## 实验报告书写要求

说明：

1、每个同学按照以下要求写实验报告，每次实验将该实验对应的内容copy过去。

2、只交电子稿，把每个实验的实验原理、过程和总结等补充完整。

3、每次实验以自己的学号姓名命名，例如：013301734209王欢

4、电子文稿交到：ftp://192.168.40.14/郭海如/”本学期”/本课程/”自己班级”/对应的实验名称

目录

[实验报告书写要求 1](#_Toc66298814)

[实验一 STC51联机调试 1](#_Toc66298815)

[实验二 STC51闪烁灯 7](#_Toc66298816)

[实验三 STC51定时闪烁灯 8](#_Toc66298817)

[实验四 STC51串口发送 10](#_Toc66298818)

[实验五 STM32系统认知实验 11](#_Toc66298819)

[实验六 STM32的LED灯 28](#_Toc66298820)

[实验七 LCD显示例程 29](#_Toc66298821)

[实验八 STM32外部中断 30](#_Toc66298822)

[实验九 STM32基本定时器 30](#_Toc66298823)

[实验十 PWM输出 31](#_Toc66298824)

[实验十一 STM32串口 32](#_Toc66298825)

[实验十二 STM32模数转换 33](#_Toc66298826)

[实验十三 EEPROM存储器 34](#_Toc66298827)

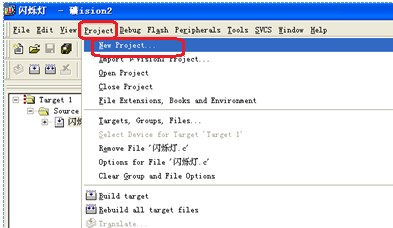
[实验十四 STM32实时时钟 34](#_Toc66298828)

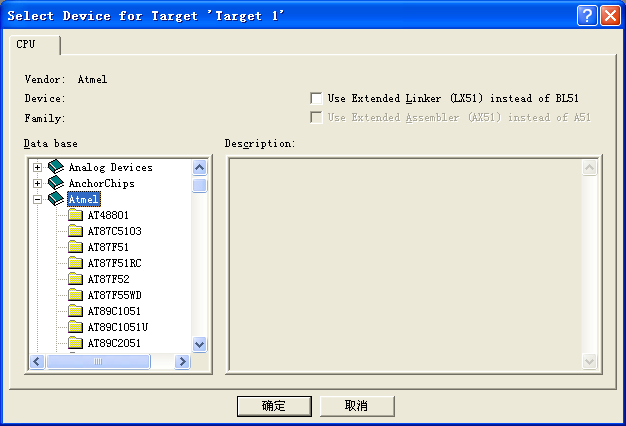
## 实验一 STC51联机调试

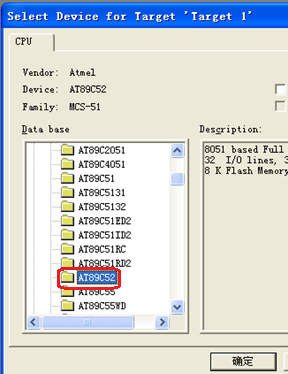
1. **实验目的**
2. 熟悉Keil的安装过程；
3. 熟悉Proteus的安装过程；
4. 熟悉Keil和Proteus联机调试。
5. **实验内容**

