

YT12832ZK

液晶显示模块使用说明书

一、液晶模块概述

YT12832ZK 液晶显示模块是 128×32 点阵的汉字图形型液晶显示模块,可显示汉字及图形,内置国标 GB2312 码简体中文字库 (16X16 点阵)、128 个字符 (8X16 点阵) 及 64X256 点阵显示 RAM (GDRAM)。可与 CPU 直接接口,提供两种界面来连接微处理机: 8-位并行及串行两种连接方式。具有多种功能: 光标显示、画面移位、睡眠模式等。

二、模块基本特性

外型尺寸: 80.0×40.0×13.0 mm

视域尺寸: 60.0×18.0mm

显示类型: 黄底黑字

LCD 显示角度: 6 点

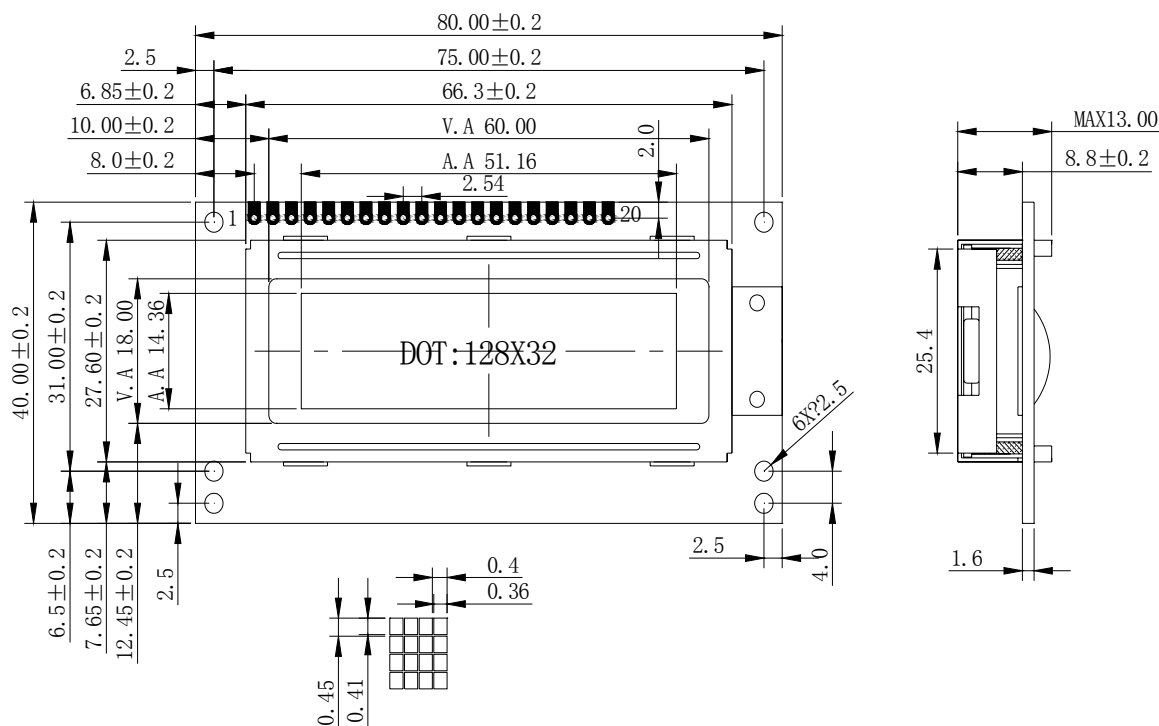
驱动方式: 1/32 duty, 1/6 bias

连接方式: 导电胶条, 铁框

- 补充说明: 模块外观尺寸可根据用户的要求进行适度调整。

三、外形尺寸

YT12832ZK 外型尺寸图 (Unit: mm): 此模组请注意有 2 组定位尺寸。



四、模块引脚说明

YT12832ZK 模块引脚说明

引脚	名称	方向	说明
1	VO	—	LCD 驱动电压调整端
2	VR	—	LCD 驱动电压调整端
3	VSS	—	地（0V）
4	VDD	—	工作电压（DC +5v）
5	NC	—	悬空
6	RS (CS)	I	H: 数据 L: 指令
7	R/W (STD)	I	H: 读操作 L: 写操作
8	E (SCLK)	I	使能信号，高电平有效
9	DB0	I/O	数据 0
10	DB1	I/O	数据 1
11	DB2	I/O	数据 2
12	DB3	I/O	数据 3
13	DB4	I/O	数据 4
14	DB5	I/O	数据 5
15	DB6	I/O	数据 6
16	DB7	I/O	数据 7
17	PSB	I	H: 并口模式 L: 串口模式
18	/RST	I	复位信号，低电平有效
19	LEDA	—	背光源正极（+5V）
20	LEDA	—	背光源正极（+5V）

