

单片机原理与应用 实验指导书

合肥学院人工智能与大数据学院

2019 年 10 月

目 录

目 录	1
前 言	3
实验要求	4
第一章 实验平台及开发工具介绍	5
第二章 烧录第一个程序	21
第三章 实验指导	32
实验一 IO 口实验——跑马灯	32
实验二 键盘显示实验——密码键盘	34
实验三 定时器/中断实验——电子钟	40
实验四 定时器综合实验——PWM 调光灯	42
实验五 串口通信实验——远程控制	44
实验六 液晶显示电子钟	46
实验七 智能家居控制系统	48
附录一 试验箱结构及电路结构	53
附 1 51 核心板电路电源电路	53
附 2 底板单色流水灯	53
附 3 底板彩色流水灯	54
附 4 双色二极管	54
附 5 四位一体共阳数码管	55
附 6 74HC595 驱动	55
附 7 LCD1602 液晶	56
附 8 LCD12864 液晶	56
附 9 OLED 液晶	57
附 10 2.8 寸 TFT 屏	57
附 11 独立按键	58
附 12 核心板 USB 供电 5V 电源电路	58
附 13 核心板 3V3 电源电路	59
附 14 核心板 1V8 电源电路	59
附 15 矩阵键盘	60
附 16 DS18B20 电路	60
附 17 DS1302 电路	61
附 18 AT24C02 电路	61
附 19 MAX485 通信电路	61
附 20 PCF8591 AD/DA 转换模块	62

附 21	2 路串口模块	62
附 22	红外收发头模块	63
附 23	SD 卡模块	63
附 24	无源蜂鸣器模块	63
附 25	两路继电器加步进电机模块	64
附 26	两路直流电机加步进电机模块	64
附 26	光敏/火焰/霍尔传感器模块	65
附 27	电源引出口	65
附 28	串转并模块	65
附 29	并转串模块	66
附 30	双色点阵模块	66
附 31	74HC573 八位锁存模块	67
附 32	74LS138 三八译码器模块	67
附 33	模拟量输入模块	67
附 34	五向摇杆模块	68
附 35	一位数码管模块	68
附录二	8051 系列单片机汇编语言指令速查表	69

前 言

由于单片机具有高可靠性、超小型、价格低、容易产品化等特点，在仪器仪表智能化、实时工业控制、实时数据采集、智能终端、通信设备、导航系统、家用电器等控制应用领域，具有十分广泛的用途。由于目前在国内单片机应用中，MCS-51 系列单片机仍然是一种主流单片机，所以本实验指导书为学习 MCS-51 单片机的本系学生，配合《单片机原理与应用》课程的教学，结合实验开发平台编写了这本实验指导书。

《单片机原理与应用》是一门实践性很强的课程，提高教学质量的一个重要环节是上机实习和训练，无论是学习汇编语言程序设计，还是学习接口电路和外设，或者软硬兼施学习单片机应用系统，不通过加强动手时不能获得预期效果的。本实验指导书提供 10 个实验的指导性材料，涵盖单片机内部资源的使用、外围常用接口的设计和使用及单片机典型应用系统开发，对培养学生动手能力、单片机系统搭建能力、编程能力有一定的提升。

由于时间紧迫，需要赶课程进度与实验时间的同步，加上编者学识有限，如有不妥之处，欢迎批评指正。

编者 龙夏
2019 年 10 月

实验要求

一、实验素养和一般知识

1. 实验前准备工作

(1) 明确实验目的、熟悉实验内容、掌握实验步骤、了解所用单片机实验开发系统及仪器设备的性能。

(2) 做好实验前预习和必要的准备（先编好程序），作到有的放矢。

2. 进入实验室后

(1) 熟悉所用的单片机实验开发系统及仪器设备，了解其外观、性能。

(2) 接线要简单、正确、明了，并检查接线是否正常，确保没有短路情况发生。

(3) 通电瞬间，应注意仪器是否正常工作，不要只看开关，如有不正常现象，应立即断电找出原因。

(4) 不允许自带光盘和软盘上机，违反者取消实验资格！（指导教师允许情况除外）。

3. 实验进行时

(1) 正常请况下，按规定步骤进行，读取数据，并及时纪录。

(2) 各种操作要有目的进行，不可盲动。

(3) 实验完后，经老师察看或签字，并把仪器、导线、工具清理、收拾完毕后，学生签到方可离开实验室。

二、实验报告内容和要求

1. 实验报告要用学校统一的纸。

2. 实验内容应包括，是实验目的、设备、原理、步骤简述、原始数据、结果和分析以及程序清单等。

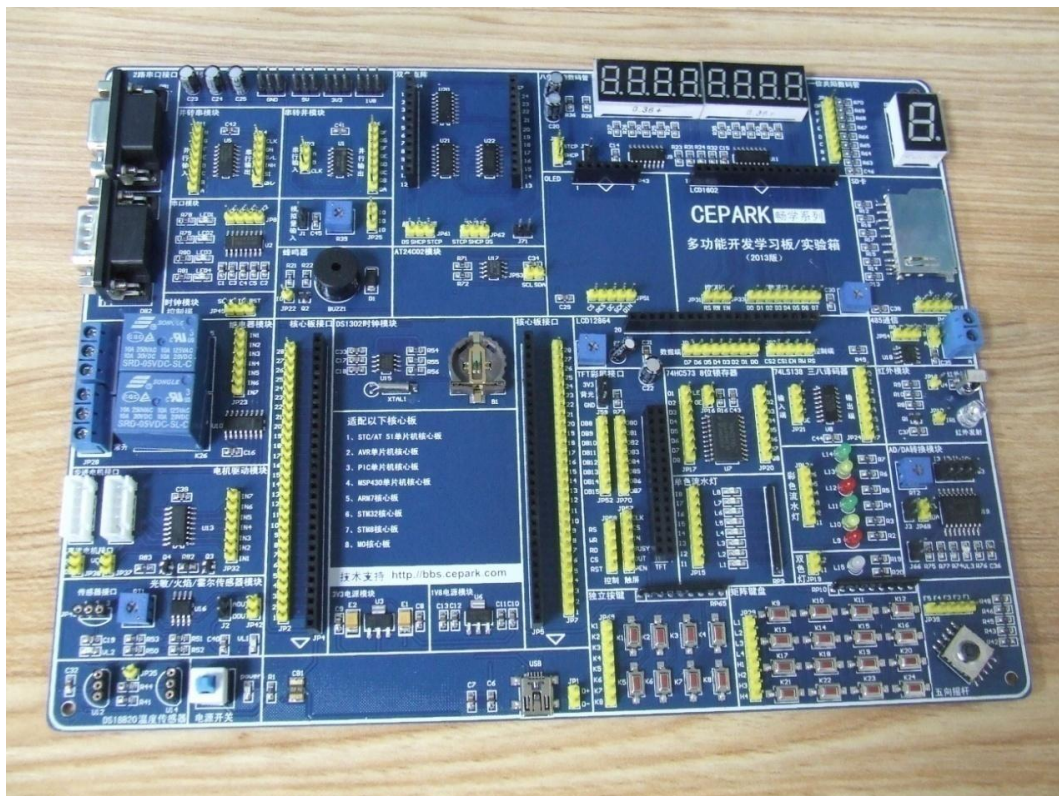
3. 每个实验后的思考题必须完成。

4. 对实验中发现的问题应加以讨论，并提出自己的改进意见和要求、希望等。

第一章 实验平台及开发工具介绍

畅学实验开发平台由功能主板和扩展核心板组成，扩展核心板可以直接插到底板扩展接口，组合成一个多功能开发学习板，这种独特的开放式接插设计可以完美兼容常见芯片，功能底板和扩展核心板可以一起使用，也可以分开单独使用，灵活性、扩展性极强。扩展板除可以51单片机核心板外，还支持以下核心板：AVR16单片机核心板、PIC单片机核心板、MSP430单片机核心板、ARM7核心板、STM32核心板、AVR64/128核心板等其他类型处理器。

一、底板介绍



功能底板板载19个芯片，集成38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。由于主板模块较多，所以使用之前必须看懂电路图，并能在开发板上找到对应位置，只有对硬件连接了解后，才能去学习写程序。

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位数码管
- 5) 1个1位数码管
- 6) 1个8×8 红绿双色点阵
- 7) LCD1602液晶接口
- 8) LCD12864液晶接口
- 9) OLED模块接口
- 10) TFT彩屏接口
- 11) 8个独立按键
- 12) 4×4 矩阵键盘
- 13) 5向摇杆按键
- 14) 两路DS18B20
- 15) 热敏/光敏/霍尔传感器模块
- 16) 2路直流电机接口
- 17) 2路步进电机接口
- 18) 2路继电器
- 19) 2路DB9串口（1公1母）
- 20) DS1302时钟模块
- 21) 无源蜂鸣器
- 22) AT24C02模块
- 23) 模拟量输入
- 24) 74HC164串转并模块
- 25) 74HC165并转串模块

- 26) SD卡模块
- 27) MAX485模块
- 28) 红外发射模块
- 29) 红外接收模块
- 30) PCF8591AD/DA模块
- 31) 74HC573锁存芯片
- 32) 74LS138译码器模块
- 33) 3.3V电源电路
- 34) 1.8V电源电路
- 35) 3.3V、5V、1.8V、GND电源引出
- 36) USB通讯接口
- 37) 自恢复保险丝电源保护
- 38) 核心板扩展接口

从上述资源介绍可以看出，畅学系列开发平台资源较为丰富，包含了所有常用的资源，同时留有扩展口，方便外接电路。

二、核心板介绍



(1)核心板介绍

畅学系列51单片机最小系统核心板支持AT和STC两种类型双列直插DIP40系列的51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，即可以插到开发平台底板上组合使用，也可以单独使用。

该核心板标配一片STC12C5A60S2单片机，使用一条USB线即可实现核心板及底板供电及在线虚拟串口通许实现程序下载。

支持在线仿真功能，将51单片机芯片更换为SST89E516RD仿真芯片，短接仿真功能跳线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。

外部晶振使用圆孔插座，可以方便更换外部晶振，核心板标配3个不同频率晶振。

板载自恢复保险丝，可以对核心板功能进行电流保护，防止过大电流烧毁核心板芯片级电路。

板载多组电源引出，5V电源4组、3.3V电源4组，GND4组，方便扩展模块功能使用。

(2)处理器介绍

畅学51单片机核心板采用了宏晶公司最新的1T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、ST89S52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AD转换(250K/S，即25万次/秒)，针对电机控制、强干扰场合。具有以下特点：

- 1) 工作电压：5.5V-3.3V
- 2) 工作频率范围：0-35mhz，相当于普通8051的0-420mhz
- 3) 用户应用程序空间：60K
- 4) 片上集成1280Byte RAM
- 5) 通用IO口 36个
- 6) 双串口

7) 工作温度范围: -40~+85℃(工业级)/0-75℃(商业级)

8) 封装: PDIP40

STC12系列单片机取代传统8051注意事项:

STC12C5A60S2系列单片机的定时器0/定时器1与传统8051兼容, 上电复位后, 定时器部分缺省还是除12再技术的, 而串口由定时器1控制速度, 所以定时器/串口完全兼容。

增加了独立波特率发生器, 省去了传统8052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。

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。

三、Keil C51介绍

keil uvision2 C51 软件是目前功能最强大的单片机 c 语言集成开发环境。

下面我们通过图解的方式来 Keil C51 软件的使用教程, 学习最简单的, 如何输入源程序 → 新建工程 → 工程详细设置 → 源程序编译得到目标代码文件。

第一步: 双击 Keil uVision2 的桌面快捷方式 (图 1-1), 启动 Keil 集成开发开发软件。软件启动后的界面如图 1-2 所示。



图 1-1: 启动 Keil uVision2 单片机集成开发环境

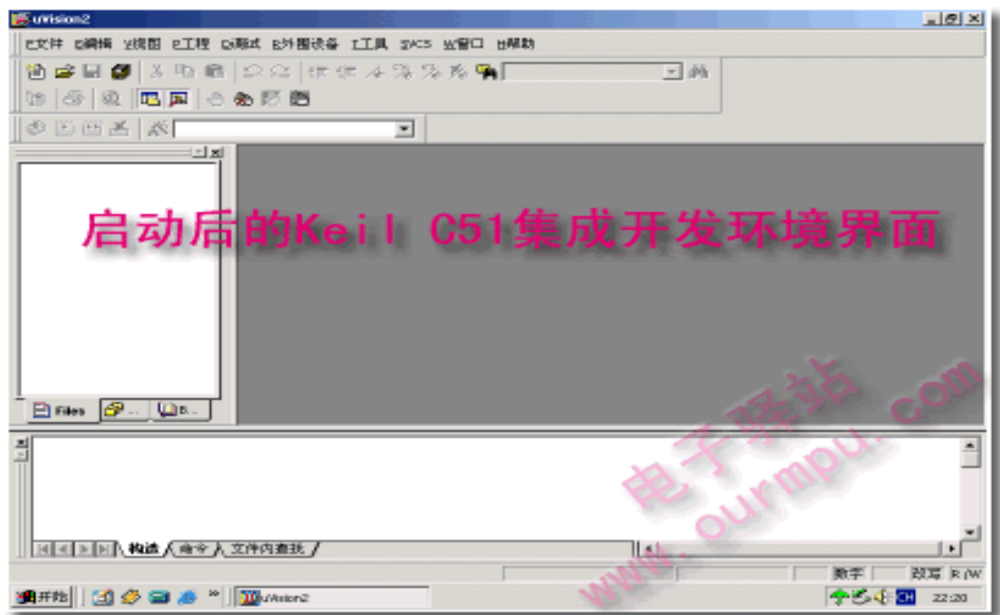


图 1-2：软件启动后的界面

第二步：新建文本编辑窗。点击工具栏上的新建文件快捷按钮，即可在项目窗口的右侧打开一个新的文本编辑窗。见图 1-3 所示。

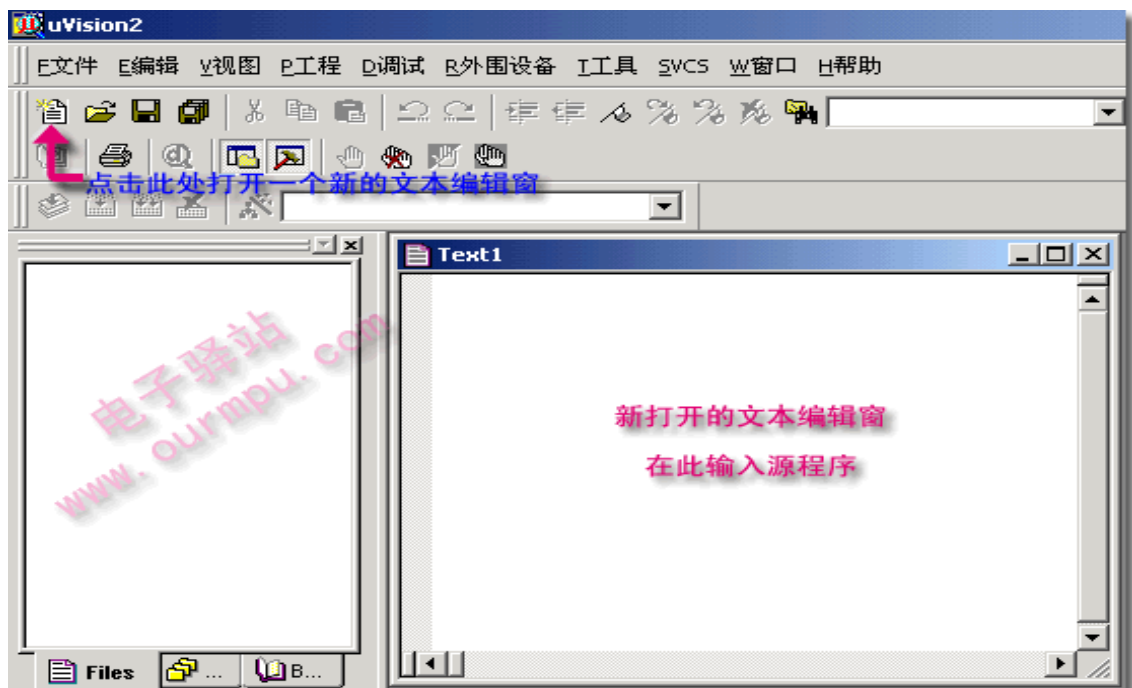


图 3：新建文本编辑窗

第三步：输入源程序。在新的文本编辑窗中输入源程序，可以输入 C 语言程序，也可以输入汇编语言程序。

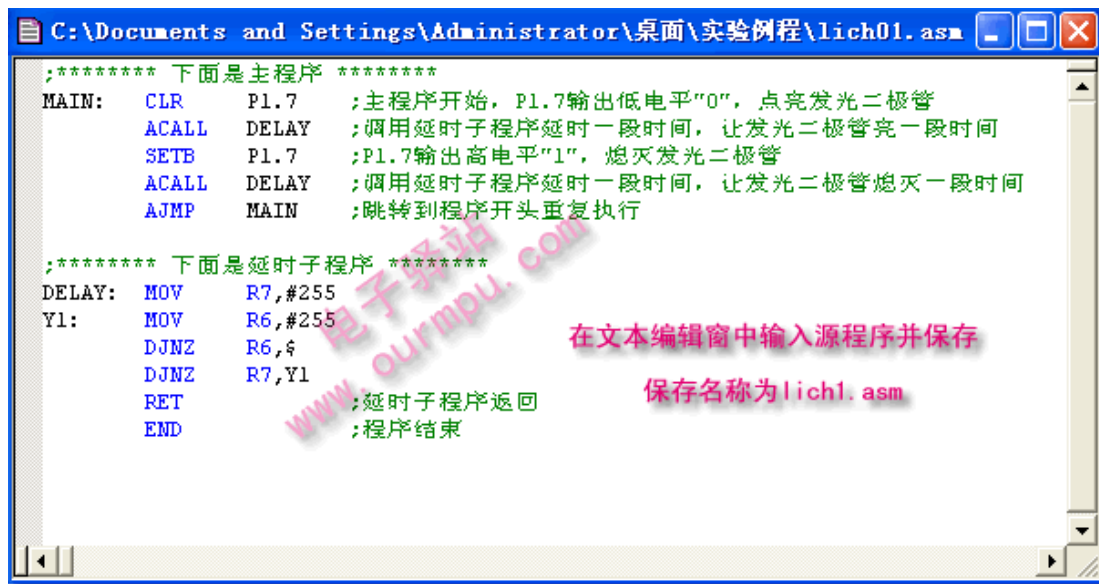


图 4：输入源程序

第四步：保存源程序。保存文件时必须加上文件的扩展名，如果你使用汇编语言编程，那么保存时文件的扩展名为 ".asm"，如果是 C 语言程序，文件的扩展名使用 "*.C"。

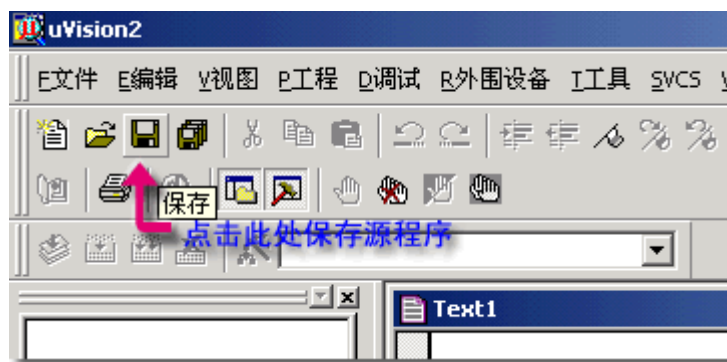


图 5：保存源程序

注：第 3 步和第 4 步之间的顺序可以互换，即可以先输入源程序后保存，也可以先保存后输入源程序。

第五步：新建立 Keil 工程。如图 6 所示，点击“工程”→“新建工程”命令，将出现保存对话框（图 7）。

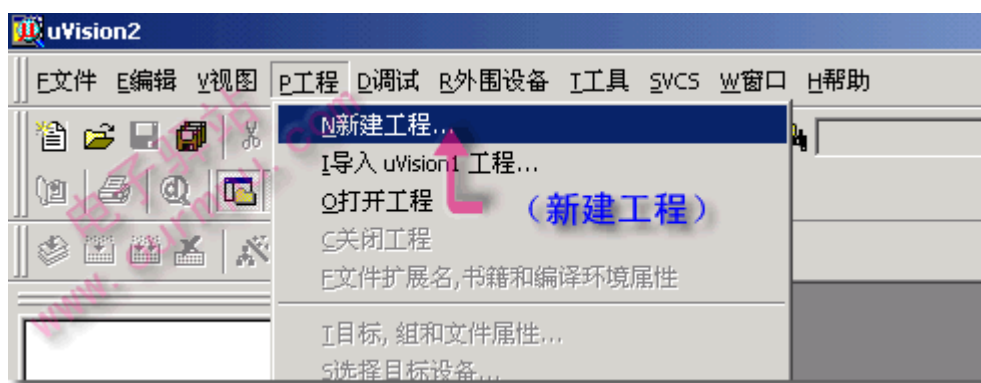


图 6：新建工程

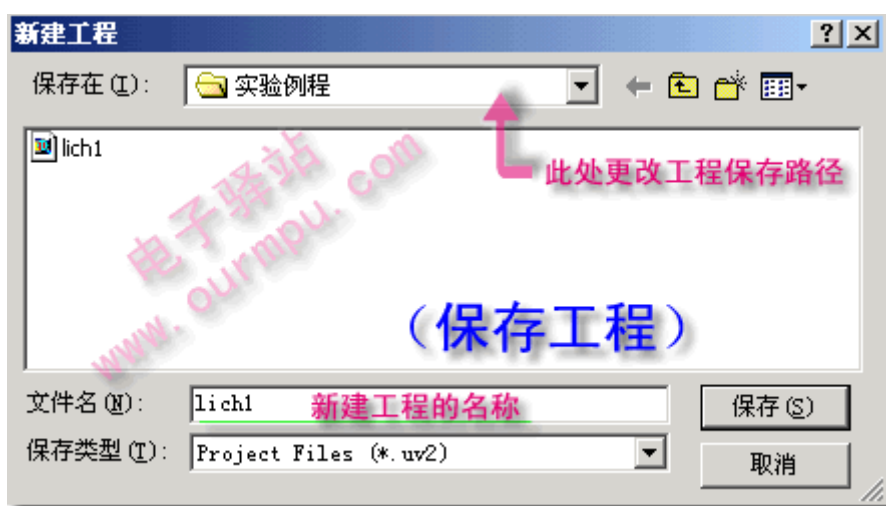


图 7：保存工程对话框

在保存工程对话框中输入你的工程的文件名，Keil 工程默认扩展名为“.uv2”，工程名称不用输入扩展名（见图 7），一般情况下使工程文件名称和源文件名称相同即可，输入名称后保存，将出现“选择设备”对话框（图 8），在对话框中选择 CPU 的型号。