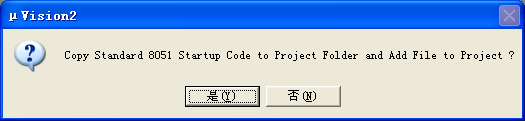
**1、Keil的简单调试**

新建工程：

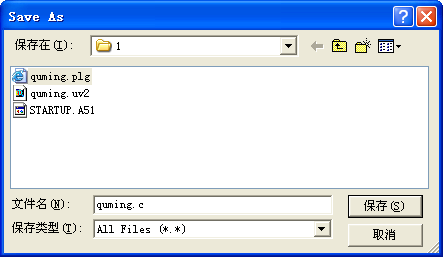
放在某个地方，并取名

选择型号

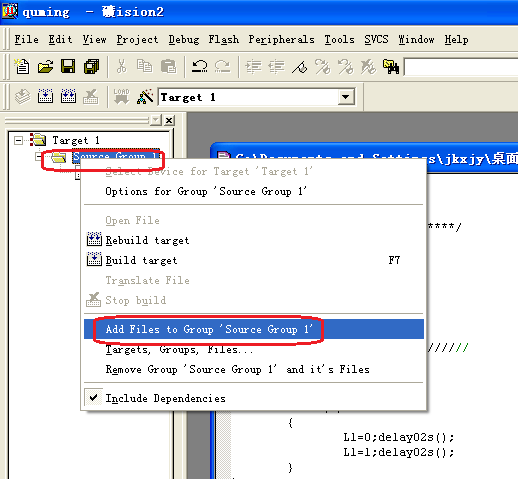
型号AT89C51或AT89C52

自动加入头文件

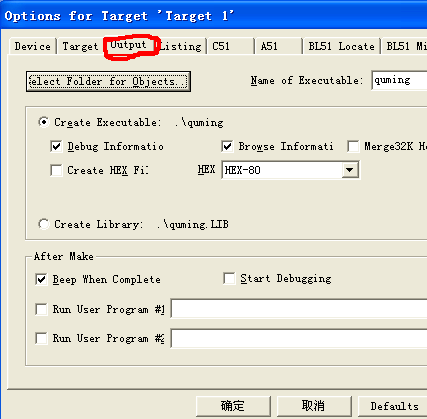
新建文件

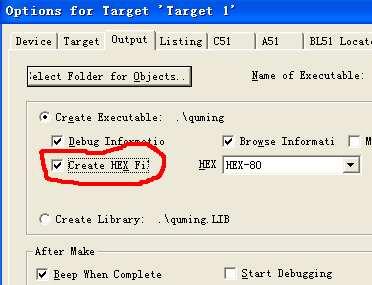
输入程序后，以“文件名+.c”命名

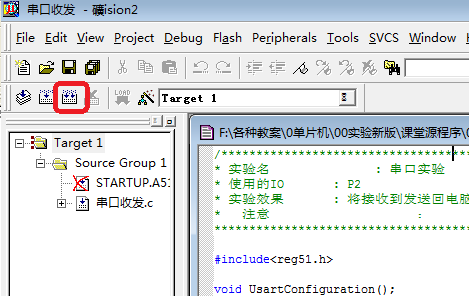
输入C语言源程序，或者copy源程序进行修改。

右键Source Group1再选择

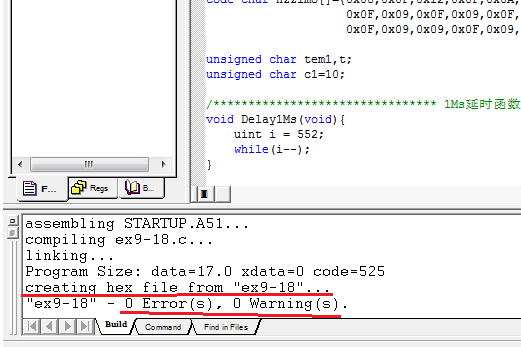
点击设置按钮

弹出对话框，选择“output”

选择下面红色部分创建hex文件

编译文件：点

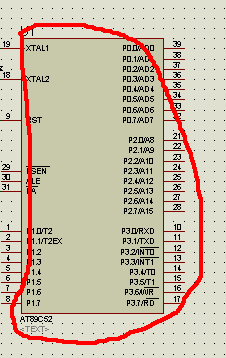
无错误的话，会生成hex文档。

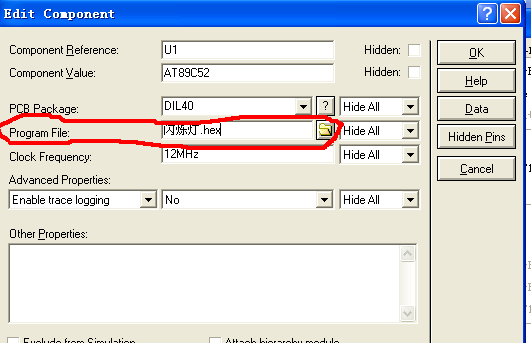
左下角

Creating hex file from ““:创建hex二进制文档，后面可以将该文件下载到单片机，或者用于proteus联机仿真

0 Error(s),0 Warning(s):表示没任何错误，若有错误上面就有提示错误行，且不能生成hex文件。

**2、Hex文件加载至Proteus仿真**

双击Proteus中的单片机AT89C5xx

找到keil中生成的hex文件

在Proteus左下角点击play按钮即可看到现象

点击stop即停下

接下来，更改程序，继续编译，生成hex文件，再按看现象。

**三、实验总结**

## 实验二 STC51闪烁灯

**一、实验目的及要求**

1.认识IO端口。

2.认识延时程序原理。

3.点亮led发光二极管。

**二、实验原理**

单片机最小系统：

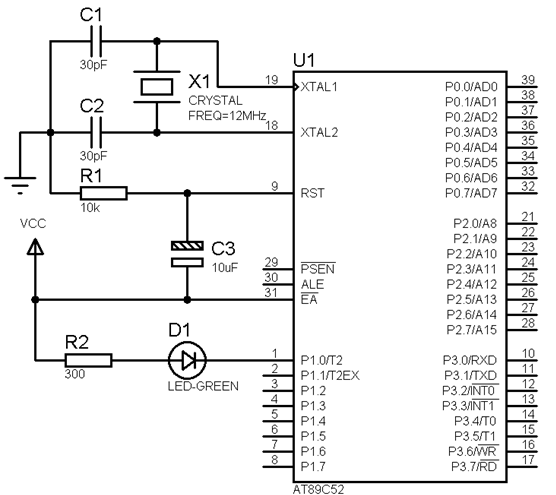
单片机的时序单位：

51单片机内部组成：

**三、实验过程**

闪烁灯：在单片机某端口上接一个发光二极管L1，使L1周期性地一亮一灭

①实验电路图：



②程序流程图及源程序：

#include <REG52.H>

sbit L1=P1^0;

/\*\*延时0.2秒子程序\*\*/

void delay02s(void)

{

unsigned char i,j,k;

for(i=20;i>0;i--)

for(j=40;j>0;j--)

for(k=248;k>0;k--);

}

///////////////////

void main(void)

{

while(1)

{

L1=0;delay02s();

L1=1;delay02s();

}

}

③实验结果：

**四、实验总结**

## 实验三 STC51定时闪烁灯

**一、实验目的及要求**

1、理解定时计数器原理

2、理解定时器工作方式1

**二、实验原理**

常用实现定时的方法：

单片机定时器的工作过程：

51单片机定时器相关的寄存器：

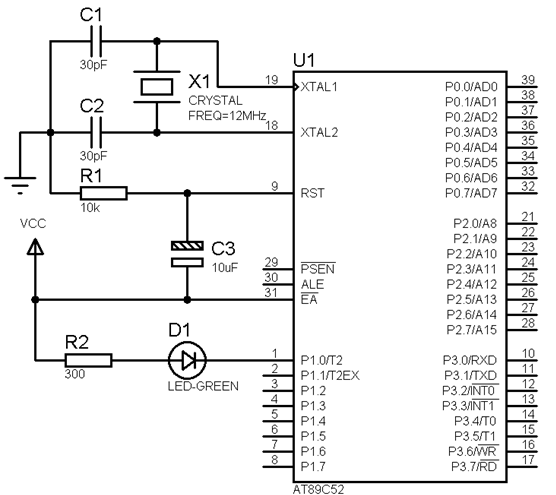
51单片机方式1的工作原理：

时间常数的计算：

**三、实验过程**

实现功能：在单片机某端口上接一个发光二极管L1，使L1周期性地一亮一灭，闪烁间隔为0.5秒， 使用晶振的频率f=12MHz。

①实验电路图：



②程序流程图及源程序：

//主程序：

#include <REG52.H>

#define uchar unsigned char

sbit L1=P1^0;//L1,发光二极管引脚

ucharicount; //记录溢出次数

void main(void)

