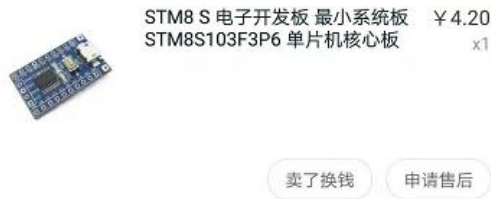


新建一个工程

本教程基于 C 语言，ST 官方开发工具 stvd，官方库等。
由于笔者还做 linux 开发，所以有时会把目录说成路径，希望读者能理解含义。

教程的资料（库，开发工具，烧写工具，部分例程，部分文档）来自笔者在淘宝买的核心板的店铺。



新建工程的方法之前的旧帖子写过，但是整体的风格不太好，见：
http://bbs.elecfans.com/jishu_569529_1_1.html
现参考网友 2974786048 的博客建立工程，博客链接：
https://blog.csdn.net/sy_lixiang/article/details/8461829

在建立工程之前，我们先搭建开发环境，

1，安装软件

开发工具 **ST Visual Develop**，简称 stvd，可到官网或者网络上搜索下载并安装。另笔者所使用的安装包与本文档一同提供，可供读者下载使用。

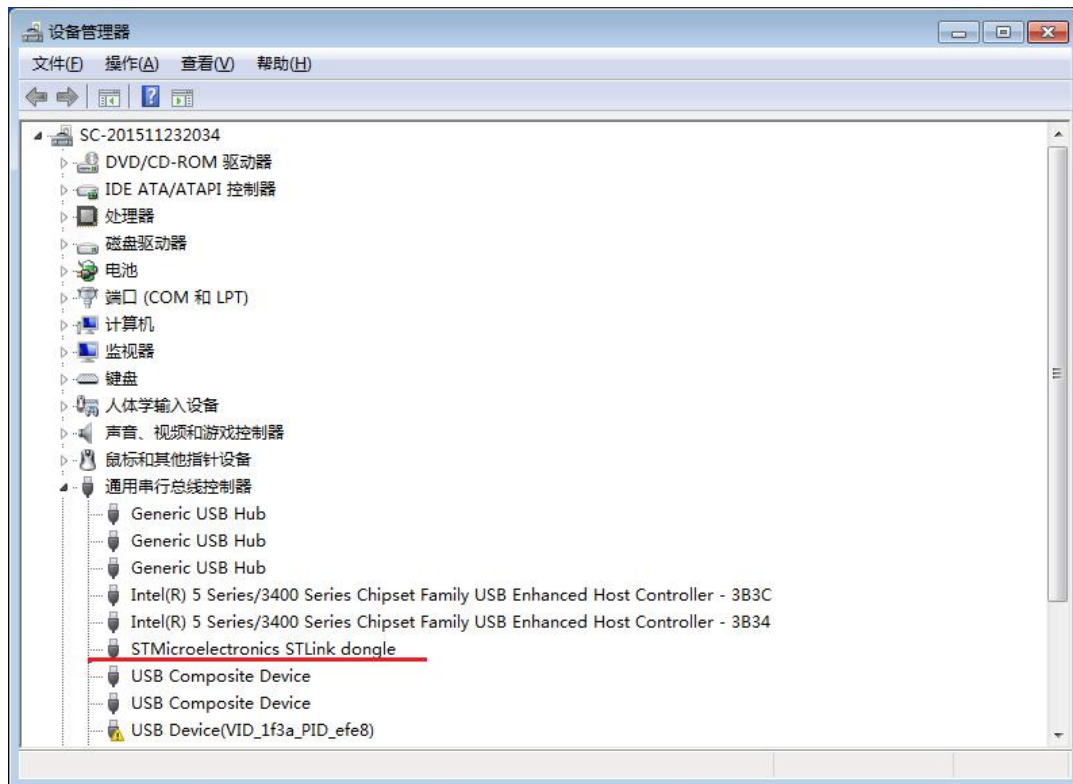


编译插件 **stm8 32K cosmic**，简称 cosmic，安装时请留意安装目录，后面会用到。我的安装目录是 C:\Program Files (x86)\COSMIC\CXSTM8_32K

附 1：本文及提供附件仅供学习交流使用，请勿用于商业目的，如若侵权，请及时联系删除。

2，烧录调试工具 stlink

请自行安装 stlink 驱动，笔者电脑 win10 64bit 版，stlink 驱动是插上 stlink 后自动安装的，安装成功后在设备管理器中可以看到如下图中标出的一项



