



银领工程系列

单片机实验 与综合训练

王宗和 主 编

童一帆 副主编



高等教育出版社
Higher Education Press

银领工程系列

单片机实验与综合训练

王宗和 主 编

童一帆 副主编

高等教育出版社

内容提要

本书用实例方式讲解了单片机的应用及接口电路的扩展方法。全书由两大部分组成:第一部分是基础训练,包括键盘、显示、I/O 接口扩展等基础内容,提供了基本硬件电路连接、软件设计思想和参考程序段;第二部分是综合训练,以典型应用题目为载体,介绍了工程项目的设计方法,提供了 4 个项目,每个项目自总体设计方案开始,讲解了硬件设计过程、软件设计思想和方法,给出了参考电路和程序。

本书适用于高职高专院校计算机、自动控制、数字控制、仪器仪表等专业,也可供从事相关专业的工程技术人员参考。

图书在版编目(CIP)数据

单片机实验与综合训练/王宗和主编. —北京:高等教育出版社,2005. 12

ISBN 7-04-018116-9

I. 单... II. 王... III. 单片微型计算机-高等学校:技术学校-教材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2005)第 133695 号

策划编辑	孙 杰	责任编辑	王莉莉	封面设计	王凌波	责任绘图	朱 静
版式设计	胡志萍	责任校对	张 颖	责任印制	宋克学		

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京凌奇印刷有限责任公司

开 本 787×1092 1/16
印 张 8.25
字 数 190 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landrace.com>
<http://www.landrace.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2005 年 12 月第 1 版
印 次 2005 年 12 月第 1 次印刷
定 价 11.90 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 18116-00

出版说明

为了认真贯彻《国务院关于大力推进职业教育改革与发展的决定》，落实《2003—2007 年教育振兴行动计划》，缓解国内劳动力市场技能型人才紧缺现状，为我国走新型工业化道路服务，自 2001 年 10 月以来，教育部在永州、武汉和无锡连续三次召开全国高等职业教育产学研经验交流会，明确了高等职业教育要“以服务为宗旨，以就业为导向，走产学研结合的发展道路”，同时明确了高等职业教育的主要任务是培养高技能人才。这类人才，既要能动脑，更要能动手，他们既不是白领，也不是蓝领，而是应用型白领，是“银领”。从而为我国高等职业教育的进一步发展指明了方向。

培养目标的变化直接带来了高等职业教育办学宗旨、教学内容与课程体系、教学方法与手段、教学管理等诸多方面的改变。与之相应，也产生了若干值得关注与研究的新课题。对此，我们组织有关高等职业院校进行了多次探讨，并从中遴选出一些较为成熟的成果，组织编写了“银领工程”丛书。本丛书围绕培养符合社会主义市场经济和全面建设小康社会发展要求的“银领”人才的这一宗旨，结合最新的教改成果，反映了最新的职业教育工作思路和发展方向，有益于固化并更好地推广这些经验和成果，很值得广大高等职业院校借鉴。我们的这一想法和做法也得到了教育部领导的肯定，教育部副部长吴启迪专门为首批“银领工程”丛书提笔作序。

我社出版的高等职业教育各专业领域技能型紧缺人才培养培训工程系列教材也将陆续纳入“银领工程”丛书系列。

“银领工程”丛书适用于高等职业学校、高等专科学校、成人高校及本科院校开办的二级职业技术学院、继续教育学院和民办高校使用。

高等教育出版社

2004 年 9 月

前 言

《单片机实验与综合训练》是高职高专院校计算机、自动控制、数字控制、仪器仪表等专业的实践教材,为适应高等工程技术教育,本书努力体现以应用为目的的特点,重点放在基本实践技能和能力的培养上。

《单片机实验与综合训练》符合高职高专对单片机实践能力的基本要求,本书以实例为载体,讲解了单片机的基本功能和应用,力争实现理论与实践相结合,使学生从课堂逐渐走向工程实际,实现理论与实践相结合的目的。全书由两大部分组成:基础训练部分包括7个单元,分别提供了基本硬件电路连接、软件设计思想和参考程序段,完成基本单元的实践学习;综合训练部分提供了4个项目,以应用题目为载体,介绍了工程项目的设计思想和方法,自总体设计方案开始,讲解了硬件设计过程、软件设计思想和方法,给出了参考电路和程序,达到训练解决实际工程问题的目的。

本书的编者有:童一帆、刘焕平(单元1、3、4、6,项目1、2、3),苏梅(单元2、5),王宗和(单元7,项目4,附录)。童一帆负责程序校对。承德石油高等专科学校高嵩副教授担任主审,他对全书内容和形式提出了许多宝贵意见,在此表示真诚的感谢!

