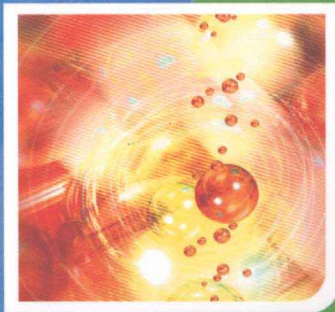
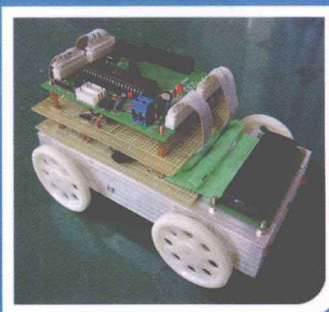


单片机应用 系统设计

洪志刚 杜维玲 井娥林 编著



DANPIANJI YINGYONG XITONG SHEJI



 机械工业出版社
CHINA MACHINE PRESS

单片机应用系统设计

洪志刚 杜维玲 井娥林 编著



机械工业出版社

前 言

传统的单片机教学，均以单片机的结构为主线，先讲单片机的硬件结构，再讲指令系统和软件编程，然后是单片机接口的扩展和各种外围元器件的应用，最后简单介绍几个应用实例。此种结构模式，读者普遍觉得单片机难学。因为现有教材，大部分都是以单片机为蓝本来学习微型计算机原理，而不是为学习单片机技术，在教材、教学过程的安排上，也没有太多地考虑学习者的接受能力，使得学习者的学习过程变成了一个艰难的过程，很多人认为单片机入门难。

基于以上情况，作者尝试编写一套全新的教材，将单片机系统的知识模块化，项目任务化，使得单片机内容的整个学习过程变成完成一个个任务并不断成功的过程。当完成一个任务时，把需要用到的相关知识点加以总结。完成第一个项目时，就能使学习者清楚地了解单片机的开发过程；完成第二、三个项目后，就能使其模仿性地编出自己的程序等。即便只完成部分任务，也可以去做一些程序，使学习者的能力得到提高，激发其学习单片机的兴趣。

本书具体采用的是以项目为导向、以任务为教学单元的学习理念，其具体内容不按单片机硬件结构、指令、编程的先后顺序进行；教学方法也打破原有的界限，将单片机的整体知识分割开来，循序渐进地引入到各项目任务中，即将各部分知识分解成一个个知识点，穿插于各个子任务中，以不断强化单片机技术的学习过程。当所有项目和任务全部完成后，知识点也就学习完了。

本书的主要特点是

1. 以提升学习者的单片机开发能力为目标，遵循以人的认知规律为主线，而不是以课程结构为主线。通过模块一的学习，使读者掌握单片机的开发过程。

2. 以项目开发为导向，全书选用了8个实用项目；把每一个项目分解为若干个任务，构建知识单元，而不是以单片机功能为单元构建。

3. 完成第一个项目即可进行单片机的初步应用尝试，不必学完单片机的全部知识体系。随着任务的逐渐进行，知识逐渐完善，能力逐渐提高，所有任务完成时，学习者便已具有单片机应用初步开发的能力。

4. 每个任务都具有相对的完整性，基础知识以“相关知识”的形式出现，便于查阅。

5. 研制了与教材配套的模块化、积木式单片机应用实训装置。可提供智能小车、数字钟等套件。装置包括单片机最小系统板、键盘输入板、输出显示板、输出控制板、并口扩展板、串口扩展板、模/数转换板、数/模转换板等共20余款；书中把相关的原理图都附在书后供大家分享。可以根据原理图自己制板、焊接、调试来完成，也可以选购电路板来实验；相关的程序清单，力求正确，也提供给读者。

为了便于读者学习，本书内容分成单片机应用系统开发基础、单片机最小系统应用设计、单片机并口扩展应用系统设计、单片机串口扩展应用系统设计、其他总线芯片应用设计、综合应用系统设计等6个模块。书中列举了8个项目，每个项目又分为多个任务，读者通过完成多个任务的学习，最后达到掌握相关项目开发能力的目的。

本书所介绍的项目，可以作为课程实训和设计练习。可提供实验电路板和实训套件，配套指导资料。

本书中原理图均使用 Protel 绘图软件所做，为保持与软件一致，便于读者对照，图中图形符号未按国标符号修改，请读者注意。

本书既可作为普通高等院校和高职高专计算机控制、自动化、机电一体化、电子信息、通信等专业学生的教材，也适合自学和从事单片机工作的工程技术人员参考。

本书由中山职业技术学院洪志刚统稿，并编写了模块一、模块二和模块六，杜维玲编写了模块三、模块四；泰州学院井娥林编写了模块五；中山职业技术学院计算机控制专业全体教师对本书的出版也做了大量工作。

由于作者水平有限，时间仓促，错误和不妥之处在所难免，敬请大家批评指正。

作者

2010 年 11 月

目 录

前言

模块一 单片机应用系统开发基础	1
项目一 基于单片机的单键单灯亮/灭控制系统设计	1
任务 1.1 单片机应用系统的开发流程	1
任务 1.2 认知单片机	6
任务 1.3 基于单片机的单键单灯亮/灭控制电路设计	19
任务 1.4 单键单灯闪亮控制程序设计	28
任务 1.5 基于单片机的单键单灯控制程序的输入和调试	35
任务 1.6 单片机程序的写入和系统调试	41
模块二 单片机最小系统应用设计	44
项目二 基于单片机的键控花样灯设计	44
任务 2.1 基于单片机的 8 键 8 灯亮/灭控制系统设计	44
任务 2.2 基于外部中断控制的单键单灯闪烁系统设计	54
任务 2.3 基于单片机定时中断的流水灯控制系统设计	75
任务 2.4 基于单片机的“多键、多灯、多花样”控制系统设计	91
项目三 直流电动机驱动的智能小车设计	97
任务 2.5 直流电动机驱动的智能小车硬件设计	97
任务 2.6 直流电动机驱动的智能小车软件设计	100
项目四 步进电动机驱动的智能小车设计	107
任务 2.7 步进电动机驱动的智能小车硬件设计	107
任务 2.8 步进电动机驱动的智能小车软件设计	110
项目五 基于单片机最小系统的数字钟控制器设计	114
任务 2.9 基于单片机最小系统的数字钟控制器硬件电路设计	115
任务 2.10 六位 LED 数码管动态扫描显示程序设计	119
任务 2.11 实时时钟程序设计	123
任务 2.12 数字钟时间设置程序设计	128
任务 2.13 定时闹钟时间程序设计	132
任务 2.14 数字钟控制器系统调试	134
模块三 单片机并口扩展应用系统设计	138
项目六 智能温度测控系统设计	138
任务 3.1 总线及简单 I/O 口扩展技术	139
任务 3.2 程序存储器扩展	144
任务 3.3 数据存储器扩展	146
任务 3.4 用 8155 扩展键盘和显示器	152
任务 3.5 温度信号变送与 A/D 转换	158
任务 3.6 数值处理程序设计	167
任务 3.7 用 DAC0832 扩展灯光亮度控制器	176
模块四 单片机串口扩展应用系统设计	184
任务 4.1 串口-并口转换	184

