

SG12864-15B 系列说明书

(Version1.0)

可选型号：

产品型号 及其后缀	LCD 类型 (显示模式)	背光类型 (LED)	时序方 式	逻辑电压 (VDD)	背光 电压	接口方式 及其预留配置
SG12864SYD-15BLYE(ND)	STN 黄绿底色	黄绿底背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864SYD-15BSYE(ND)	STN 黄绿底色	黄绿侧背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864FPD-15BSWE(ND)	FSTN 黑白正像	白背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864SBD-15BSWE(ND)	STN 蓝底色	白背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864SYD-15BLYE(TC)	STN 黄绿底色	黄绿底背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864SYD-15BSYE(TC)	STN 黄绿底色	黄绿侧背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864FPD-15BSWE(TC)	FSTN 黑白正像	白背光	80 时序	5.0V	5.0V	单排插针焊位
SG12864SBD-15BSWE(TC)	STN 蓝底色	白背光	80 时序	5.0V	5.0V	单排插针焊位

SG12864-15B 系列产品可扩展功能介绍：

1. 可以选择在逻辑电压为 3.3V 环境下使用，但需要单独声明，特殊配置。
2. 可以选择使用 LCM 中配置的温度补偿电路。

注：

1. 以上列出的 LCD 类型为我公司的标准品。
2. (ND) 表示：LCM 中没有配置负电源，需要客户外部提供。
3. (TC) 表示：LCM 中配置负电源和温度补偿电路。
4. 如有其他需求，请与我公司销售部联系！

销售部：

电话：010-51601528 13683669085
13910537028

QQ：1056456564

文档修订记录

修订 次第	修订 日期	修订人	修订前 版本号	修订			批准人
				页 次	章节 编号	修订内容简述	
1	2007-1-23	王梅	/	/	/	新建文档	趙鵬

目 录

1、液晶显示模块整体描述	4
2. 最大典型值	5
3. 电气特性	5
4. 光学特性	6
5. 光学特性测定方法	6
6. 原理框图	7
7. 时序图	7
8. 功能说明及指令集	8
9. 字符代码表	17
10. LCD 驱动电源连接方式	18
11. 出厂测试报告	19
12. 接口引脚定义	20
13. 外形尺寸图纸	20
14.液晶显示模块在使用过程中的注意事项	22
15、驱动程序	24

1、液晶显示模块整体描述

项目	说明	单位
液晶显示模块组成	液晶显示屏,背光灯箱,线路板,铁框,导电橡胶等	
液晶显示屏类型	正像反射型,半透型,负像型	
液晶屏显示类型	STN 型:黄绿模式,灰模式,蓝模式	
	FSTN 型:黑白模式	
液晶显示屏视角	6 O'clock or 12 O'clock	
液晶模块外形尺寸 (LED*)	78.00(长)×70.00(宽)×13.40(厚)	mm
液晶模块视域	62.00(长)×44.00(宽)	mm
液晶模块铁框尺寸 (LED*)	76.00(长)×58.00(宽)×13.40(厚)	mm
液晶显示模块点阵数	128×64 点阵	
液晶显示屏点尺寸	0.39(长)×0.55(宽)	mm
液晶显示屏点间距	0.44(长)×0.60(宽)	mm
液晶显示屏占空比	1/64	
液晶显示屏偏置电压	1/9	
液晶显示模块控制器	T6963C or SAP1024B or RA6963	
液晶显示模块使用温度范围(E*)	-20~+70	
液晶显示模块存储温度范围(E*)	-30~+80	
背光灯箱	LED: 黄绿底色、白色、蓝色	
液晶显示模块数据输入格式	八位并行输入格式,80 时序方式	
电源输入电压	3.3~5V 输入供电,内置 DC/DC 电路	V
液晶显示模块理论寿命	50,000	小时

注意:

LED*: LED 背光
E*: 宽温等级

2. 最大典型值

2.1 电气最大典型值

V_{SS}=0V

Item	Symbol	Min	Max	Unit	Note
逻辑电源	V _{dd} -V _{SS}	0	7.0	V	
LCD 驱动电压	V _{dd} -V _o	0	13.0	V	
I/O 口输入电压	V _i	0	V _{dd}	V	

2.2 使用环境最大典型值

Item	Symbol	Min	Max	Unit
工作温度	T ₀	-20	+70	
储存温度	T _s	-30	+80	
湿度	---	---	85	%RH

3. 电气特性

3.1 电气特性

V_{SS}=0V

Item	Symbol	Condition	Min	Typ	Max	Unit
逻辑电压	V _{dd}	----	2.7	5.0	5.5	V
LCD 驱动电压	V _{dd} -V _o	----	10.5	---	13.0	V
输入电压最大值	V _{IH}	----	V _{dd} -2.2	---	V _{dd}	V
输入电压最小值	V _{IL}	----	0	---	0.8	V
频率	FELM	V _{dd} =5V	65	78	85	Hz

3.2 LED 背光特性说明书

Color	Item	Symbol	Min	Typ	Max	Unit	Condition
黄绿底背光	正向电压	V _f	4.0	4.1	4.2	V	I _f =440mA
黄绿侧背光	正向电压	V _f	2.8	3.0	3.2	V	I _f =65mA
白色侧背光	正向电压	V _f	2.8	3.0	3.2	V	I _f =80mA
蓝色侧背光	正向电压	V _f	2.8	3.0	3.2	V	I _f =80mA

4. 光学特性

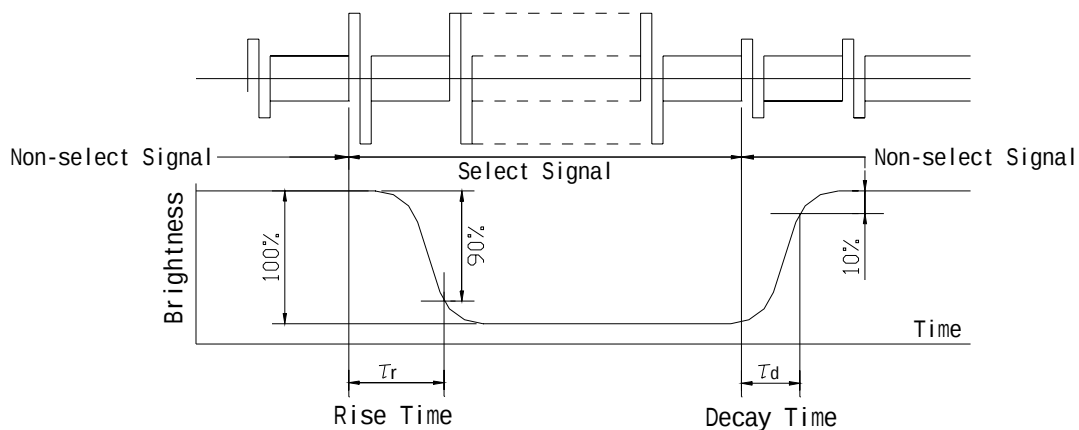
STN TYPE

Ta=25

Item	Symbol	Condition	Min	Typ	Max	Unit	Reference
Viewing angle	θ	K 2.0 $\varphi=0^\circ$	40°	---	---	deg	
Contrast ration	K	$\theta=5^\circ \varphi=0^\circ$	---	5	---	---	
Response time(rise)	Tr	$\theta=5^\circ \varphi=0^\circ$	---	110	165	ms	
Response time(fall)	Tf	$\theta=5^\circ \varphi=0^\circ$	---	110	165	ms	