{ icount = 0; //记录溢出次数

TMOD = 0x01; //设定T/C0工作在定时器方式1

L1 = 1;

TH0 = (65536-10000)/256;

TL0 = (65536-10000)%256;//装载计数初值

EA = 1; //开总中断

ET0 = 1; //开T0中断

TR0 = 1; //启动T0

while(1);

}

void timer0\_10ms(void) interrupt 1 //中断程序：

{ TH0 = (65536-10000)/256;

TL0 = (65536-10000)%256; //重置计数初值

icount++; //溢出次数加1

if(icount == 50)

{icount = 0; //累计够0.5秒时次数加清0

L1 = !L1; //控制灯闪烁

}

}

③实验结果：

**四、实验总结**

## 实验四 STC51串口发送

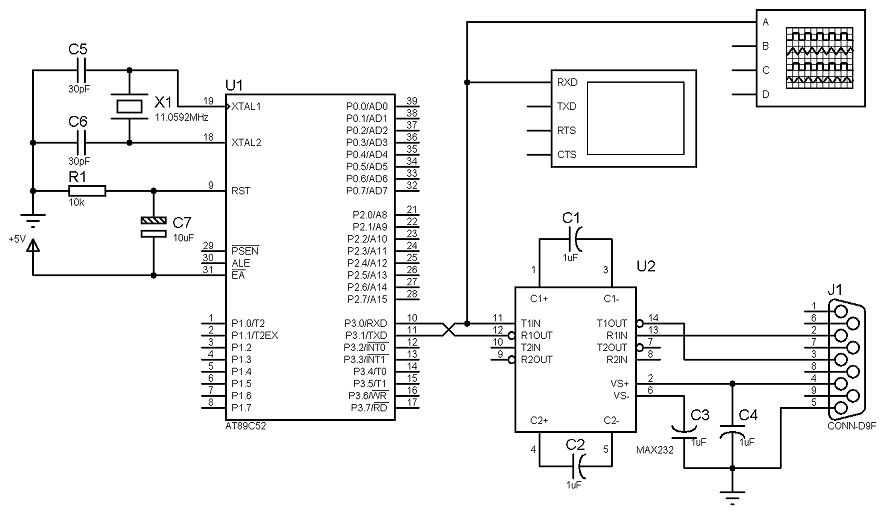
**一、实验目的及要求：**

1、理解串口工作原理

2、理解串口发送过程

**二、实验内容**

单片机通过串行口不停的向其他设备发送字符串"Welcome to Beijing''



#include <reg52.h>

#include <intrins.h>

char code str[] = "Welcome to Xiao Gan \n\r";

voidsend\_str();

main()

{int j;

TMOD = 0x20; // 定时器1工作于8位自动重载模式, 用于产生波特率

TH1 = 0xFD; // 波特率9600

TL1 = 0xFD;

SCON = 0x50; // 设定串行口工作方式

PCON &= 0xef; // 波特率不倍增

TR1 = 1; // 启动定时器1

IE = 0x0; // 禁止任何中断

while(1)

{

send\_str(); // 传送字串"welcome!

for(j=1;j<200;j++);

}

}

voidsend\_str()

// 传送字串

{

unsigned char i = 0;

while(str[i] != '\0')

