

FYD12864-0403A 液晶显示模块服务文件



实物图片

一. 基本特性

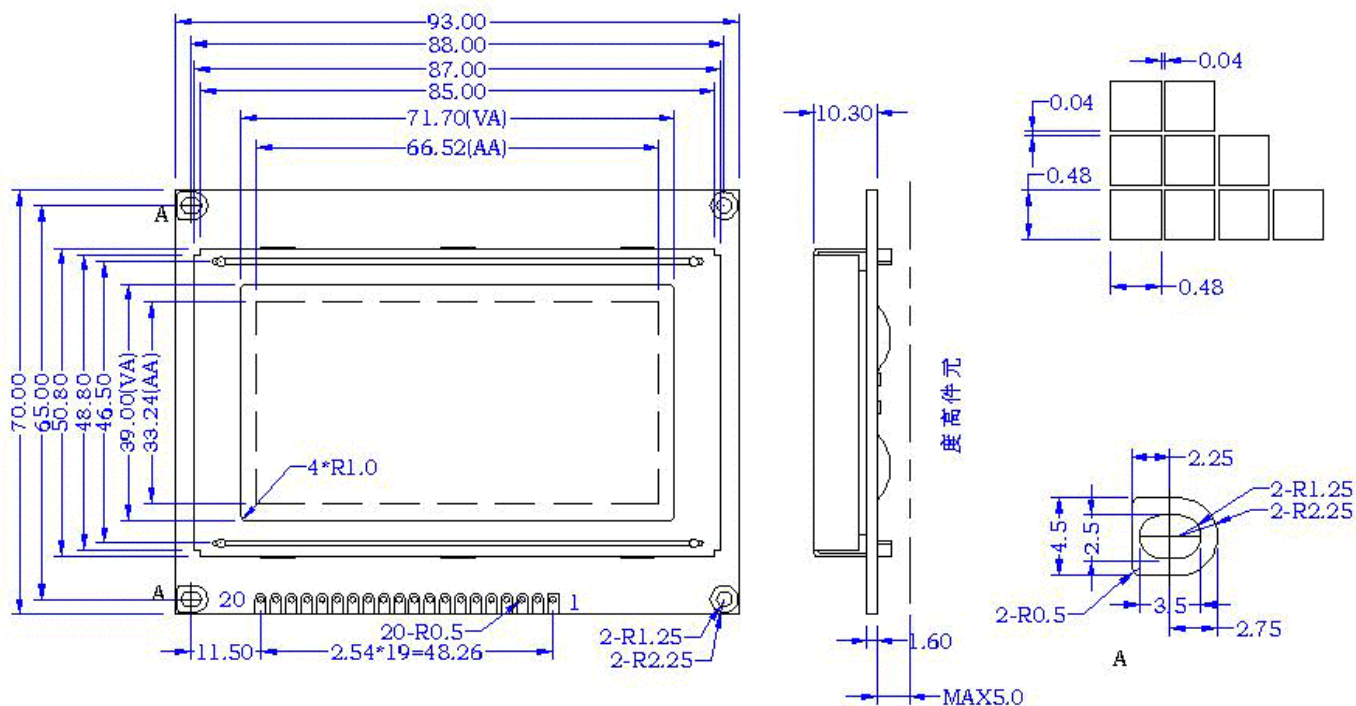
- * 电源电压 5V
- * 显示分辨率:128×64 点
- * 显示方式: STN、半透、正显
- * 驱动方式: 1/64DUTY, 1/9BIAS
- * 视角方向: 6 点
- * 背光方式: 底部黄绿 LED 背光
- * 通讯方式: 8 线并口
- * 内置 DC-DC 转换电路, 无需外加负压
- * 工作温度:-10℃ - +60℃ , 存储温度: -20℃ - 70℃

成都市飞宇达实业有限公司

www.cdfyd.com

二. 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	93.0×70.0×12.8	mm
定 位 尺 寸	88.0×65.0	mm
视 域	71.7×39	mm
点 阵 数	128×64	像素
点 距 离	0.52×0.52	mm
点 大 小	0.48×0.48	mm

