

## T6963C 液晶显示驱动控制器

在中规模图形液晶显示模块中，内置 T6963C 控制器的液晶显示模块是目前较为常用的品种较多的内置控制器型图形液晶显示模块。

### 第一节 内置 T6963C 控制器型图形液晶显示模块的电路特性

内置 T6963C 控制器型液晶显示模块的驱动控制系统是由液晶显示控制器 T6963C 及其周边电路，行驱动器组，列驱动器组以及液晶驱动偏压电路组成。从模块的外接口考察模块的电路特性，实际上就是 T6963C 的电路特性。

T6963C 的最大特点是具有独特的硬件初始值设置功能，显示驱动所需的参数如占空比系数，驱动传输的字节数 / 行及字符的字体选择等均由引脚电平设置，这样 T6963C 的初始化在上电时就已经基本设置完成，软件操作的主要精力就可以全部用于显示画面的设计上了。这就是 T6963C 广泛应用于内置控制器型的液晶显示模块的原因。

T6963C 可以管理 64K 的显示存储器（实际液晶显示模块上提供的显存大小请参照具体产品的规格书，目前产品中集成 8K 显存的居多）。显存是用来存储显示数据的，可以分为两个或三个区来使用：文本区、图形区和字符发生器区（即 CGRAM，T6963C 内部集成了字符发生器 CGROM，见附表。如果使用内部 CGROM 的话，可以不分配此区）。例如：

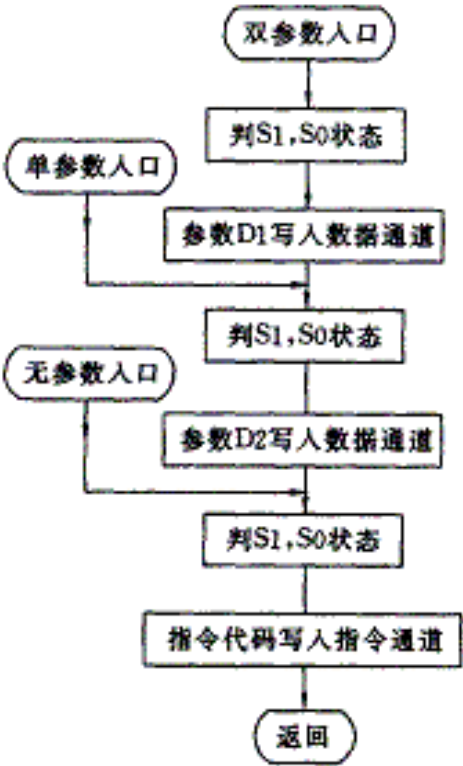
0x0000	字符区
0x07FF	
0x0800	图形区
0x17FF	
0x1800	
0x1FFF	CGRAM

### 第二节 内置 T6963C 控制器型图形液晶显示模块的软件特性

由于 T6963C 使用了硬件初始化设置，所以使得其指令功能集中于显示功能的设置上，从而加强了 T6963C 的显示控制能力。

T6963C 指令的运行时间有些是不能确定的，这是因为有些指令的运行要受到当时控制部的状态影响。

在 T6963C 指令中有的指令需要参数的补充，如地址指针的设置。T6963C 指令参数的输入是在指令代码写入之前。为此 T6963C 的指令写入的流程图如图所示。如果在带有参数的指令代码写入之前，写入多于该指令参数的数据，那么 T6963C 认可最近写入的数据为该指令的参数。



T6963C 的指令表如下：

指令名称	控制状态			指令代码								参数 量	运行时间
	CD	RD	WR	D7	D6	D5	D4	D3	D2	D1	D0		
读状态字	1	0	1	S7	S6	S5	S4	S3	S2	S1	S0	无	—
地址指针设置	1	1	0	0	0	1	0	0	N2	N1	N0	2	状态检测
显示区域设置	1	1	0	0	1	0	0	0	0	N1	N0	2	状态检测
显示方式设置	1	1	0	1	0	0	0	CG	N2	N1	N0	无	32x1/Fosc
显示状态设置	1	1	0	1	0	0	1	N3	N2	N1	N0	无	32x1/Fosc
光标形状设置	1	1	0	1	1	0	0	0	N2	N1	N0	无	32x1/Fosc
数据自动读写设置	1	1	0	1	0	1	1	0	0	N1	N0	无	32x1/Fosc
数据一次读写设置	1	1	0	1	1	0	0	0	N2	N1	N0	1	32x1/Fosc
屏读（一字节）设置	1	1	0	1	1	1	0	0	0	0	0	无	状态检测
屏拷贝（一行）设置	1	1	0	1	1	1	0	1	0	0	0	无	状态检测
位操作	1	1	0	1	1	1	1	N3	N2	N1	N0	无	状态检测
数据写操作	0	1	0	数 据								无	状态检测
数据读操作	0	0	1	数 据								无	状态检测

T6963C 指令的详细说明如下。

● 读状态字（STATUS READ）

格式	S7	S6	S5	S4	S3	S2	S1	S0
----	----	----	----	----	----	----	----	----

T6963C 的状态字由七位标志位组成，它们是：

- S0（STA0） 指令读写状态
- 1：准备好 0：忙
- S1（STA1） 数据读写状态
- 1：准备好 0：忙

S2 (STA2)	数据自动读状态	1: 准备好	0: 忙
S3 (STA3)	数据自动写状态	1: 准备好	0: 忙
S4 (STA4)	未用		
S5 (STA5)	控制器运行检测可能性	1: 可能	0: 不能
S6 (STA6)	屏读 / 屏拷贝出错状态	1: 出错	0: 正确
S7 (STA7)	闪烁状态检测	1: 显示	0: 关显示

这七个标志位各有各的应用场合，并非同时都有效。在计算机写指令或一次读 / 写数据时，S0 和 S1 要同时有效，即“准备好”状态；当计算机使用自动读 / 写功能时，S2 或 S3 将取代 S0 和 S1 作为忙标志位，此时计算机就要判别它是否有效；S6 标志是考察 T6963C 屏读或屏拷贝指令执行情况的标志位。S5 和 S7 表示控制器内部运行状态，在 T6963C 的应用上不会使用它们。

对 T6963C 的软件操作每一次之前都要进行判“忙”。只有仅在不“忙”的状态下计算机对 T6963C 的操作才有效。

### ●地址指针设置 ( REGISTER SET)

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

该指令为双参数 (D1, D2) 指令。指令代码中的 N2, N1, N0 取值“1”为有效，“0”为无效，而且不能同时为“1”，根据 N 的取值，该指令有三种含义，如下表所示。

