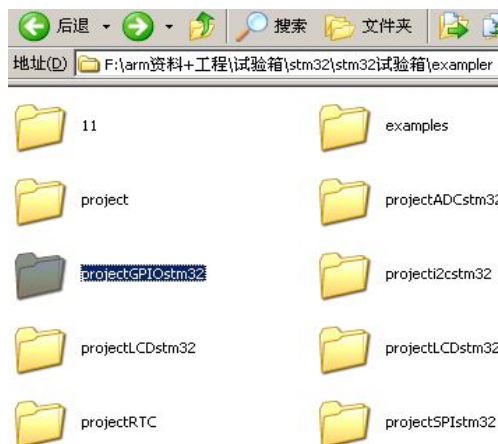
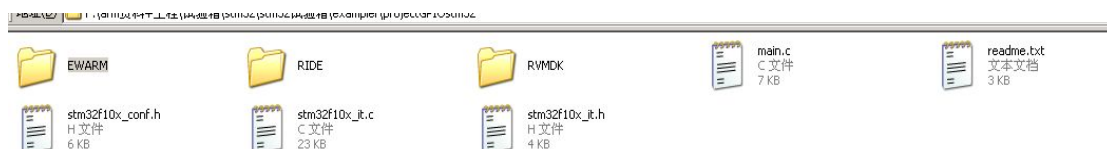


## 一 工程的建立

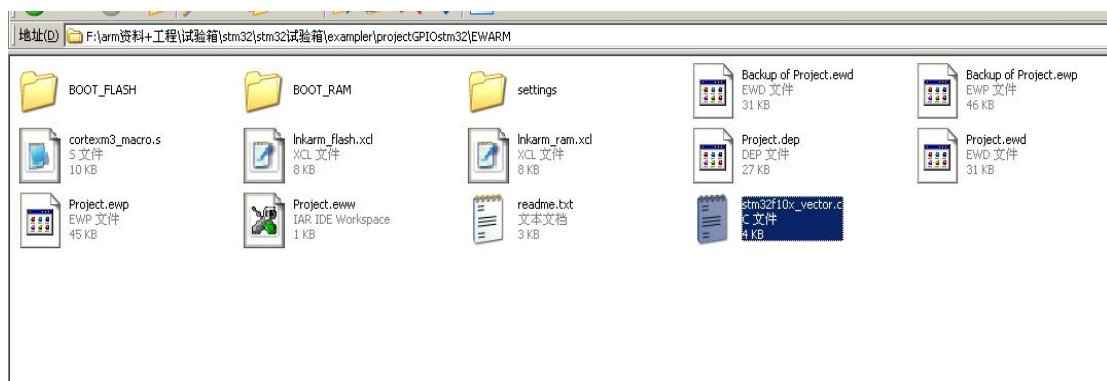
我们先来回忆一下我们熟悉的 IAR 编译软件工程的结构, 请看下图我们以 gpio 的工程文件为参考:



打开该文件后如图, 几个文本文档是我们写的程序。



IAR---使用 ewarm 环境, 所以建了一些文件夹, 为了更清晰此文件夹下为工程文件和几个我们需要使用但不用更改的文件。



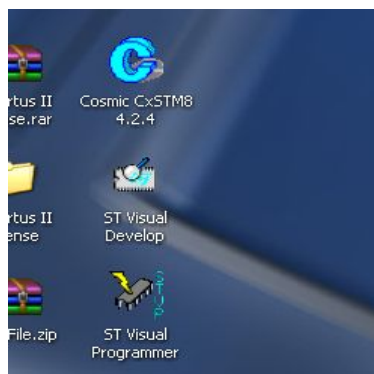
仿真----stm8 的 stvd 编译器工程文件结构和此相似, 当然具体什么文件放在哪也是按照自己意愿来, 但是做为初涉该软件的人, 我们还是按照规矩一点的来。首先我们先按装必备的软件



