

目 录

第一章 单片机实验箱.....	3
1、51 单片机核心板.....	3
2、配套 51 主芯片 STC12C5A60S2.....	3
3、实验箱功能底板.....	4
第二章 烧录第一个程序.....	7
1、STC51 单片机下载（烧写程序）方法.....	7
2、下载软件.....	7
3、硬件连接.....	8
4、软件设置.....	10
5、程序下载.....	11
6、常见问题及解决.....	12
第三章 硬件实验及功能测试.....	15
实验一 单色流水灯测试.....	15
实验二 彩色流水灯测试.....	18
实验三 双色灯测试.....	20
实验四 单个数码管显示.....	22
实验五 矩阵键盘显示.....	26
实验六 8X8 红绿双色点阵.....	29
实验七 LCD1602 液晶显示.....	32
实验八 LCD12864 液晶显示.....	36
实验九 OLED 模块显示.....	39
实验十 TFT 彩屏模块测试.....	41
实验十一 五向摇杆按键测试.....	43
实验十二 DS18B20 温度显示.....	46
实验十三 直流电机测试.....	50
实验十四 步进电机控制.....	52
实验十五 继电器控制.....	55
实验十六 串口通讯测试.....	57
实验十七 DS1302 时钟模块测试.....	59
实验十八 蜂鸣器测试.....	62
实验十九 AT24C02 模块测试.....	65
实验二十 74HC164 串转并测试.....	67
实验二十一 74HC165 并转串测试.....	69
实验二十二 74HC138 三八译码器.....	72
实验二十三 74HC573 锁存器.....	75
实验二十四 红外发送接收测试.....	78
实验二十五 红外解码测试.....	80
实验二十六 SD 卡模块测试.....	83
实验二十七 PCF8591AD/DA 模块测试.....	86
实验二十八 自主脉冲发生器测试.....	89
第四章 C 语言开发平台 keil c51.....	91

1、 μ Vision 集成开发环境简介.....	91
2、 μ Vision3 集成开发环境的安装.....	92
3、基于 Keil C51 的第一个 C 语言程序项目的建立.....	95
4、 μ Vision3 集成开发环境下的软件模拟仿真.....	105
第五章 仿真.....	109
1、仿真知识.....	109
2、硬件连接.....	109
3、keil 打开工程.....	109
4、设置项目属性.....	110
5、开始仿真.....	111

第一章 单片机实验箱

1、51 单片机核心板



单片机最小系统核心板支持 STC89X 系列和 AT89X 系列双列直插单片机芯片，核心板采用 DIP40 双列直插锁紧座，方便更换芯片，芯片 IO 全部引出 2 组，可以直接插到实验箱底板上，也可以单独使用。

单片机最小系统核心板标配一个 STC12C5A60S2 单片机。

- 板载标准 ISP 接口，使用 AT 系列单片机时可以使用 USB-ASPIISP 下载器在线编程，
- 支持在线仿真功能，将 51 单片机核心板更换为 51 仿真器（也可以直接插入），即可实现在线仿真功能，可以跟踪程序执行过程，更方便程序调试。
- 板载多组电源引出，5V 电源 8 路、3.3V 电源 8 路、1.8V 电源 8 路、GND16 路，都留足余量，安全可靠，方便扩展模块供电使用。

2、配套 51 主芯片 STC12C5A60S2

单片机核心板采用了宏晶公司最新的增强型 8051 单片机，51 芯片选用的是 DIP 封装的 STC12C5A60S2，具有 ISP 编程功能，完全兼容 STC89C52，AT89S52 单片机。

STC12C5A60S2 系列单片机是宏晶科技生产的单时钟/机器周期(1T)的单片机，是高速/低功耗/超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051，但速度快 8-12 倍。内部

集成 MAX810 专用复位电路, 2 路 PWM, 8 路高速 10 位 A/D 转换 (250K/S, 即 25 万次/秒), 针对电机控制, 强干扰场合。

--工作电压: 3.3v-5.5v

--工作频率范围: 0-35mhz, 相当于普通 8051 的 0~420mhz

--用户应用程序空间: 60k

--片上集成 1280 字节 ram

--通用 i/o 口 36 个

--双串口, rxd2/p1.2(可通过寄存器设置到 p4.2), txd2/p1.3(可通过寄存器设置到 p4.3)

--工作温度范围: -40~+85℃(工业级)/0-75℃(商业级)

--封装: pdip-40

注意: STC12 系列单片机取代传统 8051 注意事项

1、STC12C5A60 系列单片机的定时器 0/定时 1 与传统 8051 完全兼容, 上电复位后, 定时器部分缺省还是除 12 再计数, 而串口由定时器 1 控制速度, 所以定时器/串口完全兼容。

2、增加了独立波特率发生器, 省去了传统 8051 的定时器 2, 如是用 T2 做波特率的, 请改用独立波特率发生器做波特率发生器。

3、传统 8051 的 111 条指令执行速度全面提速, 最快的指令快 24 倍, 最慢的指令快 3 倍, 靠软件延时实现精确延时的程序需要调整。

3、实验箱功能底板



单片机核心板既可直接插到功能底板组合为多功能实验箱, 也可以作为一个最小系统核心板单独使用, 灵活度高、扩展性极强。

功能底板板载多个芯片, 集成 40 大功能模块及接口。各个功能模块使用粗线条把对应的模块分开, 从上图可以清晰看出模块区域。

具体硬件资源信息如下：

- 1) 8 个单色流水灯
- 2) 6 个彩色流水灯（红 2、黄 2、绿 2）
- 3) 1 个双色灯
- 4) 2 个 4 位一体数码管
- 5) 1 个单位数码管
- 6) 1 个 8×8 红绿双色点阵
- 7) 1 个 LCD1602 液晶显示器和接口
- 8) 1 个 LCD12864 液晶显示器（可选配）和接口
- 9) 1 个 OLED 显示器（可选配）和接口
- 10) 1 个 TFT 彩色显示器（可选配）和接口
- 11) 8 个独立按键
- 12) 4×4 矩阵键盘
- 13) 5 向摇杆控制模块
- 14) 两路 DS18B20
- 15) 热敏传感器模块
- 16) 光敏传感器模块
- 17) 霍尔传感器模块
- 18) 1 路直流电机和控制电路
- 19) 1 路步进电机和控制
- 20) 2 路继电器
- 21) 2 路 DB9 串口（1 公 1 母）
- 22) DS1302 时钟模块
- 23) 无源蜂鸣器
- 24) AT24C02 模块
- 25) 模拟量输出模块
- 26) 自主脉冲发生器模块
- 27) 8 位拨码开关模块作开关量输出
- 28) 74HC164 串口转并口模块
- 29) 74HC165 并口转串口模块
- 30) SD 卡模块
- 31) MAX485 模块

- 32) 红外发射和接收模块
- 33) PCF8591 AD/DA 一体化数模转换模块
- 34) 74HC573 锁存电路
- 35) 74LS138 三八译码器电路
- 36) 3V3 电源电路
- 37) 1.8V 电源电路
- 38) 多路 3V3/5V/1V8/GND 电源输出
- 39) 核心板扩展接口

除 51 单片机核心板之外，实验箱还支持以下单片机系统的实验和开发：

- AVR16 单片机核心板
- PIC 单片机核心板
- MSP430 单片机核心板
- ARM(STM32)核心板

第二章 烧录第一个程序

程序编写、编译、完毕后，编译器会自动生成单片机能够识别的可执行文件（一般是 .HEX 格式的文件），只有将这些文件“安装”到单片机里面，才能够让单片机执行相应的功能。通常情况下，我们把将文件“安装”到单片机里面的步骤称为“烧录”或者“下载”。

1、STC51 单片机下载（烧写程序）方法

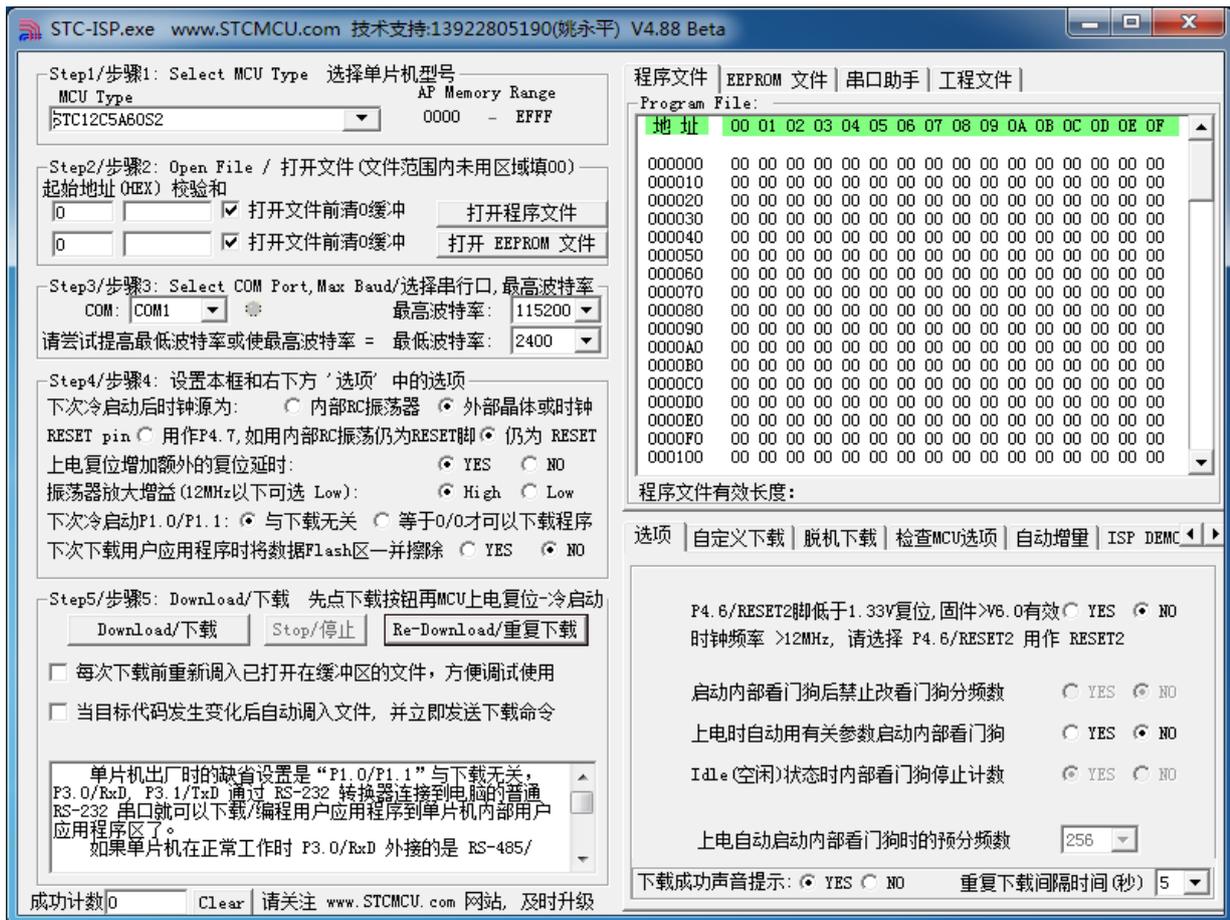
具体如何将编译好的 HEX 文件烧录到单片机里面呢？这需要借助专用的烧录软件和烧录工具。STC 单片机集成了 ISP 在系统编程功能，这使得我们只需要一个烧录软件就可以轻松的将程序烧录到单片机里面。

STC 系列单片机支持串口下载，不需要另外的下载器就可以直接下载程序。

STC12 系列单片机内部固化有 ISP 系统引导固件，配合 PC 端的控制程序即可将用户的程序代码下载进单片机内部，具有在系统可编程 (ISP) 特性，ISP 的好处是：省去购买通用编程器，单片机在用户系统上即可下载/烧录用户程序（速度比通用编程器快，几秒一片），而无须将单片机从已生产好的产品上拆下，再用通用编程器将程序代码烧录进单片机内部。

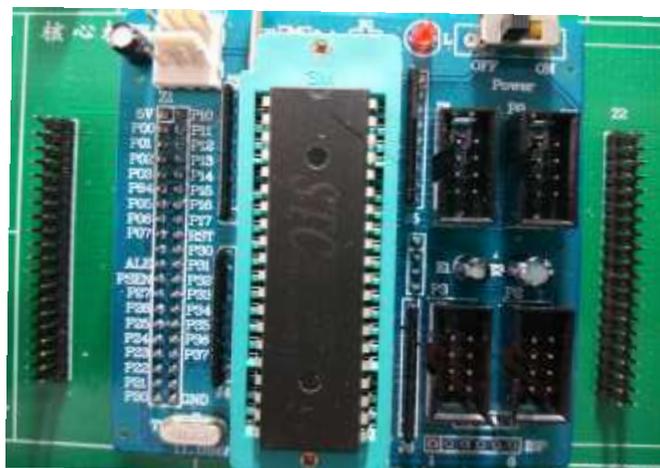
2、下载软件

在这里，我们使用的下载软件是 stc-isp，版本号为 stc-isp-v4.86，在配套光盘中有提供。软件界面如下：



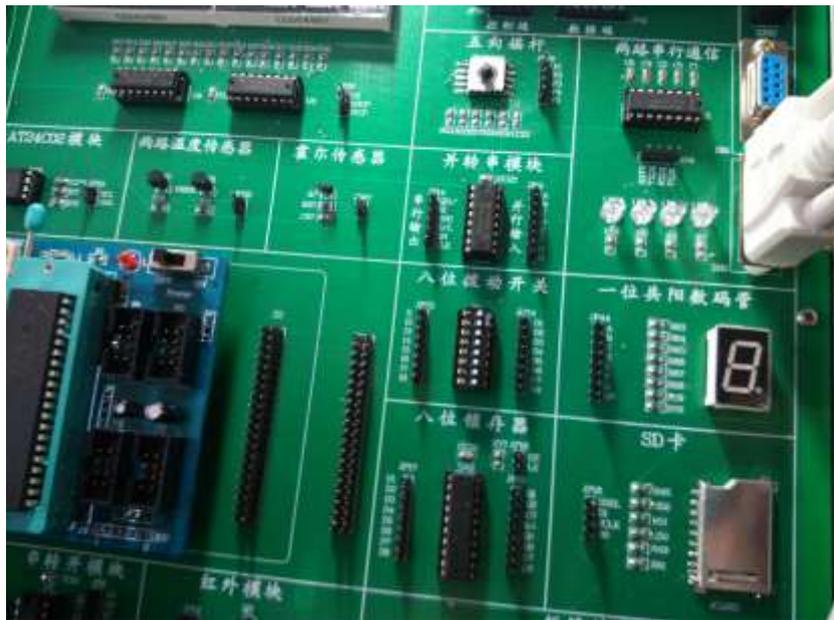
3、硬件连接

- 1) 将随机配备的电源线（交流 220V）接到机箱后侧面的电源插座上。
- 2) 将单片机核心板插到实验箱功能底板上（出厂时已插好）。



注意核心板插入的方向，一定不能插反，按上面图示插入。

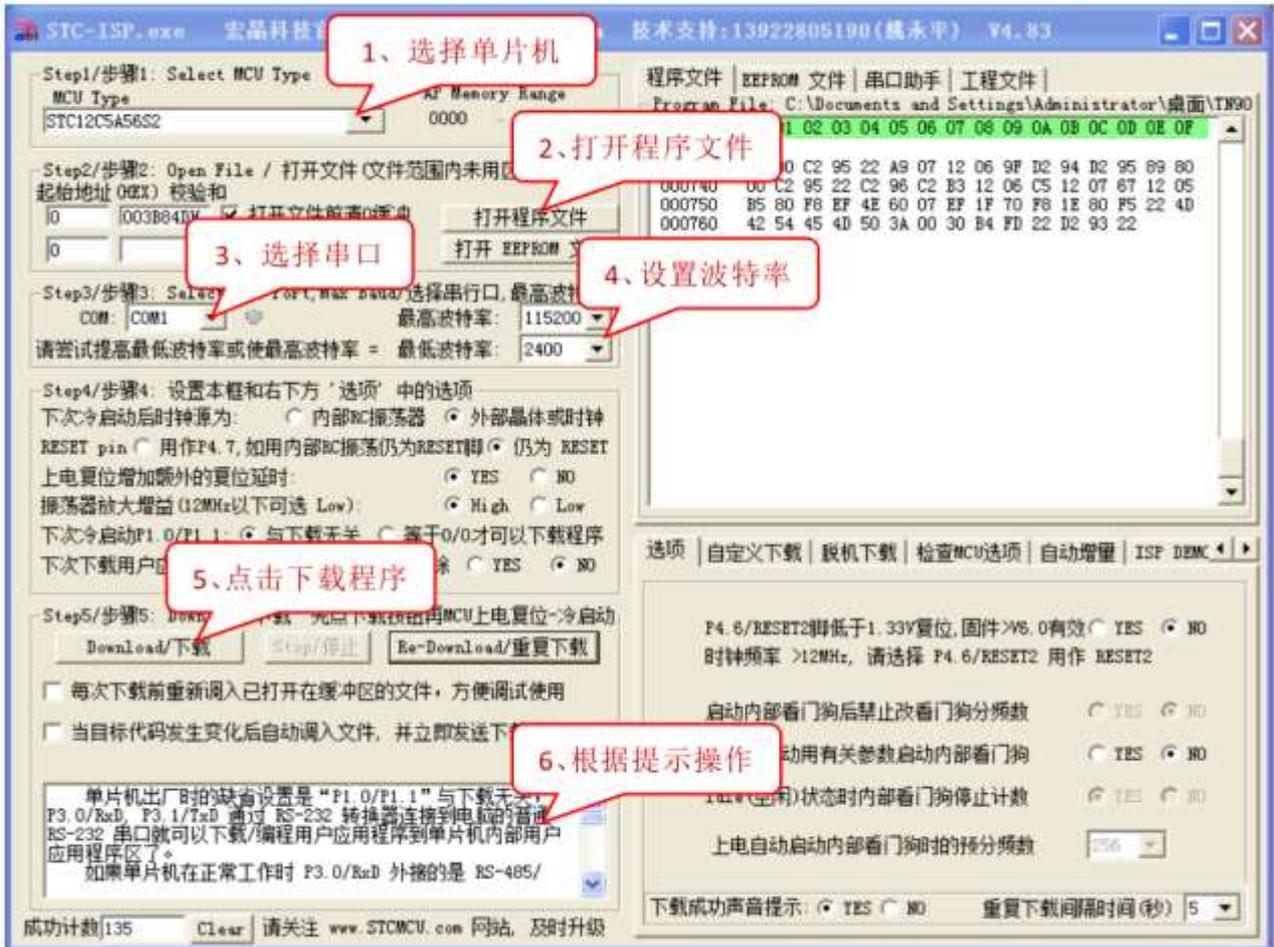
3) 将串口线一端与机箱 DB25 口连接，一端接电脑串口。



这样我们就完成了实验箱和电脑的硬件连接，可以准备下载程序了。

4) 可靠连接后，机箱右侧面电源开关拨到 I 位置，功能底板上的电源指示灯应该会点亮，就可以继续下面的操作了。如果不亮，立即关闭电源，检查上面的 1-3 个步骤是否有误，直到正常为止。

4、软件设置



1) 选择单片机型号

在图中 1 处选择单片机芯片型号，我们使用 **STC12C5A60S2**。

2) 设置串口号

在图中 3 处选择串口号，一般选择为 COM1 口，根据实际使用哪个口来选。

3) 设置波特率

在图中 4 处选择，最低波特率设置为 1200，最高波特率设置为 1152000。

如果下载不成功，可以适当调低波特率设置，以便正常下载，一般不需要特殊设置。

4) 注意事项

其他选项选择默认值，不要随意更改。

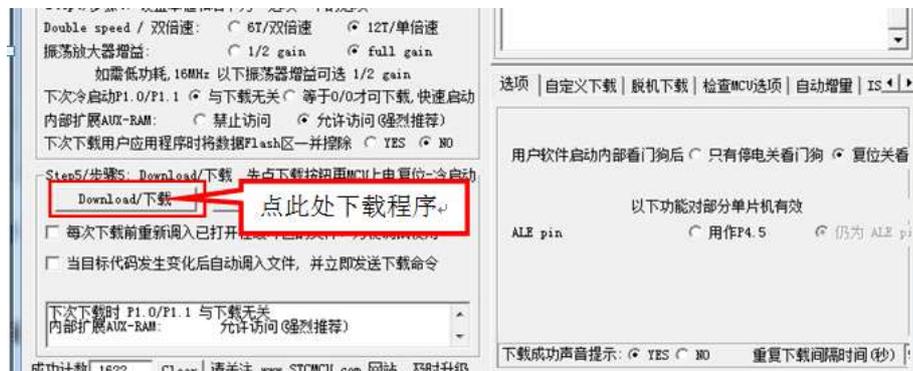
5、程序下载

1) 加载程序

点击上图中 2 处选择要加载的程序（HEX 格式），在随机光盘中有提供配套测试程序以及配套实验程序，选择你指定位置的 HEX 程序即可。

2) 点击下载

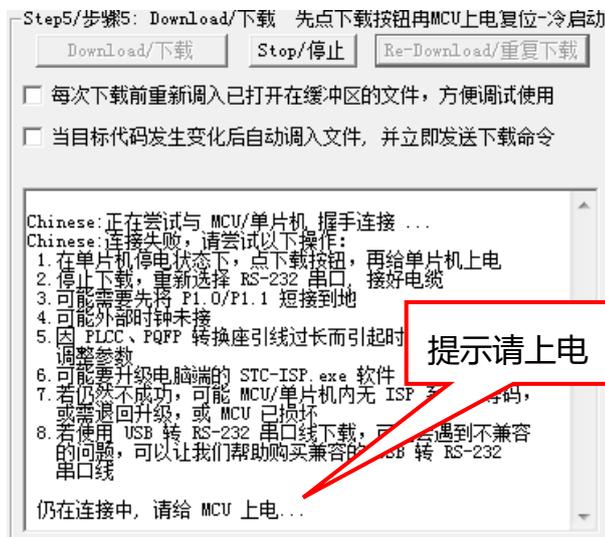
点击上图中的 5 处“Download/下载”，准备向单片机写入程序。



3) 上电

注意：在本操作前，实验箱已通电，但核心板右上方的开关 Power 必须在 OFF 位置，否则软件检测不到合法的下载命令流，单片机就直接跑用户程序了。也就是说一定要在选择“Download/下载”按钮后，再给要下载的单片机芯片上电。

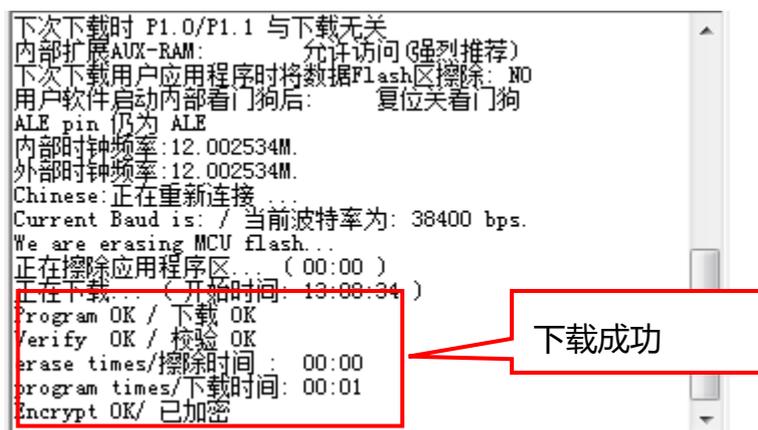
下载时注意看提示，主要看是否要给单片机上电，参看下图。



软件提示请上电后，再给核心板上电，在实验箱已通电的情况下，将核心板上方的开关 Power 拨在 ON 位置，软件将自动下载程序到单片机中。

4) 下载成功

程序下载成功后的提示如下：



5) 运行程序

下载完成后，按核心板上复位键就可以运行刚下载的程序了。

6、常见问题及解决

1) 串口占用或者串口号不正确

提示信息：

Chinese:正在尝试与 MCU/单片机 握手连接 ...

打开串口失败!

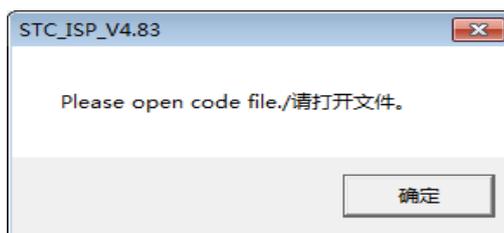
Chinese:串口已被其它程序打开或该串口不存在。

解决办法：

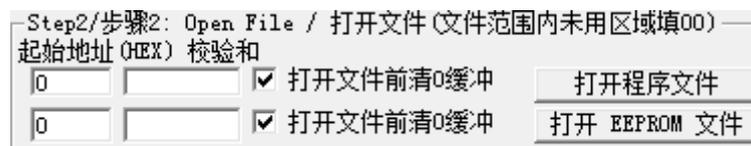
请到设备管理器检查串口号，如串口号正确请检查是否有其他软件打开占用串口

2) 未打开需要下载的 HEX 文件

提示信息：



解决办法:



点击打开程序文件，选择需要下载的 HEX 文件。

3) 波特率设置太高

提示信息:

Chinese:正在重新连接 ...

Connection failed. / 握手失败 (End: 16:55:10)

解决办法:

调低波特率，极端情况将最高、最低波特率都调整至 1200，如果下载成功，可以尝试调高最高波特率，至仍可以正常下载。

友情提示: 如波特率不稳定，将最高波特率和最低波特率设置为一样，可以获得比较高的下载成功率。

4) 芯片型号选择不正确

提示信息:

MCU Type is wrong. / MCU 型号错，请重选

或

仍在连接中，请给 MCU 上电...

解决办法:

请查看芯片上面型号，对照检查 STC-ISP 软件中芯片型号设置，选择正确的芯片型号。

5) 核心板外部晶振没有插上或者晶振坏了

提示信息:

仍在连接中，请给 MCU 上电...

解决办法:

确认 51 单片机核心板晶振 TI 是完好的。

6) 51 芯片锁死

提示信息:

仍在连接中，请给 MCU 上电...

故障原因:

下载时候不小心选择了错误选项

Step4/步骤4: 设置本框和右下方 '选项' 中的选项

Double speed / 双倍速: 6T/双倍速 12T/单倍速

振荡放大器增益: 1/2 gain full gain

 如需低功耗, 16MHz 以下振荡器增益可选 1/2 gain

下次冷启动P1.0/P1.1 与下载无关 等于0/0才可下载, 快速启动

内部扩展AUX-RAM: 禁止访问 允许访问 (强烈推荐)

下次下载用户应用程序时将数据Flash区一并擦除 YES NO

正确: 下次冷启动P1.0/P1.1 与下载无关 等于0/0才可下载, 快速启动

错误: 下次冷启动P1.0/P1.1 与下载无关 等于0/0才可下载, 快速启动

下载完成后必须按照说明书使用杜邦线将芯片指定引脚连接到相应的功能模块对应 IO 引脚，才能看到程序正常运行的效果，每个程序的头部介绍部分都会注明连接方式。

第三章 硬件实验及功能测试

实验一 单色流水灯测试

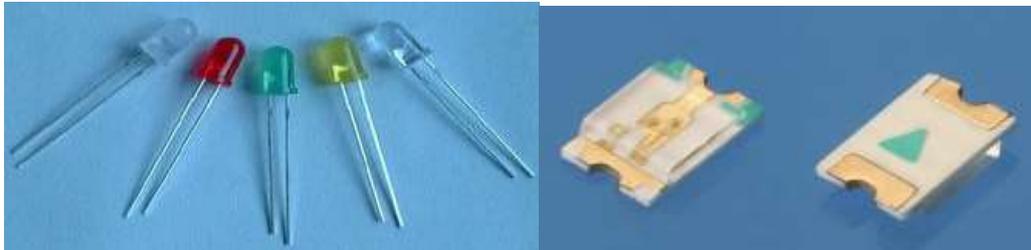
一、实验目的

- 1、掌握单片机 P1 口简单使用。
- 2、掌握发光二极管点亮的原理。
- 3、学习延时程序的编写和使用。

二、实验内容

1、P1 口做输出口，依次 P1.0-P1.7 接“单色流水灯”模块的 8 个发光二极管 L1-L8，编写程序，使 P1 口接的 8 个发光二极管 L1-L8 按流水的方式点亮。

三、实验说明



发光二极管：它具有单向导电性，通过5mA左右电流即可发光，电流越大，其亮度越强，但若电流过大会烧毁二极管；一般我们设计电路时电流控制在3mA~20mA之间，那么就要给发光二极管串联一个电阻，其目的就是为了限制通过发光二极管的电流不要太大，这个电阻习惯称为“限流电阻”。发光二极管的发光亮度与通过的工作电流成正比，限流电阻R可用下式计算：

$$R = (E - U_d) / I_d$$

式中E 为电源电压， U_d 为LED 的正向压降， I_d 为LED 的一般工作电流。普通发光二极管的正向饱和压降为1.4~2.1V，正向工作电流为5~20mA，如发光二极管正常导通时，两端电压约为1.5V，发光管的阴极为低电平(0V)；阳极串接一个电阻，电阻的另一端接VCC(+5V)，计算出加在电阻两端的电压为5V-1.7V=3.3V，穿越电阻的电流选择3.3mA，带入公式 $R=3.3V/3.3mA$ ，得到 $R=1K$ 电阻。



实际电路应用中发光二极管主要有三种颜色，然而三种发光二极管的压降都不相同，具体压降参考值如下：

- 红色发光二极管的压降为2.0--2.2V；
- 黄色发光二极管的压降为1.8—2.0V；
- 绿色发光二极管的压降为3.0—3.2V；

正常发光时的额定电流约为20mA，实验中发光二极管根据电路图，应该是低电平点亮。

四、实验原理图

见图 3-1-1 所示

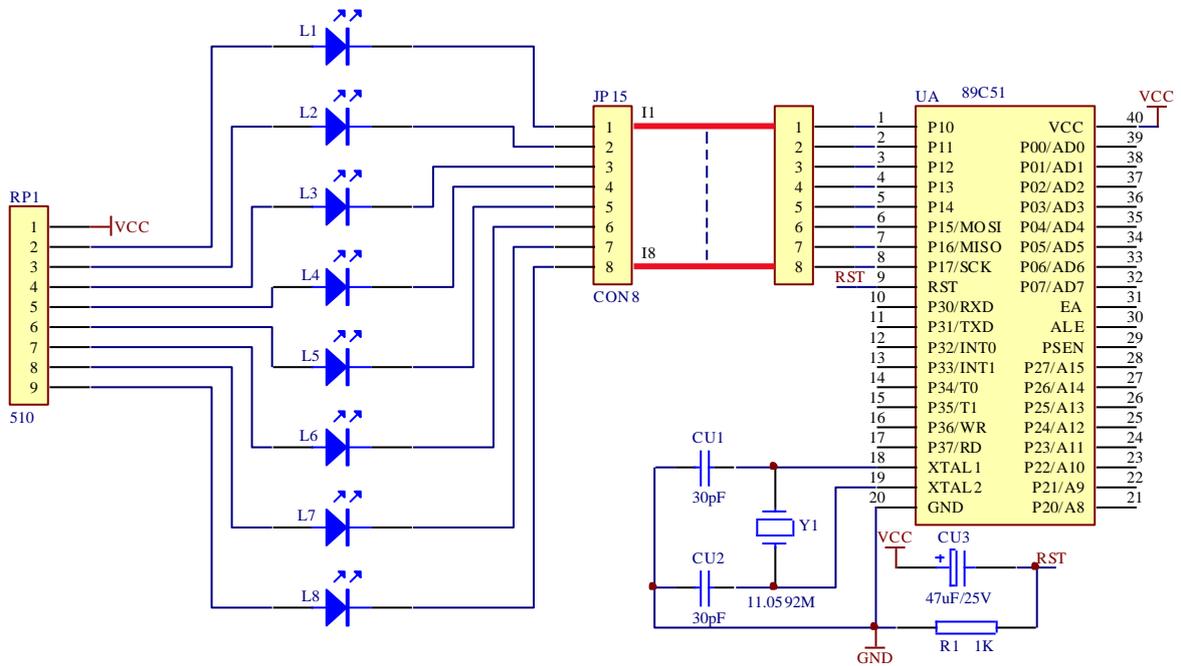


图 3-1-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-1-2 所示

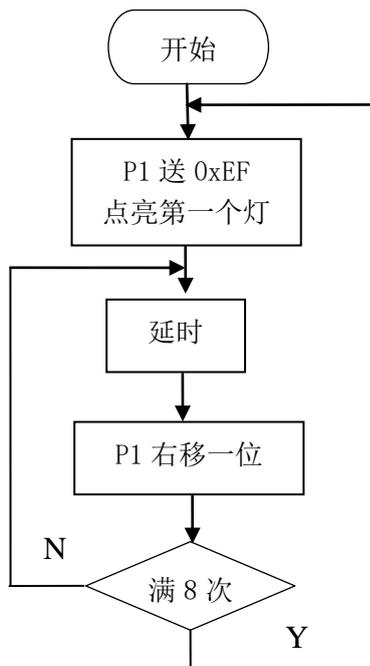


图 3-1-2

六、实验步骤

- 1、准备 8P 杜邦线 1 根, 将 P1.0~P1.7 用 8 芯杜邦线连至“单色流水灯”单元的 JP15(I1~I8) ;
- 2、在 KEIL 软件中调试 LED1 工程文件, 用 STC 软件下载 LED1.HEX 程序;
- 3、实验现象: 8 个 LED 灯 **L8-L1** 依次点亮, 之后同时亮灭。

实验二 彩色流水灯测试

一、实验目的

- 1、掌握单片机 P1 口简单使用。
- 2、掌握发光二极管点亮的原理。
- 3、学习延时程序的编写和使用。

二、实验内容

1、P1 口做输出口，接“彩色流水灯”模块的 6 个发光二极管 L9—L14，编写程序，使 P1 口接的 6 个发光二极管先亮 2 个绿灯，再亮 2 个黄灯，最后 2 个红灯。

