

SHENZHEN BRILLIANT CRYSTAL TECHNOLOGIC CO.,LTD.

深 圳 市 彩 晶 科 技 有 限 公 司

本中文液晶说明书适应以下型号：

型号：CM12864-12SLYA

TEL:+86-755-21306020/26137169

FAX:+86-755-29995580

Http://www.szcm-lcd.com

E-mail:szcm@szcm-lcd.com

QQ:366910/66218549

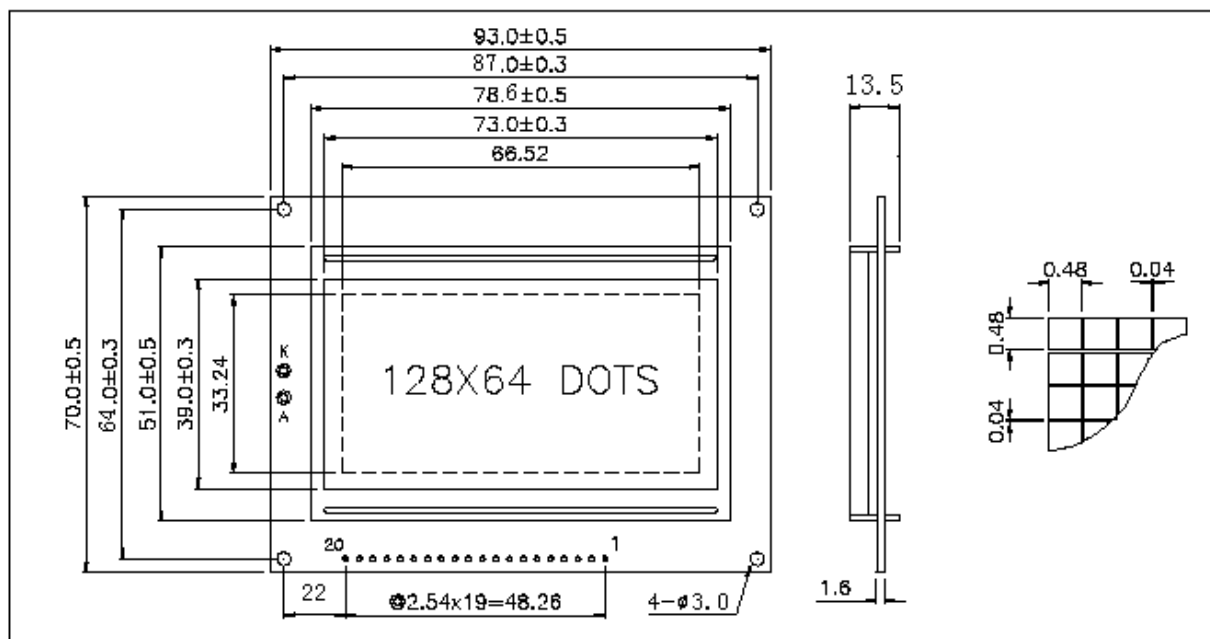
MSN:qq366910@hotmail.com

地址：深圳市南山区西丽留仙洞工业区顺和达厂区 B 栋东六楼

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：93×70×13.5mm
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	- GND (0V)	11	DB4	I 数据 4
2	VDD	- Supply Voltage For Logic (+5V)	12	DB5	I 数据 5
3	VO	- Supply Voltage For LCD	13	DB6	I 数据 6
4	RS (CS)	0 H: Data L: Instruction Code	14	DB7	I 数据 7
5	R/W (SID)	0 H: Read L: Write	15	PSB	0 H: Parallel Mode L: Serial Mode
6	E (SCLK)	0 Enable Signal		16	NC
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 × 70 × 13.5	mm
视域	73 × 39	mm
行列点阵数	128 × 64	dots
点距离	0.04 × 0.04	mm
点大小	0.48 × 0.48	mm

