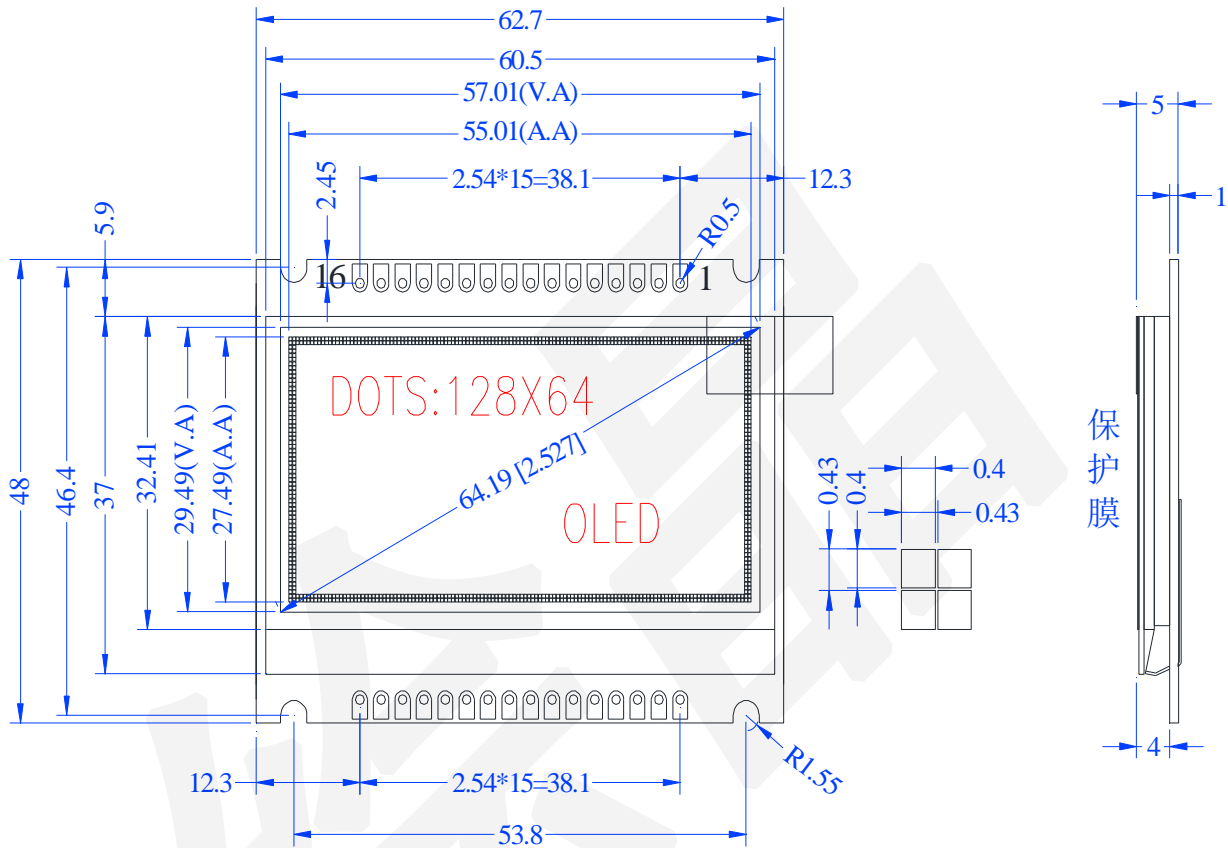


一、模块尺寸信息

a) 模块尺寸图



b) 模块尺寸图

项目	参考值
LCM 尺寸 (长×宽×厚)	62.7×48×5.0 mm
可视区域 (长×宽)	57.01×29.49mm
OLED 显示大小 (英寸)	2.5 (2.527) in
点间距 (长×宽)	0.43×0.43 mm
点尺寸 (长×宽)	0.40×0.40 mm

二、模块功能简介

- 1. 模块工作电压 3.3V/5V 可选（默认 5V）
- 2. 单片机通讯接口：
  - ┆ 并口：8 位 6800/8080
  - ┆ 串口：4 线、SPI、I<sup>2</sup>C 总线
- 3. 128\*64 个 LED 组合的 OLED 面板，128\*64 DDRAM 显示内存
- 4. 软件可调亮度精度为 1/256
- 5. 软件功能介绍
  - ┆ 显示顺序\方向设置
  - ┆ 屏保内容滚动功能（多种滚动方式详见指令表）
  - ┆ 可编程帧速度
  - ┆ 行、列重新映射（设置硬件参数）
  - ┆ 显示内容反显
  - ┆ 设置页、起始行、列地址
  - ┆ 对比度（亮度）调节
- 6. 宽工作温度-40 度到+85 度、存储温度-65 度到+150 度
- 7. 可配合中文字库 IC 使用，需要带字库的用户，请参考 12864-OLED-1 带中文字库说明书。

三、接口功能、定义、详细说明

管脚 编号	定义	方向	功能
1	R/W#	I	读/写选择信号或写信号 当 68XX 系列单片机，高电平为“读”模式；低电平为“写”模式 当 80XX 系列单片机，该脚做为写信号，低电平为写入状态，同时 CS#须为低电平 当 <b>串口</b> 或 <b>I2C</b> 模式时，该脚必须连接到 VSS。
2	E/RD#	I	读/写使能或读信号 当 68XX 系列单片机，该脚做为读/写全能信号，高电平有效，还必须 CS#为低电平。 当 80XX 系列单片机，该脚下为低电平时读取信号有效，与些同时 CS#须为低电平。 当 <b>串口</b> 或 <b>I2C</b> 模式时，该脚必须连接到 VSS。
3	VSS	P	OLED 系列地线，外部供电地脚

4	VDD	P	模块供电正极引脚, 3.3V 或者 5.0V, 默认为 5V
5	VCC	P	OLED 面板电源正极 (7-16V), 默认模块内部提供电压/悬空
6	CS#	I	片选信号 (低电平有效) 信号 当 CS=0 时, 芯片被使能, 单片机才能和芯片通讯
7	D/C#	I	数据/命令控制信号 并口模式下: 高电平时认为是显示数据输入, 低电平是认为是指令数据输入 <b>串口模式下</b> : SDIN 认为是显示数据输入, 低电平是认为是指令数据输入 <b>I2C 模式下</b> : 做为 SA0 for slave 地址选择 详细与单片机的接口信号, 请参考时序特征图
8	RES#	I	控制器和驱动电源复位信号 当该引脚为低电平时, 芯片被初始化, 保持高电平芯片正常工作
9-16	D0-D7	I/O	8 位双向数据总线与单片机数据通讯 ( <b>并口通讯</b> ), 当选择了 <b>串口通讯</b> 模式时, D1 为数据输入 SDIN, D0 为时钟输入 SCLK 当选择了 <b>I2C 通讯</b> 模式时, D2, D1 应该邦在一起作为串口数据 SDA 输出, SDA 输入的作用, D0 作为时钟 SCL 的输入。 未使用的引脚必须接到 VSS, 除在串口模式下的 D2
13	ROM-CS#	I	字库 IC 的片选输入脚 (只有串口模式下才能选用)
14	ROM-SCLK#	I	字库 IC 的时钟输入脚 (只有串口模式下才能选用)
15	ROM-SI#	I	字库 IC 的指令输入脚 (只有串口模式下才能选用)
16	ROM-SO#	O	字库 IC 的数据输出脚 (只有串口模式下才能选用)

#### 四、模块供电参数 (DC)

名称	符号	测试条件	参数范围			单位
			最小	标准	最大	
模块工作电压	VDD	—	4.8/3.1	5.0/3.3	5.2/3.5	V
OLED 驱动电压	VOLED	—	7	12	16	V
IO 输入高电平	VIH	—	0.7VDD	—	VDD	V
IO 输入低电平	VIL	—	0	—	0.6	V
LCM 输出高电平	VOH	—	0.7VDD	—	—	V
LCM 输出低电平	VOL	—	—	—	0.4	V
模块工作电流	IDD	=VDD	—	4.0	40	MA
模块待机电流	ID0	=VDD	—	—	10	uA

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

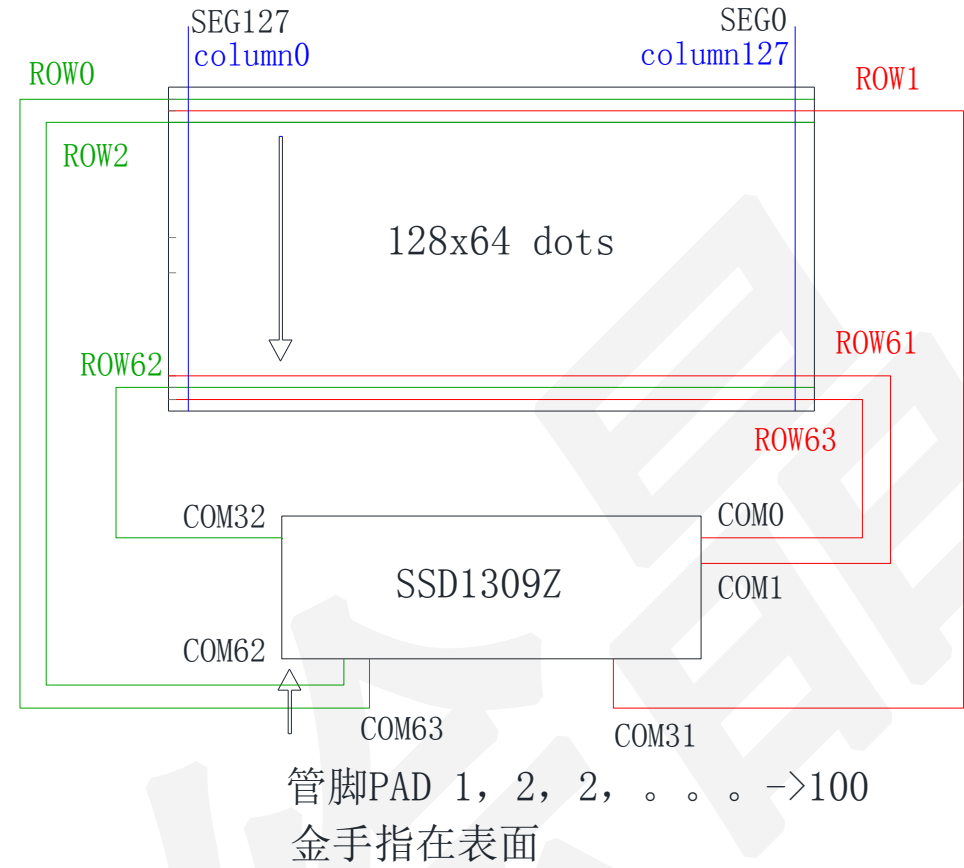
TEL: 0755-23146001 FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

五、显示结构、原理

- a) 模块硬件连接图
- i. COM0-COM63、SEG0-SEG127 硬件连接关系
  - ii. ROW0-ROW63、column0-127 可由软件设置的显示方向



