

51 单片机汇编语言教程：24 课动态扫描显示接口电路及程序

(基于 HJ-1G、HJ-3G 实验板)

在单片机系统中动态扫描显示接口是单片机中应用最为广泛的一种显示方式之一。其接口电路是把所有显示器的 8 个笔划段 a-h 同名端连在一起，而每一个显示器的公共极 COM 是各自独立地受 I/O 线控制。CPU 向字段输出口送出字形码时，所有显示器接收到相同的字形码，但究竟是那个显示器亮，则取决于 COM 端，而这一端是由 I/O 控制的，所以我们就能自行决定何时显示哪一位了。而所谓动态扫描就是指我们采用分时的办法，轮流控制各个显示器的 COM 端，使各个显示器轮流点亮。

在轮流点亮扫描过程中，每位显示器的点亮时间是极为短暂的（约 1ms），但由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位显示器并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感。

下图所示就是我们的单片机实验板上的动态扫描接口。由 89c51 的 P0 口能灌入较大的电流，所以我们采用共阳的数码管，并且不用限流电阻，而只是用两只 1N4004 进行降压后给数码管供电，这里仅用了两只，实际上还能扩充。它们的公共端则由 PNP 型三极管 8550 控制，显然，如果 8550 导通，则对应的数码管就能亮，而如果 8550 截止，则对应的数码管就不可能亮，8550 是由 P2.7，P2.6 控制的。这样我们就能通过控制 P27、P26 达到控制某个数码管亮或灭的目的。

下面的这个单片机程序，就是用实验板上的数码管显示 0 和 1。

```
FIRST EQU P2.7 ;第一位数码管的位控制
SECOND EQU P2.6 ;第二位数码管的位控制
DISPBUFF EQU 5AH ;显示缓冲区为5AH 和5BH

ORG 0000H
AJMP START

ORG 30H
START:
MOV SP, #5FH ;设置堆栈
MOV P1, #0FFH
MOV P0, #0FFH
MOV P2, #0FFH ;初始化，所显示器，LED 灭
MOV DISPBUFF, #0 ;第一位显示0
MOV DISPBUFF+1, #1 ;第二握显示1
LOOP:
```

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

```
LCALL DISP ;调用显示程序
AJMP LOOP
;主程序到此结束
DISP:
PUSH ACC ;ACC 入栈
PUSH PSW ;PSW 入栈
MOV A, DISPBUFF ;取第一个待显示数
MOV DPTR, #DISPTAB ;字形表首地址
MOVC A, @A+DPTR ;取字形码
MOV P0, A ;将字形码送 P0位 (段口)
CLR FIRST ;开第一位显示器位口
LCALL DELAY ;延时1毫秒
SETB FIRST ;关闭第一位显示器 (开始准备第二位的数据)
MOV A, DISPBUFF+1 ;取显示缓冲区的第二位
MOV DPTR, #DISPTAB
MOVC A, @A+DPTR
MOV P0, A ;将第二个字形码送 P0口
CLR SECOND ;开第二位显示器
LCALL DELAY ;延时
SETB SECOND ;关第二位显示
POP PSW
POP ACC
RET
DELAY: ;延时1毫秒
PUSH PSW
SETB RS0
MOV R7, #50
D1: MOV R6, #10
D2: DJNZ R6, $
DJNZ R7, D1
```

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

```
POP PSW
```

```
RET
```

```
DISPTAB:DB 28H, 7EH, 0a4H, 64H, 72H, 61H, 21H, 7CH, 20H, 60H
```

```
END
```

从上面的单片机例程中能看出，动态扫描显示必须由 CPU 持续地调用显示程序，才能保证持续持续的显示。

上面的这个程序能实现数字的显示，但不太实用，为什么呢？这里仅是显示两个数字，并没有做其他的工作，因此，两个数码管轮流显示1毫秒，没有问题，实际的工作中，当然不可能只显示两个数字，还是要做其他的事情的，这样在二次调用显示程序之间的时间间隔就不一不定了，如果时间间隔比较长，就会使显示不连续。而实际工作中是很难保证所有工作都能在很短时间内完成的。况且这个显示程序也有点“浪费”，每个数码管显示都要占用1个毫秒的时间，这在很多合是不允许的，怎么办呢？我们能借助于定时器，定时时间一到，产生中断，点亮一个数码管，然后马上返回，这个数码管就会一直亮到下一次定时时间到，而不用调用延时程序了，这段时间能留给主程序干其他的事。到下一次定时时间到则显示下一个数码管，这样就很少浪费了。

```
Counter EQU 59H ;计数器，显示程序通过它得知现正显示哪个数码管
```

```
FIRST EQU P2.7 ;第一位数码管的位控制
```

```
SECOND EQU P2.6 ;第二位数码管的位控制
```

```
DISPBUFF EQU 5AH ;显示缓冲区为5AH和5BH
```

```
ORG 0000H
```

```
AJMP START
```

```
ORG 000BH ;定时器 T0 的入口
```

```
AJMP DISP ;显示程序
```

```
ORG 30H
```

```
START:
```

```
MOV SP, #5FH ;设置堆栈
```

```
MOV P1, #0FFH
```

```
MOV P0, #0FFH
```

```
MOV P2, #0FFH ;初始化，所显示器，LED 灭
```

```
MOV TMOD, #00000001B ;定时器 T0 工作于模式1（16位定时/计数模式）
```

