

HC6800-EM3
使用操作手册

普中科技

www.prechin.com

www.prechin.net

电话：0755-61190227

第1章 HC6800-EM3硬件知识

- 1.1 HC6800-EM3 硬件资源介绍
- 2.1 HC6800-EM3 跳线帽说明

第2章 软件安装及使用

- 2.1 USB驱动程序
- 2.2 51 Keil安装及其使用
- 2.3 ARM RVMDK eil安装及其使用
- 2.4 如何打开第一个Keil例程源码
- 2.5 第一次给51单片机烧录程序。
- 2.6 如何建立自己Keil工程
- 2.7 Keil工程设置
- 2.8 给STM32 ARM核心板烧写程序

第3章 硬件原理分析

- 3.1 51打片机和STM32最小系统
- 3.2 液晶屏显示
- 3.3 动态数码管和静态数码管
- 3.4 LED灯
- 3.5 独立按键
- 3.6 IIC总线
- 3.7 EEPROM存储器
- 3.8 Ds1302
- 3.9 AD/DA模数数模转换
- 3.10 步进电机
- 3.11 继电器
- 3.12 蜂鸣器
- 3.13 Rs232通信
- 3.14 温度DS1302
- 3.15 红外接收与发射
- 3.16 矩阵键盘

简介

HC6800-EM3是一款功能超强的32位ARM处理器和51二合一学习板，整个开发板分底板和核心板。核心板ARM处理器是STM32F103C8T6。STM32具有性价比极高优点，它采用ARM公司的、具有先进架构的Cortex-M3内核，出色的实时性能、优越的功效、高级的和创新实用型外设、最大的集成性、配置丰富灵活、功耗超低、易于开发，加速了面市时间等优点，STM32已经被大众企业所接纳。学习ARM从STM32开始是最佳的选择。底板资源继承了HC6800E开发板资源，并增加了3.3V和5V的双电源系统并又增加了2路电源接口，基本可以通吃所有器件或设备的供电，为做不同电压实验提供了更多方便，还增加了一些锁存器的功能。从51过渡到ARM嵌入式系统,HC600-EM3是您明智的选择

STM32F103系列增强型资源介绍

1. STM32采用 ARM32位 Cortex-M3的 CPU,工作频率可以达到72M 1.25DMIPS/MHZ
2. 0等待访问存储器
3. 能够有单周期乘法和硬件除法
4. 32K到128K的FLASH
5. 6-20K的SRAM
6. 时钟、复位和供电管理
7. 2.0至3.6伏供电和I/O管脚
8. 上电 / 断电复位(POR / PDR)、可编程电压监测器(PVD)
9. 内嵌4至16MHz高速晶体振荡器
10. 内嵌经出厂调校的8MHz Rc振荡器
11. 内部40kHz的RC振荡器
12. PLL供应CPU时钟
13. 带校准的32kHz RTC振荡器
14. 低功耗
 - a. 睡眠、停机和待机模式
 - b. VBAT为RTC和后备寄存器供电
15. 调试模式, 串行线调试(SWD)和JTAG调试接口
16. DMA
 - a. 7通道DMA控制器
 - b. 支持的外设: 定时器、ADC、SPI、I2C和USART
17. 1个12位模数转换器, 1us转换时间(16通道)
18. 多达80个快速I/O口, 26/37/51/80个多功能双向5V兼容的I/O口, 所有I/O口可以映像到16个外部中断

19. 多达6个定时器，多达3个16位定时器，

a. 每个定时器有多达4个用于输入捕获 / 输出比较 / PWM或脉冲计数的通道

b. 2个16位看门狗定时器(独立的和窗口型的)

c. 系统时间定时器：24位自减型

22. 多达7个通信接口

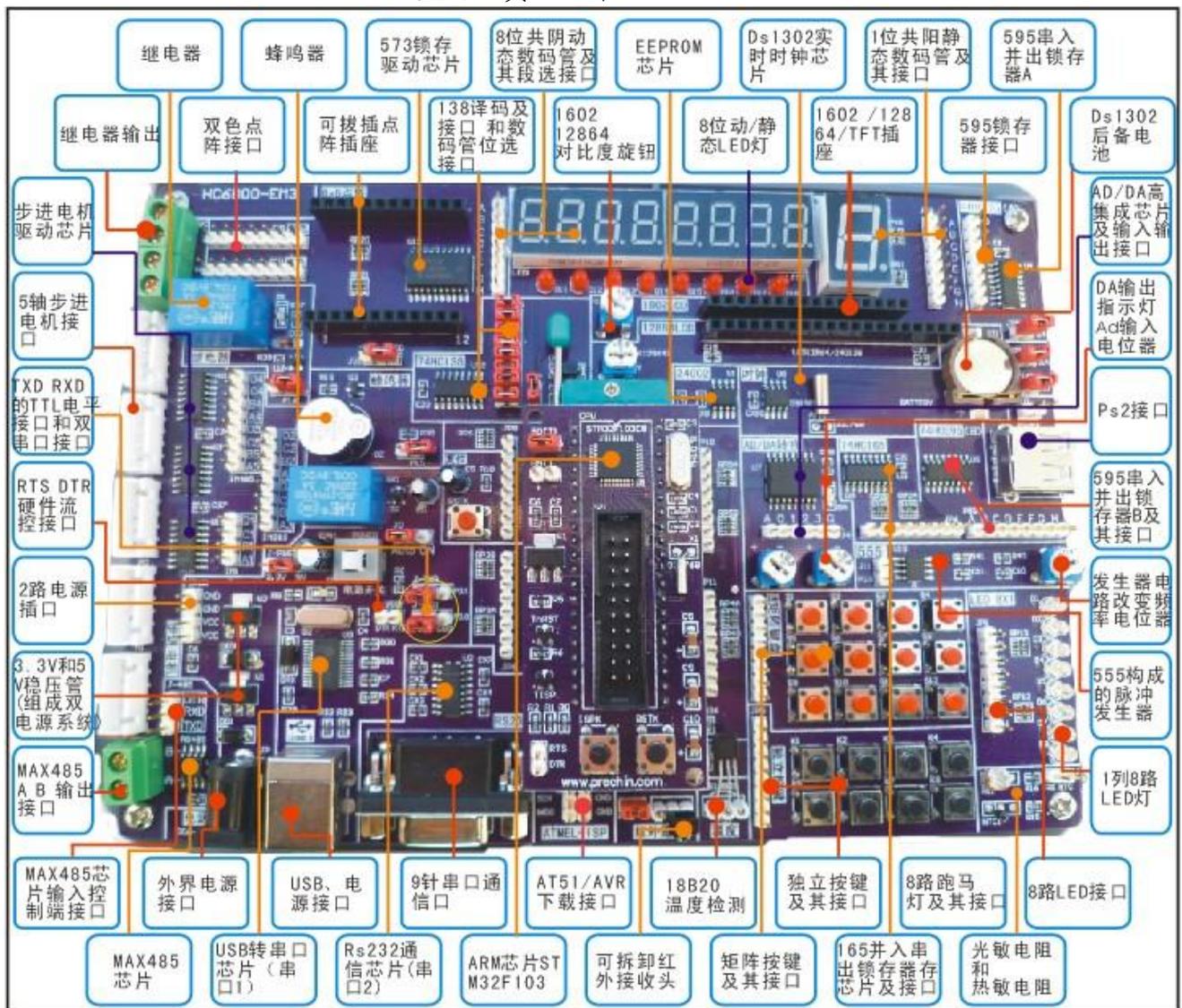
a. 多达2个I2C接口 (SMBus/PMBus)

b. 多达3个USART接口，支持ISO7816, LIN, IrDA接口和调制解调控制

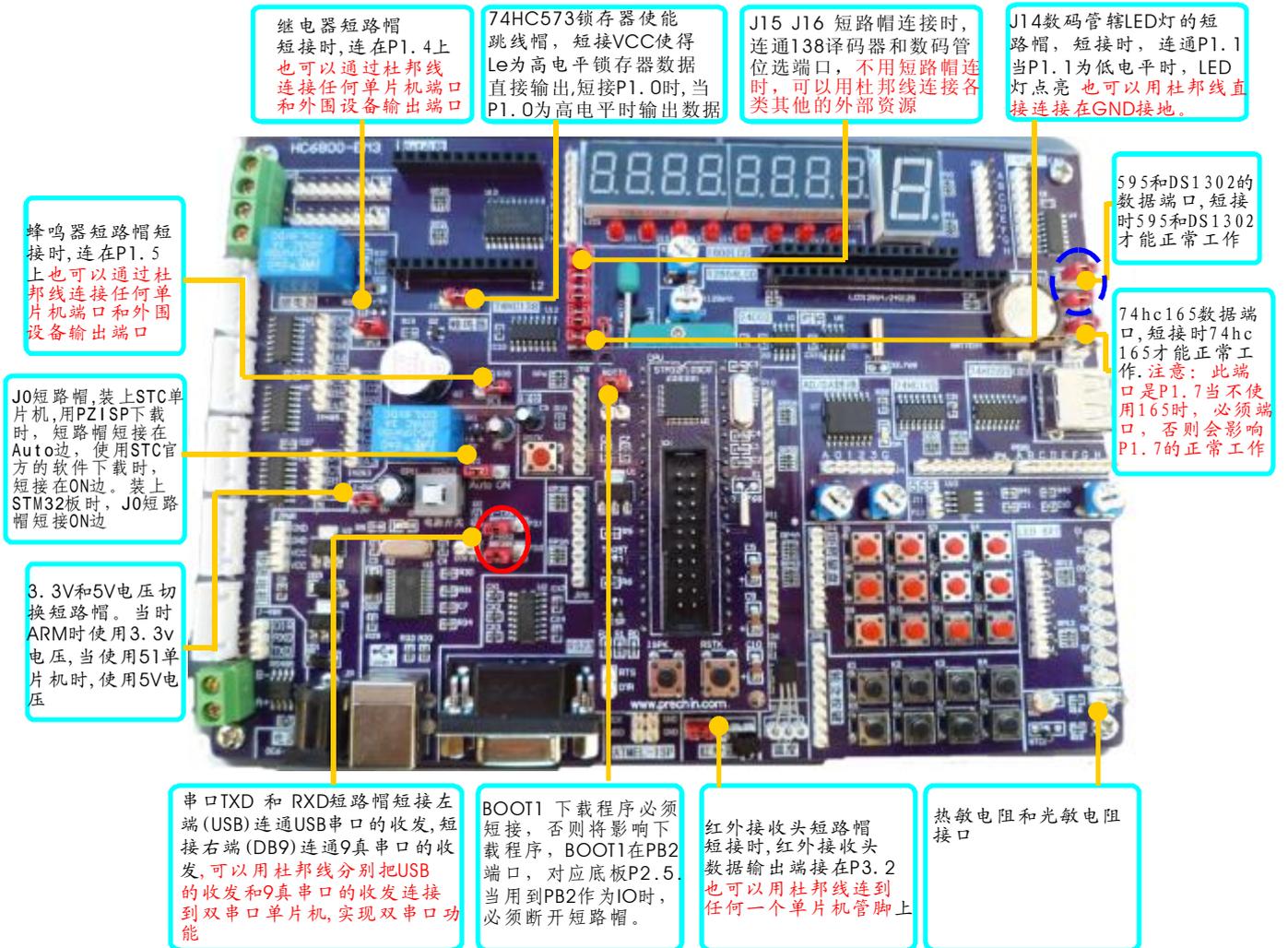
c. 多达2个SPI同步串行接口 (18兆位/秒)

23. 1个12位模数转换器，1us转换时间(16通道)

整板资源介绍



跳线帽说明

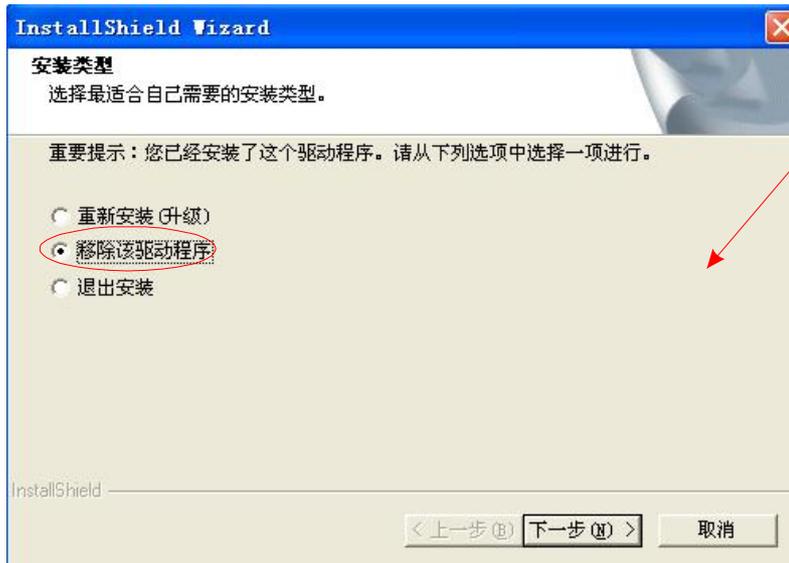


USB驱动程序安装

在光盘中找到“安装USB驱动程序”的文件夹,在文件夹中,有XP和WIN7两种驱动程序。

XP安装演示:

双击USB驱动程序的安装程序



弹出此对话框,说明你电脑里已经安装了此类驱动程序,此时强烈建议选择"移除该驱动程序",根据提示重启计算机。

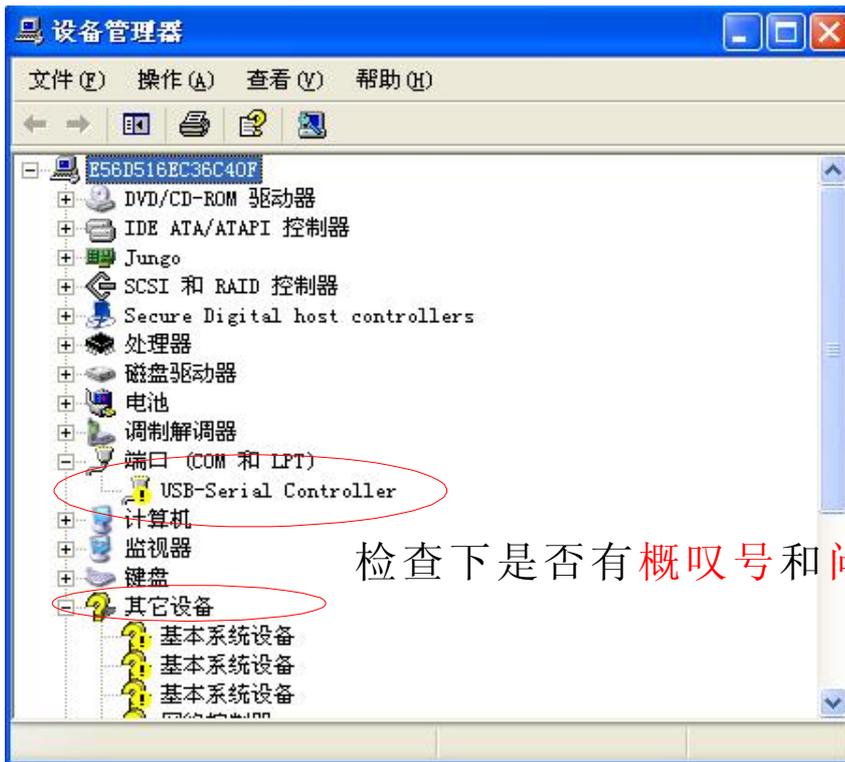
卸载成功后,重新安装我们光盘中的驱动程序,我们提供驱动程序是兼容性最好的版本



点“下一步”即将开始安装,完成后,按照提示重启计算机

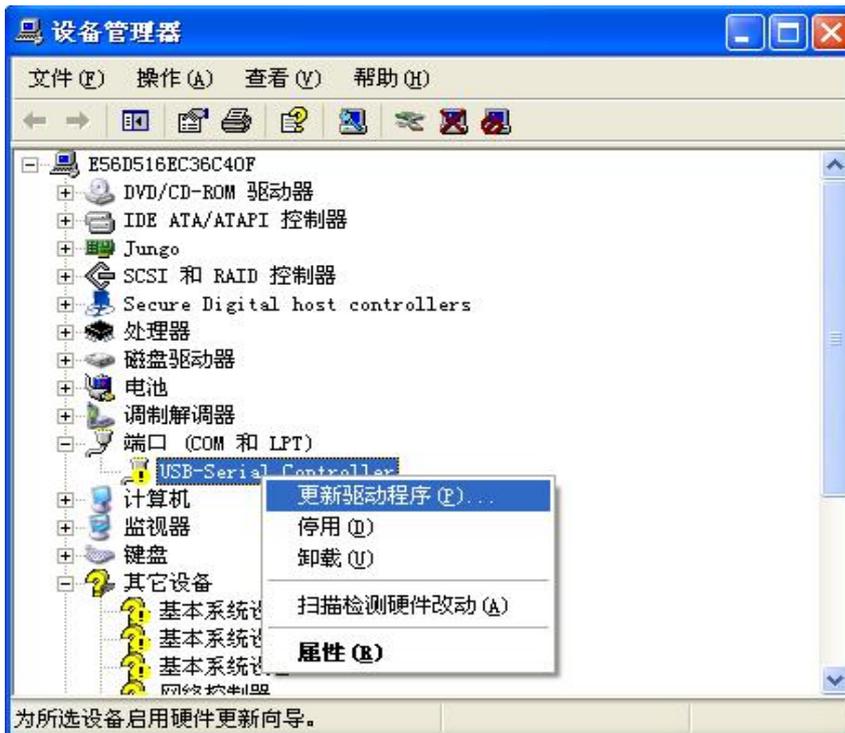
安装不成功解决办法

安装完驱动程序后,发现不能正常烧录程序时,此时打开设备管理器

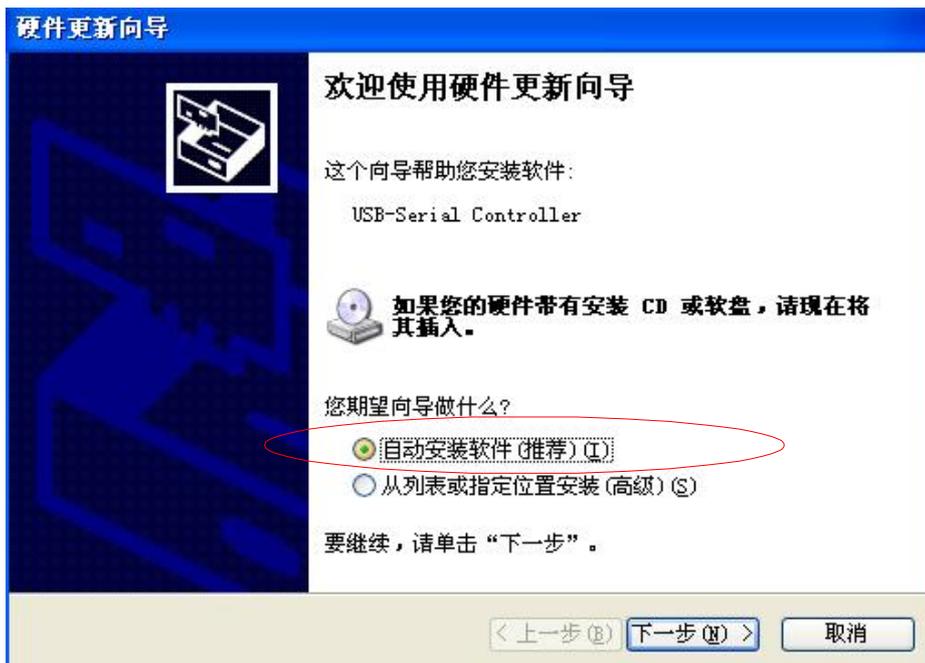


检查下是否有概叹号和问号出现

此时,重新启动计算机,再次打开设备管理器,点击右键选择"更新驱动程序"



选择“自动安装软件（推荐）”



