

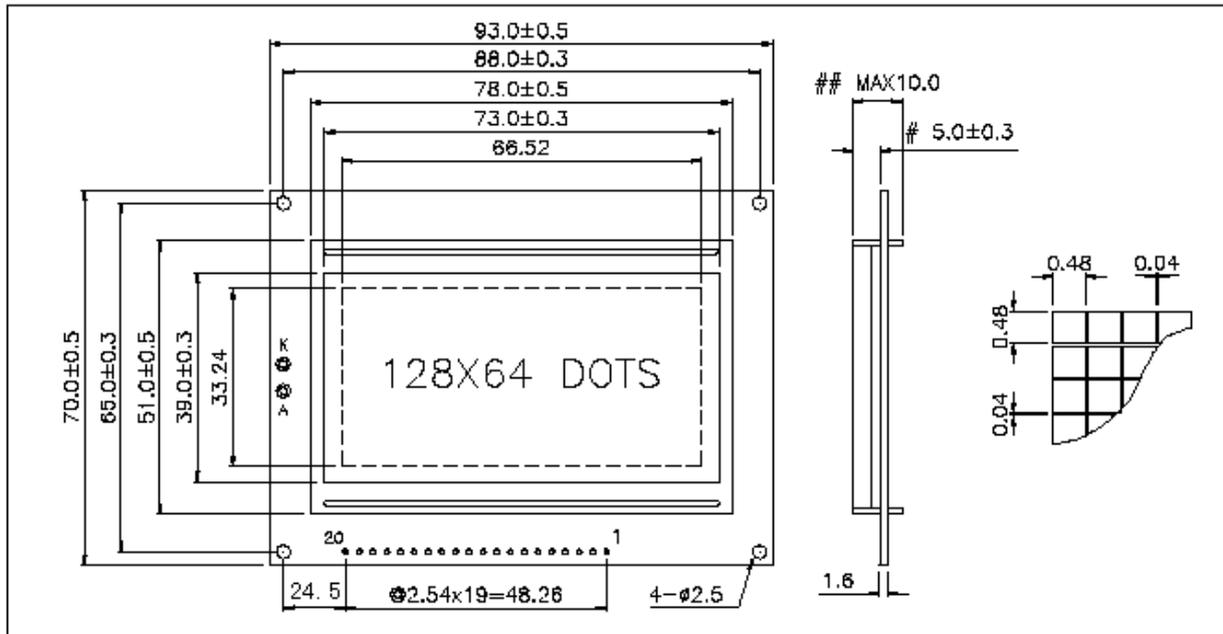
中文字库液晶显示模块使用手册

LCD12864

一、液晶显示模块概述

1. 液晶显示模块是 128×64 点阵的汉字图形型液晶显示模块，可显示汉字及图形，内置 8192 个中文汉字（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。
2. 外观尺寸：93×70×12.5mm
3. 视域尺寸：73×39mm

外形尺寸图



二、模块引脚说明

128X64 引脚说明

引脚名称	方向	说明	引脚	名称	方向	说明	
1	VSS	-	电源地 (0V)	11	DB4	I	数据 4
2	VDD	-	电源正 (+5v)	12	DB5	I	数据 5
3	VO	-	对比度调节 (+5v)	13	DB6	I	数据 6
4	RS (CS)	0	H: 数据 L: 指令	14	DB7	I	数据 7
5	R/W (SID)	0	H: 读 L: 写	15	PSB	0	H: 8 位或 4 位并行方式 L: 串行方式
6	E (SCLK)	0	使能信号	16	NC	-	空脚
7	DB0	I	数据 0	17	/RST	0	Reset Signal 低电平有效
8	DB1	I	数据 1	18	NC	-	空脚
9	DB2	I	数据 2	19	LEDA	-	背光源正极 (LED+5v)
10	DB3	I	数据 3	20	LEDK	-	背光源负极 (LED-0V)

