

必須濱ノ業大学 远程教育学院

第12章 MCS-51的键盘显示接口



主要内容:

72.00

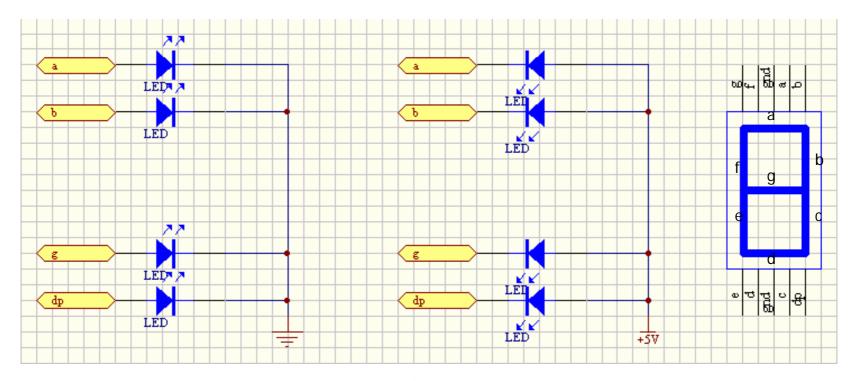
- ■LED显示原理及接口电路设计
- ■键盘原理及接口电路设计
- ■键盘显示接口电路设计
- ■MCS-51单片机与BCD码拨盘的接口设计

12.1 LED显示电路设计



LED显示器(数码管)分类

- 常用的数码管可分为7段和"米"字段两种
- 从电气特性上可分为"共阴极"和"共阳极"两种



为使LED显示不同的符号或数字,要为LED提供段码(或称字型码)。

提供给LED显示器的段码(字型码)正好是一个字节(8段)。各段与字节中各位对应关系如下:

代码位	D7	D6	D5	D4	D3	D2	D1	DO
显示段	dp	g	f	е	d	С	b	a

按上述格式,8段LED的段码如下表所示。

LED段码表(8段)

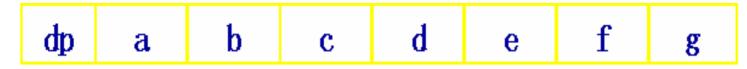
显示字符	共阴极 段码	共阳极 段码	显示字符	共阴极 段码	共阳极 段码
	72.73	1×1-7		TANT	12.79
0	3FH	СОН	С	39H	C6H
1	06H	F9H	d	5EH	A1H
2	5BH	A4H	E	79 H	86H
3	4FH	ВОН	F	71H	8EH
4	66H	99H	Р	73H	8CH
5	6DH	92H	U	3EH	C1H
6	7DH	82H	T	31H	CEH
7	07H	F8H	у	6EH	91H
8	7FH	80H	Н	76H	89H
9	6FH	90H	L	38H	С7Н
A	77FH	88H	"灭"	ООН	FFH
b	7CH	83H	:	•••	•••

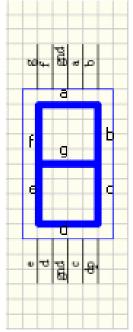


注意: 段码是相对的,它由各字段在字节中所处的位决定。例如前面表中8段LED段码是按格式:

dp g f e d c b a

而形成的,"0"的段码为3FH(共阴)。反之,如将格式改为下列格式:



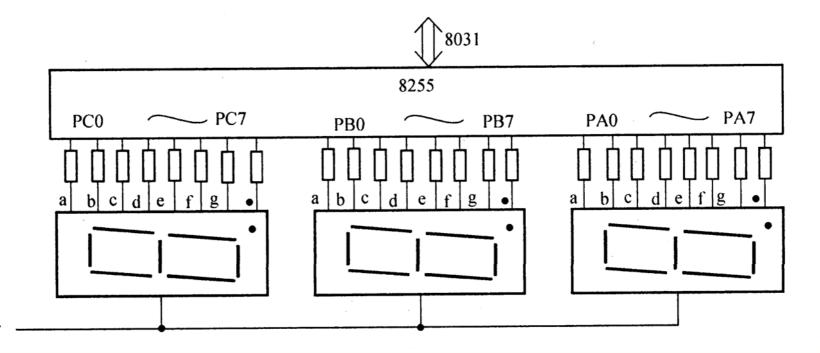


则 "0"的段码为7EH(共阴)。字型及段码由设计者自行设定,习惯上还是以"a"段对应段码的最低位。

12.1.2 LED的静态显示

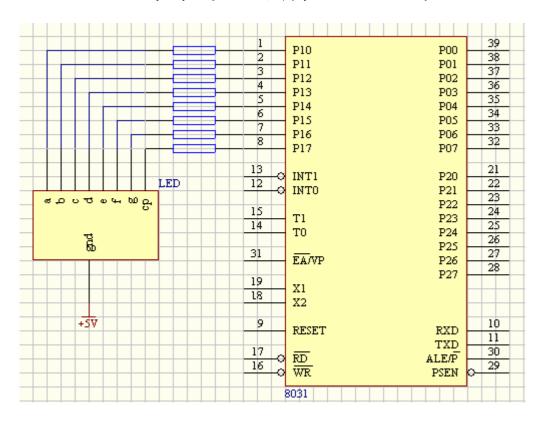
基本原理

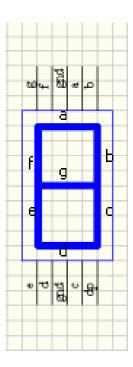
- 多个数码管显示时,同时点亮显示,每一位数码管恒定的显示,不闪烁。
- 显示程序简单,但是要求较多的I/O口线
- 驱动电流较小



LED的静态显示

简单的静态显示



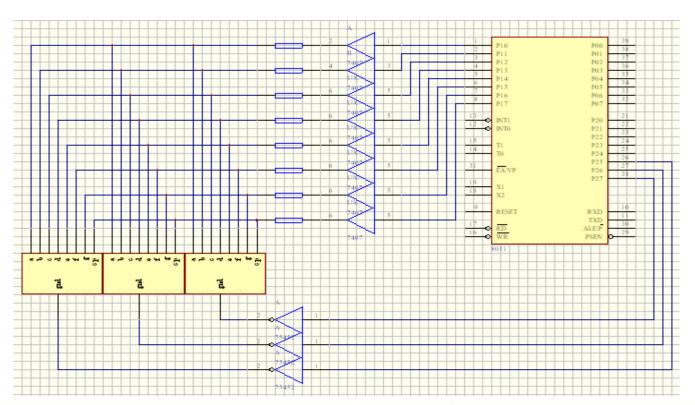


共阳极数码管静态显示示意图

12.1.3 LED的动态显示



- 多个数码管显示时,依次循环点亮每一个数码管,利用人的视觉 暂留看到整个显示内容,只有循环速度足够快,才不闪烁。
- 显示程序较复杂,但是节省I/O口线
- 驱动电流较大



12.2 键盘接口的设计



要点:

◆键盘的分类

编码式:由专门的硬件(8279等)识别按下的键码。非编码式:依靠软件实现键码的识别。

- ◆非编码键盘的结构
 - 独立式键盘
 - 矩阵式键盘
- ◆软件实现按键识别的方法
 - 扫描法
- ◆键盘使用中的注意事项
 - 键盘的抖动

12.2.1 键盘接口的工作原理

独立式键盘接口和矩阵式键盘接口。

1.独立式键盘接口

各键相互独立,每个按键各接一根输入线,通过检测输入线的电平状态可很容易判断那个键被按下。

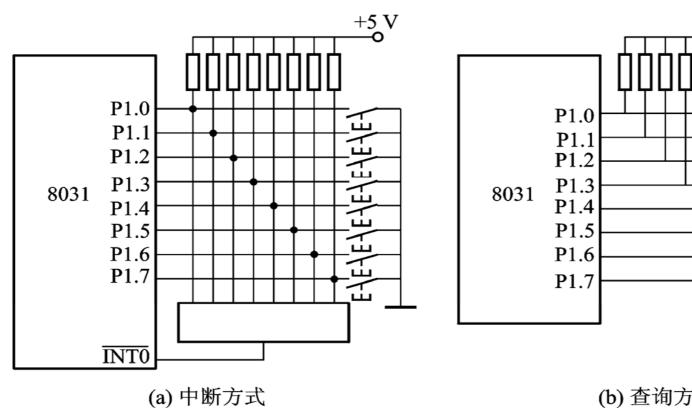
此种接口适于键数较少或操作速度较高的场合。

独立式键盘的检测方式



+5 V

€\<u></u>€\€\€\€\€\€\€\€\

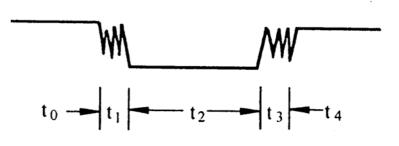


(b) 查询方式

键盘的抖动问题

键盘的抖动

- 抖动时间一般为5~10ms。
- 为了保证CPU对键的闭合作一次, 而且是仅作一次处理,必须消除抖 动
- 可采用软、硬件方法消除抖动。



键盘工作时输出的电压波形

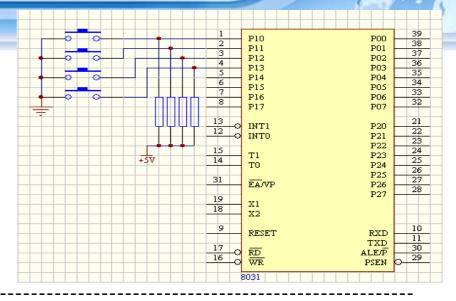
**软件消抖原理

- 1) 判断是否有键按下;
- 2) 若有键按下,调用延时程序(延时时间大于10ms);
- 3) 再次判断是否有键按下,并读入相应的键值.