推荐使用慧净 51 实验板。推荐 51 学习网 WWW.HLMCU.COM 淘宝网: <http://shop37031453.taobao.com/>

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

```
MOV TH0, #HIGH(65536-2000)
MOV TL0, #LOW(65536-2000)
SETB TR0
SETB EA
SETB ET0
MOV Counter, #0 ;计数器初始化
MOV DISPBUFF, #0 ;第一位始终显示0
MOV A, #0
LOOP:
MOV DISPBUFF+1, A ;第二位轮流显示0-9
INC A
LCALL DELAY
CJNE A, #10, LOOP
MOV A, #0
AJMP LOOP ;在此中间能安排任意程序，这里仅作示范。
;主程序到此结束
DISP: ;定时器 T0的中断响应程序
PUSH ACC ;ACC 入栈
PUSH PSW ;PSW 入栈
MOV TH0, #HIGH(65536-2000) ;定时时间为2000个周期，约2170微秒（11.0592M）
MOV TL0, #LOW(65536-2000)
SETB FIRST
SETB SECOND ;关显示
MOV A, #DISPBUFF ;显示缓冲区首地址
ADD A, Counter
MOV R0, A
MOV A, @R0 ;根据计数器的值取对应的显示缓冲区的值
MOV DPTR, #DISPTAB ;字形表首地址
MOVC A, @A+DPTR ;取字形码
MOV P0, A ;将字形码送 P0位（段口）
```

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

```
MOV A, Counter ;取计数器的值
JZ DISPFIRST ;如果是0则显示第一位
CLR SECOND ;不然显示第二位
AJMP DISPNEXT
DISPFIRST:
CLR FIRST ;显示第一位
DISPNEXT:
INC Counter ;计数器加1
MOV A, Counter
DEC A ;如果计数器计到2, 则让它回0
DEC A
JZ RSTCOUNT
AJMP DISPEXIT
RSTCOUNT:
MOV Counter, #0 ;计数器的值只能是0或1
DISPEXIT:
POP PSW
POP ACC
RETI
DELAY: ;延时130毫秒
PUSH PSW
SETB RS0
MOV R7, #255
D1: MOV R6, #255
D2: NOP
NOP
NOP
NOP
DJNZ R6, D2
DJNZ R7, D1
```

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

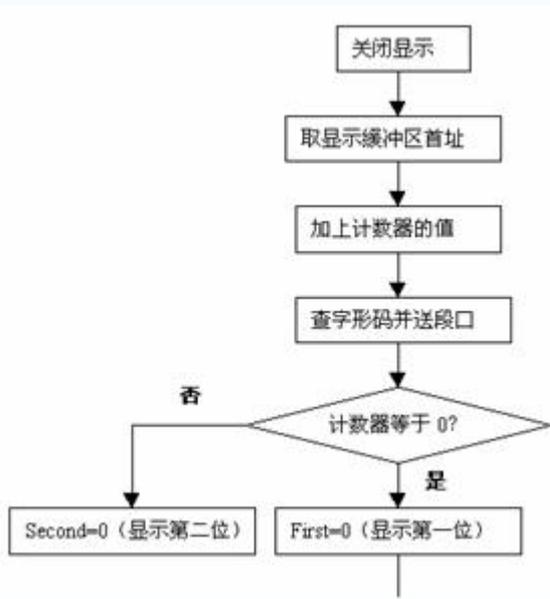
POP PSW

RET

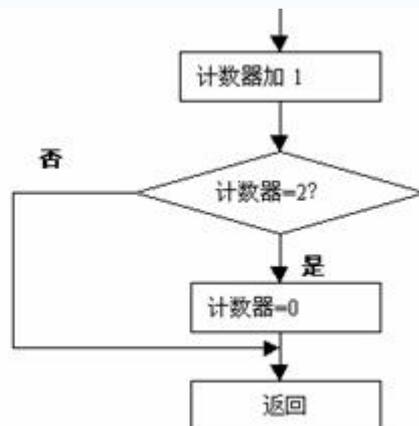
DISPTAB:DB 28H, 7EH, 0a4H, 64H, 72H, 61H, 21H, 7CH, 20H, 60H

END

从上面的单片机程序能看出，动态显示和静态显示相比，程序稍有点复杂，不过，这是值得的。这个程序有一定的通用性，只要改变端口的值及计数器的值就能显示更多位数了。下面给出显示程序的流程图。



<动态扫描程序框图>



[51 单片机汇编语言教程-慧净电子会员收集整理 \(全部 28 课\)](#)

[51 实验板推荐\(点击下面的图片可以进入下载资料链接\)](#)