环境准备完毕，我们开始建立一个 stvd 工程，

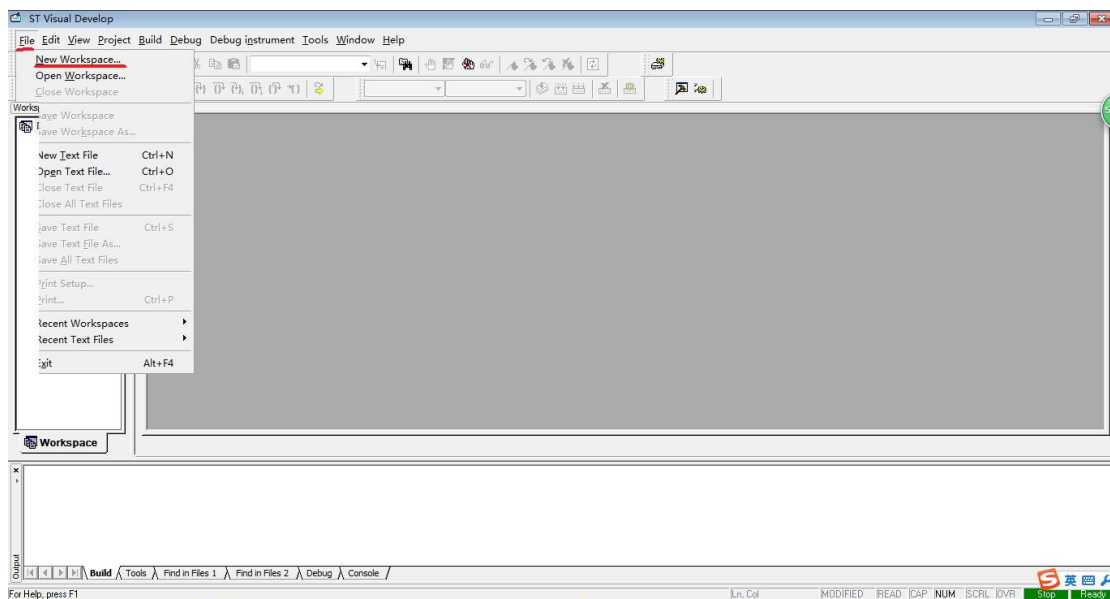
3，建立工程

笔者的 stm8 的具体型号是 stm8s103f3p6，所以后面都默认建立 stm8s103f3p6 的工程。

3.1 双击启动 stvd

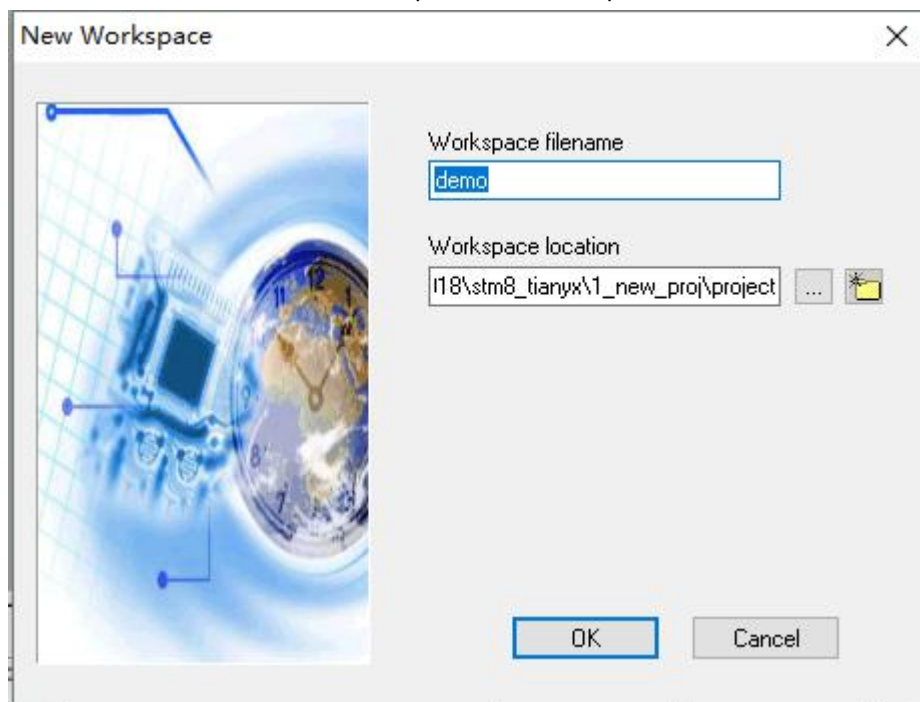
3.2 file --> New Workspace --> Creat workspace and project --> 确定

如下图





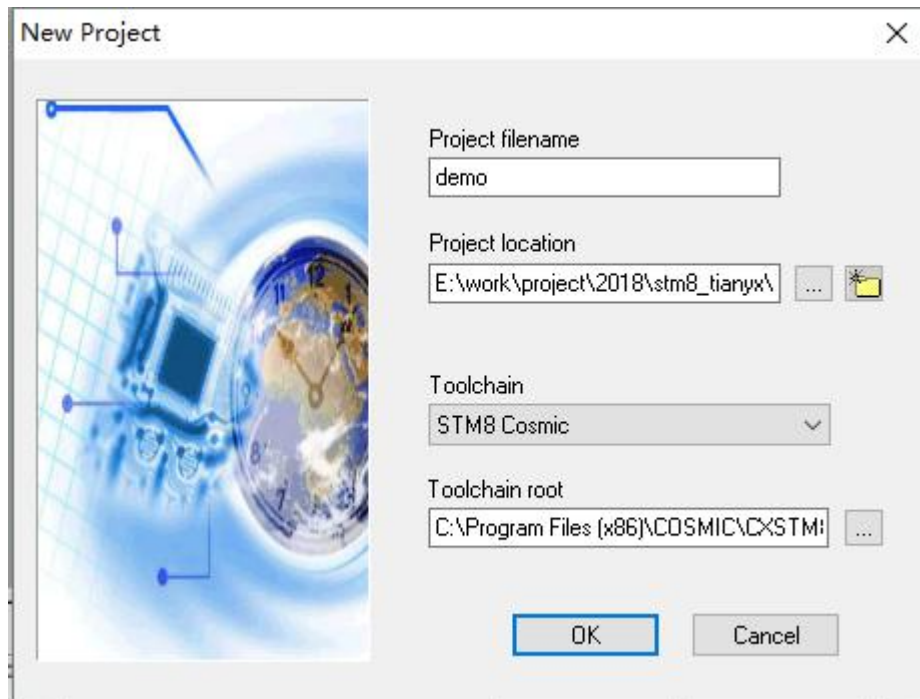
3.3 在弹出的对话框中填写 workspace 名和 workspace 存放路径 ---> ok



