

**内藏 T6963C 控制器
点阵图形液晶显示模块
使用手册**

目 录

前 言	2
注意事项	2
第一章：T6963C 一般介绍	4
第二章：6963C 的引脚说明及其功能	4
一、 T6963C 外形图	4
二、 T6963C 引脚说明	5
第三章：T6963C 指令集	6
第四章：内藏 T6963C 控制器的液晶显示器的外特性	10
第五章、内藏 T6963C 的液晶显示模块与 MPU 的接口方法	12
第六章、T6963C 控制器液晶模块应用程序示例	16

前 言

T6963C液晶显示控制器多用于小规模液晶显示器件，常被装配在图形液晶显示模块上，以内藏控制器型图形液晶显示模块的形式出现。

注意事项

十分感谢您购买我公司的产品。在使用前请您首先仔细阅读以下注意事项，以免给您造成不必要的损失。您在使用过程中遇到困难时，请拨打我公司电话，我们将尽力为您提供服务和帮助。

一、处理保护膜

在装好的模块成品表面贴有一层保护膜，以防在装配时沾污显示表面。在整机装配结束前不得揭去，以免弄脏或损坏显示面。

二、加装衬垫

在模块与前面板之间最好加装一块约0.1 毫米左右的衬垫。面板应保持平整，以免在装配后产生扭曲，并可提高其抗振性能。

三、严防静电

模块中的控制、驱动电压是低压、微功耗的CMOS 电路，极易被静电击穿。静电击穿是一种不可修复的损坏，而人体有时会产生高达几十伏或上百伏的高压静电，所以在操作、装配以及使用中都应极其小心，严防静电。为此：

1. 不要用手随意去摸外引线、电路板上的电路及金属框；
2. 如必须直接接触时，应使人体与模块保持在同一电位或将人体良好接地；
3. 焊接使用的烙铁和操作用的电动工具，必须良好接地，没有漏电；
4. 不得使用真空吸尘器进行清洁处理，因为它会产生很强的静电；
5. 空气干燥也会产生静电，因此工作间湿度应在RH60%以上；
6. 取出或放回包装袋或移动位置时，也需小心防止产生静电，不要随意更换包装或舍弃原包。

四、装配操作时的注意事项

1. 模块是经过精心设计组装而成的，请勿随意自行加工修整；
2. 金属框爪不得随意扭动、拆卸；
3. 不要随意修改加工PCB 板外形、装配孔、线路极其部件；
4. 不得修改导电胶条；
5. 不得修改任何内部支架；
6. 不要碰、摔、折曲、扭动模块。

五、焊接

在焊接模块外引线接口电路时，应按如下规程进行操作：

1. 烙铁头温度小于 280°C ；
2. 焊接时间小于 $3\sim 4\text{s}$ ；
3. 焊接材料：共晶型、低熔点；
4. 不要使用酸性助焊剂；
5. 重复焊接不要超过3次，且每次重复需间隔5分钟。

六、模块的使用与保养

1. 模块的外引线决不允许接错，在您想调试液晶模块时，请注意正确接线，尤其是正负电源的接线不能有错，否则可能造成过流、过压烧电路上的芯片等，对液晶模块元器件有损的现象。

2. 模块在使用时接入电源及断开电源，必须在正电源稳定接入以后才能输入信号电平。如在电源稳定前或断开后输入信号电平，有可能损坏模块中的IC及电路。

3. 点阵液晶模块显示时的对比度视角与温度驱动电压关系很大，所以如果VEE调整过高，不仅会影响显示还会缩短模块的使用寿命。

4. 因为液晶材料的物理特性，液晶的对比度会随着温度的变化而相应变化，所以您加的负压值，应该随温度作相应的调整。大致是温度变化 10°C 电压变化1伏。为满足这一要求您可做一个温度补偿电路或者安排一个电位器，随温度变化，调整负电压值。

5. 不应在规定工作温度范围以外使用。并且不应在超过存储极限温度的范围外存储。如果温度低于结晶温度，液晶就会结晶。如果温度过高，液晶将变成各向同性的液晶，破坏分子取向。使器件报废。

6. 用力按压显示部分会产生异常显示，这时切断电源稍待片刻，重新上电即恢复正常。

7. 液晶显示器件或模块表面结雾时，不要通电工作。因为这将引起电极化学反应，产生断线。

8. 长期用于阳光及强光下时，被遮部分会产生残留影像。

七、模块的存储

若长期，如几年以上存储，我们推荐以下方式：

1. 装入聚乙烯口袋，最好有防静电涂层，并将口封住；
2. 在 $-10^{\circ}\text{C}\sim +35^{\circ}\text{C}$ 之间存储；
3. 放暗处，避强光；
4. 决不能在表面压放任何物品；
5. 严格避免在极限温/湿度条件下存放。