第六步：选择 CPU 型号。如图 8 所示，为工程选择 CPU 型号，本新建工程选择了 ATMEL 公司的 AT89C51 单片机。

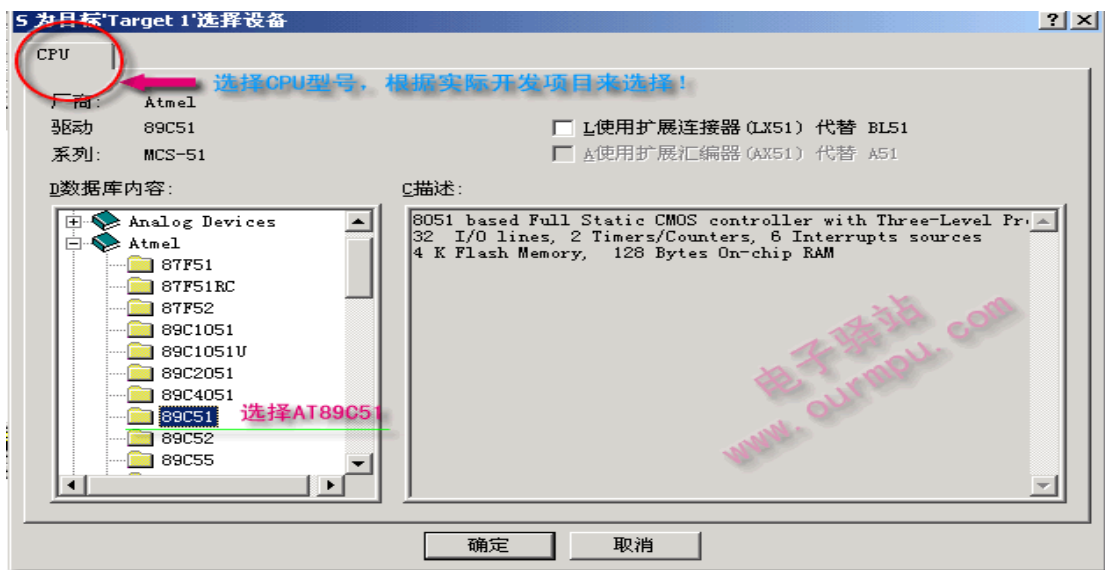


图 8： 选择 CPU 型号对话框

第七步：加入源程序到工程中。在选择好 CPU 型号后，点击“确定”按钮返回主界面，此时可见到工程管理窗中出现“Target 1”，点击“Target 1”前面的“+”号展开下一层的“Source Group 1”文件夹，此时的新工程是空的，“Source Group 1”文件夹中什么文件都没有，必须把刚才输入的源程序加入到工程当中。如图 9 所示，右击工程管理窗中的“Source Group 1”，出现下拉菜单，点击“增加文件到组 Source Group 1”命令，将出现添加文件对话框（图 10）。

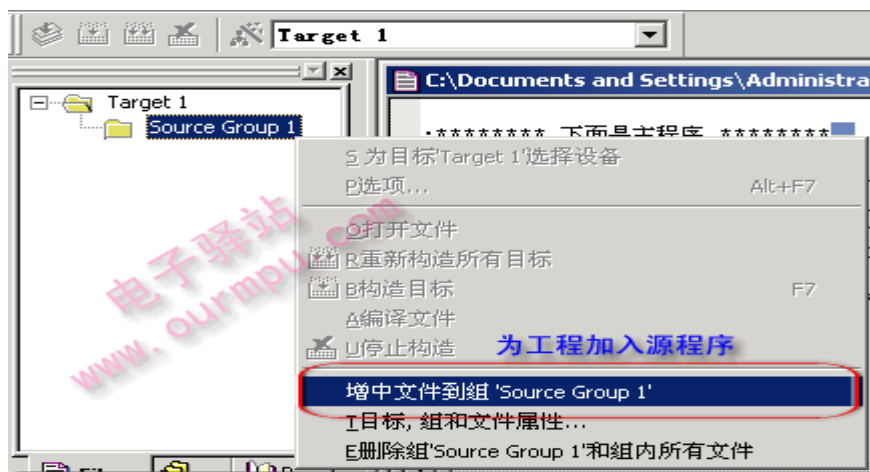


图 9： 添加文件到工程命令

在添加文件对话框（图 10）中，找到要添加到工程中的源程序文件。注意：在对话框中的文件类型默认为“C 源文件 (*.c)”，如果你要添加到工程中的是汇编语言程序，则在文件类型中必须选中“Asm 源文件 (*.asm; *.src)”，以*.asm 为扩展名的汇编源程序才会出现在文件列表框中。



图 10： 浏览选择 ASM 源程序

双击该文件 lich1.asm，即可将该文件添加到工程当中，另外也可以单击 lich1.asm 选中该文件，再点击“Add”按钮，也可以把文件加入工程中（见图 11）。

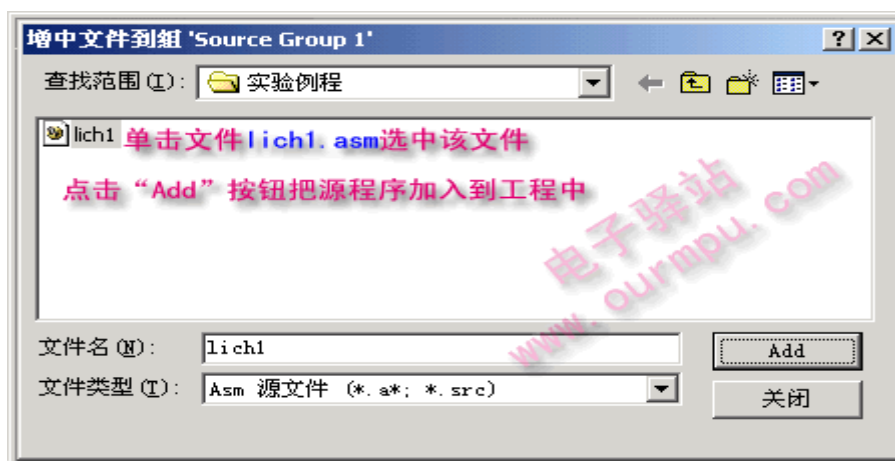


图 11： 选中 ASM 源程序，加入到工程中

点击 Add 按钮后，把文件添加到工程中，此时添加文件对话框并不会自动关闭，而是等待继续添加其他文件，初学者往往以为没有加入成功，再次双击该文件，则会出现图 12 对话框，表示该文件不再加入目标。此时此时应该点击“确定”按钮，返回到前一对话框，再点击“关闭”按钮，返回到主界面。



图 12：重复加入文件对话框

当给工程添加源程序文件成功后，工程管理器中的“Source Group 1”文件夹的前面会出现一个“+”号，单击“+”号，展开文件夹，可以看到 lich1.asm 已经出现在里面，双击即可打开该文件进行编辑修改源程序（见图 13）

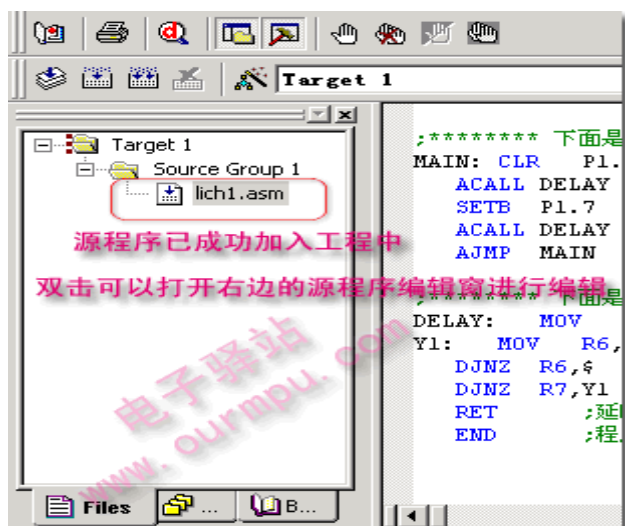


图 13：文件成功加入工程

第八步：工程目标' Target 1' 属性设置。如下图 14 所示，在工程项目管理窗中的“Target 1”文件夹上右击，出现下拉菜单，点击“目标' Target 1' 属性”命令，就进入目标属性设置界面。

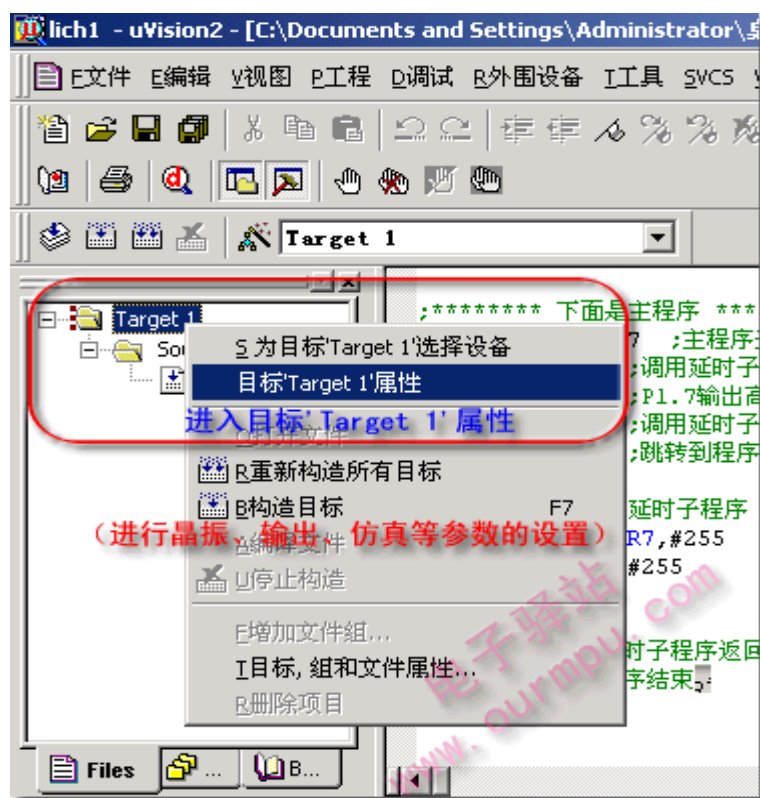


图 14： 进入工程目标属性设置

工程目标' Target 1' 属性设置对话框（图 15）中有 8 个页面，设置的项目繁多复杂，大部分使用默认设置即可，我们主要设置其中的“目标”、“输出”、“调试”三个页面，下面对这三个页面的设置进行详细介绍。

1、工程目标属性设置。该页面单片机的晶振频率、存储器等，把晶振的频率改为 11.0592，频率设置和你实验板上的实际晶振频率相同即可（图 15）。



图 15: 晶振频率设置

2、工程输出设置。该页面设置如下图 16 所示。注意：如果要进行单片机写片实验，则一定要把“E 生成 HEX 文件”选项选中，程序编译后才能生成我们写单片机需要的 HEX 格式目标文件。

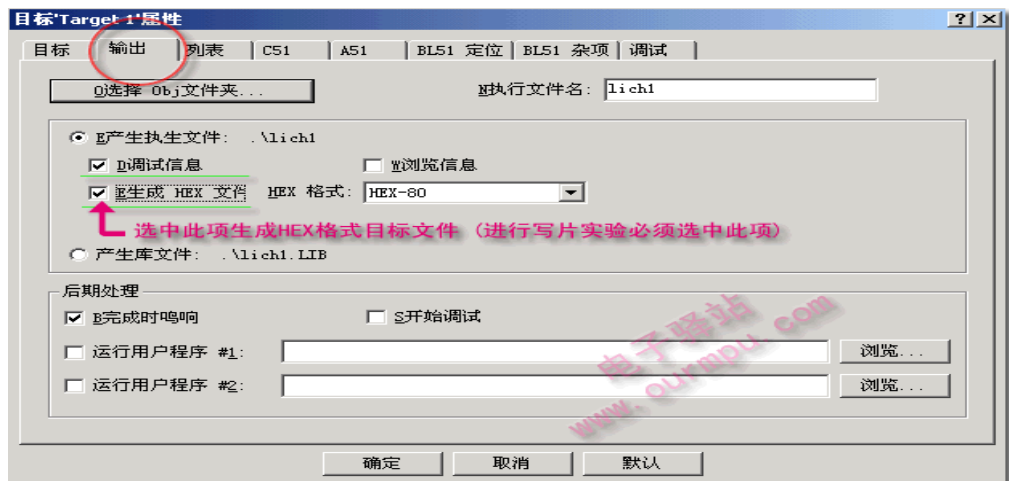


图 16: 工程输出设置

3、工程调试设置。“调试”页面设置如下图 17 所示。该页分为左右两半，左半边是软件仿真设置，而右半边是硬件仿真设置，当你使用软件仿真时，选中左边的“S 使用仿真器”；如果你使用硬件仿真器，那么就按下图所示设置硬件仿真，同时把仿真器连接到你的电脑串口上。

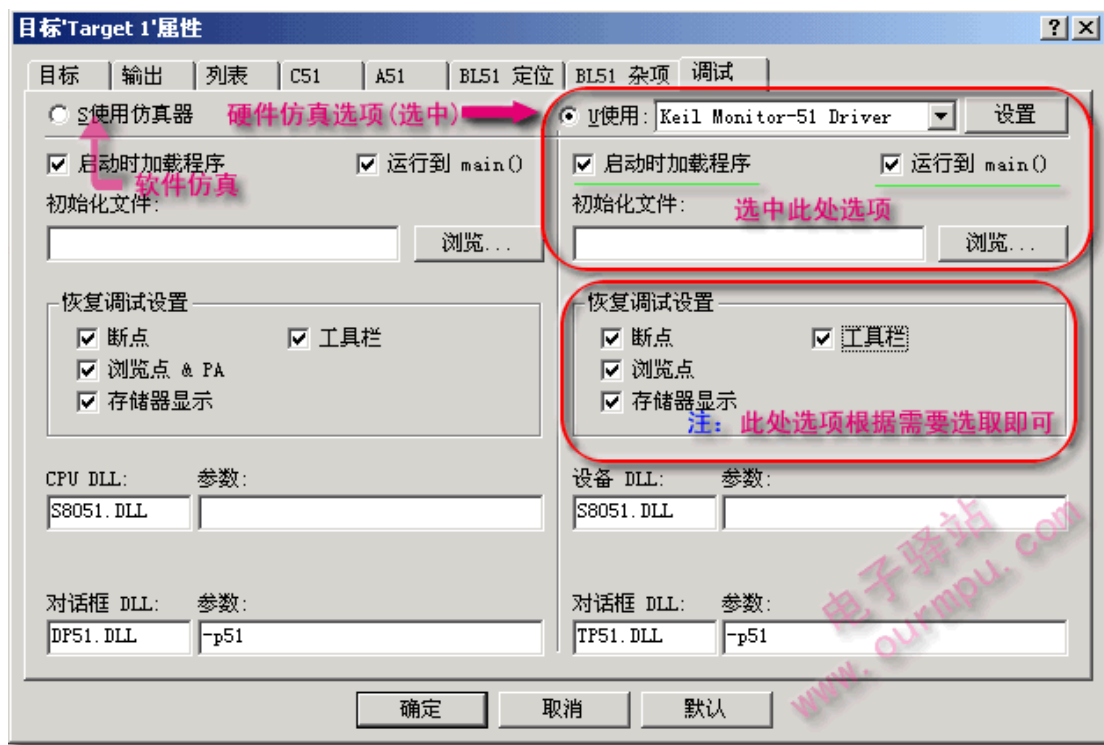


图 17：仿真调试设置

4、串口设置。串口设置如下图 18 所示。串口号根据你的仿真器实际连接来设置，如你把仿真器接到 COM2，那么就选择 COM2；通信波特率选择 38400 即可。

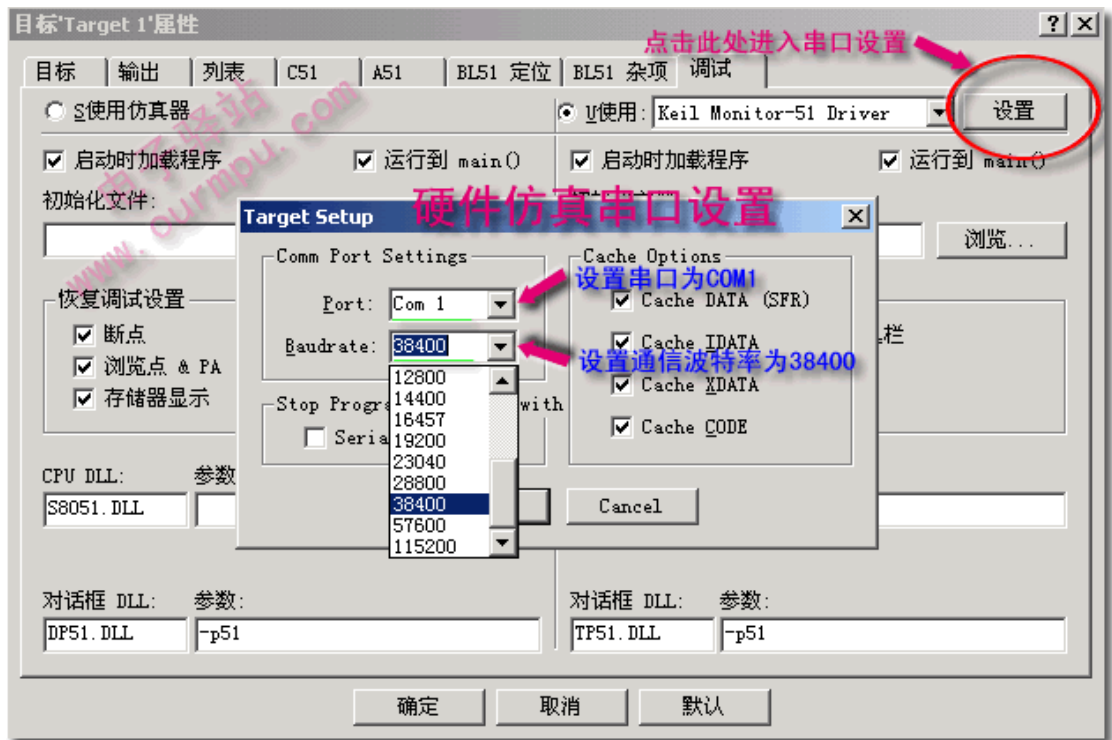


图 18：串口设置

第九步：源程序的编译与目标文件的获得

我们已经完成了从源程序输入、工程建立、工程详细设置的工作，接下来我们将完成最后的步骤，此时我们可以在文本编辑窗中继续输入或修改我们的源程序，使程序实现我们的目标，在检查程序无误后保存工程。接着如下图 19 所示，点击“构造目标”快捷按钮，进行源程序的编译连接，源程序编译相关的信息会出现在输出窗口中的“构造”页中。下图 19 显示编译结果为 0 错误，0 警告，同时产生了目标文件 lich1.hex。如果源程序中有错误，则不能通过编译，错误会在输出窗口中报告出来，双击该错误，就可以定位到源程序的出错行，我们可以对源程序进行反复修改，再编译，直到没有错误为止。注意：每次修改源程序后一定要保存。

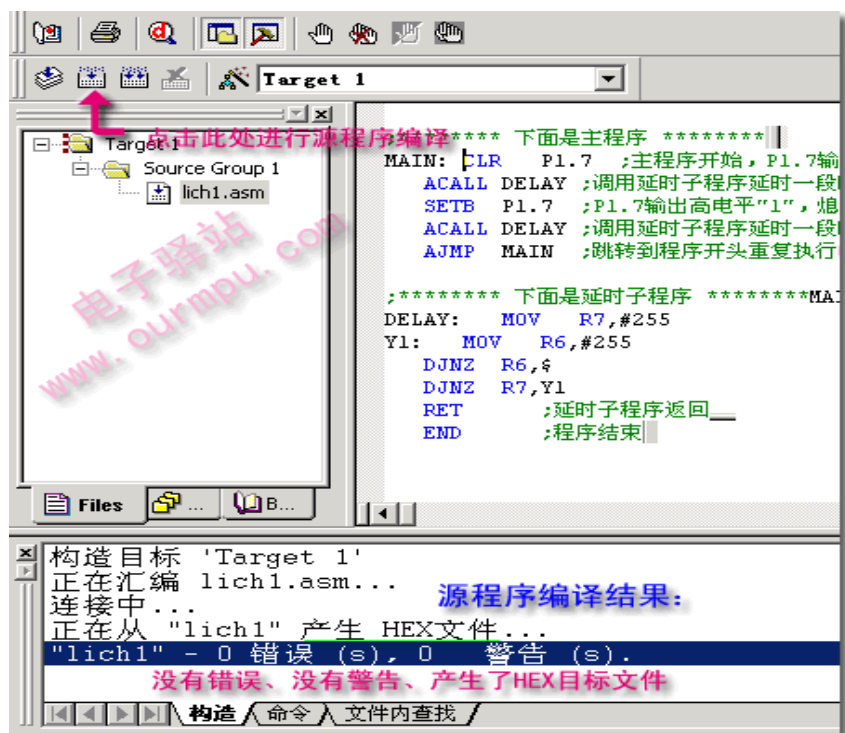


图 19: 源程序的编译

编译通过后, 我们打开工程文件夹(图 20), 可以看到文件夹中有了“lich1.hex”, 这就是我们需要的最终目标文件, 用编程器把该文件写入单片机, 单片机就可以实现我们程序的功能了。

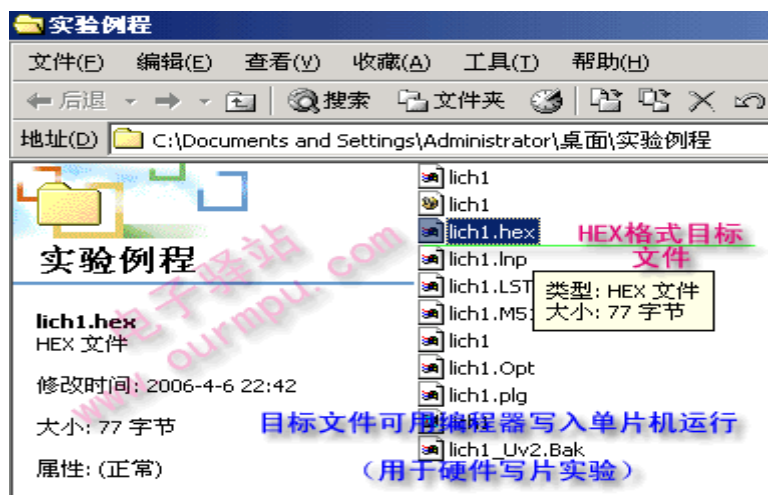


图 20: 编译成功, 获得目标文件 lich1.hex

第二章 烧录第一个程序

程序编写、编译、仿真器调试完毕后，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IAP在应用编程功能，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。