任务 4.2 单片机与 PC 之间的 RS-232C 串行通信	193
任务 4.3 单片机与单片机之间的 RS-485 串行通信	197
任务 4.4 单片机与多机之间的通信——主从式	200
模块五 单片机 I²C、SPI 和 Flash 芯片应用设计	205
任务 5.1 I ² C 总线存储器 AT24C02 扩展设计	205
任务 5.2 单线温度传感器 DS18B20 应用设计	214
任务 5.3 实时时钟芯片 DS1302 的应用	228
任务 5.4 IC 存储卡 (IC 总线) 的读写	234
模块六 单片机综合应用系统设计	240
项目七 预付费式用电控制箱语音报警器设计	245
任务 6.1 预付费式用电控制箱语音报警器硬件电路设计	246
任务 6.2 预付费式用电控制箱语音报警器程序设计	251
项目八 基于单片机的车载自动饮水机控制系统设计	256
任务 6.3 车载自动饮水机硬件电路设计	258
任务 6.4 车载自动饮水机程序设计	261
附录	264
附录 1 MCS-51 单片机指令表	264
附录 2 ASCII 码对照表	268
附图 1 单片机实训装置原理图和 PCB 图	270
附图 1-1 单片机最小系统板原理图和 PCB 图	270
附图 1-2 8 位独立键盘原理图和 PCB 图	271
附图 1-3 8 位 LED 指示灯及其驱动电路原理图和 PCB 图	272
附图 1-4 输出控制与电源原理图和 PCB 图	273
附图 1-5 6 位数码管显示器原理图和 PCB 图	274
附图 1-6 16 位拨码开关和钮子开关原理图和 PCB 图	275
附图 1-7 串口 RS-232 原理图	276
附图 1-8 实时时钟芯片 DS1302 与单片机的连接电路原理图和 PCB 图	276
附图 1-9 6 位 LED 数码管及晶体管位驱动原理图和 PCB 图	277
附图 1-10 RS-485 与单片机接口电路原理图和 PCB 图	277
附图 1-11 直流电动机驱动电路原理图	278
附图 1-12 混合式两相步进电动机驱动电路原理图	278
附图 1-13 并行接口扩展板原理图和 PCB 图	279
附图 1-14 扩展 8 位输入、输出电路原理图	280
附图 1-15 行列式键盘电路原理图 (2×8 键)	281
附图 1-16 ADC0809 与单片机连接电路原理图和 PCB 图	281
附图 1-17 8155 扩展键盘和显示驱动电路原理图和 PCB 图	282
附图 1-18 DAC0832 数/模转换器与单片机连接电路原理图和 PCB 图	283
附图 2 基于单片机最小系统构建的数字钟控制器电路原理图	285
附图 3 直流电动机驱动的智能小车部分电路原理图	286
附图 4 步进电动机驱动的智能小车驱动电路原理图	287
附图 5 基于单片机的花样灯控制电路板	288
附图 6 数字钟控制器模块式电路板	288
附图 7 智能温控系统模块式电路板	289
参考文献	290

模块一 单片机应用系统开发基础

本模块学习目标:

- ★ 先从“基于单片机的单键单灯亮/灭控制系统”项目的开发着手，初步认识单片机；
- ★ 了解单片机实训装置；
- ★ 掌握单片机应用系统设计所需开发软件的使用和开发流程；
- ★ 深入了解单片机内部资源。

项目一 基于单片机的单键单灯亮/灭控制系统设计

一、学习要求

- ★ 掌握单片机应用系统的开发流程；认知单片机；
- ★ 了解学习单片机应用系统设计开发的实验装置；
- ★ 了解单片机程序设计指令格式、程序结构；
- ★ 掌握单片机仿真调试软件的使用及程序编辑和调试方法；
- ★ 掌握单片机程序的写入和系统调试方法。

二、总体设计

本项目要求利用单片机实现单键控制单灯的亮/灭功能，共分成 6 个任务来学习这部分内容。

在学习本项目时，可以考虑让学生在学习电子技能训练时，练习焊接电路板，把单片机实训套件电路板焊好。这样，在学习单片机时，就可以直接使用；也可以考虑要求学生在课外时间完成电路板焊接。

任务 1.1 单片机应用系统的开发流程

能力目标：会单片机开发工具的使用，会把编写的程序输入到计算机，并进行编辑和下载。

知识目标：掌握单片机应用项目的开发流程。

1. 任务描述

选用单片机为控制核心，用一个按钮控制一只 LED 灯的点亮和熄灭。要求上电后，按一次按钮（按钮与单片机的第 12 脚即 P3.2 连接），灯（通过单片机的第 1 脚即 P1.0 连接）点亮；再按一次，灯熄灭；第三次按又点亮，如此循环。

2. 任务分析

根据系统要求，用一个按钮输入，一个指示灯作输出，采用单片机最小系统（包括复位电路、晶振电路）作为控制核心，电源采用开关电源，输出 +5V 电压。当然，在这个任务中，主要是想告诉大家单片机项目的开发流程，所以，不刻意追求此项目的性能价格比。

