学会简单的自定义库函数操作方法

很多编程的人都有这样的经历，当程序比较大的时候，阅读和查找程序中的语句或者调试程序都比较麻烦，特别是对于已经成功测试的程序进行移植的时候，拷贝大量的函数和语句容易出错。在一些程序中，某些函数和语句是专门服务于某一个功能的，类似于功能驱动函数和程序，这些程序就可以把它组合起来，形成一个文件和自定义库函数。

自定义库函数可以很好地解决上述问题，下面就以PCF8591的实验范例演示自定义库函数的操作流程，学会这个处理方法，可以让主程序“瘦身”，且具有更好的可读性和可移植性。

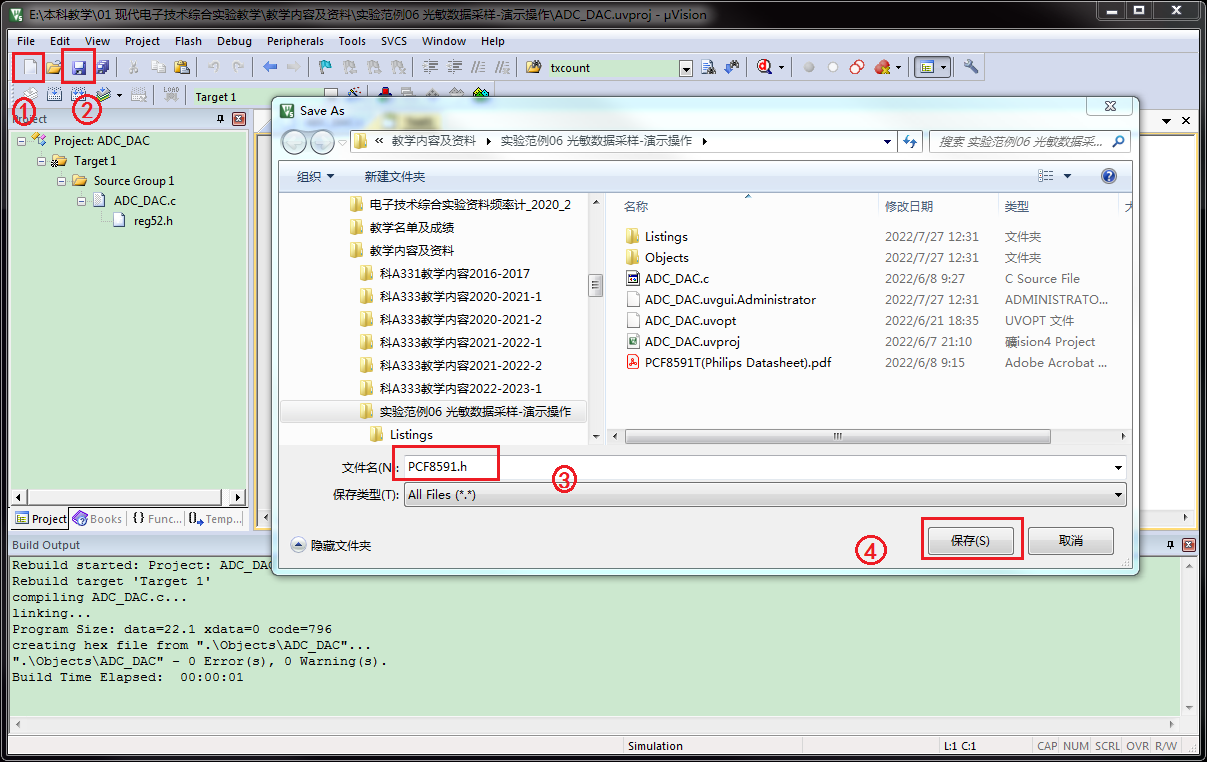
在操作前必须对所要处理的程序比较熟悉，特别是与之密切相关的硬件说明、管脚定义、函数功能等，各种驱动控制字和接口程序要有明确的概念，然后就可以完成库函数和文件生成工作。

1. 源程序工程文件夹复制

在这一操作中，为保护已有的程序，避免修改中出错，大多数情况下会在另外一个文件夹内去操作。简单的操作办法新建一个文件夹，将全部源程序工程文件拷贝到新建的文件夹，然后直接打开进行修改；也可以新建工程和文件，把源工程的程序语句拷贝过来修改，这里我们选前一种方式。

