

一、概 述

16032ZA 中文图形液晶模块使用 ST7920 作为控制驱动器，它的软件特性主要由 ST7920 控制驱动器决定。ST7920 同时作为控制器和驱动器，它可提供 3 路 com 输出和 64 路 seg 输出。在驱动器 ST7921 的配合下，最多可以驱动 256×32 点阵液晶。

16032ZA 中文图形液晶模块硬件特性如下：

- 提供 8 位，4 位并行接口及串行接口可选
- 并行接口适配 M6800 时序
- 自动电源启动复位功能
- 内部自建振荡源
- 64×16 位字符显示 RAM (DDRAM 最多 16 字符×4 行，LCD 显示范围 16×2 行)
- 2M 位中文字型 ROM (CGROM)，总共提供 8192 个中文字型 (16×16 点阵)
- 16K 位半宽字型 ROM (HCGROM)，总共提供 126 个西文字型 (16×8 点阵)
- 64×16 位字符产生 RAM (CGRAM)
- 15×16 位总共 240 点的 ICON RAM (ICONRAM)

16032ZA 中文图形液晶模块软件特性如下：

- 文字与图形混合显示功能
- 画面清除功能
- 光标归位功能
- 显示开/关功能
- 光标显示/隐藏功能
- 显示字体闪烁功能
- 光标移位功能
- 显示移位功能
- 垂直画面旋转功能
- 反白显示功能
- 休眠模式

中文字库选择：

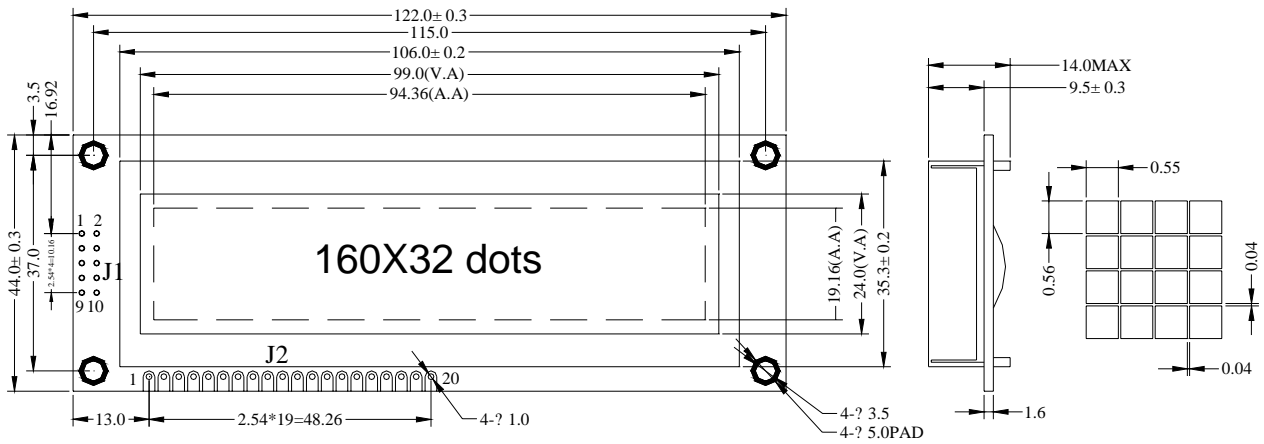
ST7920-0A 内建 BIG-5 码繁体中文字型库

ST7920-0B 内建 GB 码简体中文字型库

用户在选用之前务必注明。

二、外形结构

1. 外形图



2. 主要外形尺寸

项 目	标 准 尺 寸	单 位
模 块 体 积	122.0L×44.0W×14.0 (max) H	mm
视 域	99.0×24.0	mm
行 列 点 阵 数	160×32	dots
点 距 离	0.55×0.56	mm
点 大 小	0.59×0.60	mm

三、模块硬件说明

16032ZA 配有两套接口，一为串口 J1，一为并口 J2。其中 J2 中的某些线也可以做串口用。用户可以通过设置 PSB 脚来选择使用串口还是并口。也可以将 PSB 脚悬空，通过液晶模块背面的短接点来选择串口或者并口。

1. 模块接口

接口(J2)	名称	电 平	功能描述	
			并 口	串 口
1	GND	0V	电源地	
2	VCC	5.0V	模块电源输入（未注明为 5V）	
3	V0	-	对比度调节端	
4	RS(CS)	H/L	寄存器选择端： H: 数据； L: 指令	片选，高有效
5	R/W(SID)	H/L	读/写选择端： H: 读操作； L: 写操作	串行数据线
6	E(SCLK)	H/L	使能信号	串行时钟输入

7-10	DB0-DB3	H/L	数据总线低四位,用 4 位并口时空接	空接
11-14	DB4-DB7	H/L	数据总线高四位	空接
15	PSB	H/L	并口/串口选择: H 并口; L 串口	
16	RST	H/L	复位信号, 低有效	
17	Vout	-	液晶驱动电压	
18	LEDA	5.0V	背光正 (或名 BLA)	
19	LEDK	0V	背光负 (或名 BLK)	
20	NC	-	空脚	

2. 电气特性

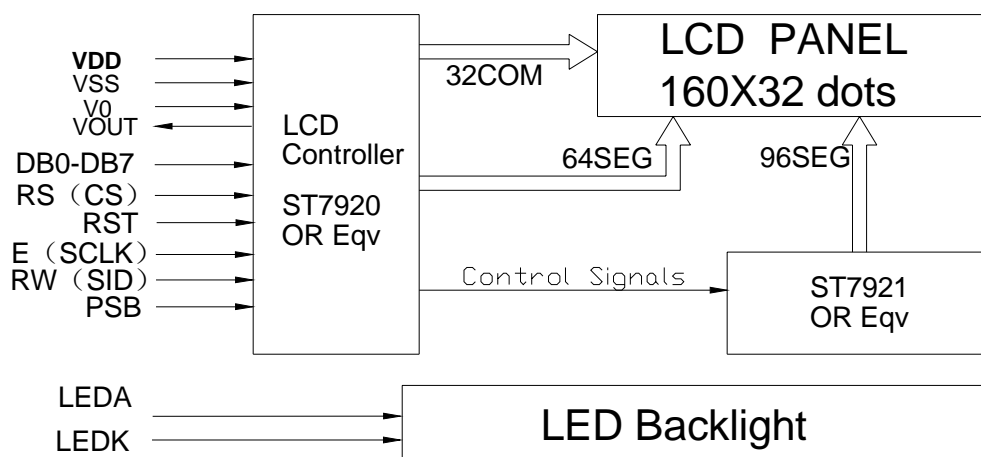
- 1) 输入高电平 (V_{ih}): $0.7V_{dd} \sim V_{dd}$
- 2) 输入低电平 (V_{il}): $0.6V_{max}$
- 3) 输出高电平 (V_{oh}): $0.8V_{dd} \sim V_{dd}$
- 4) 输出低电平 (V_{ol}): $0.4V_{max}$
- 5) 模块工作电流: $3 \sim 5mA$ (不含背光)
- 6) 侧白光工作电流: -
- 7) 底黄绿光工作电流: $568mA$