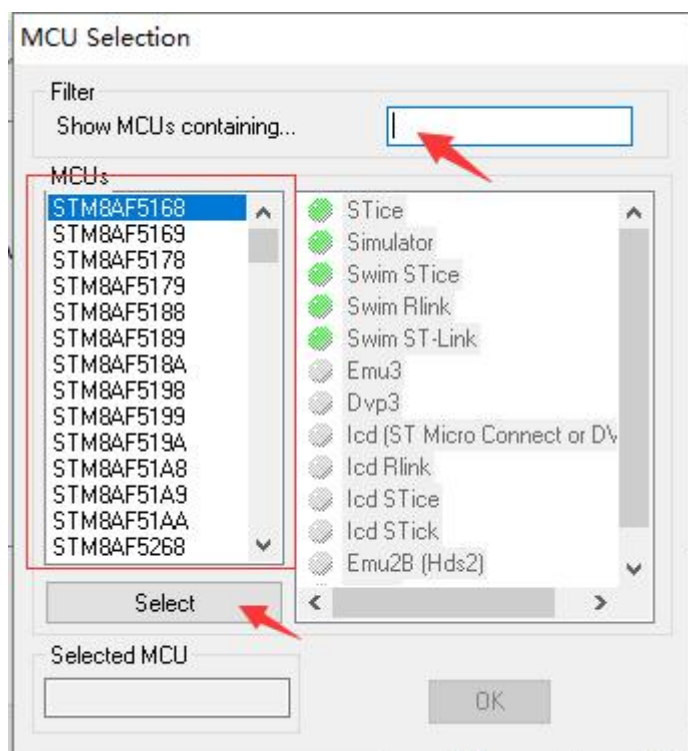
笔者 workspace filename: demo

workspace location: E:\work\project\2018\stm8_tianyx\1_new_proj\project

3.4 在弹出的对话框中填写 project 名和路径，然后选择编译工具链，本教程使用 stm8 cosmic,然后填写工具链的路径，即前述提示记住的安装目录，笔者填写 C:\Program Files (x86)\COSMIC\CXSTM8_32K ---> ok

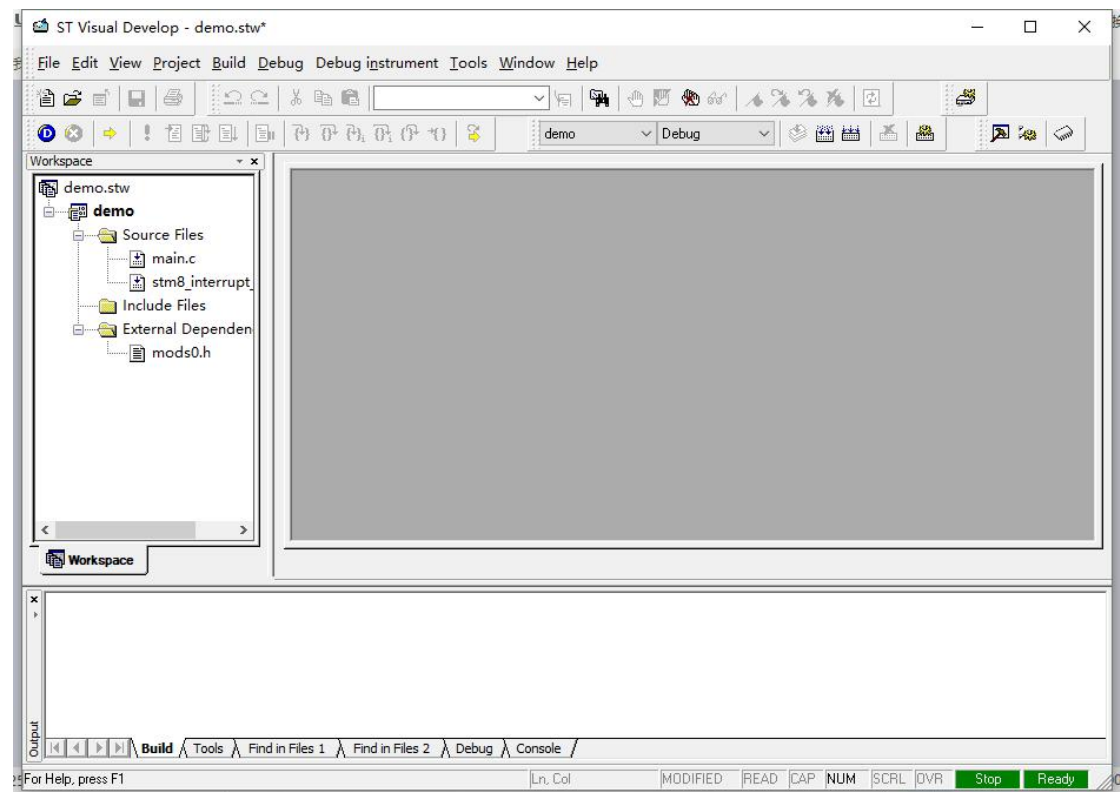


3.5 在弹出的对话框中的 filter 中输入自己的芯片型号，选择对应芯片，或者直接在 MCUs 栏选择，点击 Select，本人使用 stm8s103f3p6 --> ok

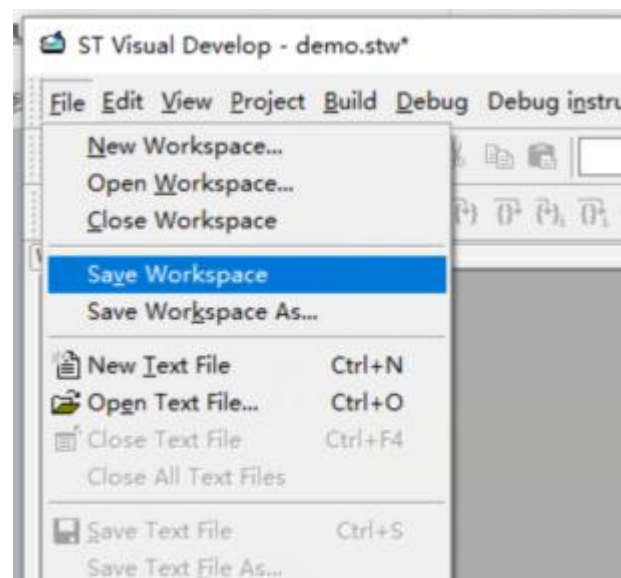


3.6 工程已经基本创建完成，如下图，
双击 source file 展开可以看到在该目录下有 2 个重要文件 main.c 和 stm8_interrupt_vector.c