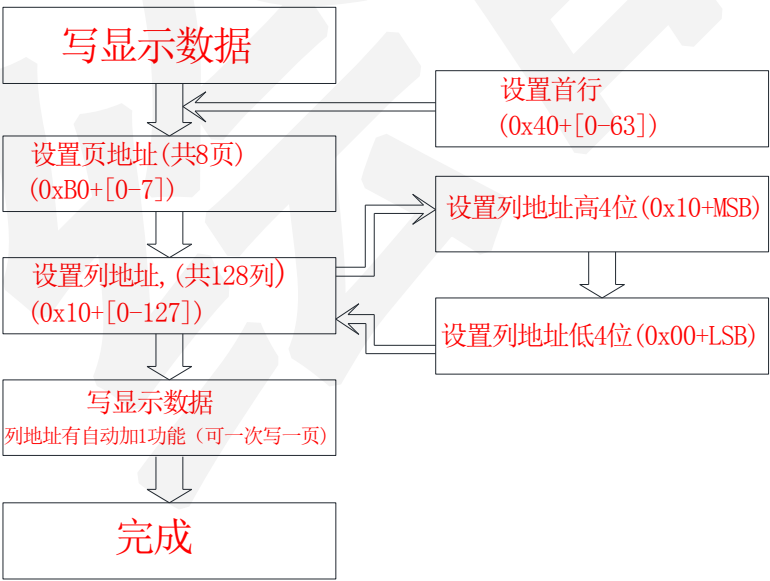
- b) 模块显示内存结构
- i. 显示内存按页地址（0-7 页）、列地址（0-127）的寻址方式
  - ii. 假设地址 0 页、0 列，那么数据宽度为一个字节（ROW0-ROW7）

正常 行	显示面板	行 映射
ROW0-ROW7	0 页	ROW63-ROW56
ROW8-ROW15	1 页	ROW55-ROW48
ROW16-ROW23	2 页	ROW47-ROW40
ROW24-ROW31	3 页	ROW39-ROW32
ROW32-ROW39	4 页	ROW31-ROW24
ROW40-ROW47	5 页	ROW23-ROW16
ROW48-ROW55	6 页	ROW15-ROW8
ROW56-ROW63	7 页	ROW7-ROW0
正常 列	列 0-----列 127	
列 映射	列 127-----列 0	

c) 显示内存页的结构

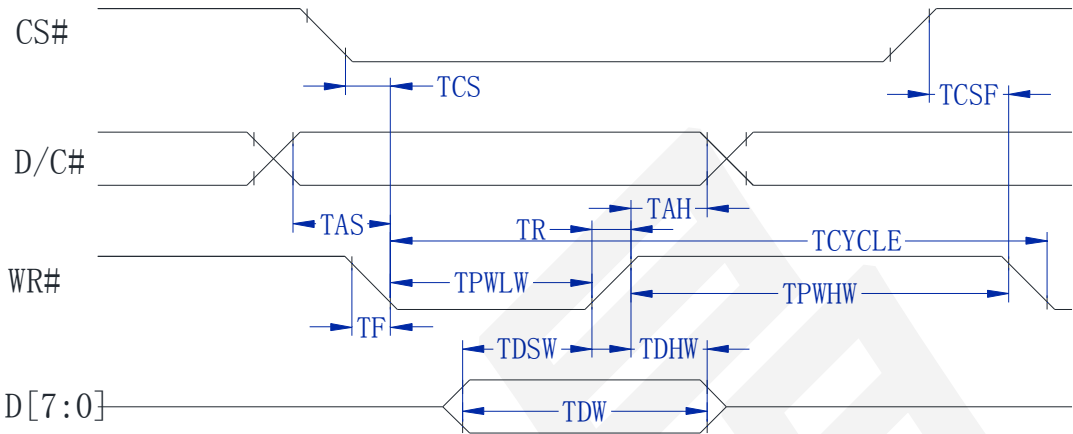
页 \ 列	列 0		、 、 、 、	列 127
假设 页 0	ROW0 (行)	D0 (低位)		
	ROW1	D1		
	ROW2	D2		
	ROW3	D3		
	ROW4	D4		
	ROW5	D5		
	ROW6	D6		
	ROW7	D7 (高位)		

d) 写显示数据程序流程

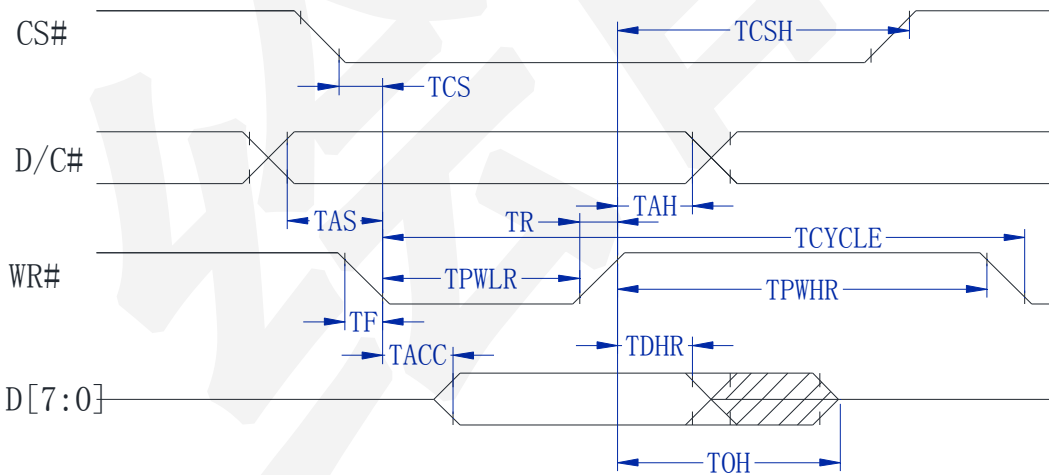


六、模块时序图

a) 8080 时序（写）



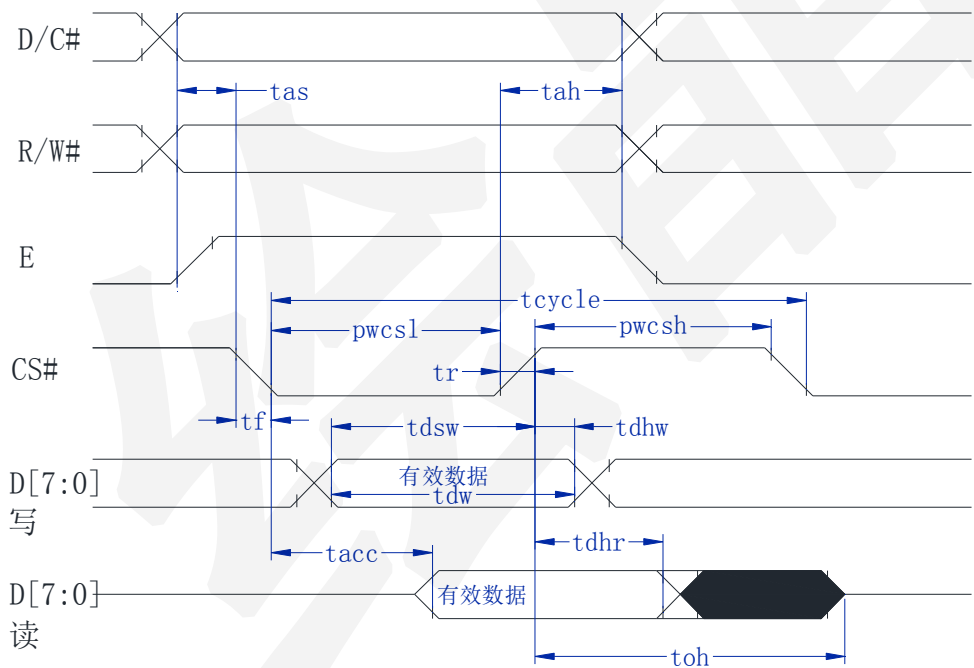
b) 8080 时序（读）



符号	参数	最小	典型	最大	单位
Tcycle	时钟周期时间	300	—	—	NS
Tas	地址建立时间	20	—	—	Ns
Tah	地址保持时间	0	—	—	Ns
Tdw	写数据时间	70	—	—	Ns
Tdsw	写数据建立时间	40	—	—	NS
Tdhw	写数据保持时间	15	—	—	Ns

Tdhr	读数据保持时间	20	-	-	Ns
Toh	输出禁能时间	-	-	70	Ns
Tacc	访问时间	-	-	140	Ns
Tpwlr	读低时间	120	-	-	Ns
Tpwlw	写低时间	60	-	-	Ns
Tpwhr	读高时间	60	-	-	Ns
Tpwhw	写高时间	60	-	-	Ns
Tr	上升时间	-	-	40	Ns
Tf	下降时间	-	-	40	Ns
Tcs	片选建立时间	0	-	-	Ns
Tcsh	片选保持读信号	0	-	-	Ns
Tcsf	片选保持保持时间	20	-	-	NS

c) 6800 时序



符号	参数	最小	典型	最大	单位
Tcycle	时钟周期时间	300	-	-	NS
Tas	地址建立时间	20	-	-	Ns
Tah	地址保持时间	0	-	-	Ns
Tdw	写数据时间	80	-	-	Ns
Tdsw	写数据建立时间	40	-	-	Ns
Tdhw	写数据保持时间	20	-	-	Ns

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

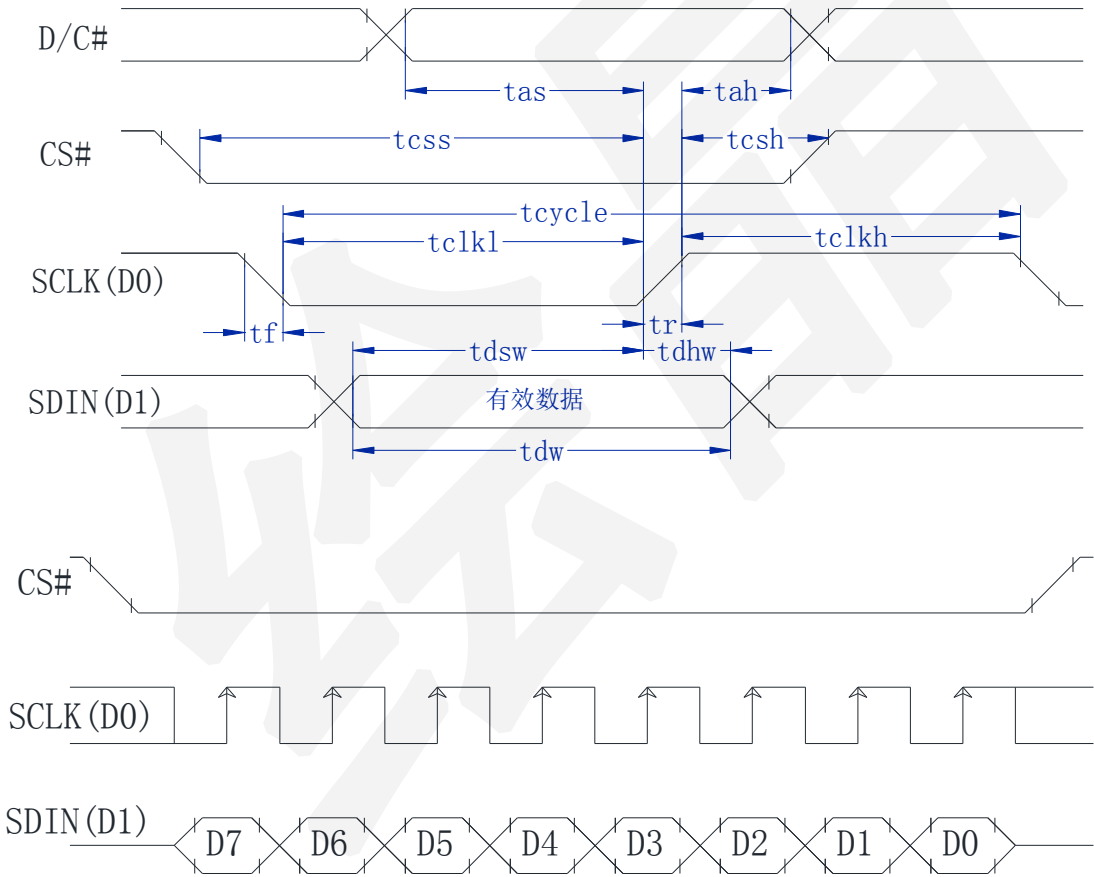
TEL: 0755-23146001      FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