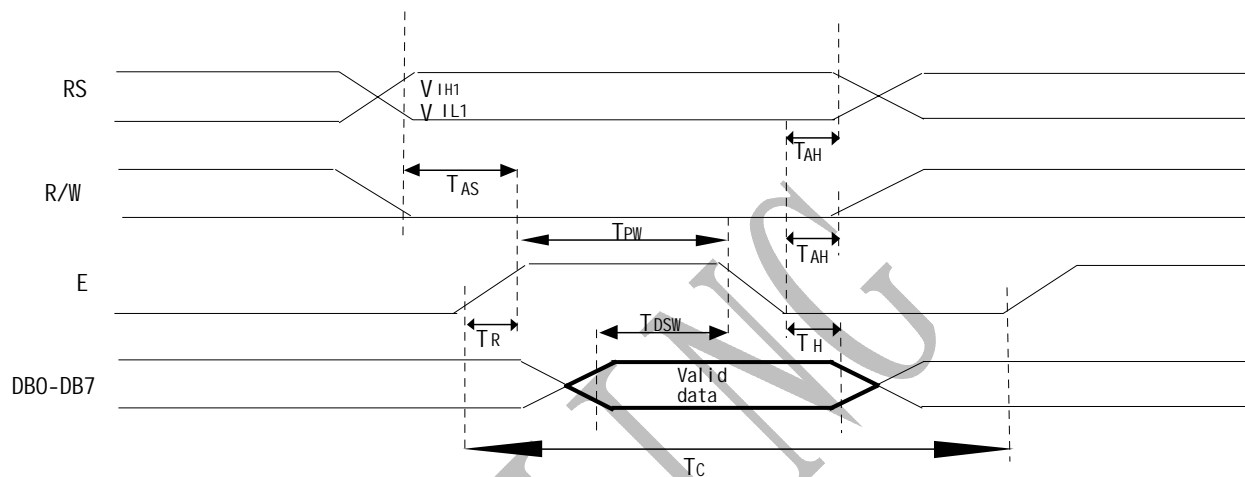
三、液晶硬件接口

- 1、逻辑工作电压(VDD) : 4.75 ~ 5.25V
- 2、电源地(GND) : 0V
- 3、工作温度(T_a) : 0 ~ 60 (常温) / -20 ~ 70 (宽温)

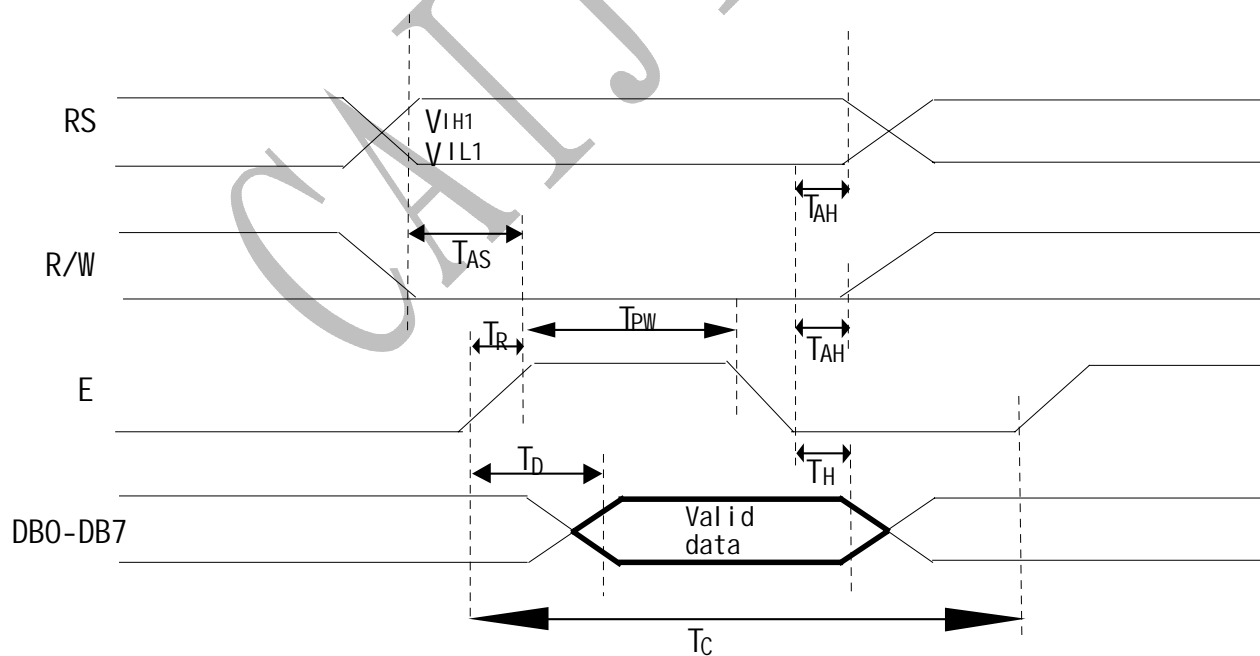
模块有并行和串行两种连接方法 (时序如下):