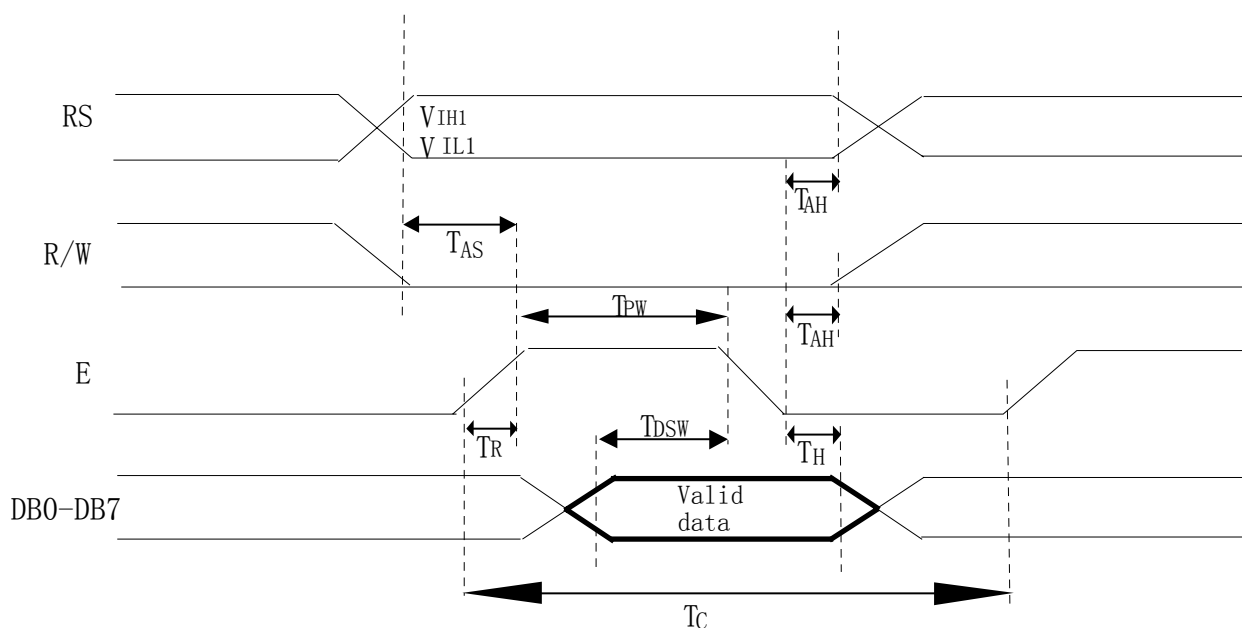
五、YT12832ZK 液晶硬件接口

- 1、逻辑工作电压(VDD): 3.3V~5.5V
- 2、电源地(GND): 0V
- 3、LCD 驱动电压(V0): 0~-10V
- 4、工作温度(Ta): 0~55℃(常温) / -20~70℃(宽温) 保存温度(Tstg): -10~65℃(常温)
- 5、电气特性见附图 1 外部连接图(参考附图 2)