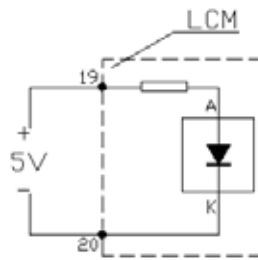
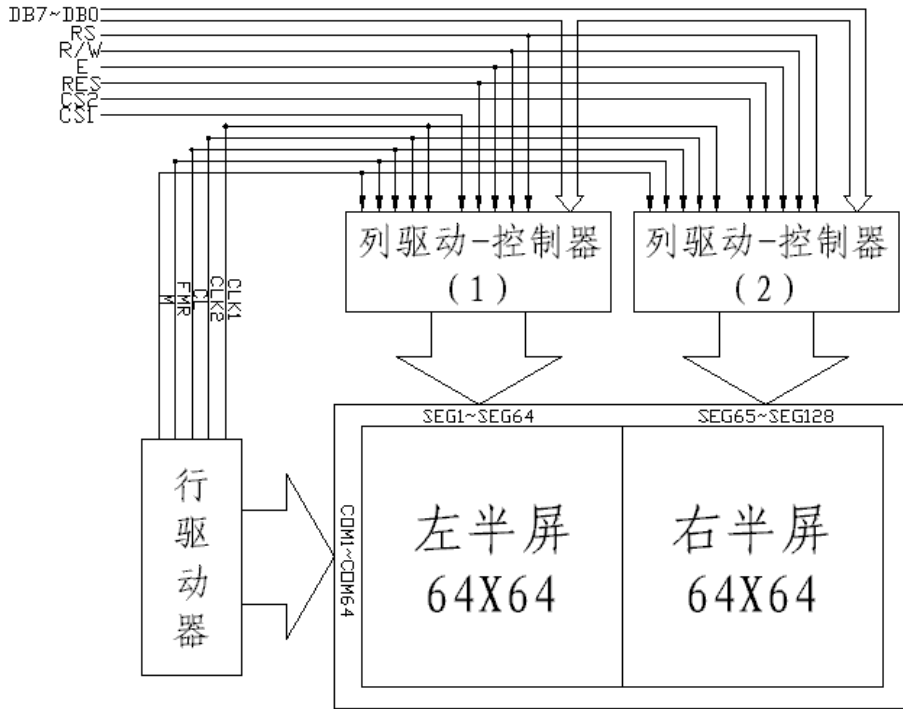
三. 硬件说明

1. 引脚特性

引脚号	引脚名称	电平	引 脚 功 能 描 述
1	VSS	0V	电源地
2	VDD	+5V	电源电压
3	V0	0~-10V	LCD 驱动负电压, 调节 LCD 对比度
4	D/I	H/L	H:DB0-DB7 为显示数据 L:DB0-DB7 为显示指令数据

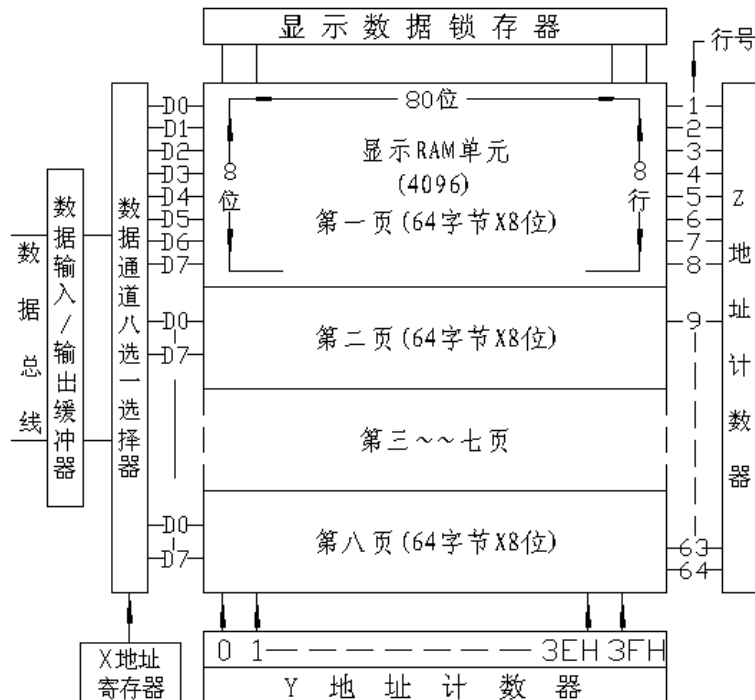
5	R/W	H/L	R/W=H, E=H 数据被读到 DB0-DB7 R/W=L, E=H—L, DB0-DB7 数据写到 IR 或 DR
6	E	H/L	使能信号:R/W=L. E 信号下降沿锁存 DB0-DB7;R/W=H, E=H, DDRAM 数据读到 DB0-DB7
7	DB0	H/L	8 位三态并行数据总线
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	CS1	H/L	片选信号, 当 CS1=H 时, 液晶右半屏显示
16	CS2	H/L	片选信号, 当 CS2=H 时, 液晶左半屏显示
17	/RET	H/L	复位信号, RET=0 有效
18	VEE	-10V	输出-10V 的负电压(单电源供电)
19	A	+5V	背光电源, 背光功耗 \leq 300mA
20	K	0V	