STC12C5A60S2系列单片机具有在系统可编程(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，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。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，故无须编程器（速度比通用编程器快，几秒一片）。

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。也可以到<http://www.mcu-memory.com/>下载最新版本。软件界面如图2-1所示。

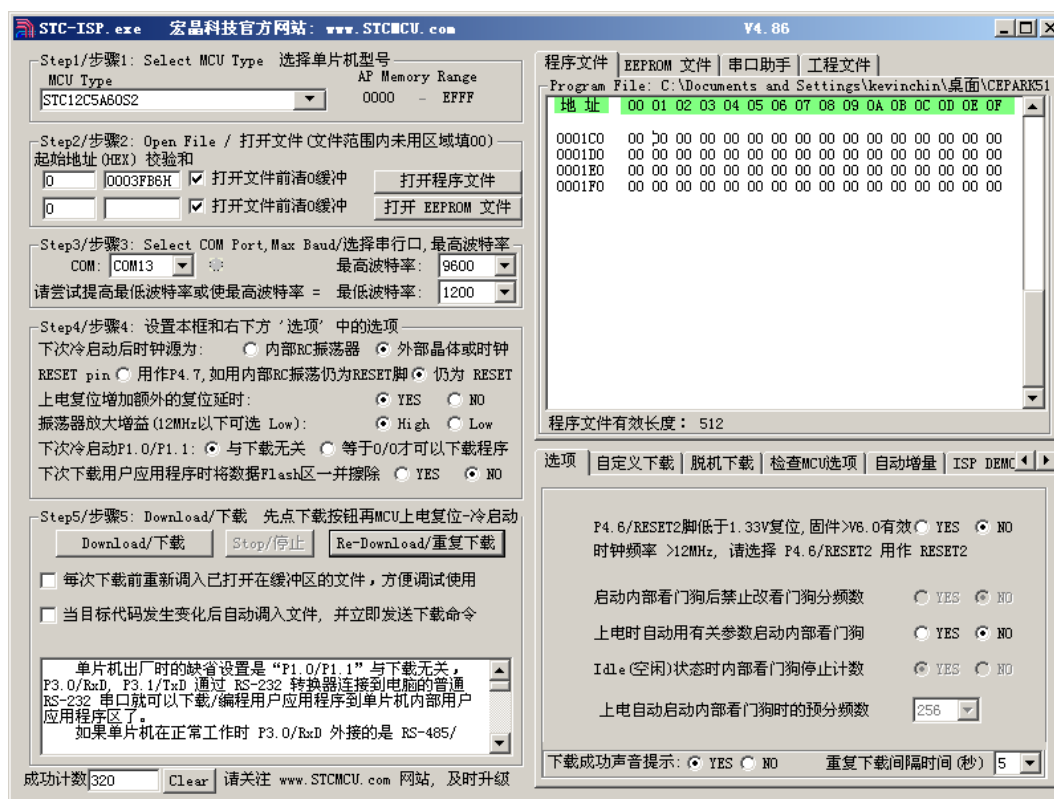


图2-1 STC-ISP软件主界面

3、硬件连接

- 1) 将畅学51单片机核心板插到畅学多功能底板上，如图2-2。

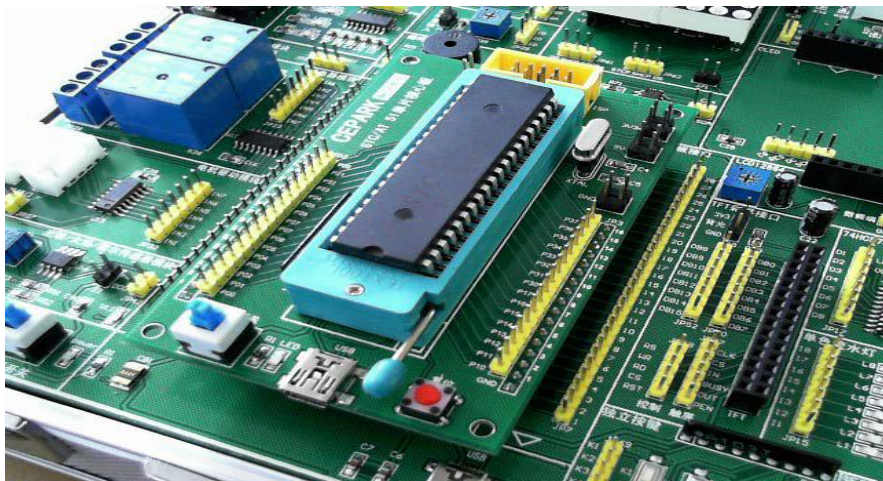


图2-2 接插核心板

注意核心板引脚1脚和底板引脚1脚对齐，如图2-3所示。

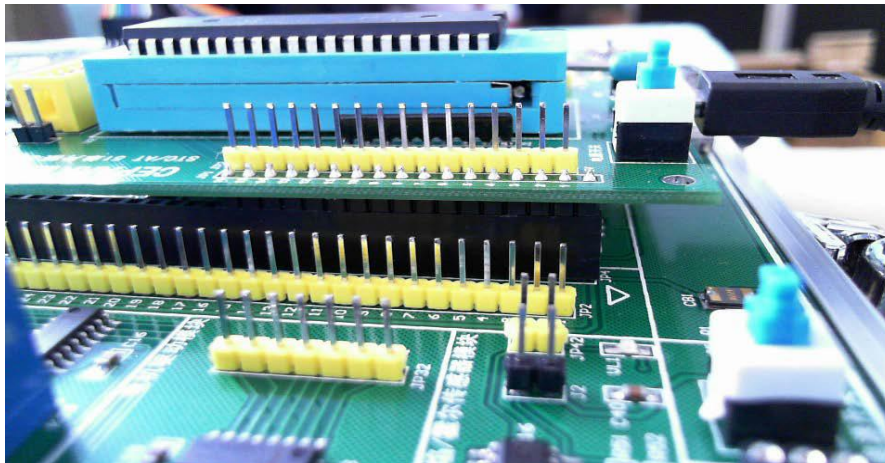
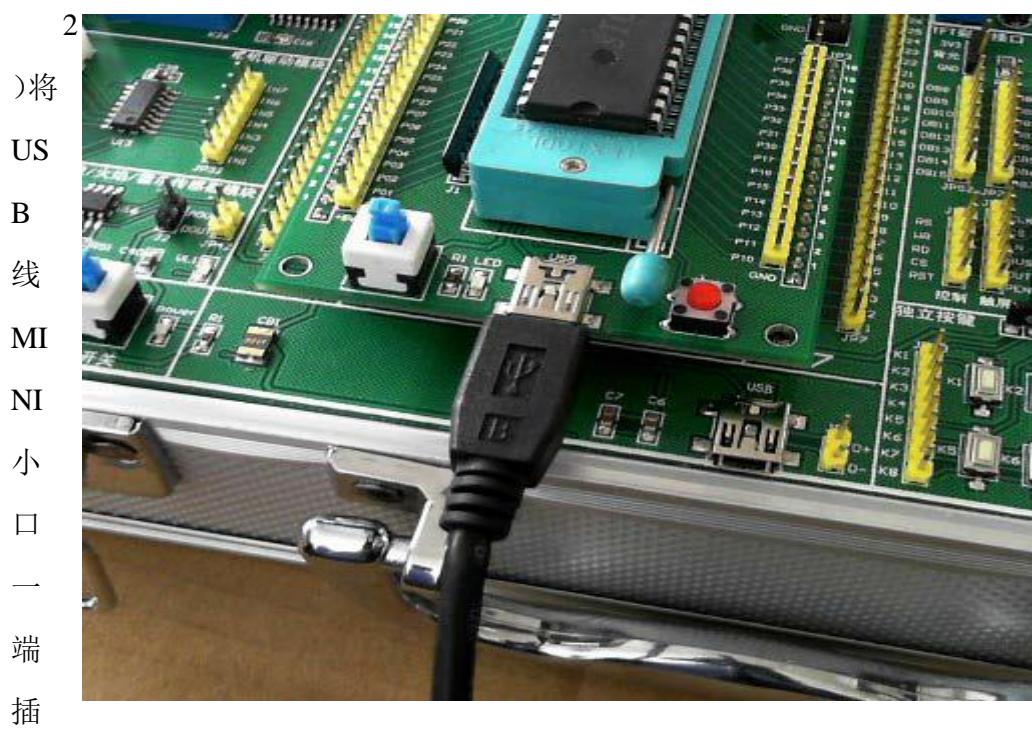


图2-3 核心板引脚对齐图



到畅学51单片机核心板USB接口，另外扁口A端连接电脑USB接口，如图2-4所示。

图2-4 USB电源连接

3) 将晶振插到畅学51单片机核心板晶振插座，默认为11.0592M，如图2-5所

示。

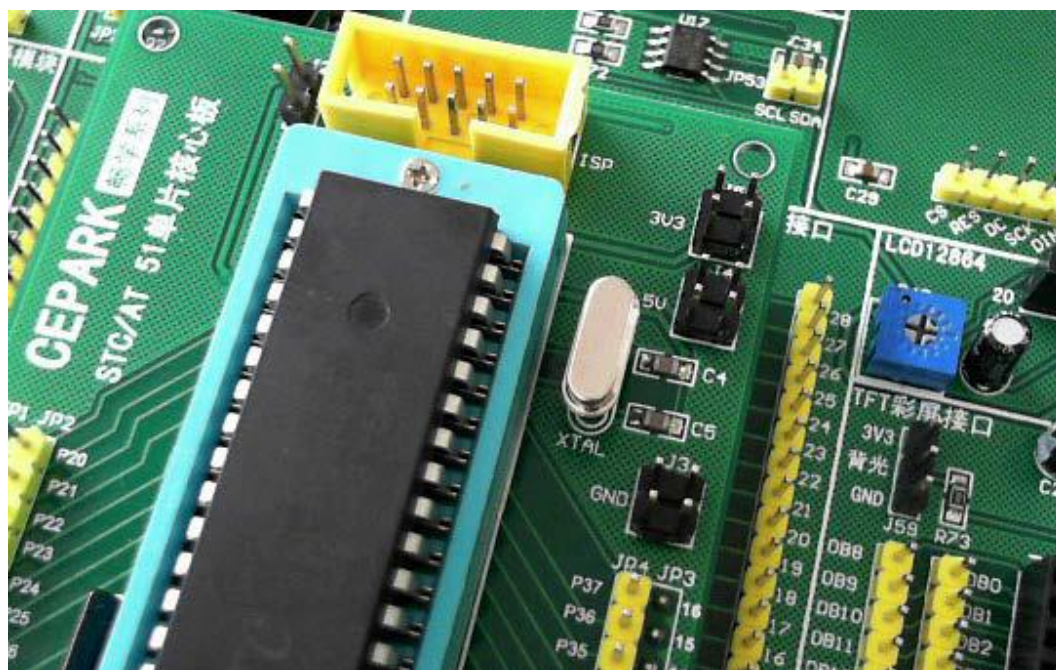


图2-5 晶振连接

这样我们就完成了畅学51开发板和电脑的硬件连接，CEPARK 畅学51单片机核心板集成了USB转串口功能，一条USB线既可以给开发板供电也可以实现虚拟串口通讯，下载程序。

连接后检查51单片机核心板电源指示灯是否低亮，按下电源开关后指示灯是否高亮，如果正常就可以继续下面的操作了。

4、驱动安装

畅学51单片机核心板集成了CP2102芯片，可以在将电脑一个USB接口虚拟为串口，这样通过一根USB线就能实现51单片机核心板的供电和程序程序下载以及串口通信，笔记本或台式电脑没有串口也可以很方便直接使用畅学51单片机核心板。

在使用之前需要先安装CP2102驱动程序。打开光盘中的驱动程序CP210x_VCP_Win_XP_S2K3_Vista_7.exe，出现如下图界面，如图2-6所示。