一个完整的单片机应用系统由硬件和软件两部分共同构成。

3. 硬件电路的设计及接线

1) 基于单片机的单键单灯亮/灭控制系统组成结构如图 1-1 所示。系统由单片机芯片、键盘、指示灯、复位电路、晶振电路及直流电源组成。采用 Protel 99SE/DXP 电子绘图软件绘制电路原理图和 PCB 图。

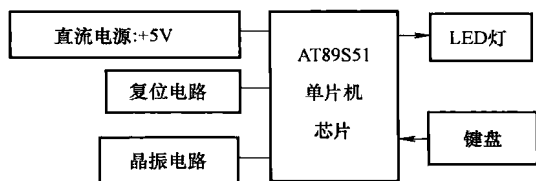


图 1-1 单键单灯亮度控制系统组成结构框图

2) 原理电路如图 1-2 所示。电源由外部 +5V 供电。按键采用外接一个微动按钮 AN3 与单片机的第 12 脚相连；输出指示灯采用发光二极管 3LED1，与发光二极管相连的电阻 1R1 起限流作用。调节 1R1 的阻值可以改变输出灯的亮度。单片机选用 AT89S51，采用单片机内部程序存储器工作，单片机的第 31 脚要接高电平；单片机工作还需要可靠的复位电路和外部晶振电路。

为了方便大家实现此功能，我们采用实验电路板来演练。

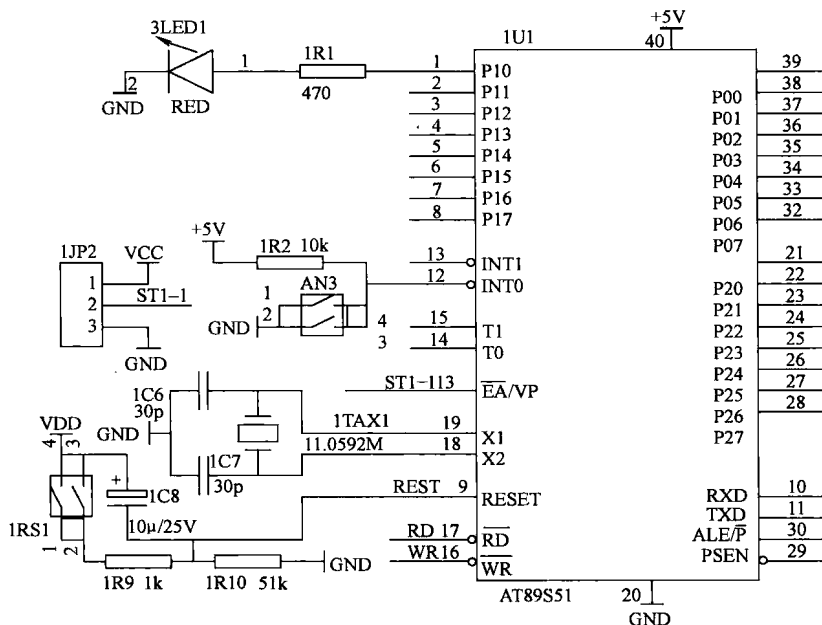


图 1-2 基于单片机的单键单灯亮/灭控制系统原理图

3) 在实验装置上的接线原理图如图 1-3 所示。采用的实验板包括：

单片机最小系统板 (1 号板)，如附图 1-1-2 所示；8 位独立键盘板 (2 号板)，如附图 1-2-2 所示；8 个 LED 灯显示板 (3 号板)，如附图 1-3-2 所示；开关电源为 +5V。在此，假设电路板都已焊好。

选择 8 个按键中的 AN3，该键信号从单片机芯片的第 12 脚 (P3.2) 输入；AN3 按下时，P3.2 为低电平，松开后 P3.2 为高电平；发光二极管选择 3LED1，该灯从单片机的第 1 脚 (P1.0) 输出，当 P1.0 输出高电平 (即 P1.0 = 1) 时，灯亮；若 P1.0 = 0 则灯灭。

跳线：用短路块把 1 号板上的 1JP2 的 2-1 短接，使单片机的 31 脚接高电平，使单片机工作时运行片内程序；3 号板上的跳线 3JP1 用短路块使 2-3 连接，使发光二极管的负端

