



中显液晶  
技术资料



型号: ZX19264A

2009年3月15日

北京市海淀区中关村大街32号和盛大厦811室  
电话：(86)-010-52926620 传真：(86)-010-52926621  
企业网站：<http://www.zxlcd.com>

## 一. 概述

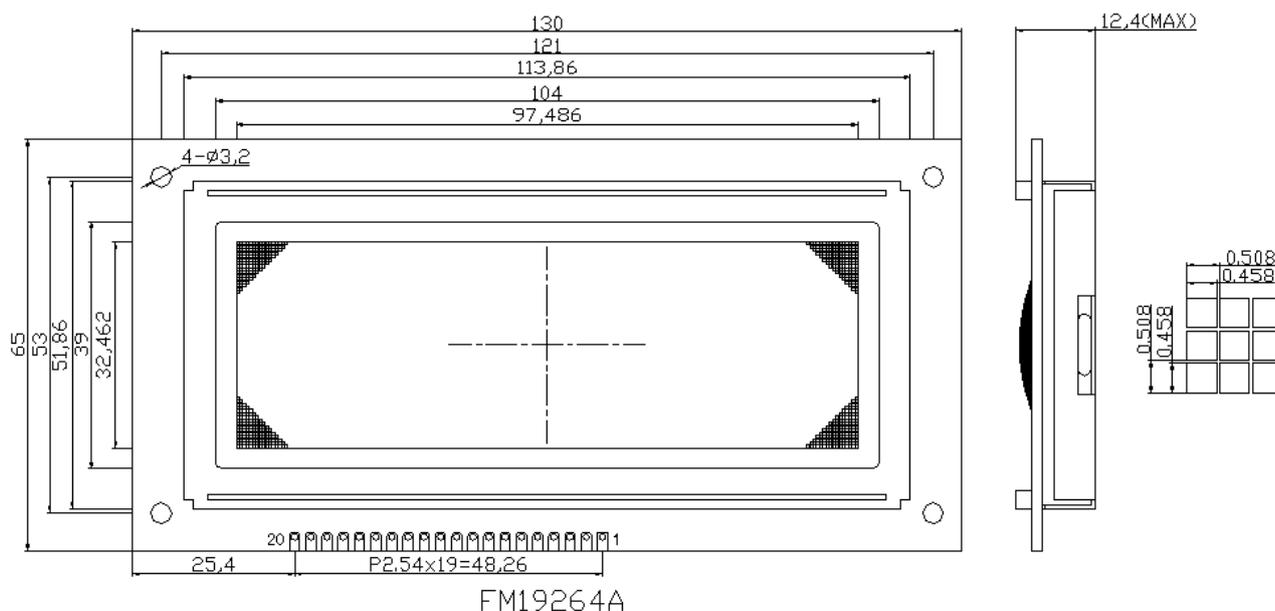
ZX19264A 是一种图形点阵液晶显示器。它主要采用动态驱动原理由行驱动—控制器和列驱动器两部分组成了 192(列) × 64(行)的全点阵液晶显示。此显示器采用了 COB 的软封装方式, 通过导电橡胶和压框连接 LCD, 使其寿命长, 连接可靠。

## 二. 特性

1. 工作电压为  $+5V \pm 10\%$  , 可自带驱动 LCD 所需的负电压。
2. 全屏幕点阵, 点阵数为 192(列) × 64(行), 可显示 12(/行) × 4(行) 个 (16 × 16 点阵) 汉字, 也可完成图形, 字符的显示。
3. 与 CPU 接口采用 5 条位控制总线和 8 位并行数据总线输入输出, 适配 M6800 系列时序。
4. 内部有显示数据锁存器
5. 简单的操作指令 显示开关设置, 显示起始行设置, 地址指针设置和数据读/写等指令。

## 三. 外形尺寸

### 1. 外形尺寸图



## 2. 主要外形尺寸

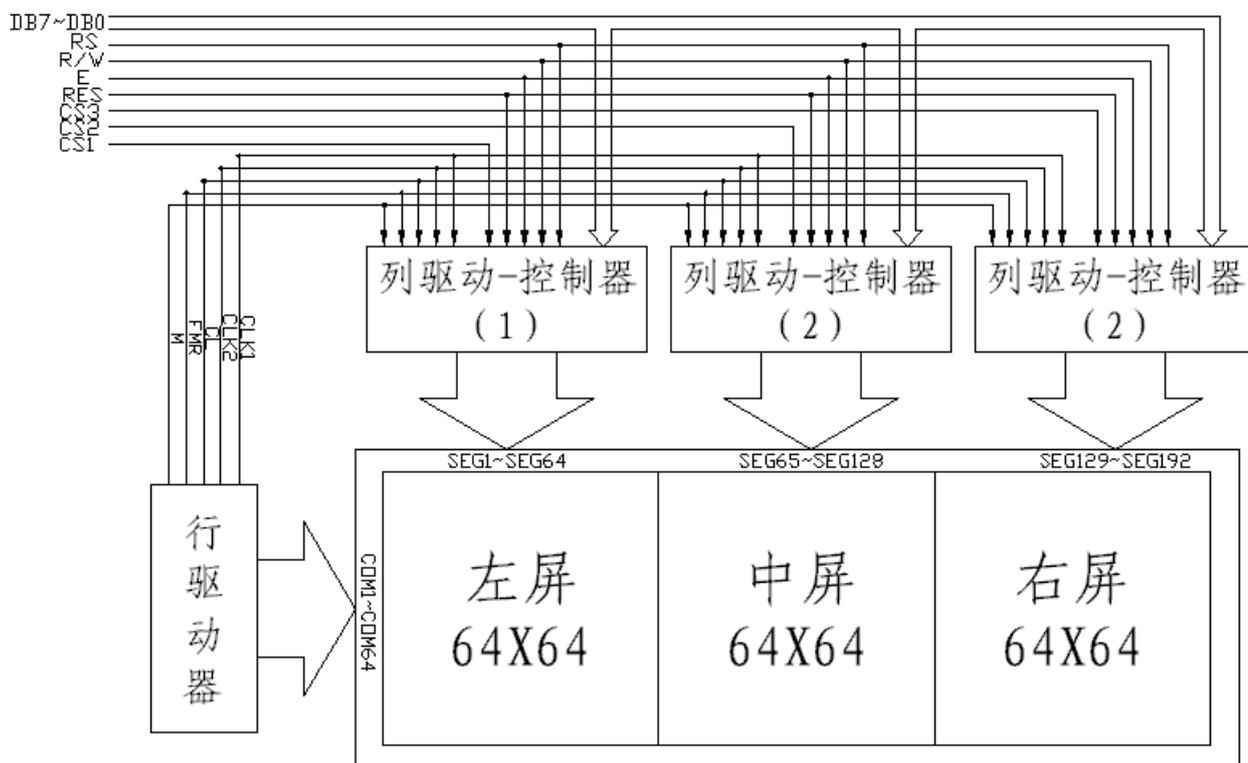
项 目	标 准 尺 寸	单 位
模 块 体 积	130.0 × 65.0 × 12.4	mm
定 位 尺 寸	121.0 × 53.0	mm
视 域	97.486 × 32.462	mm
行 列 点 阵 数	192 × 64	dots
点 距 离	0.508 × 0.508	mm
点 大 小	0.458 × 0.458	mm

## 四. 硬件说明

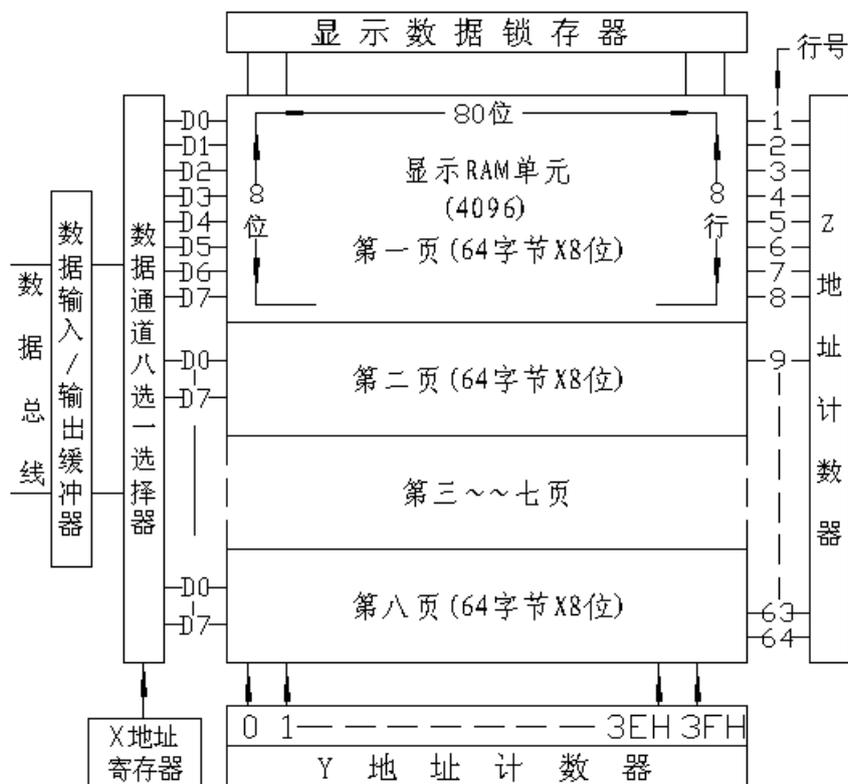
## 1. 引脚特性

引脚号	引脚名称	级 别	引 脚 功 能 描 述
1	VSS	0V	电源地
2	VDD	+5V	电源电压
3	VLCD	0 ~ -10V	对比度调节输入参考电压
4	RS	H/L	寄存器选择信号
5	R/W	H/L	读/写操作选择信号
6	E	H/L	使能信号
7	DB0	H/L	八位三态并行数据总线
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	CS1	H/L	片选信号, 当 CS1=L 时, 液晶左屏显示
16	/RES	H/L	复位信号, 低有效
17	CS2	H/L	片选信号, 当 CS2=L 时, 液晶中屏显示
18	CS3	H/L	片选信号, 当 CS3=L 时, 液晶右屏显示
19	VEE		输出负电压(单电源供电)
20	LED+ (EL)	+5V	LED 电源电压, $I_{dd} \leq \text{mA}$

### 2. 原理简图



### 3. 主要各部分详解



### 1) 显示数据 RAM (DDRAM)

DDRAM ( $64 \times 8 \times 8$  bits) 是存储图形显示数据的。此 RAM 的每一位数据对应显示面板上一个点的显示 (数据为 H) 与不显示 (数据为 L)。DDRAM 的地址与显示位置关系对照图 (见附录一)

### 2) I/O 缓冲器 (DB0 ~ DB7)

I/O 缓冲器为双向三态数据缓冲器。是 LCM (液晶显示模块) 内部总线与 MPU 总线的结合部。其作用是将两个不同时钟下工作的系统连接起来, 实现通讯。I/O 缓冲器在片选信号 CS 有效状态下, I/O 缓冲器开放, 实现 LCM (液晶显示模块) 与 MPU 之间的数据传递。当片选信号为无效状态时, I/O 缓冲器将中断 LCM (液晶显示模块) 内部总线与 MPU 数据总线的联系, 对外总线呈高阻状态, 从而不影响 MPU 的其他数据操作功能。

### 3) 输入寄存器

输入寄存器用于接收在 MPU 运行速度下传送给 LCM (液晶显示模块) 的数据并将其锁存在输入寄存器内, 其输出将在 LCM (液晶显示模块) 内部工作时钟的运作下将数据写入指令寄存器或显示存储器内。

### 4) 输出寄存器

输出寄存器用于暂存从显示存储器读出的数据, 在 MPU 读操作时, 输出寄存器将当前锁存的数据通过 I/O 缓冲器送入 MPU 数据总线上。

### 5) 指令寄存器

指令寄存器用于接收 MPU 发来的指令代码, 通过译码将指令代码置入相关的寄存器或触发器内。

### 6) 状态字寄存器

状态字寄存器是 LCM (液晶显示模块) 与 MPU 通讯时唯一的“握手”信号。状态字寄存器向 MPU 表示了 LCM (液晶显示模块) 当前的工作状态。尤其是状态字中的“忙”标志位是 MPU 在每次对 LCM (液晶显示模块) 访问时必须读出判别的状态位。当处于“忙”标志位时, I/O 缓冲器被封锁, 此时 MPU 对 LCM (液晶显示模块) 的任何操作 (除读状态字操作外) 都将是无效的。

### 7) X 地址寄存器

X 地址寄存器是一个三位页地址寄存器, 其输出控制着 DDRAM 中 8 个页面的选择, 也是控制着数据传输通道的八选一选择器。X 地址寄存器可以由 MPU 以指令形式设置。X 地址寄存器没有自动修改功能, 所以要想转换页面需要重新设置 X 地址寄存器的内容。

### 8) Y 地址计数器

Y 地址计数器是一个 6 位循环加一计数器。它管理某一页面上的 64 个单元。Y 地址计数器可以由 MPU 以指令形式设置, 它和页地址指针结合唯一选通显示存储器的一个单元, Y 地址计数器具有自动加一功能。在显示存储器读/写操作后 Y 地址计数将自动加一。当计数器加至 3FH 后循环归零再继续加一。

### 9) Z 地址计数器

Z 地址计数器是一个 6 位地址计数器, 用于确定当前显示行的扫描地址。Z 地址计数器具有自动加一功能。它与行驱动器的行扫描输出同步, 选择相应的列驱动的数据输出。

## 10) 显示起始行寄存器

显示起始行寄存器是一个 6 位寄存器，它规定了显示存储器所对应显示屏上第一行的行号。该行的数据将作为显示屏上第一行显示状态的控制信号。

## 11) 显示开/关触发器

显示开/关触发器的作用就是控制显示驱动输出的电平以控制显示屏的开关。在触发器输出为“关”电平时，显示数据锁存器的输入被封锁并将输出置“0”，从而使显示驱动输出全部为非选择波形，显示屏呈不显示状态。在触发器输出为“开”电平时，显示数据锁存器被控制，显示驱动输出受显示驱动数据总线上数据控制，显示屏将呈显示状态。

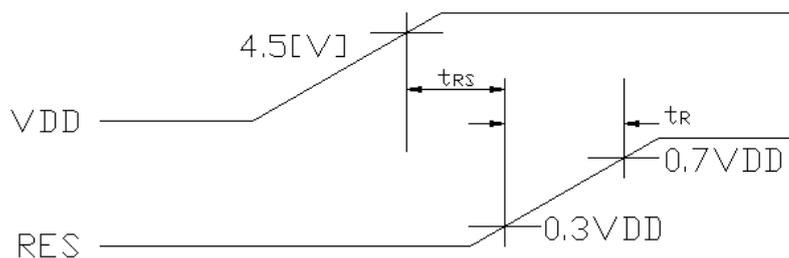
## 12) 复位端/RES

复位端/RES 用于在 LCM (液晶显示模块) 上电时或需要时实现硬件电路对 LCM (液晶显示模块) 的复位。该复位功能将实现：

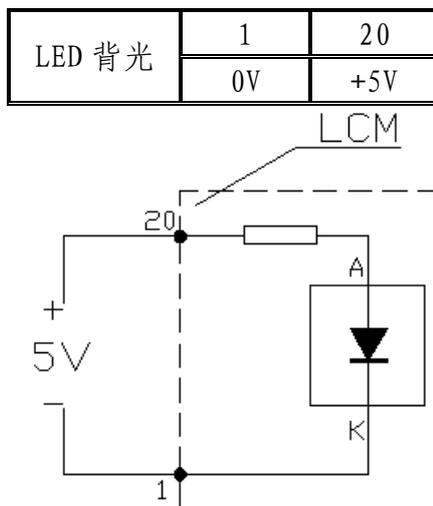
- 设置显示状态为关显示状态
- 显示起始寄存器清零。显示 RAM 第一行对应显示屏上的第一行。
- 在复位期间状态字中 RESET 位置“1”。

初始化条件：

项目	名称	最小值	标准值	最大值	单位
Reset Time	$t_{RS}$	1.0	-	-	us
Rise Time	$t_R$	-	-	200	ns



## 4. 背光接线图



## 五. 电气特性

### 1. 限定参数

项 目	名称	值	单位	备注
Operating Voltage	VDD	-0.3 to +5.5	V	*1
Supply Voltage	VEE	VDD-19.0 to VDD+0.3	V	*2
Driver Supply Voltage	V <sub>B</sub>	-0.3 to VDD+0.3	V	*1, *3
Operating Temperature	T <sub>OPR</sub>	-20 to +70	°C	
Storage Temperature	T <sub>STG</sub>	-30 to +80	°C	

\*1. Based on VSS=0V

\*2. Applies to V<sub>LCD</sub>

\*3. Applies to CS, E, R/W, RS, DB0 ~ DB7

### 2. 直流特性 (VDD=+5V ± 10%, VSS=0V, VDD-VLCD=8 ~ 17V, Ta=-20 ~ +70°C)

项 目	名称	测试条件	Min	Typ	Max	单位	备注
Input High Voltage	V <sub>IH</sub>	-	2.0	-	VDD	V	*1
Input Low Voltage	V <sub>IL</sub>	-	0	-	0.8	V	*1
Output High Voltage	V <sub>OH</sub>	I <sub>OH</sub> =-200uA	2.4	-	-	V	*2
Output Low Voltage	V <sub>OL</sub>	I <sub>OL</sub> =1.6mA		-	0.4	V	*2
Input Leakage Current	I <sub>LKG</sub>	V <sub>IN</sub> =VSS ~ VDD	-1.0	-	1.0	uA	*3
Three-state(OFF) input Current	I <sub>TSL</sub>	V <sub>IN</sub> =VSS ~ VDD	-5.0	-	5.0	uA	*4
Operating Current	I <sub>DD1</sub>	During Display	-	-	0.5	mA	*5
	I <sub>DD2</sub>	During Access			2	mA	*5
On Resistance	R <sub>ON</sub>		-	-	7.5	KΩ	*6

\*1. CS, E, RW, RS, DB0 ~ DB7

\*2. DB0 ~ DB7

\*3. Except DB0 ~ DB7

\*4. DB0 ~ DB7 at High Impedance

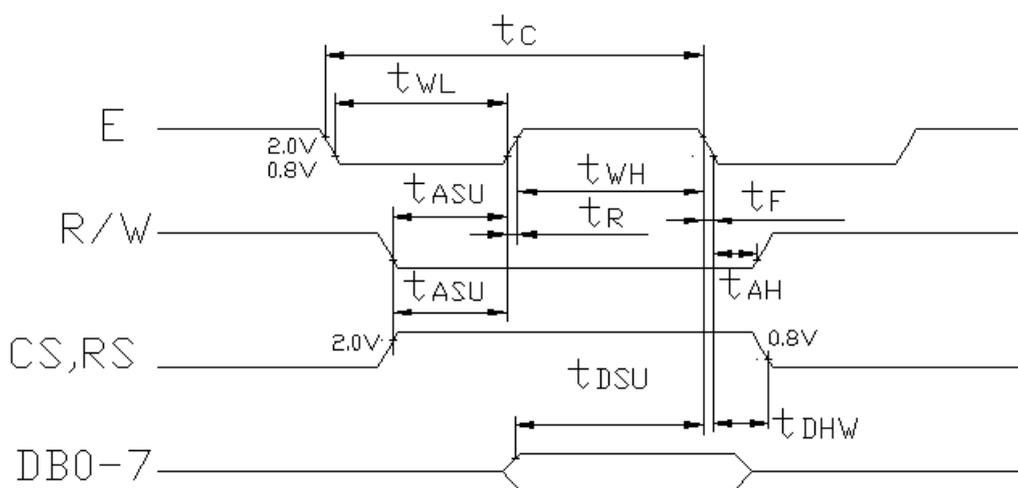
\*5. 1/64 duty, FCLK=250KHZ, Frame Frequency=70HZ, Output: NO Load

\*6. VDD ~ VEE=15.5

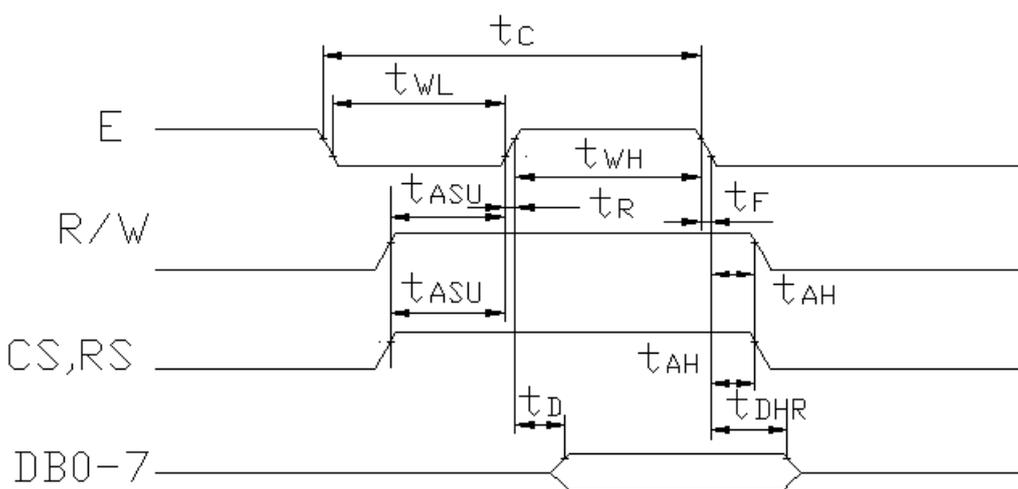
### 3. 交流特性 (VDD=+5V ± 10%, VSS=0V, Ta=-20 ~ +70°C)

项 目	名称	Min	Typ	Max	单 位
E Cycle	t <sub>C</sub>	1000	-	-	ns
E Hight Level Width	t <sub>WH</sub>	450	-	-	ns
E Low Level Width	t <sub>WL</sub>	450	-		ns
E Rise Time	t <sub>R</sub>	-	-	25	ns
E Fall Time	t <sub>F</sub>	-		25	ns

项 目	名 称	Min	Typ	Max	单 位
Address Set-up Time	$t_{ASH}$	140	-	-	ns
Address Hold Time	$t_{AH}$	10	-	-	ns
Data Set-up Time	$t_{DSU}$	200	-	-	ns
Data Delay Time	$t_D$	-	-	320	ns
Data Delay Time	$t_{DHW}$	10			ns
Data Delay Time	$t_{DHR}$	20			ns



MPU Write timing



MPU Read timing

## 四. 软件说明

### 1. 指令表

指令名称	控制信号		控制代码							
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
显示开关设置	0	0	0	0	1	1	1	1	1	D
显示起始行设置	0	0	1	1	L5	L4	L3	L2	L1	L0
页面地址设置	0	0	1	0	1	1	1	P2	P1	P0
列地址设置	0	0	0	1	C5	C4	C3	C2	C1	C0
读取状态字	0	1	BUSY	0	ON/OFF	RESET	0	0	0	0
写显示数据	1	0	数 据							
读显示数据	1	1	数 据							

详细解释各个指令功能

#### 1) 读状态字

格式

BUSY	0	ON/OFF	RESET	0	0	0	0
------	---	--------	-------	---	---	---	---

状态字是 MPU 了解 LCM（液晶显示模块）当前状态，或 LCM 向 MPU 提供其内部状态的唯一的信息渠道。

BUSY 表示当前 LCM 接口控制电路运行状态。BUSY=1 表示 LCM 正在处理 MPU 发过来的指令或数据。此时接口电路被封锁，不能接受除读状态字以外的任何操作。BUSY=0 表示 LCM 接口控制电路已外于“准备好”状态，等待 MPU 的访问。

ON/OFF 表示当前的显示状态。ON/OFF=1 表示关显示状态，ON/OFF=0 表示开显示状态。

RESET 表示当前 LCM 的工作状态，即反映/RES 端的电平状态。当/RES 为低电平状态时，LCM 处于复位工作状态，标志位 RESET=1。当/RES 为高电平状态时，LCM 为正常工作状态，标志位 RESET=0。

在指令设置和数据读写时要注意状态字中的 BUSY 标志。只有在 BUSY=0 时，MPU 对 LCM 的操作才能有效。因此 MPU 在每次对 LCM 操作之前，都要读出状态字判断 BUSY 是否为“0”。若不为“0”，则 MPU 需要等待，直至 BUSY=0 为止。

#### 2) 显示开关设置

格式

0	0	1	1	1	1	1	D
---	---	---	---	---	---	---	---

该指令设置显示开/关触发器的状态，由此控制显示数据锁存器的工作方式，从而控制显示屏上的显示状态。D 位为显示开/关的控制位。当 D=1 为开显示设置，显示数据锁存器正常工作，显示屏上呈现所需的显示效果。此时在状态字中 ON/OFF=0。当 D=0 为关显示设置，显示数据锁存器被置零，显示屏呈不显示状态，但显示存储器并没有被破坏，在状态字中 ON/OFF=1。

## 3) 显示起始行设置

格式	1	1	L5	L4	L3	L2	L1	L0
----	---	---	----	----	----	----	----	----

该指令设置了显示起始行寄存器的内容。LCM 通过 CS 的选择分别具有 64 行显示的管理能力，该指令中 L5~L0 为显示起始行的地址，取值在 0~3FH (1~64 行) 范围内，它规定了显示屏上最顶一行所对应的显示存储器的行地址。如果定时间隔地，等间距地修改（如加一或减一）显示起始行寄存器的内容，则显示屏将呈现显示内容向上或向下平滑滚动的显示效果。

## 4) 页面地址设置

格式	1	0	1	1	1	P2	P1	P0
----	---	---	---	---	---	----	----	----

该指令设置了页面地址—X 地址寄存器的内容。LCM 将显示存储器分成 8 页，指令代码中 P2~P0 就是要确定当前所要选择的页面地址，取值范围为 0~7H，代表第 1~8 页。该指令规定了以后的读/写操作将在哪一个页面上进行。

## 5) 列地址设置

格式	0	1	C5	C4	C3	C2	C1	C0
----	---	---	----	----	----	----	----	----

该指令设置了 Y 地址数计数器的内容，LCM 通过 CS 的选择分别具有 64 列显示的管理能力，C5~C0=0~3FH (1~64) 代表某一页面上的某一单元地址，随后的一次读或写数据将在这个单元上进行。Y 地址计数器具有自动加一功能，在每一次读/写数据后它将自动加一，所以在连续进行读/写数据时，Y 地址计数器不必每次都设置一次。

页面地址的设置和列地址的设置将显示存储器单元唯一地确定下来，为后来的显示数据的读/写作了地址的选通。

## 6) 写显示数据

格式		数					据	
----	--	---	--	--	--	--	---	--

该操作将 8 位数据写入先前已确定的显示存储器的单元内。操作完成后列地址计数器自动加一。

## 7) 读显示数据

格式		数					据	
----	--	---	--	--	--	--	---	--

该操作将 LCM 接口部的输出寄存器内容读出，然后列地址计数器自动加一。

## 2. 控制时序表

CS1	CS2	CS3	RS	R/W	E	DB7~DB0	功能
X	X	X	X	X	0	高阻	总线释放
0	0	0	0	0	下降沿	输入	写指令代码

CS1	CS2	CS3	RS	R/W	E	DB7 ~ DB0	功能
0	0	0	0	1	1	输出	读状态字
0	0	0	1	0	下降沿	输入	写显示数据
0	0	0	1	1	1	输出	读显示数据

### 3. DDRAM 地址表

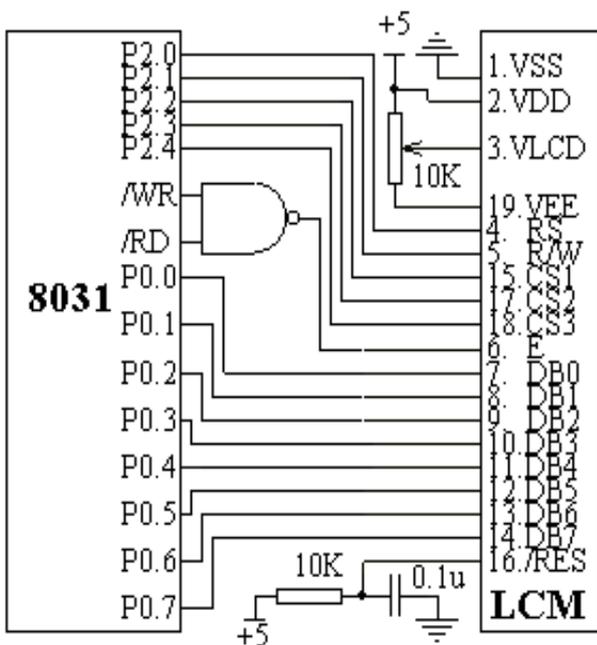
CS1=1						CS2=1					CS3=1					行号
Y=	0	1	..	62	63	0	1	..	62	63	0	1	..	62	63	
X=0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	0
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	7
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	8
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	55
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	56
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	63
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓

### 4. LCM 与 MPU 接口及驱动程序

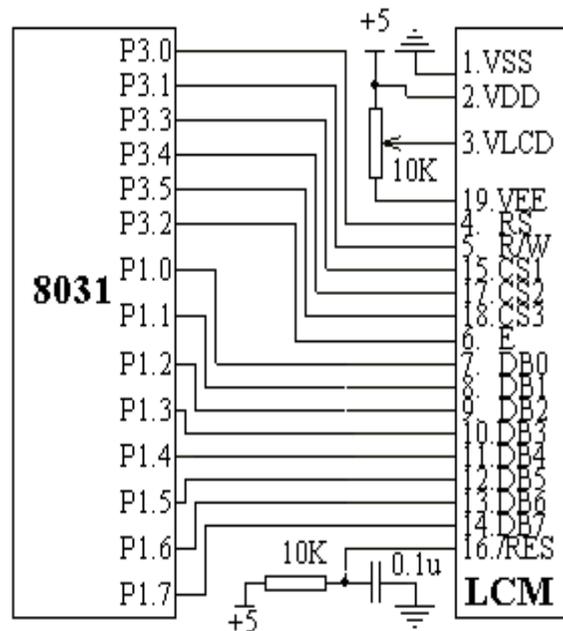
FM19264A 图形液晶显示模块与 MPU 的连接方式有两种：一种为直接访问方式，一种为间接控制方式。

#### 1) 接口电路(以 8031 为例)

##### 直接访问方式



注：双电源负压直接由 3. VLCD 引入  
间接访问方式



注：双电源负压直接由 3. VLCD 引入

## 2) 驱动程序(以 8031 汇编为例)

## 直接访问方式

A12=CS3, A11=CS2, A10=CS1, A9=R/W, A8=RS

COM EQU 20H ; 指令寄存器

DAT EQU 21H ; 数据寄存器

CWADD1 EQU 0400H ; 写指令代码地址左

CRADD1 EQU 0600H ; 读状态字地址左

DWADD1 EQU 0500H ; 写显示数据地址左

DRADD1 EQU 0700H ; 读显示数据地址左

CWADD2 EQU 0800H ; 写指令代码地址中

CRADD2 EQU 0A00H ; 读状态字地址中

DWADD2 EQU 0900H ; 写显示数据地址中

DRADD2 EQU 0B00H ; 读显示数据地址中

CWADD3 EQU 1000H ; 写指令代码地址右

CRADD3 EQU 1200H ; 读状态字地址右

DWADD3 EQU 1100H ; 写显示数据地址右

DRADD3 EQU 1300H ; 读显示数据地址右

## 1. 左区驱动子程序

## 1) 写指令代码子程序(左)

PRL0: PUSH DPL

PUSH DPH

MOV DPTR, #CRADD1

; 设置读状态字地址

PRL01: MOVX A, @DPTR ; 读状态字

JB ACC. 7, PRL01

; 判“忙”标志为“0”否, 否再读

MOV DPTR, #CWADD1

; 设置写指令代码地址

MOV A, COM ; 取指令代码

MOVX @DPTR, A ; 写指令代码

POP DPH

POP DPL

RET

## 2) 写显示数据子程序(左)

PRL1: PUSH DPL

PUSH DPH

MOV DPTR, #CRADD1

; 设置读状态字地址

PRL11: MOVX A, @DPTR ; 读状态字

JB ACC. 7, PRL11

## 间接访问方式

CS1 EQU P3.3 ; 片选左

CS2 EQU P3.4 ; 片选中

CS3 EQU P3.5 ; 片选右

RS EQU P3.0 ; 寄存器选择信号

RW EQU P3.1 ; 读/写选择信号

E EQU P3.2 ; 使能信号

## 1. 左区驱动子程序

## 1) 写指令代码子程序(左)

PRL0: SETB CS1

CLR CS2

CLR CS3

CLR RS ; RS=0

SETB RW ; R/W=1

PRL01: MOV P1, #0FFH ; P1 口置“1”

SETB E ; E=1

MOV A, P1 ; 读状态字

CLR E ; E=0

JB ACC. 7, PRL01

; 判“忙”标志为“0”否, 否再读

CLR RW ; R/W=0

MOV P1, COM ; 写指令代码

SETB E ; E=1

CLR E ; E=0

RET

## 2) 写显示数据子程序(左)

PRL1: SETB CS1

CLR CS2

CLR CS3

CLR RS ; RS=0

SETB RW ; R/W=1

PRL11: MOV P1, #0FFH ; P1 口置“1”

SETB E ; E=1

MOV A, P1 ; 读状态字

CLR E ; E=0

JB ACC. 7, PRL11

; 判“忙”标志为“0”否, 否再读

SETB RS ; RS=1

CLR RW ; R/W=0

```

; 判“忙”标志为“0”否, 否再读
MOV DPTR, #DWADD1
; 设置写显示数据地址
MOV A, DAT ; 取数据
MOVX @DPTR, A ; 写数据
POP DPH
POP DPL
RET

```

### 3) 读显示数据子程序 (左)

```

PRL2: PUSH DPL
      PUSH DPH
      MOV DPTR, #CRADD1
; 设置读状态字地址
PRL21: MOVX A, @DPTR ; 读状态字
        JB ACC. 7, PRL21
; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #DRADD1
; 设置读显示数据地址
        MOVX A, @DPTR ; 读数据
        MOV DAT, A ; 存数据
        POP DPH
        POP DPL
        RET

```

## 2. 中区驱动子程序

### 1) 写指令代码子程序 (中)

```

PRM0: PUSH DPL
      PUSH DPH
      MOV DPTR, #CRADD2
; 设置读状态字地址
PRM01: MOVX A, @DPTR ; 读状态字
        JB ACC. 7, PRM01
; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #CWADD2
; 设置写指令代码地址
        MOV A, COM ; 取指令代码
        MOVX @DPTR, A ; 写指令代码
        POP DPH
        POP DPL

        MOV P1, DAT ; 写数据

```

```

SETB E ; E=1
CLR E ; E=0
RET

```

### 3) 读显示数据子程序 (左)

```

PRL2: SETB CS1
      CLR CS2
      CLR CS3
      CLR RS ; RS=0
      SETB RW ; R/W=1
PRL21: MOV P1, #0FFH ; P1口置“1”
        SETB E ; E=1
        MOV A, P1 ; 读状态字
        CLR E ; E=0
        JB ACC. 7, PRL21
; 判“忙”标志为“0”否, 否再读
        SETB RS ; RS=1
        MOV P1, #0FFH ; P1口置“1”
        SETB E ; E=1
        MOV DAT, P1 ; 写数据
        CLR E ; E=0
        RET

```

## 2. 中区驱动子程序

### 1) 写指令代码子程序 (中)

```

PRM0: CLR CS1
      SETB CS2
      CLR CS3
      CLR RS ; RS=0
      SETB RW ; R/W=1
PRM01: MOV P1, #0FFH ; P1口置“1”
        SETB E ; E=1
        MOV A, P1 ; 读状态字
        CLR E ; E=0
        JB ACC. 7, PRM01
; 判“忙”标志为“0”否, 否再读
        CLR RW ; R/W=0
        MOV P1, COM ; 写指令代码
        SETB E ; E=1

```

```

RET
-----
2) 写显示数据子程序 (中)
PRM1: PUSH DPL
      PUSH DPH
      MOV DPTR, #CRADD2
                ; 设置读状态字地址
PRM11: MOVX A, @DPTR      ; 读状态字
        JB ACC.7, PRM11
                ; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #DWADD2
                ; 设置写显示数据地址
        MOV A, DAT        ; 取数据
        MOVX @DPTR, A    ; 写数据
        POP DPH
        POP DPL
        RET
-----

```

```

3) 读显示数据子程序 (中)
PRM2: PUSH DPL
      PUSH DPH
      MOV DPTR, #CRADD2
                ; 设置读状态字地址
PRM21: MOVX A, @DPTR      ; 读状态字
        JB ACC.7, PRM21
                ; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #DRADD2
                ; 设置读显示数据地址
        MOVX A, @DPTR      ; 读数据
        MOV DAT, A        ; 存数据
        POP DPH
        POP DPL
        RET
-----

```

### 3. 右区驱动子程序

#### 1) 写指令代码子程序 (右)

```

PRR0: PUSH DPL
      PUSH DPH
      MOV DPTR, #CRADD3
                ; 设置读状态字地址

      CLR E                ; E=0
-----

```

```

RET
-----
2) 写显示数据子程序 (中)
PRM1: CLR CS1
      SETB CS2
      CLR CS3
      CLR RS                ; RS=0
      SETB RW              ; R/W=1
PRM11: MOV P1, #0FFH      ; P1 口置“1”
        SETB E                ; E=1
        MOV A, P1           ; 读状态字
        CLR E                ; E=0
        JB ACC.7, PRM11
                ; 判“忙”标志为“0”否, 否再读
        SETB RS              ; RS=1
        CLR RW              ; R/W=0
        MOV P1, DAT        ; 写数据
        SETB E                ; E=1
        CLR E                ; E=0
        RET
-----

```

```

3) 读显示数据子程序 (中)
PRM2: CLR CS1
      SETB CS2
      CLR CS3
      CLR RS                ; RS=0
      SETB RW              ; R/W=1
PRM21: MOV P1, #0FFH      ; P1 口置“1”
        SETB E                ; E=1
        MOV A, P1           ; 读状态字
        CLR E                ; E=0
        JB ACC.7, PRM21
                ; 判“忙”标志为“0”否, 否再读
        SETB RS              ; RS=1
        MOV P1, #0FFH      ; P1 口置“1”
        SETB E                ; E=1
        MOV DAT, P1        ; 写数据
        CLR E                ; E=0
        RET
-----

```

```

PRR01: MOVX A, @DPTR      ; 读状态字
        JB ACC. 7, PRR01
           ; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #CWADD3
           ; 设置写指令代码地址
        MOV A, COM        ; 取指令代码
        MOVX @DPTR, A    ; 写指令代码
        POP DPH
        POP DPL
        RET

```

### 2) 写显示数据子程序 (右)

```

PRR1:  PUSH DPL
        PUSH DPH
        MOV DPTR, #CRADD3
           ; 设置读状态字地址
PRR11: MOVX A, @DPTR     ; 读状态字
        JB ACC. 7, PRR11
           ; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #DWADD3
           ; 设置写显示数据地址
        MOV A, DAT       ; 取数据
        MOVX @DPTR, A   ; 写数据
        POP DPH
        POP DPL
        RET

```

### 3) 读显示数据子程序 (右)

```

PRR2:  PUSH DPL
        PUSH DPH
        MOV DPTR, #CRADD3
           ; 设置读状态字地址
PRR21: MOVX A, @DPTR     ; 读状态字
        JB ACC. 7, PRR21
           ; 判“忙”标志为“0”否, 否再读
        MOV DPTR, #DRADD3
           ; 设置读显示数据地址
        MOVX A, @DPTR   ; 读数据
        MOV DAT, A     ; 存数据
        POP DPH
        POP DPL

```

### 3. 右区驱动子程序

#### 1) 写指令代码子程序 (右)

```

PRR0:  CLR CS1
        CLR CS2
        SETB CS3
        CLR RS      ; RS=0
        SETB RW     ; R/W=1
PRR01: MOV P1, #0FFH ; P1口置“1”
        SETB E      ; E=1
        MOV A, P1   ; 读状态字
        CLR E       ; E=0
        JB ACC. 7, PRR01
           ; 判“忙”标志为“0”否, 否再读
        CLR RW     ; R/W=0
        MOV P1, COM ; 写指令代码
        SETB E     ; E=1
        CLR E      ; E=0
        RET

```

#### 2) 写显示数据子程序 (右)

```

PRR1:  CLR CS1
        CLR CS2
        SETB CS3
        CLR RS      ; RS=0
        SETB RW     ; R/W=1
PRR11: MOV P1, #0FFH ; P1口置“1”
        SETB E      ; E=1
        MOV A, P1   ; 读状态字
        CLR E       ; E=0
        JB ACC. 7, PRR11
           ; 判“忙”标志为“0”否, 否再读
        SETB RS     ; RS=1
        CLR RW     ; R/W=0
        MOV P1, DAT ; 写数据
        SETB E     ; E=1
        CLR E      ; E=0
        RET

```

#### 3) 读显示数据子程序 (左)

```

PRR2:  CLR CS1

```

```
RET
-----
-----
```

```
CLR CS2
SETB CS3
CLR RS ; RS=0
SETB RW ; R/W=1
PRR21: MOV P1, #0FFH ; P1 口置 "1"
SETB E ; E=1
MOV A, P1 ; 读状态字
CLR E ; E=0
JB ACC.7, PRR21
; 判"忙"标志为"0"否, 否再读
SETB RS ; RS=1
MOV P1, #0FFH ; P1 口置 "1"
SETB E ; E=1
MOV DAT, P1 ; 写数据
CLR E ; E=0
RET
-----
-----
```

### 3) 举例程序(以 8031 汇编为例)

```
ORG 0000H
LJMP INT
ORG 100H
INT: MOV COM, #0C0H ; 设置显示起始行为第一行
LCALL PRL0
LCALL PRM0
LCALL PRR0
MOV COM, #3FH ; 开显示设置
LCALL PRL0
LCALL PRM0
LCALL PRR0
CLEAR: MOV R4, #00H ; 页面地址暂存器
MOV DPTR, #CCW0
CLEAR1: MOV A, R4
ORL A, #0B8H ; "或" 页面地址设置代码
MOV COM, A ; 页面地址设置
LCALL PRL0
LCALL PRM0
LCALL PRR0
MOV COM, #40H ; 列地址设置为 "0"
```

```
LCALL PRL0
LCALL PRM0
LCALL PRR0
MOV R3, #10H           ; 显示 10 列
CLEAR2: MOV A, #00H    ; 显示
        MOVC A, @A+DPTR
        MOV DAT, A
        LCALL PRL1
        LCALL PRM1
        LCALL PRR1
        INC DPTR
        INC DPTR
        DJNZ R3, CLEAR2
        MOV DPTR, #CCW0
        INC DPTR
        INC R4
        CJNE R4, #02H, CLEAR1
        MOV DPTR, #CCW1
CLEAR11: MOV A, R4
        ORL A, #0B8H
        MOV COM, A
        LCALL PRL0
        LCALL PRM0
        LCALL PRR0
        MOV COM, #40H
        LCALL PRL0
        LCALL PRM0
        LCALL PRR0
        MOV R3, #10H
CLEAR21: MOV A, #00H   ; 显示
        MOVC A, @A+DPTR
        MOV DAT, A
        LCALL PRL1
        LCALL PRM1
        LCALL PRR1
        INC DPTR
        INC DPTR
        DJNZ R3, CLEAR21
        MOV DPTR, #CCW1
        INC DPTR
        INC R4
```