八、责任范围及维修

在您购买液晶显示模块时我公司将会为您做显示模块的检测，确保您所买的显示模块为完好的器件。在您使用过程中，因不小心将显示模块损坏，您可送至我公司维修液晶模块。如果出现屏的问题，比如玻璃面破损，玻璃屏角碎裂等等，将无法进行修理您的液晶模块。只能更换玻璃。

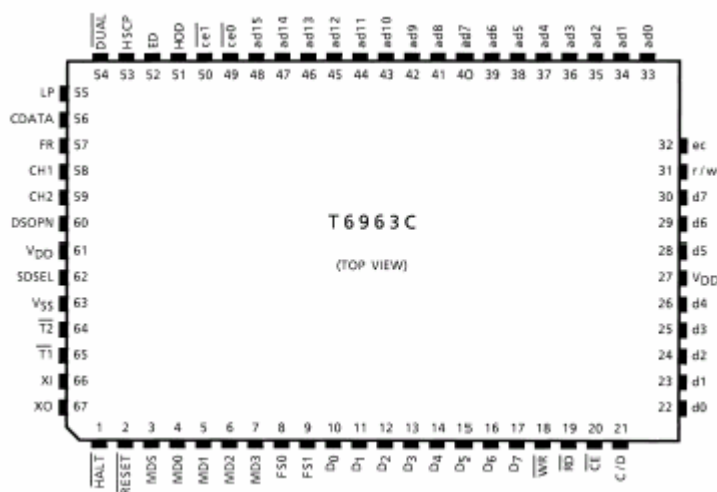
公司产品质量都是通过严格检测及时间考验的，请用户放心使用。如果在使用中发现问题，可以拿到我公司进行检测，确实是质量问题，而且购买时间在一年之内，我公司给予换货处理，但是因为用户使用不当，例如静电、焊接连线不当、过流、过压使用等所造成的损坏，可送到我们维修部修理。如是邦定IC烧毁，则无法维修，只能更换板子。

第一章、T6963C 一般介绍

- 1、与 80 系列 8 位微处理器直接接口。
- 2、内部具有 128 个字符的 ROM 字符发生器。
- 3、6963C 物字体由硬件设置，其字体有四种：5×8、6×8、7×8、8×8。
- 4、可对 8K Byte 的显示 RAM 内存操作
- 5、字符与图形可同时显示，可以选择"OR","AND","EXOR"方式
- 6、6963C 的占空比可从 1 / 16 到 1 / 128。

第二章、6963C 的引脚说明及其功能

一、T6963C 外形图



二、T6963C 引脚说明

T6963C 的 QFD 封装共有 67 个引脚，各引脚说明如下：

- 1、D0 - D7：T6963C 与 MPU 接口的数据总线，三态。

2、RD, WR : 读 . 写选通信号, 低电平有效, 输入信号。

3、CE : T6963C 的片选信号, 低电平有效。

4、C / D : 通道选择信号, 1 为指令通道, 0 为数据通道。

5、RESET, HALT : RESET 为低电平有效的复位信号, 它将行、列计数器和显示寄存器清零, 关显示; HALT 具有 RESET 的基本功能, 还将中止内部时钟振荡器的工作, 以保护液晶显示器件。

以上引脚为 T6963C 与 MPU 接口的引脚, 时序关系符合 8080 系列时序。

6、UAL, SDSEL :

DUAL = 1 为单屏结构, DUAL = 0 为双屏结构;

SDSEL = 0 为一位串行数据传输方式, SDSEL = 1 为 2 位并行数据传输方式。

7、MD2, MD3 : 设置显示窗口长度, 从而确定了列数据传输个数的最大值, 其组合逻辑关系如下:

MD3	1	1	0	0
MD2	1	0	1	0
每行字符数	32	40	64	80

8、MDS, MD1, MD0 : 设置显示窗口宽度 (行), 从而确定 T6963C 的帧扫信号的时序和显示驱动的占比系数, 当 DUAL = 1 时, 其组合功能如下:

MDS	0	0	0	0	1	1	1	1
MD1	1	1	0	0	1	1	0	0
MD0	1	0	1	0	1	0	1	0
字符行	2	4	6	8	10	12	14	16
总行数	16	32	48	64	80	96	112	128
占空比	1/16	1/32	1/48	1/64	1/80	1/96	1/112	1/128