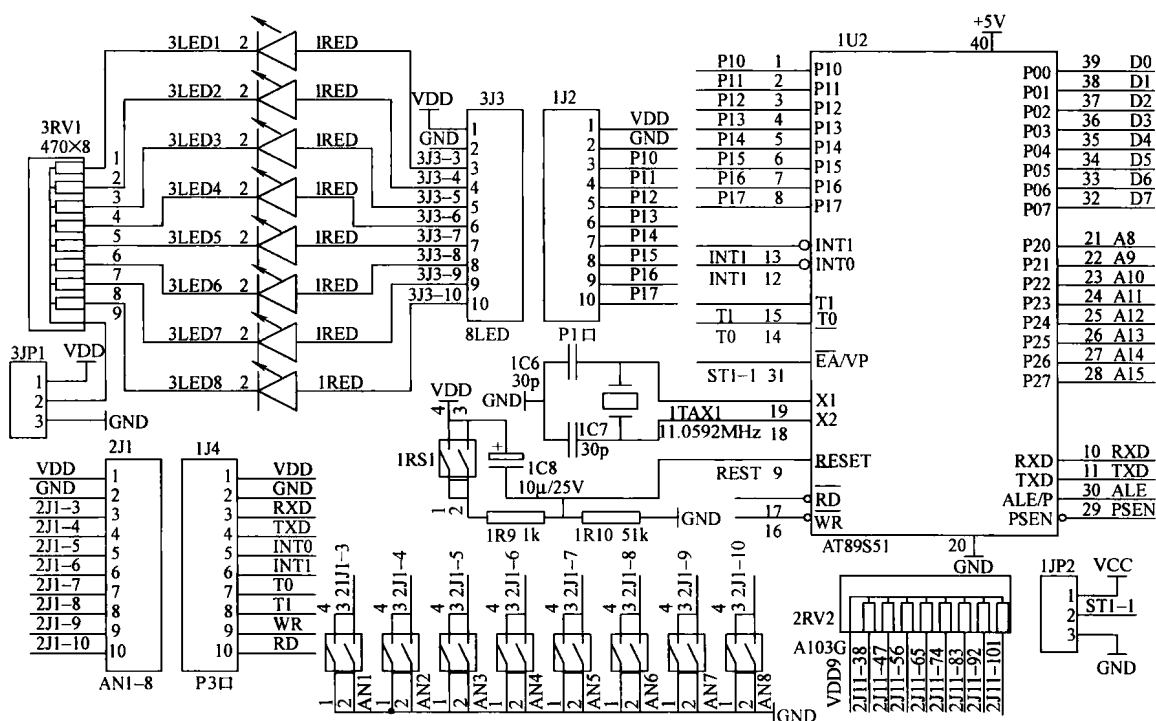


图 1-3 键控灯接线实验原理图

接地。

连线：用 2 条单列 10 芯的排线分别连接 1J4—2J1（按钮信号从单片机的 P3 口输入）和 1J2—3J3（指示灯从 P1 口输出）。电路图上信号的连接通过网络标号来指示，这一点学过 Protel 电子绘图软件后都应该能看懂，所以，在原理图上没有用导线连，而是用网络标号来标注。

注意电路板上，方形焊盘的引脚为第 1 脚。电源可以直接从开关电源提供 +5V 到 1CZ1 输入。

4. 程序设计

单片机应用系统除了硬件电路外，还需要有程序才能工作。

买回来的单片机芯片内部的程序不一定适应你的具体的工作对象。所以，单片机应用系统的设计开发需要编写应用程序。具体过程如下：

（1）分析系统工作要求，明确系统详细工作过程

此实例采用按钮控制灯的开关，工作过程是：上电后，系统复位，执行初始化，然后扫描键盘看是否有按钮按下，如果有按钮按下（即 P3.2 = 0），则标志位“置 1”（以 20H 单元的 D0 位为例，就使 20H.0 = 1），相当于做一个标记；如果该按钮没有按下（即 P3.2 = 1）则标志位不变。接着单片机检测标志位，当标志位 20H.0 = 1 时，表示按了奇数次按钮，要求亮灯（P1.0 = 1）；如果标志位不等于 1，表示按了偶数次按钮，要求灭灯（P1.0 = 0）。

（2）画流程框图

基于单片机的单键单灯亮/灭控制要求，画出流程框图，如图 1-4 所示。

系统上电之后, 先进行初始化, 主要对所用端口进行设置: P1 口为输出, P3 口为输入, 堆栈指针为 60H; 按下按键标志位寄存器清零; 然后, 检测是否有键按下, 如果有键按下, 则把标志位取反, 如果没有键按下, 标志位不变; 接着判别按键次数, 如果按了奇数次按键, 则亮灯, 如果按了偶数次按键, 则灭灯; 接下来, 单片机循环回来继续检测按键是否按下, 判断按键次数, 控制灯亮灯灭。

(3) 单片机资源分配

硬件资源分配: 即外部引脚分配, 按钮对应 P3.2 引脚; 指示灯从 P1.0 输出。

软件资源分配: 即内部 RAM 分配, 用 20H 单元作为标志存储单元, 20H.0 作为按键按下次数的标志位, 按下奇数次, 20H.0 = 1; 按下偶数次, 20H.0 = 0; 60H 之后为堆栈区。

(4) 编写程序清单

```

ORG 0000H;
SJMP MAIN;           跳到主程序
ORG 0030H;           主程序从 0030H 开始
MAIN: MOV P1, #00H;   设 P1 口为输出
      MOV P3, #0FFH;   设 P3 口为输入
      MOV 20H, #0;     清标志位 20H 单元
LOOP:  JB P3.2, CHABZ;  判断 AN3 是否按下
      CPL 20H.0;       如果按下了, 取反标志
CHABZ: JNB 20H.0, MIED; 判断键被按次数
      SETB P1.0;       按下奇数次亮灯
      SJMP LOOP
MIED:  CLR P1.0;       按下偶数次灭灯
      SJMP LOOP;       大循环, 再次检测按键
END;                伪指令, 表示程序结束

```

5. 程序输入与编译

单片机程序需要设计者自行编写, 输入、编译后写入到单片机中, 可在万利、伟福等任意一种编译软件下进行, 在计算机上输入, 再经过编译, 把汇编语言格式的程序指令翻译成机器代码。在此仅给出程序清单环节, 程序输入和编译细节后面再介绍。

打开万利仿真器程序 MedWin V3.0, 在文件主菜单的“项目管理”下, 单击“新建项目”, 建立一个文档, 另存为“danjiandandengkz.asm”, 扩展名取.asm, 表示汇编语言程序, 如图 1-5 所示。

6. 程序调试

单片机应用系统的调试分为硬件调试、程序调试、软硬件联调、独立运行调试和现场调

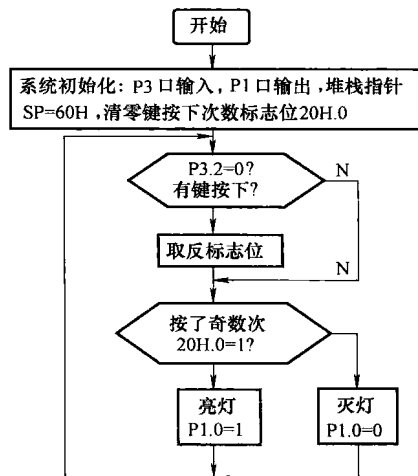


图 1-4 单键单灯亮/灭控制流程框图

试等。硬件调试包括通电前是否短路的测试、通电后各相应点电位的测试及工作状态是否正确等测试；程序调试包括在程序输入后，编译过程中，仿真软件可以找出语法错误等；然后，进行仿真调试。借助仿真器也可以对用户板的硬件进行调试，查找设计或制作中的错误。

如图 1-6 中，编译后出现错误提示，第 13 行；细查是“分号;”采用了中文格式，需改成英文模式的“;”；修改后再一次编译，错误消失，然后保存。

软硬件联调是指在仿真器和用户板相连的情况下，调试系统的软件和硬件电路功能协调一致工作。

7. 程序写入芯片

程序调试无误后就可以把程序写入芯片，使单片机脱离 PC 独立运行。编译调试成功后的程序最终要下载到单片机芯片中去才能使单片机执行；这一过程称为程序的写入或程序下载。程序的下载方法有两种：离线下载和在线下载。离线下载是指把单片机芯片放到专用编程器（也叫写片机）上，再把调试好的单片机程序目标文件写入到单片机芯片中，然后，再把单片机从专用编程器上取下，安装到用户板的 IC 座上使用；在线编程是指不需要把芯片从用户板上拿下来，而是通过串行接口等下载接口，直接把计算机内的程序代码下载到用户板上的单片机芯片中，这样可以先把芯片在电路板上焊好再编程。