出现此对话框，正在搜索，匹配的驱动程序



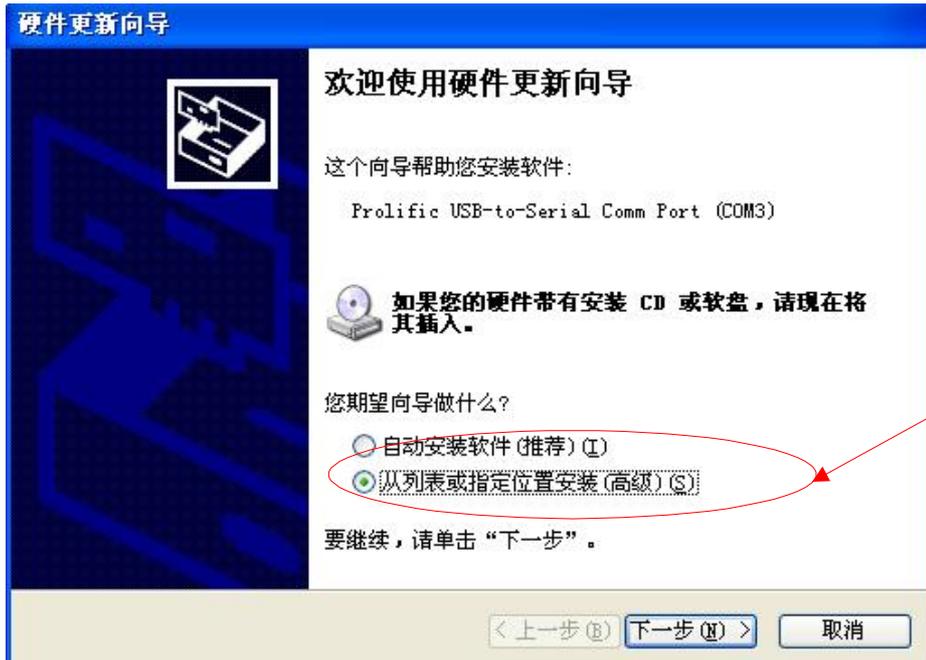
如果弹出下面对话框，表示安装完成，点击"完成"



如果弹出此对话框，安装不成功。用安装软件自带的“移除”功能卸载后，重新安装



对于一些电脑由于各种各样的原因造成驱动程序无法安装成功，
最终的解决办法：



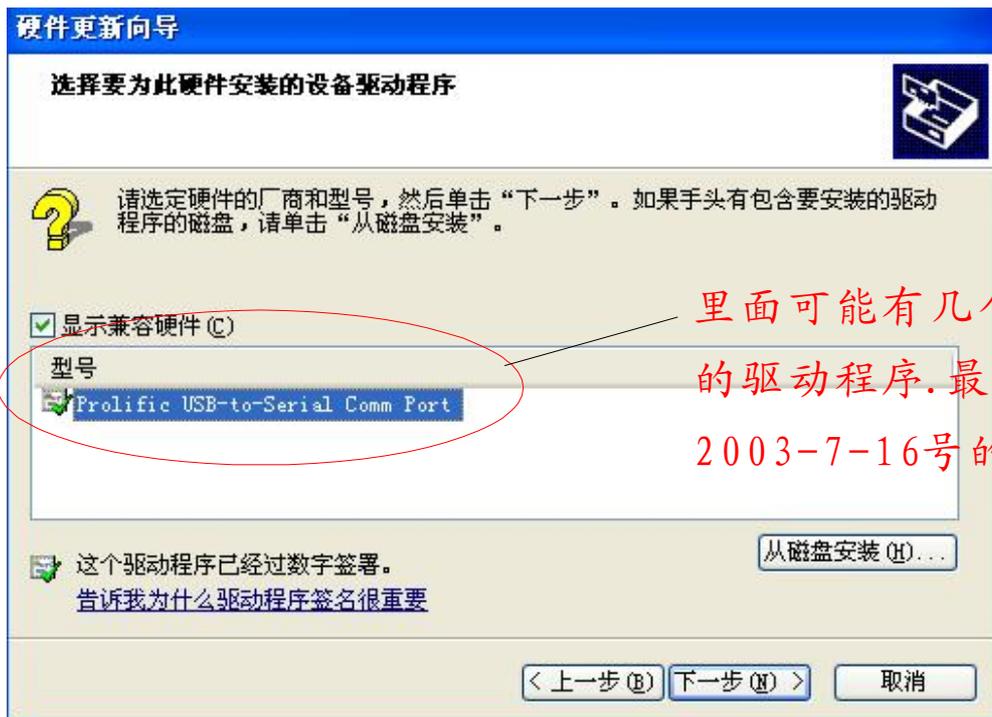
点"下一步"



点"下一步"



点"下一步"



里面可能有几个不同版本的驱动程序. 最好选择 2003-7-16号的驱动

点"下一步, 应该可以解决安装问题."

安装Keil开发软件安装

在光盘中找到"安装单片机开发工具"文件夹,在文件下面有"Keil2-Full"和Keil3_Full两个版本的开发软件,推荐使用Keil3-Full。

下面Keil3-Full安装演示

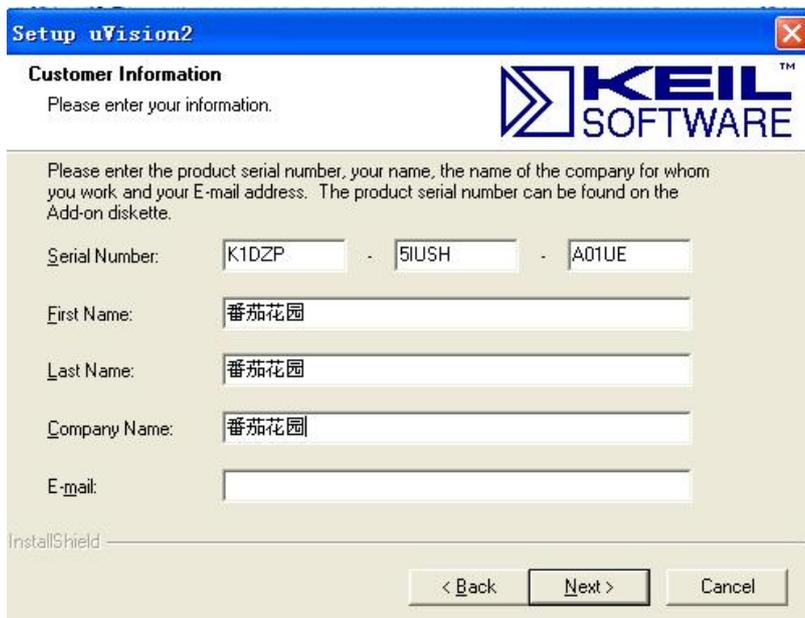
在setup文件夹中双击Setup.exe,即可启动安装程序



按照相应的提示走到这一步时,注意选择"Full Version"



按照相应提示来到这一步,需要输入序列号



序列号在"安装说明"中



打开"安装说明",可以找到下列的序列号

这组安装码可以使用keil C51:
Ident = Y1DZKM (这个号码已
SN = K1DZP-5IUSH-A01UE

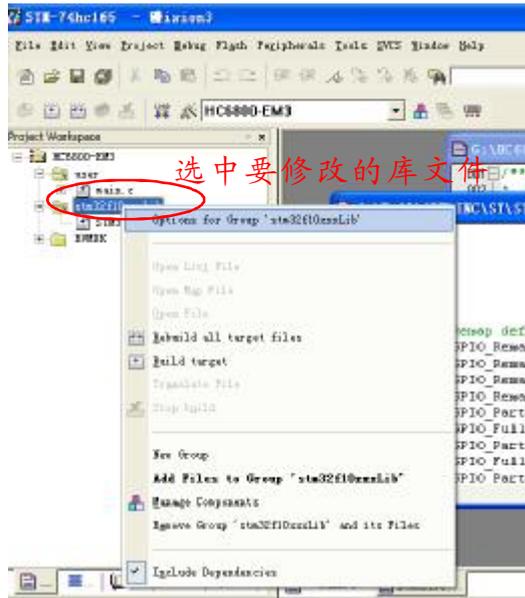
按照相应的步骤提示点击"下一步",即可完成软件的安装。

ARM-MDK的安装和使用

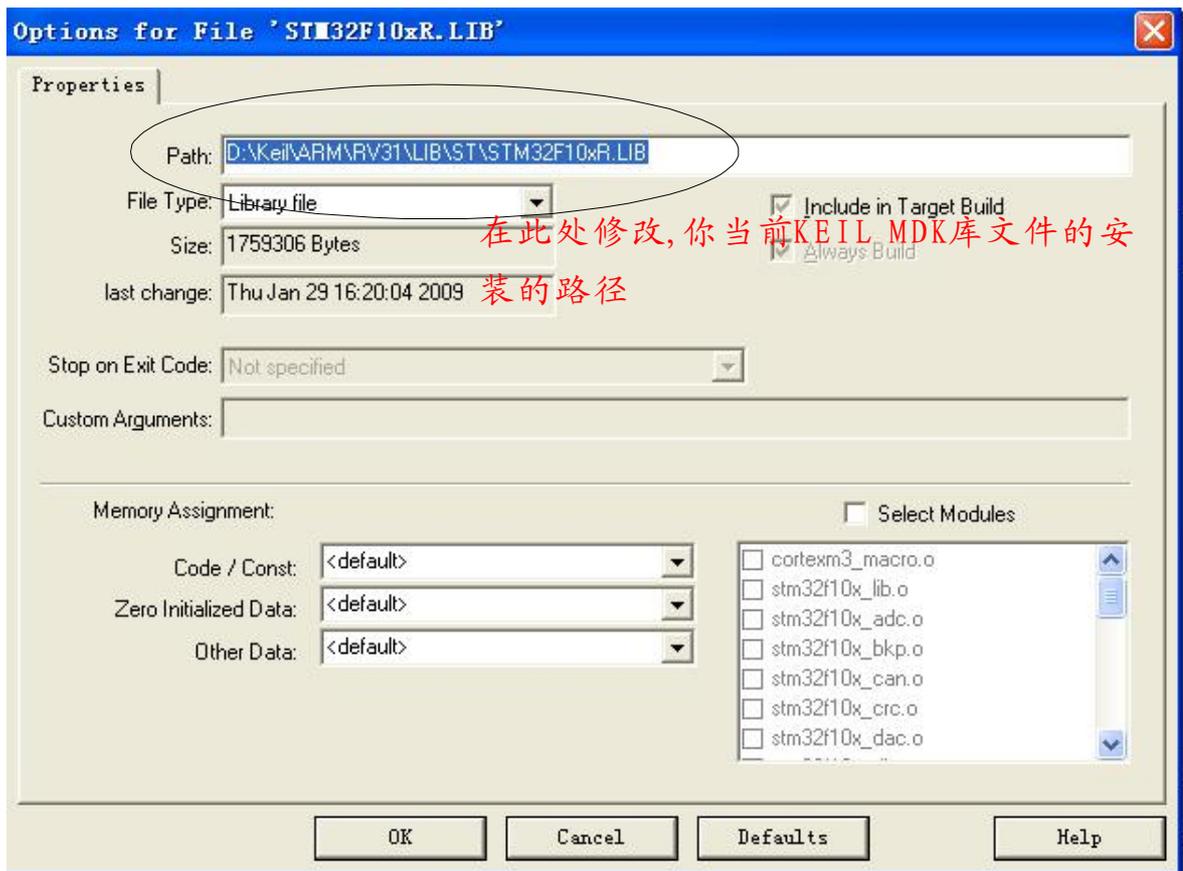
安装ARM 开发工具RVMDK时, 安装方法类似, 但是要注意: 51和ARM 安装路径不能一样。如果一样, 将会造成打开51工程和ARM工程, 出现软件使用混乱。请自己修改安装路径, 加以区别。

MDK库文件路径修改

MDK修改库文件路径，例子工程包含一些库文件，库文件一般在安装目录下面，而每个用户的KEIL MDK安装的路径不一样，致使同一个工程在不同的电脑里面，找不到库文件，这时需要手动修改库文件路径。以下图片演示：



为了方便学习，
HC6800-EM3中的
库文件大部分都拷贝
在例子工程下面。



区分ARM与51的Keil开发环境

51和ARM的开发环境极其相似,对初学这来说,如果装了开发51和开发ARM的软件后打开工程时,很容弄混淆。下面是开启工程时,欢迎界面的区别。



此图为 ARM开发软件

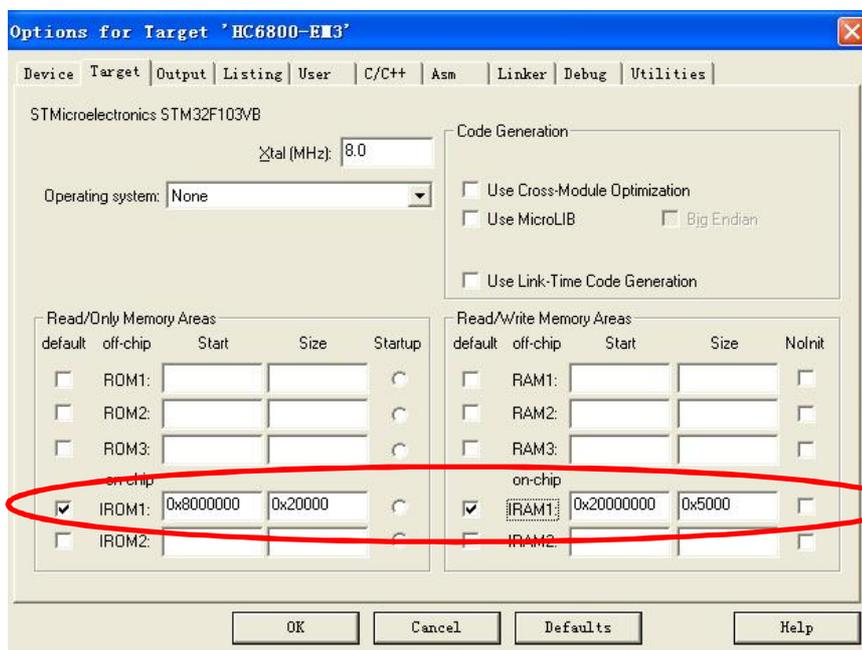
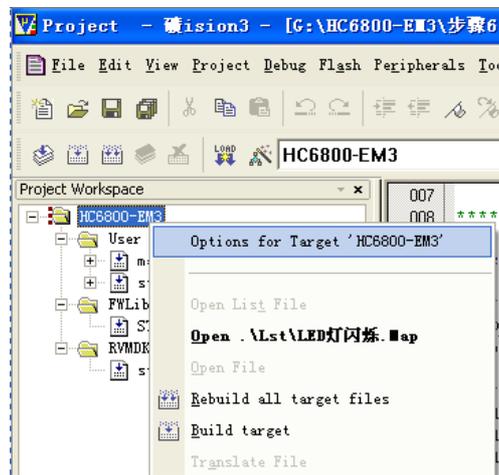


此图为 51开发软件

有时候不小心用51的KEIL软件打开了ARM工程后，造成ARM的KEIL的一些设置发生了变化，致使再用ARM的KEIL打开ARM工程时，出现下图的错误，而编译不成功。

```
Build target 'HC6800-EM3'  
compiling main.c...  
compiling stm32f10x_it.c...  
assembling stm32f10x_vector.s...  
*** Error: Referred Memory Range 'ROM1' is undefined.  
Target not created
```

这时需要重新设置KEIL 参考下图重新设置。



此两项选中
即可更正

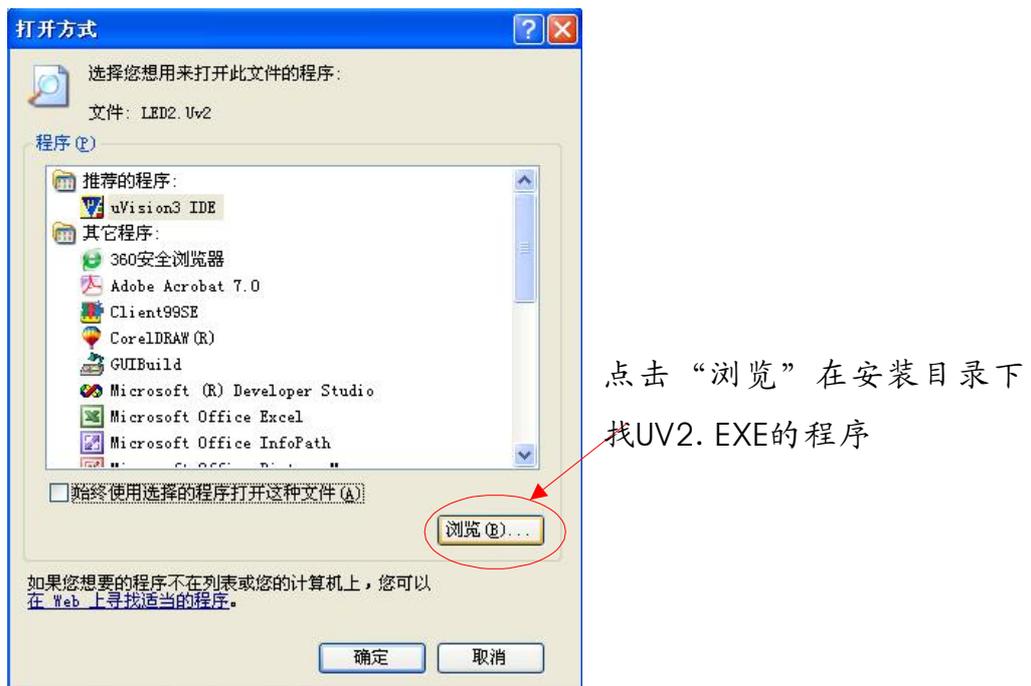
2.5 打开第一个Keil例程源码

安装好Keil开发软件，大家可以马上打开源码瞧瞧，里面有使用的相关详细说明。有些电脑安装好后可能关联不了Keil软件，致使初学者不知道怎么打开例子工程。这介绍手动关联的方法，打开步骤如下：

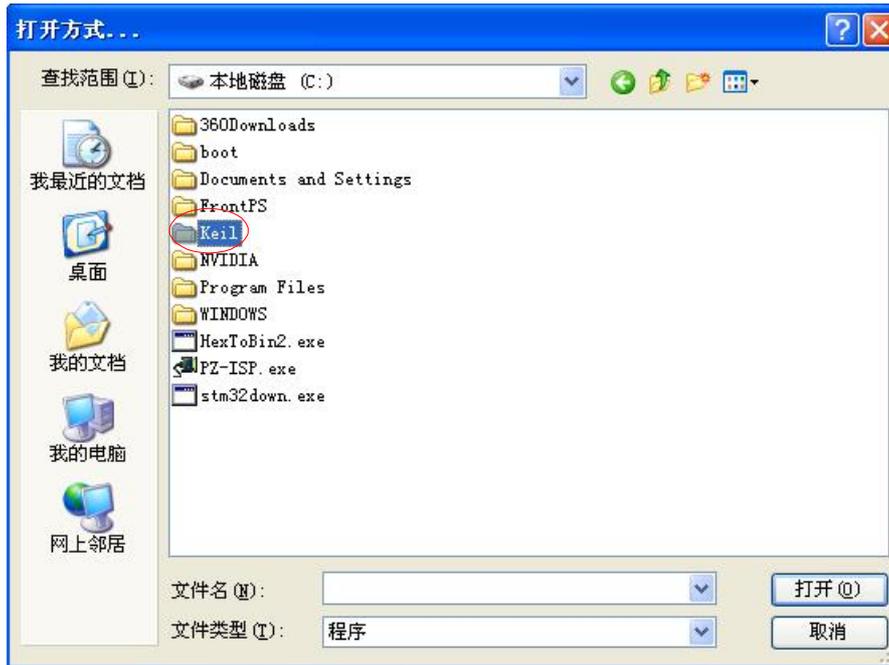
1. 首先在单片机例程里面随便找个例子程序，在工程下找到***.uv2的文件，



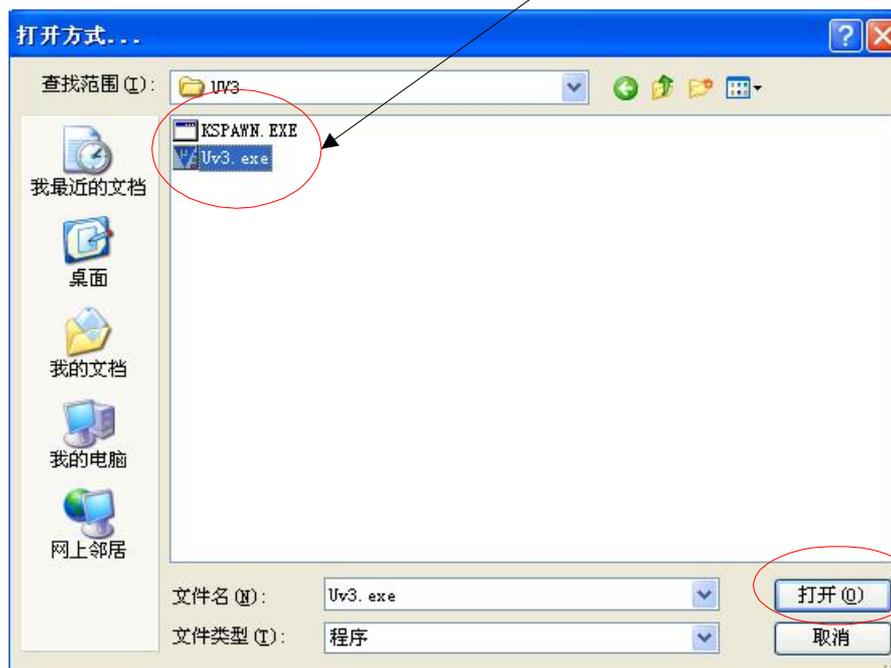
选择“打开方式”弹出下面对话框



如果你的Keil是安装在C盘下，你将会在C盘下找到Keil文件夹



用这个软件打开Keil例子工程



最后,回到这个对话框



注意:这个框要选中,这次操作后,就再也不用上述操作,直接双击那.UV2的文件就会自动打开例子工程

点“确定”后,例子工程中.UV2的文件的图标就会变成此
就可以直接打开工程文件



以后双击此图标

第一次给单片机烧录程序

烧录程序前,先检测一下条件是否满足:

1. 安装好USB驱动程序并确定串口号



2. 检查开发仪的J-RXD和J-TXD的两个短路帽是否在USB端(左端)。

注意:使用板载USB的串口时,按上述连接,如果使用9针串口时,J-RXD和J-TXD的两个短路帽要跳到DB9端(右端)

3. 检查J0短路帽短接情况是否与烧写软件匹配。

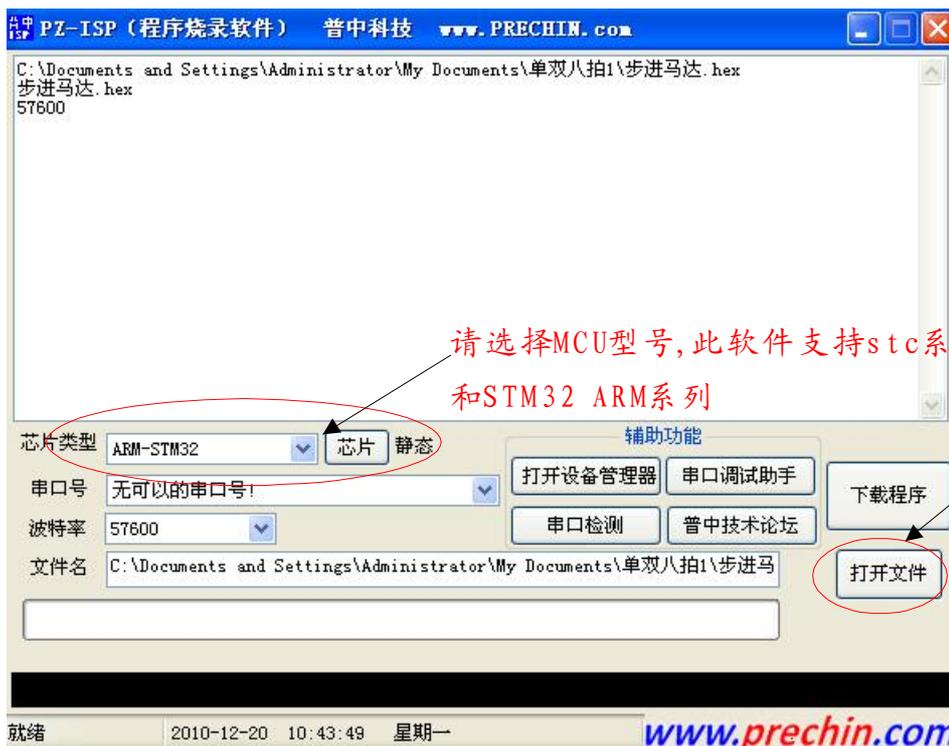
PZISP烧录软件(普中): J0的短路帽需要短接

STCISP烧录软件(STC官方): J0短路帽需要断开

4. 打开电源开关



5. 打开PZISP烧写软件(推荐使用),此软件具有自动断电功能



请选择MCU型号,此软件支持stc系列
和STM32 ARM系列

通过此按钮
用来选择单片
机的程序文件
HEX文件

用鼠标点击"打开文件"按钮,即可弹出下面对话框,然后在相应的工程(光盘中的例程或自建的工程)里面选择.HEX的文件.



当使用PZISP时，如果启动出现下图错误,这是由于你系统缺少文件引起.



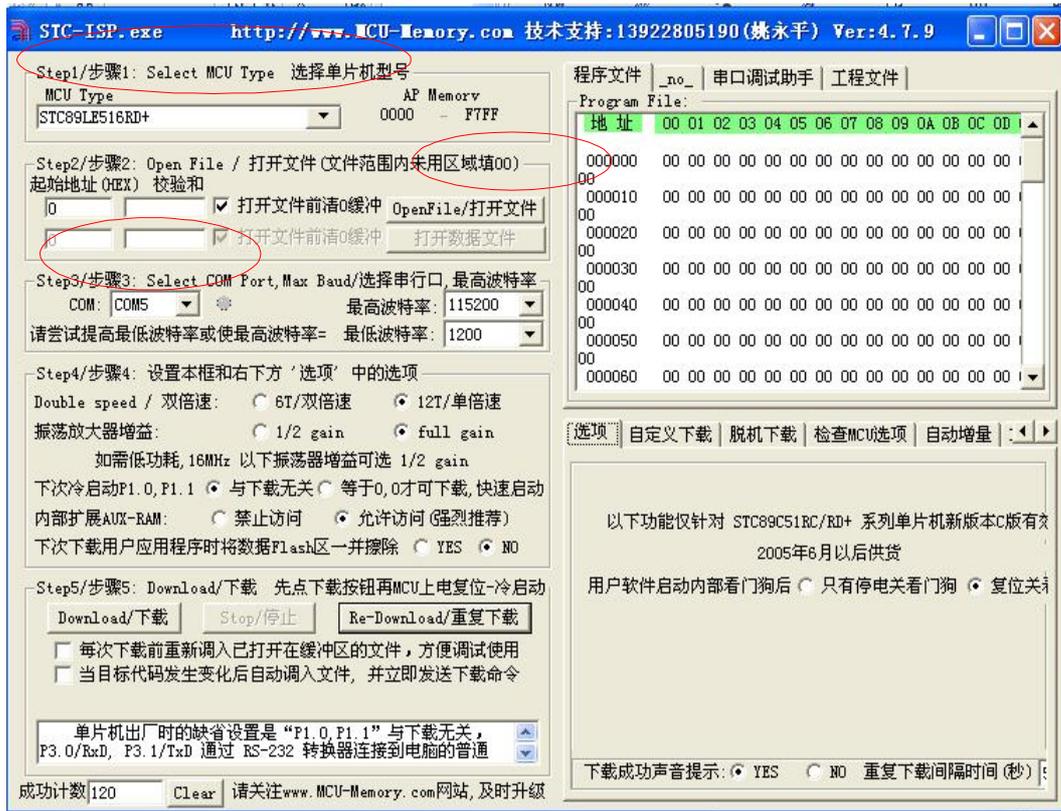
解决办法时,需要把这个文件安装你的系统中.我们光盘提供了此安装文件.

程序烧录工具\VCredist_x86.exe

双击下图的安装文件，安装完成后，即可是用PZISP软件。



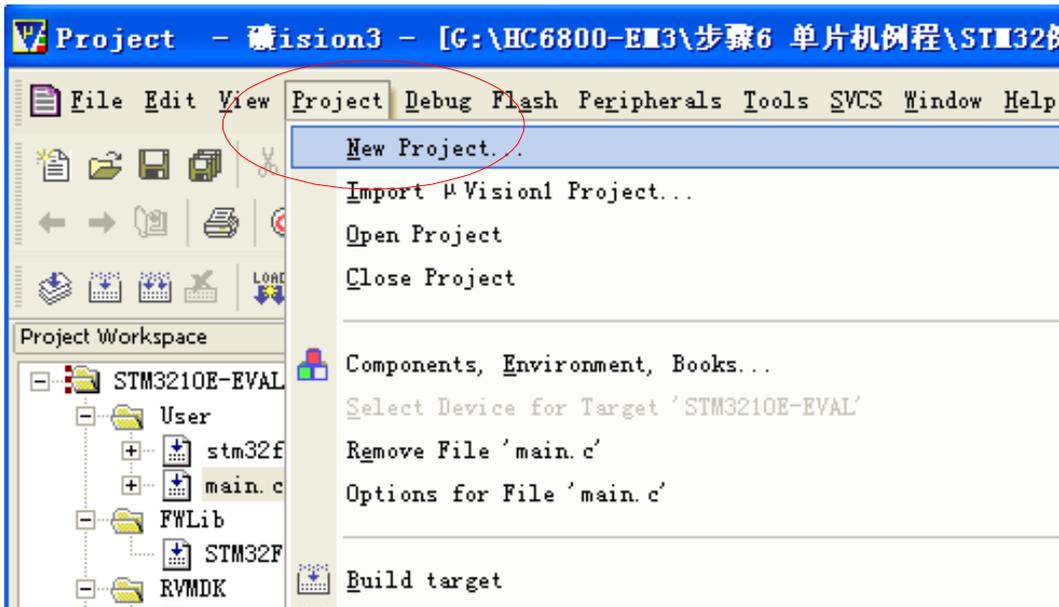
6. 使用STCISP不能自动断电,需要冷启动才能完成程序烧录。**冷启动方法: 先关掉电源开关,再点"下载",然后马上打开电源**



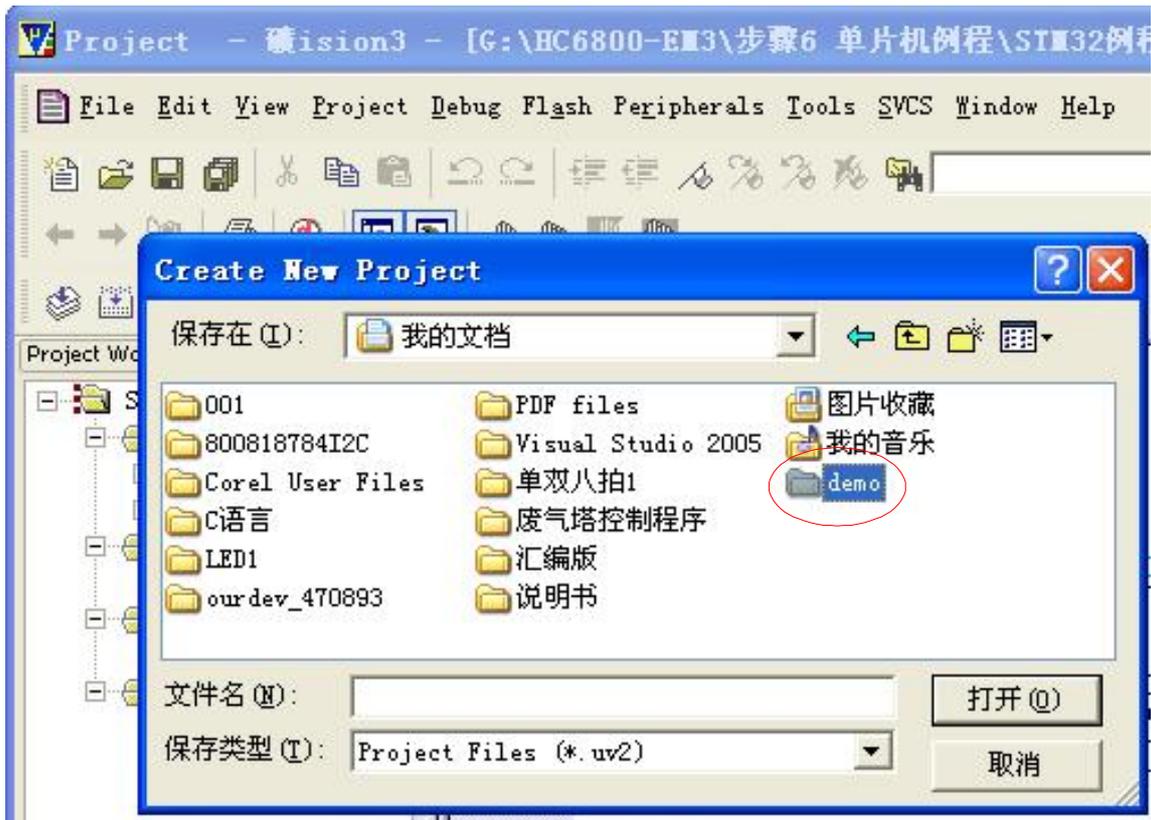
设置好
红圈中
的参数
即可

2.7 如何建立自己Keil工程

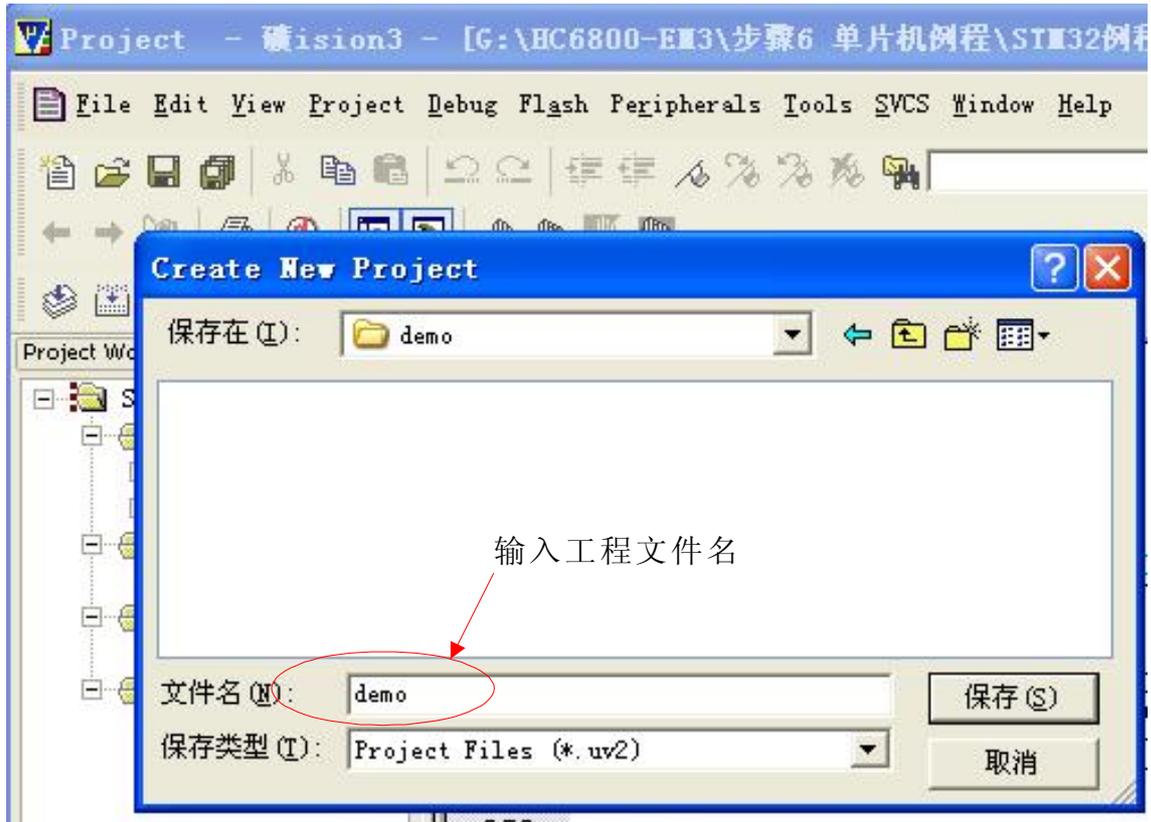
1. 打开Keil软件, 选择"project"下的"NewProject"



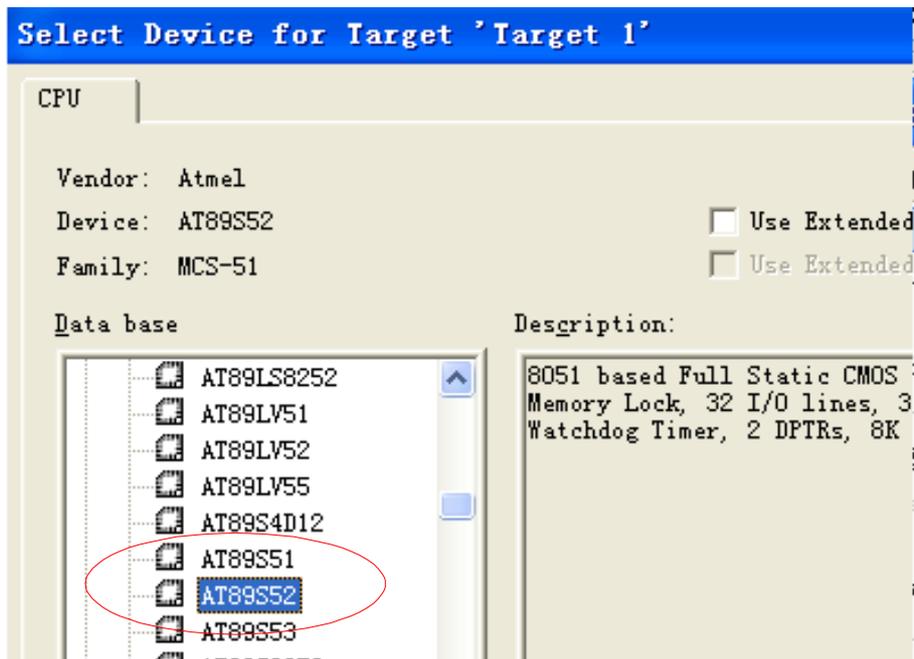
1. 新建一个demo的文件夹



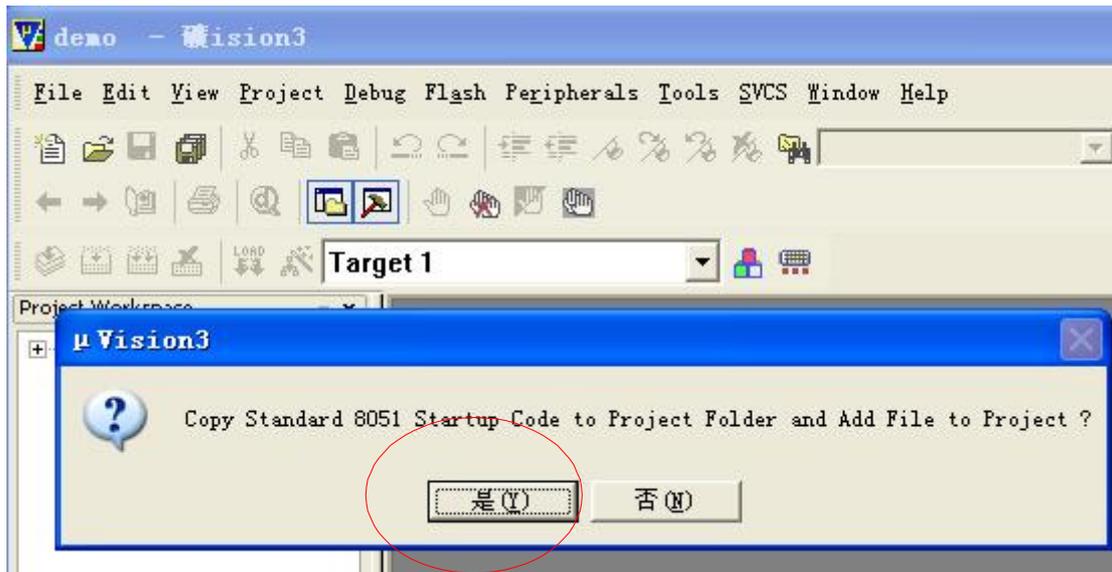
打开demo文件,输入要见工程的文件名,然后点"保存"



