

## 目 录

第一章、	显示器外形结构尺寸图.....	3
第二章、	显示器基本功能介绍.....	4
第三章、	显示器接口定义说明.....	5
第一节、	针座管脚说明.....	5
第二节、	金手指管脚说明.....	6
第四章、	显示器的电性参数.....	7
第一节、	直流电参数.....	7
第五章、	显示器的显示结构原理.....	8
第一节、	显示器原理图方框图.....	8
第二节、	显示器控制器方框图.....	8
第三节、	内存与显示的关系.....	9
第六章、	驱动程序时序图说明.....	12
第一节、	8080 时序图 1.....	12
第二节、	8080 时序图 2.....	13
第三节、	6800 时序图 1.....	14
第四节、	6800 时序图 2.....	15
第七章、	驱动程序的指令说明.....	16
第一节、	显示模块指令表:.....	16
第二节、	基本指令详细说明.....	17
一、	初始化设置 system set.....	17
二、	休眠方式设置 sleep in.....	21
三、	显示状态设置 display on/off.....	22
四、	显示区域设置 scroll.....	23
五、	光标形状设置 csrform.....	24
六、	CGRAM 地址设置 cgram adr.....	24
七、	光标移动方向设置 csrdir.....	25
八、	点位移设置 hdot scr.....	25
九、	显示合成方式设置 overlay.....	26
十、	光标指针设置 csrw.....	27
十一、	光标指针读取 csrr.....	28
十二、	数据写入设置 write.....	28
十三、	数据读取设置 mread.....	28
第八章、	单片机与显示器连接说明.....	29
第一节、	接口应用原理图.....	29
第二节、	不带控制器连接原理图.....	29
第三节、	单片机并口连接.....	30
第九章、	单片机驱动程序源代码.....	30
第一节、	源代码解释定义声明.....	30
第二节、	接口时序函数.....	32
第三节、	液晶模块初始化.....	33
第四节、	应用函数.....	35

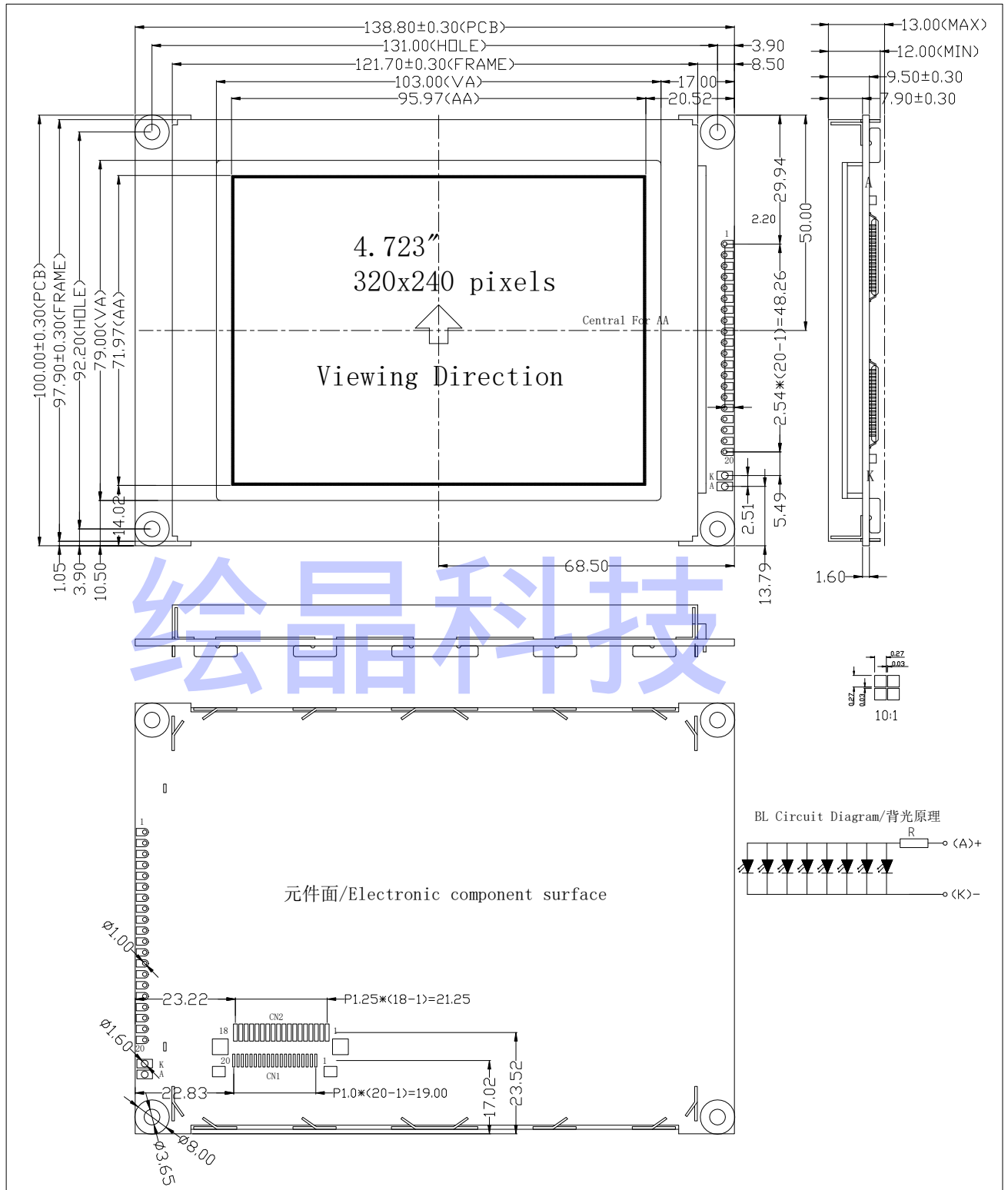
---

第五节、	主调用函数.....	40
第六节、	动态显示函数.....	43
第七节、	标准字符数据表.....	44
第十章、	版本信息.....	52

[返回目录 Ctrl+Home 或者 wep 里的返回箭头](#)

# 绘晶科技

# 第一章、显示器外形结构尺寸图



项目	参考值 (mm)
LCM 尺寸 (长×宽×厚)	138.8×100.0×12.0
可视区域 (长×宽)	103.0×79.0
点间距 (长×宽)	0.30×0.30
点尺寸 (长×宽)	0.27×0.27

## 第二章、显示器基本功能介绍

- ◇ 工作电压 5v ，可做 3.3v，**下单时请注明。**
- ◇ 显示器控制芯片为 RA8835AP3N，兼容 SED1335。
- ◇ 通讯方式：8 位 8080 并行通讯接口，可做 6800 **下单时请注明。**
- ◇ 320x240 点阵显示内存,横向取模。
- ◇ ASCII 字符图案
  - 160 个 5X7 点阵的字符
  - 缓冲存储器中的外部字符发生器最多可包含 256 个 8X16 点阵的字符
  - 本产品提供了编写不同大小字体的程序范例供用户参考或者移植用
- ◇ 功耗(背光 120Ma, TBD 非决定值)
  - 正常模式 (80mA typ VDD=5v)
- ◇ 指令方面，
  - 支持文本、图形及文图混合显示模式
  - 支持分屏显示及垂直/水平方向的平滑滚动功能
- ◇ 占空比 1/240 偏压比 1/16
- ◇ 工作温度-20 度~+70 度
- ◇ 储存温度-30 度~+80 度
- ◇ 视角方向 6 点钟
- ◇ 显示效果 ; 黑底白字 (DFSTN, 白背光) , 可做其它颜色**下单时请注明。**

## 第三章、显示器接口定义说明

### 第一节、针座管脚说明

(注: CON3, 过孔接口定义)

引脚	名称	方向	说明
1	VSS	- -	电源负端(0V)
2	VDD	- -	电源正端(+5v)
3	V0	- -	LCD 对比度调节输入, 取 VEE 电压
4	WR	I	写信号
5	RD	I	读信号
6	CS	I	片选控制
7	A0	I	数据指令线
8	REST	I	复位信号
9	D0	I/O	数据线
10	D1	I/O	数据线
11	D2	I/O	数据线
12	D3	I/O	数据线
13	D4	I/O	数据线
14	D5	I/O	数据线
15	D6	I/O	数据线
16	D7	I/O	数据线
17	A\NC	- -	背光正极\空脚
18	VEE	- -	LCD 电压输出, 用于调节对比度 V0 电压
19	BLA	- -	背光电源的正极 (+5v)
20	BLK	- -	LED(-0V)

## 第二节、金手指管脚说明

(注: CON2、CON1 相同)

引脚	名称	方向	说明
1	VSS	- -	电源负端(0V)
2	VDD	- -	电源正端(+5v)
3	V0	- -	LCD 对比度调节输入, 取 VEE 电压
4	A0(CD,RS)	I	数据指令线
5	WR	I	写信号
6	RD	I	读信号
7	D0	I/O	数据线
8	D1	I/O	数据线
9	D2	I/O	数据线
10	D3	I/O	数据线
11	D4	I/O	数据线
12	D5	I/O	数据线
13	D6	I/O	数据线
14	D7	I/O	数据线
15	CS	I	片选控制
16	RST	I	复位信号
17	VOUT(VEE)	- -	LCD 电压输出, 用于调节对比度 V0 电压
18	SEL1	I	0:8080 时序; 1:6800 时序 (默认 8080)
19	BLA	- -	背光电源的正极 (+5v)
20	BLK	- -	LED(-0V)

## 第四章、显示器的电性参数

### 第一节、 直流电参数

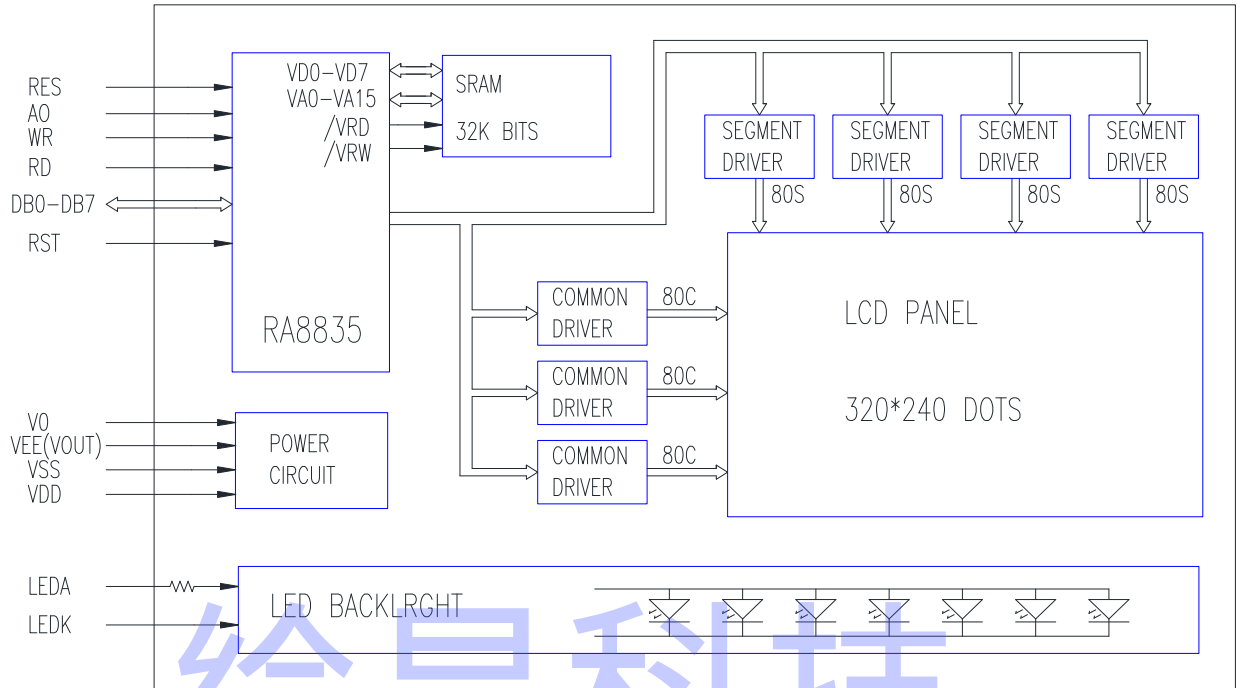
名称	符号	测试条件	参数范围			单位
			最小	标准	最大	
模块工作电压	VDD	-	4.9/3.2	5.0/3.3	5.1/3.4	V
玻璃电压	VOP/VLCD	VDD-VO	-	21.0	25.0	V
背光工作电压	VLED	-	4.9/3.2	5.0/3.3	5.1/3.4	V
IO 输入高电平	VIH	-	0.5VDD	-	VDD	V
IO 输入低电平	VIL	-	VSS	-	0.2VDD	V
LCM 输出高电平	VOH	-	VDD-0.4	-	-	V
LCM 输出低电平	VOL	-	-	-	VSS+0.4	V
模块工作电流	IDD	=VDD	-	60	120	MA
背光工作电流	ILED	=VLED	-	120	160	MA
工作温度	Tstg		-20	25	70	°C
储存温度	Tsolder		-30	25	80	°C