程序离线写入芯片需要用到编程器，例如 RF-853。

独立运行调试是把写好程序的单片机插到单片机用户板上，脱开仿真器工作，看系统工作是否正常。

如果单片机应用系统在仿真器下工作正常，独立工作时不能正常工作，需要重点检查的主要有电源系统、复位电路、晶振电路以及单片机的程序存储器选择端电位（单片机的 31 脚为高电平，表示系统复位后程序从单片机内部开始运行；如果 31 脚为低电平，表示程序从单片机外部程序存储器开始运行）。

8. 系统调试

编好的程序可能还有很多不如意或不完善的地方，需要经过不断修改和反复实验。即便是用仿真器调试好的程序，当写好程序的单片机芯片放回到用户板上独立工作时，也有可能系统

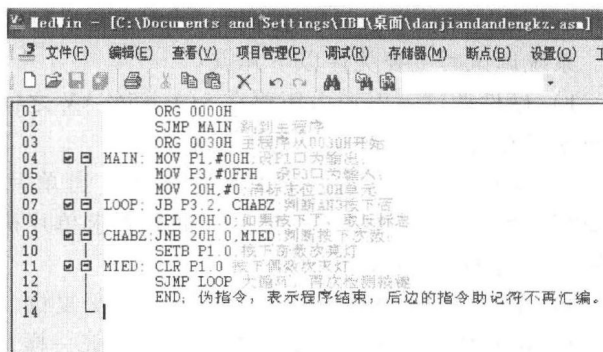


图 1-5 程序输入界面

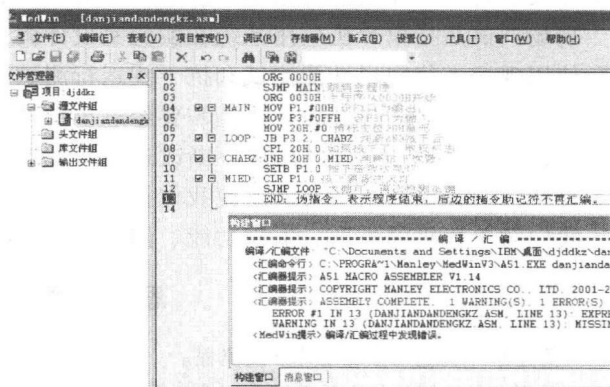


图 1-6 编译后出错提示

工作不正常,或在线下载程序之后,需对系统进行软、硬件调试,以及最后的现场使用调试。

系统调试时主要关注以下几点:

1) 硬件调试:电源电压要在 $5(1 \pm 5\%)V$ 之间,即在 $4.75 \sim 5.25V$ 之间;电源电压太高或太低都有可能使系统工作不正常。

2) 单片机运行程序所在地的选择:当单片机要求从 CPU 内部开始运行程序时,AT89S51 单片机的 31 脚接高电平;如果要求单片机运行程序从单片机之外的程序存储器开始执行,则 31 脚接低电平。

3) 复位电路:AT89S51 单片机是高电平复位,低电平时运行。

4) 晶振电路:晶振电路就如同人的心脏一样,如果没有这个时钟,单片机也就不工作了。检测此引脚可以用示波器观测单片机的 30 脚 (ALE) 信号的波形。

现在来操作看看我们的设计是否与我们所要求的一致。

仿真调试完成之后,把经编程器写好程序的单片机芯片 (AT89S51/52) 插到单片机座上,也可以通过下载线,在线把调试好的程序写入到单片机芯片 (AT89S51/52) 中。通电后,按下按钮 AN3 一次,指示灯 LED1 点亮;再按一次 AN3,指示灯熄灭;按第三次 AN3,指示灯又点亮;这样,按奇数次 AN3,指示灯点亮,按第偶数次 AN3 时,指示灯熄灭。

现场调试是指单片机应用系统在实验室设计调试通过满足功能和性能指标后,还需要到实际使用现场进行现场性能测试和抗干扰运行测试。

一个产品在研制成功之后,还需要经过小批量试产、中批量试产之后才能进入大批量生产阶段,走向市场,并在各阶段不断改进完善。

总结:通过这样一个实例,使我们初步认识了单片机应用项目的组成和开发过程,接下来,我们对单片机的外形、引脚功能、内部结构、特点及发展作进一步详细的学习。

思考与练习题

1. 简述单片机应用系统设计开发流程。

2. 采用电子绘图软件设计单键单灯控制系统电路原理图和 PCB 图。注意,AT89S51 的原理图库与 8031 的相同。

3. 根据任务 1.1 介绍的方法编写用按键 AN2 控制灯 LED2 亮/灭的流程框图和程序,并实现。

任务 1.2 认知单片机

1. 单片机的定义

通过键控单灯的实验,可以看到:通过按一下按钮 AN3,指示灯点亮,再按一次灯熄灭;按奇数次时灯亮,按偶数次时灯灭。它是怎样实现的呢?这中间主要利用的是单片机的“记忆”功能。为了学会使用单片机,让我们先来认识单片机。

单片机的外形如图 1-7 所示。

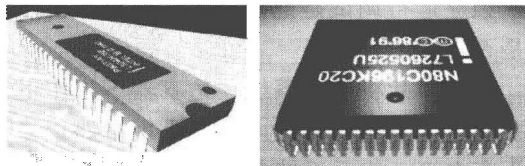


图 1-7 单片机外形图

一台能够工作的计算机由这样几个部分

构成:CPU (进行运算、控制)、ROM (程序存储)、RAM (数据存储)、输入/输出接口电路 (例如串口、并行输出口等)、定时/计数器等。在个人计算机上这些部分被分成若干块芯片,安装在一个称之为主板 的印制电路板上。而在单片机中,这些部分被全部做到一块集