3. 对比度调节电路

ST7920 内带倍压电路, 生成 2 倍于 V_{ref} 的电压, V_{ref} 是 V_{CC} 经过电阻降压产生的, $V_{ref} \leq V_{CC}$ 。倍压由 V_{out} 脚引出, 它通过模块上的线路供给 V_0 用来驱动 LCD。直接驱动 LCD 的是 V_0 , V_0 电压越高, 对比度越深。

通过调节 V_0 值以改变对比度。用户可以将 V_0 与地之间接一个 $20K$ 的可调电阻, 用来拉低 V_0 电压值。对于对比度已经锁定的液晶模块, 不能再调节对比度。用户强行用上述方法调节对比度只会使对比度变浅。

4. 原理简图



四、ST7920 内置硬件说明

1. 中文字型产生 ROM (CGROM) 及半宽字型 ROM(HCGROM)

ST7920 的字型产生 ROM 通过 8192 个 16×16 点阵的中文字型，以及 126 个 16×8 点阵的西文字符，它用 2 个字节来提供编码选择，将要显示的字符的编码写到 DDRAM 上，硬件将依照编码自动从 CGROM 中选择将要显示的字型显示再屏幕上。

2. 字型产生 RAM(CGRAM)

ST7920 的字型产生 RAM 提供用户自定义字符生成（造字）功能，可提供 4 组 16×16 点阵的空间，用户可以将 CGROM 中没有的字符定义到 CGRAM 中。

3. 显示 RAM(DDRAM)

显示 RAM 提供 64×2 字节的空间，最多可以控制 4 行 16 字的中文字型显示。当写入显示资料 RAM 时，可以分别显示 CGROM，HCGROM 及 CGRAM 的字型。

三种字型的选择：

- 1) 显示半宽字型 将一个字节的编码写入 DDRAM 中，范围是 02~7FH
- 2) 显示 CGRAM 字型 将 2 个字节的编码写入 DDRAM 中，共有 0000H, 0002H, 0004H 及 0006H 四种编码
- 3) 显示中文字型 将 2 字节的编码写入 DDRAM 中，先写高 8 位，后写低 8 位范围是 A140H~D75FH(BIG5), A1A0H~F7FFH(GB)

4. ICON RAM(IRAM)

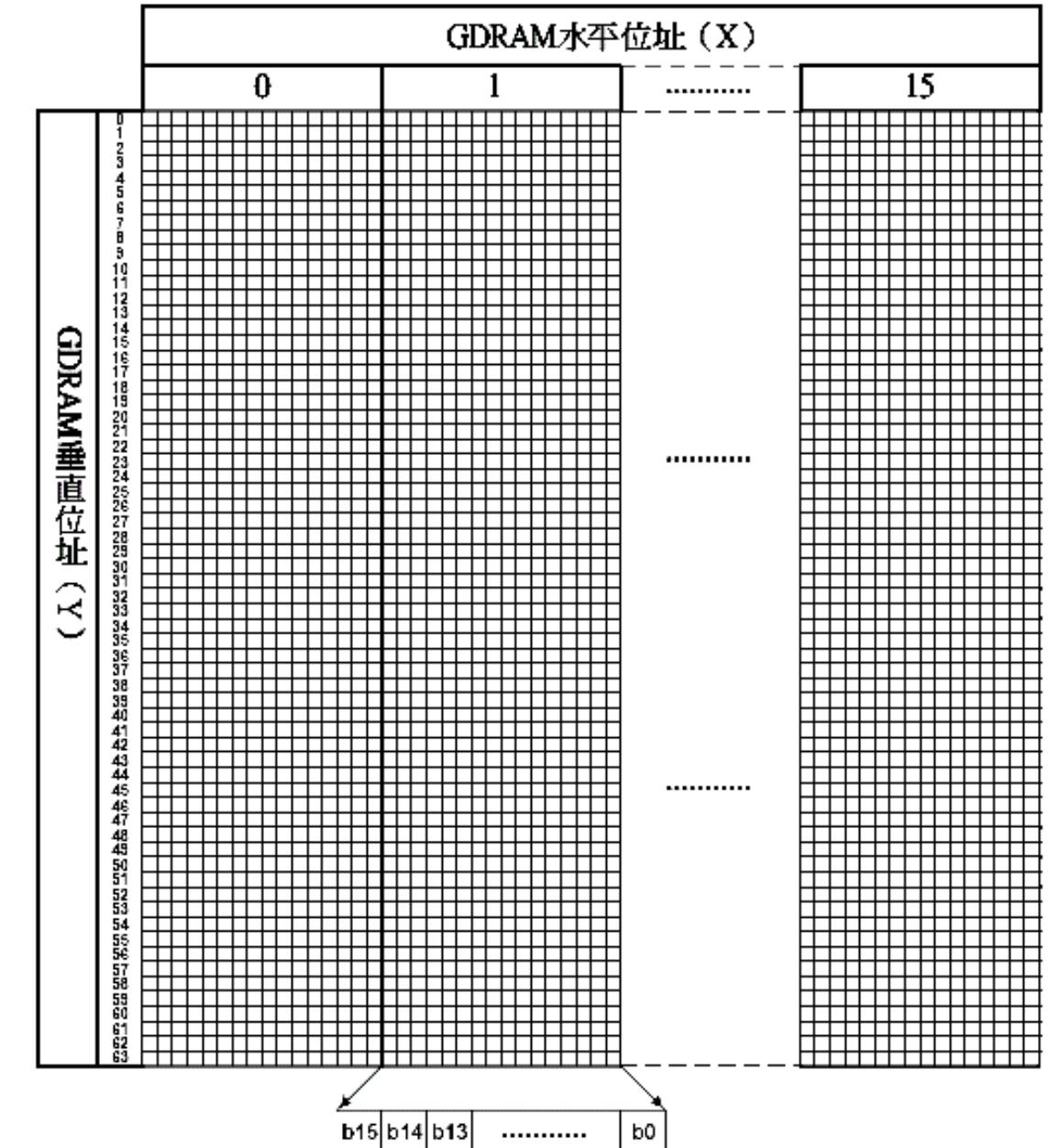
ST7920 提供 240 点的 ICON 显示，它由 15 个 IRAM 单元组成，每个单元有 16 位，每写入一组 IRAM 时，