注：5.0V 或 3.3V 工作模式只能二选一，正常默认出货 5.0V

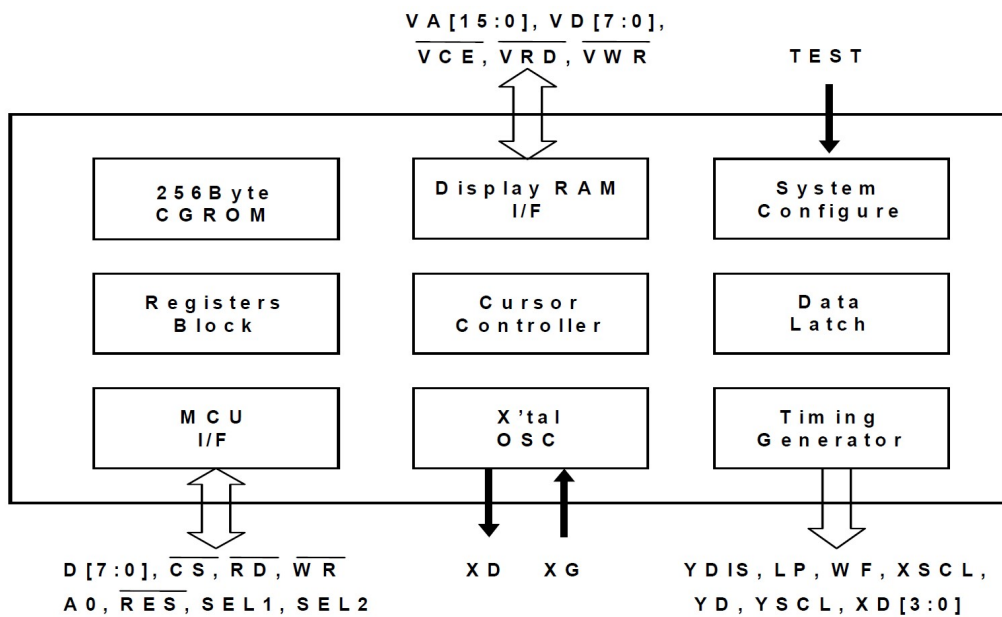
绘晶科技

## 第五章、显示器的显示结构原理

### 第一节、显示器原理图方框图



### 第二节、显示器控制器方框图





### 第三节、 内存与显示的关系

RA8835 系列支持比 LCD 面板的物理尺寸更大的虚拟屏幕地址范围,C / R。RA8835 系列的一个层可以看作是更大虚拟中的一个窗口屏幕保存在显示内存中。此窗口可分为两个块,每个块都可以显示虚拟屏幕的不同部分。

例如,这允许一个块在数据区域中动态滚动,而另一个块作为状态消息显示区域。见图 7-20 和图 7-21。

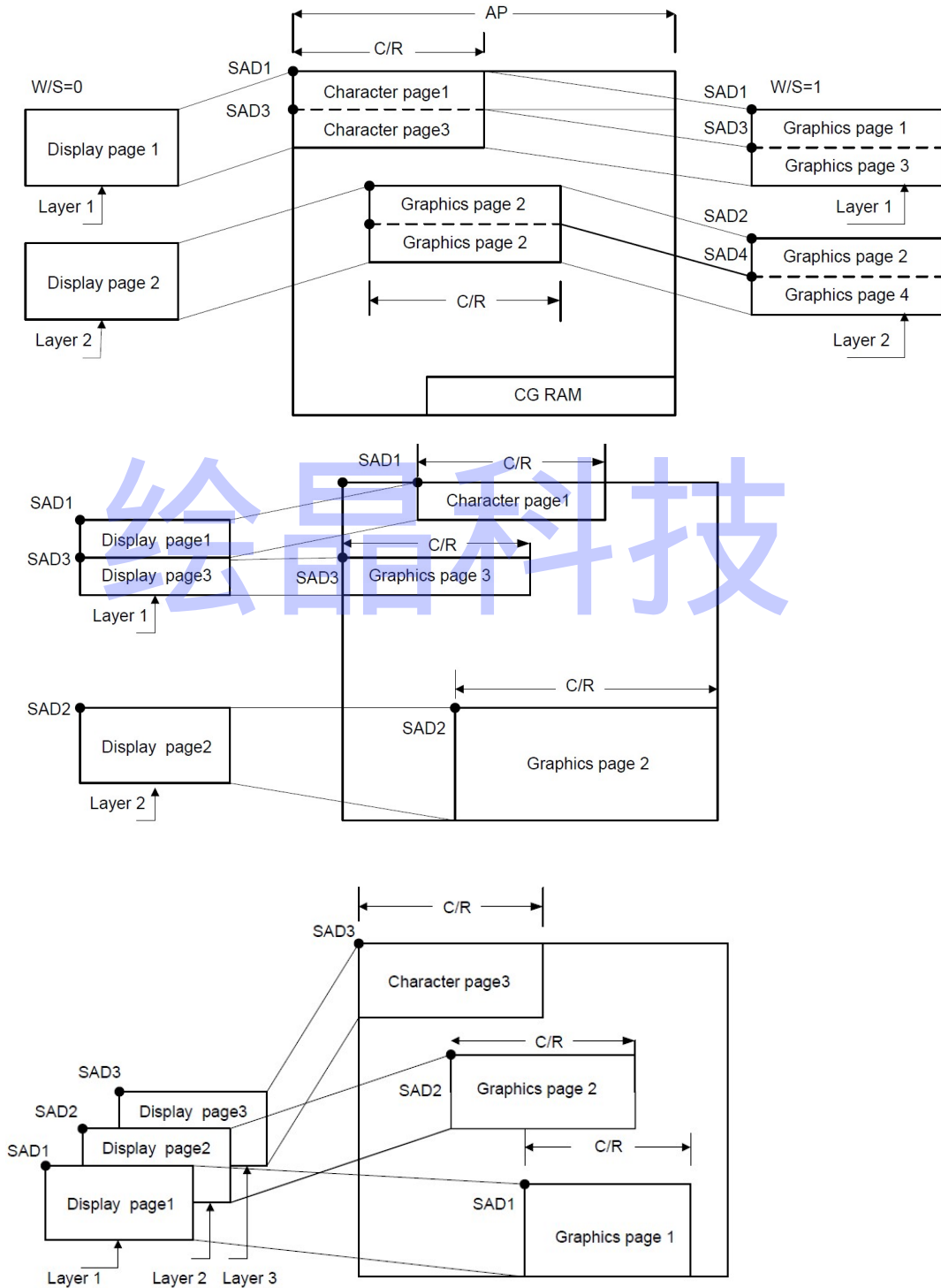


Figure 7-20: Display Layers and Memory

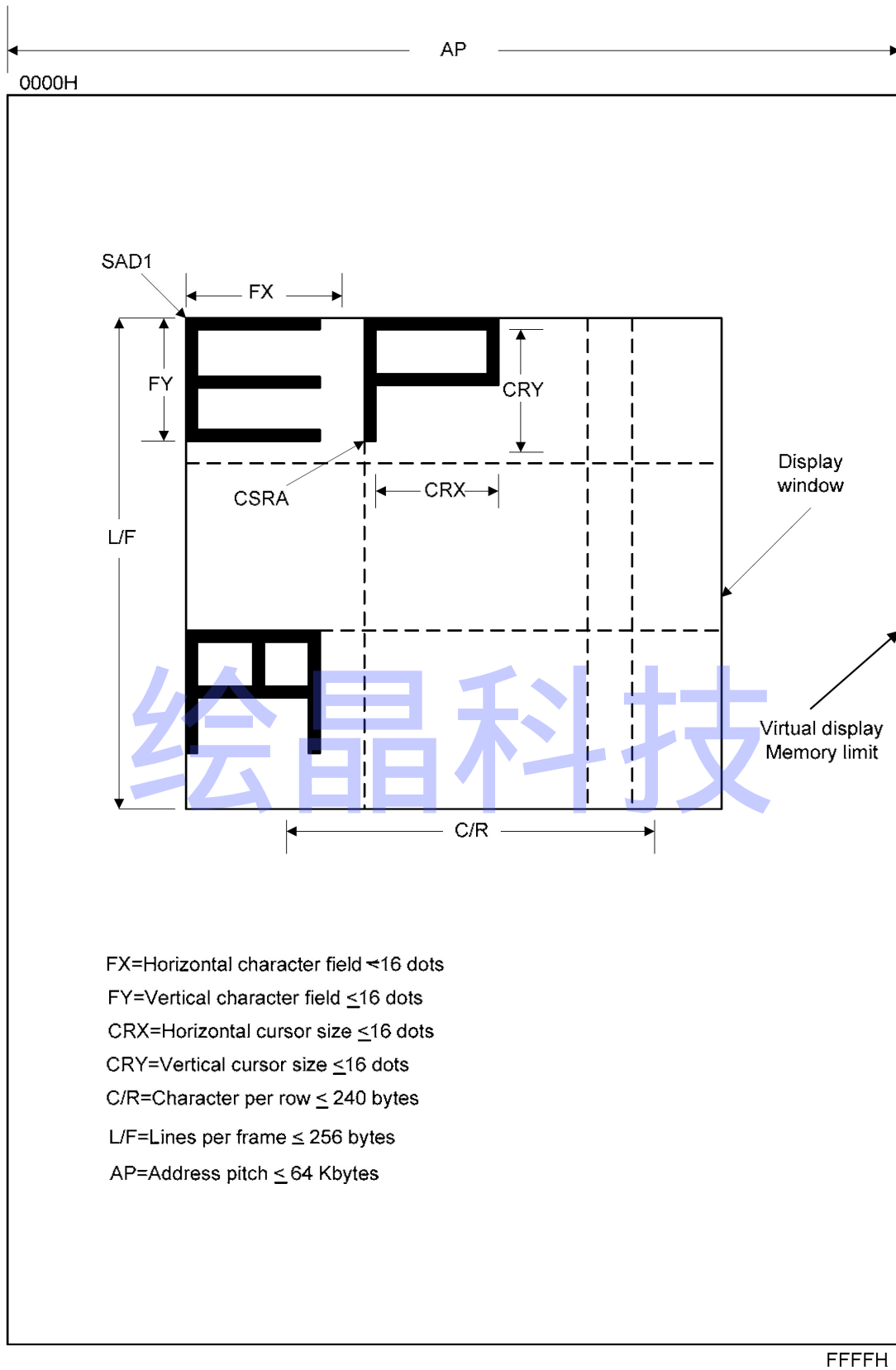


Figure 7-21: Display Windows and Memory

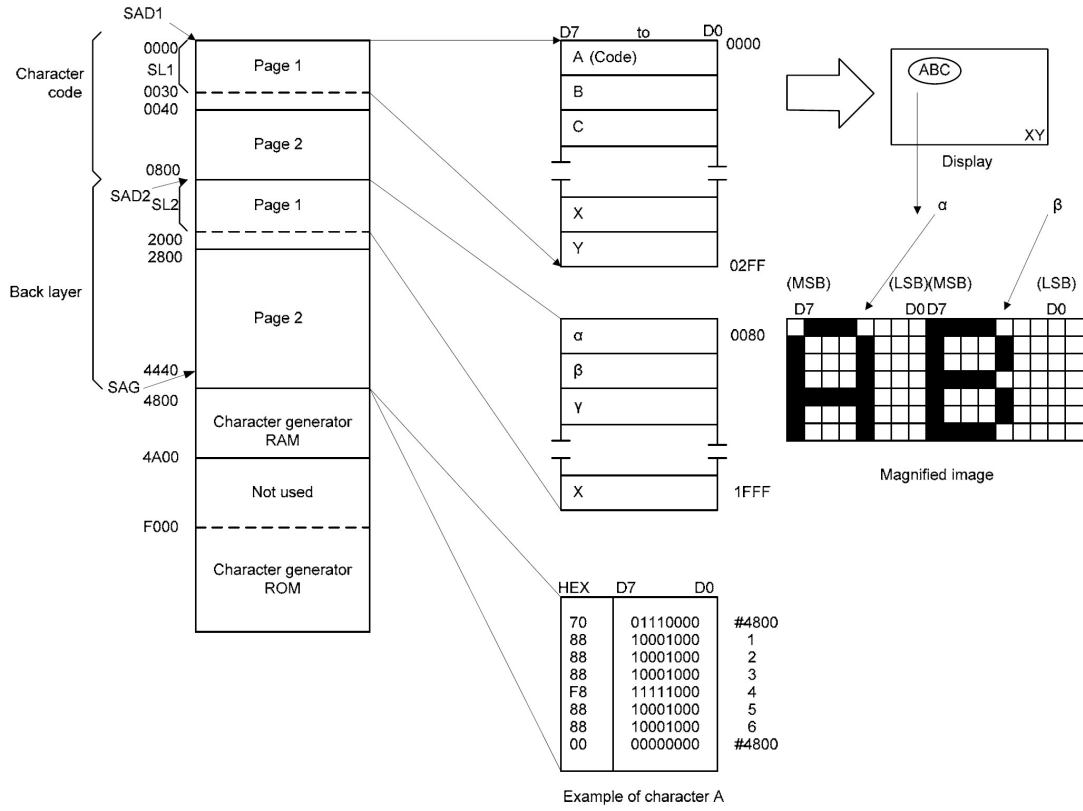
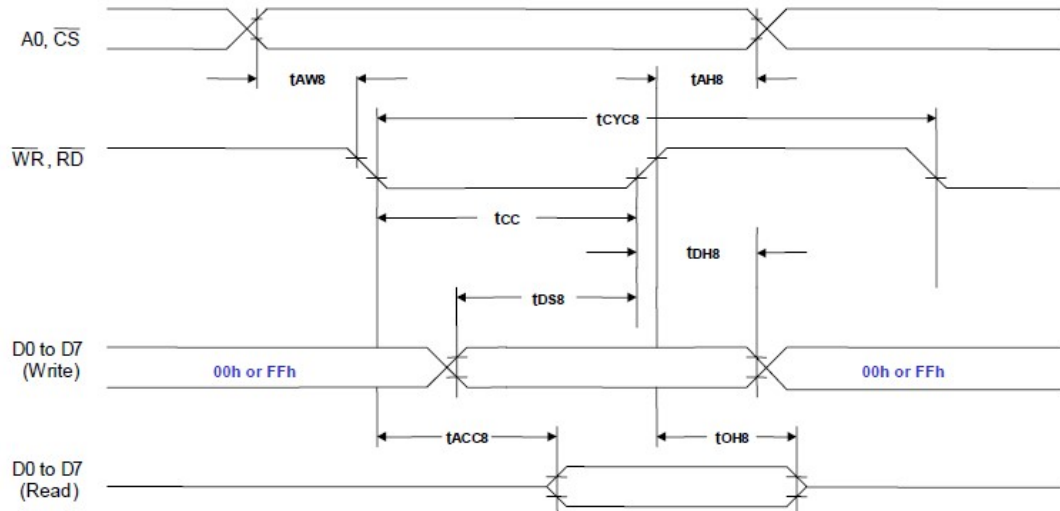