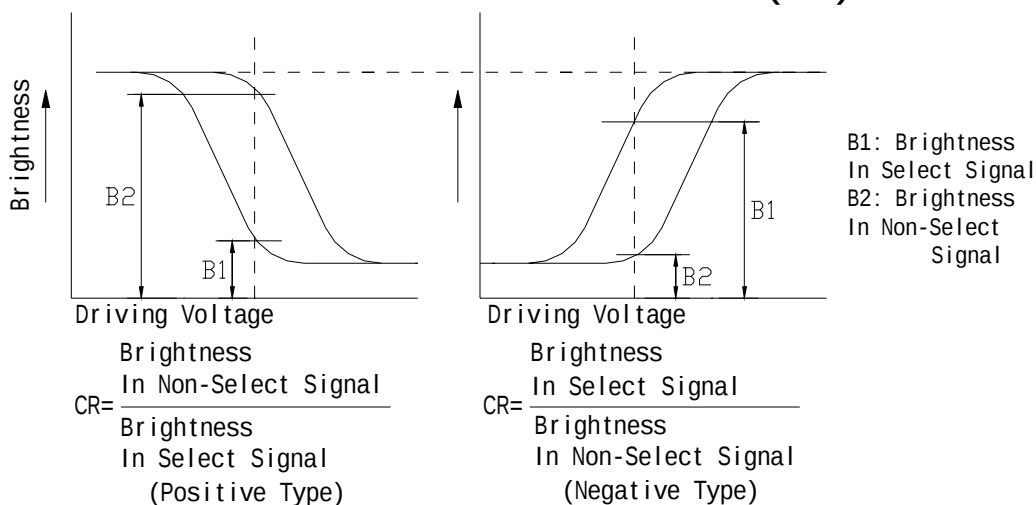
5. 光学特性测定方法

● Definition of Optical Response Time

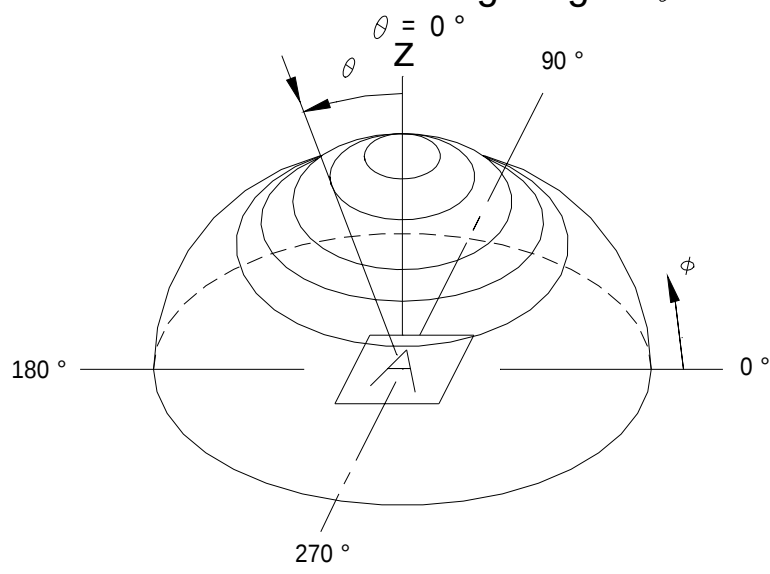


In case of Negative type,
wave form of changing brightness becomes reverse
(Non Select Signals:0%,Select Signals:100%)

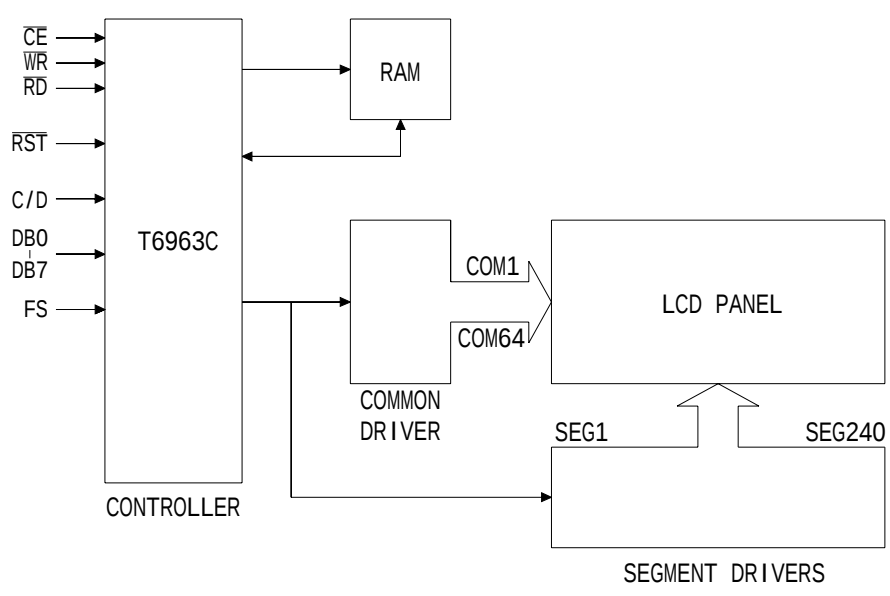
● Definition of Contrast Ratio(CR)



● Definition of Viewing Angle θ and ϕ



6. 原理框图

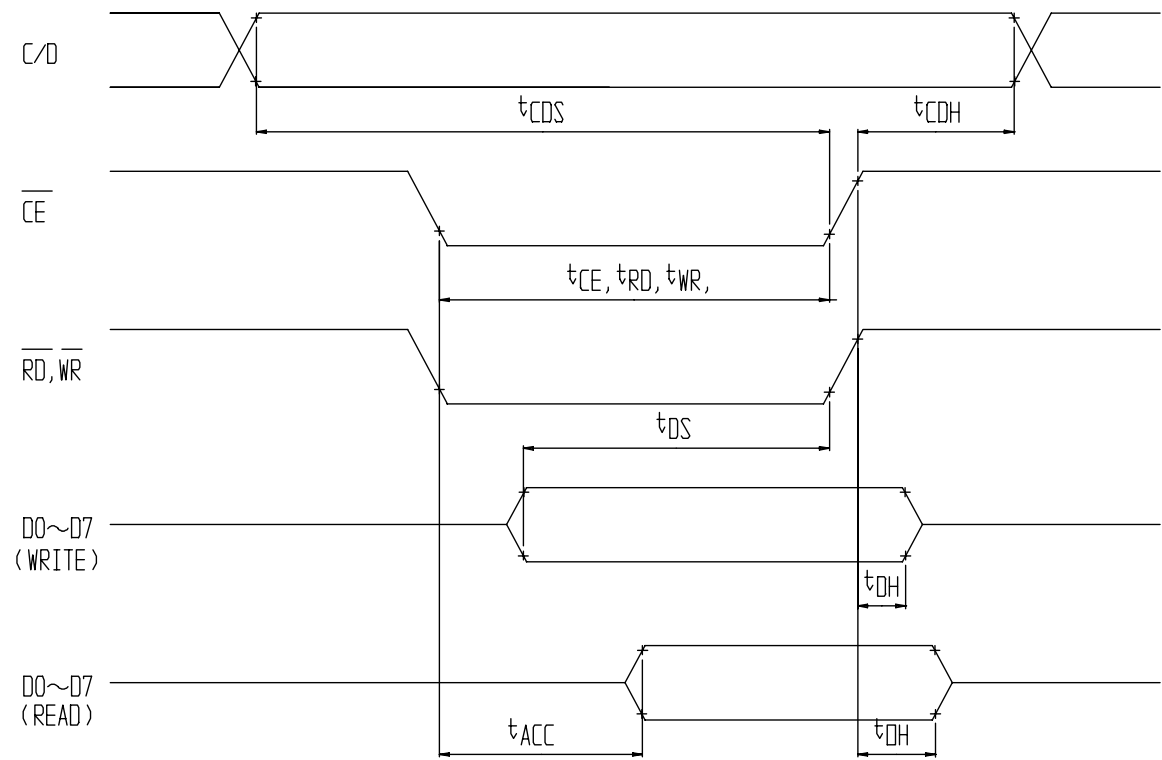


7. 时序图

7.1 AC CHARACTERISTICS

Characteristic	Symbol	Min	Max	Unit
C/D set up time	t _{CDS}	100	---	ns
C/D hold time	t _{CDH}	10	---	ns
CE, RD, WR pulse width	t _{CE} , t _{RD} , t _{WR}	80	---	ns
Data set up time	t _{DS}	80	---	ns
Data hold time	t _{DH}	40	---	ns
Access time	t _{ACC}	---	150	ns
Output hold time	t _{OH}	10	50	ns