{

SBUF = str[i]; //数据传送

while(!TI); // 等特数据传送

TI = 0; // 清除数据传送标志

i++; // 下一个字符

}

}

**三、实验结果**

发送字符“W”的实验结果：

分析波形图：

**四、实验总结**

## 实验五 STM32系统认知实验

**一、实验目的**

1. 熟悉程序下载过程
2. 熟悉硬件环境
3. 熟悉软件环境及新建工程的过程

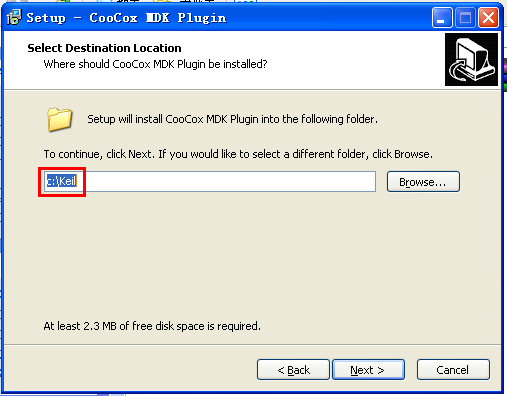
**二、实验内容**

1. 安装驱动程序

（1）进入相关目录，打开驱动程序文件



点击“Next”，之后，将目录更改为C:盘



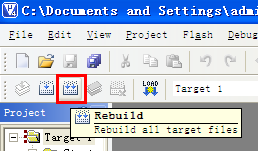
后面一直点击Next直到完成。

（2）进入相关目录，打开KEY\_TEST实验工程



（3）接好数据线，USB接口接到离液晶屏近的端口。

（4）编译



（5）一键下载：若出现下载不了，检查端口是否接错。等待一分钟，等接口识别之后再下载。

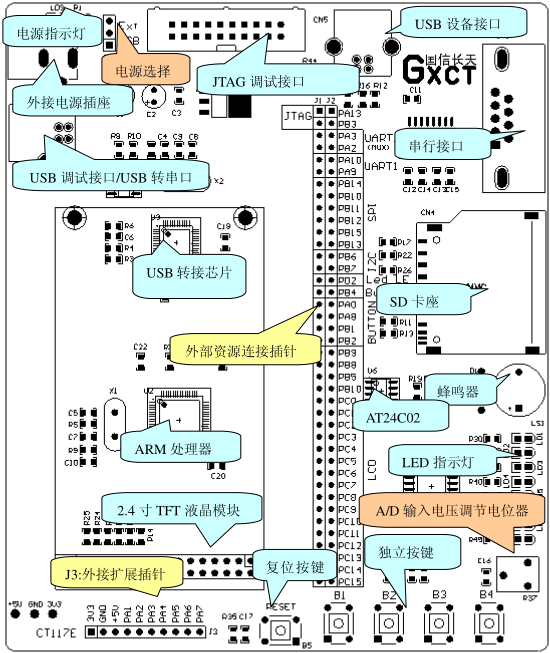


（6）看现象：

液晶屏会显示字符，否则，有问题，有问题必须反馈。

2、熟悉硬件电路：

（1）各个基本功能模块在板上的布局如下图所示。



（2.） 功能模块

CT117E 嵌入式竞赛板由以下基本功能模块组成。

\*处理器：STM32F103RBT6

\*两个 RS232 串口

\*一个 SD 卡接口

\*一个 USB 从设备接口

\* 2.4 寸 TFT-LCD

\*四路功能按键输入

\*八路 LED 输出

\*板载 USB 转串口

\*有源蜂鸣器

\*可调模拟电压输入

\* EEPROM 芯片 AT24C02

\*板载 JTAG 调试功能（USB 接口，无需外接调试器）

（3）. 跳线说明

JP1：电源选择选择

EXT 短接：选择外部 5V 电源

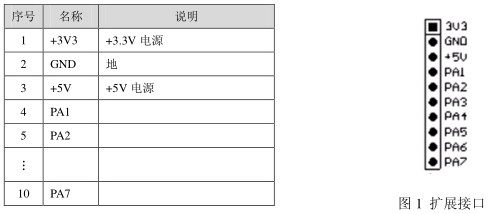
USB 短接：选择 USB 供电

（4.） 电位器功能说明

电位器 R37，用于调节 A/D 输入电压值。

（5）. 接口说明

1）J3 为竞赛板扩展接口，引脚定义如图 1 所示：



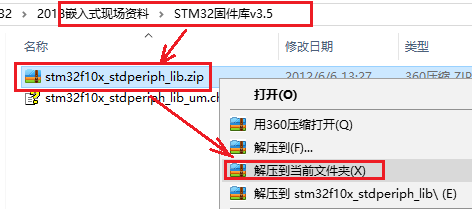
2）J1、J2 外部资源连接插针；引脚定义方式如下表所示：





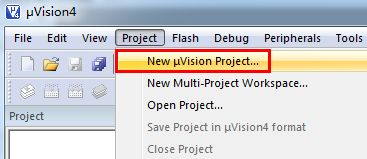
1. 新建工程：需要做两三遍

新建工程可以用keil4或者keil5做，建议打印出来照着做。本课程采用keil4讲解新建工程过程，keil4没有代码自动联想补全功能，所以如果使用keil5进行平常练习，建议关掉代码联想功能，以提高自己的编程能力。新建工程的过程中会用到固件库中的一些文件，先解压固件库v3.5。

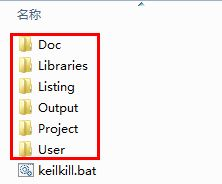


① 点击桌面 UVision4 图标，启动软件。如果是第一次使用的话会打开一个自带的工程文件，我们可以通过工具栏 Project->Close Project 选项把它关掉。

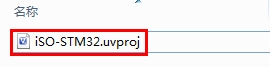
② 在工具栏 Project->New μVision Project新建我们的工程文件，见图。



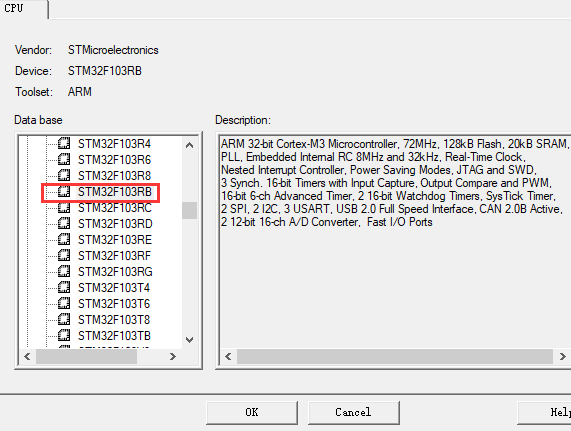
③ 我们将新建的工程文件保存在一个文件夹里面。首先我们建一个名为工程模板的文件夹，再在工程模板里面建几个文件夹：Doc、Libraries、 Listing、 Output 、Project、User，见图。（文件夹名称也可以根据自己的习惯更改）



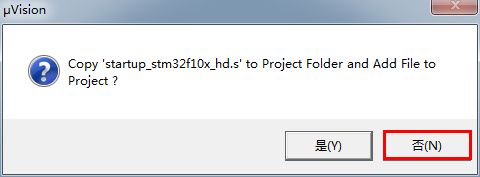
④ 在 Project 文件夹里面再建一个 RVMDK（uv4）文件夹，这个文件夹就是我们用来存放工程文件的，其中工程名字我在这里取为 ISO-STM32，见图。

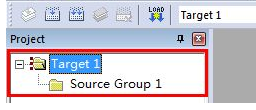


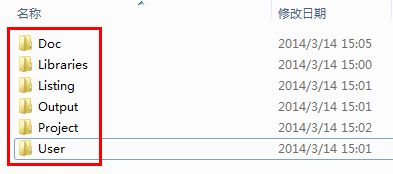
⑤ 接下来的窗口是让我们选择公司跟芯片的型号，见图。竞赛板用的芯片是 ST 公司的STM32F103RBT6，按如下选择即可。



⑥ 接下来的窗口问我们是否需要拷贝 STM32 的启动代码到工程文件中，这份启动代码在STM32 系列中都是适用的，一般情况下我们都点击是，但我们这里用的是 ST 的库，库文件里面也自带了这一份启动代码，所以为了保持库的完整性，我们就不需要开发环境为我们自带的启动代码了，稍后我们自己手动添加，这里我们点击否，见图。