Figure 7 racters -22: Memory Map and Magnified Characters

绘晶科技

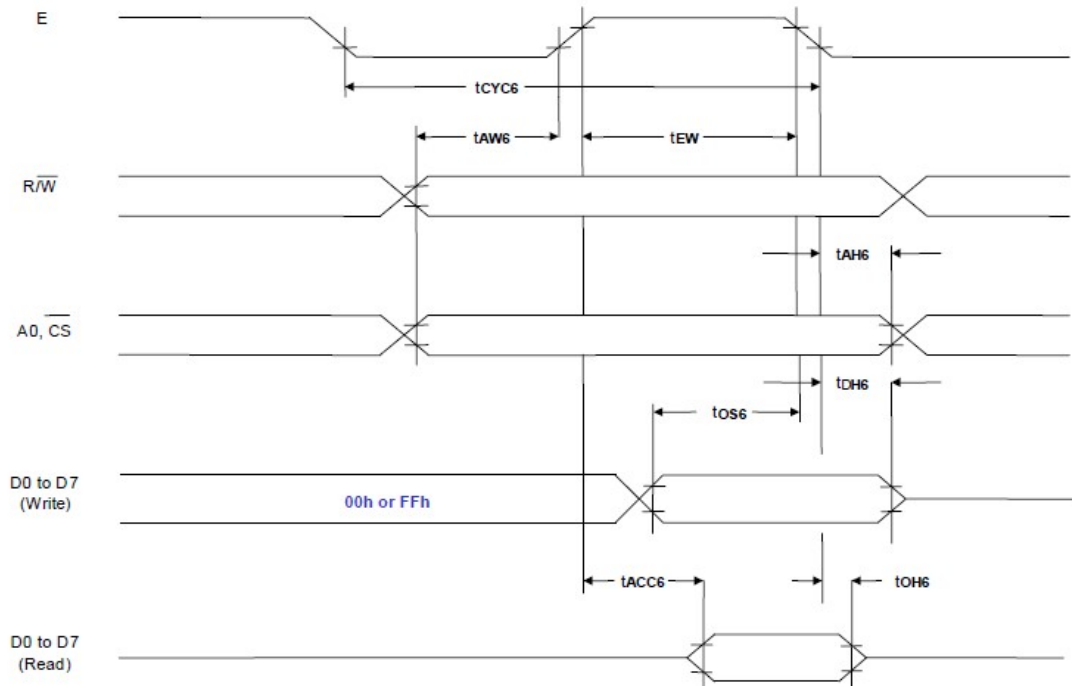
## 第六章、驱动程序时序图说明

### 第一节、8080 时序图 1



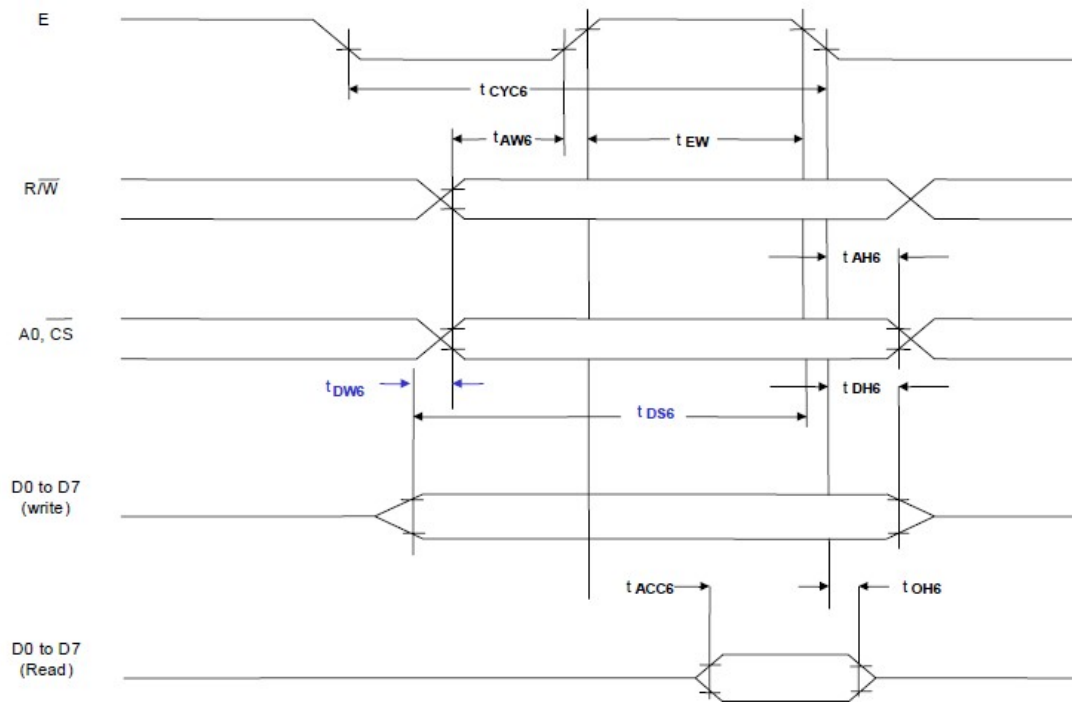
Signal	Symbol	Parameter	$V_{DD} = 4.5 \text{ to } 5.5\text{V}$		$V_{DD} = 2.7 \text{ to } 4.5\text{V}$		Unit	Condition
			Min.	Max.	Min.	Max.		
A0, $\overline{\text{CS}}$	$t_{AH8}$	Address hold time	10	—	10	—	ns	CL = 100pF
	$t_{AW8}$	Address setup time	0	—	0	—	ns	
$\overline{\text{WR}}$ , $\overline{\text{RD}}$	$t_{CYC8}$	System cycle time	note.	—	note.	—	ns	
	$t_{CC}$	Strobe pulse width	$20+t_c$	—	$50+t_c$	—	ns	
D0 to D7	$t_{DS8}$	Data setup time	120	—	120	—	ns	
	$t_{DH8}$	Data hold time	5	—	5	—	ns	
	$t_{ACC8}$	$\overline{\text{RD}}$ access time	—	50	—	80	ns	
	$t_{OH8}$	Output disable time	10	50	10	55	ns	

## 第二节、 8080 时序图 2



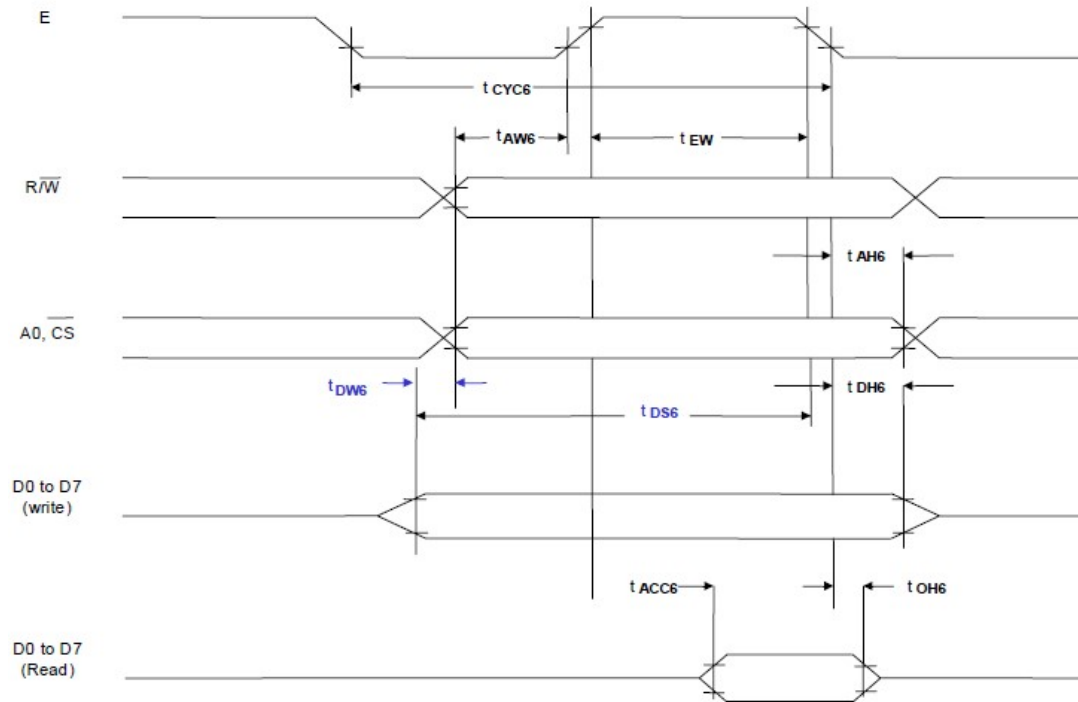
Signal	Symbol	Parameter	$V_{DD} = 4.5 \text{ to } 5.5\text{V}$		$V_{DD} = 2.7 \text{ to } 4.5\text{V}$		Unit	Condition
			Min.	Max.	Min.	Max.		
A0, $\overline{\text{CS}}$ , R/(W)	$t_{CYC6}$	System cycle time	note.	—	note.	—	ns	CL = 100 pF
	$t_{AW6}$	Address setup time	0	—	10	—	ns	
	$t_{AH6}$	Address hold time	0	—	0	—	ns	
D0 to D7	$t_{DS6}$	Data setup time	100	—	120	—	ns	
	$t_{DH6}$	Data hold time	0	—	0	—	ns	
	$t_{OH6}$	Output disable time	10	50	10	75	ns	
	$t_{ACC6}$	Access time	—	85	—	130	ns	
E	$t_{EW}$	Enable pulse width	$20+t_C$	—	$50+t_C$	—	ns	

### 第三节、 6800 时序图 1



Signal	Symbol	Parameter	$V_{DD} = 4.5 \text{ to } 5.5\text{V}$		$V_{DD} = 2.7 \text{ to } 4.5\text{V}$		Unit	Condition
			Min.	Max.	Min.	Max.		
A0, $\overline{\text{CS}}$ , R/(W)	$t_{CYC6}$	System cycle time	note.	—	note.	—	ns	CL = 100 pF
	$t_{AW6}$	Address setup time	0	—	10	—	ns	
	$t_{AH6}$	Address hold time	0	—	0	—	ns	
D0 to D7	$t_{DS6}$	Data setup time	$25+t_C$	—	$60+t_C$	—	ns	
	$t_{DH6}$	Data hold time	0	—	0	—	ns	
	$t_{OH6}$	Output disable time	10	50	10	75	ns	
	$t_{ACC6}$	Access time	—	85	—	130	ns	
	$t_{Dw6}$	Access time	5	—	10	—	ns	
E	$t_{EW}$	Enable pulse width	$20+t_C$	—	$50+t_C$	—	ns	

## 第四节、 6800 时序图 2



Signal	Symbol	Parameter	$V_{DD} = 4.5 \text{ to } 5.5\text{V}$		$V_{DD} = 2.7 \text{ to } 4.5\text{V}$		Unit	Condition
			Min.	Max.	Min.	Max.		
A0, $\overline{\text{CS}}$ , R/(W)	$t_{CYC6}$	System cycle time	note.	—	note.	—	ns	CL = 100 pF
	$t_{AW6}$	Address setup time	0	—	10	—	ns	
	$t_{AH6}$	Address hold time	0	—	0	—	ns	
D0 to D7	$t_{DS6}$	Data setup time	$25+t_c$	—	$60+t_c$	—	ns	
	$t_{DH6}$	Data hold time	0	—	0	—	ns	
	$t_{OH6}$	Output disable time	10	50	10	75	ns	
	$t_{ACC6}$	Access time	—	85	—	130	ns	
E	$t_{EW}$	Enable pulse width	$20+t_c$	—	$50+t_c$	—	ns	

## 第七章、驱动程序的指令说明

### 第一节、显示模块指令表：

带参数的指令

分类	NO	指令	指令码											HEX	指令说明	带参数	
			RD	WR	RS	D7	D6	D5	D4	D3	D2	D1	D0				
系统控制	1	系统设置	1	0	1/0	0	1	0	0	0	0	0	0	0	40H	初始化设备和显示	8
	2	SLEEP IN	1	0	1	0	1	0	1	0	0	1	1	53H	进入待机模式	0	
显示控制	3	显示开/关	1	0	1/0	0	1	0	1	1	0	0	D	58H 59H	启用和禁用显示和显示闪烁	1	
	4	SCROLL	1	0	1/0	0	1	0	0	0	1	0	0	44H	设置显示起始地址和显示区域	10	
	5	CSRFORM	1	0	1/0	0	1	0	0	1	1	0	1	5DH	设置光标类型	2	
	6	CGRAM ADR	1	0	1/0	0	1	0	1	1	1	0	0	5C	设定开始的字符发生器内存地址	2	
	7	CSRDIR	1	0	1	0	1	0	0	1	1	D I R 1	D I R 0	4CH 4FH	光标移动方向设定	0	
	8	HDOT SCR	1	0	1/0	0	1	0	1	1	0	1	0	5AH	设置水平滚动位置	1	
9	OVLAY	1	0	0/1	0	1	0	1	1	0	1	1	5BH	设置显示重叠格式	1		
图形控制	10	CSRW	1	0	1/0	0	1	0	0	0	1	1	0	46H	设置光标地址	2	
	11	CSRR	1/0	0/1	1	0	1	0	0	0	1	1	1	47H	光标地址读	2	
存储器控制	12	MWRITE	1	0	1/0	0	1	0	0	0	0	1	0	42H	写显示内存	-	
	13	MREAD	1/0	0/1	1	0	1	0	0	0	0	1	1	43H	从显示存储器读	-	