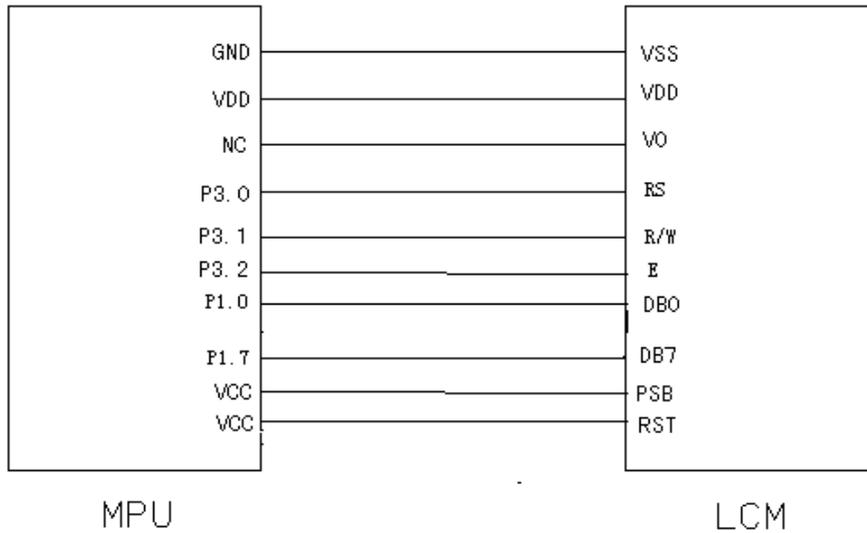
外形尺寸

ITEM	NOMINAL DIMEN	UNIT
模块体积	93×78×12.5	mm
视域	70.7×38.8	mm
行列点阵数	128×64	dots

点距离	0.52×0.52	mm
点大小	0.48×0.48	mm

三、液晶硬件接口

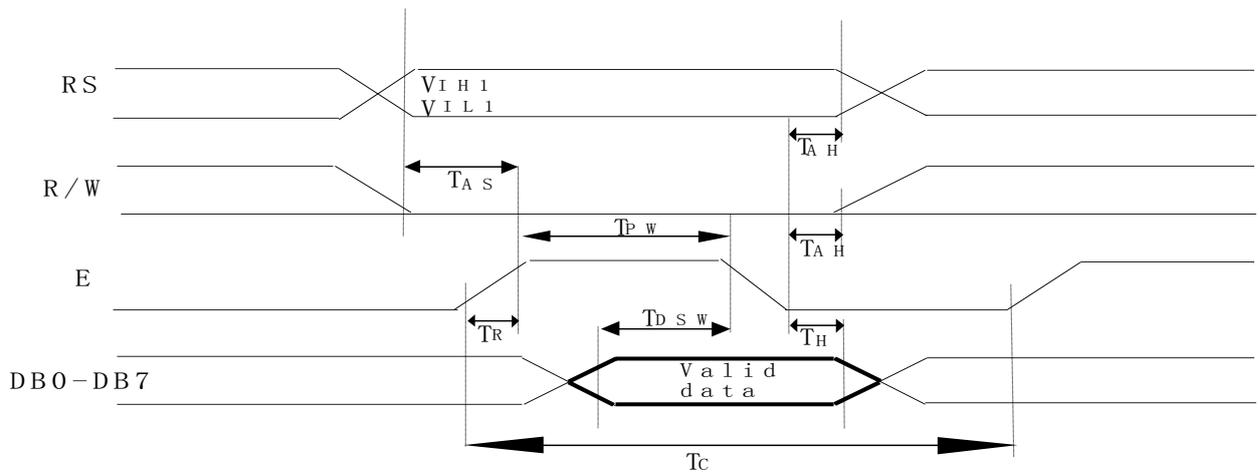
- 1、逻辑工作电压 (VDD): 4.5~5.5V
- 2、电源地 (GND): 0V
- 3、工作温度 (Ta): 0~60°C (常温) / -20~75°C (宽温)
- 4、电气特性见附图 1 外部连接图 (参考附图 2)



