

# 单片机是怎样在液晶上显示字符的

· 吴景逖 ·

## 液晶 (LCD) 是如何显示的

**1. 线段的显示** 点阵图形式液晶由 M 行×N 列个显示单元组成, 假设 LCD 显示屏有 64 行, 每行有 128 列, 每 8 列对应 1 个字节的 8 个位, 即每行由 16 字节, 共  $16 \times 8 = 128$  个点组成, 屏上  $64 \times 16$  个显示单元和显示 RAM 区 1024 个字节相对应, 每一字节的内容和屏上相应位置的亮暗对应。例如屏的第一行的亮暗由 RAM 区的 000H~00FH 的 16 个字节的内容决定, 当 (000)=FFH 时, 则屏的左上角显示一条短亮线, 长度为 8 个点; 当 (3FFH)=FFH 时, 则屏的右下角显示一短亮线; 当 (000H)=FFH, (001H)=00H, (002H)=FFH, (003H)=00H, ... (00EH)=FFH, (00FH)=00H 时, 则在屏的顶部显示一条由 8 段亮线和 8 条暗线组成的虚线。这就是 LCD 显示的基本意思。

**2. 字符的显示** 当用 LCD 显示一个字符时就较复杂了, 因为一个字符由  $6 \times 8$  或  $8 \times 8$  点阵组成, 即要找到和屏上某几个位置对应的显示 RAM 区的 8 个字节, 并且要使每个字节的不同的位为 '1', 其它的为 '0', 为 '1' 的点亮, 为 '0' 的点暗, 这样一来就组成某个字符。但对于内带字符发生器的控制器 (如 T6963C) 来说, 显示字符就比较简单了, 可让控制器工作在文本方式, 根据在 LCD 上开始显示的行列号及每行的列数找出显示 RAM 对应的地址, 设立光标, 在此送上该字符对应的代码即可。

**3. 汉字的显示** 汉字的显示一般采用图形方式, 事先从微机中提取要显示的汉字的点阵码, 每个汉字占 32 字节, 分左右两半部, 各占 16 字节, 左边为 1、3、5..., 右边为 2、4、6..., 根据在 LCD 上开始显示的行列号及每行的列数可找出显示 RAM 对应的地址, 设立光标, 送上要显示的汉字的第一个字节, 光标位置加 1, 送第二字节, 换行按列对齐, 送第三字节...直到 32 字节显示完就可在 LCD 上得到一个完整的汉字。

## 内带 T6963C 的液晶图形显示模块和 8031 单片机的连接

**1. LCD 显示控制器 T6963C** T6963C 是点阵式液晶图形显示控制器, 能直接和 8031 单片机连接 (见附图); 可以以图形、字符方式或合成显示; 内部有字符发生器, 共 128 个字符, 允许 CPU 随时访问显示 RAM, 并可进行位操作。

**2. 内带 T6963C 的液晶图形显示模块 (MGLS12864T) 和 8031 单片机的连接** 内带 T6963C 的液晶图形显示模块 MGLS12864T 和 8031 单片机的连接见附图。由图可以看出, MGLS12864T 的数据线和 8031 的数据总线相接, 它的 C/D 接 8031 的地址线 A0, /CE

接 8031 外扩系统地址译码器 74LS138 的一个输出端; MGLS12864T 的 /WR, /RD 分别和 8031 的 /WR, /RD 相接, /RESET, VCC, /HALT 接+5V, VO 接-5V~-15V 可调, VEE 接-5V~-15V; FG、GND 接 8031 的 GND。T6963C 的主要指令见表 1。

	参数 1	参数 2	指令代码	功 能
1.	水平位置	垂直位置	21H	光标指针设置
2.	地址 (低 5 位)	00H	22H	CGRAM 偏置地址设置
3.	低字节	高字节	24H	地址指针位置
4.	低字节	高字节	40H	文本区首址
5.	低字节	00H	41H	文本区宽度 (字节数/行)
6.	低字节	高字节	42H	图形区首址
7.	低字节	00H	43H	图形区宽度 (字节数/行)
8.	无	无	8XH	显示方式设置
9.	无	无	9XH	光标闪烁, 显示, 文本, 图形显示
10.	无	无	AXH	光标形状选择
11.	无	无	BXH	数据自动读/写方式设置
12.	D1 (仅写才有)	无	C0—C5H	数据一次写/读, 地址加 1 或不变
13.	无	无	D0H	屏读
14.	无	无	D8H	屏拷贝
15.	无	无	FXH	位操作