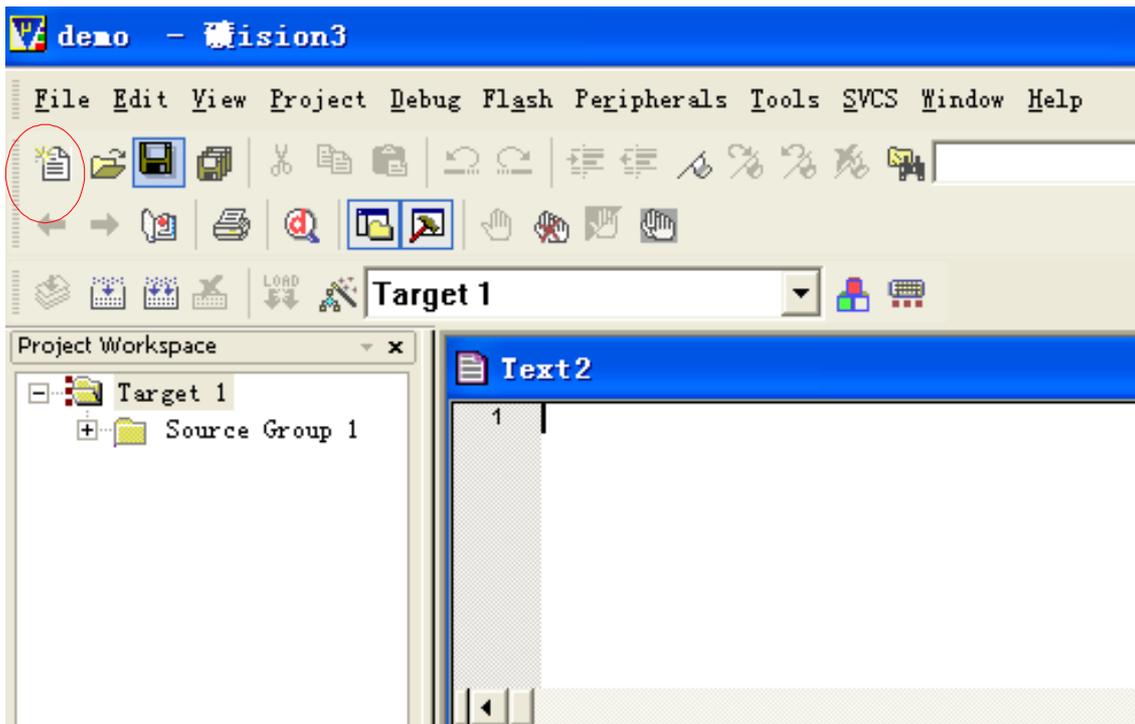
开发仪使用的是STC单片机, STC在keil里面没有对应的选择项,这里选择AT89S52, 这里选择别的型号也没有关系,都是51的内核,头文件都差不多。



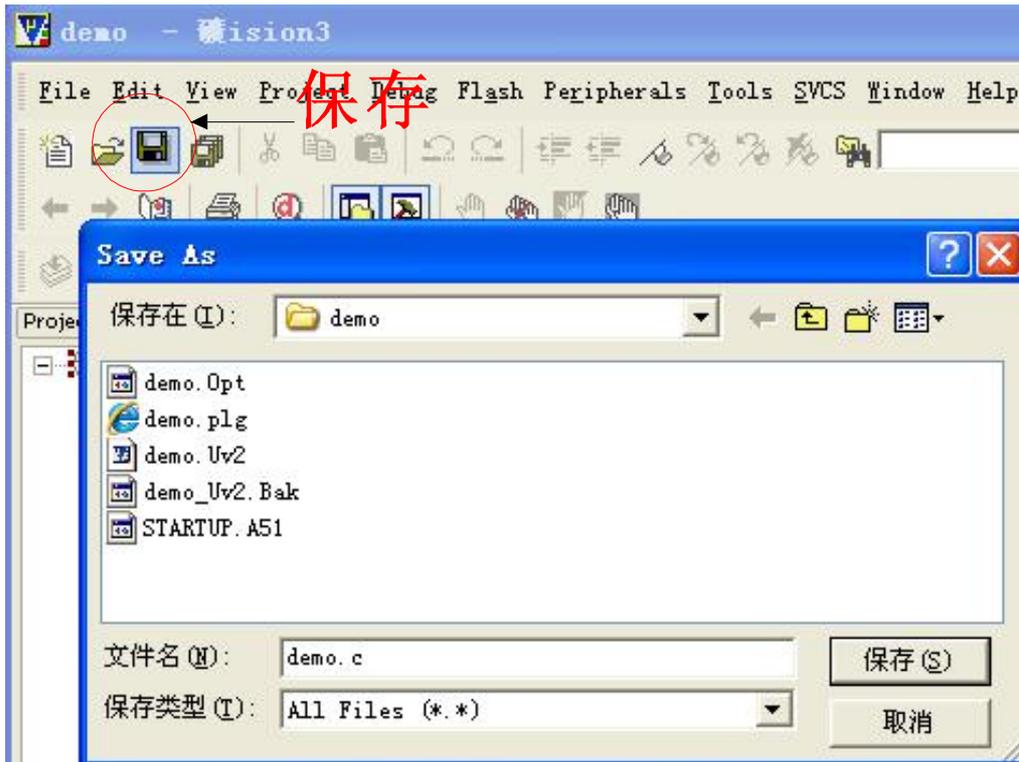
点击"确认"



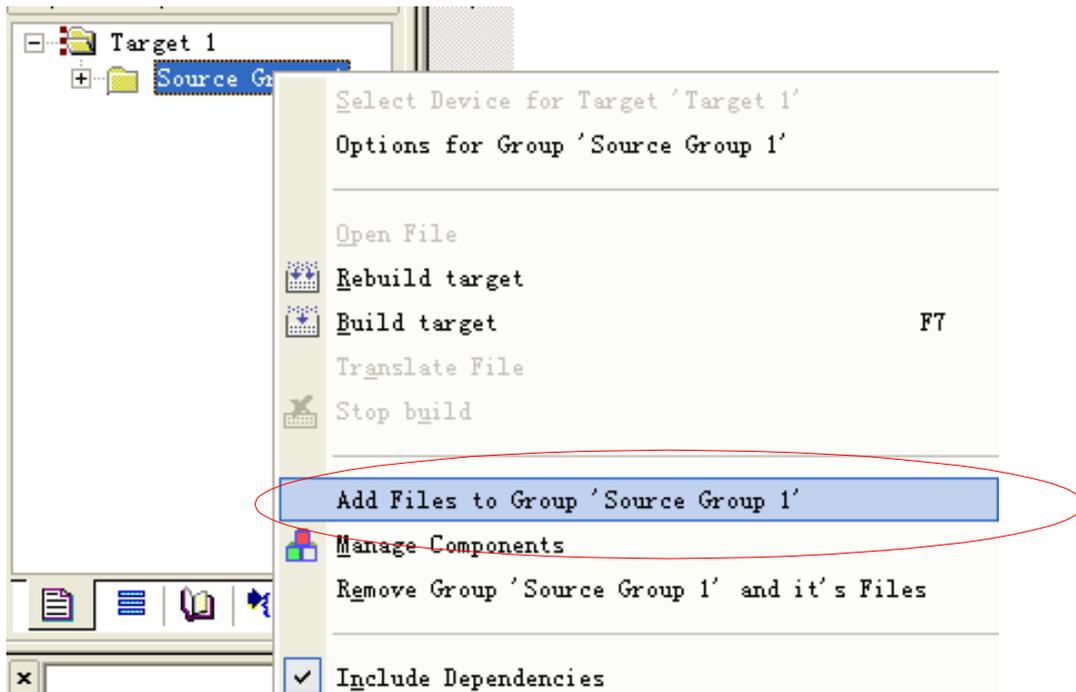
点"新建"图标,新建一个.C或.asm(汇编)的文件



点击"保存"图标,保存为.c和.asm(汇编)的文件



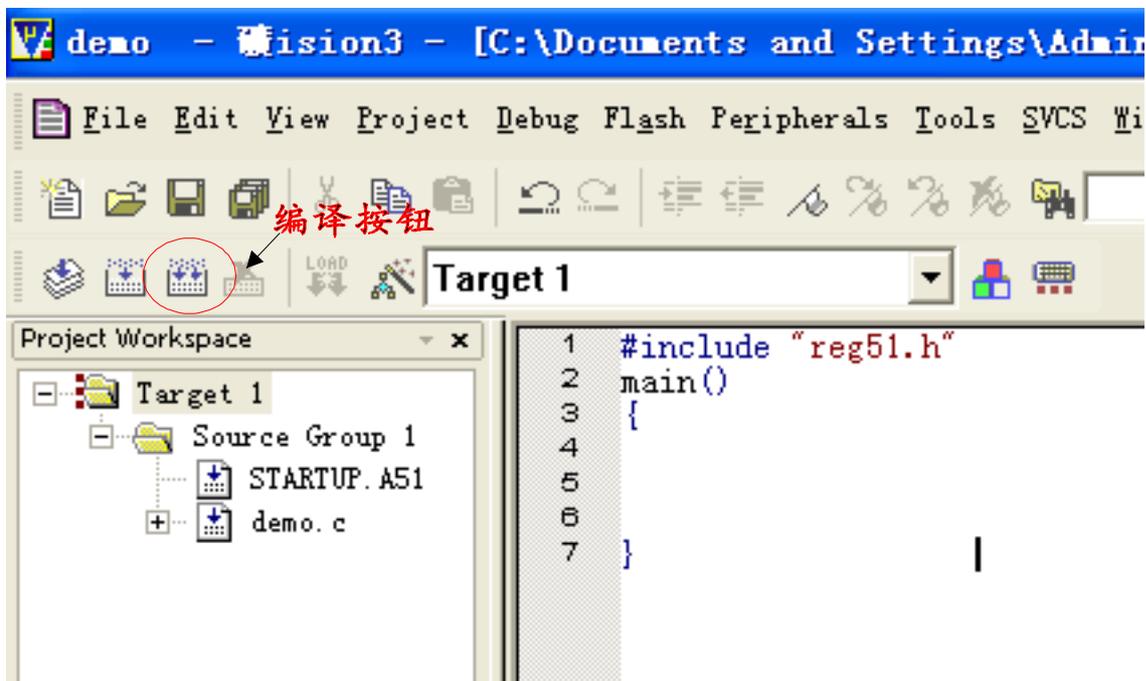
将刚才新建的demo.c文件添加到工程中



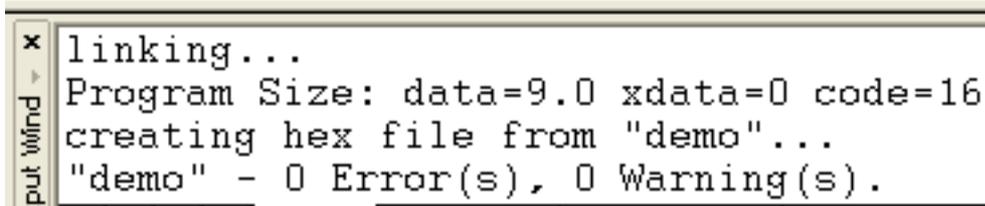
在工程工作区即可看到刚才添加的文件



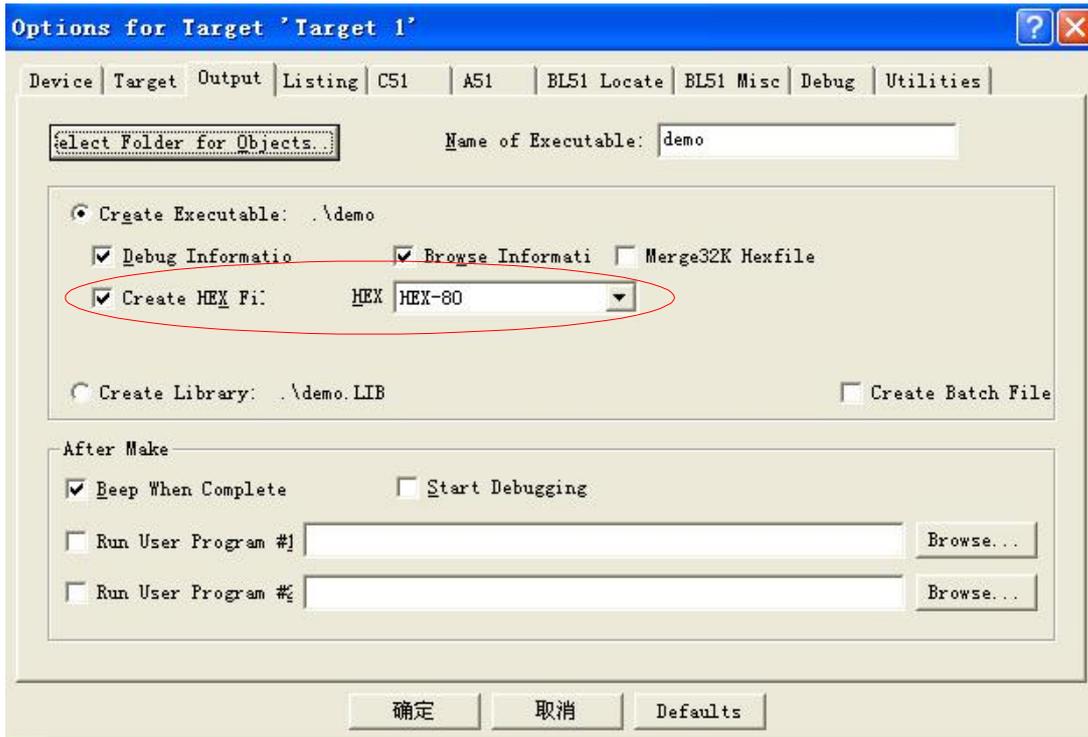
这时可以在刚才新建的.C文件中,编写自己的程序,写完后点击"编译"。完成了工程建立和代码生成



下面是生成代码的信息,表示编译通过,没有语法错误,代码生成成功

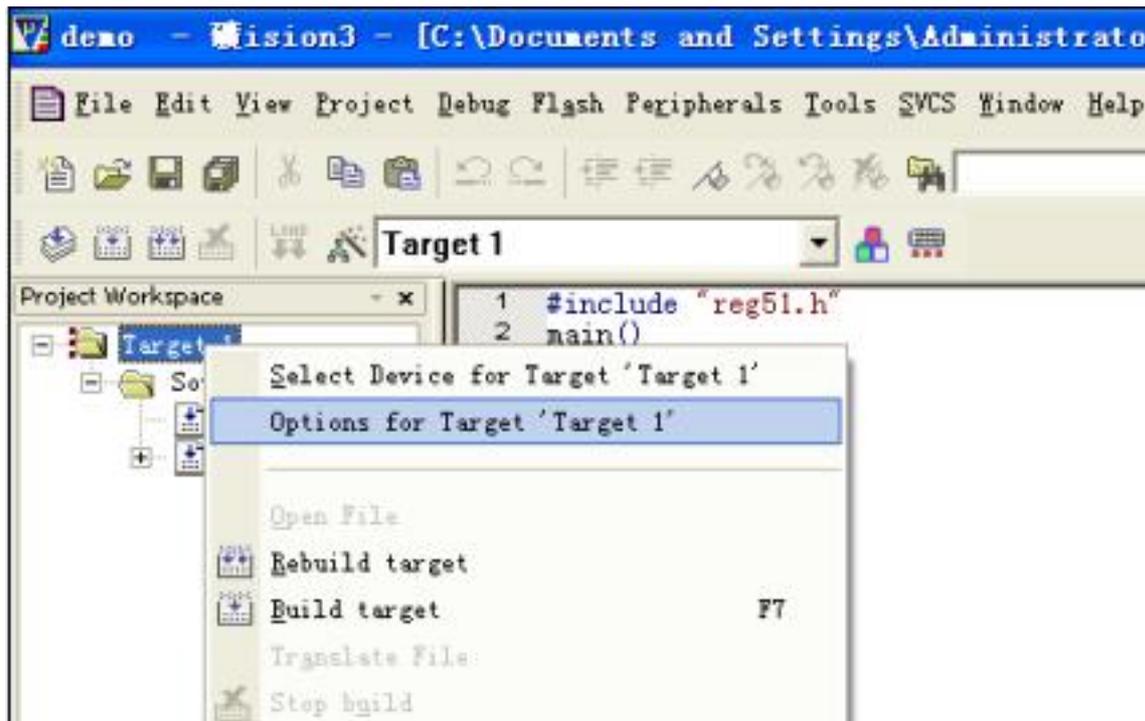


新建的工程,默认是不能生成HEX文件,必须要选中下面的选项.

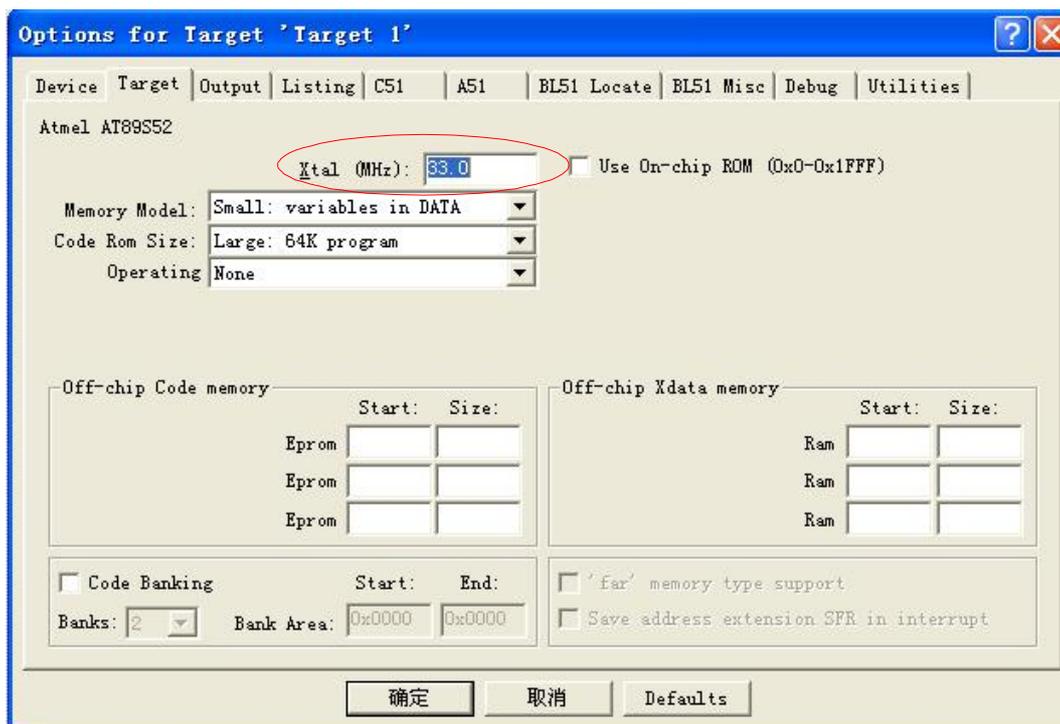


2.8 Keil工程设置

工程建好后,要对工程进一步设置,才能满足一些特定需要.