Sttoolset.exe 是编译软件安装程序，按照提示即可安装完成

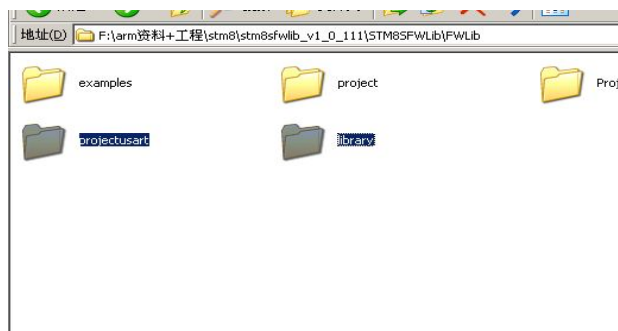
Cosmic stm8 是第三方编译软件，和上面的软件联合使用，用来编译我们写的程序。

安装完后图标如图：我们只使用 st visual develop 来建立工程



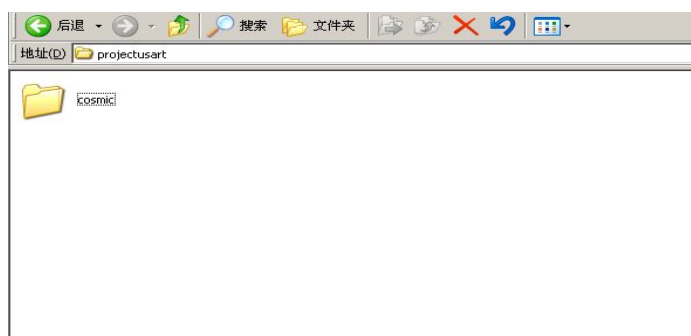
好现在我们来建立工程，首先我们在电脑里建立几个文件，用来放我们将要建立的工程文件：

我们建立一个 stm8uart 工程 如图



Library 是公用的文件 和 32 一样

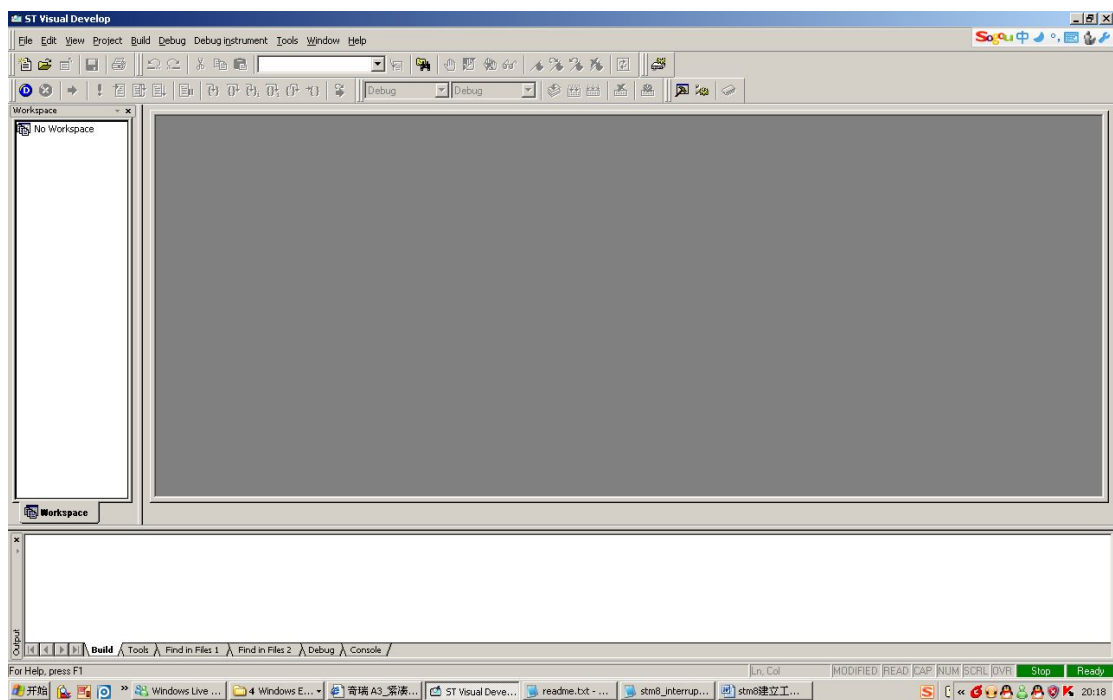
在 projectusart 文件夹下建立 cosmic 文件夹 因为我们使用 cosmic 编译器