Main.c 中的函数 main() 是应用程序入口，
stm8_interrupt_vector.c 中主要是中断向量表



3.7 保存工程

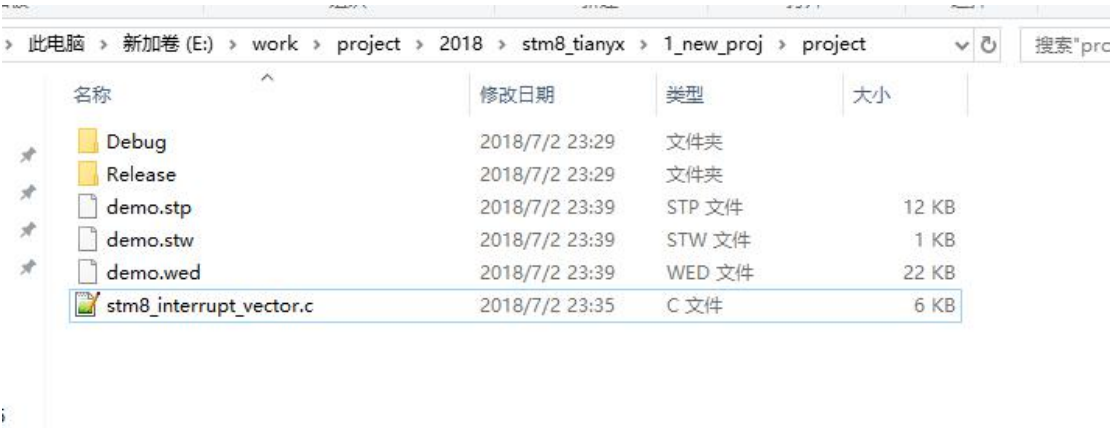


3.8 背景：stvd 自动生成的 main.c 和 stm8_interrupt_vector.c 更适合一些人所说的寄存器开发，寄存器开发相对库函数开发具有代码更紧凑占用 flash 少，cpu 效率高等优势，而库函数开发的优势则是减轻开发工作，缩短开发周期，而且用库函数开发好程序之后，是可以很方便的修改成寄存器版本的，对于目前的市场来说，缩短开发周期意味着产品更快的推向

市场，既可以缩减成本，也可以抢占市场，所以使用库函数开发是更优选择。

3.9 为了方便使用库函数，本人将库中的 main.c 和 stm8_interrupt_vector.c 替换掉本工程源码中的对应文件，

3.9.1 删除 main.c，用新的 stm8_interrupt_vector.c 覆盖原工程的 stm8_interrupt_vector.c，如下图



名称	修改日期	类型	大小
Debug	2018/7/2 23:29	文件夹	
Release	2018/7/2 23:29	文件夹	
demo.stp	2018/7/2 23:39	STP 文件	12 KB
demo.stw	2018/7/2 23:39	STW 文件	1 KB
demo.wed	2018/7/2 23:39	WED 文件	22 KB
stm8_interrupt_vector.c	2018/7/2 23:35	C 文件	6 KB

3.9.2 进入上一级目录，新建文件夹 src 和 lib，如下图



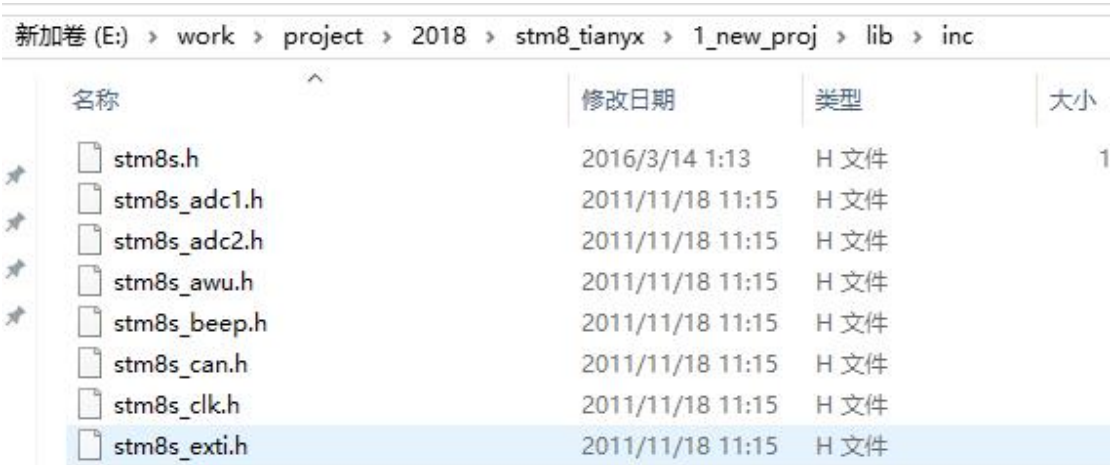
名称	修改日期	类型	大小
lib	2018/5/16 23:36	文件夹	
project	2018/7/2 23:39	文件夹	
src	2018/6/26 23:09	文件夹	