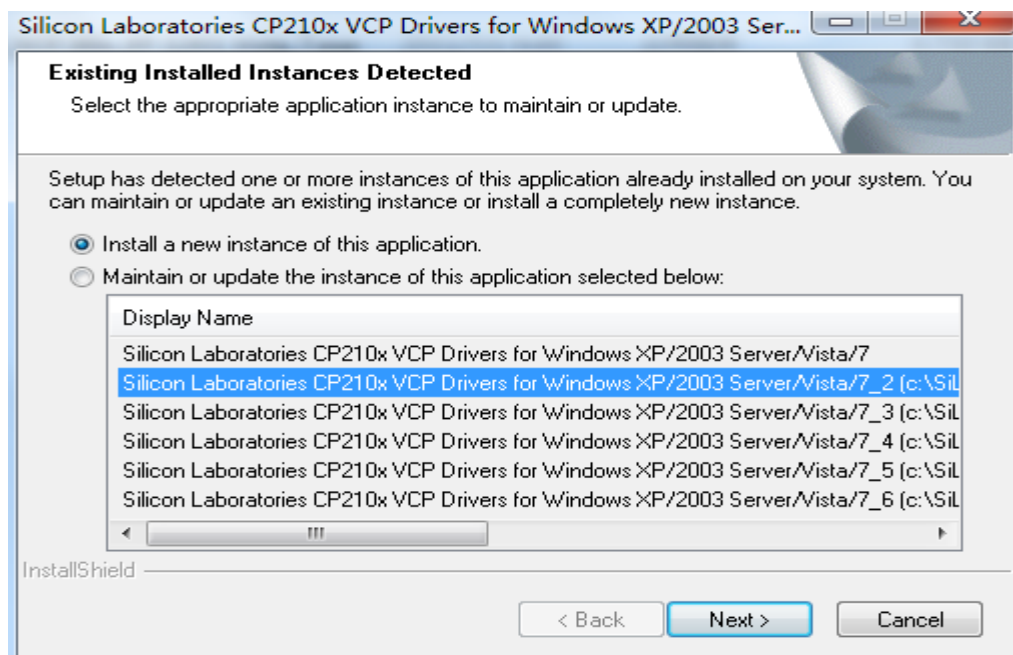


图2-6 CP210X安装步骤1

点击NEXT即可进入安装界面，如图2-7所示。

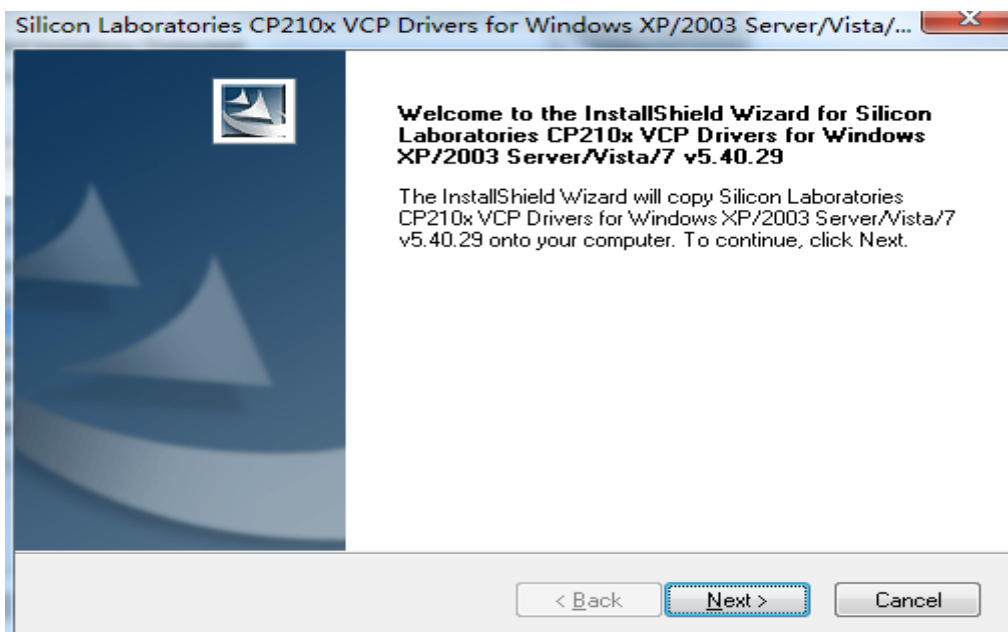


图2-7 CP210X安装步骤2

点击NEXT进入用户协议，选择同意，点Next，如图2-8所示。

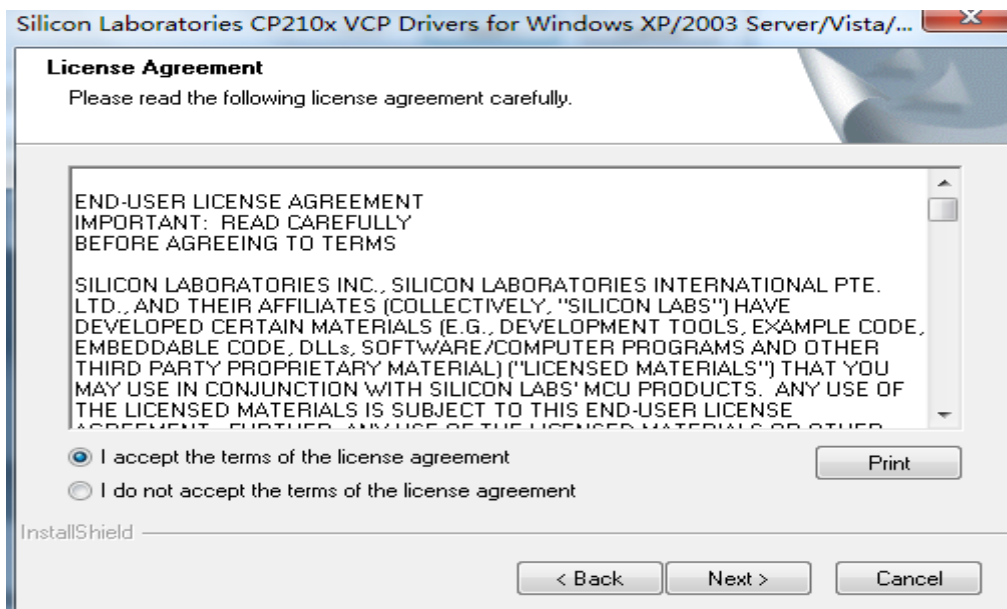


图2-8 CP210X安装步骤3

选择安装路径，这里选择默认路径，点Next继续，如图2-9所示。

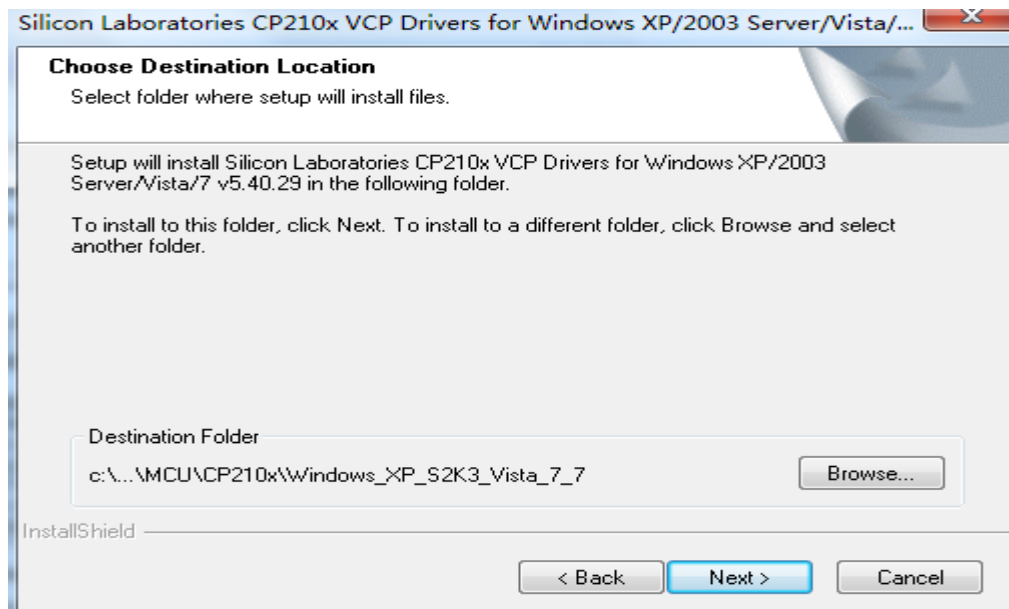


图2-9 CP210X安装步骤4

点击install开始安装，如图2-10, 2-11所示。

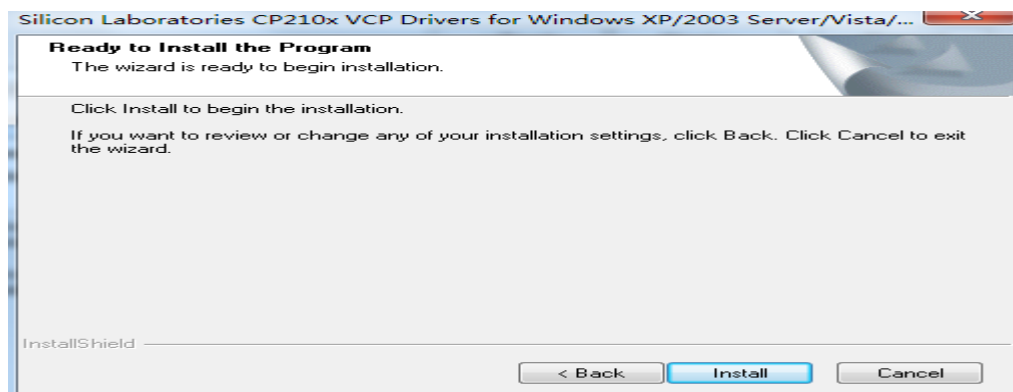


图2-10 CP210X安装步骤5

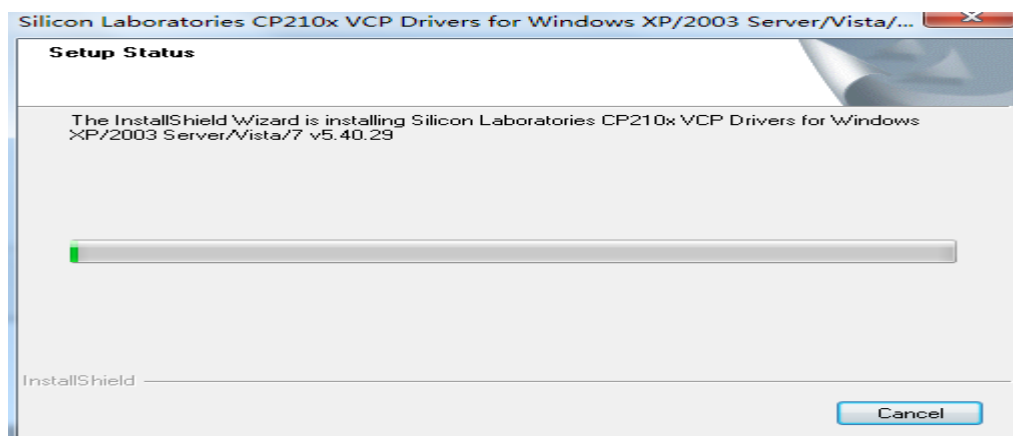


图2-11 CP210X安装步骤6

自动加载完成后，出现下面的对话框，点Finish，如图2-12所示。

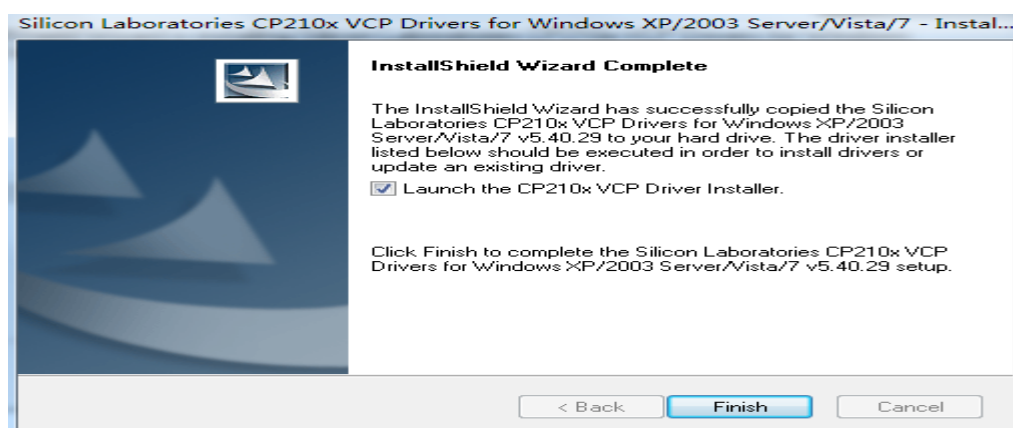


图2-12 CP210X安装步骤7

出现下面的对话框，点击install，系统将会自动安装好CP2102驱动，如图2-13所示。

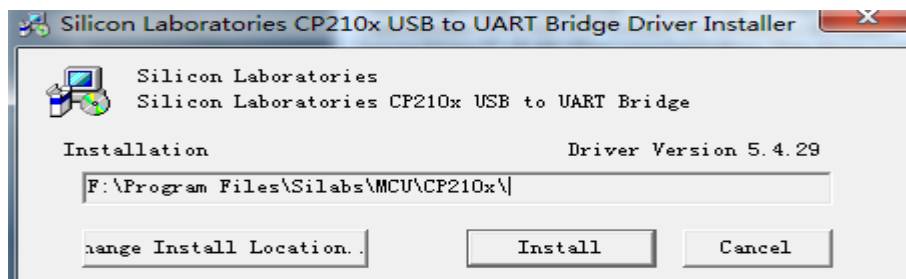


图2-13 CP210X安装步骤8

重新插拔USB接口，如果驱动安装正常，设备管理器里将会发现一个新的USB转串口端口，如图2-14所示。



图2-14 CP210X安装步骤9

在带有CP210X字样的串口后面，括号里面有提示当前虚拟串口的串口号，这个要记住，在给单片机下载程序时候，需要设置这个串口号。

5、软件设置

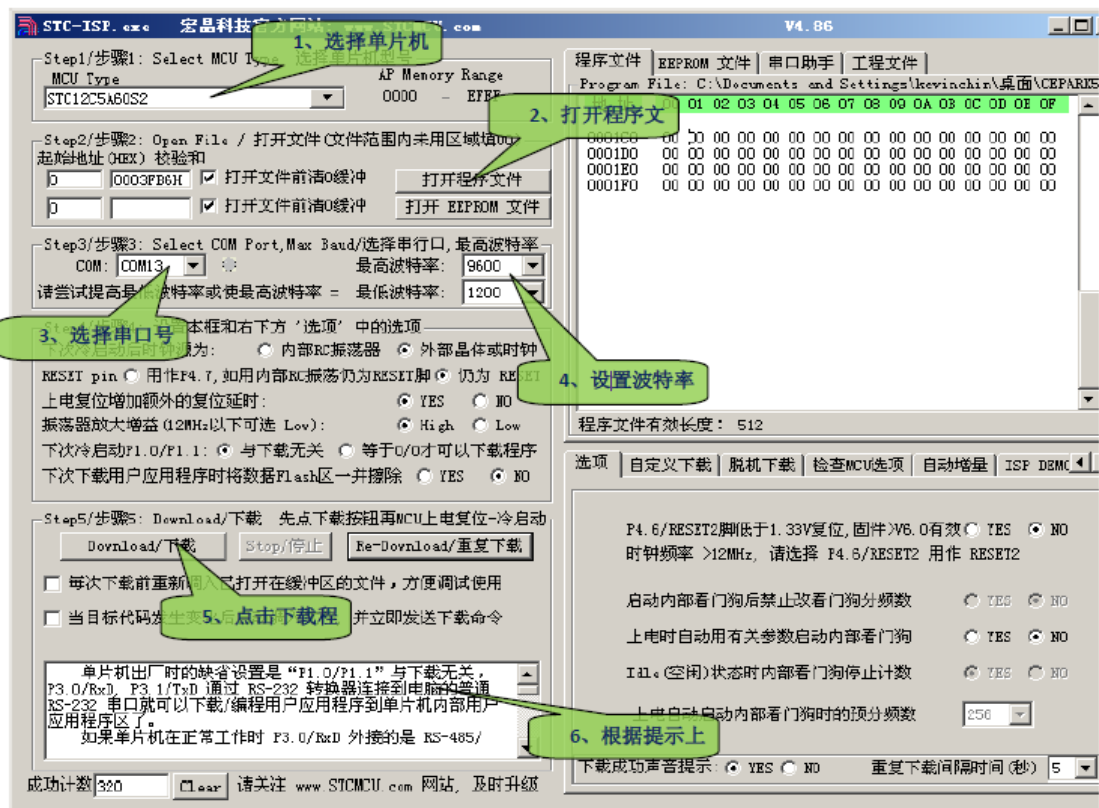


图2-15 SRC-ISP软件主界面

1) 选择单片机型号

CEPARK 畅学51单片机核心板板使用STC12C5A60S2，在图中1处选择。

2) 设置串口号

在电脑设备管理器，端口处可以看到端口号，畅学51单片机核心板板载USB-232串口芯片CP2102，安装驱动后，在设备管理里面会多一个虚拟的串口，根据实际情况选择即可。

3) 设置波特率

最低波特率设置为1200，最高波特率设置为9600。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。

6、程序下载

1) 加载程序

打开你要写入单片机的HEX程序，随板光盘中有提供配套测试程序以及配套实验程序，选择指定位置的HEX程序即可，在图中2处选择。

2) 点击下载

点击“下载”开始准备向单片机写入程序，如图2-16所示。

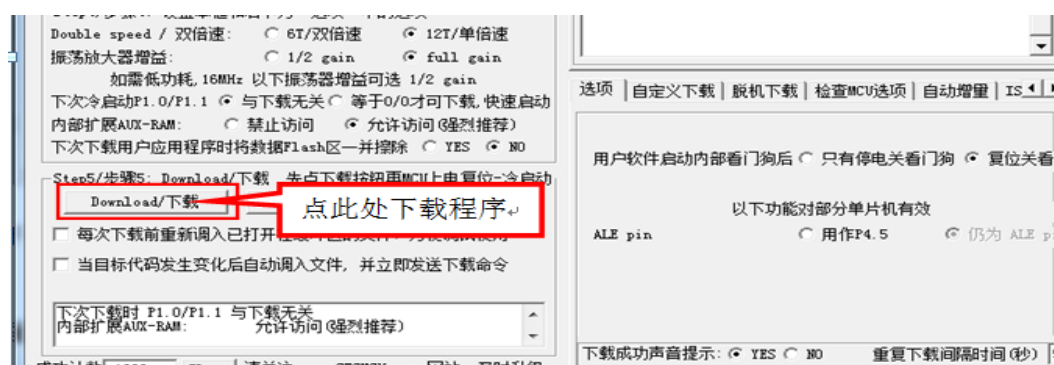


图2-16 SRC-ISP下载

3) 上电

下载时注意看提示，主要看是否要给单片机上电，如图2-17所示。

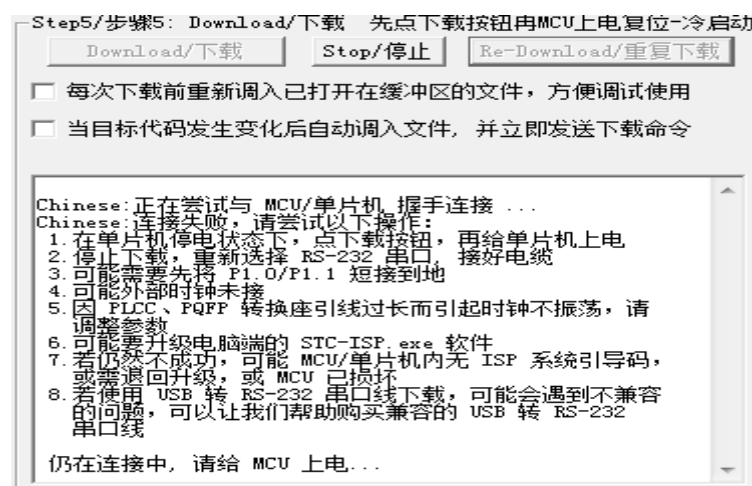
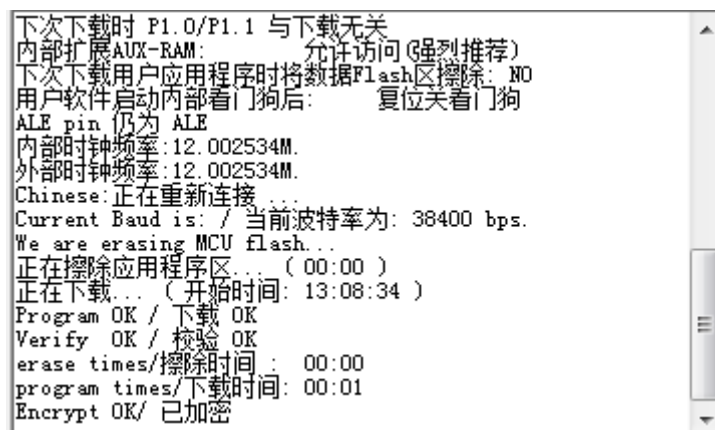


图2-16 SRC-ISP下载中

提示请上电后，给核心板断电然后重新上电，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图2-17所示：

A screenshot of a text-based message box from SRC-ISP. The text is in Chinese and English, providing status updates during a microcontroller download. It includes information about ALE pin settings, internal/external clock frequencies (12.002534M), baud rate (38400 bps), and the progress of erasing the MCU flash and downloading the program. The final status shows 'Program OK / 下载 OK', 'Verify OK / 校验 OK', 'erase times/擦除时间: 00:00', 'program times/下载时间: 00:01', and 'Encrypt OK/ 已加密'.

```
下次下载时 P1.0/P1.1 与下载无关
内部扩展AUX-RAM: 允许访问(强烈推荐)
下次下载用户应用程序时将数据Flash区擦除: NO
用户软件启动内部看门狗后: 复位关看门狗
ALE pin 仍为 ALE
内部时钟频率: 12.002534M.
外部时钟频率: 12.002534M.
Chinese: 正在重新连接...
Current Baud is: / 当前波特率为: 38400 bps.
We are erasing MCU flash...
正在擦除应用程序区... ( 00:00 )
正在下载... ( 开始时间: 13:08:34 )
Program OK / 下载 OK
Verify OK / 校验 OK
erase times/擦除时间: 00:00
program times/下载时间: 00:01
Encrypt OK/ 已加密
```

图2-17 SRC-ISP下载完成

5) 运行程序

下载完成后，按核心板上复位键运行程序。

注意：

一定要先选择“Download/下载”按钮，然后再给单片机上电复位(先关闭电源再打开电源)，而不要先上电，先上电，检测不到合法的下载命令流，单片机就直接跑用户程序了。

第三章 实验指导

实验一 IO 口实验——跑马灯

一、实验目的与要求

熟悉 Keil C51 集成环境软件的使用方法。

熟悉 MCS51 汇编指令，能自己编写简单的程序，控制硬件。

熟悉畅学开发平台，掌握单片机最小系统及 IO 口的简单控制及不同组 IO 口的特点。

二、实验设备

CEPARK 畅学系列开发平台一套、PC 机一台。

三、实验内容

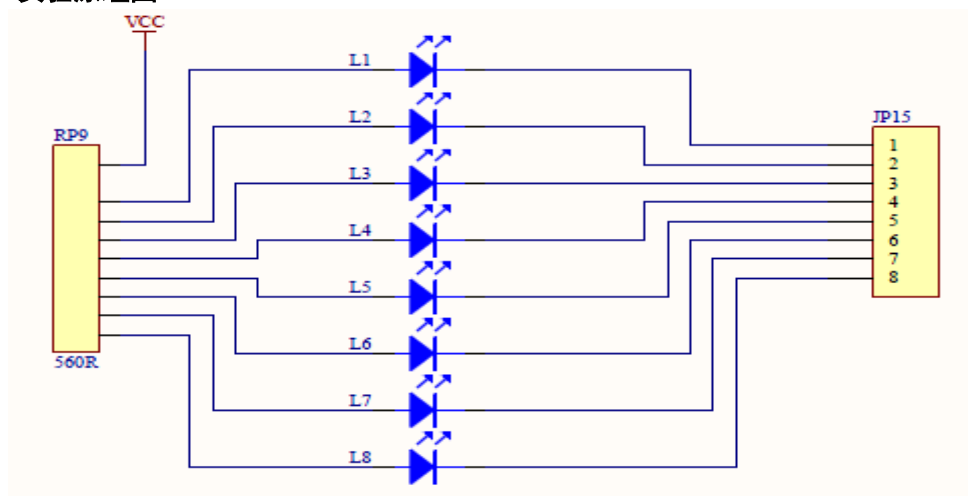
1、熟悉 Keil C51 集成环境软件的安装和使用方法。

2、照接线图编写程序，完善给出程序，实现以下功能。

要求 LED 灯实现从上到下依次点亮，再从下到上依次点亮。完成后实现从两侧向中间循环点亮，再从中间往外循环点亮。功能完成后循环上述过程。

3、观察实验结果，验证程序是否正确。

四、实验原理图



五、实验步骤

1、连线说明：

JP4 (P1 口)	——	JP15
------------	----	------

2、编写程序或运行参考程序。

3、实验结果：通过单色流水灯区的 LED 指示灯（8 个指示灯往复点亮），观察实验的输出结果是否正确。

六、演示程序（部分，请完善）

```

ORG 0000H
    LJMP    START
ORG 0100H
    START:
        MOV    A , #0FFH
        CLRC
        MOV    R4 , #08H
    LOOP1:
        RRC    A
        MOV    P1 , A
        LCALL  DELAY
        DJNZ   R4 , LOOP1
        MOV    A , #0FFH
        CLR    C
        MOV    R4 , #08H
    LOOP2:
        RLCA
        MOV    P1 , A
        LCALL  DELAY
        DJNZ   R4 , LOOP2
        JMP    START

    DELAY:
        MOV    R1 , #20H
    D0:
        MOV    R2 , #00H
    D1:
        MOV    R3 , #00H
        DJNZ   R3 , $
        DJNZ   R2 , D1
        DJNZ   R1 , D0
        RET

END

```

七、思考

- 1、循环间隔时间通过延时怎么控制？
- 2、跑马灯模式如何实现间隔亮、多位同时亮等其他效果？

实验二 键盘显示实验——密码键盘

一、实验目的与要求

掌握 4*4 矩阵键盘的驱动方法，掌握并理解通过 74HC595 串转并芯片如何驱动数码管的方法，以及在控制过程中如何实现数码管不同的显示方式。

认真预习，做好实验前的准备工作，自行编写程序，填写实验报告

二、实验设备

CEPARK 畅学系列开发平台一套、PC 机一台。

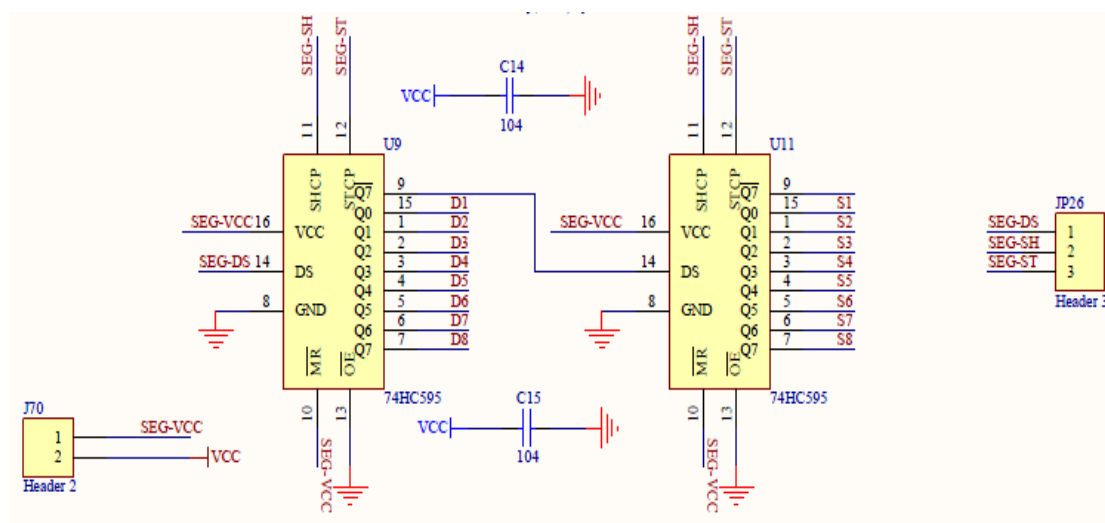
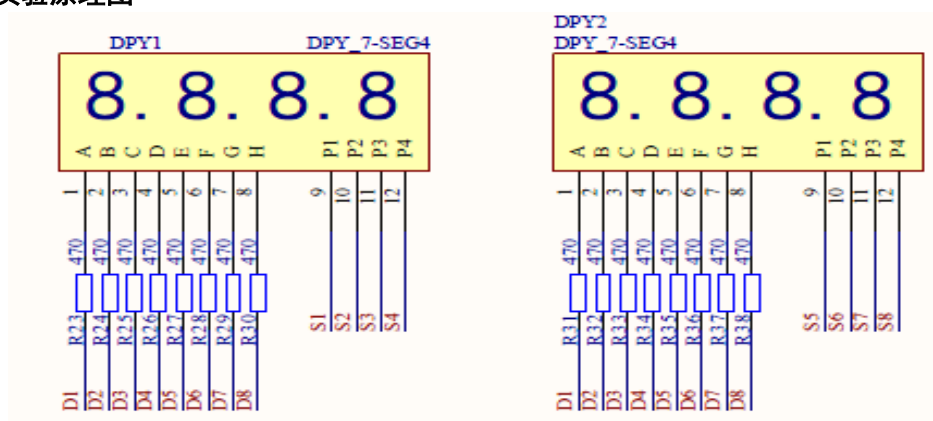
三、实验内容

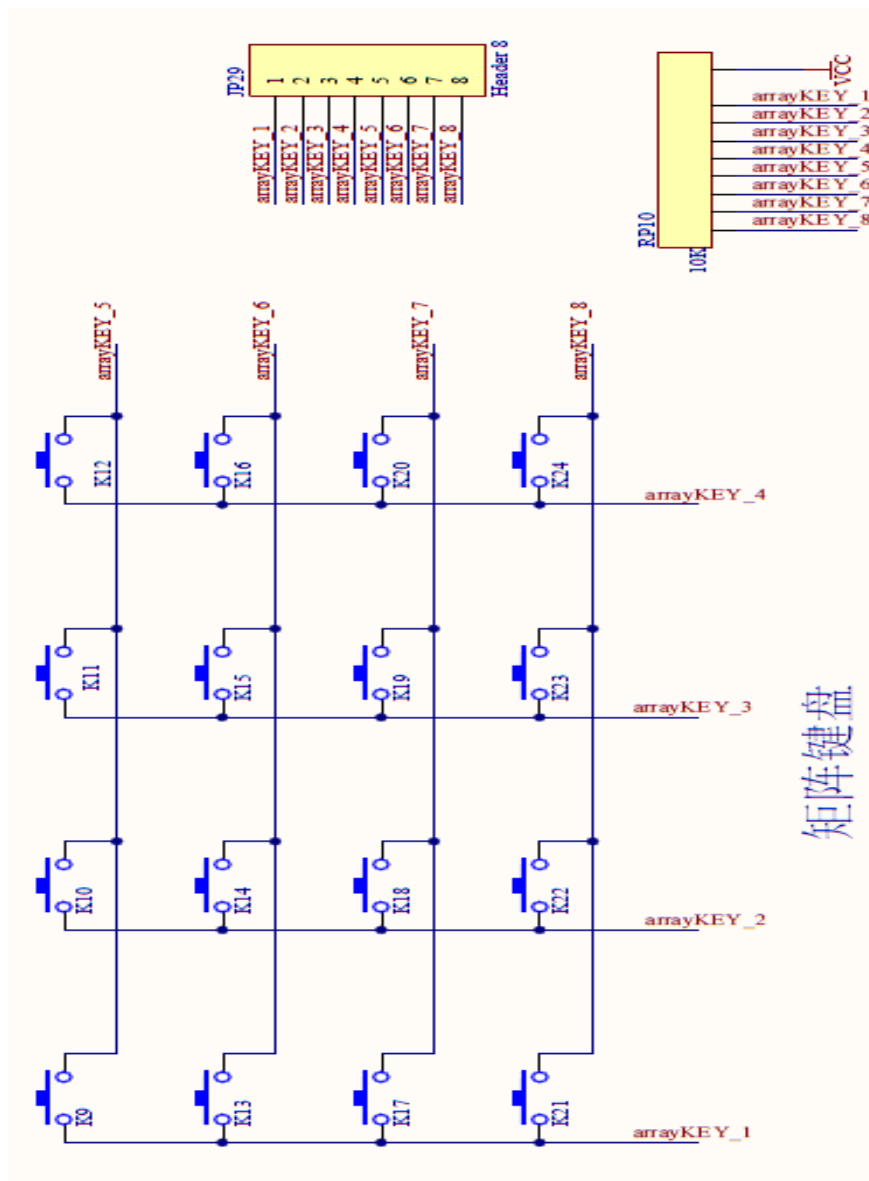
1、编写程序：通过行列反转法驱动 4*4 矩阵键盘，并通过 74HC595 通过 LED 数码管，同时将键盘扫描键值显示在数码管上。