Tdhr	读数据保持时间	20	-	-	Ns
Toh	输出禁止时间	-	-	70	Ns
Tacc	防问时间	-	-	140	Ns
Pwchl	片选低脉冲宽度（读）	120	-	-	Ns
	片选低脉冲宽度（写）	60	-	-	
Pwchsh	片选高脉冲宽度（读）	60	-	-	Ns
	片选高脉冲宽度（写）	60	-	-	
Tr	上升时间	-	-	40	Ns
Tf	下降时间	-	-	40	Ns

d) 4 线 SPI

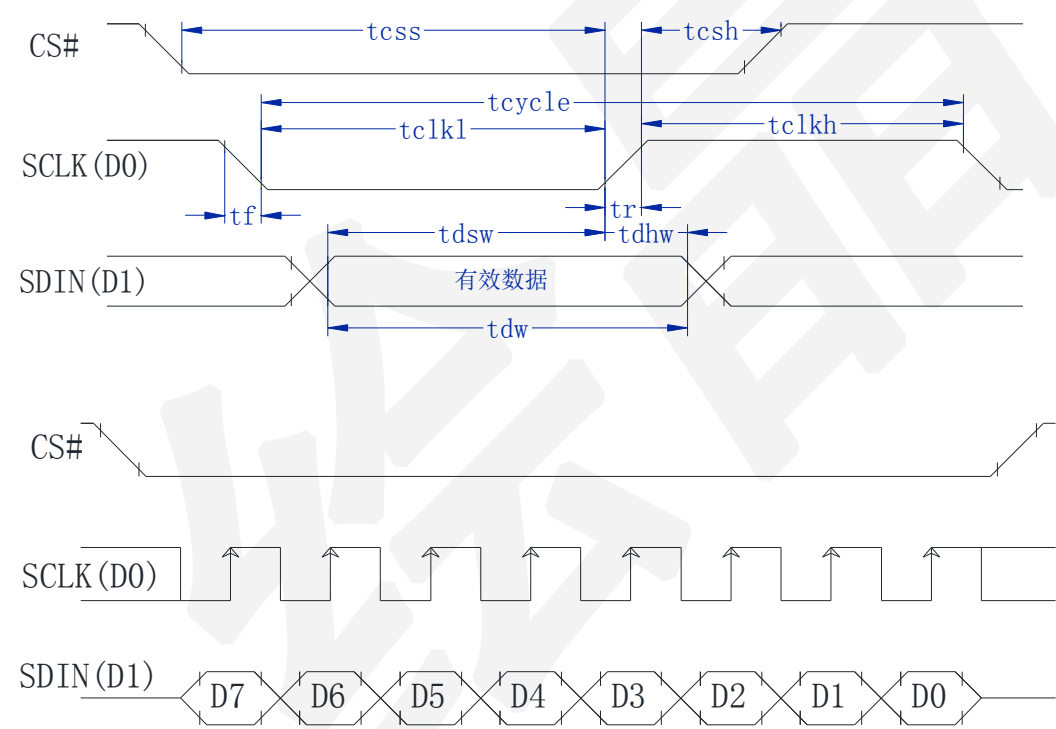


符号	参数	最小	典型	最大	单位
Tcycle	时钟周期时间	100	-	-	Ns
Tas	地址建立时间	15	-	-	Ns
Tah	地址保持时间	15	-	-	Ns
Tcss	片选建立时间	20	-	-	Ns



Tcsh	片选保持时间	50	-	-	Ns
Tdw	写数据时间	55	-	-	Ns
Tdsw	写数据建立时间	15	-	-	Ns
Tdhw	写数据保持时间	15	-	-	Ns
Tclk1	时钟低电平时间	50	-	-	Ns
Tclkh	时钟高电平时间	50	-	-	Ns
Tr	上升时间	-	-	40	Ns
Tf	下降时间	-	-	40	Ns

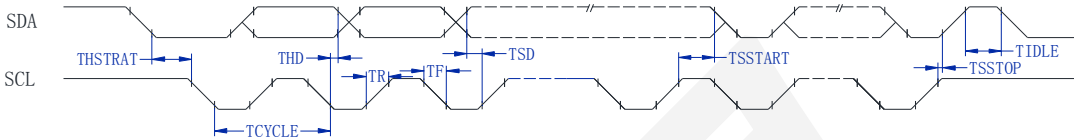
e) 3 线 SPI 通讯（本产品没预留此功能）



符号	参数	最小	典型	最大	单位
Tcycle	时钟周期时间	100	-	-	Ns
Tcss	片选建立时间	20	-	-	Ns
Tcsh	片选保持时间	50	-	-	Ns
Tdw	写数据时间	55	-	-	Ns
Tdsw	写数据建立时间	15	-	-	Ns
Tdhw	写数据保持时间	15	-	-	Ns
Tclk1	时钟低电平时间	50	-	-	Ns
Tclkh	时钟高电平时间	50	-	-	Ns

Tr	上升时间	-	-	40	Ns
Tf	下降时间	-	-	40	Ns

f) I<sup>2</sup>C 总线通讯



符号	参数	最小	典型	最大	单位
Tcycle	时钟周期时间	2.5	-	-	Us
Thstart	启动条件保持时间	0.6	-	-	Us
Thd	数据保持时间 (SDA OUT 脚)	0	-	-	Ns
	数据保持时间 (SDA IN 脚)	300	-	-	Ns
Tsd	数据建立时间	100	-	-	Ns
Tsstart	启动条件建立时间 (只有一个重新相关的启动条件)	0.6	-	-	Us
Tsstop	停止条件建立时间	0.6	-	-	Us
Tr	上升时间的数据和时钟引脚	-	-	300	Ns
Tf	下降时间的数据和时钟引脚	-	-	300	Ns
Tidle	一个新的传输可以开始的空闲时间	1.3	-	-	Us

七、模块指令表说明

(D/C#=0, R/W# (WR#)=0, E (RD#=1) 除非有具体的设置规定

1. 基本指令表 (没有找到 D8H)

1. 基本指令表				
初始	HEX	D[7: 0]	指令	描述
	81H	1000 0001	对比度控制	双字节指令
0X8F	软件演示	A[7: 0]		A[7: 0] 1-256 对比度级别, 对比度增加作为增加的值 (复位=7FH)

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001      FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

0XA4	A4/A5H	1010 010X	全显开	A4H; 恢复到 RAM 的内容显示 (复位) RAM 内容显示如下 A5H; 全部显示打开 输出忽略了 RAM 的内容
0XA6	A6/A7H	1010 011X	设置正常/反向显示	A6H; 正常显示 (复位) 0 在 ROM; 关显示面板 1 在 ROM; 开显示面板 A7H; 反向显示 0 在 ROM; 开显示面板 1 在 ROM; 关显示面板
0xae	AE/AF	1010 111X	设置显示开/关	AE; 显示关 (睡眠模式) (复位) AF; 正常模式显示开
	E3	1110 0011	保留	没有操作指令
	FDH	1111 1101	设置指令锁	双字节指令
0X12		0001 0x10 A[2]		A[2]=1 单片机的保护现状 A[2]=0; 解锁 OLED 驱动电路的 单片机接口进入指令 (复位) A[2]=1; 锁 OLED 驱动电路的单 片机接口进入指令 注: 锁定 OLED 驱动电路的单片 机接口禁止所有的命令和存储 器访问, 除 FDH 指令外

2. 滚动指令表

2. 滚动指令表（水平滚动）													
初始	HEX	D[7: 0]	指令	描述									
	26/27h	0010 011X	连续水平滚动设置	26H; 向右水平滚动 27H; 向左水平滚动									
		A[7: 0]		A[7: 0]; 虚拟字节（设置 00H） 水平滚动一列									
		B[2: 0]		B[2: 0] 定义起始页地址 <table><tr><td>000b- 页 0</td><td>011b- 页 3</td><td>110b- 页 6</td></tr><tr><td>001b- 页 1</td><td>100b- 页 4</td><td>111b- 页 7</td></tr><tr><td>010b- 页 2</td><td>101b- 页 5</td><td></td></tr></table>	000b- 页 0	011b- 页 3	110b- 页 6	001b- 页 1	100b- 页 4	111b- 页 7	010b- 页 2	101b- 页 5	
000b- 页 0	011b- 页 3	110b- 页 6											
001b- 页 1	100b- 页 4	111b- 页 7											
010b- 页 2	101b- 页 5												

		C[2:0]		C[2:0]; 设定的时间间隔每个滚动步之间的帧频率 <table><tr><td>000b-5 帧</td><td>100b-2 帧</td></tr><tr><td>001b-64 帧</td><td>101b-3 帧</td></tr><tr><td>010b-128 帧</td><td>110b-4 帧</td></tr><tr><td>011b-256 帧</td><td>111b-1 帧</td></tr></table>	000b-5 帧	100b-2 帧	001b-64 帧	101b-3 帧	010b-128 帧	110b-4 帧	011b-256 帧	111b-1 帧	
000b-5 帧	100b-2 帧												
001b-64 帧	101b-3 帧												
010b-128 帧	110b-4 帧												
011b-256 帧	111b-1 帧												
		D[2:0]		D[2:0]定义结束页地址 <table><tr><td>000b- 页 0</td><td>011b- 页 3</td><td>110b- 页 6</td></tr><tr><td>001b- 页 1</td><td>100b- 页 4</td><td>111b- 页 7</td></tr><tr><td>010b- 页 2</td><td>101b- 页 5</td><td></td></tr></table>	000b- 页 0	011b- 页 3	110b- 页 6	001b- 页 1	100b- 页 4	111b- 页 7	010b- 页 2	101b- 页 5	
000b- 页 0	011b- 页 3	110b- 页 6											
001b- 页 1	100b- 页 4	111b- 页 7											
010b- 页 2	101b- 页 5												
		E[7:0]		E[7:0]: 虚拟字节 (设置 00H)									
		F[7:0]		F[7:0]; 定义开始列 (复位=0)									
		G[7:0]		G[7:0]: 定义结束列地址 (复位=7FH)									
				注: D[2; 0) 必须大于等于 B[2: 0] G[7; 0) 必须大于等于 F[7: 0]									