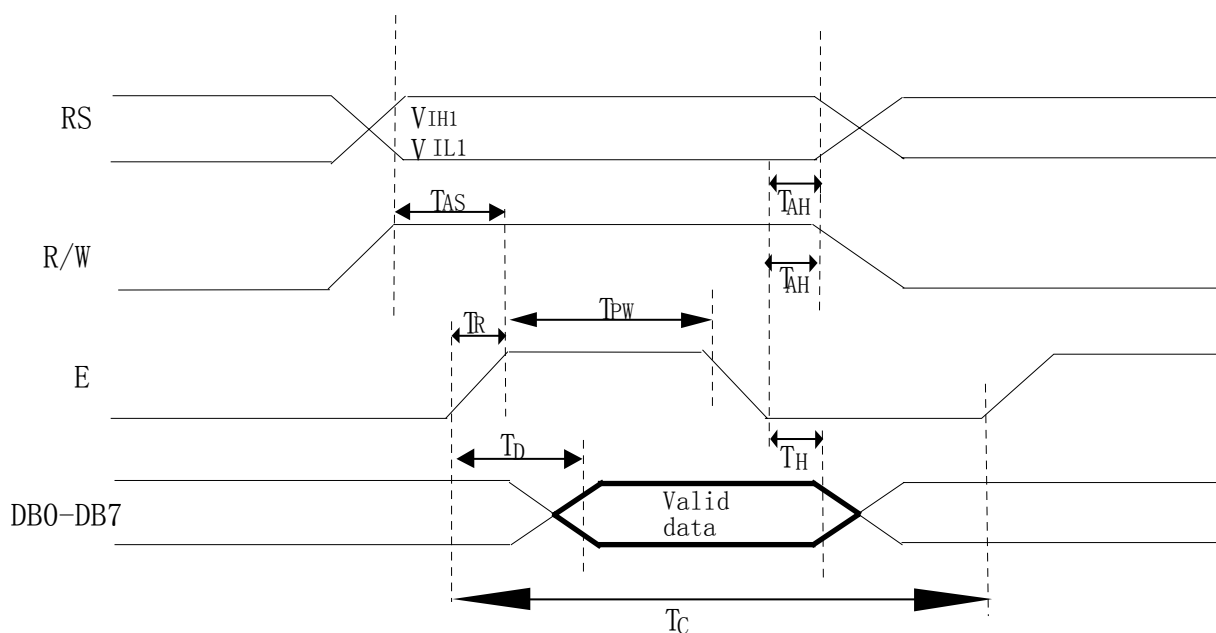
模块有并行和串行两种连接方法(时序如下):

1、8 位并行连接时序图

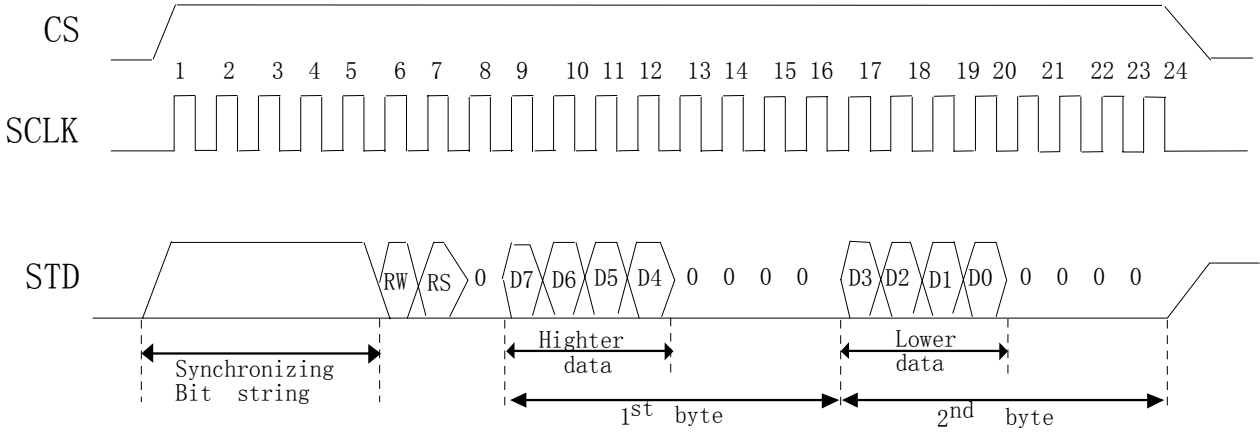
MPU 写资料到模块



MPU 从模块读出资料



2、串行连接时序图



3、AC 电气特性 (TA=25℃ VCC=5V)

SYMBOL	Characteristics	Test condition	Min.	Typ.	Max.	Unit
Fosc	OSC frequency	Rf=39K Ω	480	540	600	KHz
PARAMETER		SYMBOL		MESURE TIME		UNIT
FOR WRITE MODE(WRITING DATA FROM MPU TO ST7920)						
System cycle time		TC		13,000		ns
Address setup time		TAS		1,500		ns
Address hold time		TAH		1,500		ns
Data setup time		TDSW		1,000		ns
Data hold time		TH		20		ns
Enable pulsewidth		TPW		1,500		ns
Enable rise/fall time		TR,TF		25		ns
FOR READ MODE(READING DATA FROM ST7920 TO MPU)						
System cycle time		TC		13,000		ns
Address setup time		TAS		1,500		ns
Address hold time		TAH		1,500		ns
Data setup time		TDDR		1,000		ns
Data hold time		TH		20		ns
Enable pulsewidth		TPW		1,500		ns
Enable rise/fall time		TR,TF		25		ns