1、8 位并行连接时序图

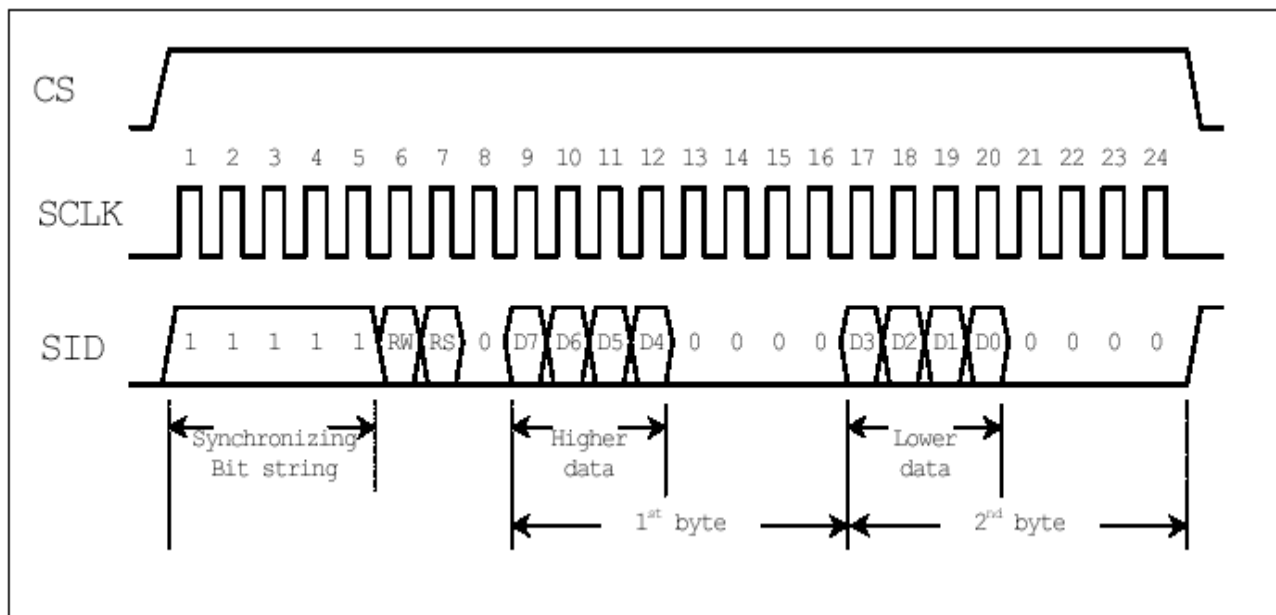
MPU 写资料到模块



MPU 从模块读出资料



2、串行连接时序图



串行数据传送共分三个字节完成：

第一字节：串口控制—格式 11111ABC

A 为数据传送方向控制：H 表示数据从 LCD 到 MCU，L 表示数据从 MCU 到 LCD

B 为数据类型选择：H 表示数据是显示数据，L 表示数据是控制指令

C 固定为 0

第二字节：(并行)8 位数据的高 4 位—格式 DDDD0000

第三字节：(并行)8 位数据的低 4 位—格式 0000DDDD

串行接口时序参数：(测试条件：T=25℃ VDD=4.5V)

Symbol	Characteristics	Test Condition	Min.	Typ.	Max.	Unit
<i>Internal Clock Operation</i>						
f _{OSC}	OSC Frequency	R = 33KΩ	470	530	590	KHz
<i>External Clock Operation</i>						
f _{EX}	External Frequency	-	470	530	590	KHz
	Duty Cycle	-	45	50	55	%
T _R , T _F	Rise/Fall Time	-	-	-	0.2	μs
T _{SCYC}	Serial clock cycle	Pin E	400	-	-	ns
T _{SHW}	SCLK high pulse width	Pin E	200	-	-	ns
T _{SLW}	SCLK low pulse width	Pin E	200	-	-	ns
T _{SDS}	SID data setup time	Pins RW	40	-	-	ns
T _{SDH}	SID data hold time	Pins RW	40	-	-	ns
T _{CSS}	CS setup time	Pins RS	60	-	-	ns
T _{CSH}	CS hold time	Pins RS	60	-	-	ns

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) 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

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	H	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	H	I/D	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	H	D	C	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	H	S/C	R/L	X	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