2. 滚动指令表（垂直水平滚动）													
初始	HEX	D[7：0]	指令	描述									
	29/2Ah	0010 10XX	连续垂直滚动设置	29H；垂直和水平向右滚动 2AH；垂直和水平向左滚动									
		A[0]		A[0]；设置列滚动偏移 =0 没有水平滚动 =1 水平滚动一列									
		B[2：0]		B[2：0]定义起始页地址 <table><tr><td>000b-页 0</td><td>011b-页 3</td><td>110b-页 6</td></tr><tr><td>001b-页 1</td><td>100b-页 4</td><td>111b-页 7</td></tr><tr><td>010b-页 2</td><td>101b-页 5</td><td></td></tr></table>	000b-页 0	011b-页 3	110b-页 6	001b-页 1	100b-页 4	111b-页 7	010b-页 2	101b-页 5	
000b-页 0	011b-页 3	110b-页 6											
001b-页 1	100b-页 4	111b-页 7											
010b-页 2	101b-页 5												
		C[2:0]		C[2:0]；设定的时间间隔每个滚动步之间的帧频率 <table><tr><td>000b-5 帧</td><td>100b-2 帧</td></tr><tr><td>001b-64 帧</td><td>101b-3 帧</td></tr><tr><td>010b-128 帧</td><td>110b-4 帧</td></tr><tr><td>011b-256 帧</td><td>111b-1 帧</td></tr></table>	000b-5 帧	100b-2 帧	001b-64 帧	101b-3 帧	010b-128 帧	110b-4 帧	011b-256 帧	111b-1 帧	
000b-5 帧	100b-2 帧												
001b-64 帧	101b-3 帧												
010b-128 帧	110b-4 帧												
011b-256 帧	111b-1 帧												

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

		D[2:0]		D[2:0] 定义结束页地址 <table><tr><td>000b- 页 0</td><td>011b- 页 3</td><td>110b- 页 6</td></tr><tr><td>001b- 页 1</td><td>100b- 页 4</td><td>111b- 页 7</td></tr><tr><td>010b- 页 2</td><td>101b- 页 5</td><td></td></tr></table>	000b- 页 0	011b- 页 3	110b- 页 6	001b- 页 1	100b- 页 4	111b- 页 7	010b- 页 2	101b- 页 5	
000b- 页 0	011b- 页 3	110b- 页 6											
001b- 页 1	100b- 页 4	111b- 页 7											
010b- 页 2	101b- 页 5												
		E[5:0]		E[5:0] 垂直滚动偏移，例如 =01h; 参考偏移=1 行（排） =3FH; 参考偏移=63 行（排）									
		F[7:0]		F[7:0]；定义开始列（复位=0									
		G[7:0]		G[7:0]：定义结束列地址（复 位=7FH）									
				注： D[2； 0）必须大于等于 B[2： 0] G[7； 0）必须大于等于 F[7： 0]									
	2E	0010 1110	停止滚动	停止滚动 通过指令 26H/27H/29H/2AH 配置 注：发送 2Eh 指令停止滚动动 作，RAM 中的数据需要被重写									

2. 滚动指令表（激活滚动）				
初始	HEX	D[7: 0]	指令	描述
	2Fh	0010 1111	激活滚动	开始滚动，由滚动配置设置命 令；26h/27h/29h/2ah 与以下 有效的序列 有效的命令序列 1；26h; 2fh 有效的命令序列 2；27h; 2fh 有效的命令序列 3；29h; 2fh 有效的命令序列 4；2ah; 2fh 例如 如果 26H; 2AH; 2FH 指令 发出，最后的命令为滚动的设 置，这种情况下，后面的指令 2AH 将被执行

				换句话说，在过去的滚动设置指令将覆盖设置之前滚动设置指令
	A3h	1010 0011	设置垂直滚动区域	
		A[5:0]		A[5:0]; 设置顶点区域的行号，顶区的行号在 GDDRAM 的顶部（即 0 行）；[复位=0]
		B[6:0]		B[6:0]; 设置滚动区域的行，这是可以垂直滚动的行数，滚动从顶部第一行到下方的固定行[复位=64] 注意 1、A[5:0]+B[6:0]<=复用率 2、B[6:0]<=复用率（MUX ratio） 3 垂直滚动偏移 (E[5:0]in29h/2ah)<B[6:0] 31 设置显示起始行 (X5X4X3X2X1X0 of 40h-7fh)<B[6:0] 4 在滚动区域移动到滚动区域的第一行的最后一行 5 FOR 64D MUX 显示 A[5:0]=0, B[6:0]=64 整个区域滚动 A[5:0]=0, B[6:0] <64 顶部区域滚动 A[5:0]+B[6:0] <64 中间区域滚动 A[5:0]+B[6:0] =64 底部区域滚动 6 当垂直滚动条指令启用 29H/2AH, 垂直滚动区域由该命令定义

2. 滚动指令表（内容滚动设置）				
初始	HEX	D[7: 0]	指令	描述
	2C/2D	0010 110X	内容滚动设置	2CH; 水平向右滚动一列 2DH; 水平向左滚动一列
		A[7: 0]		A[7:0]虚拟字节（设置 00H） 水平滚动一列

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

		B[2: 0]		B[2: 0]定义起始页地址 <table><tr><td>000b-页 0</td><td>011b-页 3</td><td>110b-页 6</td></tr><tr><td>001b-页 1</td><td>100b-页 4</td><td>111b-页 7</td></tr><tr><td>010b-页 2</td><td>101b-页 5</td><td></td></tr></table>	000b-页 0	011b-页 3	110b-页 6	001b-页 1	100b-页 4	111b-页 7	010b-页 2	101b-页 5	
000b-页 0	011b-页 3	110b-页 6											
001b-页 1	100b-页 4	111b-页 7											
010b-页 2	101b-页 5												
		C[7:0]= 0000 0001		A[7:0]虚拟字节（设置 00H）									
		D[2:0]		D[2:0]定义结束页地址 <table><tr><td>000b-页 0</td><td>011b-页 3</td><td>110b-页 6</td></tr><tr><td>001b-页 1</td><td>100b-页 4</td><td>111b-页 7</td></tr><tr><td>010b-页 2</td><td>101b-页 5</td><td></td></tr></table>	000b-页 0	011b-页 3	110b-页 6	001b-页 1	100b-页 4	111b-页 7	010b-页 2	101b-页 5	
000b-页 0	011b-页 3	110b-页 6											
001b-页 1	100b-页 4	111b-页 7											
010b-页 2	101b-页 5												
		E[5:0]= 0000 0000		A[7:0]虚拟字节（设置 00H）									
		F[7:0]		F[7:0]；定义开始列（复位=0									
		G[7:0]		G[7:0]：定义结束列地址（复位=7FH）									
				注： D[2; 0) 必须大于等于 B[2: 0] G[7; 0) 必须大于等于 F[7: 0] 延时时间必须 2/帧频率，如果连续发送 2CH/2DH 指令									

3. 地址设置指令表

3. 地址设置指令表				
初始	HEX	D[7: 0]	指令	描述
1	00-0FH	0000 XXXX	设置页面寻址方式低列的起始地址	页面寻址方式使用 X[3: 0]数据位设置列的起始地址指令低字节，初始显示线寄存器 0000H 复位后 注：此指令仅用于页寻址方式
	10-1FH	0001 XXXX	设置页面寻址方式高列的起始地址	页面寻址方式使用 X[3: 0]数据位设置列的起始地址指令高字节，初始显示线寄存器 0000H 复位后

				注：此指令仅用于页寻址方式
	20H	0010 0000	设置内存寻址方式	
0X02		A[1: 0]		=00; 水平寻址方式 =01; 垂直寻址方式 =10; 页面寻址方式（复位） =11; 无效的
	21H	0010 0001	设置列地址	设置列的开始和结束地址
		A[7: 0]		列的起始地址，范围 0-127（复位=0）
		B[7: 0]		列的结束地址，范围 0-127（复位=127）
				注：该指令仅用在水平/垂直寻址模式
	22H	00100010	设置页面地址	设置页面起始/结束地址
		A[2: 0]		页起始地址，范围 0-7（复位=0）
		B[2: 0]		页面结束地址，范围 0-7（复位=7）
				注：该指令仅用于水平/垂直寻址模式
	B0-B7H	1011 0XXX	页面寻址方式设置页的起始地址	在页地址寻址模式设置 GDDRAM 页起始地址（页 0-页 7）使用 X[2: 0] 注：该指令仅用于页寻址模式

## 4. 硬件配置指令表

4. 硬件配置（面板分辨率与布局相关）指令表				
初始	HEX	D[7: 0]	指令	描述
0X00	40-7FH	01XX XXXX	设置显示起始行	设置显示 RAM 的首行指令，使用 X[5: 0] 显示首行寄存器复位为 000000 在复位
0XA1	A0/A1H	1010 000X	设置 SEG 接线映射	A0H; 列地址 0 连接到 SEGO 上（复位） A1H; 列地址 127 连接到 SEGO 上
	A8H	1010 1000	设置复用率/点空比	设置 MUX 比率到 N+1 MUX
0X3F		A[5: 0]		N=A[5: 0]; 的 16MUX 到 64MUX

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)



				复位=111111（即 63，64MUX） A[5: 0]的 0-14 无效
0XC8	C0/C8H	1100 X000	设置 COM 输出扫描方向	C0H; 正常模式（复位）扫描为从 COM0 到 COM[N-1] C8H; 重新定义模式，扫描从 COM[N-1] 到 COM0 其中 N 是复用率
	D3	1101 0011	设置显示偏移	由 COM 从 0-63 垂直移动 该值复位为 00H
0X00		A[5: 0]		
	DA	1101 1010	设置 COM 引脚的硬件配置	
0X12		A[5]; A[4] 0054 0010		A[4]=0; 连续的 COM 引脚配置 A[4]=1; （复位）选择 COM 引脚配置 A[5]=0; （复位）禁止 COM 左/右映射 remap A[5]=1; 启用 COM 左/右 remap
	DCH	1101 1100	设置 GPIO	
		A[1: 0]		00 引脚 HIZ, 无效输入 01 引脚 HIZ, 启用输入 10 引脚输出低（复位） 11 引脚输出高

## 5. 定时各驱动方案设置指令表

5. 定时各驱动方案设置指令表				
初始	HEX	D[7: 0]	指令	描述
	D5H	1101 0101	设置显示时钟分频比/振荡器频率	
0XA0		A[7: 0]		A[3: 0]定义分频比（D）的显示时钟（DCLK） 分频比=A[3: 0]+1, 复位为 000（分频比=1） A[7: 4]设置振荡频率，频率增加振荡频率的值 A[7: 4]反之亦然，复位是 0111B 范围 000B-111B 设定值增大频率增加
	D9H	1101 1001	设置预充电周期	

0X82		A[7: 0]		A[3: 0]; 1 期高达 15 个时钟周期 0 无效的输入 (复位=2h) A[7: 4];2 期多达 15 时钟周期 0 无效输入 (复位=2H)		
0X34	DBH	1101 1011 A[5:2]	设置共通高电位 取消水平	A[52]	HEX	共通高位取消水平
				0000B	00H	~0.64xVcc
				1101B	34H	~0.78xVcc (复位
				1111B	3CH	~0.84xVcc

6. 数据/读写

5. 定时各驱动方案设置指令表					
初始	HEX	D[7: 0]	指令	描述	
	D[7:0]	D[7:0]	读状态寄存器	D[7]; 保留 D[6]; ‘1’ 显示关/ ‘0’ 显示开 D[5: 0]; 保留	