三、实验说明

发光二极管根据电路图，低电平点亮。

硬件准备：8P 杜邦线 1 根或 1P 杜邦线 6 根。

四、实验原理图

见图 3-2-1 所示

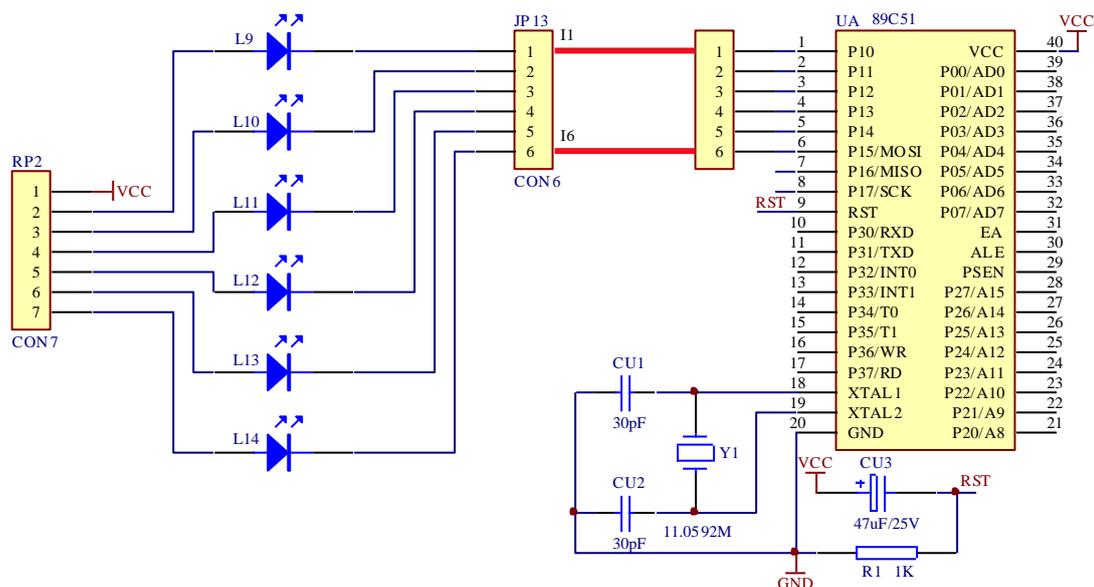


图 3-2-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-2-2 所示

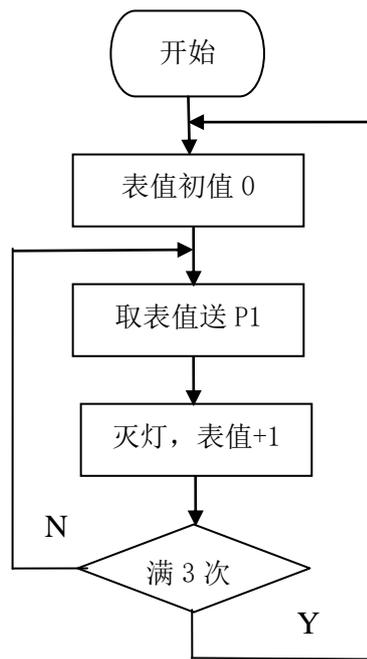


图 3-2-2

六、实验步骤

- 1、P1.0~P1.7 用 8 芯线对应连至“彩色流水灯”单元的 JP13 (I1~I6)；
- 2、在 KEIL 软件中调试 LED2 工程文件。用 STC 软件下载 LED2.HEX 程序；
- 3、实验现象：先亮 2 个绿灯，再亮 2 个黄灯，最后 2 个红灯。

实验三 双色灯测试

一、实验目的

- 1、掌握单片机 P1 口简单使用。
- 2、掌握双色二极管点亮的工作原理。
- 3、学习延时程序的编写和使用。

二、实验内容

- 1、P1 口做输出口，接“双色灯”模块的发光二极管 L15，编写程序，红灯绿灯交替点亮。

三、实验说明

发光二极管根据电路图，低电平点亮，红灯绿灯同时亮时为橙色。

硬件准备：1P 杜邦线 2 根。

四、实验原理图

见图 3-3-1 所示

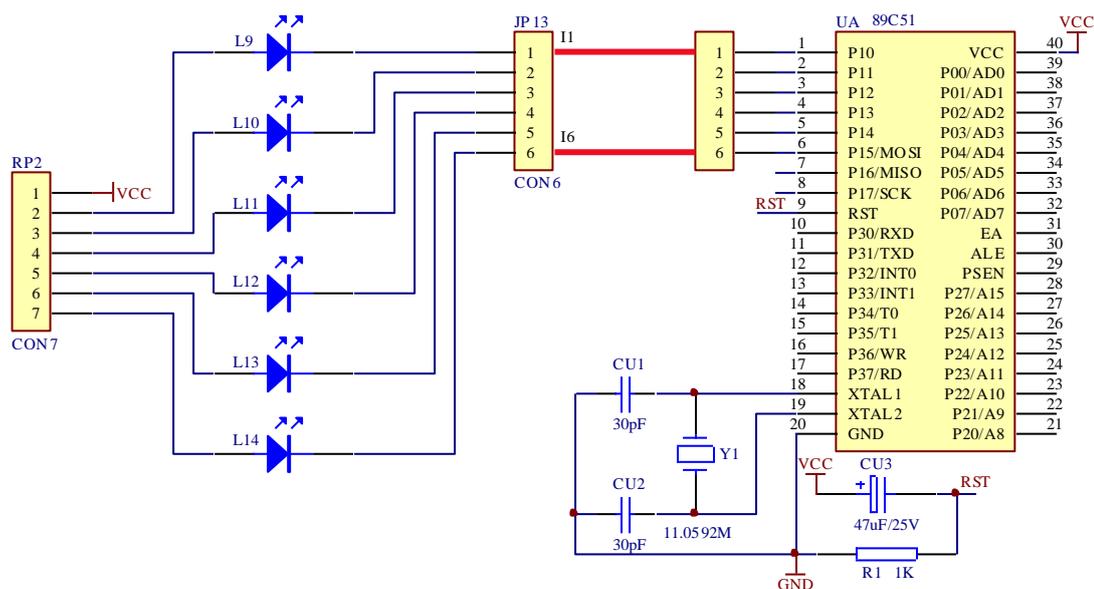


图 3-3-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-3-2 所示

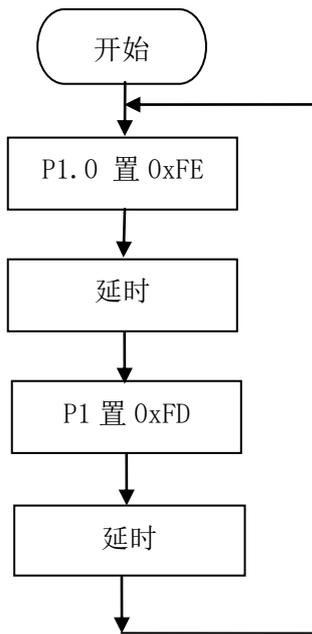


图 3-3-2

六、实验步骤

- 1、P1.0~P1.1 用杜邦线对应连至“双色灯”单元的 JP19 (I1~I2)；
- 2、在 KEIL 软件中调试 LED3 工程文件。用 STC 软件下载 LED3. HEX 程序；
- 3、实验现象：先绿灯亮，后红灯亮，交替闪烁。

实验四 单个数码管显示

一、实验目的

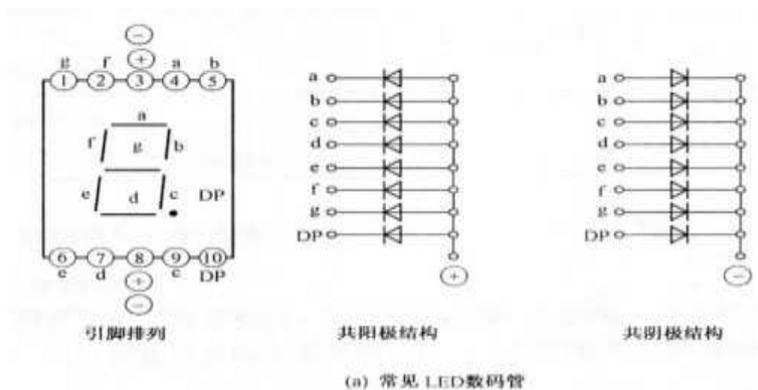
- 1、掌握数码管的工作原理。
- 2、掌握数码管的显示编码。
- 3、掌握单片机端口作为输入口的应用

二、实验内容

- 1、P3 口做输出口，去驱动数码管。
- 2、P1 口作输入口，为独立按键的输入。
- 3、编制程序使不同的按键对应不同的显示值。

三、实验说明

数码管的显示原理都是一样的，都是靠点亮内部的发光二极管来发光，下面我们就来讲解一个数码管是如何亮起来的。数码管内部电路如下图所示，从左图可看出，一位数码管的引脚是 10 个，显示一个 8 字需要 7 个小段，另外还有一个小数点，所以其内部一共有 8 个小的发光二极管，最后还有一个公共端，生产商为了封装统一，单位数码管都封装 10 个引脚，其中第 3 和第 8 引脚是连接在一起的作为公共端。而公共端又可分为共阳极和共阴极，中间图为共阴极内部原理图，右图为共阳极内部原理图。



上图展出了常用的两种数码管的引脚排列和内部结构。总所周知，就是要给予它足够大的正向压降。所以点亮数码管其实也就是给它内部相应的发光二极管正向压降。如上图左（一共 a、b、c、d、e、f、g、DP 八段），如果要显示“1”，则要点亮 b、c 两段；显示“A”则点亮 a、b、c、e、f、g 这六段；我们还知道，既然点亮发光二极管要加载的是正向压降，它的两端电压必然会有高低之分：如果公共端作为电压高的一端，我们称之为共阳极数码管（如上图左）；如果公共端作为电压低的一端，则称之为共阴极数码管（上图右）。所以，要点亮共阳极数码管，则要在公共端给予高于非公共端的电平；反之点亮共阴极数码管，则要在公共端给予低于非公共端的电平。

1. 共阴极数码管

其内部 8 个发光二极管的阴极在数码管内部全部连接在一起，所以称“共阴”，而它们的阳极是独立的，通常在设计电路时一般把阴极接地。当我们给数码管的任意一个阳极加一个高电平时，对应的这个发光二极管就点亮了。如果想要显示出一个 8 字，并且把右下角的小数点也点亮的话，可以给 8 个阳极全部送高电平，如果想让它显示出一个 0 字，那么我们可以除了给第“g, dp”这两位送低电平外，其余引脚全部都送高电平，这样它就显示 0 字了。想让它显

示几，就给相对应的发光二极管送高电平，因此我们在显示数字的时候，首先做的就是给0-9十个数字编码，在要它亮什么数字的时候直接把这个编码送到它的阳极就行了。

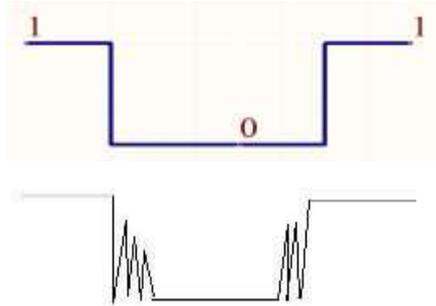
2. 共阳极数码管

其内部8个发光二极管的所有阳极全部连接在一起，电路连接时，公共端接高电平，因此我们要点亮的那个发光管二极管就需要给阴极送低电平，此时显示数字的编码与共阳极编码是相反的关系，数码管内部发光二极管点亮时，也需要5mA以上的电流，而且电流不可过大，否则会烧毁发光二极管。由于单片机的I/O口送不出如此大的电流，所以数码管与单片机连接时需要加驱动电路，可以用上拉电阻的方法或使用专门的数码管驱动芯片，本实验使用上拉电阻的方法实现驱动。

3. 独立按键

按键种类繁多，功能有简有繁，但是无论如何，所有的按键其实都有一个原型，来源于同一种原理，所有的按键无论多复杂，多华丽，都是从这样一个原型发展而成的。平日所见到的绝大部分的按键，其实都可以归类为一种，叫“接触式按键”。

按键常常遇到的问题_抖动问题。



看图2-4-1，按键未按下之前，单片机引脚P1口即按键的右端显示高电平，左端显示低电平，按键按下后，按键闭合，整个通路都显示低电平，然后按键松开，又回到按键按下之前的电平状态。如果只看按键右端的电平变化，应该是上图中所显示的一个负脉冲波形。但是，实际上，正确的波形应该是下图。相比于上图，大家都看到了在高低电平转换时有一段锯齿一样的波形，这就是所谓的按键抖动。为什么会有按键抖动呢，原因很简单，接触式按键是靠机械的接触来实现开关作用的。这种接触方式就注定了它要经历一个“接触不稳定——正在稳定中——彻底稳定”的一种过程。接触式按键就处于一种徘徊在“闭合”与“断开”两者之间的状态。体现在电路中，就是在一小段时间内有非常多的“按下——抬起”动作。而这段抖动的时间，大概是10~20 毫秒，依不同的环境条件而定。

解决这个问题常见的方法有软件去抖动和硬件去抖动。

我们解释一下去抖动：关于按键去抖动的解释，我们在手动按键的时候，由于机械抖动或是其它一些非人为的因素很有可能会造成误识别，一般手动按下一次键然后接着释放，按键两片金属膜接触的时间大约为 50ms 左右，在按下瞬间到稳定的时间为 5-10ms，在松开的瞬间到稳定的时间也为 5-10ms，如果我们在首次检测到键被按下后延时 10ms 左右再去检测，这时如果是干扰信号将不会被检测到，如果确实是有键被按下，则可确认，以上为按键识别去抖动的原理。

硬件准备：8P 杜邦线 2 根。

本实验采用共阳数码管。

四、实验原理图

如图 3-4-1 所示

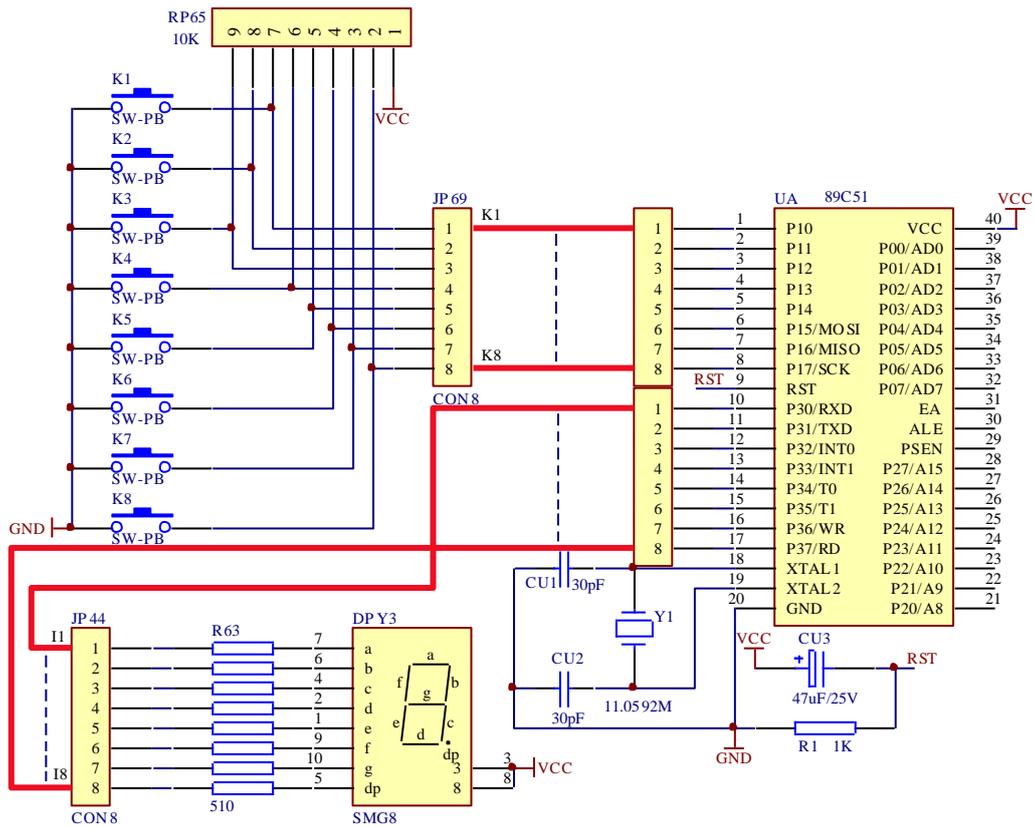


图 3-4-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-4-2 所示

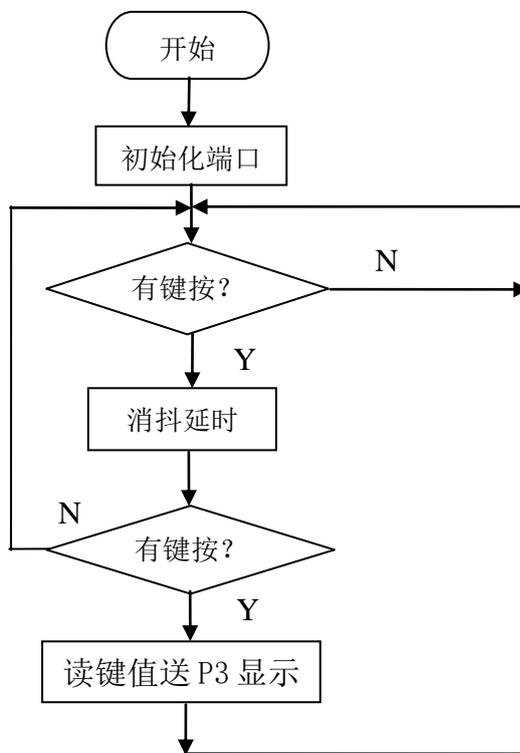


图 3-4-2

六、实验步骤

- 1、使用 1 根 8P 杜邦线连接单片机 P3 口与底板“1 位共阳数码管”区 JP44 排针， 分别为：P30-A， P31-B， P32-C， P33-D， P34-E， P35-F， P36-G， P37-DP；
- 2、使用 1 根 8P 杜邦线连接单片机 P1 口与底板“独立按键区”JP69 排针， 分别为：P10-K1， P11-K2， P12-K3， P13-K4， P14-K5， P15-K6， P16-K7， P17-K8；
- 3、在 KEIL 软件中调试 smg 工程文件。用 STC 软件下载 smg.Hex 程序；
- 4、实验现象：依次按键盘， 会显示该键值。

实验五 矩阵键盘显示

一、实验目的

- 1、掌握单片机 P1 口简单使用。
- 2、掌握矩阵键盘扫描的工作原理。
- 3、掌握数码管扫描显示驱动方法。

二、实验内容

- 1、P1 口做键盘的输入输出口。
- 2、数码管通过 74LS595 驱动位码和段码，通过 P2 口来控制。
- 3、编程，每按一次矩阵键盘上的键，会在数码管上显示相应的键值。

三、实验说明

检测扫描矩阵键盘的按键，然后再数码管上显示该键值。
 硬件准备：1P 杜邦线 3 根，8P 杜邦线 1 根。

四、实验原理图

见图 3-5-1 所示

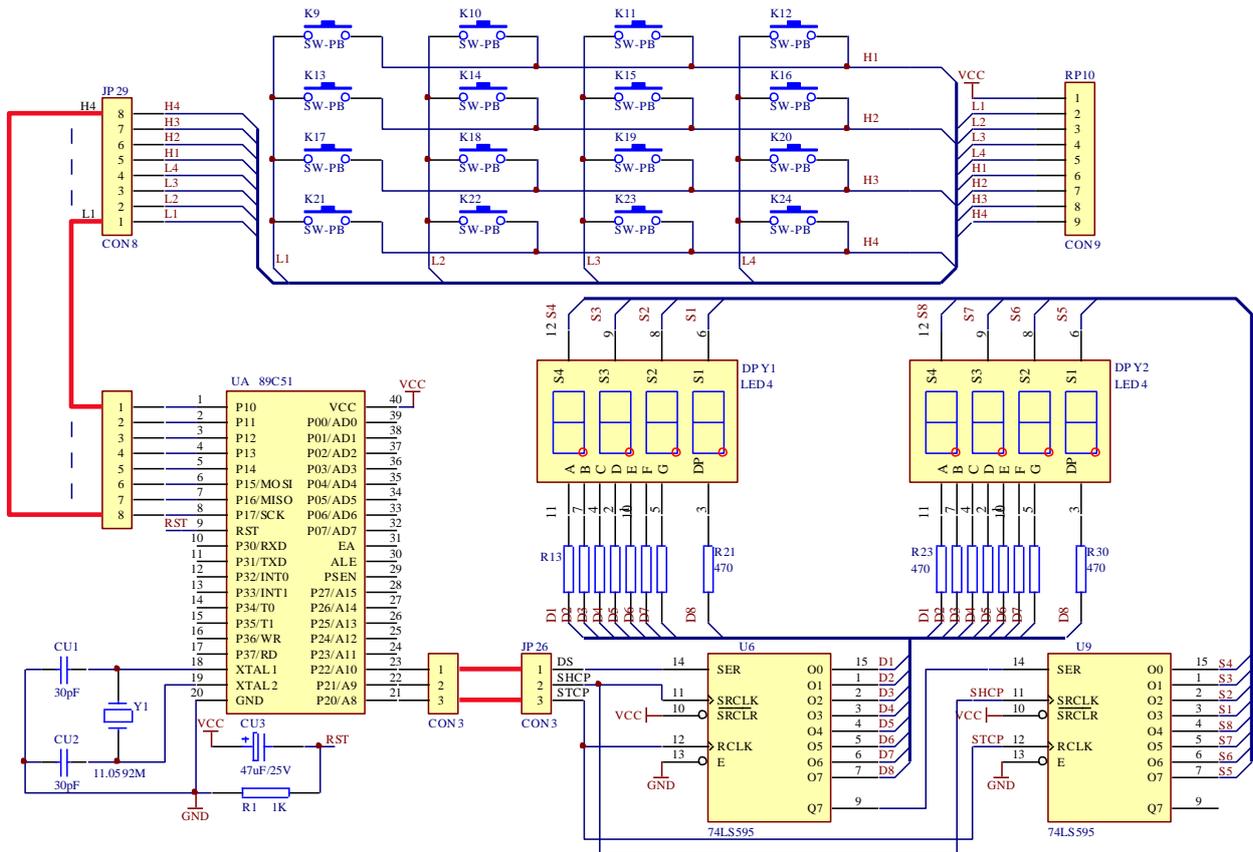
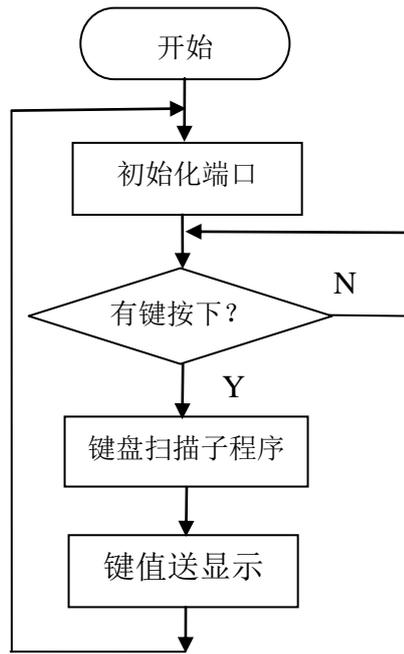


图 3-5-1

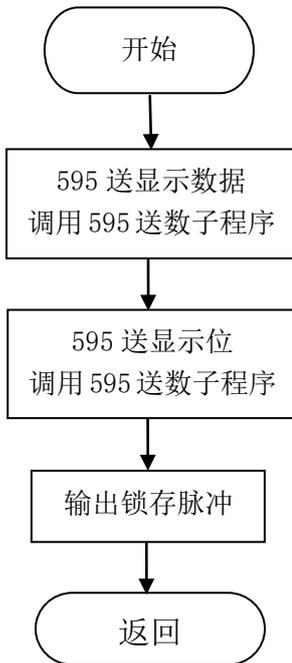
五、实验程序与框图

源程序见随机光盘
 流程图见图 3-5-2 所示

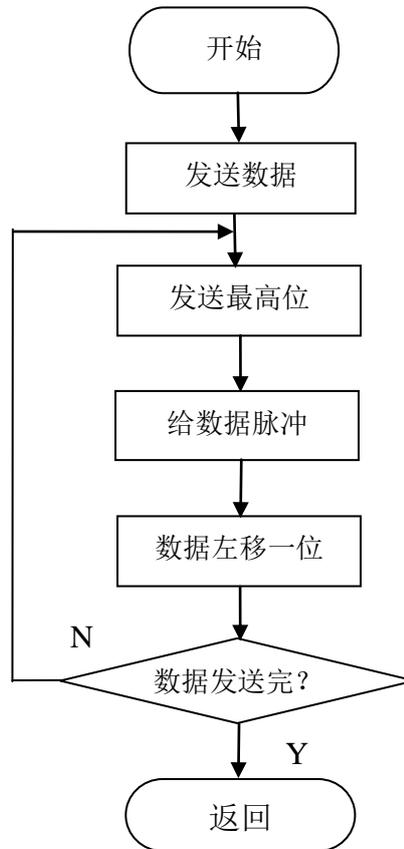
主程序



显示子程序



595 送数据子程序



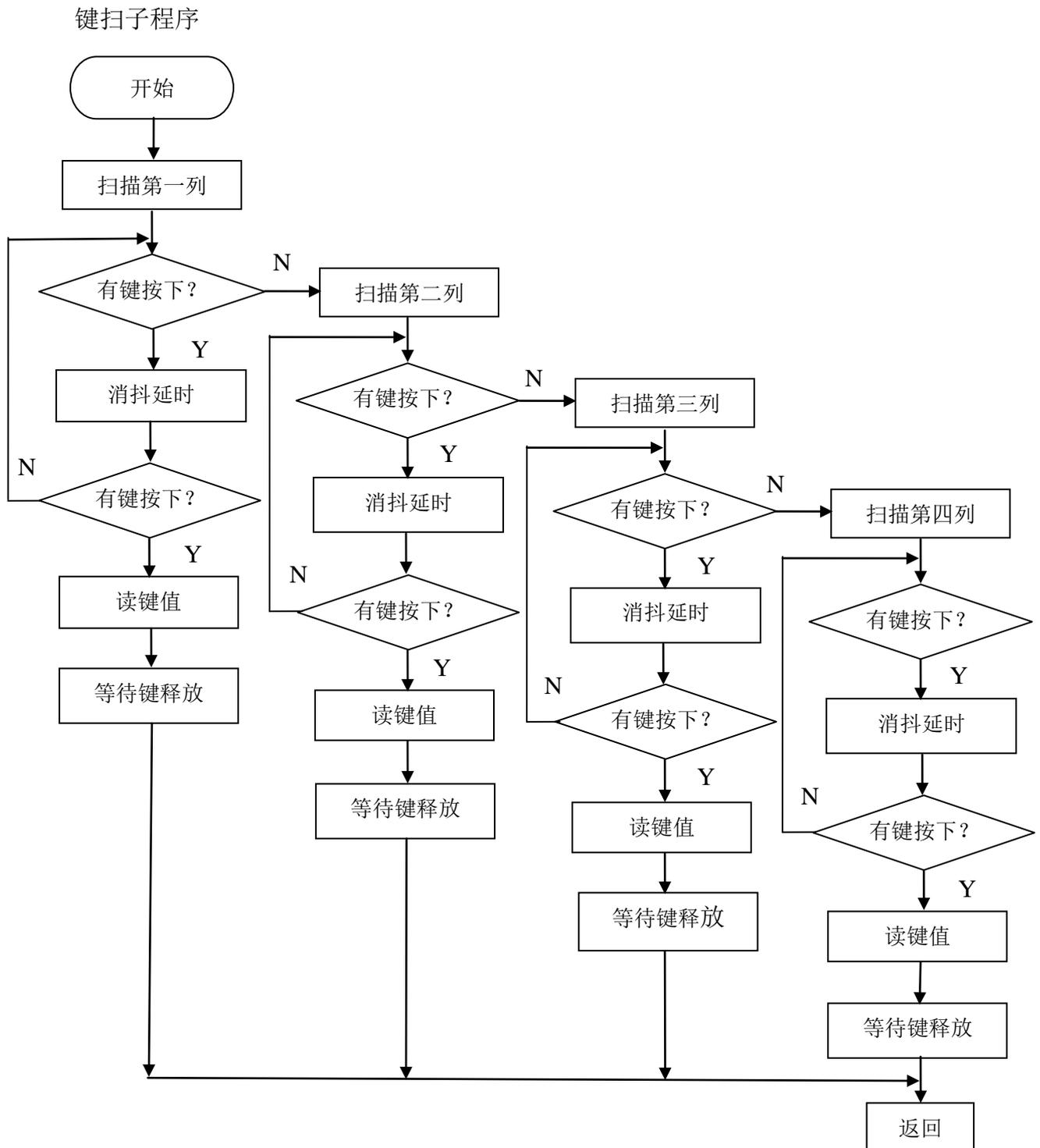


图 3-5-2

六、实验步骤

- 1、使用杜邦线连接单片机 P20~P22 口接底板“八位共阳数码管”区的 JP26，具体接线为：P22-STCP，P21-SHCP，P20-DS。
- 2、使用 1 根 8P 杜邦线连接单片机 P1 口接底板 JP29，具体接线为：P10-L1，P11-L2，P12-L3，P13-L4，P14-H1，P15-H2，P16-H3，P17-H4。
- 3、在 KEIL 软件中调试 JPXS 工程文件。用 STC 软件下载 JPXS.HEX 程序。
- 4、实验现象：每按一次矩阵键盘上的键，会在数码管上显示相应的键值。。

实验六 8X8 红绿双色点阵

一、实验目的

- 1、掌握双色点阵的工作原理。
- 2、掌握 74HC595 芯片的编程。
- 3、掌握字库的生成。

二、实验内容

- 1、编制程序，在点阵上显示不同的字符。

三、实验说明

双色点阵内部结构图：

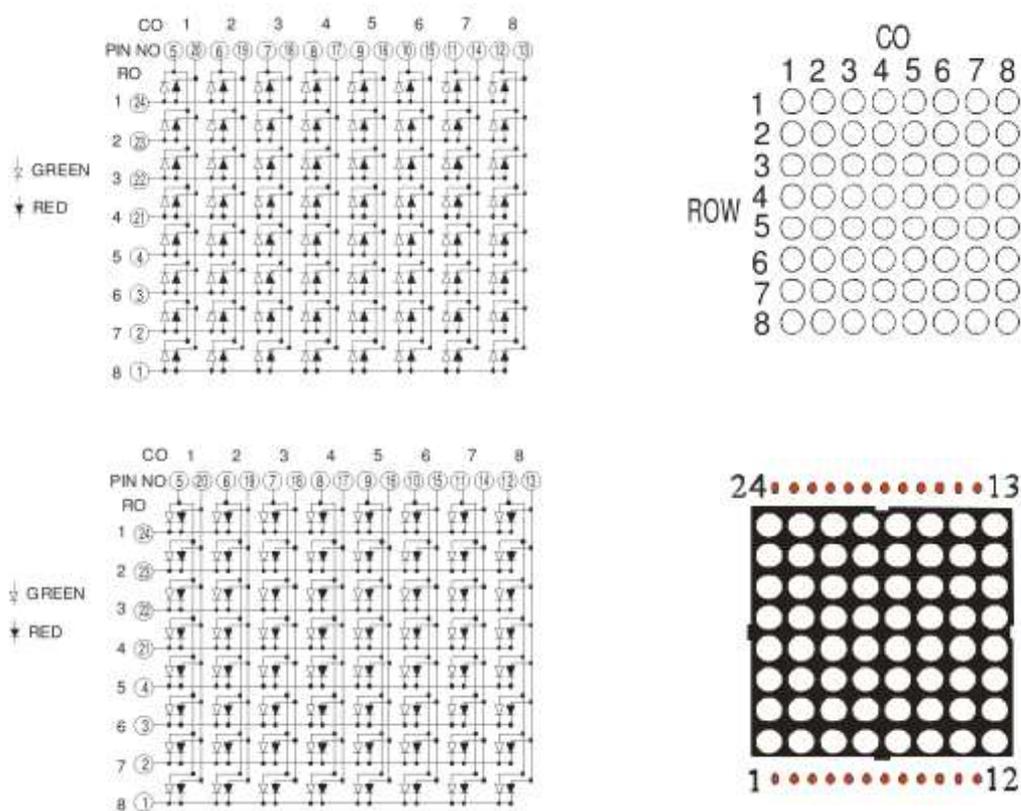


图 3-6-1

点阵 LED 显示器是将许多 LED 类似矩阵一样排列在一起组成的显示器件，当单片机输出的控制信号使得点阵中有些 LED 发光，有些不发光，即可显示出特定的信息，包括汉字、图形等。车站广场由单片机控制的点阵 LED 大屏幕广告宣传牌随处可见。

8×8 点阵 LED 相当于 8×8 个发光二极管（双色 8×8×2）组成的阵列，对于共阳极 LED 来说，其中每一行共用一个阳极（列控制），每一列共用一个阴极（行控制），行控制和列控制满足正确的电平就可以使相应的发光管点亮。点阵模块上的点阵 LED 的管脚及相应的行、列控制位如图 3-6-1 所示。

模块上有一个共阳极 8×8 双色点阵的显示器，其点阵结构如图 3-6-1 中的左上所示。该点阵对外引出 24 条线，其中 8 条行线，16 条列线，若使某一个 LED 发光，只要将与其相连的行线加高电平，对应列线加低电平即可。

实验内容的电路原理图如图 3-6-2 所示，8 位行代码用 74LS595(U12)的输出 00-07 控制。行代码输出的数据通过串行驱动器 74LS595 加至点阵的 8 条行线上，绿色 LED 列码用 74LS595(U13)的输出 00-07 控制，红色 LED 列码用 74LS595(U15)的输出 00-07 控制。

硬件准备：单根杜邦线 6 根。

四、实验原理图

如图 3-6-2 所示

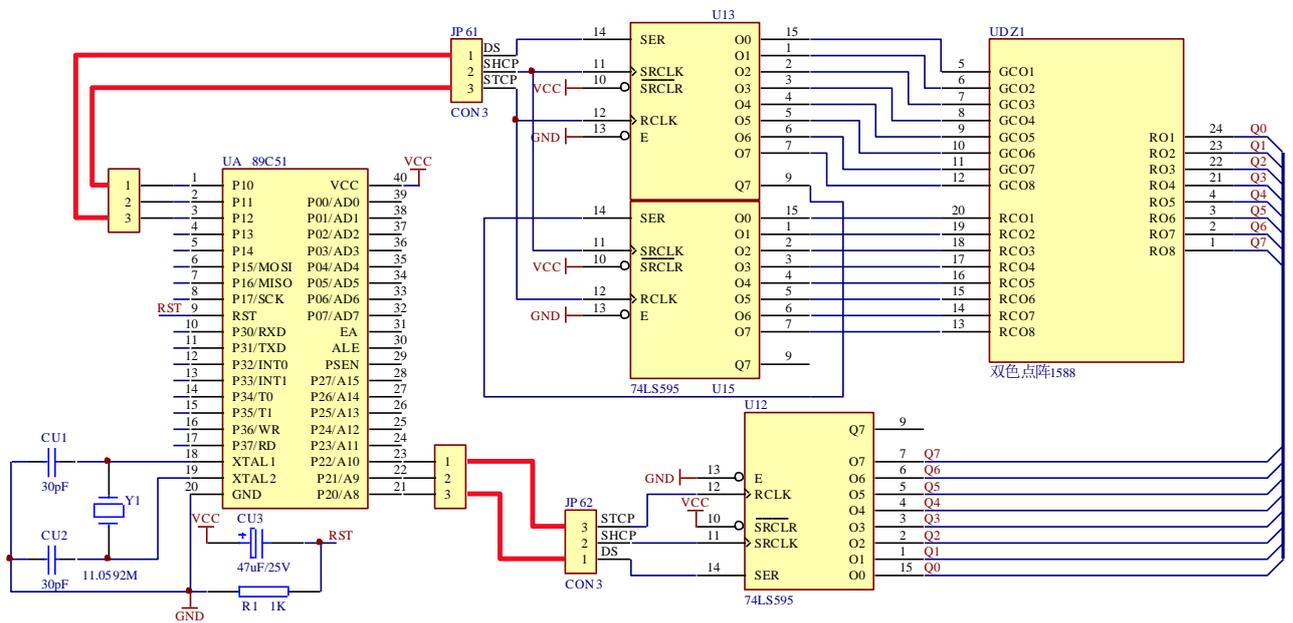


图 3-6-2

五、实验程序与框图

源程序见随机光盘

流程图见图 3-6-3 所示

六、实验步骤

1、底板 JP62 与单片机 P20~P22 相连，JP61 与单片机 P10~P12 相连。

具体接法如下： JP62 共端 DS-P20，SHCP-P21，STCP-P22。

JP61 颜色控制端 DS-P12，SHCP-P11，STCP-P10。

2、在 KEIL 软件中调试 dianz 工程文件。用 STC 软件下载 dianz. HEX 程序。

3、实验现象：点阵首先显示红色“电子”两个汉字，然后流动显示“CEPARK”6 个字符。

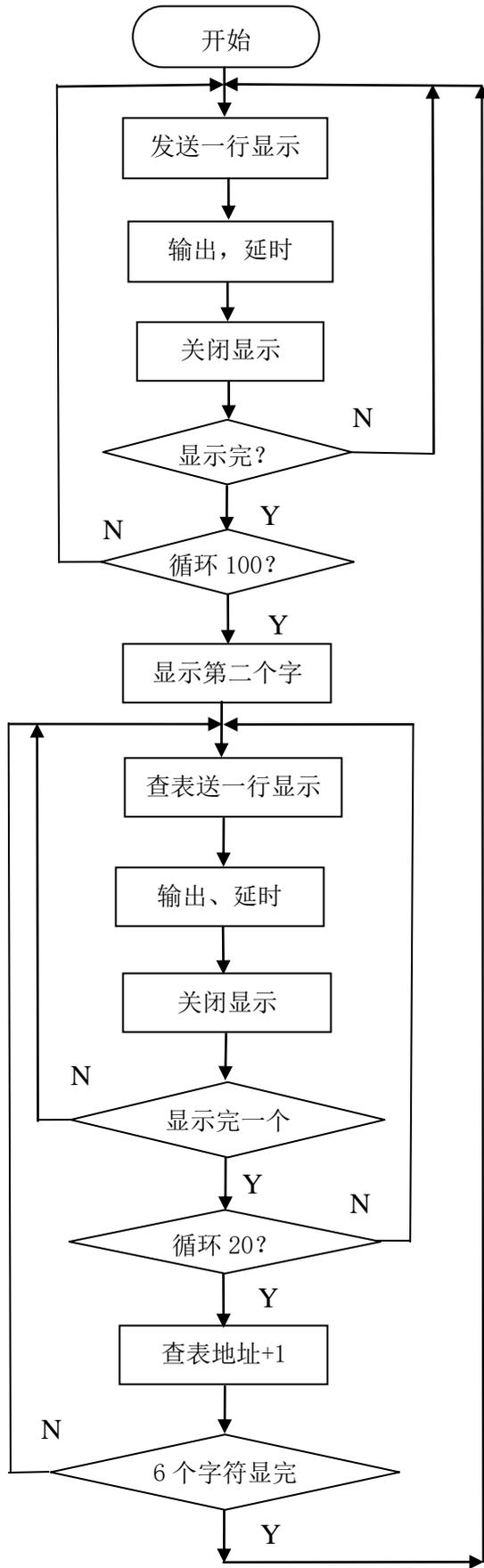


图 3-6-3

实验七 LCD1602 液晶显示

一、实验目的

- 1、了解液晶显示模块的工作原理
- 2、掌握液晶显示屏的驱动
- 3、了解常用符号的 ASCII 编码

二、实验内容

- 1、编写液晶屏的驱动程序，在液晶屏上显示英文字母、数字和标点符号。

三、实验说明

1、1602 概述

如下图所示，液晶屏 1602 是一种非常易用的字符型 LCD，可显示 ASCII 码对应的内容，如大小写英文字母，0-9 的数字，标点符号等。大写字母 A~Z 对应的 ASCII 编码为 41H~5AH，小写字母 a~z 对应的 ASCII 码的编码为 61~7AH，数字 0~9 对应的编码为 30H~39H。