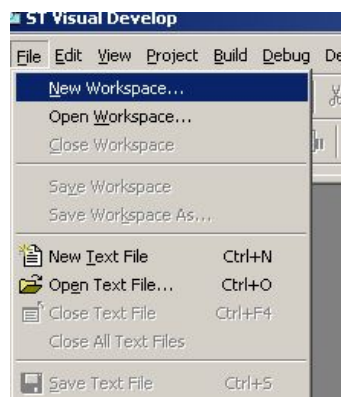
Ok 下面开始建立工程： 点击桌面的图标 如图：



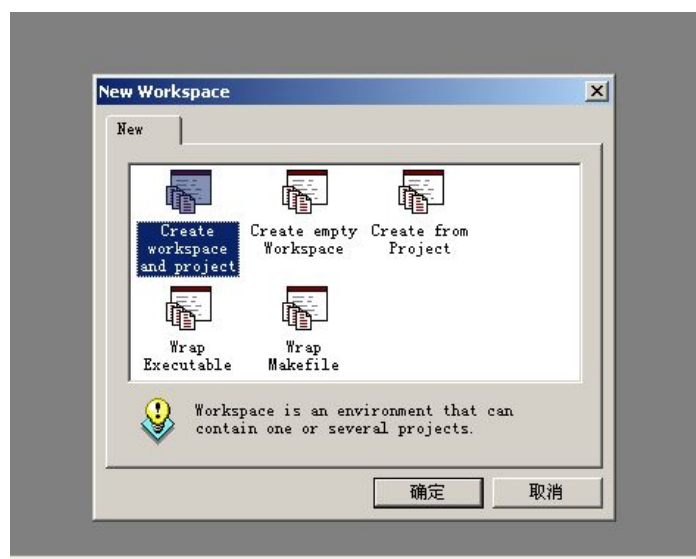
打开后：



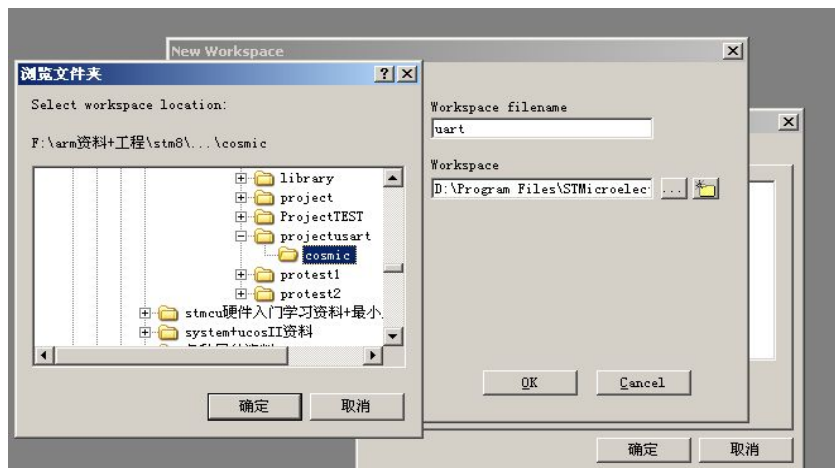
File---new workspace



出现如图



选择阴影部分 如图 工程环境我们输入 **uart** 路径我们选择我们上面建立的 **cosmic** 路径