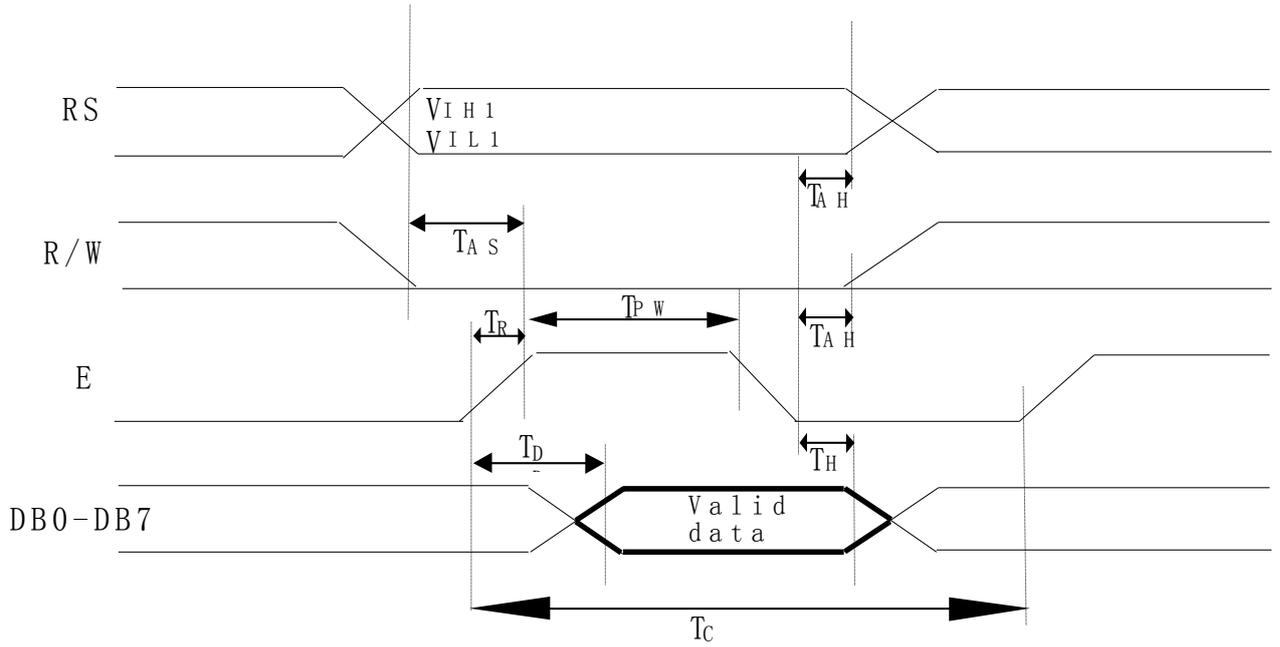
模块有并行和串行两种连接方法 (时序如下):

1、8 位并行连接时序图

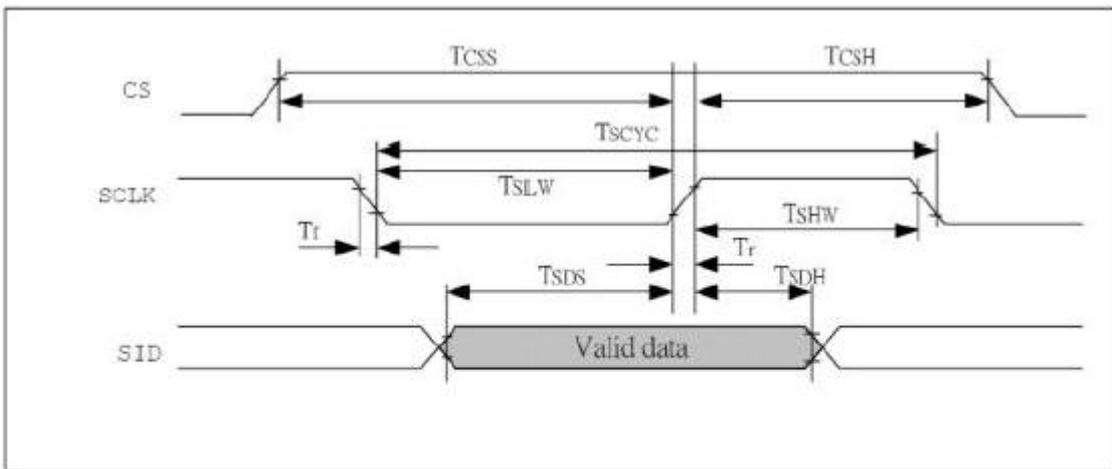
MPU 写资料到模块



MPU 从模块读出资料



2、串行连接时序图



串口方式下 MPU 写数据到 ST7920

四、用户指令集

1、指令表 1: (RE=0: 基本指令集)