2、按图连线，运行程序，观察实验结果，能熟练掌握键盘和数码管的驱动方法，为后续试验做准备。

3、要求以汇编编写该程序，可以尝试采用 C51 编写。

四、实验原理图





五、实验步骤

1、连线说明：

JP4 (P1 口)	——	JP29
JP2 (P2. 0)	——	JP26 (STCP)
JP2 (P2. 1)	——	JP26 (SHCP)
JP2 (P2. 2)	——	JP26 (DS)
J70	——	短路帽

2、编写程序或运行参考程序。

3、实验结果：查看按键键值是否能正确显示在数码管上，同时观察按键扫描程序是否具备较好的灵敏度和容错性。

六、演示程序

```
;          P1.0-P1.3为列线,P1.4-P1.7为行线
;-----定义变量-----
        DS_595    EQU P2.2
        SHCP_595  EQU P2.1
        STCP_595  EQU P2.0
        DIS_DAT    EQU 31H
        DIS_ADDR   EQU 32H
        KEY_NUM    EQU 30H

;-----定义地址-----
ORG     0000H
        LJMP      MAIN
ORG     0030H
;-----主函数-----
MAIN:
        LCALL     KEY_IN
        MOV        DIS_DAT,KEY_NUM
        MOV        DIS_ADDR,#00H
        LCALL     SEND_LED
        CALL       DELAY
        LJMP      MAIN

;-----固定位置送显示-----
;32H:    位选地址
;31H:    段选数据
SEND_LED:
        MOV        A,DIS_ADDR
        MOV        DPTR,#SEG_ADDR
        MOVC       A,@A+DPTR
        LCALL     OUT_595
        MOV        A,DIS_DAT
        MOV        DPTR,#SEG_DAT
        MOVC       A,@A+DPTR
        LCALL     OUT_595
        RET
```

```

;-----595输出-----
OUT_595:
    LCALL WR_595    ;写595
    CLRSTCP_595     ;存储寄存器输入
    NOP
    NOP
    SETB    STCP_595
    NOP
    NOP
    CLRSTCP_595
    RET

;-----写595-----
WR_595:
    MOV     R5,#8    ;数码管的8位
LOOP1:
    RLC     A
    MOV     DS_595,C ;送数据进595
    SETB    SHCP_595 ;移位寄存器时钟，根据DATASHEET，送数据
进去要一个SCK时钟周期。
    NOP
    NOP
    CLR     SHCP_595
    DJNZ    R5,LOOP1 ;8位数据是否送完
    RET

;-----DELAY-----
DELAY:
    MOV     R2,#02H
L0:
    MOV     R6,#00H
L1:
    MOV     R7,#00H
L2:
    DJNZ    R7,L2
    DJNZ    R6,L1
    DJNZ    R2,L0
    RET

;-----清显示-----
DIS_CLR:

```

```

        MOV    R3,#00H
LOOP_DIS:  INCR3
        MOV    DIS_ADDR,R3
        MOV    DIS_DAT,#0FFH
        LCALL  SEND_LED
        CJNE   R3,#08H,LOOP_DIS
        RET

;-----按键扫描-----
KEY_IN:   MOV   P1,#0F0H           ;置列线为0，行线为1
        MOV   A,P1
        ANL   A,#0F0H
        MOV   B,A
        MOV   P1,#0FH           ;置列线为1，行线为0
        MOV   A,P1
        ANL   A,#0FH
        ORL   A,B               ;高四位与低四位重新组合
        CJNE  A,#0FFH,KEY_IN1    ;0FFH为末按键
        RET
KEY_IN1:  MOV   B,A
        MOV   DPTR,#KEYTABLE
        MOV   R3,#0FFH          ;
KEY_IN2:  INC   R3
        MOV   A,R3
        MOVC  A,@A+DPTR
        CJNE  A,B,KEY_IN3
        MOV   A,R3              ;找到，取顺序码
        MOV   KEY_NUM,A
        RET
KEY_IN3:  CJNE  A,#0FFH,KEY_IN2  ;未完，继续查
        RET                    ;0FFH为结束码

;-----
SEG_DAT   : DB
0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H,88H,83H,0C6H,0A1H,86H,8EH
SEG_ADDR: DB 01H,02H,04H,08H,10H,20H,40H,80H,0FFH,00H
KEYTABLE:
        DB   0EEH,0EDH,0EBH,0E7H,0DEH  ;0,1,2,3,4, 顺序码
        DB   0DDH,0DBH,0D7H,0BEH,0BDH  ;5,6,7,8,9,

```

```
DB 0BBH,0B7H,07EH,07DH,07BH ;A,B,C,D,E,
DB 077H,0FFH ;F 0FF为结束码
END
```

七、实验扩展及思考

- 1、在按键扫描过程中如何实现对组合键的识别？如Ctrl+V之类的识别。
- 2、在按键扫描过程中如何实现一键功能复用的识别？如单击、双击如何识别？
- 3、如何实现动态显示过程中闪烁显示等特定显示效果？
- 4、请分析单片机IO口扫描键盘方式下如何提高按键的灵敏度及容错性？

实验三 定时器/中断实验——电子钟

一、实验目的

熟悉 MCS51 类 CPU 的定时器、中断系统编程方法，了解定时器的应用、实时程序的设计和调试技巧。

二、实验设备

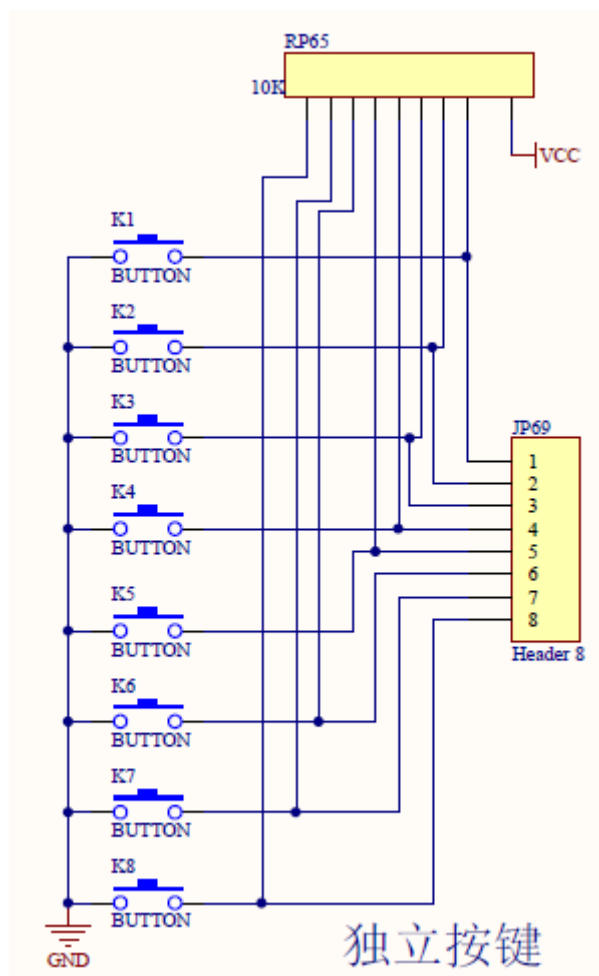
CEPARK 畅学系列开发平台一套、PC 机一台。

三、实验内容

1、编写一个时钟程序，使用单片机定时器 1 产生一个 50ms 的定时中断，对定时中断计数，将时、分、秒显示在数码管上。

2、将单片机的 P3.5 口与独立按键 K1 相连。要求通过计数器对按键按压次数进行计数，并将计数结果显示在 LED 上，或者通过外部中断实现按键计数。要求显示范围为 0-99。记满后从零开始重新计数。

四、实验原理图(8 位 LED 显示原理图见实验 3)



五、实验步骤

1、连线说明：

JP4 (P3. 5)	——	JP69 (K1)
JP2 (P2. 0)	——	JP26 (STCP)
JP2 (P2. 1)	——	JP26 (SHCP)
JP2 (P2. 2)	——	JP26 (DS)
J70	——	短路帽

2、编写程序或运行参考程序。

3、实验结果：注意单独按键扫描时如何提高扫描的准确性，以及不产生多次计数的问题。

六、演示程序(略)

七、实验扩展及思考

- 1、动态扫描LED时如何解决闪屏及拖影的现象？
- 2、通过计数器如何实现外部中断的效果？
- 3、在动态显示过程中如何解决单独按键扫描的问题？

实验四 定时器综合实验——PWM 调光灯

一、实验目的

熟悉 MCS51 中断系统的工作原理和编程方法，掌握定时器初值控制方法，深入理解 PWM 信号的特点，以及占空比调节方法。

二、实验设备

CEPARK 畅学系列开发平台一套、PC 机一台。

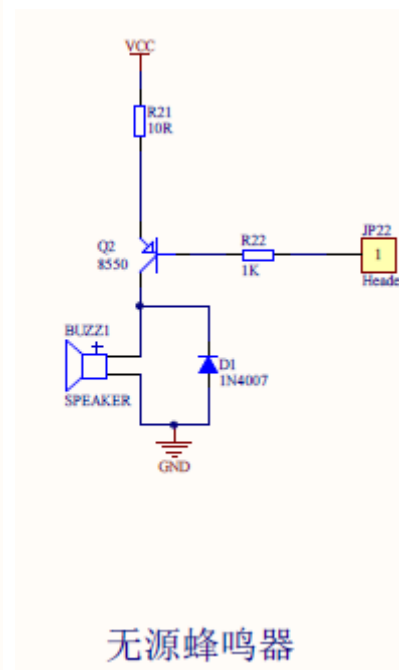
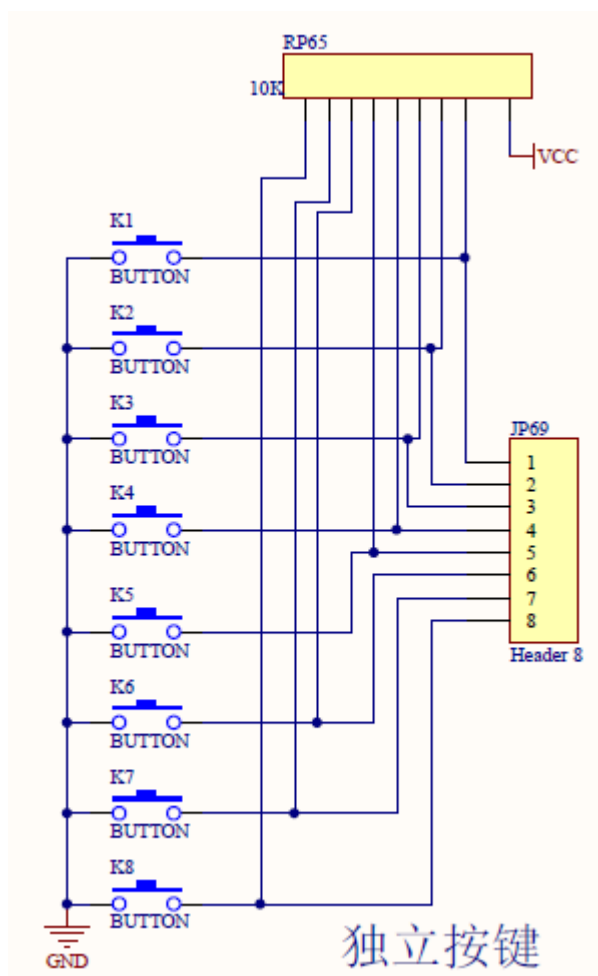
三、实验内容

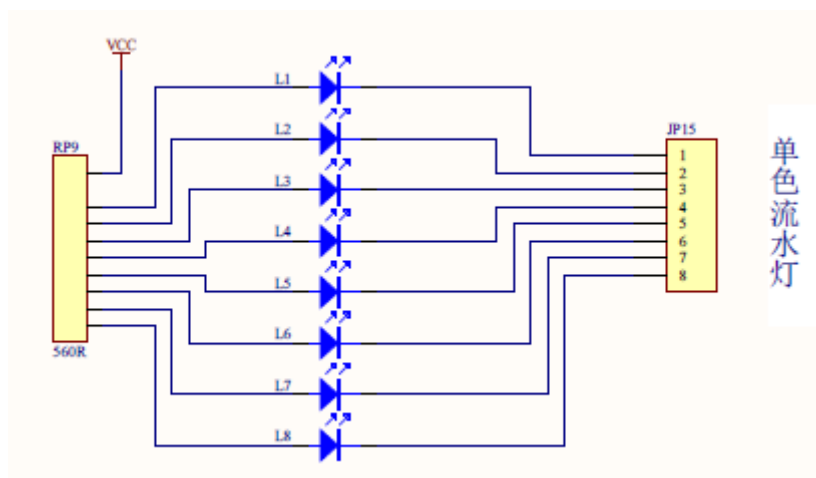
1、单片机定时器 T0 产生一个频率为 10HZ 的 PWM 脉冲信号，占空比可调，并在 P1.0 口和 P1.1 口同时输出该脉冲信号，并分别驱动一位 LED 灯和蜂鸣器。

2、对应 5 个独立按键，分别为 1 号、2 号、3 号、4 号、5 号和 6 号键，按下 1 号键，PWM 脉冲占空比为 0%，2 号—5 号键按下对应输出占空比为 20%、40%、60%和 80%，而按下 6 号键时输出占空比为 100%，上电默认占空比为 50%。

3、通过 LED 灯的亮度以及蜂鸣器的声调大小来判别调光成功与否。

四、实验原理图(8 位 LED 显示原理图见实验 3)





五、实验步骤

1、连线说明：

JP1 (P0. 0)	——	JP69 (K1)	6 个独立按键 控制 PWM 信号 占空比
JP1 (P0. 1)	——	JP69 (K2)	
JP1 (P0. 2)	——	JP69 (K3)	
JP1 (P0. 3)	——	JP69 (K4)	
JP1 (P0. 4)	——	JP69 (K5)	
JP1 (P0. 5)	——	JP69 (K6)	
JP3 (P1. 1)	——	JP22	无源蜂鸣器
JP2 (P1. 0)	——	JP15 (L1)	LED 灯

2、编写程序或运行参考程序。

3、实验结果：注意调节占空比过程中，定时初值的合理选择。

六、演示程序(略)

七、实验扩展及思考

1、定时初值选择和占空比调节精度的关系？

2、PWM信号频率和占空比调节精度的关系？

3、如果要求占空比的调节以1%为梯度，该如何实现？以及实现过程中对信号频率的要求是什么？

实验五 串口通信实验——远程控制

一、实验目的

掌握单片机串行口的工作原理，掌握串口通信程序设计方法，了解 MAX232 芯片的工作原理。

二、实验设备

CEPARK 畅学系列开发平台一套、PC 机一台, 串口线一根。

三、实验内容

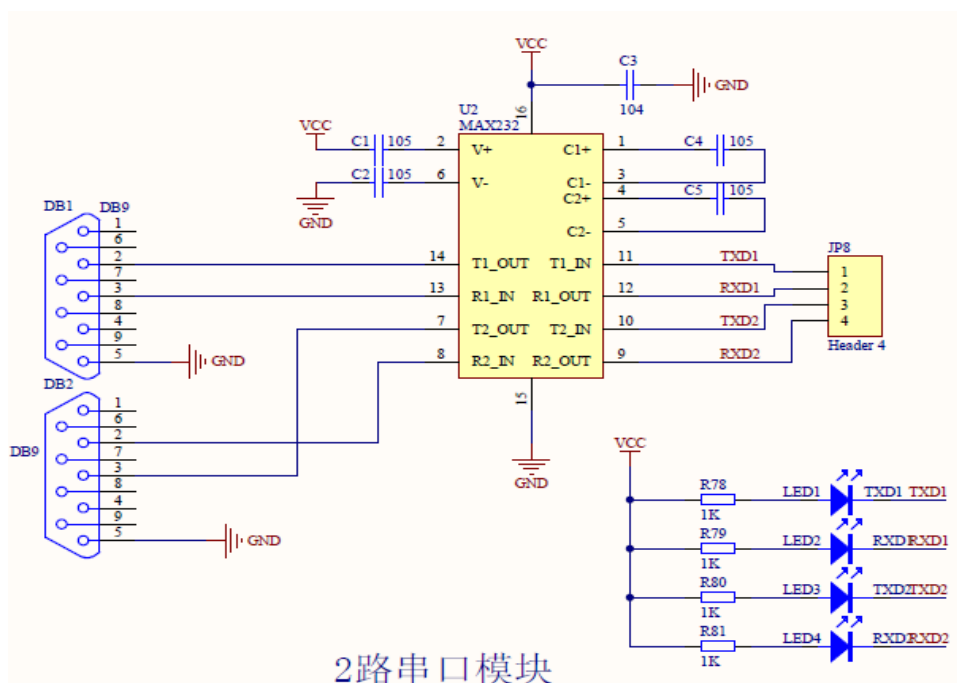
1、利用 PC 机的串口调试助手 (STC-ISP 中有集成) 通过串口向单片机发送一个数据 (0-99) 作为单片机倒计时的起始时间 (秒)，单片机收到该信号后启动倒计时程序，倒计时时间显示在 LED 数码管上，当倒计时完成后蜂鸣器响 2-5s 以起到提示作用，波特率自定。

2、倒计时过程中如果收到串口新数据，应该结束当前倒计时计数，以新数据重新开始计数。

3、同时单片机外部中断 0 外接一位独立按键，统计独立按键的单击次数，将单击次数显示在数码管上，显示范围为 0-99。并每隔一段时间将单击次数发送给 PC 机，通过串口助手观察结果。

4、可以通过 51 汇编或 C51 编程实现。

四、实验原理图 (8 位 LED 显示原理图见实验 3)



五、实验步骤

1、连线说明：

JP4 (P3. 0)	——	JP8 (RX1)
JP4 (P3. 1)	——	JP8 (TX1)
JP4 (P1. 0)	——	JP22
JP2 (P2. 0)	——	JP26 (STCP)
JP2 (P2. 1)	——	JP26 (SHCP)
JP2 (P2. 2)	——	JP26 (DS)
J70	——	短路帽

2、编写程序或运行参考程序。

3、实验结果：注意波特率的设置问题。

六、演示程序(部分)

//12MHZ 9600波特率，定时器1T方式，串口1，波特率倍数开启

void UART_Init(void)

```
{
    PCON |= 0x80;          //使能波特率倍速位SMOD
    SCON = 0x50;           //8位数据,可变波特率
    AUXR |= 0x40;          //定时器1时钟为Fosc,即1T
    AUXR &= 0xfe;          //串口1选择定时器1为波特率发生器
    TMOD &= 0x0f;          //清除定时器1模式位
    TMOD |= 0x20;          //设定定时器1为8位自动重装方式
    TL1 = 0xB2;            //设定定时初值
    TH1 = 0xB2;            //设定定时器重装值
    ET1 = 0;              //禁止定时器1中断
    TR1 = 1;              //启动定时器1
}
```

七、实验扩展及思考

- 1、串行通信的波特率对实现收发有什么影响？
- 2、如何实现多位字符同时发送和接收(实现帧收发)？

实验六 液晶显示电子钟

一、实验目的

掌握液晶 LCD12864 的驱动原理，掌握 DS1302 时钟芯片的使用方法，掌握综合程序的写法。

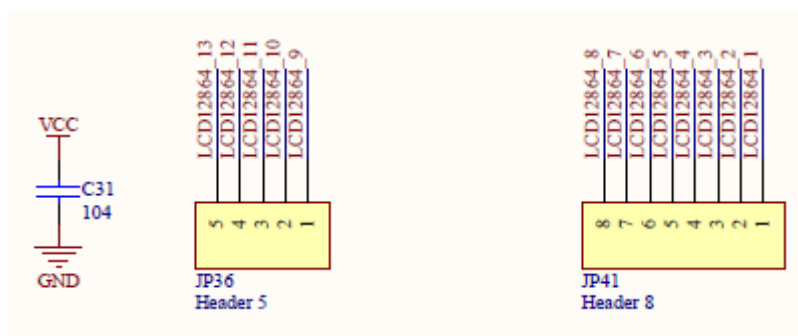
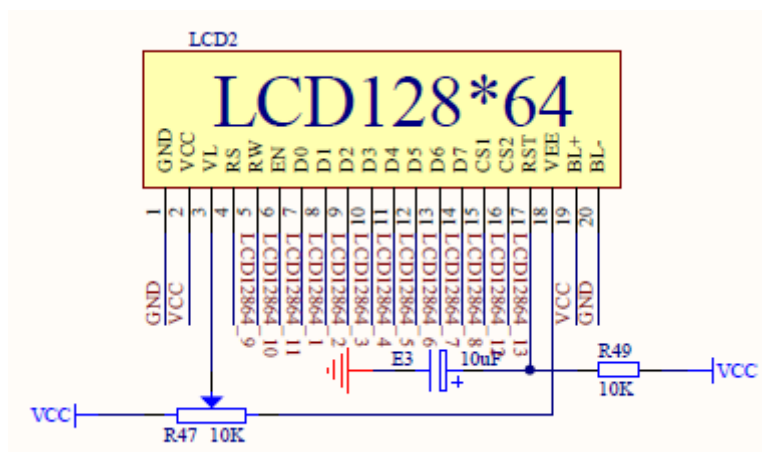
二、实验设备