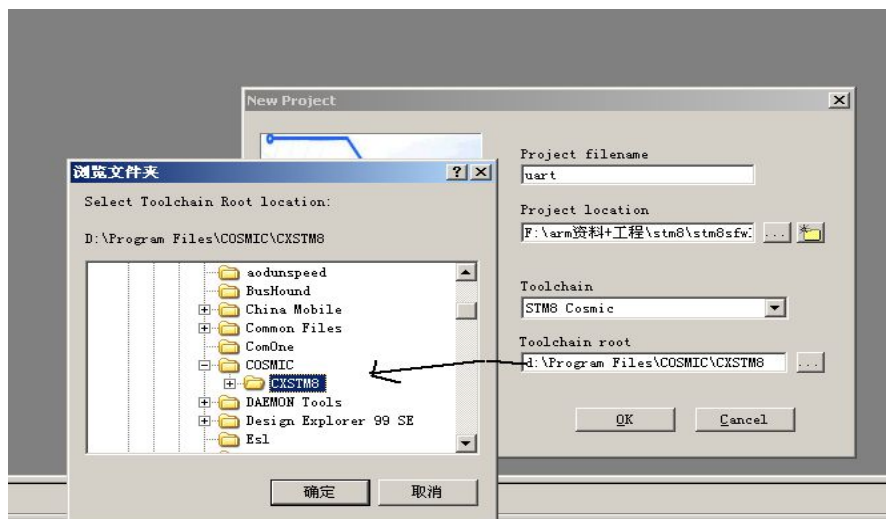


点击 **ok** 工程文件 我们仍输入 **uart** 当然可以不一样

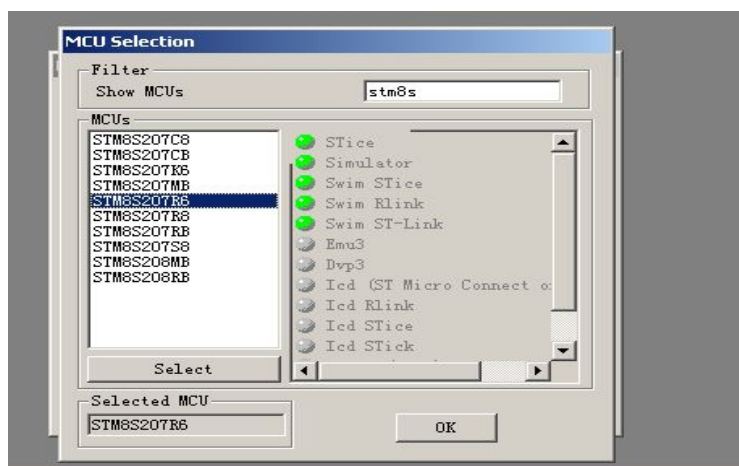
**project location** 和上面一样

**toolchain** 如图所示

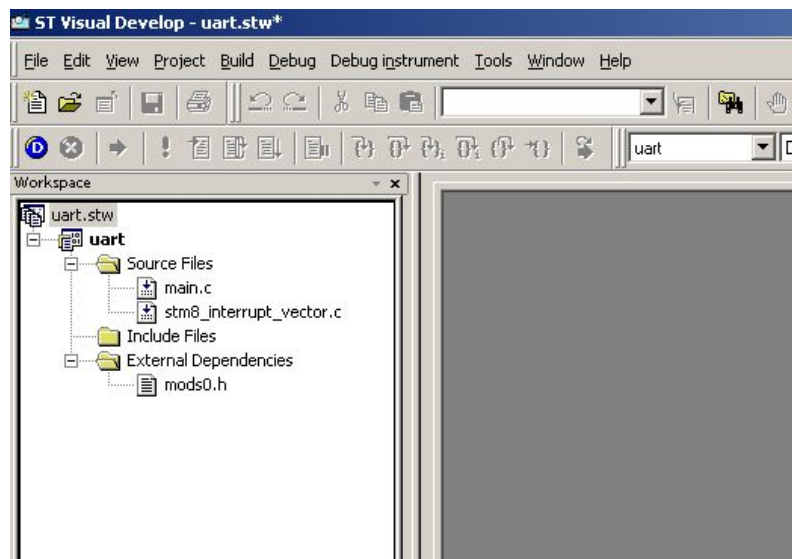
**toolchain root** 选择我们安装的 **cosmicstm8** 编译软件的路径 找到 **cxstm8** 处 如图