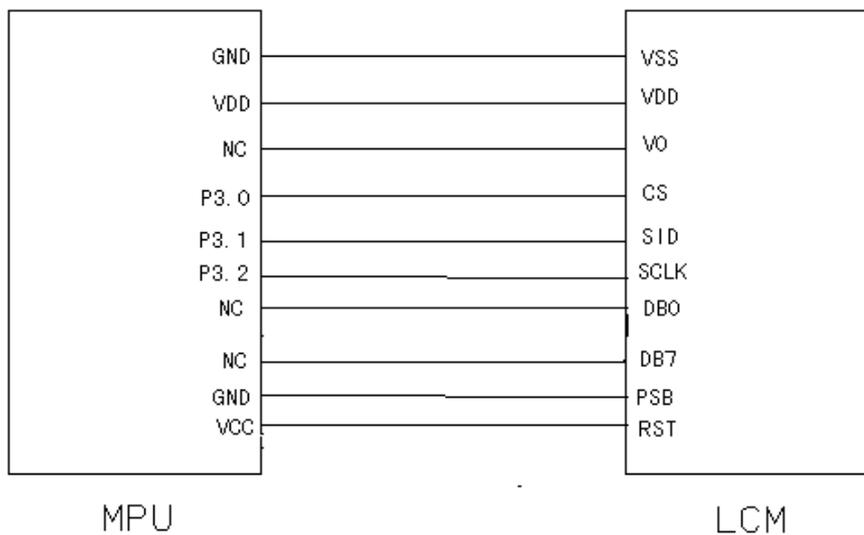
指令	指令码										说明	执行时间 (540KHZ)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
清除显示	0	0	0	0	0	0	0	0	0	1	将 DDRAM 填满“20H”，并且设定 DDRAM 的地址计数器 (AC) 到“00H”	4.6ms
地址归位	0	0	0	0	0	0	0	0	1	X	设定 DDRAM 的地址计数器 (AC) 到“00H”，并且将光标移到开头原点位置；这个指令并不改变 DDRAM 的内容	4.6ms
进入点 设定	0	0	0	0	0	0	0	1	I/D	S	指定在资料的读取与写入时，设定光标移动方向及指定显示的移位	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: 光标 ON B=1: 光标位置 ON	72us
光标或显示移 位控制	0	0	0	0	0	1	S/C	R/L	X	X	设定光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) RE=1: 扩充指令集动作 RE=0: 基本指令集动作	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	将 DDRAM 填满“20H”，并且设定 DDRAM 的地址计数器 (AC) 到“00H”	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	

HS12864-12 串口接线方式:



备注:

- 1、当模块在接受指令前, 微处理顺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

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

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

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

3、位址归位

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

功能：把 DDRAM 位址计数器调整为“00H”，游标回原点，该功能不影响显示 DDRAM 功能：执行该命令后，所设置的行将显示在屏幕的第一行。显示起始行是由 Z 地址计数器控制的，该命令自动将 A0-A5 位地址送入 Z 地址计数器，起始地址可以是 0-63 范围内任意一行。Z 地址计数器具有循环计数功能，用于显示行扫描同步，当扫描完一行后自动加一。

4、显示状态 开/关

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

功能：D=1；整体显示 ON C=1；游标 ON B=1；游标位置 ON

5、游标或显示移位控制

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

功能：设定游标的移动与显示的移位控制位：这个指令并不改变 DDRAM 的内容

6、功能设定

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

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

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

CODE:	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	L	H	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
	H	L	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（12H）

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（13H）

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（14H）

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 (015H)

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 (016H)

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 (017H)