ICON RAM 位址 在擴充指令集將 SR 設 為“0”.再利用設定 I RAM 位 址指令來設定 AC3...AC0				ICON RAM 資料															
				高位元組								低位元組							
AC3	AC2	AC1	AC0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
0	0	0	1	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31
0	0	1	0	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39	SEG40	SEG41	SEG42	SEG43	SEG44	SEG45	SEG46	SEG47
0	0	1	1	SEG48	SEG49	SEG50	SEG51	SEG52	SEG53	SEG54	SEG55	SEG56	SEG57	SEG58	SEG59	SEG60	SEG61	SEG62	SEG63
0	1	0	0	SEG64	SEG65	SEG66	SEG67	SEG68	SEG69	SEG70	SEG71	SEG72	SEG73	SEG74	SEG75	SEG76	SEG77	SEG78	SEG79
0	1	0	1	SEG80	SEG81	SEG82	SEG83	SEG84	SEG85	SEG86	SEG87	SEG88	SEG89	SEG90	SEG91	SEG92	SEG93	SEG94	SEG95
0	1	1	0	SEG96	SEG97	SEG98	SEG99	SEG100	SEG101	SEG102	SEG103	SEG104	SEG105	SEG106	SEG107	SEG108	SEG109	SEG110	SEG111
0	1	1	1	SEG112	SEG113	SEG114	SEG115	SEG116	SEG117	SEG118	SEG119	SEG120	SEG121	SEG122	SEG123	SEG124	SEG125	SEG126	SEG127
1	0	0	0	SEG128	SEG129	SEG130	SEG131	SEG132	SEG133	SEG134	SEG135	SEG136	SEG137	SEG138	SEG139	SEG140	SEG141	SEG142	SEG143
1	0	0	1	SEG144	SEG145	SEG146	SEG147	SEG148	SEG149	SEG150	SEG151	SEG152	SEG153	SEG154	SEG155	SEG156	SEG157	SEG158	SEG159
1	0	1	0	SEG160	SEG161	SEG162	SEG163	SEG164	SEG165	SEG166	SEG167	SEG168	SEG169	SEG170	SEG171	SEG172	SEG173	SEG174	SEG175
1	0	1	1	SEG176	SEG177	SEG178	SEG179	SEG180	SEG181	SEG182	SEG183	SEG184	SEG185	SEG186	SEG187	SEG188	SEG189	SEG190	SEG191
1	1	0	0	SEG192	SEG193	SEG194	SEG195	SEG196	SEG197	SEG198	SEG199	SEG200	SEG201	SEG202	SEG203	SEG204	SEG205	SEG206	SEG207
1	1	0	1	SEG208	SEG209	SEG210	SEG211	SEG212	SEG213	SEG214	SEG215	SEG216	SEG217	SEG218	SEG219	SEG220	SEG221	SEG222	SEG223
1	1	1	0	SEG224	SEG225	SEG226	SEG227	SEG228	SEG229	SEG230	SEG231	SEG232	SEG233	SEG234	SEG235	SEG236	SEG237	SEG238	SEG239
1	1	1	1	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

需先写入 IRAM 地址，然后连续送入 2 个字节的数据，先高 8 位 (D15~D8)，后低 8 位 (D7~D0)。

5. 绘图 RAM

提供 64×32 个字节的存储空间（由扩充指令设定绘图 RAM 地址），最多可以控制 256×64 点阵的二维绘图缓冲空间，在更改绘图 RAM 是，由扩充指令设置 GDRAM 地址先垂直地址后水平地址（连续 2 个字节的数来定义垂直和水平地址），再 2 个字节的数给绘图 RAM（先高 8 位后低 8 位）。



6. DDRAM 内容，CGRAM 地址以及 CGRAM 内容的对照关系

DDRAM 資料 (字元代碼)				CGRAM 位址					CGRAM 資料 (高位元組)					CGRAM 資料 (低位元組)																				
B15~B4				B	B	B	B	B	B	B	B	B	D	D	D	D	D	D	D	D	D	D												
				3	2	1	0	5	4	3	2	1	0	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0					
												5	4	3	2	1	0																	
0	X	00	X	00	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0				
					0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
					0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	
					0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
					0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0
					0	1	0	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0
					0	1	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0
					0	1	1	1	1	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0
					1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0
					1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
					1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
					1	0	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
					1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
					1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
					1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
					1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	X	01	X	01	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0				
					0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0		
					0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0
					0	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	0	
					0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	
					0	1	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	
					0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	
					0	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0
					1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0
					1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0
					1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0
					1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0
					1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
					1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7. CGRAM 与中文字型的编码只能出现在 **adress counter** 的起始位置（见下表）

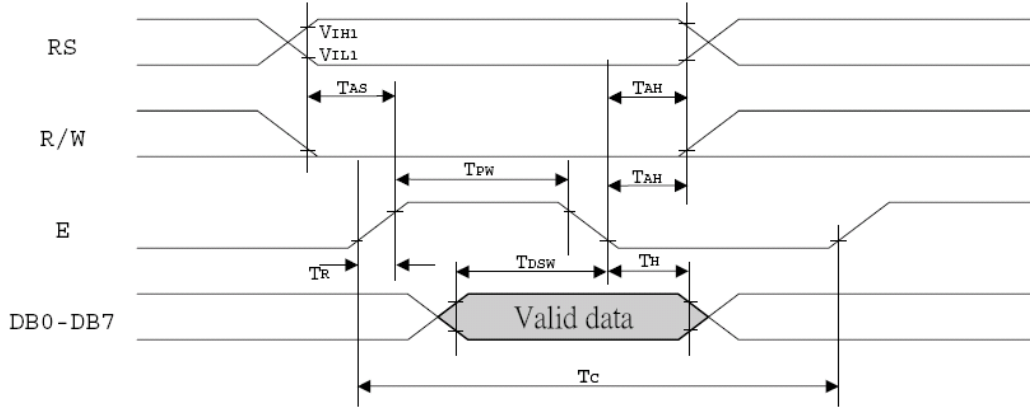
80		81		82		83		84		85		86		87		88		89		8A		8B		8C			
H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L
A	B		1	2	3	4																					
液		晶		模		块			A B		1 2		中 文				(正 确)									
液		晶		模		块		...		A B		1 2		中 文				(错 误)									