六、用户指令集

1、指令表 1: (RE=0: 基本指令集)

指令	指令码										说明	执行时间 (540KHZ)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
清除显示	0	0	0	0	0	0	0	0	0	1	将 DDRAM 填满 “20H”，并且设定 DDRAM 的地址计数器 (AC) 到 “00H”	4.6ms
地址归位	0	0	0	0	0	0	0	0	1	X	设定 DDRAM 的地址计数器 (AC) 到 “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	4.6ms
进入点 设定	0	0	0	0	0	0	0	1	I/D	S	指定在资料的读取与写入时，设定游标移动方向及指定显示的移位	72us

显示状态 开/关	0	0	0	0	0	0	1	D	C	B	D=1: 整体显示 ON C=1: 游标 ON B=1: 游标位置 ON	72us
游标或显示 移位控制	0	0	0	0	0	1	S/C	R/L	X	X	设定游标的移动与显示的移位控制位元; 这个指令并不改变 DDRAM 的内容	72us
功能设定	0	0	0	0	1	DL	X	0 RE	X	X	DL=1 (必须设为 1) <u>RE=1: 扩充指令集动作</u> <u>RE=0: 基本指令集动作</u>	72us
设定 CGRAM 地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	设定 CGRAM 地址到地址计数器 (AC)	72us
设定 DDRAM 地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	设定 DDRAM 地址到地址计数器 (AC)	72us
读取忙碌标志 (BF) 和地址	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	读取忙碌标志 (BF) 可以确认内部动作是否完成, 同时可以读出地址计数器 (AC) 的值	0us
写资料到 RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	写入资料到内部的 RAM (DDRAM/CGRAM/IRAM/GDRAM)	72us
读出 RAM 的值	1	1	D7	D6	D5	D4	D3	D2	D1	D0	从内部 RAM 读取资料 (DDRAM/CGRAM/IRAM/GDRAM)	72us

指令表—2: (RE=1: 扩充指令集)

指令	指令码										说明	执行时间 (540KHZ)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
待命模式	0	0	0	0	0	0	0	0	0	1	进入待命模式, 执行其他命令都可终止待命模式	72us
卷动地址或 IRAM 地址 选择	0	0	0	0	0	0	0	0	1	SR	SR=1: 允许输入垂直卷动地址 SR=0: 允许输入 IRAM 地址	72us
反白选择	0	0	0	0	0	0	0	1	R1	R0	选择 4 行中的任一行作反白显示, 并可决定反白与否	72us
睡眠模式	0	0	0	0	0	0	1	SL	X	X	SL=1: 脱离睡眠模式 SL=0: 进入睡眠模式	72us
扩充功能设定	0	0	0	0	1	1	X	1 RE	G	0	<u>RE=1: 扩充指令集动作</u> <u>RE=0: 基本指令集动作</u> G=1 : 绘图显示 ON G=0 : 绘图显示 OFF	72us
设定 IRAM 地址或卷动 地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	SR=1: AC5—AC0 为垂直卷动地址 SR=0: AC3—AC0 为 ICON IRAM 地址	72us
设定绘图 RAM 地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	设定 CGRAM 地址到地址计数器 (AC)	72us

备注: 1、当模块在接受指令前, 微处理必须先确认模块内部处于非忙碌状态, 即读取 BF 标志时 BF 需为 0, 方可接受新的指令; 如果在送出一个指令前并不检查 BF 标志, 那么在前一个指令和这个指令中间必须延迟一段较长的时间, 即是等待前一个指令确实执行完成, 指令执行的时间请参考指令表中的个别指令说明。

2、“RE”为基本指令集与扩充指令集的选择控制位元, 当变更“RE”位元后, 往后的指令集将维持在最后的状态, 除非再次变更“RE”位元, 否则使用相同指令集时, 不需每次重设“RE”位元。

具体指令介绍:**1、清除显示(指令代码为 01H)**

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	L	L	L	H

功能: 清除显示屏幕, 把 DDRAM 位址计数器调整为 “00H”