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

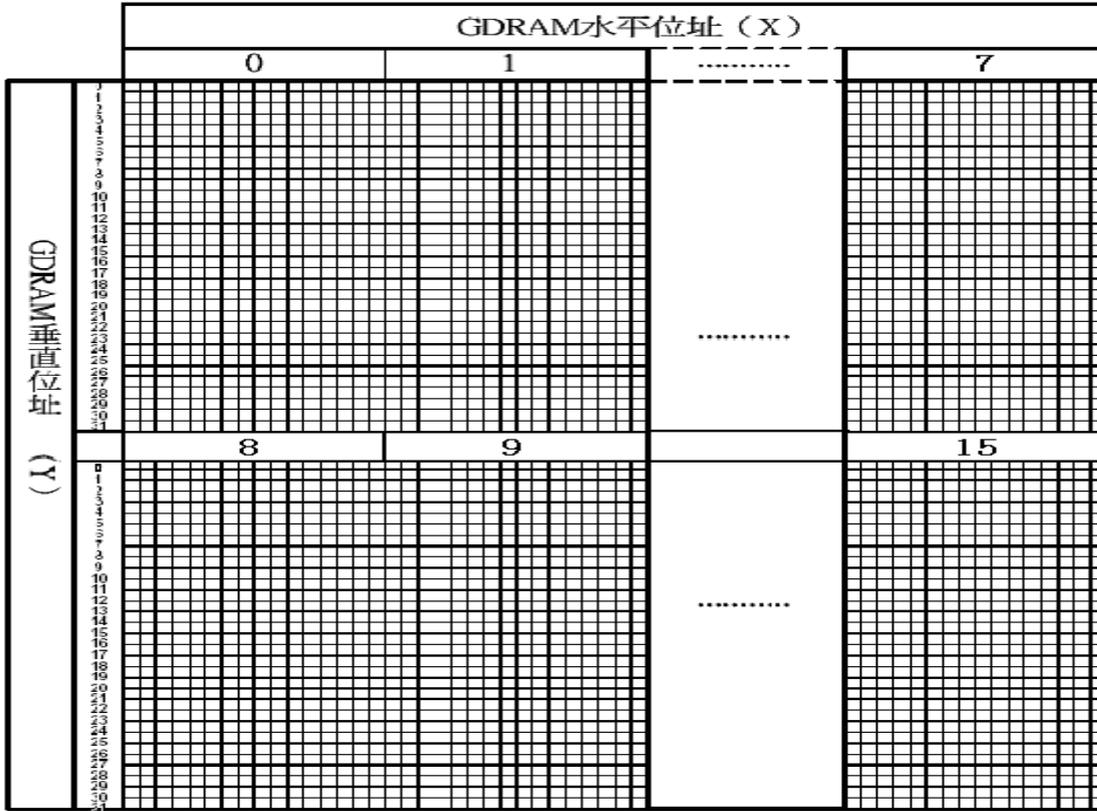
18、设定绘图 RAM 位址 (018H)

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)