当 DUAL = 0 时, 以上设置中的字符行和总行数增至原来的 2 倍, 其它都不变, 这种情况下的液晶屏结构为双屏结构。

9、S1, FS0 : 显示字符的字体选择。

FS1	1	1	0	0
FS0	1	0	1	0
字体	5×8	6×8	7×8	8×8

10、X1, X0 : 振荡时钟引脚。

11、AD0 - 15 : 输出信号, 显示缓冲区 16 位地址总路线。

12、D0 - 7 : 三态, 显示缓冲区 8 位数据总路线。

13、R / W : 输出, 显示缓冲区读 / 写控制信号。

14、CE : 输出, 显示缓冲区片选信号, 低电平有效。

15、CE0, CE1 : 输出, DUAL = 1 时的存储器片选信号。

16、T1, T2, CH1, CH2 : 用来检测 T6963C 工作使用情况, T1, T2 作为测试信号输入端, CH1,

CH2 作为输出端。

17、HOD , HSCP , LOD LSCP (CE1) , ED LP , CDATA , FR 为 T6963C 驱动部分信号 , 我们可以不作太多了解。

第三章、T6963C 指令集

T6963C 的初始化设置一般都由硬件作为设置 , 因此其指令系统将集中于显示功能的设置上。T6963C 的指令可有一个或两个参数 , 或无参数。每条指令的执行都是先送入参数 (如果有的话) , 再送入指令代码。每次操作之前最好先进行状态字检测。

一、T6963C 的状态字如下所示 :

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

由于状态位作用不一样 , 因此执行不同指令必须检测不同状态位。在 MPU 一次读 / 写指令和数据时 , STA0 和 STA1 要同时有效 (“准备好”状态)。当 MPU 读 / 写数据时 , 判断 STA2 或 STA3 状态。屏读、屏拷贝指令使用 STA6。STA5 和 STA7 反映 T6963C 内部运行状态。

二、T6963C 指令系统的说明如下 :

1、设置指令 , 格式如下 :

D1,D2	0	0	1	0	0	N2	N1	N0
-------	---	---	---	---	---	----	----	----

D1,D2 为第一和第二个参数,后一个字节为指令代码,根据 N0,N1,N2 的取值,该指令有三个含义 (N0,N1,N2 不能有两个同时为 1)

D1	D2	指令代码	功能
水平位置 (有效位七位)	垂直位置 (有效 5 位)	21H(N0=1)	光标指针设置
地址 (有效位 5 位)	00H	22H(N1=1)	CGRAM 偏置地址设置
低字节	高字节	24H(N2=1)	地址指针位置

N2 - N0：合成显示方式控制位，其组合功能如下表：

N2	N1	N0	合成方式
0	0	0	逻辑“或”合成
0	0	1	逻辑“异或”合成
0	1	1	逻辑“与”合成
1	0	0	文本特征

当设置文本方式和图形方式均打开时，上述合成显示方式设置才有效。其中的文本特征方式是指将图形区改为文本特征区，该区大小与文本区相同，每个字节作为对应的文本区的每个字符显示的特征，包括字符显示与不显示、字符闪烁及字符的“负向”显示。通过这种字符方式，T6963C 可以控制每个字符的文本特征。文本特征区内，字符的文本特征码由一个字节的低四位组成，即：

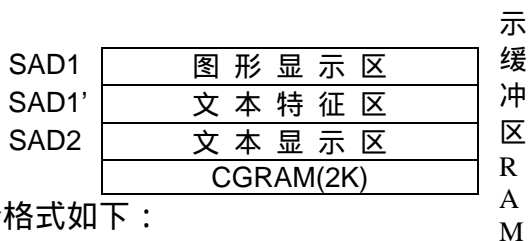
D7	D6	D5	D4	D3	D2	D1	D0
*	*	*	*	d3	d2	d1	d0