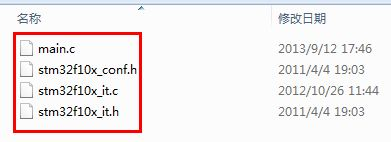


⑦ 此时我们的工程新建成功，但我们的工程中还没有任何文件，见图 3-7。接下来我们需要在我们的工程中添加所需文件。

⑧ 在工程模板文件夹下，我们在刚开始建的文件夹下添加我们所需要的文件，见图。

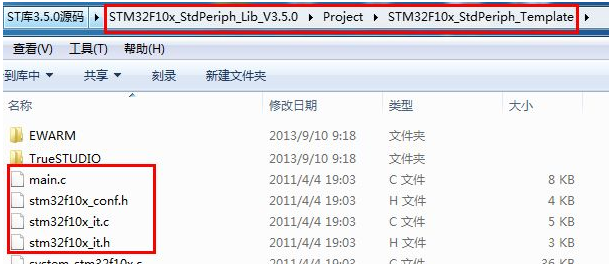
Doc 用来放一些说明文件，对工程做一些简单的说明，一般文件格式为 XXX.TXT，一般习惯用 readme.txt。

USER 用来存放用户写的驱动文件。 其中还包含了下面这四个文件：main.c、stm32f10x\_conf.h、stm32f10x\_it.c、stm32f10x\_it.h，见图。

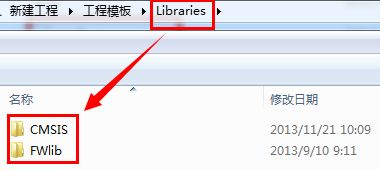


这四个文件来自于 ST 的库，我们需要拷贝过来即可，根据下面红色框中的路径即可

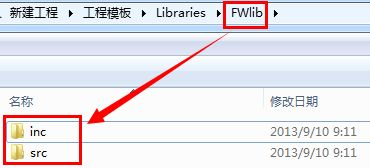
找到这三个文件，见图。



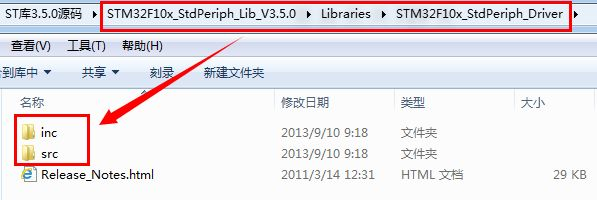
Libraries 用来放 ST 库里面最核心的文件，里面包含了 FWlib 和 CMSIS 这两个文件夹，见图。



FWlib:用来存放 STM32 库里面的 inc 和 src 这两个文件，这两个文件包含了芯片上的所有驱动，见图。

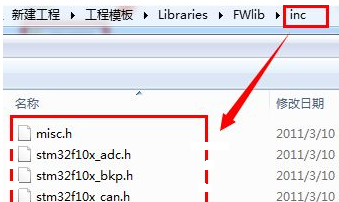


Inc 和 src 这两个文件夹也是直接从 ST 的库里面复制过来的，见图。



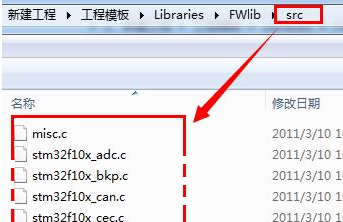
inc 里面是 ST 片上资源的驱动的头文件，见图。如要用到某个资源，则必须把

相应的头文件包含进来。

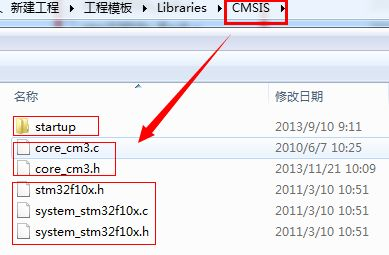


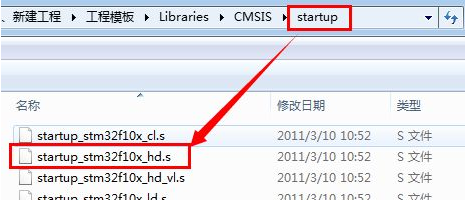
src 里面是 ST 片上资源的驱动文件，见图。这些驱动里面涉及了大量的 c 语言的

知识，是我们学习库的重点。

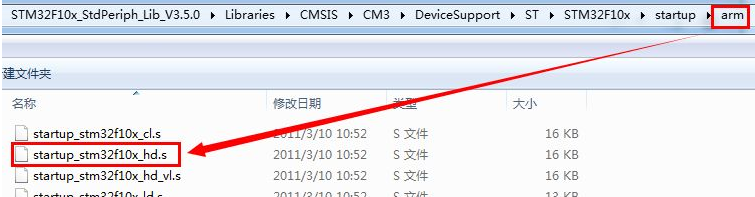


CMSIS 用来存放库为我们自带的启动文件和一些 M3 系列单片机通用的文件，见图。CMSIS 里面存放的文件适合任何 M3 内核的单片机。CMSIS 的缩写为: Cortex Microcontroller Software Interface Standard，是 ARM Cortex 微控制器软件接口标准，是ARM 公司为芯片厂商提供的一套通用的且独立于芯片厂商的处理器软件接口。