注：模式比那些在指令表被禁止的进入芯片一个意想不到的指令，可能发生的结果。

数据读/写

在正常的数据读取模式的列地址指针 GRRDRAM 将增加一个后自动每个数据读

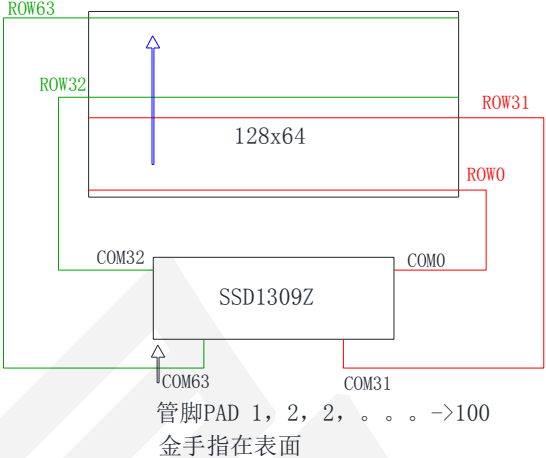
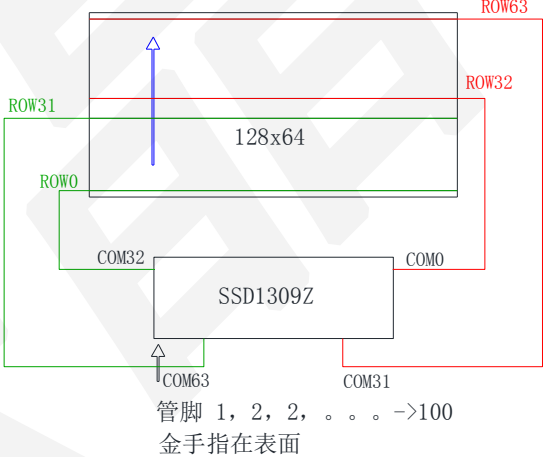
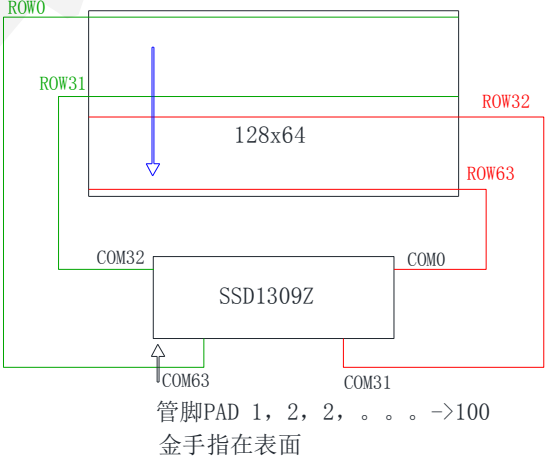
同时，一个虚拟的读是第一读取数据之前需要。

串行接口模式一直是在写模式。

GDDRAM 列地址的指针将自动控制增加一个在每个数据写入

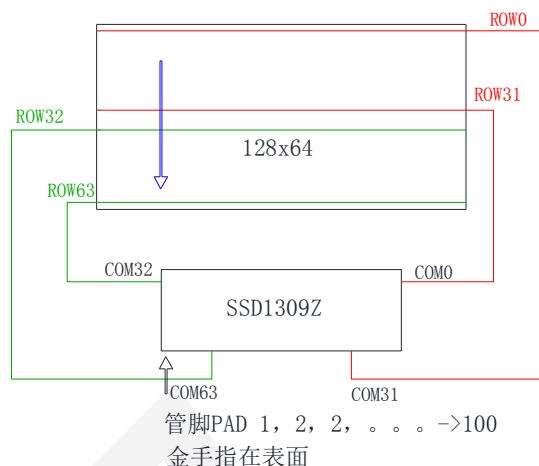
八、部分指令的详细说明

1. 设置页面寻址方式下列的的起始地址 (00-0FH)  
此命令指定下一位列的起始地址的显示数据 RAMD 页面寻址方式。列地址将自动递增每个数据访问。
2. 设置页寻址高列的起始地址 (10H-1FH)  
该指令指定的 8 位高一系列显示数据 RAM 的起始地址为下页寻址方式。列地址将递增每个数据访问。参数 9. 3 节 10. 3 部分细节
3. 设置内存寻址模式 (20H)  
有 3 种不同的存储寻址方式：面寻址，控制器水平寻址模式和垂直寻址方式。这个指令集的存储器寻址方式进入一个以上的三种模式。在这里，“COL”指图形显示数据 DRM 列。
4. 设置 COM 引脚的硬件配置 (DAh)

条件	COM 引脚配置
<p>1、连续 COM 引脚配置 (DAH A[4]=0 COM 扫描方向：从 COM0-COM63 (COH) 禁用 COM 左/右映射 (DAH A[5]=0)</p> <p>注：COM 从下到上的顺序</p>	 <p>管脚 PAD 1, 2, 2, ... -&gt;100 金手指在表面</p>
<p>2、连续 COM 引脚配置 (DAH A[4]=0 COM 扫描方向：从 COM0-COM63 (COH) 使能 COM 左/右映射 (DAH A[5]=1) 注：映射解说 (芯片左右交换 COM 管脚) COM32-COM63 (对应到 ROW0-ROW31) COM0-COM31 (对应到 ROW32-ROW63)</p> <p>注：COM 从下到上的顺序</p>	 <p>管脚 1, 2, 2, ... -&gt;100 金手指在表面</p>
<p>3、连续 COM 引脚配置 (DAH A[4]=0 COM 扫描方向：从 COM63-COM0 (C8H) 禁用 COM 左/右映射 (DAH A[5]=0)</p> <p>注：COM 从上到下的顺序</p>	 <p>管脚 PAD 1, 2, 2, ... -&gt;100 金手指在表面</p>

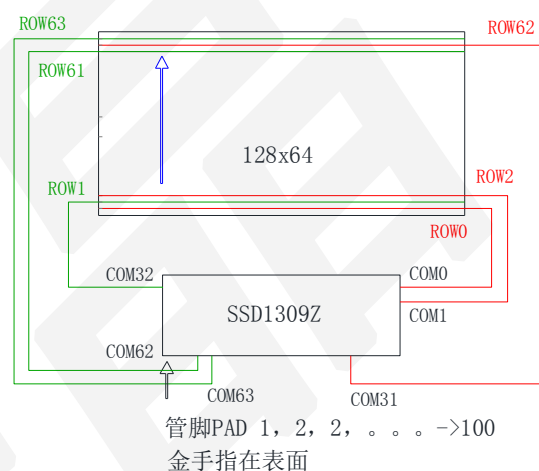
4、连续 COM 引脚配置 (DAH A[4]=0)  
COM 扫描方向：从 COM63-COM0 (C8H)  
使能 COM 左/右映射 (DAH A[5]=1)

注：COM 从上到下的顺序



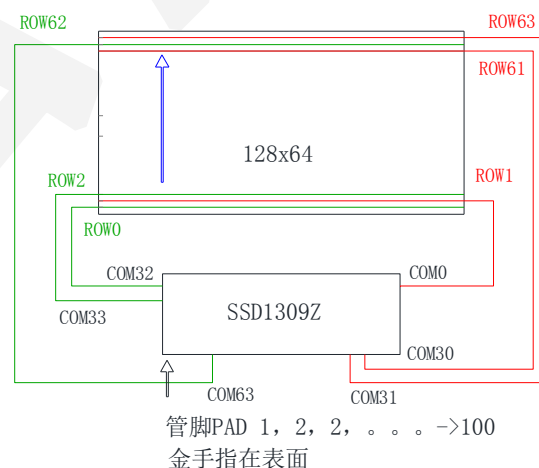
5、选择 COM 引脚配置 (DAH A[4]=1)  
COM 扫描方向：从 COM0-COM63 (COH)  
禁用 COM 左/右映射 (DAH A[5]=0)

注：COM 从下到上的顺序



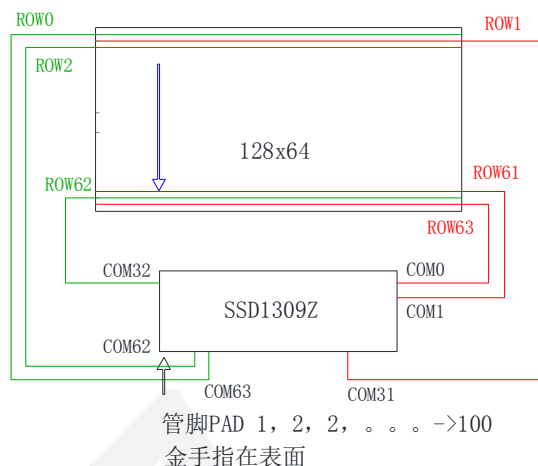
6、选择 COM 引脚配置 (DAH A[4]=1)  
COM 扫描方向：从 COM0-COM63 (COH)  
使能 COM 左/右映射 (DAH A[5]=1)

注：COM 从下到上的顺序



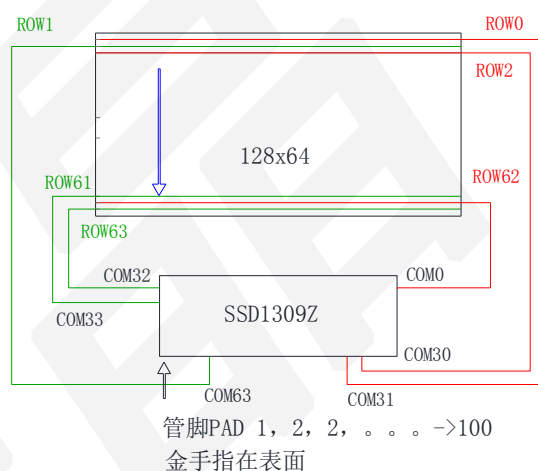
7、选择 COM 引脚配置 (DAH A[4]=1  
COM 扫描方向: 从 COM63-COM0 (C8H)  
禁用 COM 左/右映射 (DAH A[5]=0)

注: COM 从上到下的顺序



8、选择 COM 引脚配置 (DAH A[4]=1  
COM 扫描方向: 从 COM63-COM0 (C8H)  
使能 COM 左/右映射 (DAH A[5]=1)