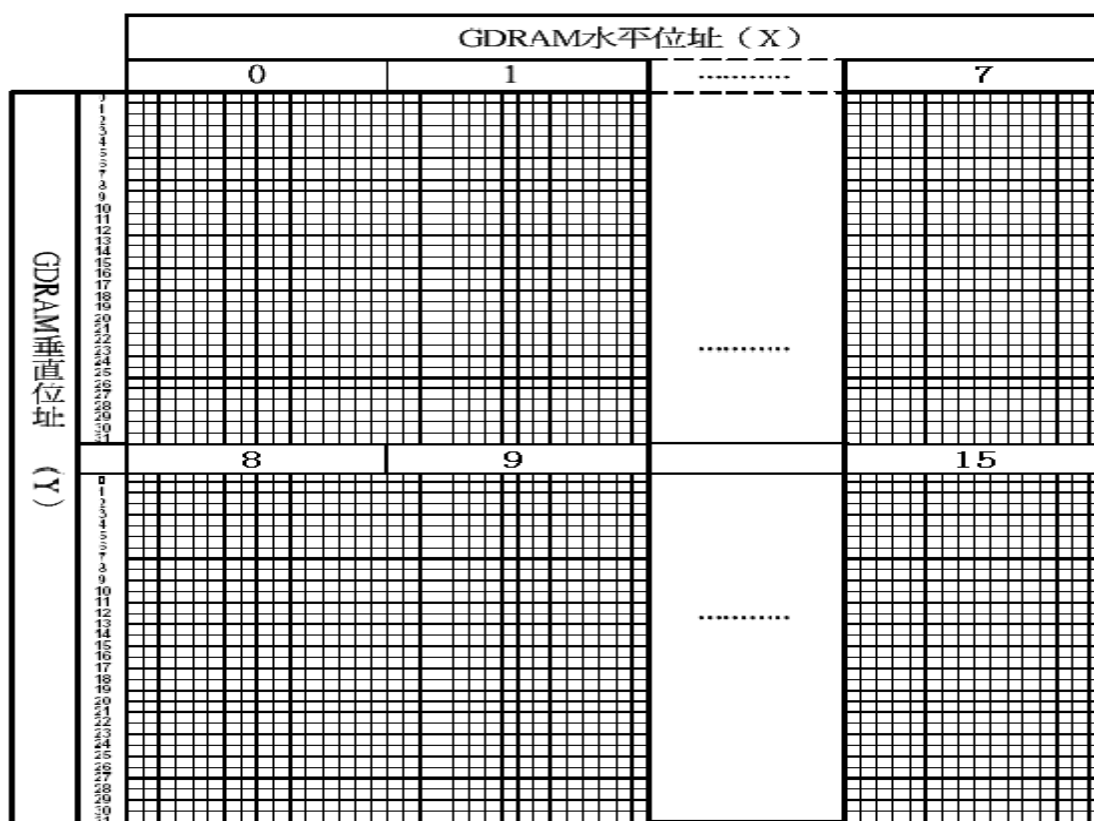
18、设定绘图 RAM 位址 (018H)

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							
Li ne1	80H	81H	82H	83H	84H	85H	86H	87H
Li ne2	90H	91H	92H	93H	94H	95H	96H	97H
Li ne3	88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
Li ne4	98H	99H	9AH	9BH	9CH	9DH	9EH	9FH

3、字符表

代码 (02H---7FH)

☒	☒	☒	♥	♦	♣	♣	•	●	○	◉	♂	♀	♪	♪	※
▶	◀	↑	!!	¶	§	—	±	↑	↓	→	←	└	↔	▲	▼
□	!	"	#	\$	%	&	'	()	*+,-.	/				
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	{	!	}	~	△

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：

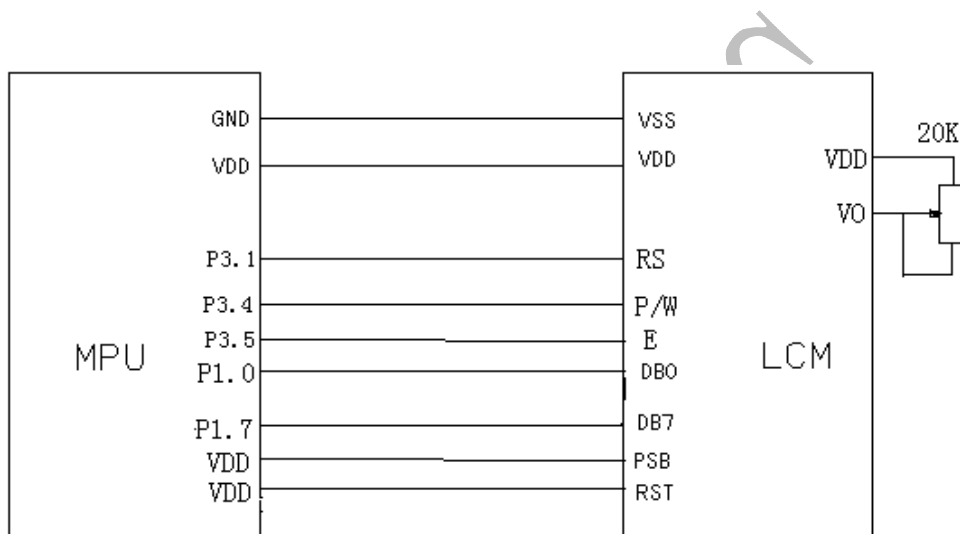
- 1、关闭绘图显示功能。
- 2、先将水平的位元组坐标 (X) 写入绘图 RAM 地址；
- 3、再将垂直的坐标 (Y) 写入绘图 RAM 地址；
- 4、将 D15——D8 写入到 RAM 中；
- 5、将 D7——D0 写入到 RAM 中；
- 6、打开绘图显示功能。