五、显示坐标关系

1、图形显示坐标

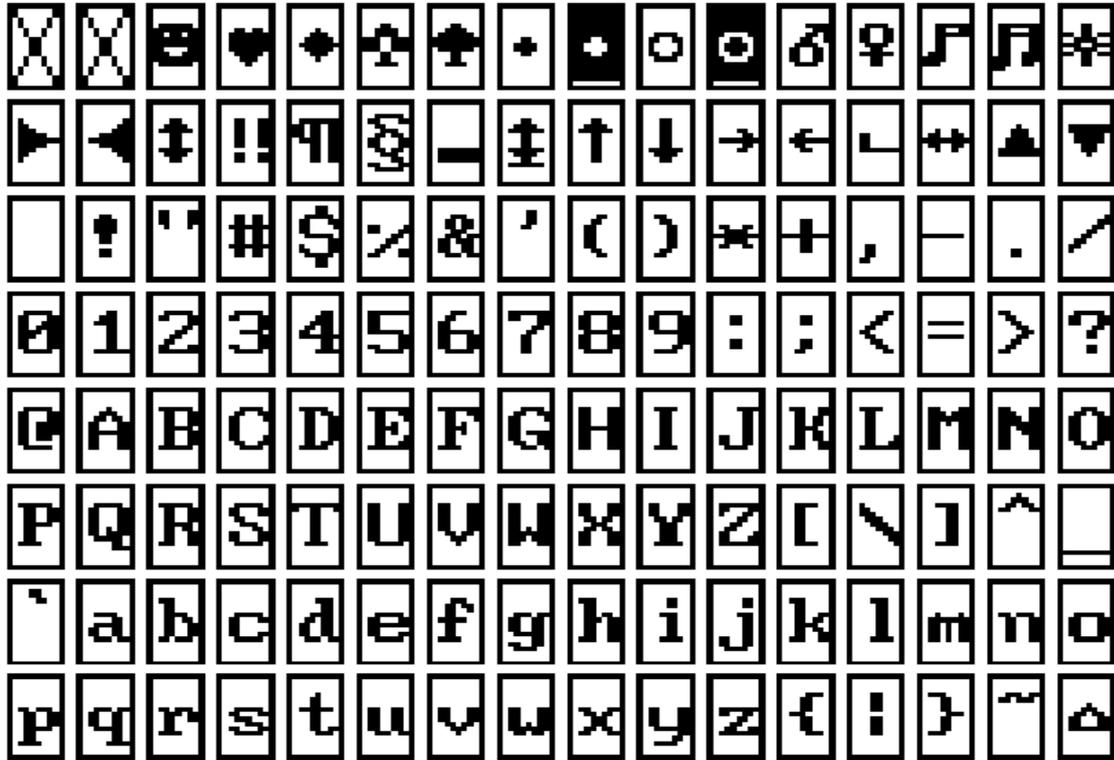


2、汉字显示坐标

	X 坐标							
Line1	80H	81H	82H	83H	84H	85H	86H	87H
Line2	90H	91H	92H	93H	94H	95H	96H	97H
Line3	88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
Line4	98H	99H	9AH	9BH	9CH	9DH	9EH	9FH

3、字符表

代码 (02H---7FH)



六、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 点的二维也纳绘图缓冲空间,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、先将水平的位元组坐标 (X) 写入绘图 RAM 地址;
- 3、再将垂直的坐标 (Y) 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。

七. 应用举例

LCD12864F 与单片机 8031 的一种接口如图 5. 所示:

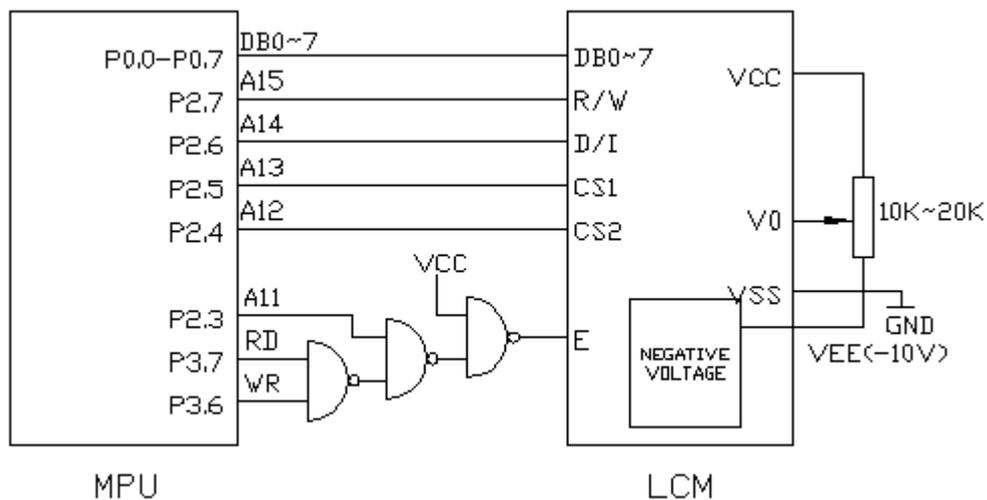


图 5

利用图 5 举例介绍编程实例

```
ORG 0000H
LJMP INITM
ORG 0100H
```

```
INITM: MOV SP, #67H           ; SET STACK ADDRESS
      MOV DPTR, #3800H       ; SELECT CHIP1 AND CHIP2
      MOV A, #3EH           ; OFF DISPLAY
      LCALL OUTI
      LCALL MS40
      LCALL MS40
      LCALL MS40
      MOV A, #3FH           ; ON DISPLAY
      LCALL OUTI
      LCALL MS40
      LCALL MS40
      LCALL MS40
```