CEPARK 畅学系列开发平台一套、PC 机一台, 串口线一根。

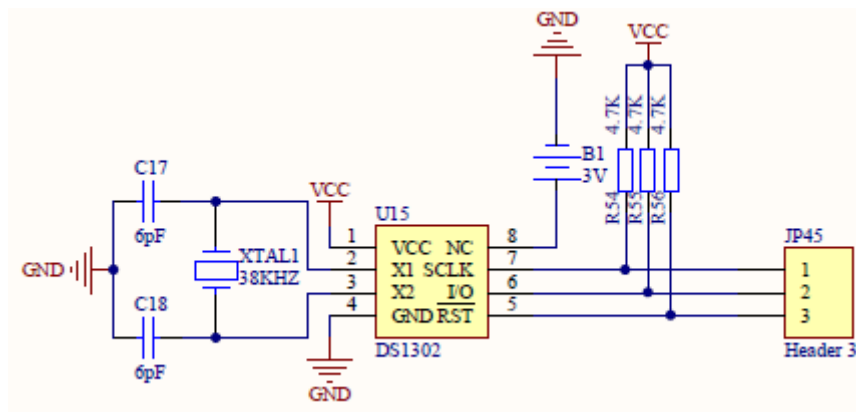
三、实验内容

- 1、液晶屏上显示时钟，时钟上可以显示年月日、时分秒。时间要求通过 DS1302 时钟芯片实现。
- 2、在主界面可以通过按键进入设置界面，在设置界面要求可以设置时间的初值以及重要日期提醒设置。如生日、会议、考试等重要日期可以设置。
- 3、提醒日期到时间后通过蜂鸣器提醒，同时在液晶屏上通过文字提醒。
- 4、可以通过 51 汇编或 C51 编程实现。

四、实验原理图



液晶 LCD12864 显示电路



DS1302 电路

五、实验步骤

1、连线说明：

	——	JP41
	——	JP36
	——	JP45

2、编写程序或运行参考程序。

3、实验结果。

六、演示程序(无)

七、实验扩展及思考

- 1、DS1302 设置初值的注意事项？
- 2、如何在液晶屏上实现模拟时钟，如表带？

实验七 智能家居控制系统

一、实验目的

了解几种类型 AD 转换的工作原理，理解并行和串行的优缺点，掌握 PCF8591 的驱动方法。

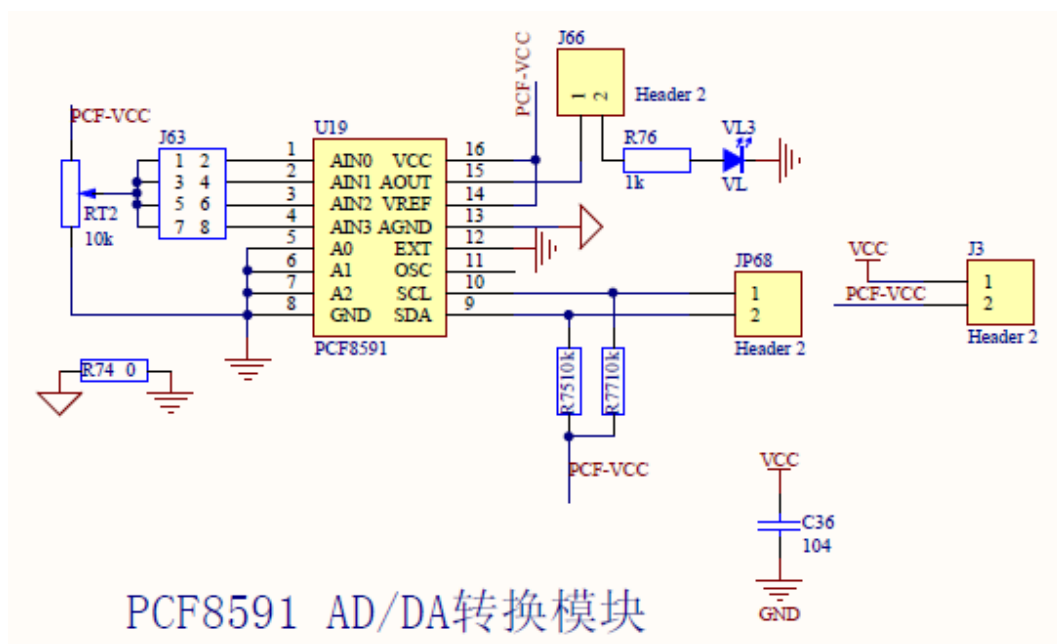
二、实验设备

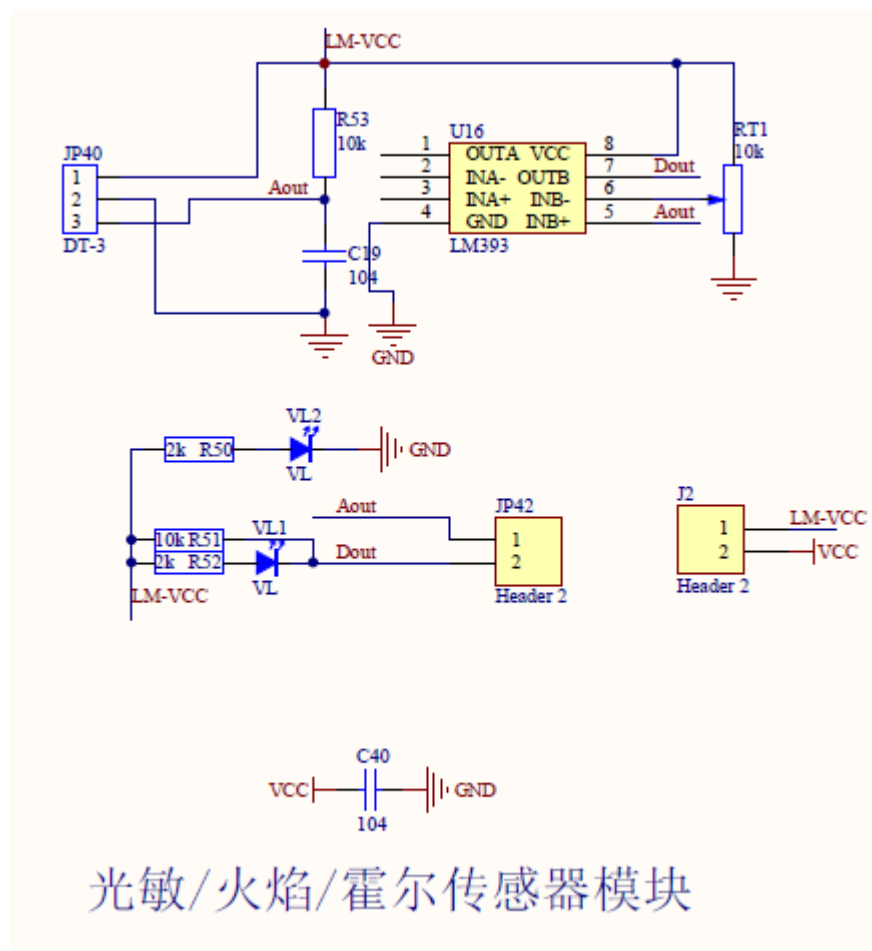
CEPARK 畅学系列开发平台一套、PC 机一台。

三、实验内容

- 1、通过 PCF8591 采样光敏电阻的阻值，并转换为对应的光照强度(Lux)。
- 2、将测量结果显示在数码管上。
- 3、通过按键可以设定光照阈值（上限和下限），光照低于下限时，关闭窗帘、室内开灯。光照高于上限时，打开窗帘、同时光灯。窗帘的开关通过步进电机模拟。
- 4、同时通过 DS18B20 采样室内温度，显示在数码管上。

四、实验原理图(8 位 LED 显示原理图见实验 3)





五、实验步骤

1、连线说明：

JP4 (P1. 1)	——	JP68 (SCL)	PCF8591 控制信号
JP4 (P1. 0)	——	JP68 (SDA)	
J3	——	短路帽	PCF8591 电源接入
JP2 (P2. 0)	——	JP26 (STCP)	HC595 信号线
JP2 (P2. 1)	——	JP26 (SHCP)	
JP2 (P2. 2)	——	JP26 (DS)	
J70	——	短路帽	LED 数码管电源接入

2、编写程序或运行参考程序。

3、实验结果：注意 PCF 串行采样时序(IIC)。

六、演示程序(部分)

//IIC时序

/**i2c 操作函数**/

void i2c_start(void) //起始条件

```
{
    sda=1;
    delay5us();
    scl=1;
    delay5us();
    sda=0;
    delay5us();
    scl=0;
    delay5us();
}
```

void i2c_stop(void) //停止条件

```
{
    sda=0;
    delay5us();
    scl=1;
    delay5us();
    sda=1;
    delay5us();
}
```

//向总线写一字节,并返回有无应答

unsigned char i2c_write(unsigned char c)

```
{
    unsigned char i;
    unsigned char ack=1;
    for(i=0;i<8;i++)
    {
        if(c&0x80)
        {
            sda=1;
        }
        else
```

```

    {
        sda=0;
    }
    delay5us();
    scl=1;
    delay5us();
    scl=0;
    c<<=1;
    delay5us();
}
sda=1;
delay5us();
scl=1;
delay5us();
if(sda==1)
{
    ack=0;    //失败,无应答
}
scl=0;
delay5us();
return ack;
}

//读一字节 ack=1 时应答,ack=0 时不应答
unsigned char i2c_read(unsigned char ack)
{
    unsigned char ttt=0;
    unsigned char i;

    sda=1;
    for(i=0;i<8;i++)
    {
        delay5us();
        scl=0;
        delay5us();
        scl=1;
        delay5us();
        ttt<<=1;
    }
}

```

```

        if(sda==1)
        {
            ttt++;
        }
    }
    delay5us();
    scl=0;
    delay5us();
    if(ack)        //非应答
    {
        sda=0;
    }
    else           //应答
    {
        sda=1;
    }
    delay5us();
    scl=1;
    delay5us();
    scl=0;
    delay5us();
    return(ttt);
}

//11059200Hz  stc 1T 时钟  5uS 延时函数
void delay5us(void)    //误差 -0.026765046296us
{
    unsigned char a,b;
    for(b=3;b>0;b--)
        for(a=2;a>0;a--);
    _nop_();    //if Keil,require use intrins.h
}

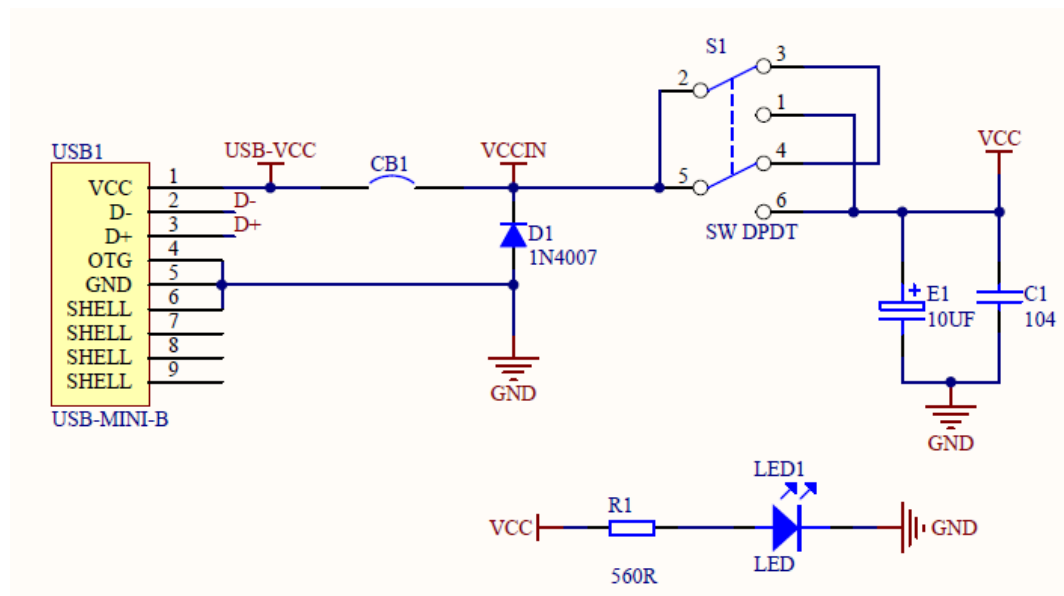
```

七、实验扩展及思考

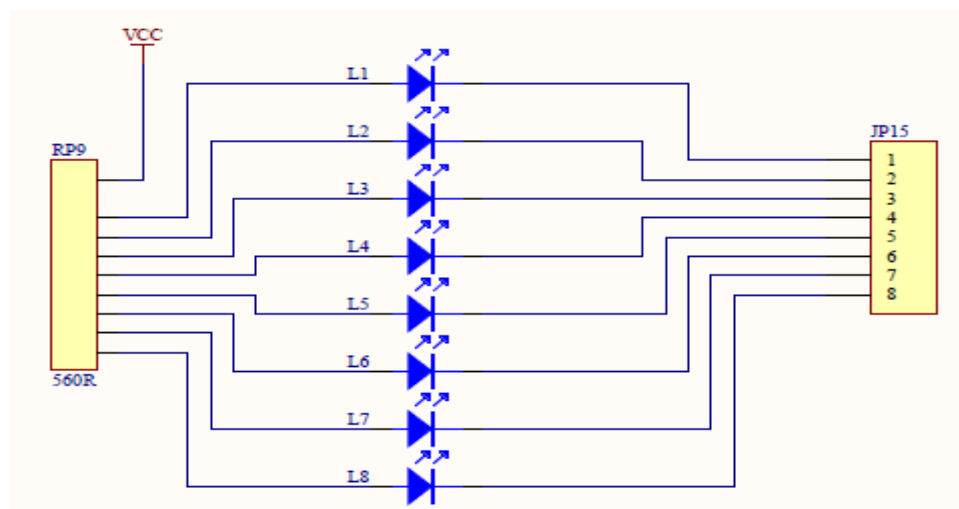
- 1、IIC 总线时序如何精确控制，如何准确算出 C51 延时程序？
- 2、PCF8591 如何实现 4 通道轮训采样？
- 3、光敏电阻的阻值变化与光照强度之间的关系。

附录一 试验箱结构及电路结构

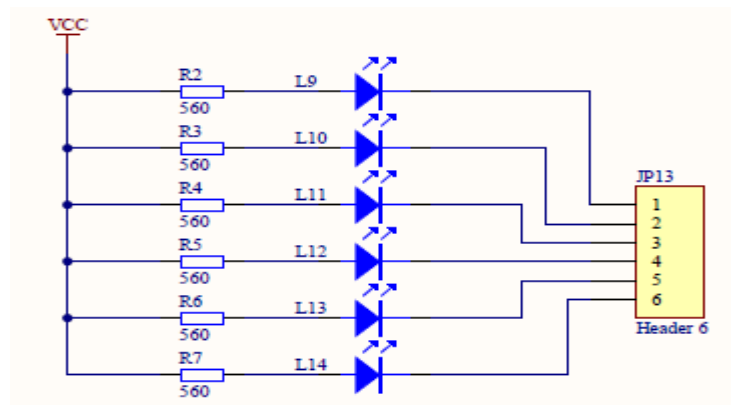
附 1 51 核心板电路电源电路



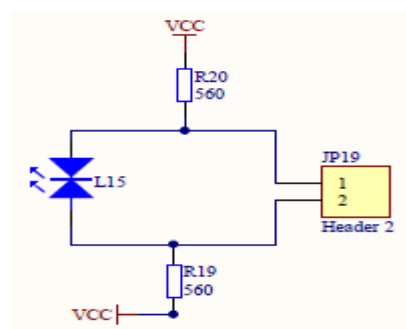
附 2 底板单色流水灯



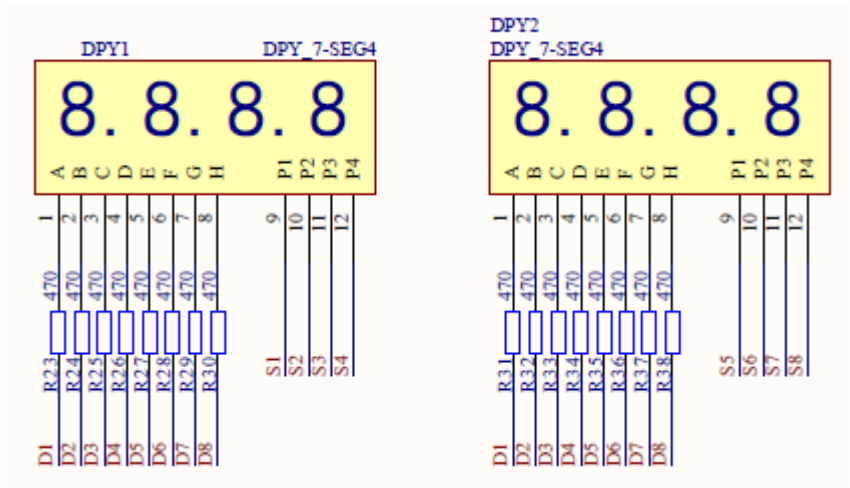
附 3 底板彩色流水灯



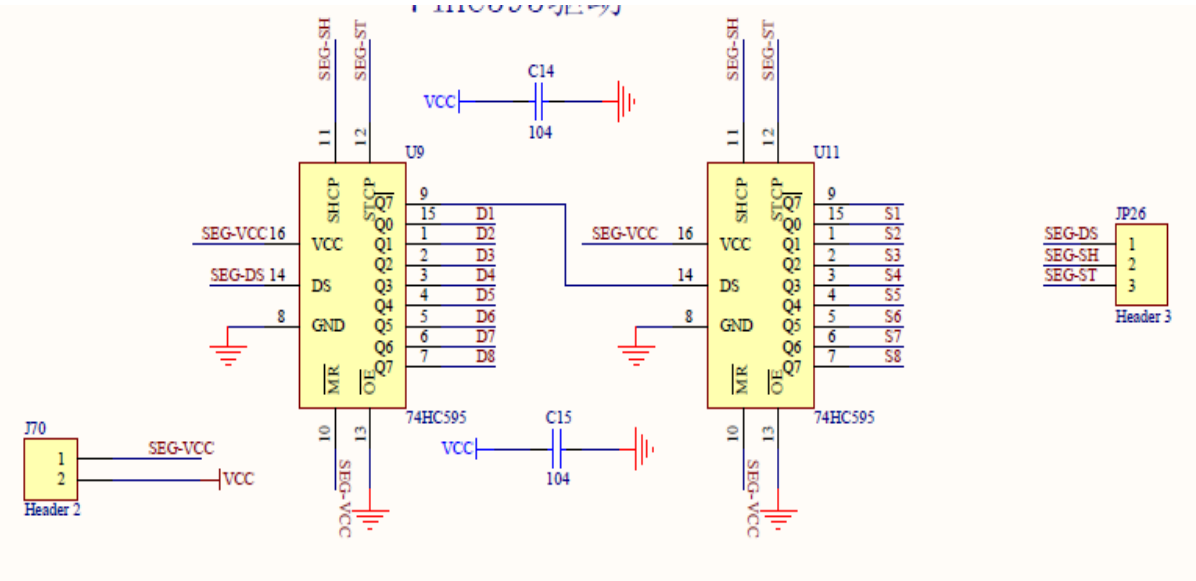
附 4 双色二极管



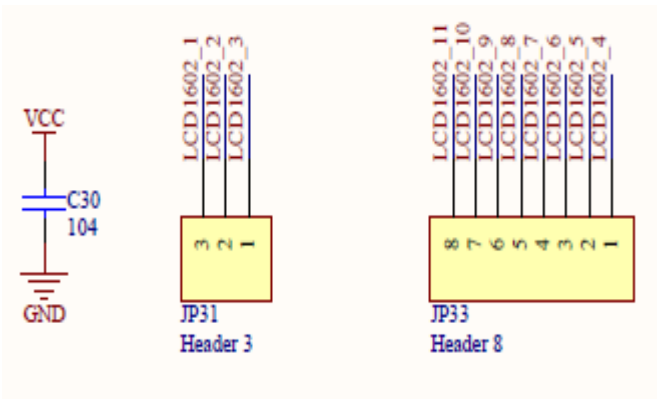
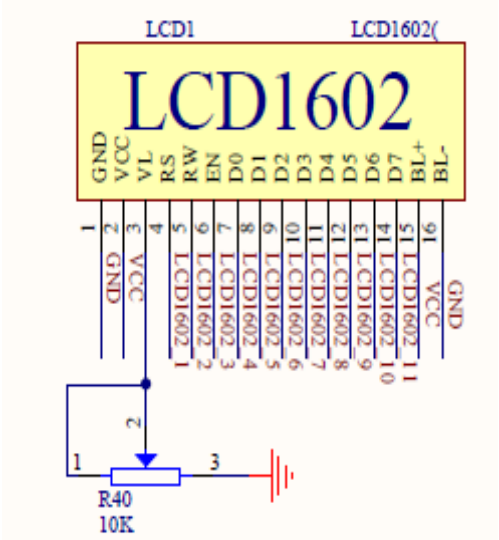
附 5 四位一体共阳数码管