2、位址归位(02H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	L	L	H	X

功能: 把 DDRAM 位址计数器调整为 “00H”, 光标回原点, 该功能不影响显示 DDRAM

3、点设定(07H/04H/05H/06H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	L	H	I/D	S

功能: 设定光标移动方向并指定整体显示是否移动。

I/D=1 光标右移, I/D=0 光标左移。

SH=1 且 DDRAM 为写状态: 整体显示移动, 方向由 I/D 决定 (I/D=1 左移, I/D=0 右移)

SH=0 或 DDRAM 为读状态: 整体显示不移动

4、显示状态 开/关(08H/0CH/0EH/0FH)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	H	D	C	B

功能: D=1; 整体显示 ON C=1; 光标 ON B=1; 光标位置 ON

5、光标或显示移位控制(10H/14H/18H/1CH)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	H	S/C	R/L	X	X

功能: 10H/14H: 光标左/右移动; 18H/1CH: 整体显示左右移动, 光标跟随移动, AC 值不变

6、功能设定(36H/30H/34H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	H	DL	X	0 RE	X	X

功能: DL=1 (必须设为 1) RE=1; 扩充指令集动作 RE=0: 基本指令集动作

7、设定 CGRAM 位址(40H-7FH)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	H	AC5	AC4	AC3	AC2	AC1	AC0

功能: 设定 CGRAM 位址到位址计数器 (AC)

8、设定 DDRAM 位址(80H-9FH)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	H	AC6	AC5	AC4	AC3	AC2	AC1	AC0

功能: 设定 DDRAM 位址到位址计数器 (AC)

9、读取忙碌状态(BF)和位址(BF=1, 状态忙)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	H	L	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

功能: 读取忙碌状态 (BF) 可以确认内部动作是否完成, 同时可以读出位址计数器 (AC) 的值

10、写资料到 RAM

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	H	D7	D6	D5	D4	D3	D2	D1	D0

功能: 写入资料到内部的 RAM (DDRAM/CGRAM/TRAM/GDRAM)

11、读出 RAM 的值

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	H	H	D7	D6	D5	D4	D3	D2	D1	D0

功能：从内部 RAM 读取资料（DDRAM/CGRAM/TRAM/GDRAM）

12、待命模式 (01H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	L	L	L	H

功能：进入待命模式，执行其他命令都可终止待命模式

13、卷动位址或 IRAM 位址选择 (02H/03H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	L	L	H	SR

功能：SR=1；允许输入卷动位址 SR=0；允许输入 IRAM 位址

14、反白选择 (04H\05H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	L	H	R1	R0

功能：选择 4 行中的任一行作反白显示，并可决定反白的与否

15、睡眠模式 (08H/0CH)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	L	L	H	SL	X	X

功能：SL=1；脱离睡眠模式 SL=0；进入睡眠模式

16、扩充功能设定 (36H/30H/34H)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	L	H	H	X	1 RE	G	L

功能：RE=1；扩充指令集动作 RE=0；基本指令集动作 G=1；绘图显示 ON G=0；绘图显示 OFF

17、设定 IRAM 位址或卷动位址 (40H-7FH)

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	L	H	AC5	AC4	AC3	AC2	AC1	AC0

功能：SR=1；AC5~AC0 为垂直卷动位址 SR=0；AC3~AC0 写 ICONRAM 位址

18、设定绘图 RAM 位址 (80H-FFH)