; 显示“*”号

```
      MOV R3, #04H           ; PAGE NUMBER (2*4=8PAGES)
      MOV A, #0B8H           ; PAGE0
DISP1: PUSH ACC
      LCALL CHIN1
      POP ACC
      INC A
      INC A
      DJNZ R3, DISP1
      LCALL MS40
      LCALL MS40
      LCALL MS40
      LCALL MS40
      LCALL MS40
```

; 显示竖条

```
      MOV R3, #04H
      MOV A, #0B8H
DISP2: PUSH ACC
      LCALL CHIN2
      POP ACC
      INC A
      INC A
      DJNZ R3, DISP2
      LCALL MS40
      LCALL MS40
      LCALL MS40
      LCALL MS40
      LCALL MS40
```

; 显示横条

```
MOV R3, #04H
MOV A, #0B8H
DISP3: PUSH ACC
LCALL CHIN3
POP ACC
INC A
INC A
DJNZ R3, DISP3
LCALL MS40
LCALL MS40
LCALL MS40
LCALL MS40
LCALL MS40
```

; 显示汉字

```
MOV R3, #04H
MOV A, #0B8H
DISP4: PUSH ACC
LCALL CHIN4
POP ACC
INC A
INC A
DJNZ R3, DISP4
LCALL MS40
LCALL MS40
LCALL MS40
LCALL MS40
LCALL MS40
LJMP INITM
```

```
CHIN1: PUSH ACC ; PUT A (PAGE NUMBER) INTO STACK
```

```
LCALL OUT1
```

```
MOV A, #40H ; SET Y ADDRESS
```

```
LCALL OUTI
```

```
MOV R2, #32
```

```
LOAD1: MOV A, #55H
```

```
LCALL OUTD
```

```
MOV A, #0AAH
```

```
LCALL OUTD
```

```
DJNZ R2, LOAD1
```

```
POP ACC
```

```
INC A
```

```
LCALL OUTI
```

```
MOV A, #40H
```

```
LCALL OUTI
MOV R2, #32
LOAD12: MOV A, #55H
LCALL OUTD
MOV A, #0AAH
LCALL OUTD
DJNZ R2, LOAD12
RET

CHIN2: PUSH ACC ; PUT A (PAGE NUMBER) INTO STACK
LCALL OUT1
MOV A, #40H ; SET Y ADDRESS
LCALL OUTI
MOV R2, #32
LOAD2: MOV A, #00H
LCALL OUTD
MOV A, #0FFH
LCALL OUTD
DJNZ R2, LOAD2
POP ACC
INC A
LCALL OUTI
MOV A, #40H
LCALL OUTI
MOV R2, #32
LOAD21: MOV A, #00H
LCALL OUTD
MOV A, #0FFH
LCALL OUTD
DJNZ R2, LOAD21
RET

CHIN3: PUSH ACC ; PUT A (PAGE NUMBER) INTO STACK
LCALL OUT1
MOV A, #40H ; SET Y ADDRESS
LCALL OUTI
MOV R2, #64
LOAD3: MOV A, #55H
LCALL OUTD
DJNZ R2, LOAD3
POP ACC
INC A
LCALL OUTI
MOV A, #40H
```

```
        LCALL OUTI
        MOV R2, #64
LOAD31: MOV A, #55H
        LCALL OUTD
        DJNZ R2, LOAD31
        RET

CHIN4:  PUSH ACC
        LCALL OUT1
        MOV A, #40H
        LCALL OUTI
        MOV R2, #64
        MOV R1, #00H
        MOV DPTR, #CHINESE
LOAD4:  MOV A, R1
        MOVC A, @A+DPTR
        LCALL OUTD
        INC DPTR
        DJNZ R2, LOAD4
        POP ACC
        INC A
        LCALL OUTI
        MOV R2, #64
LOAD41: MOV A, R1
        MOVC A, @A+DPTR
        LCALL OUTD
        INC DPTR
        DJNE R2, LOAD41
        RET

MS40:  MOV R7, #0E8H
MS2:   MOV R6, #0FFH
MS1:   DJNZ R6, MS1
        DJNZ R7, MS2
        RET

; OUT INSTRUCTION FOR CHIP1 AND CHIP2
OTUI:  PUSH DPH
        PUSH DPL
        MOV DPTR, #3800H
        MOVX @A+DPTR, A
        POP DPL
        POP DPH
        RET

OUTD:  PUSH DPH
        PUSH DPL
```

```

MOV DPTR, #7800
MOVX @DPTR, A
POP DPL
POP DPH
RET