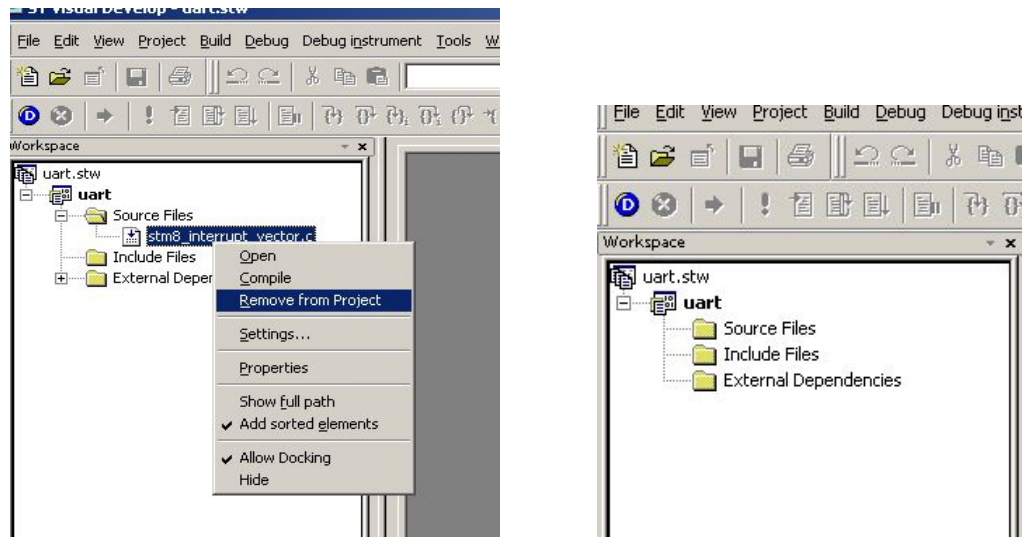
点击 **ok** 选择我们要使用的芯片 **show MCUs** 里面可以方便我们关键字搜索



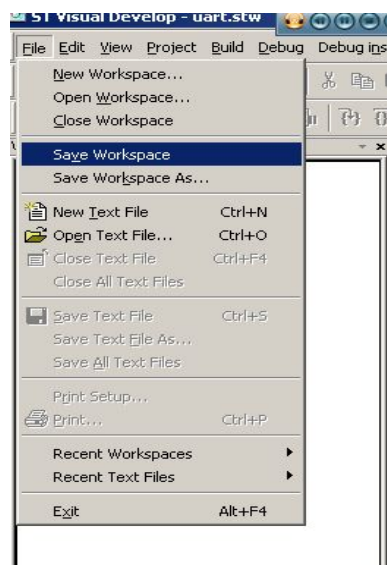
点击 ok 界面如图



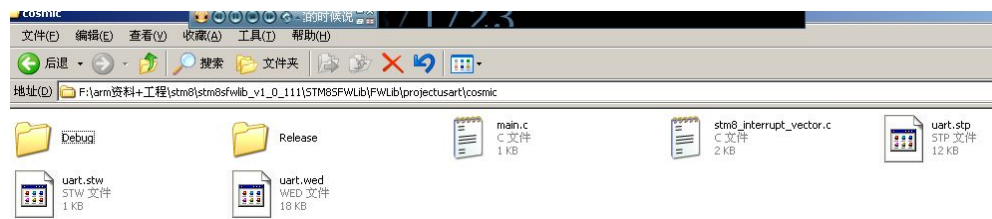
我们把里面的 main.c 和....cs 删除 因为我们这里不自己编程序 而是使用已有的例程