2. 原理简图



背光接线图

3. 硬件功能描述

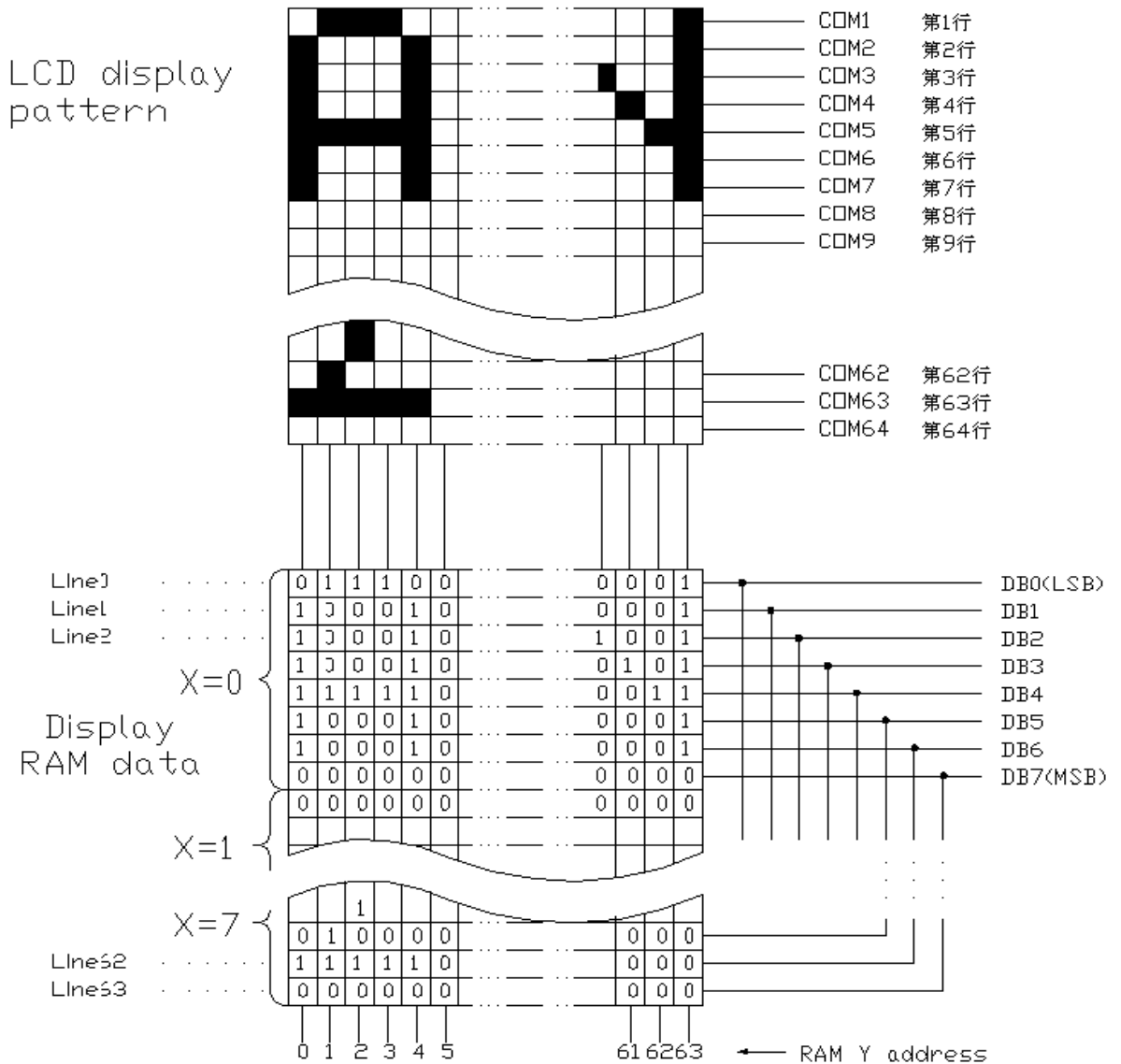


成都市飞宇达实业有限公司

www.cdfyd.com

1) 显示数据 RAM(DDRAM)

DDRAM (64×8×8 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(见下图)



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

成都市飞宇达实业有限公司

www.cdfyd.com

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。

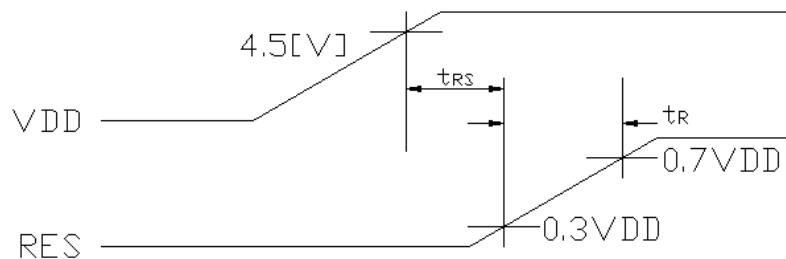
12) 复位端/RES

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



四. 电气特性

1. 极限参数

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 _B	-0.3 to VDD+0.3	V	*1,*3
Operating Temperature	T _{OPR}	-20 to +70	°C	
Storage Temperature	T _{STG}	-30 to +80	°C	