7.2 TIMING CHARACTERISTICS



8. 功能说明及指令集

8.1. 读状态

写入任何读写命令之前，必须检查“状态”寄存器。

T6963C “状态”寄存器读取逻辑

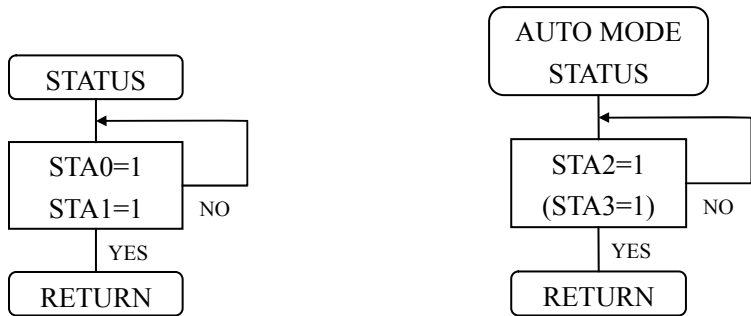
RD	L
WR	H
CE	L
C/D	H
D0-D7	Status word

T6963C “ 状态 ” 寄存器格式

MSB				LSB			
STA7 D7	STA6 D6	STA5 D5	STA4 D4	STA3 D3	STA2 D2	STA1 D1	STA0 D0
STA0	可执行命令					0 : Disable 1 : Enable	
STA1	可读写数据					0 : Disable 1 : Enable	
STA2	可自动读数据					0 : Disable 1 : Enable	
STA3	可自动写数据					0 : Disable 1 : Enable	
STA4	无用						
STA5	操作可执行					0 : Disable 1 : Enable	
STA6	错误标志(执行屏获取和拷贝命令)					0 : 无错误 1 : 错误	
STA7	检查是否正常显示					0 : 显示关 1 : 正常显示	

- 注意 1：必须同时检查 STA0 和 STA1, 在执行命令时发送数据会产生错误。
注意 2：状态检查时，检查 STA0 和 STA1 即可（不必考虑其他标志位）
注意 3：数据自动读写方式下，STA2/STA3 有效，STA0/STA1 无效。

“ 状态 ” 寄存器检查流程



注意 4：不能在 MSB0 有效情况下省去 “ 状态 ” 寄存器检查，采用延时方法也不能省去 “ 状态 ” 寄存器检查。

8.2. 数据设置

命令写入顺序



注意：如果写入超过 2 data ，最后 2 data 有效。

8.3. 显示控制命令

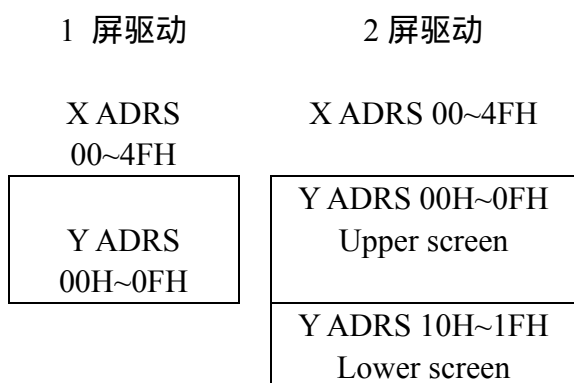
Command 命令	Code 代码	D1 参数 1	D2 参数 2	Function 功能
寄存器设置	00100001	X 地址	Y 地址	光标位置设定
	00100010	数据	00H	寄存器偏移设定
	00100100	低位地址	高位地址	地址指针设定
控制器设置	01000000	低位地址	高位地址	文本起始地址设定
	01000001	行地址	00H	文本区域设定
	01000010	低位地址	高位地址	图形起始地址设定
	01000011	行地址	00H	图形区域设定
显示方式设置	1000x000	---	---	"或" 方式
	1000x001	---	---	"异或"方式
	1000x011	---	---	"与"方式
	1000x100	---	---	"文本属性"方式
	10000xxx	---	---	内部 CG ROM 方式
	10001xxx	---	---	外部 CG RAM 方式

显示模式设置	10010000	---	---	显示关
	1001xx10	---	---	光标开, 闪烁关
	1001xx11	---	---	光标开, 闪烁开
	100101xx	---	---	文本模式开, 图形模式关
	100110xx	---	---	文本模式关, 图形模式开
	100111xx	---	---	文本模式开, 图形模式开
光标模式选择	10100000	---	---	1 行光标
	10100001	---	---	2 行光标
	10100010	---	---	3 行光标
	10100011	---	---	4 行光标
	10100100	---	---	5 行光标
	10100101	---	---	6 行光标
	10100110	---	---	7 行光标
	10100111	---	---	8 行光标
数据自动 读/写	10110000	---	---	设置为数据自动写模式
	10110001	---	---	设置为数据自动读模式
	10110010	---	---	复位自动读写模式
数据读/写	11000000	Data	---	数据写&地址指针加 1
	11000001	---	---	数据读&地址指针加 1
	11000010	Data	---	数据写&地址指针减 1
	11000011	---	---	数据读&地址指针减 1
	11000100	Data	---	数据写&地址指针不变
	11000101	---	---	数据读&地址指针不变
屏获取	11100000	---	---	获取一字节显示数据
屏拷贝	11101000	---	---	拷贝一行显示数据到图形区域
位设置/清除	11110xxx	---	---	bit 清 0
	11111xxx	---	---	bit 置 1
	1111x000	---	---	bit 0 (LSB)
	1111x001	---	---	bit 1
	1111x010	---	---	bit 2
	1111x011	---	---	bit 3
	1111x100	---	---	bit 4
	1111x101	---	---	bit 5
	1111x110	---	---	bit 6
	1111x111	---	---	bit 7 (MSB)

8.4. 部分命令详释

8.4-1. 光标位置设定

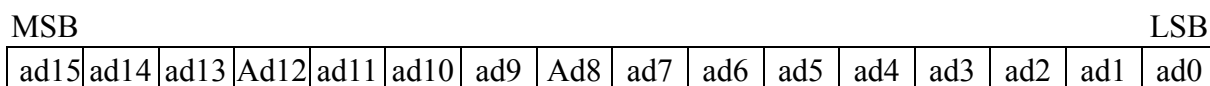
光标位置由 X ADRS, Y ADRS 指定, 也仅有此命令可以移动光标位置, 没有增加或减少功能。



8.4-2. 偏移寄存器设置

偏移寄存器用来决定外部字符发生 RAM 区域。

T6963C 有 16 位地址线



高 5 位由偏移寄存器决定。中间 8 位(ad10~ad3)由字符决定。低 3 位由垂直计数器决定。
此命令由数据 D1 (data)的低 5 位 (lower 5 bit) 有效。

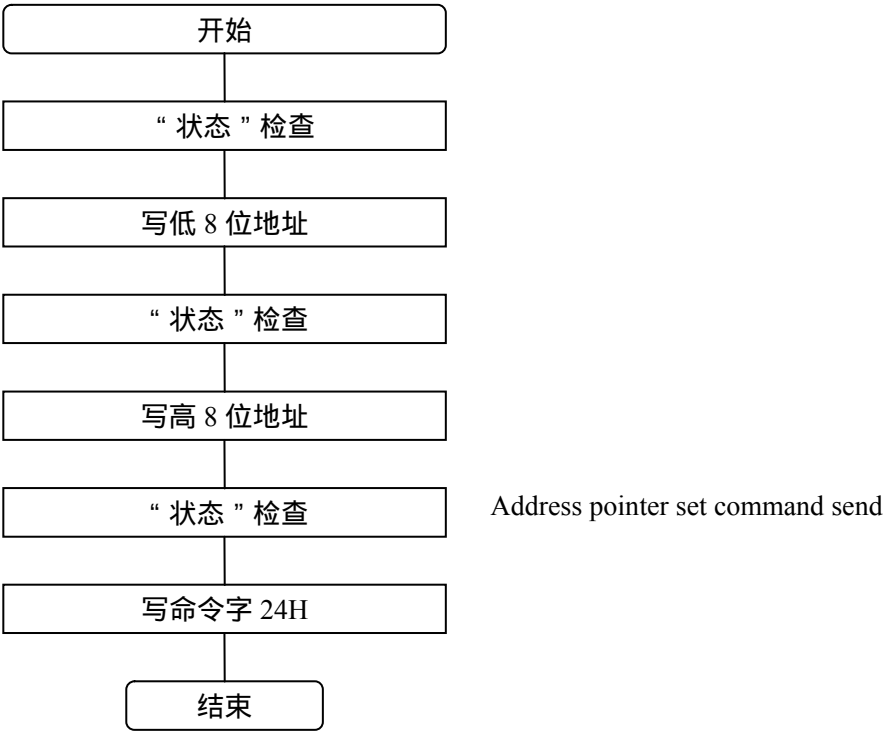
偏移量寄存器与显示 RAM 地址关系

偏移寄存器	CG RAM HEX 地址(起始-结束)
00000	0000-07FFH
00001	0800-0FFFH
00010	1000-17FFH
11100	E000-E7FFH
11101	E800-EFFFH
11110	F000-F7FFH
11111	F800-FFFFH