D1	D2	指令代码	功能
水平位置 (低 7 位有效)	垂直位置 (低 5 位有效)	21H (N0=1)	光标地址设置
偏置地址 (低 5 位有效)	00H	22H (N1=1)	CGRAM 偏置地址设置
低字节	高字节	24H (N2=1)	显示地址设置

#### 1. 光标地址设置 D1—D2—21H

T6963C 的光标控制是独立于显示地址控制的，它专门有一个光标指针寄存器存放当前的光标地址，而且光标地址不会自动修改。该指令设置或称修改了光标在显示屏上的位置。光标的地址以二维坐标形式，以字符为单位设置。水平方向的位置由 D1 参数确定，取值范围在 00H~4FH (1~80 字符位)，表示显示屏左起第几个字符位；垂直方向由 D2 参数确定，取值范围在 00H~1FH (1~32 字符行)，表示显示屏上起第几个字符行。光标在双屏结构的显示屏上的垂直方向位置的规定为上半屏为 00H~0FH，下半屏为 10H~1FH。这个规定不考虑显示屏的实际点行数。

#### 2. CGRAM 偏置地址设置 D1—D2—22H

T6963C 可以管理 2K 的 CGRAM。在显示存储器内要划出 2K 的区域作 CGRAM 使用，只需确定 16 位地址的高 5 位 (ad15~ad11) 即可。CGRAM 偏置地址寄存器就是用来存储这个地址值的。用户可以通过将这个寄存器的内容与自定义字符代码值组合出显示存储器中该字符字模数组所在的首地址：

ad15 ad14 ad13 ad12 ad11 ad10 ad9 ad8 ad7 ad6 ad5 ad4 ad3 ad2 ad1 ad0  
 | ←—偏置地址值 (5 位) —→ + ←——— 字符代码值 (8 位) ———→ | 0 0 0  
 ad2~ad0 从 0 至 7 指向将该字符 8 个字节的字模。

偏置地址的设置由参数 D1 的低 5 位值来实现，D2 设置为 00H。例如：

指令 03H—00H—22H 设置字符代码为 80H。那么该字符字模组在显示存储器的首地址为 1C00H。字模存放在 1C00H~1C07H 单元内。

#### 3. 显示地址设置 D1—D2—24H

该指令将计算机所要访问的显示存储器的地址写入 T6963C 的地址指针计数器中。该地址指针计数器为 16 位字长，需要两个字节。D1 为低 8 位地址，D2 为高 8 位地址。

● 显示区域设置 (CONTROL WORD SET)

格式	D1	D2	0	1	0	0	0	0	N1	N0
----	----	----	---	---	---	---	---	---	----	----

该指令是双参数指令，它将在显示存储器内划分出各显示区域的范围。它是由设定显示区域的首地址和宽度来确定该显示区域的范围，同时也确定了显示存储器单元与显示屏上各点像素的对应关系。该指令中 N1, N0 有四种组合，每个组合有不同的含义。具体功能如下表所示。

N1	N2	D1	D2	指令代码	功能
0	0	低字节	高字节	40H	文本显示区首地址
0	1	字节数	00H	41H	文本显示区宽度
1	0	低字节	高字节	42H	图形显示区首地址
1	1	字节数	00H	43H	图形显示区宽度

1. 文本显示区首地址设置 D1—D2—40H

该指令设置了文本显示区在显示存储器中的起始地址。该地址对应显示屏上左上角的第一个字符位 (home)。定时间定间隔地修改这个地址将会产生显示画面的平滑滚动。参数 D1 为该地址的低 8 位，D2 为该地址的高 8 位。

2. 文本显示区宽度设置 D1—00H—41H

该指令规定了在文本显示区中作为一行显示所占的单元 (字节) 数。该数据与文本显示首地址一起确定了显示单元与显示屏上各点像素的对应关系。比如设文本显示区首地址为 SAD，文本显示区宽度为 CR，n 为字符位数，则文本显示单元与显示屏上字符位的对应关系如下图所示。

字符行\	1	2	.....	n	字符位
1	SAD	SAD+1	.....	SAD+CR-1	
2	SAD+CR	SAD+CR+1	.....	SAD+2CR-1	

图 文本显示单元与显示屏上字符位的对应关系

当设置的显示区宽度超过了实际显示屏上显示所需的单元数，那么超出的部分不显示。但是要注意的是这里使用软件设置的显示区域宽度仅是 T6963C 作为计算每行显示数据传输的起始单元地址的参数，实际传输的字节数已由硬件引脚设置完成。比如 MGLS240128T 模块，通常为了计算地址方便，设立文本显示区域宽度为 20H (32 字节)，但 T6963C 实际一行传输数据为 40 个字节，也就是说 T6963C 将所设置的下一行单元的数据也作为这一行的数据传输了，只是由于屏点阵数小于所传输的数据量不显示而已。但是 T6963C 在传输下一行数据时要使用这个宽度参数计算起始单元的地址，如上图所示那样。

3. 图形显示区首地址设置 D1—D2—42H

该指令设置了图形显示区在显示存储器中的起始地址。该地址对应显示屏上左上角的第一个 8 点列像素 (home)。一个水平 8 点像素作为一个像素组由一个字节表示。定时间定间隔地修改这个地址将会产生显示画面的平滑滚动。参数 D1 为该地址的低 8 位，D2 为该地址的高 8 位。

4. 圆形显示区宽度设置 D1—00H—43H

该指令规定了在图形显示区中作为一行显示所占的单元 (字节) 数。该数据与图形显示首地址一起确定了显示单元与显示屏上各像素组 (字节) 的对应关系。比如设图形显示区首地址为 SAD，图形显示区宽度为 CR，n 为像素点数，则图形显示单元与显示屏上像素组的对应关系如下图所示。

点\	1	2	...	n/8	像素组
行\	1-8	9-16	...	n-8-n	像素点
1	SAD	SAD+1	.....	SAD+CR-1	
2	SAD+CR	SAD+CR+1	.....	SAD+2CR-1	

图 图形显示单元与显示屏上显示像素的对应关系

当设置的显示区宽度超过了实际显示屏上显示所需的单元数，那么超出的部分不显示。但是要注意的是这里使用软件设置的显示区域宽度仅是 T6963C 作为计算每行显示数据传输的起始单元地址的参数，实际传输的字节数已由硬件引脚设置完成。比如 MGLS240128T 模块，通常为了计算地址方便，设置文本显示区域宽度为 20H（32 字节），但 T6963C 实际一行输数据为 40 个字节，也就是说 T6963C 将所设置的下一行单元的数据也作为这一行的数据传输了，只是由于屏点阵数小于所传输的数据量不显示而已。但是 T6963C 在传输下一数据时要使用这个宽度参数计算起始单元的地址，如上图所示那样。