3.9.3 进入 lib 文件夹，新建文件夹 inc 和 src，如下图，inc 存放库文件的头文件，src 存放库文件的源文件。



名称	修改日期	类型	大小
inc	2018/5/16 23:36	文件夹	
src	2018/5/16 23:36	文件夹	

3.9.4 将库文件中的头文件(*.h)拷贝到工程目录的子文件夹\lib\inc 中，如下图：



名称	修改日期	类型	大小
stm8s.h	2016/3/14 1:13	H 文件	1
stm8s_adc1.h	2011/11/18 11:15	H 文件	
stm8s_adc2.h	2011/11/18 11:15	H 文件	
stm8s_awu.h	2011/11/18 11:15	H 文件	
stm8s_beep.h	2011/11/18 11:15	H 文件	
stm8s_can.h	2011/11/18 11:15	H 文件	
stm8s_clk.h	2011/11/18 11:15	H 文件	
stm8s_exti.h	2011/11/18 11:15	H 文件	

3.9.5 将库文件中的源文件(*.c)拷贝到工程目录的子文件夹\lib\src 中，如下图：

新加卷 (E:) > work > project > 2018 > stm8_tianyx > 1_new_proj > lib > src

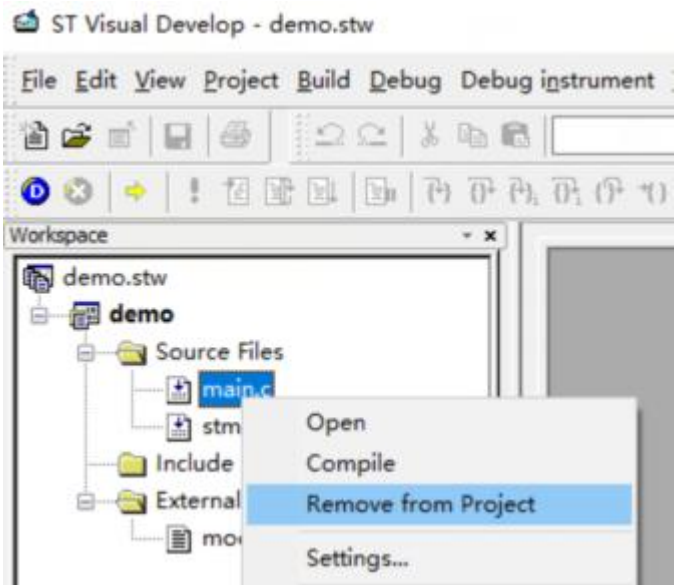
名称	修改日期	类型
stm8s_adc1.c	2011/11/18 11:15	C 文件
stm8s_adc2.c	2011/11/18 11:15	C 文件
stm8s_awu.c	2011/11/18 11:15	C 文件
stm8s_beep.c	2011/11/18 11:15	C 文件
stm8s_can.c	2011/11/18 11:15	C 文件
stm8s_clk.c	2011/11/18 11:15	C 文件
stm8s_exti.c	2011/11/18 11:15	C 文件
stm8s_flash.c	2011/11/18 11:20	C 文件

3.9.6 将 main.c, stm8s_conf.h, stm8s_it.c, stm8s_it.h 拷贝到工程目录的子文件夹\src 中，如下图，注此目录不是上面库文件中的 src 目录。

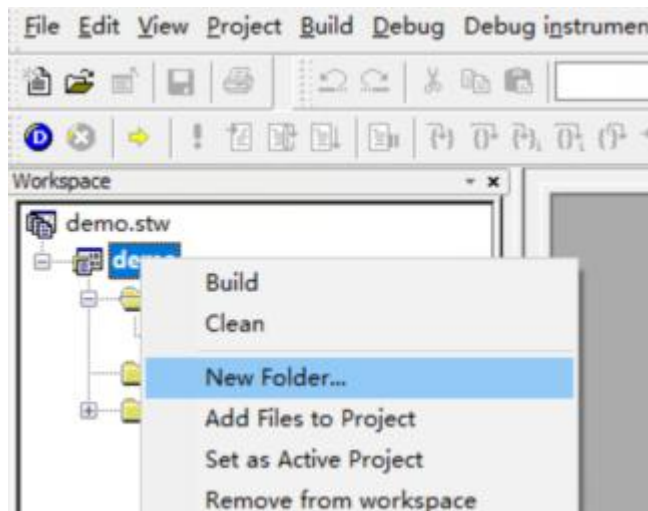
此电脑 > 新加卷 (E:) > work > project > 2018 > stm8_tianyx > 1_new_proj > src

名称	修改日期	类型	大小
main.c	2018/6/26 23:09	C 文件	3 KB
stm8s_conf.h	2011/11/21 15:54	H 文件	5 KB
stm8s_it.c	2018/5/17 0:19	C 文件	13 KB
stm8s_it.h	2011/11/21 15:54	H 文件	6 KB

3.9.7 在开发工具中将刚才的文件改动同步修改到工程中，删除原来的 main.c，单击选中 main.c，右击选择 Remove from Project，如下图