12.2.2 独立式键盘编程

- 1)每个按键占用一根口线
- 2) 电路配置灵活,软件结构简单。
- 3)适用于按键较少的场合。

特点:可直接判断哪个键按下



独立式键盘的处理程序(实际应用时要加延时消除抖动)

; ------

LOOP: MOV A,P1

ANL A,#0FH

CJNE A,#OFH,KEY

SJMP LOOP

KEY: CJNE A,#OEH,NEXT1

ACALL KEY1

SJMP LOOP

NEXT1: CJNE A,#0DH,NEXT2

ACALL KEY2

;读P1口电平状态

;取P1口低4位

; 判断是否有键按下

,如没有键按下,重新扫描

; 判断是否为一号键按下

;调用一号键处理子程序

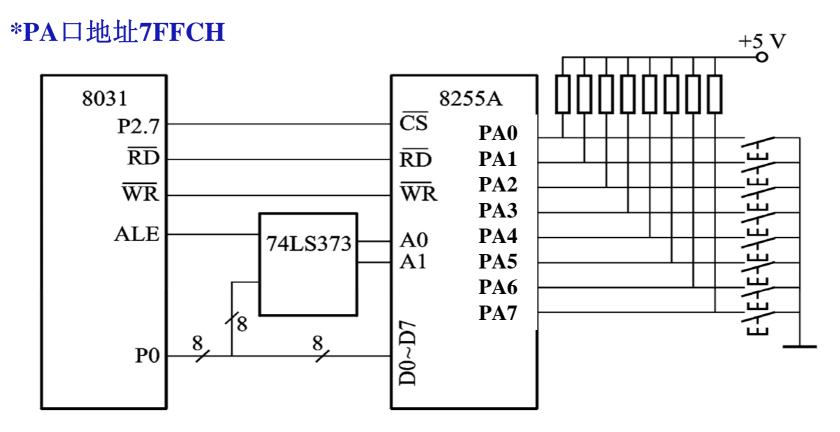
;判断是否为2号键按下

14

8255A扩展I/0口的独立式按键接口电路

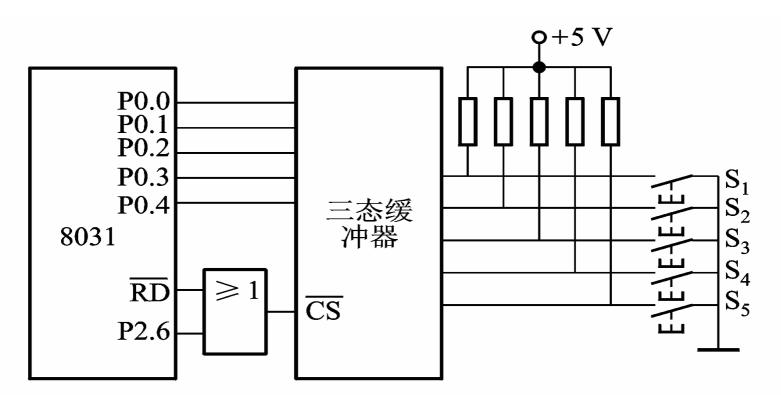


*8255A工作于基本输入输出方式



用三态缓冲器扩展的按键接口电路





三态缓冲器的地址: BFFFH

对上图独立式键盘编程,软件消抖,查询方式检测键的状态。仅有一键按下时处理(不处理多键同时按下)。

KEYIN: MOV DPTR, #OBFFFH; 键盘端口地址BFFFH

MOVX A, @DPTR ; 读键盘状态

ANL A, #1FH ; 屏蔽高三位

MOV R3, A , 保存键盘状态值

LCALL DELAY10 ; 延时10ms去键盘抖动

MOVX A, @DPTR ; 再读键盘状态

ANL A, #1FH ; 屏蔽高三位

CJNE A, R3, OUT ; 两次不同,返回

CJNE A, #1EH, KEY2 ; 相等, 有键按下,

;判断是否是S1?不是转KEY2

;于柱序PKEYT

KEY2: CJNE A, #1DH, KEY3;S2键未按下,转KEY3