1602 可显示 2 行的字符，每一行显示 16 个字符，整个屏幕可显示 32 个字符。



2、1602 显示屏地址

如下表所示，第一行为 00H-0FH，第二行为 40H-4FH。要在液晶屏上显示字符或数字，必须先给出要显示的地址，再给数据。

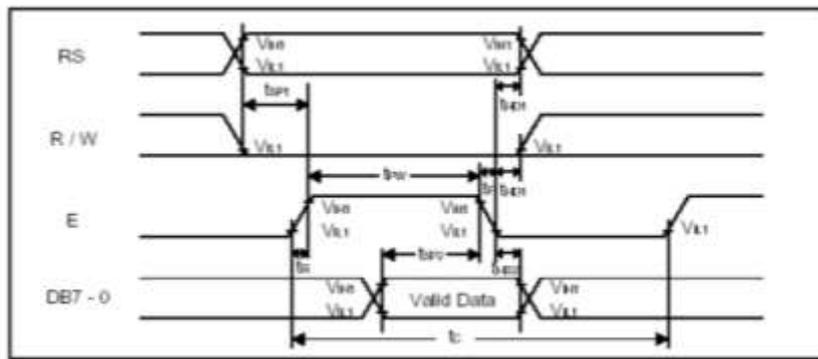
3、引脚功能

引脚编号	引脚符号	功能规格
1	VSS(输入)	接地，0V
2	Vdd(输入)	接电源，5V±5%
3	VO(输入)	反视度调整，使用可变电阻调节，可改变对比度
4	RS(输入)	寄存器选择。1：选择资料寄存器，0：选择指令寄存器
5	$\overline{R/W}$ (输入)	读写选择，1：读；0：写
6	E(输入)	使能选择。1：LCD 可读写。0：LCD 不可读写
7	DB0	数据端口的第 0 位
8	DB1	数据端口的第 1 位
9	DB2	数据端口的第 2 位
10	DB3	数据端口的第 3 位
11	DB4	数据端口的第 4 位
12	DB5	数据端口的第 5 位
13	DB6	数据端口的第 6 位

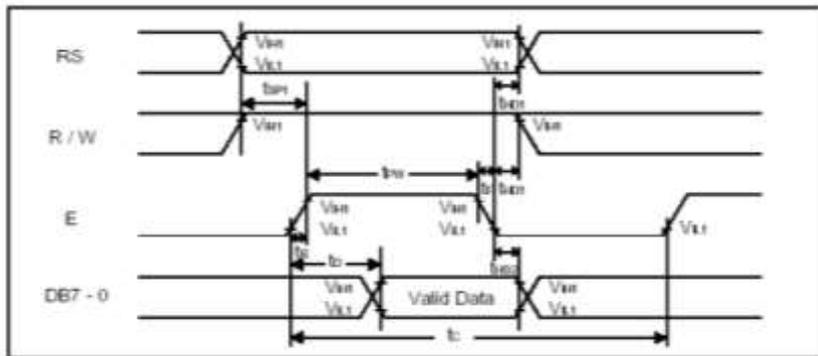
14	DB7	数据端口的第 7 位
15	Vdd(输入)	背光电源正极, 5V ± 5%
16	VSS(输入)	背光电源负极, 0V

4、读写时序图

(1) 写时序



(2) 读时序



5、指令表

1602 液晶模块内部的控制器共有 11 条控制指令，如下表所示。

	指令	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0
1	清屏	0	0	0	0	0	0	0	0	0	1
2	光标返回	0	0	0	0	0	0	0	0	1	*
3	输入模式	0	0	0	0	0	0	0	1	I/D	S
4	显示控制	0	0	0	0	0	0	1	D	C	B
5	光标/字符移位	0	0	0	0	0	1	S/C	R/L	*	*
6	功能	0	0	0	0	1	DL	N	F	*	*
7	置字符发生器地址	0	0	0	1	字符发生存贮器地址					
8	置数据存贮器地址	0	0	1	显示数据存贮器地址						
9	读忙标志和地址	0	1	BF	计数器地址						
10	写数据到指令 7.8 所设地址	1	0	要写的数据							
11	从指令 7.8 所设的地址读	1	1	读出的数据							

	数据			
--	----	--	--	--

指令 1: 清显示, 光标复位到地址 00H 位置。

指令 2: 光标复位, 光标返回到地址 00H。

指令 3: 光标和显示模式设置 I/D: 光标移动方向, 高电平右移, 低电平左移, S: 屏幕上所有文字是否左移或者右移。高电平表示有效, 低电平则无效。

指令 4: 显示开关控制。D: 控制整体显示的开与关, 高电平表示开显示, 低电平表示关显示 C: 控制光标的开与关, 高电平表示有光标, 低电平表示无光标 B: 控制光标是否闪烁, 高电平闪烁, 低电平不闪烁。

指令 5: 光标或显示移位 S/C: 高电平时移动显示的文字, 低电平时移动光标。R/L, 高向左, 低向右。

指令 6: 功能设置命令 DL: 高电平时为 4 位总线, 低电平时为 8 位总线 N: 低电平时为单行显示, 高电平时为双行显示 F: 低电平时显示 5x7 的点阵字符, 高电平时显示 5x10 的点阵字符。(有些模块是 DL: 高电平时为 8 位总线, 低电平时为 4 位总线)

指令 7: 字符发生器 RAM 地址设置, 地址: 字符地址*8+字符行数。(将一个字符分成 5*8 点阵, 一次写入一行, 8 行就组成一个字符)

指令 8: 置显示地址, 第一行为: 00H—0FH, 第二行为: 40H—4FH。

指令 9: 读忙信号和光标地址 BF: 为忙标志位, 高电平表示忙, 此时模块不能接收命令或者数据, 如果为低电平表示不忙。

指令 10: 写数据。

指令 11: 读数据。

6、液晶屏的操作

液晶屏的软件编程控制操作主要包含初始化, 写指令和写数据三个部分。凡是写到液晶屏内部, 用来控制液晶屏显示的内容都属于指令, 写入到液晶屏后能直接显示出来的结果就属于数据。

7、请详细查看相关资料

硬件准备: 1P 杜邦线 3 根, 8P 杜邦线 1 根

四、实验原理图

如图 3-7-1 所示

五、实验程序与框图

源程序见随机光盘

流程图见图 3-7-2 所示

六、实验步骤

1、用 1 根 8P 杜邦线将单片机 P1 口与底板“LCD1602 液晶”区 JP33 排针相连, 分别为: P10-DB0, P11-DB1, P12-DB2, P13-DB3, P14-DB4, P15-DB5, P16-DB6, P17-DB7;

2、用 1 根 3P 杜邦线将单片机 P35~P37 口与底板“LCD1602 液晶”JP31 相连。分别为: P35-RS, P36-RW, P37-EN;

3、在 KEIL 软件中调试 LCD1602 工程文件。用 STC 软件下载 lcd1602.HEX 程序。

4、实验现象: 在液晶上显示两行字符

第一行显示“1A2B3C4D5E6F7G8H”

第二行显示“GOOD LUCK！”

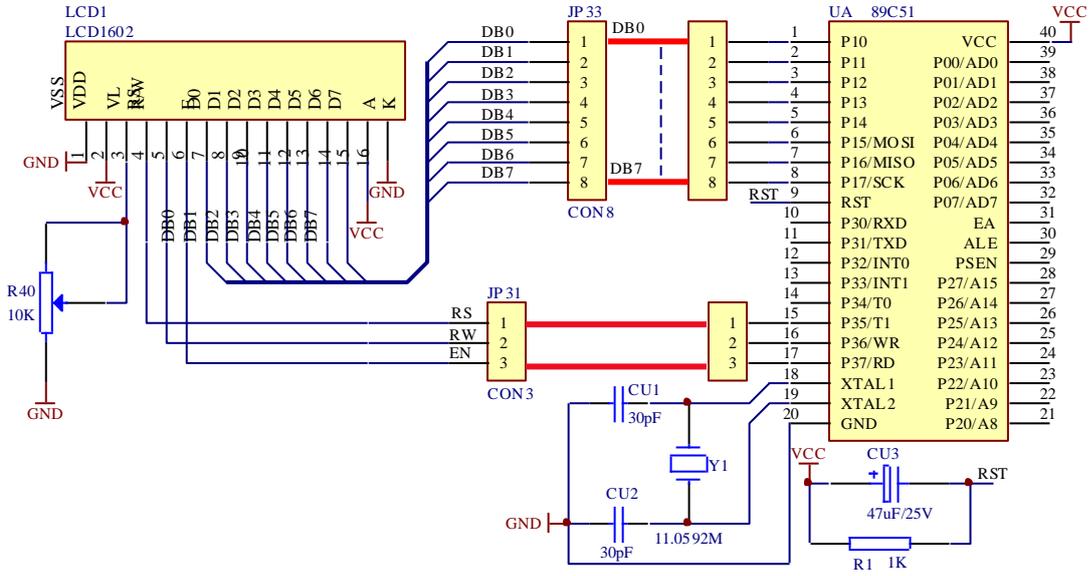


图 3-7-1

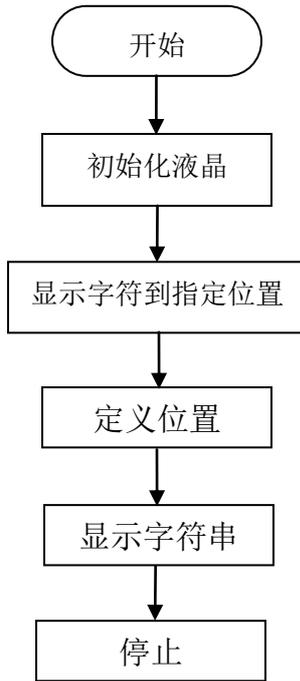


图 3-7-2

实验八 LCD12864 液晶显示

一、实验目的

- 1、了解字符 LCD 模块的使用方法
- 2、掌握 8051 单片机控制字符 LCD 模块显示程序的设计方法。

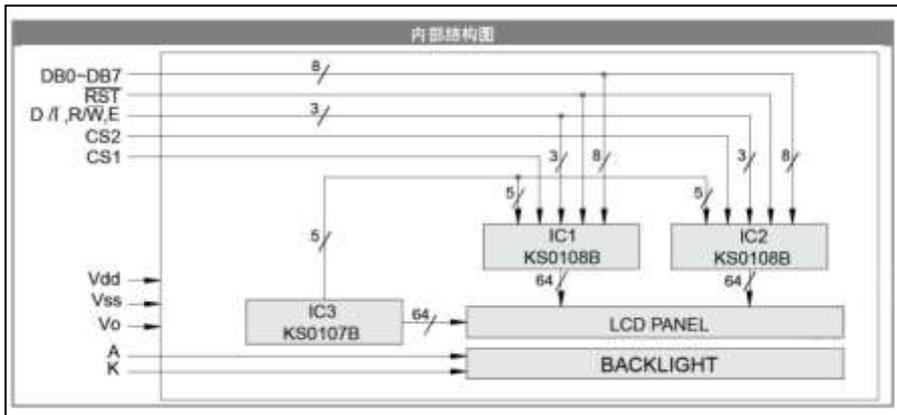
二、实验内容

利用 LCD 循环显示图形和汉字。

三、实验说明

字符图形 LCD 模块是一种专用显示字符、数字或符号的液晶显示模块。模块本身附有显示驱动控制电路，可以与单片机的 I/O 口线直接连接，使用方便。实验系统采用的 128×64 液晶为 KS0108 驱动控制器，详细见液晶显示相关资料。

1、内部结构图



2、引脚说明

引脚说明			
标号	名称	参数	说明
1	Vdd	+5V	模块电源 +5V DC
2	Vss	GND 0V	模块电源 地
3	Vo	0V ~ -10V	外部液晶偏置电压（负压）输入端
4	DB0	H/L	数据线 0 （H 表示高电平，L 表示低电平）
5	DB1	H/L	数据线 1
6	DB2	H/L	数据线 2
7	DB3	H/L	数据线 3
8	DB4	H/L	数据线 4
9	DB5	H/L	数据线 5
10	DB6	H/L	数据线 6
11	DB7	H/L	数据线 7
12	CS1	H/L	左半屏使能 （低电平有效）
13	CS2	H/L	右半屏使能 （低电平有效）
14	RST	H/L	LCM 复位 （低电平有效）
15	R/W	H/L	H: MPU→LCM 读数据； L: MPU→LCM 写数据；
16	D/I	H/L	H: 数据； L: 指令；
17	E	H/L	LCM 使能
18	Vss	GND 0V	模块电源 地
19	LED-A	+5V	LED 背光电源 +5V DC
20	LED-K	GND 0V	LED 背光电源 地

3、指令表

指令	数据位										功能描述
	D/I	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
显示开关	0	0	0	0	1	1	1	1	1	P	P=1: 开显示 P=0: 关显示
设置列 (Y 地址)	0	0	0	1	A5	A4	A3	A2	A1	A0	列地址设置 (0~63)
设置页码 (X 地址)	0	0	1	0	1	1	1	A2	A1	A0	页码设置 (0~7)
设置起始 (Z 地址)	0	0	1	1	A5	A4	A3	A2	A1	A0	显示起始地址设置 (0~63)
读状态	0	1	B	0	P	R	0	0	0	0	B=1: LCM 忙 B=0: LCM 空闲 P=1: 开显示 P=0: 关显示 R=1: 复位 R=0: 正常
写数据	1	0	A7	A6	A5	A4	A3	A2	A1	A0	写 8 位显示数据
读数据	1	1	A7	A6	A5	A4	A3	A2	A1	A0	读 8 位显示数据

■ BF 读忙 可以使用读忙函数或根据工作耗时使用相应延时程序

硬件准备：1P 杜邦线 5 根，8P 杜邦线 1 根

四、实验原理图

如图 3-8-1 所示

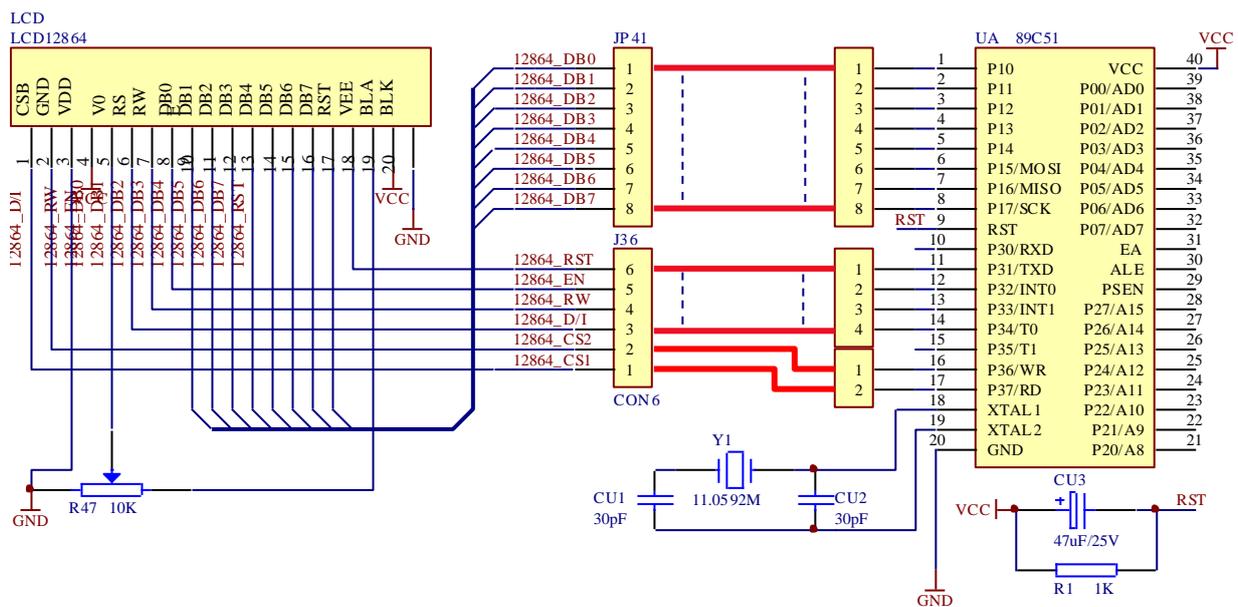


图 3-8-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-8-2 所示

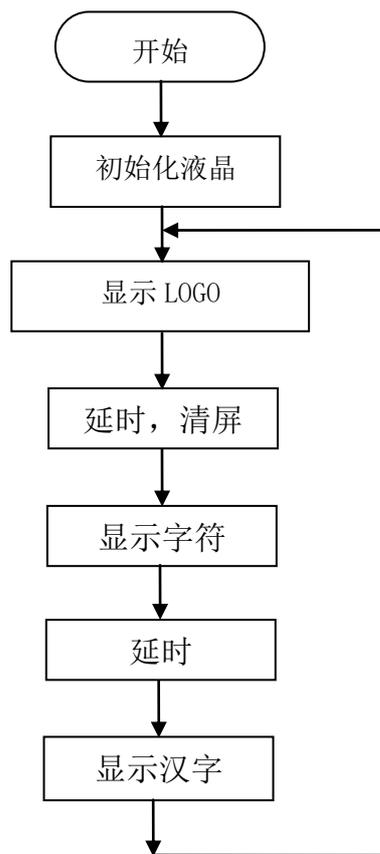


图 3-8-2

六、实验步骤

- 1、用 1 根 8P 杜邦线将单片机 P1 口与底板“LCD12864 液晶”区 JP41 排针相连，分别为：P10-DB0，P11-DB1，P12-DB2，P13-DB3，P14-DB4，P15-DB5，P16-DB6，P17-DB7；
- 2、用杜邦线将单片机 P3 口与底板“LCD12864 液晶”区 JP36 相连。分别为：**P31-RST(STC 下载时先不插，下载完成后插上)**，P32-EN，P33-RW，P34-D/I，P36-CS2，P37-CS1；
- 3、在 KEIL 软件中调试 LCD12864 工程文件。用 STC 软件下载 LCD12864.HEX 程序；
- 4、实验现象：在液晶上显示图形和文字。

实验九 OLED 模块显示

一、实验目的

- 1、了解字符 OLED 模块的使用方法；
- 2、掌握 8051 单片机控制字符 OLED 模块显示程序的设计方法；
- 3、掌握 OLED 取字模的方法。

二、实验内容

利用 OLED 屏显示汉字。

三、实验说明

1、OLED 简介

OLED，即有机发光二极管（Organic Light Emitting Diode）。OLED 由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制造较简单等特性，被认为是下一代的平面显示器新兴应用技术。

LCD 都需要背光，而 OLED 不需要，因为它是自发光的。这样同样的显示，OLED 效果要来得好一些。以目前的技术，OLED 的尺寸还难以大型化，但是分辨率可以做到很高。在此我们使用的是 0.96 寸 OLED 显示屏，该屏有以下特点：

1) 0.96 寸 OLED 有黄蓝、白、蓝三种颜色可选；其中黄蓝是屏上 1/4 部分为黄光，下 3/4 为蓝；而且是固定区域显示固定颜色，颜色和显示区域均不能修改；白光则为纯白，也就是黑底白字；蓝色则为纯蓝，也就是黑底蓝字；

2) 分辨率为 128×64；

3) 多种接口方式：OLED 裸屏总共有两种接口，包括：6800、8080 两种并行接口方式、3 线或 4 线的串行 SPI 接口方式、IIC 接口方式，这五种接口是通过屏上的 BS0~BS2 来配置的。

我们用的 0.96 寸 OLED 屏是 SPI 接口方式。

SPI/IIC 接口模块接口定义：

GND： 电源地

VCC： 电源正（3~5.5V）

D0： OLED 的 D0 脚，在 SPI 和 IIC 通信中为时钟管脚

D1： OLED 的 D1 脚，在 SPI 和 IIC 通信中为数据管脚

RES： OLED 的 RES#脚，用来复位（低电平复位）

DC： OLED 的 D/C#E 脚，数据和命令控制管脚

硬件准备：1P 杜邦线 5 根，

四、实验原理图

如图 3-9-1

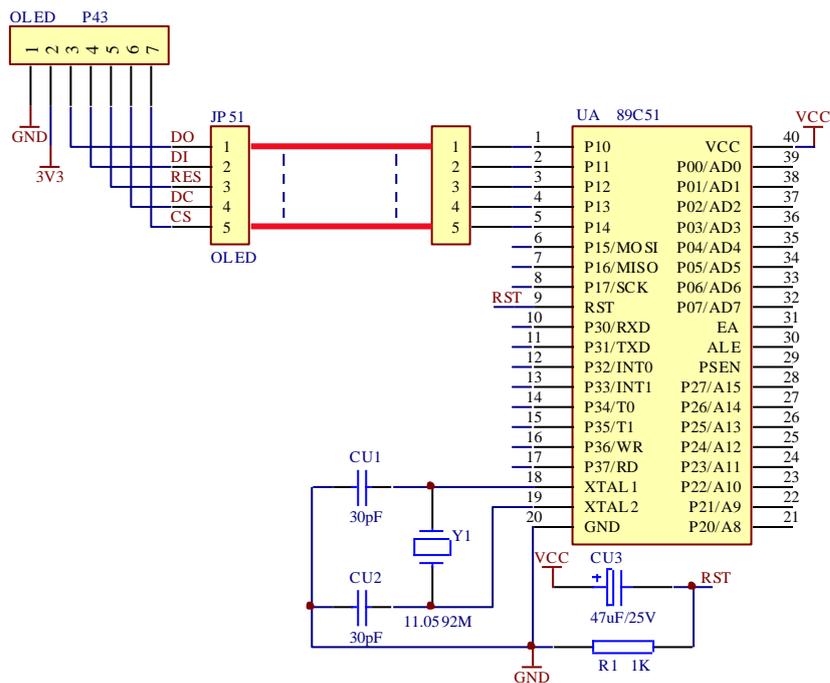


图 3-9-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-9-2 所示

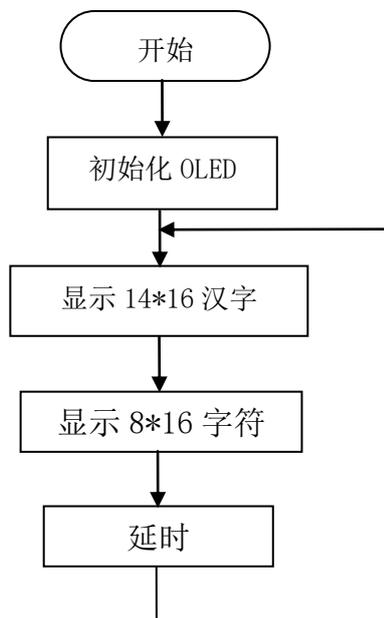


图 3-9-2

六、实验步骤

- 1、用杜邦线将单片机 P10~P14 口与底板“OLED 液晶”区 JP51 相连，分别为：P10-DO，P11-D1，P12-RES，P13-DC，P14-CS；
- 2、在 KEIL 软件中调试 OLED 工程文件。用 STC 软件下载 OLED.HEX 程序；
- 3、实验现象：在 OLED 屏上显示文字。

实验十 TFT 彩屏模块测试

一、实验目的

- 1、 3.2' TFT 彩色液晶屏的控制。
- 2、 常用线阻式触摸屏的应用及控制芯片 XPT2046 的编程控制。

二、实验内容

- 1、 在 TFT 屏上显示字符和图片。
- 2、 触摸屏触摸画点并显示座标。

三、实验说明

3.2 寸 TFT 屏接口模式，详细见相关资料。

序号	名称	说明
1	GND	电源地
2	VCC	电源正。
3	NC	空
4	RS	数据/命令切换
5	WR	写数据时钟
6	RD	读数据时钟
7-14	DB8-DB15	高 8 位数据总线
15	CS	片选
16	F_CS	FLASH 片选（模块预留 FLASH 芯片 W25X16, 默认不贴件，做预留用途）
17	REST	复位（复位操作过程：拉低电平，持续 8ms；再拉高电平，持续 8ms）
18	NC	空
19	LED_A	背光电源（模块已串联 3.9 欧姆限流电阻，LED_A 脚可接 3.3V，如果接 5V 需修改限流电阻为 25 欧姆，否则会烧坏背光）
20	NC	空
21-28	DB0-DB7	低 8 位数据总线
29	T_CLK	触摸控制器(XPT2046)时钟
30	T_CS	触摸控制器(XPT2046)片选
31	T_DIN	触摸控制器(XPT2046)的数据入（即 MOSI）
32	NC	空
33	T_DO	触摸控制器(XPT2046)的数据出（即 MISO）
34	T_IRQ	触摸控制器(XPT2046)数据中断（平时为高，触摸时为低）
35	SD_DO	SD 卡接口的 MISO
36	SD_CLK	SD 卡接口时钟
37	SD_DIN	SD 卡接口的 MOSI
38	SD_CS	SD 卡接口片选
39	NC	空
40	NC	空

硬件准备：8P 杜邦线 2 根， 1P 杜邦线 10 根。

四、实验原理图

如图 3-10-1 所示

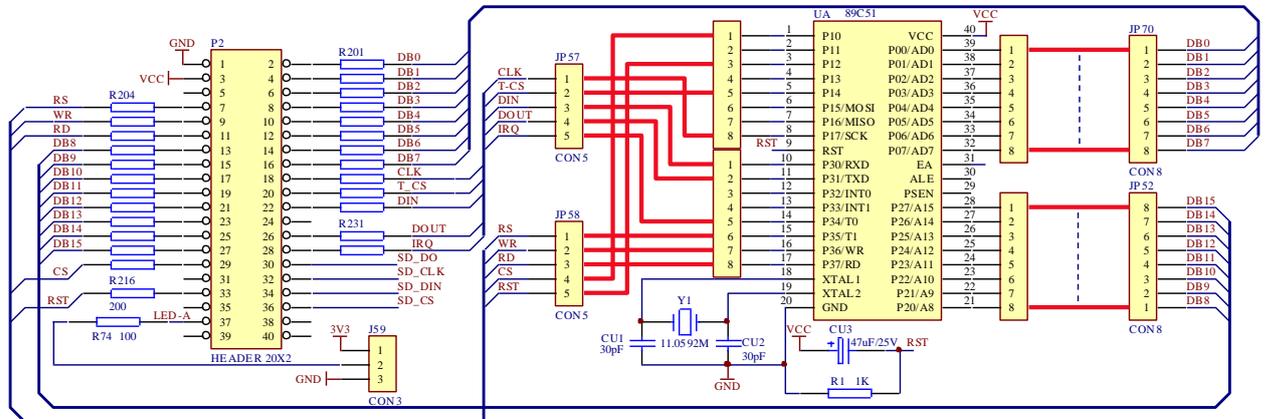


图 3-10-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-10-2 所示

六、实验步骤

- 1、用 1 根 8P 杜邦线将单片机 P0 口与底板“TFT 彩屏”区 JP70 排针相连，分别为：P00-DB0，P01-DB1，P02-DB2，P03-DB3，P04-DB4，P05-DB5，P06-DB6，P07-DB7，
- 2、用 1 根 8P 杜邦线将 P2 与“TFT 彩屏”区 JP52 排针相连，分别是 P20-DB8，P21-DB9，P22-DB10，P23-DB11，P24-DB12，P25-DB13，P26-DB14，P27-DB15。
- 3、用 5 根 1P 杜邦线将单片机口与底板 JP58 控制端相连。分别为：P35-RS，P36-WE，P37-RD，P10-CS，P12-RST；
- 4、用 5 根 1P 杜邦线将单片机口与底板 JP57 控制端相连。分别为：P17-CLK，P14-CS，P30-DIN，P31-DOUT，P34-IRQ；
- 5、“TFT 彩屏”区短接跳线 J59 的背光与 3V3 两个引脚；
- 6、在 KEIL 软件中调试 ceshi 工程文件。用 STC 软件下载 ceshi.HEX 程序；
- 7、实验现象：在 3.2' TFT 彩色液晶屏显示字符和图形，接触摸屏画点、画线。

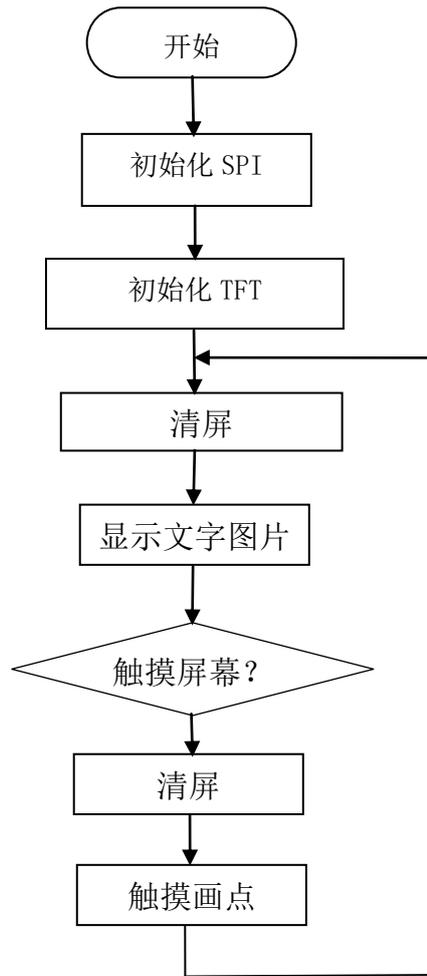


图 3-10-2

实验十一 五向摇杆按键测试

一、实验目的

- 1、了解掌握五向摇杆的工作原理。
- 2、进一步掌握点阵的编程和使用。

二、实验内容

- 1、编写拨动五向摇杆，在点阵屏上显示对应的箭头。

三、实验说明

五向开关中央有一个固定的定点，也称之为公用触点，可以通过这个触点来完成断电与通电功能。五向开关大致是一个正方形，周边被一层金属外壳包裹着并对应着多个固定的触点，分别设置了多个固定触点金属弹片。五向开关不可以独立存在完成工作，它需要配合 PCB 线路板或其它来完成工作。

硬件准备：1P 杜邦线 5 根， 3P 杜邦线 2 根。

四、实验原理图

如图所示 3-11-1

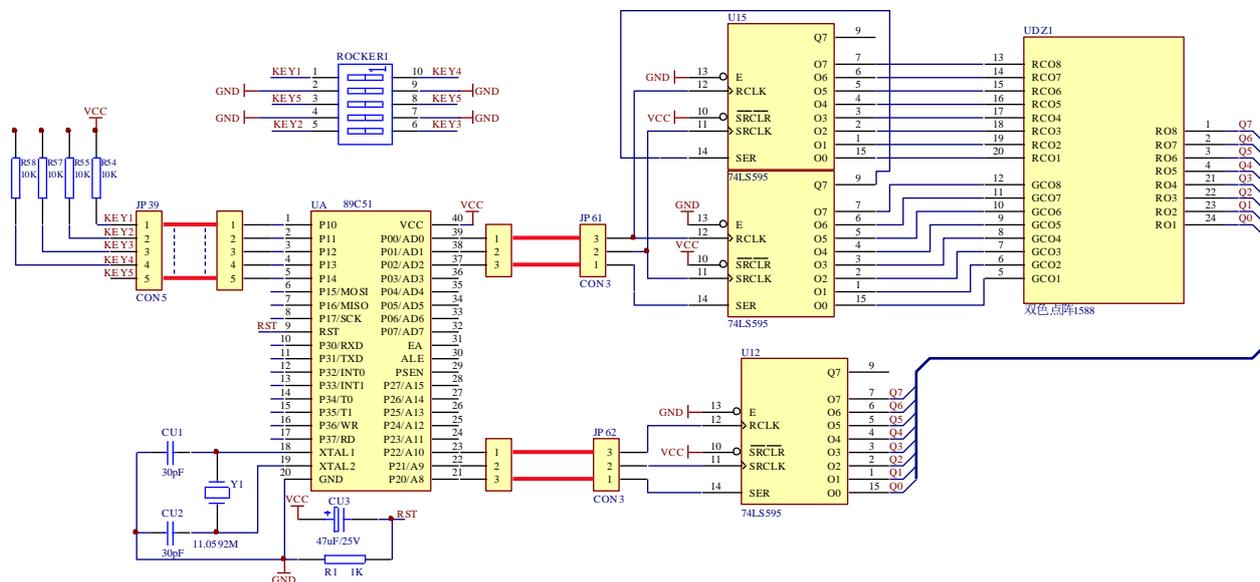


图 3-11-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-11-2 所示

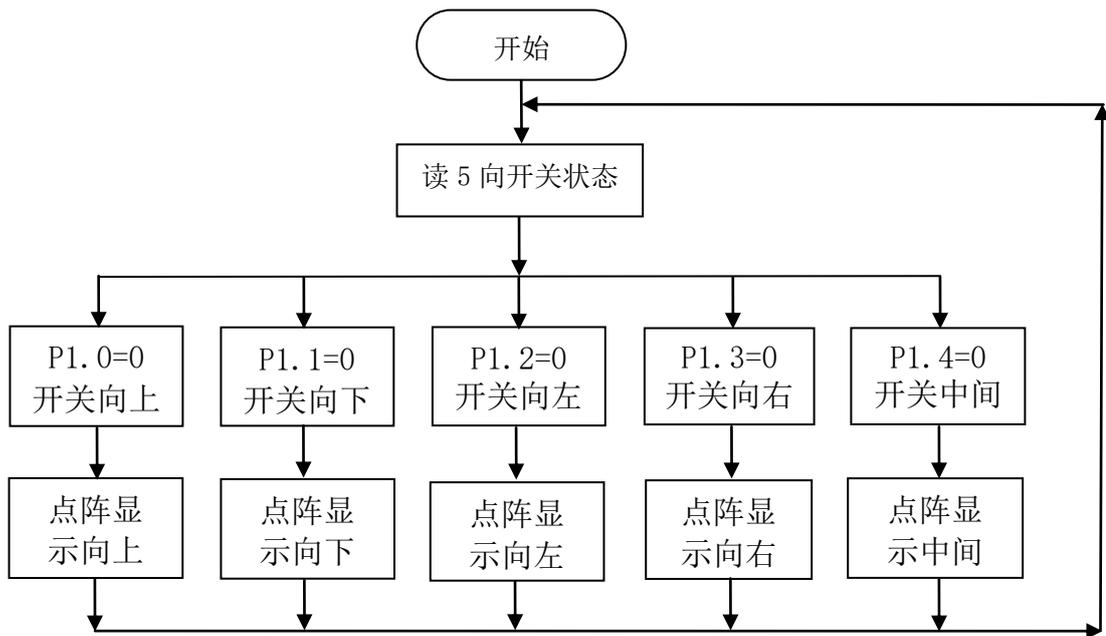


图 3-11-2

六、实验步骤

- 1、使用 1 根 5P 杜邦线连接单片机 P1 口接底板“五相摇杆”区 JP39 分别为：P10-F1，P11-F2，P12-F3，P13-F4，P14-F5；
- 2、用 1 根 3P 杜邦线将底板“双色点阵区” JP62 与单片机 P20~P22 相连，具体接法如下：公共端 DS-P20，SHCP-P21，STCP-P22；
- 3、用 1 根 3P 杜邦线将底板“双色点阵区” JP61 与单片机 P00~P02 相连。具体接法如下：颜色控制端 DS-P02，SHCP-P01，STCP-P00；
- 4、在 KEIL 软件中调试 wxyg 工程文件。用 STC 软件下载 wxyg.HEX 程序；
- 5、实验现象：拨动五向摇杆，点阵显示箭头位置。