## 第二节、基本指令详细说明

### 一、初始化设置 system set

#### a) 1 system set (40h), 8个参数 初始化设置

初始化设备，设置窗口的大小，选择LCD接口格式。自本指令集的RA8835系列的基本操作参数，不正确的系统设置命令可能会导致其他命令来操作不正确。

P1 (参数1)

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	0	0	IV	1	W/S	M2	M1	M0	

M0字符发生器选择位

M0=0 为内部字符发生器

M0=1 为外部字符发生器

注意如果CG ROM的地址空间重叠显示存储器的地址空间，这部分显示内存不能写入。

M1外部字符发生器CGRAM代码区选择位

M1=0 选择 CGRAM1，字符代码为 80H-9FH（没有 D6 修正，与 CGRAM2 不连续，被视为特征的发电机的 ROM

M1=1 选择 CGRAM1+CGRAM2，字符代码为 80H-9FH 和 E0H-FFH。（D6 修正，是连续的，均视为特征的发电机的 RAM

M2外部字符发生器字符选择位

M2=0 为8X8点阵字体，（2716或等效的ROM）

M2=1 为8X16点阵字体，（2732或等效的ROM）

M0, M1, M2组合设置了字符发生器的结构

M2	M1	M0	字符发生器结构
0	0	0	内部CGROM+CGRAM1（8X8点阵）
0	0	1	外部CGROM+CGRAM1（8X8点阵）
0	1	0	内部CGROM+CGRAM1+CGRAM2（8X8点阵）
0	1	1	外部CGROM+CGRAM1+CGRAM2（8X8点阵）
1	0	1	外部CGROM+CGRAM1（8X16点阵）
1	1	1	外部CGROM+CGRAM1+CGRAM2（8X16点阵）

## 内部字符发生器的字体CGROM (160个)

		Character code bits 0 to 3															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Character code bit 4 to 7	2																
	3																
	4																
	5																
	6																
	7																
	A																
	B																
	C																
	D																
1																	

注意：阴影的位置表明，整个6×8点阵熏黑的特点。

下图显示的字符码和码分配确认通知。如果不使用内部ROM可以使用所有的代码。

低4位	高4位																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0				0	@	P	`	p	CGRAM 1								
1			!	1	A	Q	a	q									
2			“	2	B	R	b	r									
3			#	3	C	S	c	s									
4			\$	4	D	T	d	t									
5			&	5	E	U	e	u									
6			%	6	F	V	f	v									
7			‘	7	G	W	g	w									
8			(	8	H	X	h	x									
9			)	9	I	Y	i	y									
a			*	:	J	Z	j	x									
b			+	;	K	[	k	{									
c			;	<	L	¥	o	!									
d			.	=	M	]	m	}									
e			-	>	N	^	n	→									
f			/	?	0	_	o	←									

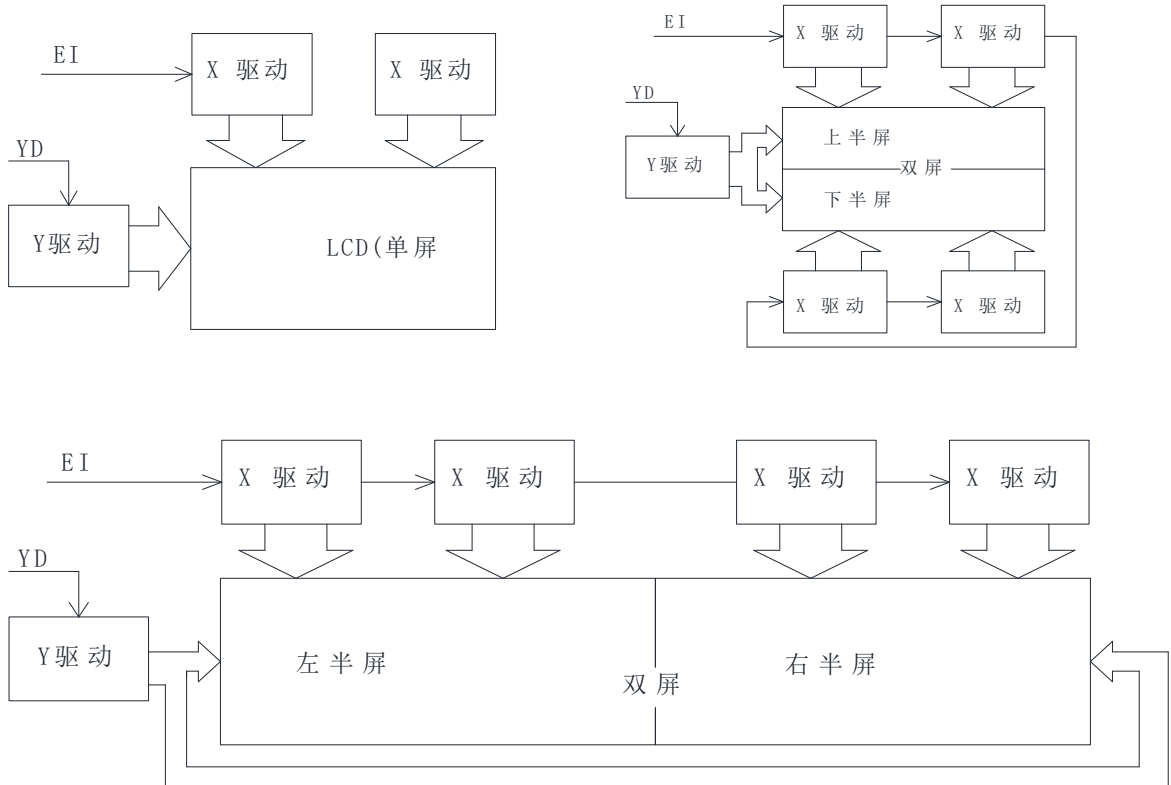
芯片上的字符代码

W/s 选择液晶显示器的驱动方法。产品驱动时序信号的类型。

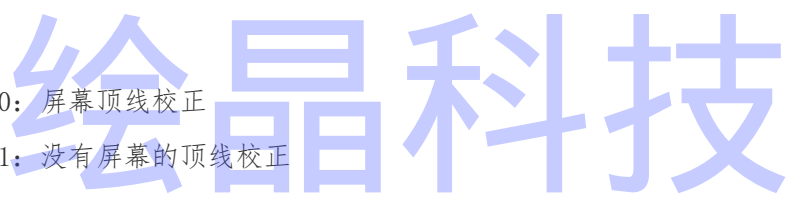
W / S = 0: 单面板驱动

W / S = 1: 双板驱动

认识一下单屏/双屏的结构(看下图)



IV = 0: 屏幕顶线校正  
 IV = 1: 没有屏幕的顶线校正



P2 (参数2) 字符宽度

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	WF	0	0	0	0	FX			

WF 交流驱动波形选择位，  
 =0 C形波形，每16行翻转一次  
 =1 B形波形，每一帧翻转一次（通常为1）  
 FX 字符宽度设置（范围8点）

P3 (参数3) 字符高度

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	WF	0	0	0	FY				

FY 字符高设置（范围16点）

P4 (参数4) 宽度字节数

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	C/R								

C/R 显示屏一行占内存的多少个字节（范围00H-EFH（1-240）），可以大一点，预留的部分不能超过64点阵

## P5 (参数5) 驱动频率的时序常数

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	TC/R								

TC/R是晶振频率FOSC转换成液晶显示驱动工作频率FR的时序常数。它可以简单的由下式示出  $TC/R \geq C/R+4$

说明 该参数的设置有时是非常重要的,错误的设置可引起以下几种现象

行扫描停止且出现高对比度的水平线

显示不稳定,出现闪烁或全亮或全暗

LP信号不足或不存在。

在取值时还是允许有一定的选取范围的。

TC/R与振荡频率的关系为:  $FOSE \geq 9 * TC/R * L/F * FR$

TC/R也可以由上两个公式联合示出。FR为液晶显示帧扫描频率,通常取什为  $FR=70HZ$ 。在控制大规模液晶显示模块时,在最大振荡频率10MHZ下,要满足 TC/R取值要求,就要降低液晶显示的帧扫描频率FR了。

## P6 (参数6)

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0		L/F								

L/F 扫描驱动的帧参数。L/F什设置了控制器所要控制的液晶显示驱动系统的扫描行数,其倒数为点空比系数。当液晶显示驱动系统为单屏结构时,L/F为一帧扫描行数的二倍,此时L/F必须为偶数值,L/F取值在0-FFH范围,对应扫描行为1-256行。

## P7

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	APL								

## P8

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	APH								

AP: 在显示存储器内分配给予一个显示行的单元数(字节数)。AP为双字节参数,APL为低8位。通常AP取值为C/R+1或大于C/R值的整数,以方便计算各显示行的显示存储器地址。

## 二、休眠方式设置 sleep in

### b) 2 sleep in (53h) 休眠方式设置

该指令强制8835等进入休眠工作方式。在该指令输入后，8835等在扫描最后一行时，将列驱动器清零度；置YDIS=0，使液晶显示驱动系统关闭显示驱动电源；将显示存储器管理信号VCE=1，以保护显示数据不被破坏；中止包括振荡在内的所有内部逻辑电路运行，但保存原设置的参数，使其进入休闲工作方式。退出休闲工作方式需要计算机重新写入SYSTEM SET指令的P1参数。该指令无参数。

### 三、显示状态设置 display on/off

#### c) 3 display on/off(58h,59h) 显示状态设置

该指令设置了显示的各种状态。它们有显示开关的设置，光标的显示器状态的设置和各显示区显示状态的设置。指令码中就含有一个参数位；该参数位控制着显示的总开关状态。

参数

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	FP5	FP4	FP3	FP2	FP1	FP0	FC1	FC0	

FC1 FC0I光标状态设置位

FC1	FC0	光标显示状态
0	0	光标显示关闭
0	1	光标显示，但不闪烁
1	0	光标以FR/32Hz的频率闪烁显示
1	1	光标以FR/64Hz的频率闪烁显示

注意：光标闪烁的亮与暗占空比为7：3。

FP1, FP0 显示一区状态设置位。FP1和FP0组合设置了显示一区显示的四种状态。

FP3, FP2 显示二，四区状态设置位。FP3和FP2组合设置了显示二，四区显示的四种状态。

FP5, FP5 显示三区状态设置位。FP5和FP4组合设置了显示三区显示的四种状态。

FC1	FC0	显示一区显示状态
FC3	FC2	显示二、区显示状态
FC5	FC4	显示三区显示状态
0	0	光标显示关闭
0	1	光标显示，但不闪烁
1	0	光标以FR/32Hz的频率闪烁显示
1	1	光标以FR/64Hz的频率闪烁显示

说明：要打开总开关以上显示才有效（468）

## 四、显示区域设置 scroll

### d) 4 scroll (44h) 十个参数 显示区域设置

该指令设置了四个显示区的起始地址SAD和各区所控制的显示行数SL。该指令设置的参数与SYSTEM SET指令设置的参数结合将确定显示RAM中的各显示区的区域范围与有效的显示区域，如图16-3所示。定时间定量的递增或递减某区的起始地址SAD，将产生显示画面的平滑流动，该指令带有10个参数。部分参数的设置是有效的，但设置的顺序不能改变也不能省略，各参数的功能如下

P1

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD1L								

P2

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD1H								

SAD1 显示一区首地址值。SAD1确定了第一显示区的起始地址。它由双字节组成。SAD1L为SAD1的低8位，SAD1H为SAD1的高8位。

P3

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SL1								

SL1 显示一区所控制的显示屏上的行数，SL1取值为00H- (L/F) H

P4

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD2L								

P5

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD2H								

SAD2 显示二区首地址值。SAD2确定了第二显示区的起始地址。它由双字节组成。SAD2L为SAD2的低8位，SAD2H为SAD2的高8位。

P6

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
0	0	1	SL2								

SL2 显示二区所控制的显示屏上的行数，SL1取值为00H- (L/F) H

P7

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD3L								

P8

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD3H								

SAD3 显示三区首地址值。SAD3确定了第三显示区的起始地址。它由双字节组成。SAD3L为SAD3的低8位，SAD3H为SAD3的高8位。

P9

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD4L								

P10

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAD4H								

SAD4 显示四区首地址值。SAD4确定了第四显示区的起始地址。它由双字节组成。SAD4L为SAD4的低8位，SAD4H为SAD4的高8位。

## 五、光标形状设置 `csrform`

### e) 5 `csrform` (5dh) 二个参数 光标形状设置

该指令设置了光标的显示开形状。光标的显示形状有线状形式各块状形式两种，光标的形状是可以设置的。该指令所带的两个参数将实现对形状的设置，

P1

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	0	0	0	0	0	CRX			

CRX 光标形状水平向点列数。CRX取值在0-7H范围内，对应光标水平向点据的点列数为1-8点， $CRX \leq FX$ 。

P2

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	CM	0	0	0	CRY				

CRY 光标形状垂直向点行数。CRY取值在1-FH范围内，对应光标垂直向占据的点行数2-16点。CRY=0或CRY>FY均无效。

CM 光标形状选择位。CM=0为线状光标显示，CM=1为块状光标显示。

说明：块状光标的大小形状由CRX和CRY设置。

线状光标的形状由CRX确定其长度。由CRY设定其所占的显示行，当CRY=FY时，光标显示呈底线光标形式；

光标显示仅在文本方式下有效。

## 六、CGRAM 地址设置 `cgram adr`

### f) 6 `cgram adr` (5ch) CGRAM首地址设置

该指令设置了CGRAM在显示存储器内的起始地址SAG。从该地址SAG起的2K字节空间 (M2=0) 或4K字节空间 (M2=1) 为CGRAM区域。该指令带有二个参数，它们是：