LJMP PKEY2 ; S2键按下,转PKEY2处理

KEY3: CJNE A, #1BH, KEY4; S3未按下,转KEY4

LJMP PKEY3 ; S3按下,转PKEY3处理

KEY4: CJNE A, #17H, KEY5; S4键未按下,转KEY5

LJMP PKEY4 ; S4按下,转PKEY4处理

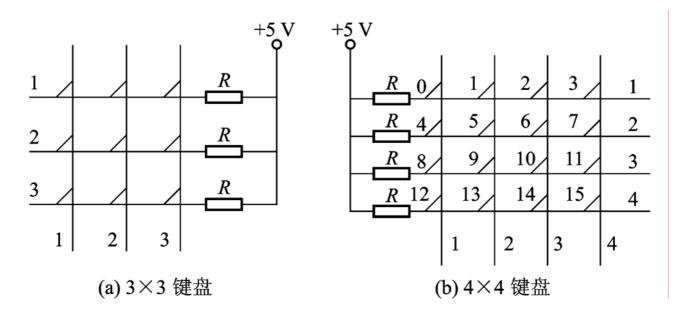
KEY5: CJNE A, #0FH, OUT; S5未按下,转OUT

LJMP PKEY5 ; S5按下,转PKEY5处理

OUT: RET;重键或无键按下,从子程序返回

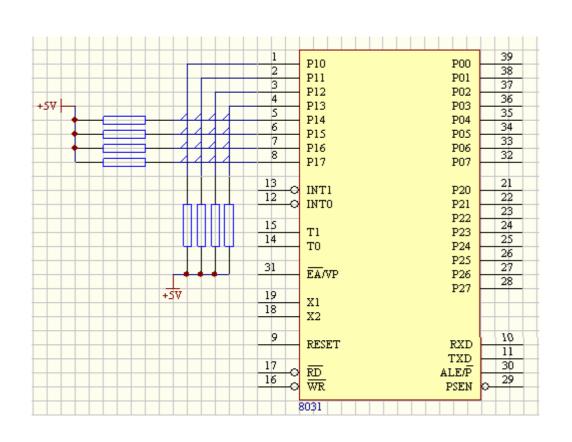
12.2.3 矩阵式(行列式)键盘接口

用于按键数目较多的场合,由行线和列线组成,按键位于行、列的交叉点上,如下图。



**在需要按键数目较多的场合,矩阵式键盘与独立式键盘相比,要节省较多的I/O口线。

矩阵式键盘的一种接线方式



矩阵式键盘示意图

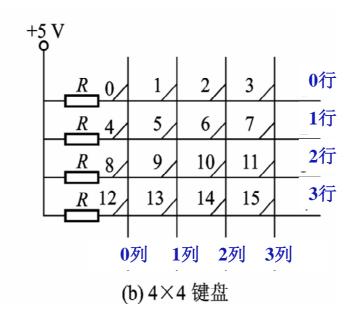
矩阵式键盘的工作过程



- ▶ 键没有按下时行线和列线是分开的,键按下将相应的行线列线短接。
- ➤ 连接行线的管脚作为输入口,没有键按下时,被 外电路拉高为"1"
- ➤ 连接列线的管脚作为输出口,需要扫描键盘时, 逐次输出低电平,若此时有键按下,则与其交叉 的行线回读值为"0",否则为"1"。
- ➤ 根据当前的扫描列线和回读电平为"0"的行线可确 定按下的键值。

矩阵式键盘键码的判别

▶对于一个规则编码的矩阵式键盘,可以通过对键盘的列线依次进行扫描(输出低电平),然后回读行线,如果哪一行的电平为低,说明该行、该列的键被按下。该键的键码为: 键码=该行首键码+该列号

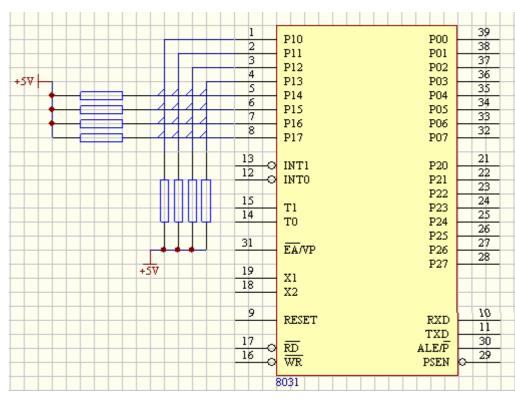


矩阵式按键的识别方法(扫描法)