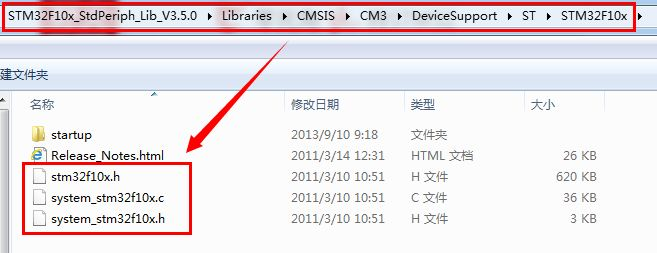
其中 startup 文件夹下的文件是从 ST 库里面复制过来的，路径如下红框中所示，见图。



core\_cm3.c 和 core\_cm3.h 这两个文件也可从 ST 库里面复制过来，路径如下，见图。



stm32f10x.h、system\_stm32f10x.c、system\_stm32f10x.h 这三个文件也可从 ST 库里面复制过来，路径如下，见图。



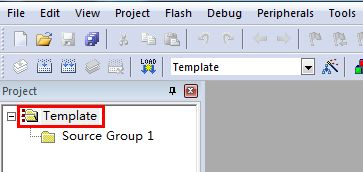
Output 用来保存软件编译后输出的文件，当我们编译出的.axf 或者.hex 文件就放在这个文件夹，.axf 是 JLINK 下载程序用到的文件，.hex 则为串口下载用到的文件，现在暂时为空。

Listing 用来存放一些编译过程中产生的文件，暂时为空。

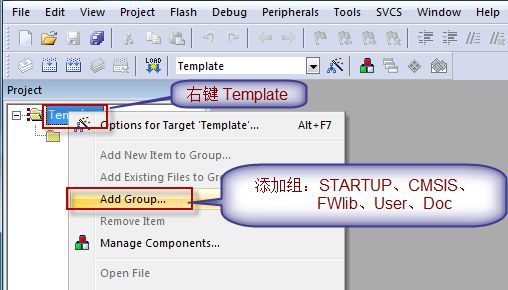
现在我们就已经基本完成了往模板里面添加库文件的工作。

⑨ 回到我们刚刚新建的工程界面，把刚刚那些文件夹里面的内容添加进工程，这样我们在 MDK 界面里面就可以管理那些文件了。

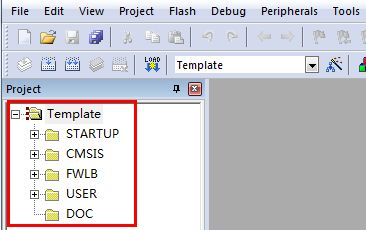
1. 把 Target 的名称改为 Template，见图。如果我们建的工程是 LED 的就改为LED，其实不改也可以，改了只是为了见名知义。



2. 往工程里面添加 5 个组文件夹，并命名为 STARTUP、CMSIS 、FWLIB 、USER 、DOC，见图。



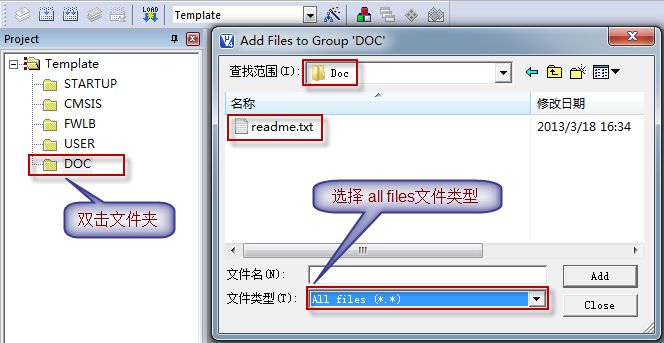
组添加完成之后如下，见图。



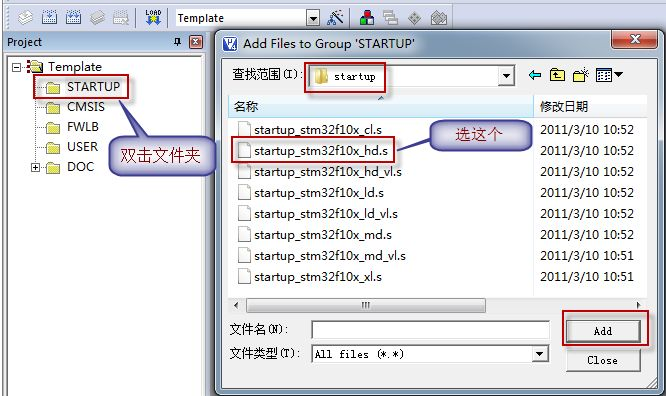
从名字就可以看得出 STARTUP 是用它来放我们的启动代码的，USER 用来存放用户自定义的应用程序，FWlib 用来存放库文件，CMSIS 用来存放 M3 系列单片机通用的文件。

3. 接下来我们往这些新建的组中添加文件，双击哪个组就可以往哪个组里面添加文件，如果该组里面已经有文件的话，双击则把组里面的文件都显示出来，然后再双击该组的话，则可以继续添加文件。