8.4-3. 地址指针

地址指针设置命令用来设定读写外部 RAM 的起始地址。

命令流程

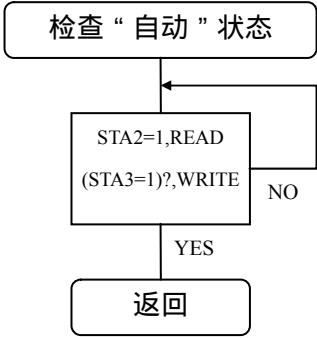
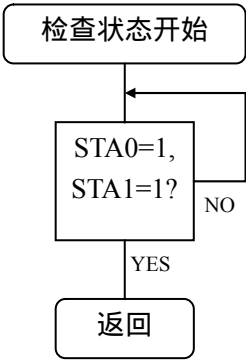
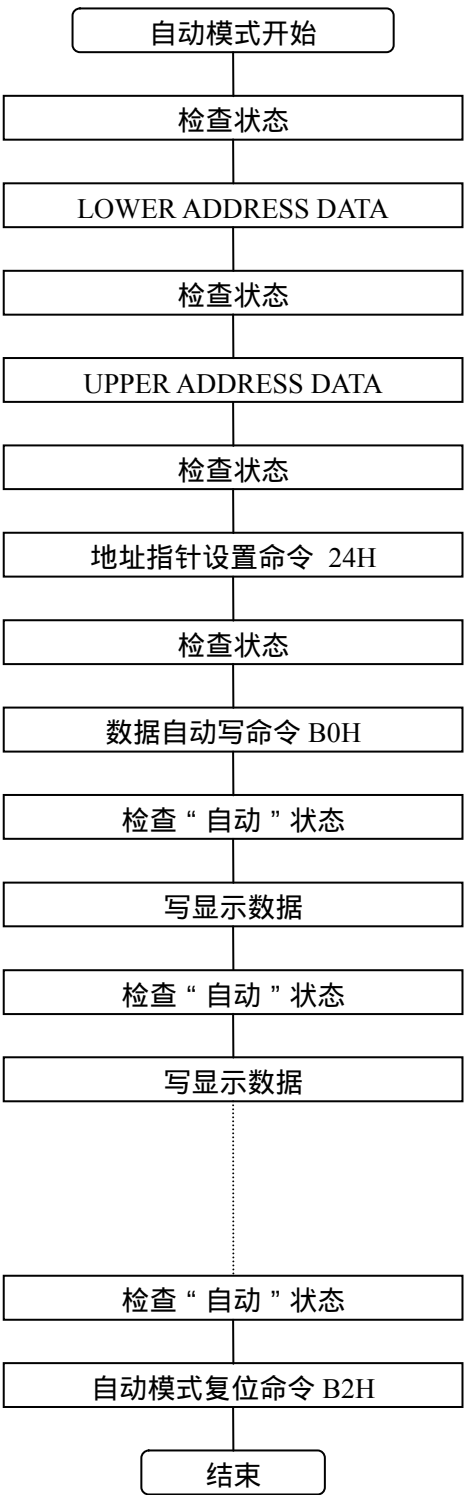


8.4-4. 数据自动读/写

Code	Hex	Function	Operand
10110000	B0H	数据自动写	---
10110001	B1H	数据自动读	---
10110010	B2H	自动模式复位	---

这条命令对整屏数据操作十分方便，设置自动模式后，每个数据之间不再需要读/写命令。“数据自动读/写”命令后应跟“地址指针设置”命令，每个数据读/写后，地址指针自动 + 1，所有数据读/写完成后，需要执行自动模式复位命令返回到正常操作模式，注意：每个数据读/写之间应当检查状态寄存器自动模式位(STA2 或,STA3)，自动模式复位也应当在检查自动模式位(STA2 或,STA3)后执行。

数据自动读/写流程



8.4-5. 数据读写

Code	Hex	Function	Operan d
1100000 0	C0H	数据写&地址指针加 1	Data
1100000 1	C1H	数据读&地址指针加 1	---
1100001 0	C2H	数据写&地址指针减 1	Data
1100001 1	C3H	数据读&地址指针减 1	---
1100010 0	C4H	数据写&地址指针不变	Data
1100010 1	C5H	数据读&地址指针不变	---

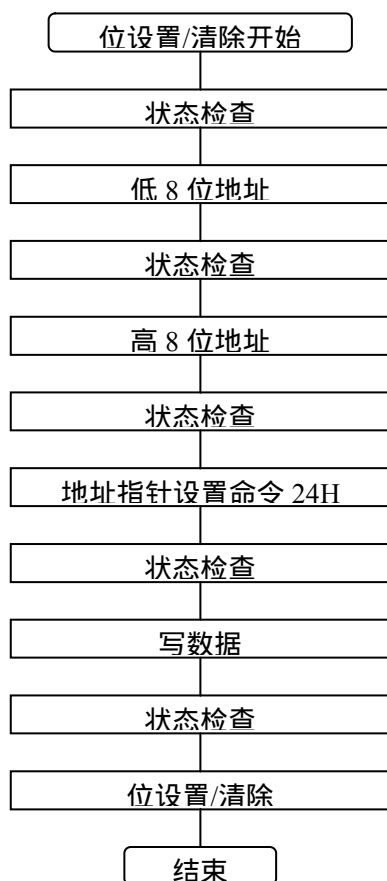
这条命令用来实现 MPU 和外部显示 RAM 之间的数据交换。数据读写操作应在地址指针设定命令后执行。地址指针可以设置为在每个数据读写后自动增 1 或减 1。
注意：这条命令对每一字节读写操作都是必须的。参见以下流程图



8.4-6. 位 设置/清除

Code	Function	Operand
11110xxx	bit 清除	---
11111xxx	bit 设置	---
1111x000	bit 0 (LSB)	---
1111x001	bit 1	---
1111x010	bit 2	---
1111x011	bit 3	---
1111x100	bit 4	---
1111x101	bit 5	---
1111x110	bit 6	---
1111x111	bit 7 (MSB)	---