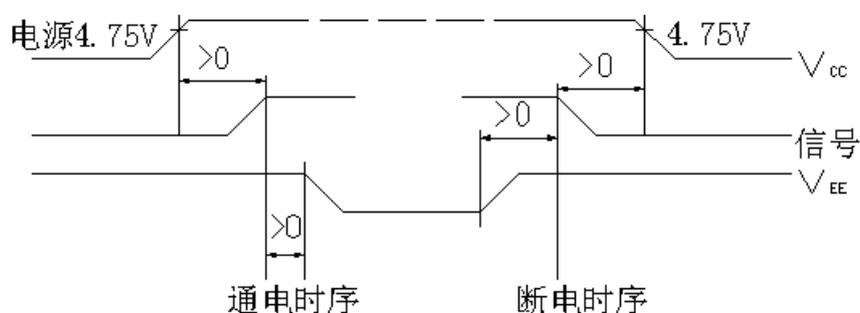
```
CJNE R4, #04H, CLEAR11
MOV DPTR, #CCW2
CLEAR12: MOV A, R4
        ORL A, #0B8H
        MOV COM, A
        LCALL PRL0
        LCALL PRM0
        LCALL PRR0
        MOV COM, #40H
        LCALL PRL0
        LCALL PRM0
        LCALL PRR0
        MOV R3, #10H
CLEAR22: MOV A, #00H           ; 显示“液”
        MOVC A, @A+DPTR
        MOV DAT, A
        LCALL PRL1
        LCALL PRM1
        LCALL PRR1
        INC DPTR
        INC DPTR
        DJNZ R3, CLEAR22
        MOV DPTR, #CCW2
        INC DPTR
        INC R4
        CJNE R4, #06H, CLEAR12
        MOV DPTR, #CCW3
CLEAR13: MOV A, R4
        ORL A, #0B8H
        MOV COM, A
        LCALL PRL0
        LCALL PRM0
        LCALL PRR0
        MOV COM, #40H
        LCALL PRL0
        LCALL PRM0
        LCALL PRR0
        MOV R3, #10H
CLEAR23: MOV A, #00H           ; 显示“晶”
        MOVC A, @A+DPTR
        MOV DAT, A
```

```
LCALL PRL1
LCALL PRM1
LCALL PRR1
INC DPTR
INC DPTR
DJNZ R3, CLEAR23
MOV DPTR, #CCW3
INC DPTR
INC R4
CJNE R4, #08H, CLEAR13
LJMP INT
NOP
```

```
CCW0: DB 010H, 004H, 021H, 0FEH, 0C6H, 001H, 030H, 008H, 0F4H, 009H, 054H, 009H, 05FH, 0FFH, 054H, 009H
      DB 0F4H, 049H, 000H, 020H, 0FEH, 01FH, 022H, 041H, 022H, 081H, 0FEH, 07FH, 000H, 000H, 000H, 000H ; 潮
CCW1: DB 000H, 004H, 000H, 004H, 048H, 004H, 048H, 004H, 048H, 004H, 048H, 004H, 048H, 004H, 0FFH, 0FFH
      DB 048H, 004H, 048H, 004H, 048H, 004H, 068H, 004H, 04CH, 004H, 008H, 006H, 000H, 004H, 000H, 000H ; 丰
CCW2: DB 010H, 004H, 061H, 004H, 006H, 0FFH, 0E0H, 000H, 018H, 001H, 084H, 000H, 0E4H, 0FFH, 01CH, 041H
      DB 084H, 021H, 065H, 012H, 0BEH, 00CH, 024H, 01BH, 0A4H, 061H, 064H, 0C0H, 004H, 040H, 000H, 000H ; 液
CCW3: DB 000H, 000H, 000H, 07FH, 000H, 025H, 000H, 025H, 07EH, 025H, 02AH, 025H, 02AH, 07FH, 02AH, 000H
      DB 02AH, 000H, 02AH, 07FH, 02AH, 025H, 07EH, 025H, 000H, 025H, 000H, 025H, 000H, 07FH, 000H, 000H ; 晶
      END
```

## 五. 液晶显示模块使用注意事项

1. 请勿随意自行加工、整修、拆卸。
2. 避免对液晶屏表面施加压力。
3. 不要用手随意去摸外引线、电路板上的电路及金属框。
4. 如必须直接接触时，应使人体与模块保持同一电位，或将人体良好接地。
5. 焊接使用的烙铁、操作用的电动改锥等工具必须良好接地，没漏电。
6. 严防各种静电。
7. 模块使用接入电源及断开电源时，必须按图时序进行。即必须在正电源（ $5 \pm 0.25V$ ）稳定接入后，才能输入信号电平。如在电源稳定接入前，或断开后就输入信号电平，将会损坏模块中的集成电路，使模块损坏。



8. 点阵模块在调节时，应调整 VEE 至最佳对比度、视角时为止。如果 VEE 调整过高，不仅会影响显示，还会缩短液晶的寿命。
9. 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。
10. 模块要存储在暗处（避阳光），温度在  $-10^{\circ}\text{C} \sim +35^{\circ}\text{C}$ ，湿度在 RH60% 以上的地方。如能装入聚乙烯口袋（最好有防静电涂层）并将口封住最好。

以上使用说明由北京中显电子有限公司编制，有问题请电话联络，我们将竭诚为您服务，同时，提供完善的保修服务！因为每种液晶使用的控制器都不一样，控制器的型号基本就决定了液晶的指令形式和使用方式，所以，在说明书里一般不会详细照搬控制器说明书的每个细节，只会简要介绍常用指令，如果需要了解详细的指令和具体电气参数，请参照 [WWW.ZXLCD.COM](http://WWW.ZXLCD.COM) 网站里的“技术支持”菜单下，均有对应控制器手册免费下载，直接对应现有各类液晶使用的各种控制器，使用手册里一般有具体电气参数说明，指令详细介绍，同时辅以编程实例，以便客户详细参照，同时提高编程及操作技巧。

服务电话：010-52926620, 82626833

公司地址：北京市中关村大街 32 号蓝天和盛大厦 811 室

# 附 录 一