P1

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAGL								

P2

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	SAGH								

SAG CGRAM起始地址。SAGL为SAG的低8位地址；SAGH为SAG的高8位地址。  
说明：SAG所指地址对应的是字符代码00H的字模首地址，所以对于CGRAM1的首地址应为SAG+80H\*8。

CGRAM2区域的字符代码为E0H-FFH，但由于控制部在CGRAM2寻址上安置了异或40H的电路，所以CGRAM2的地址应按A0H-BFH计算，即CGRAM2R的首地址为SAG+A0H\*8。

## 七、光标移动方向设置 csrdir

### g) 7 csrdir (4ch, 4fh) 光标移动方向设置

该指令规定了光标指针CSR在计算机读写显示数据后自动修改方式，从显示效果上说就是光标移动的方向设置。该指令的两个参数位DIR1和DDIRO设置了光标指针的移动方向的设置，设置参数如表

DIR1	DIR0	代码	光标移动方向	地址计算公式
0	0	4CH	光标向右移动	CSR+1》CSR
0	1	4DH	光标向左移动	CSR-1》CSR
1	0	4EH	光标向上移动	CSR-AP》CSR
1	1	4FH	光标向下移动	CSR+AP》CSR

## 八、点位移设置 hdot scr

### h) 8 hdot scr (5ah) 一个参数 点位移设置

该指令设置了以点为单位的显示画面的水平位移量。该指令的作用有二：其一是在制作负向显示字符时与SYSTEM SET参数IV结合使用。通过设置该指令可以产生左边界，使得行首字符有比较好的显示效果。其二是在制作画面滚动显示效果时与SCROLL指令结合产生水平向点滚动的显示效果。但要注意的是该指令控制在显示数据合成之后向驱动系统输出之处，故该指令所产生的滚动效果是针对所有参加显示的显示区，因此要想使某一显示区产生点滚动而其化显示区不动是不可能的。该指令带有一个参数

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
----	----	----	----	----	----	----	----	----	----	----	-----

1/0	0	1	0	0	0	0	0	D	
-----	---	---	---	---	---	---	---	---	--

D 水平点位移量参数。D取值在0H-7H范围内。当D取值由0H向7H方向递增时，显示画面呈现左移效果；当D取值由7H向0H方向递减时，显示画面呈现右移效果。

说明：该指令与SCROLL指令结合使用方法：显示左移时，首先使用HDOT SCR 指令使点位移参数从0增到7，然后使用SCROLL指令将显示起始地址SDA加一，并同时 will HDOTSCR参数清零。循环上述过程，将产生显示水平向左点位移的效果。显示右移时，首先使用HDOT SCR 指令使点位移参数从7减到0，然后使用SCROLL指令将显示起始地址SAD减一，并同时 will HDOT SCR参数置7。循环上述过程，将产生显示水平向右点位移的效果。

## 九、显示合成方式设置 overlay

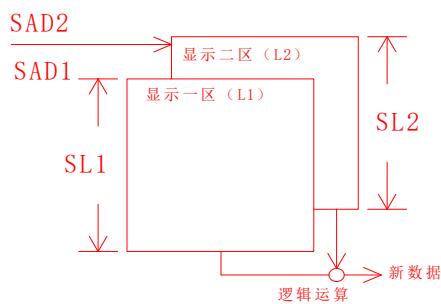
### i) 9 overlay (5bh) 一个参数 显示合成方式设置

该指令设置了显示区的合成显示方式以及显示一区和三区的显示方式。该指令带一个参数

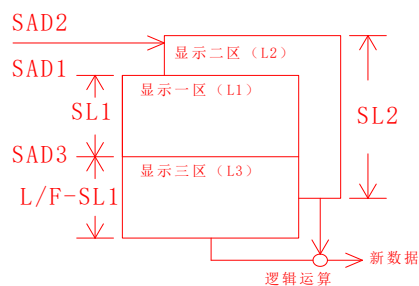
RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	0	0	0	0V	DM2	DM1	MX1	MX0	

0V 显示合成设置位。0V=0为二重合成显示，用于文本与图形或图形的混合显示；0V=1为三重合成显示，仅用于图形混合显示。

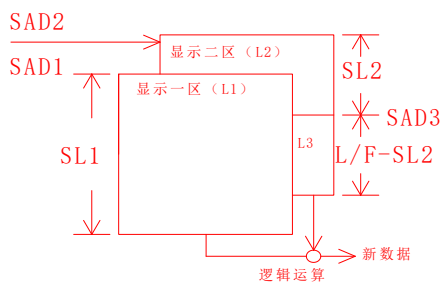
说明1：二重合成显示方式是将两个显示区的内容经某一种逻辑合成出新显示数据从而产生的显示画面。该方式将以第一，第二显示区为主，第三，第四显示区为辅的合成，如图a-d



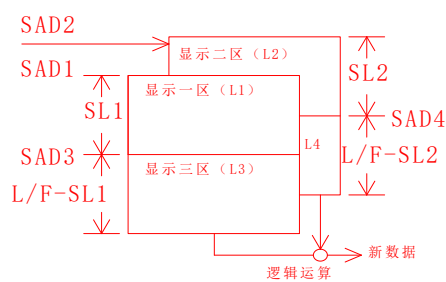
(a) 单屏结构：显示一、二区参加的二重合成显示方式



(b) 单屏结构：显示一、二、三区参加的二重合成显示方式(一)

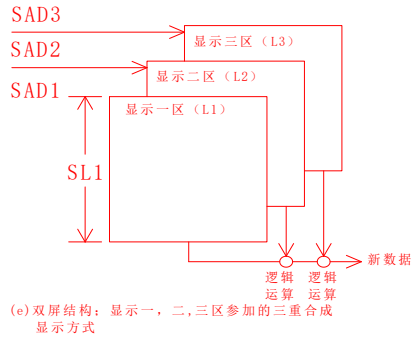


(c) 单屏结构：显示一、二、三区参加的二重合成显示方式(二)



(d) 双屏结构：显示一、二、三、四区参加的二重合成显示方式

说明2; 三重合成显示方式是将三个显示区的内容经某一种或两种逻辑运算合成出新的显示数据从而产生新的显示画面。该方式将第一, 第二, 第三显示区参加的合成, 合成方式如图e



MX1	MX0	合成方式	用途
0	0	L1+L2+L3简单逻辑或合成	底线坐标和文本图形混合显示
0	1	(L1+L2) +L3逻辑异或。或合成	负向字符显示, 区域闪烁底线
1	0	(L1。L2) +L3逻辑与。或合成	简单动画立体表达式
1	1	L1>L2>L3优先叠加	

说明1; 合成显示产生的结果仅改变显示画面而不改变各显示区的内容。

说明2: 优先叠加合成显示是三个显示区以L1覆盖L2, L2覆盖L3的方式逻辑“或”合成显示, 当关闭L1时, L2和L3与L1相同显示的部分(为“1”处)被关闭; 当L2被关闭时, L3与L2相同显示的部分(为“1”处)被关闭。

DM1 显示一区的显示属性位。DM1=0设置显示一区为文本显示属性, 其数据均作为字符代码处理; DM1=1设置显示一区为图形显示属性, 其数据均作为图形数据处理。

DM2 显示三区的显示属性位。DM2=0Y设置显示三区为文本显示属性, 其数据均作为字符代码处理; DM2=1设置显示三区为图形显示属性, 其数据均作为图形数据处理。

## 十、光标指针设置 csw

### j) 10 csw (46h) 光标指针设置

该指令设置了光标指针CSR, 该指针有两个功能, 其一作为当前光标显示的位置; 其二作为存储器当前的地址指针。两个参数

P1

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
----	----	----	----	----	----	----	----	----	----	----	-----

1/0	0	1	CSRL								
P2											
RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	CSRH								

CSR I 光标指针，CSRL为光标指针的低8位；CSRH为光标指针的高8位。

说明；当光标指针指向有效显示区域之外时，光标将自然消失，但不影响显示数据的读写。

光标指针在计算机的每次写显示数据操作后，将根据最近设置的光标指针移动方向而自动修正。

光标指针不受画面流动操作的影响。

## 十一、光标指针读取 csrr

### k) 11 csrr (47h) 光标指针读取

影该指令代码写入后，计算机将从8835等数据通道读出当前的光标指针值。光标指针读出的顺序是先CSRL，再CSRH。

P1

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1	0/1	1/0	CSRL								

P2

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1	0/1	1/0	CSRH								

## 十二、数据写入设置 write

### l) 12 nwrite (42h) 数据写入设置

RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1/0	0	1	数据								

该指令写入后将允许计算机连续的向显示存储器内写入数据。写入的单元地址由光标指针CSR提供，写操作完成后光标指针自动根据所设置的光标指针移动方向进行修改。显示数据写入操作在计算机写入下一个指令代码后结束。

## 十三、数据读取设置 mread

### m) 13 mread (43h) 数据读取设置

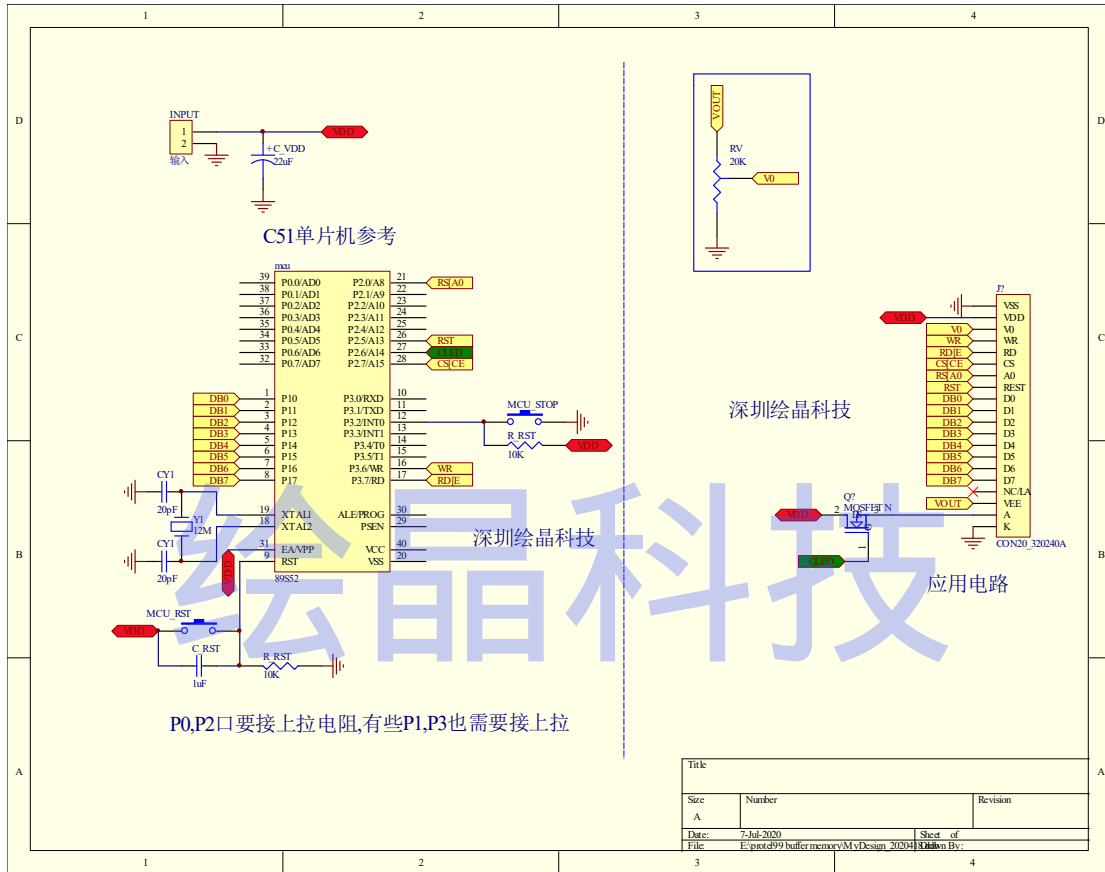
RS	WR	RD	D7	D6	D5	D4	D3	D2	D1	D0	HEX
1	0/1	1/0	数据								