这条命令用来设置/清除地址指针指定的外部显示 RAM 中的某一位。一次命令不能清除一个字节中的多位，参见下面流程图：



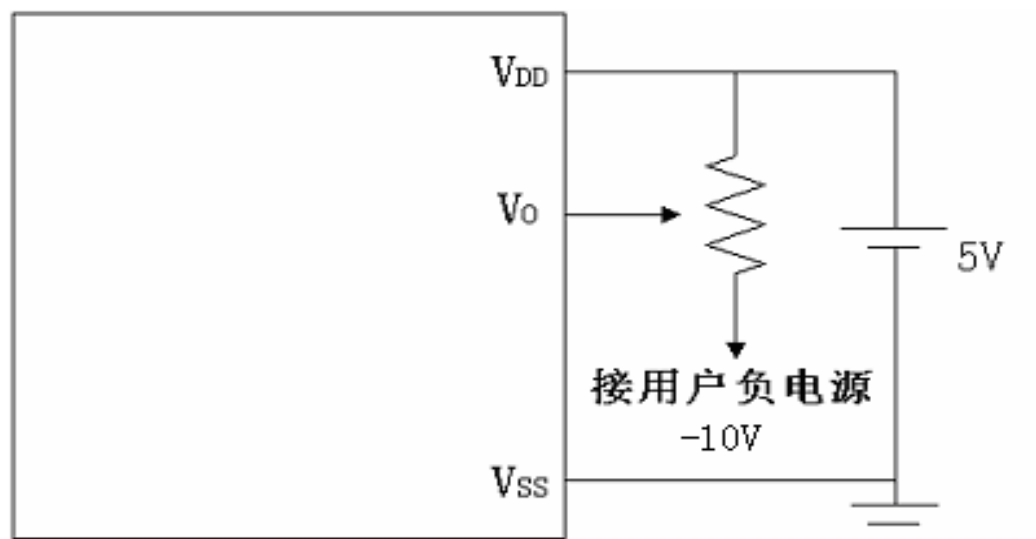
9. 字符代码表

ROM CODE 0101

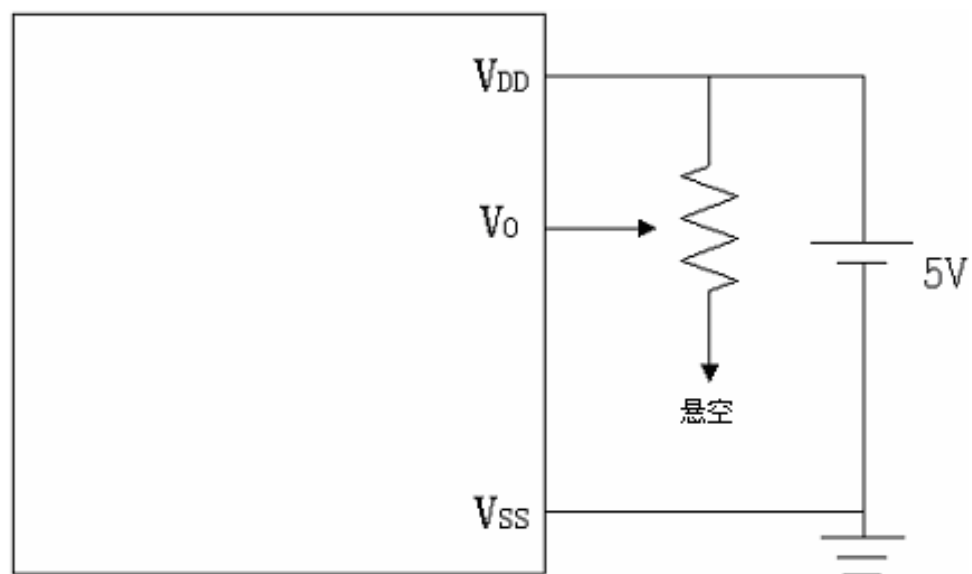
LSB MSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
2	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
4	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
6	ç	ü	ë	ä	å	ä	ä	ç	ë	ë	ë	ï	î	ï	ä	ä
7	ë	æ	æ	ö	ö	ö	ü	ü	ü	ö	ü	ø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

10. LCD 驱动电源连接方式

后缀为 ND 的模块



$V_R=10K\sim20K$



$V_R=10K\sim20K$

11. 出厂测试报告

Vdd=5V Ta=25

Test Item	Content of Test	Test Condition	Applicable Standard
High temperature storage	Endurance test applying the high storage temperature for a long time.	60 °C 200 hrs	-----
Low temperature storage	Endurance test applying the low storage temperature for a long time.	-10 °C 200 hrs	-----
High temperature operation	Endurance test applying the electric stress (Voltage & Current) and the thermal stress to the element for a long time.	50 °C 200 hrs	-----
Low temperature operation	Endurance test applying the electric stress under low temperature for a long time.	0 °C 200 hrs	-----
High temperature / Humidity storage	Endurance test applying the high temperature and high humidity storage for a long time.	60 °C , 90 %RH 96 hrs	MIL-202E-103B JIS-C5023
High temperature / Humidity operation	Endurance test applying the electric stress (Voltage & Current) and temperature / humidity stress to the element for a long time.	40 °C , 90 %RH 96 hrs	MIL-202E-103B JIS-C5023
Temperature cycle	Endurance test applying the low and high temperature cycle.	-10°C/ 60°C 10 cycles	-----
Vibration test	Endurance test applying the vibration during transportation and using.	10~22Hz → 1.5mmp-p 22~500Hz → 1.5G Total 0.5hrs	MIL-202E-201 A JIS-C5025 JIS-C7022-A-10
Shock test	Constructional and mechanical endurance test applying the shock during transportation.	50G half sign wave 11 msdc 3 times of each direction	MIL-202E-213B
Atmospheric pressure test	Endurance test applying the atmospheric pressure during transportation by air.	115 mbar 40 hrs	MIL-202E-105C
Static electricity test	Endurance test applying the electric stress to the terminal.	VS=800V , RS=1.5 kΩ CS=100 pF 1 time	MIL-883B-3015 .1

*** Supply voltage for logic system = 5V. Supply voltage for LCD system = Operating voltage at 25°C.

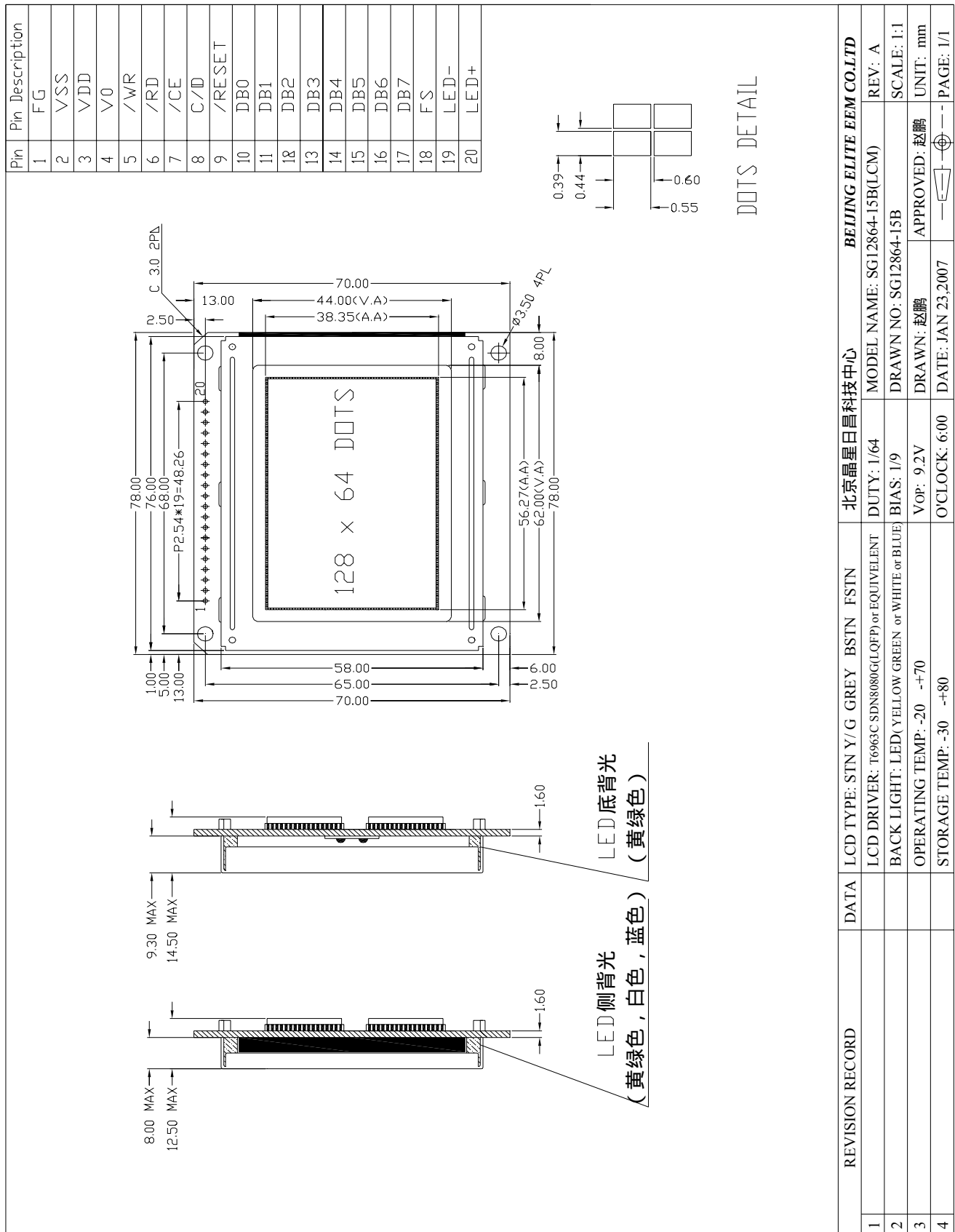
12. 接口引脚定义

Pin No.	Symbol	Level	Function
1	FG	----	液晶显示模块铁框接地引出端。
2	VSS	0V	逻辑电源地
3	VDD	5.0V	逻辑电源正
4	VO	---	LCD 驱动电压输入端，请参照第 10 项的说明连接
5	/WR	L-H	写数据信号，上升沿锁存
6	/RD	L-H	读数据信号，上升沿锁存
7	/CE	L	LCD 选通信号，低有效
8	C/D	H/L	命令/数据，高电平：命令,低电平：数据
9	/RST	L	复位信号，低有效
10	DB0	H/L	8 位数据总线
11	DB1	H/L	
12	DB2	H/L	
13	DB3	H/L	
14	DB4	H/L	
15	DB5	H/L	
16	DB6	H/L	
17	DB7	H/L	
18	FS	H/L	字体选择(H : 6 x 8 dots, L : 8 x 8 dots)
19	LED-	0V	背光电源负端
20	LED+	5.0V	背光电源正端