d3：闪烁控制位，d3 = 1 为闪烁，d3 = 0 为不闪烁，

d2 - d0 的组合如下：

d3	d3	d3	显示效果
0	0	0	正向显示
1	0	1	负向显示
0	1	1	禁止显示，空白

启用文本特征方式时，可在原有图形区和文本区外，用图形区域设置指令，另开一区作为文本特征区，以保持原图形区的数据。显示缓冲区可划分如下（单屏结构）：



4、显示开关，指令格式如下：

无参数	1	0	0	1	N3	N2	N1	N0
-----	---	---	---	---	----	----	----	----

N0：1/0，光标闪烁启用/禁止

N1：1/0，光标显示启用 / 禁止

N2：1/0，文本显示启用 / 禁止

N3：1/0，图形显示启用 / 禁止

5、光标形状选择，指令格式如下：

无参数	1	0	1	0	0	N2	N1	N0
-----	---	---	---	---	---	----	----	----

光标形状为 8 点（列）×N 行，N 的值为 0 - 7H，由 N2-N0 确定。

6、数据自动读 / 写方式设置：

无参数	1	0	1	1	0	0	N1	N0
-----	---	---	---	---	---	---	----	----

该指令执行后，MPU 可以连续地读 / 写显示缓冲区 RAM 的数据，每读 / 写一次，地址指针自动增 1。自动读 / 写结束后，必须写入自动结束命令以使 T6963C 退出自动读 / 写状态，开始接受其它指令。

N1, N0 组合功能如下：

N1	N0	指令代码	功 能
0	0	B0H	自动写设置
0	1	B1H	自动读设置
1	*	B2H / B3H	自动读 / 写结束

7、数据一次读 / 写方式，指令格式如下：

D1	1	1	0	0	0	N2	N1	N0
D1		N2	N1	N0	指令代码	功 能		
数据		0	0	0	C0H	数据写，地址加 1		
		0	0	1	C1H	数据读，地址加 1		
数据		0	1	0	C2H	数据写，地址减 1		
		0	1	1	C3H	数据读，地址减 1		
数据		1	0	0	C4H	数据写，地址不变		
		1	0	1	C5H	数据读，地址不变		

8、屏读，指令格式为：

无参数	1	1	1	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

该指令将当前由地址指针指向的某一位置上的显示状态（8×1 点阵）作为一个字节的数据送到 T6963C 的数据栈内，等待 MPU 的读取，该数据是文本数据与图形数据在该位置上的逻辑合成值。地址指针应在图形区内设置。

9、屏拷贝，指令代码为：

无参数	1	1	1	0	1	0	0	0
-----	---	---	---	---	---	---	---	---

该指令将当前地址指针（图形区内）指向的位置开始的一行显示状态拷贝到相对应的图形显示区的一组单元内，该指令不能用于文本特征方式或双屏结构液晶显示屏的应用上。

10、位操作：

无参数	1	1	1	1	N3	N2	N2	N0
-----	---	---	---	---	----	----	----	----

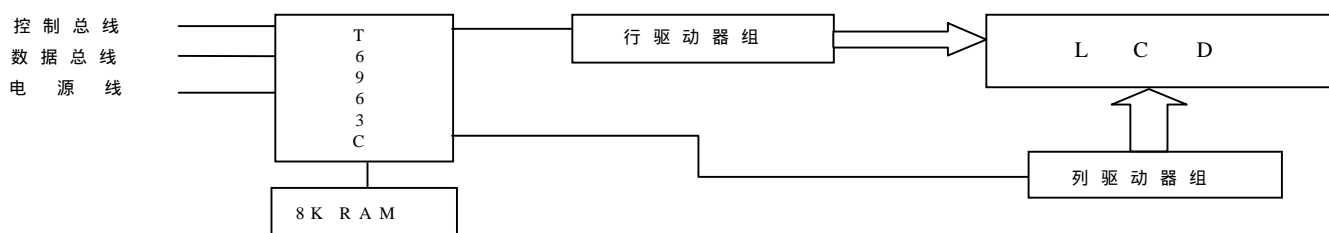
该指令可将显示缓冲区某单元的某一位清零或置 1，该单元地址由当前地址指针提供。N3 = 1 置

1, N3 = 0 清零。N2 - N0：操作位对应单元的 D0 - D7 位。

至此，T6963C 的指令系统全部讲述完毕。

第四章、内藏 T6963C 控制器的液晶显示器的外特性

内藏 T6963C 的液晶显示器已经实现了 T6963C 与行、列驱动器及显示缓冲区 RAM 的接口，同时也已用硬件设置了液晶屏的结构（单 / 双屏），数据传输方式，显示窗口长度、宽度等等。我们常用的液晶显示模块一般都是单屏结构，因此我们这里只讨论单屏结构的液晶显示模块。内藏 T6963C 的单屏结构点阵图形液晶显示模块的方框图如下：