绘图显示的记忆体对应分布请参考表

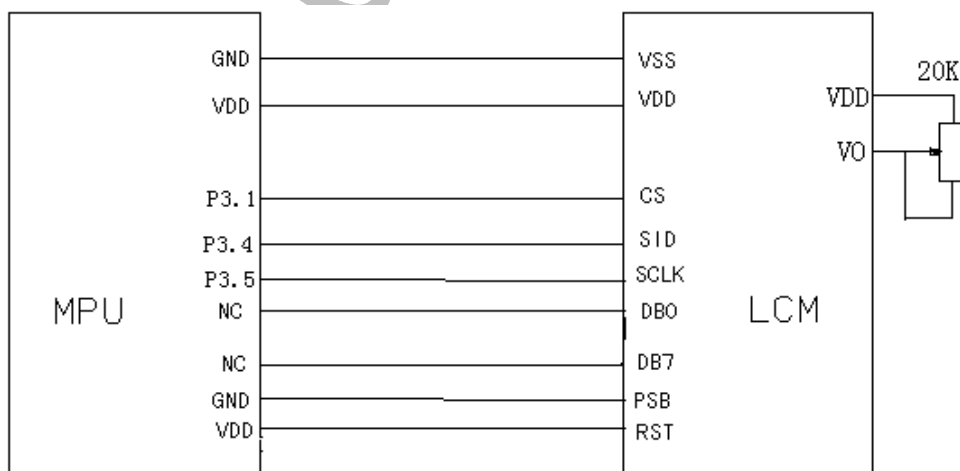
2、游标/闪烁控制

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。

CM12864-12 并口接线方式：



CM12864-12 串口接线方式：



```

ORG    0000H
LJMP   MAIN
ORG    0003H
LJMP   ZHONGDUAN
ORG    00100H

```

ZHONGDUAN:

```

PUSH    07H
PUSH    06H
PUSH    05H
SJMP    $
POP     05H
POP     06H
POP     07H
RETI
COMMAND EQU    30H
DATA1    EQU    31H
DATA2    EQU    32H
DATA3    EQU    33H
RS        EQU    P3.1
RW        EQU    P3.4
E         EQU    P3.5

```

;*****延时程序*****

DELAY:

```

MOV     R6,#00H
MOV     R7,#00H

```

DELAY1:

```

NOP
DJNZ    R7,DELAY1
DJNZ    R6,DELAY1
RET

```

;*****读 BF 和 AC 值*****

PR0:

```

PUSH    ACC
MOV     P1,#0FFH
CLR     RS
SETB    RW
SETB    E
MOV     COMMAND,P1
CLR     E
POP     ACC
RET

```

;*****写指令代码*****

PR1:

```

PUSH    ACC
CLR     RS
SETB    RW

```

```

PR11:MOV    P1,#0FFH
      SETB   E
      MOV    A,P1
      CLR    E
      JB     ACC.7,PR11
      CLR    RW
      MOV    P1,COMMAND
      SETB   E
      CLR    E
      POP    ACC
      RET

```

;*****写显示数据*****

```

PR2:
      PUSH   ACC
      CLR    RS
      SETB   RW
PR21:MOV    P1,#0FFH
      SETB   E
      MOV    A,P1
      CLR    E
      JB     ACC.7,PR21
      SETB   RS
      CLR    RW
      MOV    P1,DATA1
      SETB   E
      CLR    E
      POP    ACC
      RET

```

;*****读显示数据*****

```

PR3:
      PUSH   ACC
      CLR    RS
      SETB   RW
PR31:MOV    P1,#0FFH
      SETB   E
      MOV    A,P1
      CLR    E
      JB     ACC.7,PR31
      SETB   RS
      SETB   RW
      MOV    P1,#0FFH
      SETB   E
      MOV    COMMAND,P1
      CLR    E
      POP    ACC
      RET

```

,*****写 CGROM*****

CG:

```
LCALL PR1
MOV R0,#8
```

CG1:

```
MOV DATA1,DATA3
LCALL PR2
LCALL PR2
MOV DATA1,DATA2
LCALL PR2
LCALL PR2
DJNZ R0,CG1
RET
```

,*****写 DDRAM*****

DD:

```
LCALL PR1
MOV R0,#8
```

```
MM: MOV DATA1,#00H
LCALL PR2
MOV DATA1,DATA2
LCALL PR2
DJNZ R0,MM
RET
```

;

CM0:

```
MOV COMMAND,#01H ;清除显示
LCALL PR1
LCALL CLEARPHIC
```

```
MOV DPTR,#CHINESETAB0
MOV DATA2,#80H
MOV DATA3,#80H
```