扫描法

- 依次置低键盘的列扫描线
- 回读键盘的行线
- 根据键盘的行、列线状态 来计算按下键的键号



矩阵式键盘示意图

键盘的扫描方式



程序控制扫描方式

- 当单片机空闲时,才调用键盘扫描程序 定时扫描方式
 - 每隔一定时间对键盘扫描一次,通常采用单片机内定时器确定键盘扫描时间间隔

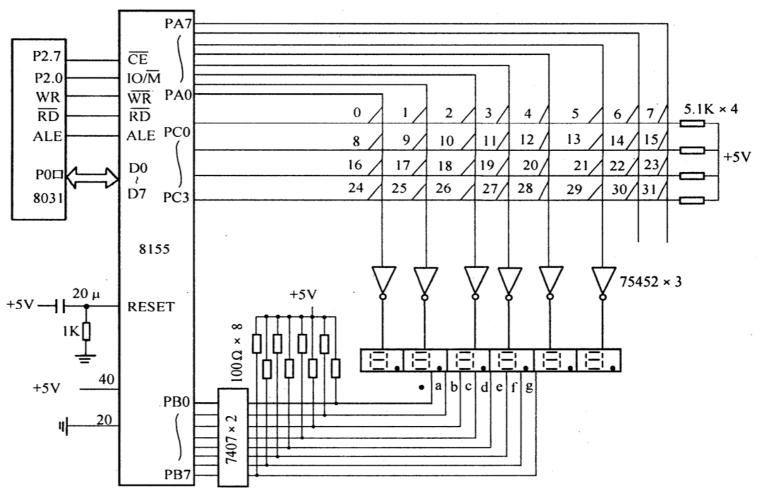
中断扫描方式

- 利用按键操作产生的中断来扫描键盘,优点是 及时响应键盘操作,节省CPU时间

12.3 键盘显示接口电路设计

12.3.1 8155扩展键盘显示

例1 8155PA口地址7F01H 8155PB口地址7F02H 8155PC口地址7F03H

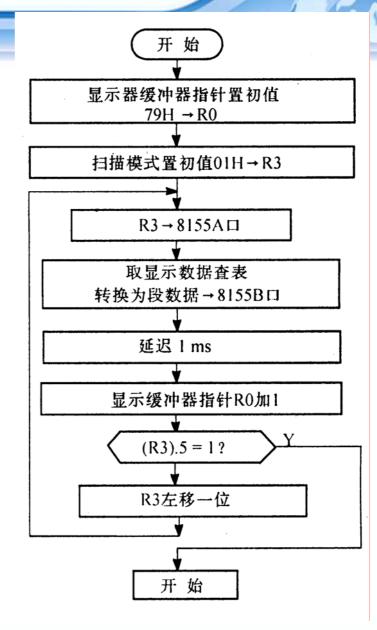


显示程序设计

软件任务

- 假设在79H~7EH已经 存放了用于显示的6位 数据
- 要求设计显示子程序, 实现6个数码管的动态 显示
- 显示字符码可采用共阴 极字符码

程序流程



显示子程序流程

显示子程序

显示子程序,完成显示缓冲区79H~7EH中6位数据的显示工作

主程序中应设好8155的工作方式; PA 方式0输出, PB方式0输出

PC 方式0输入

·______

DIR: MOV R0,#79H

MOV R3,#01H

MOV A,R3

LD0: MOV DPTR,#7F01H

MOVX @DPTR,A

MOV DPTR,#tab1

MOV A,@R0

MOVC A,@A+DPTR

DIR1: MOV DPTR,#7F02H

MOVX @DPTR,A

ACALL DL11

INC R0

MOV A,R3

CJNE A,#00100000B,LD1

RET

LD1: RL A

MOV R3,A

AJMP LD0

;置缓冲器指针初值

;显示位数

;指向8155的PA口

;置相应数码管位选择

;指向字符码表格头地址

;取待显示数据

;取待显示数据的字符码

;指向8155的PB口

;送出段数据

;延迟1ms

显示子程序

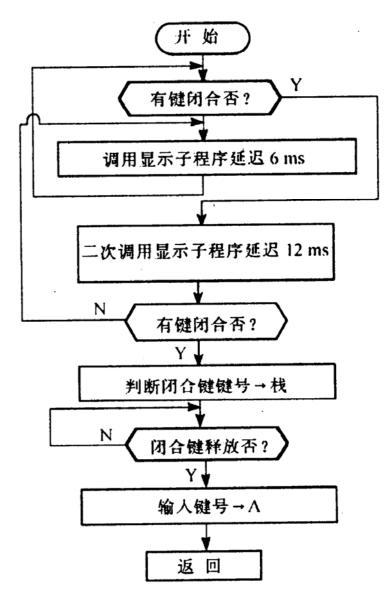
```
延时子程序
             R7,#02H
DL11:
      MOV
             R6,#0FFH
      MOV
            R6,DL6
DL6:
      DJNZ
            R7,DL6
      DJNZ
      RET
   显示字符码表格(共阴极LED段数据表格,对应显示0~F)
           3FH,06H,5BH,4FH,66H,6DH,7DH,07H
tab1:
      \mathbf{DB}
           7FH,6FH,77H,7CH, 39H,5EH,79H,71H
      DB
```

键盘程序设计

设计要求

- 判别键盘上有无键闭合
- 去除键盘的机械抖动
- 判别闭合键的键号
- 使处理器对键的一次闭 合仅作一次处理

程序流程



键盘处理子程序

;-----

键盘输入子程序,调用结束后累加器A中为键号

;-----

KEYI: ACALL KS1 ;调用判有无键闭合子程序,若累加器A不为0,

;则有键按下

JNZ LK1 ; 有键按下,程序转

NI: ACALL DIR ;调显示子程序,延迟6ms

AJMP KEYI

AJMP NI

LK1: ACALL DIR ACALL DIR ;调显示子程序,延迟12ms

ACALL KS1 ,则业小 1 生儿, 处处12ms

JNZ LK2 ;若有键按下,则开始辨别键号

LK2: MOV R2,#0FEH ; R2存放列扫描数据(第0列)