保存工程



然后我们反过去看一下 电脑上工程文件夹里面

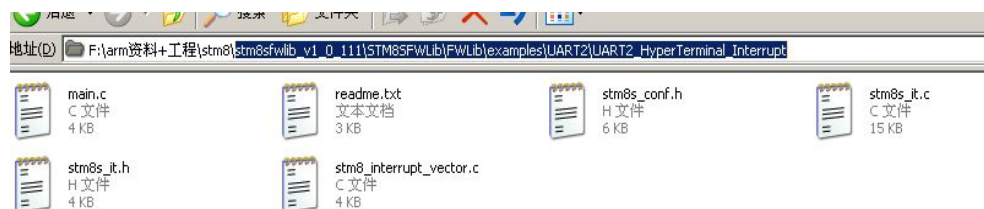


删除两个.c 文件

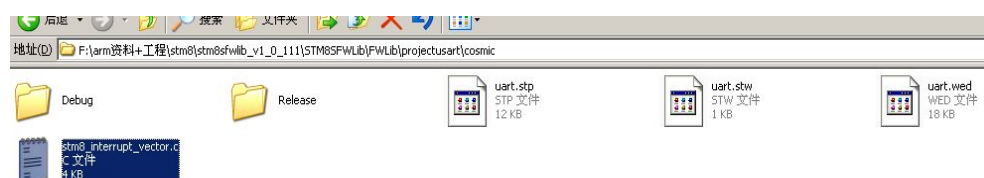


好,我们现在先向文件夹里添加 例程中的文件:

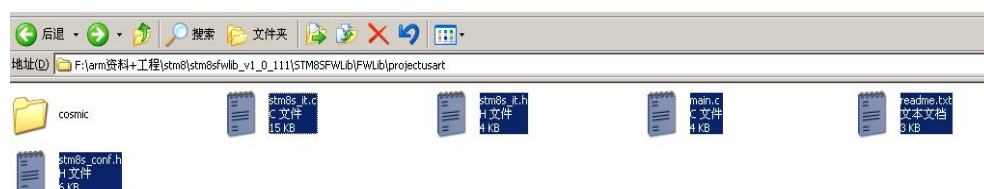
打开 usrat 例程文件



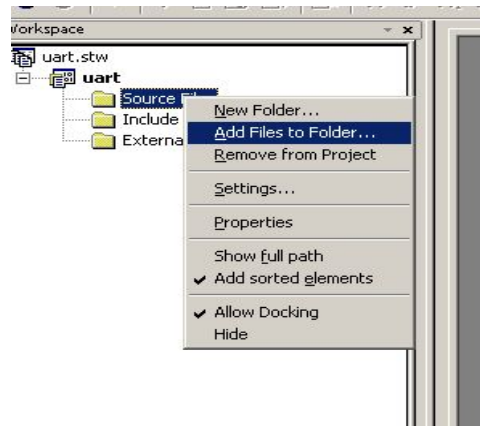
我们把 上面 vector.c 文件添加到 cosmic 文件夹下



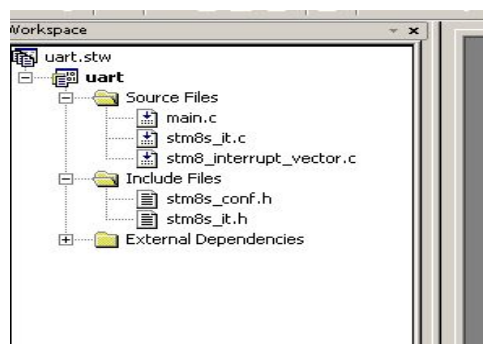
剩下的加到 工程文件 projectusart 根文件下:



好了,现在我们把 .c 和.h 文件分别添加到工程:



如图：

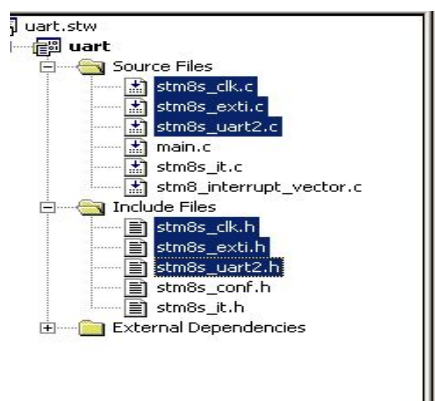


下面我们添加库文件里我们要用到的 .c 和.h 文件,和 stm32 一样  
看工程下的 readme.txt 文件

```
In order to make the program work, you must do the following :
- Create a project and setup all project configuration
- Compile the directory content files and required library files :
+ stm8s_clk.c
+ stm8s_exti.c
+ stm8s_uart2.c
- Rebuild all files: Project->Rebuild all.
- Load project image: Debug->Start/Stop Debug Session
- Run program: Debug->Run (F5)
```