由于时间紧迫,本教材尚有待改进之处,希望得到同行专家的认同和指正,希望得到广大读者的意见和建议。

编 者

2005年8月

目 录

第一部分 基础训练

单元 1 键盘及其接口	2	单元 5 并行接口扩展电路	48
一、键盘类型	2	一、并行接口	48
二、非编码键盘与单片机的接口	2	二、并行接口扩展电路	48
三、键盘应用	7	三、并行接口扩展应用举例	48
单元 2 显示器件	12	单元 6 串行接口	53
一、显示类型	12	一、串行接口类型	53
二、显示接口电路	16	二、单片机的串行接口	54
三、显示应用举例	20	三、串行接口应用	58
单元 3 打印机及接口电路	31	单元 7 信号转换与处理	64
一、微型打印机类型	31	一、信号放大技术	64
二、微型打印机与单片机的接口	32	二、多通道输入技术	65
三、打印机应用	34	三、采样/保持电路(S/H)	66
单元 4 A/D、D/A 转换器及接口电路	37	四、滤波电路	66
一、A/D、D/A 转换器类型	37	五、电平转换电路	67
二、接口电路	38	六、开关量信号输出接口电路	68
三、A/D、D/A 转换器应用举例	41	七、量程转换电路	68

第二部分 综合训练

项目 1 温度采集显示系统	72	项目 3 自动车库控制系统	97
一、设计要求	72	一、设计要求	97
二、设计方案	72	二、设计方案	97
三、硬件设计	74	三、硬件设计	98
四、软件设计	76	四、软件设计	101
五、调试提示	80	项目 4 超市人数统计显示装置	111
项目 2 电梯控制系统模型	81	一、设计要求	111
一、设计要求	81	二、设计方案	111
二、设计方案	81	三、硬件设计	111
三、硬件设计	81	四、软件设计	112
四、软件设计	83		
附录 1 51 单片机指令速查表	119		
附录 2 ASCII 码表	124		
参考文献	125		

第一部分

基础训练

把计算机的五个组成部分:运算器、存储器、控制器、输入电路及输出电路集成在一个芯片上形成的芯片级特殊计算机称为单片计算机(single chip microcomputer),简称单片机,当前单片机均采用冯·诺依曼原理,即存储程序-执行程序的过程,因此,单片机的工作原理及过程与普通计算机完全一样。在此,重复计算机执行程序的工作过程。

在计算机通电正常工作后,就进入如下过程:

- (1) PC 中的指令地址放到地址总线;
- (2) 从地址指示的存储器中取出指令,由数据总线送到 CPU;
- (3) PC 中的地址递增指令字节数,即指向下一指令的首地址;
- (4) CPU 分析指令,完成指令功能;
- (5) 若有中断且允许中断,转中断处理;
- (6) 返回(1)。

注意:计算机的工作是周而复始、无穷无尽的,因此在书写完某种工作程序时,需要使计算机进入无限循环或迫使 CPU 停机。

单片机从功能和组成上都是应控制领域的要求而发展的,特别是近几年新一代单片机着重发展的是各种控制功能单元,如 A/D、PWM、PAC、WDT 等,已完全突破了传统微型计算机的内容。虽然各种控制系统仍以单片机为核心,但单片机已经完全融入应用系统中,是应用系统的一个组成部分,因此也将其称为微控制器(microcontroller)。

由于单片机是实践性很强的课程,学习中注意单片机内部资源,将各部分灵活运用,在实践中提高,同时,学习功能接口的处理方法,以便组成运用系统。

键盘及其接口

键盘是单片机控制系统最常用、最简单的输入设备。用户可以通过键盘输入数据或命令,实现简单的人机对话。

一、键盘类型

单片机控制系统所用的键盘有编码键盘和非编码键盘两种。

编码键盘除了键开关外,还有专门的硬件电路,用于识别闭合键并产生键代码。不仅如此,编码键盘一般还有去键抖动电路和防串键保护电路等。这种键盘的优点是所需软件简短,但硬件电路较复杂,价格较昂贵。目前在单片机控制系统中使用不多。

