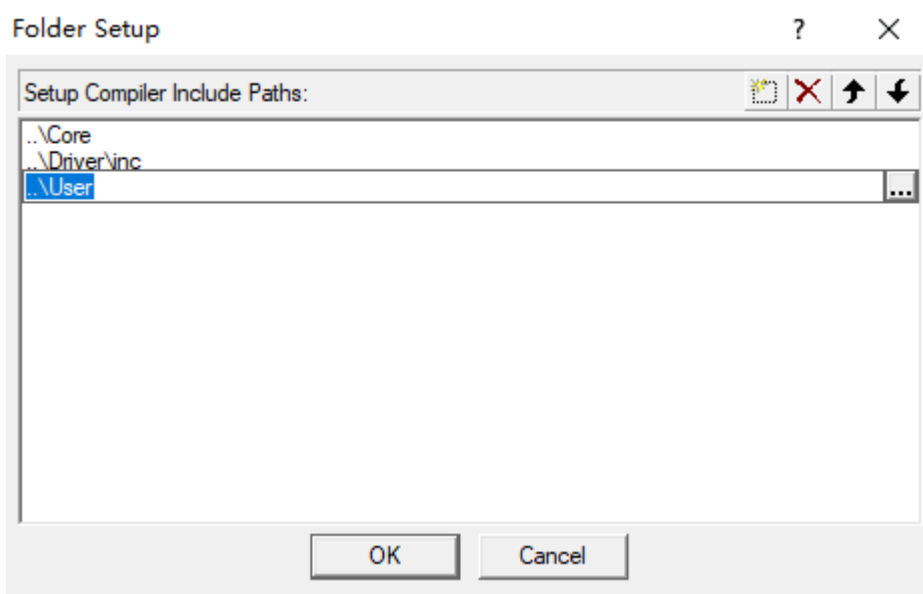
点击 ，进行设定 hex 文件的路径，将路径设定为 MDK/Output

(5) Listing 选项卡

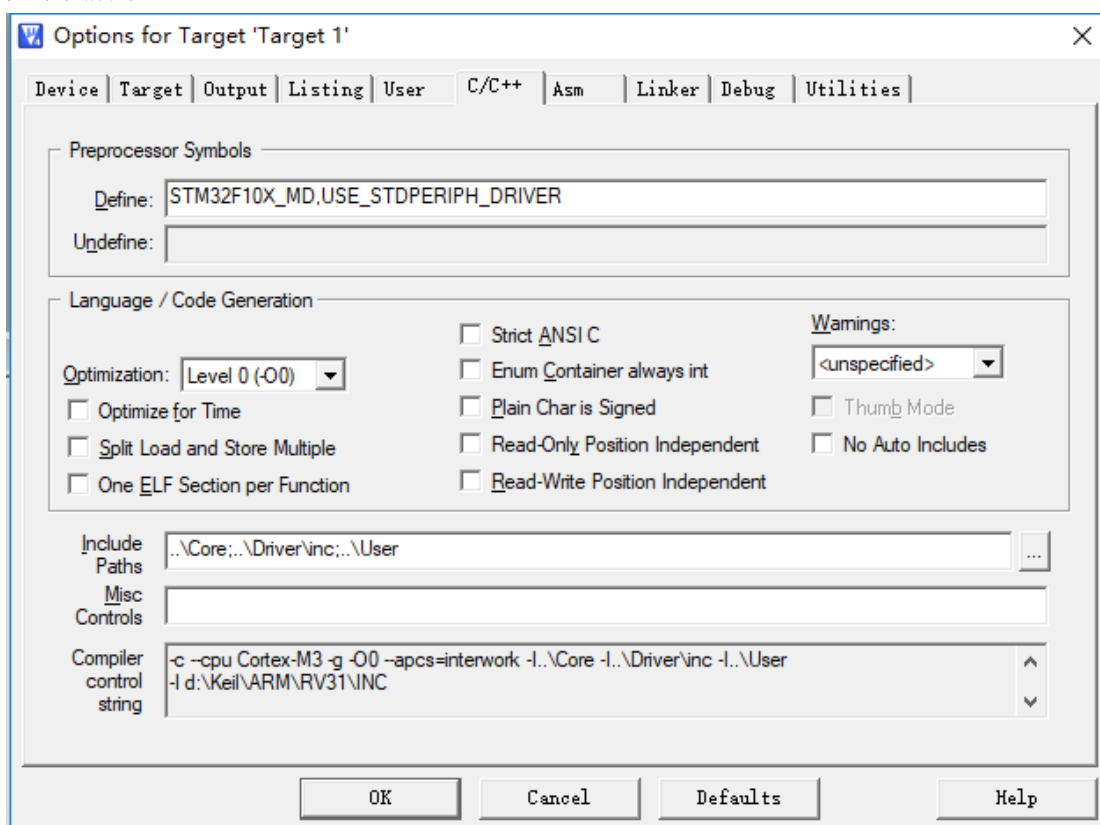
在此选项卡中，点击 ，进行 list 文件路径的设定、将路径设定为 MDK/List

(6) C/C++选项卡

在 Include Paths 中设定头文件的路径，主要是 Core、Driver\inc，User。路径添加完整之后，如图所示



声明 Symbols, 在 Define 输入以下内容 STM32F10X_MD,USE_STDPERIPH_DRIVER。
效果如下图所示



注意, 在此也需要根据芯片型号将 MD 或者其他的标识, HD,LD 等等。

(7) Debug 选项卡和 Utilities 选项卡

这两个选项卡主要是设定仿真工具的, 根据实际情况设定即可。

五、编译

编程成功。

Build Output

```
compiling stm32f10x_dma.c...
compiling stm32f10x_exti.c...
compiling stm32f10x_flash.c...
compiling stm32f10x_fsmc.c...
compiling stm32f10x_gpio.c...
compiling stm32f10x_i2c.c...
compiling stm32f10x_iwdg.c...
compiling stm32f10x_pwr.c...
compiling stm32f10x_rcc.c...
compiling stm32f10x_rtc.c...
compiling stm32f10x_sdio.c...
compiling stm32f10x_spi.c...
compiling stm32f10x_tim.c...
compiling stm32f10x_usart.c...
compiling stm32f10x_wwdg.c...
compiling main.c...
compiling stm32f10x_it.c...
compiling system_stm32f10x.c...
linking...
Program Size: Code=844 RO-data=268 RW-data=20 ZI-data=1636
FromELF: creating hex file...
".\Output\StdProject.axf" - 0 Errors, 0 Warning(s).
```