实验十二 DS18B20 温度显示

一、实验目的

- 1、了解 DS18B20 的工作原理及使用方法；
- 2、掌握 DS18B20 读写时序的编程方法；
- 3、锻炼单片机综合应用和开发的能力。

二、实验内容

- 1、显示当前温度：在数码管上显示当前 DS18B20 采集到的温度值。

三、实验说明

1、DS18B20 是 DALLAS 公司生产的 1-Wire 数字温度传感器，它是单总线器件，具有线路简单，体积小的特点。因此用它来组成一个测温系统，具有线路简单，只要一根通信线，可以挂很多这样的数字温度传感器，十分方便。

2、DS18B20 产品的特点

- (1) 只要求一个端口即可实现通信。
- (2) 在 DS18B20 中的每个器件上都有独一无二的序列号。
- (3) 实际应用中不需要外部任何元器件即可实现测温。
- (4) 测量温度范围在 -55°C 到 $+125^{\circ}\text{C}$ 之间。
- (5) 数字温度计的分辨率用户可以从 9 位到 12 位选择。
- (6) 内部有温度上、下限告警设置。

3、DS18B20 的引脚介绍

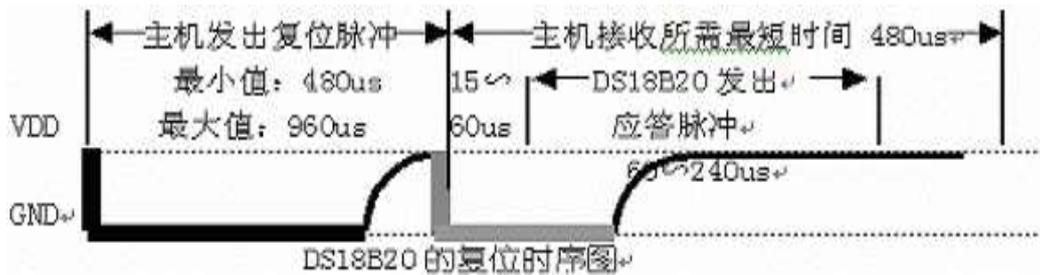
序号	名称	引脚功能描述
1	GND	地信号
2	DQ	数据输入/输出引脚。开漏单总线接口引脚。当被用着在寄生电源下，也可以向器件提供电源。
3	VDD	可选择的 VDD 引脚。当工作于寄生电源时，此引脚必须接地。

4、DS18B20 的使用方法

由于 DS18B20 采用的是 1-Wire 总线协议方式，即用一根数据线实现数据的双向传输，而对 AT89S51 单片机来说，硬件上并不支持单总线协议，因此，我们必须采用软件的方法来模拟单总线的协议时序来完成对 DS18B20 芯片的访问。

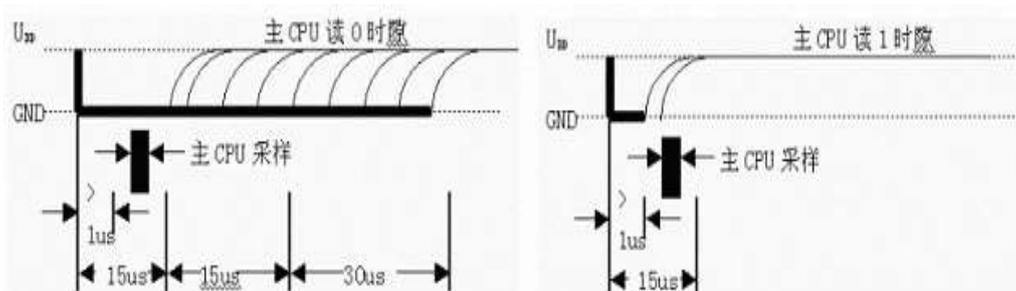
由于 DS18B20 是在一根 I/O 线上读写数据，因此，对读写的数据位有着严格的时序要求。DS18B20 有严格的通信协议来保证各位数据传输的正确性和完整性。该协议定义了几种信号的时序：初始化时序、读时序、写时序。所有时序都是将主机作为主设备，单总线器件作为从设备，而每一次命令和数据的传输都是从主机主动启动写时序开始，如果要求单总线器件回送数据，在进行写命令后，主机需启动读时序完成数据接收。数据和命令的传输都是低位在先。

1) DS18B20 的复位时序如下：



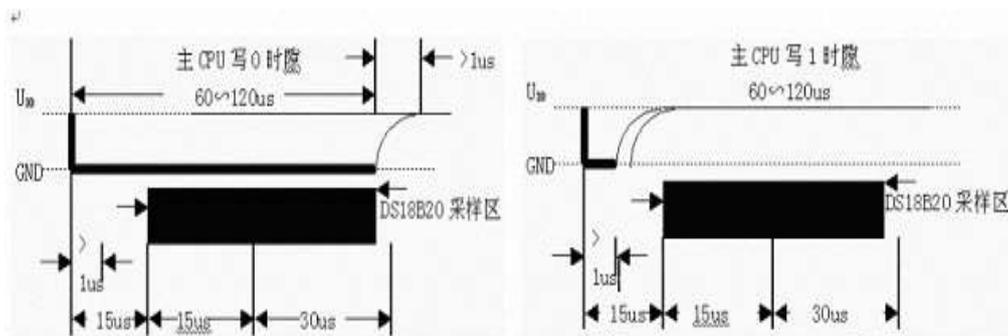
2) DS18B20 的读时序

对于 DS18B20 的读时序分为读 0 时序和读 1 时序两个过程。对于 DS18B20 的读时隙是从主机把单总线拉低之后，在 15 秒之内就得释放单总线，以让 DS18B20 把数据传输到单总线上。DS18B20 在完成一个读时序过程，至少需要 60us 才能完成。



3) DS18B20 的写时序

对于 DS18B20 的写时序仍然分为写 0 时序和写 1 时序两个过程。对于 DS18B20 写 0 时序和写 1 时序的要求不同，当要写 0 时序时，单总线要被拉低至少 60us，保证 DS18B20 能够在 15us 到 45us 之间能够正确地采样 IO 总线上的“0”电平，当要写 1 时序时，单总线被拉低之后，在 15us 之内就得释放单总线。



5、工作原理：本实验通过 DS18B20 采集环境温度，当单片机检测到 DS1820 的存在便可以发出 ROM 操作命令之一，Read ROM(读 ROM)、Match ROM(匹配 ROM)、Skip ROM(跳过 ROM)、Search ROM(搜索 ROM)、Alarm search(告警搜索)，然后对发存储器操作命令，对 DS18B20 进行读写数据转换等操作。单片机使用时间隙(time slots)来读写 DS1820 的数据位和写命令字的位，然后将读到的数据转换 BCD 码在数码管显示出来。

硬件准备：3P 杜邦线 1 根，1P 杜邦线 1 根。

四、实验原理图

如图 3-12-1 所示

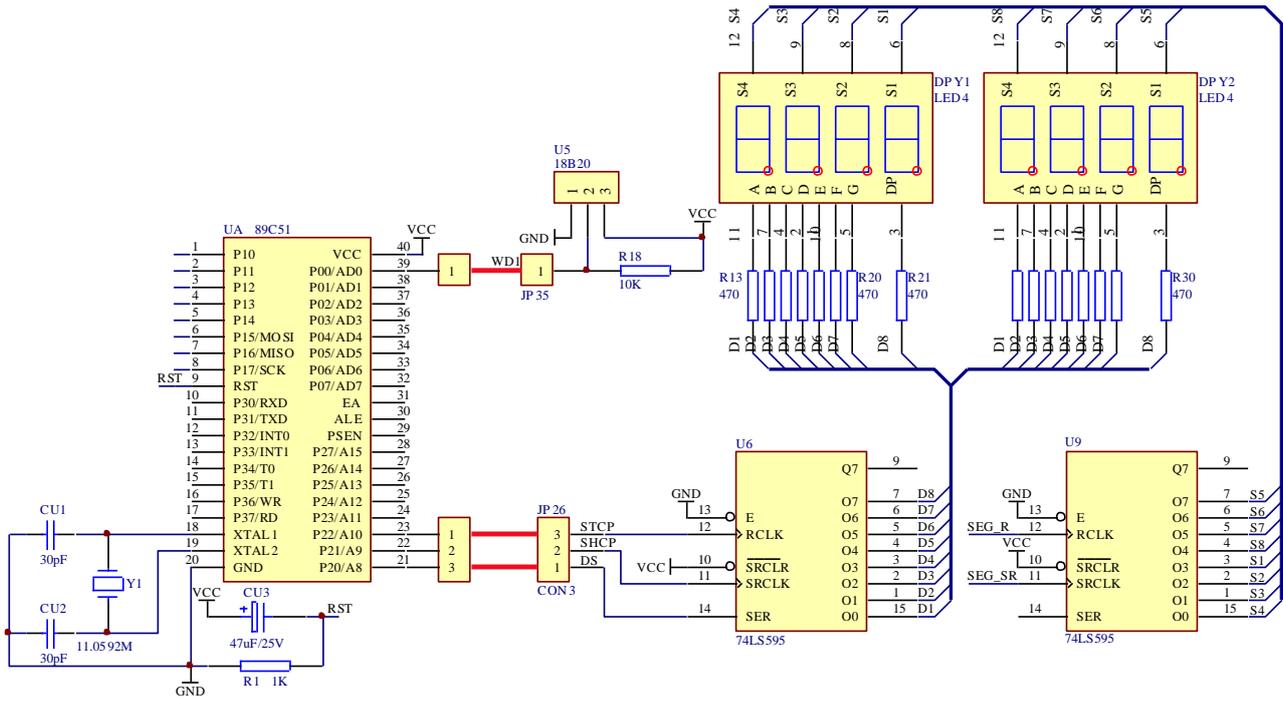


图 3-12-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-12-2 所示

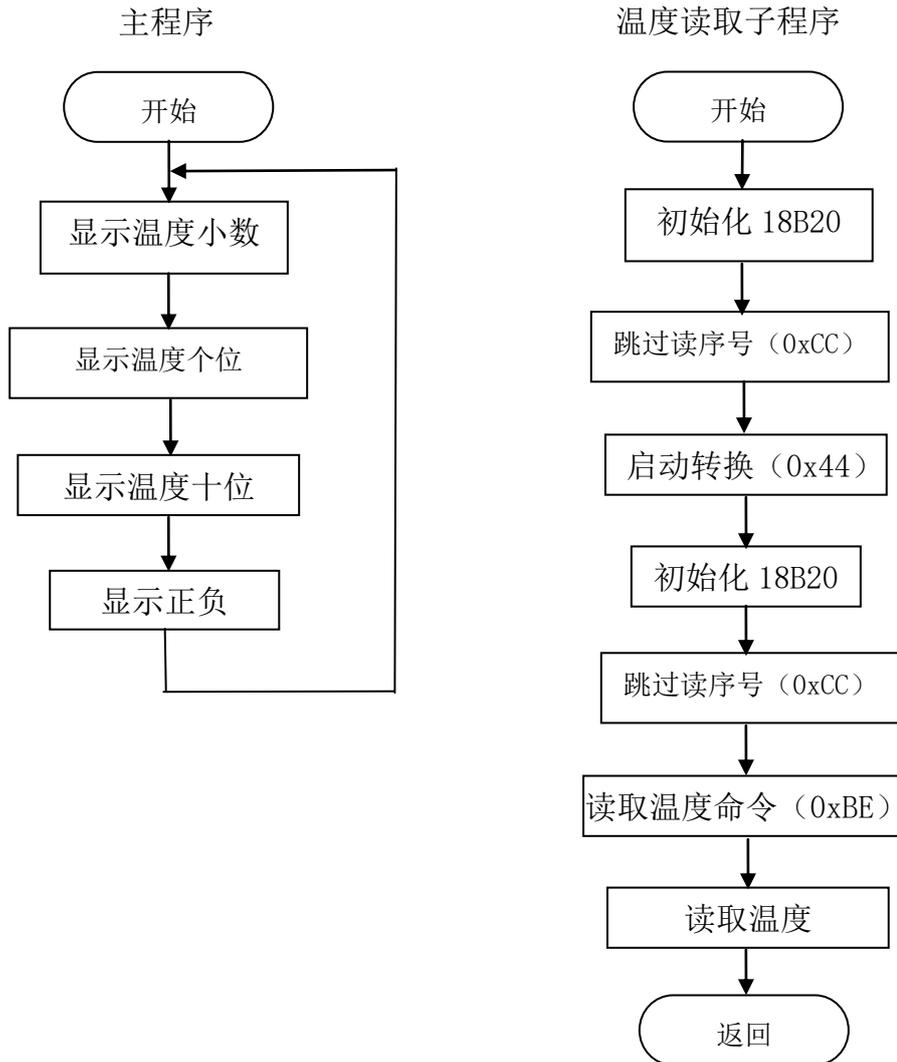


图 3-12-2

六、实验步骤

- 1、使用 1 根 3P 杜邦线连接单片机 P20~P22 口与底板“八位共阳数码管”区 JP26 相连，分别为：P20-DS，P21-SHCP，P22-STCP；
- 2、使用 1 根 1P 杜邦线将单片机 P00 与底板“两路温度传感器”区 JP35 的 WD1 相连；
- 3、在 KEIL 软件中调试 18B20 工程文件。用 STC 软件下载 18B20.HEX 程序；
- 4、实验现象： 数码管前四位显示当前温度值。

实验十三 直流电机测试

一、实验目的

了解直流电机控制的基本原理，掌握电机转动编程方法。

二、实验内容

编程实现直流电机的控制。

三、实验说明

在电压允许范围内，直流电机的转速随着电压的升高而加快，若加上的电压为负电压，则电机反向旋转。

硬件准备：2P 杜邦线 2 根。

四、实验原理图

如图 3-13-1 所示

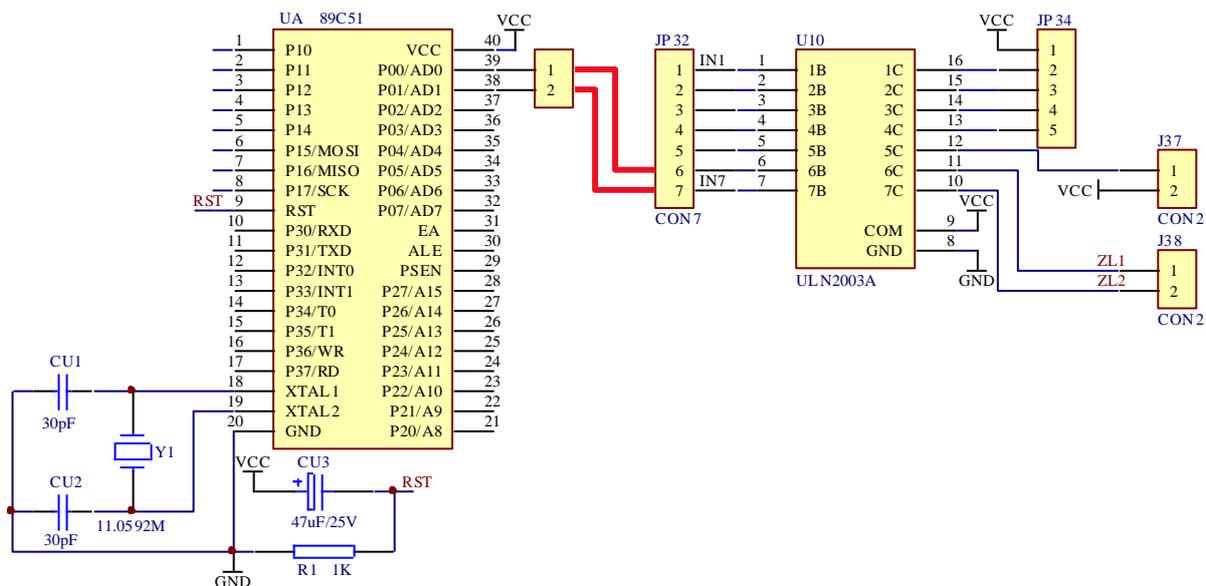


图 3-13-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-13-2 所示

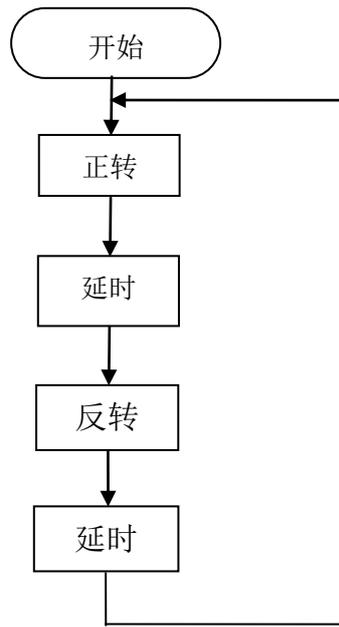


图 3-13-2

六、实验步骤

1、方式 1：直流电机使用“电机驱动模块”区 JP38 接口时，单片机 P00 口与底板“电机驱动模块”区 JP32 相连，即：P00-IN6，P01-IN7。

2、方式 2：直流电机使用“电机驱动模块”区 JP37 接口时，单片机 P00 口与底板“电机驱动模块”区 JP32 相连，即：P00-IN5；

3、在 KEIL 软件中调试 z1dj 工程文件。用 STC 软件下载 z1dj.HEX 程序。

4、实验现象：方式 1：电机正反转控制；方式 2：转动和停止反复。

实验十四 步进电机控制

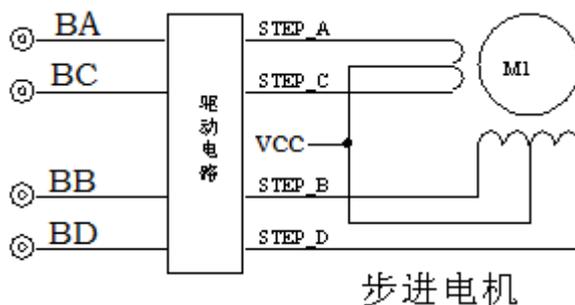
一、实验目的

了解步进电机控制的基本原理，掌握步进电机转动编程方法。

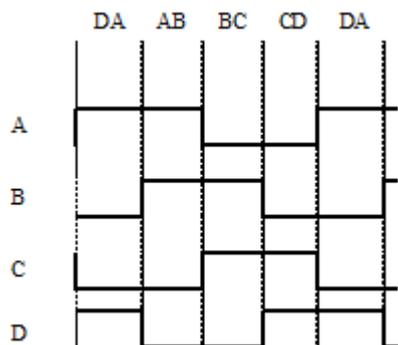
二、实验内容

编程实现步进电机的控制。

三、实验说明



步进电机驱动原理是通过对每相线圈中的电流的顺序切换来使电机作步进式旋转。切换是通过单片机输出脉冲信号来实现的。所以调节脉冲信号的频率便可以改变步进电机的转速，改变各相脉冲的先后顺序，可以改变电机的旋转方向。我们编制的程序是使步进电机的转速由慢到快逐步加速。电机驱动方式可以采用双四拍(AB→BC→CD→DA→AB)方式，各种工作方式的时序图如下：（高电平有效）



上图中示意的脉冲信号是高有效，但实际控制时公共端是接在 VCC 上的，所以实际控制脉冲是低有效。单片机口输出的脉冲信号经 (ULN2003A) 反相驱动后，向步进电机输出脉冲信号序列来控制步进电机的运转。

硬件准备：4P 杜邦线 1 根，5P 杜邦线 1 根。

四、实验原理图

如图 3-14-1 所示

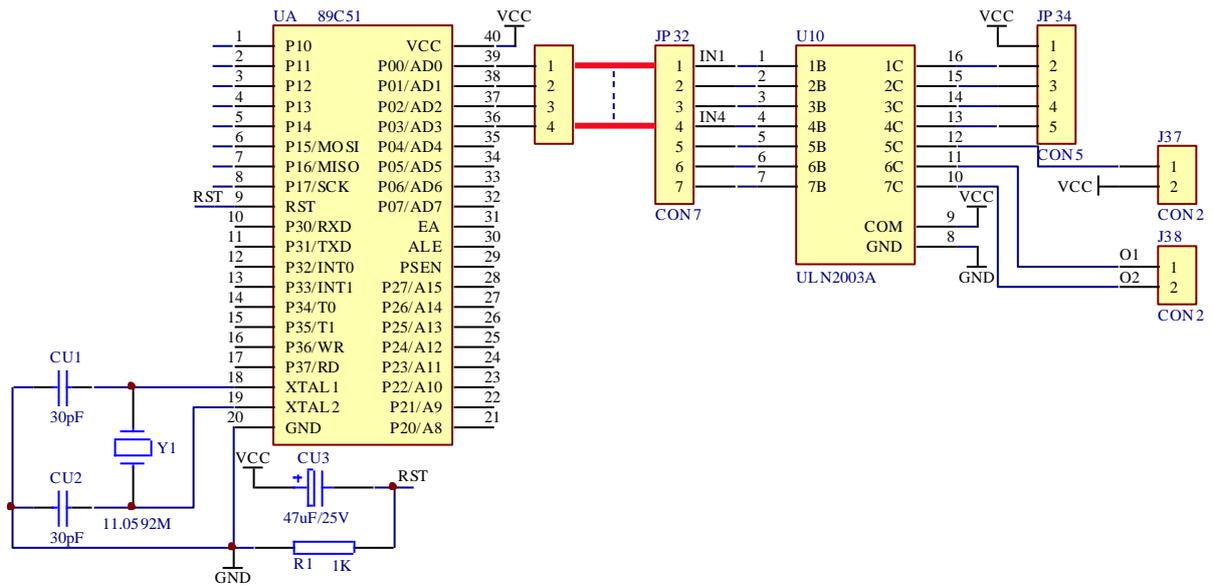


图 3-14-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-14-2 所示

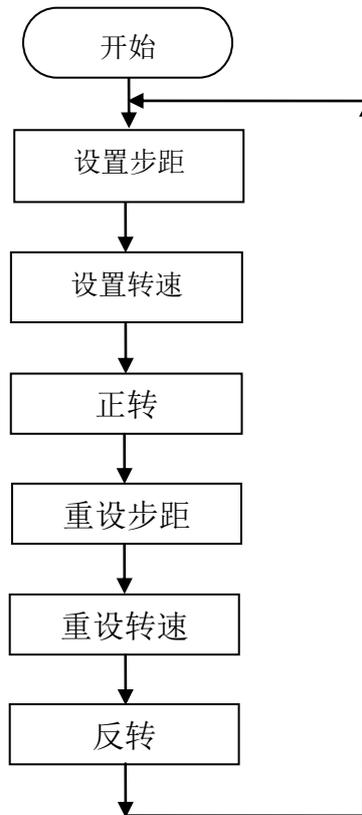


图 3-14-2

六、实验步骤

1、当“步进电机”区的步进电机接口 JP93 和“电机驱动模块”区的 JP30 连接时，单片机 P00~P03 口与底板“继电器模块”区 JP23 相连，分别为： P00-IN1， P01-IN2， P02-IN3， P03-IN4；

2、当“步进电机”区的步进电机接口 JP93 和“电机驱动模块”区的 JP34 接口时，单片机 P00~P03 口与底板 JP32 相连，分别为： P00-IN1， P01-IN2， P02-IN3， P03-IN4；

3、在 KEIL 软件中调试 bjdj 工程文件。用 STC 软件下载 bjdj.HEX 程序；

4、实验现象：单双八拍工作方式，步进电机先正转再反转。

实验十五 继电器控制

一、实验目的

- 1、学习 I/O 端口的使用方法；
- 2、掌握继电器的控制的基本方法。

二、实验内容

用单片机的端口输出电平控制继电器的吸合和断开，实现对外部装置的控制。本实验通过继电器控制 LED 的亮和灭。

三、实验说明

现代自动控制设备中，都存在一个电子电路与电气电路的互相连接问题，一方面要使电子电路的控制信号能够控制电气电路的执行元件（电动机，电磁铁，电灯等），另一方面又要为电子线路的电气电路提供良好的电气隔离，以保护电子电路和人身的安全。继电器便能完成这一桥梁作用。本实验采用的继电器其控制电压是 5V，控制端为高电平时，继电器工作常开触点吸合，连触点的 LED 灯被点亮。当控制端为低电平时，继电器不工作。执行时，对应的 LED 将随继电器的开关而亮灭。

硬件准备：1P 杜邦线 4 根。

四、实验原理图

如图 3-15-1 所示

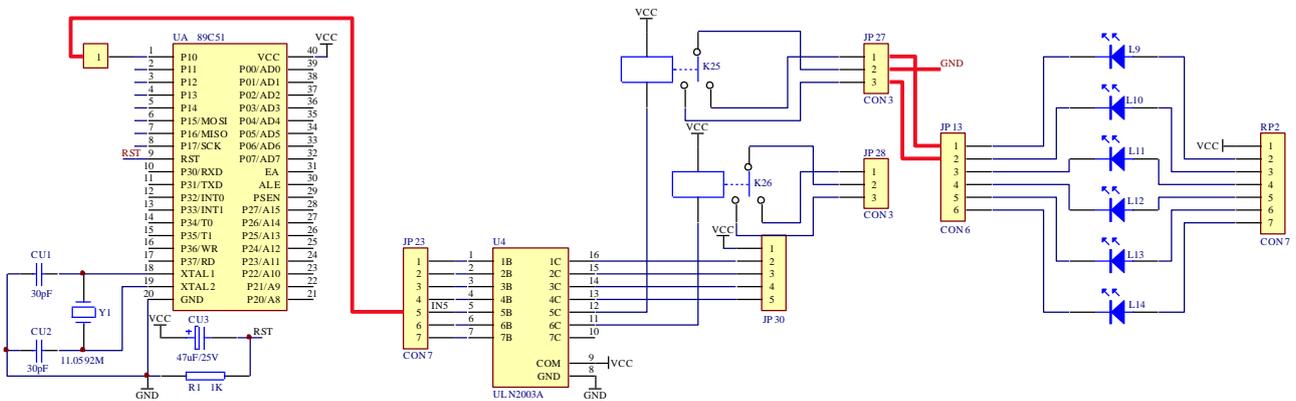


图 3-15-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-15-2 所示

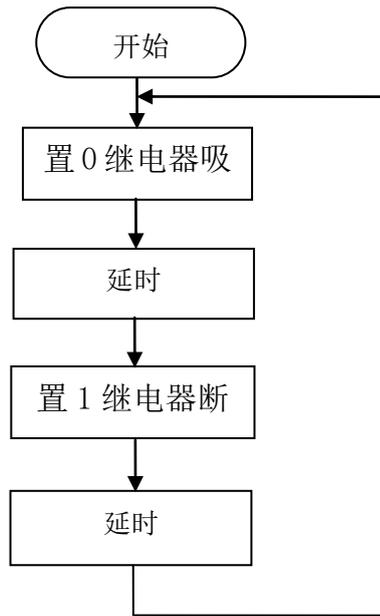


图 3-15-2

六、实验步骤

- 1、如果使用继电器 K25（上），单片机 P10 接底板“继电器模块”区单排针 JP23 的 IN5；
 - 2、如果使用继电器 K26（下），单片机 P10 接底板“继电器模块”区单排针 JP23 的 IN6。
- 常开接 JP13 的 L1，常闭接 JP13 的 L3, 公共端接地；
- 3、在 KEIL 软件中调试 jdq 工程文件。用 STC 软件下载 jdq. HEX 程序；
 - 4、实验现象：红绿灯交替点亮。

实验十六 串口通讯测试

一、实验目的

- 1、了解 RS232 串行通信的原理，学习 RS232 串行通信程序的设计和操作；
- 2、学习使用上位机软件“串行调试助手”。

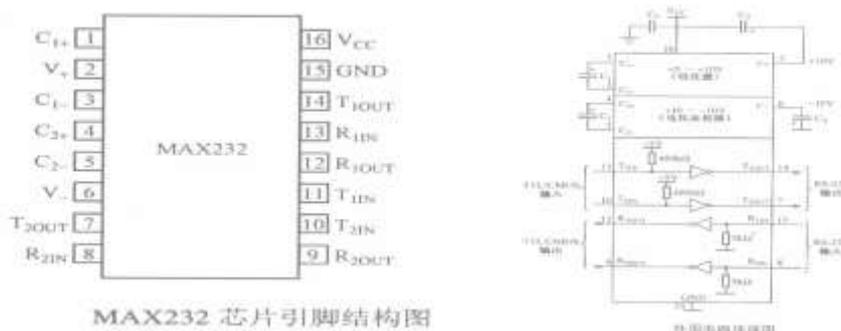
二、实验内容

MAX232 将单片机发出的 TTL 电平信号转化为 RS232 电平信号。收发信号从 9 芯插座通过 RS232 串行通信线传送到 PC 机。

三、实验说明

1、MAX232 芯片实现 RS-232 电平与 TTL 电平转换

MAX232 芯片是 MAXIM 公司生产的、包含两路收发器和驱动器的 IC 芯片，它的内部有一个电源电压变换器，可以把输入的+5V 电源电压变换成为 RS-232 输出电平所需的+10V 电压。所以，采用此芯片接口的串行通信系统只需单一的+5V 电源就可以了。对于没有+12V 电源的场合，其适应性更强，加之其价格适中，硬件接口简单，所以被广泛采用。



上右图中上半部分电容 C1、C2、C3、C4、V+及 V-是电源变换电路部分。在实际应用中，器件对电源噪声很敏感，因此 Vcc 必须要对地加去耦电容 C5，其值为 0.1uF。按芯片手册中介绍，电容 C1，C2，C3，C4 应取 10uF/16V 的电解电容，经大量实验及实际应用，这 4 个电容都可以选用 0.1uF 的非极性瓷片电容代替 10uF/16V 的电解电容，在具体设计电路时，这 4 个电容要尽量靠近 MAX232 芯片，以提高抗干扰能力。下半部分为发送和接收部分，实际应用中，T1IN, T2IN 可直接连接 TTL/CMOS 电平的 51 单片机的串行发送端 TXD；R1OUT, R2OUT 可直接连接 TTL/CMOS 电平的 51 单片机的串行接收端 RXD；T1OUT, T2OUT 可直接连接代机的 RS-232 串口的接收端 RXD；R1IN, R2IN 可直接连接 PC 机的 RS-232 串口的发送端 TXD。现从 MAX232 芯片中两路发送、接收中任选一路作为接口。要注意其发送、接收的引脚要对应。如使 T1IN 连接单片机的发送端 TXD，则 PC 机的 RS-232 接收端 RXD 一定要对应接 T1OUT 引脚。同时，R1OUT 连接单片机的 RXD 引脚，PC 机的 RS-232 发送端 TXD 对应接 R1IN 引脚。

2、硬件准备：串口线 1 根。

3、软件准备：串口调试助手软件或者 STC_ISP（带串口调试助手）。

四、实验原理

如图 3-16-1 所示

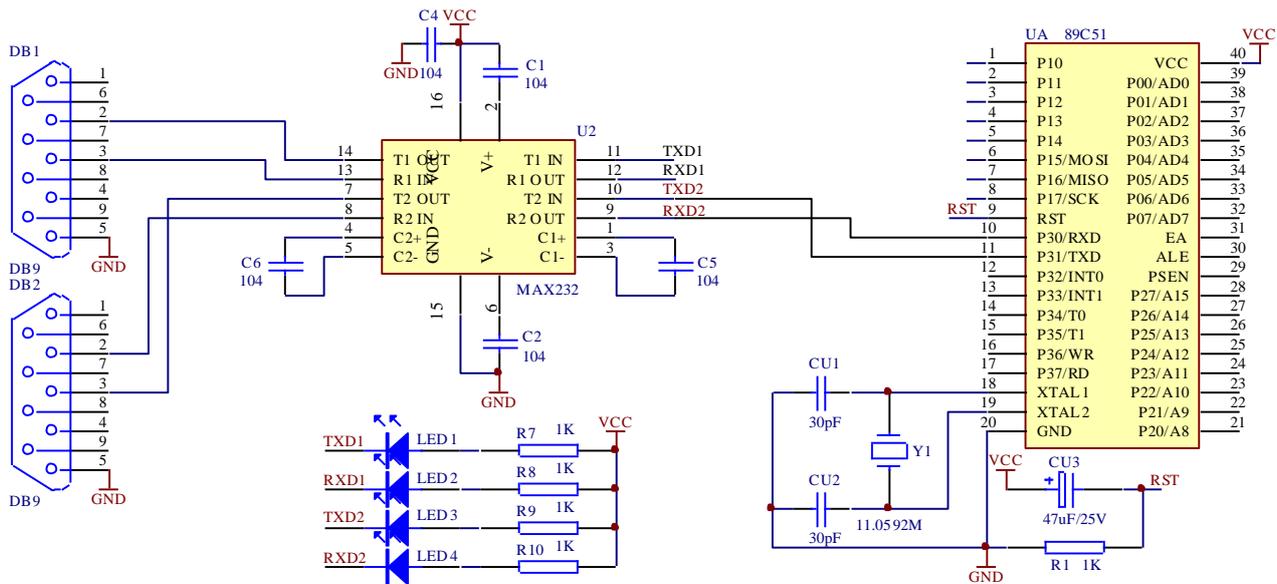


图 3-16-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-16-2 所示

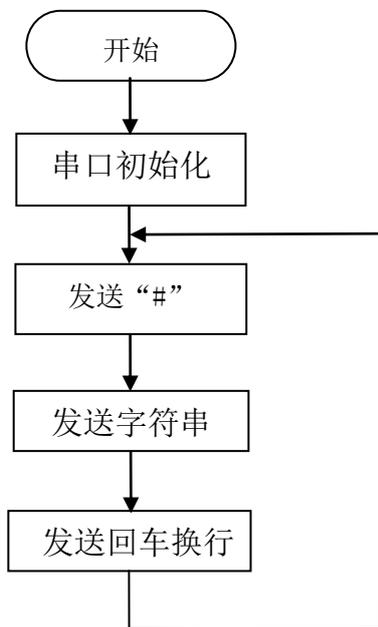


图 3-16-2

六、实验步骤

- 1、在 STC 下载方式无需接线；
- 2、在 KEIL 软件中调试 cktx 工程文件。用 STC 软件下载 cktx.HEX 程序；
- 3、实验现象：打开串口助手，波特率为 9600。字符方式显示 “#welcome to dvcc.china.b2b.cn”。