8. 16×8 半宽字型表

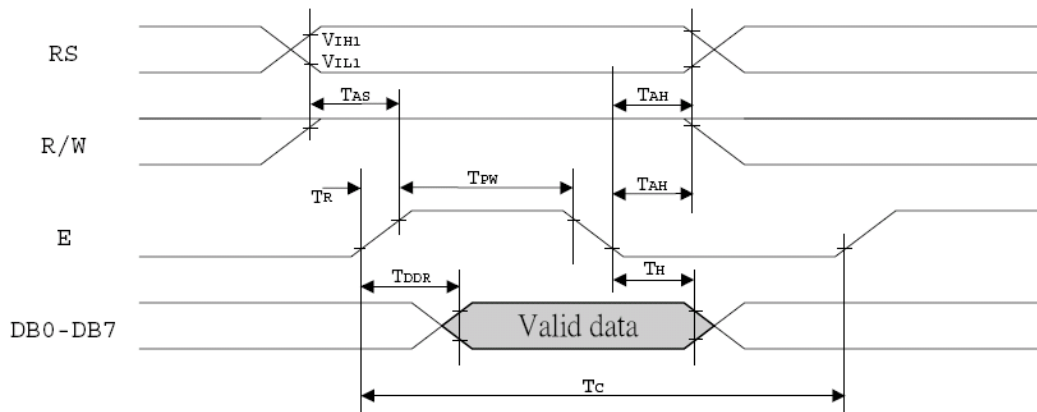
☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
▶	◀	↑	!!	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
	!	"	#	\$	%	&	'	()	*	+	,	-	.	/										
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?										
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O										
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_										
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o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

五、操作时序

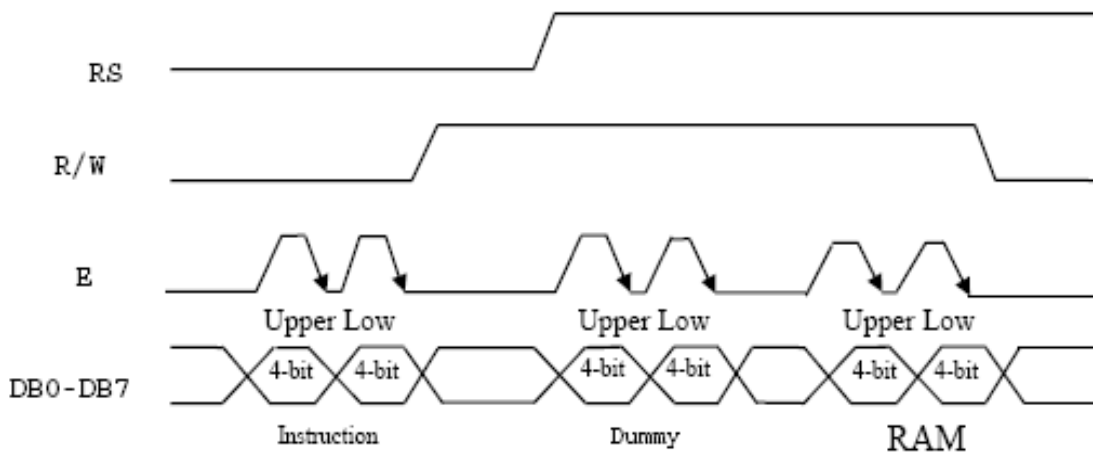
1. 8 位并口写操作时序图



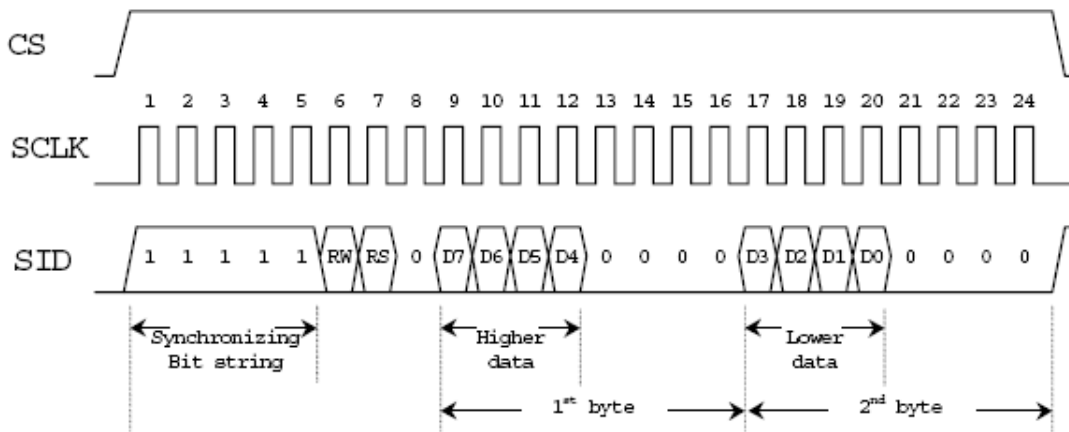
2. 8 位并口读操作时序图



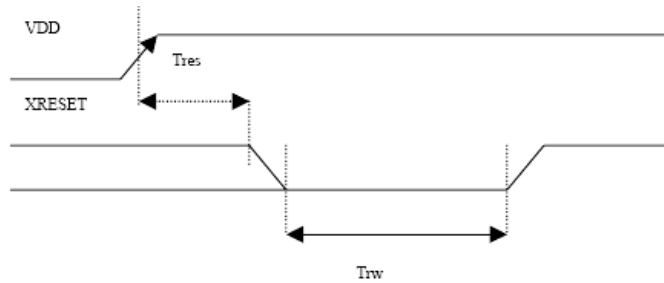
3. 4 位并口时序图



4. 串口时序图



5. 外部复位时序图



XRESET pulse width	Trw	10us
RESET start time	Tres	50ns

六、指令说明

1. 指令表 1 (RE=0,基本指令集)