非编码键盘仅由键开关组成,其他工作如键识别、键代码的产生、去抖动等,不是由硬件完成而是由软件完成的。为了简化硬件电路、降低成本,目前单片机控制系统中大多数采用非编码键盘。

非编码键盘的键开关可以排列成线性形式或矩阵形式,因此非编码键盘有线性非编码键盘和矩阵非编码键盘两种。

二、非编码键盘与单片机的接口

1. 线性非编码键盘与单片机的接口

线性非编码键盘的键开关排成一行或一列的形式,它与单片机的接口电路如图 1-1-1 所示。

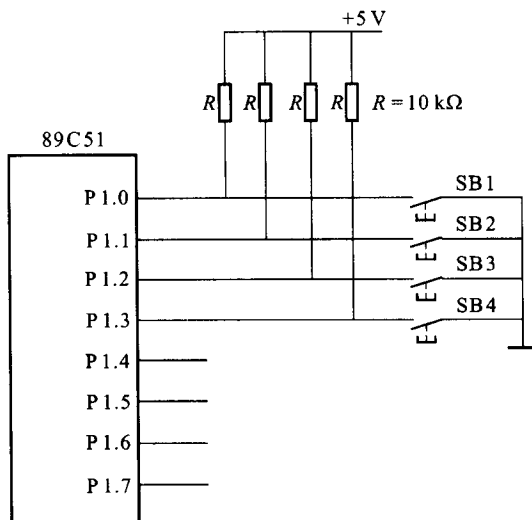


图 1-1-1 线性非编码键盘与单片机接口电路

线性非编码键盘的工作原理非常简单,每个按键的一端与单片机的 I/O 线直接相连,同时经上拉电阻接至 +5 V 电源,另一端为公共接地端。当键未被按下时,与此键相连的 I/O 线获得高电平;当键被按下时,与此键相连的 I/O 线获得低电平,单片机只要读取 I/O 口状态,就可以获取按键信息,识别有无键按下和哪个键被按下。

单片机进行键处理时,首先读 I/O 口状态并判断是否有键按下,如果有键按下,则等待 10 ms 以消除抖动,然后再读 I/O 口状态并再一次判断是否有键按下,若仍有键按下,则认为键盘上有键处于稳定的闭合状态,于是逐个检测按键的状态,并执行与按下键相对应的子程序;若第二次判断时,无键按下,则认为第一次是键抖动引起的,属于误判。

键处理流程如图 1-1-2 所示。

键处理程序如下:

```

MOV    P1, #0FFH
UP1:   MOV    A, P1           ;读 I/O 口状态
        ANL    A, #0FH        ;屏蔽无用位
        CJNE   A, #0FH, NEXT1  ;有闭合键?
        SJMP   UP1
NEXT1:  LCALL  D10ms           ;延时 10 ms 消除抖动
        MOV    A, P1           ;再读 I/O 口状态
        ANL    A, #0FH
        CJNE   A, #0FH, NEXT2  ;有闭合键?
        SJMP   UP1
NEXT2:  JB     P1.0, NEXT3     ;SB1 按下?
        LCALL  SB1             ;SB1 键处理程序
        LJMP   UP1
NEXT3:  JB     P1.1, NEXT4     ;SB2 按下?
        LCALL  SB2             ;SB2 键处理程序
        LJMP   UP1
NEXT4:  JB     P1.2, NEXT5     ;SB3 按下?
        LCALL  SB3             ;SB3 键处理程序
        LJMP   UP1
NEXT5:  JB     P1.3, UP1        ;SB4 按下?
        LCALL  SB4             ;SB4 键处理程序
        LJMP   UP1