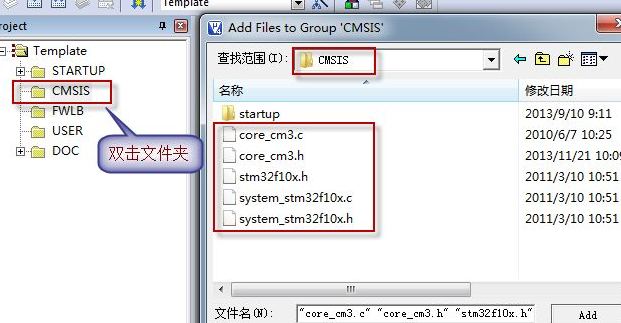
a 、往 DOC 文件夹内添加文件，见图。



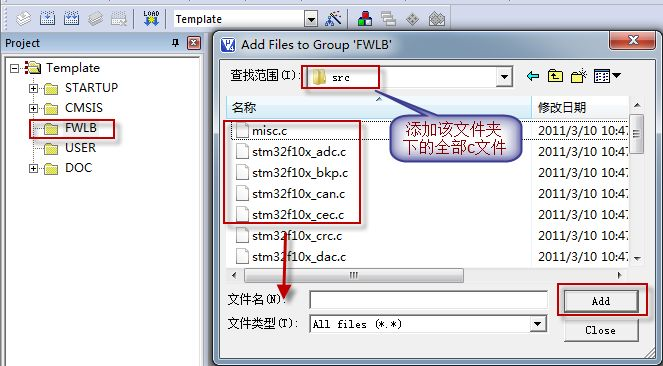
b 、往 STARTUP 文件夹内添加文件，见图。



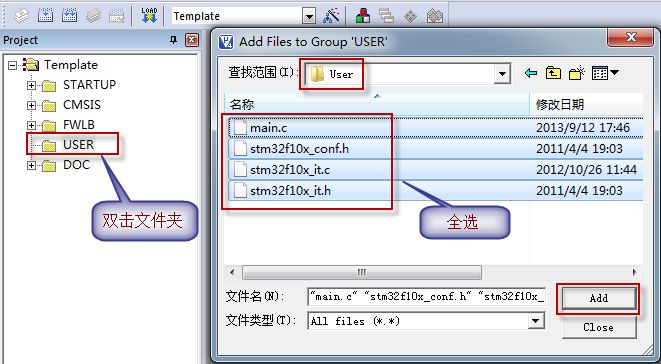
c 、往 CMSIS 文件夹内添加文件，见图。



d 、往 FWLIB 文件夹内添加文件，见图。



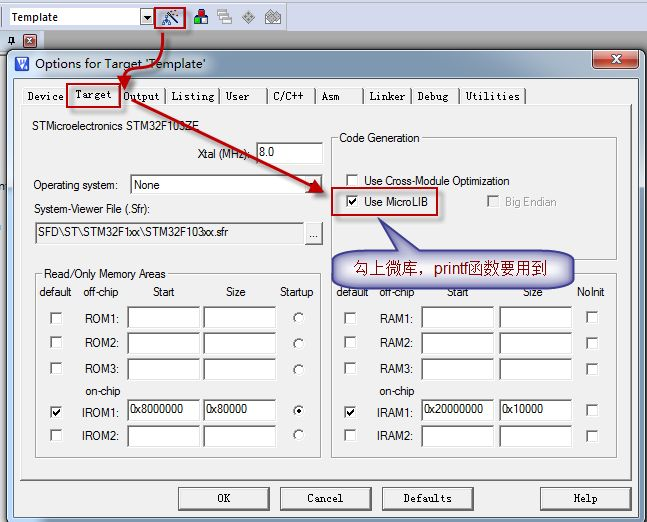
e、往 USER 文件夹下添加文件，见图。



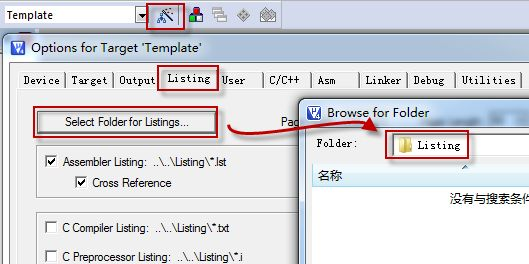
至此，我们的工程已经基本建好，但是还有一些工作需要完成。下面来配置一下

MDK 的配置选项，点击工具栏中的魔术棒按钮。

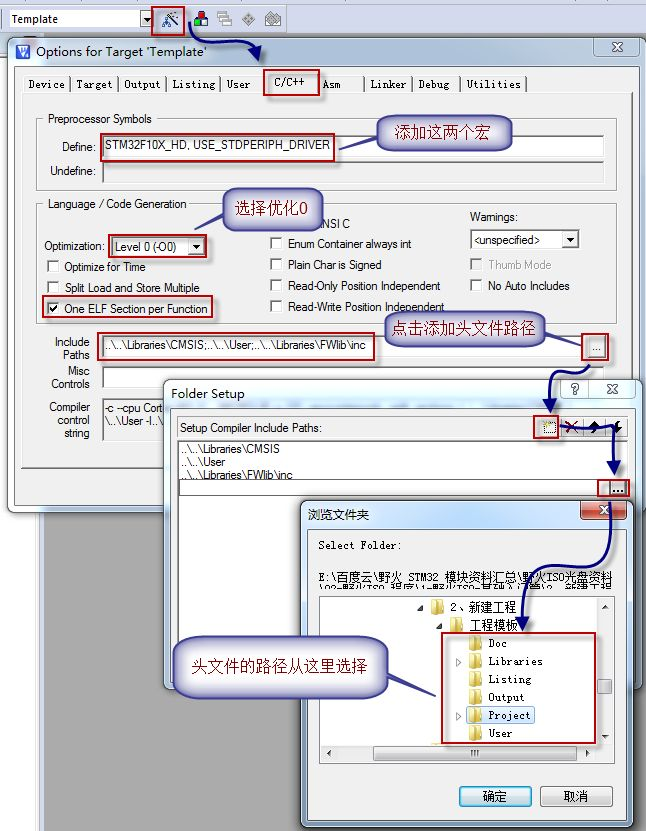
1）在 Target 这个选项卡中，把微库的勾打上，这样是为了后面的串口例程可以使用printf 这个函数，见图。

2) Output 选项卡如下设置，见图。点击 Select Folder for Objects... 设置编译后输出文件保存的位置。同时把 Debug Information 、Create HEX File 和 Browse information 这三个选项框也选上。

3) 在 Listing 这个选项卡中，点击 Select Folder listings…定位到模板中的 Listing 文件夹，见图。



4)在 C/C++这个选项卡上需要 设置的比较多，见图。

但当编译器在我们指定的路径下 搜索不到的话还是会回到标准目录去搜索，就像有些ANSIC C 的库文件，如 stdin.h 、stdio.h。

5）修改 main.c 文件，见图。因为刚刚我们的 main.c 文件是从官方库里面复制过来的，里面有许多的东西我们是不需要的，为了简化 main.c 文件，我们将修改如下，见图。