注：

1. SG12864-15B 系列产品标准配置是在 5V 逻辑电源环境下使用，但可以选择在 3.3V 逻辑电源环境下使用，但需要事先声明，配置将有随之调整。
2. SG12864-15B 系列产品可以选择使用 LCM 上的负电源，也可以使用外置负电源；
后缀为（ND）的型号需要客户使用外部负电源。
后缀为（TC）的型号客户只需将第 4 引脚“VO”悬空，就可以使用 LCM 中配置的负电源和温度补偿电路。
3. 第 19 引脚：LED-、第 20 引脚：LED+可以颠倒使用，短接 J3、J4，挑开 J1、J2。

13. 外形尺寸图纸



14.液晶显示模块在使用过程中的注意事项

14.1 液晶显示模块

▼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) 液晶显示模块避免受到强烈的震动.
- (2) 液晶显示模块避免扭动,拆卸金属钮角.
- (3) 液晶显示模块避免在印有线路的工作平台上操作.
- (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.
- (5) 避免接触,调整,修改导电橡胶.

▼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.

- (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) 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.

- (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.

- (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.

- (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.

- (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.

- (7) 注意厂房的湿度

厂房湿度范围: 50~60%RH

▼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.

- (1) 焊接时所需条件:

电铁的温度: 280 ± 10

焊接时间: $< 3-4S$

焊接材料: 低熔点,可充分熔化的焊锡

避免使用融化后易流动的焊锡,因为在焊接时易渗透到液晶显示模块里面,在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.

(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后再移动连接线.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.

▼ 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.

(1) 储存时尽可能使用出厂时的原包装.

(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.

(3) 储存时应保持低湿度,储存温度最佳范围: 0 ~35

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.

▼ 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.

14.2 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 ,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.

15、驱动程序

```
/******  
//6963c 通用程序  
//8X8 像素  
//RAM 空间：32K，地址：0 - 7FFF；  
//取模方式：横向正序取模  
  
//然后将 unsigned char code sch[]={} 里的数据改为新的图片数据，默认的是 12864 的  
/******  
#include <reg52.h>  
#include <intrins.h>  
/******  
  
sbit    ce=P1^0;          /*    chip enable */  
sbit    d_i=P1^3;        /*    command or data select , H : command   L:data*/  
  
sbit    r_d=P1^4;        /*    rd   EQU P1.4 */  
sbit    w_r=P1^5;        /*    wr   EQU P1.5 */  
sbit    rst=P1^6;        /*RESET EQU P1.5 */  
  
sbit    sta0_bit=P2^0;  
sbit    sta1_bit=P2^1;  
sbit    sta3_bit=P2^3;  
  
  
unsigned char *p;  
unsigned char *q;  
unsigned char *r;  
unsigned char *s;  
  
//以下的变量比较多，建议用结构体  
unsigned int size_lie=128;    //LCD 像素点的列数目,例如 240128 的屏是 240  
unsigned int size_hang=64;    //LCD 像素点的行数,例如 240128 的屏是 128  
unsigned int    textsize; //LCD text 区域空间，  
unsigned int    graphicsize; //LCD graphic 区域空间  
unsigned char textsize_h;    //textsize 的高字节  
unsigned char textsize_l;    //低字节  
unsigned char graphicsize_h; //graphic 的高字节  
unsigned char graphicsize_l; //graphic 的低字节
```