● 显示方式设置（MODE SET）

格式	1	0	0	0	CG	N2	N1	N0
----	---	---	---	---	----	----	----	----

该指令为显示方式的设置，无参数。它的几个设置位功能如下：

CG 位 字符发生器选择位。

当 CG=0 时，启用内部字符发生器 CGROM，该字符库有 128 种字符，其代码为 00H~7FH；同时可以建立 128 种 8 × 8 点阵的自定义字符发生器 CGRAM，其字符代码规定在 80H~FFH 范围内；

当 CG=1 时，禁止内部 CGROM，字符显示完全取自自定义字符发生器 CGRAM，该字符库为 2K 字节容量，字符代码为 00H~FFH。

N2, N1, N0 位为显示方式设置位，它们的组合所产生的显示方式如下表所示。

N2	N1	N0	显示方式	说明
0	0	0	逻辑“或”	文本与图形以逻辑“或”的关系合成显示
0	0	1	逻辑“异或”	文本与图形以逻辑“异或”的关系合成显示
0	1	1	逻辑“与”	文本与图形以逻辑“与”的关系合成显示
1	0	0	文本属性	文本显示特征以双字节表示

在设置了文本属性显示方式后，图形显示区将转换成文本属性区，用于存储字符的属性代码，其地址与显示屏上的对应关系与文本显示区相同。因此在显示屏上某位置上显示的字符是由双字节数据组成，第一字节为字符代码存储在文本显示区内，第二字节为属性代码存储在文本属性区内。这种表示方法与计算机的字符显示数据相似。在文本属性显示方式下，字符的属性代码由一字节的低 4 位组成：

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	D3	D2	D1	D0

其中 D3 位是字符闪烁控制位。D3=0 为不闪烁；D3=1 为闪烁。

D2-D0 组合功能如下表所示。

D2	D1	D0	显示效果
0	0	0	正向显示
0	0	1	负向显示
0	1	1	禁止显示（正向）

1	0	0	禁止显示（负向）
---	---	---	----------

### ●显示状态设置（DISPLAY MODE）

格式	1	0	0	1	N3	N2	N1	N0
----	---	---	---	---	----	----	----	----

该指令设置了当前的显示状态，无参数。该指令有 4 个设置位，每一位都代表一种显示状态的设置，它们可以同时有效，也可以部分有效，也可以都无效。这 4 个设置位的功能如下：

N0 光标闪烁设置开关 N0=1 为启用光标闪烁；N=0 为禁止光标闪烁。

N1 光标显示设置开关 N0=1 为启用光标显示；N=0 为禁止光标显示。

N2 文本显示设置开关 N0=1 为启用文本显示；N=0 为禁止文本显示。

N3 图形显示设置开关 N0=1 为启用图形显示；N=0 为禁止图形显示。

在文本显示与图形显示合成显示时，文本显示开关与图形显示开关应同时启用，在文本属性显示方式下，图形显示开关也应启用，只是特性不同。光标显示及光标闪烁功能的启用要在文本显示启用时进行，否则无效。

### ●光标形状设置（CURSOR PATTERN SELECT）

格式	1	0	1	0	0	N2	N1	N0
----	---	---	---	---	---	----	----	----

该指令设置光标的显示形状。无参数补充。在指令中的三个设置位设置了光标点数。光标是以 8 点列×N 行，行的取值由设置位 N2，N1，N0 组合完成。

### ●数据自动读写设置（DATA AUTO READ WRITE）

格式	1	0	1	1	0	0	N1	N0
----	---	---	---	---	---	---	----	----

使用该指令将进入或退出数据的自动读或自动写方式。在自动读或自动写方式中，计算机可以连续地将显示数据写入显示存储器中或从显示存储器中读取数据。在每次读或写的操作后，显示地址自动加一。进入自动读方式或自动写方式时，状态位将由 S2（自动读方式）或 S3（自动写方式）代替 S1 和 S0。在自动读或自动写方式完成时要输入退出自动读写方式指令。在自动读写方式中写入其他指令都是无效的。

该指令中有两位设置位 N1，N0。这两位将产生三个子指令，如下表所示。

N1	N0	指令代码	功能
0	0	B0H	进入自动写方式
0	1	B1H	进入自动读方式
1	1	B2H / B3H	退出自动读写方式

### ●数据一次读写设置（DATA READ WRITE）

格式	D1	1	1	0	0	0	N2	N1	N0
----	----	---	---	---	---	---	----	----	----

该指令是一次读写数据操作指令。在每次读写数据操作后，显示地址都要根据指令代码的设置而修正：加一，减一或不变。该指令在写入数据时，所带的一个参数就是所要写入的显示数据。当读数据操作时，该指令不带参数，直接写入指令代码，T6963C 在接收到该指令后将当前显示地址计数器所指的显示存储器单元的内容送入接口部的数据栈内，紧接着计算机的读数据操作将其读出。该指令有三个设置位 N2，N1，N0（见下表）。它们的组合功能如下。

参数 D1	N2	N1	N0	指令代码	功能
-------	----	----	----	------	----

数据	0	0	0	C0H	数据写，地址加一
—	0	0	1	C1H	数据读，地址加一
数据	0	1	0	C2H	数据写，地址减一
—	0	1	1	C3H	数据读，地址减一
数据	1	0	0	C4H	数据写，地址不变
—	1	0	1	C5H	数据读，地址不变

### ●屏读（一字节）设置（SCREEN PEEK）

格式	1	1	1	0	0	0	0	0
----	---	---	---	---	---	---	---	---

所谓屏读是指把显示屏上显示的内容取出来作为数据提供给计算机使用，这个内容为一个字节的当前显示数据，它有可能是图形显示数据，也可能是文本显示的某一个字符上的某一行的字模数据，更多的是文本与图形合成显示的内容。屏读指令将使计算机能够直接获得显示屏上的数据，这是其他控制器所没有的功能。屏读指令要求当前显示地址指针指在图形显示区内，所以屏读指令只能在图形显示功能有效时才能使用。在屏读指令写入后要立即检查状态 S6，判断该指令执行是否正确。如果执行正确，就可以读取数据。

该指令为无参数指令。不能应用在文本属性显示方式下。

由于屏读操作所得的数据是由 T6963C 驱动部数据合成后反馈产生的，所对应的单元地址与硬件引脚设置窗口长度有关，所以要求显示区域宽度的软件设置与硬件显示数据传输量设置端 MD3，MD2 所设置的显示窗口长度相同才能正确的计算出对应单元的地址，屏读出正确的数据来。