注: COM 从上到下的顺序

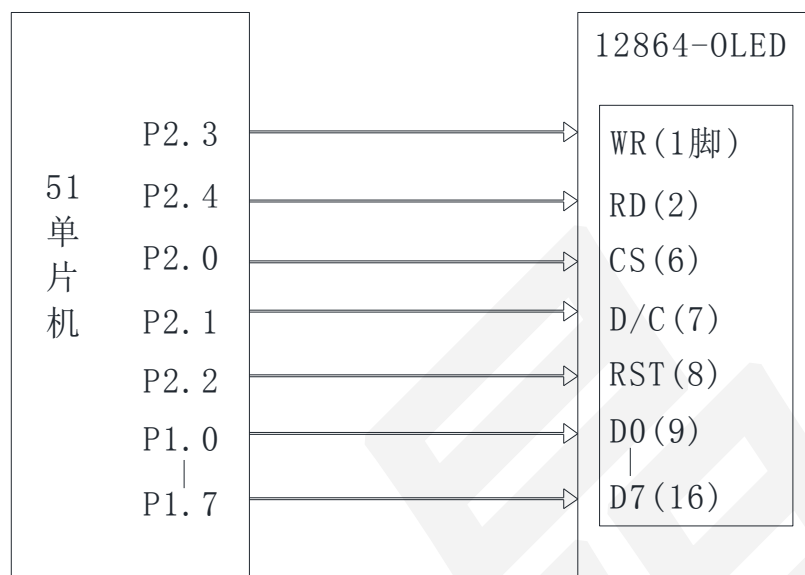


### 九、模块与单片机连接图

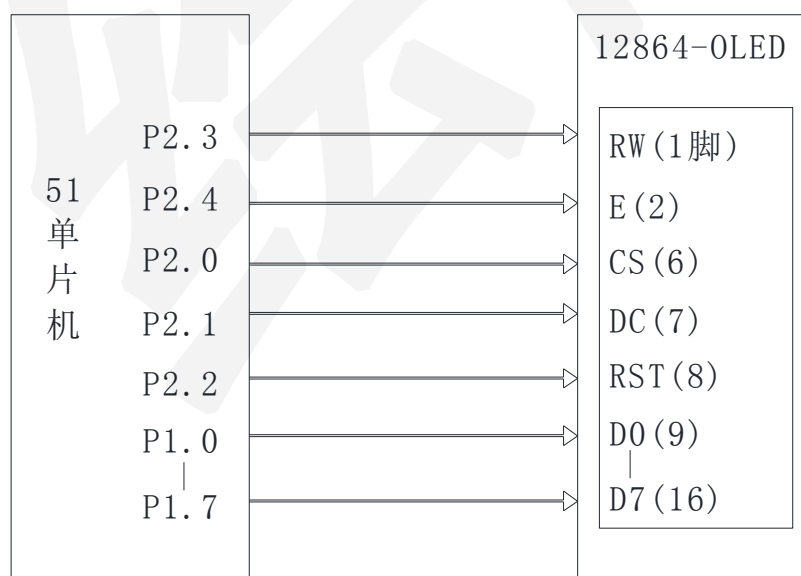
a) 默认出厂 8080 时序, 若要更改通讯方式, 需要更改跳点。如下

BS2 、BS1 (在模块上)	通讯方式
00	4 线 SPI
01	I <sup>2</sup> C 总线
11	8 位 8080 并口
10	8 位 6800 并口

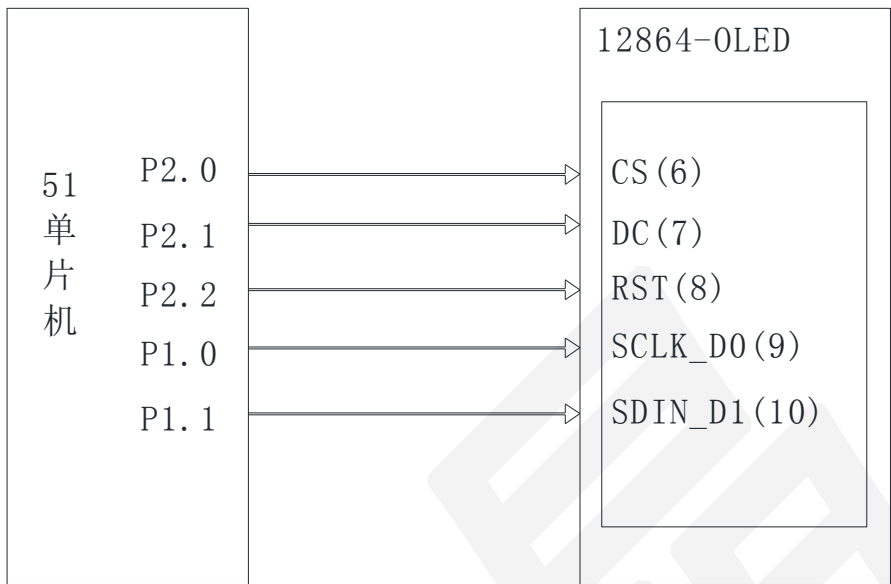
### b) 8 位 8080 时序连接



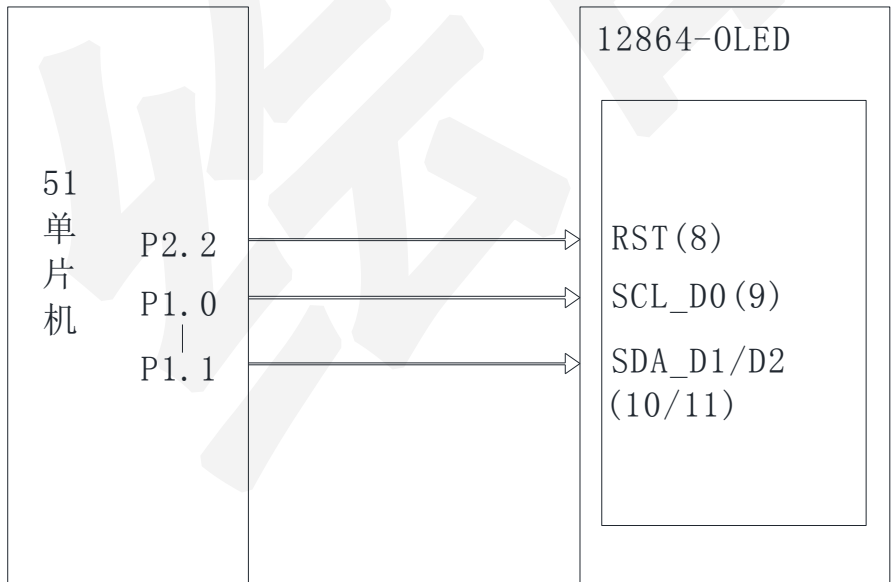
### c) 8 位 6800 时序连接



d) 4 线 SPI 串口时序



e) I<sup>2</sup>C 总线时序



十、程序例程

//-----  
-----

```
// HJ12864-OLED-1
// www.huijinglcm.com
// Dot Matrix: 128*64
// Driver IC : SSD1309 (Solomon Systech)
// Interface : 8-bit 68XX/80XX Parallel, 4-wire Serial
// Revision :
// Date : 2015/03/03
//=====
=====

#include <at89x51.h>
#include <Ascii_p.h>
#define M68 // 8-bit 68XX Parallel
// BS1=0; BS2=1
#define I80 // 8-bit 80XX Parallel
// BS1=1; BS2=1
#define SPI // 4-wire Serial
// BS1=0; BS2=0
// The unused pins should be connected with VSS mostly
or floating (D2).
// Please refer to the SSD1309 specification for
detail.
//=====
// Pin Definition
//=====
#define xData P1 // Parallel Data Input/Output

#define SCLK P1_0 // Serial Clock Input
#define SDIN P1_1 // Serial Data Input

#define RES P2_2 // Reset
#define CS P2_0 // Chip Select
#define DC P2_1 // Data/Command Control

#define E P2_4 // Read/Write Enable
#define RW P2_3 // Read/Write Select

#define RD P2_4 // Read Signal
#define WR P2_3 // Write Signal

unsigned char ml,z,zl,d,d1,s,s1,s10,s100;
unsigned char code hui1[];
```



```
unsigned char code jing1[];
unsigned char code kel[];
unsigned char code jil[];
unsigned char code Ascii_2[107][5];
unsigned char code Ascii_1[240][5];
unsigned char code Huijing[][48];
unsigned char code ascii_table_8x16[95][16];
unsigned char code ascii_table_5x8[95][5];
//-----
// Delay Time
//-----
void uDelay(unsigned char l)
{
    while(l--);
}

void Delay(unsigned char n)
{
    unsigned char i,j,k;

    for(k=0;k<n;k++)
    {
        for(i=0;i<40;i++)
        {
            for(j=0;j<15;j++)
            {
                uDelay(203);
            }
        }
    }
}
/*
void Delay(unsigned char n)
{
    unsigned char i,j,k;

    for(k=0;k<n;k++)
    {
        for(i=0;i<131;i++)
        {
            for(j=0;j<15;j++)
```

```
        {
            uDelay(203);
        }
    }
}

*/
//-----
// Read/Write Sequence
//-----
#ifdef M68                // 8-bit 68XX Parallel
void Write_Command(unsigned char Data)
{
    DC=0;
    CS=0;
    RW=0;
    E=1;
    xData=Data;
    E=0;
    RW=1;
    CS=1;
    DC=1;
}

void Write_Data(unsigned char Data)
{
    DC=1;
    CS=0;
    RW=0;
    E=1;
    xData=Data;
    E=0;
    RW=1;
    CS=1;
    DC=1;
}

#endif

#ifdef I80                // 8-bit 80XX Parallel
void Write_Command(unsigned char Data)
```

```
{
    DC=0;
    CS=0;
    WR=0;
    xData=Data;
    WR=1;
    CS=1;
    DC=1;
}

void Write_Data(unsigned char Data)
{
    DC=1;
    CS=0;
    WR=0;
    xData=Data;
    WR=1;
    CS=1;
    DC=1;
}
#endif

#ifdef SPI // 4-wire Serial
void Write_Command(unsigned char Data)
{
    unsigned char i;

    CS=0;
    DC=0;
    for (i=0; i<8; i++)
    {
        SCLK=0;
        SDIN=(Data&0x80)>>7;
        Data = Data << 1;
        //    uDelay(1);
        SCLK=1;
        //    uDelay(1);
    }
    //    SCLK=0;
    DC=1;
}
```

```
        CS=1;
    }

void Write_Data(unsigned char Data)
{
    unsigned char i;

    CS=0;
    DC=1;
    for (i=0; i<8; i++)
    {
        SCLK=0;
        SDIN=(Data&0x80)>>7;
        Data = Data << 1;
        //    uDelay(1);
        SCLK=1;
        //    uDelay(1);
    }
    // SCLK=0;
    DC=1;
    CS=1;
}
#endif

//-----
//  Instruction Setting
//-----
void Set_Start_Column(unsigned char d)
{
    Write_Command(0x00+d/16);    // Set Lower Column Start Address for Page
    Addressing Mode
                                // Default => 0x00
    Write_Command(0x10+d/16);    // Set Higher Column Start Address for Page
    Addressing Mode
                                // Default => 0x10
}

void Set_Addresssing_Mode(unsigned char d)
{

```

```
Write_Command(0x20);          // Set Memory Addressing Mode
Write_Command(d);              // Default => 0x02
                                // 0x00 => Horizontal Addressing Mode
                                // 0x01 => Vertical Addressing Mode
                                // 0x02 => Page Addressing Mode
}

void Set_Column_Address(unsigned char a, unsigned char b)
{
    Write_Command(0x21);        // Set Column Address
    Write_Command(a);           // Default => 0x00 (Column Start Address)
    Write_Command(b);           // Default => 0x7F (Column End Address)
}

void Set_Page_Address(unsigned char a, unsigned char b)
{
    Write_Command(0x22);        // Set Page Address
    Write_Command(a);           // Default => 0x00 (Page Start Address)
    Write_Command(b);           // Default => 0x07 (Page End Address)
}

void Set_Start_Line(unsigned char d)
{
    Write_Command(0x40|d);       // Set Display Start Line
                                // Default => 0x40 (0x00)
}

void Set_Contrast_Control(unsigned char d)
{
    Write_Command(0x81);         // Set Contrast Control for Bank 0
    Write_Command(d);           // Default => 0x7F
}

void Set_Segment_Remap(unsigned char d)
{
    Write_Command(d);           // Set Segment Re-Map
                                // Default => 0xA0
}
```

```
//      0xA0 => Column Address 0 Mapped to SEG0
//      0xA1 => Column Address 0 Mapped to SEG127
}

void Set_Entire_Display(unsigned char d)
{
    Write_Command(d);          // Set Entire Display On / Off
                                // Default => 0xA4
                                //      0xA4 => Normal Display
                                //      0xA5 => Entire Display On
}

void Set_Inverse_Display(unsigned char d)
{
    Write_Command(d);          // Set Inverse Display On/Off
                                // Default => 0xA6
                                //      0xA6 => Normal Display
                                //      0xA7 => Inverse Display On
}

void Set_Multiplex_Ratio(unsigned char d)
{
    Write_Command(0xA8);        // Set Multiplex Ratio
    Write_Command(d);          // Default => 0x3F (1/64 Duty)
}

void Set_Display_On_Off(unsigned char d)
{
    Write_Command(d);          // Set Display On/Off
                                // Default => 0xAE
                                //      0xAE => Display Off
                                //      0xAF => Display On
}

void Set_Start_Page(unsigned char d)
{
    Write_Command(0xB0|d);      // Set Page Start Address for Page
```