*1. Based on VSS=0V

*2. Applies to V_{LCD}

*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 _{IH}	-	2.0	-	VDD	V	*1
Input Low Voltage	V _{IL}	-	0	-	0.8	V	*1
Output High Voltage	V _{OH}	I _{OH} =-200uA	2.4	-	-	V	*2
Output Low Voltage	V _{OL}	I _{OL} =1.6mA		-	0.4	V	*2
Input Leakage Current	I _{LKG}	V _{IN} =VSS~VDD	-1.0	-	1.0	uA	*3
Three-state(OFF) input Current	I _{TSL}	V _{IN} =VSS~VDD	-5.0	-	5.0	uA	*4
Operating Current	I _{DD1}	During Display	-	-	0.5	mA	*5
	I _{DD2}	During Access			2	mA	*5
On Resistance	R _{ON}		-	-	7.5	KΩ	*6

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

*2. DB0~DB7

*3. Except DB0~DB7

*4. DB0~DB7 at High Impedance

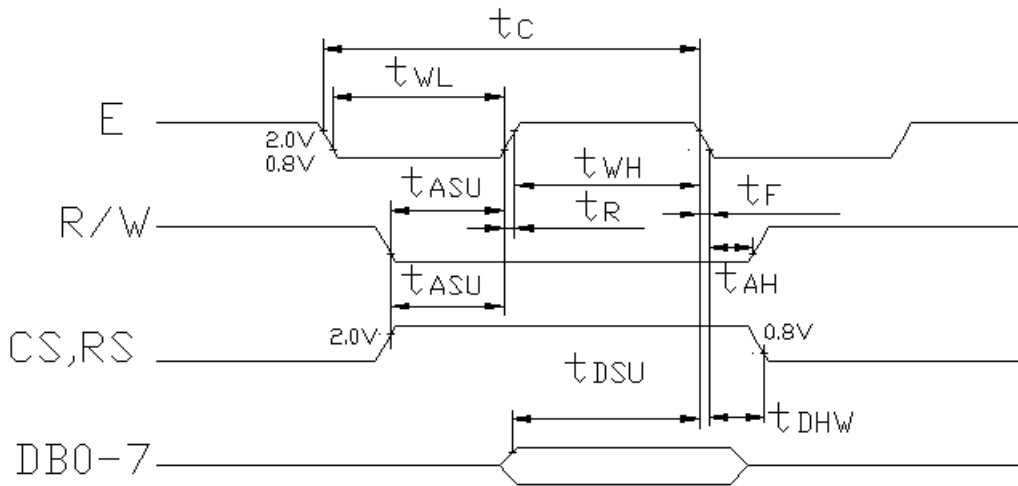
*5. 1/64 duty, FCLK=250KHZ, Frame Frequency=70HZ, Output: NO Load

*6. VDD~VEE=15.5

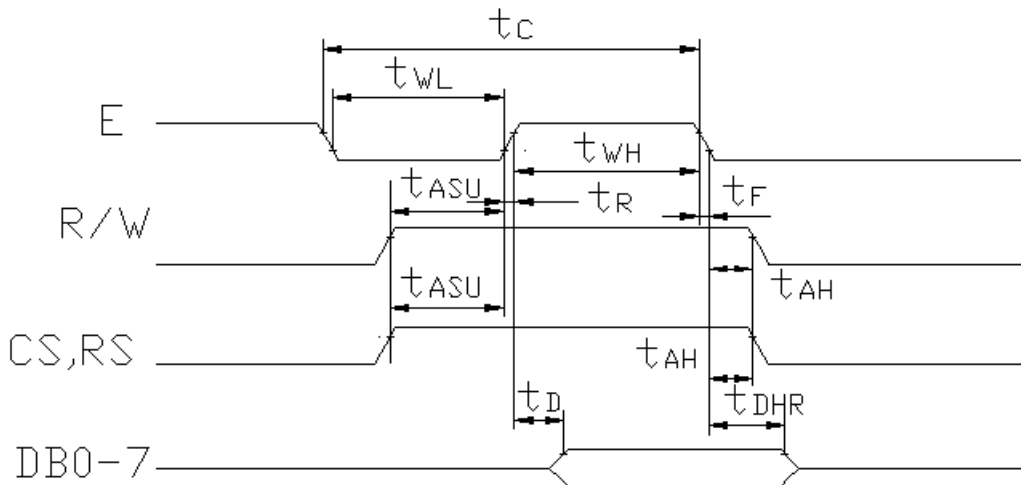
3. 交流特性 (VDD=+5V±10%, VSS=0V, Ta=-20~+70°C)