实验十七 DS1302 时钟模块测试

一、实验目的

- 1、了解实时时钟芯片 DS1302 的原理，掌握其编程方法；
- 2、进一步掌握 OLED 屏的使用。

二、实验内容

编程实现 DS1302 读写，并利用 DS1302 实现电子时钟；

三、实验说明

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。

1、引脚功能及结构

如图 3-17-1 示出 DS1302 的引脚排列，其中 Vcc1 为后备电源，VCC2 为主电源。在主电源关闭的情况下，也能保持时钟的连续运行。DS1302 由 Vcc1 或 Vcc2 两者中的较大者供电。当 Vcc2 大于 Vcc1+0.2V 时，Vcc2 给 DS1302 供电。当 Vcc2 小于 Vcc1 时，DS1302 由 Vcc1 供电。X1 和 X2 是振荡源，外接 32.768kHz 晶振。RST 是复位/片选线，通过把 RST 输入驱动置高电平来启动所有的数据传送。RST 输入有两种功能：首先，RST 接通控制逻辑，允许地址/命令序列送入移位寄存器；其次，RST 提供终止单字节或多字节数据的传送手段。当 RST 为高电平时，所有的数据传送被初始化，允许对 DS1302 进行操作。如果在传送过程中 RST 置为低电平，则会终止此次数据传送，I/O 引脚变为高阻态。上电运行时，在 $V_{cc} \geq 2.5V$ 之前，RST 必须保持低电平。只有在 SCLK 为低电平时，才能将 RST 置为高电平。I/O 为串行数据输入输出端(双向)，后面有详细说明。SCLK 始终是输入端。

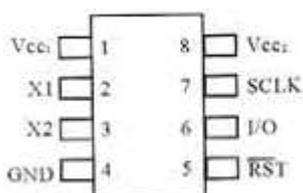
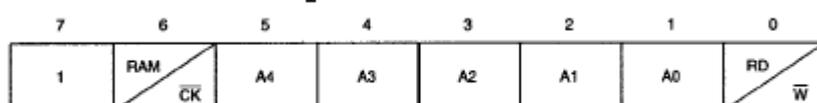


图 3-17-1 ds1302 引脚图

2、DS1302 的控制字节

DS1302 的控制字如下图所示。控制字的最高有效位(位 7)必须是逻辑 1，如果它为 0，则不能把数据写入 DS1302 中，位 6 如果为 0，则表示存取日历时钟数据，为 1 表示存取 RAM 数据；位 5 至位 1 指示操作单元的地址；最低有效位(位 0)如为 0 表示要进行写操作，为 1 表示进行读操作，控制字节总是从最低位开始输出。



3、数据输入输出(I/O)

在控制指令字输入后的下一个 SCLK 时钟的上升沿时，数据被写入 DS1302，数据输入从低位即位 0 开始。同样，在紧跟 8 位的控制指令字后的下一个 SCLK 脉冲的下降沿读出 DS1302 的数据，读出数据时从低位 0 位到高位 7。

4、DS1302 的寄存器

寄存器名	命令字节		取值范围	寄存器内容							
	写	读		7	6	5	4	3	2	1	0
秒寄存器	80H	81H	00~59	CH	10S			SEC			
分寄存器	82H	83H	00~59	0	10MIN			MIN			
时寄存器	84H	85H	00~23 或 00~12	12/2 4	0	10A/P		HR			
日寄存器	86H	87H	01~ 28, 29, 30, 31	0	0	10DATE		DATE			
月寄存器	88H	89H	01~12	0	0	0	10 M	MONTH			
周寄存器	8AH	8BH	01~07	0	0	0	0	0	DAY		
年寄存器	8CH	8DH	00~99	10YEAR				YEAR			

硬件准备：1P 杜邦线 5 根，3P 杜邦线 1 根。

四、实验原理图

如图 3-17-2 所示

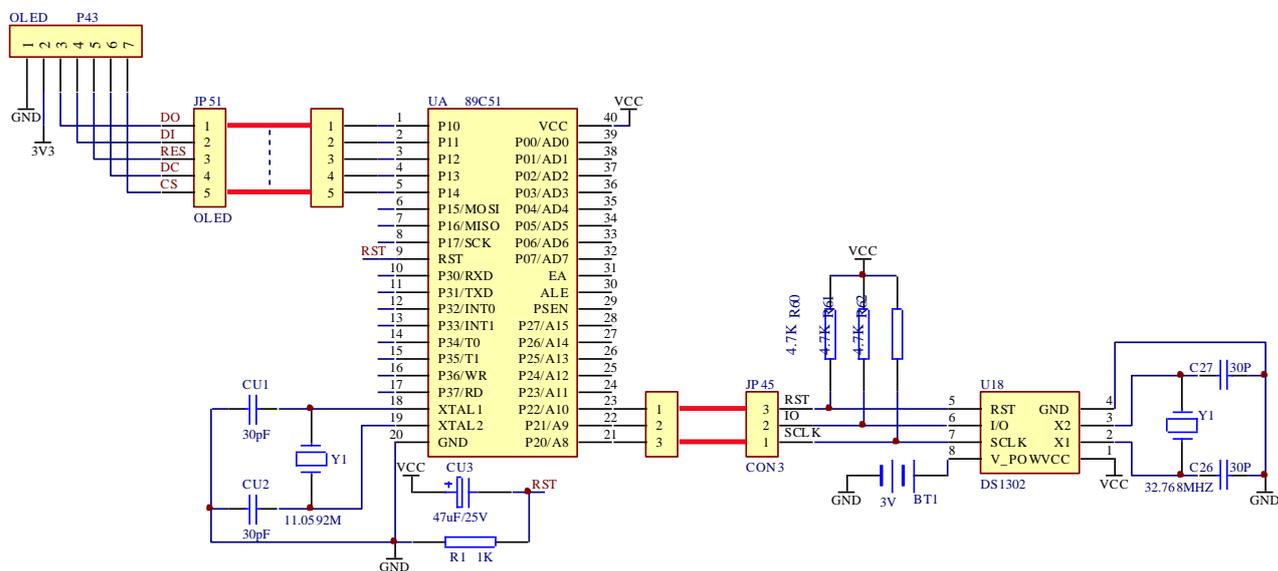


图 3-17-2

五、实验程序与框图

源程序见随机光盘

流程图见图 3-17-3 所示

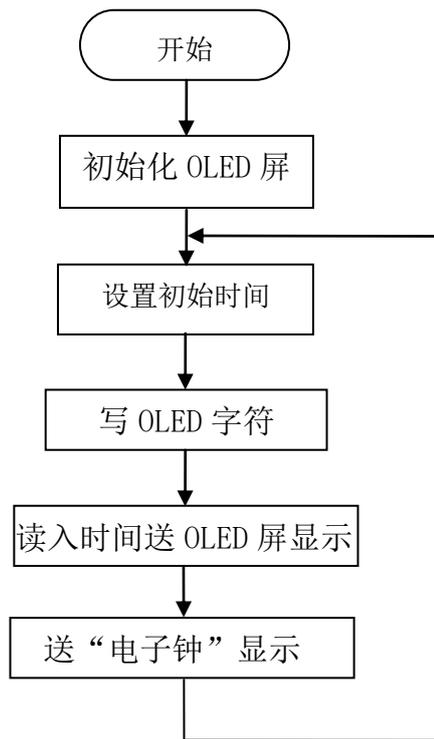


图 3-17-3

六、实验步骤

1、用 1 根 5P 杜邦线将单片机 P10~P14 口与底板“OLED 液晶”区 JP51 相连，分别为：P10-D0，P11-D1，P12-RES，P13-CD，P14-CS；

2、用 1 根 3P 杜邦线将单片机 P20~P22 口与底板“DS1302 时钟”区 JP45 相连。分别为：P20-SCLK，P21-IO，P22-RST；

3、在 KEIL 软件中调试 DS1302 工程文件。用 STC 软件下载 DS1302.HEX 程序；

4、实验现象： OLED 第一行显示：“ -DS1302 TEST- ”，第二行显示：-年-月-日，第三行显示： 时：分：秒，第四行显示：电子钟。

实验十八 蜂鸣器测试

一、实验目的

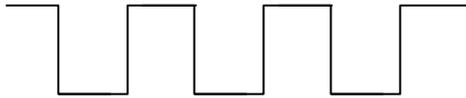
- 1、学习 I/O 端口的使用方法；
- 2、掌握蜂鸣器的控制的基本方法。

二、实验内容

用单片机的端口，输出电平控制蜂鸣器。

三、实验说明

蜂鸣器是一种一体化结构的电子音响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。我们开发板上常用的蜂鸣器就是常说的交流蜂鸣器或直流蜂鸣器（自激式蜂鸣器）。直流蜂鸣器是给一定的驱动直流电压就会响。而交流蜂鸣器是需要给蜂鸣器一个脉冲才会响。常见的有 PWM 控制蜂鸣器的频率。脉冲就是高低电平的切换，如下图的一个方波脉冲。



我们用单片机的 IO 口实现一种这样不同频率的方波，驱动蜂鸣器发音。

消防车：以低频频率 650Hz 至 750Hz，高频频率 1450Hz 至 1550Hz，由低频升至高频时间 1.5 秒，再由高频降至低频为 3.5 秒。

救护车：以低频频率 650Hz 至 750Hz，高频频率 900z 至 1000Hz，低频持续时间 0.4 秒，高频持续时间 0.6 秒，高低频交替进行。

警车：以低频频率 650Hz 至 750Hz，高频频率 1450Hz 至 1550Hz，由低频升至高频时间 0.23 秒，再由高频降至低频为 0.1 秒。

硬件准备：1P 杜邦线 1 根。

五、实验原理图

如图 3-18-1 所示

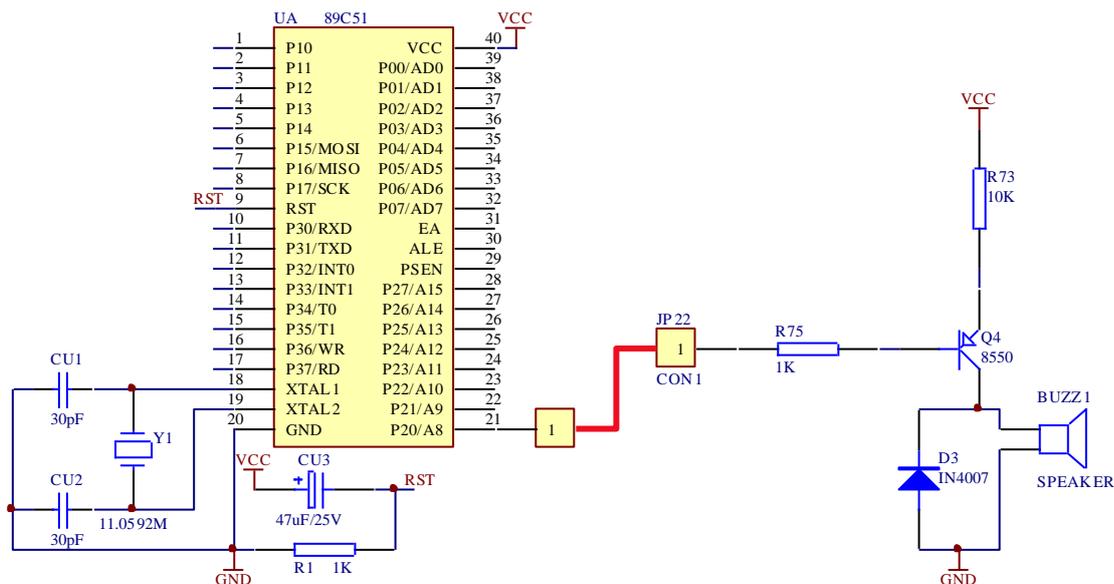


图 3-18-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-18-2 所示

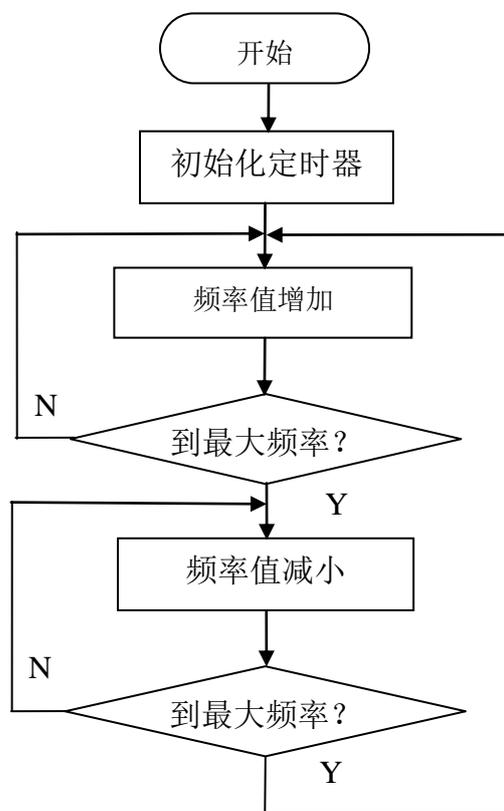


图 3-18-2

六、实验步骤

- 1、用 1P 杜邦线将单片机 P20 口与底板“蜂鸣器”区 JP22 相连；
- 2、在 KEIL 软件中调试 FMQ 工程文件。用 STC 软件下载 FMQ.HEX 程序；
- 3、实验现象：蜂鸣器发出消防车的响声。

实验十九 AT24C02 模块测试

一、实验目的

通过本实验，初步掌握 IIC 总线的读写操作。

二、实验内容

向 24C02 写入数据，然后读出数据，在 LCD1602 屏上显示。

三、实验说明

1、AT24C01A/AT24C02 为 I²C 总线型 EEPROM 存储器，容量为 1K/2K 位 (128/256*8)，读/写时序遵循 I²C 总线协议标准。AT24C01A/AT24C02 内部设有一个控制寄存器，其每一位的含义如下：

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	0	1	0	A2	A1	A0	R/W

其中 A2/A1/A0 用于选择总线上待访问的 I²C 器件，R/W=1 读操作，R/W=0 写操作；从上述时序可以看出，I²C 总线上最多可以扩展 2³=8 片同样的 1K/2K 容量 EEPROM 存储器或者可以扩展 1 片容量为 16K Bits 的 EEPROM 存储器。如果扩展 8 片 2K 以内容量的 EEPROM 存储器，每片存储器将对应一个地址，这个地址由芯片的 A2/A1/A0 决定，因此在同一个 I²C 总线上扩展多片同样的 EEPROM 存储器时，需要注意任意两片的地址不能相同。由于实验箱上的 AT24C01A/AT24C02 的 A2/A1/A0 引脚全部接地，等效为实验箱上的 AT24C01A/AT24C02 的地址为 0，所以在实验中读写控制字分别为：0xa1/0xa0。

2、硬件准备：3P 杜邦线 1 根，2P 杜邦线 1 根，8P 杜邦线 1 根。

四、实验原理图

如图 3-19-1 所示

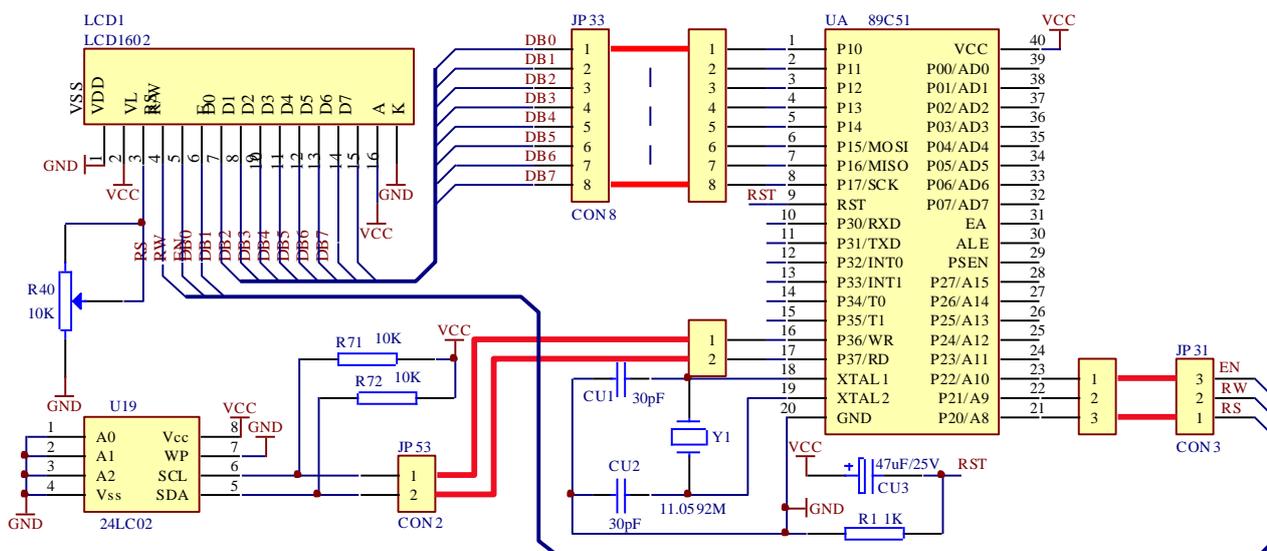


图 3-19-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-19-2 所示

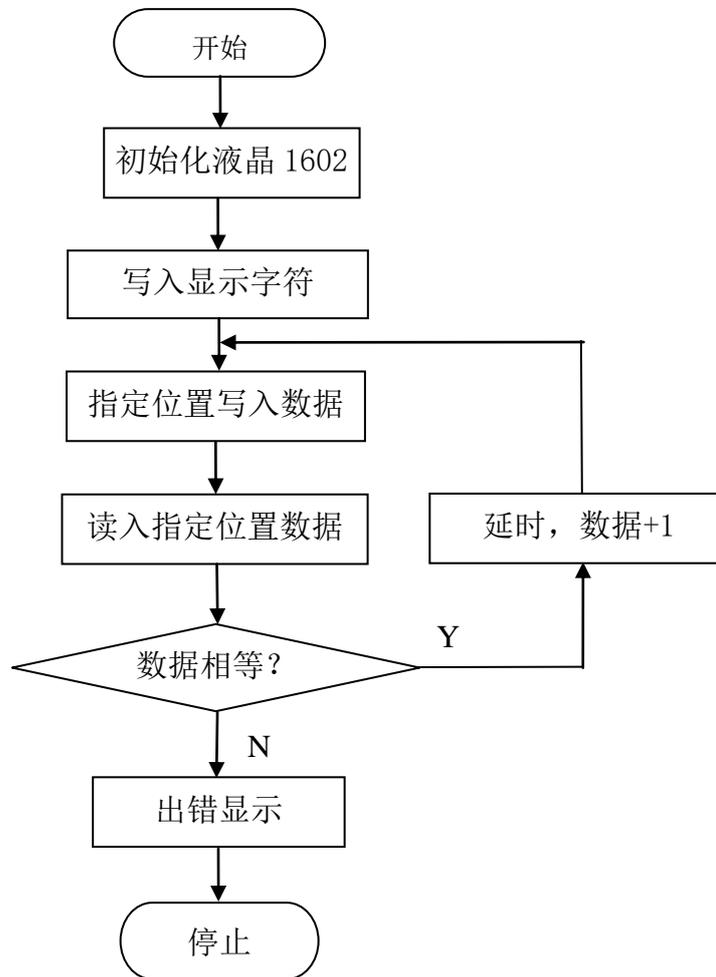


图 3-19-2

六、实验步骤

1、用 1 根 8P 杜邦线将单片机 P10~P17 口与底板“LCD1602 液晶”区 JP33 相连，分别为：P10-DB0，P11-DB1，P12-DB2，P13-DB3，P14-DB4，P15-DB5，P16-DB6，P17-DB7；

2、用 1 根 3P 杜邦线将单片机 P20~P22 口与底板 JP31 相连，分别为：P20-RS，P21-RW，P22-EN；

3、用 1 根 2P 杜邦线将单片机 P36~P37 口与底板“LCD1602 液晶”区 JP53 相连。分别为：P36-SCL，P37-SDA；

4、在 KEIL 软件中调试 24C02 工程文件。用 STC 软件下载 24C02.HEX 程序；

5、实验现象：第一行显示：“-AT24CXX TEST-”，第二行显示：“Read Data:00000!”，然后数据自动加 1。

实验二十 74HC164 串转并测试

一、实验目的

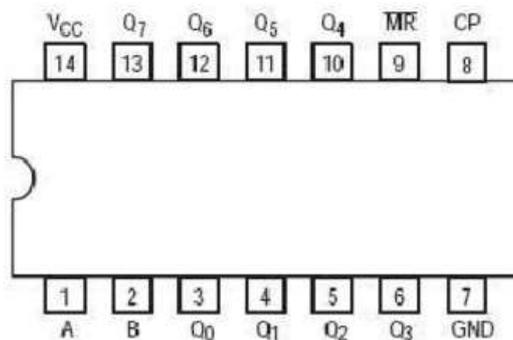
1、掌握利用 IO 口通过 74HC164 扩展 I/O 通道的方法。

二、实验内容

使用串行输入并行输出移位寄存器 74LS164，扩展一个 8 位输出通道，用于驱动发光二极管产生流水灯现象。

三、实验说明

(1) 74LS164 芯片如右图所示，它是 8 位串入并出移位寄存器，串行输入数据，然后并行输出。各引脚功能如下：



A、B (1、2 引脚)：数据输入端，数据通过这两个输入端之一串行输入；任一输入端可以用作高电平使能端，控制另一输入端的数据输入。当其中任意一个为低电平，则禁止新数据输入；当其中有一个为高电平，则另一个就允许输入数据。因此两个输入端或者连接在一起，或者把不用的输入端接高电平，一定不要悬空。Q0~Q7 (3~6, 10~13 引脚)：数据输出端 CP (8 号引脚)：时钟输入端。CP 每次由低变高时，数据右移一位，输入到 Q0，Q0 是两个数据输入端的逻辑与，它将上升时钟沿之前保持一个建立时间的长度。

MR：复位清除端，当 MR 为低电平时，其它所有输入端都无效，同时所有输出端均为低电平。

(2) 硬件准备：8P 杜邦线 1 根，3P 杜邦线 1 根。

四、实验原理图

如图 3-20-1 所示

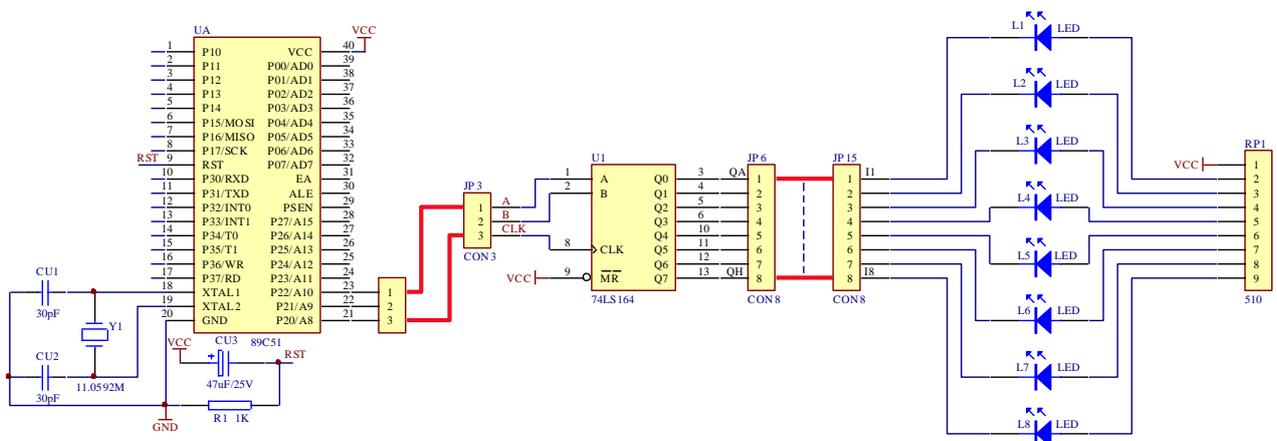


图 3-20-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-20-2 所示

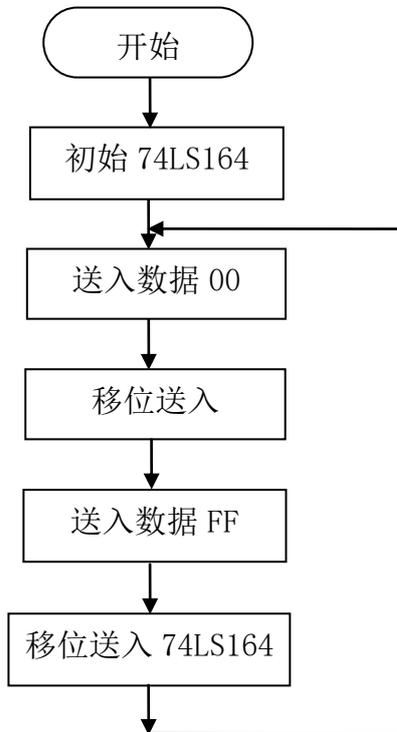


图 3-20-2

六、实验步骤

1、用 1 根 3P 杜邦线将单片机的 P20~P22 与底板“串转并模块”区 JP3 相连, P22-A, P21-B, P20-CLK;

2、用 1 根 8P 杜邦线将底板“串转并模块”区 JP6 与底板“单色流水灯”区 JP15 相连。QA-I1, QB-I2, QC-I3, QD-I4, QE-I5, QF-I6, QG-I7, QH-I8;

3、在 KEIL 软件中调试 cbzh 工程文件。用 STC 软件下载 cbzh. HEX 程序;

4、实验现象: 分别点亮 8 个 LED。

实验二十一 74HC165 并转串测试

一、实验目的

1、掌握利用 I/O 口通过 74HC165 扩展 I/O 输入通道的方法。

二、实验内容

使用并行输入串行输出 74LS165，扩展一个 8 位输入通道，读放开关量的值，通过 P1 给指示灯显示。

三、实验说明

74HC165/74LS165 是典型并入串出芯片，引脚如图 3-22-1 所示。

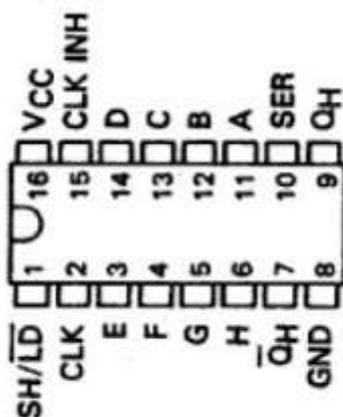


图 3-21-1

引出端符号	功能
CLK, CLK INH	时钟输入端（上升沿有效）
A—H	并行数据输入端
SER	串行数据输入端
QH	输出端
QHn	互补输出端
SH/LD	移位控制/置入控制（低电平有效）

硬件准备：8P 杜邦线 2 根，3P 杜邦线 1 根，1P 杜邦线 1 根。

四、实验原理图

如图 3-21-2 所示

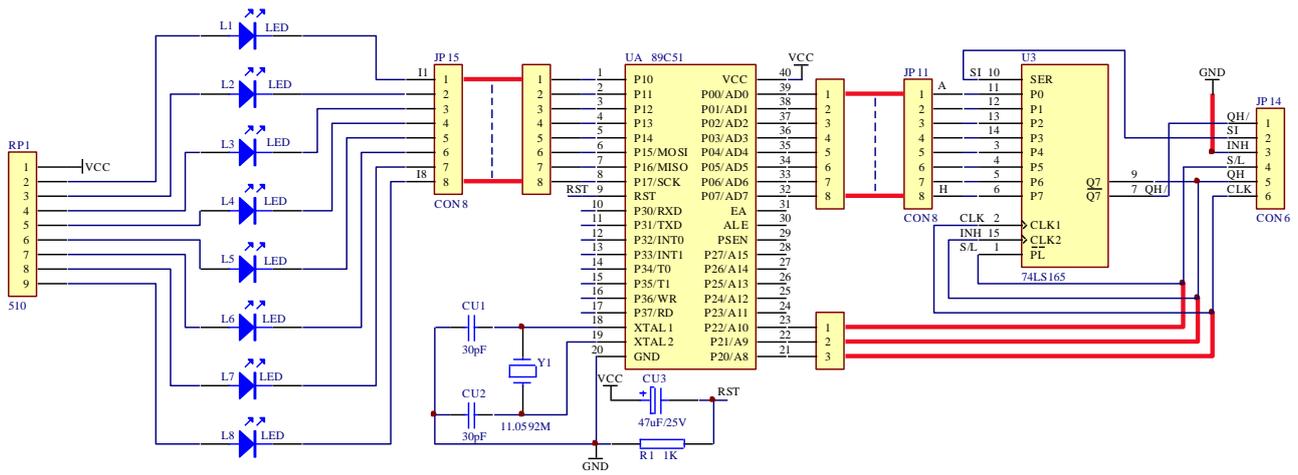


图 3-21-2

五、实验程序与框图

源程序见随机光盘

流程图见图 3-22-3 所示

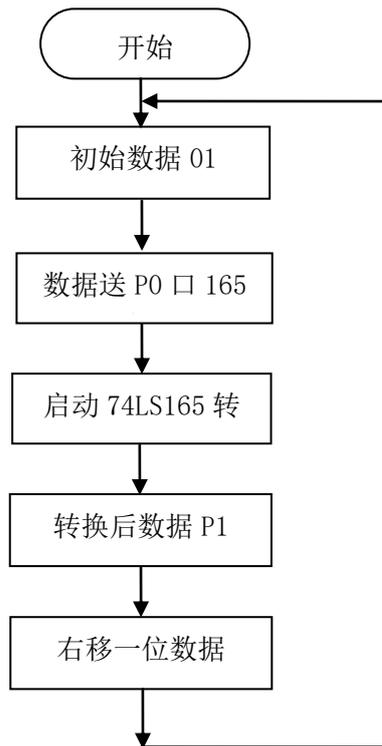


图 3-21-3

六、实验步骤

1、用 1 根 8P 杜邦线将单片机 P1 口与底板“单色流水灯”区 JP15 排针相连,分别为:P10-I1, P11-I2, P12-I3, P13-I4, P14-I5, P15-I6, P16-I7, P17-I8;

2、用 1 根 3P 杜邦线将单片机 P20~P22 口与底板“并转串模块”区 JP14 相连,分别为: P20-CLK, P21-QH, P22-S/L;

3、用 1 根 8P 杜邦线将单片机 P0 口与底板“并转串模块”区 JP11 相连，分别为：P00-A，P01-B，P02-C，P03-D，P04-E，P05-F，P06-G，P07-H。用一根 1P 杜邦线连接底板“并转串模块”区 JP14 的 INH 引脚与 GND；

4、在 KEIL 软件中调试 bczh 工程文件。用 STC 软件下载 bczh.HEX 程序；

5、实验现象： P0 口将并行数据送入 74HC165，P1 口接收 74HC165 串行输出。使得 8 个 LED 灯依次点亮。

实验二十二 74HC138 三八译码器

一、实验目的

1、掌握利用 I/O 口通过 74HC138 的译码功能。

二、实验内容

使用 74HC138 分别点亮 8 个发光二极管。

三、实验说明

74LS138 为 3 线—8 线译码器，引脚如图 3-22-1 所示。

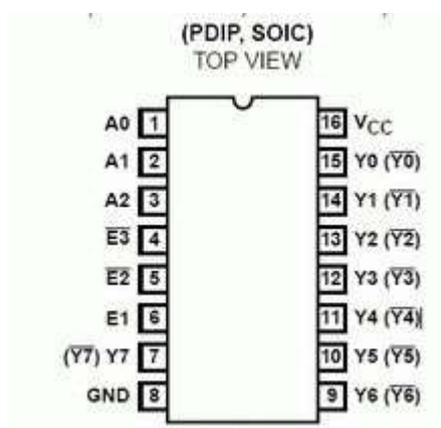


图 3-22-1

引脚功能

A0~A2: 地址输入端

STA (E1): 选通端

\overline{STB} (E2)、 \overline{STC} (E3): 选通端 (低电平有效)

$\overline{Y0} \sim \overline{Y7}$: 输出端 (低电平有效)

VCC: 电源正

GND: 地

A0~A2 对应 Y0—Y7；A0，A1，A2 以二进制形式输入，然后转换成十进制，对应相应 Y 的序号输出低电平，其他均为高电平。

硬件准备：1P 杜邦线 4 根，8P 杜邦线 1 根。

五、实验原理图

如图 3-22-2 所示

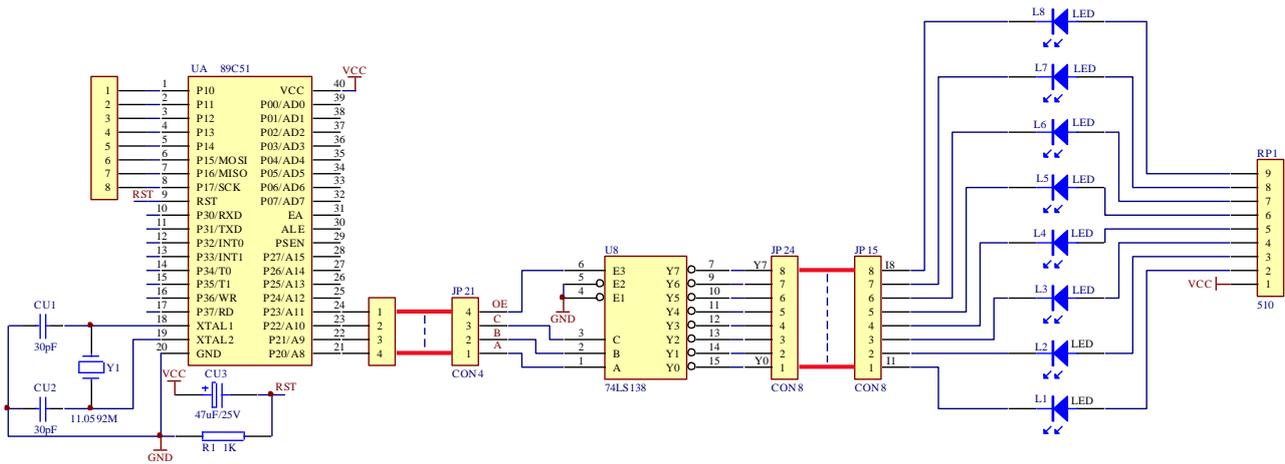


图 3-22-2

五、实验程序与框图

源程序见随机光盘

流程图见图 3-22-3 所示

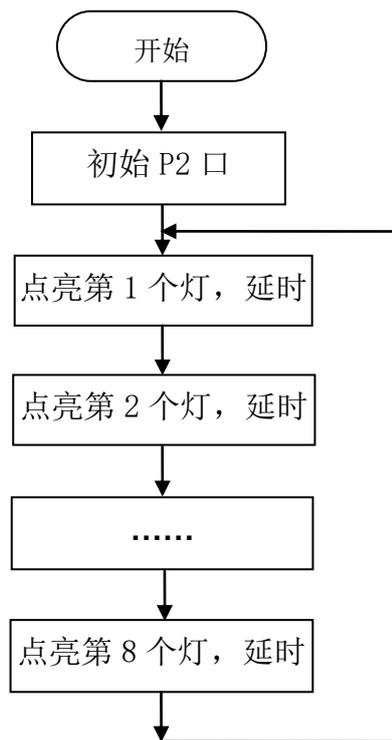


图 3-22-3

六、实验步骤

1、用 2 根 2P 杜邦线将单片机 P20~P23 口与底板“三八译码器”区 JP21 排针相连，分别为：P20-A，P21-B，P22-C，P23-OE；

2、用 1 根 8P 杜邦线将底板“三八译码器”区 JP24 与“单色流水灯”区 JP15 相连。分别为：Y0-I1，Y1-I2，Y2-I3，Y3-I4，Y4-I5，Y5-I6，Y6-I7，Y7-I8；