### ●屏拷贝（一行）设置（SCREEN COPY）

格式	1	1	1	0	1	0	0	0
----	---	---	---	---	---	---	---	---

所谓屏拷贝是指把显示屏上的某一行显示的内容取出来作为图形显示数据返写回图形显示区相应的显示单元内，这个内容为一行数个字节的当前显示数据，它有可能是图形显示数据，也可能是文本显示的某一个字符上的某一行的字模数据，更多的是文本与图形合成显示的内容。屏拷贝功能将当前显示屏上的显示内容拷贝到图形显示区内作为计算机的处理使用。这是其他控制器所没有的功能。屏拷贝指令要求当前显示地址指针指在图形显示区内，所以屏拷贝指令只能在图形显示功能有效时才能使用。在屏拷贝指令写入后要立即检查状态 S6，判断该指令执行得是否正确。如果执行正确，就可以读取数据。

该指令为无参数指令。不能应用在文本属性显示方式下，也不能应用在双屏结构的液晶显示器件的控制上。

由于屏拷贝操作所得的数据是由 T6963C 驱动部数据合成后反馈产生的，所对应的单元地址与硬件引脚设置窗口长度有关，所以要求显示区域宽度的软件设置与硬件显示数据传输量设置端 MD3，MD2 所设置的显示窗口长度相同才能正确的计算出对应单元的地址，屏拷贝出正确的数据来。

### ●位操作（BIT SET RESET）

格式	1	1	1	1	N3	N2	N1	N0
----	---	---	---	---	----	----	----	----

该指令可以对当前显示地址指针所指的显示单元中的数据的一位写“0”或写“1”。操作应由 N2，N1，N0 确定，它们取值在 0~7 之间，对应着数据的 D0~D7 位。N3 为写入的数据，是“1”表示该位将置“1”；是“0”表示该位将清“0”。该指令一次仅能操作一位。该指令无参数。

●数据写操作（DATA WRITE）

格式	数	据
----	---	---

数据写操作是向数据通道里写数据，指令的参数也同样是这样操作。一次写数据或参数时，该数据将写入到数据栈中，再由紧接着写入的指令代码决定将该数据是作为数据写入当前的显示地址指针所指的单元内，还是作为参数写入相应的寄存器中。

●数据读操作（DATA READ）

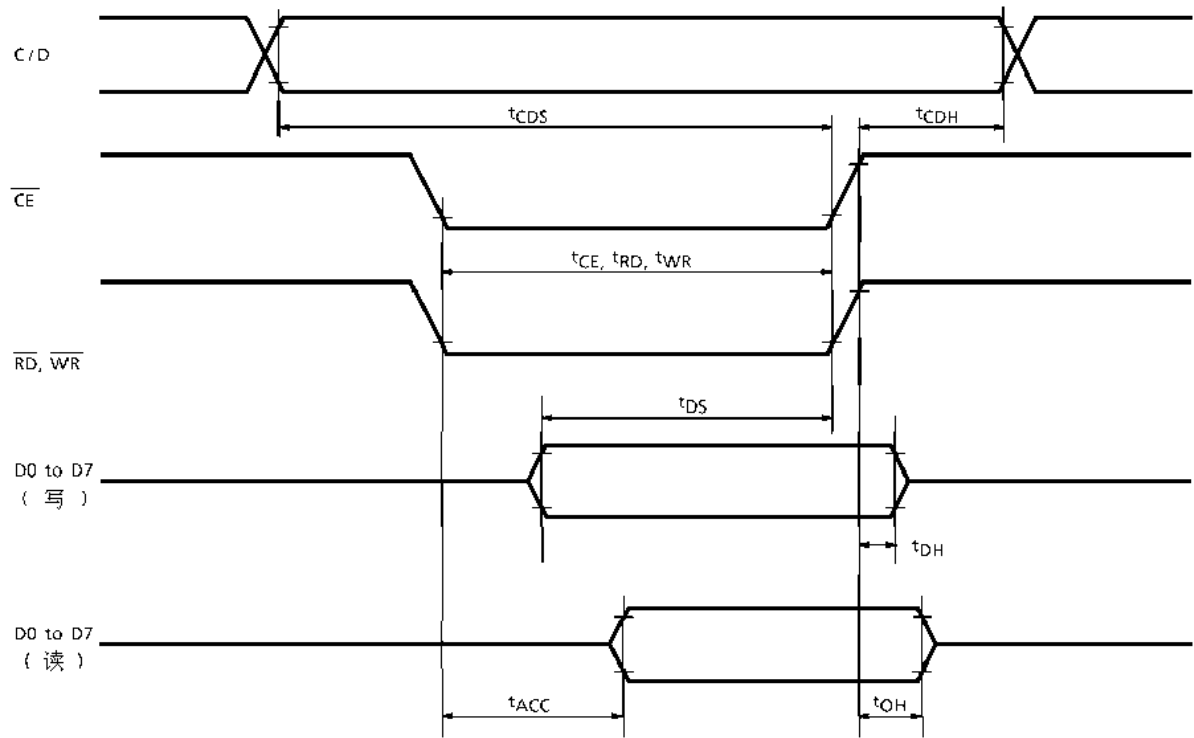
格式	数	据
----	---	---

数据读操作是从数据通道中读取数据。在一次读数据操作时，读数据指令的写入将当前显示地址指针所指的数据取出写入数据栈中，读数据操作将该数据提出送入数据总线上供计算机获取。在自动读操作时连续的读操作将连续地从显示存储器内读取数据，显示地址将自动加一。

第三节内置 T6963C 控制器的液晶显示模块的接口技术

内置 T6963C 控制器的液晶显示模块与计算机的接口时序为 Intel8080 时序。本节以常用的单片机 8031 为实验样机，叙述计算机与内置 T6963C 控制器的液晶显示模块的接口技术。