指令名称	控制信号		控制代码								执行时间
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
清除显示	0	0	0	0	0	0	0	0	0	1	1.6 ms
地址归 0	0	0	0	0	0	0	0	0	1	X	72us
进入设定点	0	0	0	0	0	0	0	1	I/D	S	72us
显示开关设置	0	0	0	0	0	0	1	D	C	B	72us
移位控制	0	0	0	0	0	1	S/C	R/L	X	X	72us
功能设定	0	0	0	0	1	DL	X	0/RE	X	X	72us
设定 CGRAM 地址	0	0	0	1	A5	A4	A3	A2	A1	A0	72us
设定 DDRAM 地址	0	0	1	0	A5	A4	A3	A2	A1	A0	72us
读忙标志和地址	0	1	BF	A6	A5	A4	A3	A2	A1	A0	72us
写显示数据	1	0	显示数据								72us
读显示数据	1	1	显示数据								72us

2. 指令表 2 (RE=1,扩充指令集)

指令名称	控制信号		控制代码								执行时间
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
待命模式	0	0	0	0	0	0	0	0	0	1	72 μ s
滚动地址或 RAM 地址选择	0	0	0	0	0	0	0	0	1	SR	72 μ s
反白显示	0	0	0	0	0	0	0	1	R1	R0	72 μ s
睡眠模式	0	0	0	0	0	0	1	SL	X	X	72 μ s
扩充功能设定	0	0	0	0	1	DL	X	1/RE	G	0	72 μ s
设定 IRAM 地址或滚动地址	0	0	0	1	A5	A4	A3	A2	A1	A0	72 μ s
设定绘图 RAM 地址	0	0	1	0	0	0	A3	A2	A1	A0	72 μ s
				A6	A5	A4	A3	A2	A1	A0	

备注：当 ST7920 在接受指令前，MCU 必须先确认 ST7920 处于非忙状态。即读取 BF = 0，才能接受新的指令；如果在送出一条指令前不检查 BF 状态，则需要延时一段时间，以确保上一条指令执行完毕，具体指令执行时间参照指令表。

“RE”是基本指令集与扩充指令集的选择控制位，当变更“RE”的状态后，以后的指令维持在最后的状态。除非再次变更“RE”的状态，否则使用相同的指令集时，不需要重新设置“RE”。

3. 基本指令详细说明

1) 清除显示 (CLEAR)

格式	0	0	0	0	0	0	0	1
----	---	---	---	---	---	---	---	---

将 DDRAM 填满“20H”（空格）代码，并且设定 DDRAM 的地址计数器 (AC) 为 00H；更新设置进入设定点将 I/D 设为 1，游标右移 AC 加 1。

2) 地址归 0 (HOME)

格式	0	0	0	0	0	0	1	X
----	---	---	---	---	---	---	---	---

设定 DDRAM 的地址寄存器为 00H，并且将游标移到开头原点位置；这个指令并不改变 DDRAM 的内容。

3) 进入设定点 (ENTRY MODE SET) 初始值：06H

格式	0	0	0	0	0	1	I/D	S
----	---	---	---	---	---	---	-----	---

指定在显示数据的读取与写入时，设定游标的移动方向及指定显示的移位

I/D=1，游标右移，DDRAM 地址计数器 (AC) 加 1

I/D=0，游标左移，DDRAM 地址计数器 (AC) 减 1

S: 显示画面整体位移

S	I/D	功能描述
H	H	画面整体左移
H	L	画面整体右移

4) 显示开关设置 (DISPLAY STATUS) 初始值: 08H

格式	0	0	0	0	1	D	C	B
----	---	---	---	---	---	---	---	---

控制整体显示开关, 游标开关, 游标位置显示反白开关

D=1, 整体显示开; D=0, 整体显示关, 但是不改变 DDRAM 内容

C=1, 游标显示开; C=0, 游标显示关

B=1, 游标位置显示反白开, 将游标所在地址上的内容反白显示; B=0, 正常显示

5) 游标或显示移位控制(CURSOR AND DISPLAY SHIFT CONTROL)

初始值: 0001 XXXX B (X=0,1)

格式	0	0	0	1	S/C	R/L	X	X
----	---	---	---	---	-----	-----	---	---

这条指令不改变 DDRAM 的内容

S/C	R/L	方向	AC 的值
L	L	游标向左移动	AC=AC-1
L	H	游标向右移动	AC=AC+1
H	L	显示向左移动, 游标跟着移动	AC=AC
H	H	显示向右移动, 游标跟着移动	AC=AC

6) 功能设定(FUNCTION SET) 初始值: 0011 X0XX B (X=0,1)

格式	0	0	1	DL	X	0/RE	X	X
----	---	---	---	----	---	------	---	---

DL: 8/4 位接口控制位

DL=1, 8 位 MPU 接口; DL=0, 4 位 MPU 接口

RE: 指令集选择控制位

RE=1, 扩充指令集; RE=0, 基本指令集

同一指令的动作不能同时改变 DL 和 RE, 需先改变 DL 再改变 RE 才能确保设置正确

7) 设定 CGRAM 地址

格式	0	1	A5	A4	A3	A2	A1	A0
----	---	---	----	----	----	----	----	----

设定 CGRAM 地址到地址计数器 (AC), AC 范围为 00H~3FH 需确认扩充指令中 SR=0 (卷动位置或 RAM 地址选择)

8) 设定 DDRAM 地址

格式	1	0	A5	A4	A3	A2	A1	A0
----	---	---	----	----	----	----	----	----

设定 DDRAM 地址到地址计数器 (AC)

第一行 AC 范围 80H~89H

第二行 AC 范围 90H~99H

9) 读取忙标志和地址(RS=0,R/W=1)

格式	BF	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----	----

读取忙标志以确定内部动作是否完成, 同时可以读出地址计数器 (AC) 的值

10) 写显示数据到 RAM(RS=1,R/W=0)

格式	D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----	----

当显示数据写入后会使得 AC 改变, 每个 RAM (CGRAM, DDRAM, IRAM) 地址都可以连续写入 2 个字节的显示数据, 当写入第二个字节时, 地址计数器 (AC) 的值自动加一。

11) 读取显示 RAM 数据 (RS=1, R/W=1)