```

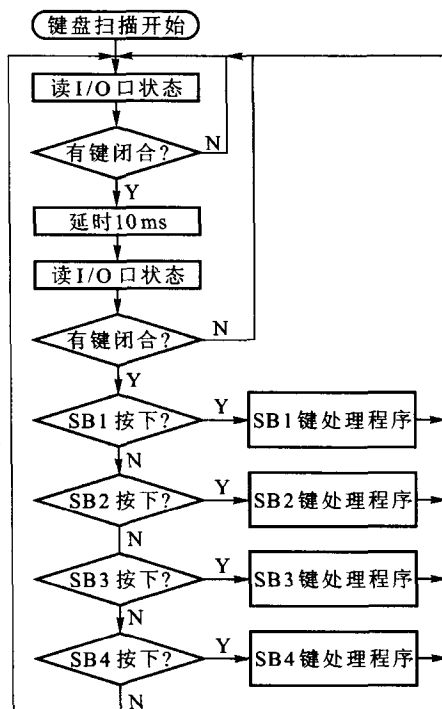


图 1-1-2 键处理流程图

线性非编码键盘的接口电路简单,驱动程序也简单。但是,由于每个键占用一条 I/O 线,当按键较多时,I/O 口线浪费较大,因此线性非编码键盘适用于按键较少的场合。当按键较多时,通常采用矩阵非编码键盘。

2. 矩阵非编码键盘与单片机的接口

矩阵非编码键盘的键开关处于行、列线的交叉点上。每个键开关的一端与行线相连,另一端与列线相连。图 1-1-3 所示是一个 4×4 矩阵非编码键盘与单片机的接口电路图。

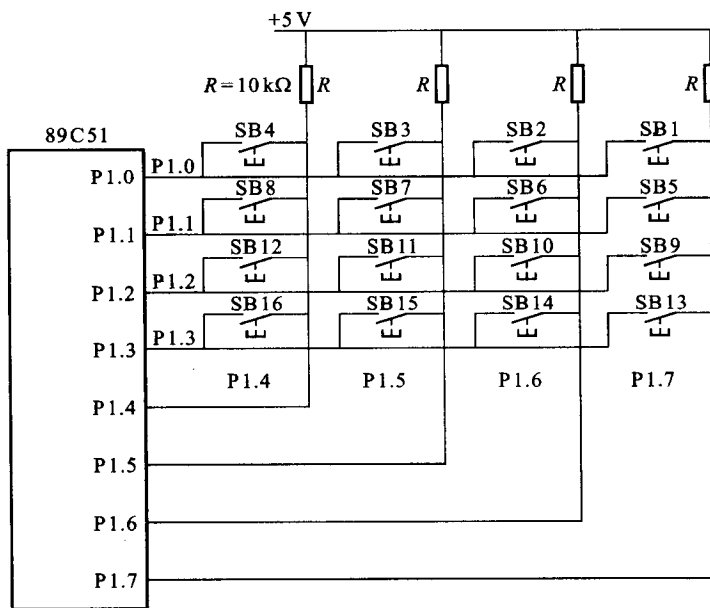


图 1-1-3 4×4 矩阵非编码键盘与单片机的接口电路

这个矩阵非编码键盘的行线接到单片机的 P1.0 ~ P1.3 上,列线接到 P1.4 ~ P1.7 上,同时列线经上拉电阻接至 +5 V 电源上。当键被按下时,它所在的行线和列线被短接。如果预先使某行线输出低电平,则在读入列线状态时,若该行线上的 4 个按键中有键被按下,那么闭合键所在的列线状态为 0,其余列线状态为 1。据此可以确定闭合键的键代码。

在以上硬件接口电路的基础上,通过程序就可以确定:

(1) 键盘中是否有键按下。方法是先使所有行线输出低电平,然后读所有列线的状态,若列线状态全为高电平,则说明没有键按下;若列线状态有低电平,则说明有键按下。

(2) 去除键的机械抖动。方法是在判别到键盘上有键按下后,延迟一段时间,再判别键盘状态,若仍有键按下,则认为键盘上有键处于稳定的闭合状态;否则认为是键抖动。

(3) 如有键被按下,则寻找闭合键所在位置,求出其键代码。传统的方法是逐行扫描。扫描的方法是使 0 号行线输出低电平,其余行线为高电平,读取列线状态,若列线状态有 0,则闭合键处于该行和状态为 0 的列线的交叉点上,据此求出此键的键代码(键代码 = 行号 + 列号);若列线状态为全 1,则表明闭合键不在此行,紧接着使 1 号行线输出低电平,其余行线为高电平,继续读列线状态,用同样方法,检查闭合键是否在此行;依此类推。