一、T6963C 接口操作时序



测试条件（默认环境： VDD=5.0V±10%，VSS=0V，工作温度：-20 -- 75℃）

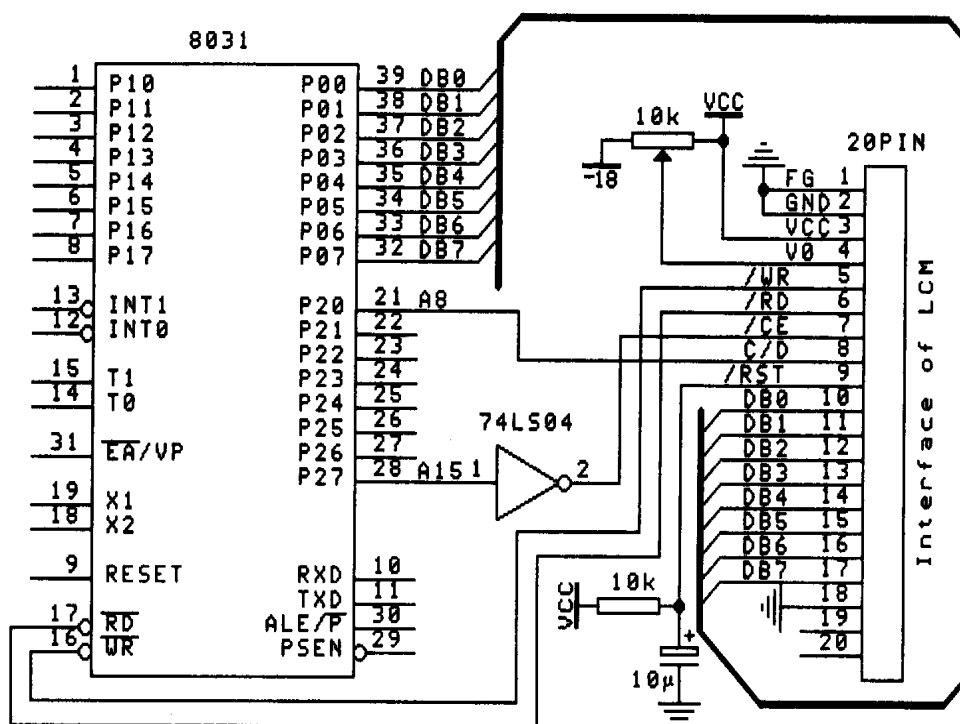
项目	符号	条件	最小值	最大值	单位
C/D 建立时间	$t_{CDS}$	--	100	-	ns
C/D 保持时间	$t_{CDH}$	--	10	-	ns
/CE, /WR, /RD 脉冲宽度	$t_{CE}, t_{WR}, t_{RD}$	--	80	-	ns



数据建立时间	$t_{DS}$	--	80	—	ns
数据保持时间	$t_{DH}$	--	40	—	ns
处理时间	$t_{ACC}$	--	—	150	ns
输出保持时间	$t_{OH}$	--	10	50	ns

## 二、直接访问方式

计算机直接访问方式下与内置 T6963C 控制器的液晶显示模块的实用接口电路如下图所示。在图中使用了地址线 A8 作为模块的寄存器选择线 C/D, 使用地址线 A15 作为模块操作的片选线; 模块的 V0 端所接的电位器是作为液晶驱动电源的调节器, 调节显示的对对比度。



DAT1	EQU	30H	； 第一参数单元
DAT2	EQU	31H	； 第二参数 / 数据单元
COM	EQU	32H	； 指令代码单元
C__ADD	EQU	8100H	； 指令通道地址
D ADD	EQU	8000H	； 数据通道地址

; 读状态字子程序

```

;
; 判状态位 S1, S0 子程序 (读写指令和读写数据状态)
ST01:  LCALL  R__ST
        JNB   ACC.0, ST01
        JNB   ACC.1, ST01
        RET

;
; 判状态位 S2 子程序 (数据自动读状态)
ST2:   LCALL  R__ST
        JNB   ACC.2, ST2
        RET

; -----
; 判状态位 S3 子程序 (数据自动写状态)
ST3:   LCALL  R__ST
        JNB   ACC.3, ST3
        RET

; -----
; 判状态位 S6 子程序 (屏读 / 屏拷贝状态)
ST6:   LCALL  R__ST
        JB    ACC.6, ERR
        RET
ERR:   LJMP  ST6                      ; 出错处理程序

; -----
; 写指令和写数据子程序
PR1:   LCALL  ST01                      ; 双字节参数指令写入入口
        MOV   A, DAT1                  ; 取第一参数单元数据
        LCALL PR13                      ; 写入参数
PR11:  LCALL  ST01                      ; 单字节参数指令写入入口
        MOV   A, DAT2                  ; 取第二参数单元数据
        LCALL PR13                      ; 写入参数
PR12:  LCALL  ST01                      ; 无参数指令写入入口
        MOV   A, COM                   ; 取指令代码单元数据
        LJMP  PR14                      ; 写入指令代码
PR13:  MOV    DPTR, #D__ADD             ; 设置数据通道地址 / 数据写入入口
PR14:  MOVX   @DPTR, A                  ; 写入操作
        RET

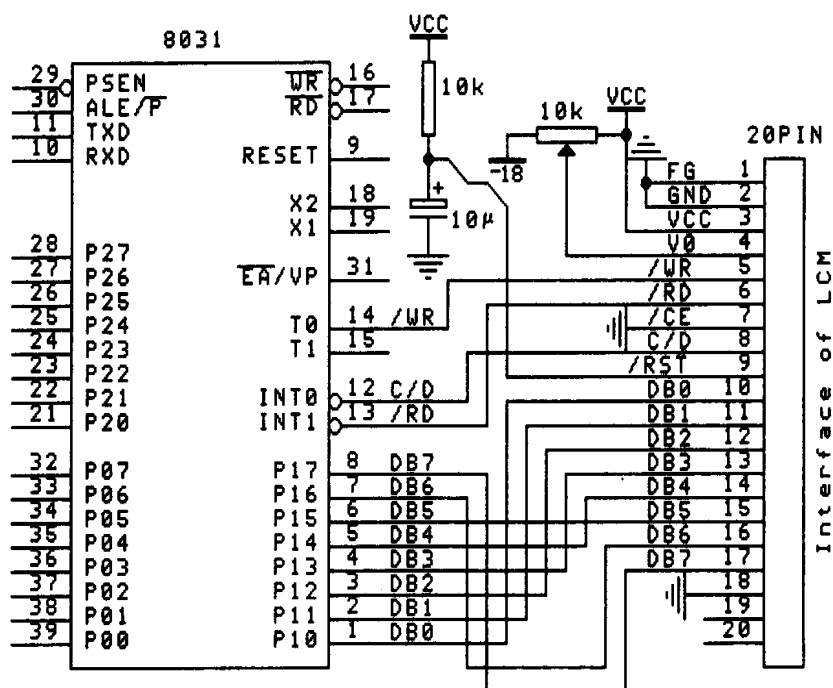
; -----
; 读数据子程序
PR2:   LCALL  ST01                      ; 判状态位
        MOV   DPTR, #D__ADD             ; 设置数据通道地址
        MOVX  A, @DPTR                  ; 读数据操作
        MOV   DAT2, A                   ; 数据存入第二参数 / 数据单元
        RET