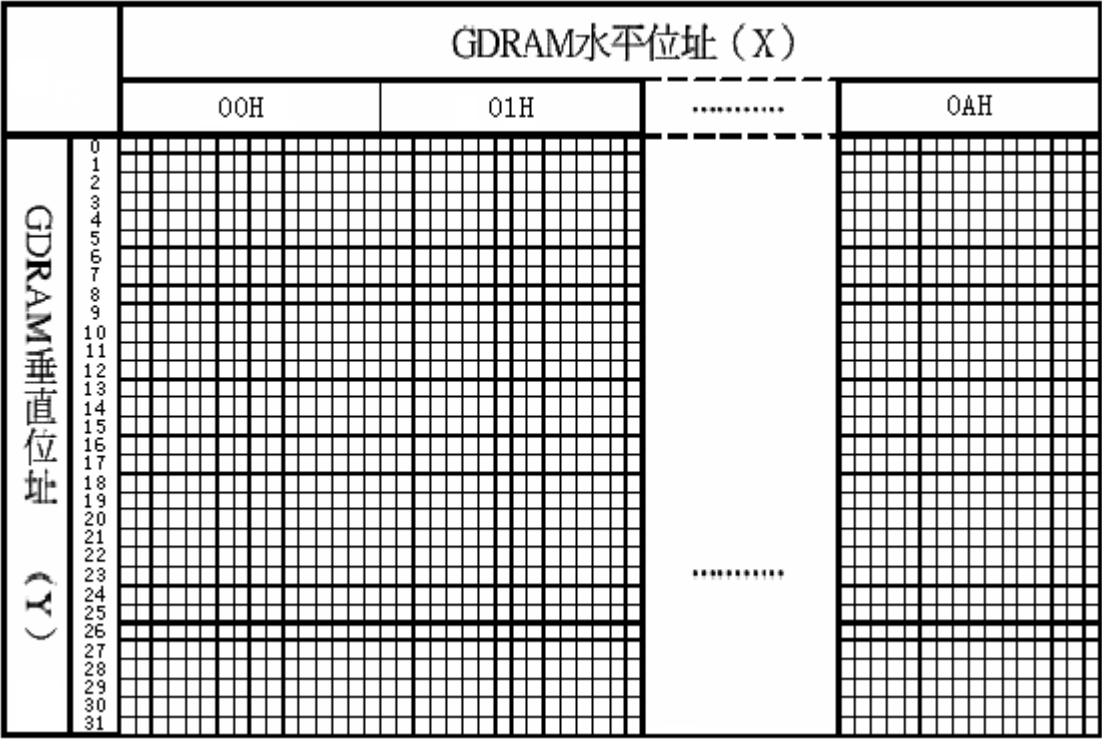
CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	L	H	AC6	AC5	AC4	AC3	AC2	AC1	AC0

功能：设定 GDRAM 位址到位址计数器（AC）

0

七、显示坐标关系

1、图形显示坐标



2、汉字显示坐标

行号	对应地址							
Line1	80H	81H	82H	83H	84H	85H	86H	87H
Line2	90H	91H	92H	93H	94H	95H	96H	97H

3、字符表

代码 (02H---7FH)

☒	☒	☒	♥	♦	♣	♣	•	•	◉	◉	♂	♀	♫	♫	※
▶	◀	‡	!!	¶	§	—	‡	↑	↓	→	←	└	↔	▲	▼
□	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
Q	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	{	}	~	△	

八、显示步骤

1、显示资料 RAM（DDRAM）

显示资料 RAM 提供 64×2 个位元组的空间，最多可以控制 4 行 16 字（64 个字）的中文字型显示，当写入显示资料 RAM 时，可以分别显示 CGROM、HCGROM 与 CGRAM 的字型；ST7920A 可以显示三种字型，分别是半宽的 HCGROM 字型、CGRAM 字型及中文 CGROM 字型，三种字型的选择，由在 DDRAM 中写入的编码选择，在 0000H—0006H 的编码中将自动的结合下一个位元组，组成两个位元组的编码达成中文字型的编码（A140—D75F），各种字型详细编码如下：

- 1、显示半宽字型：将 8 位元资料写入 DDRAM 中，范围为 02H—7FH 的编码。
- 2、显示 CGRAM 字型：将 16 位元资料写入 DDRAM 中，总共有 0000H，0002H，0004H，0006H 四种编码。
- 3、显示中文字形：将 16 位元资料写入 DDRAMK，范围为 A1A1H—F7FEH 的编码。

绘图 RAM（GDRAM）

绘图显示 RAM 提供 64×32 个位元组的记忆空间，最多可以控制 256×64 点的二维绘图缓冲空间，在更改绘图 RAM 时，先连续写入水平与垂直的坐标值，再写入两个 8 位元的资料到绘图 RAM，而地址计数器（AC）会自动加一；在写入绘图 RAM 的期间，绘图显示必须关闭，整个写入绘图 RAM 的步骤如下：

- 1、关闭绘图显示功能。
- 2、先将垂直的坐标（Y）写入绘图 RAM 地址；
- 3、再将水平的位元组坐标（X）写入绘图 RAM 地址；
- 4、将 D15——D8 写入到 RAM 中；
- 5、将 D7——D0 写入到 RAM 中；
- 6、打开绘图显示功能。