成电路芯片中了，所以就称为单片（单芯片）机（英文：Singlechip）。所谓“单片机”是指把 CPU、存储器、输入输出接口芯片（PIO）及定时计数器（CTC）都集成到一个芯片中的微控制器。而且有一些单片机中除了上述部分外，还集成了其他部分，如 A/D、D/A 转换器等。

这么多东西做在一起，那单片机价格会不会太高呢？这块芯片应该非常大吧？不，实际上单片机的价格并不高，从几元人民币到几十元人民币，体积也不大，一般采用 40 脚封装，当然功能多一些单片机也有引脚比较多的，如 68 引脚，功能少的只有 10 多个或 20 多个引脚，有的甚至只有 8 引脚。

2. 单片机的发展概况

单片机的发展大致可分为 4 个阶段：

1) 单片机探索阶段：以 Intel 公司 MCS-48 和 Motorola 公司的 6801 为代表，属低档型 8 位单片机。

2) 单片机完善阶段：以 Intel 公司的 MCS-51 和 Motorola 公司 68HC05 为代表，属于高档型 8 位机。随着 8 位单片机体系进一步完善，特别是 MCS-51 系列单片机在全球得到了广泛应用，奠定了它在单片机领域的地位，形成了事实上的 8 位单片机标准结构。

3) 由单片机向微控制器发展的阶段：此阶段 Intel 公司推出了 16 位的 MCS-96 系列单片机，其他芯片制造商也纷纷推出了性能优异的 16 位单片机，但由于 16 位机价格太高，其应用面受到一定限制。反观 MCS-51 系列单片机，由于其性价比高，得到了广泛应用，并吸引了世界许多知名芯片制造厂商竞相以 80C51 为内核，扩展部分测控系统中使用的电路技术、接口技术、A/D 转换、D/A 转换和看门狗等功能部件，推出了许多与 80C51 兼容的 8 位单片机。强化了微控制器的特征，进一步巩固和发展了 8 位单片机的主流地位。

4) 微控制器全面发展阶段：随着单片机在各个领域全面深入的发展和应用，世界各大电气、半导体厂商普遍投入，出现了高速、大寻址范围、强运算能力的 8 位/16 位/32 位通用型单片机以及小型廉价的专用型单片机，促使单片机进入一个可广泛选择和全面发展的应用时代。

3. 单片机的分类

单片机作为工业控制和数据处理的计算机，它的结构与指令功能都是按照工业控制要求设计的，对其有多种分类方法：按数据线位数来分，单片机主要有 4 位、8 位、16 位、32 位等；从应用的角度来看，单片机大致可分为以下几类：

1) 通用型/专用型。所谓通用/专用，是指其应用范围。如 80C51，属于通用型单片机，它不是为某种专门用途设计的。还有一些单片机是针对某一类产品，甚至是针对某一个产品而设计生产的，如 VCD、DVD 以及 PC 声卡、显卡中的 CPU 芯片。专用型单片机可最大限度地简化系统结构，提高资源利用率，降低成本。目前，开发专用型芯片是单片机发展的一个重要分支。

2) 总线型/非总线型。总线型单片机普遍设置有并行地址总线、数据总线和控制总线，这些引脚可以用来并行扩展外围元器件。非总线型单片机从使用的角度可分为两类：一类是有并行总线但不并行扩展，原用于并行扩展的地址总线、数据总线引脚直接用于 I/O 口，即使需要扩展也是通过串口扩展；另一类是将需要的外围元器件及外设接口直接集成在单片机内，省去原用于并行扩展的地址总线、数据总线和无用的控制端线，这样减少了芯片引脚数

和芯片体积。由于串行扩展技术的发展,以及在 Flash ROM 的应用,非总线型单片机逐渐成为单片机发展的主流方向。

3) CISC 结构体系和 RISC 结构体系。MCS-51 系列单片机采用 CISC 结构体系,其指令复杂,指令代码、周期数不统一,因此指令运行很难实现流水线操作,大大阻碍了运行速度的提高。采用 RISC 结构体系后,精简指令,绝大部分成为单周期指令,而且通过增加程序存储器的宽度,实现一个存储地址单元存放一条指令,从而实现流水线操作,在晶振频率相同的条件下,大大提高了指令运行速度,如 PIC 系列单片机。

4) OTPROM 型/EPROM 型/Flash ROM 型。单片机片内 ROM 主要有以下几种形式:Mask ROM、OTPROM、EPROM 和 Flash ROM。Mask ROM 和 OTPROM 仅适用于大批量生产的成熟产品。在开发研制阶段或小批量试产阶段的产品,一般用 EPROM 和 Flash ROM,早期的单片机系统多用 EPROM 或 E²PROM,但由于其价贵和使用不便,EPROM 和 E²PROM 已逐渐退出。Flash ROM 因其可多次编程擦写,价廉且使用方便,目前已成为应用的主流产品。

与其他集成电路芯片一样,单片机也可按所能适用的环境温度划分为三个等级,即民用级(0~70℃);工业级(-40~85℃);军用级(-65~125℃)。

下面说说 MCS-51 系列单片机和 8051、8031、AT89S51 等的关系。

我们平常老是讲 8051,还有什么 8031,现在又有 AT89S51、AT89C51 等,它们之间究竟是什么关系呢?

MCS-51 是指由美国 Intel 公司生产的一系列单片机的总称,这一系列单片机包括了好多品种,如 8031、80C51、8751、8032、8052、8752 等,其中 80C51 是最早最典型的产品,该系列其他单片机都是在 80C51 的基础上进行功能的增、减改变而来的,所以人们习惯于用 80C51 来称呼 MCS-51 系列单片机,而 8031 是前些年在我国最流行的单片机,所以很多场合会看到 8031 的名称。Intel 公司将 MCS-51 的核心技术授权给了其他很多公司,所以有很多公司在做以 80C51 为核心的单片机,当然,功能或多或少有些改变,以满足不同的需求,其中 AT89C51 就是这几年来在我国非常流行的单片机,它是由美国 ATMEL 公司开发生产的。现在大部分使用 AT89S51 单片机,其他功能一样,只是 AT89S51 能实现在线编程下载。以后我们将用 AT89S51 来完成一系列的实验。