Addressing Mode

```
        // Default => 0xB0 (0x00)
    }

void Set_Common_Remap(unsigned char d)
{
    Write_Command(d);          // Set COM Output Scan Direction
                                // Default => 0xC0
                                // 0xC0 => Scan from COM0 to 63
                                // 0xC8 => Scan from COM63 to 0
}

void Set_Display_Offset(unsigned char d)
{
    Write_Command(0xD3);       // Set Display Offset
    Write_Command(d);          // Default => 0x00
}

void Set_Display_Clock(unsigned char d)
{
    Write_Command(0xD5);       // Set Display Clock Divide Ratio / Oscillator
Frequency
    Write_Command(d);          // Default => 0x70
                                // D[3:0] => Display Clock Divider
                                // D[7:4] => Oscillator Frequency
}

void Set_Low_Power(unsigned char d)
{
    Write_Command(0xD8);       // Set Low Power Display Mode
    Write_Command(d);          // Default => 0x04 (Normal Power Mode)
}

void Set_Precharge_Period(unsigned char d)
{
    Write_Command(0xD9);       // Set Pre-Charge Period
    Write_Command(d);          // Default => 0x22 (2 Display Clocks [Phase 2])
}
```

```
/ 2 Display Clocks [Phase 1])  
                                //      D[3:0] => Phase 1 Period in 1~15 Display Clocks  
                                //      D[7:4] => Phase 2 Period in 1~15 Display Clocks  
}
```

```
void Set_Common_Config(unsigned char d)  
{  
    Write_Command(0xDA);          // Set COM Pins Hardware Configuration  
    Write_Command(d);              //      Default => 0x12  
                                   //      Alternative COM Pin Configuration  
                                   //      Disable COM Left/Right Re-Map  
}
```

```
void Set_VCOMH(unsigned char d)  
{  
    Write_Command(0xDB);          // Set VCOMH Deselect Level  
    Write_Command(d);              //      Default => 0x34 (0.78*VCC)  
}
```

```
void Set_NOP()  
{  
    Write_Command(0xE3);          // Command for No Operation  
}
```

```
void Set_Command_Lock(unsigned char d)  
{  
    Write_Command(0xFD);          // Set Command Lock  
    Write_Command(d);              //      Default => 0x12  
                                   //      0x12 => Driver IC interface is unlocked from  
entering command.  
                                   //      0x16 => All Commands are locked except 0xFD.  
}
```

```
//=====
```

```
==
```

```
// Global Variables
```

```
//=====
```

---

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001     FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)



```
==  
#define XLevelL      0x00  
#define XLevelH      0x10  
#define XLevel       ((XLevelH&0x0F)*16+XLevelL)  
#define Max_Column   128  
#define Max_Row      64  
#define Brightness   0xBF//0xBF  
  
//-----  
==  
  
//-----  
//  Show Regular Pattern (Full Screen)  
//-----  
void Fill_RAM(unsigned char Data)  
{  
    unsigned char i,j;  
  
    for(i=0;i<8;i++)  
    {  
        Set_Start_Page(i);  
        Set_Start_Column(0x00);  
  
        for(j=0;j<128;j++)  
        {  
            Write_Data(Data);  
        }  
    }  
}  
  
//-----  
//  Show Regular Pattern (Partial or Full Screen)  
//  
//    a: Start Page  
//    b: End Page  
//    c: Start Column  
//    d: Total Columns  
//-----  
void Fill_Block(unsigned char Data, unsigned char a, unsigned char b, unsigned char
```

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001     FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

```
c, unsigned char d)
{
    unsigned char i,j;

    for(i=a;i<(b+1);i++)
    {
        Set_Start_Page(i);
        Set_Start_Column(c);

        for(j=0;j<d;j++)
        {
            Write_Data(Data);
        }
    }
}

//-----
// Show Checkboard (Full Screen)
//-----
void Checkerboard(unsigned char a, unsigned char b)
{
    unsigned char i,j;

    for(i=0;i<8;i++)
    {
        Set_Start_Page(i);
        Set_Start_Column(0x00);

        for(j=0;j<64;j++)
        {
            Write_Data(a);
            Write_Data(b);
        }
    }
}

//-----
// Show Frame (Full Screen)
//-----
void Frame()
```

```
{
    unsigned char i,j;

    Set_Start_Page(0x00);
    Set_Start_Column(XLevel);

    for(i=0;i<Max_Column;i++)
    {
        Write_Data(0x01);
    }

    Set_Start_Page(0x07);
    Set_Start_Column(XLevel);

    for(i=0;i<Max_Column;i++)
    {
        Write_Data(0x80);
    }

    for(i=0;i<8;i++)
    {
        Set_Start_Page(i);

        for(j=0;j<Max_Column;j+=(Max_Column-1))
        {
            Set_Start_Column(XLevel+j);

            Write_Data(0xFF);
        }
    }
}

//-----
// Show Character (5x7)
//
//   a: Database
//   b: Ascii
//   c: Start Page
//   d: Start Column
//-----

void Show_Font57(unsigned char a, unsigned char b, unsigned char c, unsigned char
```

```
d)
{
    unsigned char *Src_Pointer;
    unsigned char i;

    switch(a)
    {
        case 1:
            Src_Pointer=&Ascii_1[(b-1)][0];
            break;
        case 2:
            Src_Pointer=&Ascii_2[(b-1)][0];
            break;
    }
    Set_Start_Page(c);
    Set_Start_Column(d);

    for(i=0;i<5;i++)
    {
        Write_Data(*Src_Pointer);
        Src_Pointer++;
    }
    Write_Data(0x00);
}

//-----
//  Show String
//
//  a: Database
//  b: Start Page
//  c: Start Column
//  * Must write "0" in the end...
//-----

void Show_String(unsigned char a, unsigned char *Data_Pointer, unsigned char b,
unsigned char c)
{
    unsigned char *Src_Pointer;

    Src_Pointer=Data_Pointer;
    Show_Font57(1,96,b,c);           // No-Break Space
                                     // Must be written first before the string start...
```

```
while(1)
{
    Show_Font57(a,*Src_Pointer,b,c);
    Src_Pointer++;
    c+=6;
    if(*Src_Pointer == 0) break;
}
}

//-----
// Show Pattern (Partial or Full Screen)
//
//   a: Start Page
//   b: End Page
//   c: Start Column
//   d: Total Columns
//-----

void Show_Pattern(unsigned char *Data_Pointer, unsigned char a, unsigned char b,
unsigned char c, unsigned char d) // (内容, 页首, 页末, 列首, 列宽)
{
    unsigned char *Src_Pointer;
    unsigned char i,j;

    Src_Pointer=Data_Pointer;
    for(i=a;i<(b+1);i++)//写 4 页
    {
        Set_Start_Page(i);//设置首页
        Set_Start_Column(c);//设置首列

        for(j=0;j<d;j++)//写 48 列宽
        {
            Write_Data(*Src_Pointer);
            Src_Pointer++;
        }
    }
}

//-----
// Vertical / Fade Scrolling (Partial or Full Screen)
```

```
//
// a: Scrolling Direction
// "0x00" (Upward)
// "0x01" (Downward)
// b: Set Top Fixed Area
// c: Set Vertical Scroll Area
// d: Set Numbers of Row Scroll per Step
// e: Set Time Interval between Each Scroll Step
//-----
void Vertical_Scroll(unsigned char a, unsigned char b, unsigned char c, unsigned
char d, unsigned char e)// (上/下滚动, 首行, 尾行, 滚行, 滚程)
{
    unsigned int i,j;

    Write_Command(0xA3);          // Set Vertical Scroll Area 设置垂直滚动
    Write_Command(b);             // Default => 0x00 (Top Fixed Area)首行
    Write_Command(c);             // Default => 0x40 (Vertical Scroll Area)尾
    行

    switch(a)
    {
        case 0://上滚动
            for(i=0; i<c; i+=d)
            {
                Set_Start_Line(i);
                for(j=0; j<e; j++)
                {
                    uDelay(200);
                }
            }
            break;
        case 1://下滚动
            for(i=0; i<c; i+=d)
            {
                Set_Start_Line(c-i);
                for(j=0; j<e; j++)
                {
                    uDelay(200);
                }
            }
            break;
    }
}
```

```
Set_Start_Line(0x00);
}

//-----
// Continuous Horizontal Scrolling (Partial or Full Screen)
//
// a: Scrolling Direction
// "0x00" (Rightward)
// "0x01" (Leftward)
// b: Define Start Page Address
// c: Define End Page Address
// d: Define Start Column Address
// e: Define End Column Address
// f: Set Time Interval between Each Scroll Step in Terms of Frame Frequency
// g: Delay Time
//-----
void Horizontal_Scroll(unsigned char a, unsigned char b, unsigned char c, unsigned
char d, unsigned char e, unsigned char f, unsigned char g)
{
    Write_Command(0x26|a);          // Horizontal Scroll Setup
    Write_Command(0x00);
    Write_Command(b);
    Write_Command(f);
    Write_Command(c);
    Write_Command(0x00);
    Write_Command(d);
    Write_Command(e);
    Write_Command(0x2F);          // Activate Scrolling
    Delay(g);
}

//-----
// Continuous Vertical / Horizontal / Diagonal Scrolling (Partial or Full Screen)
//
// a: Scrolling Direction
// "0x00" (Vertical & Rightward)
// "0x01" (Vertical & Leftward)
// b: Set Numbers of Column Scroll per Step (Horizontal / Diagonal Scrolling)
// "0x00" (Horizontal / Diagonal Scrolling Off)
// "0x01" (Horizontal / Diagonal Scrolling by 1 Column)
```

```
// c: Define Start Row Address (Horizontal / Diagonal Scrolling)
// d: Define End Page Address (Horizontal / Diagonal Scrolling)
// e: Define Start Column Address
// f: Define End Column Address
// g: Set Top Fixed Area (Vertical Scrolling)
// h: Set Vertical Scroll Area (Vertical Scrolling)
// i: Set Numbers of Row Scroll per Step (Vertical / Diagonal Scrolling)
// j: Set Time Interval between Each Scroll Step in Terms of Frame Frequency
// k: Delay Time
// * e+f must be less than or equal to the Multiplex Ratio...
//-----
void Continuous_Scroll(unsigned char a, unsigned char b, unsigned char c, unsigned
char d, unsigned char e, unsigned char f, unsigned char g, unsigned char h, unsigned
char i, unsigned char j, unsigned char k)
{
    // (垂直水平向左/右, 有无水平滚动, 首页, 尾页, 首列, 尾列, 区域顶点, 区域底,
    滚动行数, 滚动间隔, 延时)
    Write_Command(0xA3);           // Set Vertical Scroll Area 设置垂直滚动区域
    Write_Command(g);              // Default => 0x00 (Top Fixed Area)
    Write_Command(h);              // Default => 0x40 (Vertical Scroll Area)

    Write_Command(0x29+a);         // Continuous Vertical & Horizontal Scroll
    Setup 垂直和水平向右滚动
    Write_Command(b); // 有无水平滚动
    Write_Command(c); // 起始页
    Write_Command(j); // 滚动间隔
    Write_Command(d); // 结束页
    Write_Command(i); // 滚动行数
    Write_Command(e); // 首列
    Write_Command(f); // 尾列
    Write_Command(0x2F);           // Activate Scrolling 激活
    Delay(k);
}

//-----
// Deactivate Scrolling (Full Screen)
//-----
void Deactivate_Scroll()
{
    Write_Command(0x2E);           // Deactivate Scrolling
}
```



```
//-----  
// Fade In (Full Screen)  
//-----  
void Fade_In()  
{  
    unsigned int i;  
  
    Set_Display_On_Off(0xAF);  
    for(i=0;i<(Brightness+1);i++)  
    {  
        Set_Contrast_Control(i);//渐亮  
        uDelay(200);  
        uDelay(200);  
        uDelay(200);  
    }  
}  
  
//-----  
// Fade Out (Full Screen)  
//-----  
void Fade_Out()  
{  
    unsigned int i;  
  
    for(i=(Brightness+1);i>0;i--)  
    {  
        Set_Contrast_Control(i-1);//渐暗  
        uDelay(200);  
        uDelay(200);  
        uDelay(200);  
    }  
    Set_Display_On_Off(0xAE);  
}  
  
//-----  
// Sleep Mode  
//  
// "0x00" Enter Sleep Mode
```

```
//      "0x01" Exit Sleep Mode
//-----
void Sleep(unsigned char a)
{
    switch(a)
    {
        case 0:
            Set_Display_On_Off(0xAE);
            Set_Entire_Display(0xA5);
            break;
        case 1:
            Set_Entire_Display(0xA4);
            Set_Display_On_Off(0xAF);
            break;
    }
}

//-----
// Connection Test
//-----
void Test()
{
    unsigned char i;

    RES=0;
    for(i=0;i<200;i++)
    {
        uDelay(200);
    }
    RES=1;

    Set_Entire_Display(0xA5);      // Enable Entire Display On (0xA4/0xA5)

    while(1)
    {
        Set_Display_On_Off(0xAF); // Display On (0xAE/0xAF)
        Delay(2);
        Set_Display_On_Off(0xAE); // Display Off (0xAE/0xAF)
        Delay(2);
    }
}
```

```
//=====
==
// Initialization
//=====
==
void OLED_Init()
{
    unsigned char i;

    RES=0;
    for(i=0;i<200;i++)
    {
        uDelay(200);
    }
    RES=1;

    Set_Command_Lock(0x12);          // (0x12 解锁/0x16)      解锁驱动 IC FDH
    Set_Display_On_Off(0xAE);        // (0xAE 关/0xAF 开)      显示关
    Set_Display_Clock(0xA0);         // Set Clock as 116 Frames/Sec      显
示频率
    Set_Multiplex_Ratio(0x3F);        // 1/64 Duty (0x0F~0x3F)      占
空比
    Set_Display_Offset(0x00);         // Shift Mapping RAM Counter (0x00~0x3F) 显示偏
移量
    Set_Start_Line(0x00);             // Set Mapping RAM Display Start Line
(0x00~0x3F) 设置起始行
    Set_Low_Power(0x04);             // Set Normal Power Mode (0x04/0x05) 设置正
常的功率模式（未找到
    Set_Addressing_Mode(0x02);        // (0x00 水平/0x01 垂直/0x02 页面) 设置页面
寻址方式

    Set_Segment_Remap(0xA0);          // (A0 地址 0 列 SEG0) (A1 地址 0 列 S127)
    Set_Common_Remap(0xC8);           // (0xC0 正常/0xC8 反向) COM 的扫描方向
C63-C0
    Set_Common_Config(0x12);          // Set Alternative Configuration (0x02/0x12) 选
择 COM, 映射

    Set_Contrast_Control(Brightness); // Set SEG Output Current      设置对
比度
    Set_Precharge_Period(0x82);       // Set Pre-Charge as 8 Clocks & Discharge as
```