```

### 三、间接控制方式

间接控制方式是将内置 T6963C 控制器的液晶显示模块与计算机系统中的一个并行 I/O 接口连接，计算机通过对该 I/O 接口的操作间接的实现模块的控制。间接控制方式在硬件电路上需要一个 8 位并行接口（如 8031 的 P1 口）与模块的数据线连接，作为数据总线，还需要一个 3 位并行接口作为时序控制信号线如 8031 的 P3.2 作为 C/D，P3.3 作为 WR，以及 P3.4 作为 RD。由于使用了专用的并行接口连接模块，而且该并行接口自身在计算机系统中有相应的片选地址，所以模块的片选信号可以直接接地作选通态，间接控制方式的接口电路与时序无关，时序完全靠软件编程实现。计算机间接控制方式下与内置 T6963C 控制器的液晶显示模块的实用接口电路如下图所示。模块的 V0 端所接的电位器是作为液晶驱动电源的调节器，调节显示的对比度。

图 间接控制方式下的接口电路图



间接控制方式的驱动程序如下：

CD	EQU	P3.2	；通道选择信号
WR	EQU	P3.3	；写操作信号
RD	EQU	P3.4	；读操作信号

；—————间接控制方式驱动子程序—————

；读状态字子程序

```

R_ST: MOV    P1, #0FFH    ; P1口置“1”
      SETB   CD           ; CD=1
      CLR    RD           ; RD=0
      MOV    A, P1        ; 读操作
      SETB   RD           ; RD=1
      RET

```

；

；写指令和写数据子程序

```

PR1:  LCALL  ST01         ; 双字节参数指令写入入口
      MOV    A, DAT1      ; 取第一参数单元数据
      LCALL  PR13         ; 写入参数

```

```

PR11:  LCALL  ST01          ; 单字节参数指令写入入口
        MOV    A, DAT2      ; 取第二参数单元数据
        LCALL  PR13          ; 写入参数
PR12:  LCALL  ST01          ; 无参数指令写入入口
        MOV    A, COM        ; 取指令代码单元数据
        LJMP   PR14          ; 写入指令代码
PR13:  CLR     CD            ; C / D=0 / 数据写入入口
PR14:  MOV     P1, A          ; 设置数据
        CLR     WR            ; WR=0
        SETB   WR            ; WR=1
        RET
; -----
; 读数据子程序
PR2:   LCALL  ST01          ; 判状态位
        CLR     CD            ; C / D=0
        MOV     P1, #0FFH     ; P1 口置“1”
        CLR     RD            ; RD=0
        MOV     A, P1          ; 读取数据
        SETB   RD            ; RD=1
        MOV     DAT2, A        ; 数据存入第二参数 / 数据单元
        RET

```

#### 第四节内置 T6963C 控制器型图形液晶显示模块的应用软件

本节以 GTG-2401287 为例屏，应用上一节接口电路及驱动程序，列举一些应用程序，在软件编程方面进一步对内置 T6963C 控制器的液晶显示模块的使用作一说明，并对一些指令的使用特性提供演示程序，供使用者参考。

##### 示例一初始化设置子程序

初始化设置主要是对显示区域的设置和显示方式的设置。这个程序只需在主程序对系统初始化时调用一次就可以了，除非要在应用中需要改变设置。初始化子程序清单如下：

```

INT:   MOV     DAT1, #00H      ; 设置文本显示区域首地址
        MOV     DAT2, #00H
        MOV     COM, #40H
        LCALL  PR1
        MOV     DAT1, #20H     ; 设置文本显示区域宽度
        MOV     DAT2, #00H     ; 即一行显示所占字节数
        MOV     COM, #41H
        LCALL  PR1
        MOV     DAT1, #00H     ; 设置图形显示区域首地址
        MOV     DAT2, #08H     ; 或为文本属性区域首地址
        MOV     COM, #42H
        LCALL  PR1
        MOV     DAT1, #20H     ; 设置图形显示区域宽度

```

```

MOV      DAT2, #00H      ; 或为文本属性区域宽度
MOV      COM, #43H       ; 即一行显示所占字节数
LCALL    PR1
MOV      COM, #0A7H      ; 光标形状设置
LCALL    PR12
MOV      COM, #80H       ; 显示方式设置逻辑“或”合成
LCALL    PR12
MOV      COM, #9CH       ; 显示开关设置 开文本和图形显示
LCALL    PR12
RET

```

需要说明的是为了计算地址的方便，设置显示区域的宽度为 20H

初始化子程序通常在主程序开始时调用。如：

```

ORL      P3, #1CH        ; 间接控制方式控制线初始化
LCALL    INT

```

初始化完成后，开文本和图形显示。此时调整对比度电压，显示屏上将因显示存储器上电时的随机数据而显示出随机的图形和字符。以此可以验证接口电路和驱动程序的正确性。

#### 示例二 清显示存储器子程序

该子程序对显示存储器完全清零，不分显示区域性质。所以在初始化时使用和在全显示存储器需要清零时使用。另外该程序也可以作为接口电路及驱动程序的调试使用，将写入的数据不为“0”，即可以观察显示效果。

```

CLEAR:   MOV      DAT1, #00H      ; 设置显示存储器首地址
          MOV      DAT2, #00H
          MOV      COM, #24H
          LCALL    PR1
          MOV      R3, #00H       ; 设置循环量（8K 字节）
          MOV      R4, #20H
          MOV      COM, #0B0H     ; 设置自动写方式
          LCALL    PR12
CLEAR1:  LCALL    ST3             ; 判状态位 S2
          MOV      A, #00H        ; 置数据“0”
          LCALL    PR13          ; 写入数据
          DJNZ     R3, CLEAR1     ; 循环
          DJNZ     R4, CLEAR1
          MOV      COM, #0B2H     ; 设置自动写结束指令
          LCALL    PR12
          RET

```

#### 示例三 建立 CGRAM 子程序（地址设定在代码为 80H 起处）

该程序是在内部 CGROM 有效时建立 CGRAM 的子程序，所以使用的字符代码是从 80H 开始的。子程序有一个参数为 COUNT，它用于设置写入  $8 \times 8$  点阵字符的字模的个数。如果是汉字字模，那么一个汉字字模就需要 4 个  $8 \times 8$  点阵字符字模组合。子程序如下：

```

COUNT   EQU      34H           ; 西文字符（ $8 \times 8$  点阵）个数
CGRAM:   MOV      DAT1, #03H     ; 设置 CGRAM 偏置地址
          MOV      DATA2, #00H
          MOV      COM, #22H