Target选项卡



Xtal: Xtal后面的数值是晶振频率值, 该数值也编译产生的目标代码无关,

仅用于软件模拟调试时, 显示的程序执行时间。

Memory Model: 用于设置RAM的使用情况, 有以下3个选项:

Small 所有变量都在单片机内部RAM中。

Compact 可以使用一页外部扩展RAM。

Large 可以使用全部外部扩展RAM。

Code Rom Size 用于设置ROM空间的使用, 有以下3个选项: :

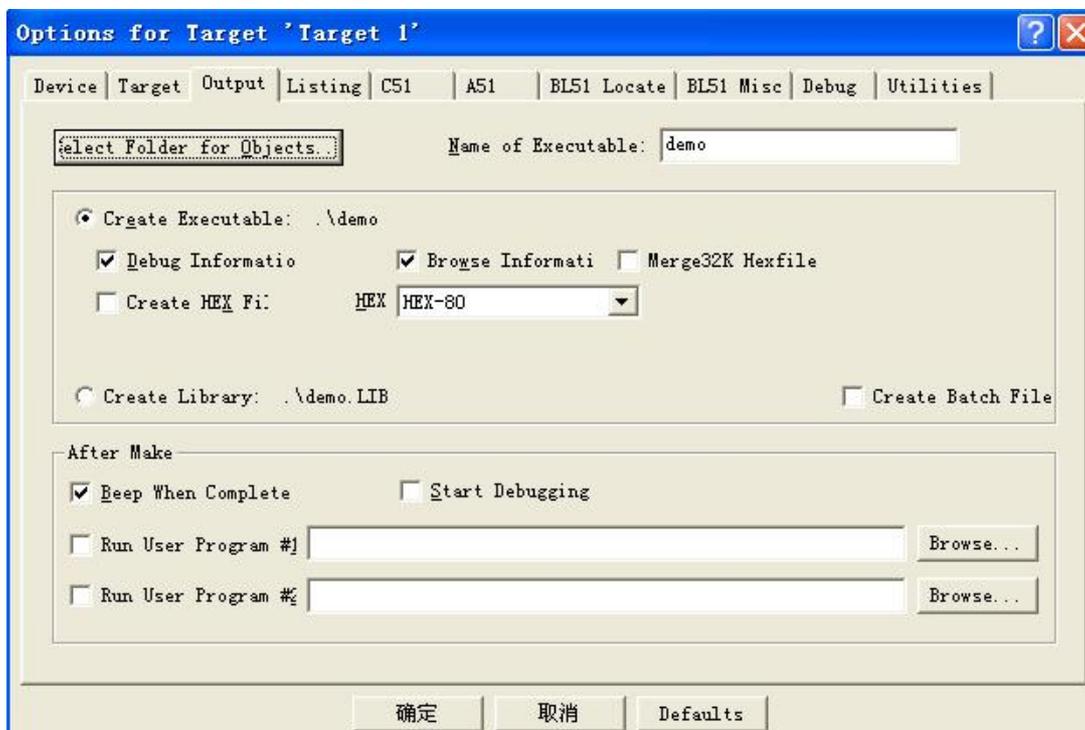
Small 只用低于2K的程序空间

Compact 单个函数代码量不能超过2KB, 整个程序可以使用64KB程序空间

Large 可以全部使用64KB空间

Operating: 该下拉框用于操作系统选择。KEIL 提供了RTX tiny和RTX full两个操作系统。如果不使用操作系统 选择默认值NONE

Output选项卡



1. Creat Hex File: 该复选框用于生成可执行代码文件, 此文件就是烧录到单片机中的程序文件, 先前建立的KEIL 工程如果没有选择这个选项, 将不能生成目标文件。
2. Debug Information: 用于产生调试信息。如果需要对程序调试, 需要选中此项。
3. Browse Information: 用于产生浏览信息。该信息可以用菜单 View → Browse 来查看。
- 4, Select Folder for Object: 改按钮用来选择最终生成的目标文件所在的文件夹
默认情况下与工程文件在同一个文件夹。
5. Name of Executable: 用于指定最终生成的目标文件的名称, 默认情况下与工程名同名。
6. Creat Library: 用于将目标文件生成库文件。

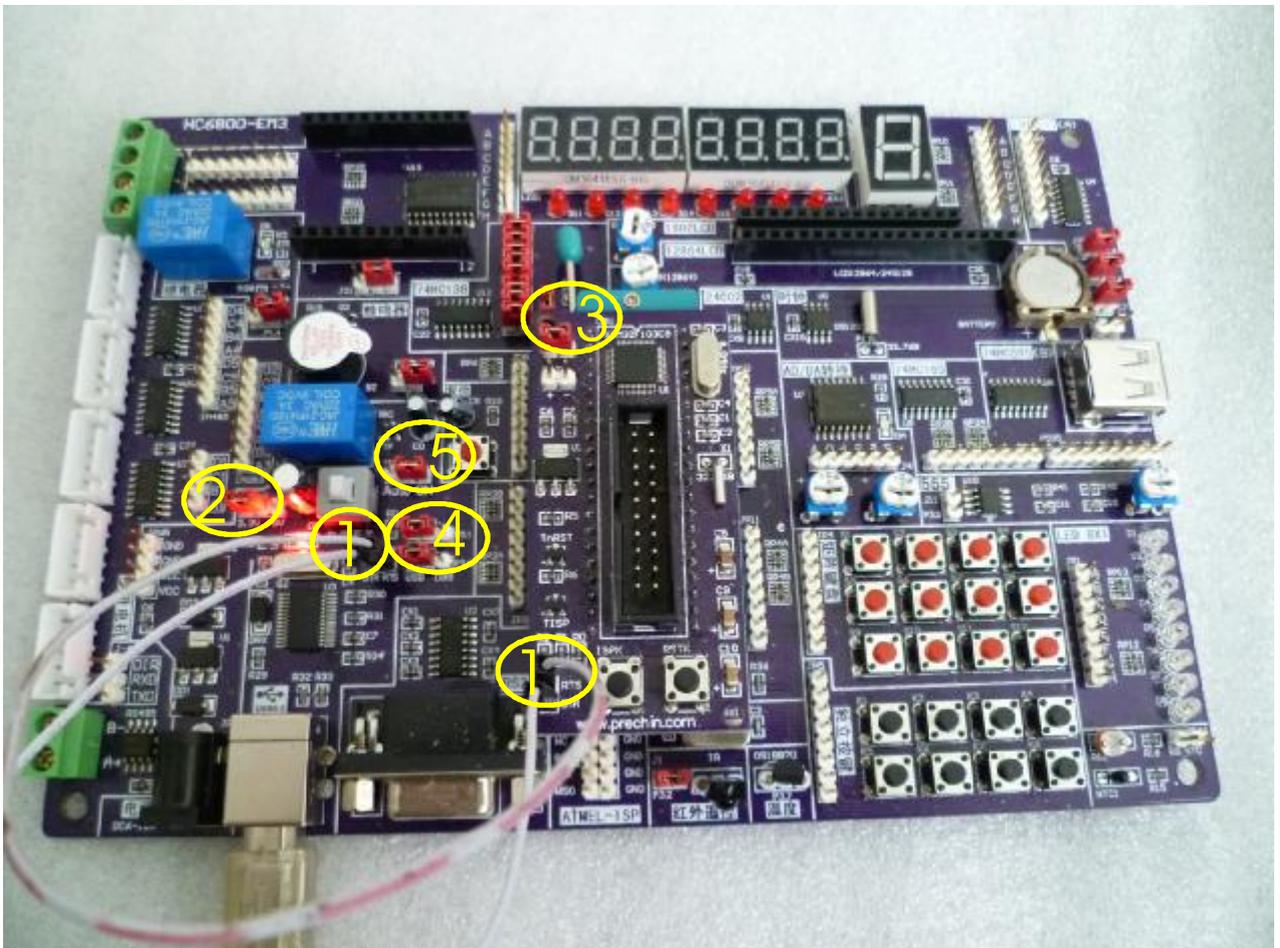
2.8 给STM32 ARM核心板烧写程序

第一步：先把硬件条件配置好

1. 用杜邦线把核心板的RTS、DTR 与底板的RTS、DTR对应连接起来。
2. 把底板的电压跳成3.3v 供电。
3. 把BOOT1用跳线帽短接起来
4. 当使用 板载的USB转串口时，短路帽应跳在左端（USB）

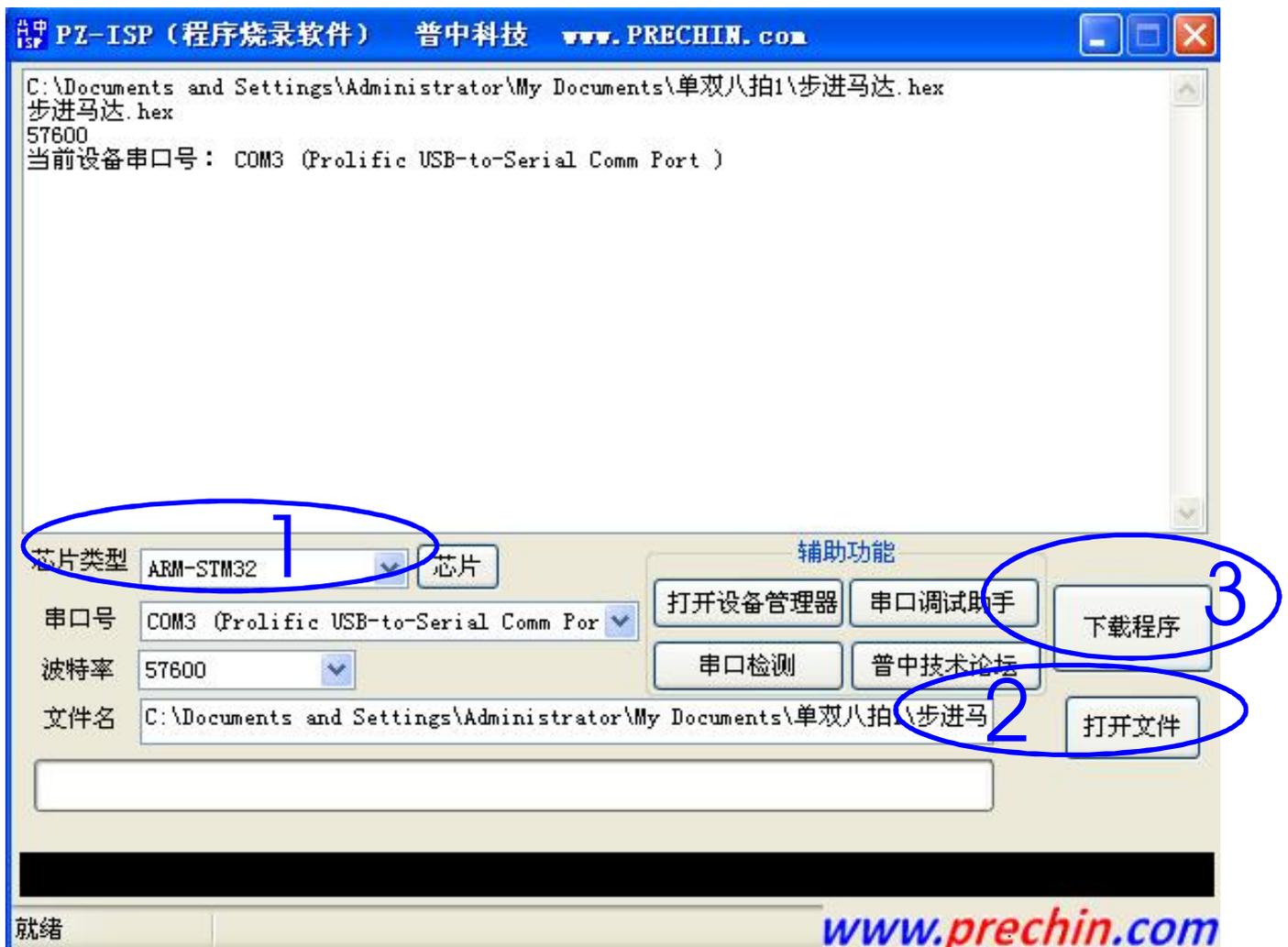
当使用9针串口是，短路帽应跳在右端（DB9）

5. J0的短路帽跳到0N端，千万不要跳到Auto端，否则影响ARM下载



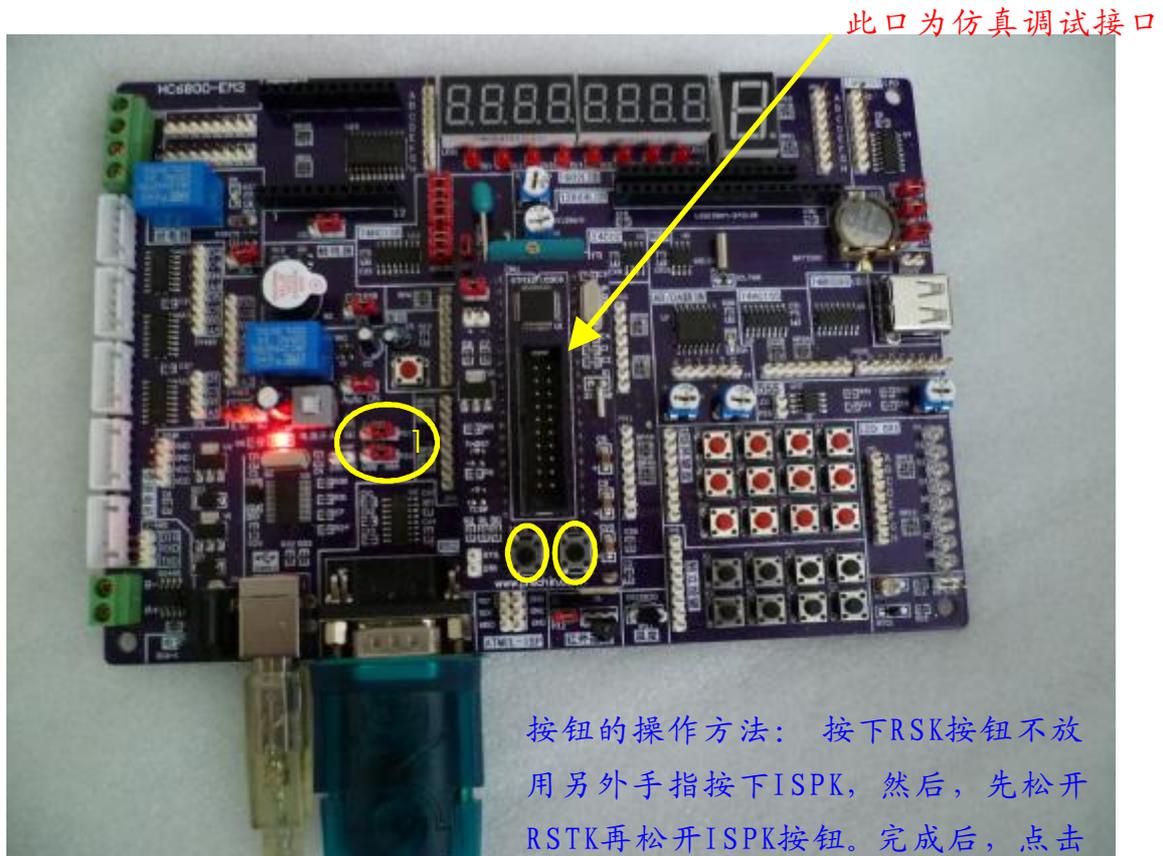
第一步：再把软件设置好

1. 芯片类型选择 ARM-STM32
2. 在选中要下载的STM32的HEX文件
3. 点击下载即可



使用标准9针串口下载程序时应注意的操作

1. 将J-TXD和J-RXD短路帽跳到右边
2. 按ISP按钮和RST按钮

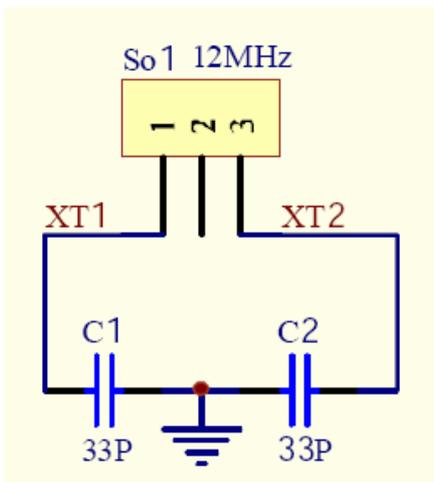
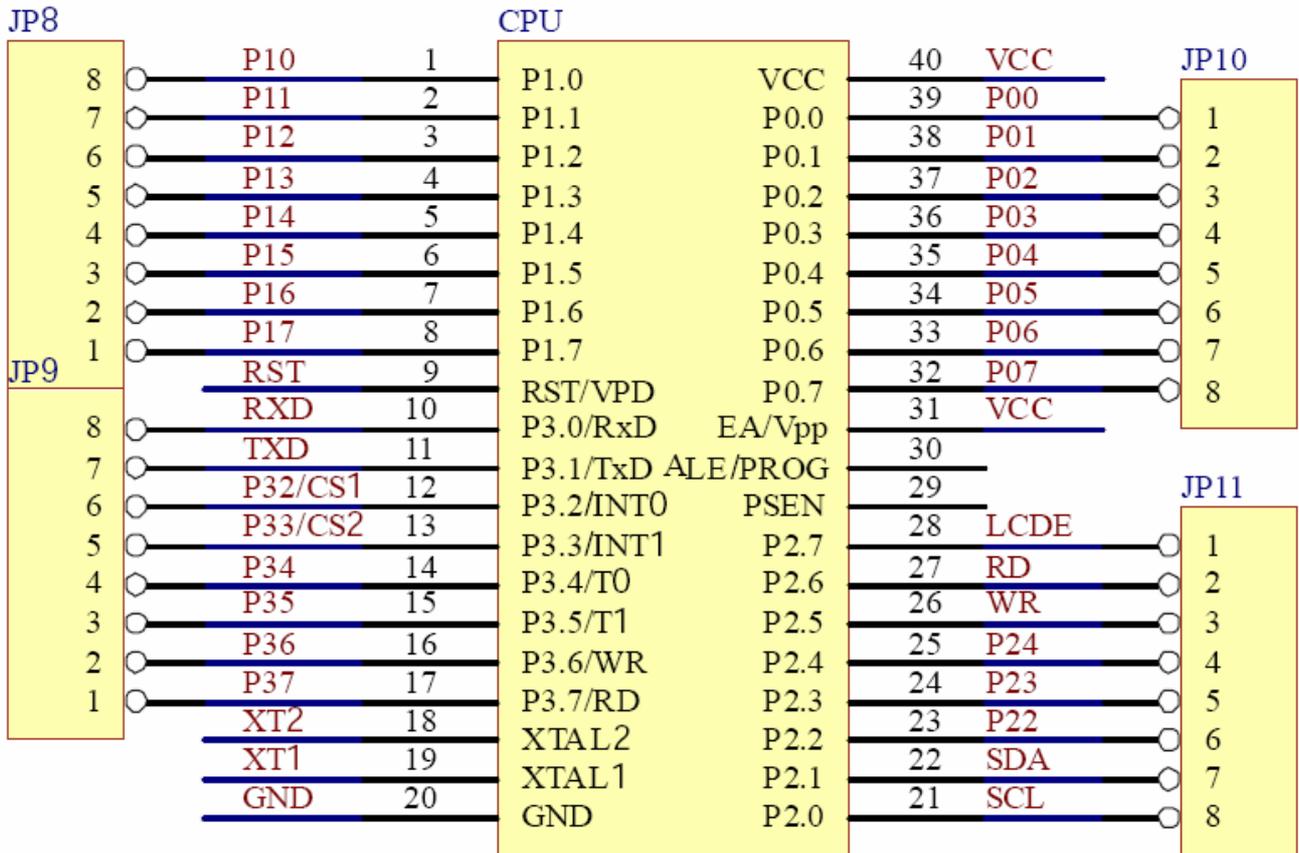