2 Clocks 设置预充 8 变到 2

Set\_VCOMH(0x34); // Set VCOM Deselect Level 去 除 水

波纹

Set\_Entire\_Display(0xA4); // Disable Entire Display On (0xA4/0xA5) 全部显示开

Set\_Inverse\_Display(0xA6); // Disable Inverse Display On (0xA6/0xA7)  
正常显示, 正显

Fill\_RAM(0x00); // Clear Screen 清屏

Set\_Display\_On\_Off(0xAF); // Display On (0xAE/0xAF) 显示开

}

void display\_graphic\_16x16(unsigned char fb,unsigned char page,unsigned char column,unsigned char \*dp)

{

unsigned int i,j;

for(j=0;j<2;j++)

{

Set\_Start\_Page(page+j);

Set\_Start\_Column(column);

for (i=0;i<16;i++)

{

if(fb==1) Write\_Data(\*dp);

else Write\_Data(~(\*dp));

dp++;

}

}

}

void ini\_int1(void)

{

EA=1;

EX0=1;//允许外部 INT0 的中断

IT0=1;// 允许中断

}

int scankey1() interrupt 0 using 1 //使用外部中断 1, 寄存器组 3

{

while(P3^2==0){for(;;);}

IE1=0;//中断标志清零

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)

```
}

void display_string_8x16(unsigned char fb,unsigned int page,unsigned int
column,unsigned char *text)
{
    unsigned int i=0,j,k,n;
    //cs1=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;

            for(n=0;n<2;n++)
            {
                // lcd_address(page+n,column);
                Set_Start_Page(page+n);
                Set_Start_Column(column);
                for(k=0;k<8;k++)
                {
                    if (fb==1) Write_Data(ascii_table_8x16[j][k+8*n]);
                    else Write_Data(~(ascii_table_8x16[j][k+8*n]));
                }
            }
            i++;
            column+=8;
        }
        else
        {
            i++;
        }
    }
}
```

```
void display_string_5x8(unsigned int page,unsigned int column,unsigned char *text)

{
    unsigned int i=0,j,k;
    //cs1=0;
    while(text[i]>0x00)
    {
```

```
        if((text[i]>=0x20)&&(text[i]<0x7e))
        {
            j=text[i]-0x20;
//    lcd_address(page,column);
            Set_Start_Page(page);
            Set_Start_Column(column);
                for(k=0;k<5;k++)
                {
                    Write_Data(ascii_table_5x8[j][k]);
                }
            i++;
            column+=6;
        }
    else
    i++;
}
}

void display_string_8x16_t(unsigned char fb,unsigned int page,unsigned int
column,unsigned char text)
{
    unsigned int j,k,n;
    j=text+16;
    for(n=0;n<2;n++)
    {
//    lcd_address(page+n,column);
        Set_Start_Page(page+n);
        Set_Start_Column(column);
            for(k=0;k<8;k++)
            {
                if (fb==1) Write_Data(ascii_table_8x16[j][k+8*n]);
                else Write_Data(~(ascii_table_8x16[j][k+8*n]));
            }
    }
}

/*延时*/
void delay_t(unsigned int i)
{
    unsigned int j,k;
    for(j=0;j<i;j++)
```



```
display_string_8x16(0,5,93,"OLED");
Delay(1);//延时 400MS (X)

//~~~~~@~~~~&~~~~@~~~~9 屏，自建汉字图形

Fill_RAM(0x00);          // Clear Screen
display_string_8x16(0,0,1,"HUIJINGKEJI");
display_string_8x16(1,0,92,"OLED");
display_string_8x16(1,2,1,"WELCOME TO");
display_string_5x8(2,88,"HUIJING");
display_string_5x8(3,88,"KEJI");
display_string_5x8(4,1,"TEL: 0755-23146001");
display_string_5x8(5,1,"FAX: 0755-23146002");
display_string_5x8(6,1,"www.huijinglcm.com");
display_string_5x8(7,1,"huijinglcm@sina.com");
Delay(1);//延时 400MS (X)

// Checkerboard (Test Pattern)
Checkerboard(0xaa,0xaa);
Delay(1);
Checkerboard(0x55,0x55);
Delay(1);
Checkerboard(0xff,0x00);
Delay(1);
Checkerboard(0x00,0xff);
Delay(1);
Checkerboard(0x55,0xaa);
Delay(1);
Checkerboard(0xaa,0x55);
Delay(1);
Fill_RAM(0x00);          // Clear Screen

//~~~~~横竖点扫描演示

// Frame (Test Pattern)
Frame();
// Show String - -HuiJing -0755-23146001
Show_String(1,&Name,0x03,XLevel+0x28);//
Show_String(1,&Tel,0x05,XLevel+0x16);
Delay(1);
Fill_RAM(0x00);          // Clear Screen

//~~~~~显示字符显示边框
```





域底, 滚动行数, 滚动间隔, 延时)

```
Continuous_Scroll(0x00,0x00,0x00,0x00,XLevel,XLevel+Max_Column-1,0x00,0x20,
0x01,0x00,0x01);
```

```
// Upward - Top Area
```

```
Continuous_Scroll(0x00,0x00,0x00,0x00,XLevel,XLevel+Max_Column-1,0x00,0x20,
0x1F,0x00,0x01);
```

```
// Downward - Top Area
```

```
Continuous_Scroll(0x00,0x01,0x00,0x03,XLevel,XLevel+Max_Column-1,0x00,0x20,
0x01,0x00,0x02);
```

```
// Up & Rightward - Top Area
```

```
Continuous_Scroll(0x01,0x01,0x00,0x03,XLevel,XLevel+Max_Column-1,0x00,0x20,
0x1F,0x00,0x02);
```

```
// Down & Leftward - Top Area
```

```
Continuous_Scroll(0x01,0x01,0x04,0x07,XLevel,XLevel+Max_Column-1,0x00,0x20,
0x01,0x00,0x02);
```

```
// Upward - Top Area
```

```
// Leftward - Bottom Area
```

```
Continuous_Scroll(0x00,0x01,0x04,0x07,XLevel,XLevel+Max_Column-1,0x00,0x20,
0x1F,0x00,0x02);
```

```
// Downward - Top Area
```

```
// Rightward - Bottom Area
```

```
Deactivate_Scroll();//设置列地址
```

```
//~~~~~滚动演示
```

```
Fill_RAM(0x00); // Clear Screen
```

```
// Show String --HuiJing -0755-23146001
```

```
Show_String(1,&Name,0x01,XLevel+0x28);
```

```
Show_String(1,&Tel,0x03,XLevel+0x16);
```

```
Delay(2);
```

```
Fill_RAM(0x00); // Clear Screen
```

```
//~~~~~显示边框和字符
```

```
}
```

```
Fill_RAM(0x00); // Clear Screen
```

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

[www.huijinglcm.com](http://www.huijinglcm.com)

E-mail: [huijinglcm@sina.com](mailto:huijinglcm@sina.com)