4. MCS-51 单片机的外部引脚功能

单片机的种类、型号很多,封装形式各异。有 40 脚 DIP40 封装的 AT89S51,有 20 脚封装的 AT89C2051,也有 44 脚方形封装的贴片型等。

拿到一块芯片,想要使用它,首先必须要知道芯片的引脚功能,我们用一块称之为 AT89S51 的芯片来看一下单片机的引脚功能及其使用连接,如图 1-8 所示。

(1) 电源引脚

单片机使用的是 5V 电源,其中正极接第 40 引脚,负极(地)接第 20 引脚。

(2) 时钟振荡电路引脚 XTAL1 (19 脚) 和 XTAL2 (18 脚)

单片机是一种时序电路,必须提供脉冲信号才能正常工作。在单片机内部已集成了振荡器,XTAL1 和 XTAL2 分别用作晶体振荡器的输入端和输出端。振荡频率取决于石英晶体的振荡频率,使用内部振荡电路时,只要买来合适的晶振、电容,接于第 18、19 脚之间就可以了,电容取值通常为 10~30pF;采用外部时钟时,18 脚为外部时钟信号输入端;19 脚

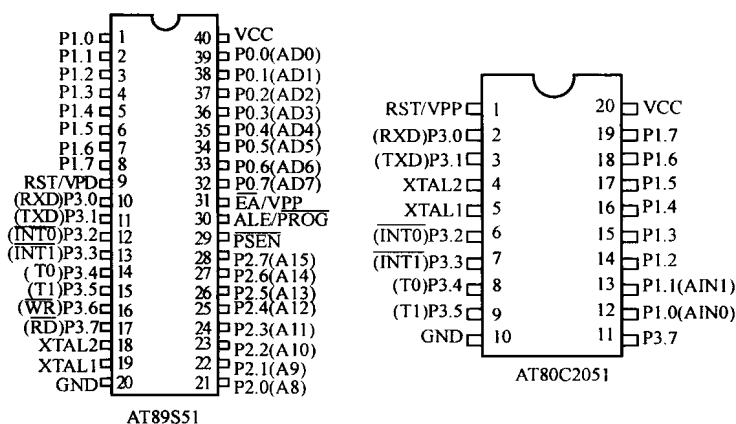


图 1-8 单片机引脚功能图

接地。

(3) 控制线

1) RST/VPD 引脚 (9 脚)——复位/备用电源引脚。

① RST 功能：正常工作时，只要在该引脚上连续保持 2 个机器周期以上高电平，AT89S51 芯片即实现复位操作，复位后一切从头开始，CPU 从 0000H 开始执行指令。

② VPD 功能：在 VCC 掉电情况下，该引脚可接上备用电源，有 VPD 向片内 RAM 供电，以保持片内 RAM 中的数据不丢失。

2) EA/VPP 引脚 (31 脚)——芯片内外 ROM 选择/片内 EPROM 编程电源引脚。

① EA 功能：单片机正常工作时，该引脚为内外 ROM 选择端。80C51 单片机 ROM 寻址范围为 64KB，每字节有 8 位。其中 4KB 在片内，60KB 在片外（8031 芯片无片内 ROM，全部在片外）。当 EA 引脚保持高电平时，先访问内部 ROM，但当 PC（程序地址指针寄存器）值超过 4KB（0FFFH）时，将自动转向执行外部 ROM 中的程序。当 EA 保持低电平时，则只访问外 ROM，而不管芯片内有无 ROM。对于 8031 芯片，片内无 ROM，因此 EA 必须接地。

② VPP 功能：片内有 EPROM 的芯片，在 EPROM 编程期间，此引脚用于施加编程电源。

3) ALE/PROG 引脚 (30 脚)——地址锁存允许/片内 EPROM 编程脉冲。

① ALE 功能：用来锁存允许 P0 口送出的低 8 位地址。

AT89S51 单片机在并行扩展外存储器时，P0 口用于分时传送低 8 位地址和数据信号，且均为二进制数。那么，如何区分是低 8 位地址还是 8 位数据信号呢？当 ALE 信号有效时（高电平），P0 口传送的是低 8 位地址信号；ALE 信号无效时，P0 口传送的是 8 位数据信号。在 ALE 信号的下降沿，锁定 P0 口传送的内容，即低 8 位地址信号。

需要指出的是，当 CPU 不执行访问外 RAM 指令（MOVX）时，ALE 以 1/6 时钟振荡频率的固定频率输出。因此，ALE 信号也可作为外部芯片 CLK 时钟或其他需要。但是，当 CPU 执行 MOVX 指令时，ALE 将跳过一个 ALE 脉冲。ALE 端可以驱动 8 个 LSTTL 门电路。

② PROG 功能：片内有 EPROM 的芯片，在 EPROM 编程期间，此引脚输入编程脉冲。

4) PSEN (29 脚)——外部 ROM 读选通信号。

AT89S51 读外 ROM 时, 每个机器周期内 $\overline{\text{PSEN}}$ 两次有效输出。 $\overline{\text{PSEN}}$ 可作为外 ROM 芯片输出允许 OE 的选通信号。在读内 ROM 或读外 RAM 时, $\overline{\text{PSEN}}$ 无效。该信号可驱动 8 个 LSTTL 门电路。

(4) I/O 引脚 (P0 ~ P3 口)

并行 I/O 接口引脚 (共 32 个, 分成 4 个 8 位并行口)。

1) P0 口 (32 ~ 39 脚)——8 位双向 I/O 口: 在不并行扩展外存储器 (包括并行扩展 I/O 口) 时, P0 口可用作双向 I/O 口。在并行扩展外存储器 (包括并行扩展 I/O 口) 时, P0 口用于分时传送低 8 位地址 (地址总线) 和 8 位数据信号 (数据总线)。P0 口能驱动 8 个 LSTTL 门。

2) P1 口 (1 ~ 8 脚)——8 位准双向 I/O 口 (准双向是指该口内部有固定的上拉电阻)。P1 口能驱动 4 个 LSTTL 门。

3) P2 口 (21 ~ 27 脚)——8 位准双向 I/O 口。通用 I/O 口引脚或地址高 8 位总线引脚; 在不并行扩展外存储器 (包括并行扩展 I/O 口) 时, P2 口可用作双向 I/O 口; 在并行扩展外存储器 (包括并行扩展 I/O 口) 时, P2 口用于传送高 8 位地址 (属地址总线)。P2 口能驱动 4 个 LSTTL 门。