CM1:

```
MOV COMMAND,#34H
LCALL PR1
```

```
MOV COMMAND,DATA3
LCALL PR1
```

```
MOV COMMAND,DATA2
LCALL PR1
MOV COMMAND,#30H
LCALL PR1
```

```
CLR A
MOVC A,@A+DPTR
```

```

MOV     DATA1,A
LCALL   PR2
INC     DPTR
CLR     A
MOVC    A,@A+DPTR
MOV     DATA1,A
LCALL   PR2
INC     DPTR
INC     DATA2
MOV     A,DATA2
CJNE    A,#88H,CM1
MOV     DATA2,#80H
INC     DATA3
MOV     A,DATA3
CJNE    A,#0A0H,CM1

```

```

MOV     DATA2,#88H
MOV     DATA3,#80H

```

CM2:

```

MOV     COMMAND,#34H
LCALL   PR1

MOV     COMMAND,DATA3
LCALL   PR1

MOV     COMMAND,DATA2
LCALL   PR1
MOV     COMMAND,#30H
LCALL   PR1

```

```

CLR     A
MOVC    A,@A+DPTR
MOV     DATA1,A
LCALL   PR2
INC     DPTR
CLR     A
MOVC    A,@A+DPTR
MOV     DATA1,A
LCALL   PR2
INC     DPTR
INC     DATA2
MOV     A,DATA2
CJNE    A,#90H,CM2
MOV     DATA2,#88H
INC     DATA3

```

```
MOV    A,DATA3
CJNE   A,#0A0H,CM2
```

```
MOV    COMMAND,#36H
LCALL  PR1
LCALL  DELAY
LCALL  DELAY
LCALL  DELAY
LCALL  DELAY
RET
```

```
=====
```

CLEARPHIC:

```
MOV    DATA2,#80H
MOV    DATA3,#80H
```

CM10:

```
MOV    COMMAND,#34H
LCALL  PR1
```

```
MOV    COMMAND,DATA3
LCALL  PR1
```

```
MOV    COMMAND,DATA2
LCALL  PR1
```

CM11:

```
MOV    COMMAND,#30H
LCALL  PR1
```

```
MOV    DATA1,#00H
LCALL  PR2
MOV    DATA1,#00H
LCALL  PR2
```

```
INC    DATA2
MOV    A,DATA2
CJNE   A,#90H,CM10
MOV    DATA2,#80H
INC    DATA3
MOV    A,DATA3
CJNE   A,#0A0H,CM10
```

```
MOV    COMMAND,#36H
LCALL  PR1
LCALL  DELAY
```



```

    LCALL    DELAY
    LCALL    DELAY
    LCALL    DELAY
    RET
;*****主程序*****
MAIN:
    NOP
    SETB     EA
    SETB     EX0
    SETB     IT0
    MOV      SP,#67H
    MOV      COMMAND,#30H
    CLR      RS
    SETB     RW
    MOV      P1,#0FFH
    SETB     E
    MOV      A,COMMAND
    MOV      A,P1
    CLR      E
    LCALL    DELAY
    MOV      COMMAND,#30H
    CLR      RS
    SETB     RW
    MOV      P1,#0FFH
    SETB     E
    MOV      A,COMMAND
    MOV      A,P1
    CLR      E
    LCALL    DELAY
    MOV      COMMAND,#30H      ;功能设置 8 位数据，基本指令
    LCALL    PR1
    MOV      COMMAND,#0CH      ;显示状态 ON，游标 OFF，反白 OFF
    LCALL    PR1
    MOV      COMMAND,#01H      ;清除显示
    LCALL    PR1
    MOV      COMMAND,#02H      ;地址归位
    LCALL    PR1

    MOV      COMMAND,#01H      ;清除显示
    LCALL    PR1

    LCALL    CM0

    MOV      COMMAND,#30H      ;功能设置 8 位数据，基本指令
    LCALL    PR1

```

```

MOV    COMMAND,#01H    ;清除显示
LCALL  PR1
MOV    COMMAND,#80H    ;设置 DDRAM 地址
LCALL  PR1
LCALL  DELAY
MOV    R0,#10H        ;显示 HCGROM
MOV    DATA1,#02H
L:  LCALL PR2
    INC  DATA1
    DJNZ R0,L
    MOV  COMMAND,#90H
    LCALL PR1
    MOV  DATA1,#41H
    MOV  R0,#10H
L1:  LCALL PR2
    INC  DATA1
    DJNZ R0,L1

    MOV  COMMAND,#88H
    LCALL PR1
    MOV  DATA1,#51H
    MOV  R0,#10H
L2:  LCALL PR2
    INC  DATA1
    DJNZ R0,L2

    MOV  COMMAND,#98H
    LCALL PR1
    MOV  DATA1,#61H
    MOV  R0,#10H
L3:  LCALL PR2
    INC  DATA1
    DJNZ R0,L3

    LCALL DELAY
    LCALL DELAY
    LCALL DELAY
;*****写 CGRAM*****