按钮的操作方法：按下RSK按钮不用另外手指按下ISPK，然后，先松开RSTK再松开ISPK按钮。完成后，点击下载软件的"下载"按钮即可完成程序下载

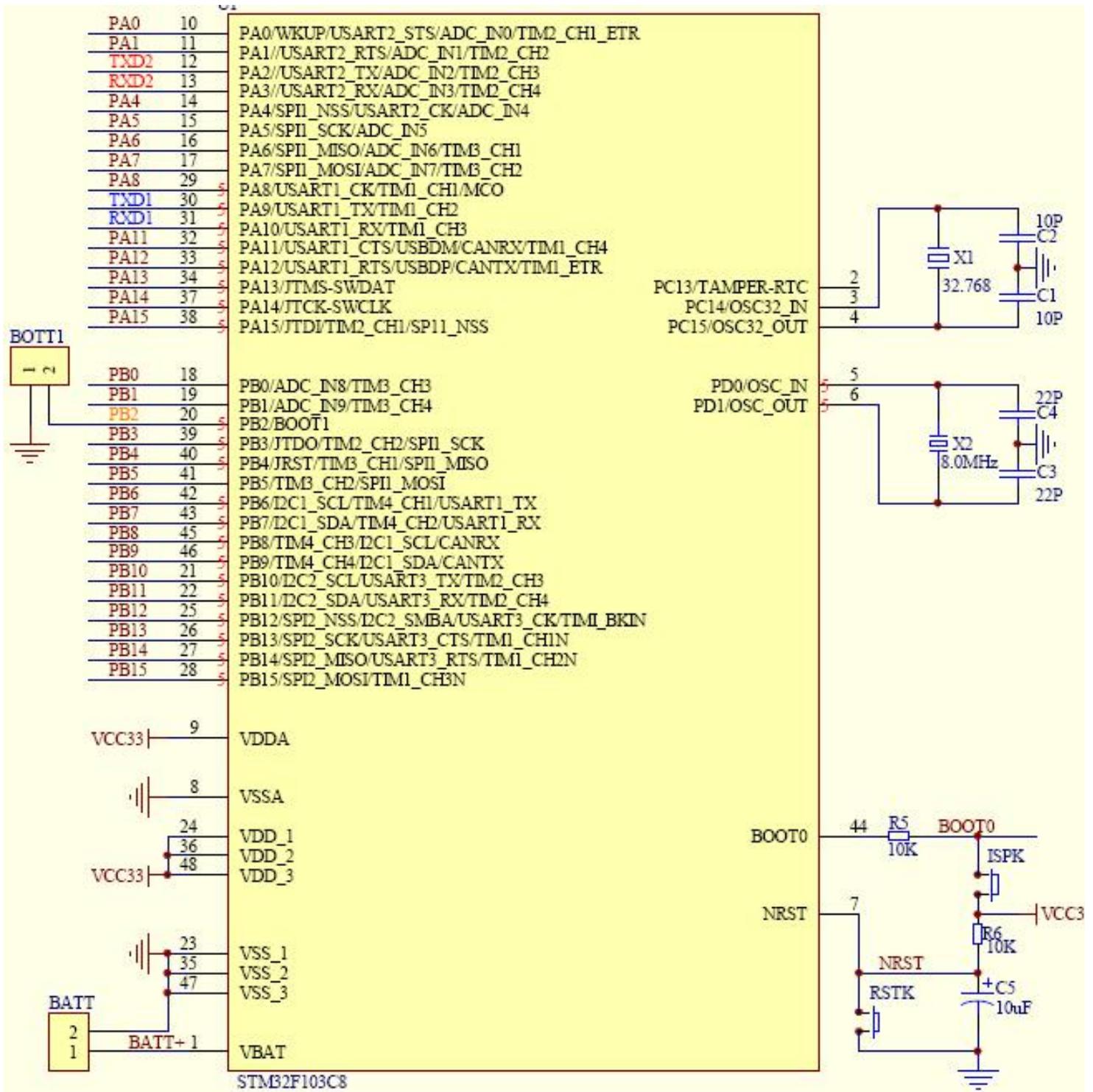
第五章 硬件原理分析

51单片机最小系统

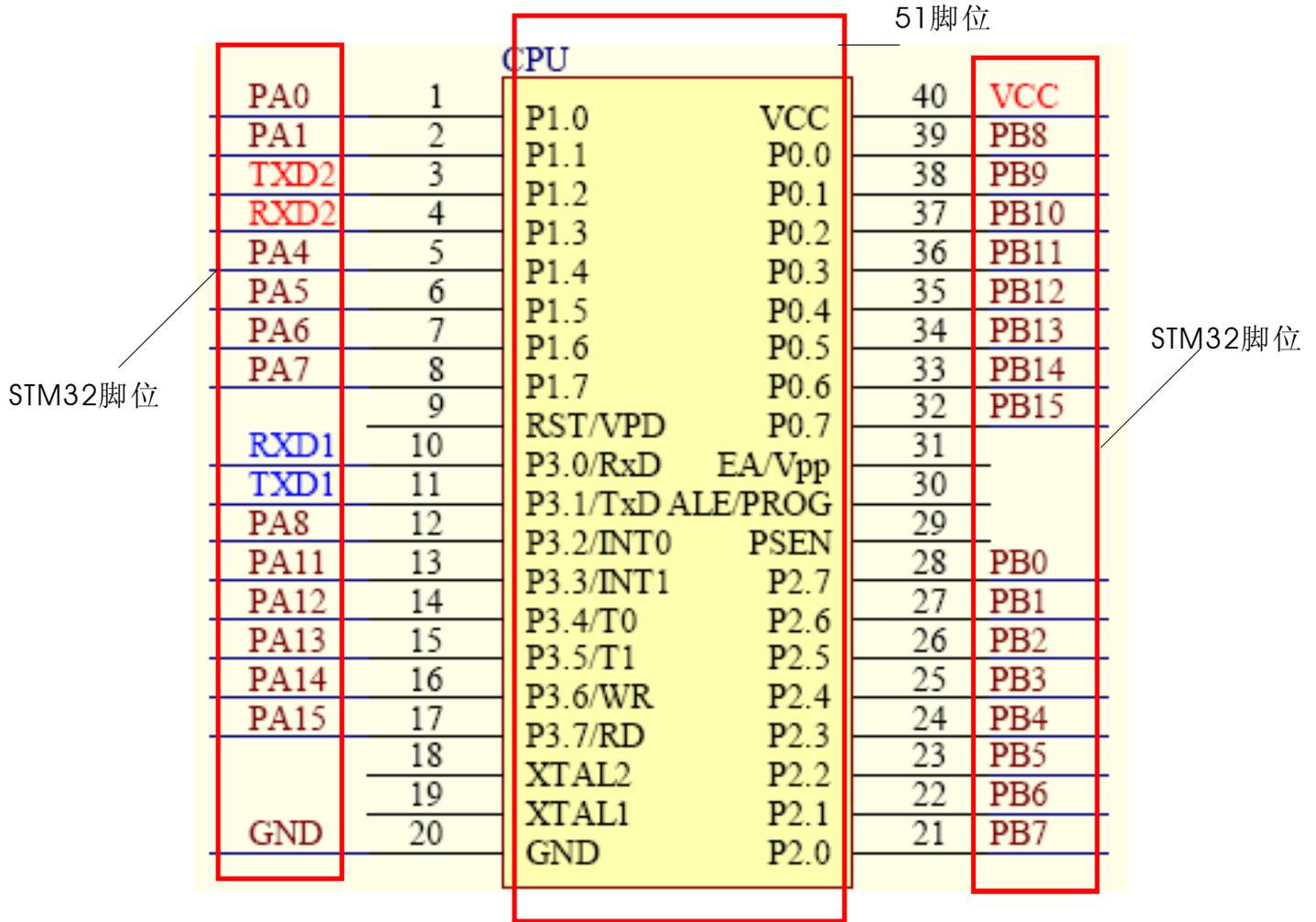
CPU为STC89系列增强型8位单片机，频率高达80MHz，可工作于6Clock，32 I/O，3定时器，内置WDT、EEPROM。支持ISP，ESD。晶振采用12M/11.0592M（可更换）。



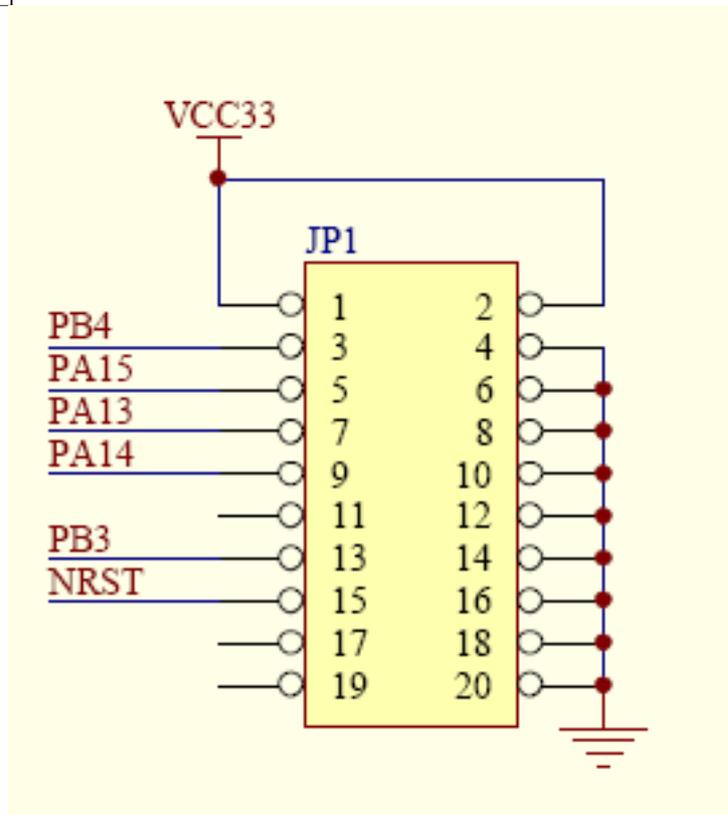
STM32最小系统核心板，原理图



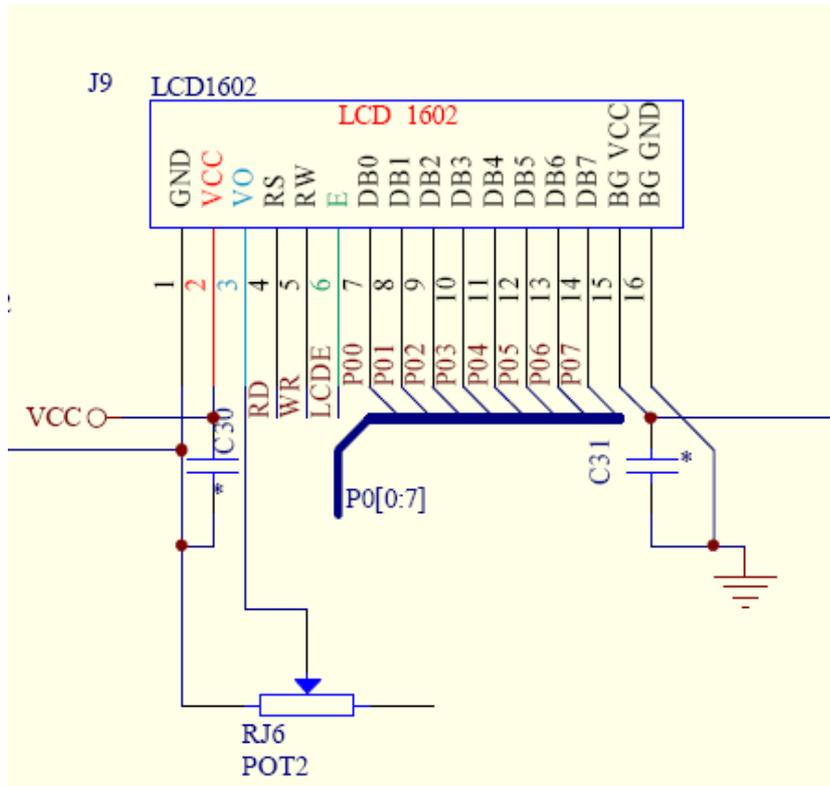
STM32与51管脚兼容原理图，



STM32在线调试接口



LCD显示



LCD (1602) 原理图:

LCD (1602) 器件说明:

1602采用标准的16脚接口,其中:

第1脚: VSS为地电源

第2脚: VCC接5V正电源

第3脚: VO为液晶显示器对比度调整端,接正电源时对比度最弱,接地电源时对比度最高,对比度过高时会产生重影,使用时可以通过一个1K的电位器调整对比度。

第4脚: RS为寄存器选择,高电平时选择数据寄存器、低电平时选择指令寄存器。

第5脚: RW为读写信号线,高电平时进行读操作,低电平时进行写操作。当RS和RW共同为低电平时可以写入指令或者显示地址,当RS为低电平RW为高电平时可以读忙信号,当RS为高电平RW为低电平时可以写入数据。

第6脚: E端为使能端,当E端由高电平跳变成低电平时,液晶模块执行命令。

第7~14脚: D0~D7为8位双向数据线。

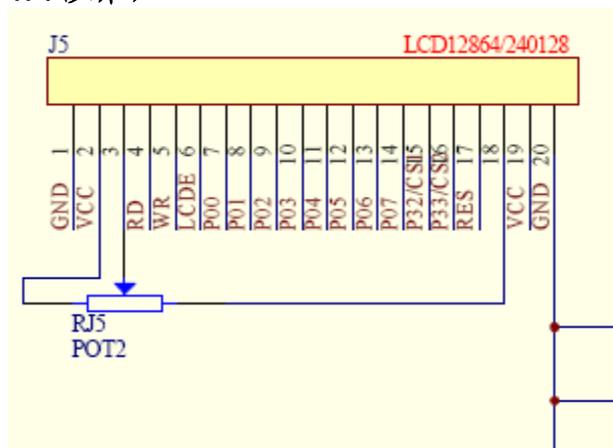
第15~16脚：背光灯电源。

1602液晶模块内部的字符发生存储器 (CGROM)已经存储了160个不同的点阵字符图形,如表1所示,这些字符有: 阿拉伯数字、英文字母的大小写、常用的符号、和日文假名等,每一个字符都有一个固定的代码,比如大写的英文字母“A”的代码是01000001B (41H),显示时模块把地址41H中的点阵字符图形显示出来,我们就能看到字母“A”

器件应用例程:

在HC6800E-V2.8开发上, 1602数据线与单片机P0相连, 具体例子看光盘例程。(学习光盘\单片机例程\LCD)。

LCD (12864/240128/TFT彩屏)



LCD (12864/240128/TFT彩屏) 原理图

一. 12864LCD

1. 12864LCD概述

带中文字库的128X64是一种具有4位/8位并行、2线或3线串行多种接口方式, 内部含有国标一级、二级简体中文字库的点阵图形液晶显示模块; 其显示分辨率为128×64, 内置8192个16×16点汉字, 和128个16×8点ASCII字符集. 利用该模块灵活的接口方式和简单、方便的操作指令, 可构成全中文人机交互图形界面. 可以显示8×4行16×16点阵的汉字. 也可完成图形显示. 低电压低功耗是其又一显著特点. 由该模块构成的液晶显示方案与同类型的图形点阵液晶显示模块相比, 不论硬件电路结构或显示程序都要简洁得多, 且该模块的价格也略低于相同点阵的图形液晶模块。

2. 基本特性:

- (1) 低电源电压 (VDD:+3.0--+5.5V)
- (2) 显示分辨率: 128×64点
- (3) 内置汉字字库, 提供8192个16×16点阵汉字(简繁体可选)
- (4) 内置 128个16×8点阵字符
- (5) 2MHZ时钟频率

- (6)、显示方式：STN、半透、正显
- (7) 驱动方式：1/32DUTY, 1/5BIAS
- (8)、视角方向：6点
- (9) 背光方式：侧部高亮白色LED, 功耗仅为普通LED的1/5—1/10
- (10)、通讯方式：串行、并口可选
- (11) 内置DC-DC转换电路, 无需外加负压
- (12) 无需片选信号, 简化软件设计
- (13) 工作温度:0℃ - +55℃ ,存储温度: -20℃ - +60℃

3、图形显示

先设垂直地址再设水平地址(连续写入两个字节的资料来完成垂直与水平的坐标地址)

垂直地址范围 AC5...AC0

水平地址范围 AC3...AC0

绘图RAM的地址计数器(AC)只会对水平地址(X轴)自动加一,当水平地址=0FH时会重新设为00H但并不会对垂直地址做进位自动加一,故当连续写入多笔资料时,程序需自行判断垂直地址是否需重新设定。GDRAM的坐标地址与资料排列顺序如下图:

4、应用说明

用带中文字库的128X64显示模块时应注意以下几点:

- ①欲在某一个位置显示中文字符时,应先设定显示字符位置,即先设定显示地址,再写入中文字符编码。
- ②显示ASCII字符过程与显示中文字符过程相同。不过在显示连续字符时,只须设定一次显示地址,由模块自动对地址加1指向下一个字符位置,否则,显示的字符中将会有空ASCII字符位置。
- ③当字符编码为2字节时,应先写入高位字节,再写入低位字节。
- ④模块在接收指令前,向处理器必须先确认模块内部处于非忙状态,即读取BF标志时BF需为“0”,方可接受新的指令。如果在送出一个指令前不检查BF标志,则在前一个指令和这个指令中间必须延迟一段较长的时间,即等待前一个指令确定执行完成。指令执行的时间请参考指令表中的指令执行时间说明。
- ⑤“RE”为基本指令集与扩充指令集的选择控制位。当变更“RE”后,以后的指令集将维持在最后的状态,除非再次变更“RE”位,否则使用相同指令集时,无需每次均重设“Re”位。

引脚号	标识	说明
PIN1	GND	接0V
PIN2	VCC	接4.8V-5V
PIN3	V0	VCC和VEE接可调电阻,中间抽头接至V0
PIN4	RS CS	并行模式:RS=0,指令寄存器;RS=1,数据寄存器。串行模式:片选

PIN5	R/W SID	并行模式: R/W=0, 写; R/W=1, 读。串行模式: 数据
PIN6	SCK	并行模式: 允许信号。串行模式: 脉冲
PIN7	D0	并行模式: 数据0; 串行模式: 不连接
PIN8	D1并行模式:	数据1; 串行模式: 不连接
PIN9	D2并行模式:	数据2; 串行模式: 不连接
PIN10	D3并行模式:	数据3; 串行模式: 不连接
PIN11	D4并行模式:	数据4; 串行模式: 不连接
PIN12	D5并行模式:	数据5; 串行模式: 不连接
PIN13	D6并行模式:	数据6; 串行模式: 不连接
PIN14	D7并行模式:	数据7; 串行模式: 不连接
PIN15	PSB并行模式:	PSB=1; 串行模式: PSB=0
PIN16	NC	不需连接
PIN17	RST	复位
PIN18	Nc	不需连接
PIN19	LED+	背光正极, 接4.8V - 5V
PIN20	LED-	背光负极, 接0V

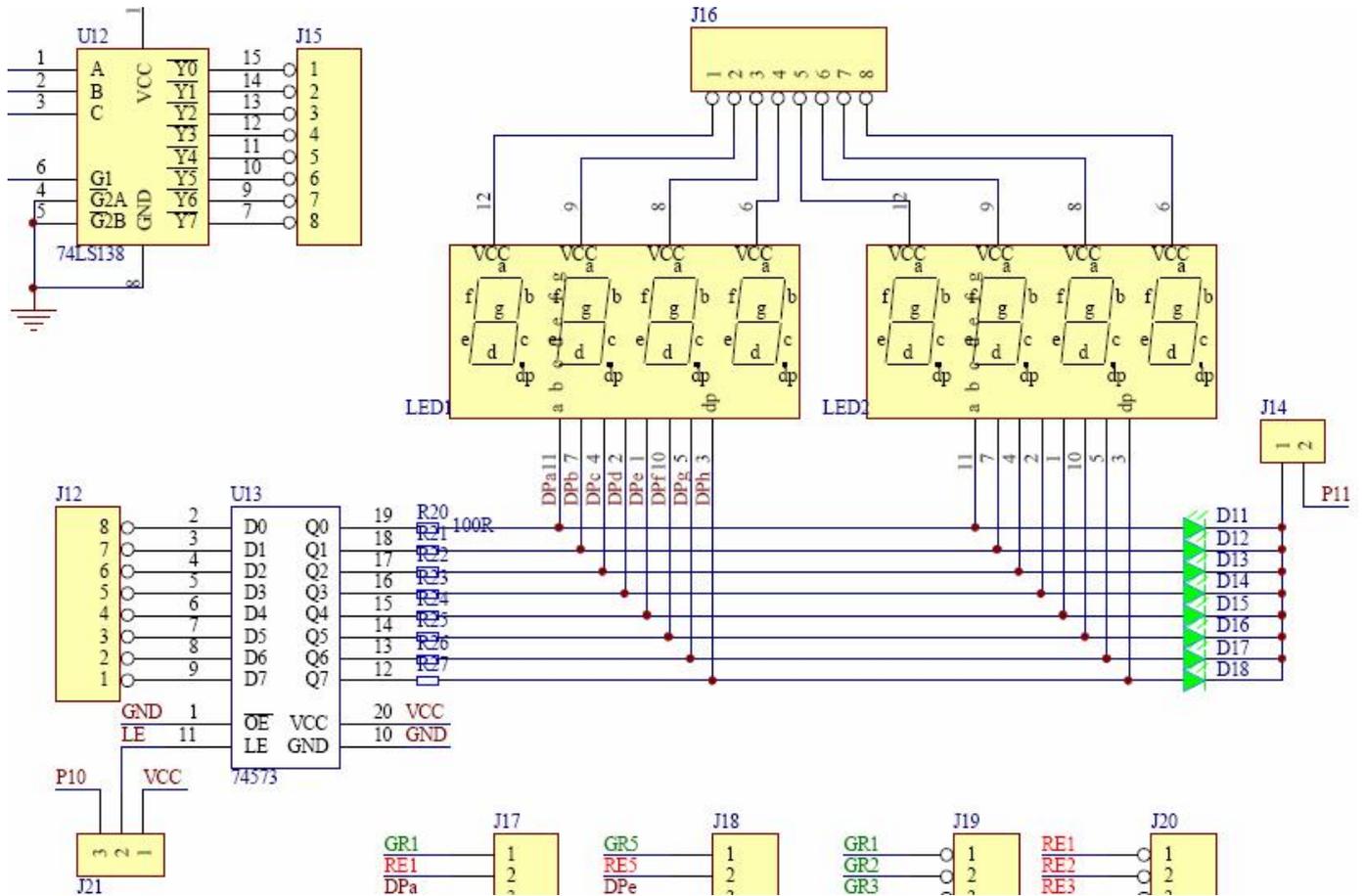
二. TFT彩屏

动态数码管和静态数码管

有2个四位动态数码管和1个一位静态数码管。

当使用四位动态数码管时，请用8P排线将J12与单片机的I/O脚(P0-P3任选)相连，

当使用一位静态数码管时，有两种连接方式：1.用8P排线将JP3与单片机的I/O脚(P0-P3任选)相连，实现用单片机脚直接控制数码管。2.用8P排线将JP2与JP3相连，然后将JP12用短路帽全部短接，此时为单片机控制74HC595，74HC595再控制数码管的动态扫描。