项 目	名称	Min	Typ	Max	单 位
E Cycle	t _C	1000	-	-	ns
E Hight Level Width	t _{WH}	450	-	-	ns
E Low Level Width	t _{WL}	450	-	-	ns
E Rise Time	t _R	-	-	25	ns
E Fall Time	t _F	-	-	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

五. 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	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							

指令功能详解

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 接口控制电路已处于“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	RS	R/W	E	DB7~DB0	功能
X	X	X	X	0	高阻	总线释放
1	1	0	0	下降沿	输入	写指令代码

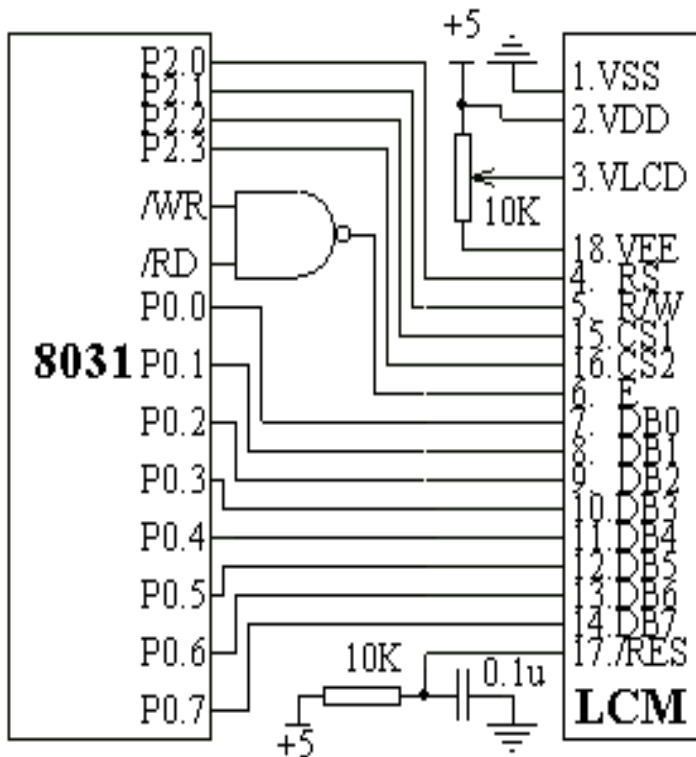
CS1	CS2	RS	R/W	E	DB7~DB0	功能
1	1	0	1	1	输出	读状态字
1	1	1	0	下降沿	输入	写显示数据
1	1	1	1	1	输出	读显示数据

3. DDRAM 地址表

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

六. 应用举例

1. 直接访问方式的子程序



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 ;写指令代码地址右

CRADD2 EQU 0A00H ;读状态字地址右

DWADD2 EQU 0900H ;写显示数据地址右

DRADD2 EQU 0B00H ;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

; 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. 右区驱动子程序

1) 写指令代码子程序 (右)

PRR0: PUSH DPL	
PUSH DPH	
MOV DPTR, #CRADD2	; 设置读状态字地址
PRR01: MOVX A, @DPTR	; 读状态字
JB ACC. 7, PRR01	; 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	
RET	

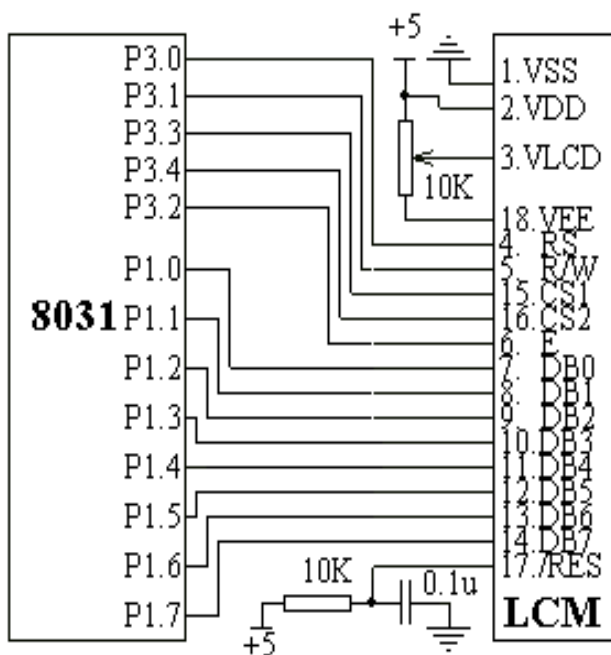
2) 写显示数据子程序 (右)

PRR1: PUSH DPL	
PUSH DPH	
MOV DPTR, #CRADD2	; 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, #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 (右)

PRR2: PUSH DPL	
PUSH DPH	
MOV DPTR, #CRADD2	; 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, #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	