Go to main file: UART2\_HyperTerminal\_Interrupt\main.c

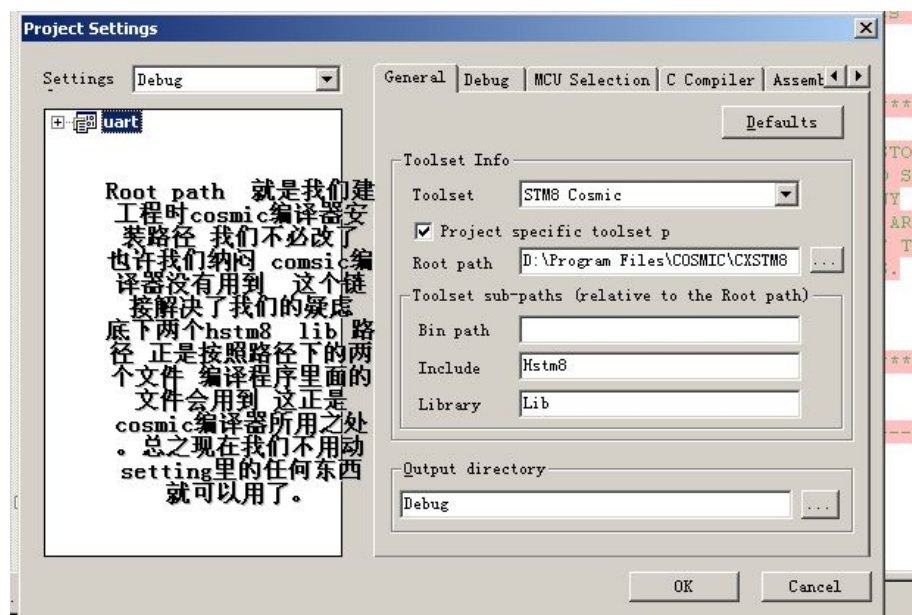
在 library 找到这些文件加入添加方法如上,当然我们也可以另加文件,把这些不用我们编写只供调用的文件放在这个文件下,在这里我们先不这样做,直接加到一起如图：



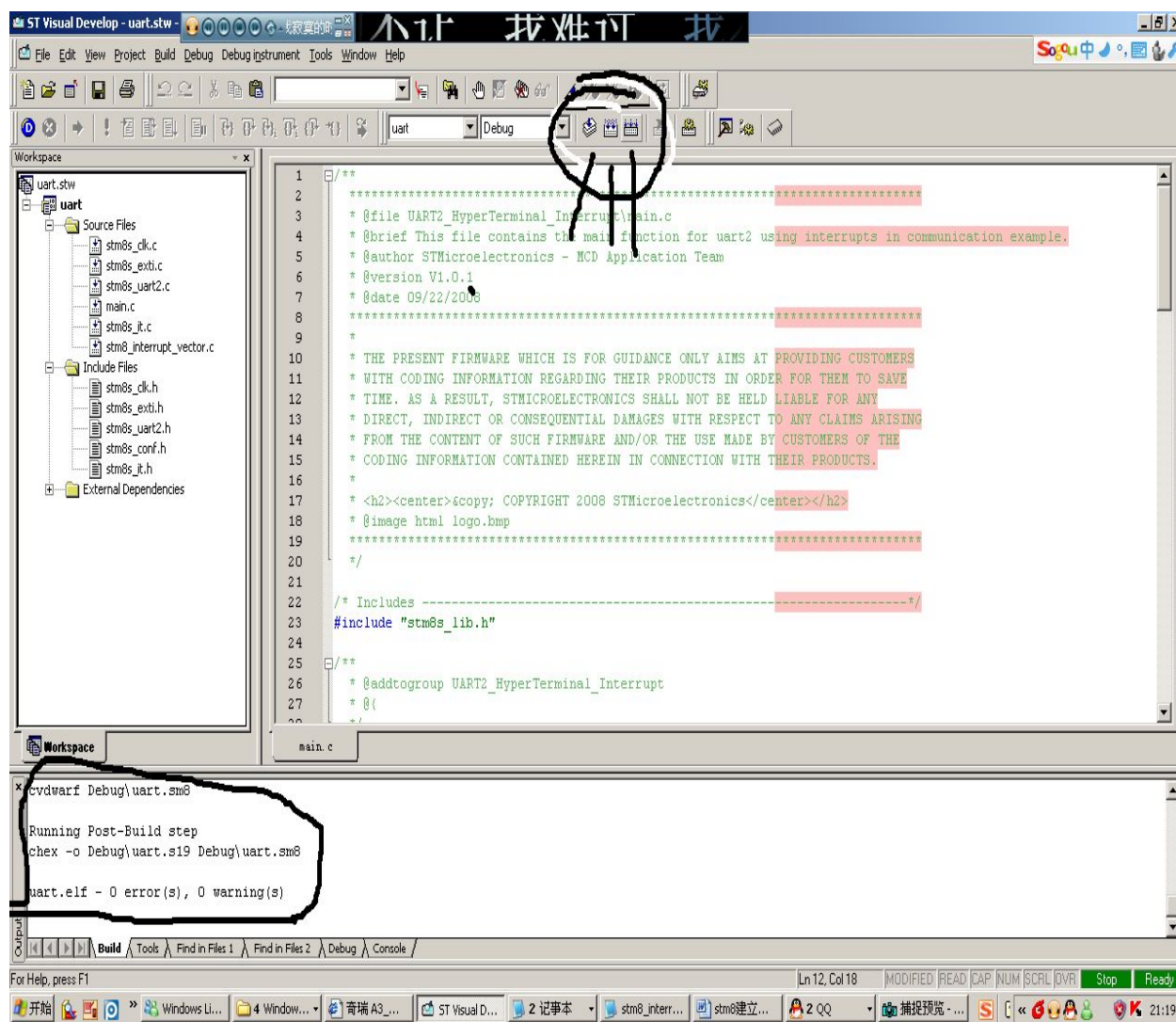
Ok 配置我们简单看一下,这里我们不需要设置,就可以编译了,当然当我熟练了,里面有些客户需要