```

```

                LCALL      PR1
MOV            DAT1, #00H          ; 设置存储器地址指针
                MOV       DAT2, #1CH
MOV            COM, #24H
                LCALL      PR1
MOV            COM, #0B0H          ; 设置自动写方式
                LCALL      PR12
MOV            A, COUNT            ; 计算写入的字节数
MOV            B, #08H
MUL            AB
MOV            COUNT, A            ; 计算器低 8 位数据
INC            B                  ; 计算器高 8 位数据
MOV            DPTR, #CGTAB        ; 取字符库首地址
PUSH           DPH                ; 存字符库指针
PUSH           DPL
CGRAM1:        LCALL      ST3        ; 判自动写状态位
                POP        DPL        ; 取字符库指针
                POP        DPH
CLR            A
MOVC           A, @A+DPTR          ; 取数据
INC            DPTR               ; 字符库指针加一
PUSH           DPH                ; 存字符库指针
PUSH           DPL
                LCALL      PR13        ; 写入数据
DJNZ           COUNT, CGRAM1       ; 计数循环
DJNZ           B, CGRAM1
MOV            COM, #0B2H          ; 写入结束自动写指令
                LCALL      PR12
                POP        ACC        ; 修正栈指针
                POP        ACC
RET

```

;

#### 建立 CGRAM 的步骤

1. 在程序区内建立汉字字模表 CGTAB;
2. 汉字代码定义 (以“冀”=80H 为例)
  - 80H 汉字代码 (表示汉字左上半部字模代码)
  - 81H 隐含代码 (表示汉字左下半部字模代码)
  - 82H 隐含代码 (表示汉字右上半部字模代码)
  - 83H 隐含代码 (表示汉字右下半部字模代码)
3. 写入两个汉字的字模数据, 设置 COUNT 值为 8;
- 4 调用 CGRAM 子程序

```

MOV            COUNT, #08H
                LCALL      CGRAM
                SJMP       $

```

CCTAB:

```
DB 02H, 7EH, 02H, 0FEH, 00H, 3FH, 21H, 3FH
DB 21H, 3FH, 04H, 3FH, 04H, 0FFH, 10H, 60H
DB 80H, 0F8H, 82H, 0FEH, 08H, 0FCH, 08H, 0F8H
DB 08H, 0F8H, 40H, 0F8H, 40H, 0FEH, 10H, 0CH ;冀=80
```

```
DB 00H, 40H, 20H, 27H, 04H, 04H, 0E4H, 27H
DB 24H, 24H, 24H, 2EH, 35H, 24H, 08H, 10H
DB 20H, 28H, 24H, 0FEH, 20H, 20H, 24H, 0A4H
DB 0A8H, 0A8H, 90H, 90H, 30H, 4AH, 8AH, 06H ;诚=84
```

注：汉字字模排列格式为第 1~16 字节为汉字左半部分自上而下排列的字符数据，第 17~32 字节为汉字右半部分自上而下排列的字模数据。

示例四 汉字写入子程序（文本方式）

在文本方式下显示汉字，首先要建立汉字 CGRAM（如上一例所示），要定义汉字代码。在文本方式下写汉字需要写入 4 个代码以组成一个汉字，子程序如下：

```
CODE EQU 33H ; 汉字字符代码
O__Y EQU 35H ; Y 坐标 (O__15)
O__X EQU 36H ; X 坐标 (O__29)
;
WRI_CT: MOV A, O__Y ; 计算显示 RAM 地址
        MOV B, #20H
        MUL AB
        ADD A, O__X
        MOV O__X, A
        MOV DAT1, A
        MOV A, B
        ADDC A, #00H ; 文本显示区首地址为 0000H
        MOV O__Y, A
        MOV DAT2, A ; 设置显示 RAM 地址
        MOV COM, #24H
        LCALL PR1
        MOV A, CODE ; 取汉字代码
        MOV DAT2, A ; 写入左上半部汉字代码
        MOV COM, #0C0H
        LCALL PR11
        MOV A, CODE
        ADD A, #02H
        MOV DAT2, A ; 写入右上半部汉字代码
        LCALL PR11
        MOV A, #20H ; 显示 RAM 地址修正
        ADD A, O__X
        MOV DAT1, A
        CLR A
        ADDC A, O__Y
```

```

MOV      DAT2, A
MOV      COM, #24H      ; 设置显示 RAM 地址
LCALL    PR1
MOV      A, CODE
INC      A
MOV      DAT2, A        ; 写入左下半部汉字代码
MOV      COM, #0C0H
LCALL    PR11
MOV      A, CODE
ADD      A, #03H
MOV      DAT2, A        ; 写入右下半部汉字代码
LCALL    PR11
RET

```

文本方式下汉字显示演示程序:

```

MOV      CODE, #80H
MOV      O__X, #03H
MOV      O__Y, #02H
LCALL    WRI__CT
MOV      CODE, #84H
MOV      O__X, #07H
MOV      O__Y, #02H
LCALL    WRI__CT

```

示例五 汉字写入子程序（图形方式）

在图形方式下显示汉字是比较常用的方法。汉字库作在程序区内，按排序定义各汉字的代码。写入汉字字模如同写入图形数据一般，这里提供单字节汉字代码使用的汉字写入子程序。

```

CODE      EQU      33H      ; 汉字字符代码
O__Y      EQU      35H      ; Y 坐标 (0__127)
O__X      EQU      36H      ; X 坐标 (0__29)
COUNT    EQU      34H      ; 计数器
WRI__CC:  MOV      A, O__Y    ; 计算图形显示 RAM 地址
          MOV      B, #20H    ; 图形显示区宽度为 20H
          MUL      AB
          ADD      A, O__X
          MOV      O__X, A
          MOV      A, B
          ADDC     A, #08H     ; 图形显示区首地址为 0800H
          MOV      O__Y, A
          MOV      DPTR, #CCTAB ; 计算汉字库字符首地址
          MOV      A, CODE
          MOV      B, #20H
          MUL      AB
          ADD      A, DPL
          MOV      DP, A
          MOV      A, B

```



```

        ADDC    A, DPH
        MOV     DPH, A
        PUSH    DPH          ; 存字库地址指针
        PUSH    DP
WRI__CC1: MOV     COUNT, #10H      ; 计数器设置为 16 次
        MOV     DAT1, 0__X      ; 设置显示 RAM 地址
        MOV     DAT2, 0__Y
        MOV     COM, #24H
        LCALL   PR1
        POP     DPL          ; 取字库地址指针
        POP     DPH
        CLR     A
        MOVC    A, @A+DPTR      ; 取左部字模数据
        MOV     DAT2, A        ; 设置数据
        INC     DPTR          ; 字库地址指针加一
        CLR     A
        MOVC    A, @A+DPTR      ; 取右部字模数据
        MOV     DAT1, A        ; 暂存数据
        INC     DPTR          ; 字库地址指针加一
        PUSH    DPH          ; 存字库地址指针
        PUSH    DP
        MOV     COM, #0COH      ; 设置一次写数据指令代码
        LCALL   PR11          ; 写入数据
        MOV     DAT2, DAT1      ; 设置右部数据
        LCALL   PR11          ; 写入数据
        MOV     A, #20H        ; 显示 RAM 地址修正
        ADD     A, 0__X
        MOV     0__X, A
        CLR     A
        ADDC    A, 0__Y
        MOV     0__Y, A
        DJNZ    COUNT, WRI__CC1 ; 计数器循环
        POP     ACC          ; 修正栈指针
        POP     ACC
        RET
; -----
; 汉字库
CCTAB: DB 02H, 80H, 7EH, 0F8H, 02H, 82H, 0FEH, 0FEH
        DB 00H, 08H, 3FH, 0FCH, 21H, 08H, 3FH, 0F8H
        DB 21H, 08H, 3FH, 0F8H, 04H, 40H, 3FH, 0F8H
        DB 04H, 40H, 0FFH, 0FEH, 10H, 10H, 60H, 0CH      ; “冀” 代码: 00

        DB 00H, 20H, 40H, 28H, 20H, 24H, 27H, 0FEH
        DB 04H, 20H, 04H, 20H, 0E4H, 24H, 27H, 0A4H