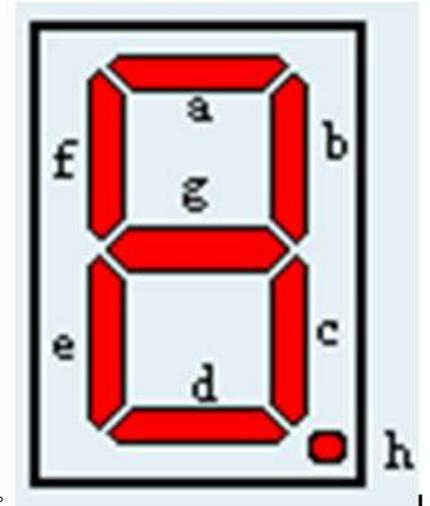
数码管原理图

数码管原理：

数码管实际上是由7个发光管组成8字形构成的，加上小数点就是8个。动态扫描显示接口是单片机中应用最为广泛的一种显示方式之一。其接口电路是把所有显示器的8个笔划段a-h同名端连在一起，而每一个显示器的公共极COM是各自独立地受I/O线控制。CPU向字段输出口送出字形码时所有显示器接收到相同的字形码，但究竟是那个显示器亮，则取决于COM端，所以我们可以自行决定何时显示哪一位了。而所谓动态扫描就是指我们采用分时的方法，轮流控制各个显示器的COM端，使各个显示器轮流点亮。每位显示器的点亮时间是极为短暂的（约1ms），但由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位显示器并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感。如何让一个数码管显示呢，

如图：如果要显示一个3字，首先使得COM为低电平使能数码管，然后可以控制abgcd高电平将会亮；而fe为低电平不亮。

如图：



J21 短路帽可以接VCC和P10 如果接P1.0需要用程序控制使能，接VCC

直通不需要程序控制使用。

J16 数码管位选接口，可以用138译码器进行位选，也可以又单片机脚直接控制实现位选。

J15 138译码器输出接口

J14 动/静LED功能，如短接，改变P11的电平可以实现动/静LED功能。

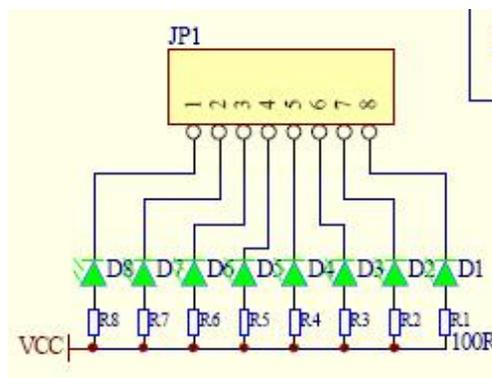
数码管应用例程：

在HC6800E-V2.8开发板上有8个数码管。数码管由74HC595和74LS138控制。74HC595控制数码管的8个笔划段，由74LS138控制那位有效，也可以单片机脚直接控制那位有效。具体例子看光盘例程。(学习光盘\单片机例程\数码管)

LED灯

JP1为8路LED灯的接口，使用此功能时，请将JP1与JP8JP11中任何相连，即可实现单片机控制8路LED。当JP1接口为低电平时，LED灯将被点亮。

LED灯原理图



LED灯应用例程

具体例子看光盘例程。(学习光盘\单片机例程\LED)

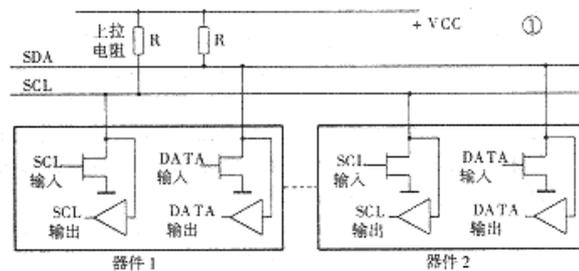
独立按键

Jp5为独立按键的接口，可以与单片机任何脚相连，组成单片机的输入设备，也可做外部中断输入，也可以做74HC165(并入串出锁存器)的输入设备。

I2C总线：

I2C总线是一种用于IC器件之间连接的二线制总线。它通过SDA（串行数据线）及SCL（串行时钟线）两根线在连到总线上的器件之间传送信息，并根据地址识别每个器件：不管是单片机、存储器、LCD驱动器还是键盘接口。

1. I2C总线的基本结构 采用I2C总线标准的单片机或IC器件，其内部不仅有I2C接口电路，而且将内部各单元电路按功能划分为若干相对独立的模块，通过软件寻址实现片选，减少了器件片选线的连接。CPU不仅能通过指令将某个功能单元电路挂靠或摘离总线，还可对该单元的工作状况进行检测，从而实现对硬件系统的既简单又灵活的扩展与控制。I2C总线接口电路结构如图所示。



2. 双向传输的接口特性 传统的单片机串行接口的发送和接收一般都各用一条线，如MCS51系列的TXD和RXD，而I2C总线则根据器件的功能通过软件程序使其可工作于发送或接收方式。当某个器件向总线上发送信息时，它就是发送器(也叫主器件)，而当其从总线上接收信息时，又成为接收器(也叫从器件)。主器件用于启动总线上传送数据并产生时钟以开放传送的器件，此时任何被寻址的器件均被认为是从器件。I2C总线的控制完全由挂载在总线上的主器件送出的地址和数据决定。在总线上，既没有中心机，也没有优先机。

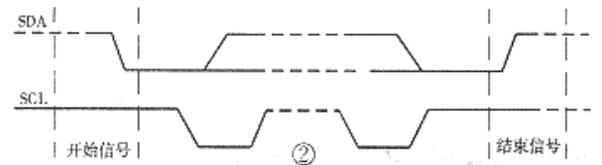
总线上主和从(即发送和接收)的关系不是一成不变的，而是取决于此时数据传送的方向。SDA和SCL均为双向I/O线，通过上拉电阻接正电源。当总线空闲时，两根线都是高电平。连接总线的器件的输出级必须是集电极或漏极开路，以具有线“与”功能。I2C总线的数据传送速率在标准工作方式下为100kbit/s，在快速方式下，最高传送速率可达400kbit/s。

3. I2C总线上的时钟信号 在I2C总线上传送信息时的时钟同步信号是由挂载在SCL时钟线上的所有器件的逻辑“与”完成的。SCL线上由高电平到低电平的跳变将影响到这些器件，一旦某

个器件的时钟信号下跳为低电平，将使SCL线一直保持低电平，使SCL线上的所有器件开始低电平时。此时，低电平周期短的器件的时钟由低至高的跳变并不能影响SCL线的状态，于是这些器件将进入高电平等待的状态。

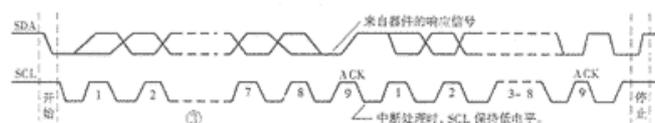
当所有器件的时钟信号都上跳为高电平时，低电平期结束，SCL线被释放返回高电平，即所有的器件都同时开始它们的高电平期。其后，第一个结束高电平期的器件又将SCL线拉成低电平。这样就在SCL线上产生一个同步时钟。可见，时钟低电平时间由时钟低电平期最长的器件确定，而时钟高电平时间由时钟高电平期最短的器件确定。

4. 数据的传送 在数据传送过程中，必须确认数据传送的开始和结束。在I2C总线技术规范中，开始和结束信号（也称启动和停止信号）的定义如图2所示。当时钟线SCL为高电平时，数据线SDA由高电平跳变为低电平定义为“开始”信号；当SCL线为高电平时，SDA线发生低电平到高电平的跳变为“结束”信号。开始和结束信号都是由主器件产生。在开始信号以后，总线即被认为处于忙状；在结束信号以后的一段时间内，总线被认为是空闲的。



I2C总线的数据传送格式是：在I2C总线开始信号后，送出的第一个字节数据是用来选择从器件地址的，其中前7位为地址码，第8位为方向位(R/W)。方向位为“0”表示发送，即主器件把信息写到所选择的从器件；方向位为“1”表示主器件将从从器件读信息。开始信号后，系统中的各个器件将自己的地址和主器件送到总线上的地址进行比较，如果与主器件发送到总线上的地址一致，则该器件即为被主器件寻址的器件，其接收信息还是发送信息则由第8位(R/W)确定。

在I2C总线上每次传送的数据字节数不限，但每一个字节必须为8位，而且每个传送的字节后面必须跟一个认可位（第9位），也叫应答位（ACK）。数据的传送过程如图3所示。每次都是先传最高位，通常从器件在接收到每个字节后都会作出响应，即释放SCL线返回高电平，准备接收下一个数据字节，主器件可继续传送。如果从器件正在处理一个实时事件而不能接收数据时，（例如正在处理一个内部中断，在这个中断处理完之前就不能接收I2C总线上的数据字节）可以使时钟SCL线保持低电平，从器件必须使SDA保持高电平，此时主器件产生1个结束信号，使传送异常结束，迫使主器件处于等待状态。当从器件处理完毕时将释放SCL线，主器件继续传送。



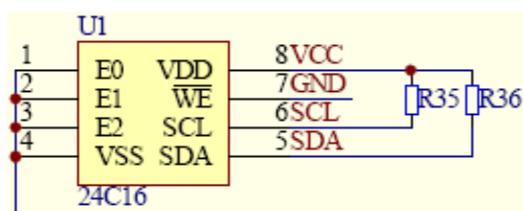
当主器件发送完一个字节的的数据后，接着发出对应于SCL线上的一个时钟（ACK）认可位，在此时钟内主器件释放SDA线，一个字节传送结束，而从器件的响应信号将SDA线拉成低电平，使SDA在该时钟的高电平期间为稳定的低电平。从器件的响应信号结束后，SDA线返回高电平，进入下一个传送周期。

I2C总线还具有广播呼叫地址用于寻址总线上所有器件的功能。若一个器件不需要广播呼叫寻址中所提供的任何数据，则可以忽略该地址不作响应。如果该器件需要广播呼叫寻址中提供的数据，则应对地址作出响应，其表现为一个接收器。

5. 总线竞争的仲裁 总线上可能挂接有多个器件，有时会发生两个或多个主器件同时想占用总线的情况。例如，多单片机系统中，可能在某一时刻有两个单片机要同时向总线发送数据，这种情况叫做总线竞争。I2C总线具有多主控能力，可以对发生在SDA线上的总线竞争进行仲裁，其仲裁原则是这样的：当多个主器件同时想占用总线时，如果某个主器件发送高电平，而另一个主器件发送低电平，则发送电平与此时SDA总线电平不符的那个器件将自动关闭其输出级。总线竞争的仲裁是在两个层次上进行的。首先是地址位的比较，如果主器件寻址同一个从器件，则进入数据位的比较，从而确保了竞争仲裁的可靠性。由于是利用I2C总线上的信息进行仲裁，因此不会造成信息的丢失。

6. I2C总线接口器件 目前在视频处理、移动通信等领域采用I2C总线接口器件已经比较普遍。另外，通用的I2C总线接口器件，如带I2C总线的单片机、RAM、ROM、A/D、D/A、LCD驱动器等器件，也越来越多地应用于计算机及自动控制系统中。

E2PROM存储器



E2PROM AT24C02说明：

AT24C02是美国ATMEL公司的低功耗CMOS串行EEPROM，它是内含256×8位存储空间，具有工作电压宽（2.5~5.5V）、擦写次数多（大于10000次）、写入速度快（小于10ms）等特点。该开发板上带有一片AT24C02芯片可以提供试验。

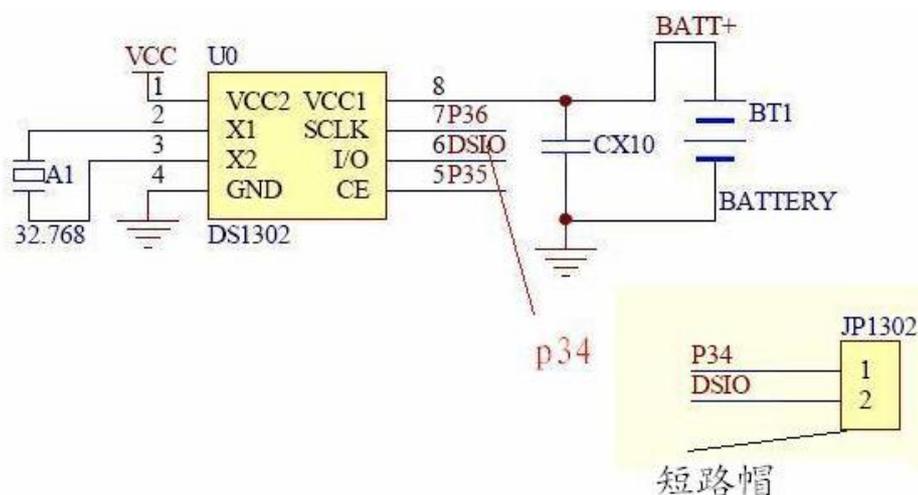
电路图中AT24C02的1、2、3脚是三条地址线，用于确定芯片的硬件地址。第8脚和第4脚分别为正、负电源。第5脚SDA为串行数据输入/输出，数据通过这条双向I2C总线串行传送。第7脚需要接地。

24C02中带有片内地址寄存器。每写入或读出一个数据字节后，该地址寄存器自动加1，以实现
对下一个存储单元的读写。所有字节均以单一操作方式读取。为降低总的写入时间，一次
操作可写入多达8个字节的数据。如果用户想详细了解可以参考光盘上的数据手册。

AT24C02应用：

在HC6800E-V2.8开发板上有I2C总线，而上面包括E2PROM、RTC(实时时钟)；现在先介绍E2PR
OM的使用。E2PROM的型号是AT24C02，E2PROM的SCL与单片机的P2.0连接，SDA与单片机的
P2.1连接。在24C02的设备地址为：0xA0。写保护引脚(WP)接地，也就是说不保护。具体例子
看光盘例程。(学习光盘\单片机例程\EEPROM读写)。

时钟芯片DS1302



Ds1302原理图

Ds1302介绍：

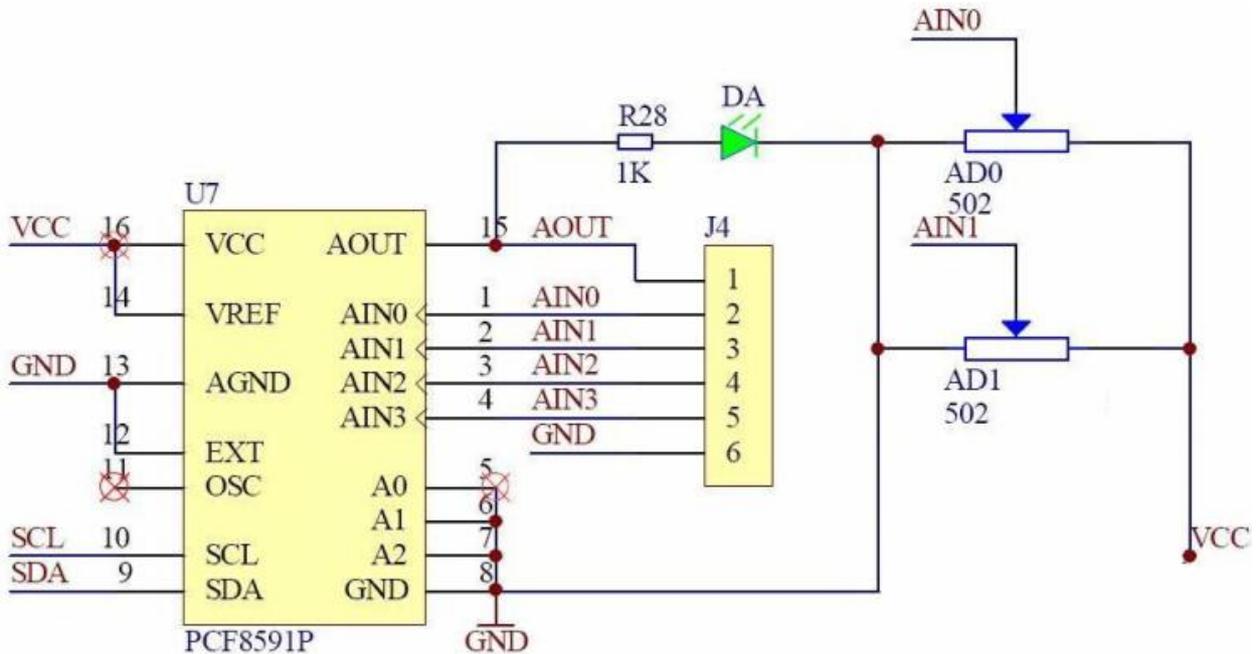
DS1302 是美国DALLAS公司推出的一种高性能、低功耗、带RAM的实时时钟电路，它可以对
年、月、日、周日、时、分、秒进行计时，具有闰年补偿功能，工作电压为2.5V~5.5V。采
用三线接口与CPU进行同步通信，并可采用突发方式一次传送多个字节的时钟信号或RAM数据。
DS1302内部有一个31×8的用于临时性存放数据的RAM寄存器。DS1302是DS1202的升级产品，
与DS1202兼容，但增加了主电源/后背电源双电源引脚，同时提供了对后背电源进行涓细电流
充电的能力。如果用户想详细了解可以参考光盘上的数据手册。

DS1302应用例程：

在HC6800E-V2.8开发板上有DS1302时钟芯片具体例子看光盘例程。(学习光盘\单片机例程\ RTC
实时时钟(DS1302))

DA/AD转换 PCF8591T:

PCF8591T使用I2C与单片机通信，P2.1(SDA)串行数据线、P2.0(SCL)串行时钟线。AD0和AD1是两路模拟输入，改变AD0和AD1位置的电位器，实现了2两路模拟输入，在数码管中可以看到数值变化。当PCF8591T数模端口数据变化时，DA位置的LED亮度随之改变。



PCF8591T 电路图

1. 组成

PCF8591是一款单电源、低功耗8位COMS型A/D、D/A转换芯片，它具有4路模拟量输入通道、一路模拟量输出通道和1个I2C总线接口。该器件I2C从地址的低三位由芯片的A0、A1和A2三个地址引脚决定，所以在不增加任何硬件的情况下同一条I2C总线最多可以连接8个同类型的器件。该器件具有多路模拟量输入、片上跟踪保持、8位A/D转换和8位D/A转换等功能。A/D与D/A的最大转换速率由I2C总线的最大传输速率决定，在单片机应用系统中得到了广泛的应用。

2. 特性

PCF8591具有以下特点：

1. 单电源供电；
2. 正常工作电源电压范围为2.5V~6V；
3. 通过I2C总线完成数据的输入/输出；
4. 器件地址由3个地址引脚决定；

5. 采样频率由I2C总线传输速率决定;
6. 4路模拟量输入可编程为单端输入或差分输入;
7. 可配置转换通道号自动增加功能;
8. 模拟电压范围为VSS~VDD;
9. 片上跟踪保持功能;
10. 8位逐次逼近A/D转换;
11. 带有一路模拟量输出的乘法D/A转换。