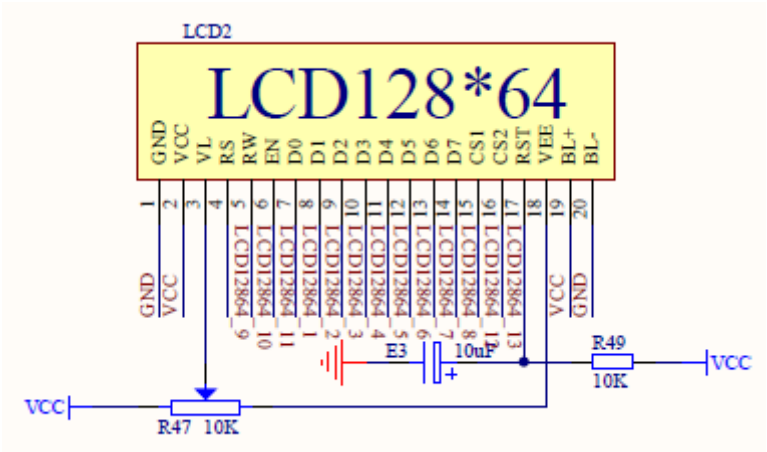
附 6 74HC595 驱动

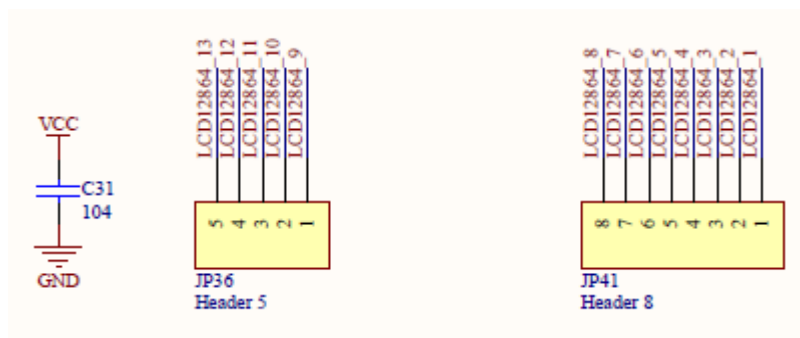


附 7 LCD1602 液晶

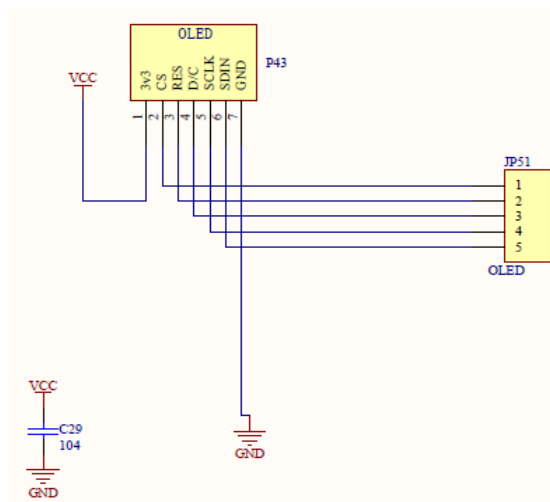


附 8 LCD12864 液晶

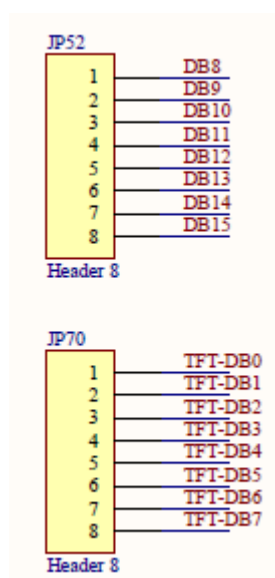


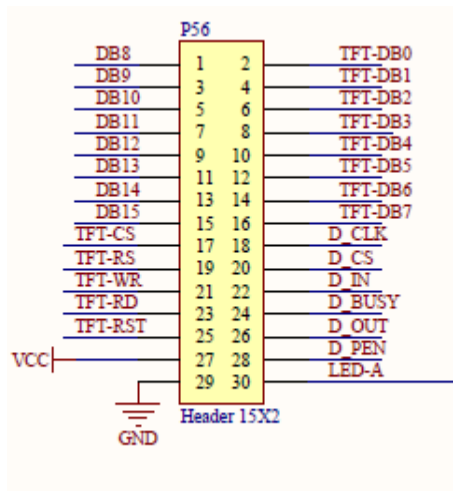


附 9 OLED 液晶

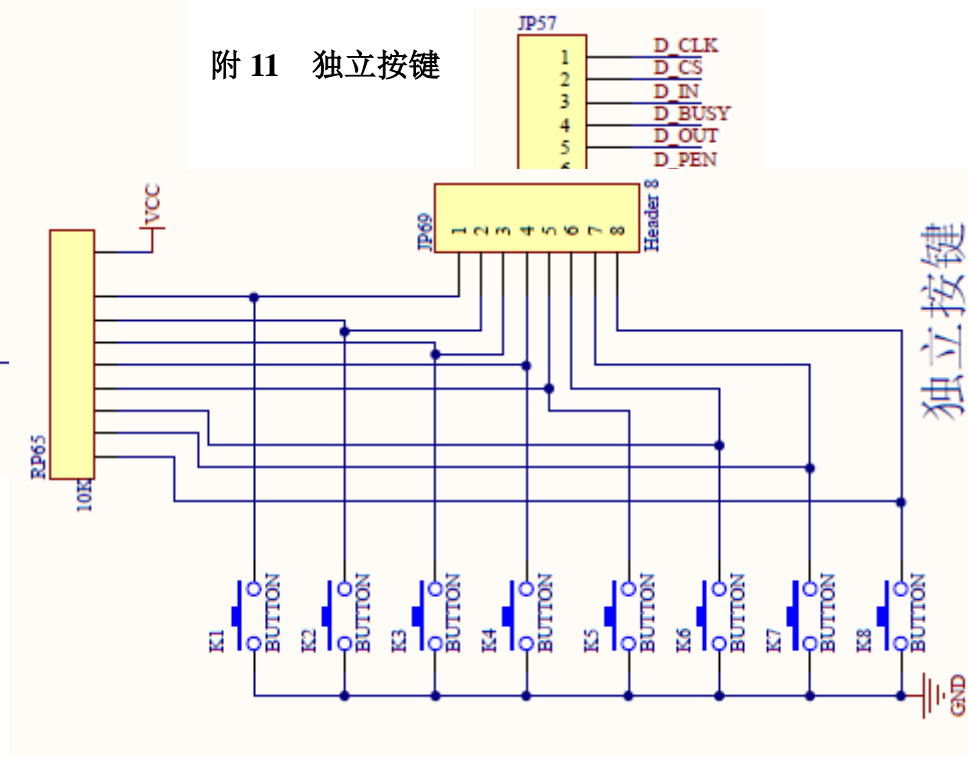


附 10 2.8 寸 TFT 屏

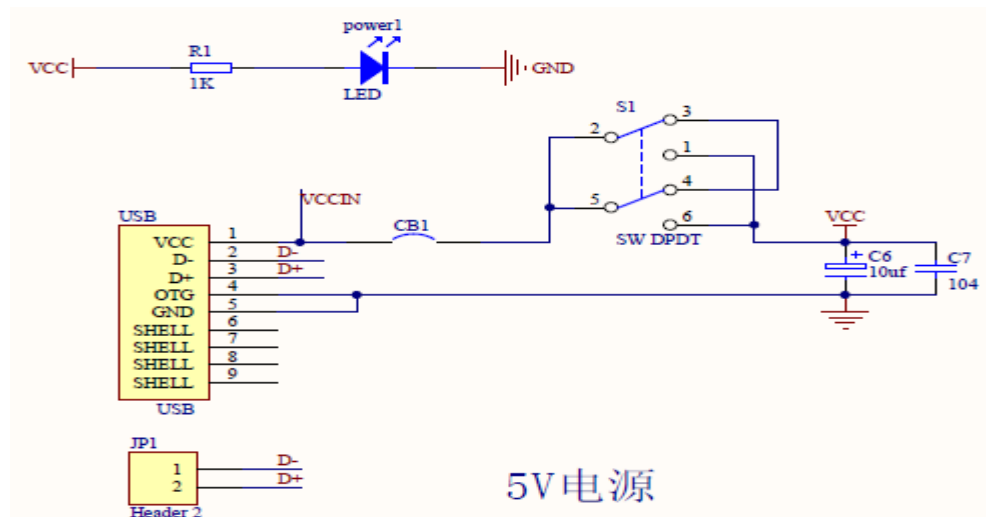




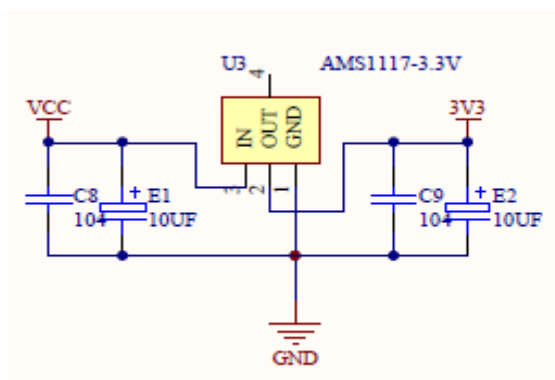
附 11 独立按键



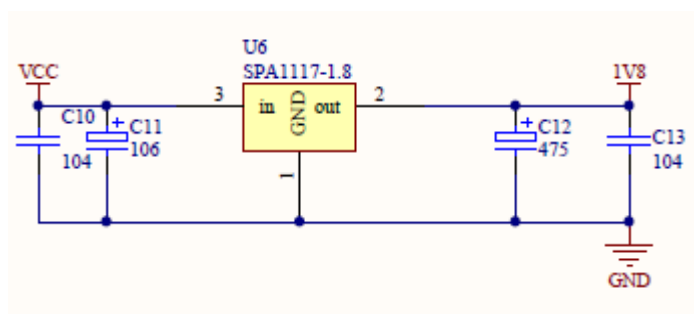
附 12 核心板 USB 供电 5V 电源电路



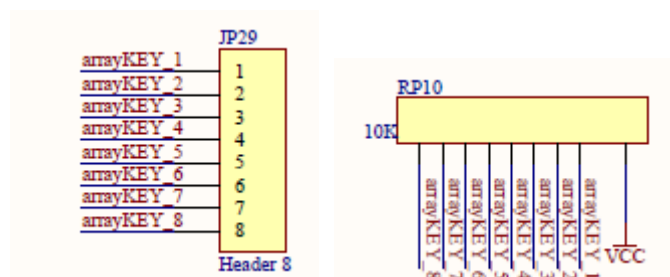
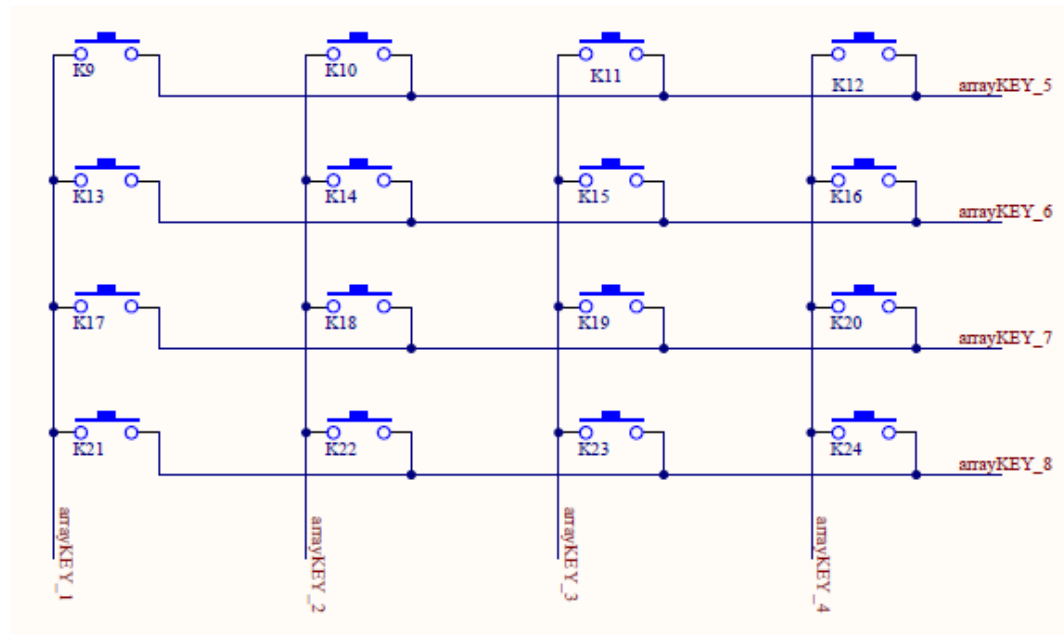
附 13 核心板 3V3 电源电路



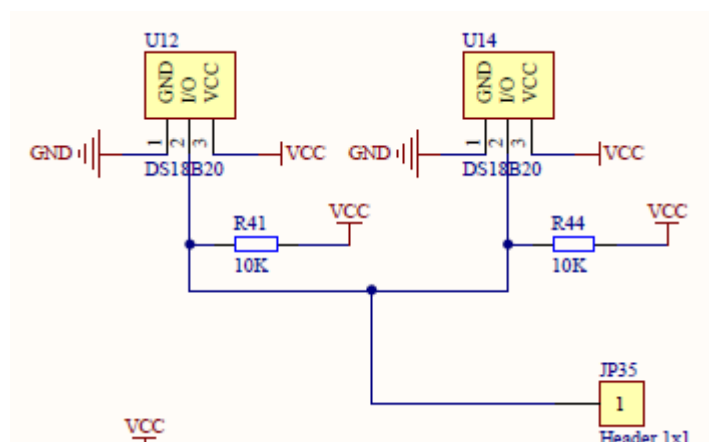
附 14 核心板 1V8 电源电路



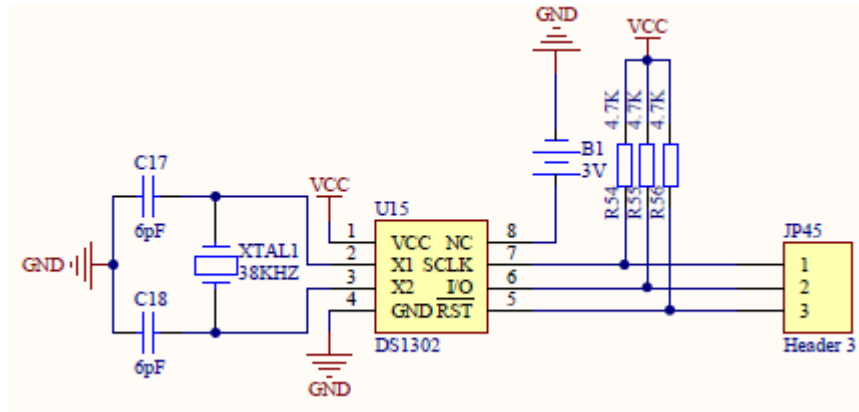
附 15 矩阵键盘



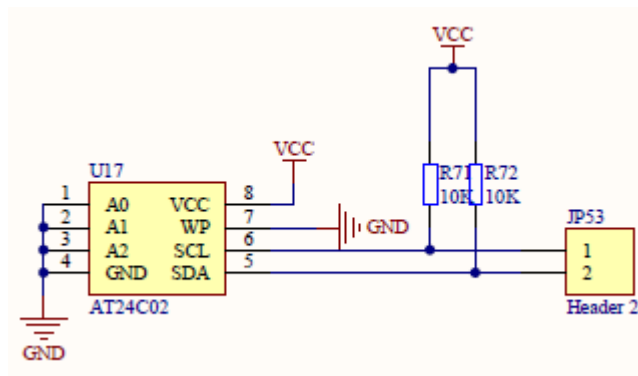
附 16 DS18B20 电路



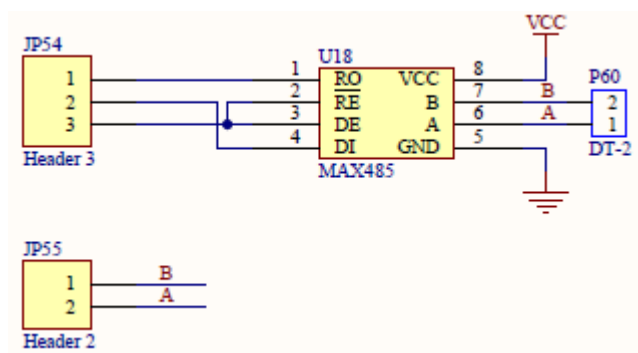
附 17 DS1302 电路



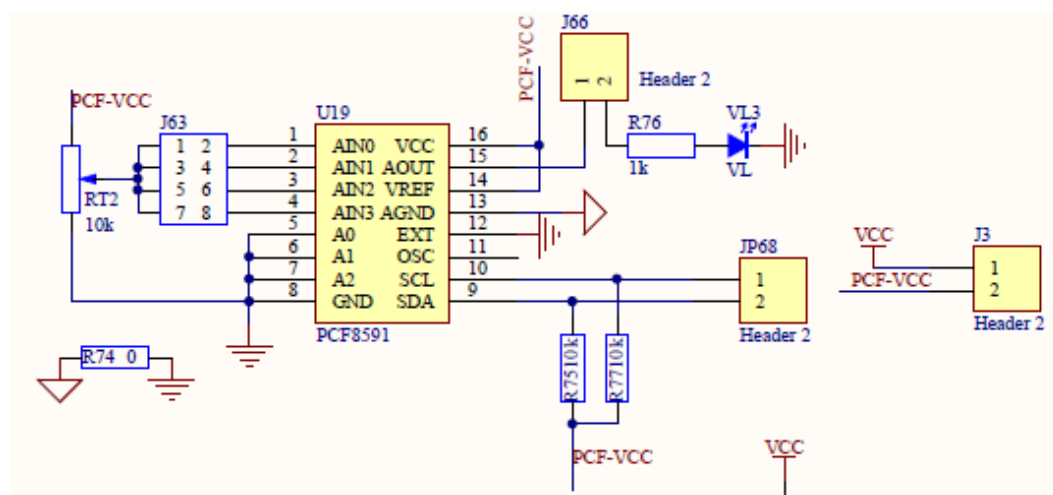
附 18 AT24C02 电路



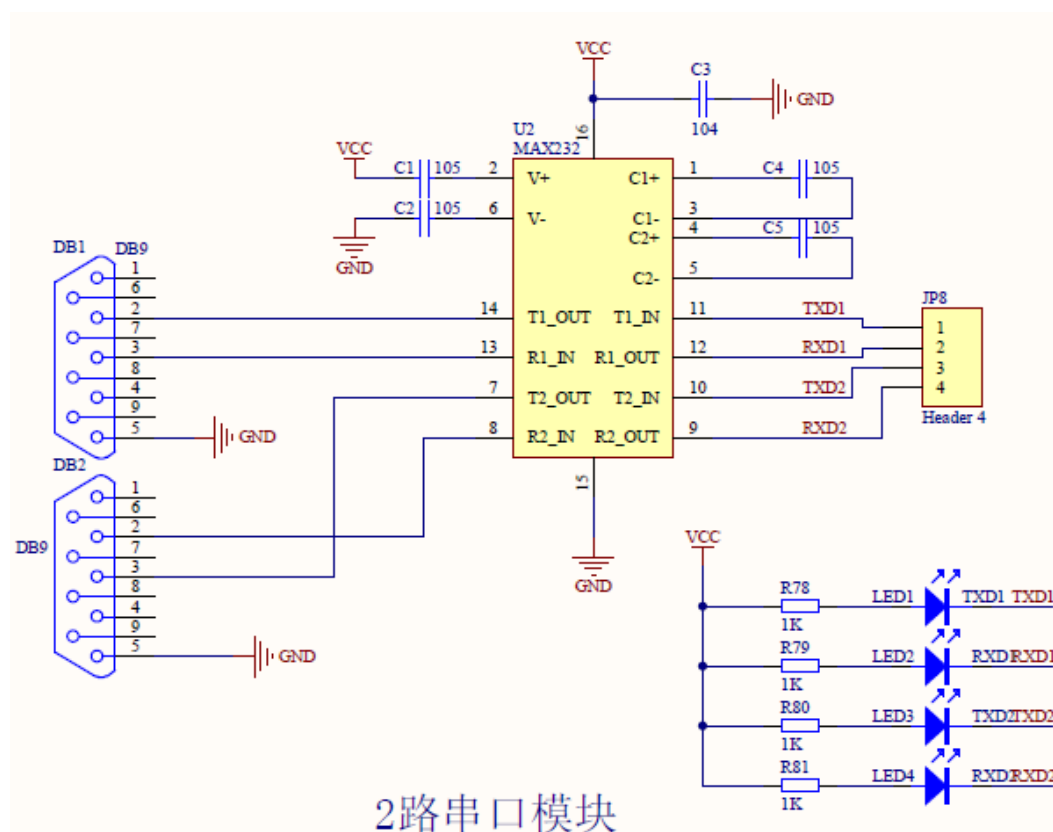
附 19 MAX485 通信电路



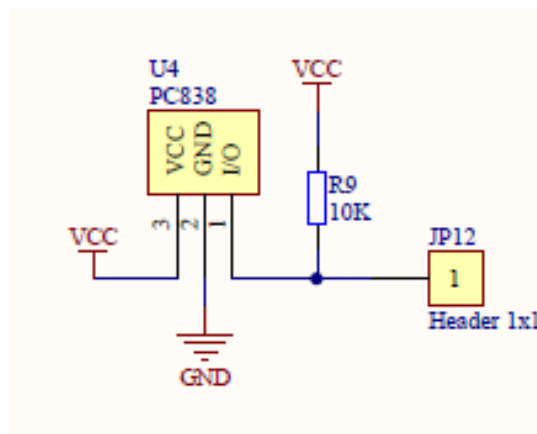
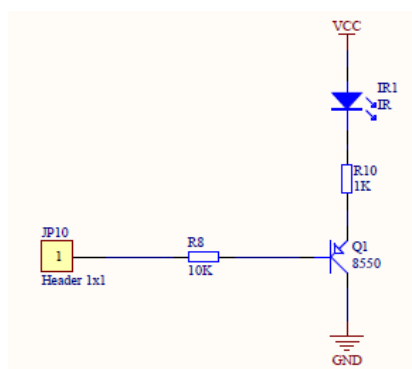
附 20 PCF8591 AD/DA 转换模块