1. 生成头文件
2. 完成第一个步骤后，打开拷贝好的工程文件，编译一次确认程序拷贝过程中没有错误。在工具栏点击“新建”，新文件窗体打开后，紧接着点击“保存”工具，注意查看路径是否指向当前工程的文件夹！保存文件的时候扩展名为h，例如：PCF8591.h。

操作流程如下图所示。



1. 在该文件内输入以下语句（拷贝斜体字部分）：

*/\**

*PCF8591.h*

*STC89C52单片机驱动PCF8591的程序头文件*

*\*/*

*#ifndef \_PCF8591\_H\_*

*#define \_PCF8591\_H\_*

*#include<reg52.h>*

*//-----------模拟I2C总线的必备定义---------------*

*//-----------子函数声明-------------*

*#endif //\_PCF8591\_H\_*

说明：以后生成其他头文件基本都是这个模板，修改\_H前的文件名即可。

1. 找出**源程序**中和PCF8591相关的全部**必备定义**内容，有如下部分：

*//定义I2C总线时钟的延时值，要根据实际情况修改，取值1～255*

*//SCL信号周期约为(I2C\_DELAY\_VALUE\*4+15)个机器周期*

*#define I2C\_DELAY\_VALUE 10*

*//定义I2C总线停止后在下一次开始之前的等待时间，取值1～65535*

*//等待时间约为(I2C\_STOP\_WAIT\_VALUE\*8)个机器周期*

*//对于多数器件取值为1即可；但对于某些器件来说，需要较长的延时*

*#define I2C\_STOP\_WAIT\_VALUE 1*

*//模拟I2C总线的引脚定义*

*sbit I2C\_SCL = P2^1; //I2C 时钟*

*sbit I2C\_SDA = P2^0; //I2C 数据*

*sbit RST\_DS = P2^4; //加上后可以关掉DS1302时钟芯片输出*

*//定义PCF8591 地址*

*#define AddWr 0x90*

将上述定义语句**剪切**到PCF8591.h对应位置下方。

找出源程序中和PCF8591相关的全部函数（其中一部分是I2C驱动），将**函数定义语句**依次**拷贝**都过来放在PCF8591.h文件的“子函数声明”下面，注意要在每个函数定义后面加上**分号**！构成全部**函数声明**。有如下几个函数：

*void I2C\_Delay();*

*void I2C\_Init();*

*void I2C\_Start();*

*void I2C\_Stop();*

*void I2C\_Write(unsigned char dat);*

*unsigned char I2C\_Read();*

*bit I2C\_GetAck();*

*void I2C\_PutAck(bit ack);*

*bit Pcf8591\_DaConversion(unsigned char addr,unsigned char channel, unsigned char Val);*