2. 间接访问方式的子程序



CS1 EQU P3.3 ;片选左
 CS2 EQU P3.4 ;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 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 RS           ; RS=0
    
```

```

        SETB RW                ; R/W=1
PRL11: MOV P1, #OFFH          ; 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
        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 RS                ; RS=0
        SETB RW               ; R/W=1
PRL21: MOV P1, #OFFH          ; 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, #OFFH          ; P1 口置 “1”
        SETB E                 ; E=1
        MOV DAT, P1            ; 写数据
        CLR E                  ; E=0
        RET

```

2. 右区驱动子程序

1) 写指令代码子程序 (右)

```

PRR0:  CLR CS1
        SETB CS2
        CLR RS                ; RS=0
        SETB RW               ; R/W=1
PRR01: MOV P1, #OFFH          ; 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
      SETB CS2
      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 (右)

```
PRR2: CLR CS1
      SETB CS2
      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 8051 汇编为例, [如需要示例源程序, 请向成都市飞宇达公司索要](#))

```
CS1 EQU P1.6;片选左
CS2 EQU P1.7;片选右
RS EQU P1.3 ;寄存器选择信号
RW EQU P1.4 ;读/写选择信号
E EQU P1.5 ;使能信号
RST EQU P2.7
COM EQU 20H ;指令数据寄存器
```

```

DAT      EQU    21H      ;显示数据寄存器
;数据口为 P0,且数据总线接 20K 上拉电阻
;*****
      ORG 0000H
      LJMP STR
      ORG 100H
STR:MOV  COM, #0COH      ;设置显示起始行为第一行
      LCALL PRL0
      LCALL PRRO
      MOV  COM, #3FH      ;开显示设置
      LCALL PRL0
      LCALL PRRO

      MOV  A, #0B8H
      MOV  COM, A        ;页面地址设置
      LCALL PRL0
      LCALL PRRO
      MOV  COM, #40H      ;列地址设置为“0”
      LCALL PRL0
      LCALL PRRO

      MOV  DPTR, #TAB1
      LCALL LEFT
      MOV  DPTR, #TAB2
      LCALL RIGHT      ;写入一幅图画“飞宇达液晶显示世界”
      LCALL TIME      ;延时程序
      LJMP STR
;*****
LEFT:  MOV    R1, #0B8H
AB:    MOV    COM, R1      ;第一页
      LCALL  PRL0      ;左半屏写指令子程序
      MOV    COM, #40H      ;第一列
      LCALL  PRL0      ;左半屏写指令子程序
      MOV    R0, #64
S027B: MOV    A, #00H
      MOVC   A, @A+DPTR
      MOV    DAT, A
      LCALL  PRL1      ;左半屏写数据子程序
      INC    DPTR
      DJNZ   R0, S027B
      INC    R1
      CJNE   R1, #0COH, AB
      RET
;=====以上是左半屏写数据子程序
RIGHT: MOV    R1, #0B8H

```

```

ABC:    MOV     COM, R1           ;第一页
        LCALL  PRR0             ;右半屏写指令子程序
        MOV     COM, #40H       ;第一列
        LCALL  PRR0             ;右半屏写指令子程序
        MOV     R0, #64
S027D:  MOV     A, #00H
        MOVC   A, @A+DPTR
        MOV     DAT, A
        LCALL  PRR1             ;右半屏写数据子程序
        INC     DPTR
        DJNZ   R0, S027D
        INC     R1
        CJNE   R1, #0COH, ABC
        RET

```

*****以上是右半屏写数据子程序

```

PRL0:
SETB CS1
CLR CS2
CLR RS           ;RS=0
SETB RW         ;R/W=1
PRL01:MOV P0, #0FFH ;P0 口置“1”
SETB E         ;E=1
MOV A, P0      ;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 P0, COM   ;写指令代码
SETB E       ;E=1
CLR E       ;E=0
RET

```

```

PRL1:
SETB CS1
CLR CS2
CLR RS           ;RS=0
SETB RW         ;R/W=1
PRL11:MOV P0, #0FFH ;P0 口置“1”
SETB E         ;E=1
MOV A, P0      ;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
MOV P0, DAT   ;写数据

```

```

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

PRL2:
SETB CS1
CLR CS2
CLR RS                ;RS=0
SETB RW              ;R/W=1
PRL21:MOV P0, #0FFH   ;P0 口置 “1”
SETB E                ;E=1
MOV A, P0             ;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 P0, #0FFH        ;P0 口置 “1”
SETB E                ;E=1
MOV DAT, P0          ;写数据
CLR E                 ;E=0
RET

PRR0:
CLR CS1
SETB CS2
CLR RS                ;RS=0
SETB RW              ;R/W=1
PRR01:MOV P0, #0FFH   ;P0 口置 “1”
SETB E                ;E=1
MOV A, P0             ;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 P0, COM          ;写指令代码
SETB E                ;E=1
CLR E                 ;E=0
RET