绘图显示的记忆体对应分布请参考表

2、游标/闪烁控制

ST7920A 提供硬体游标及闪烁控制电路，由地址计数器（address counter）的值来指定 DDRAM 中的游标或闪烁位置。

九、显示示例程序

以下程序为 51 系列驱程

1、发送子程序

```
SEND_DATA:  LCALL  CHK_BUSY      ;检测模块内部工作状态
              NOP
              SETB   RS          ; RS=1  选择数据寄存器
              CLR    RW          ; RW=0  写状态
              SETB   E           ;
              MOV    P1, A       ; 送数据到 I/O 口
```

```

        LCALL    DELAY01          ; 延时
        CLR      E                ;
        NOP
        MOV      P1, #0FFH
        RET
SEND_INT: LCALL    CHK_BUSY       ;检测模块内部工作状态
        NOP
        CLR      RS                ; RS=0  选择指令寄存器
        CLR      RW                ; RW=0  写状态
        SETB     E                ;
        MOV      P1, A            ; 送数据到 I/O 口
        LCALL    DELAY01          ; 延时
        CLR      E                ;
        NOP
        MOV      P1, #0FFH
        RET

```

2、读子程序

```

CHK_BUSY: CLR      RS                ; RS=0  选择指令寄存器
          SETB     RW                ; RW=1  读状态
          SETB     E                ;
          MOV      P1, A            ; 读入数据
HEHE1:    JB       P1. 7, HEHE1     ; 判别 BF 位
          CLR      E                ;
          RET
READ:     LCALL    CHK_BUSY       ;检测模块内部工作状态
          SETB     RS                ; RS=1  选择数据寄存器
          SETB     RW                ; RW=1  读状态
          SETB     E                ;
          MOV      P1, A            ; 从 I/O 口读数据
          LCALL    DELAY01          ; 延时
          CLR      E                ;
          NOP
          MOV      P1, #0FFH
          RET

```

3、串口写子程序:

```

;-----
;Serial    ModeWrite Data/Write Instruction
;Use  CS=RS\SCLK=E\STD=R/W
;rs_stu=1  Write Data
;Rs_stu=0  Write Instruction
;-----

```

SERIAL_WRITE:

PUSH	A	CLR	SCLK
MOVA,DA_IN		SETB	SCLK
SETB	CS	CLR	SCLK
SETB	STD	SETB	SCLK

CLR SCLK	SETB SCLK
SETB SCLK	CLR SCLK
CLR SCLK	SETB SCLK
SETB SCLK	CLR SCLK
CLR SCLK	JB ACC.3,WD31
SETB SCLK	CLR STD
CLR SCLK	SJMP WD2
CLR STD	WD31: SETB STD
SETB SCLK	WD2: SETB SCLK
CLR SCLK	CLR SCLK
JNB RS_STU,WSTU	JB ACC.2,WD21
SETB STD	CLR STD
SJMP WSTU1	SJMP WD1
WSTU: CLR STD	WD21: SETB STD
WSTU1: SETB SCLK	WD1: SETB SCLK
CLR SCLK	CLR SCLK
CLR STD	JB ACC.1,WD11
SETB SCLK	CLR STD
CLR SCLK	SJMP WD0
JB ACC.7,WD7	WD11: SETB STD
CLR STD	WD0: SETB SCLK
SJMP WD6	CLR SCLK
WD7: SETB STD	JB ACC.0,WD01
WD6: SETB SCLK	CLR STD
CLR SCLK	SJMP WD02
JB ACC.6,WD61	WD01: SETB STD
CLR STD	WD02: SETB SCLK
SJMP WD5	CLR SCLK
WD61: SETB STD	CLR STD
WD5: SETB SCLK	SETB SCLK
CLR SCLK	CLR SCLK
JB ACC.5,WD51	SETB SCLK
CLR STD	CLR SCLK
SJMP WD4	SETB SCLK
WD51: SETB STD	CLR SCLK
WD4: SETB SCLK	SETB SCLK
CLR SCLK	CLR SCLK
JB ACC.4,WD41	POP A
CLR STD	RET
SJMP WD42	
WD41: SETB STD	
WD42: SETB SCLK	
CLR SCLK	
CLR STD	
SETB SCLK	
CLR SCLK	
SETB SCLK	
CLR SCLK	