- 3、在 KEIL 软件中调试 74138 工程文件。用 STC 软件下载 74138.HEX 程序；
- 4、实验现象：分别点亮 8 个 LED。

实验二十三 74HC573 锁存器

一、实验目的

1、掌握 74HC573 的锁存功能。

二、实验内容

使用 74HC573 驱动 8 个发光二极管。

三、实验说明

74HC573 八 D 透明锁存器，引脚图如图 2-23-1 所示。

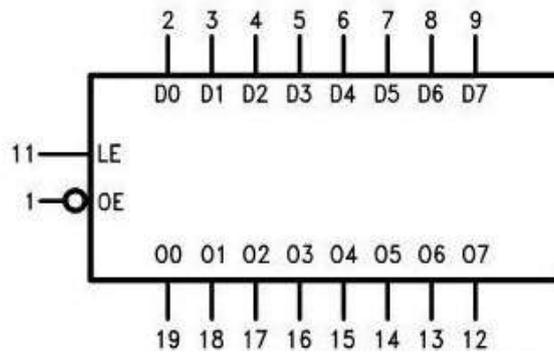


图 2-23-1

74LS573 是三态总线驱动输出

- 置数全并行存取
- 缓冲控制输入
- 使能输入有改善抗扰度的滞后作用

原理：74LS573 的八个锁存器都是透明的 D 型锁存器，当使能（G）为高时，Q 输出将随数据（D）输入而变。当使能为低时，输出将锁存在已建立的数据电平上。输出控制不影响锁存器的内部工作，即老数据可以保持，甚至当输出被关闭时，新的数据也可以置入。

这种电路可以驱动大电容或低阻抗负载，可以直接与系统总线接口并驱动总线，而不需要外接口。特别适用于缓冲寄存器，I/O 通道，双向总线驱动器和工作寄存器。

硬件准备：8P 杜邦线 2 根，2P 杜邦线 1 根。

四、实验原理图

如图 3-23-2 所示

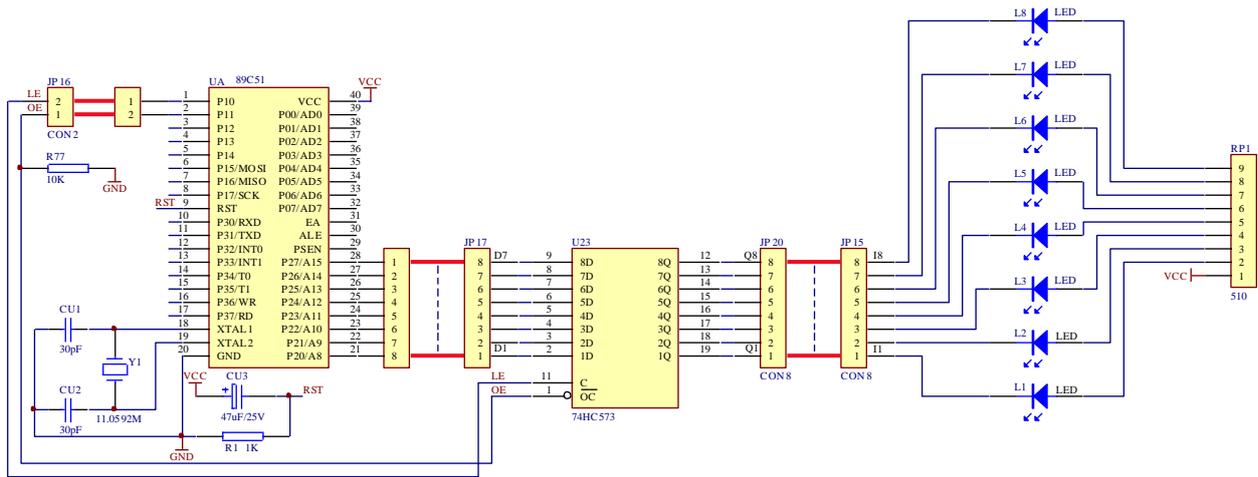


图 3-23-2

五、实验程序与框图

源程序见随机光盘

流程图见图 3-23-3 所示

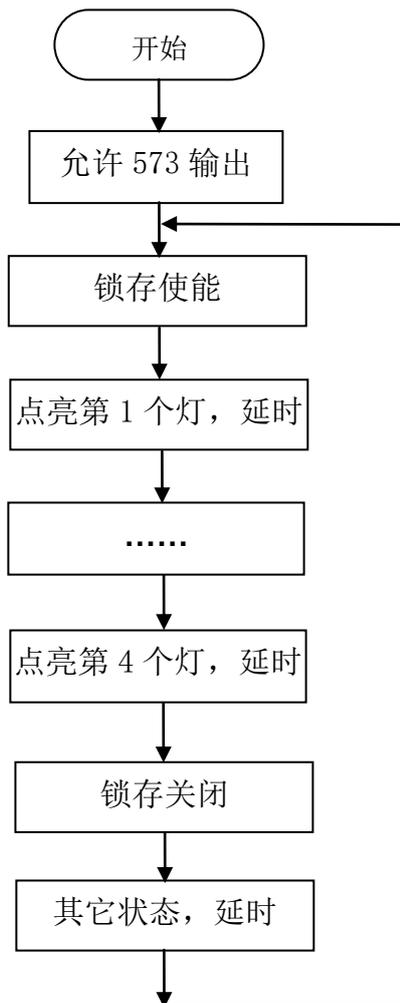


图 3-23-3

六、实验步骤

1、用 1 根 2P 杜邦线将单片机 P10~P11 接底板“八位锁存器”区 JP16，具体为：P10-LE，P11-OE；

2、用 1 根 8P 杜邦线将单片机 P20~P27 接底板“八位锁存器”区 JP17，具体为：P20-D1，P21-D2，P22-D3，P23-D4，P24-D5，P25-D6，P26-D7，P27-D8；

3、用 1 根 8P 杜邦线将底板 JP20 接“单色流水灯”区 JP15。具体为：Q1-I1，Q2-I2，Q3-I3，Q4-I4，Q5-I5，Q6-I6，Q7-I7，Q8-I8；

4、在 KEIL 软件中调试 74LS573 工程文件。用 STC 软件下载 74LS573.HEX 程序；

5、实验现象：控制 74HC573，点亮前 4 个 LED，后 4 个 LED 灭。

实验二十四 红外发送接收测试

一、实验目的

1、掌握红外发送接收的工作原理。

二、实验内容

编程控制红外发送管发送 38kHz 方波信号，通过红外接收管接收这个信号，然后通过发光二极管看是否接收到这个 38kHz 方波信号。

三、实验说明

VS1838B 为红外接收集成模块，用红外发射管对着 VS1838B 发射 38kHz 方波(占空比可调为 20%-30%)，当 VS1838B 接收到 38kHz 的方波时，VS1838B 第三脚输出脚输出低电平 0，该脚接到单片机的中断 0 输入脚 P3.2，产生中断信号，然后单片机控制发光管 L1；当没有接收到信号时时，输出脚输出高电平 1。

硬件准备：1P 杜邦线 3 根。

四、实验原理图

如图 3-24-1 所示

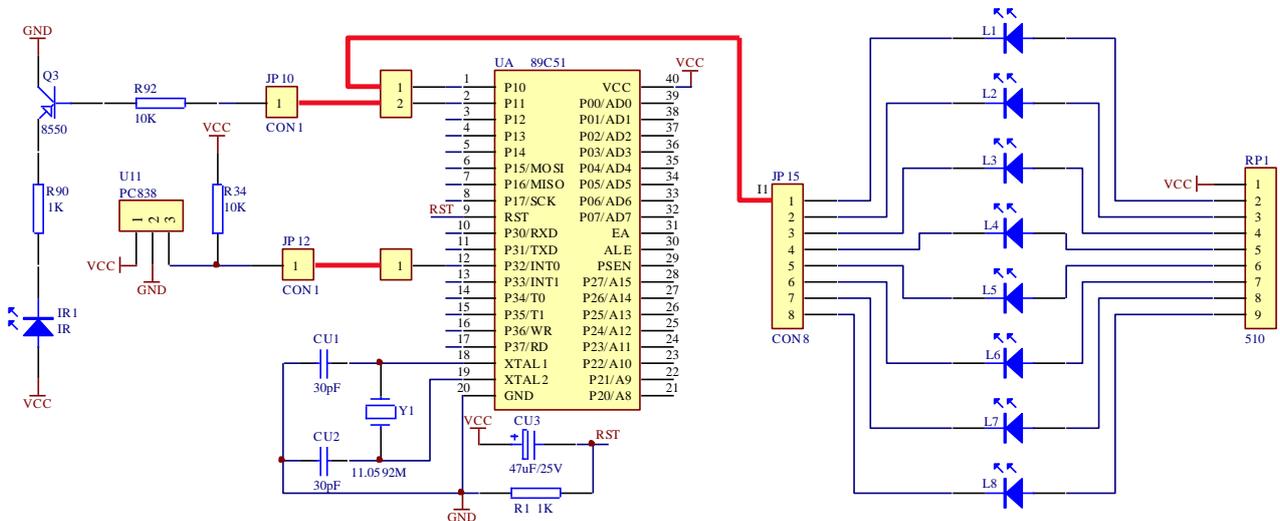


图 3-24-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-24-2 所示

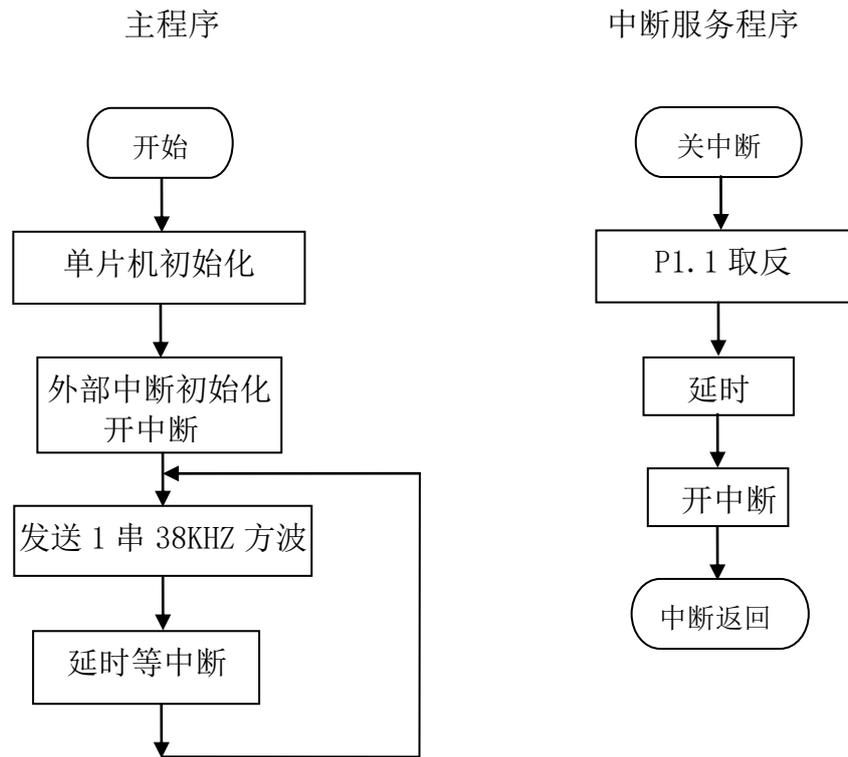


图 3-24-2

六、实验步骤

- 1、用 1 根 1P 杜邦线将单片机 P10 口接底板“单色流水灯”区 JP15 的 I1 引脚；
- 2、用 1 根 1P 杜邦线将单片机 P11 接底板“红外模块”区 JP10；
- 3、用 1 根 1P 杜邦线将单片机外部中断 P32 口接底板“红外模块”区 JP12；
- 4、在 KEIL 软件中调试 fsjs 工程文件。用 STC 软件下载 fsjs.HEX 程序；
- 5、实验现象：若红外发射能被接收头正确的接收，二极管 L1 将以一定的频率闪烁。使用小纸板遮掩物或者将发射二极管的连接杜邦线拔掉，发光二极管将不再闪烁（亮或灭）。

实验二十五 红外解码测试

一、实验目的

- 1、掌握红外遥控的基本原理；
- 2、通过实验了解红外通信协议。

二、实验内容

编制程序，将遥控器上的按键编码值用 1602 液晶显示器显示出来。

三、实验说明

遥控器如图 3-25-1 所示



图 3-25-1

硬件准备：1P 杜邦线 1 根，8P 杜邦线 1 根，3P 杜邦线 1 根，红外遥控器 1 个。

四、实验原理图

如图 3-25-2 所示

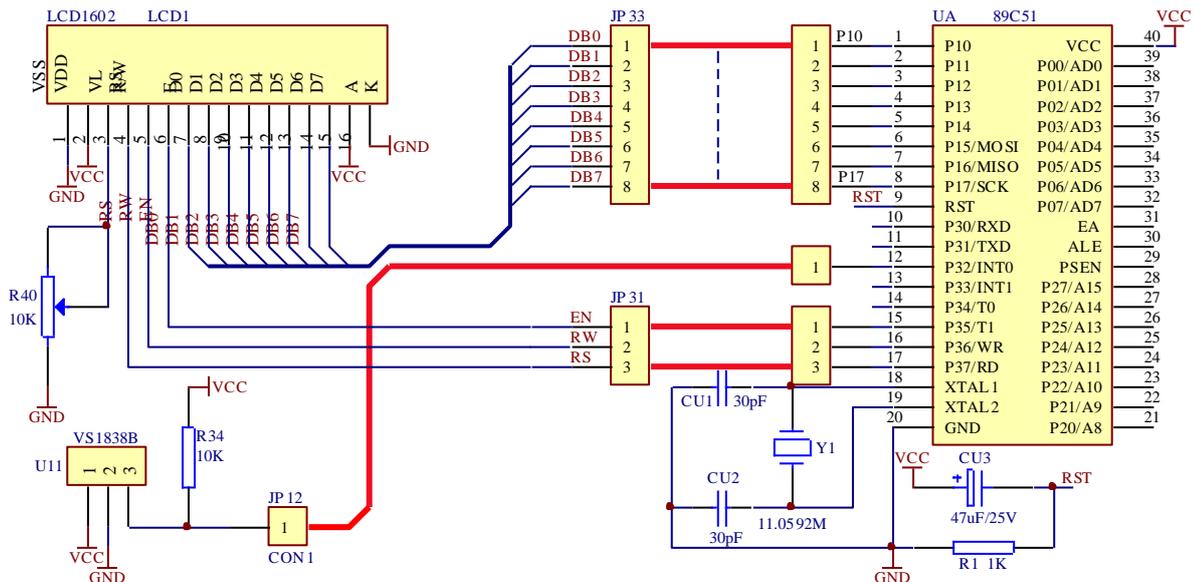


图 3-25-2

五、实验程序与框图

源程序见随机光盘

流程图见图 3-25-3 所示

六、实验步骤

- 1、用 1 根 1P 杜邦线将单片机外部中断 P32 口接底板“红外模块”区 JP12;
- 2、用 3P 杜邦线将单片机 P35~P37 接底板“LCD1602 液晶”区 JP31。具体接法为：P35-EN, P36-RW, P37-RS;
- 3、用 8P 杜邦线将单片机 P1 口接底板“LCD1602 液晶”区 JP33。具体接法为：P10-DB0, P11-DB1, P12-DB2, P13-DB3, P14-DB4, P15-DB5, P16-DB6, P17-DB7;
- 4、在 KEIL 软件中调试 hwjm 工程文件。用 STC 软件下载 hwjm.HEX 程序;
- 5、实验现象：遥控器对着红外接收头按下数字键，在 LCD1602 上显示对应红外遥控器上按键的编码。第一行“dvcc.com.cn”，第二行“CODE + 遥控编码”。

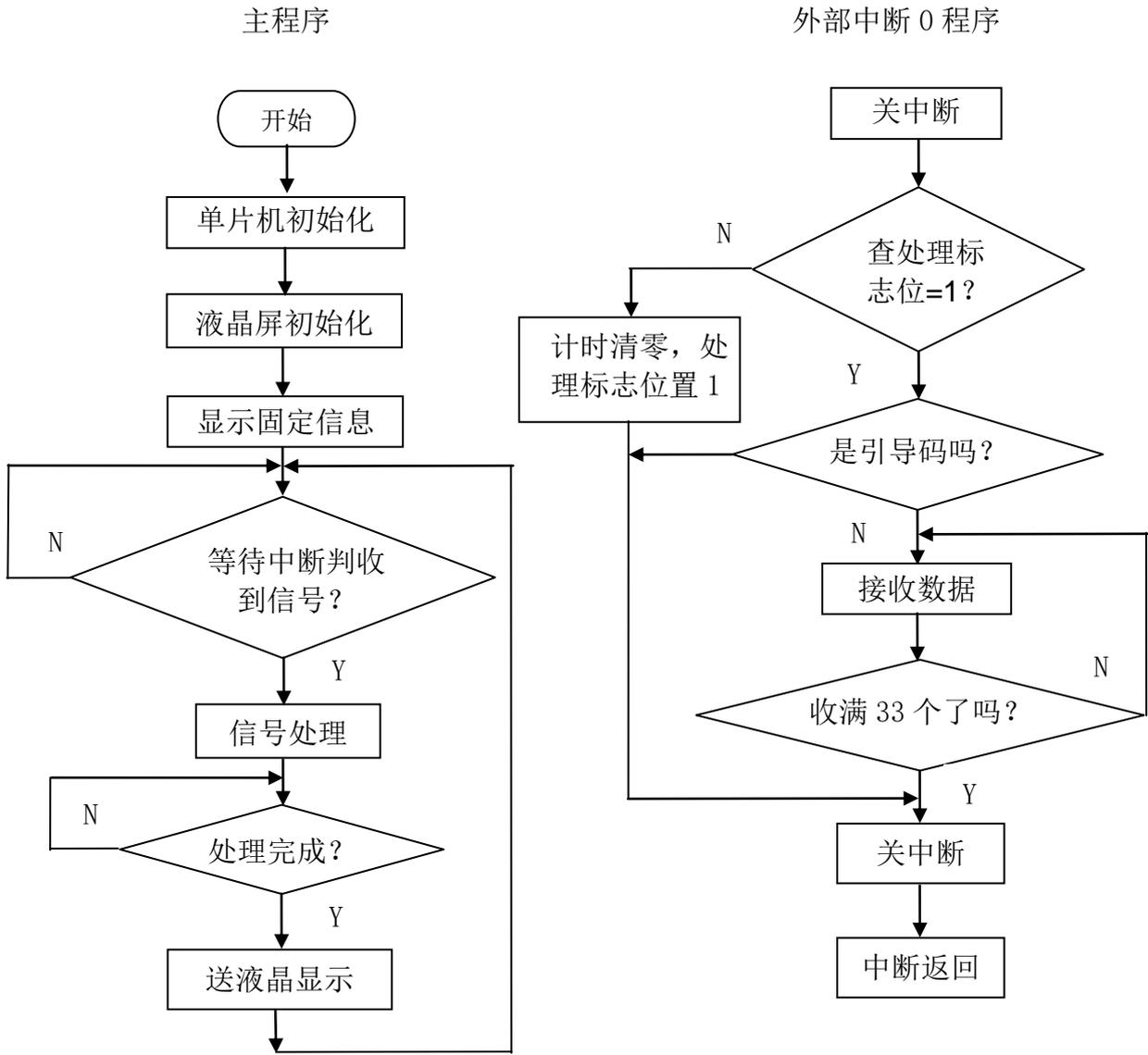


图 3-25-3

实验二十六 SD 卡模块测试

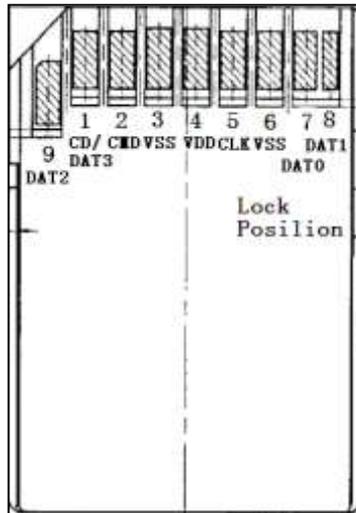
一、实验目的

了解掌握 SD 卡的工作模式和 SPI 模式下的编程读写 SD 卡。

二、实验内容

编写程序用 SPI 方式写 256 个字节的数据，然后读出写的数据块。

三、实验说明



1、SD 卡总线模式引脚定义

引脚号	引脚名	类型	SD 卡描述
1	CD/DAT3	I/O	SD 卡检测和数据脚 3
2	CMD	I/O	命令和返回参数
3	VSS1	S	电源地
4	VDD	S	电源
5	CLK	I	时钟
6	VSS2	S	电源地
7	DAT0	I/O	数据脚 0
8	DAT1	I/O	数据脚 1
9	DAT2	I/O	数据脚 2

2、SPI 模式引脚定义

引脚号	引脚名	类型	SD 卡描述
1	CS	I	片选信号（低有效）
2	DATAIN	I	数据输入
3	VSS1	S	电源地

4	VDD	S	电源
5	CLK	I	时钟
6	VSS2	S	电源地
7	DATAOUT	O	数据输出
8	RSV	I/	保留
9	RSV	I	保留

3、SD 卡在上电初期自动进入 SD 总线模式，在此模式下向 SD 卡发送复位命令 CMD0。如果 SD 卡在接收复位命令过程中 CS 低电平有效，则进入 SPI 模式，否则工作在 SD 总线模式。对于不带 SPI 串行总线接口的 STC89C52 单片机来说，用软件来模拟 SPI 总线操作的具体做法是：将（模拟 CLK 线）的初始状态设置为 1，而在允许接收后再置为 0。这样，MCU 在输出 1 位 SCK 时钟的同时，将使接口芯片串行左移，从而输出 1 位数据至单片机的（模拟 MISO 线），此后再置 CLK 为 1，使单片机从（模拟 MOSI 线）输出 1 位数据（先为高位）至串行接口芯片。至此，模拟 1 位数据输入输出便完成。此后再置 CLK 为 0，模拟下 1 位数据的输入输出，依此循环 8 次，即可完成 1 次通过 SPI 总线传输 8 位数据的操作。

4、硬件准备：写好测试内容的 SD 卡 1 个，1P 杜邦线 4 根。

四、实验原理图

如图 3-26-1 所示

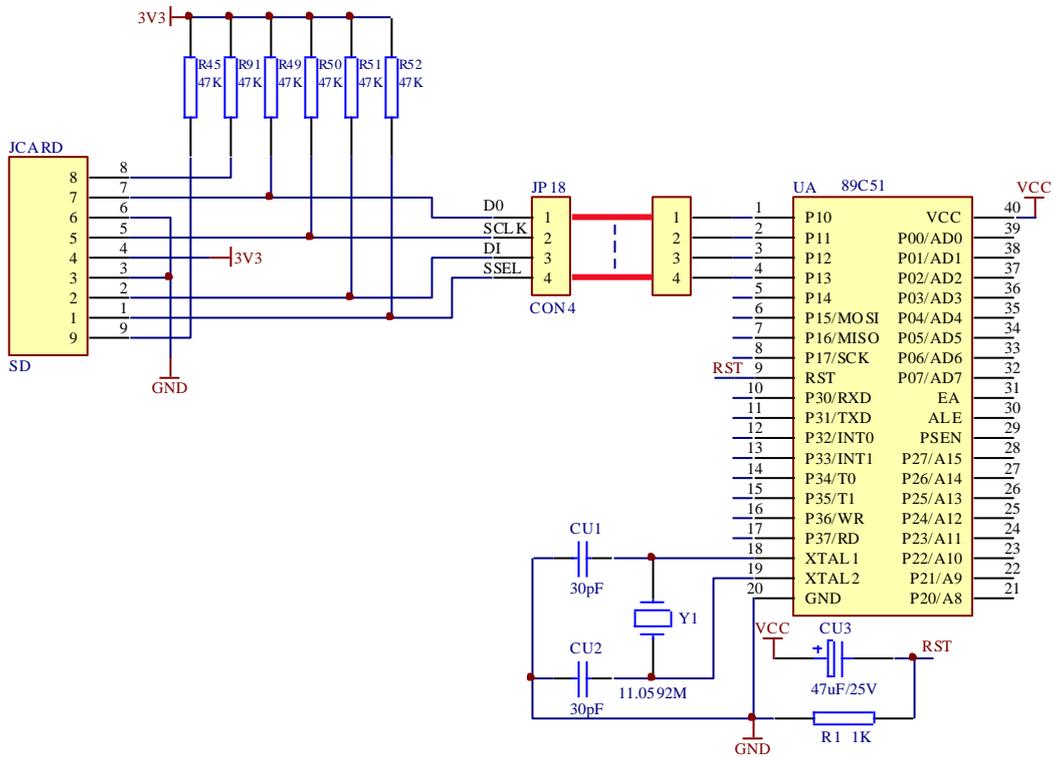


图 3-26-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-26-2 所示

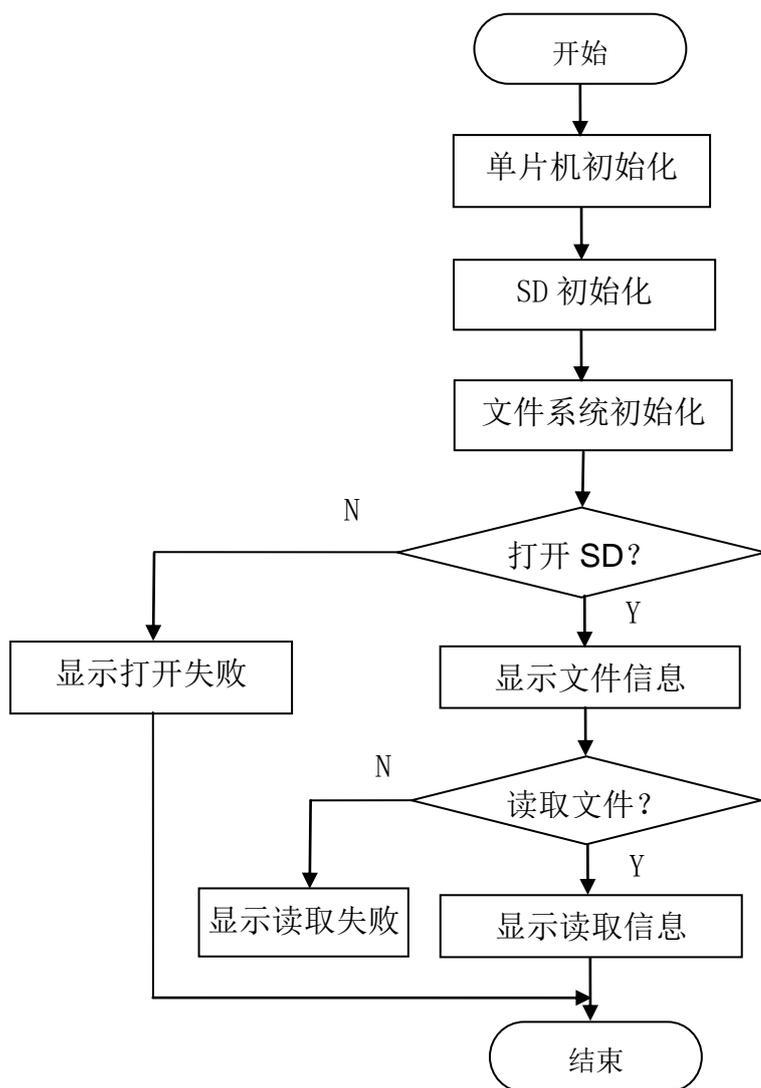


图 3-1-2

六、实验步骤

- 1、将 SD 卡插到底板“SD 卡”区的卡座里，SD 卡有金属引脚一面朝下。
- 2、用 2 根 2P 杜邦线将单片机 P10~P13 口与底板“SD 卡”区的 JP18 相连，分别为：P10-D0，P11-SCLK，P12-DI，P13-SSEL；
- 3、在 KEIL 软件中调 SD 工程文件。用 STC 软件下载 SD.HEX 程序；
- 4、实验现象： 在串口助手中显示测试内容。

友情提示：

1) 请自备 SD 卡，SD 卡格式化为 FAT32 格式，根目录放一文件名为 test.txt 文件，里面写上测试文字。

实验二十七 PCF8591AD/DA 模块测试

一、实验目的

了解掌握 PCF8591 的工作原理。

二、实验内容

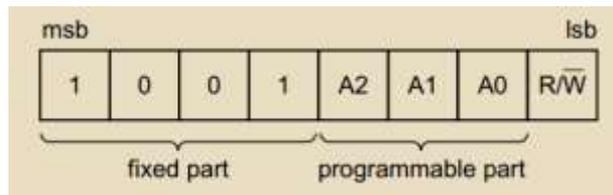
编写程序，用模数转换进行采集，采集的电压值在液晶显示器上显示，同时将这个电压值通过数模转换，转换成模拟量送发光二极管 VL3 显示。

三、实验说明

PCF8591 是在一个芯片上集成有数模转换和模数转换两种电路，单电源供电、低功耗、8-bit CMOS 数据获取器件。PCF8591 具有 4 个模拟输入、1 个模拟输出和 1 个串行 I²C 总线接口。PCF8591 的 3 个地址引脚 A0, A1 和 A2 可用于硬件地址编程，允许在同个 I²C 总线上接入 8 个 PCF8591 器件，而无需额外的硬件。在 PCF8591 器件上输入输出的地址、控制和数据信号都是通过双线双向 I²C 总线以串行的方式进行传输。

PCF8591 的通信接口是 I²C，那么编程肯定是要符合这个协议的。单片机对 PCF8591 进行初始化，一共发送三个字节即可。

第一个字节：器件地址字节



其中 7 位代表地址，1 位代表读写方向。地址高 4 位固定是 1001，低三位是 A2, A1, A0，这三位我们电路上都接了 GND，因此也就是 000。

第二个字节：器件控制字节



控制字节的第 6 位是 DA 使能位，这一位置 1 表示 DA 输出引脚使能，会产生模拟电压输出功能。

第 4 位和第 5 位可以实现把 PCF8591 的 4 路模拟输入配置成单端模式和差分模式，是配置 AD 输入方式的控制位，有单端模式和差分模式。

控制字节的第 2 位是自动增量控制位，自动增量的意思就是，比如我们一共有 4 个通道，当我们全部使用的时候，读完了通道 0，下一次再读，会自动进入通道 1 进行读取，不需要我们指定下一个通道。

注意：由于 A/D 每次读到的数据，都是上一次的转换结果，所以在使用自动增量功能的时候，要特别注意，当前读到的是上一个通道的值。

控制字节的第 0 位和第 1 位就是通道选择位了，00、01、10、11 代表了从 0 到 3 的

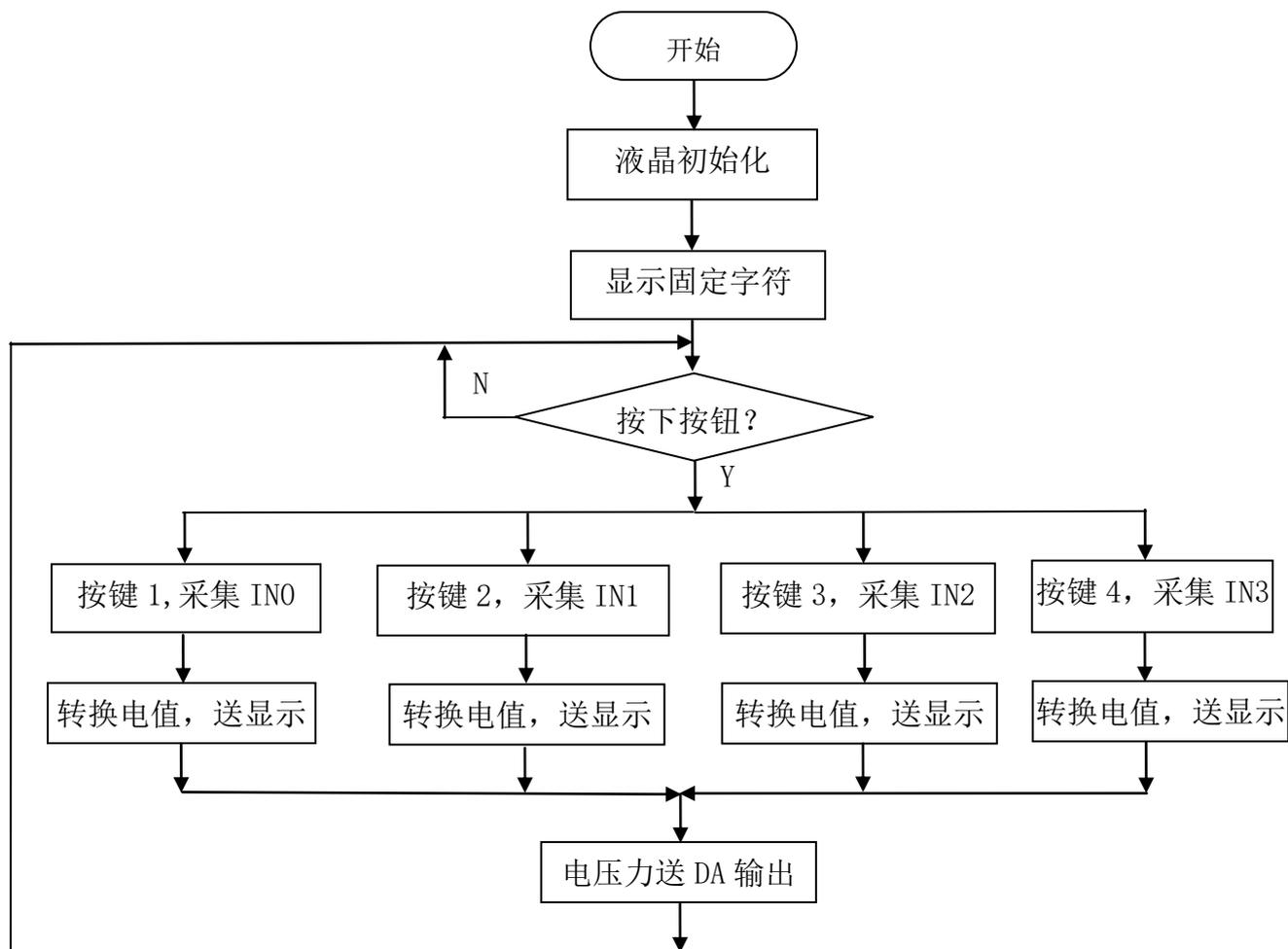


图 3-27-2

六、实验步骤

1、用 1 根 2P 杜邦线将单片机 P10~P11 接底板“AD/DA 转换”区 JP68，具体为：SCL-P10，SDA-P11；

2、用 1 根 4P 杜邦线将单片机 P12~P15 接底板“独立按键”区 JP69，具体为：K1-P12，K2-P13，K3-P14，K4-P15；

3、用 1 根 3P 杜邦线将单片机 P20~P22 口与底板“LCD1602 液晶”区 JP31 相连。具体为：P20-RS，P21-RW，P22-EN；

4、用 1 根 8P 杜邦线将单片机 P0 口与底板“LCD1602 液晶”区 JP33 排针相连，具体为：P00-DB0，P01-DB1，P02-DB2，P03-DB3，P04-DB4，P05-DB5，P06-DB6，P07-DB7；