MOV R4,#00H ; 列号

LK4: MOV DPTR,#7F01H ;指向PA口

MOV A,R2 ; 送出列扫描数据

MOVX @DPTR,A ;置相应列线为低电平

INC DPTR :指向PC口

MOVX A,@DPTR ;读取行线(PC口)状态

ANL A,#0FH

CJNE A,#0EH,LONE ;依次判断各行有无键按下

键盘处理子程序

7.73

MOV A,#00H

AJMP LKP

LONE: CJNE A,#0DH,LTW0

MOV A,#08H

AJMP LKP

LTW0: CJNE A,#0BH,LTHR

MOV A,#10H

AJMP LKP

LTHR: CJNE A,#07H,NEXT

MOV A,#18H

LKP: ADD A,R4 ;计算键号并压栈

PUSH A

LK3: ACALL DIR

ACALL KS1

JNZ LK3 ;若键未释放则等待

POP A ;键号出栈

RET

NEXT: INC R4 ;下一列的列号

MOV A,R2

CJNE A,#7FH,KND

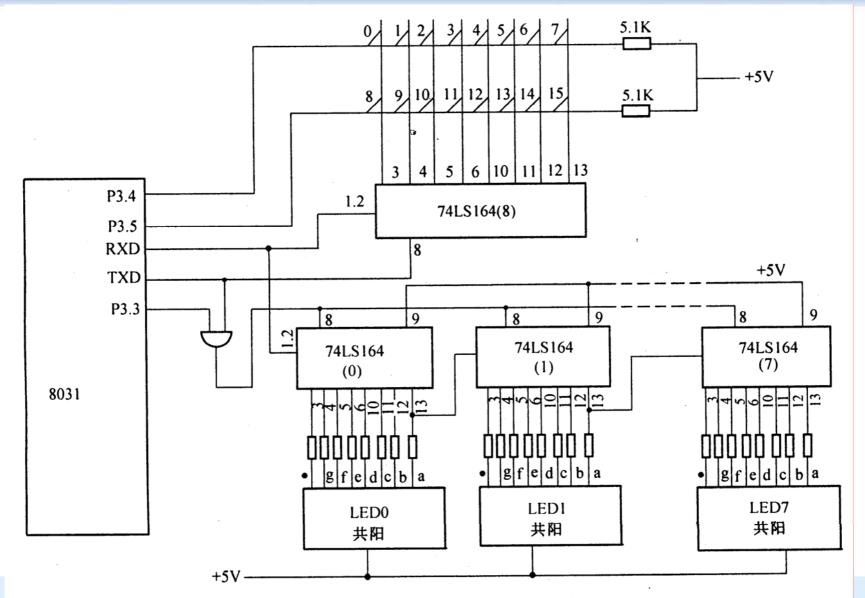
AJMP KEYI

键盘处理子程序

```
KND: RL A
     MOV R2,A
     AJMP LK4
  判断有无键按下子程序,若累加器A不为0,则表示有键按下
KS1:
     MOV DPTR,#7F01H
     MOV A,#00H
     MOVX @DPTR,A ;置所有列为低电平
     INC DPTR
     INC DPTR
                     ;读键盘行状态
     MOVX A,@DPTR
     CPL A
     ANL A,#0FH
     RET
```

12.3.2 串行口扩展键盘显示接口电路设计

例2 串行口控制键盘和显示电路



键盘及显示程序设计



显示程序设计

图中8个数码管的显示方式为静态显示, 此时显示程序可不必扫描数码管,程序相 对简单

键盘程序设计

- 判别键盘上有无键闭合
- 去除键盘的机械抖动
- 判别闭合键的键号

显示子程序

显示子程序,显示数据缓冲区位于78H~7FH,

主程序中设串行口工作于方式0输出

DIR: SETB P3.3

MOV R7,#08H

MOV R0,#7FH

DLO: MOV A,@R0

MOV DPTR,#TAB ;指向显示码首地址

MOVC A,@A+DPTR ;查表取出字符码

MOV SBUF,A

TI,DL1 DL1: **JNB**

> CLR TI

> **DEC** $\mathbf{R0}$

DJNZ R7,DL0

CLR P3.3

RET

;开放显示输出

;置显示缓冲区地址

;取要显示的数据

;送出显示

;输出完成了吗

;输出完成后,清中断标志

;指向下一个显示数据

;关闭显示输出

共阳极显示字符码表格, 段数据表格

TAB: 0C0H,0F9H,0A4H,0B0H,99H,92H DB

> DB 82H,0F8H,90H,88H,83H,0C6H

DB 0A1H,86H,8FH,8CH,0FFH

键盘扫描程序

;-----

键盘扫描子程序

·_____

KEYI: MOV A,#00H

MOV SBUF,A

;使键盘所有列为低电平

;是由抖动引起的吗?