```

```

MOV    COMMAND,#40H
MOV    DATA3,#0FFH
MOV    DATA2,#00H
LCALL  CG

```

```

MOV    COMMAND,#50H
MOV    DATA3,#00H
MOV    DATA2,#0FFH

```

LCALL	CG
MOV	COMMAND,#60H
MOV	DATA3,#0AAH
MOV	DATA2,#0AAH
LCALL	CG
MOV	COMMAND,#70H
MOV	DATA3,#55H
MOV	DATA2,#55H
LCALL	CG
MOV	COMMAND,#80H
MOV	DATA2,#00H
LCALL	DD
MOV	COMMAND,#90H
MOV	DATA2,#00H
LCALL	DD
MOV	COMMAND,#88H
MOV	DATA2,#00H
LCALL	DD
MOV	COMMAND,#98H
MOV	DATA2,#00H
LCALL	DD
LCALL	DELAY
LCALL	DELAY
LCALL	DELAY
MOV	COMMAND,#80H
MOV	DATA2,#02H
LCALL	DD
MOV	COMMAND,#90H
MOV	DATA2,#02H
LCALL	DD
MOV	COMMAND,#88H
MOV	DATA2,#02H
LCALL	DD
MOV	COMMAND,#98H
MOV	DATA2,#02H
LCALL	DD
LCALL	DELAY
LCALL	DELAY
LCALL	DELAY
MOV	COMMAND,#80H
MOV	DATA2,#04H

```
LCALL DD
MOV COMMAND,#90H
MOV DATA2,#04H
LCALL DD
MOV COMMAND,#88H
MOV DATA2,#04H
LCALL DD
MOV COMMAND,#98H
MOV DATA2,#04H
LCALL DD
LCALL DELAY
LCALL DELAY
LCALL DELAY

MOV COMMAND,#80H
MOV DATA2,#06H
LCALL DD
MOV COMMAND,#90H
MOV DATA2,#06H
LCALL DD
MOV COMMAND,#88H
MOV DATA2,#06H
LCALL DD
MOV COMMAND,#98H
MOV DATA2,#06H
LCALL DD
LCALL DELAY
LCALL DELAY
LCALL DELAY

MOV COMMAND,#40H
MOV DATA3,#0AAH
MOV DATA2,#55H
LCALL CG
MOV COMMAND,#50H
MOV DATA3,#55H
MOV DATA2,#0AAH
LCALL CG
MOV COMMAND,#60H
MOV DATA3,#0FFH
MOV DATA2,#0FFH
LCALL CG

MOV COMMAND,#80H
MOV DATA2,#00H
LCALL DD
MOV COMMAND,#90H
```

```
MOV    DATA2,#00H
LCALL  DD
MOV    COMMAND,#88H
MOV    DATA2,#00H
LCALL  DD
MOV    COMMAND,#98H
MOV    DATA2,#00H
LCALL  DD
LCALL  DELAY
LCALL  DELAY
LCALL  DELAY
```

```
MOV    COMMAND,#80H
MOV    DATA2,#02H
LCALL  DD
MOV    COMMAND,#90H
MOV    DATA2,#02H
LCALL  DD
MOV    COMMAND,#88H
MOV    DATA2,#02H
LCALL  DD
MOV    COMMAND,#98H
MOV    DATA2,#02H
LCALL  DD
LCALL  DELAY
LCALL  DELAY
LCALL  DELAY
```

```
MOV    COMMAND,#80H
MOV    DATA2,#04H
LCALL  DD
MOV    COMMAND,#90H
MOV    DATA2,#04H
LCALL  DD
MOV    COMMAND,#88H
MOV    DATA2,#04H
LCALL  DD
MOV    COMMAND,#98H
MOV    DATA2,#04H
LCALL  DD
LCALL  DELAY
LCALL  DELAY
LCALL  DELAY
```

```
MOV    COMMAND,#01H
LCALL  PR1
```

```
MOV    COMMAND,#80H
LCALL  PR1
MOV    DPTR,#CHINESE1
MOV    R0,#10H
```

CHINA1:

```
CLR    A
MOVC   A,@A+DPTR
MOV    DATA1,A
LCALL  PR2
INC    DPTR
DJNZ   R0,CHINA1
```

```
MOV    COMMAND,#90H
LCALL  PR1
MOV    DPTR,#CHINESE2
MOV    R0,#10H
```

CHINA2:

```
CLR    A
MOVC   A,@A+DPTR
MOV    DATA1,A
LCALL  PR2
INC    DPTR
DJNZ   R0,CHINA2
```

```
MOV    COMMAND,#88H
LCALL  PR1
MOV    DPTR,#CHINESE3
MOV    R0,#10H
```

CHINA3:

```
CLR    A
MOVC   A,@A+DPTR
MOV    DATA1,A
LCALL  PR2
INC    DPTR
DJNZ   R0,CHINA3
```

```
MOV    COMMAND,#98H
LCALL  PR1
MOV    DPTR,#CHINESE4
MOV    R0,#10H
```

CHINA4:

```
CLR    A
MOVC   A,@A+DPTR
MOV    DATA1,A
```

```

LCALL    PR2
INC       DPTR
DJNZ     R0,CHINA4
LCALL    DELAY
LCALL    DELAY
LCALL    DELAY

```

```

;=====

```

```

LJMP     MAIN

```

```

CHINESE1:

```

```

DB 0B2H, 0CAH, 0BEH, 0A7H, 0BFH, 0C6H, 0BCH, 0BCH, 0D3H, 0DOH, 0CFH, 0DEH, 0B9H, 0ABH, 0CBH, 0BEH

```

```

CHINESE2:

```

```

DB 0D6H, 0DOH, 0CEH, 0C4H, 0D7H, 0D6H, 0BFH, 0E2H, 0D2H, 0BAH, 0BEH, 0A7H, 0CFH, 0D4H, 0CAH, 0BEH

```

```

CHINESE3:

```

```

DB 0B2H, 0CAH, 0BEH, 0A7H, 0BFH, 0C6H, 0BCH, 0BCH, 0D3H, 0DOH, 0CFH, 0DEH, 0B9H, 0ABH, 0CBH, 0BEH

```

```

CHINESE4:

```

```

DB 0D6H, 0DOH, 0CEH, 0C4H, 0D7H, 0D6H, 0BFH, 0E2H, 0D2H, 0BAH, 0BEH, 0A7H, 0CFH, 0D4H, 0CAH, 0BEH

```

```

CHINESETAB0:

```

```

DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH

```

```

DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 030H, 000H, 004H, 002H, 004H, 008H, 020H, 002H, 000H, 000H, 020H, 002H, 000H, 000H, 083H

```

```

DB 0C7H, 0C0H, 081H, 0FEH, 007H, 004H, 008H, 020H, 002H, 008H, 0F7H, 0F0H, 012H, 003H, 0FFH, 0C3H

```

```

DB 0C0H, 091H, 001H, 004H, 03CH, 024H, 008H, 024H, 0FFH, 0FCH, 094H, 020H, 012H, 000H, 000H, 083H

```

```

DB 0C4H, 092H, 001H, 0FCH, 004H, 014H, 07DH, 0FEH, 004H, 000H, 097H, 0E0H, 021H, 000H, 004H, 083H

```

```

DB 0C2H, 024H, 001H, 004H, 03FH, 014H, 008H, 020H, 008H, 020H, 0A4H, 020H, 041H, 007H, 0FEH, 083H

```

```

DB 0C0H, 080H, 081H, 0FCH, 004H, 044H, 008H, 020H, 01FH, 0F0H, 0A4H, 020H, 080H, 080H, 000H, 083H

```

```

DB 0CFH, 0F9H, 000H, 000H, 004H, 024H, 00DH, 0FCH, 028H, 020H, 097H, 0E1H, 008H, 071H, 0F8H, 083H

```

```

DB 0C0H, 082H, 00FH, 09FH, 00EH, 024H, 018H, 084H, 04FH, 0E0H, 095H, 012H, 008H, 021H, 008H, 083H

```

```

DB 0C1H, 0CCH, 008H, 091H, 015H, 007H, 068H, 088H, 088H, 020H, 095H, 020H, 010H, 001H, 008H, 083H

```

```

DB 0C2H, 0A0H, 088H, 091H, 014H, 0FCH, 008H, 050H, 00FH, 0E0H, 0D5H, 040H, 012H, 001H, 0F8H, 083H

```

```

DB 0C2H, 091H, 00FH, 09FH, 024H, 004H, 008H, 020H, 008H, 020H, 0A4H, 080H, 021H, 001H, 008H, 083H

```

```

DB 0C4H, 082H, 008H, 091H, 004H, 004H, 008H, 050H, 008H, 020H, 084H, 040H, 040H, 080H, 000H, 083H

```

```

DB 0C8H, 08CH, 00FH, 09FH, 004H, 004H, 028H, 088H, 008H, 020H, 087H, 038H, 0FFH, 0C0H, 002H, 083H

```

```

DB 0C0H, 0B0H, 008H, 091H, 004H, 004H, 013H, 006H, 008H, 060H, 084H, 010H, 040H, 040H, 001H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

```

DB 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H

```

[illegible]

END