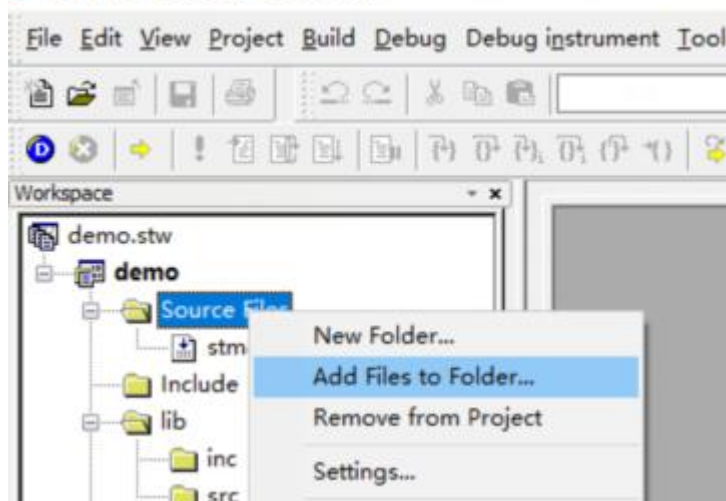


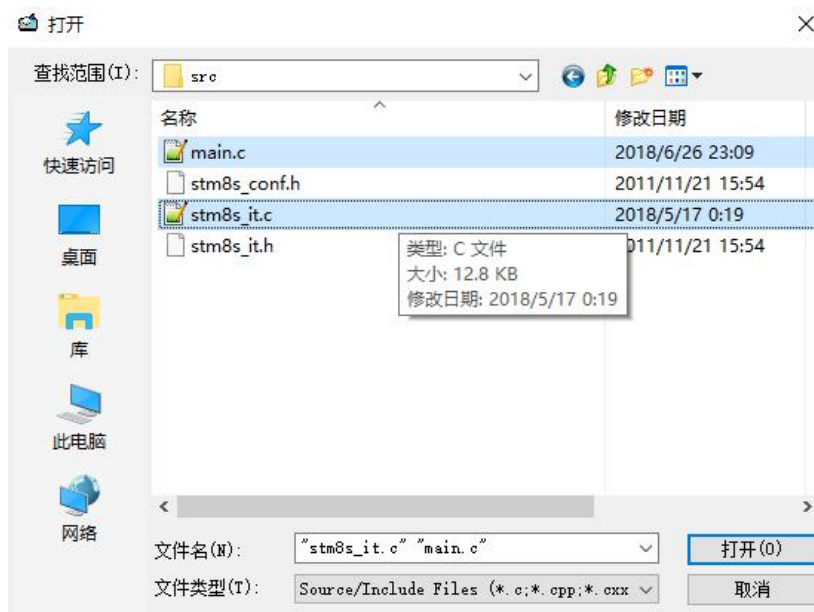
新建 lib/src, lib/inc 文件夹。单击选中 demo，右击选择 New folder，输入 lib，确定。如下图，单击选中 lib，右击选择 New folder，输入 inc，确定。lib/src 文件夹建立方法同理。



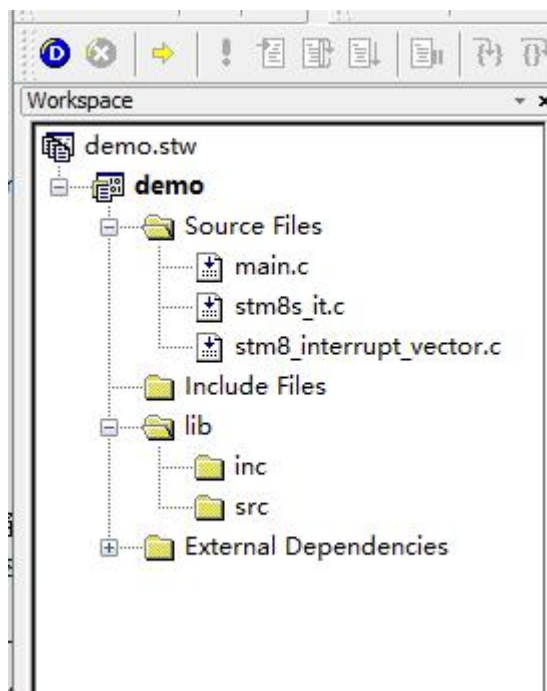
将 main.c, stm8s_it.c 导入到工程的 Source Files 目录下, 单击 Source Files, 右击选择 Add Files to Folder, 选择 main.c, stm8s_it.c 点击打开。

ST Visual Develop - demo.stw

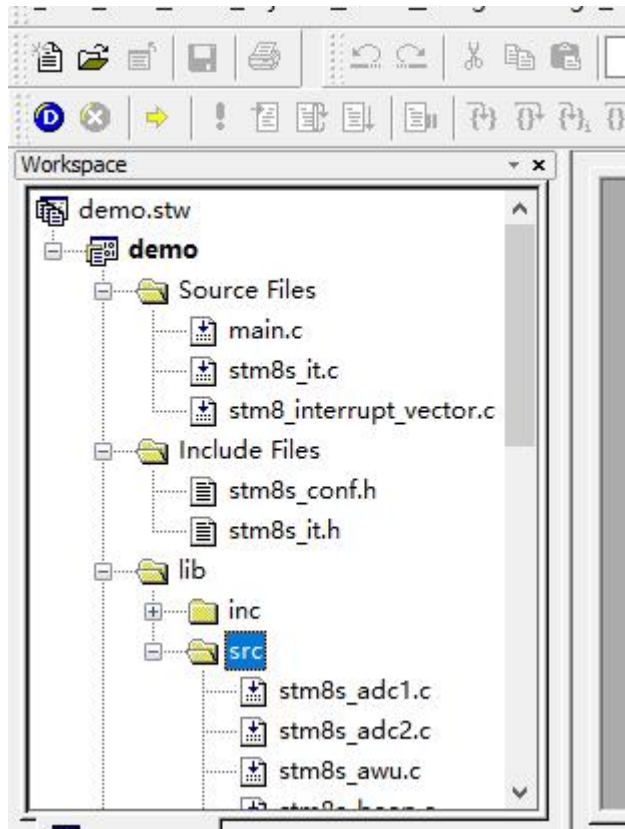




结果如下:

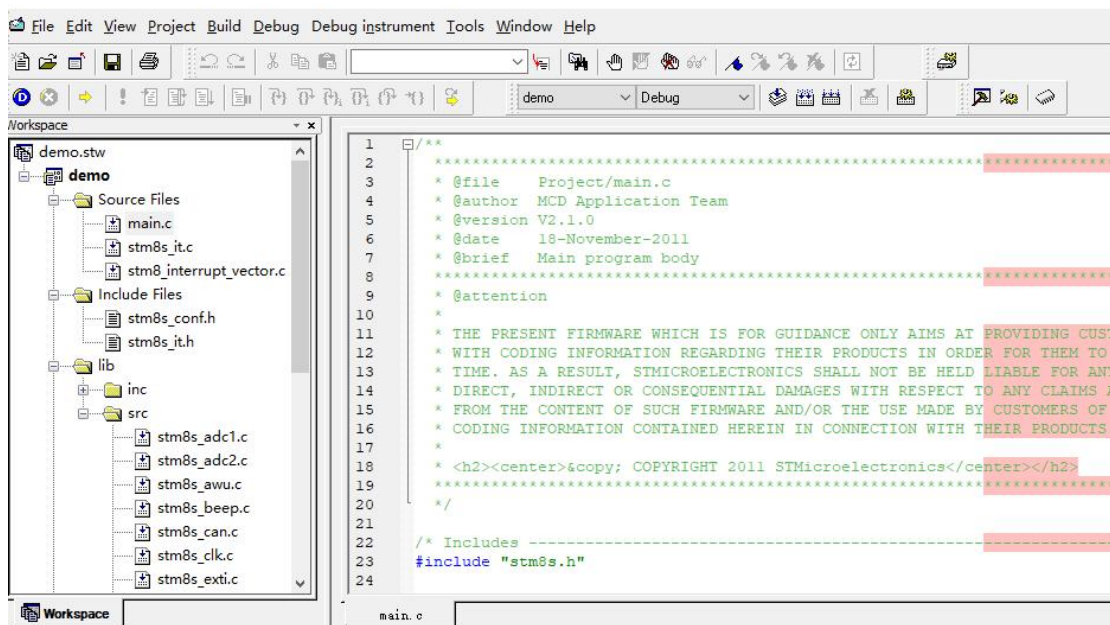


同理导入 stm8s_conf.h, stm8s_it.h 到 Include Files, 导入库头文件到 lib/inc, 导入库源文件到 lib/src。结果如下:



3.9.8

双击 `main.c` 可以打开 `main.c`，显示在右侧区域，如下图，可以在这里编辑修改 `main.c` 文件，也可以使用其他文件编辑软件编辑，编辑完到 `stvd` 中会提示同步修改，确认即可。



对于库文件中的头文件，除文件 `stm8s.h` 之外的文件都是包含库函数和相关结构体枚举类型等的声明，如 `stm8s_adc1.h` 是用于 `adc1` 的所有库函数声明，而 `stm8s.h` 则定义了一些重要的编译信息以及将寄存器地址映射成符号等；

Src 中则是库函数的实现，每一个 `xxx.c` 文件在 `inc` 中都有一个对应名的 `xxx.h` 文件。

3.10 因为这是官方出的 stm8 的整个系列的通用库，但是各芯片间还是有点区别，所以需要区分我们编译出来是用于哪款 ic，需要在在文件 stm8s.h 中指定 ic 的型号，如下图，笔者是 stm8s103

```
/* Define to prevent recursive inclusion */
#ifndef __STM8S_H
#define __STM8S_H

/** @addtogroup STM8S_StdPeriph_Driver
 *  @{
 */

/* Uncomment the line below according to the target STM8S or STM8A device
 application. */

/* #define STM8S208 */           /*!< STM8S High density devices with CAN */
/* #define STM8S207 */           /*!< STM8S High density devices without CAN */
/* #define STM8S007 */           /*!< STM8S Value Line High density devices */
/* #define STM8AF52Ax */          /*!< STM8A High density devices with CAN */
/* #define STM8AF62Ax */          /*!< STM8A High density devices without CAN */
/* #define STM8S105 */           /*!< STM8S Medium density devices */
/* #define STM8S005 */           /*!< STM8S Value Line Medium density device */
/* #define STM8AF626x */          /*!< STM8A Medium density devices */
#define STM8S103                 /*!< STM8S Low density devices */
/* #define STM8S003 */           /*!< STM8S Value Line Low density devices */
/* #define STM8S903 */           /*!< STM8S Low density devices */

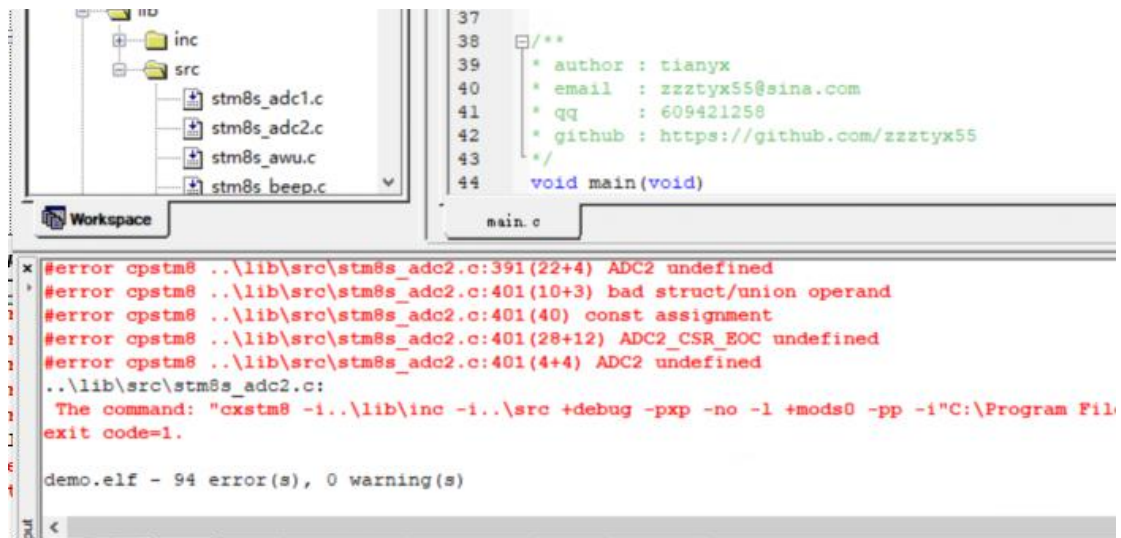
/*
 Tip: To avoid modifying this file each time you need to switch betw
 devices, you can define the device in your toolchain compiler pr

- High-Density STM8A devices are the STM8AF52xx STM8AF6269/8x/Ax,
STM8AF51xx, and STM8AF6169/7x/8x/9x/Ax microcontrollers where the Fl
density ranges between 32 to 128 Kbytes
- Medium-Density STM8A devices are the STM8AF622x/4x, STM8AF6266/68,
STM8AF612x/4x, and STM8AF6166/68 microcontrollers where the Flash me
density ranges between 8 to 32 Kbytes
- High-Density STM8S devices are the STM8S207xx, STM8S007 and STM8S208
where the Flash memory density ranges between 32 to 128 Kbytes.
- Medium-Density STM8S devices are the STM8S105x and STM8S005 microcon
where the Flash memory density ranges between 16 to 32-Kbytes.
- Low-Density STM8S devices are the STM8S103xx, STM8S003 and STM8S903x
where the Flash density is 8 Kbytes. */
```