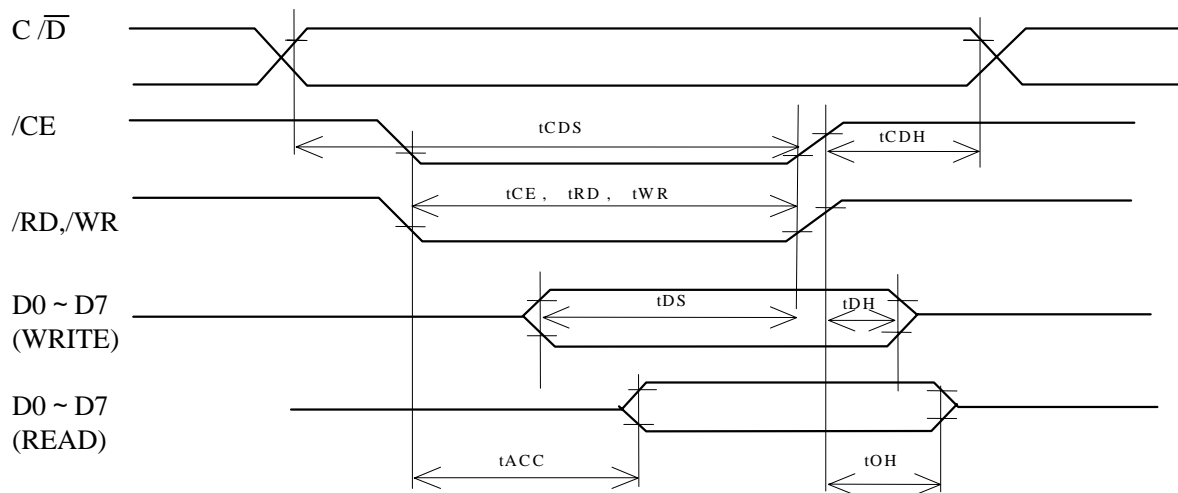
电特性：绝对最大范围

名称	符号	条件	范围	单位
电源电压	VDD	Ta=25°C	-0.3 ~ 0.7	V
输入电压	VIN	Ta=25°C	-0.3 ~ VDD+0.3	V
工作温度	Topr		-10 ~ +70	°C
存储温度	Tstg		-55 ~ +125	°C

电特性：电气参数

名称	符号	条件	MIN	TYP	MAX	单位
工作电压	VDD		+4.5	+5.0	+5.5	V
“H”输入电压	VIN		VDD-2.2	-	VDD	V
“L”输入电压	VIL		0	-	0.8	V
“H”输出电压	VOH		VDD-0.3	-	VDD	V
“L”输出电压	VOL		0	-	0.3	V
“H”输出电压	ROH	VOUT=VDD-0.5	-	-	400	Ω
“L”输出电压	ROL	VOUT=0.5V	-	-	400	Ω
输入上拉电阻	RPU		50	100	200	KΩ
工作频率	fosc		0.4	-	5.5	MHz
工作时电流损耗	IDD(1)	VDD=5.0V Fosc=3.0MHz	-	3.3	6.0	mA
暂停时电流损耗	IDD(2)	VDD=5.0v	-	-	3.0	μA