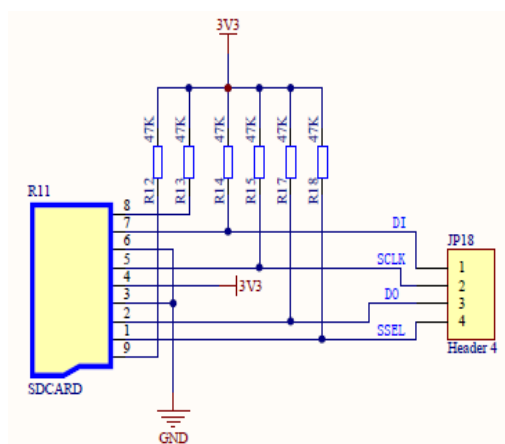
附 21 2 路串口模块



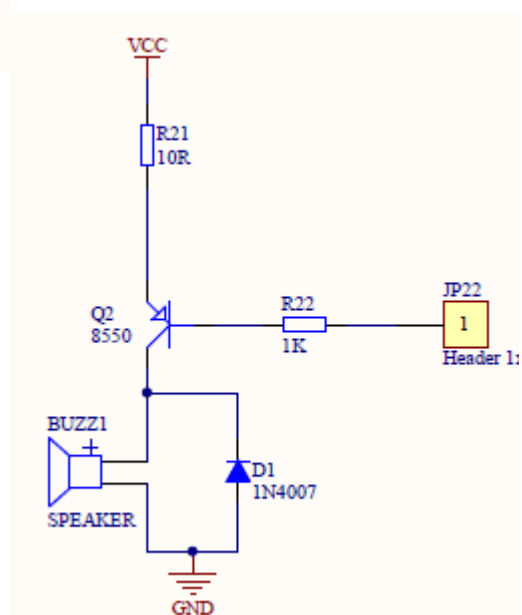
附 22 红外收发头模块



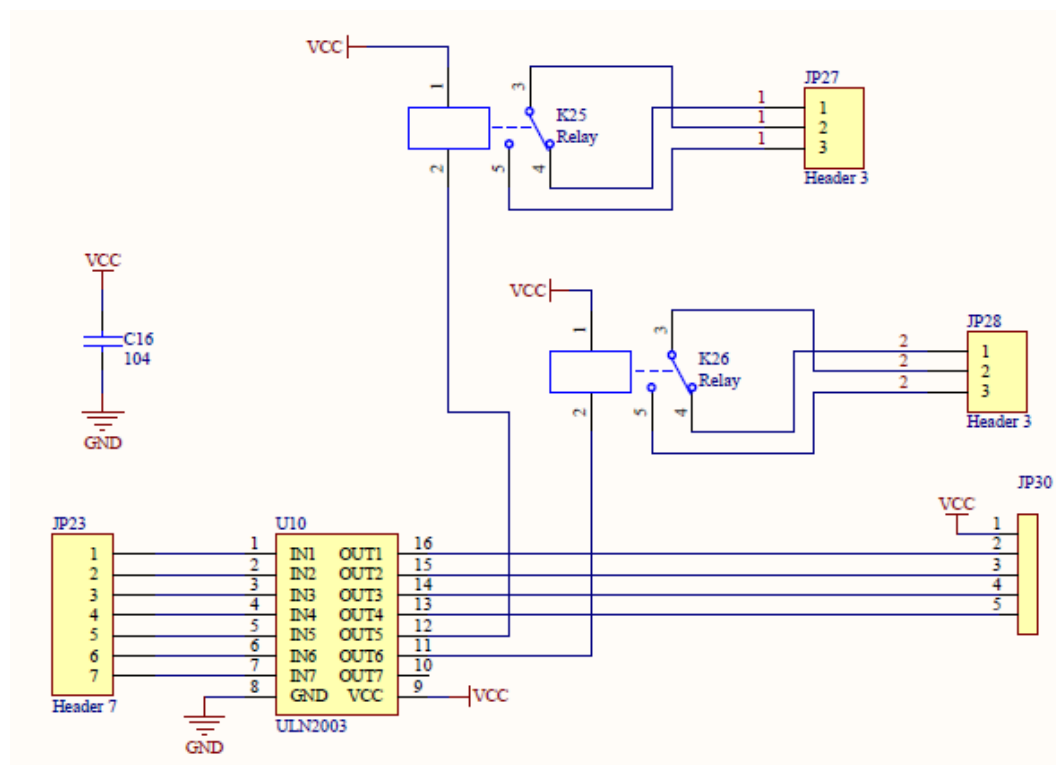
附 23 SD 卡模块



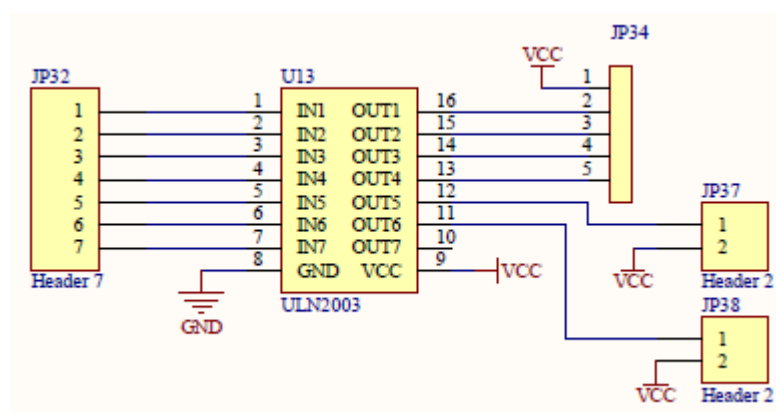
附 24 无源蜂鸣器模块



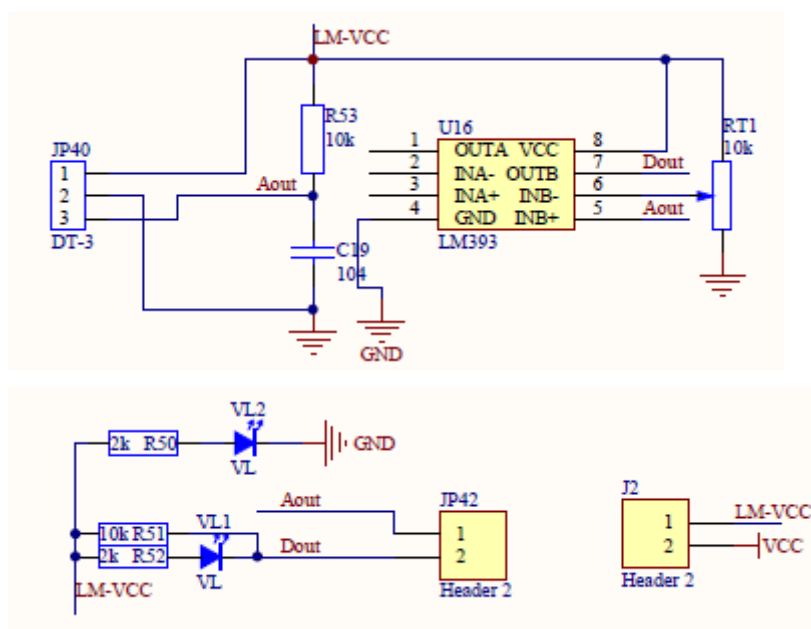
附 25 两路继电器加步进电机模块



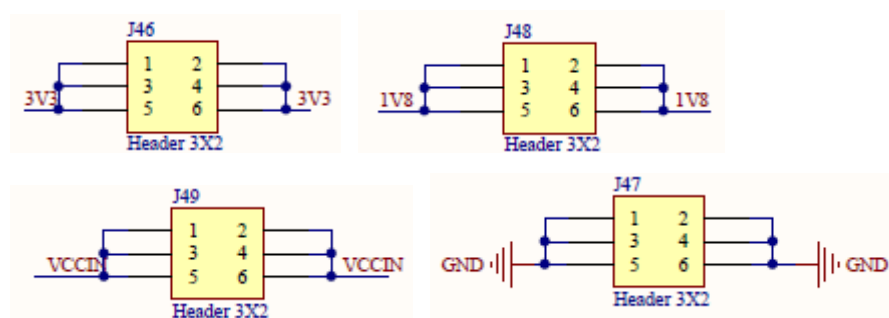
附 26 两路直流电机加步进电机模块



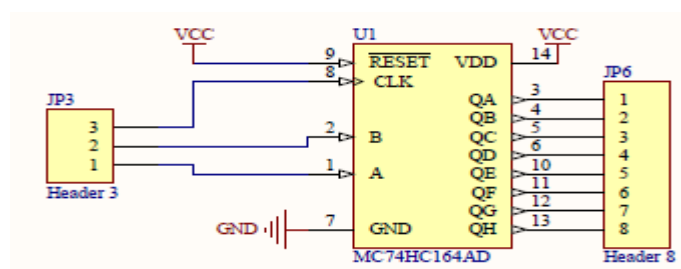
附 26 光敏/火焰/霍尔传感器模块



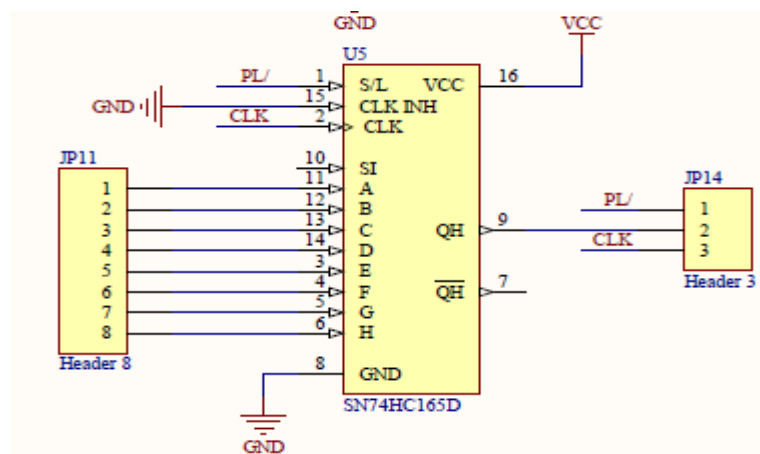
附 27 电源引出口



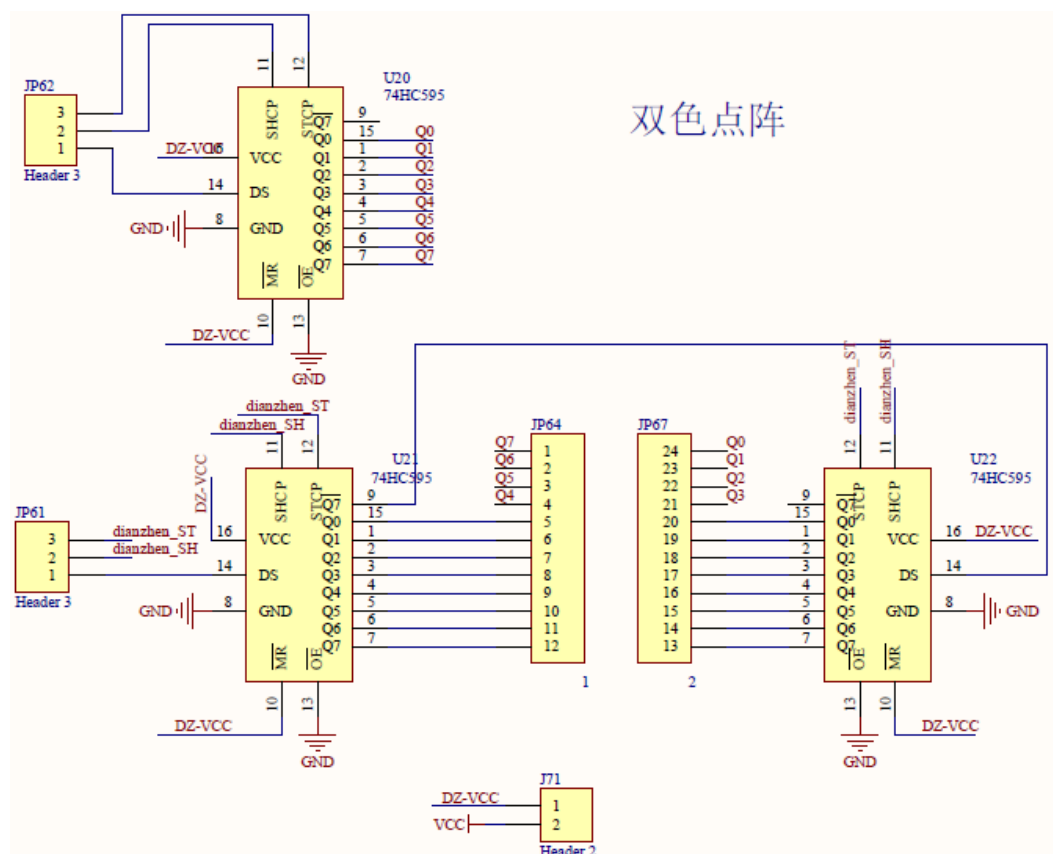
附 28 串转并模块



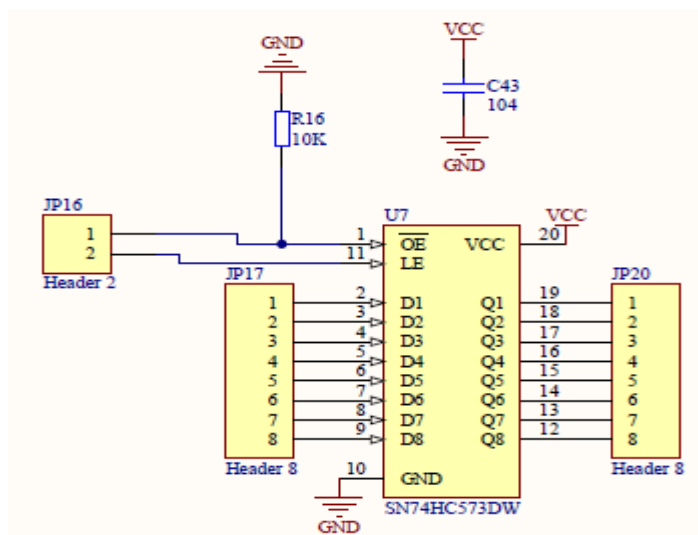
附 29 并转串模块



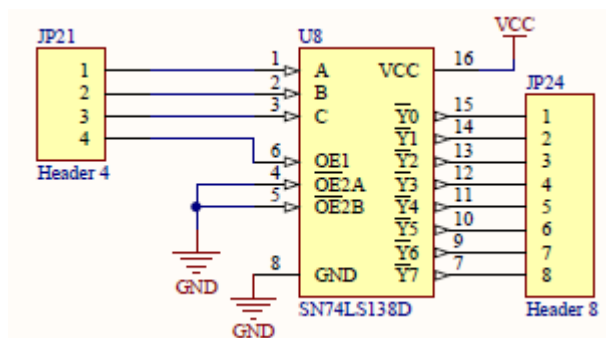
附 30 双色点阵模块



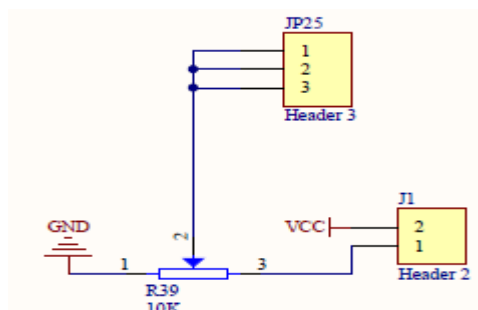
附 31 74HC573 八位锁存模块



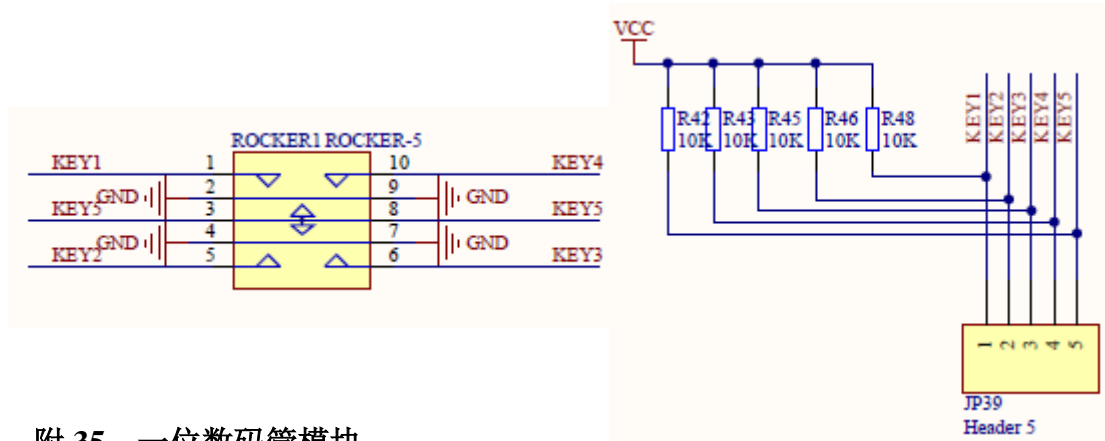
附 32 74LS138 三八译码器模块



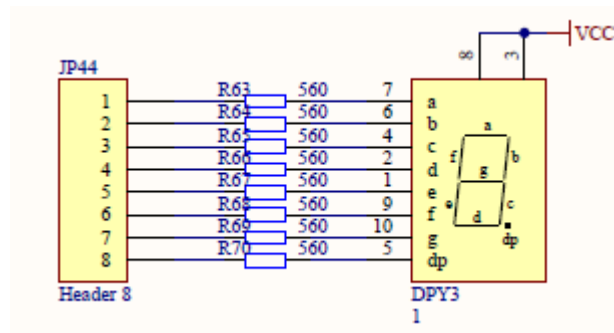
附 33 模拟量输入模块



附 34 五向摇杆模块



附 35 一位数码管模块



附录二 8051 系列单片机汇编语言指令速查表

类别	指令格式	功能简述	字节数	周期
数据 传 送 类 指 令	MOV A, Rn	寄存器送累加器	1	1
	MOV Rn, A	累加器送寄存器	1	1
	MOV A, @Ri	内部 RAM 单元送累加器	1	1
	MOV @Ri, A	累加器送内部 RAM 单元	1	1
	MOV A, #data	立即数送累加器	2	1
	MOV A, direct	直接寻址单元送累加器	2	1
	MOV direct, A	累加器送直接寻址单元	2	1
	MOV Rn, #data	立即数送寄存器	2	1
	MOV direct, #data	立即数送直接寻址单元	3	2
	MOV @Ri, #data	立即数送内部 RAM 单元	2	1
	MOV direct, Rn	寄存器送直接寻址单元	2	2
	MOV Rn, direct	直接寻址单元送寄存器	2	2
	MOV direct, @Ri	内部 RAM 单元送直接寻址单元	2	2
	MOV @Ri, direct	直接寻址单元送内部 RAM 单元	2	2
	MOV direct2, direct1	直接寻址单元送直接寻址单元	3	2
	MOV DPTR, #data16	16 位立即数送数据指针	3	2
	MOVX A, @Ri	外部 RAM 单元送累加器(8 位地址)	1	2
	MOVX @Ri, A	累加器送外部 RAM 单元(8 位地址)	1	2
	MOVX A, @DPTR	外部 RAM 单元送累加器(16 位地址)	1	2
	MOVX @DPTR, A	累加器送外部 RAM 单元(16 位地址)	1	2
算 术 运 算 类 指 令	MOVC A, @A+DPTR	查表数据送累加器(DPTR 为基址)	1	2
	MOVC A, @A+PC	查表数据送累加器(PC 为基址)	1	2
	XCH A, Rn	累加器与寄存器交换	1	1
	XCH A, @Ri	累加器与内部 RAM 单元交换	1	1
	XCHD A, direct	累加器与直接寻址单元交换	2	1
	XCHD A, @Ri	累加器与内部 RAM 单元低 4 位交换	1	1
	SWAP A	累加器高 4 位与低 4 位交换	1	1
	POP direct	栈顶弹出指令直接寻址单元	2	2
	PUSH direct	直接寻址单元压入栈顶	2	2
	ADD A, Rn	累加器加寄存器	1	1
	ADD A, @Ri	累加器加内部 RAM 单元	1	1

	ADD A, direct	累加器加直接寻址单元	2	1
	ADD A, #data	累加器加立即数	2	1
	ADDC A, Rn	累加器加寄存器和进位标志	1	1
	ADDC A, @Ri	累加器加内部 RAM 单元和进位标志	1	1
	ADDC A, #data	累加器加立即数和进位标志	2	1
	ADDC A, direct	累加器加直接寻址单元和进位标志	2	1
	INC A	累加器加 1	1	1
	INC Rn	寄存器加 1	1	1
	INC direct	直接寻址单元加 1	2	1
	INC @Ri	内部 RAM 单元加 1	1	1
	INC DPTR	数据指针加 1	1	2
	DA A	十进制调整	1	1
	SUBB A, Rn	累加器减寄存器和进位标志	1	1
	SUBB A, @Ri	累加器减内部 RAM 单元和进位标志	1	1
	SUBB A, #data	累加器减立即数和进位标志	2	1
	SUBB A, direct	累加器减直接寻址单元和进位标志	2	1
	DEC A	累加器减 1	1	1
	DEC Rn	寄存器减 1	1	1
	DEC @Ri	内部 RAM 单元减 1	1	1
	DEC direct	直接寻址单元减 1	2	1
逻辑运算类指令	MUL AB	累加器乘寄存器 B	1	4
	DIV AB	累加器除以寄存器 B	1	4
	ANL A, Rn	累加器与寄存器	1	1
	ANL A, @Ri	累加器与内部 RAM 单元	1	1
	ANL A, #data	累加器与立即数	2	1
	ANL A, direct	累加器与直接寻址单元	2	1
	ANL direct, A	直接寻址单元与累加器	2	1
	ANL direct, #data	直接寻址单元与立即数	3	1
	ORL A, Rn	累加器或寄存器	1	1
	ORL A, @Ri	累加器或内部 RAM 单元	1	1
	ORL A, #data	累加器或立即数	2	1
	ORL A, direct	累加器或直接寻址单元	2	1
	ORL direct, A	直接寻址单元或累加器	2	1
	ORL direct, #data	直接寻址单元或立即数	3	1

	XRL A, Rn	累加器异或寄存器	1	1
	XRL A, @Ri	累加器异或内部 RAM 单元	1	1
	XRL A, #data	累加器异或立即数	2	1
	XRL A, direct	累加器异或直接寻址单元	2	1
	XRL direct, A	直接寻址单元异或累加器	2	1
	XRL direct, #data	直接寻址单元异或立即数	3	2
	RL A	累加器左循环移位	1	1
	RLC A	累加器连进位标志左循环移位	1	1
	RR A	累加器右循环移位	1	1
	RRC A	累加器连进位标志右循环移位	1	1
	CPL A	累加器取反	1	1
	CLR A	累加器清零	1	1
控制 转移 类 指令	ACCALL addr11	2KB 范围内绝对调用	2	2
	AJMP addr11	2KB 范围内绝对转移	2	2
	LCALL addr16	2KB 范围内长调用	3	2
	LJMP addr16	2KB 范围内长转移	3	2
	SJMP rel	相对短转移	2	2
	JMP @A+DPTR	相对长转移	1	2
	RET	子程序返回	1	2
	RET1	中断返回	1	2
	JZ rel	累加器为零转移	2	2
	JNZ rel	累加器非零转移	2	2
	CJNE A, #data, rel	累加器与立即数不等转移	3	2
	CJNE A, direct, rel	累加器与直接寻址单元不等转移	3	2
	CJNE Rn, #data, rel	寄存器与立即数不等转移	3	2
	CJNE @Ri, #data, rel	RAM 单元与立即数不等转移	3	2
	DJNZ Rn, rel	寄存器减 1 不为零转移	2	2
布 尔 操 作 类 指 令	DJNZ direct, rel	直接寻址单元减 1 不为零转移	3	2
	NOP	空操作	1	1
	MOV C, bit	直接寻址位送 C	2	1
	MOV bit, C	C 送直接寻址位	2	1
	CLR C	C 清零	1	1
	CLR bit	直接寻址位清零	2	1
	CPL C	C 取反	1	1

令	CPL bit	直接寻址位取反	2	1
	SETB C	C 置位	1	1
	SETB bit	直接寻址位置位	2	1
	ANL C, bit	C 逻辑与直接寻址位	2	2
	ANL C, /bit	C 逻辑与直接寻址位的反	2	2
	ORL C, bit	C 逻辑或直接寻址位	2	2
	ORL C, /bit	C 逻辑或直接寻址位的反	2	2
	JC rel	C 为 1 转移	2	2
	JNC rel	C 为零转移	2	2
	JB bit, rel	直接寻址位为 1 转移	3	2
	JNB bit, rel	直接寻址为 0 转移	3	2
	JBC bit, rel	直接寻址位为 1 转移并清该位	3	2

说明：

1) Ri, Rn 指当前工作寄存器，i, n = 0 – 7，当前工作寄存器由程序状态字寄存器 PSW 的 2 个位 RS1, RS0 决定