/*送*/0x42,0x21,0x21,0x07,0x00,0x00,0xEF,0x20,0x21,0x21,0x22,0x24,0x20,0x50,0x8F,0x00,
0x20,0x20,0x40,0xF8,0x80,0x80,0xFC,0x80,0x40,0x20,0x10,0x10,0x00,0x00,0xFE,0x00,
/*别*/0x00,0x3F,0x21,0x21,0x21,0x3F,0x29,0x08,0x7F,0x09,0x11,0x11,0x21,0x27,0x42,0x80,
0x04,0x04,0x24,0x24,0x24,0x24,0x24,0x24,0x24,0x24,0x04,0x04,0x04,0x1C,0x08,
/*下*/0x00,0x7F,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x00,
0x04,0xFE,0x00,0x00,0x00,0xC0,0x60,0x30,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
/*马*/0x00,0x3F,0x00,0x08,0x08,0x08,0x08,0x0F,0x00,0x00,0x7F,0x00,0x00,0x00,0x00,0x00,
0x00,0xE0,0x20,0x20,0x20,0x20,0x20,0xFC,0x04,0x04,0xF4,0x04,0x04,0x34,0x08,0x00,
/*饮*/0x20,0x20,0x20,0x3D,0x45,0x4A,0x94,0x10,0x10,0x10,0x10,0x11,0x15,0x1A,0x14,0x00,
0x80,0x80,0x80,0xFE,0x04,0x48,0x40,0x40,0x40,0xA0,0xA0,0x10,0x08,0x0E,0x04,0x00,
/*君*/0x00,0x3F,0x02,0x02,0xFF,0x04,0x04,0x3F,0x04,0x08,0x1F,0x18,0x28,0x48,0x8F,0x08,
0x00,0xF8,0x08,0x08,0xFE,0x08,0x08,0xF8,0x08,0x00,0xF8,0x08,0x08,0x08,0xF8,0x08,
/*酒*/0x40,0x2F,0x21,0x01,0x97,0x55,0x15,0x25,0x25,0x26,0xC4,0x47,0x44,0x47,0x44,0x00,
0x00,0xFE,0x20,0x20,0xFC,0x24,0x24,0x24,0x24,0x3C,0x04,0xFC,0x04,0xFC,0x04,0x00,
/*问*/0x10,0x0B,0x08,0x20,0x20,0x27,0x24,0x24,0x24,0x27,0x20,0x20,0x20,0x20,0x20,0x20,
0x00,0xFC,0x04,0x04,0x04,0xE4,0x24,0x24,0x24,0xE4,0x04,0x04,0x04,0x04,0x14,0x08,
/*何*/0x08,0x0C,0x1B,0x10,0x30,0x27,0x64,0xA4,0x24,0x27,0x24,0x20,0x20,0x20,0x20,0x00,
0x00,0x00,0xFE,0x08,0x08,0xC8,0x48,0x48,0x48,0xC8,0x48,0x08,0x08,0x28,0x10,0x00,
/*所*/0x0E,0x70,0x40,0x40,0x7C,0x44,0x44,0x44,0x7C,0x40,0x41,0x41,0x42,0x42,0x84,0x08,
0x1E,0xE0,0x80,0x80,0x80,0xFE,0x88,0x88,0x88,0x88,0x08,0x08,0x08,0x08,0x08,0x08,
/*之*/0x02,0x01,0x00,0x7F,0x00,0x00,0x00,0x00,0x00,0x01,0x06,0x18,0x28,0xE7,0x40,0x00,
0x00,0x00,0x88,0xFC,0x08,0x10,0x20,0x40,0x80,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,
/*言*/0x02,0x01,0x7F,0x00,0x00,0x1F,0x00,0x1F,0x00,0x00,0x1F,0x10,0x10,0x10,0x1F,0x10,
0x00,0x00,0xFE,0x00,0x00,0xF0,0x00,0xF0,0x00,0x00,0xF0,0x10,0x10,0x10,0xF0,0x10,
/*不*/0x00,0x7F,0x01,0x01,0x01,0x01,0x03,0x05,0x09,0x11,0x21,0x41,0x01,0x01,0x01,0x00,
0x04,0xFE,0x80,0x00,0x00,0x00,0x60,0x30,0x18,0x0C,0x08,0x00,0x00,0x00,0x00,0x00,
/*得*/0x13,0x1A,0x23,0x4A,0x8B,0x10,0x13,0x30,0x50,0x97,0x10,0x11,0x11,0x11,0x10,0x10,
0xF8,0x08,0xF8,0x08,0xF8,0x00,0xFE,0x10,0x10,0xFE,0x10,0x10,0x10,0x10,0x50,0x20,
/*意*/0x01,0x3F,0x08,0x04,0xFF,0x00,0x1F,0x12,0x11,0x1F,0x00,0x29,0x28,0x68,0x07,0x00,
0x00,0xFC,0x20,0x40,0xFE,0x00,0xF0,0x10,0x10,0xF0,0x00,0x08,0x84,0x14,0xF0,0x00,
/*归*/0x08,0x08,0x0B,0x48,0x48,0x48,0x48,0x49,0x48,0x48,0x48,0x08,0x10,0x13,0x20,0x40,
0x00,0x00,0xFC,0x04,0x04,0x04,0x04,0xFC,0x04,0x04,0x04,0x04,0x04,0xFC,0x00,0x00,
/*卧*/0x00,0x7F,0x48,0x48,0x48,0x7F,0x41,0x41,0x7F,0x48,0x48,0x48,0x48,0x7F,0x00,0x00,
0x40,0x40,0x40,0x40,0x40,0x60,0x50,0x4C,0x46,0x42,0x40,0x40,0x40,0x40,0x00,
/*南*/0x01,0x01,0xFF,0x01,0x02,0x3F,0x24,0x22,0x2F,0x21,0x3F,0x21,0x21,0x21,0x21,0x00,
0x00,0x04,0xFE,0x00,0x00,0xFC,0x24,0x44,0xF4,0x04,0xFC,0x04,0x04,0x14,0x08,0x00,
/*山*/0x01,0x01,0x01,0x01,0x01,0x21,0x21,0x21,0x21,0x21,0x21,0x21,0x7F,0x20,0x00,
0x00,0x00,0x00,0x00,0x00,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0xFC,0x04,0x00,
/*陞*/0x78,0x4B,0x50,0x57,0x61,0x51,0x49,0x4F,0x49,0x69,0x51,0x47,0x40,0x40,0x43,0x40,
0x3C,0xC0,0x40,0xFE,0x50,0x50,0x50,0xFE,0x50,0x50,0x50,0xFE,0x40,0x40,0xFC,0x00,
/*但*/0x10,0x1B,0x12,0x22,0x22,0x73,0xA2,0x22,0x22,0x23,0x22,0x22,0x20,0x2F,0x20,0x00,
0x00,0xF8,0x08,0x08,0x08,0xF8,0x08,0x08,0x08,0xF8,0x08,0x00,0x00,0xFE,0x00,0x00,
/*去*/0x01,0x01,0x01,0x3F,0x01,0x01,0x01,0x01,0x7F,0x02,0x06,0x08,0x10,0x3F,0x10,0x00,

```

0x00,0x00,0x00,0xFC,0x00,0x00,0x00,0x00,0xFE,0x00,0x00,0x20,0x10,0xF8,0x10,0x00,
/*莫*/0x04,0x04,0x7F,0x04,0x1F,0x10,0x1F,0x10,0x1F,0x01,0xFF,0x02,0x04,0x08,0x10,0x60,
0x20,0x20,0xFE,0x20,0xF0,0x10,0xF0,0x10,0xF0,0x00,0xFE,0x80,0x40,0x20,0x1C,0x08,
/*复*/0x08,0x0F,0x10,0x2F,0x48,0x0F,0x08,0x0F,0x04,0x07,0x0C,0x12,0x21,0x06,0x18,0x60,
0x00,0xFC,0x00,0xF8,0x08,0xF8,0x08,0xF8,0x00,0xF8,0x10,0x60,0x80,0x60,0x1E,0x04,
/*闻*/0x10,0x0B,0x40,0x40,0x5F,0x44,0x47,0x44,0x47,0x44,0x44,0x5F,0x40,0x40,0x40,0x40,
0x00,0xFC,0x04,0x04,0xF4,0x44,0xC4,0x44,0xC4,0x44,0xF4,0x44,0x44,0x44,0x54,0x08,
/*白*/0x02,0x02,0x04,0x3F,0x20,0x20,0x20,0x20,0x3F,0x20,0x20,0x20,0x20,0x3F,0x20,0x00,
0x00,0x00,0x00,0xF8,0x08,0x08,0x08,0x08,0xF8,0x08,0x08,0x08,0x08,0xF8,0x08,0x00,
/*云*/0x00,0x00,0x3F,0x00,0x00,0x00,0xFF,0x03,0x07,0x06,0x0C,0x18,0x31,0x7F,0x20,0x00,
0x00,0x30,0xF8,0x00,0x00,0x0C,0xFE,0x00,0x00,0x40,0x20,0x10,0xF8,0x0C,0x08,0x00,
/*无*/0x00,0x3F,0x01,0x01,0x01,0x01,0x7F,0x01,0x02,0x02,0x04,0x08,0x10,0x20,0xC0,0x00,
0x00,0xFC,0x00,0x00,0x00,0x00,0xFE,0x80,0x80,0x80,0x80,0x80,0x82,0x82,0x7E,0x00,
/*尽*/0x00,0x1F,0x10,0x10,0x1F,0x10,0x10,0x10,0x26,0x21,0x40,0x84,0x03,0x01,0x00,0x00,
0x00,0xF8,0x08,0x08,0xF8,0x80,0x80,0x40,0x20,0x98,0x8E,0x04,0x00,0x80,0x80,0x00,
/*时*/0x00,0x00,0x7C,0x44,0x47,0x44,0x7C,0x45,0x44,0x44,0x7C,0x00,0x00,0x00,0x00,0x00,
0x10,0x10,0x10,0x10,0xFE,0x10,0x10,0x10,0x90,0x90,0x10,0x10,0x10,0x10,0x50,0x20,
/*****/
//Write Data
/*****/
void wdata(unsigned char com)          //PR14
{
    ce=0;//片选
    d_i=0;//数据
    r_d=1;
    w_r=0;//写数据
    nop();
    P2=com;//送出数据
    nop();
    w_r=1;
}
/*****/
//写命令
/*****/
void wcom(unsigned char dat)          //wdata
{
    ce=0;
    d_i=1;//命令
    r_d=1;
    w_r=0;//写
    nop();
    P2=dat;
    nop();
}

```

```

        w_r=1;
    }

/*****
//查状态 0 和 1 位
*****/
void check_sta01(void)
{
    unsigned char temp;
    w_r=1;
    d_i=1;
    ce=0;
    while(1)
    {
        P2=0XFF;
        r_d=0;
        nop();
        temp=P2;
        nop();
        r_d=1;
        if((temp&0x03)==0x03)//STA0 AND STA1
            return;
    }
}

/*****
// check_sta3
*****/
void check_sta3(void)
{
    unsigned char temp;
    w_r=1;
    d_i=1;//命令
    ce=0;
    while(1)
    {
        P2=0XFF;
        r_d=0;
        temp=P2;
        nop();
        r_d=1;
        if((temp&0x08)==0x08)
            return;
    }
}

```

```

    }
}

/*****/
//双字节命令
/*****/
void double_byte_parameter(unsigned char coma,unsigned char comb,unsigned char data_b)
{
    check_sta01();
    wdata(coma);
    check_sta01();
    wdata(comb);
    check_sta01();
    wcom(data_b);
}

/*****/
//无参数命令
/*****/
void no_parameter(unsigned char com)
{
    check_sta01();
    wcom(com);
}

/*****/
//清屏或全显
//state=0,TEXT 区 , vlaue = 0 , 清屏 , value = 0xff , 全显
/*****/
void all_display(unsigned char state,unsigned char vlue)
{
    unsigned int i;
    if(state==0)
    {
        double_byte_parameter(text_startaddr_l,text_startaddr_h,0x24);
        no_parameter(0xb0);
        for(i=textsize;i>0;i--)
        {
            check_sta3();
            wdata(vlue);
        }
        check_sta3();
        wcom(0xb2);
    }
}