(4) 根据键代码转入闭合键所对应的程序中去。常用的方法是设置一个转移指令表,表中

每一条转移指令与一个键相对应,转移指令中的转移地址就是完成此键功能的子程序的入口地址。根据闭合键的键代码查转移指令表,就可以转到对应的键功能子程序去。

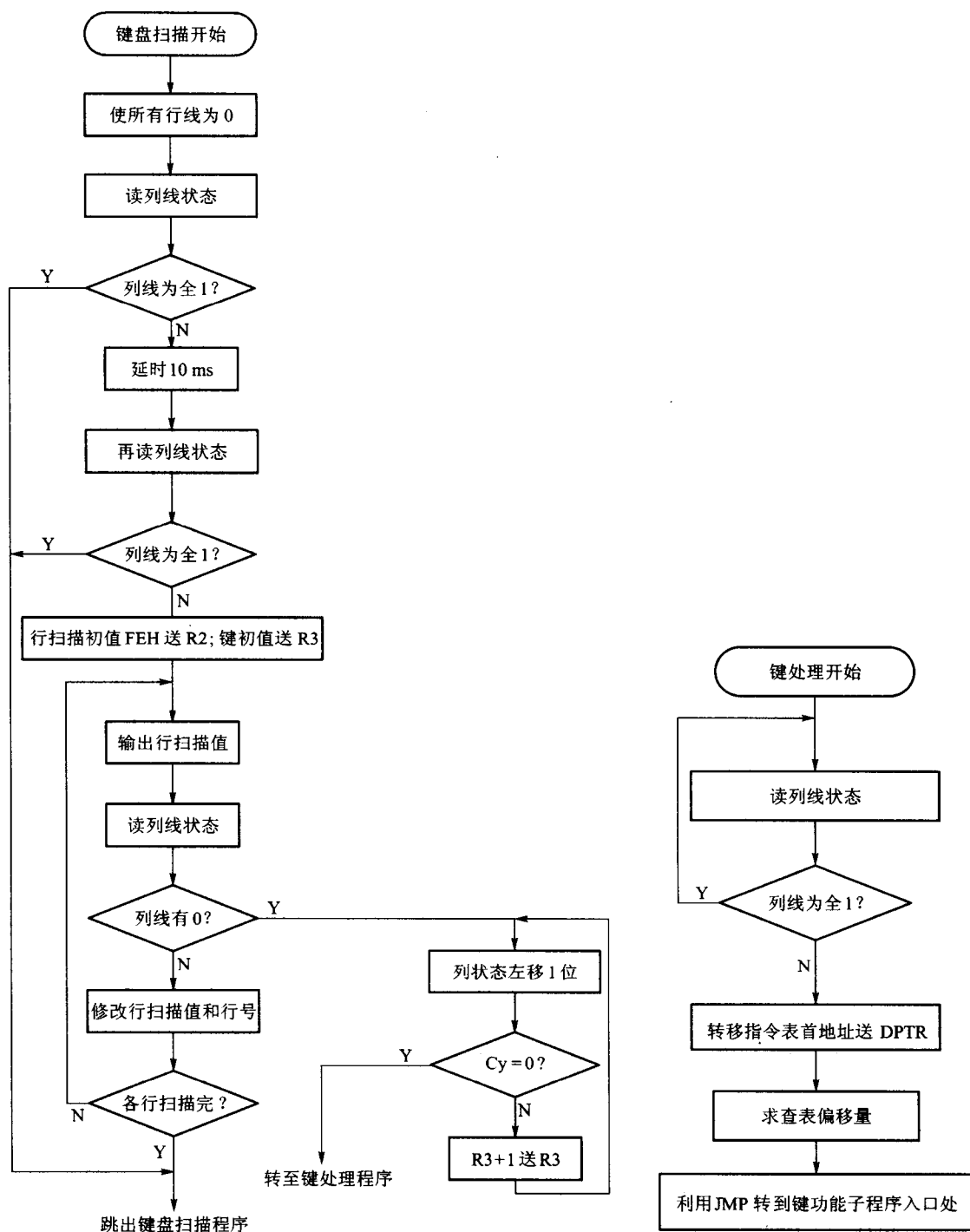


图 1-1-4 键扫描和键处理流程图

键扫描和键处理程序流程如图 1-1-4 所示。键值寄存器与按键的关系为:

R3	0	1	2	...	D	E	F
键	SB1	SB2	SB3	...	SB14	SB15	SB16

键扫描和键处理程序如下:

```

MOV    P1,#0F0H      ;使所有行线为 0
MOV    A,P1           ;读列线状态
ANL    A,#0F0H        ;屏蔽无用位
CJNE   A,#0F0H,NEXT1 ;列线为全 1?
LJMP   DISP

NEXT1: LCALL  D10ms    ;延时 10 ms 消除抖动
MOV    A,P1           ;再读列线状态
ANL    A,#0F0H
CJNE   A,#0F0H,NEXT2 ;列线为全 1?
LJMP   DISP

NEXT2: MOV    R2,#0FEH ;行扫描初值送 R2
MOV    R3,#00H        ;键初值送 R3
MOV    R7,#4          ;扫描次数送 R7
UP1:   MOV    P1,R2    ;输出行扫描值
MOV    A,P1           ;读列线状态
ANL    A,#0F0H
CJNE   A,#0F0H,NEXT3 ;列线有 0?
MOV    A,R2           ;修改行扫描值,准备扫描下一行
RL     A
MOV    R2,A
MOV    A,R3            ;修改键值
ADD    A,#4
MOV    R3,A
DJNZ   R7,UP1         ;各行都扫描一遍?
LJMP   DISP

NEXT3: RLC    A        ;列线状态左移
JNC    NEXT4          ;某条列线为 0?
INC    R3             ;键值加 1,以求键代码
SJMP   NEXT3

NEXT4: MOV    A,P1     ;读列线状态
ANL    A,#0F0H        ;屏蔽无用位
CJNE   A,#0F0H,NEXT4 ;列线全 1? 以等待键释放
MOV    DPTR,#TAB      ;转移指令表首地址送 DPTR
MOV    A,R3           ;键代码乘 3,以求查表偏移量

```

```

MOV    B, #3
MUL    AB
JMP     @ A + DPTR      ;转到相应的转移指令处
TAB:   LJMP  SB1
       LJMP  SB2
       .....
       LJMP  SB16

```

程序中的 SB1 ~ SB16 是各个键功能子程序的入口地址。(SB1 ~ SB16 子程序略。)

上述键盘驱动程序没有考虑防串键问题。串键是指多个键同时按下时或前面键没有释放又按下新的键时所产生的问题。判别是否有串键的方法是:在程序中一定将所有行都扫描一次,而不是检测到列线状态有 0 时就结束。在所有行都扫描一遍之后,如果不止一次得到的列状态有 0,则说明出现了串键。解决串键的最简单的方法是这次扫描作废,再来一遍。实际上,由于扫描速度很快,真正找到两个键同时按下的情况是很少的。

三、键盘应用

例 1.1.1 某单片机控制系统, P1 口接有 8 个发光二极管, P3.0、P3.1、P3.2、P3.3 接有 4 个开关 SB1、SB2、SB3、SB4, 试画出接口电路, 并编程使得当 SB1 按下时 8 个发光二极管全亮; 当 SB2 按下时 8 个发光二极管闪亮; 当 SB3 按下时 8 个发光二极管由左向右点亮; 当 SB4 按下时 8 个发光二极管全灭。

解: 电路如图 1-1-5 所示。

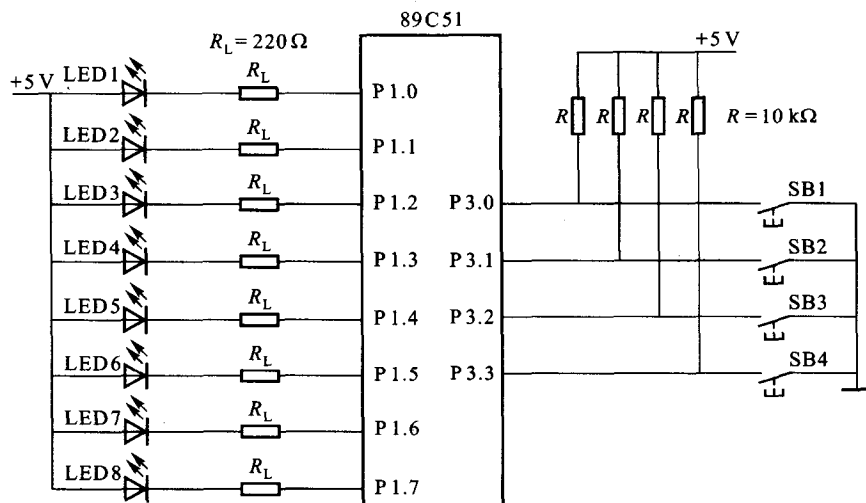


图 1-1-5 例 1 电路图

程序如下:

```

MOV    B, #0FEH      ;置显示初值
MOV    P3, #0FFH     ;使口锁存器置 1
MOV    P1, #0FFH     ;使发光二极管熄灭