;不是抖动,确定有键按下

:按列扫描键盘,判别键号

;扫描码,此时对第0列扫描

;有键按下吗?

;延迟

; 列号

KL0: JNB TI,KL0

CLR TI

KL1: JNB P3.4,PK1

JB **P3.5,KL1**

PK1: ACALL DL10

JNB P3.4,PK2

JB P3.5,KL1

PK2: MOV R7,#08H

MOV R6,#0FEH

MOV R3,#00H

MOV A,R6

KL5: MOV SBUF,A

KL2: JNB TI,KL2

CLR TI

JNB P3.4,PKONE

JB P3.5,NEXT

MOV R4,#08H

;第2行有键按下,该行首键号

键盘扫描程序

A.IMP PK3 ;第1行有键按下,该行首键号 **PKONE: MOV** R4,#00H **MOV** ;等待键释放 **PK3**: SBUF,#00H **KL3: JNB** TI,KL3 CLR TI P3.4,KL4 **KL4**: JNB P3.5,KL4 **JNB** ;键释放,计算键码 MOV A,R4;该行首键号+列号 **ADD** A,R3 **RET** MOV ;准备作下一列的扫描 **NEXT:** A,R6RL A MOV **R6,A** INC **R3** ; 列号 : 是否8列均扫描结束? DJNZ R7,KL5 AJMP KEYI 延迟10ms子程序 **DL10: R7,#0AH** MOV DL: **MOV** R6,#0FFH **DL6:** DJNZ **R6,DL6**

DJNZ

RET

R7,DL

12.4 MCS-51单片机与BCD码拨盘的接口设计

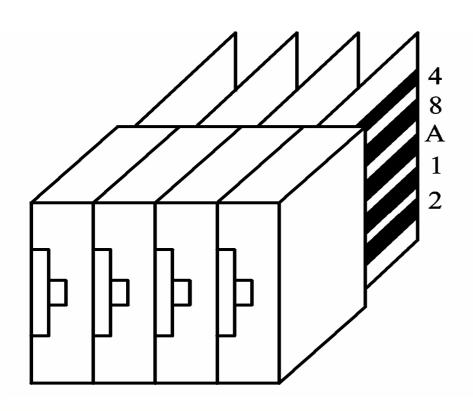
12.4.1 BCD码拨盘工作原理

有时需输入一些控制参数,设定完将维持不变。

使用的最方便的拨盘是十进制输入,BCD码输出的BCD码拨盘。

下图为四片BCD码拨盘拼接的4位十进制输入拨盘组。

每片拨盘具有0~9十个位置,每个位置都有相应的数字显示。



这部分内容作 为提高阶段的 学习

32.00

BCD码拨盘后面有5个接点,A为输入控制线,另外4是BCD码输出线。

拨盘拨到不同位置时,输入控制线A分别与4根BCD码输出线中的某根或某几根接通,其接通的BCD码输出线状态正好与拨盘指示的十进制数相一致。

BCD码拨盘的输入输出状态表如下所示。

BCD码拨盘的输入输出状态

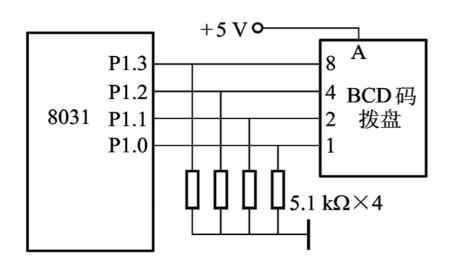
		输出状态			
拨盘输入	控制端A	8	4	2	1
0	1	0	0	0	0
1	1	0	0	0	1
2	1	0	0	1	0
3	1	0	0	1	1
4	1	0	1	0	0
5	1	0	1	0	1
6	1	0	1	1	0
7	1	0	1	1	1
8	1	1	0	0	0
9	1	1	0	0	1 40

12.4.2 BCD码拨盘与MCS-51的接口



1.与单片BCD码拨盘的接口

下图是8031通过P1.0~P1.3与单片BCD码拨盘的接口电路。

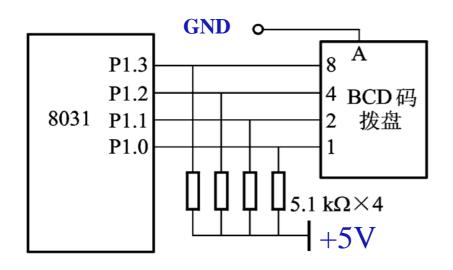


A端接+5V,当拨盘拨至某十进制数时,相应的8,4,2,1有效端输出高电平,无效端为低电平(如拨至"6"时,8 端输出"0",4,2,端为"1",1端输出"0")。也称正逻辑输出