PRR1:
CLR CS1
SETB CS2
CLR RS                ;RS=0
SETB RW              ;R/W=1
PRR11:MOV P0, #0FFH   ;P0 口置 “1”
SETB E                ;E=1
MOV A, P0             ;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 P0, DAT         ;写数据
SETB E              ;E=1
CLR E                ;E=0
RET

```

PRR2:

```

CLR CS1
SETB CS2
CLR RS              ;RS=0
SETB RW            ;R/W=1
PRR21: MOV P0, #0FFH ;P0口置"1"
SETB E              ;E=1
MOV A, P0           ;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 P0, #0FFH      ;P0口置"1"
SETB E              ;E=1
MOV DAT, P0        ;写数据
CLR E                ;E=0
RET

```

TIME:

```

S0240: MOV R5, #08H ;延时
L0242: MOV R7, #0FFH
L0244: MOV R6, #0FFH
L0246: DJNZ R6, $
        DJNZ R7, L0244
        DJNZ R5, L0242
RET

```

TAB1:

```

DB 0FFH, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 081H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 021H, 0C1H, 081H, 087H
DB 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 0FFH, 0FFH, 001H, 000H, 000H, 080H
DB 0C0H, 070H, 038H, 01EH, 00CH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 080H, 0C0H, 0F8H, 078H, 008H, 008H, 008H, 008H, 008H, 008H, 008H, 008H, 008H, 008H, 008H, 008H, 008H
DB 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 07FH, 0FFH, 002H, 006H, 00DH, 008H
DB 018H, 030H, 070H, 0E0H, 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H

```

DB 000H, 000H, 000H, 000H, 004H, 004H, 004H, 004H, 004H, 004H, 004H, 004H, 004H, 004H, 0FCH, 0FCH
DB 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 00FH, 03FH, 0F8H, 0E0H, 080H
DB 000H, 000H, 000H, 000H, 001H, 000H, 080H, 0F8H, 000H, 000H, 000H, 000H, 000H, 000H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 0FFH, 0FFH
DB 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 001H, 003H
DB 003H, 007H, 006H, 00EH, 00CH, 00CH, 01FH, 01FH, 01CH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 004H, 004H, 004H, 00CH, 01CH, 01EH, 00FH, 007H
DB 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 010H, 020H, 040H, 000H, 020H, 020H, 0E0H, 020H, 0F0H, 020H, 020H, 020H, 020H, 020H, 000H, 000H
DB 000H, 0A0H, 050H, 050H, 050H, 050H, 050H, 050H, 050H, 050H, 050H, 0F0H, 000H, 000H, 000H, 000H
DB 000H, 0F0H, 090H, 090H, 090H, 090H, 090H, 090H, 090H, 090H, 090H, 090H, 0F0H, 000H, 000H, 000H
DB 0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 001H, 0E2H, 024H, 018H, 0FCH, 003H, 010H, 01CH, 023H, 0C5H, 099H, 061H, 011H, 00FH, 000H, 000H
DB 0F8H, 049H, 04AH, 04AH, 04AH, 0FAH, 002H, 0F2H, 04AH, 04AH, 04AH, 049H, 048H, 0F8H, 000H, 000H
DB 000H, 00BH, 074H, 084H, 0FCH, 004H, 004H, 004H, 0FCH, 004H, 084H, 074H, 00BH, 000H, 000H, 000H
DB 0FFH, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H
DB 083H, 080H, 080H, 080H, 081H, 080H, 082H, 082H, 081H, 080H, 080H, 081H, 081H, 082H, 082H, 080H
DB 083H, 082H, 082H, 082H, 082H, 083H, 080H, 081H, 082H, 082H, 082H, 082H, 082H, 082H, 083H, 080H, 080H
DB 082H, 082H, 082H, 082H, 083H, 082H, 082H, 082H, 083H, 082H, 082H, 082H, 082H, 082H, 082H, 080H, 080H

TAB2:

DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 041H, 081H, 081H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 0C1H, 0C1H, 081H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 0FFH
DB 008H, 008H, 008H, 008H, 008H, 008H, 008H, 088H, 048H, 038H, 03CH, 018H, 010H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H, 00FH, 00EH, 000H, 080H, 080H, 080H, 080H
DB 080H, 080H, 080H, 080H, 080H, 0FFH, 0FFH, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 0C0H, 0E0H
DB 0C0H, 080H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH
DB 004H, 004H, 004H, 004H, 004H, 002H, 003H, 003H, 002H, 080H, 0C0H, 080H, 080H, 000H, 000H, 000H
DB 000H, 000H, 000H, 004H, 004H, 004H, 004H, 004H, 0FEH, 0FEH, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 0C0H, 0F8H, 03FH, 027H, 040H, 0C0H, 080H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH
DB 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 001H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 080H, 0C0H, 040H, 03FH, 07FH, 080H, 080H, 040H, 020H, 020H, 010H
DB 008H, 00EH, 007H, 001H, 000H, 000H, 000H, 000H, 000H, 001H, 003H, 007H, 00EH, 03CH, 078H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 002H, 007H, 001H, 000H, 000H, 000H, 000H, 000H, 001H, 003H, 002H, 006H, 006H
DB 004H, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH, 00CH
DB 004H, 004H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH
DB 000H, 010H, 010H, 010H, 010H, 010H, 010H, 010H, 010H, 010H, 010H, 010H, 010H, 000H, 000H, 000H
DB 000H, 000H, 0F0H, 000H, 000H, 000H, 0F0H, 000H, 000H, 000H, 0F0H, 000H, 000H, 000H, 000H, 000H
DB 000H, 0F0H, 050H, 050H, 050H, 050H, 0F0H, 050H, 050H, 050H, 050H, 050H, 0F0H, 000H, 000H, 000H