至此，我们的工程模板就建成了。学会新建工程，是学习 stm32 的第一步。

**三、实验总结**

## 实验六 STM32的LED灯

**一、实验目的：**

1.掌握外设时钟的配置方法

2.掌握STM32处理器GPIO的操作方法

\* 程序说明：使用程序前，确认LED相关引脚已经通过跳线正确连接。

**二、实验原理**

1、GPIO配置方法

2、外设时钟配置方法

实现功能：关闭所有LED灯，然后让两个LED灯轮流闪烁。

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验七 LCD显示例程

**一、实验目的：**

1.掌握Systick 的配置和使用方法

2.掌握中断服务程序设计方法

3.熟悉LCD例程相关函数

\* 程序说明：使用程序前，确认LED相关引脚已经通过跳线正确连接。

**二、实验原理**

1、中断概念、中断分组和中断优先级

2、Systick 的配置和使用方法

3、LCD相关函数

实现功能：将systick抢占优先级设置为1

**三、实验过程截图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验八 STM32外部中断

**一、实验目的：**

STM32处理器外部中断的操作与配置方法

\* 程序说明：1.使用程序前，确认按键相关引脚已经通过跳线正确连接

\* 2.B1-PA0 B2-PA8 B3-PB1 B4-PB2

**二、实验原理**

外部中断的操作与配置方法

实现功能：

（1）屏幕初始化显示如下：

第1行：" KEY TEST DEMO "

第3行：" Press the button... "

（2）当按键B1-B4某一个被按下时，分别显示" ButtonN pressed..."（N表示对应的按键号）。

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验九 STM32基本定时器

**一、实验目的：**

掌握STM32定时器功能和配置方法

\* 程序说明：通过定时器TIM3 50毫秒中断一次,LED每隔1S移动一位

**二、实验原理**

STM32定时器功能和配置方法

实现功能：通过定时器TIM3 50毫秒中断一次,LED每隔1S移动一位。

（1）屏幕初始化：

第1行：" TIMER DEMO "

第3行：" See The LEDs! "

（2）在LCD上继续显示点亮的灯的序号。

第6行： " -- LED ON:LD\* " （\*表示灯的序号）

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验十 PWM输出

**一、实验目的：**

掌握STM32定时器功能和配置方法

\* 程序说明：配置TIM2 CH2、CH3为PWM输出模式，CH2输出1KHZ，占空比50%信号，

\* CH3输出1KHz，占空比70%信号

**二、实验原理**

STM32定时器PWM输出模式的功能和配置方法

实现功能： 定时器TIM2 的CH2输出1KHZ，占空比50%信号，TIM2 的CH3输出1KHz，占空比70%信号。

（1）屏幕初始化：

第1行：" TIMER DEMO "

第3行：" TIM2 PWM MODE "

第6行："PA1-PWMVALUE:50% "

第8行："PA2-PWMVALUE:70% "

（2）用逻辑分析仪、示波器分别观察PA1、PA2的波形。

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验十一 STM32串口

**一、实验目的：**

掌握STM32 USART的操作与配置方法

\* 程序说明：1. 使用程序前，确认USART2引脚PA3、PA2已经通过跳线正确连接。

\* 2.通过串口调试工具显示连接PC机与串口调试软件，将波特率设定为19200

\* 3.发送字符串"asdasdad"+x,在LCD第7行显示asdasdad,以'x'位结束标志

**二、实验原理**

STM32 USART的操作与配置方法

实现功能：

（1）通过串口调试工具显示连接PC机与串口调试软件，将波特率设定为19200。

（2）从开发板发送“Welcome to HBEU”，在电脑的串口助手上查看接收的字符。

（3）在串口助手中发送字符串"asdasdad"+x到开发板,在LCD第7行显示asdasdad,以'x'位结束标志。

LCD显示要求：

（1）屏幕初始化：

第1行：" USART DEMO "

第3行：" Receive & Display "

（2）在LCD上继续显示从串口接收到的字符串。

第6行： "Receive:"

第7行：接收到的字符。

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验十二 STM32模数转换

**一、实验目的：**

STM32处理器ADC的操作与配置方法

\* 程序说明：调节电位器R37输出电压，观察LCD上显示的数值

\* ADC工作模式配置:

\* ADC1 独立模式,单次转换

**二、实验原理**

ADC的操作与配置方法

实现功能：调节点位器R37输出电压，要求LCD显示如下：

1. 屏幕初始化：

第1行：" ADC DEMO "

第3行：" PB0-ADC channel 8 "

（2）在LCD上继续显示转换的数值（小数点后保留3位）

第7行：： " ADC Value:数值 "

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验十三 EEPROM存储器

**一、实验目的：**

1.掌握EEPROM AT24C02的操作方法

\* 2.掌握IIC总线协议和通讯原理

\* 程序说明：1.从EEPROM 0xff地址读出数据dat,++dat后重新写回0xff地址

\* 2.通过LCD显示0xff地址存储的数据,每次复位后，数值加1

**二、实验原理**

1、EEPROM AT24C02的基本原理和操作方法

2、IIC总线协议和通讯原理

实现功能：从EEPROM的0xff地址读出数据dat，++dat后重新写回0xff地址，通过LCD显示0xff地址存储的数据,每次复位后，数值加1

（1）屏幕初始化：

第1行：" I2C DEMO "

第3行：" AT24C02 R/W "

（2）在LCD上继续显示当前数据值。

第6行：： "ADDR:0xFF,VAL:\*" （\*表示dat的当前值）

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**

## 实验十四 STM32实时时钟

**一、实验目的：**

掌握STM32外设RTC的配置和使用方法

\* 程序说明：CT117E竞赛平台无外部低速晶振，因此例程中使用内部低速振荡器LSI作为

\* RTC时钟。

**二、实验原理**

STM32外设RTC的配置和使用方法

实现功能：使用内部低速振荡器LSI作为RTC时钟，实现时、分、秒显示以及整点报时功能（蜂鸣器响5声）。

LCD显示要求：

（1）屏幕初始化：

第1行：" RTC DEMO "

第3行：" RTC\_Calendar\_Test "

（2）在LCD上继续显示时间。

第7行： " Time：\*\*：\*\*：\*\*" （\*\*表示时分秒））

初始时间为：10:59:55

**三、实验原理图：**

**四、实验主要程序**

**五、实验现象**

**六、总结**