格式	D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----	----

读取后会改变 AC

设定 RAM (CGRAM, DDRAM, IRAM) 地址后, 先要 Dummy read 一次后才能读取到正确的显示数据, 第二次读取不需要 Dummy read, 除非重新设置了 RAM 地址

4. 扩充指令详细说明表

1) 待命模式

格式	0	0	0	0	0	0	0	1
----	---	---	---	---	---	---	---	---

进入待命模式, 执行如何其它指令都可以结束待命模式; 该指令不能改变 RAM 的内容。

2) 卷动位置或者 RAM 地址选择 初始值: 02H

格式	0	0	0	0	0	0	1	SR
----	---	---	---	---	---	---	---	----

当 SR=1 时, 允许输入垂直卷动地址

当 SR=0 时, 允许输入 IRAM 地址 (扩充指令) 及允许设定 CGRAM 地址 (基本指令)

3) 反白显示 初始值: 04H

格式	0	0	0	0	0	1	0	R0
----	---	---	---	---	---	---	---	----

选择 2 行中的任意一行作反白显示, 并可决定反白与否。R0 初始值为 0, 第一次执行时为反白显示, 再次执行时为正常显示

通过 R0 选择要作反白处理的行:

R0=0 第一行, R0=1 第二行

说明: 参考基本指令详细说明中的 DDRAM 地址说明

4) 睡眠模式 初始值: 0000 10XXB(X=0,1)

格式	0	0	0	0	1	SL	0	0
----	---	---	---	---	---	----	---	---

SL=1, 脱离睡眠模式

SL=0, 进入睡眠模式

5) 扩充功能设定

初始值: 001 DL X100 B (DL=1,8BIT 并口;DL=0,4BIT 并口 X=0,1)

格式	0	0	1	DL	X	RE	G	X
----	---	---	---	----	---	----	---	---

DL: 8/4 位接口控制位

DL=1, 8 位 MPU 接口; DL=0, 4 位 MPU 接口

RE: 指令集选择控制位

RE=1, 扩充指令集; RE=0, 基本指令集

G: 绘图显示控制位

G=1, 绘图显示开; G=0, 绘图显示关

同一指令的动作不能同时改变 RE 及 DL、G, 需先改变 DL 或 G 再改变 RE 才能确保设置正确

6) 设定 IRAM 地址或卷动地址

格式	0	1	A5	A4	A3	A2	A1	A0
----	---	---	----	----	----	----	----	----

SR=1, A5~A0 为垂直卷动地址; SR=0, A3~A0 为 IRAM 地址

7) 设定绘图 RAM 地址

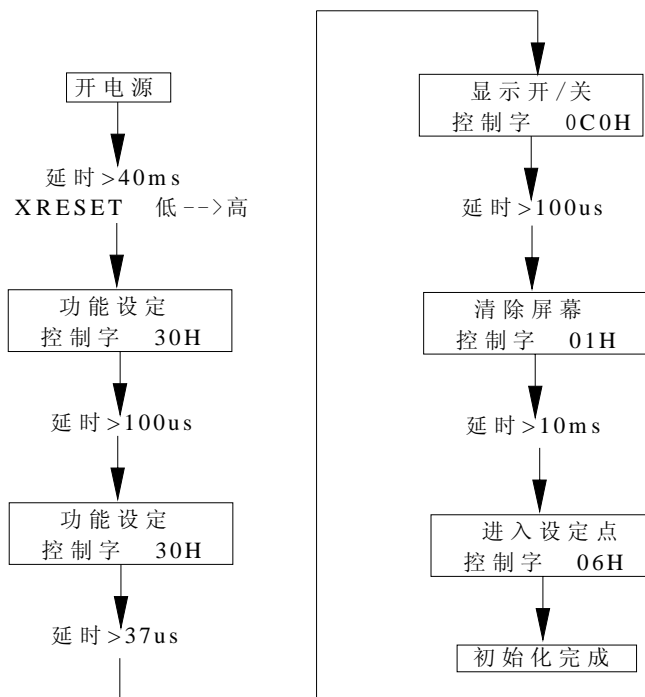
格式	1	0	0	0	A3	A2	A1	A0
		A6	A5	A4	A3	A2	A1	A0