4) P3 口 (10 ~ 17 脚)——8 位准双向 I/O 口。

P3 口每一引脚除可作一般 I/O 口用外, 同时还具有第二功能, 用于特殊信号输入输出和控制信号 (属控制总线)。P3 口引脚对应的第二功能如下:

P3.0—RXD: 串口接收端; 输入。

P3.1—TXD: 串口发送端, 输出。

P3.2— $\overline{\text{INT0}}$: 外部中断 0 请求输入端。

P3.3— $\overline{\text{INT1}}$: 外部中断 1 请求输入端。

P3.4—T0: 定时/计数器 0 外部信号输入端。

P3.5—T1: 定时/计数器 1 外部信号输入端。

P3.6— $\overline{\text{WR}}$: 外 RAM 写选通信号输出端。

P3.7— $\overline{\text{RD}}$: 外 RAM 读选通信号输出端。

P3 口能驱动 4 个 LSTTL 门。

上述的 4 个 I/O 口各有各的用途: 在不并行扩展外存储器 (包括并行扩展 I/O 口) 时, 4 个 I/O 口都可作为双向 I/O 口用; 在并行扩展外存储器 (包括并行扩展 I/O 口) 时, P0 口专用于分时传送低 8 位地址信号和 8 位数据信号, P2 口专用于传送高 8 位地址信号, P3 口根据需要常用于第二功能, 真正可提供给用户使用的 I/O 口是 P1 口和部分未用作第二功能的 P3 口端线。

5. MCS-51 单片机内部结构

MCS-51 系列以 AT89S51 为例, AT89S51 单片机内部组成结构如图 1-9 所示。

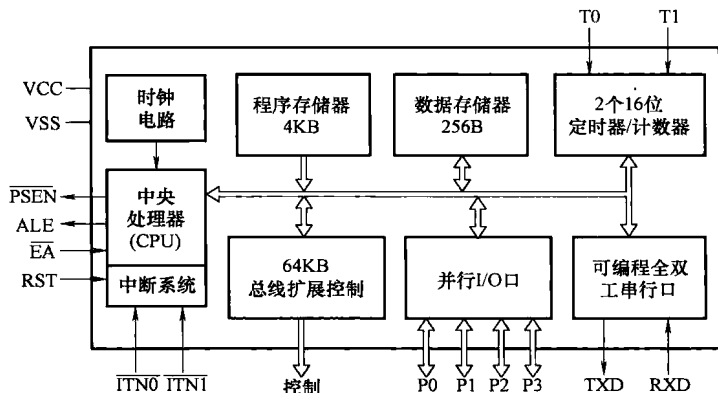


图 1-9 AT89S51 单片机内部组成结构框图

AT89S51 单片机内部集成有中央处理器 (CPU)、存储器 (RAM、ROM)、输入输出 (I/O) 及其他功能部件, 如定时器/计数器、中断系统等, 它们通过地址总线 (AB)、数据总线 (DB) 和控制总线 (CB) 连接在一起。

AT89S51 单片机内部包含的功能部件详细内容如下:

- 1) 1 个 8 位的 CPU;
- 2) 128B 数据存储器 RAM;
- 3) 21 个特殊功能寄存器;
- 4) 4KB 程序存储器 ROM (80C51 有 4KB 掩膜 ROM, 87C51 有 4KB EPROM, 80C31 片内无 ROM);
- 5) 可寻址 64KB 的外部 ROM 和外 RAM 控制电路;
- 6) 2 个 16 位的可编程定时器/计数器; 可用中断;
- 7) 4 个 8 位的并行 I/O 口, 共 32 条可编程 I/O 端线, 用于对外数据传输;
- 8) 1 个可编程全双工串口;
- 9) 5 个中断源, 包括 2 个外中断、2 个定时器中断、1 个串行中断; 可设置成 2 个优先级;
- 10) 1 个片内振荡器和时钟电路, 需要外接晶振和电容。

单片机内部组成细节, 可从如下几个方面考虑:

(1) 微处理器

微处理器是单片机的核心。它主要由寄存器阵列、运算器和控制器三部分组成。

MCS-51 内部有一个功能很强的 8 位中央处理器 (CPU), 它决定了单片机的主要功能特性, 完成对指令的解释和运算, 为执行指令提供必需的运算、逻辑和控制线路。它由运算器、控制器和专用寄存器组等构成。

1) 运算器。

运算器包括算术逻辑运算部件 (ALU)、累加器 (ACC)、寄存器 (B)、暂存器 (TMP1 和 TMP2)、程序状态字寄存器 (PSW)、十进制调整电路以及布尔处理器等。该模块的主要功能是实现算术运算、逻辑运算、数据传送和位变量处理等。具体包括加、减、乘、除算术运算; 与、或、异或、循环、求补等逻辑运算; 十进制调整、加 1、减 1 运算; 置 1、清 0、取反操作; 数据传送操作等。同时还具有一般微机所不具有的位处理功能, 这也是单片机能成为面向控制的微处理器的重要原因。

2) 控制器。

控制器控制着单片机系统完成各种操作。控制器一般包括时钟电路、复位电路、定时控制逻辑、指令寄存器 (IR)、指令译码器 (ID)、程序计数器 (PC) 和信息传送控制部件等。它以主振频率为基准, 由定时控制逻辑发出 CPU 时序, 将指令寄存器中存放的指令码取出送指令译码器进行译码, 再由信息传送控制部件发出一系列控制信号, 控制单片机各部分运行, 完成指令指定的功能。

(2) 存储器空间配置和功能

AT89S51 的存储器配置方式与其他常用的微机系统不同, 属哈佛结构, 它把程序存储器和数据存储器分开, 各有自己的寻址系统、控制信号和功能。程序存储器用于存放程序和表格常数; 数据存储器用于存放程序运行数据和结果。AT89S51 存储空间配置如图 1-10 所示。

AT89S51 的存储器组织结构可以分为 3 个不同的存储空间, 分别是: