



佳显电子技术

应用手册

字符型模组

SJXD1601ADBSWWN

编写	审核	批准

目 录

■	液晶显示模块整体描述-----	2
■	原理框图-----	2
■	外形尺寸图纸-----	3
■	接口引脚定义-----	4
■	液晶屏显示控制部分-----	5
1	原理图-----	5
2	极限参数-----	5
3	电气特性-----	5
1)	直流特性-----	5
2)	交流特性-----	6
4	液晶屏使用说明-----	8
1)	主要部分说明-----	8
2)	指令集-----	10
3)	指令详细解释-----	10
4)	初始化指令-----	16
■	背光使用说明-----	18
1	背光特性-----	18
2	原理框图-----	18
3	极限特性-----	18
4	光电特性-----	18
■	DDRAM 地址映射关系-----	19
■	电源的连接方式-----	19
■	示例-----	19
1	硬件部分-----	19
2	软件部分-----	20
1)	间接控制方式-----	20
2)	直接控制方式-----	24
■	液晶显示模块在使用过程中的注意事项-----	28
1	液晶显示模块在装配过程中的注意事项-----	28
2	液晶显示模块在使用过程中的注意事项-----	29
■	附录 A-----	30

■ 液晶显示模块整体描述

项 目	说 明	单位
主要组成部分	液晶显示屏，LED 背光灯箱，线路板，铁框，导电橡胶	
外形尺寸	80.00(长)×36.00(宽)×12.00(最大厚度)	mm
视域	64.5(长)×14.00(宽)	mm
显示内容类型	字符型	
连接方式	COB	
显示模式	STN：蓝膜（负性）	
视角	6	点钟
显示字符数	16 个字符/行×1 行	
显示字符尺寸	6.35(长)×3.2(宽)	mm
显示字符间距	3.75(宽)	mm
显示屏占空比	1/16	Duty
显示屏偏置电压	1/5	Bias
液晶显示模块使用温度范围	-20~+70	℃
液晶显示模块存储温度范围	-30~+80	℃
背光	LED：白色	
数据输入格式	4 位/8 位并行输入格式，6800 时序方式	
电源输入电压	5.0V 输入供电(注：可使用 3.0V，但需另行开发)	
特点	1. 内置字符发生器 ROM(CGROM)：10080 位(204 个 5×8 点阵格式英日字符和 32 个 5×11 点阵格式英日字符)； 英欧/英俄字符字体库需专门指定 2. 字符发生器 RAM(CGRAM)：5×8 位(8 个 5×8 点阵格式字符)，特殊字符图案可以对 CGRAM 直接编程得到	
液晶显示模块理论寿命	50000	小时

注：当液晶显示屏显示模式为负性时，在显示时必须点亮背光

■ 原理框图

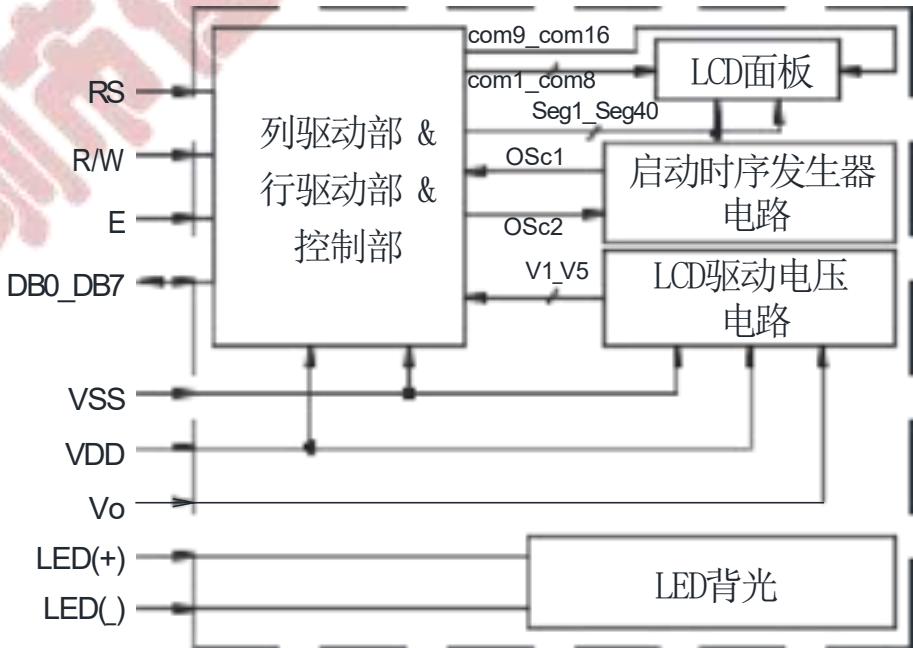
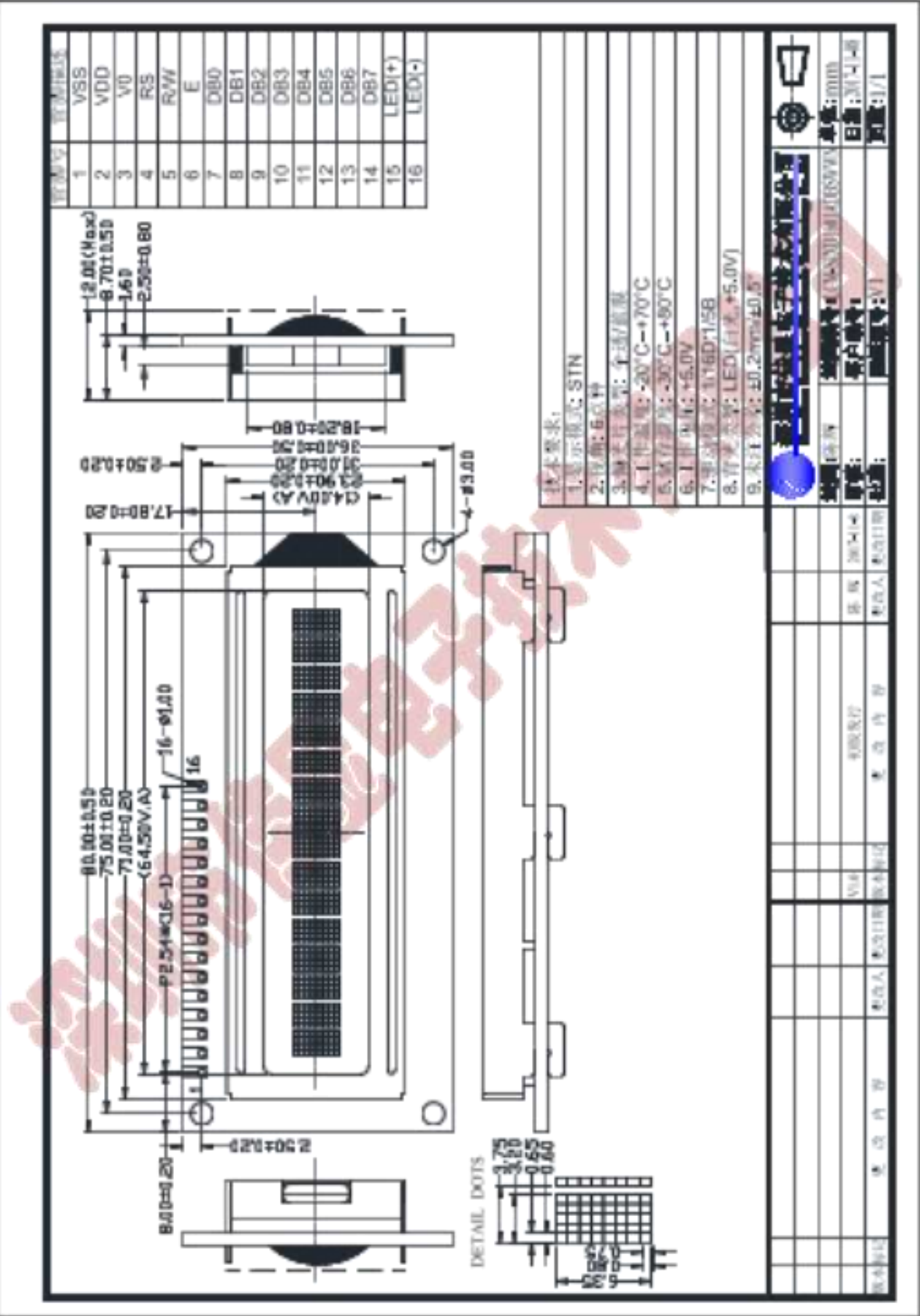


图 1. 原理框图

■ 外形尺寸图纸



■ 接口引脚定义

引脚号	符号	状态	功 能
1	VSS	0V	液晶显示供电电源地
2	VDD	+5.0V	液晶显示供电电源正
3	VO	-	液晶显示驱动电源（调节显示对比度） VDD-VO=4.7V, 显示为最佳值，客户可在此基础上微调 如 VDD-VO 电压过高，显示字符位全部点都显示 如 VDD-VO 电压过低，显示字符位不显示
4	RS	H/L	数据/指令寄存器选择信号 L：选择指令寄存器 H：选择数据寄存器
5	R/W	H/L	读/写选择信号 L：写指令或数据 H：读指令或数据
6	E	H/L	使能信号 H：允许读指令或数据 H → L：允许写指令或数据
7	DB0	H/L	数据总线 当设置数据总线为 8 位长度时，DB7~DB0 有效 当设置数据总线为 4 位长度时，DB7~DB4 有效
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	LED(+)	+5.0V	背光供电电源正
16	LED(-)	0V	背光供电电源地

■ 液晶屏显示控制部分

1 原理图

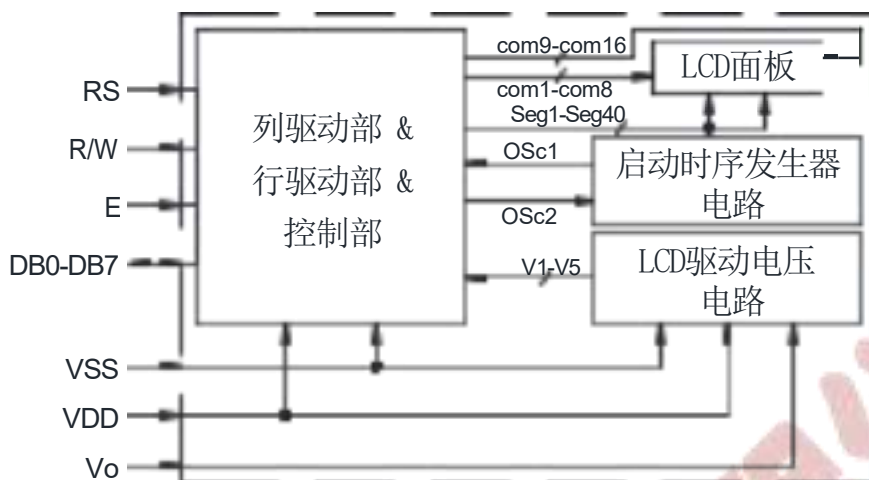


图 2. 液晶屏显示控制部分原理框图

2 极限参数

条件: $T_a=25^{\circ}\text{C}$

参 数	符 号	值	单位
电源电压	VDD	-0.3~+7.0	V
驱动电压	VO	$VDD-15\sim VDD+0.3$	V
输入电压	VIN	-0.3~ $VDD+0.3$	V
工作温度	Topr_LCD	-20~+70	$^{\circ}\text{C}$
存储温度	Tstg_LCD	-30~+80	$^{\circ}\text{C}$

3 电气特性

1) 直流特性

条件: $VDD=4.5\text{Vto}5.5\text{V}$, $T_a=-20\text{to}+70^{\circ}\text{C}$

参数	符号	测试条件	最小	典型	最大	单位
工作电压	VDD	—	4.5	—	5.5	V
电源电流 (不含背光)	IDD	内部振荡或者外部时钟 ($VDD=5.0\text{V}$, $f_{osc}=270\text{KHz}$)	—	0.6	—	mA
输入电压	V_{IH1}	—	2.2	—	VDD	V
	V_{IL1}	—	-0.3	—	0.6	V
输出电压	V_{OH1}	$I_{OH1}=-0.205\text{mA}$	2.4	—	—	V
	V_{OL1}	$I_{OL1}=1.2\text{mA}$	—	—	0.4	V

条件: $VDD=2.7\text{Vto}4.5\text{V}$, $T_a=-20\text{to}+70^{\circ}\text{C}$

参数	符号	测试条件	最小	典型	最大	单位
工作电压	VDD	—	2.7	—	4.5	V
电源电流 (不含背光)	IDD	内部振荡或者外部时钟 ($VDD=3.0\text{V}$, $f_{osc}=270\text{KHz}$)	—	0.5	—	mA
输入电压	V_{IH1}	—	0.7VDD	—	VDD	V
	V_{IL1}	—	-0.3	—	0.55	V
输出电压	V_{OH1}	$I_{OH1}=-0.1\text{mA}$	0.75VDD	—	—	V
	V_{OL1}	$I_{OL1}=0.1\text{mA}$	—	—	0.2VDD	V

2) 交流特性

条件: VDD=4.5Vto5.5V, Ta=-30to+85°C

模式	参数	符号	最小	典型	最大	单位
写模式 (参照图 3)	E 周期	t_C	500	—	—	ns
	E 上升/下降时间	$t_{R,tF}$	—	—	20	
	E 脉冲宽度	t_W	230	—	—	
	R/W 和 RS 建立时间	t_{SU1}	40	—	—	
	R/W 和 RS 保持时间	t_{H1}	10	—	—	
	数据建立时间	t_{SU2}	80	—	—	
	数据保持时间	t_{H2}	10	—	—	
读模式 (参照图 4)	E 周期	t_C	500	—	—	ns
	E 上升/下降时间	$t_{R,tF}$	—	—	20	
	E 脉冲宽度	t_W	230	—	—	
	R/W 和 RS 建立时间	t_{SU}	40	—	—	
	R/W 和 RS 保持时间	t_H	10	—	—	
	数据输出延迟时间	t_D	—	—	120	
	数据保持时间	t_{DH}	5	—	—	

条件: VDD=2.7Vto4.5V, Ta=-30to+85°C

模式	参数	符号	最小	典型	最大	单位
写模式 (参照图 3)	E 周期	t_C	1000	—	—	ns
	E 上升/下降时间	$t_{R,tF}$	—	—	25	
	E 脉冲宽度	t_W	450	—	—	
	R/W 和 RS 建立时间	t_{SU1}	60	—	—	
	R/W 和 RS 保持时间	t_{H1}	20	—	—	
	数据建立时间	t_{SU2}	195	—	—	
	数据保持时间	t_{H2}	10	—	—	
读模式 (参照图 4)	E 周期	t_C	1000	—	—	ns
	E 上升/下降时间	$t_{R,tF}$	—	—	25	
	E 脉冲宽度	t_W	450	—	—	
	R/W 和 RS 建立时间	t_{SU}	60	—	—	
	R/W 和 RS 保持时间	t_H	20	—	—	
	数据输出延迟时间	t_D	—	—	360	
	数据保持时间	t_{DH}	5	—	—	

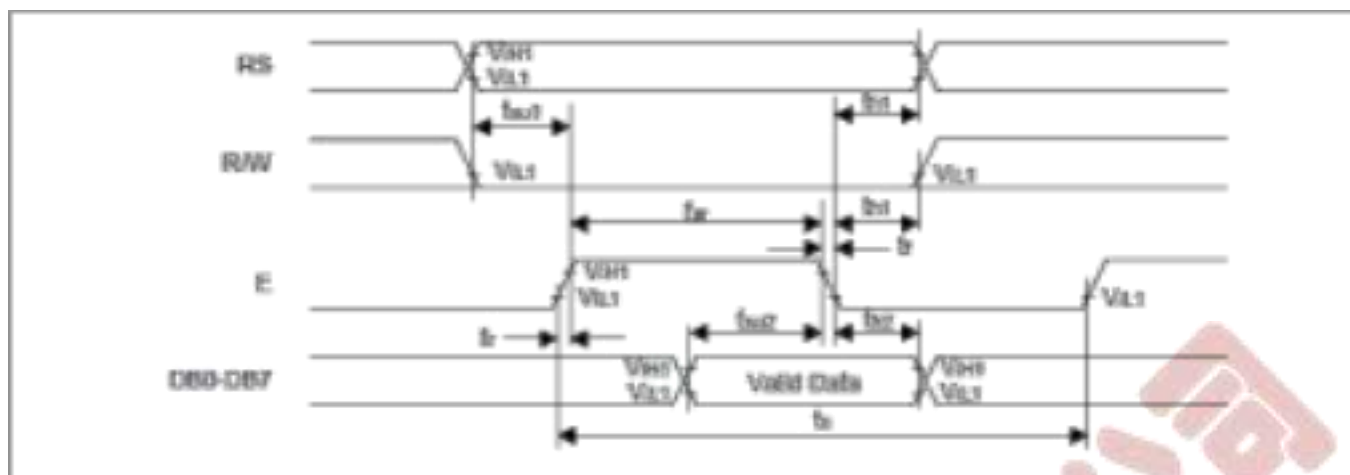


图 3. 写模式时序图

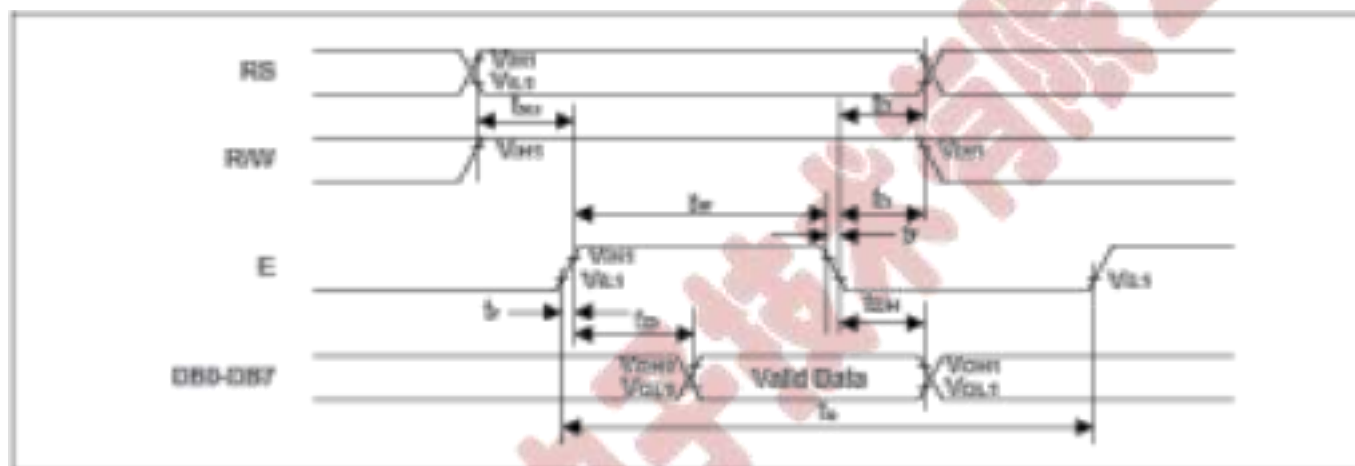


图 4. 读模式时序图

4 液晶屏使用说明

1) 主要部分说明

1.1 地址指针计数器 AC

地址指针计数器 AC 是可读可写计数器。它是 DDRAM 和 CGRAM 共同的地址指针计数器，指示当前 DDRAM 或 CGRAM 的地址。指示哪种存储器的地址是由 MPU 最近写入的地址设置指令的标识码来确定。地址指针计数器 AC 可以由 MPU 设置成加一计数器或减一计数器。当 MPU 进行读/写数据操作后地址指针计数器 AC 自动进行修正。地址指针计数器 AC 有两个作用，其一是指示当前的 DDRAM 或 CGRAM 地址；其二为光标和闪烁的位置地址指针，指示当前光标和闪烁的位置地址。

1.2 光标闪烁控制电路

系统具有光标和闪烁功能。光标是以底线形式呈现在字符位的第 8 行上。闪烁是在一定的频率下当前显示和全显示交替产生的一种显示效果。闪烁有字符的闪烁和光标的闪烁两种。字符的闪烁效果可以造成阴影块光标的效果。光标和闪烁的控制电路可以受 MPU 的控制，由 MPU 写入控制指令设置光标和闪烁的状态。

1.3 字符发生器

系统内置两种字符发生器。一种为 CGROM 即已固化好的字模库，见附录 A。它含有 208 种 5×8（含光标位）点阵字体的字符字模数据和 32 种 5×11（含光标位）点阵字体的字符字模数据。MPU 只要写入某个字符的字符代码，系统将以其作为字模库的地址将该字符输出显示。另一种为 CGRAM 即可随时定义的字符字模库。系统提供 64 个字节的 CGRAM，地址为 00H~3FH。它可以生成 8 个 5×8 点阵（作为光标行也可以占用）的自定义字符（字符代码 00H~07H）或 4 个 5×11 点阵（作为光标行也可以占用）的自定义字符（字符代码 00H~04H）。由于系统仅使用一行 5 位数据为字符点阵，所以作为 CGRAM 字模库仅使用存储单位字节的低 5 位，而高 3 位虽然存在但不作为字模数据使用。系统提供给 CGRAM 的字符字模代码为 00H~07H 或 08H~0FH。作为 5×8 点阵字符的字模库，CGRAM 每 8 个字符为一个字符的字模数据，字模数据存储顺序是从上到下排列。每个字符代码都对应着 CGRAM 的 8 个单元；作为为 5×11 点阵字符的字模库，CGRAM 每 16 个字节为一个字符的字模数据，其中前 11 个字节为字模数据存储单元，后 5 个字节与字模无关。字符代码与 CGRAM 地址的对应关系为：

5×8 点阵字符 (CGRAM)		5×11 点阵字符 (CGRAM)	
字符代码	CGRAM 地址	字符代码	CGRAM 地址
00H (08H)	00H~07H	00H (08H)	00H~0FH
01H (09H)	08H~0FH	01H (09H)	10H~1FH
02H (0AH)	10H~17H	02H (0AH)	20H~2FH
03H (0BH)	18H~1FH	03H (0BH)	30H~3FH
04H (0CH)	20H~27H		
05H (0DH)	28H~2FH		
06H (0EH)	30H~37H		
07H (0FH)	38H~3FH		

如示例图 5，图 6 给出了字符代码为 00H 的 CGRAM 地址及字模数据与显示效果的对应关系

1.4 显示存储器 (DDRAM)

系统拥有 80 个字节的显示存储器 DDRAM。DDRAM 用于存储当前所要显示的字符的字符代码。DDRAM 的地址由地址指针计数器 AC 提供，MPU 可以对 DDRAM 进行读/写操作。DDRAM 各单元对应着显示屏上的各字符位。DDRAM 地址定义分两种：一种为一行显示的地址定义，一种为两行显示的地址定义，如图 7，图 8 所示。在两行显示时，第二行总是从 DDRAM 的后半部分开始，即第 41 单元起单元定义为第二行 DDRAM 单元。

CGRAM 地址	CGRAM 单元数据	显示屏的显示效果
00H	08H	
01H	0FH	
02H	10H	
03H	0FH	
04H	0AH	
05H	1FH	
06H	02H	
07H	02H	
08H	02H	
09H	02H	
0AH	00H	

图 5. ”年”字的 5×11 点阵字模数据

CGRAM 地址	CGRAM 单元数据	显示屏的显示效果
00H	0FH	
01H	09H	
02H	0FH	
03H	09H	
04H	0FH	
05H	09H	
06H	13H	
07H	00H	

图 6. ”月”字的 5×8 点阵字模数据

显示位置	1	2	3	4	79	80
DDRAM 地址	00H	01H	02H	03H	4EH	4FH

图 7. 一行显示 DDRAM 地址定义

显示位置	1	2	3	4	39	40
第一行	00H	01H	02H	03H	26H	27H
第二行	40H	41H	41H	43H	66H	67H

图 8. 两行显示 DDRAM 地址定义

1.5 复位电路

系统在电源上电时复位电路被首先启动实现硬件初始化:

- a) 清屏，即清 DDRAM
- b) 工作方式设置为 8 位总线接口，一行 5×8 字体字符显示
- c) 关显示
- d) 输入方式为地址指针计数器 AC 加一形式

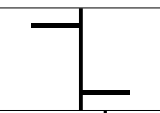
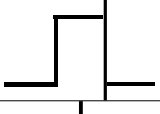
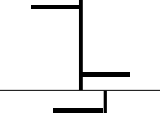
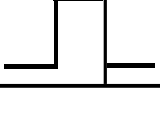
初始化过程中，接口部对 MPU 呈“忙”状态

复位电路的启动是有条件的，当电源从 0.2V 上升到 4.5V 时所需时间在 (0.1~10ms) 之外，或电源瞬间抖动低电平 (<0.2V) 持续时间小于 1ms (即使此时的电源上电时间满足要求) 时，复位电路都不会启动

1.6 接口时序

系统的接口时序为 M6800 时序，其特点是读/写操作时序是由使能信号 E 完成，E 信号是正脉冲信号，不操作时为低电平状态，操作时产生一个正脉冲。在读操作时，E 信号在高电平时，控制部分将所需数据送入数据总线上，供 MPU 读取；在写操作时，E 信号的下降沿处将数据总线上的数据写入控制部分接口部的寄存器内。系统对读/写操作的识别是判断 R/W 信号端上的电平状态，R/W=1 为读操作选择，R/W=0 为写操作选择。R/W 信号的宽度要大于 E 信号的宽度才能保证 MPU 的操作正确。RS 信号是系统识别数据总线上的数据是属于指令代码还是属于显示数据。RS=0 选通指令寄存器通道，数据总线传输的是指令代码或标示位；RS=1 选通数据寄存器通道，数据总线传输的是显示数据或自定义字符的字模数据。接口部信号端的逻辑功能组合表如表 9 所示。

表 9

RS	R/W	E	DB0-DB7	功能
0	0		输入态	写指令代码
0	1		输出态	读 BF 及 AC 值
1	0		输入态	写数据
1	1		输出态	读数据

2) 指令集

指令名称	控制信号		控制代码								运行时间 fosc=270kHz
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
清屏	0	0	0	0	0	0	0	0	0	1	1.53ms
归 home 位	0	0	0	0	0	0	0	0	1	-	1.53ms
输入方式设置	0	0	0	0	0	0	0	1	I/D	S	39us
显示状态设置	0	0	0	0	0	0	1	D	C	B	39us
光标或画面滚动	0	0	0	0	0	1	S/C	R/L	-	-	39us
工作方式设置	0	0	0	0	1	DL	N	F	-	-	39us
CGRAM 地址设置	0	0	0	1	A5	A4	A3	A2	A1	A0	39us
DDRAM 地址设置	0	0	1	A6	A5	A4	A3	A2	A1	A0	39us
读 BF 和 AC 值	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	0us
写数据	1	0	数据								43us
读数据	1	1	数据								43us

注: “-”表示任意值, 在实际应用时一般认为是“0”

3) 指令详细解释

3.1 清屏

	D7	D6	D5	D4	D3	D2	D1	D0	
格式	0	0	0	0	0	0	0	1	代码: 01H

该指令完成下列功能:

将空码 (20H) 写入 DDRAM 的全部的单元内。

将地址计数器 AC 清零, 光标或闪烁归 home 位。

设置输入方式参数 I/D=1, 即地址指针 AC 为自动加一输入模式。

该指令多用于上电时或更新全屏显示内容时。在使用该指令之前要确认 DDRAM 的当前内容是否是有用。

3.2 归 home 位

	D7	D6	D5	D4	D3	D2	D1	D0	
格式	0	0	0	0	0	0	1	0	代码: 02H

该指令将地址指针计数器 AC 清零。执行该指令的效果有：将光标或闪烁位返回到显示屏的左上第一字符位上，即 DDRAM 地址 00H 单元位置；这是因为光标和闪烁位都是以地址指针计数器 AC 当前值定位的。如果画面已滚动，则撤消滚动效果，将画面拉回到 home 位。

3.3 输入方式设置

	D7	D6	D5	D4	D3	D2	D1	D0	
格式	0	0	0	0	0	1	I/D	S	代码:04H~07H

该指令的功能在于设置了显示字符的输入方式，即在 MPU 读写 DDRAM 或 CGRAM 后，地址指针计数器 AC 的修改方式；反映在显示效果上，当写入一个字符后画面或光标的移动。该指针的两个参数位 I/D 和 S 确定了字符的输入方式。

I/D 表示当 MPU 读/写 DDRAM 或 CGRAM 的数据后，地址指针计数器 AC 的修改方式，由于光标位置也是由 AC 值确定，所以也是光标移动的方式。

I/D=0 AC 为减 1 计数器，光标左移一个字符位

I/D=1 AC 为加 1 计数器，光标右移一个字符位

S 表示在写入字符时，是否允许显示画面的滚动。

S=0 禁止滚动

S=1 允许滚动

S=1 且 I/D=0 显示画面向右滚动一个字符位

S=1 且 I/D=1 显示画面向左滚动一个字符位

综合而论，该指令可以实现四种字符的输入方式，见下表所示：

输入方式	指令代码	参数状态	示例
画面不动光标左移	04H	S=0, I/D=0	
画面右滚动	05H	S=1, I/D=0	
画面不动光标右移	06H	S=0, I/D=1	
画面左滚动	07H	S=1, I/D=1	

注：“-”为光标符号

注：画面滚动方式在 MPU 读 DDRAM 数据，或在读/写 CGRAM 时无效，也就是说该指令主要应用在 MPU 写入 DDRAM 数据的操作时，所以称该指令为输入方式设置指令。需要说明的是在 MPU 读 DDRAM 数据或在读/写 CGRAM 数据时，建议将 S 置 0。

3.4 显示状态设置

	D7	D6	D5	D4	D3	D2	D1	D0	
格式	0	0	0	0	1	D	C	B	代码:08H~0FH

该指令控制着画面, 光标及闪烁的开与关。该指令有三个状态为 D, C, B, 这三个状态位分别控制着画面, 光标和闪烁的显示状态。

D 画面显示状态位

D=1 开显示

D=0 关显示

注: 关显示仅是画面不显示, 而 DDRAM 内容不变。着与清屏指令截然不同。

C 光标显示状态位。

C=1 光标显示

C=0 光标消失

注: 光标为底线形式 (5×1 点阵), 出现在第八行 (5×8 字符字体) 或第十一行 (5×11 字符字体) 上。光标的位置有地址指针计数器 AC 确定, 并随其变动而移动。当 AC 值超出了画面的显示范围, 光标将随之消失。

B 闪烁显示状态位。

B=1 闪烁启用

B=0 闪烁禁止

注: 闪烁是指一个字符位交替进行正常显示态和全亮显示态, 闪烁频率受控制器的工作频率影响。闪烁位置同光标一样受地址指针计数器 AC 的控制。闪烁出现在有字符或光标显示的字符位时, 正常显示态为当前字符或光标的显示; 全亮显示态为该字符所有点全显示。若出现在无字符或光标显示的字符位时, 正常显示态为无显示, 全亮显示态为该字符位所有点全显示。这种闪烁方式可以设计成块光标, 如同计算机 CRT 上块状光标闪烁提示符的效果。

该指令代码表见下表所示:

指令代码	状态位			功能
	D	C	B	
08H-0BH	0	—	—	关显示
0CH	1	0	0	画面显示
0DH	1	0	1	画面, 闪烁显示
0EH	1	1	0	画面, 光标显示
0FH	1	1	1	画面, 光标, 闪烁显示

注: “—”表示任意值, 在实际应用时一般认为是“0”

3.5 光标或画面滚动

	D7	D6	D5	D4	D3	D2	D1	D0
格式	0	0	0	1	S/C	R/L	0	0

执行该指令将产生画面或光标向左或向右滚动一个字符位。如果定时间隔地执行该指令将产生画面或光标的平滑滚动。画面的滚动是在一行内连续循环进行的, 也就是说一行的第一单元与最后一个单元连接起来, 形成了闭环式的滚动。画面滚动的的显示效果如下表所示。当未开光标显示时, 执行画面滚动指令时不修改地址指针计数器 AC 值; 当有光标显示时, 由于执行任意一条滚动指令时都将使光标产生位移, 所以地址指针计数器 AC 都需要被修改。光标的滚动是在 DDRAM 内全程进行的, 它不分是一行显示还是两行显示。如果用光标的指针——地址指针计数器 AC 加一和减一功能来解释, 就能理解光标从第 1 显示位左移至第 80 显示位, 或从第 80 显示位右移至第 1 显示位原理了。光标的滚动功能可以用于搜寻需要修改的显示字符。

该指令有两个参数位:

S/C 滚动对象的选择

S/C=1 画面滚动

S/C=0 光标滚动

R/L 滚动方向的选择

R/L=1 向右滚

R/L=0 向左滚

该指令代码表如下表所示:

指令代码	状态位		功能
	S/C	R/L	
10H	0	0	光标左滚动
14H	0	1	光标右滚动
18H	1	0	画面左滚动
1CH	1	1	画面右滚动

显示位置	1	2	3	4	79	80
DDRAM 地址	00H	01H	02H	03H	4EH	4FH

(a) 一行显示 DDRAM 单元与显示字符位的原始位置关系

显示位置	1	2	3	4	79	80
DDRAM 地址	4FH	00H	01H	02H	4DH	4EH

(b) 画面向右滚动时 DDRAM 单元与显示字符位的关系变化

显示位置	1	2	3	4	79	80
DDRAM 地址	01H	02H	03H	04H	4FH	00H

(c) 画面向左滚动时 DDRAM 单元与显示字符位的关系变化

一行显示下画面滚动的地址对应关系

显示位置	1	2	3	4	39	40
第一行 DDRAM 地址	00H	01H	02H	03H	26H	27H
第二行 DDRAM 地址	40H	41H	42H	43H	66H	67H

(a) 两行显示 DDRAM 单元与显示字符位的原始位置关系

显示位置	1	2	3	4	39	40
第一行 DDRAM 地址	27H	00H	01H	02H	25H	26H
第二行 DDRAM 地址	67H	40H	41H	42H	65H	66H

(b) 画面向右滚动时 DDRAM 单元与显示字符位的关系变化

显示位置	1	2	3	4	79	80
第一行 DDRAM 地址	01H	02H	03H	04H	27H	00H
第二行 DDRAM 地址	41H	42H	43H	44H	67H	40H

(c) 画面向左滚动时 DDRAM 单元与显示字符位的关系变化

两行显示下画面滚动的地址对应关系

该指令与输入方式设置指令都可以产生光标或画面的滚动, 区别在于该指令专用于滚动功

能, 执行一次, 显示呈现一次滚动效果; 而输入方式设置指令仅是完成了一种字符输入方式的设置, 仅在计算机对 DDRAM 等进行操作时才能产生滚动的效果。

3.6 工作方式设置

	D7	D6	D5	D4	D3	D2	D1	D0
格式	0	0	1	DL	N	F	0	0

该指令设置了控制部分的工作方式, 包括有控制部分与 MPU 的接口形式和控制部分显示驱动的占空比系数等。该指令有三个参数 DL, N, F。他们的作用是:

DL 设置控制部分与 MPU 的接口形式

DL=0 数据总线为 8 位长度

DL=1 数据总线为 4 位长度

注: 总线为 8 位长度时, DB7~DB0 有效; 总线为 4 位长度时, DB7~DB4 有效。在该方式下 8 位指令代码和数据将按先高 4 位后低 4 位的顺序分两次传输。

N 设置显示的字符行数

N=0 一行字符行

N=1 两行字符行

F 设置显示字符的字体

F=0 5×8 点阵字符体

F=1 5×11 点阵字符体

该指令可以说是系统初始化设置指令, 也是唯一的软件复位指令。

3.7 CGRAM 地址设置

	D7	D6	D5	D4	D3	D2	D1	D0
格式	0	1	A5	A4	A3	A2	A1	A0

代码: 7FH~40H

该指令将 6 位的 CGRAM 地址写入地址指针计数器 AC 内, 随后 MPU 对数据的操作是对 CGRAM 的读/写操作

3.8 DDRAM 地址设置

	D7	D6	D5	D4	D3	D2	D1	D0
格式	1	A6	A5	A4	A3	A2	A1	A0

代码: FFH~80H

该指令将 6 位的 CGRAM 地址写入地址指针计数器 AC 内, 随后 MPU 对数据的操作是对 DDRAM 的读/写操作

3.9 读“忙”标志和地址指针值

	D7	D6	D5	D4	D3	D2	D1	D0
格式	BF	A6	A5	A4	A3	A2	A1	A0

BF “忙”标示位

BF=1 表示内部工作正处于忙碌中, 不能继续下条指令, 需要等待。

BF=0 表示内部工作正处于空闲中, 能继续下条指令。

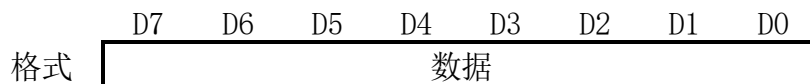
注: a) MPU 对指令寄存器通道读操作 (RS=0, R/W=1) 时, 将读出此格式的“忙”标示 BF 值和 7 位地址指针计数器 AC 的当前值。MPU 随时都可以对系统读“忙”操作。

b) BF 值反应系统的接口状态。MPU 在对系统每次操作时首先都要读 BF 值判断系统的当前接口状态, 仅有在 BF=0 时 MPU 才可以向系统写指令代码或显示数据和读出显示数据。

c) MPU 读出的地址指针计数器 AC 当前值可能是 DDRAM 地址也可能是 CGRAM 的地址,

这取决于最近一次 MPU 向 AC 写入的是哪类地址。

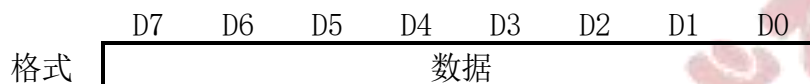
3.10 写数据



MPU 向数据寄存器通道写入数据 (RS=1, R/W=0)，系统根据当前地址指针计数器 AC 值的属性及数值将该数据送入相应的存储器内的 AC 所指的单元里。如果 AC 值为 DDRAM 地址指针，则认为写入的数据为字符代码并送入 DDRAM 内 AC 所指的单元里；如果 AC 值为 CGRAM 地址指针，则认为写入的数据是自定义字符的字模数据并送入 CGRAM 内 AC 所指的单元里。所以系统在写数据操作之前要先设置地址指针或人为的确认地址指针的属性及数值。在写入数据后地址指针计数器 AC 将根据最近设置的输入方式自动修改。

由此可知，系统在写数据操作之前要作两项工作：其一是设置或确认地址计数器 AC 值的属性及数值，以保证所写数据能够正确到位；其二是设置或确认输入方式，以保证连续写入数据时 AC 值的修改方式符合要求。

3.11 读数据

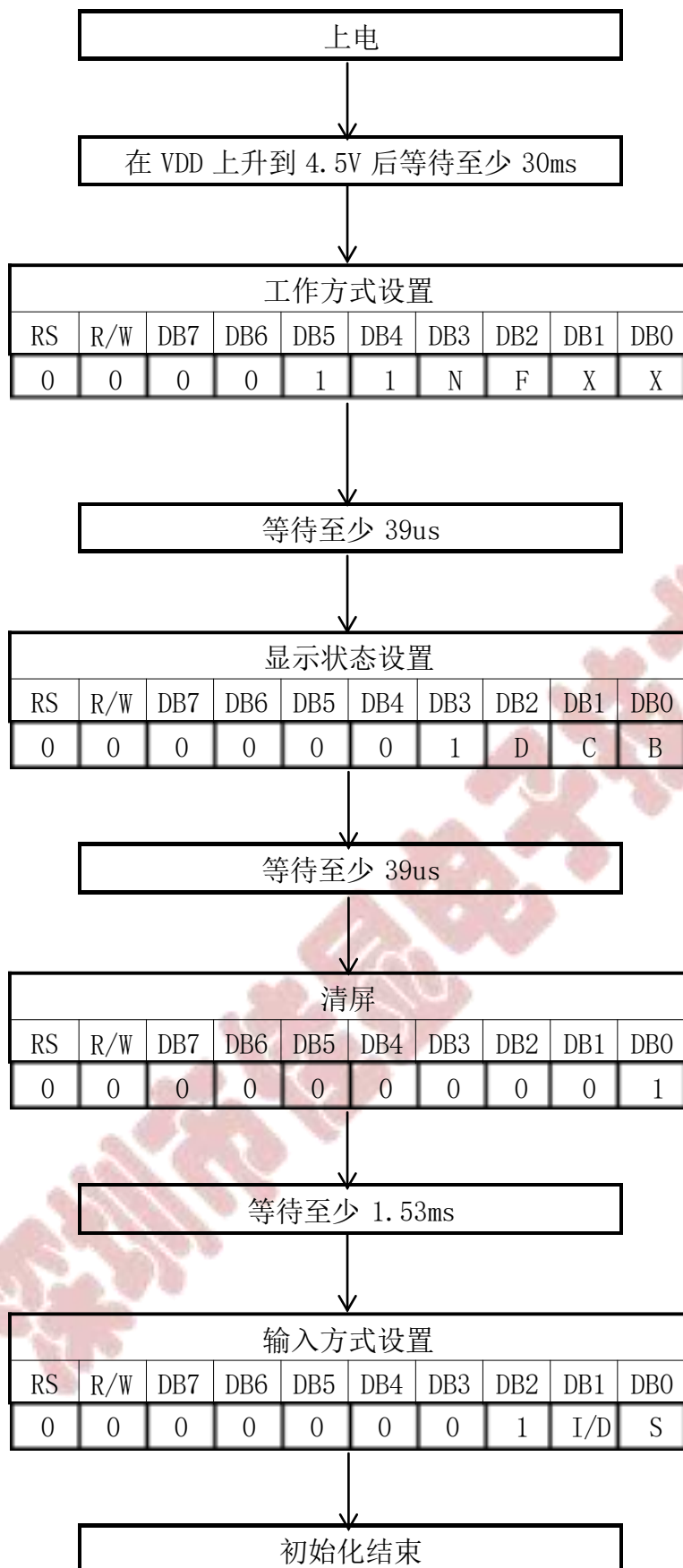


系统的读数据 (RS=1, R/W=1) 是从数据寄存器通道中数据输出寄存器读取当前所存放的数据。所以系统在首次读数据操作之前需要重新设置一次地址指针 AC 值，或用光标滚动指令将地址指针计数器 AC 值修改到所需的地址上，然后进行的读数据操作将能获得所需的数据。在读取数据后地址指针计数器 AC 将根据最近设置的输入方式自动修改。如果读操作前，没有写入 RAM 地址设置指令，且读出多次数据，则从第二个数据开始是有效的，第一个数据是不对的，因为没有时序配合 RAM 数据输出。在 DDRAM 的读操作中，光标转移指令起到了 DDRAM 地址设置指令相同的作用，同样将 RAM 数据送至输出寄存器。在读操作后，地址计数器根据输入模式指令自增或自减，在 CGRAM 读操作后，显示移位可能不能正确执行。

由此可知，系统在读数据操作之前要作两项工作：其一是设置或确认地址计数器 AC 值的属性及数值，以保证所读数据的正确性；其二是设置或确认输入方式，以保证连续读取数据时 AC 值的修改方式符合要求。

4) 初始化指令

4.1 8 位接口模式



N	0	1 行字符行
	1	2 行字符行

F	0	5×8 点钟字符体
	1	5×11 点钟字符体

注：此款产品设置 N=1

D	0	关显示
	1	开显示

C	0	光标消失
	1	光标显示

B	0	闪烁禁止
	1	闪烁启用

I/D	0	光标左移
	1	光标右移

S	0	禁止滚动
	1	允许滚动

4.2 4 位接口模式



N	0	1 行字符行
	1	2 行字符行

F	0	5×8 点钟字符体
	1	5×11 点钟字符体

注：此款产品设置 N=1

D	0	关显示
	1	开显示

C	0	光标消失
	1	光标显示

B	0	闪烁禁止
	1	闪烁启用

I/D	0	光标左移
	1	光标右移

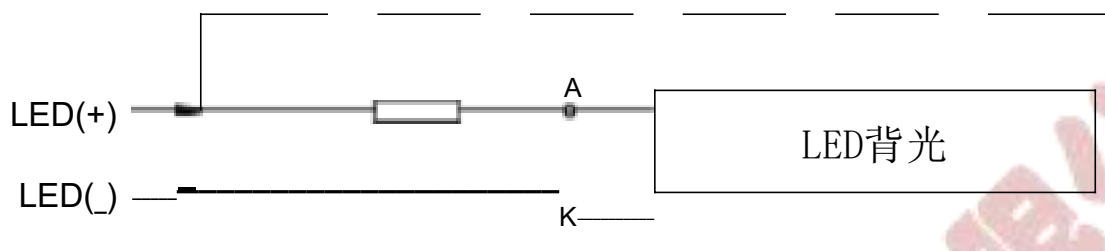
S	0	禁止滚动
	1	允许滚动

■ 背光使用说明

1 背光特性

背光类型	LED
颜 色	白色
供电电压	+5.0V

2 原理框图



3 极限特性

条件: $T_a=25^{\circ}\text{C}$

参 数	符号	条件	值	单位
极限直流正向电流	I_{fm}		20	mA
脉冲驱动极限电流	I_{fp}	1 msec 脉冲, 1/10 占空比	60	mA
反向电压	V_r		5	V
工作温度	T_{opr_LED}		$-30\sim+70$	$^{\circ}\text{C}$
存储温度	T_{stg_LED}		$-40\sim+80$	$^{\circ}\text{C}$
寿 命	$Hour_LED$	正向电流=15mA	50000	h

注: 当工作温度高于 25°C 时, I_{fm} , I_{fp} 必须降低; 电流降低率是 $-0.36\text{mA}/^{\circ}\text{C}$ (直流驱动), 或 $-0.9\text{mA}/^{\circ}\text{C}$ (脉冲驱动)。产品的工作电流不能大于对应工作温度条件下 I_{fm} 或 I_{fp} 的 70%。

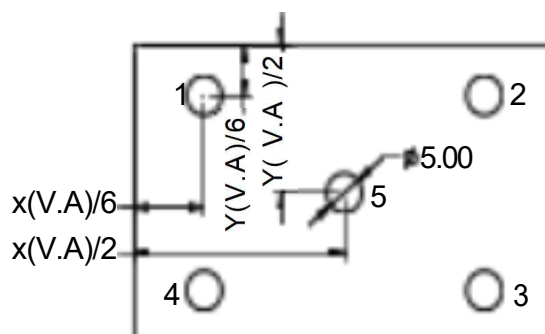
4 光电特性

条件: $T_a=25^{\circ}\text{C}$

参 数	符号	测试条件	最小	典型	最大	单位
工作电压 (LED(+)-LED(-))	VDD		—	5.0	5.5	V
正向电压 (A-K)	V_f	正向电流=15mA	2.9	3.1	3.3	mV
反向电流	I_r	$V_r=5\text{V}$			100	μA
色坐标	λ_p	正向电流=15mA	$X=0.25$ $Y=0.26$	$X=0.27$ $Y=0.28$	$X=0.29$ $Y=0.30$	nm
均匀性	Δ	正向电流=15mA	75%			

$$\text{均匀性} = \frac{\text{5个点中(1~5)亮度最小值}}{\text{5个点中(1~5)亮度最大值}} \times 100\%$$

注: 亮度值使用 BM-7 亮度色度仪测试, 测量光圈 $\phi 5\text{mm}$

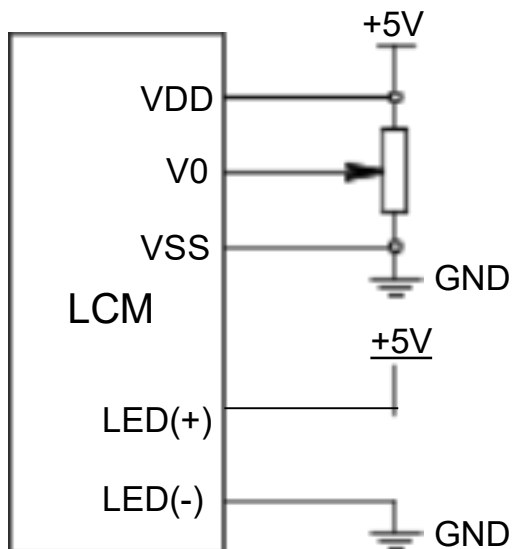


■ DDRAM 地址映射关系

字符显示位置	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
第一行 DDRAM 地址	00H	01H	02H	03H	04H	05H	06H	07H	40H	41H	42H	43H	44H	45H	46H	47H

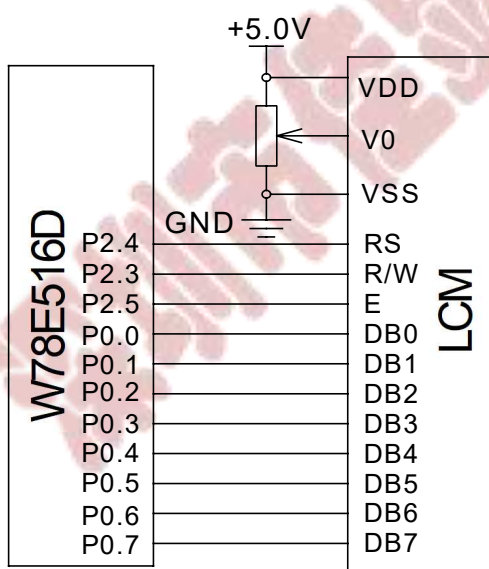
注：执行了光标或画面滚动指令后此对应关系会发生改变

■ 电源的连接方式

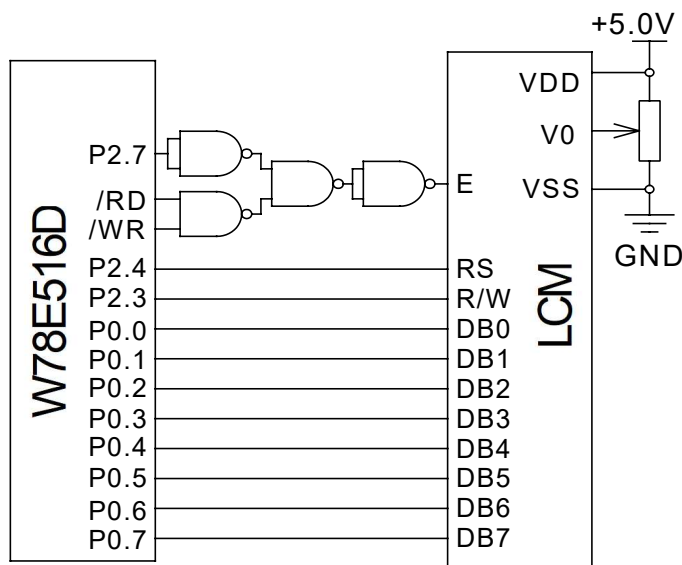


■ 示例

1 硬件部分



间接控制方式



直接控制方式

2 软件部分

1) 间接控制方式

```

/*W78E516B 12M crystal*/
#include <reg54.h>
#include <intrins.h>
//sfr CHPENR=0xF6;    //定义 W78E516B 的特殊功能寄存器
//sfr CHPCON=0xBF;    //用来打开片上的 256Byte 的 RAM
sbit LCM_RW =P2^3;//定义 RW 引脚
sbit LCM_RS = P2^4;//定义 RS 引脚
sbit LCM_E = P2^5;//定义 E 引脚
#define LCM_Data P0 //或者 sfr LCM_Data=0x80, 定义 P0 引脚
#define Busy 0x80 //用于检测 LCM 状态字中的 Busy 表示
void WriteDataLCM(unsigned char WDLCM); //写数据
void WriteCommandLCM(unsigned char WCLCM, unsigned char BuysC); //写指令
unsigned char ReadDataLCM(void);    //读数据
unsigned char ReadStatusLCM(void); //读 BF 和 AC 值
void LCMInit(void);                //LCM 初始化
void WriteCGRAM(unsigned char *Character_Add, unsigned char Character_Code); //写 CGRAM
void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char DData); //在 Y 行第 X 个字符位显示一个字符 (DData)
unsigned char ReadOneChar(unsigned char X, unsigned char Y);                //在 Y 行第 X 个字符位读取字符
void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code *DData); //在 Y 行第 X 个字符位显示一串字符 (DData)
void Delay40us(void);                //40us 延时
void Delay5Ms(void);                //5ms 延时
void Delay40Ms(void);                //40ms 延时
void Delay500Ms(void);                //400ms 延时
unsigned char code CGRAM[]={
    0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,0X1F, //全黑 0
    0X0E,0X11,0X01,0X0D,0X17,0X11,0X0E,0X00, //@ 1
    0X00,0X1F,0X00,0X1F,0X00,0X1F,0X00,0X1F, //横 2
    0X15,0X15,0X15,0X15,0X15,0X15,0X15,0X15, //竖 3
    0X0A,0X0A,0X0A,0X0A,0X0A,0X0A,0X0A,0X0A, //竖 4
    0X1F,0X11,0X11,0X11,0X11,0X11,0X11,0X1F, //方框 5
    0X00,0X11,0X0A,0X04,0X0A,0X11,0X00,0X00, //叉 6
    0X00,0X01,0X02,0X14,0X08,0X00,0X00,0X00, //勾 7
};
unsigned char code test1[]={ 'W','E','L','C','O','M','E','!',0X20,0X20,'W','E','L','C','O','M','E','!',0X20,0X20,'W','E','L',
    'C','O','M','E','!',0X20,0X20,'W','E','L','C','O','M','E','!',0X20,0X20,'\0'};
unsigned char code test2[]={ 'h','t','t','p',':','/','/', 'W','W','W','.', 'g','o','o','d','L','C','M','.', 'C','O','M',0X20,0X01,0X02,
    0X03,0X04,0X05,0X06,0X07,'\0'};
void main(void)
{
    unsigned char d,d1,d2;
    unsigned char l;
    CHPENR = 0x87;
    CHPENR = 0x59;
    CHPCON |= 0x10;
    CHPENR = 0x00;

    //输入用户程序代码
    LCM_E=0;
    LCMInit();
    for(l=0;l<8;l=l+1)
    {

```

```
    WriteCGRAM(CGRAM+I*8,I);
}

DisplayListChar(1,1,test1);           //显示 test1 数组内容
DisplayListChar(1,2,test2);           //显示 test2 数组内容
if((ReadStatusLCM() & 0x80)==0x00)
{
    d=ReadStatusLCM();
    if((d & 0x40)==0x40)
        d1=2;
    else
        d1=1;
    d2=(d & 0x3F)+1;
    if(ReadOneChar(d2-1,d1)==0x07)
        DisplayOneChar(d2,d1,0x00);
    else
        return;
}
else
    return;
Delay500Ms();
Delay500Ms();
Delay500Ms();
while(1)
{
    WriteCommandLCM(0X18,1);
    Delay500Ms();
    Delay500Ms();
    Delay500Ms();
}
}

unsigned char ReadStatusLCM(void)
{
    unsigned char recdata;
    LCM_Data=0xFF;
    LCM_RS = 0;
    LCM_RS = 0;    //RS==1,后一条用于延时
    LCM_RW = 1;
    LCM_RW = 1;    //RW==1,后一条用于延时
    LCM_E = 1;
    LCM_E = 1;
    recdata=LCM_Data;
    recdata=LCM_Data;
    Delay40us();
    LCM_E=0;
    LCM_E=0;
    return(recdata);//返回 0,空闲
}

unsigned char ReadDataLCM(void)
{
    unsigned char a;
    while((ReadStatusLCM() & 0x80) ==Busy); //检测忙
    LCM_RS = 1;    //RS==1,后一条用于延时
    LCM_RS = 1;
    LCM_RW = 1;    //RW==1,后一条用于延时
    LCM_RW = 1;
    LCM_E = 1;
```

```
    LCM_E = 1; //E 在高电平时读数据
    a=LCM_Data;
    a=LCM_Data;
    Delay40us();
    LCM_E=0; //恢复现场,已备下一步骤
    LCM_E=0;
    LCM_Data = 0xFF;//P0 口作为输入口时需置"1"
    return(a);
}

void WriteDataLCM(unsigned char WDLCM)
{
    while((ReadStatusLCM() & 0x80)==Busy); //检测忙
    LCM_RS = 1;
    LCM_RS = 1;
    LCM_RW = 0;
    LCM_RW = 0;
    LCM_E = 1;
    LCM_E = 1;
    LCM_Data = WDLCM;
    LCM_Data = WDLCM;
    LCM_E = 0; //E 下降沿写入数据
    LCM_E = 0;
    Delay40us(); //延时 5ms
}

void WriteCommandLCM(unsigned char WCLCM,unsigned char BusyC) //BusyC 为 0 时忽略忙检测
{
    if(BusyC==1) while((ReadStatusLCM() & 0x80)==Busy); //根据需要检测忙
    LCM_RW = 0;
    LCM_RW = 0;
    LCM_RS = 0;
    LCM_RS = 0;
    LCM_E = 1;
    LCM_E = 1;
    LCM_Data = WCLCM;
    LCM_Data = WCLCM;
    LCM_E = 0; //E 下降沿写入指令
    LCM_E = 0;
    Delay40us(); //延时 5ms
}

void LCMInit(void)
{
    Delay40Ms(); //启动等待,等 LCM 进入工作状态
    WriteCommandLCM(0x38,1); //功能设置,,8 位、2 行, 5*8 点阵
    Delay40us();
    WriteCommandLCM(0x0c,1); //显示开关控制, 整体显示开,光标关,光标位的字符闪烁禁止
    Delay40us();
    WriteCommandLCM(0x01,1); //显示清屏
    Delay5Ms();
    WriteCommandLCM(0x06,1); //输入方式:增量, 无移位
    Delay40us();
}

void WriteCGRAM(unsigned char *Character_Add, unsigned char Character_Code)
{
    unsigned char l;
```

```

    WriteCommandLCM(0x40+Character_Code*8,1);    //通过字符代码 (00-07)设置 CGRAM 地址
    for(l=0;l<8;l=l+1)
    {
        WriteDataLCM(*(Character_Add+l));
    }
}

void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char DData)
{
    X=X-1;
    Y=Y-1;
    if(Y==1) X=X+0x40; //当要显示第二行时,地址码+0X40
    X=X+0x80;          //算出指令码(DDRAM 地址设置),即地址码加最高位的"1"
    WriteCommandLCM(X, 0); //这里不检测忙信号, 发送地址码
    WriteDataLCM(DData);
}

unsigned char ReadOneChar(unsigned char X, unsigned char Y)
{
    unsigned char a;
    X=X-1;
    Y=Y-1;
    if(Y==1) X=X+0x40; //当要显示第二行时,地址码+0X40
    X=X+0x80;          //算出指令码(DDRAM 地址设置),即地址码加最高位的"1"
    WriteCommandLCM(X, 0); //这里不检测忙信号, 发送地址码
    a=ReadDataLCM();
    return(a);
}

void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code *DData)
{
    unsigned char ListLength= 0;
    while (DData[ListLength]!='\0') //若到达字符串尾则退出
    {
        DisplayOneChar(X, Y, *DData);
        DData++;
        X++;
    }
}

void Delay40us(void) //40us 延时
{
    unsigned char a,b;
    for(b=7;b>0;b--)
        for(a=1;a>0;a--);
}

void Delay5Ms(void) //5ms 延时
{
    unsigned char a,b;
    for(b=185;b>0;b--)
        for(a=12;a>0;a--);
}

void Delay40Ms(void) //40ms 延时
{
    unsigned char a,b;
    for(b=95;b>0;b--)
        for(a=209;a>0;a--);
}

```

```
}

void Delay500Ms(void) //500ms 延时
{
    unsigned char a,b,c;
    for(c=205;c>0;c--)
        for(b=116;b>0;b--)
            for(a=9;a>0;a--);
}
```

2) 直接控制方式

```

/*W78E516B 12M crystal*/
/*RS=P2^4, RW=P^3, Data=P0*/
#include<reg54.h>
#include<absacc.h>
#define uchar unsigned char
#define uint unsigned int
sbit control_E = P2^7;//定义 E 引脚,P2.7=1,E=0
#define READ_LCMDATA XBYTE[0x7FFF]//读数据地址,RS=1,RW=1; 011(1,1)111,1111,1111
#define WRITE_LCMDATA XBYTE[0x77FF] //写数据地址,RS=1,RW=0; 011(1,0)111,1111,1111
#define READ_LCMCOM XBYTE[0x6FFF] //读命令,RS=0,RW=1; 011(0,1)111,1111,1111
#define WRITE_LCMCOM XBYTE[0x6700] //写命令,RS=0,RW=0; 011(0,0)111,1111,1111

#define Busy 0x80 //用于检测 LCM 状态字中的 Busy 表示
void WriteDataLCM(unsigned char WDLCM); //写数据
void WriteCommandLCM(unsigned char WCLCM, unsigned char BuysC); //写指令
unsigned char ReadDataLCM(void); //读数据
unsigned char ReadStatusLCM(void); //读 BF 和 AC 值
void LCMIInit(void); //LCM 初始化
void WriteCGRAM(unsigned char *Character_Add, unsigned char Character_Code); //写 CGRAM
void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char DData);
//在 Y 行第 X 个字符位显示一个字符(DData)
unsigned char ReadOneChar(unsigned char X, unsigned char Y); //在 Y 行第 X 个字符位读取字符
void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code *DData);
//在 Y 行第 X 个字符位显示一串字符(DData)
void Delay40us(void); //40us 延时
void Delay5Ms(void); //5ms 延时
void Delay40Ms(void); //40ms 延时
void Delay500Ms(void); //400ms 延时
unsigned char code CGRAM[]={
    0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,//全黑 0
    0X0E,0X11,0X01,0X0D,0X17,0X11,0X0E,0X00,//@ 1
    0X00,0X1F,0X00,0X1F,0X00,0X1F,0X00,0X1F,//横 2
    0X15,0X15,0X15,0X15,0X15,0X15,0X15,0X15,//竖 3
    0X0A,0X0A,0X0A,0X0A,0X0A,0X0A,0X0A,0X0A,//竖 4
    0X1F,0X11,0X11,0X11,0X11,0X11,0X11,0X1F,//方框 5
    0X00,0X11,0X0A,0X04,0X0A,0X11,0X00,0X00,//叉 6
    0X00,0X01,0X02,0X14,0X08,0X00,0X00,0X00,//勾 7
};
unsigned char code test1[]={'W','E','L','C','O','M','E','!',0X20,0X20,'W','E','L','C','O','M','E','!',0X20,0X20,'W','E','L','C','O','M','E','!',0X20,0X20,'W','E','L','C','O','M','E','!',0X20,0X20,'W','E','L','C','O','M','E','!',0X20,0X20,'\0'};
unsigned char code test2[]={ 'h','t','t','p','://','W','W','W','.','g','o','o','d','L','C','M','.','C','O','M',0X20,0X01,0X02,0X03,0X04,0X05,'a',0x07,'\0'};

```

```
void main(void)
{
    unsigned char d,d1,d2;
    unsigned char l;
    CHPENR    =    0x87;
    CHPENR    =    0x59;
    CHPCON    |=    0x10;
    CHPENR    =    0x00;

    //输入用户程序代码
    control_E=1;
    LCMInit();
    for(l=0;l<8;l=l+1)
    {
        WriteCGRAM(CGRAM+l*8,l);
    }

    DisplayListChar(1,1,test1);           //显示 test1 数组内容
    DisplayListChar(1,2,test2);           //显示 test2 数组内容
    if((ReadStatusLCM() & 0x80)==0X00)
    {
        d=ReadStatusLCM();
        if((d & 0x40)==0x40)
            d1=2;
        else
            d1=1;
        d2=(d & 0x3F)+1;
        if(ReadOneChar(d2-1,d1)==0x07)
            DisplayOneChar(d2,d1,0x00);
        else
            return;
    }
    else
        return;
    Delay500Ms();
    Delay500Ms();
    Delay500Ms();

    while(1)
    {
        WriteCommandLCM(0X18,1);
        Delay500Ms();
        Delay500Ms();
        Delay500Ms();
    }
}

unsigned char ReadStatusLCM(void)
{
    unsigned char recdata;
    recdata=READ_LCMCOM;
    Delay40us();
    control_E=1;
    control_E=1;
    return(recdata);//返回 0,空闲
}

unsigned char ReadDataLCM(void)
{
    unsigned char a;
```

```
while((ReadStatusLCM() & 0x80) == Busy); //检测忙
a=READ_LCMDATA;
Delay40us();
control_E=1;
control_E=1;
return(a);
}

void WriteCommandLCM(unsigned char WCLCM,unsigned char BusyC) //BusyC 为 0 时忽略忙检测
{
    if(BusyC==1) while((ReadStatusLCM() & 0x80)==Busy); //根据需要检测忙
    WRITE_LCMCOM=WCLCM;
    control_E=1;
    control_E=1;
    Delay40us(); //延时 5ms
}

void WriteDataLCM(unsigned char WDLCM)
{
    while((ReadStatusLCM() & 0x80)==Busy); //检测忙
    WRITE_LCMDATA=WDLCM;
    control_E=1;
    control_E=1;
    Delay40us(); //延时 5ms
}

void LCMInit(void)
{
    Delay40Ms(); //启动等待,等 LCM 进入工作状态
    WriteCommandLCM(0x38,1); //功能设置,,8 位、2 行, 5*8 点阵
    Delay40us();
    WriteCommandLCM(0x0c,1); //显示开关控制, 整体显示开,光标关,光标位的字符闪烁禁止
    Delay40us();
    WriteCommandLCM(0x01,1); //显示清屏
    Delay5Ms();
    WriteCommandLCM(0x06,1); //输入方式:增量, 无移位
    Delay40us();
}

void WriteCGRAM(unsigned char *Character_Add, unsigned char Character_Code)
{
    unsigned char l;
    WriteCommandLCM(0x40+Character_Code*8,1); //通过字符代码 (00-07)设置 CGRAM 地址
    for(l=0;l<8;l=l+1)
    {
        WriteDataLCM(*(Character_Add+l));
    }
}

void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char DData)
{
    X=X-1;
    Y=Y-1;
    if(Y==1) X=X+0x40; //当要显示第二行时,地址码+0X40
    X=X+0x80; //算出指令码(DDRAM 地址设置),即地址码加最高位的"1"
    WriteCommandLCM(X, 0); //这里不检测忙信号, 发送地址码
    WriteDataLCM(DData);
}
```

```
unsigned char ReadOneChar(unsigned char X, unsigned char Y)
{
    unsigned char a;
    X=X-1;
    Y=Y-1;
    if(Y==1) X=X+0x40; //当要显示第二行时,地址码+0X40
    X=X+0x80;          //算出指令码(DDRAM 地址设置),即地址码加最高位的"1"
    WriteCommandLCM(X, 0); //这里不检测忙信号, 发送地址码
    a=ReadDataLCM();
    return(a);
}

void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code *DData)
{
    unsigned char ListLength= 0;
    while (DData[ListLength]!='\0') //若到达字符串尾则退出
    {
        DisplayOneChar(X, Y, *DData);
        DData++;
        X++;
    }
}

void Delay40us(void) //40us 延时
{
    unsigned char a,b;
    for(b=7;b>0;b--)
        for(a=1;a>0;a--);
}

void Delay5Ms(void) //5ms 延时
{
    unsigned char a,b;
    for(b=185;b>0;b--)
        for(a=12;a>0;a--);
}

void Delay40Ms(void) //40ms 延时
{
    unsigned char a,b;
    for(b=95;b>0;b--)
        for(a=209;a>0;a--);
}

void Delay500Ms(void) //500ms 延时
{
    unsigned char a,b,c;
    for(c=205;c>0;c--)
        for(b=116;b>0;b--)
            for(a=9;a>0;a--);
}
```

■ 液晶显示模块在使用过程中的注意事项

1 液晶显示模块在装配过程中的注意事项

1) 液晶显示模块在操作过程中的注意事项

我们在出厂前已经针对液晶显示模块进行了精确的装配和调试, 因此在客户使用操作时请注意以下几点:

- 1.1 液晶显示模块避免受到强烈的震动。
- 1.2 液晶显示模块避免扭动, 拆卸金属钮角。
- 1.3 液晶显示模块避免在印有线路的工作平台上操作。
- 1.4 除了液晶显示模块的焊盘 (输入/输出接线处), 禁止在线路板上的其它地方焊接。
- 1.5 避免接触, 调整, 修改导电橡胶。

2) 严防静电

液晶显示模块的控制, 驱动电路是 CMOS 电路, 极易被静电击穿, 因此我们在制造和运输整个过程中都采取了严格的防静电措施。请在使用过程中小心, 要严防静电, 以保持 CMOS IC 的正常工作状态。

- 2.1 在装配使用液晶显示模块前, 请不要将其从包装袋中取出。

液晶显示模块所使用的包装袋是经过防静电处理的特殊包装袋。因此在焊接模块连线之前请不要将其从包装袋中取出。在储存液晶显示模块时也要带有包装袋储存, 或者储存在做过防静电处理的容器中, 或者放在能充分接地的容器中储存。

- 2.2 在操作液晶显示模块时, 要始终保持操作人充分接地。

将液晶模块从防静电袋里取出时必须保持操作人的充分接地, 使人体和液晶模块保持同一电位。从防静电袋里取出的液晶显示模块需要挪动时, 应将其放在能充分接地的容器中进行挪动。

此外, 操作时应避免穿化学纤维的工作服, 最好穿棉的或者经过抗静电处理的工作服。

- 2.3 使用绝缘的, 良好接地的电烙铁进行焊接液晶显示模块。

焊接使用的电烙铁必须良好接地, 没有漏电。

- 2.4 在操作过程中所需的设备要充分接地。

在操作液晶显示模块时需要的设备, 尤其是驱动器, 必须良好接地, 没有漏电, 以避免干扰。

- 2.5 使操作台同一电位等于接地。

如果操作台用铝或钢作为接地材料, 由于它们抗阻太低, 所以可能损坏液晶显示模块或者产生电震。因此, 操作台应使用橡胶垫。

- 2.6 应慢慢揭去液晶显示模块保护膜。

液晶模块表面都有一层保护膜, 目的在于避免造成 LCD 的偏光片划伤, 沾染污渍等。如果快速揭去保护膜都将产生静电, 因此要慢慢揭去保护膜。

- 2.7 注意厂房的湿度

厂房湿度范围: 50~60%RH

3) 焊接液晶显示模块时的注意事项:

在焊接液晶显示模块时应注意以下事项:

※ 液晶显示模块上只有输入/输出连线处可以焊接。

※ 焊接所需的烙铁必须绝缘。

- 3.1 焊接时所需条件:

电铁的温度: $280^{\circ}\text{C} \pm 10^{\circ}\text{C}$

焊接时间: $< 3-4\text{S}$

焊接材料: 低熔点, 清理时易对液晶模块造成污染。此外, 为了避免焊接时焊锡对液晶显示模块的污染, 应在焊接完成后再揭去液晶显示模块的保护膜。

3.2 重复焊接时注意事项:

由于连接线是穿过模块的焊盘与模块焊接的, 所以在拆除时需等到焊锡完全熔化时再移动连接线。若焊锡未能完全熔化就用力移动连接线, 就极易造成焊盘损坏或脱落。在拆除连接线时最好使用“吸枪”。此外还应注意, 重复焊接不得超过 3 次。

4) 长时间储存时注意事项:

当液晶显示模块需要长时间储存时, 应遵循以下原则:

如果储存方法不当, 将影响偏光片的质量, 使显示效果不佳; 还容易造成焊盘的氧化, 不容易焊接。

4.1 储存时尽可能使用出厂时的原包装。

4.2 储存散装的液晶显示模块时, 应先装入防静电袋里, 封口严密。置放在免受太阳光, 日光灯照射的地方储存。

4.3 储存时应保持低湿度, 储存温度最佳范围: $0^{\circ}\text{C} \sim 35^{\circ}\text{C}$

储存时应查阅说明书, 根据不同模块的最佳储存温度和储存湿度进行储存。

5) 关于电流保护装置

液晶显示模块上没有装电流保护装置, 因此, 在使用时应预备好电流保护装置。

2 液晶显示模块在使用过程中的注意事项

- 1) 防止受到振荡, 冲击。
- 2) 防止用较硬的材料擦拭液晶显示屏表面。
- 3) 防止受到挤压。
- 4) 防止施加直流电。
- 5) 防止太阳光或日光灯的长时间照射。
- 6) 避免在高温, 高湿度的环境中储存。
- 7) 长时间储存时, 温度应高于 40°C , 湿度应低于 60%。
- 8) 液晶显示屏中的液晶材料是有害物质, 当不慎溅落到手, 身体, 衣服等处时, 绝对避免入口, 应尽快冲洗干净。

■ 附录 A

upper 4 bit	0000	0001	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CU DAN 121														
0001	121	!	1	P	Q	a	q			ト	ア	チ	ナ	ヨ	ウ
0010	121	!!	2	B	R	b	r			ト	イ	リ	キ	フ	ウ
0011	141	!!	3	C	S	c	s			ル	ウ	デ	サ	ミ	ウ
0100	151	!!	4	D	T	d	t			ノ	エ	ト	カ	ハ	ウ
0101	101	!!	5	E	U	e	u			・	オ	ナ	エ	ウ	ウ
0110	171	!!	6	F	V	f	v			ラ	カ	ニ	ヨ	ウ	ウ
0111	101	!!	7	G	W	g	w			フ	キ	ウ	リ	ウ	ウ
1000	111	!!	8	H	X	h	x			フ	ウ	ホ	リ	ウ	ウ
1001	101	!!	9	I	Y	i	y			カ	テ	ル	ル	ウ	ウ
1010	121	!!	*	J	Z	j	z			ハ	コ	ル	ル	ウ	ウ
1011	141	!!	+	K	K	ク	ク			ホ	サ	ロ	ウ	ウ	ウ
1100	151	!!	,	L	キ	リ	リ			カ	シ	フ	ウ	ウ	ウ
1101	101	!!	—	M	J	B	ク			フ	ス	ン	ウ	ウ	ウ
1110	171	!!	■	N	ク	ウ	ウ			ヨ	セ	ホ	ウ	ウ	
1111	101	!!	/	O	L	O	ウ			シ	リ	ウ	ウ	ウ	