设定 GDRAM 地址到地址计数器 (AC)，先设置垂直位置再设置水平位置 (连续写入 2 字节数据来完成垂直与水平坐标的设置)。

垂直地址范围:AC6~AC0

水平地址范围:AC3~AC0

5. 初始化流程

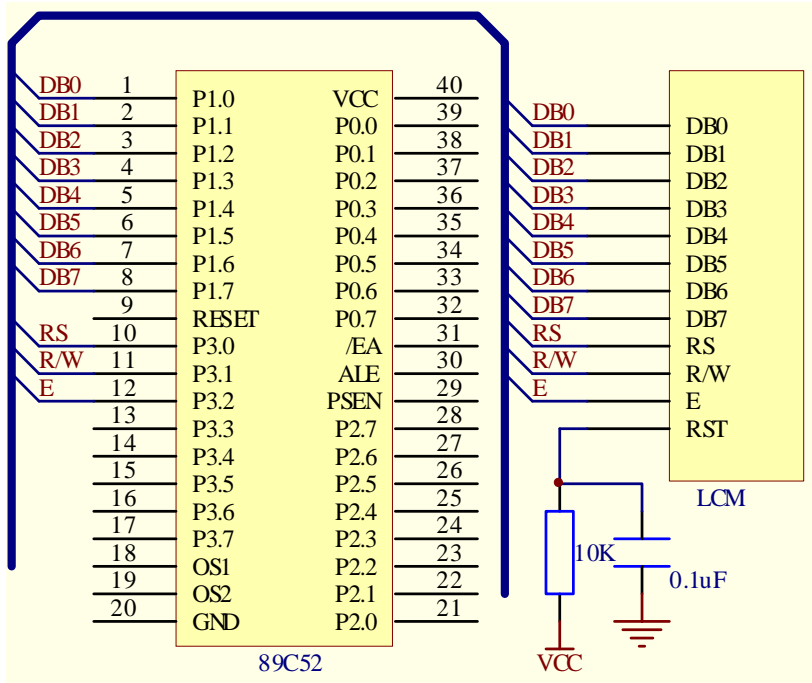


七、屏幕与 DDRAM 地址的对应关系

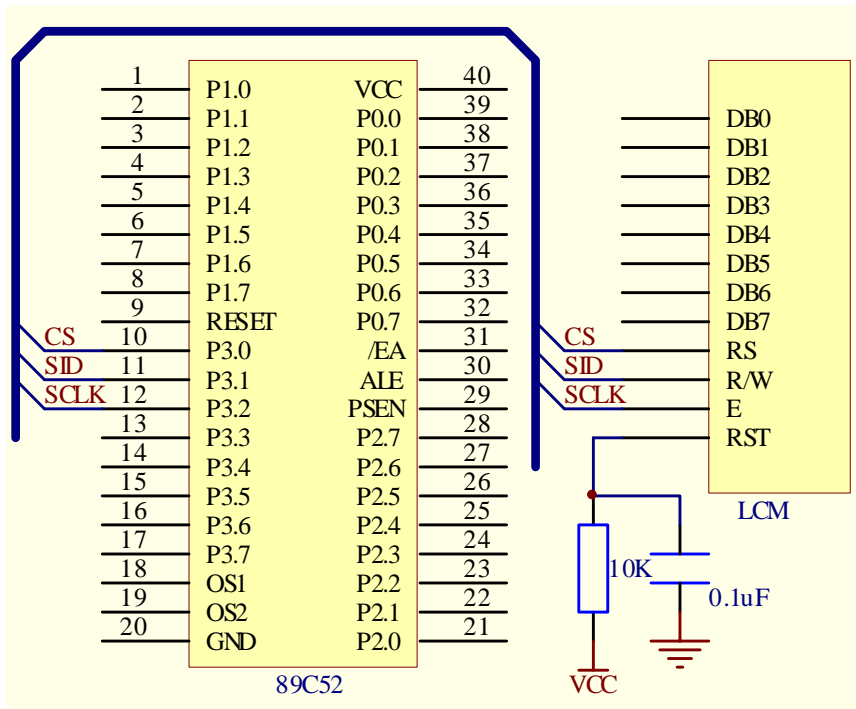
	第 1 字	第 2 字	第 9 字	第 10 字
第一行	80H	81H	88H	89H
第二行	90H	91H	98H	99H

八、16032ZA 与 MCU 的连接（示例）

1. 并口接线方式



2. 串口接线图



九、16032ZA 示例程序

本说明书中提供的示例程序并没有演示 ST7920 所有的功能，只是给用户做时序上的参考。最主要的是写指令和写数据子程序。下面是针对 51 单片机写的间接访问方式示例程序。

```

;*****
;CONTROLLER:ST7920(8BIT INTERFACE) *
;CRYSTAL:12MHz *
;RS=P3.0 RW=P3.1 E=P3.2 D0-D7=P1 *
;*****

RS EQU P3.0 ;H=DATA, L=COM
RW EQU P3.1 ;H=READ, L=WRITE
E EQU P3.2 ;

COM EQU 30H ;控制字暂存单元
DAT EQU 31H ;显示数据暂存单元
ADDR EQU 32H ;文字地址暂存单元
ADY EQU 33H ;图形模式 Y 地址
ADX EQU 34H ;图形模式 X 地址

ORG 0000H
AJMP MAIN
ORG 0030H

MAIN:
MOV SP, #60H
MOV P3, #0F8H
LCALL DEL_20MS
LCALL DEL_20MS
LCALL INI ;进入初始化子程序

MOV ADDR, #80H
MOV DPTR, #WEL_1
LCALL W_10HZ
MOV ADDR, #90H
MOV DPTR, #WEL_2
LCALL W_10HZ
LCALL DEL_700MS

MOV COM, #01H ;清除显示
LCALL WCOM

MOV DAT, #0FFH
LCALL WLINE
MOV DAT, #55H
LCALL WDOT
MOV DAT, #0AAH
LCALL WDOT

MOV DPTR, #BMP1
LCALL WBMP
LJMP MAIN

INI:
MOV COM, #30H ;功能设定, 8-bit 模式
LCALL WCOM
MOV COM, #30H ;功能设定, 基本指令

```

```

LCALL WCOM
MOV COM, #0CH ;显示开, 游标关, 反白关
LCALL WCOM
MOV COM, #01H ;清除显示
LCALL WCOM
MOV COM, #06H ;进入设定点, 游标 7 右移, 画面不移动
LCALL WCOM
RET

W_10HZ: ;写 8 汉字子程序
MOV COM, #30H ;功能设定, 基本指令, 确保送入的是 DDRAM 地址
LCALL WCOM
MOV COM, ADDR
LCALL WCOM
MOV R2, #20 ;20 个循环, 连续写入 10 个汉字或是 20 个西文字符
W_10HZ1:
MOV A, #00H
MOVC A, @A+DPTR
MOV DAT, A
LCALL WDAT
INC DPTR
DJNZ R2, W_10HZ1
RET

WDOT:
MOV COM, #34H ;功能设定, 扩充指令, 确保送入的是绘图 RAM 地址
LCALL WCOM
MOV COM, #36H ;功能设定, 扩充指令, 绘图显示开
LCALL WCOM
MOV ADY, #00H
MOV R2, #32
WDOT1:
MOV A, ADY
ORL A, #80H
MOV COM, A
LCALL WCOM
MOV COM, #80H
LCALL WCOM
MOV R3, #32
WDOT2:
LCALL WDAT
DJNZ R3, WDOT2
MOV A, DAT
CPL A
MOV DAT, A
INC ADY
DJNZ R2, WDOT1
LCALL DEL_700MS
MOV COM, #34H ;绘图关
LCALL WCOM
RET

WLINE:
MOV COM, #34H ;功能设定, 扩充指令, 确保送入的是绘图 RAM 地址
LCALL WCOM
MOV COM, #36H ;功能设定, 扩充指令, 绘图显示开
LCALL WCOM
MOV ADY, #00H
MOV R2, #32