5、用短路片短接“AD/DA 转换”区 J63 上的 I0, I1, I2, I3；

6、在 KEIL 软件中调试 PCF8591 工程文件。用 STC 软件下载 23.HEX 程序；

7、实验现象：分别按 K1-K4 键，采样 4 个通道的模拟量，在 LCD 上显示出来（比如 2.5V），调节电位器，LCD 上显示的电压值会变化。由于四个通道是接在一起的，所以四个通道的显示值会相同。在将 4 个通道的模拟量转换成数字量后，再将这个数字量转换成模拟量，通过 AOUT 端输出模拟量，在 VL3 灯上显示。电压越大 VL3 显示灯越亮；LED 显示电压越小，VL3 灯越暗。

实验二十八 自主脉冲发生器测试

一、实验目的

了解掌握 555 定时器的工作原理。

二、实验内容

编写程序采用中断方式检测脉冲信号，然后通过 P1 口检测脉冲频率。

三、实验说明

通过 555 定时器产生方波，频率范围大约在 6Hz 和 9Hz 之间，通过两个电位器调节频率和占空比。

硬件准备：1P、8P 杜邦线各一根。

四、实验原理图

如图 3-28-1 所示

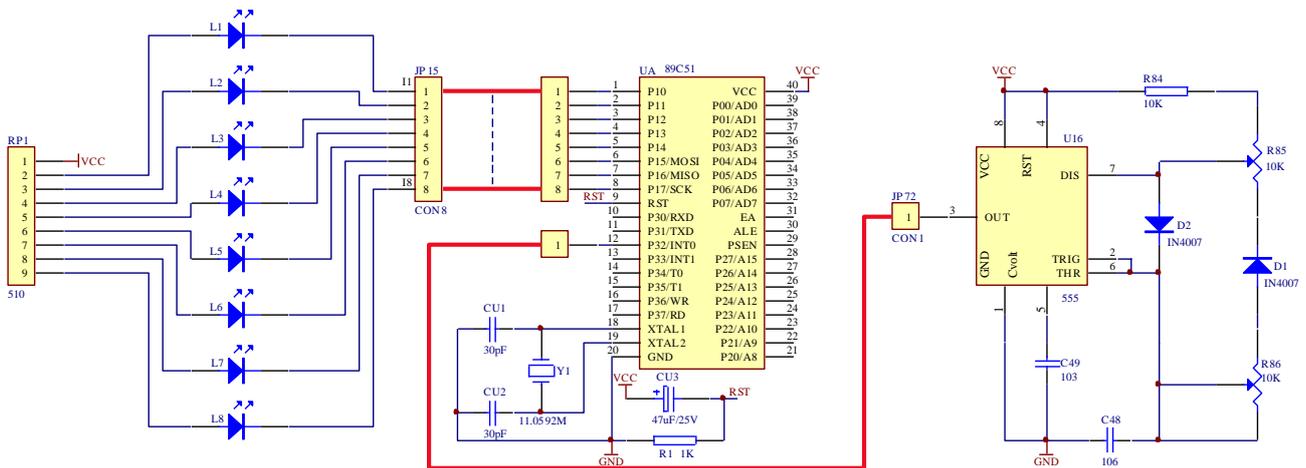


图 3-28-1

五、实验程序与框图

源程序见随机光盘

流程图见图 3-28-2 所示

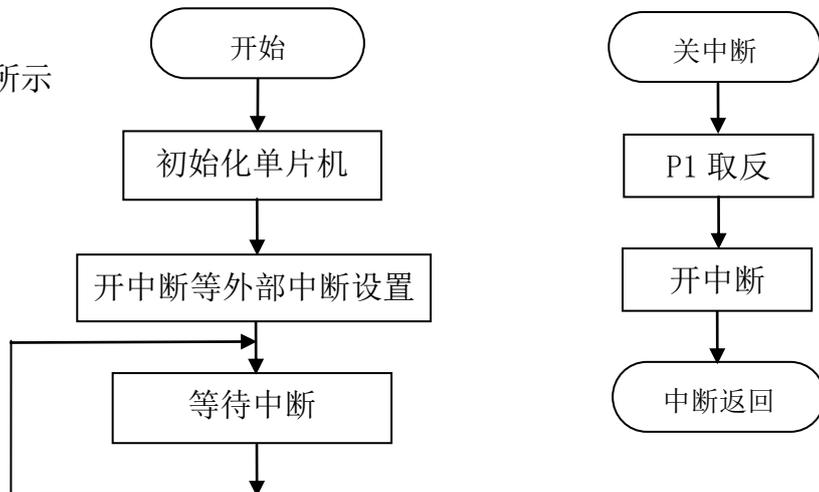


图 3-28-2

六、实验步骤

- 1、单片机 P1 口接“单色流水灯”区 JP15，具体接法为：P10-I1，P11-I2，P12-I3，P13-I4，P14-I5，P15-I6，P16-I7，P17-I8；
- 2、P32 口接“自主脉冲发生器”的输出口 JP72；
- 3、在 KEIL 软件中调试 MC 工程文件。用 STC 软件下载 MC.HEX 程序；
- 4、实验现象：每隔一段时间，P1 口接的 LED 灯亮灭变化一次。

第四章 C 语言开发平台 keil c51

基于 51 单片机的 C 语言，即 C51 语言，在语法结构上跟通用的 C 语言是一致的。所不同的是 C51 语言运行的环境是 51 单片机，而 C 语言运行的环境是个人计算机。由于运行环境的不同，C51 语言在数据结构和处理方面跟 C 语言有一定的区别。

下面给出一个完整的 C51 语言的程序代码。其功能是实现 51 单片机 P0.0~P0.7 这 8 个端口连续输出 8 个不同的值。

```
#include <reg51.h>           //
Void main(void)
{
    unsigned char temp = 0;
    for(temp = 0;temp < 8)
    {
        P0 = temp;
    }
    While(1);
}
```

从上面这段程序中可以看出，C51 语言和 C 语言结构基本一致，都包含头文件、主函数等。程序中用到的赋值语句、循环语句、函数结构都与 C 语言完全一致。因为 C51 语言运行于单片机系统，P0 为单片机的硬件资源，C51 可以对端口直接操作，这是区别于 C 语言的地方。

1、 μ Vision 集成开发环境简介

目前，对于 51 单片机的 C 语言开发，一般采用德国 Keil 公司的 Keil C51 集成开发环境。Keil C51 支持 51 系列单片机及其兼容单片机的编程开发，Keil C51 的集成开发环境是 μ Vision 系列。 μ Vision3 提供了丰富的库函数和编译工具，能够对 51 系列单片机以及与 51 系列单片机兼容的大部分类型的单片机进行编程设计。 μ Vision3 除了支持 C51 语言开发外，它也支持汇编语言的开发。

Keil C51 是一款功能强大的 C 语言编译器，它能够生成形式简洁、效率极高的程序代码，其产生的代码质量基本接近于汇编程序代码。

Keil C51 是一个基于 windows 系统的 51 单片机集成开发环境，不仅支持程序代码的编写，同时还支持程序代码的编译、仿真，该软件界面友好、易学易用，因此受到广大 51 单片机开发者的青睐。

μ Vision 系列集成开发环境的版本

目前的最新版本是 μ Vision4，版本号为 v9.00。本书以目前常用的 μ Vision3 版本进行实例开发。

μ Vision3 集成开发环境对计算机系统的要求

μ Vision3 集成开发环境对计算机的软件和硬件配置要求不高，一般的计算机系统都能够满足运行 μ Vision3 集成开发环境。但是为了达到比较好的运行效果，建议计算机的软硬件配置适当高一些。 μ Vision3 集成开发环境对计算机系统的最低要求如下：

- 内存不小于 16MB；
- 硬盘的剩余空间不小于 60MB；
- Windows95 及以上操作系统。

2、 μ Vision3 集成开发环境的安装

这里以 Keil C51 v8.02 版本为例，介绍 μ Vision3 集成开发环境的安装过程。

第一步：双击安装包中的 setup.exe 文件，弹出如图 1.1 所示的安装程序对话框，点击“Next”按钮，开始安装进程。

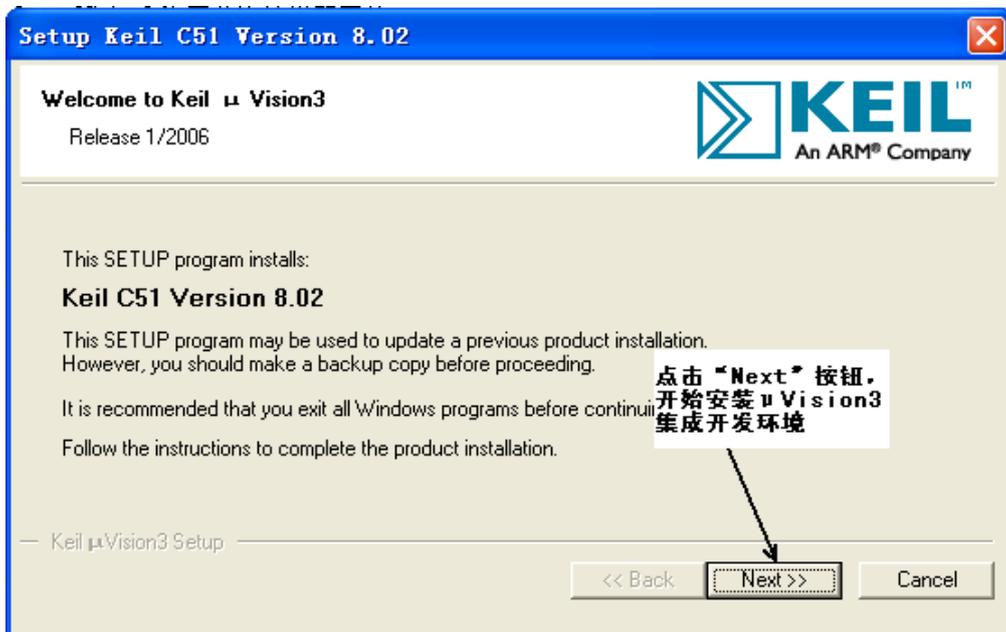


图 1.1 μ Vision3 的安装启动界面

第二步：如图 1.2 所示，选择接受使用协议，点击“Next”进入下一步操作。



图 1.2 使用协议选择对话框

第三步：如图 1.3 所示，选择软件的安装目录，根据实际情况选择合适的安装目录，再次

选择系统默认的安装目录。点击“Next”进入下一步操作。

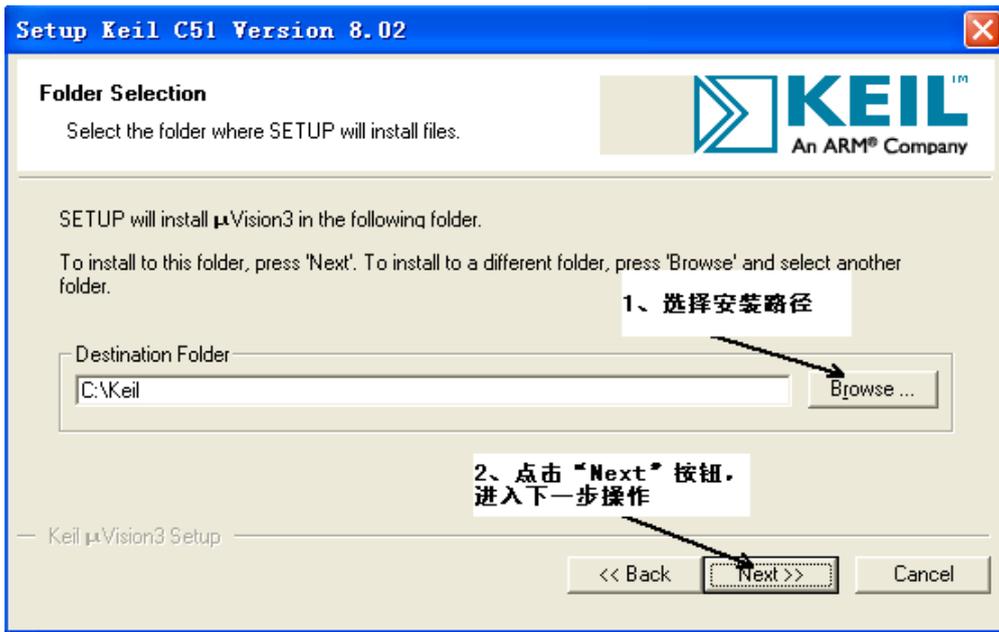


图 1.3 选择安装目录

第四步: 如图 1.4 所示, 根据提示, 输入用户姓名、公司名称、电子邮箱等信息, 点击“Next”进入下一步操作。

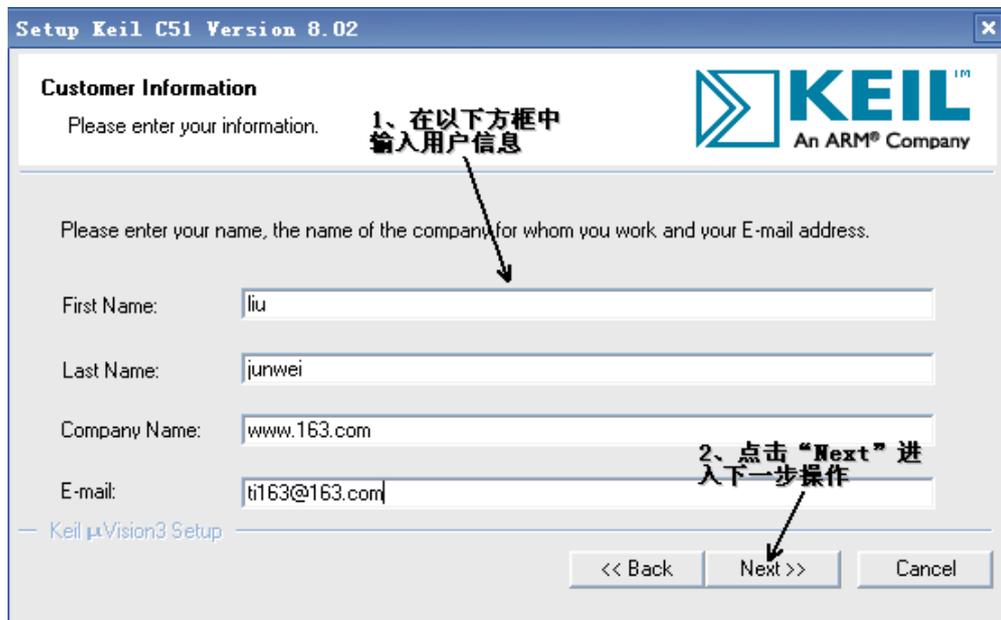


图 1.4 输入用户信息

第五步: 如图 1.5 所示, 开始正式安装 μVision3 集成开发环境, 同时以蓝色小方块的形式显示软件的安装进度。

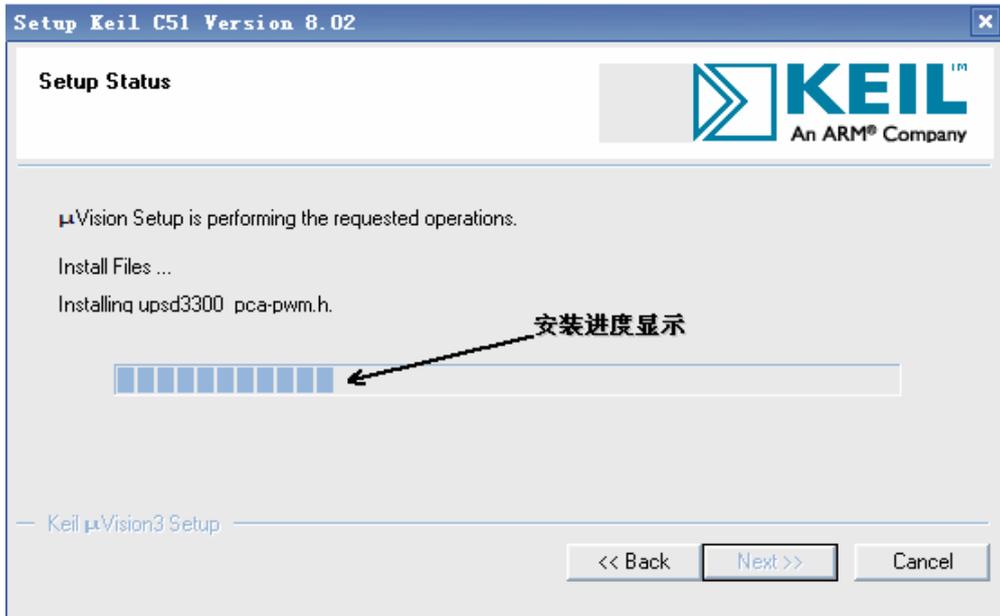


图 1.5 安装进度显示

第六步: 如图 1.6 所示，用户可选择查看版本信息和项目区设置，也可不选这两个选项，点击“Finish”按钮完成安装。



图 1.6 完成安装

第七步: 进行软件注册，如图 1.7 所示，首先打开 µVision3 集成开发环境软件，进入“File”菜单，选择“License Management...”（注册管理）选项。

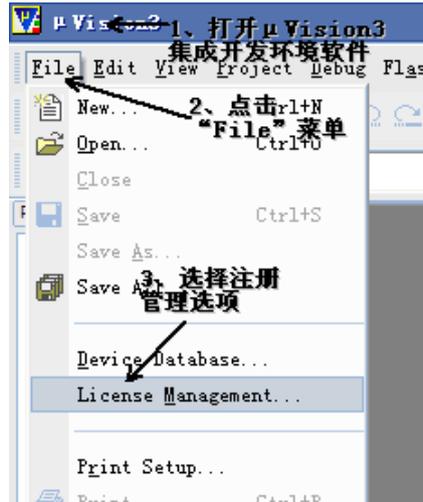


图 1.7 进入注册管理页面

第八步：完成软件注册，如图 1.8 所示，用户图中提示的步骤将购买的序列号输入相应位置，完成整个软件的注册。

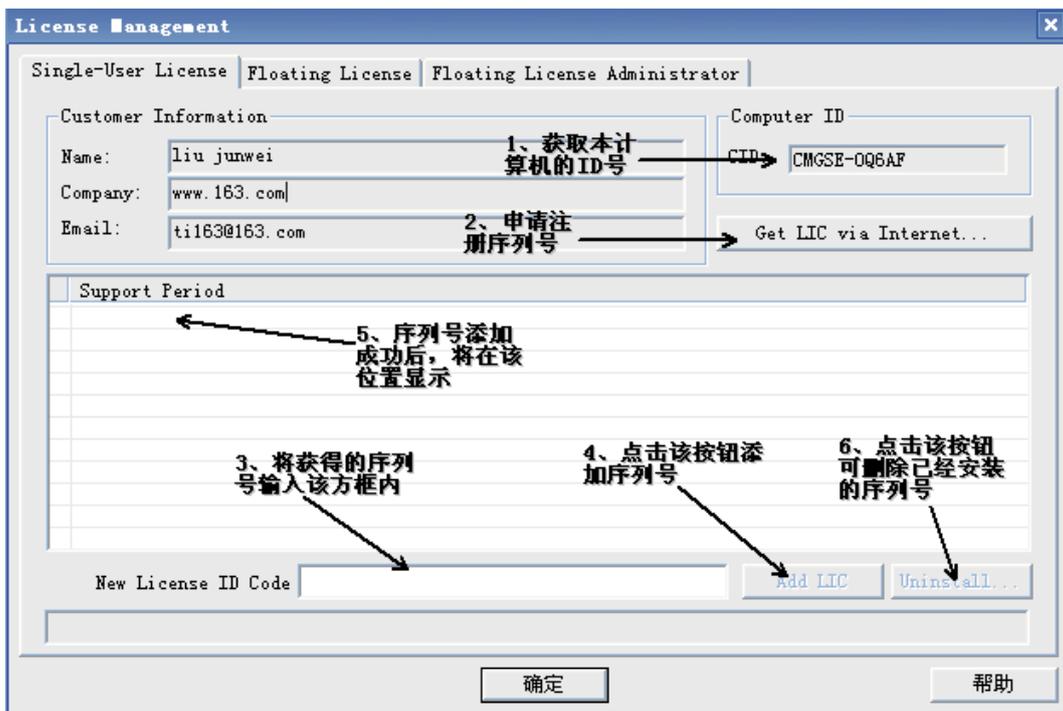


图 1.8 填写注册信息

3、基于 Keil C51 的第一个 C 语言程序项目的建立

接下来通过一个简单的例子演示如何使用 μ Vision3 集成开发环境创建一个 51 单片机 C51 程序项目。

在 μ Vision3 集成开发环境下建立 C51 程序项目的过程如下：

- (1) 建立一个工程项目，选择芯片类型；

- (2) 建立 C 语言源程序文件；
- (3) 用项目管理器生成各种应用文件；
- (4) 检查并修改源程序文件中的错误；
- (5) 编译连接程序项目文件，并生成单片机能够识别的烧录文件。

μVision3 集成开发环境建立 C51 程序项目的具体过程如下：

第一步：双击桌面上的 μVision3 集成开发环境快捷方式(如图 1.9 所示)，打开 μVision3 集成开发环境，界面如图 1.10 所示，与一般的 Windows 应用程序一样，μVision3 集成开发环境的界面由菜单栏、工具栏、工作区、项目管理窗口、输出窗口等部分组成，在不同的编译环境下，还有其它一些调试或观察窗口。



图 1.9 μVision3 集成开发环境快捷方式

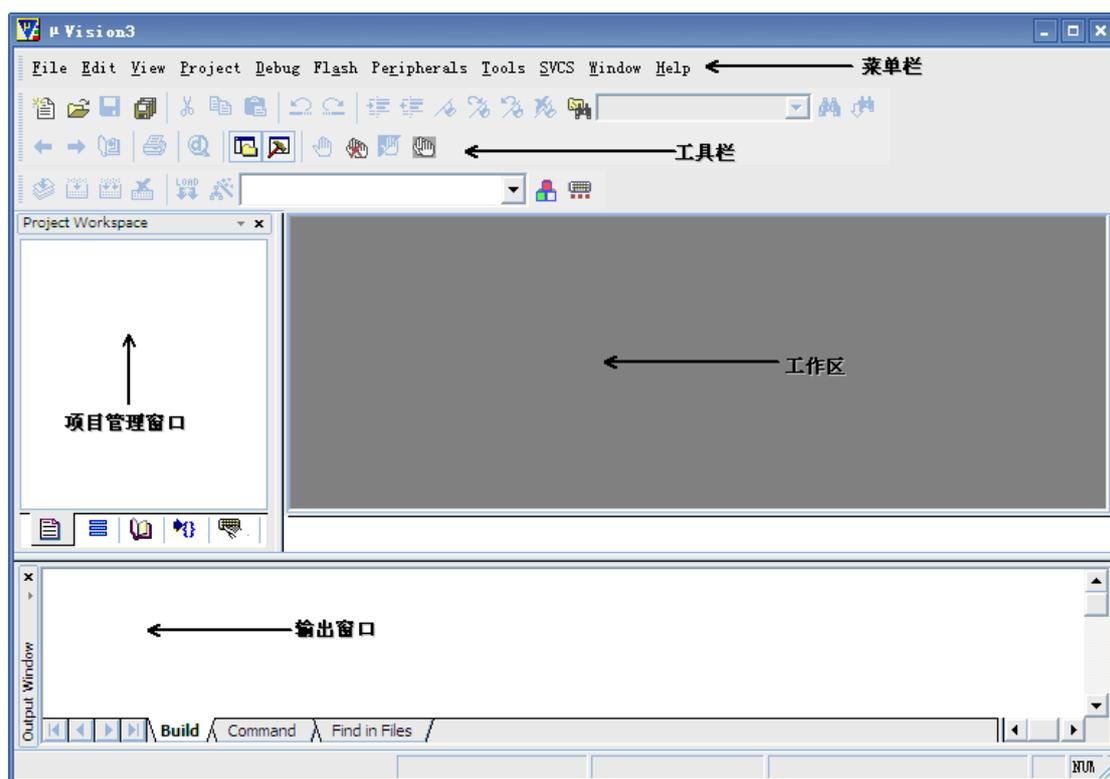


图 1.10 μVision3 集成开发环境的界面

第二步：建立一个工程项目，选择芯片类型。如图 1.11 所示，在菜单栏中选择“Project”→“New Project...”选项，弹出创建新项目对话框，如图 1.12 所示，首先在“保存在”对话框中选择项目保存的路径，接着在“文件名”对话框中输入项目名称，然后在“保存类型中选择“Project Files(*.uv2)”，最后单击“保存”按钮，将该新建项目保存。

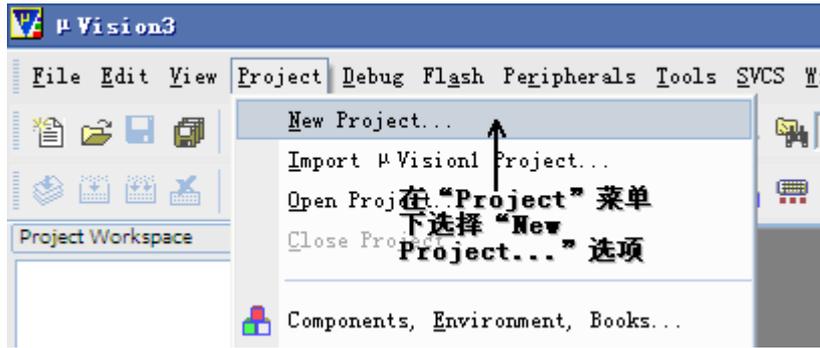


图 1.11 选择新建项目选项

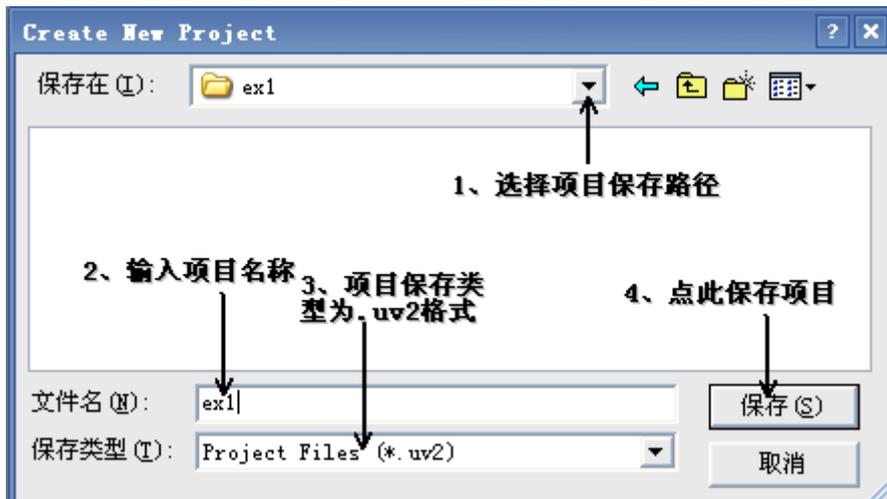


图 1.12 新建并保存项目

点击“保存”按钮后，将弹出芯片类型选择对话框，如图 1.13 所示。在“Data base”项中选择“Atmel”点击其前面的+号，在列出的芯片类型中选“AT89C51”，然后点击确定，完成芯片类型的选择。

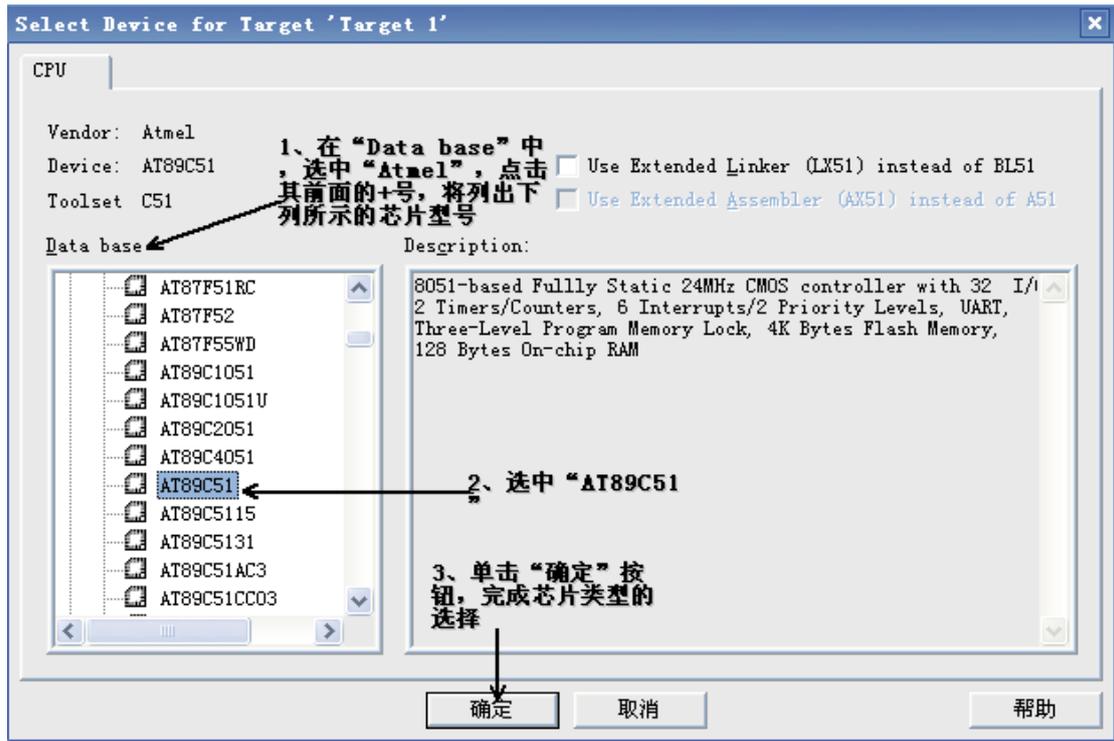


图 1.13 芯片类型的选择

此时弹出一个提示信息框，如图 1.14 所示。该信息提示是否将 8051 的起始代码添加到项目中，一般选择“是”，添加起始代码；也可以选择“否”，不添加起始代码，在以后需要时再手动添加起始代码。这里选择“是”，添加起始代码。



图 1.14 提示信息

至此完成了新项目的建立，只是由于此时还未添加源文件，所以该项目是一个空项目，如图 1.15 所示。

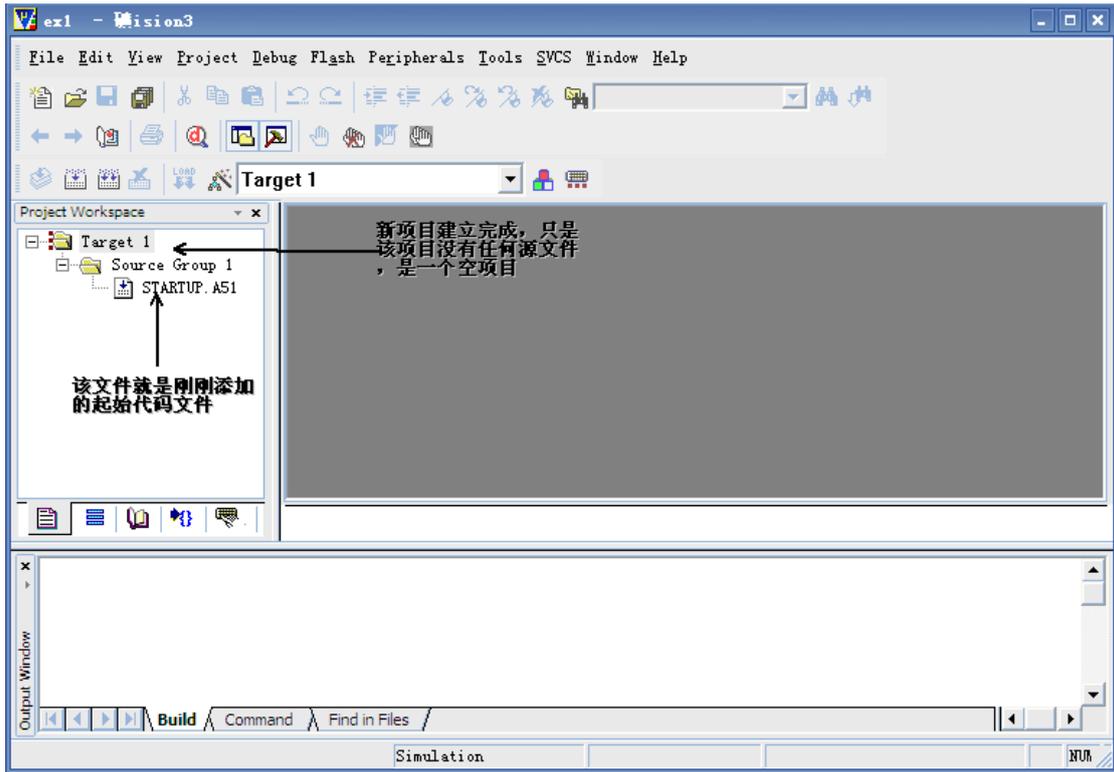


图 1.15 新项目建立完毕

第三步： 创建源文件。项目建立完毕后，就要开始进行源文件的设计，这边涉及到项目的核心内容。具体操作步骤如下：

在“File”菜单下选择“New...”选项，如图 1.16 所示。

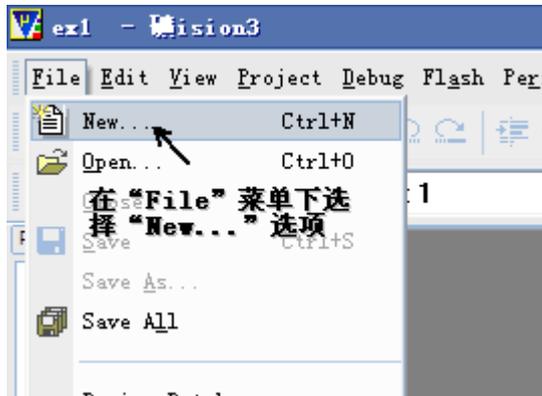


图 1.16 创建源文件

此时在文本编辑区弹出一个新的文本编辑窗口，如图 1.17 所示。用户可以在该窗口中输入程序代码。



图 1.17 新建的文本编辑窗口

在文本编辑窗口中输入如下的程序代码。如图 1.18 所示。程序输入完成后，点击工具栏中的“保存”快捷按钮，将该文件保存在上一步建立的文件夹中，注意文件保存类型必须是.c 类型。

```
#include <reg51.h>
void delay(void);
void main(void)
{
    P0 = 0x55;
    delay();
    P0 = 0xaa;
    delay();
    while(1);
}
void delay(void)
{
    unsigned int i, j;
    for(i = 0; i < 500; i++)
    {
        for(j = 0; j < 500; j++);
    }
}
```

