



字符点阵液晶显示模块

---

# 字符系列 使用说明书

本说明书的内容如有修正，恕不另行通知。未得比亿特电子的允许，不得以任何理由将本说明书的内容以电子或机械的方式，将档案转换成其它模式并予以重制，传输。

深圳市比亿特电子有限公司  
SHEN ZHEN BIYITE TECHNOLOGY CO., LTD

# 目 录

一、功能特点 .....	1
二、读写操作时序 .....	1
三、LCM 内部结构 .....	2
四、指令说明 .....	11
五、应用举例 .....	15
六、注意事项 .....	20
七、质量保证 .....	20

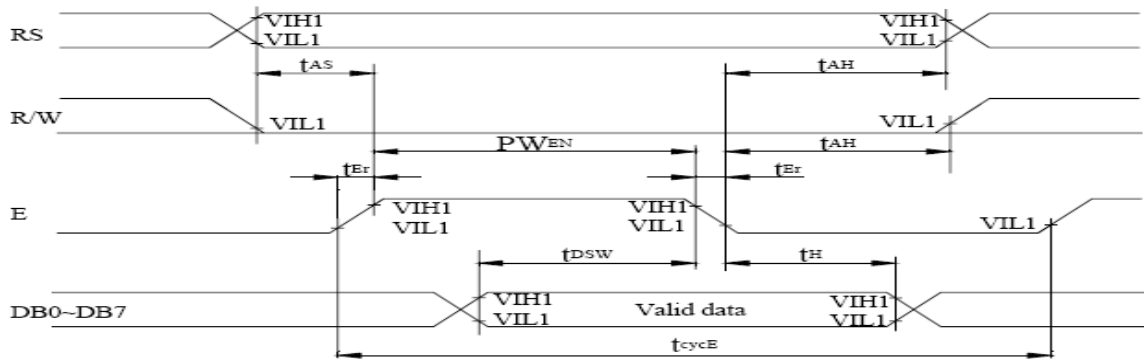
# 一、功能特点

字符型液晶显示模块是专门用于显示字母、数字、符号等的点阵型液晶显示模块。分 4 位和 8 位数据传输方式。提供 5X7 点阵 + 光标和 5X10 点阵 + 光标的显示模式。提供显示数据缓冲区 DDRAM、字符发生器 CGROM 和字符发生器 CGRAM，可以使用 CGRAM 来存储自己定义的最多 8 个 5X8 点阵的图形字符的字模数据。提供了丰富的指令设置：清显示；光标回原点；显示开/关；光标开/关；显示字符闪烁；光标移位；显示移位等。提供内部上电自动复位电路，当外加电源电压超过+4.5V 时，自动对模块进行初始化操作，将模块设置为默认的显示工作状态。

# 二、读写操作时序

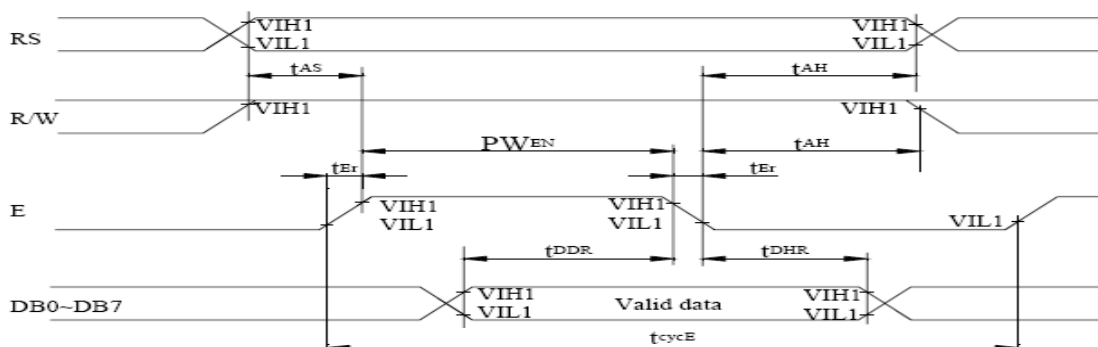
## 1、写操作时序与时序图：

项 目	符 号	条 件	最小值	最大值	单位
E 周期	tcycE	VDD=5V ± 5% VSS=0V Ta=25℃	1000	--	nS
E 脉宽	Pweh		450	--	
E 上升/下降时间	Ter, Tef		--	25	
地址设置时间	Tas		140	--	
地址保持时间	Tah		10	--	
数据设置时间	Tdsw		195	--	
数据保持时间	Th		10	--	



## 2、读操作时序与时序图：

项 目	符 号	条 件	最小值	最大值	单位
E 周期	t <sub>cycE</sub>	VDD=5V ± 5% VSS=0V Ta=25℃	1000	--	nS
E 脉宽	Pweh		450	--	
E 上升/下降时间	T <sub>er</sub> , T <sub>ef</sub>		--	25	
地址设置时间	T <sub>as</sub>		140	--	
地址保持时间	T <sub>ah</sub>		20	--	
数据设置时间	T <sub>dsw</sub>		--	320	
数据保持时间	T <sub>h</sub>		10	--	



## 三、LCM 内部结构

字符型液晶显示模块组件内部主要由 LCD 显示屏(LCD PANEL)、控制器(controller)、驱动器(driver)和偏压产生电路构成。

控制器主要由指令寄存器 IR、数据寄存器 DR、忙标志 BF、地址计数器 AC、DDRAM、CGROM、CGRAM 以及时序发生电路组成：

### 1、指令寄存器（IR）和数据寄存器（DR）

本系列模块内部具有两个 8 位寄存器：指令寄存器（IR）和数据寄存器（DR）。用

户可以通过 RS 和 R/W 输入信号的组合选择指定的寄存器，进行相应的操作。下表中列出了组合选择方式：

E	RS	R/W	说 明
1	0	0	将 DB0 ~ DB7 的指令代码写入指令寄存器中。
1→0		1	分别将状态标志 BF 和地址计数器 (AC) 内容读到 DB7 和 DB6 ~ DB0。
1	0	0	将 DB0 ~ DB7 的数据写入数据寄存器中，模块的内部操作自动将数据写到 DDRAM 或者 CGRAM 中。
1→0		1	将数据寄存器内的数据读到 DB0 ~ DB7，模块的内部操作自动将 DDRAM 或者 CGRAM 中的数据送入数据寄存器中。

## 2、忙标志位 BF

忙标志 BF=1 时，表明模块正在进行内部操作，此时不接受任何外部指令和数据。当 RS=0、R/W=1 以及 E 为高电平时，BF 输出到 DB7。每次操作之前最好先进行状态字检测，只有在确认 BF=0 之后，MPU 才能访问模块；

## 3、地址计数器 (AC)

AC 地址计数器是 DDRAM 或者 CGRAM 的地址指针。随着 IR 中指令码的写入，指令码中携带的地址信息自动送入 AC 中，并做出 AC 作为 DDRAM 的地址指针还是 CGRAM 的地址指针的选择。

AC 具有自动加 1 或者减 1 的功能。当 DR 与 DDRAM 或者 CGRAM 之间完成一次数据传送后，AC 自动会加 1 或减 1。在 RS=0、R/W=1 且 E 为高电平时，AC 的内容送到 DB6 ~ DB0。

Hight order bits			Low order bits			
AC6	AC5	AC4	AC3	AC2	AC1	AC0

## 4、显示数据寄存器 (DDRAM)

DDRAM 存储显示字符的字符码，其容量的大小决定着模块最多可显示的字符数目。DDRAM 地址与 LCD 显示屏上的显示位置的对应关系如下：

(1) 显示一行:

字符列地址	1	2	3	-----	78	79	80
DDRAM 地址	00H	01H	02H	-----	4DH	4EH	4FH

- ① 执行显示移位操作时，对应的 DDRAM 地址也发生移位，以每行 8 个字符的显示为例，(如 LCD0801 系列)，移位前的地址对应关系如下:

字符列地址	1	2	3	4	5	6	7	8
DDRAM 地址	00H	01H	02H	03H	04H	05H	06H	07H

左移一位:

字符列地址	1	2	3	4	5	6	7	8
DDRAM 地址	01H	02H	03H	04H	05H	06H	07H	08H

右移一位:

字符列地址	1	2	3	4	5	6	7	8
DDRAM 地址	4FH	00H	01H	02H	03H	04H	05H	06H

- ② 执行显示移位操作时，对应的 DDRAM 地址也发生移位，以每行 16 个字符的显示为例，(如 LCD1601 系列)，移位前的地址对应关系如下:

字符列地址	1	2	--	7	8	9	10	--	15	16
DDRAM 地址	00H	01H	--	06H	07H	40H	41H	--	46H	47H

左移一位:

字符列地址	1	2	--	7	8	9	10	--	15	16
DDRAM 地址	01H	02H	--	07H	08H	41H	42H	--	47H	48H

右移一位:

字符列地址	1	2	--	7	8	9	10	--	15	16
DDRAM 地址	27H	00H	--	05H	06H	67H	40H	--	45H	46H

(2) 显示二行:

字符列地址		1	2	3	---	38	39	40
DDRAM 地址	第 1 行	00H	01H	02H	---	25H	46H	27H
	第 2 行	40H	41H	42H	---	65H	66H	67H

- ① 执行显示移位操作时，对应的 DDRAM 地址也发生移位，以每行 8 个字符的显示为例，(如 LCD0802 系列)，移位前的地址对应关系如下:

字符列地址		1	2	3	4	5	6	7	8
DDRAM 地址	第 1 行	00H	01H	02H	03H	04H	05H	06H	07H
	第 2 行	40H	41H	42H	43H	44H	45H	46H	47H

左移一位:

字符列地址		1	2	3	4	5	6	7	8
DDRAM 地址	第 1 行	01H	02H	03H	04H	05H	06H	07H	08H
	第 2 行	41H	42H	43H	44H	45H	46H	47H	48H

右移一位:

字符列地址		1	2	3	4	5	6	7	8
DDRAM 地址	第 1 行	27H	00H	01H	02H	03H	04H	05H	06H
	第 2 行	67H	40H	41H	42H	43H	44H	45H	46H

- ② 执行显示移位操作时, 对应的 DDRAM 地址也发生移位, 以每行 16 个字符的显示为例, (如 LCD1602 系列), 移位前的地址对应关系如下:

字符列地址		1	2	3	-----	14	15	16
DDRAM 地址	第 1 行	00H	01H	02H	-----	0DH	0EH	0FH
	第 2 行	40H	41H	42H	-----	4DH	4EH	4FH

左移一位:

字符列地址		1	2	3	-----	14	15	16
DDRAM 地址	第 1 行	01H	02H	03H	-----	0EH	0FH	10H
	第 2 行	41H	42H	43H	-----	4EH	4FH	51H

右移一位:

字符列地址		1	2	3	-----	14	15	16
DDRAM 地址	第 1 行	27H	00H	01H	-----	0CH	0DH	0EH
	第 2 行	67H	40H	41H	-----	4CH	4DH	4EH

- ③ 执行显示移位操作时, 对应的 DDRAM 地址也发生移位, 以每行 20 个字符的显示为例, (如 LCD2002 系列), 移位前的地址对应关系如下:

字符列地址		1	2	3	-----	18	19	20
DDRAM 地址	第 1 行	00H	01H	02H	-----	11H	12H	13H
	第 2 行	40H	41H	42H	-----	51H	52H	53H

左移一位:

字符列地址		1	2	3	-----	18	19	20
DDRAM 地址	第 1 行	01H	02H	03H	-----	12H	13H	14H
	第 2 行	41H	42H	43H	-----	52H	53H	54H

右移一位:

字符列地址		1	2	3	-----	18	19	20
DDRAM 地址	第 1 行	27H	00H	01H	-----	10H	11H	12H
	第 2 行	67H	40H	41H	-----	50H	51H	52H

(3) 显示四行:

字符列地址		1	2	3	---	18	19	20
DDRAM 地址	第 1 行	00H	01H	02H	---	11H	12H	13H
	第 2 行	40H	41H	42H	---	51H	52H	53H
	第 3 行	14H	15H	16H	---	25H	26H	27H
	第 4 行	54H	55H	56H	---	65H	66H	67H

执行显示移位操作时, 对应的 DDRAM 地址也发生移位, 以每行 8 个字符的显示为例, (如 LCD402 系列), 移位前的地址对应关系如下:

字符列地址		1	2	3	---	18	19	20
DDRAM 地址	第 1 行	00H	01H	02H	---	11H	12H	13H
	第 2 行	40H	41H	42H	---	51H	52H	53H
	第 3 行	14H	15H	16H	---	25H	26H	27H
	第 4 行	54H	55H	56H	---	65H	66H	67H

左移一位:

字符列地址		1	2	3	---	18	19	20
DDRAM 地址	第 1 行	01H	02H	03H	---	12H	13H	14H
	第 2 行	41H	42H	43H	---	52H	53H	54H
	第 3 行	15H	16H	17H	---	26H	27H	28H
	第 4 行	55H	56H	57H	---	66H	67H	68H

右移一位:

字符列地址		1	2	3	---	18	19	20
DDRAM 地址	第 1 行	27H	00H	01H	---	10H	11H	12H



	第 2 行	67H	40H	41H	---	50H	51H	52H
	第 3 行	0FH	14H	15H	---	24H	25H	26H
	第 4 行	4FH	54H	55H	---	64H	65H	66H

## 5、字符发生器 ROM

在 CGROM 中，模块已经以 8 位二进制数的形式，生成了 5X8 点阵的字符字模组字符字

模（一个字符对应一组字模）。字符字模是与显示字符点阵相对应的 8X8 矩阵位图数据（与点阵行相对应的矩阵行的高三位为“0”），同时每一组字符字模都有一个由其在 CGROM 中存放地址的高八位数据组成的字符码对应。

字符码地址范围为 00H ~ FFH，其中 00H ~ 07H 字符码与用户在 CGRAM 中生成的自定义图形字符的字模组相对应。

## 6、字符发生器 RAM

在 CGRAM 中，用户可以生成自定义图形字符的字模组。可以生成 5 × 8 点阵的字符字模 8 组，相对应的字符码从 CGROM 的 00H ~ 0FFH 范围内选择。

CGROM 中，字符码与字符字模之间的对应关系表

Upper 4 bit Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL																
LLLH																
LLHL																
LLHH																
LHLL																
LHLH																
LHHL																
LHHH																
HLLL																
HLLH																
HLHL																
HLHH																
HHLH																
HHHL																
HHHH																

5X8 点阵字符的 CGROM 地址、字符字模和字符码三者之间的关系如下图：

CGROM Address												Data				
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D4	D3	D2	D1	D0
0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0
								0	0	0	1	1	0	0	0	0
								0	0	1	0	1	0	1	1	0
								0	0	1	1	1	1	0	0	1
								0	1	0	0	1	0	0	0	1
								0	1	0	1	1	0	0	0	1
								0	1	1	0	1	1	1	1	0
								0	1	1	1	0	0	0	0	0
								1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0
								1	0	1	0	0	0	0	0	0
								1	0	1	1	0	0	0	0	0
								1	1	0	0	0	0	0	0	0
								1	1	0	1	0	0	0	0	0
								1	1	1	0	0	0	0	0	0
								1	1	1	1	0	0	0	0	0
								1	1	1	1	0	0	0	0	0

Cursor position  
光标位置

Character code      Line position  
字符码              行地址

注释：

- 高八位 CGROM 地址 A11 ~ A4 组合形成字符码；
- 低四位 CGROM 地址 A3 ~ A0 定义字模数据存储行地址；
- 数据 D4 ~ D0 为字符字模数据；
- 必须将高三位数据 D5 ~ D7 赋值为 0；
- 对应数据 1 的位置为显示位（黑）；
- 对于 5X8 点阵字体，第九行以下（包括第九行）数据值为 0。

用户自定义 5X8 点阵字符的 CGRAM 地址、字符码和字符字模间关系如下图：

Character Code ( DD RAM Data )								CG RAM Address						Character Patterns ( CG RAM Data )							
b7	b6	b5	b4	b3	b2	b1	b0	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	X	0	0	0	0	0	0	0	0	0	X	X	X	1	1	1	1	1
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
0	0	0	0	X	0	0	1	0	0	1	0	0	1	X	X	X	0	1	1	1	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	0	1	0	0
																	0	1	1	1	0

注释： 字符码 0 ~ 2 位与 CGRAM 地址 3 ~ 5 位对应；  
CGRAM 地址 0 ~ 2 位生成字模数据行位置。第八行是光标位置， 因此构成字符字模数据时，在设置光标显示的情况下，应赋值为 0；如果赋值为 1，不论光标显示与否，第八行均处于显示状态；

字符字模数据 0 ~ 4 位的赋值状态构成了自定义字符的位图数据；  
从图中可以看出，字符码 3 位的赋值状态并不影响用户自定义 字符在 CGROM 中的字符码，用户自定义字符码的范围为 00H ~ 07H 或者 08H ~ 0FH，也就是说字符码 00H 与 08H 对应同一组用户自定义字符字模；  
CGRAM 数据为 1 时，处于显示状态。

## 四、指令说明

由于 MPU 可以直接访问模块内部的 IR 和 DR，作为缓冲区域，IR 和 DR 在模块进行内部操作之前，可以暂存来自 MPU 的控制信息。这样就给用户在 MPU 和外围控制设备的选择上，增加了余地。模块的内部操作由来自 MPU 的 RS、R/W、E 以及数据信号 DB0-DB7 决定，这些信号的组合形成了模块的指令。

本系列模块向用户提供了 11 条指令，大致可以分为四大类：

模块功能设置，诸如：显示格式、数据长度等；

设置内部 RAM 地址；

完成内部 RAM 数据传送；

完成其他功能。

一般情况下，内部 RAM 的数据传送的功能使用最为频繁，因此，RAM 中的地址指针所具备的自动加一或减一功能，在一定程度上减轻了 MPU 编程负担。此外，由于数据移位指令与写显示数据可同时进行，这样用户就能以最少系统开发时间，达到最高的编程效率。

有一点需特别注意：在每次访问模块之前，MPU 应首先检测忙标志 BF，确认 BF=0 后，访问过程才能进行。

### 1、Clear display (清显示)

指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

清显示指令将空位字符码 20H 送入全部 DDRAM 地址中，使 DDRAM 中的内容全部清除，显示消失；地址计数器 AC=0，自动增 1 模式；显示归位，光标或者闪烁回到原点（显示屏左上角）；但并不改变移位设置模式。

### 2、Return home (归位)

指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

归位指令置地址计数器 AC=0；将光标及光标所在位的字符回原点；但 DDRAM 中的内容并不改变。

### 3、Entry mode set (设置输入模式)

指令码：

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

I/D: 字符码写入或者读出 DDRAM 后 DDRAM 地址指针 AC 变化方向标志：

I/D=1, 完成一个字符码传送后, 光标右移, AC 自动加 1;

I/D=0, 完成一个字符码传送后, 光标左移, AC 自动减 1;

S: 显示移位标志:

S=1, 将全部显示向右 (I/D=0) 或者向左 (I/D=1) 移位;

S=0, 显示不发生移位;

S=1 时, 显示移位时, 光标似乎并不移位; 此外, 读 DDRAM 操作以及对 CGRAM 的访问, 不发生显示移位。

#### 4、Display on/off control (显示开/关控制)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D: 显示开/关控制标志: D=1, 开显示; D=0, 关显示;

关显示后, 显示数据仍保持在 DDRAM 中, 立即开显示可以再现;

C: 光标显示控制标志: C=1, 光标显示; C=0, 光标不显示;

不显示光标并不影响模块其它显示功能; 显示 5X8 点阵字符时, 光标在第八行显示, 显示 5X10 点阵字符时, 光标在第十一行显示;

B: 闪烁显示控制标志: B=1, 光标所指位置上, 交替显示全黑点阵和显示字符, 产生闪烁效果, Fosc=250kHz 时, 闪烁频率为 0.4ms 左右; 通过设置, 光标可以与其所指位置的字符一起闪烁。

#### 5、Cursor or display shift (光标或显示移位)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

光标或显示移位指令可使光标或显示在没有读写显示数据的情况下, 向左或向右移动; 运用此指令可以实现显示的查找或替换; 在双行显示方式下, 第一行和第二行会同时移位; 当移位越过第一行第四十位时, 光标会从第一行跳到第二行, 但显示数据只在本行内水平移位, 第二行的显示决不会移进第一行; 倘若仅执行移位操作, 地址计数器 AC 的内容不会发生改变。

S/C	R/L	说 明
0	0	光标向左移动, AC 自动减 1
0	1	光标向右移动, AC 自动加 1
1	0	光标与显示一起向左移动, AC 值不变
1	1	光标与显示一起向右移动, AC 值不变



## 6、Function set (功能设置)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

功能设置指令设置模块数据接口宽度和 LCD 显示屏显示方式，即 MPU 与模块接口数据总线为 4 位或者是 8 位、LCD 显示行数和显示字符点阵规格；所以建议用户最好在执行其它指令设置（读忙标志指令除外）之前，在程序的开始，进行功能设置指令的执行。

DL: 数据接口宽度标志:

DL=1, 8 位数据总线 DB7 ~ DB0;

DL=0, 4 位数据总线 DB7 ~ DB4, DB3 ~ DB0 不用, 使用此方式传送数据, 需分两次进行;

N: 显示行数标志:

N=1, 两行显示模式; N=0, 单行显示模式;

F: 显示字符点阵字体标志:

F=1: 5X10 点阵 + 光标显示模式; F=0: 5X7 点阵 + 光标显示模式。

## 7、Set CGRAM address (CGRAM 地址设置)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	ACG5	ACG4	ACG3	ACG2	ACG1	ACG0

CGRAM 地址设置指令设置 CGRAM 地址指针，它将 CGRAM 存储用户自定义显示字符的字模数据的首地址 ACG5 ~ ACG0 送入 AC 中，于是用户自定义字符字模就可以写入 CGRAM 中或者从 CGRAM 中读出。

## 8、Set DDRAM address (DDRAM 地址设置)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

DDRAM 地址设置指令设置 DDRAM 地址指针，它将 DDRAM 存储显示字符的字符码的首地址 ADD6 ~ ADD0 送入 AC 中，于是显示字符的字符码就可以写入 DDRAM 中或者从 DDRAM 中读出;

值得注意的是: 在 LCD 显示屏一行显示方式下, DDRAM 的地址范围为: 00H ~ 4FH; 两行显示方式下, DDRAM 的地址范围为: 第一行 00H ~ 27H, 第二行 40H ~ 67H。

## 9、Read busy flag and address (读忙标志 BF 和 AC)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

当 RS=0 和 R/W=1 时,在 E 信号高电平的作用下,BF 和 AC6 ~ AC0 被读到数据总线 DB7 ~ DB0 的相应位;

BF: 内部操作忙标志, BF=1, 表示模块正在进行内部操作, 此时模块不接收任何外部指令和数据, 直到 BF=0 为止;

AC6 ~ AC0: 地址计数器 AC 内的当前内容, 由于地址计数器 AC 为 CGROM、CGRAM 和 DDRAM 的公用指针, 因此当前 AC 内容所指区域由前一条指令操作区域决定; 故只有 BF=0 时, 送到 DB7 ~ DB0 的数据 AC6 ~ AC0 才有效。

## 10、Write data to CGRAM or DDRAM (写数据到 CGRAM 或 DDRAM)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

写数据到 CGRAM 或 DDRAM 指令, 是将用户自定义字符的字模数据写到已经设置好的 CGRAM 的地址中, 或者将欲显示字符的字符码写到 DDRAM 中; 欲写入的数据 D7 ~ D0 首先暂存在 DR 中, 再由模块的内部操作自动写入地址指针所指定的 CGRAM 单元或者 DDRAM 单元中。

## 11、Read data from CGRAM or DDRAM (从 CGRAM 或 DDRAM 中读数据)

指令码:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

从 CGRAM 或 DDRAM 中读数据指令, 是从地址计数器 AC 指定的 CGRAM 或者 DDRAM 单元中, 读出数据 D7 ~ D0; 读出的数据 D7 ~ D0 暂存在 DR 中, 再由模块的内部操作送到数据总线 DB7 ~ DB0 上; 需要注意的是, 在读数据之前, 应先通过地址计数器 AC 正确指定读取单元的地址。



## 五、应用举例

接口接线方式

连接线图:

```
=====
| LCM-----51    | LCM-----51    | LCM-----51    |
|-----|
| DB0-----P1.0  | DB4-----P1.4  | RS-----P2.0    |
| DB1-----P1.1  | DB5-----P1.5  | RW-----P2.1    |
| DB2-----P1.2  | DB6-----P1.6  | E-----P2.2     |
| DB3-----P1.3  | DB7-----P1.7  | V0 接 10K 可调电阻到 GND 和 VDD |
|-----|
```

[注: AT89S52 使用 12M 晶振]

=====\*/

```
#include <reg51.h>
```

```
//#define eight_bit//定义 8 位并口
```

```
//定义引脚
```

```
sbit LCM_RS = P2^0 ;
```

```
sbit LCM_RW = P2^1 ;
```

```
sbit LCM_E  = P2^2;
```

```
#define LCM_Data P1
```

```
#define BUSY      0x80 //用于检测 LCM 状态字中的 Busy 标识
```

```
#define uchar unsigned char
```

```
void Wr_Data(uchar wr_data);    //写数据
```

```
void Wr_Command(uchar wr_command); //写指令
```

```
void Rd_Status(void);    //读忙
```

```
void Init(void);        // 初始化
```

```
void Delay5Ms(void);
```

```
void Delay400Ms(void);
```

```
void Display_String(uchar line, uchar *string);
```

---

```
uchar code *string-1[]={ "    LCD-LCD    "};
uchar code *string-2[]={ "  TEL: 2888888  "};
uchar code *string-3[]={ "WELCOME TO YOU! "};

void main(void)
{
    Delay400Ms (); //启动等待, 等 LCM 讲入工作状态
    Init (); //LCM 初始化
    while (1)
    {
        Display-String (1, *string-1);
        Display-String (2, *string-3);
        Delay400Ms ();
        Delay400Ms ();
        Delay400Ms ();
        Display-String (2, *string-2);
        Delay400Ms ();
        Delay400Ms ();
        Delay400Ms ();
    }
}

//显示字符串
void Display-String (uchar line, uchar *string)
{
    uchar addr, i;
    if (line==1) addr=0x80;
    else if (line==2) addr=0xc0;
    for (i=0; i<16; i++)
    {
        Wr-Command (addr);
        Wr-Data (string[i]);
        addr++;
    }
}
```

---

```
#ifdef eight_bit //8 位并口
//写数据
void Wr_Data (uchar wr_data)
{
    Rd_Status ();
    LCM_RS = 1;
    LCM_RW = 0;
    LCM_E = 1;
    LCM_Data = wr_data;
    LCM_E = 0; //若晶振速度太高可以在这后加小的延时
}

//写指令
void Wr_Command (uchar wr_command)
{
    Rd_Status ();
    LCM_RS = 0;
    LCM_RW = 0;
    LCM_E = 1;
    LCM_Data = wr_command;
    LCM_E = 0; //若晶振速度太高可以在这后加小的延时
}

#else //4 位并口

//写数据
void Wr_Data (uchar wr_data)
{
    Rd_Status ();
    LCM_RS = 1;
    LCM_RW = 0;

    LCM_E = 1;
    LCM_Data = wr_data;
    LCM_E = 0;
```

```
LCM_E = 1;
LCM_Data = (wr_data<<4);
LCM_E = 0;
}

//写指令
void Wr_Command(uchar wr_command)
{
    Rd_Status();
    LCM_RS = 0;
    LCM_RW = 0;

    LCM_E = 1;
    LCM_Data = wr_command;
    LCM_E = 0;

    LCM_E = 1;
    LCM_Data = (wr_command<<4);
    LCM_E = 0;
}

#endif
```

```
//检测忙信号
void Rd_Status(void)
{while(1)
{
    LCM_RS = 0;
    LCM_RW = 1;
    LCM_E = 0;
    LCM_Data = 0xFF;
    LCM_E = 1;
    if((LCM_Data & BUSY)==0)
```

```
break;
}
}
//LCM 初始化
void Init(void)
{
    Wr_Command(0x38); //三次显示模式设置
    Delay5Ms();
    Wr_Command(0x38);
    Delay5Ms();
    Wr_Command(0x38);
    Delay5Ms();
    Wr_Command(0x38); //显示模式设置, 开始要求每次检测忙信号
    Wr_Command(0x08); //关闭显示
    Wr_Command(0x01); //显示清屏
    Wr_Command(0x06); // 显示光标移动设置
    Wr_Command(0x0C); // 显示开及光标设置
}

//5ms 延时
void Delay5Ms(void)
{
    unsigned int TempCyc = 5552;
    while(TempCyc--);
}

//400ms 延时
void Delay400Ms(void)
{
    unsigned char TempCycA = 5;
    unsigned int TempCycB;
    while(TempCycA--)
    {
        TempCycB=7269;
        while(TempCycB--);
    }
}
```

## 六、注意事项

### 1. 处理

- (1) 要避免在处理机械振动和对模块施加外力，都可能使屏不显示或损坏。
- (2) 不能用手或坚硬工具或物体接触、按压、磨擦显示屏，否则屏上的偏光片被物体划坏。
- (3) 如果屏破裂液晶材料外漏，液晶可以通过空气被吸入，而且要避免液晶与皮肤接触，如果接触应立即用酒精冲洗，然后再用水彻底冲洗。
- (4) 不能使用可溶有机体来清洗显示屏。因为这些可溶的溶剂对偏光片不利，清洗显示屏时，可用棉花蘸少量石油苯轻轻擦拭或用透明胶带粘起脏物。
- (5) 要防止高压静电产生的放电，将损坏模块中的 CMOS 电路。
- (6) 不能把模块放在温度高的地方，尤其不能长时间放在湿度大的地方，最好把模块放在温度为  $0^{\circ}\text{C}-35^{\circ}\text{C}$ ，湿度低于 70% 的环境中。
- (7) 模块不能贮存在太阳直射的地方。

### 2. 操作

- (1) 当电源接通时，不能组装或拆卸模块。
- (2) 在电源电压的偏差、输入电压的偏差及环境温度等最坏条件下，也不能超过最大的额定值，否则将损坏 LCD 模块。

## 七、质量保证

如在此手册列明的正常条件下使用、储存该产品，公司将提供 7 天包换、12 个月保修的质量保证。

### 深圳市比亿特电子有限公司

公司地址：深圳市宝安区沙井万丰一号路源兴大厦 13 楼

工厂地址：深圳市宝安区福永塘尾富华工业区 11 栋 2 楼

电 话：0755-36520656 23054910

传 真：0755-23054910

电子信箱：bytlcd@yeah.net

公司网址：<http://www.bytlcd.com>