AIN0~AIN3: 模拟输入 (A/D转换)。

AOUT: 模拟输出 (D/A转换)。

A0-A2: 硬件设备地址。

GND: 电源负极地。

VREF: 参考电压输入。

EXT:振荡器输入时, 内部/外部的切换开关。

OSC: 振荡器输入/输出。

SCL: I2C BUS 时钟输入。

SDA: I2C BUS 数据输入/输出。

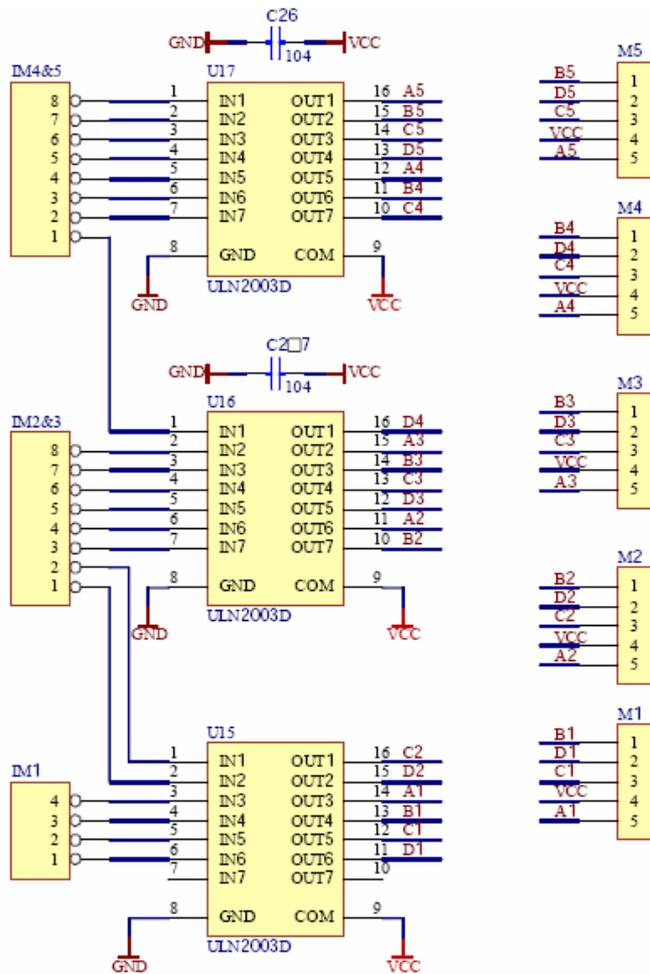
AGND: 模拟地, 模拟信号和基准电源的参考地。

PCF8591T应用:

具体例子看光盘例程。(学习光盘\单片机例程\AD-DA转换)。

步进电机:

M1-M5均为步进马达接口, 通过ULN2003D放大驱动步进马达, 采用5线四相步进马达。通过编程可以完成5个步进马达协调运行, 可以模拟5轴数控, 可以制作智能小车等。



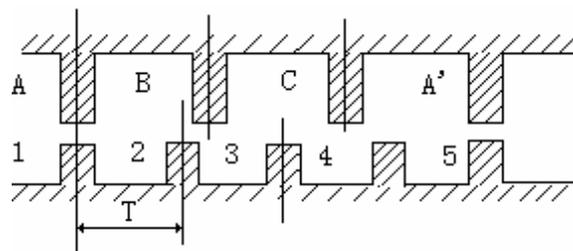
步进电机原理图

步进电机介绍:

由于反应式步进电机工作原理比较简单。下面先叙述三相反应式步进电机原理。

1、结构:

电机转子均匀分布着很多小齿，定子齿有三个励磁绕组，其几何轴线依次分别与转子齿轴线错开。0、 $1/3\tau$ 、 $2/3\tau$ ，(相邻两转子齿轴线间的距离为齿距以 τ 表示)，即A与齿1相对齐，B与齿2向右错开 $1/3\tau$ ，C与齿3向右错开 $2/3\tau$ ，A'与齿5相对齐，(A'就是A，齿5就是齿1)下面是定转子的展开图



2、旋转:

如A相通电，B，C相不通电时，由于磁场作用，齿1与A对齐，(转子不受任何力以下均同)。

如B相通电，A，C相不通电时，齿2应与B对齐，此时转子向右移过 $1/3\tau$ ，此时齿3与C偏移为 $1/3\tau$ ，齿4与A偏移 $(\tau - 1/3\tau) = 2/3\tau$ 。

如C相通电，A，B相不通电，齿3应与C对齐，此时转子又向右移过 $1/3\tau$ ，此时齿4与A偏移为 $1/3\tau$ 对齐。

如A相通电，B，C相不通电，齿4与A对齐，转子又向右移过 $1/3\tau$

这样经过A、B、C、A分别通电状态，齿4（即齿1前一齿）移到A相，电机转子向右转过一个齿距，如果不断地按A，B，C，A... ..通电，电机就每步（每脉冲） $1/3\tau$ ，向右旋转。如按A，C，B，A... ..通电，电机就反转。

由此可见：电机的位置和速度由导电次数（脉冲数）和频率成一一对应关系。而方向由导电顺序决定。不过，出于对力矩、平稳、噪音及减少角度等方面考虑。往往采用A-AB-B-BC-C-CA-A这种导电状态，这样将原来每步 $1/3\tau$ 改变为 $1/6\tau$ 。甚至于通过二相电流不同的组合使其 $1/3\tau$ 变为 $1/12\tau$ ， $1/24\tau$ ，这就是电机细分驱动的基本理论依据。

不难推出：电机定子上有m相励磁绕组，其轴线分别与转子齿轴线偏移 $1/m, 2/m, \dots, (m-1)/m$ 。并且导电按一定的相序电机就能正反转被控制这是步进电机旋转的物理条件。只要符合这一条件我们理论上可以制造任何相的步进电机，出于成本等多方面考虑，市场上一般以二、三、四、五相为多。

3、力矩：

电机一旦通电，在定转子间将产生磁场（磁通量 Φ ）当转子与定子错开一定角度产生力 F 与 $(d\Phi/d\theta)$ 成正比。

其磁通量 $\Phi = Br \cdot S$ 。Br为磁密，S为导磁面积，F与 $L \cdot D \cdot Br$ 成正比，L为铁芯有效长度，D为转子直径
 $Br = N \cdot I / R$

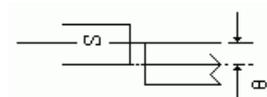
$N \cdot I$ 为励磁绕组安匝数（电流乘匝数）R为磁阻。力矩=力*半径。

力矩与电机有效体积*安匝

数*磁密成正比（只考虑线性状态）因此，电机有效体积越大，

励磁安匝数越大，定转子间

气隙越小，电机力矩越大，反之亦然。

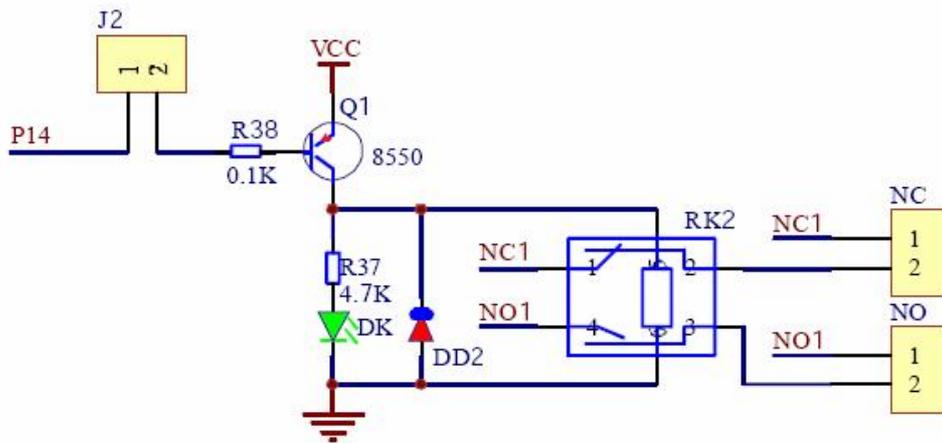


步进电机例程应用：

在HC6800E-V2.8开发板上单片机的管脚通过排线连接ULN2003驱动步进电机，使用步进电机时，具体例子看光盘例程。（学习光盘\单片机例程\步进马达或5轴步进电机）

继电器

继电器由P1.4控制，当要使用此功能时，J2必须短接。



继电器原理图

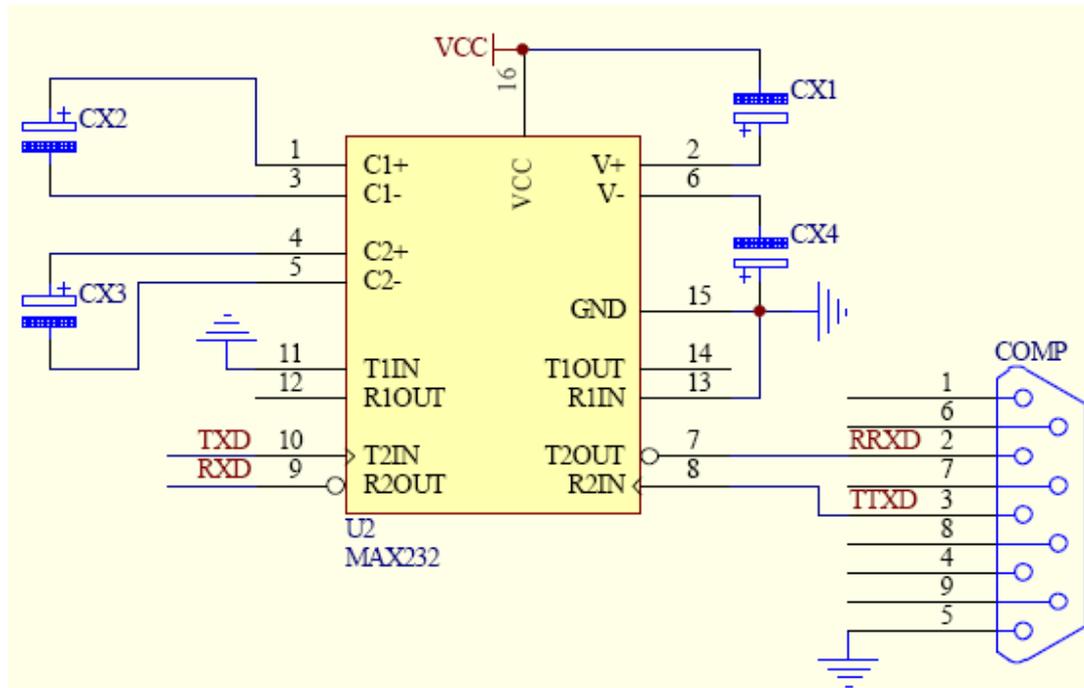
继电器是一种当输入量（电、磁、声、光、热）达到一定值时，输出量将发生跳跃式变化的自动控制器件。继电器的输入信号 x 从零连续增加达到衔铁开始吸合时的动作值 x_x ，继电器的输出信号立刻从 $y=0$ 跳跃到 $y=y_m$ ，即常开触点从断到通。一旦触点闭合，输入量 x 继续增大，输出信号 y 将不再起变化。

继电器应用例程：

在HC6800E-V2.8开发板上提供2个继电器，单片机先通过管脚P1.4驱动74HC14，通过它来驱动ULN2003，再由ULN2003控制继电器，具体例子看光盘例程。(学习光盘\单片机例程\继电器)

:

Rs232通讯



MAX232电路图

RS232介绍

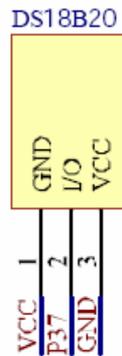
RS232接口是制定用于串行通讯的标准。该标准规定采用一个25个脚的Db25连接器，对连接器的每个引脚的信号内容加以规定，还对各种信号的电平加以规定。DB25的串口一般只用到的管脚只有2 (RXD)、3 (TXD)、7 (GND) 这三个，随着设备的不断改进，现在DB25针很少看到了，代替他的是DB9的接口，DB9所用到的管脚比DB25有所变化，是2 (RXD)、3 (TXI)、5 (GND) 这三个。因此现在都把RS232接口叫做DB9。

MAX232应用例程:

。具体例子看光盘例程。(学习光盘\单片机例程\RS232通讯)。

温度传感DS18B20

此电路的上拉电阻在《中央控制器（CPU）》模块中



DS18B20原理图

DS18B20介绍

DALLAS公司单线数字温度传感器DS18B20是一种新的“一线器件”，它具有体积小、适用电压宽等特点。一线总线独特而且经济的特点，使用户可轻松地组建传感器网络，为测量系统的构建引入全新概念。DS18B20支持“一线总线”接口，测量温度范围为 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ，在 $-10^{\circ}\text{C} \sim +85^{\circ}\text{C}$ 范围内，精度为 $\pm 0.5^{\circ}\text{C}$ ；通过编程可实现9~12位的数字值读数方式；可以分别在93.75ms和750ms内将温度值转化为9位和12位的数字量。每个DS18B20具有唯一的64位长序列号，存放于DS18B20内部ROM只读存储器中。DS18B20温度传感器的内部存储器包括1个高速暂存RAM和1个非易失性的电可擦除

E2RAM，后者存放高温度和低温度触发器TH、TL和结构寄存器。暂存存储器包含了8个连续字节，前2字节为测得的温度信息，第1个字节为温度的低8位，第2个字节为温度的高8位。高8位中，前4位表示温度的正(全“0”)与负(全“1”)；第3个字节和第4个字节为TH、TL的易失性拷贝；第5个字节是结构寄存器的易失性拷贝，此三个字节内容在每次上电复位时被刷新；第6、7、8个字节用于内部计算；第9个字节为冗余检验字节。所以，读取温度信息字节中的内容，可以相应地转化为对应的温度值。表1列出了温度与温度字节间的对应关系。

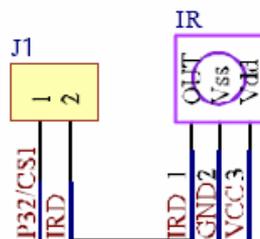
DS18B20例程应用

在HC6800E-V2.8中，DS18B20的信号管脚DQ直接和MCU的P3.7相连，具体例子看光盘例程。(学习光盘\单片机例程\温度传感)。

红外接收与发射

将J1跳线短接时，P3.2脚与红外输出端相连，此时开启了红外功能。当P3.2脚做别的用途时最好断开，以免影响。

此电路的上拉电阻在《中央控制器（CPU）》模块中



红外接收图原理图

红外通信是利用红外技术实现两点间的近距离保密通信和信息转发。它一般由红外发射和接收系统两部分组成。发射系统对一个红外辐射源进行调制后发射红外信号，而接收系统用光学装置和红外探测器进行接收，就构成红外通信系统。

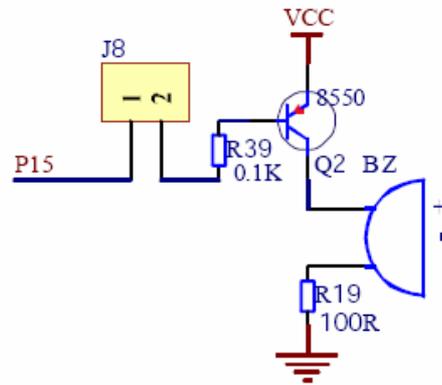
红外接收头一般是接收、放大、解调一体头，一般红外信号经接收头解调后，数据“0”和“1”的区别通常体现在高低电平的时间长短或信号周期上，单片机解码时，通常将接收头输出脚连接到单片机的外部中断，结合定时器判断外部中断间隔的时间从而获取数据。重点是找到数据“0”与“1”间的波形差别。

红外线例程应用

在HC6800E-V2.8中，红外线的信号管脚IRD直接和MCU的P3.2相连，具体例子看光盘例程。(学习光盘\单片机例程\红外线)。

蜂鸣器

J8跳线短接时，P1.5脚与蜂鸣器相连，开启了蜂鸣器功能，当P1.5脚做别的用途时最好断开，以免影响。



蜂鸣器原理图

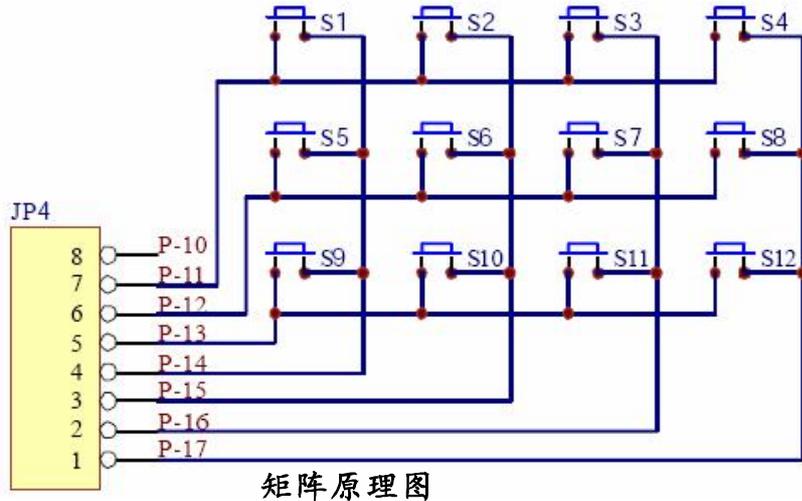
蜂鸣器例程应用：

该单片机开发板上有一个直流蜂鸣器(Buzzer)。，蜂鸣器通过单片机的P1.5脚，当单片机的管脚输出固定电平时蜂鸣器响；声音是由震动所产生的，一定频率的震动就产生了一定频率的声音。如果我们控制P1.5引脚以固定的电平输出蜂鸣器会发出一个固定的声音，改变频率将会发出其他的声音。我们可以通过控制频率来控制音阶通过控制一个频率维持的时间来控制音长，如果控制得当这样就可以让蜂鸣器发出悦耳的音乐了。

具体例子看光盘例程。(学习光盘\单片机例程\蜂鸣器)

矩阵按键

JP4为矩阵键盘的接口，P1.4、P1.5、P1.6为行，P1.0、P1.1、P1.2、P1.3为列。使用8P排线把JP4与JP8JP11中任何接口相连，实现矩阵键盘功能



矩阵式键盘的结构与工作原理：

在键盘中按键数量较多时，为了减少I/O口的占用，通常将按键排列成矩阵形式，如图所示。在矩阵式键盘中，每条水平线和垂直线在交叉处不直接连通，而是通过一个按键加以连接。这样，一个端口（如P1口）就可以构成 $4 \times 4 = 16$ 个按键，比之直接将端口线用于键盘多出了一倍，而且线数越多，区别越明显，比如再多加一条线就可以构成20键的键盘，而直接用端口线则只能多出一键（9键）。由此可见，在需要的键数比较多时，采用矩阵法来做键盘是合理的。

矩阵式结构的键盘显然比直接法要复杂一些，识别也要复杂一些，上图中，列线通过电阻接正电源，并将行线所接的单片机的I/O口作为输出端，而列线所接的I/O口则作为输入。这样，当按键没有按下时，所有的输出端都是高电平，代表无键按下。行线输出是低电平，一旦有键按下，则输入线就会被拉低，这样，通过读入输入线的状态就可得知是否有键按下了。具体的识别及编程方法如下所述。

确定矩阵式键盘上何键被按下介绍一种“行扫描法”。行扫描法又称为逐行（或列）扫描查询法，是一种最常用的按键识别方法，如上图所示键盘，介绍过程如下：

1.判断键盘中何键按下将全部行线P10-P13置低电平，然后检测列线的状态。只要有一列的电平为低，则表示键盘中有键被按下，而且闭合的键位于低电平线与4根行线交叉的4个按键之中。若所有列线均为高电平，则键盘中无键按下。

2.判断闭合键所在的位置在确认有键按下后，即可进入确定具体闭合键的过程。其方法是：依次将行线置为低电平，即在置某根行线为低电平时，其它线为高电平。在确定某根行线位置为低电平后，再逐行检测各列线的电平状态。若某列为低，则该列线与置为低电平的行线交叉处的按键就是闭合的按键。

矩阵式键盘例程应用

在HC6800E-V2.8中，具体例子看光盘例程。(学习光盘\单片机例程\矩阵按键