该指令的写入将允许计算机连续地从显示存储器内读取数据，读取的单元地址由光标指针CSR提供，读操作完成后光标指针自动根据所设置的光标移动方向

进行修改。显示数据读取操作在计算机写入下一个指令代码结束。

# 第八章、单片机与显示器连接说明

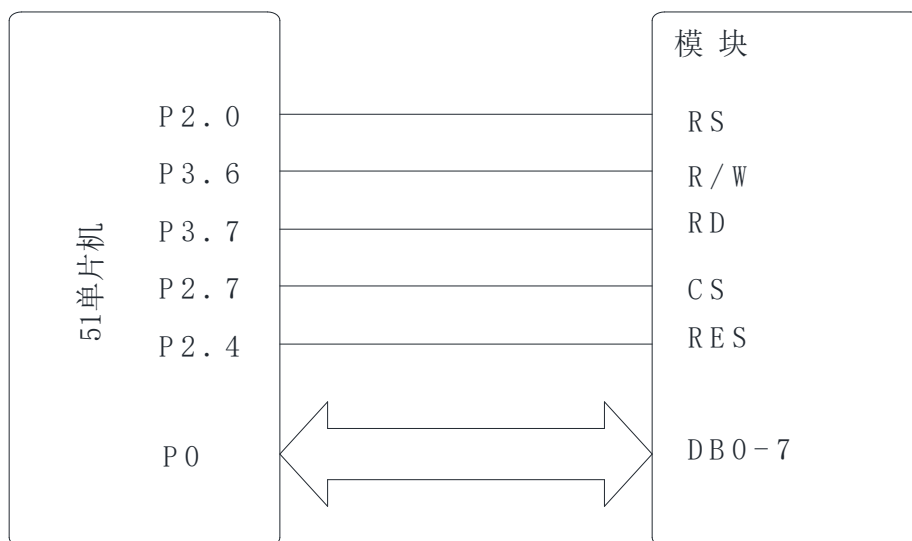
## 第一节、接口应用原理图



## 第二节、不带控制器连接原理图

欲了解更多详情可登陆我司官网查询 <http://www.huijinglcm.com>

### 第三节、单片机并口连接



## 第九章、单片机驱动程序源代码

### 第一节、源代码解释定义声明

```
//, 单片机:89S52,STC完全兼容
//晶振:12M,编译软件:KEIL 7.06
//8835控制器, 8080-8位并口时序
#include <reg51.h>
#include <intrins.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define uint unsigned int
#define uchar unsigned char
#define xchar unsigned char code
sbit A0 = P2^0;
sbit WRR = P3^6;
sbit RDD = P3^7;
sbit CS = P2^7;
sbit RES = P2^4;
sbit BF = ACC^7;
uchar code num_8x16[11][16];
uchar code num_24x48[11][144];
```

```
uchar code hui_32x32[];
uchar code jing_32x32[];
uchar code ke_32x32[];
uchar code ji_32x32[];
uchar code num_16x24_2[];
uchar code num_16x24_4[];
uchar code num_16x24_0[];
uchar code num_16x24_x[];
uchar code num_16x24_1[];
uchar code num_16x24_3[];
uchar code hui2_24x24[];
uchar code jiu_24x24[];
uchar code jiao_24x24[];
uchar code dian_24x24[];
uchar code jing2_24x24[];
uchar code ping_24x24[];
uchar code zi_24x24[];
uchar code sang_24x24[];
uint i,data1,temp,data2,j,k;
uchar m1,num,ii,z,z1,d,d1,s,s1,s10,s100;
bit k1;

/*
xchar xshm[9600]={
};
*/
//-----
void delays(uint v)
{ while(v!=0)v--;}
//-----

void time_nms(unsigned int x)//0.5ms
{
  unsigned char j;
  while(x--)
    {for(j=0;j<50;j++)
      {;}
    }
}

void delay(uint nn)
{
```

绘晶科技

```
uint mm;
while (nn-->0)
for (mm=0; mm<1000; mm++) { };
}
```

## 第二节、 接口时序函数

```
//-----
void busy()
{
CS=0; A0=0;
do
{ P0=0xff; RDD=0; delayms(1); ACC=P0; RDD=1; }
while (BF==1);
}
//-----
void wcomd(uint cdat)
{
busy();
A0=1; RDD=1; P0=cdat; WRR=0; WRR=1; CS=1;
}
//-----
void wdata(uint ddat)
{
busy();
A0=0; RDD=1; P0=ddat; WRR=0; WRR=1; CS=1;
}

//-----
void wcode(uint csrl, uint csrh)
{
uint temdat1=0, temdat2=0;
temdat1=csrl+40*csrh;
temdat2=temdat1/256;
temdat1=temdat1%256;
wcomd(0x46);
wdata(temdat1);
wdata(temdat2);
wcomd(0x42);
}
```



### 第三节、液晶模块初始化

```
//-----
void initial()
{
uint i=0;
//~~~~~指令带参数~~~~~
wcomd(0x40);    //初始化设置指令（8）
wdata(0x30);    //---P1
/*
7   DR=0 为正常运行，=1为辅助位移（双屏一行64的整倍数）
6   0
5   IV=0 首行做为边界，=1无边界（首选）空一行较理想
4   1
3   W/S=0 单屏，=1双屏
2   M2
1   M1
0   M0 （M2-M0） 组合调字符类型 详464 455
*/
wdata(0x87);    //---P2
/*
7   WF=0 C型16行反转一次，=1 B型一帧反转一次（首选）
6-3 0
2-0 FY，字符体宽度设置（点列数）0-7 多出为间隔
*/
wdata(0x07);    //---P3
/*
7   WF=0 C型16行反转一次，=1 B型一帧反转一次（首选）
6-4 0
4-0 FY，字符体高度设置（点列数）0-FH 多出为间隔
*/
wdata(39);      //---P4
//C/R
//00H-EFH（1-240个字节）有效窗口宽度设置，可长不超过64点（8字节）
wdata(66);      //---P5
/*
TC/R 驱动频率的时间常数，跟晶振有关， $TC/R \geq C/R + 4 \leq$ 
 $Fosc \geq 9*TC/R*L/F*FR$  FR为帧扫描频率通常=70Hz 晶振  $\leq 10M$  详465
*/
wdata(240);     //---P6
/*
L/F 扫描驱动的帧参数，扫描的行数，倒数为点空比
范围0-FFH（1-256行）

```

```

*/
wdata(40);          //---P7--APL
wdata(0);           //---P8--APH
//一行的单元数（字节数）APL低8位，通常C/R+1或C/R整倍
delays(0x9);
//~~~~~指令带参数~~~~~
wcomd(0x44);       //显示区域设置SCROLL
wdata(0x00); //P1 SAD1L
wdata(0x00); //P2 SAD1H
//DSA1确认第一显示区的起始地址。
wdata(240); //P3      SL1
//显示一区所控制显示屏上的行数。SL1取值为00H~(L/F) H
wdata(0x80); //P4 SAD2L
wdata(0x25); //P5 SAD2H
//DSA2确认第二显示区的起始地址。
wdata(240); //p6      SL2
//显示二区所控制显示屏上的行数。SL2取值为00H~(L/F) H
wdata(0x00); //P7 SAD3L
wdata(0x4b); //P8 SAD3H
//DSA3确认第三显示区的起始地址。
wdata(0x00); //P9 SAD4L
wdata(0x00); //P10   SAD4H
//DSA4确认第四显示区的起始地址。
delays(0x9);
//~~~~~指令带参数~~~~~
wcomd(0x5a);       //点位移设置
wdata(0x00);
//7-3   =0
//2-1   D   与SCROLL配合产生左移右移（470

delays(0x9);
//~~~~~指令带参数~~~~~
wcomd(0x5b);       //显示合成方式设置OYLAY
wdata(0x1c);
//7, 6, 5   =0,
//4,        0v=0;二重合成显示，用于文本图形，=1三重合成，用于图形混合显示
//1 0      MX1 MX0 显示合成方式设置位（底线坐标和文本图形混合并（470
//3        DM2 显示三区属性，=0为文本，=1为图形
//2        DM1 显示二区属性，=0为文本，=1为图形
delays(0x9);
//~~~~~指令带参数~~~~~
wcomd(0x59);       //显示开 58H显示关
wdata(0x04);
//0, 1    FC0 FC1 启用/禁用光标和设置闪光率 1无闪烁 2（32HZ）3（64HZ）

```

```

//2, 3  FP0 FP1 第一个屏幕块（内源）      1无闪烁 2（32HZ） 3（4HZ）
//4, 5  FP2 PF3 第二屏幕块（SAD2, sad4）
//6, 7  FP4 FP5 第三屏幕块（SAD3）
delaysms(0x9);
//~~~~~指令无参数~~~~~
wcomd(0x4c); //4C右, 4D左, 4E上, 4F下（28页）
delaysms(0x9);
//~~~~~指令带参数~~~~~
wcomd(0x46);    //光标地址
wdata(0); //低8位
wdata(0); //高8位
delaysms(0x9);
}

```

## 第四节、应用函数

```

//-----
void disp_all(uchar zf1,uchar zf2)
{
uint i=0,j=0;
wcode(0x00,0x00);
for(i=0;i<120;i++)
{
for(j=0;j<40;j++)
{ wdata(zf1);}
for(j=0;j<40;j++)
{ wdata(zf2);}
}
}
//-----
/*
void disp_bmp(xchar *str)
{
uint i=0;
wcode(0,0);
for(i=0;i<9600;i++)
{ wdata(str[i]);}
}
*/
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 2 using 3 //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
}

void ds_16x24(bit fb,uint X,uint Y,uchar *dp)//X为列坐标,以字节为单位;Y为行坐标,
以点为单位.
{
    uint addL;
    uint addH;
    uchar j;
    for(i=0;i<24;i++)
    {
        temp=0x0000+Y*40+X+40*i;
        addL=temp&0x00ff;           //取TEMP的低八位,
        addH=((temp&0xff00)>>8);    //取TEMP的高八位.
        data1=addL;
        data2=addH;
        wcomd(0x46); //显示地址
        wdata(data1);
        wdata(data2);
        wcomd(0x42);
        for(j=0;j<2;j++)
        {
            if(fb==1) wdata(*dp);
            else wdata(~(*dp));
            dp++;
        }
    }
}
```

```
void ds_24x24(bit fb,uint X,uint Y,uchar *dp)//X为列坐标,以字节为单位;Y为行坐标,
以点为单位.
{
    uint addL;
    uint addH;
    uchar j;
    for(i=0;i<24;i++)
    {
        temp=0x0000+Y*40+X+40*i;
```

```

    addL=temp&0x00ff;           //取TEMP的低八位,
    addH=((temp&0xff00)>>8);     //取TEMP的高八位.
    data1=addL;
    data2=addH;
    wcomd(0x46); //显示地址
    wdata(data1);
    wdata(data2);
    wcomd(0x42);
    for(j=0;j<3;j++)
    {
        if(fb==1) wdata(*dp);
        else wdata(~(*dp));
        dp++;
    }
}

void ds_32x32(bit fb,uint X,uint Y,uchar *dp)//X为列坐标,以字节为单位;Y为行坐标,
以点为单位.
{
    uint addL;
    uint addH;
    uchar j;
    for(i=0;i<32;i++)
    {
        temp=0x0000+Y*40+X+40*i;
        addL=temp&0x00ff;           //取TEMP的低八位,
        addH=((temp&0xff00)>>8);     //取TEMP的高八位.
        data1=addL;
        data2=addH;
        wcomd(0x46); //显示地址
        wdata(data1);
        wdata(data2);
        wcomd(0x42);
        for(j=0;j<4;j++)
        {
            if(fb==1) wdata(*dp);
            else wdata(~(*dp));
            dp++;
        }
    }
}

// 灵活的矩形绘制函数X1, Y1为起点, X2为行程, Y2为列程,

```

```
void Wr_reg(uchar x1,uchar y1,uchar x2,uchar y2,uchar d1)
{
    uint addL;
    uint addH;
    uchar j;
    for(i=0;i<y2;i++)
    {
        temp=0x0000+y1*40+x1+40*i;
        addL=temp&0x00ff;           //取TEMP的低八位,
        addH=((temp&0xff00)>>8);    //取TEMP的高八位.
        data1=addL;
        data2=addH;
        wcomd(0x46); //显示地址
        wdata(data1);
        wdata(data2);
        wcomd(0x42);
        for(j=0;j<x2;j++)
        {
            wdata(d1);
        }
    }
}

void ds_8x16(uchar x,uchar y,uchar *text)
{
    uint addL, addH;
    uchar q,i; i=0;
    temp=0x0000+y*40+x;
    addL=temp&0x00ff;           //取TEMP的低八位,
    addH=((temp&0xff00)>>8);    //取TEMP的高八位.
    data1=addL;
    data2=addH;
    wcomd(0x46); //显示地址
    wdata(data1);
    wdata(data2);
    wcomd(0x42);
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<0x7e)) // 一行5*8
        {
            q=text[i]+0x80;
            wdata(q);
            i++;
        }
    }
}
```

```
        else
            i++;
    }

}

void time_disp_8x16(uchar x,uchar y,uchar z,uchar z1,uchar d,uchar d1,uchar s,uchar
s1,uchar s10)
{
    uchar k;
    uint addL, addH;
    for(k=0;k<16;k++)
    {
        temp=0x0000+(k+y)*40+x;
        addL=temp&0x00ff;           //取TEMP的低八位,
        addH=((temp&0xff00)>>8);    //取TEMP的高八位.
        data1=addL;
        data2=addH;
        wcomd(0x46); //显示地址
        wdata(data1);
        wdata(data2);
        wcomd(0x42);
        wdata(num_8x16[z][k]);
        wdata(num_8x16[z1][k]);
        wdata(num_8x16[10][k]);
        wdata(num_8x16[d][k]);
        wdata(num_8x16[d1][k]);
        wdata(num_8x16[10][k]);
        wdata(num_8x16[s][k]);
        wdata(num_8x16[s1][k]);
        wdata(num_8x16[10][k]);
        wdata(num_8x16[s10][k]);
        wdata(num_8x16[0][k]);
    }
}

void time_disp_24x48_h(uchar x,uchar y,uchar z,uchar z1,uchar d)
{
    uchar k,j;
    uint addL, addH;

    for(k=0;k<48;k++)
```

```

{
temp=0x0200+(k+y)*40+x;
addL=temp&0x00ff;           //取TEMP的低八位,
addH=((temp&0xff00)>>8);     //取TEMP的高八位.
data1=addL;
data2=addH;
wcomd(0x46); //显示地址
wdata(data1);
wdata(data2);
wcomd(0x42);
for(j=0;j<3;j++) {wdata(num_24x48[z][k+k*2+j]);}
for(j=0;j<3;j++) {wdata(num_24x48[z1][k+k*2+j]);}
for(j=0;j<3;j++) {wdata(num_24x48[10][k+k*2+j]);}
for(j=0;j<3;j++) {wdata(num_24x48[d][k+k*2+j]);}
for(j=0;j<3;j++) {wdata(num_24x48[0][k+k*2+j]);}
}
}