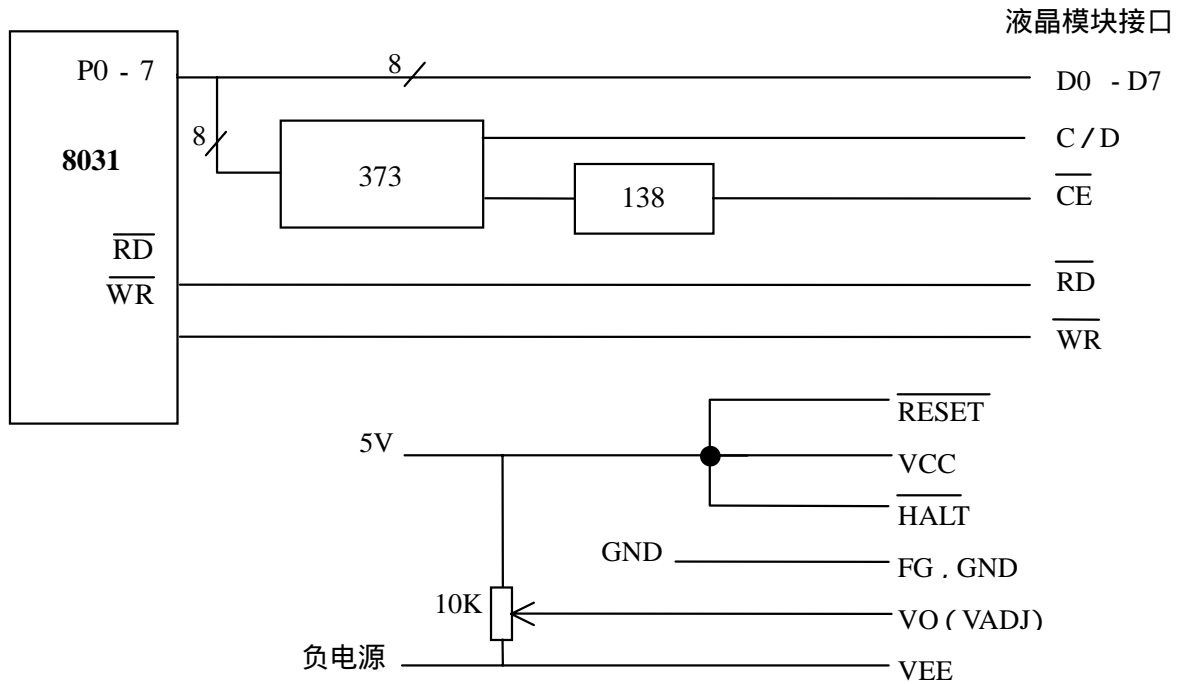
与 MPU 接口时序：



第五章、内藏 T6963C 的液晶显示模块与 MPU 的接口方法

1、直接访问方式

MPU 利用数据总线与控制信号直接采用存储器访问形式或 I/O 设备访问形式控制 T6963C 类液晶显示模块。接口电路如下图所示：



8031 数据接口 P0 直接与液晶显示模块的口连接，由于 T6963C 接口适用于 8080 系列 MPU，所以可以直接用 8031 的 RD、WR 作为液晶显示模块的读、写控制信号，液晶显示模块 RESET，HALT 挂在 5V 上。CE 信号可由地址线译码或与 MREQ 或 IORQ 共同产生。例如，取 138 译码器 Y0 输出。C/D 信号由 8031 地址线 A0 提供，A0 = 1 为指令口地址；A0 = 0 为数据口地址。各驱动子程序如下：

(1)、读标志字子程序

占用寄存器：R0, A；输出寄存器：A 存储标志字

```

PR0:  MOV R0, #01H      ; 指令口地址
      MOVX A, @R0      ; 读标志字
      RET
    
```

由此程序派生出判断有关标志位的子程序：

1) 判断 STA0, STA1 状态位子程序，在写指令的读，写数据之前这两个标志位必须同时为“1”：

```

PR01: ACALL PR0
      JNB ACC.0, PR01  ; 判别 STA0
    
```

```

JNB ACC. 1 , PR01      ; 判别 STA1
RET

```

2) 判断 STA2 标志位子程序，该位在数据自动读操作过程中取代 STA0 和 STA1 有效。在连续读过程中每读一次之前都要确认 STA2=1。

```

PR02 : ACALL PR0
        JNB ACC. 2 , PR02      ; 判别 STA2
        RET

```

同理，数据自动写标志位 STA3 的判断子程序如下：

```

PR03 : ACALL PR0
        JNB ACC. 3 , PR03      ; 判别 STA3
        RET

```

3) 屏读或屏拷贝指令执行后，紧接着要判断 STA6 标志位，若 STA6=0，则表示指令条件正确，执行有效。如：

```

PR06 : ACALL PR0
        JB  ACC. 6 , ERR        ; 判别 STA6
        RET
ERR : .....                   ; 出错处理程序

```

(2)、写指令和数据子程序

有了判断各标志位的子程序，写指令的子程序也就随之而来了。如：

占用寄存器：R0，R2，R3，R4，A；

输入寄存器：R2 参数第一字节，R3 参数第二字节，R4 指令代码

```

PR1 : ACALL PR01      ; 双字节参数指令入口
      MOV  A , R2
      ACALL PR14
PR11 : ACALL PR01     ; 单字节参数指令入口
      MOV  A , R3
      ACALL PR14
PR12 : ACALL PR01     ; 无参数指令入口
      MOV  A , R4
      SIMP PR15
PR14 : MOV  R0 , #00H ; 写数据入口，数据口地址
PR15 : MOVX @R0 , A
      RET