```

```

    }
    else
    {
        double_byte_parameter(graphic_startaddr_l,graphic_startaddr_h,0x24);
        no_parameter(0xb0);
        for(i=graphicsize;i>0;i--)
        {
            check_sta3();
            wdata(vlue);
        }
        check_sta3();
        wcom(0xb2);
    }
}

/*****/
//TEXT 模式下，写 8X8 字符
//hang:1-size_hang/8
//lie:1-size_lie/8
/*****/
void display_ascii(unsigned char *addr,unsigned char hang,unsigned char lie)
{
    unsigned int temp;//
    unsigned char low;
    unsigned char high;
    temp=(size_lie/8)*(hang-1)+(lie-1);
    //TEXT 模式下，TEXT 区域被分为 size_hang/8 行、size_lie/8 列个 8X8 点阵
    //行地址为 0 - (size_hang/8-1),列地址 0 - (size_lie/8 - 1)
    //注意，本程序的 hang 和 lie 都是从 1 开始的
    low=temp&0x00ff;
    high=(temp>>8)&0x00ff;
    double_byte_parameter(low,high,0x24);
    no_parameter(0xb0);//
    check_sta3();
    wdata(*addr);//addr：CGRAM OR CGROM 中字符的地址
    check_sta3();
    wcom(0xb2);//
}

/*****/
//显示一副图片
//graphic 模式下，共 size_hang 行，size_lie/8 列
/*****/

```

```

void display_map(unsigned char *p)
{
    unsigned char m,n;
    double_byte_parameter(graphic_startaddr_l,graphic_startaddr_h,0x24);
    no_parameter(0xb0);
    for(m=size_hang;m>0;m--)//共 size_hang 行
    {
        for(n=(size_lie/8);n>0;n--)//size_lie/8 列
        {
            check_sta3();
            wdata(*p++);
        }
    }
    check_sta3();
    wcom(0xb2);
}

```

```

void display8hotext(void)//TEXT 模式下写调用 CGROM 中字符
{
    unsigned int hang;//
    unsigned int lie;//
    for(hang=1;hang<=((size_hang*3/8)/8);hang++)//3/8 屏字符 8
    {
        for(lie=1;lie<=((size_lie/8));lie++)
            display_ascii( &h8O,hang, lie );
    }
    for(hang=(((size_hang*3/8)/8)+1);hang<=((size_hang*3/8)/8)+2;hang++)//2/8 屏 H
    {
        for(lie=1;lie<=((size_lie/8));lie++)
            display_ascii( &h8O+1,hang, lie );
    }
    for(hang=(((size_hang*3/8)/8)+3);hang<=((size_hang*3/8)/8)+5;hang++)//3/8 屏 O
    {
        for(lie=1;lie<=((size_lie/8));lie++)
            display_ascii( &h8O+2,hang, lie );
    }
}

```

//显示汉字

//hang 1 - size_hang/16, lie 1-size_lie/16

//addr CGRAM 中字符的地址

//四个 8X8 字符组成一个 16X16 的汉字字符

```

void displayHZ_at_xyaddr(unsigned char addr,unsigned char hang,unsigned char lie)
{
    unsigned int temp;
    unsigned char low;
    unsigned char high;
    temp=(size_lie/8)*2*(hang-1)+2*(lie-1);
    //TEXT 模式下，TEXT 区域被分为 size_hang/8 行、size_lie/8 列个 8X8 点阵
    //行地址为 0 - (size_hang/8-1),列地址 0 - (size_lie/8 - 1)
    //注意，本程序的 hang 和 lie 都是从 1 开始的，并且是 16X16 的
    low=temp&0x00ff;
    high=(temp>>8)&0x00ff;
    double_byte_parameter(low,high,0x24);
    no_parameter(0xb0);
    check_sta3();
    wdata(addr);//左上
    check_sta3();
    wdata(addr+2);//右上
    check_sta3();
    wcom(0xb2);
    temp=(size_lie/8)*2*(hang-1)+(size_lie/8)+2*(lie-1);//下半部分比上半部分的地址多
(size_lie/8)
    low=temp&0x00ff;
    high=(temp>>8)&0x00ff;

    double_byte_parameter(low,high,0x24);
    no_parameter(0xb0);
    check_sta3();
    wdata(addr+1);
    check_sta3();
    wdata(addr+3);
    check_sta3();
    wcom(0xb2);
}

/*****/
void displayHZ()
{
    unsigned char j;

    for(j=1;j<=6;j++)
    {
        displayHZ_at_xyaddr(0X80+tableindex[j-1+24]*4,2,j);
    }
}

```

```

        displayHZ_at_xyaddr(0X80+tableindex[j-1+24+6]*4,3,j);
    }

}

//graphic 格式下显示 8HO 字符
void display_8HO(unsigned char *q)//显示字符，三种字符的数量按照 3:2:3 的比例显示，
{
    unsigned int count_byte,count_row,count_page,page;
    for(count_row=7;count_row>3;count_row--)//共七种字符形式
    {
        double_byte_parameter(graphic_startaddr_l,graphic_startaddr_h,0x24);// 从
        GRAPHIC 的首地址开始写
        no_parameter(0xb0);//自动写
        for(page=1;page<=((size_hang*3/8)/8);page++)// 3/8 屏
        {
            s=q;
            for(count_byte=8;count_byte>0;count_byte--)//共 8 行，组成一个 8 X 8 字符
            {
                for(count_page=(size_lie/8);count_page>0;count_page--)//写一行，共 (size_lie/8)
                列

                {
                    check_sta3();
                    wdata(*s);
                }
                s++;
            }
        }

        for(page=1;page<=((size_hang*2/8)/8);page++)// 2/8
        {
            s=q;
            s=s+8;
            for(count_byte=8;count_byte>0;count_byte--)//每页 8 行，8 X 8 点阵
            {
                for(count_page=(size_lie/8);count_page>0;count_page--)//30X8 列
                {
                    check_sta3();
                    wdata(*s);
                }
                s++;
            }
        }
    }
}

```

列

```
for(page=1;page<=((size_hang*3/8)/8);page++)// 3/8
{
    s=q;
    s=s+16;
    for(count_byte=8;count_byte>0;count_byte--)//每页 8 行 , 8 X 8 点阵
    {
        for(count_page=(size_lie/8);count_page>0;count_page--)//30X8
        {
            check_sta3();
            wdata(*s);
        }
        s++;
    }
    check_sta3();
    wcom(0xb2);//退出自动写
}

void CGRAM(void)
{
    unsigned int m=0;
    double_byte_parameter(0x07,0x00,0x22); //设定最高的 1K 为 CGRAM
    double_byte_parameter(0x00,0x3c,0x24); //
    no_parameter(0xb0);
    for(m=0;m<992;m++)
    {
        check_sta3();
        wdata(CGTAB[m]);
    }
    check_sta3();
    wcom(0xb2);
}

void delay1s(void)
{
    unsigned int i;
    unsigned int j;
    for(i=0;i<1000;i++)
    {
        for(j=0;j<100;j++)
```

```

        {j=j;}
    }
}

/*****/
void main(void)
{
    rst=0;
    nop();
    nop();
    nop();
    rst=1;
    textsize=graphicsize=size_lie/8*size_hang;//
    textsize_l=textsize;//
    textsize_h=textsize>>8;//

    graphicsize_h=graphicsize>>8;
    graphicsize_l=graphicsize;

    text_startaddr=0x0000;
    text_startaddr_h=text_startaddr>>8;//
    text_startaddr_l=text_startaddr;

    graphic_startaddr=text_startaddr+textsize;//
    graphic_startaddr_h=graphic_startaddr>>8;//
    graphic_startaddr_l=graphic_startaddr;

    double_byte_parameter(text_startaddr_l,text_startaddr_h,0x40);//TEXT
    double_byte_parameter((size_lie/8),0x00,0x41);
    double_byte_parameter(graphic_startaddr_l,graphic_startaddr_h,0x42);
    double_byte_parameter((size_lie/8),0x00,0x43);

    no_parameter(0x80);//TEXT OR GRAPHIC
    no_parameter(0xa3);//
    CGRAM();

    while(1)
    {
        all_display(1,0x00);
        no_parameter(0x98);//TEXT OFF ,GRAPHIC ON
        display_map(&sch);
        delay1s();
    }
}

```

```
display_8HO(&graphich8o);
delay1s();
no_parameter(0x94); //TEXT ON,GRAPHIC OFF
all_display(0,0x00);
displayHZ();
delay1s();
all_display(0,0x00);
display8hotext();
delay1s();
}
}
```