A端接地,8,4,2,1输出端通过电阻上拉至高电平时,拨盘输出的BCD码为负逻辑(反码)输出。

BCD码盘控制端A接"0"时的输出





BCD码盘输入数字为9时,

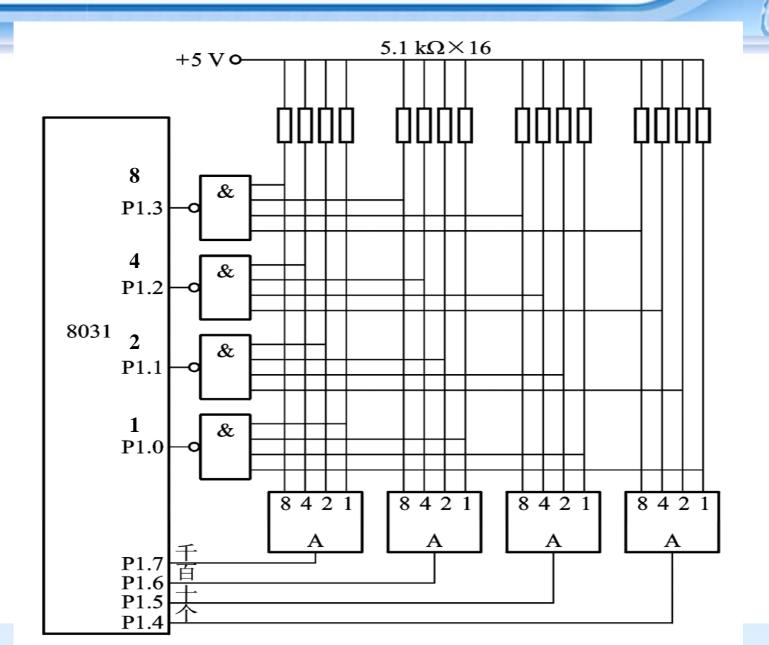
- "8"端输出0
- "4"端输出1
- "2"端输出1
- "1"端输出0

2. 多片BCD码拨盘与单片机的接口

如按上图,N位拨盘需占用4×N根1/0口线, 为减少1/0口线, 可将拨盘的输出线分别通过4个与非门与单片机的1/0口相连, 每片拨盘的控制端A不再接+5V或地, 而是分别与1/0口线相连, 用来控制选择多片拨盘中的任意一片。

这时, N位十进制拨盘, 用N片BCD码拨盘拼成时只需占用4+N根 I/O口线。下图通过P1与4片BCD码拨盘相连的4位BCD码输入电路。

多片BCD码拨盘与单片机的接口电路



4片拨盘的BCD码输出相同端接入同一个4个与非门。四个与非门输出8, 4, 2, 1端分别接入P1. 3, P1. 2, P1. 1, P1. 0。其余的P1. 6, P1. 5, P1. 4分别与千、百、十、个位BCD码拨盘的控制端相连。当选中某位时,该位的控制端置0,其它三个控制端置1。

每位BCD码输出端上有相应的端子与控制端A接通,当每一位的控制端A接1时,其BCD码输出端子为全1,当控制端A接0时,其对应数字的BCD端子输出0(即该数字BCD码的反码)。

19. A.

例如选中千位时,P1.7置0, P1.4~P1.6置1,此时四个与非门百、十、个位输入端均为1状态,因此四个与非门输出的状态完全取决于千位数BCD拨盘输出状态。由于该位的控制端置0,因此,拨盘所置之数输出为BCD反码,通过与非门输出为该千位数的BCD码。

例如输入为9345

当P1.7扫描输出0时,P1.3-P1.0状态为1001当P1.6扫描输出0时,P1.3-P1.0状态为0011当P1.5扫描输出0时,P1.3-P1.0状态为0100当P1.4扫描输出0时,P1.3-P1.0状态为0101

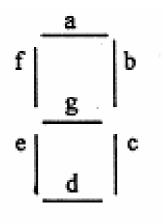
练习



编制一个显示子程序

要求: 待显示量存放在R1中, 得到的显示 码存放在R2中。

数码管段与数据位关系,显示为低电平驱动(数码管为共阳极)



g	f	е	d	С	b	a	dp
D7	D6	D 5	D4	D3	D2	D1	DO

答案

```
72.40
```

```
Push
      a
Push
      dph
Push
      dpl
Mov
     a,r1
Mov
     dptr,#tab
Movc a,@a+dptr
Mov
     r2,a
      dpl
Pop
      dph
Pop
Pop
      a
Ret
Tab:db
       81h,0f3h,49h,61h,33h,25h,05h,0f1h
    db 01h,21h,11h,07h,8dh,43h,0dh,1dh
```

END