```



```

WLINE1:
MOV  A, ADY
ORL  A, #80H
MOV  COM, A
LCALL WCOM
MOV  COM, #80H
LCALL WCOM
MOV  R3, #32
WLINE2:
LCALL WDAT
DJNZ R3, WLINE2
INC  ADY
DJNZ R2, WLINE1
LCALL DEL_700MS
MOV  COM, #34H      ;绘图关
LCALL WCOM
RET

WBMP:
MOV  COM, #34H      ;功能设定, 扩充指令, 确保送入的是绘图 RAM 地址
LCALL WCOM
MOV  COM, #36H      ;功能设定, 扩充指令, 绘图显示开
LCALL WCOM
MOV  ADY, #00H
MOV  R2, #32
WBMP1:
MOV  A, ADY
ORL  A, #80H
MOV  COM, A
LCALL WCOM
MOV  COM, #80H
LCALL WCOM
MOV  R3, #20
WBMP2:
CLR  A
MOVC A, @A+DPTR
MOV  DAT, A
LCALL WDAT
INC  DPTR
DJNZ R3, WBMP2
INC  ADY
DJNZ R2, WBMP1
LCALL DEL_700MS
MOV  COM, #34H      ;绘图关
LCALL WCOM
RET

WCOM:
CLR  RS
SETB RW
WC1:
MOV  P1, #0FFH
SETB E
MOV  A, P1
CLR  E
JB   ACC. 7, WC1
CLR  RW
MOV  P1, COM
SETB E

```

```
    CLR    E
    RET

WDAT:
    CLR    RS
    SETB   RW
WD1:
    MOV    P1, #0FFH
    SETB   E
    MOV    A, P1
    CLR    E
    JB     ACC. 7, WD1
    SETB   RS
    CLR    RW
    MOV    P1, DAT
    SETB   E
    CLR    E
    RET

DEL_20MS:
    MOV    R0, #20
D2:
    MOV    R1, #200
D1:
    NOP
    NOP
    NOP
    DJNZ   R1, D1
    DJNZ   R0, D2
    RET

DEL_100MS:
    LCALL  DEL_20MS
    LCALL  DEL_20MS
    LCALL  DEL_20MS
    LCALL  DEL_20MS
    LCALL  DEL_20MS
    RET

DEL_500MS:
    LCALL  DEL_100MS
    LCALL  DEL_100MS
    LCALL  DEL_100MS
    LCALL  DEL_100MS
    LCALL  DEL_100MS
    RET

DEL_700MS:
    LCALL  DEL_500MS
    LCALL  DEL_100MS
    LCALL  DEL_100MS
    RET

DEL_1000MS:
    LCALL  DEL_500MS
    LCALL  DEL_500MS
    RET

WEL_1:
```

DB "160 X 32 点阵液晶显示"

WEL_2:

DB "可以显示 10X2 个中文字"

BMP1:

DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,
 DB 0FFH,0FFH,0FFH,0FFH,080H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,000H,001H,0BFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,
 DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FDH,0A0H,000H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,005H,
 DB 0A0H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,005H,0A0H,000H,000H,000H,000H,00FH,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,002H,000H,000H,000H,005H,0A0H,000H,000H,000H,000H,010H,080H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,005H,000H,000H,000H,005H,0A0H,000H,000H,000H,
 DB 000H,010H,080H,000H,000H,000H,000H,000H,000H,000H,000H,002H,000H,000H,000H,005H,
 DB 0A0H,000H,000H,000H,000H,010H,080H,000H,000H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,005H,0A0H,000H,000H,000H,00FH,000H,000H,000H,000H,000H,000H,
 DB 001H,0F8H,000H,000H,000H,000H,000H,005H,0A0H,000H,000H,000H,000H,000H,000H,000H,
 DB 0FCH,000H,000H,030H,003H,0FCH,00CH,001H,003H,0F0H,000H,005H,0A0H,000H,000H,0FCH,
 DB 000H,000H,018H,001H,0FEH,006H,000H,048H,007H,0FEH,00CH,002H,087H,0F8H,018H,005H,
 DB 0A0H,001H,081H,0FEH,000H,030H,024H,003H,0FFH,006H,000H,048H,007H,0FFH,00EH,001H,
 DB 00FH,0FCH,018H,005H,0A0H,001H,083H,0FFH,000H,048H,024H,003H,0FFH,087H,000H,030H,
 DB 00FH,09FH,0F7H,0E0H,00FH,0FEH,01CH,005H,0A0H,003H,087H,0FFH,000H,048H,018H,007H,
 DB 0CFH,0FBH,0F0H,000H,00FH,05FH,0E3H,0F8H,01FH,03FH,0EFH,0C5H,0A0H,03FH,07FH,0CFH,
 DB 080H,030H,000H,007H,0AFH,0F1H,0FCH,000H,030H,00FH,0C3H,0E0H,01EH,0BFH,0C7H,0F5H,
 DB 0A0H,0FEH,03FH,0D7H,080H,000H,000H,018H,007H,0E1H,0F0H,008H,040H,00FH,0C1H,080H,
 DB 060H,01FH,087H,0C5H,0A0H,03EH,01FH,080H,061H,000H,004H,020H,007H,0E0H,0C0H,014H,
 DB 07FH,00FH,0C1H,080H,080H,01FH,083H,005H,0A0H,00CH,01FH,080H,012H,080H,00AH,03FH,
 DB 087H,0E0H,0C0H,008H,030H,00FH,0C1H,000H,0FEH,01FH,083H,005H,0A0H,00CH,01FH,087H,
 DB 0F1H,000H,004H,018H,007H,0E0H,080H,000H,008H,00FH,0E3H,000H,060H,01FH,082H,005H,
 DB 0A0H,004H,01FH,080H,060H,000H,000H,004H,007H,0F1H,080H,000H,00CH,00FH,0FEH,000H,
 DB 010H,01FH,0C6H,005H,0A0H,006H,03FH,080H,080H,000H,000H,006H,007H,0FFH,000H,000H,
 DB 003H,007H,0FCH,000H,018H,01FH,0FCH,005H,0A0H,003H,0FFH,081H,080H,000H,000H,001H,
 DB 083H,0FEH,000H,000H,000H,0FFH,0FCH,000H,006H,00FH,0F8H,005H,0A0H,001H,0FFH,006H,
 DB 000H,000H,000H,000H,07FH,0FEH,000H,000H,000H,001H,080H,000H,001H,0FFH,0F8H,005H,
 DB 0A0H,001H,0FFH,0F8H,000H,000H,000H,000H,000H,0C0H,000H,000H,000H,000H,000H,000H,
 DB 000H,003H,000H,005H,0A0H,000H,00CH,000H,000H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,005H,0A0H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,005H,0A0H,000H,000H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,005H,
 DB 0A0H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,005H,0BFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,
 DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FDH,080H,000H,000H,000H,000H,000H,000H,000H,
 DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,001H,0FFH,0FFH,0FFH,0FFH,
 DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,
 END