4，编译工程

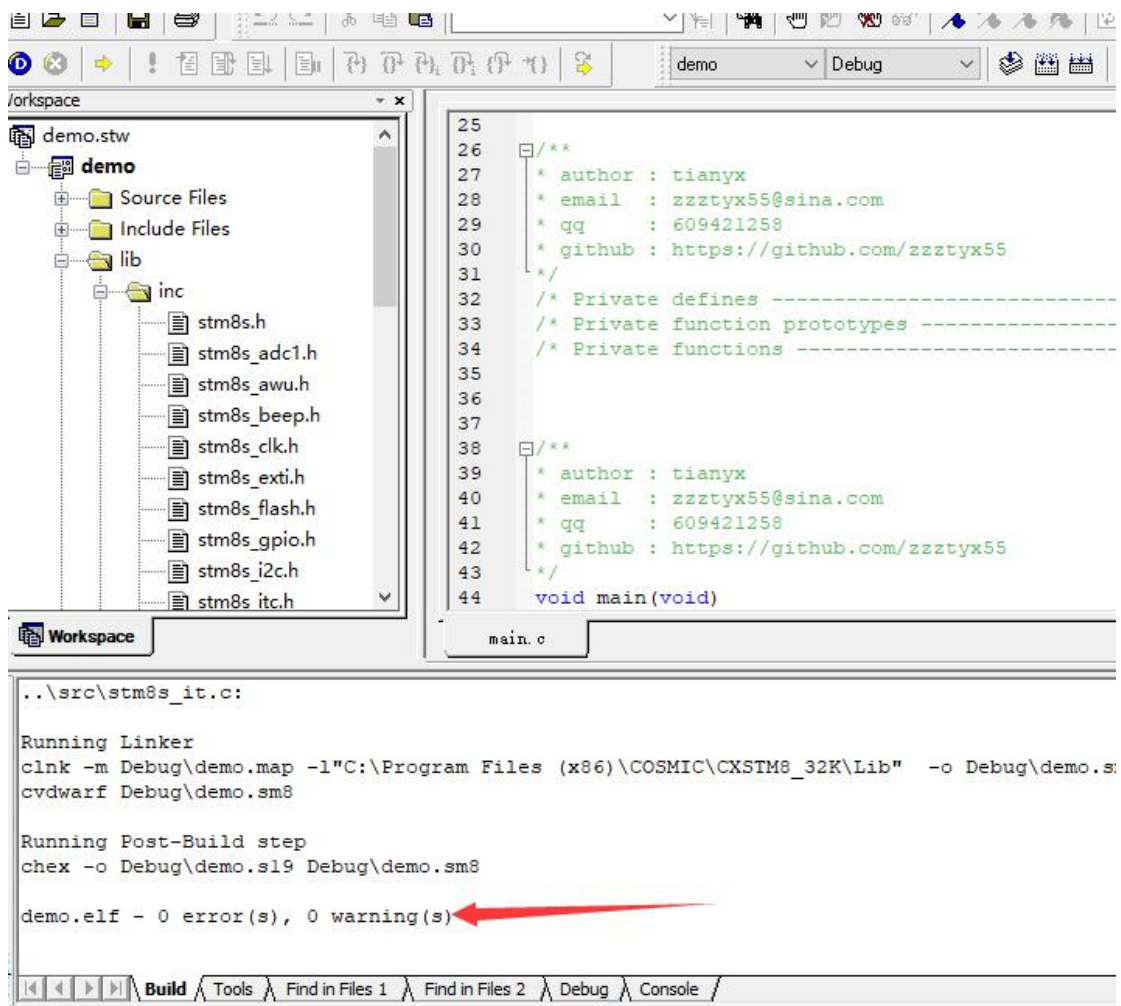
直接按快捷键 F7 或者点击工具栏图标  (中间红圈的) 或者点击菜单栏 Build --> Build. 即可编译工程。

提示错误很多，如下图红色文字：



因为 stm8s103f3p6 没有 adc2, 所以不应该导入库文件的 adc2, 从工程中删除, 其他同类错误相同操作即可。

最终编译通过, 如下图:



实际上, 我们开发时不需要建立工程后一股脑把所有的库文件全导入工程, 然后又根据错误一个个剔除。真正讲效率的方法是我们开发时用到哪个库文件就导入哪个库文件。