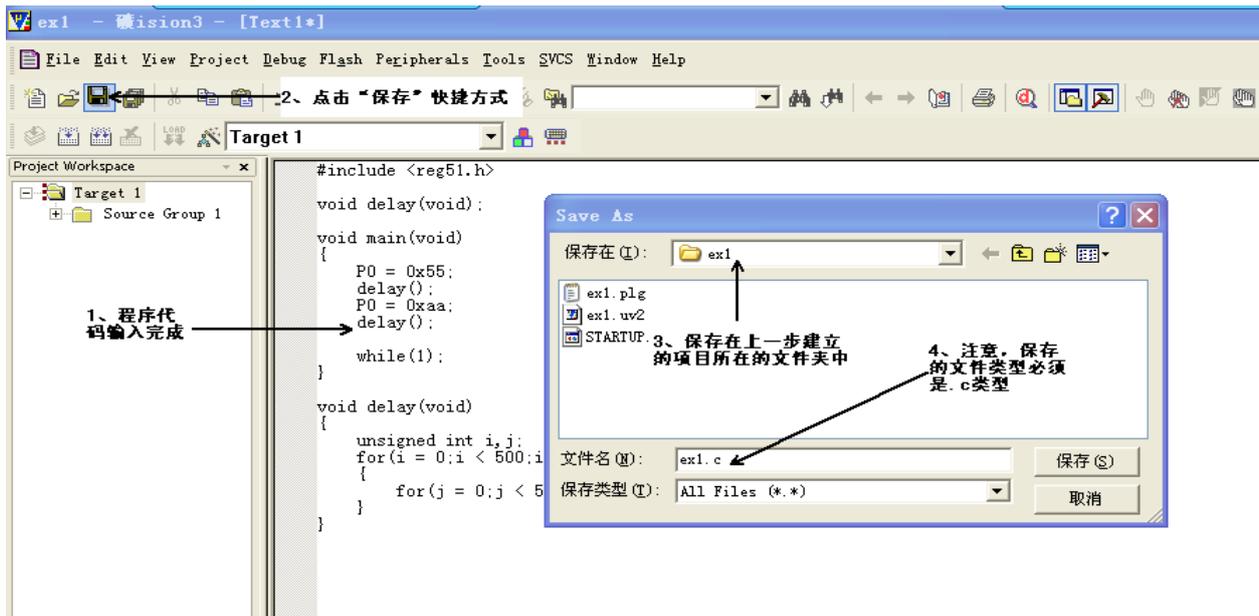


图 1.18 建立并保存源程序文件

第四步：添加源程序文件到当前项目组。如图 1.19 所示。单击项目管理窗口中“Target 1”前面的“+”号，出现“Source Group 1”后，再单击选中该项，然后单击鼠标右键，在弹出的快捷菜单中选中“Add Files to Group ‘Source Group 1’”，弹出添加文件对话框，如图 1.20 所示。

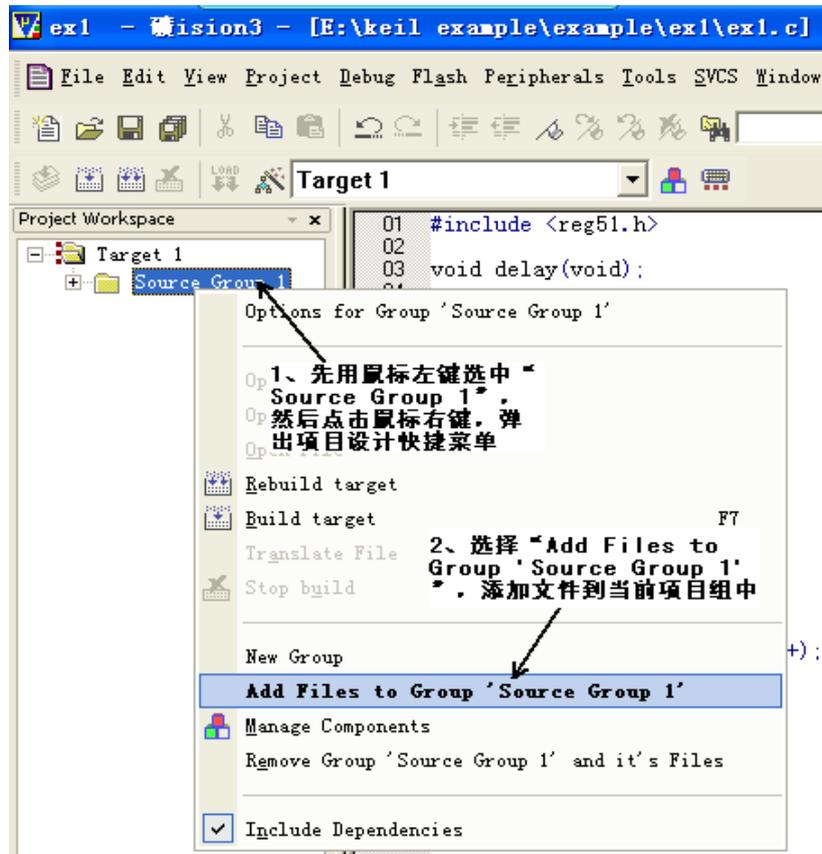


图 1.19 选择添加文件选项

在弹出的添加文件对话框中，选择将要添加到项目中的文件，然后单击“Add”按钮，将选中的文件添加到项目中。注意在添加完文件后，该窗口并不会关闭，用户可以继续添加文件到项目中，完成全部文件的添加后，单击“Close”按钮关闭该对话框。

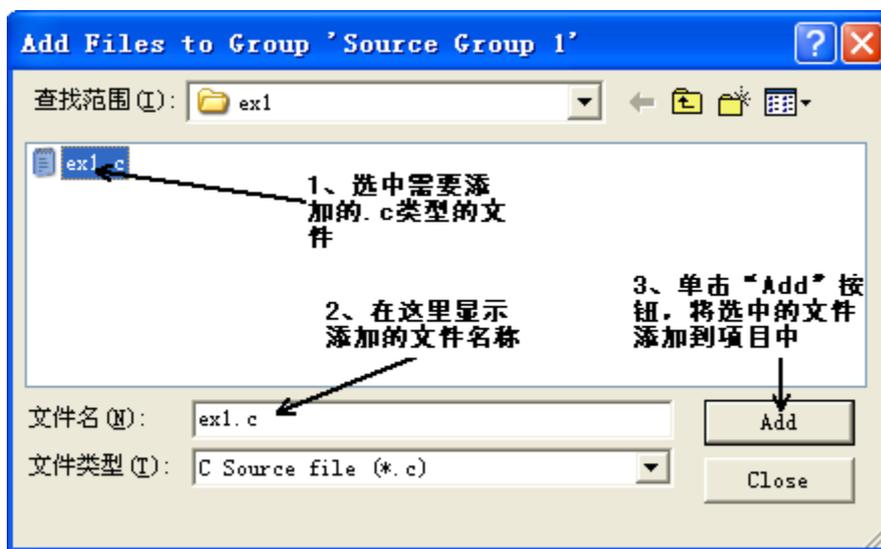


图 1.20 将选择的文件添加到项目中

第五步：项目编译选项设置。鼠标左键单击项目管理窗口中“Target 1”选中该项，然后单击鼠标右键，在弹出的快捷菜单中选中“Options for Target ‘Target 1’”，如图 1.21 所示。

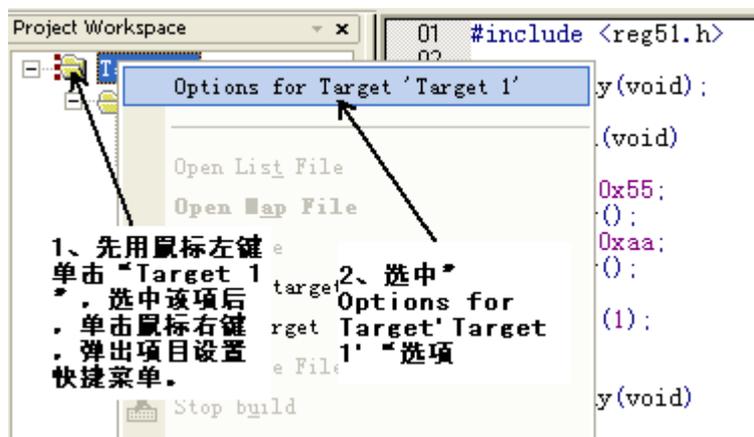


图 1.21 项目设置选择

在弹出的项目设置对话框中，首先输入晶振频率，如图 1.22 所示。

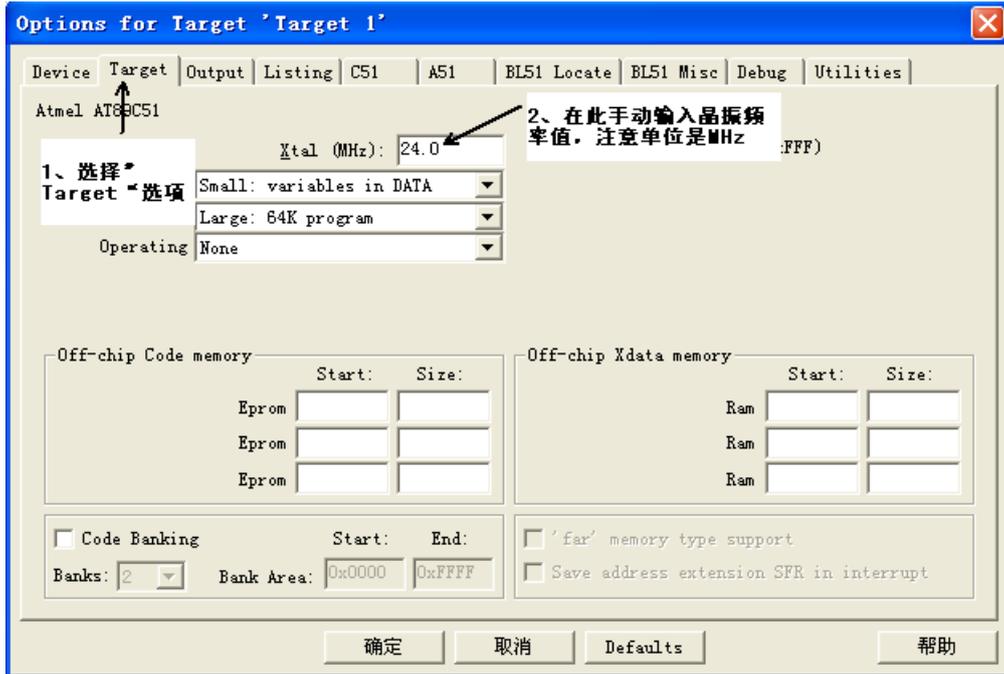


图 1.22 晶振频率选项

然后选择输出文件类型，程序编译后最终是要烧录到单片机中去执行的，所以必须要生成能够在单片机上运行的文件，该文件类型一般是 .HEX 类型，如图 1.23 所示，在“Output”选项中选择复选框“Create Hex File”。

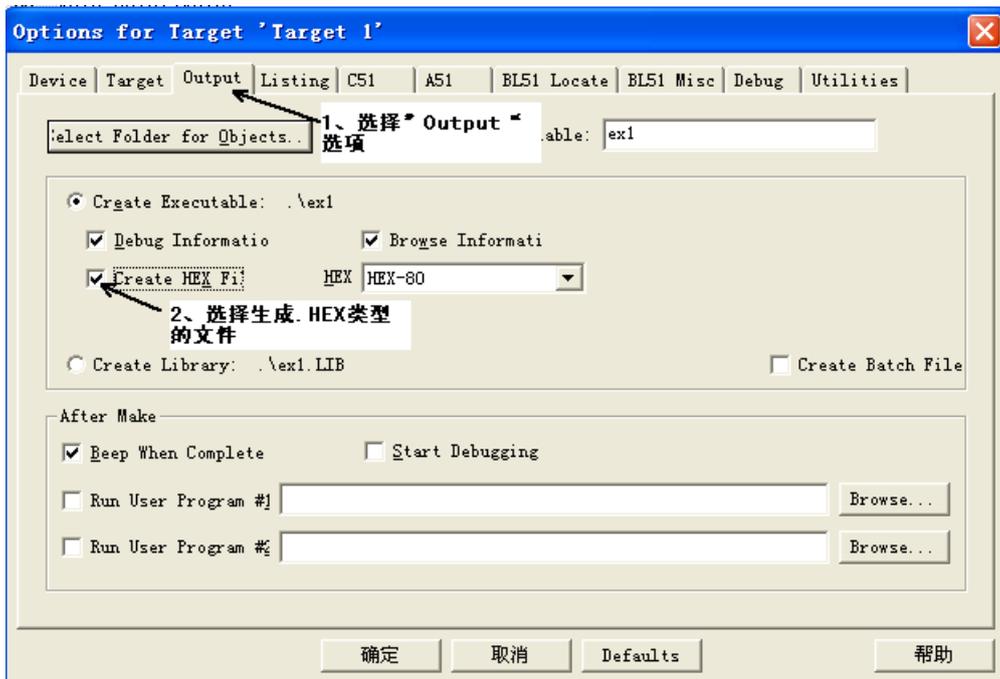


图 1.23 选择输出 .HEX 类型的文件

最后选择仿真选项设置，如图 1.24。用户可根据实际情况选择使用软件仿真模式或者硬件仿真模式。使用硬件仿真模式时，还要注意选择相应的仿真器类型。



图 1.24 仿真选项设置

点击确定按钮，完成项目的设置。从而完成新建项目、程序代码输入、在项目中添加文件、设置项目的属性等工作，接下来就可以对该项目进行编译和错误检查工作。如图 1.25 所示，在菜单栏中选择“Project”菜单，在弹出的菜单选项中选择“Rebuild all target files”选项，对项目进行编译。



图 1.25 项目编译选项

编译结束后，在输出窗口显示编译结果，如果编译中出现问题，将提示错误或者警告的类型及位置。如图 1.26 所示。用户可以根据错误或警告提示重新修改程序代码，并重新编译，直到没有错误或警告为止。编译成功后，将会在项目文件夹中生成一个.HEX 类型的文件，这个文件是在单片机上可以执行的文件，在程序烧录的时候将会用到该文件。

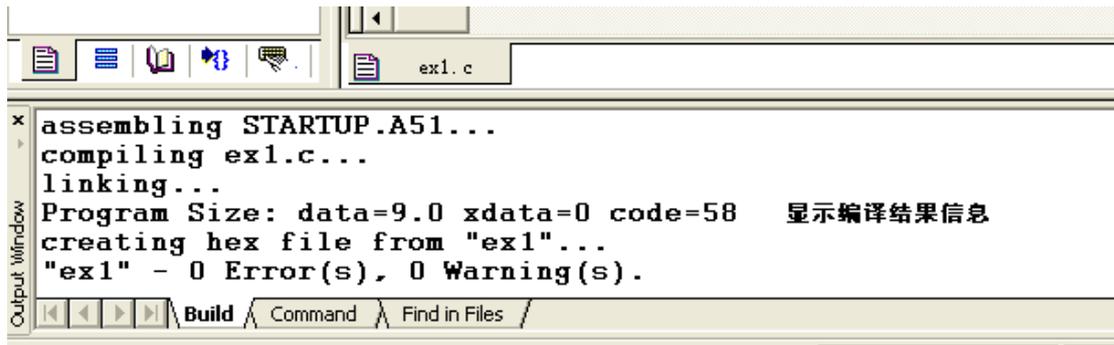


图 1.26 编译结果

至此，完成了第一个 C51 项目。

4、 μ Vision3 集成开发环境下的软件模拟仿真

项目建立并编译成功后，在烧录到单片机硬件中之前，一般需要进行仿真调试，确保程序的执行结果完全符合要求。 μ Vision3 集成开发环境提供了完善的项目仿真和调试工具，它既支持软件仿真模式、也支持硬件仿真模式，下面以软件仿真模式为例来演示如何使用 μ Vision3 集成开发环境进行项目的仿真调试。

项目仿真调试之前需要确保该项目完全通过编译。仿真调试步骤如下：

第一步：在菜单栏中选择“Debug”菜单，在弹出的快捷方式中选择“Start/Stop Debug Session”选项，或者单击工具栏内的快速调试按钮，如图 1.27 所示。

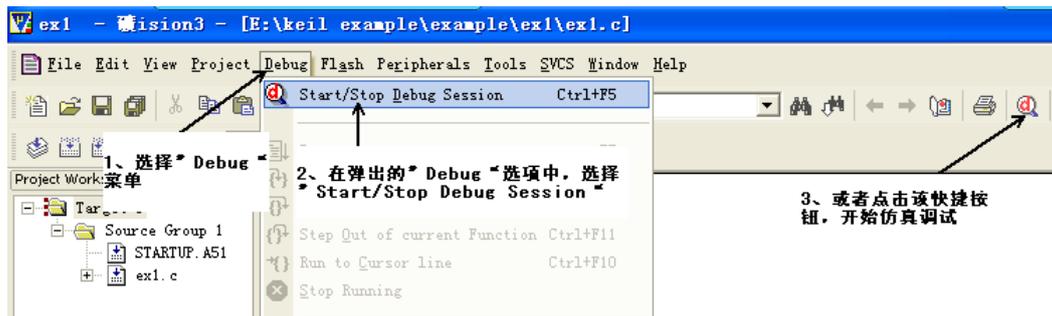


图 1.27 仿真调试选择

μ Vision3 进入仿真调试模式。此时界面的菜单栏和工具栏都会发生相应的变化。如图 1.28 所示。

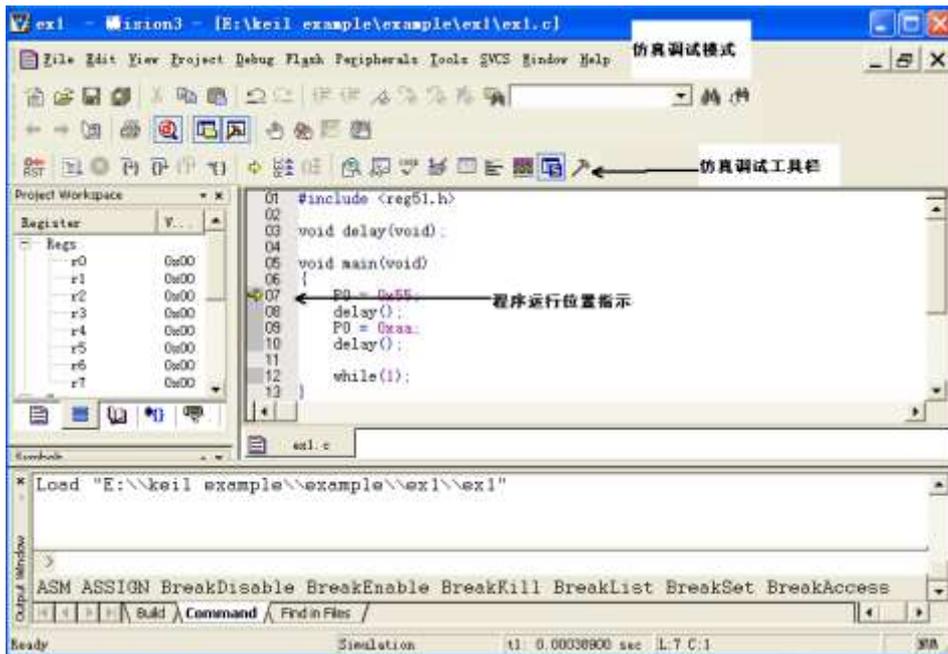


图 1.28 仿真调试界面

第二步：进入程序仿真界面后，可以利用调试工具栏的相应图标对程序进行仿真调试，如表 1.1 所示是调试工具栏各种图标的功能描述。

工具栏图标	对应的菜单命令	功能描述
	Peripherals→Reset CPU	复位 MCU，程序回到起始位置
	Debug→Go	运行程序，直到手动停止或者遇到断点
	Debug→Stop Running	停止程序运行
	Debug→Step	单步运行程序，遇到子程序则进入子程序运行
	Debug→Step over	单步运行程序，遇到子程序则将该子程序当做一步运行
	Debug→Step out of Current function	单步运行程序，跳出子程序
	Debug→Run to Cursor line	运行程序到光标所在的位置
	Debug→Show Next Statement	显示下一条指令
	Debug→Enable/Disable Trace Recording	使能/禁止程序运行轨迹的标识
	Debug→View Trace Records	显示程序运行过的指令

	View→Disassembly Window	显示/隐藏反汇编窗口
	View→Watch&Call Stack Window	显示/隐藏观察窗口和堆栈窗口
	View→Code Coverage Window	显示/隐藏代码报告窗口
	View→Serial Window #1	显示/隐藏串口 1 的观察窗口
	View→Memory Window	显示/隐藏存储器观察窗口
	View→Performance Analyzer Window	显示/隐藏性能分析窗口
	View→Logic Analyzer Window	显示/隐藏逻辑分析窗口
	View→Symbol Window	显示/隐藏符号窗口
	View→Toolbox	显示/隐藏

表 1.1 调试工具栏图表及功能

第三步：程序运行中，可以通过打开各种观察窗口来观察程序的执行结果。如图 1.29 所示，选择“Peripherals”→“I/O Ports”→“Port 0”，打开并行仿真端口 P0。在程序运行过程中，可以观察 P0 口各位的电平变化情况。如果程序中用到了 P1、P2、P3 等端口，可以分别选择打开这几个端口的仿真观察窗口。

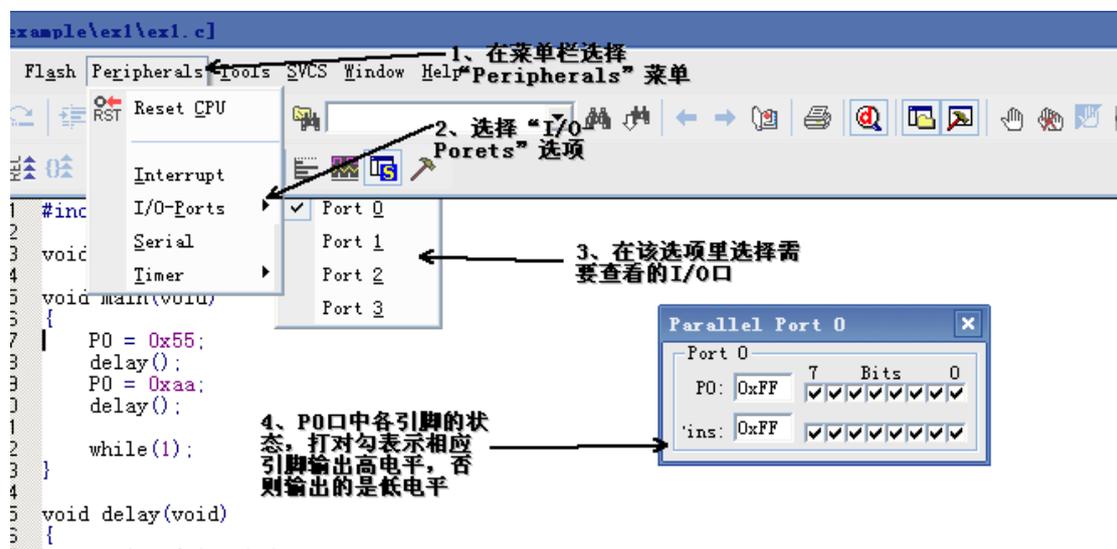


图 1.29 I/O 口状态观察窗口

第四步：依次选择“View”→“Memory Window”以及“View”→“Watch&Call Stack Window”，观察变量及 RAM 区的数据变化情况，如图 1.30 所示。



图 1.30 变量及 RAM 区数据观察窗口

第五步：观察程序运行时间以及通用寄存器、堆栈指针等变化情况，如图 1.31 所示。

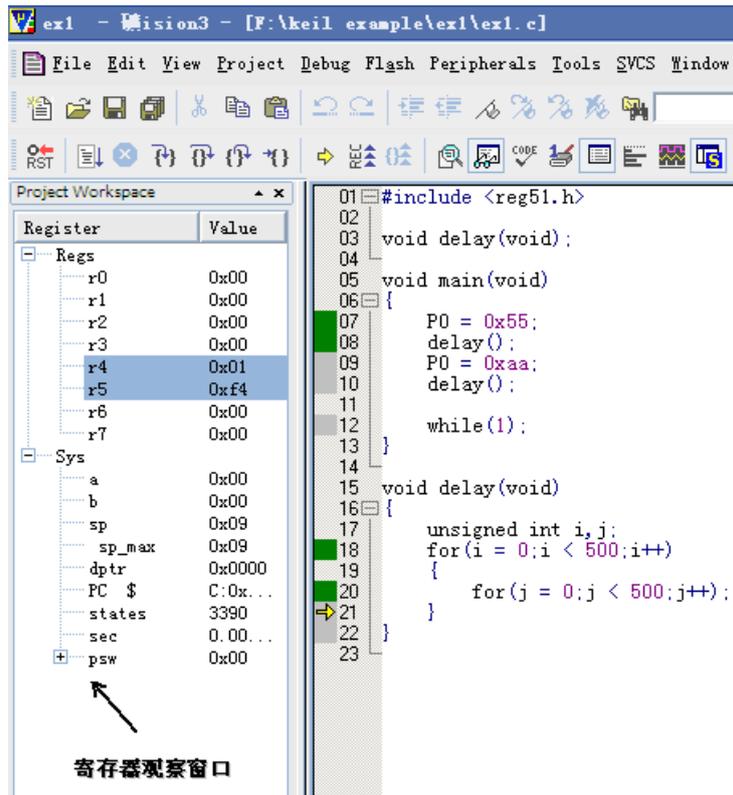


图 1.31 寄存器观察窗口

用户还可以打开其它一些仿真窗口，查看单片机其它一些资源的内容及变化情况。

第五章 仿真

1、仿真知识

仿真分为软件模拟仿真和硬件在线仿真：软件仿真是在计算机上模拟底层硬件，程序纯粹的在计算机上运行，因而与实际的嵌入式系统板运行情况还是有差距的，很多情况下，不能解决实时在系统调试的问题；

硬件仿真是利用底层系统板内运行程序，现象回读给计算机的过程，体现了程序在系统板的运行情况，与实际运行情况基本一致。

可以说硬件仿真对于初学者新手硬件理解程序，分析程序，查错，很有帮助。

下面以流水灯为例程，说明如何设置 KEIL 开发环境，来实现 STC89C52 的硬件仿真。

2、硬件连接

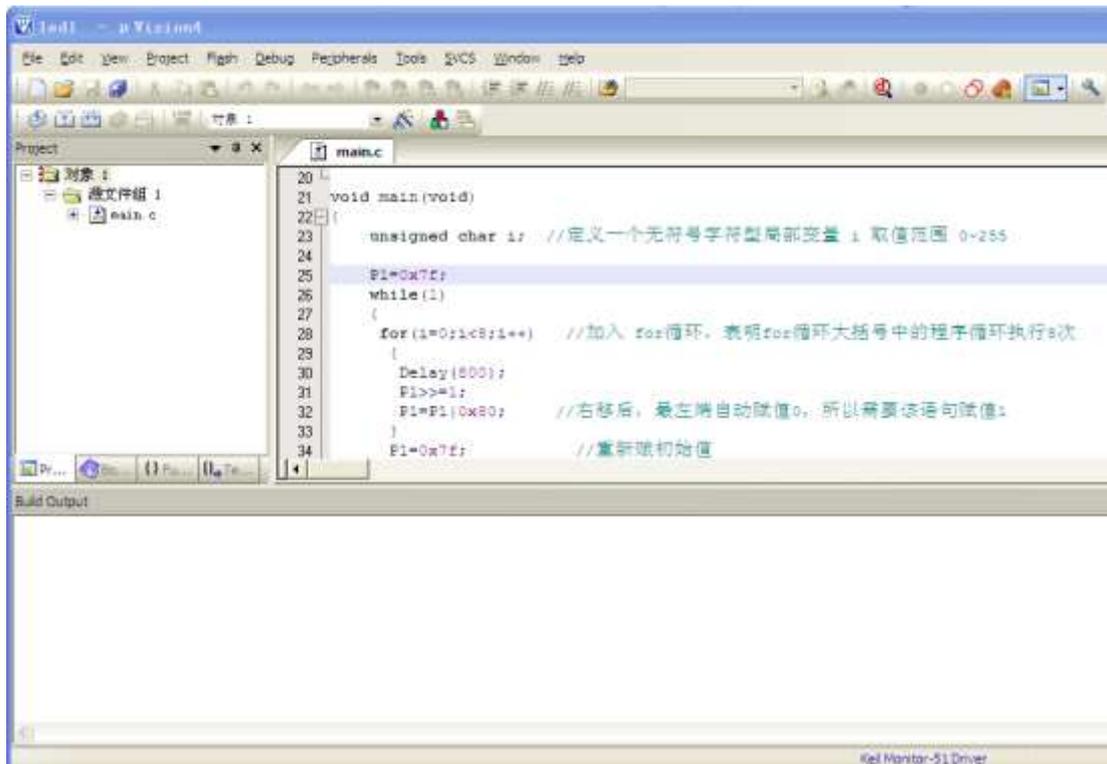
将仿真器插到核心板区座上。



2) 将 RS232 线一端插接在串行通信的 DB25 上，另外一头插接在电脑串口，打开电源开关。

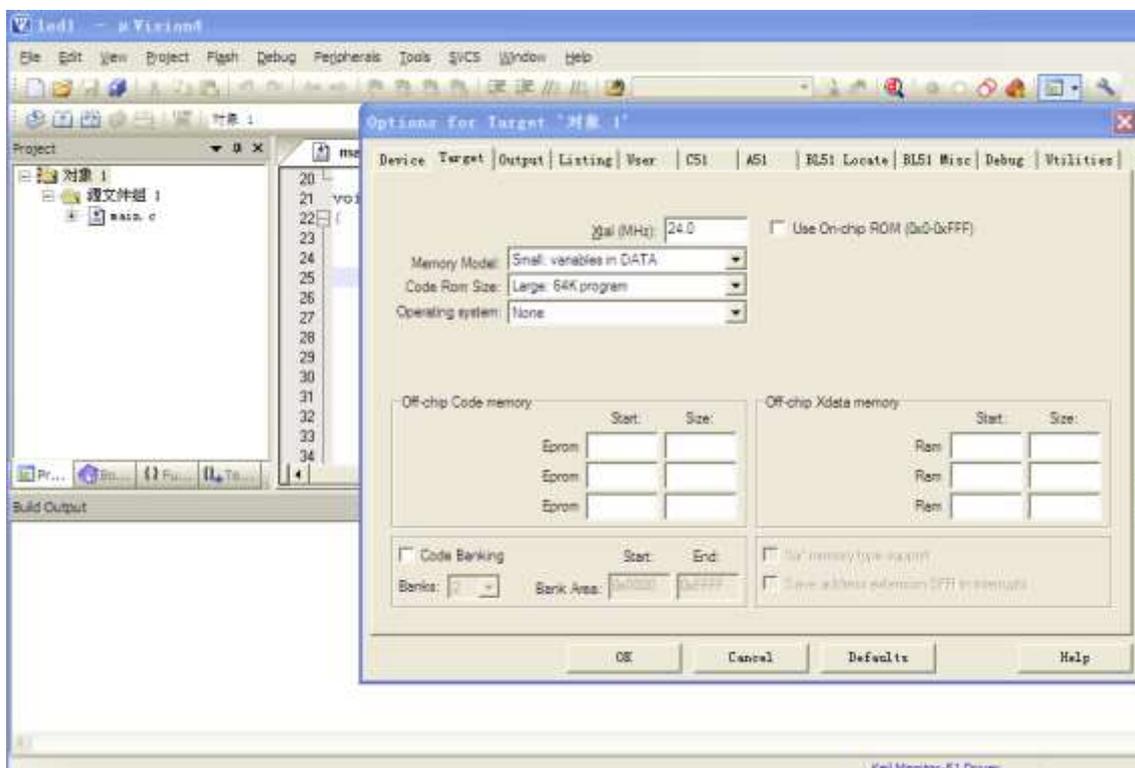
3、keil 打开工程

打开 KEIL 软件，点击 open 菜单打开工程，如图：打开我们光盘上提供的代码，流水灯实验（最好将实验代码拷贝到硬盘，便于反复修改练习。）



4、设置项目属性

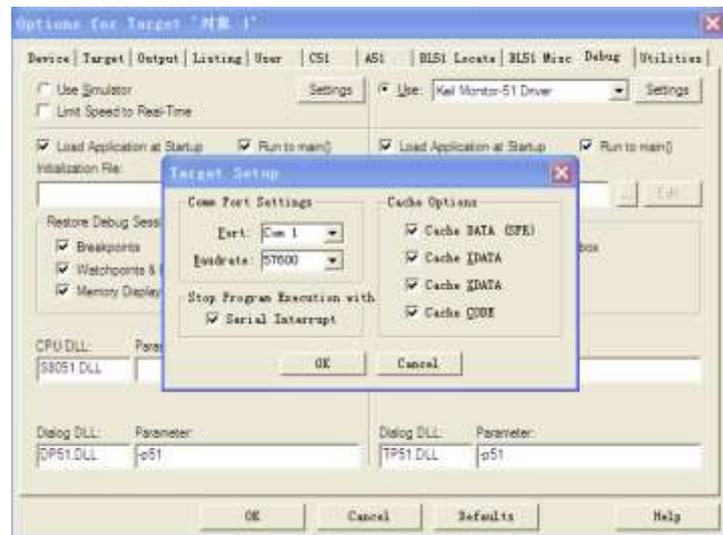
在项目名称上鼠标右键，进入项目属性设置，弹出项目属性设置窗口



我们选择右侧倒数第二个 Debug 选项卡，选择 Use Keil Montior-51 Driver 然后选择 右侧的 Settings 进入仿真通讯设置



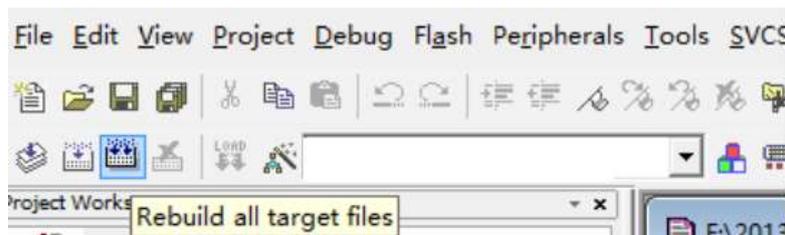
选择串口号、设置波特率



设置完毕点击 OK，确定。设置完毕。

5、开始仿真

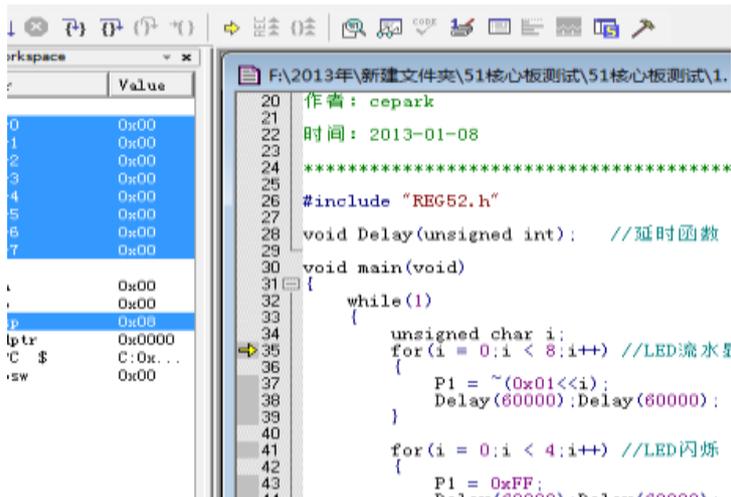
点击  重新编译工程



点击主菜单上  按钮，开始硬件仿真



连接成功会在软件下方提示如图，并且出现调试的黄色箭头，指向程序开始。



点击单步，全速运行按钮就可以在系统调试程序了

你也可以试试其他按钮，也可以点击全速运行，如果在调试中出现问题，大家可以按一下仿真器上复位按钮，重新链接调试。