```

此程序是通用程序，当单参数指令或写入数据时，应把参数值或数据送入 R3 内，其子程序入口分别为 PR11 和 PR14。

无参数指令写入子程序入口为 PR12。

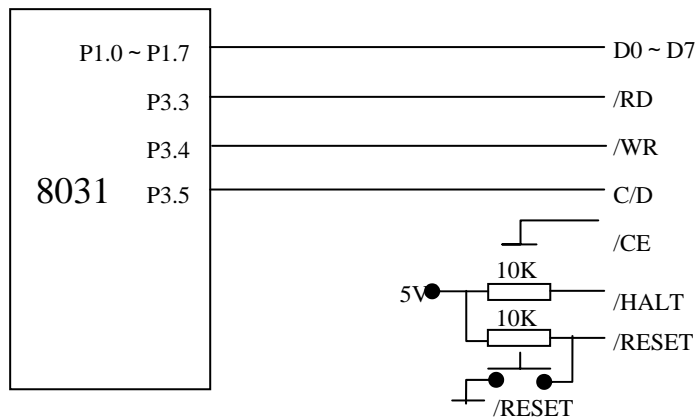
(3)、读数据子程序

占用寄存器：R0，A；输出寄存器：A 存储数据

```

PR2:  ACALL  PR01
      MOV   R0, #001H
      MOVX  A, @R0
      RET
    
```

2、间接控制方式



间接控制方式是 MPU 通过并行接口间接实现对液晶显示模块控制。根据液晶显示模块的需要，并行接口需要一个 8 位并行接口和一个 3 位并行口，由下图所示。8031 的 P1 口作为数据总线，P2 口中 3 位作为读，写及寄存器选择信号。由于并行接口只用于液晶显示模块，所以 CE 信号接地就行了。MPU 通过并行接口操纵液晶显示模块，要对其时序关系有一清楚的了解，并应在程序中明确的反映出来。间接控制方式的基本程序如下：

(1) 读标志字子程序

占用寄存器：A；输出寄存器：A 存储标志字。

```

PR0:  ORL   P3, #38H      ;控制口初始化，C/D = 1
      OLR   P1, #0FFH     ;P1 口为输入方式
      CLR   P3.3         ;RD = 0
      MOV   A, P1        ;读标志字
      SETB  P3.3         ;RD = 1
      RET
    
```

此程序可以用直接访问方式中的 PR01，PR02，PR03 和 PR06 等子程序直接调用。

(2) 写指令和数据子程序

占用寄存器：R2, R3, R4, A；

输入寄存器：R2 参数第一字节, R3 参数第二字节, R4 指令代码

```

PR1:  ACALL  PR01      ; 双字节参数指令入口
      MOV   A, R2
      ACALL PR14
PR11: ACALL  PR01      ; 单字节参数指令入口
      MOV   A, R3
      ACALL PR14
PR12: ACALL  PR01      ; 无参数指令入口
      MOV   A, R4
      SJMP PR15
PR14:  CLR   P3. 5      ; 写数据入口, C/D = 0
PR15:  CLR   P3. 4      ; WR = 0
      MOV   P1, A       ; 写入
      SETB  P3. 4       ; WR = 1
      SETB  P3. 5       ; C/D = 1
      RET

```

(3) 读数据子程序

占用寄存器：A；输出寄存器：A 存储数据

```

PR2:  ACALL  PR01
      CLR   P3. 5      ; C/D = 0
      ORL   P1, #0FFH  ; P1 口输入方式
      CLR   P3. 3      ; RD = 0
      MOV   A, P1      ; 读数据
      ORL   P3, #38H   ; RD = C/D = 1
      RET

```


第六章：T6963C 控制器液晶模块应用程序示例

以 LG1601281 为例:

```
;CE-GND,C/D-A0,A0=00H-DATA WR,A0=01H-COMMAND WR
```

```
AJMP INI
```

```
PR0: MOV R0,#01H
```

```
MOVX A,@R0
```

```
RET
```

```
RDST: ACALL PR0
```

```
JNB ACC.0 RDST ;ST0=1 R/W READY
```

```
JNB ACC.1 RDST ;ST1=1 R/W READY
```

```
RET
```

```
ST03: ACALL PR0
```

```
JNB ACC.3,ST03 ;ST3=1 AUTOW READY
```

```
RET
```

```
CDWR2:ACALL RDST
```

```
MOV A,R2
```

```
ACALL WCD0
```

```
CDWR1:ACALL RDST
```

```
MOV A,R3
```

```
ACALL WCD0
```

```
CDWR0:ACALL RDST
```

```
MOV A,R4
```

```
SJMP WCD
```

```
WCD0:MOV R0,#00H
```

```
WCD: MOVX @R0,A
```

```
RET
```

```
LCDC:MOV R2,#00H ;CLEAR SCREEN
```

```
MOV R3,#00H
```

```
MOV R4,#24H
```

```
ACALL CDWR2
```

```
MOV R4,#0B0H ;AUTO WRITE ON
```

```
ACALL CDWR0
```

```
MOV R5,#1FH
```

```
LC1: MOV R6,#0FFH ;8K=1FFF
```

```
LC2:ACALL ST03
```

```
CLR A
```

```
ACALL WCD0
DJNZ R6,LC2
DJNZ R5,LC1
MOV R4,#0B2H      ;AUTO WRITE OFF
ACALL CDWR0
RET
```