```

## 第五节、主调用函数

```

////////////////////////////////////
void main(void)
{
RES=0; delay(20);RES=1;delay(2);
initial();

ini_int1();//开中断
for(m1=0;m1<1;m1++)
{

//~~~~~程序开始

disp_all(0x00,0x00);
ds_32x32(1,12,90,hui_32x32);
ds_32x32(1,16,90,jing_32x32);
ds_32x32(1,20,90,ke_32x32);
ds_32x32(1,24,90,ji_32x32);

ds_16x24(1,13,130,num_16x24_3);
ds_16x24(1,15,130,num_16x24_2);
ds_16x24(1,17,130,num_16x24_0);
ds_16x24(1,19,130,num_16x24_x);
ds_16x24(1,21,130,num_16x24_2);

```



```
ds_16x24(1,23,130,num_16x24_4);
ds_16x24(1,25,130,num_16x24_0);

ds_24x24(0,3,16,hui2_24x24);
ds_24x24(0,7,16,jiu_24x24);
ds_24x24(1,11,16,jiao_24x24);
ds_24x24(1,15,16,dian_24x24);

ds_24x24(0,23,16,jing2_24x24);
ds_24x24(0,27,16,ping_24x24);
ds_24x24(1,31,16,zi_24x24);
ds_24x24(1,35,16,sang_24x24);

Wr_reg(0,0,40,2,0xff);//画上框
Wr_reg(0,238,40,2,0xff);//画下框
Wr_reg(0,2,1,236,0xC0);//左框
Wr_reg(39,2,1,236,0x03);//右框
delay(80);
//~~~~~2, 汉字
//~~~~~2, 字符
initial();
disp_all(0x00,0x00);
Wr_reg(0,0,40,240,0xff);//画上框
ds_32x32(0,12,90,hui_32x32);
ds_32x32(0,16,90,jing_32x32);
ds_32x32(0,20,90,ke_32x32);
ds_32x32(0,24,90,ji_32x32);

ds_16x24(0,13,130,num_16x24_3);
ds_16x24(0,15,130,num_16x24_2);
ds_16x24(0,17,130,num_16x24_0);
ds_16x24(0,19,130,num_16x24_x);
ds_16x24(0,21,130,num_16x24_2);
ds_16x24(0,23,130,num_16x24_4);
ds_16x24(0,25,130,num_16x24_0);

ds_24x24(0,3,16,hui2_24x24);
ds_24x24(0,7,16,jiu_24x24);
ds_24x24(0,11,16,jiao_24x24);
ds_24x24(0,15,16,dian_24x24);
```

```

ds_24x24(0,23,16,jing2_24x24);
ds_24x24(0,27,16,ping_24x24);
ds_24x24(0,31,16,zi_24x24);
ds_24x24(0,35,16,sang_24x24);

delay(80);
//~~~~~2, 汉字反白
disp_all(0xff,0x00);delay(50);
//~~~~~2, 扫横
disp_all(0xff,0x00);delay(50);
//~~~~~2, 扫横
disp_all(0x55,0x55);delay(50);
//~~~~~2, 扫竖
disp_all(0xaa,0xaa);delay(50);
//~~~~~2, 扫竖
disp_all(0xaa,0x55);delay(50);
//~~~~~2, 扫点
disp_all(0x55,0xaa);delay(50);
//~~~~~2, 扫点
//disp_bmp(xshm); delay(200);
disp_all(0xff,0xff);delay(50);
//~~~~~2, 全黑
}

```

```

//~~~~~@~~~~&~~~~@~~~~//老化测试

```

```

initial();
disp_all(0x00,0x00);

ds_32x32(1,12,90,hui_32x32);
ds_32x32(1,16,90,jing_32x32);
ds_32x32(1,20,90,ke_32x32);
ds_32x32(1,24,90,ji_32x32);

ds_16x24(1,13,130,num_16x24_3);
ds_16x24(1,15,130,num_16x24_2);
ds_16x24(1,17,130,num_16x24_0);
ds_16x24(1,19,130,num_16x24_x);
ds_16x24(1,21,130,num_16x24_2);
ds_16x24(1,23,130,num_16x24_4);
ds_16x24(1,25,130,num_16x24_0);

```

```
ds_32x32(0,1,20,hui_32x32);
ds_32x32(1,5,20,jing_32x32);
ds_32x32(1,9,20,ke_32x32);
ds_32x32(1,13,20,ji_32x32);

ds_32x32(1,20,160,hui_32x32);
ds_32x32(0,24,160,jing_32x32);
ds_32x32(1,28,160,ke_32x32);
ds_32x32(1,32,160,ji_32x32);

ds_32x32(1,0,160,hui_32x32);
ds_32x32(1,4,160,jing_32x32);
ds_32x32(0,8,160,ke_32x32);
ds_32x32(1,12,160,ji_32x32);

ds_24x24(0,3,200,hui2_24x24);
ds_24x24(0,7,200,jiu_24x24);
ds_24x24(0,11,200,jiao_24x24);
ds_24x24(0,15,200,dian_24x24);

ds_24x24(0,23,200,jing2_24x24);
ds_24x24(0,27,200,ping_24x24);
ds_24x24(0,31,200,zi_24x24);
ds_24x24(0,35,200,sang_24x24);

// Wr_reg(16,0,25,4,0xff); //画上框
// Wr_reg(31,64,8,64,0xff); //
```

## 第六节、 动态显示函数

```
for(z=0; z<10; z++)
{
    for(z1=0; z1<10; z1++)
    {
        for(d=0; d<6; d++)
        {
            time_disp_24x48_h(30,5,z,z1,d);
            for(d1=0; d1<10; d1++)
            {
                for(s=0; s<6; s++)
                {
```



```
uchar code num_16x24_0[]={
/*-- 文字: 0 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=12x24 --*/
/*-- 宽度不是8的倍数, 现调整为: 宽度x高度=16x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x80,0x39,0xC0,0x39,0xC0,
0x70,0xE0,0x70,0xE0,0x70,0xE0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xF0,0xE0,0x70,0xE0,
0x70,0xE0,0x70,0xE0,0x39,0xC0,0x39,0xC0,0x1F,0x80,0x00,0x00,0x00,0x00,0x00,0x00,
};
```

```
uchar code num_16x24_x[]={
/*-- 文字: x --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=12x24 --*/
/*-- 宽度不是8的倍数, 现调整为: 宽度x高度=16x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0xFF,0xE0,0x39,0xC0,0x39,0x80,0x1F,0x00,0x0F,0x00,0x07,0x00,
0x0F,0x00,0x1F,0x80,0x19,0xC0,0x31,0xC0,0xFB,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,
};
```

```
uchar code num_16x24_1[]={
/*-- 文字: 1 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=12x24 --*/
/*-- 宽度不是8的倍数, 现调整为: 宽度x高度=16x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x07,0x00,0x3F,0x00,
0x07,0x00,0x07,0x00,0x07,0x00,0x07,0x00,0x07,0x00,0x07,0x00,0x07,0x00,0x07,0x00,
0x07,0x00,0x07,0x00,0x07,0x00,0x0F,0x00,0x3F,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
};
```

```
uchar code num_16x24_3[]={
/*-- 文字: 3 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=12x24 --*/
/*-- 宽度不是8的倍数, 现调整为: 宽度x高度=16x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x80,0x33,0xC0,0x71,0xC0,
0x71,0xE0,0x71,0xE0,0x01,0xC0,0x03,0x80,0x0F,0x00,0x01,0xC0,0x00,0xE0,0x00,0xE0,
0x70,0xE0,0x70,0xE0,0x70,0xE0,0x79,0xC0,0x1F,0x80,0x00,0x00,0x00,0x00,0x00,0x00,
};
```

```
uchar code hui2_24x24[]={
/*-- 文字: 汇 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x00,0x00,0x1E,0x60,0x0C,0x0E,0x7F,0xFE,0x0E,
0xF0,0x00,0x01,0xF0,0x00,0xE1,0xF0,0x00,0x79,0xF0,0x00,0x3B,0x70,0x00,0x3B,0x70,
0x00,0x07,0x70,0x00,0x06,0x70,0x00,0x06,0x70,0x00,0x0E,0x70,0x00,0x0C,0x70,0x00,
0x1C,0x70,0x00,0x7C,0x70,0x00,0x1C,0x70,0x00,0x18,0x70,0x00,0x38,0x70,0x1E,0x3C,
```

```
0x7F,0xFE,0x1C,0x00,0x00,0x00,0x00,0x00,
};
```

```
uchar code jiu_24x24[]={
/*-- 文字: 聚 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x04,0x00,0x00,0x1E,0x00,0x7F,0xFB,0x1C,0x0E,0x3F,0xFE,0x0F,
0xF0,0x38,0x0E,0x37,0xB0,0x0E,0x31,0xE0,0x0F,0xF0,0xF8,0x0E,0x3F,0x9C,0x0F,0xF7,
0x1C,0x3C,0x3C,0xC0,0x30,0x0F,0xE0,0x03,0xFC,0x00,0x07,0x9E,0x38,0x03,0xDE,0x78,
0x07,0x1F,0xE0,0x0E,0xFF,0x80,0x39,0xFD,0xC0,0x33,0x9C,0xF8,0x07,0x1C,0x3E,0x1C,
0x1C,0x00,0x10,0x1C,0x00,0x00,0x00,0x00,
};
```

```
uchar code jiao_24x24[]={
/*-- 文字: 焦 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x18,0x00,0x03,0x9C,0x00,0x03,0xCE,0x00,0x03,0x8E,0x00,0x07,
0x06,0x1C,0x07,0xFF,0xFE,0x0E,0x0E,0x00,0x1E,0x0E,0x30,0x1E,0x0E,0x78,0x37,0xFF,
0xF8,0x76,0x0E,0x00,0xE6,0x0E,0x38,0x07,0xFF,0xF8,0x06,0x0E,0x00,0x06,0x0E,0x0C,
0x07,0xFF,0xFE,0x06,0x00,0x02,0x0C,0x06,0x30,0x0C,0xE7,0x18,0x1C,0xE3,0x9C,0x3C,
0x73,0x9C,0x78,0x60,0x0C,0x00,0x00,0x00,
};
```

```
uchar code dian_24x24[]={
/*-- 文字: 点 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x30,0x00,0x00,0x3C,0x00,0x00,0x38,0x00,0x00,0x38,0x38,0x00,
0x3F,0xFC,0x00,0x38,0x0C,0x00,0x38,0x00,0x00,0x38,0x00,0x07,0x38,0xF0,0x07,0xFF,
0xF0,0x07,0x00,0x60,0x07,0x00,0x60,0x07,0x00,0x60,0x07,0x00,0x60,0x07,0xFF,0xE0,
0x07,0x00,0x60,0x04,0x00,0x00,0x0C,0xC6,0x30,0x0C,0xE7,0x18,0x1C,0x73,0x9C,0x38,
0x73,0x9E,0x38,0x63,0x0C,0x00,0x00,0x00,
};
```

```
uchar code jing2_24x24[]={
/*-- 文字: 精 --*/
/*-- 宋体18; 此字体下对应的点阵为: 宽x高=24x24 --*/
0x00,0x00,0x00,0x0C,0x01,0x80,0x07,0x01,0xC0,0x06,0x01,0xC0,0x66,0xFF,0xFC,0x37,
0xE1,0xC6,0x3F,0xC1,0xC0,0x3F,0x9F,0xFC,0x07,0x01,0xCC,0x07,0xE1,0xCC,0x7F,0xFF,
0xFE,0x0E,0x00,0x02,0x0F,0x9C,0x18,0x1F,0xDF,0xFC,0x1F,0xDC,0x38,0x3E,0xDF,0xF8,
0x3E,0x1C,0x38,0x76,0x1C,0x38,0x66,0x1F,0xF8,0xC6,0x1C,0x38,0x06,0x1C,0x38,0x06,
0x1D,0xF8,0x06,0x1C,0x78,0x00,0x00,0x00,
};
```

```
uchar code ping_24x24[]={
```

```

/*-- 文字： 品 --*/
/*-- 宋体18； 此字体下对应的点阵为： 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x01,0xC0,0x03,0xFF,0xC0,0x03,0x81,0xC0,0x03,
0x81,0xC0,0x03,0x81,0xC0,0x03,0x81,0xC0,0x03,0x81,0xC0,0x03,0xFF,0xC0,0x03,0x81,
0xC0,0x03,0x00,0x00,0x30,0x4C,0x18,0x3F,0xF7,0xFC,0x38,0xE6,0x1C,0x38,0xE6,0x1C,
0x38,0xE6,0x1C,0x38,0xE6,0x1C,0x38,0xE6,0x1C,0x38,0xE6,0x1C,0x3F,0xE7,0xFC,0x38,
0xE6,0x1C,0x38,0x0C,0x00,0x00,0x00,0x00,
};

```

```

uchar code zi_24x24[]={
/*-- 文字： 至 --*/
/*-- 宋体18； 此字体下对应的点阵为： 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x7F,0xFF,0xFC,0x00,0x70,0x06,0x00,
0xE0,0x00,0x00,0xE6,0x00,0x01,0xC3,0x80,0x03,0x81,0xE0,0x06,0x00,0xF0,0x1F,0xFF,
0xF8,0x0F,0x38,0x38,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x70,0x1F,0xFF,0xF8,
0x00,0x38,0x18,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x0C,0x00,0x38,0x1E,0x7F,
0xFF,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,
};