```

DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH
DB 001H, 0C1H, 031H, 00DH, 001H, 001H, 001H, 0FFH, 001H, 001H, 005H, 019H, 061H, 081H, 000H, 001H
DB 001H, 001H, 0FFH, 001H, 001H, 001H, 03FH, 041H, 041H, 041H, 041H, 03FH, 001H, 001H, 000H, 020H
DB 020H, 013H, 012H, 08AH, 076H, 002H, 003H, 002H, 002H, 0F6H, 00AH, 012H, 011H, 020H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH
DB 081H, 080H, 080H, 080H, 082H, 082H, 082H, 083H, 080H, 080H, 080H, 080H, 080H, 080H, 081H, 080H, 080H
DB 080H, 080H, 083H, 082H, 082H, 082H, 082H, 082H, 082H, 082H, 082H, 082H, 082H, 082H, 080H, 080H
DB 080H, 080H, 082H, 081H, 080H, 080H, 080H, 080H, 080H, 083H, 080H, 080H, 080H, 080H, 080H, 080H, 080H
DB 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 080H, 0FFH
END

```

五. 液晶显示模块使用注意事项

1. 谨慎处理和清洁 LCD

清洁 LCD 表面时, 使用沾有异丙醇溶剂的软无毛布轻轻擦拭, 不能使用干的或硬的布料擦拭 LCD 表面, 那将会伤害偏光片的表面。不能使用下列的溶剂: 水、乙烯酮、芬芳溶剂

2. LCD 模块使用 C-MOS LSI 驱动, 因此我们建议你:

将不用的输入端连接到 V_{DD} 或 V_{SS} 上, 开电前不要输入任何信号, 工作区、工具及操作者身体都需接地, 以防静电。

3. 可靠包装

对于模块应同对待 LCD 一样, 避免从高处落下, 受到强烈的震动。

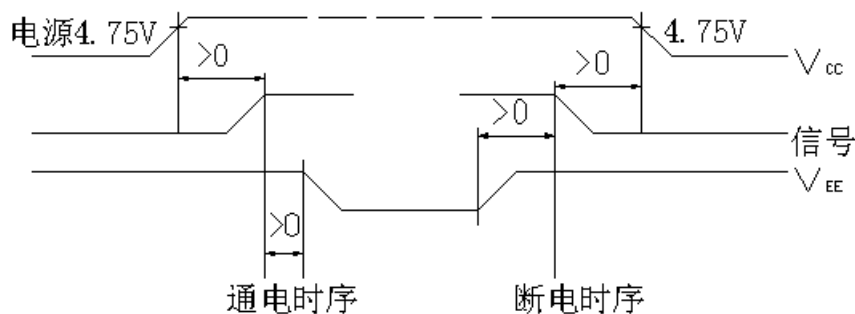
防止模块老化, 模块不能在有阳光直接照射或高温 / 高湿度条件下操作或储存。

4. 谨慎操作

在指定的限制电压下驱动 LCD 模块, 因为电压超出限制范围会缩短 LCD 模块的使用寿命。

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。



由于使用直流电驱动 LCD 模块会产生化学反应使模块出现不应该的退化, 因此避免用直流电驱动 LCD 模块。

当温度低于操作温度范围时, 响应时间将被延迟, 另一方面工作温度过高, 模块显示发黑。但是这些现象并不意味模块本身有故障, 在指定的操作温度下模块又会恢复正常。

5. 储存

如打算长期储存, 推荐以下方法。

成都市飞宇达实业有限公司

www.cdfyd.com

放在一个不漏气的密封聚乙烯袋中，不用放干燥剂。
放置在一个没有阳光直接照射，且满足储存温度范围的黑地方。
储存时不允许有东西碰到偏光片表面。

6. 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。