WDT:CLR A

```
MOVC A,@A+DPTR
MOV R3,A
MOV R4,#0C0H      ;WRITE ON
ACALL CDWR1
INC DPTR
;DJNZ R5,WDT
CJNE R3,#0AAH,WDT
RET
```

INI: MOV R4,#80H ;MODE

```
ACALL CDWR0
MOV R2,#00H      ;G HOME=0000
MOV R3,#00H
MOV R4,#42H
ACALL CDWR2
```

```
MOV R2,#14H      ;G AREA=14
MOV R3,#00H
MOV R4,#43H
```

```
ACALL CDWR2
```

```
MOV R2,#00H      ;T HOME=1000
MOV R3,#10H
MOV R4,#40H      ;TEXT
ACALL CDWR2
```

```
MOV R2,#14H      ;COLM=20(T AREA)
MOV R3,#00H
MOV R4,#41H      ;TEXT
ACALL CDWR2
```

```
ACALL LCDC
```

```
MOV R4,#9CH
ACALL CDWR0
```

```
MOV R2,#00H
MOV R3,#00H
MOV R4,#24H      ;G ADP=0000
ACALL CDWR2
```

```
MOV DPTR,#DATA2
ACALL WDT      ;W BITS(G MODE) DATA2
```

```
MOV R2,#50H
MOV R3,#10H
```

```
MOV R4,#24H      ;T ADP=1050
ACALL CDWR2
```

```
MOV DPTR,#DATA
ACALL WDT
```

```
MOV R2,#03H      ;HZ AREA 2K:1800-1FFF,HIGH 5 BITS=00011=D1=03H
                    ;0001100000000000-0001111111111111
                    ;C-CODE BEGIN FROM 80=10000000 D2=0
MOV R3,#00H      ;0001,1100,0000,0000=1C00H
                    ;D1-LOW,D2-HIGH=00,0011,0000
MOV R4,#22H      ;CG OFFSET REGISTER SET,D1,D2,CODE
ACALL CDWR2
```

```
MOV R2,#00H      ;LOW=00
MOV R3,#1CH      ;HIGE=1C
MOV R4,#24H      ;ADDRESS POINTER SET
ACALL CDWR2
```

```
MOV R4,#0B0H
ACALL CDWR0
```

```
MOV DPTR,#DATA1
```

```

MOV R2,#20H          ;32 FONTS,ONE HZ
N1: ACALL ST03
CLR A
MOVC A,@A+DPTR      ;W DATA1 CHINESE CODE
ACALL WCD0
INC DPTR
DJNZ R2,N1
;CJNE A,#0AAH,N1
MOV R4,#0B2H        ;AUTO WRITE OFF
ACALL CDWR0
    
```

```

DATA: DB 00H,00H,00H,00H,37H,25H,2CH,23H,2FH,2DH,
      DB 25H,00H,00H,34H,2FH,00H,00H,00H,00H,00H,
      DB 00H,00H,2BH,21H,29H,24H,25H,00H,00H,23H,
      DB 2FH,2DH,30H,21H,2EH,39H,01H,00H,00H,00H,
      DB 00H,00H,00H,00H,00H,00H,00H,00H,80H,82H,
      DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,
      DB 00H,00H,00H,00H,00H,00H,00H,00H,81H,83H,
      DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,0AAH
    
```

```

DATA1:DB 01H,01H,01H,7FH,41H,41H,41H,41H, ;ZHONG 80
      DB 7FH,41H,01H,01H,01H,01H,01H,01H, ; 81
      DB 00H,00H,04H,0FEH,04H,04H,04H,04H, ; 82
      DB 0FCH,04H,00H,00H,00H,00H,00H,00H, ; 83
    
```

```

DATA2:DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH,
    
```

```
DB 0AAH,  
END
```