```

```

uchar code sang_24x24[]={
/*-- 文字： 上 --*/
/*-- 宋体18； 此字体下对应的点阵为： 宽x高=24x24 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x30,0x00,0x00,
0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x30,0x00,0x30,0x78,0x00,0x3F,
0xD8,0x00,0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x00,
0x00,0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x00,0x00,0x30,0x0C,0x00,0x30,0x1E,0x7F,
0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,
};

```

```

uchar code hui_32x32[]={
/*-- 文字： 绘 --*/
/*-- 宋体24； 此字体下对应的点阵为： 宽x高=32x32 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x38,0x00,0x03,0xC0,0x3E,0x00,
0x03,0xC0,0x3C,0x00,0x07,0x80,0x7C,0x00,0x07,0x00,0x7E,0x00,0x07,0x00,0x77,0x00,
0x0E,0x38,0xE3,0x80,0x0C,0x3D,0xE3,0x80,0x1C,0x79,0xC1,0xE0,0x38,0x73,0x80,0xF0,
0x70,0xE7,0x00,0x7C,0x7F,0xEE,0x03,0xFE,0x31,0xDF,0xFF,0xF8,0x03,0xB8,0x00,0x60,
0x07,0x30,0x00,0x00,0x07,0x00,0x00,0x00,0x0E,0x00,0x00,0x30,0x1C,0x78,0x00,0x78,
0x3F,0xCF,0xFF,0xEC,0x3E,0x00,0x7C,0x00,0x10,0x00,0x78,0x00,0x00,0x00,0xF3,0x00,
0x00,0x38,0xE3,0x80,0x03,0xF1,0xC1,0xC0,0x3F,0x03,0x80,0xE0,0x7C,0x07,0x80,0xF0,
0x30,0x0F,0xFF,0xF0,0x00,0x07,0xC0,0x70,0x00,0x06,0x00,0x20,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,
};

```

```

uchar code jing_32x32[]={

```





```

0x00,0x00,0x00,0x18,0x24,0x42,0x42,0x42,0x42,0x42,0x42,0x24,0x18,0x00,0x00,
/*-- 文字: 1 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x10,0x70,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x7C,0x00,0x00,
/*-- 文字: 2 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x3C,0x42,0x42,0x42,0x04,0x04,0x08,0x10,0x20,0x42,0x7E,0x00,0x00,
/*-- 文字: 3 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x3C,0x42,0x42,0x04,0x18,0x04,0x02,0x02,0x42,0x44,0x38,0x00,0x00,
/*-- 文字: 4 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x04,0x0C,0x14,0x24,0x24,0x44,0x44,0x7E,0x04,0x04,0x1E,0x00,0x00,
/*-- 文字: 5 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x7E,0x40,0x40,0x40,0x58,0x64,0x02,0x02,0x42,0x44,0x38,0x00,0x00,
/*-- 文字: 6 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x1C,0x24,0x40,0x40,0x58,0x64,0x42,0x42,0x42,0x24,0x18,0x00,0x00,
/*-- 文字: 7 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x7E,0x44,0x44,0x08,0x08,0x10,0x10,0x10,0x10,0x10,0x00,0x00,
/*-- 文字: 8 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x3C,0x42,0x42,0x24,0x18,0x24,0x42,0x42,0x42,0x3C,0x00,0x00,
/*-- 文字: 9 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x18,0x24,0x42,0x42,0x42,0x26,0x1A,0x02,0x02,0x24,0x38,0x00,0x00,
/*-- 文字: : --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=8x16 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00,0x00,0x18,0x18,0x00,0x00,
};

```

```

uchar code num_24x48[11][144]={
/*-- 文字: 0 --*/
/*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7E,0x00,0x01,0xE7,0x80,0x03,0xC3,
0xC0,0x07,0x81,0xE0,0x0F,0x80,0xF0,0x0F,0x00,0xF0,0x1F,0x00,0xF8,0x1E,0x00,0xF8,
0x3E,0x00,0x78,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,
0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,
0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x7C,0x3E,0x00,0x78,
0x1E,0x00,0xF8,0x1F,0x00,0xF8,0x0F,0x00,0xF0,0x0F,0x81,0xF0,0x07,0x81,0xE0,0x03,
0xC3,0xC0,0x01,0xE7,0x80,0x00,0x7E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
};

```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 1 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x00, 0x00, 0x1C, 0x00, 0x00, 0x7C,  
0x00, 0x07, 0xFC, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00,  
0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00,  
0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C,  
0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00,  
0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C, 0x00, 0x00,  
0x3E, 0x00, 0x00, 0x7F, 0x00, 0x07, 0xFF, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 2 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x03, 0xC7, 0xC0, 0x07, 0x01,  
0xE0, 0x0E, 0x00, 0xF0, 0x1E, 0x00, 0xF8, 0x1E, 0x00, 0xF8, 0x3E, 0x00, 0x78, 0x3E, 0x00, 0x78,  
0x3F, 0x00, 0x78, 0x3F, 0x00, 0x78, 0x1F, 0x00, 0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xF0, 0x00,  
0x01, 0xF0, 0x00, 0x03, 0xE0, 0x00, 0x03, 0xC0, 0x00, 0x07, 0x80, 0x00, 0x0F, 0x00, 0x00, 0x1E,  
0x00, 0x00, 0x3C, 0x00, 0x00, 0x78, 0x00, 0x00, 0xF0, 0x00, 0x01, 0xE0, 0x00, 0x03, 0xC0, 0x00,  
0x07, 0x80, 0x1C, 0x07, 0x00, 0x1C, 0x0E, 0x00, 0x38, 0x1C, 0x00, 0x38, 0x3C, 0x00, 0x78, 0x3F,  
0xFF, 0xF8, 0x3F, 0xFF, 0xF8, 0x3F, 0xFF, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 3 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x00, 0x07, 0x87, 0x80, 0x0F, 0x03,  
0xC0, 0x1E, 0x01, 0xE0, 0x1E, 0x01, 0xF0, 0x1E, 0x01, 0xF0, 0x1F, 0x00, 0xF0, 0x1F, 0x00, 0xF0,  
0x1E, 0x00, 0xF0, 0x00, 0x00, 0xF0, 0x00, 0x01, 0xF0, 0x00, 0x01, 0xF0, 0x00, 0x03, 0xE0, 0x00,  
0x03, 0xC0, 0x00, 0x0F, 0x00, 0x00, 0xFE, 0x00, 0x00, 0x07, 0x80, 0x00, 0x01, 0xE0, 0x00, 0x00,  
0xF0, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x78, 0x00, 0x00, 0x7C, 0x1E, 0x00, 0x7C,  
0x3F, 0x00, 0x7C, 0x3F, 0x00, 0x7C, 0x3F, 0x00, 0x78, 0x3E, 0x00, 0xF8, 0x1E, 0x00, 0xF0, 0x0F,  
0x01, 0xE0, 0x07, 0x87, 0xC0, 0x01, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 4 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xC0, 0x00, 0x03, 0xC0, 0x00, 0x07,  
0xC0, 0x00, 0x0F, 0xC0, 0x00, 0x0F, 0xC0, 0x00, 0x1F, 0xC0, 0x00, 0x3F, 0xC0, 0x00, 0x3F, 0xC0,  
0x00, 0x77, 0xC0, 0x00, 0x77, 0xC0, 0x00, 0xE7, 0xC0, 0x01, 0xC7, 0xC0, 0x01, 0xC7, 0xC0, 0x03,  
0x87, 0xC0, 0x07, 0x07, 0xC0, 0x07, 0x07, 0xC0, 0x0E, 0x07, 0xC0, 0x1E, 0x07, 0xC0, 0x1C, 0x07,  
0xC0, 0x38, 0x07, 0xC0, 0x38, 0x07, 0xC0, 0x7F, 0xFF, 0xFE, 0x7F, 0xFF, 0xFE, 0x00, 0x07, 0xC0,  
0x00, 0x07, 0xC0, 0x00, 0x07, 0xC0, 0x00, 0x07, 0xC0, 0x00, 0x07, 0xC0, 0x00, 0x07, 0xC0, 0x00,  
0x07, 0xC0, 0x00, 0x07, 0xE0, 0x00, 0x7F, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 5 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xFF, 0xF8, 0x0F, 0xFF, 0xF8, 0x0F, 0xFF,  
0xF8, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x00,  
0x0E, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x0E, 0x7F, 0x00, 0x0D, 0xFF, 0xC0, 0x0F,  
0xC3, 0xE0, 0x1F, 0x01, 0xF0, 0x1E, 0x00, 0xF8, 0x1E, 0x00, 0xF8, 0x00, 0x00, 0x78, 0x00, 0x00,  
0x7C, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x7C, 0x1E, 0x00, 0x7C, 0x3F, 0x00, 0x7C,  
0x3F, 0x00, 0x78, 0x3F, 0x00, 0x78, 0x3E, 0x00, 0xF8, 0x1E, 0x00, 0xF0, 0x1E, 0x01, 0xF0, 0x0E,  
0x01, 0xE0, 0x07, 0x87, 0xC0, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 6 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xC0, 0x00, 0xF1, 0xE0, 0x03, 0xC1,  
0xF0, 0x07, 0x81, 0xF8, 0x07, 0x01, 0xF8, 0x0F, 0x00, 0xF0, 0x1F, 0x00, 0x00, 0x1E, 0x00, 0x00,  
0x1E, 0x00, 0x00, 0x3E, 0x00, 0x00, 0x3E, 0x00, 0x00, 0x3E, 0x00, 0x00, 0x3E, 0x3F, 0x80, 0x3E,  
0xFF, 0xE0, 0x3F, 0xE3, 0xF0, 0x3F, 0x80, 0xF8, 0x3F, 0x00, 0xF8, 0x3F, 0x00, 0x7C, 0x3E, 0x00,  
0x7C, 0x3E, 0x00, 0x7C, 0x3E, 0x00, 0x3C, 0x3E, 0x00, 0x3C, 0x3E, 0x00, 0x3C, 0x3E, 0x00, 0x3C,  
0x3E, 0x00, 0x7C, 0x1E, 0x00, 0x7C, 0x1F, 0x00, 0x78, 0x0F, 0x00, 0x78, 0x0F, 0x80, 0xF0, 0x07,  
0xC0, 0xE0, 0x03, 0xE3, 0xC0, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 7 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0xFF, 0xFC, 0x1F, 0xFF, 0xFC, 0x1F, 0xFF,  
0xF8, 0x1F, 0x00, 0x38, 0x1C, 0x00, 0x70, 0x1C, 0x00, 0xE0, 0x38, 0x00, 0xE0, 0x38, 0x01, 0xC0,  
0x00, 0x01, 0xC0, 0x00, 0x03, 0x80, 0x00, 0x03, 0x80, 0x00, 0x07, 0x80, 0x00, 0x07, 0x00, 0x00,  
0x0F, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x3C, 0x00, 0x00, 0x3C,  
0x00, 0x00, 0x3C, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x78, 0x00, 0x00, 0x78, 0x00, 0x00, 0xF8, 0x00,  
0x00, 0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00,  
0xF8, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
/\*-- 文字: 8 --\*/  
/\*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --\*/  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x07, 0xC3, 0xC0, 0x0F, 0x00,  
0xE0, 0x1E, 0x00, 0xF0, 0x1E, 0x00, 0x78, 0x3C, 0x00, 0x78, 0x3C, 0x00, 0x78, 0x3C, 0x00, 0x7C,  
0x3E, 0x00, 0x78, 0x3E, 0x00, 0x78, 0x1F, 0x00, 0x78, 0x1F, 0x80, 0xF0, 0x0F, 0xE1, 0xE0, 0x07,  
0xFB, 0xC0, 0x01, 0xFF, 0x80, 0x01, 0xFF, 0x80, 0x07, 0xBF, 0xC0, 0x0F, 0x0F, 0xE0, 0x1E, 0x03,  
0xF0, 0x3E, 0x01, 0xF8, 0x3C, 0x00, 0xF8, 0x3C, 0x00, 0x7C, 0x7C, 0x00, 0x7C, 0x78, 0x00, 0x3C,  
0x78, 0x00, 0x3C, 0x7C, 0x00, 0x3C, 0x3C, 0x00, 0x78, 0x3C, 0x00, 0x78, 0x1E, 0x00, 0x70, 0x0F,  
0x00, 0xE0, 0x07, 0xC3, 0xC0, 0x01, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
/*-- 文字: 9 --*/
/*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0xFE,0x00,0x07,0xC7,0x80,0x0F,0x01,
0xE0,0x1E,0x01,0xE0,0x1E,0x00,0xF0,0x3E,0x00,0xF8,0x3C,0x00,0x78,0x3C,0x00,0x78,
0x7C,0x00,0x7C,0x7C,0x00,0x7C,0x7C,0x00,0x7C,0x7C,0x00,0x7C,0x7C,0x00,0x7C,0x3C,
0x00,0xFC,0x3E,0x00,0xFC,0x3E,0x01,0xFC,0x1F,0x03,0xFC,0x1F,0x8F,0xFC,0x0F,0xFF,
0x7C,0x03,0xFC,0x7C,0x00,0x00,0x7C,0x00,0x00,0xF8,0x00,0x00,0xF8,0x00,0x00,0xF8,
0x00,0x00,0xF0,0x00,0x01,0xF0,0x0F,0x01,0xE0,0x1F,0x01,0xE0,0x1F,0x03,0xC0,0x1F,
0x07,0x80,0x0F,0x9F,0x00,0x03,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
/*-- 文字: : --*/
/*-- 宋体36; 此字体下对应的点阵为: 宽x高=24x48 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0x7E,0x00,0x00,0xFE,0x00,0x00,
0xFE,0x00,0x00,0x7E,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0x7E,0x00,0x00,0xFE,0x00,0x00,
0xFE,0x00,0x00,0x7E,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
};

```

## 第十章、版本信息

### 更新说明

更新日期	更新内容说明	版本
2023/11/15	第一版	V1.0