对 T6963C 进行操作前应先读状态字, T6963C 的状态字中的各位含意如表 2。

表 2

位 状态	1	0
STA0: 指令读写状态	准备好	忙
STA1: 数据读写状态	准备好	忙
STA2: 数据自动读状态	准备好	忙
STA3: 数据自动写状态	准备好	忙
STA4: 未用		
STA5: 控制器运行检测可能性	可能	不能
STA6: 屏读/拷贝出错状态	出错	正确
STA7: 闪烁状态检测	正常显示	关显示

### 3. 程序举例 (仅供参考)

#### (1) 清显示 RAM 子程序

```

MOV R2, #00H
MOV R3, #00H
MOV R4, #24H ; 设地址指针
CALL PR1
MOV R4, #0B0H ; 自动写指令
CALL PR12
MOV R2, #1FH ; 共 8K
PR31:

```

```

MOV R3, #0FFH
PR32:
CALL PR03
CLR A
CALL PR14
DJNZ R3, PR32
DJNZ R2, PR31
MOV R4, #0B2H ; 自动写结束
CALL PR12
RET

```

## (2) 检测状态:

```

PR01: ; 判 LCD 忙否?
MOV DPTR, #4001H ; LCD 指令口地址
MOVX A, @DPTR
JNB ACC. 0, PR01
JNB ACC. 1, PR01
RET

```

```

PR03: ; 判 LCD 自动写标志
MOV DPTR, #4001H
MOVX A, @DPTR
JNB ACC. 3, PR03
RET

```

## (3) 写数据或指令程序

```

PR1: ; 双字节参数指令入口
CALL PR01
MOVA, R2
CALL PR14
PR11: ; 单字节参数指令入口
CALL PR01
MOVA, R3
CALL PR14
PR12: ; 无字节参数指令入口
CALL PR01
MOVA, R4
JMP PR15

```

```

PR14: ; 写数据入口:
MOV DPTR, #4000H ; LCD 数据口地址
PR15:
MOVX @DPTR, A
RET

```

## (4) 显示汉字子程序:

```

DIS_HZ:

```

```

MOV A, R6 ; (R6) :LCD 上开始显示的行数
MOV B, A
MOV A, #10H ; 每个汉字占 16 行*8 列像素
MUL AB ; 根据 R6, R7 找显示 RAM 地址
    MOV R2, A
    MOV A, B
    MOV R3, A
    MOV A, R2
    CLR C
ADD A, R7 ; (R7) :LCD 上开始显示的列数
MOV R2, A
    MOV A, R3
    MOV R3, A
    ADDC A, #08H ; 设 0800H 为图形区首址
    MOV R7, #00H
HZ_1:
    MOV R4, #24H ; 显示 RAM 区地址指针设置命令
    MOV A, R3
    MOV R6, A
    CALL PR1
    MOV DPL, 32H ; (32H) 要显示的汉字的首地址低字节
    MOV DPH, 33H ; (33H) 要显示的汉字的首地址高字节
    MOV A, R7
    MOVC A, @A+DPTR
    MOV R3, A
    MOV R4, #0C0H ; 写一字节
    CALL PR11
    MOV A, R6
    MOV R3, A
    INC R7
    MOV A, R2
    ADD A, #01H
    MOV R2, A
    MOV A, R3
    ADDC A, #00H
    MOV R3, A
    MOV R4, #24H
    MOV A, R3
    MOV R6, A
    CALL PR1
    MOV DPL, 32H
    MOV DPH, 33H
    MOV A, R7
    MOVC A, @A+DPTR

```

```

MOV    R3, A
MOV    R4, #0C0H    ; 写下一字节
CALL   PR11
MOV    A, R6
MOV    R3, A
INC    R7
MOV    A, R2
ADD    A, #15 ; 汉字第二行的左半部
MOV    R2, A
MOV    A, R3
ADDC   A, #00H
MOV    R3, A
CJNE  R7, #20H, HZ_1 ; 共 32 字节
RET

DIS_ZF:           ; 显示一个字符 子程序
MOV A, R6         ; (R6): LCD 上开始显示的行数
MOVB, #10H       ; 设每行 16 列
MULAB
ADD    A, R7      ; (R7): LCD 上开始显示的列数
MOV    R2, A
MOV    A, B
MOV    R3, A
MOV    R4, #24H   ; 显示 RAM 区地址指针设置命令
CALL   PR1
MOV    A, R5      ; (R5): 控制器内部字符的代码
MOV    R3, A
MOV    R4, #0C0H
CALL   PR11
RET

```