```

CHINNESE:; (PAGE0)

```

DB 10H, 60H, 01H, 86H, 60H, 04H, 1cH, e4H, 04H, 04H, 04H, E4H, 1cH,
04H, 00H, 00

```

```

DB 00H, 00H, 80H, 9fH, 95H, d5H, 55H, 55H, 55H, 15H, d5H, 15H, 1fH,
00H, 00H, 00

```

```

DB 10H, 61H, 06H, E0H, 18H, 84H, E4H, 1CH, 84H, 65H, BEH, 24H, 0A4H,
64H, 04H, 00,

```

```

DB 00H, 00H, 00H, 00H, 7EH, 2AH, 2AH, 2AH, 2AH, 2AH, 2AH, 7EH, 00H,
00H, 00H, 00

```

; (PAGE1)

```

DB 04, 04H, 7eH, 01H, 40H, 20H, 20H, 10H, 0bH, 04H, 0bH, 10H, 30H,
60H, 20H, 00

```

```

DB 02, 02H, 42H, 22H, 1aH, 07H, 02H, 02H, 02H, 02H, 0ffH, 02H, 02H, 02H, 02H, 00H
DB 04H, 04H, 0FFH, 00H, 00H, 00H, FFH, 41H, 21H, 12H, 0CH, 1BH, 61H, 0COH, 40H,
00

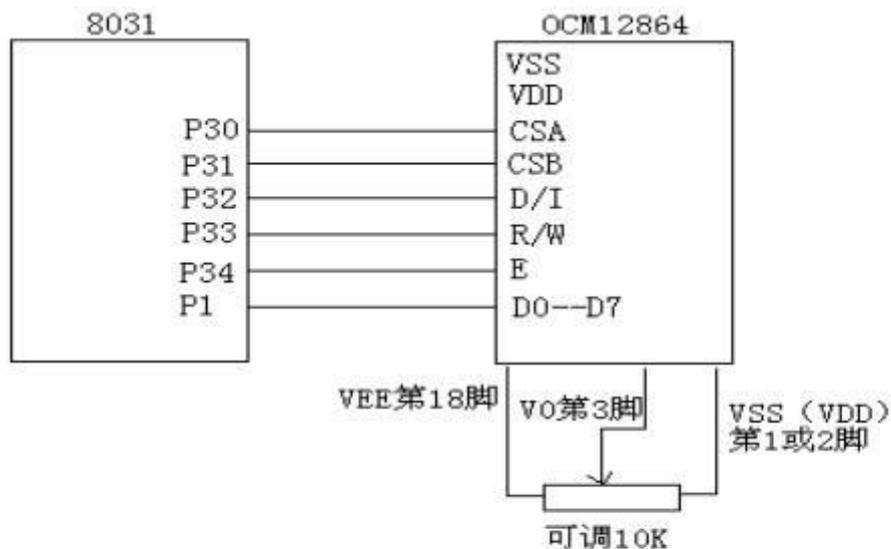
```

```

DB 00H, 7FH, 25H, 25H, 25H, 25H, 7FH, 00H, 00H, 7FH, 25H, 25H, 25H
25H, 7FH, 00H

```

读写模块程序举例



写指令子程序 (INST)

SETB E

CLR D_I

CLR R_W

MOV P1, A

CLR E

RET

写数据子程序 (DATA)

SETB E

SETB D_I

CLR R_W

MOV P1, A

CLR E

RET

写一页显示 RAM 数据 (假设指令子程序为 INST, 数据子程序为 DATA)

MOV A, #0B8H

LCALL INST ; 置页地址为 0 页

MOV A, #40H

LCALL INST ; 置列地址为 0 列

MOV R2, #40H

MOV R1, #00H

MOV DPTR, #ADDR

LOOP: MOV A, R1

MOVC A, @A+DPTR

LCALL DATA

INC R1

DJNZ R2, LOOP

模块由于内部含有中文字库，因此中文字符的显示非常简单，汇编只需给出该汉字的区位码即可显示该字。C语言直接在数组中写上汉字即可。区位码表可参见 **GB2312** 字符集。