```

```
DB 24H, 0A8H, 24H, 0A8H, 24H, 90H, 2EH, 90H
DB 35H, 30H, 24H, 4AH, 08H, 8AH, 10H, 06H ; “诚” 代码: 01
```

注：汉字字模排列格式为第 1、2 字书为汉字最顶部的 16 个点，自上而下排列的字模数据。

图形方式下汉字显示演示程序：

```
MOV CODE, #00H
MOV O__X, #00H
MOV O__Y, #00H
LCALL WRI__CC
MOV CODE, #01H
MOV O__X, #03H
MOV O__Y, #00H
LCALL WRI__CC
MOV CODE, #02H
MOV O__X, #06H
MOV O__Y, #00H
LCALL WRI__CC
MOV CODE, #03H
MOV O__X, #09H
MOV O__Y, #00H
LCALL WRI__CC
```

示例六 西文字符写入子程序（文本属性显示方式的应用）

非文本属性显示方式的西文字符写入非常简单，先设置显示地址指针（24H），再写入字符代码即可。所以这里就不再举例说明了，可以取 WRI\_\_C 程序的前 15 条指令作为参考。

该程序使用了字符代码和字符显示属性两个参数。将图形显示区域作为文本属性区。在写入字符代码后将字符的属性参数写入相应的属性区内。

```
CODE EQU 33H ; 字符代码
ATTR EQU 34H ; 字符属性码（低 4 位有效）
O__Y EQU 35H ; Y 坐标（0__15）
O__X EQU 36H ; X 坐标（0__29）
WRI_C: MOV A, O__Y ; 计算文本显示 RAM 地址
MOV B, #20H ; 文本显示区宽度为 20H
MUL AB
ADD A, O__X
MOV DAT1, A
MOV A, B
ADDC A, #00H
MOV DAT2, A
PUSH ACC ; 存地址的高 8 位
MOV COM, #24H ; 设置地址指针
LCALL PR1
MOV DAT2, CODE ; 写入字符代码
MOV COM, #0C4H
LCALL PR11
```

```

        POP        ACC                ; 取地址的高 8 位
        ADD        A, #08H           ; 计算相应的属性区 RAM 地址
        MOV        DAT2, A
        MOV        COM, #24H        ; 设置地址指针
        LCALL     PR1
        MOV        DAT2, ATTR        ; 写入属性参数
        MOV        COM, #0C4H
        LCALL     PR11
        RET

;
; 文本属性方式设置程序段
        MOV        COM, #84H        ; 设置文本属性显示方式
        LCALL     PR12
        MOV        COM, #9FH        ; 开文本, 图形显示及光标闪烁显示
        LCALL     PR12

;
; 文本属性方式显示演示程序段
        MOV        DAT1, #0EH        ; 文本光标位置设置
        MOV        DAT2, #01H
        MOV        COM, #21H
        LCALL     PR1
LOOPA:   MOV        R1, #00H          ; 设置属性寄存器=0
        MOV        R2, #00H          ; 设置表间址寄存器=0
        MOV        R3, #00H          ; 设置显示 X 坐标寄存器=0
        MOV        R4, #01H          ; 设置显示 Y 坐标寄存器=0
LOOPB:   MOV        A, R2
        MOV        DPTR, #DTAB
        MOVC       A, @A+DPTR
        MOV        CODE, A           ; 置字符代码
        MOV        ATTR, R1          ; 置字符属性值
        MOV        O__X, R3          ; 置 X 坐标
        MOV        O__Y, R4          ; 置 Y 坐标
        LCALL     WRI__C             ; 调子程序
        INC        R2
        INC        R3
        MOV        A, R2
        CJNE       A, #11, LOOPB
        LCALL     DELAY
        INC        R1
        LJMP      L00PA
DELAY:   MOV        R5, #00H
        MOV        R6, #00H
DELAY1:  NOP
        DJNZ       R5, DELAY1

```

```

        DJNZ    R6, DELAY1
        RET
DTABS      DB 37H, 45H, 4CH, 43H, 4FH, 4DH, 45H
            DB 00H, 39H, 4FH, 55H "Welcome you"

```

该演示程序有光标位置设置的程序段。字符属性参数在软件特性中已给出，这里将该参数从 0 到 FH 循环设置一遍，使用者可以从中了解参数值的特性。

参数与显示效果为：

00H~02H, 07H	正向显示
03H	禁止显示（白）
04H, 06H	禁止显示（黑）
05H	负向显示
08H~0AH, 0FH	正向闪烁显示
0BH	禁止显示（白）
0CH, 0EH	禁止显示（黑）
0DH	负向闪烁显示

#### 附：T6963C 内置字符表

低位 高位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
2	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
4	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
6	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K
7	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	

---

**冀诚公司销售网络:**

大陆工厂

地址: 河北省石家庄市新石北路 368 号 邮政编码: 050091

电话: +86-311-3856940 传真: +86-311-3851891

电子邮件: sales@gemtech-hb.com

---

深圳宾康实业有限公司

地址: 深圳福田区华强北路宝华大厦 A1205 室

邮政编码: 518031 电话: +86-755-83742919 传真: +86-755-83742632

---

冀诚电子杭州分公司

地址: 杭州市文苑路 522 号 4-2-601

电话: +86-571-88988297

---

香港代理

宾康有限公司

地址: 香港九龙湾宏照道 11 号 宝隆中心 B 座 3 字楼 12 室

电话: +852-27518691 传真: +852-27965670

电子邮件: pantage@asiansources.com

---

新加坡代理

大欣工业私人有限公司

地址: 41 Kallang Pudding Road #06-06 Golden Wheel Building Singapore 349316s

电话: +65-7463198 传真: +65-7468348

电子邮件: tassinsingnet.com.sg