还是要改,那是后话

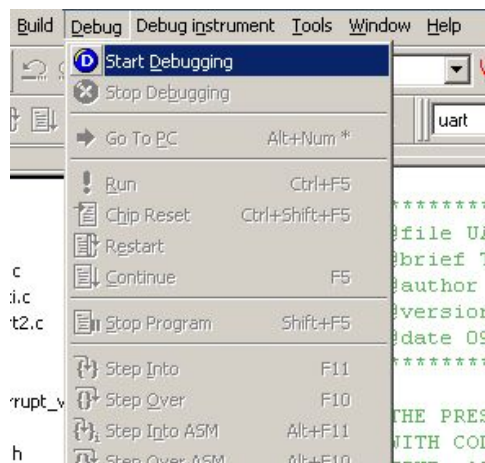
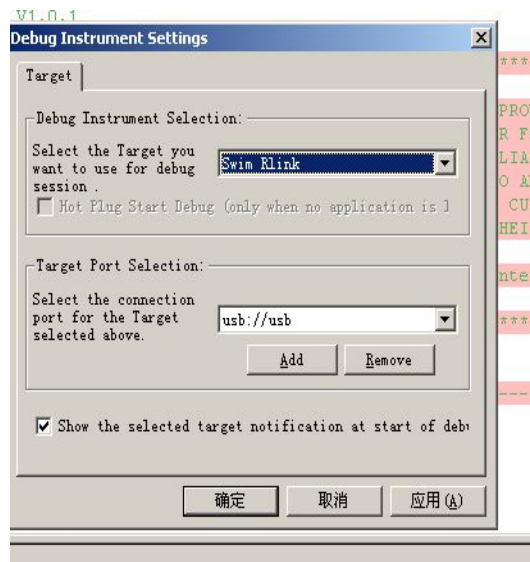
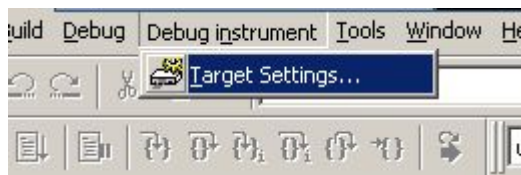


好了,现在可以编译了





设置一下设置 我们就可以 debug 下载了 用 rlink



Ok 开始我们 stm8 学习路程吧