*bit PCF8591\_SendByte(unsigned char addr,unsigned char channel);*

*unsigned char PCF8591\_RcvByte(unsigned char addr);*

检查无误后保存，见下图。

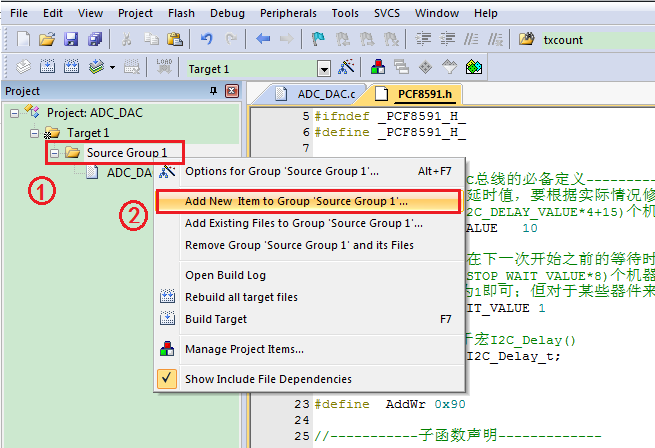


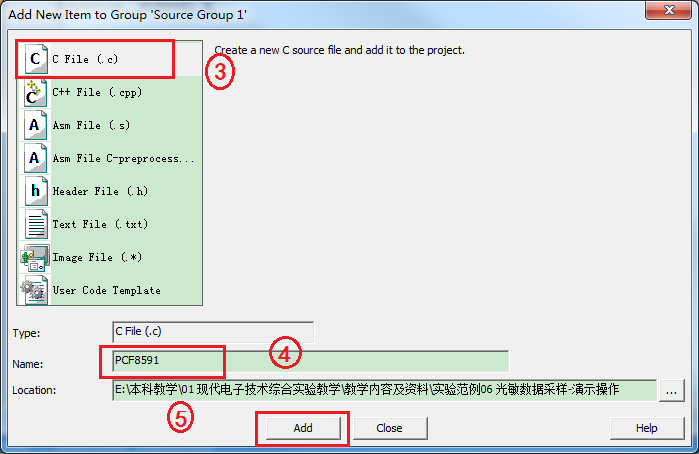
1. 生成库函数对应的C文件
2. 鼠标右键单击左侧项目窗体中“Source Group 1”，选择添加新的文件，见下图。
3. 弹出窗体文件类型选C File，文件名自己确定，比如PCF8591。点击“Add”，就可以看到左侧项目窗体出现了我们新建的文件。
4. 添加需要引用的头文件，除了常见的加入stc8.h，必须把我们生成的头文件PCF8591.h也包含进去，如下：

*#include "stc8.h"*

*#include "PCF8591.h"*

1. 把主程序中和PCF8591相关的全部函数都剪切进来。检查无误后保存文件，暂时不要编译，肯定会报错，原因自己想想就明白了。



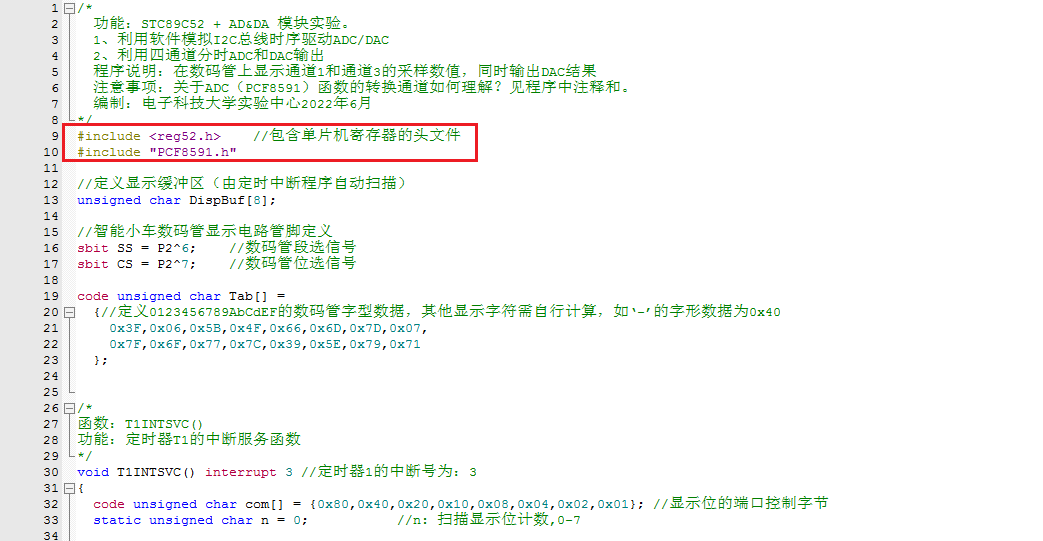


1. 引用库函数

现在，已经完成了程序的头文件和自定义库函数文件操作，只剩下引用了，只有合理的引用，才能使得两个C程序成为一个工程，等效于一个程序。

在源程序中包含的头文件加入#include "PCF8591.h"即可。

编译一下，看看有没有问题呢？如果是按照上述步骤严格操作，就一定会成功的，下载执行，会看到同样的结果。不同的是，主程序瘦身了，简练易懂，为将来的移植带来了方便。



1. 注意事项
2. 我们经常会将特定功能的模块或者接口做成头文件+库函数文件，不必强拉硬拽，反而效果不好。
3. 上述做法可以继续嵌套，各模块之间相互独立，以后组合其他程序就更加方便快捷。
4. 组合到其他主程序使用的时候，最好是把两个文件拷贝到需要新建工程的文件夹内再使用，不要直接修改路径去对应，否则整体搬移的时候会带来一些麻烦。