```

```

UP1:   MOV     A,P3           ;读 I/O 口状态
       ANL     A,#0FH        ;屏蔽无用位
       CJNE    A,#0FH,NEXT1   ;有闭合键?
       SJMP     UP1

NEXT1: LCALL    D10ms         ;延时 10 ms 消除抖动
       MOV     A,P3           ;再读 I/O 口状态
       ANL     A,#0FH
       CJNE    A,#0FH,NEXT2   ;有闭合键?
       SJMP     UP1

NEXT2: JB      P3.0,NEXT3     ;SB1 按下?
       LCALL   SB1            ;SB1 键处理程序
       LJMP    UP1

NEXT3: JB      P3.1,NEXT4     ;SB2 按下?
       LCALL   SB2            ;SB2 键处理程序
       LJMP    UP1

NEXT4: JB      P3.2,NEXT5     ;SB3 按下?
       LCALL   SB3            ;SB3 键处理程序
       LJMP    UP1

NEXT5: JB      P3.3,UP1       ;SB4 按下?
       LCALL   SB4            ;SB4 键处理程序
       LJMP    UP1


SB1:   MOV     P1,#00H        ;使发光二极管点亮
       RET

SB2:   MOV     P1,#00H        ;使发光二极管点亮
       LCALL   D2S            ;调用延时子程序
       MOV     P1,#0FFH       ;使发光二极管熄灭
       LCALL   D2S
       RET

SB3:   MOV     P1,B           ;输出循环码
       LCALL   D2S
       MOV     A,B
       RL      A              ;使循环码左移,准备使下一个灯点亮
       MOV     B,A
       RET

SB4:   MOV     P1,#0FFH       ;使发光二极管熄灭
       RET

```

程序中 D10ms、D2s 为延时子程序,从略。

程序如下:

```

START: MOV    P1, #0F0H      ;使所有行线为 0
      MOV     A, P1          ;读列线状态
      ANL     A, #0F0H      ;屏蔽无用位
      CJNE    A, #0F0H, NEXT1 ;列线全 1?
      LJMP     START
NEXT1: LCALL    D10ms         ;延时 10 ms 消除抖动
      MOV     A, P1          ;再读列线状态
      ANL     A, #0F0H
      CJNE    A, #0F0H, NEXT2 ;列线为全 1?
      LJMP     START
NEXT2: MOV     P1, #0F0H      ;P1.3 ~ P1.0 输出 0000
      MOV     A, P1          ;读 P1.7 ~ P1.4 的值
      ORL     A, #0FH        ;屏蔽列值的低 4 位
      MOV     R2, A          ;保存列值
      MOV     P1, A          ;由 P1.7 ~ P1.4 输出第一步中的值
      MOV     A, P1          ;读 P1.3 ~ P1.0 的值
      ORL     A, #0F0H      ;屏蔽行值的高 4 位
      ANL     A, R2          ;行、列组合,形成键位置码
      MOV     R7, #16
      MOV     R2, A
      MOV     DPTR, #TAB
      MOV     R3, #0         ;键代码初值送 R3
      CLR     A
UP1:   MOVC    A, @ A + DPTR  ;从键代码表中取出键位置码
      CJNE    A, R2, NEXT3    ;与所按下键的键位置码比较
      SJMP     NEXT4
NEXT3: INC     R3             ;键代码加 1
      MOV     A, R3          ;准备取下一个键位置码
      DJNZ    R7, UP1        ;所有键位置码都比较了?
      LJMP     START
NEXT4: MOV     P1, #0F0H
      MOV     A, P1          ;读列线状态
      ANL     A, #0F0H      ;屏蔽无用位
      CJNE    A, #0F0H, NEXT4 ;列线为全 1? 以等待键释放
      MOV     A, R3
      CPL     A              ;将键代码取反
      MOV     P3, A          ;按键名点亮发光二极管

```

```
LJMP    START
TAB:    DB      7EH,0BEH,0DEH,0EEH,7DH,0BDH,0DDH,0EDH
        DB      7BH,0BBH,0DBH,0EBH,77H, 0B7H,0D7H,0E7H
```

本例的键识别方法采用反转法。