



中显液晶
技术资料



中显控制器使用说明书

2009年3月15日

北京市海淀区中关村大街32号和盛大厦811室
电话：(86)-010-52926620 传真：(86)-010-52926621
企业网站：<http://www.zxlcd.com>

HD61202 液晶显示控制驱动器的应用

HD61202 液晶显示控制驱动器是一种带有驱动输出的图形液晶显示控制器，它可直接与 8 位微处理器相联，它可与 HD61203 配合对液晶屏进行行、列驱动。本手册将有选择地介绍 HD61202，详细地叙述内置 HD61202 的液晶显示模块 12864 和 19264 的应用方法。

第一章 关于 HD61202 的一般介绍

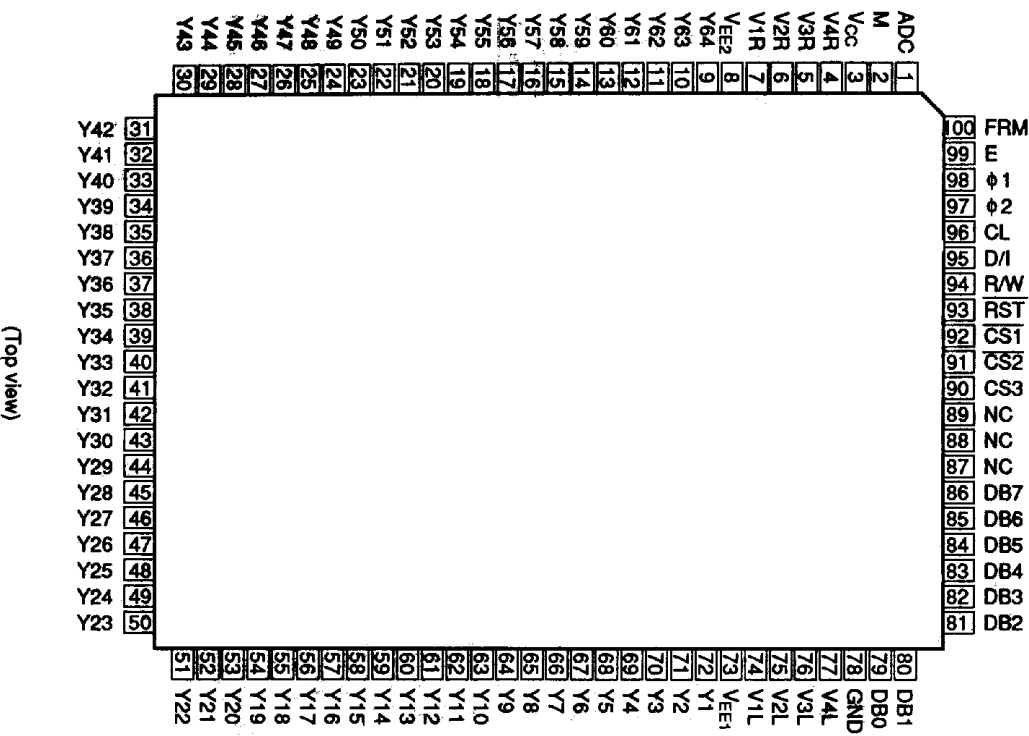
HD61202 是一种带有列驱动输出的液晶显示控制器，它可与行驱动器 HD61203 配合使用，组成液晶显示驱动控制系统。

一、HD61202 的特点

- 1. 内藏 $64 \times 64 = 4096$ 位显示 RAM，RAM 中每位数据对应 LCD 屏上一个点的亮、暗状态；
- 2. HD61202 是列驱动器，具有 64 路列驱动输出；
- 3. HD61202 读、写操作时序与 68 系列微处理器相符，因此它可直接与 68 系列微处理器接口相联；
- 4. HD61202 的占空比为 $1/32 \sim 1/64$ 。

二、HD61202 的引脚功能

HD61202 管脚图和功能原理图如图所示。



HD61202 的引脚功能如下：

引脚符号	状态	引脚名称	功 能
CS1,CS2, CS3	输入	芯片片选端	CS1 和 CS2 低电平选通，CS3 高电平选通
E	输入	读写使能信号	在 E 下降沿，数据被锁存(写)入 HD61202；在 E 高电平期间，数据被读出
R/W	输入	读写选择信号	R/W=1 为读选通， R/W=0 为写选通
D/I	输入	数据、指令选择信号	D/I=1 为数据操作， D/I=0 为写指令或读状态
DB0~DB7	三态	数据总线	
RST	输入	复位信号	复位信号有效时，关闭液晶显示，使显示起始行为 0。RST 可跟 MPU 相连，由 MPU 控制；也可直接接 Vcc，使之不起作用。

以上为与微处理器的接口信号。

引脚符号	状态	引脚名称	功 能
M	输入	交流驱动波形信号	
FRM	输入	帧同步信号	
CL	输入	锁存行显示数据的同步信号	该信号上升沿时锁存数据，同时改变显示输出地址
$\Phi 1, \Phi 2$	输入	内部操作时钟信号	

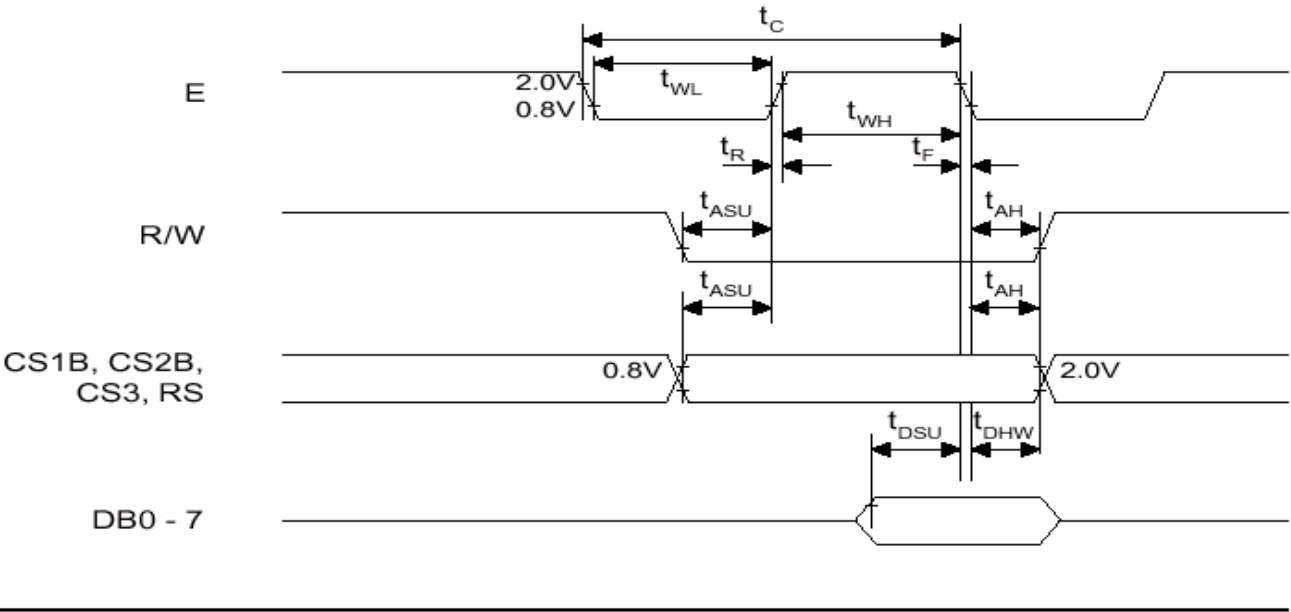
以上为与 HD61203 接口信号。

引脚符号	状态	引脚名称	功 能
Y1~Y64		液晶显示驱动端	
Vcc,GND		内部逻辑电源	
VEE1,VEE2		液晶显示驱动电路的电源	常令 VEE1=VEE2
V1L~V4L, V1R~V4R		液晶显示驱动电压	其电压值均在 Vcc 和 VEE 之间，常令 V1L=V1R,V2L=V2R, V3L=V3R,V4L=V4R
ADC		决定 Y1~Y64 与液晶屏的联接顺序	ADC=1 时，Y1= \$ 0，Y64= \$ 63 ADC=0 时，Y1= \$ 63，Y64= \$ 0 该引脚直接接 Vcc 或 GND 即可

以上为与 LCD 接口信号。

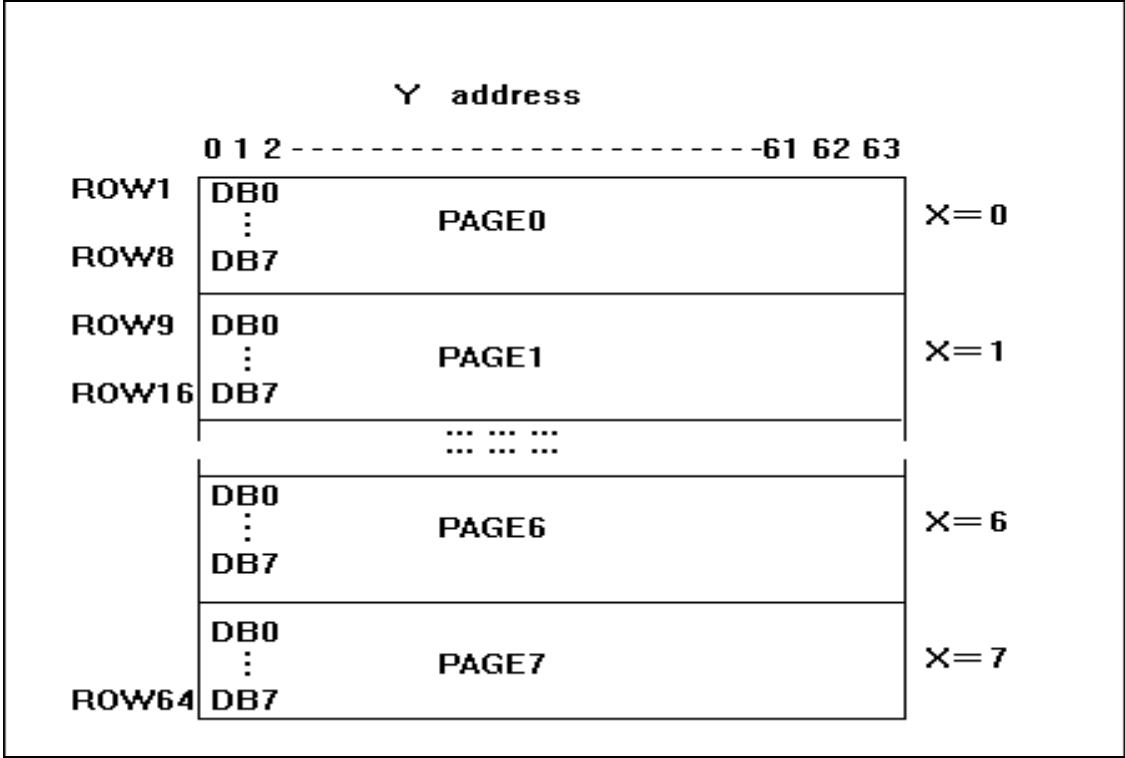
三、HD61202 的时序

HD61202 具有能与 68 系列微处理器直接接口的时序。各种信号波形对照如下。



参数项	符号	最小	典型	最大	单位
E 信号周期时间	tcyc	1000	-	-	ns
E 信号高电平宽度	Pweh	450	-	-	ns
E 信号低电平宽度	Pwel	450	-	-	ns
E 信号上升时间	tr	-	-	25	ns
E 信号下降时间	tf	-	-	25	ns
地址建立时间	tas	140	-	-	ns
地址保持时间	tah	10	-	-	ns
数据建立时间	tdsw	200	-	-	ns
数据保持时间	tdhw	10	-	-	ns
数据延时时间(读)	tddr	-	-	320	ns
数据保持时间(读)	tdhr	20	-	-	ns

四、HD61202 显示 RAM 的地址结构



五、HD61202 的指令系统

HD61202 的指令系统比较简单，总共只有七种。现分别介绍如下。

1、显示开/关指令

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	1/0

当 DB0=1 时，LCD 显示 RAM 中的内容；DB0=0 时，关闭显示。

2、显示起始行（ROW）设置指令

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	显示起始行（0~63）					

该指令设置了对应液晶屏最上一行的显示 RAM 的行号，有规律地改变显示起始行，可以使 LCD 实现显示滚屏的效果。

3、页（PAGE）设置指令

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	页号（0~7）		

显示 RAM 共 64 行，分 8 页，每页 8 行。

4、列地址（Y Address）设置指令

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	显示列地址（0~63）					

设置了页地址和列地址，就唯一确定了显示 RAM 中的一个单元，这样 MPU 就可以用读、写指令读出该单元中的内容或向该单元写进一个字节数据。

5、读状态指令

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	BUSY	0	ON/OFF	REST	0	0	0	0

该指令用来查询 HD61202 的状态，各参量含义如下：

- BUSY : 1-内部在工作 0-正常状态
- ON/OFF : 1-显示关闭 0-显示打开
- REST : 1-复位状态 0-正常状态

在 BUSY 和 REST 状态时，除读状态指令外，其它指令均不对 HD61202 产生作用。
在对 HD61202 操作之前要查询 BUSY 状态，以确定是否可以对 HD61202 进行操作。

6、写数据指令

R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	写 数 据							

7、读数据指令

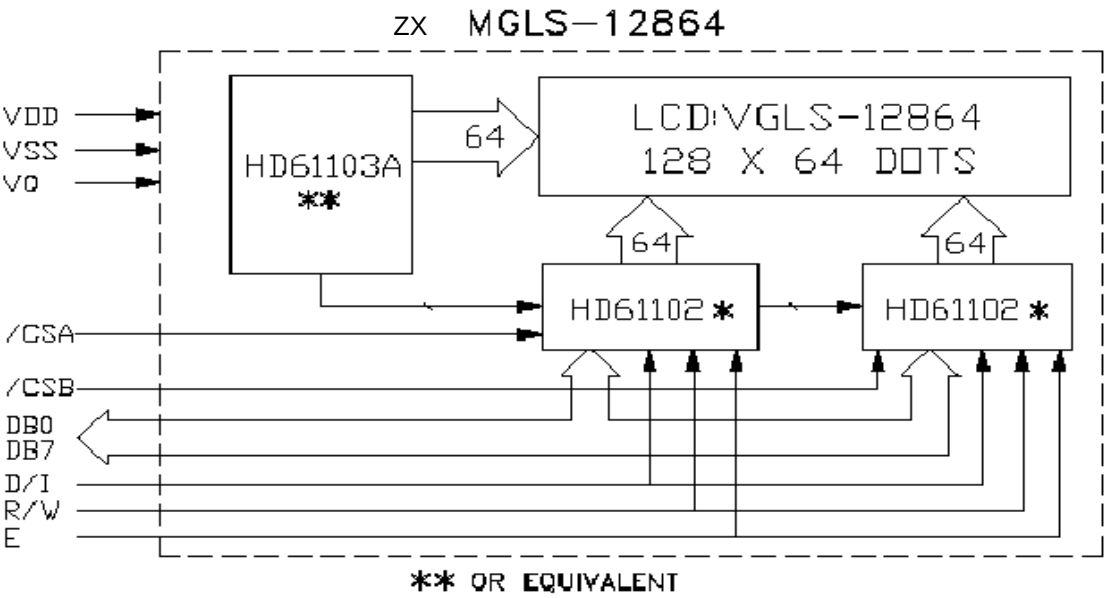
R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	读 显 示 数 据							

读、写数据指令每执行完一次读、写操作，列地址就自动增一。必须注意的是，进行读操作之前，必须有一次空读操作，紧接着再读才会读出所要读的单元中的数据。

第二章 ZXM-128641 和 ZXM-192641 的电路结构特点

ZX-128641 和 ZX--19264 均是使用 HD61202 作为列驱动器，同时使用 HD61203 作为行驱动器的液晶模块。由于 HD61203 不与 MPU 发生联系，只要提供电源就能产生行驱动信号和各种同步信号，比较简单，因此这里就不作介绍了。下面主要介绍一下 ZXM-128641和 ZX-19264 这两个模块的逻辑电路图 。

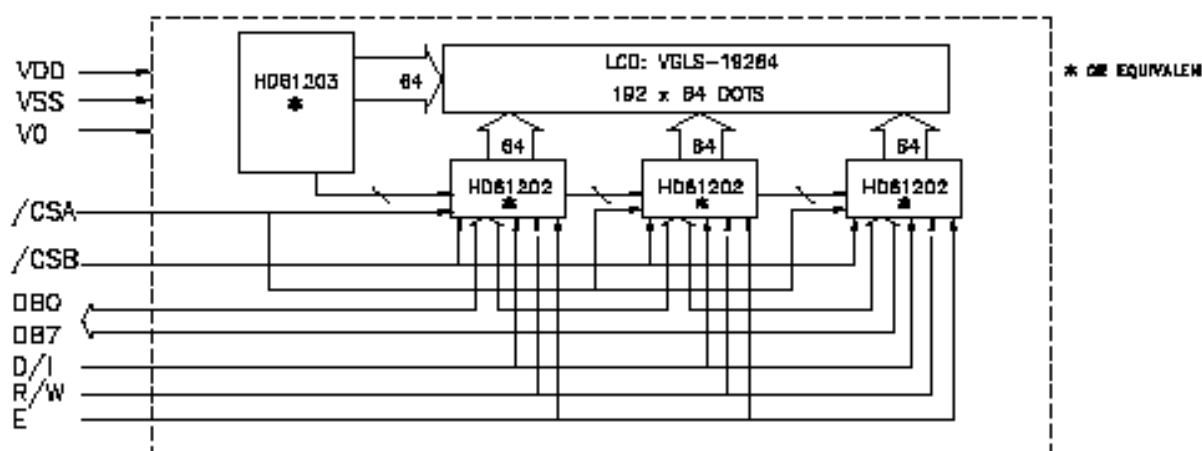
ZXM-12864 共有两片 HD61202 和一片 HD61203 ，如下图：



在 ZXM-12864 中，两片 HD61202 的 ADC 均接高电平，RST 也接高电平，这样在使用 ZXM-12864 时就不必再考虑这两个引脚的作用。/CSA 跟 HD61202(1)的 CS1 相连；/CSB 跟 HD61202(2)的 CS1 相连，因此 /CSA、/CSB 选通组合信号为 /CSA，/CSB=01 选通(1)，/CSA，/CSB=10 选通(2)。

ZXM-19264 中共有三片 HD61202 和一片 HD61203。HD61203 和三片 HD61202 之间的连接法也同 ZXM-12864 一样。三片 HD61202 中，引脚 ADC 和 RST 的接法也同 GXM-12864，所有在使用 ZXM-19264 时也不必考虑这两个引脚的影响。

ZXM-19264 也只有两个片选端 CSA 和 CSB 引出供 MPU 接口选通，如下图所示：



由图中可以看出，/CSA，/CSB 选通组合信号为 /CSA，/CSB=00 选通 HD61202(1)；/CSA，/CSB=01 选通 HD61202(2)；/CSA，/CSB=10 选通 HD61202(3)。

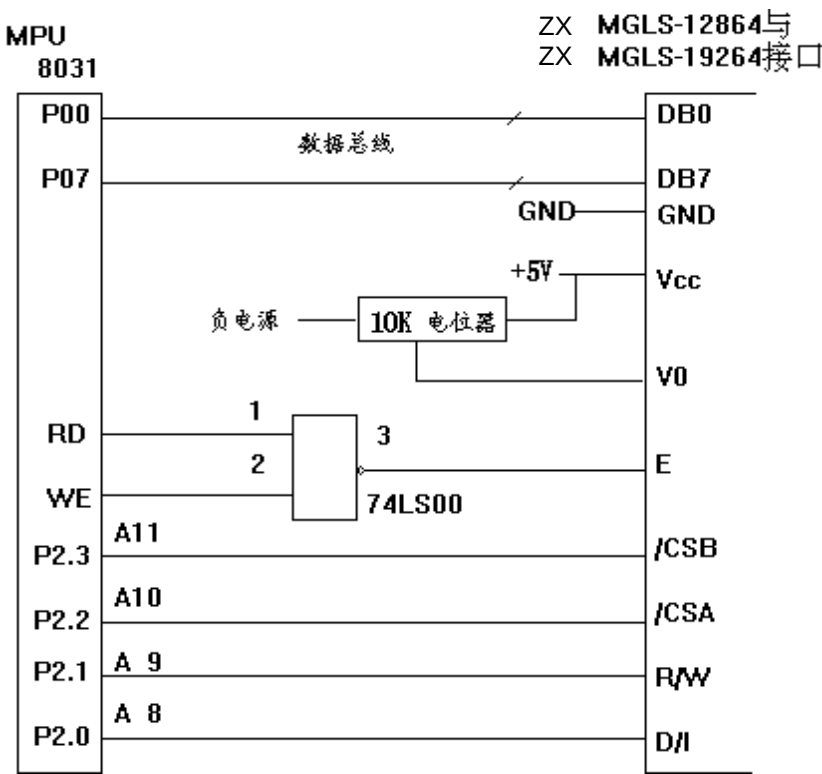
无论对于 ZXM-12864 还是对于 ZXM-19264，都只要供给 GND、Vcc 和 V0 即可，HD61202 和 HD61203 所需的电源将由模块内部电路在 Vcc 和 V0、GND 的作用下产生。

第三章 ZXM12864SL 和 ZXM19264SL 的应用

以下内容均以单片机 8031 为例机。

一、直接访问方式

(一) 接口电路



(二) 直接访问方式驱动子程序

A11 = /CSB, A10 = /CSA, A9 = R/W, A8 = A0			
COM	EQU	20H	; 指令寄存器
DAT	EQU	21H	; 数据寄存器
CWADD1	EQU	0000H	; 写指令代码地址(左)
CRADD1	EQU	0200H	; 读状态字地址(左)
DWADD1	EQU	0100H	; 写显示数据地址(左)
DRADD1	EQU	0300H	; 读显示数据地址(左)
CWADD2	EQU	0800H	; 写指令代码地址(中)
CRADD2	EQU	0A00H	; 读状态字地址(中)
DWADD2	EQU	0900H	; 写显示数据地址(中)
DRADD2	EQU	0B00H	; 读显示数据地址(中)
CWADD3	EQU	0400H	; 写指令代码地址(右)
CRADD3	EQU	0600H	; 读状态字地址(右)
DWADD3	EQU	0500H	; 写显示数据地址(右)
DRADD3	EQU	0700H	; 读显示数据地址(右)

1、写指令代码子程序(左)

PRL0: PUSH	DPL	; 片选设置为"00"
PUSH	DPH	

```

        MOV        DPTR,#CRADD1    ; 设置读状态字地址
PRL01: MOVX        A,@DPTR          ; 读状态字
        JB         ACC.7,PRL01      ; 判"忙"标志为"0"否,否再读
        MOV        DPTR,#CWADD1    ; 设置写指令代码地址
        MOV        A,COM            ; 取指令代码
        MOVX       @DPTR,A         ; 写指令代码
        POP        DPH
        POP        DPL
        RET

```

2、写显示数据子程序(左)

```

PRL1:  PUSH        DPL              ; 片选设置为"00"
        PUSH       DPH
        MOV        DPTR,#CRADD1    ; 设置读状态字地址

PRL11: MOVX        A,@DPTR          ; 读状态字
        JB         ACC.7,PRL11      ; 判"忙"标志为"0"否,否再读
        MOV        DPTR,#DWADD1    ; 设置写显示数据地址
        MOV        A,DAT           ; 取数据
        MOVX       @DPTR,A         ; 写数据
        POP        DPH
        POP        DPL
        RET

```

3、读显示数据子程序(左)

```

PRL2:  PUSH        DPL              ; 片选设置为"00"
        PUSH       DPH
        MOV        DPTR,#CRADD1    ; 设置读状态字地址

PRL21: MOVX        A,@DPTR          ; 读状态字
        JB         ACC.7,PRL21      ; 判"忙"标志为"0"否,否再读
        MOV        DPTR,#DRADD1    ; 设置读显示数据地址
        MOVX       A,@DPTR          ; 读数据
        MOV        DAT,A           ; 存数据
        POP        DPH
        POP        DPL
        RET

```

4、写指令代码子程序(中)

```

PRM0:  PUSH        DPL              ; 片选设置为"01"
        PUSH       DPH

```

MOV DPTR,#CRADD2 ; 设置读状态字地址

PRM01: MOVX A,@DPTR ; 读状态字
 JB ACC.7,PRM01 ; 判"忙"标志为"0"否,否再读
 MOV DPTR,#CWADD2 ; 设置写指令代码地址
 MOV A,COM ; 取指令代码
 MOVX @DPTR,A ; 写指令代码
 POP DPH
 POP DPL
 RET

5、写显示数据子程序(中)

PRM1: PUSH DPL ; 片选设置为"01"
 PUSH DPH
 MOV DPTR,#CRADD2 ; 设置读状态字地址

PRM11: MOVX A,@DPTR ; 读状态字
 JB ACC.7,PRM11 ; 判"忙"标志为"0"否,否再读
 MOV DPTR,#DWADD2 ; 设置写显示数据地址
 MOV A,DAT ; 取数据
 MOVX @DPTR,A ; 写数据
 POP DPH
 POP DPL
 RET

6、读显示数据子程序(中)

PRM2: PUSH DPL ; 片选设置为"01"
 PUSH DPH
 MOV DPTR,#CRADD2 ; 设置读状态字地址

PRM21: MOVX A,@DPTR ; 读状态字
 JB ACC.7,PRM21 ; 判"忙"标志为"0"否,否再读
 MOV DPTR,#DRADD2 ; 设置读显示数据地址
 MOVX A,@DPTR ; 读数据
 MOV DAT,A ; 存数据
 POP DPH
 POP DPL
 RET

7、写指令代码子程序(右)

PRR0: PUSH DPL ; 片选设置为"10"

	PUSH	DPH	
	MOV	DPTR,#CRADD3	; 设置读状态字地址
PRR01:	MOVX	A,@DPTR	; 读状态字
	JB	ACC.7,PRR01	; 判"忙"标志为"0"否,否再读
	MOV	DPTR,#CWADD3	; 设置写指令代码地址
	MOV	A,COM	; 取指令代码
	MOVX	@DPTR,A	; 写指令代码
	POP	DPH	
	POP	DPL	
	RET		

8、写显示数据子程序(右)

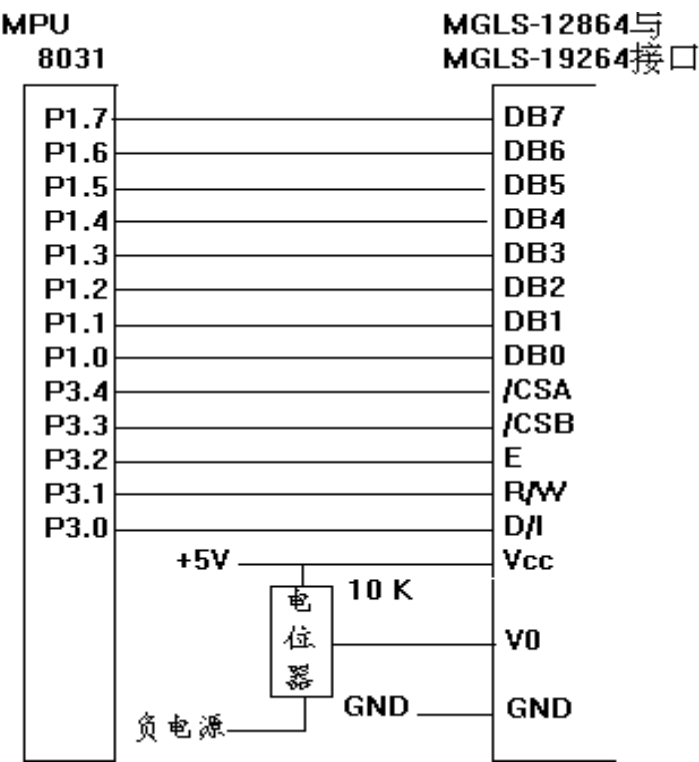
PRR1:	PUSH	DPL	; 片选设置为"10"
	PUSH	DPH	
	MOV	DPTR,#CRADD3	; 设置读状态字地址
PRR11:	MOVX	A,@DPTR	; 读状态字
	JB	ACC.7,PRR11	; 判"忙"标志为"0"否,否再读
	MOV	DPTR,#DWADD3	; 设置写显示数据地址
	MOV	A,DAT	; 取数据
	MOVX	@DPTR,A	; 写数据
	POP	DPH	
	POP	DPL	
	RET		

9、读显示数据子程序(右)

PRR2:	PUSH	DPL	; 片选设置为"10"
	PUSH	DPH	
	MOV	DPTR,#CRADD3	; 设置读状态字地址
PRR21:	MOVX	A,@DPTR	; 读状态字
	JB	ACC.7,PRR21	; 判"忙"标志为"0"否,否再读
	MOV	DPTR,#DRADD3	; 设置读显示数据地址
	MOVX	A,@DPTR	; 读数据
	MOV	DAT,A	; 存数据
	POP	DPH	
	POP	DPL	
	RET		

二、间接控制方式

(一) 接口电路



(二) 间接控制方式驱动子程序

CSA	EQU	P3.0	; 片选/CSA
CSB	EQU	P3.1	; 片选/CSB
D/I	EQU	P3.2	; 寄存器选择信号
R/W	EQU	P3.3	; 读/ 写选择信号
E	EQU	P3.4	; 使能信号

1、写指令代码子程序(左)

PRL0:	CLR	CSA	; 片选设置为"00"
	CLR	CSB	
	CLR	D/I	; D/I=0
	SETB	R/W	; R/W=1
PRL01:	MOV	P1,#0FFH	; P1 口置"1"
	SETB	E	; E=1
	MOV	A,P1	; 读状态字
	CLR	E	; E=0
	JB	ACC.7,PRL01	; 判"忙"标志为"0"否,否再读
	CLR	R/W	; R/W=0
	MOV	P1,COM	; 写指令代码
	SETB	E	; E=1
	CLR	E	; E=0

RET

2、写显示数据子程序(左)

```
PRL1:  CLR      CSA           ; 片选设置为"00"
        CLR      CSB
        CLR      D/I         ; D/I=0
        SETB     R/W         ; R/W=1

PRL11: MOV      P1,#0FFH      ; P1 口置"1"
        SETB     E            ; E=1
        MOV      A,P1         ; 读状态字
        CLR      E            ; E=0
        JB       ACC.7,PRL11   ; 判"忙"标志为"0"否,否再读
        SETB     D/I         ; D/I=1
        CLR      R/W         ; R/W=0
        MOV      P1,DAT       ; 写数据
        SETB     E            ; E=1
        CLR      E            ; E=0
        RET
```

3、读显示数据子程序(左)

```
PRL2:  CLR      CSA           ; 片选设置为"00"
        CLR      CSB
        CLR      D/I         ; D/I=0
        SETB     R/W         ; R/W=1

PRL21: MOV      P1,#0FFH      ; P1 口置"1"
        SETB     E            ; E=1
        MOV      A,P1         ; 读状态字
        CLR      E            ; E=0
        JB       ACC.7,PRL21   ; 判"忙"标志为"0"否,否再读
        SETB     D/I         ; D/I=1
        MOV      P1,#0FFH      ; P1 口置"1"
        SETB     E            ; E=1
        MOV      DAT,P1       ; 读数据
        CLR      E            ; E=0
        RET
```

4、写指令代码子程序(中)

```
PRM0:  CLR      CSA           ; 片选设置为"01"
        SETB     CSB
```

CLR	D/I	; D/I=0
SETB	R/W	; R/W=1
PRM01: MOV	P1,#0FFH	; P1 口置"1"
SETB	E	; E=1
MOV	A,P1	; 读状态字
CLR	E	; E=0
JB	ACC.7,PRM01	; 判"忙"标志为"0"否,否再读
CLR	R/W	; R/W=0
MOV	P1,COM	; 写指令代码
SETB	E	; E=1
CLR	E	; E=0
RET		

5、写显示数据子程序(中)

PRM1: CLR	CSA	; 片选设置为"01"
SETB	CSB	
CLR	D/I	; D/I=0
SETB	R/W	; R/W=1
PRM11: MOV	P1,#0FFH	; P1 口置"1"
SETB	E	; E=1
MOV	A,P1	; 读状态字
CLR	E	; E=0
JB	ACC.7,PRM11	; 判"忙"标志为"0"否,否再读
SETB	D/I	; D/I=1
CLR	R/W	; R/W=0
MOV	P1,DAT	;写数据
SETB	E	; E=1
CLR	E	; E=0
RET		

6、读显示数据子程序(中)

PRM2: CLR	CSA	; 片选设置为"01"
SETB	CSB	
CLR	D/I	; D/I=0
SETB	R/W	; R/W=1
PRM21: MOV	P1,#0FFH	; P1 口置"1"
SETB	E	; E=1
MOV	A,P1	;读状态字

CLR	E	; E=0
JB	ACC.7,PRM21	; 判"忙"标志为"0"否,否再读
SETB	D/I	; D/I=1
MOV	P1,#0FFH	; P1 口置"0"
SETB	E	; E=1
MOV	DAT,P1	;读数据
CLR	E	; E=0
RET		

7、写指令代码子程序(右)

PRR0:	SETB	CSA	; 片选设置为"10"
	CLR	CSB	
	CLR	D/I	; D/I=0
	SETB	R/W	; R/W=1
PRR01:	MOV	P1,#0FFH	; P1 口置"1"
	SETB	E	; E=1
	MOV	A,P1	; 读状态字
	CLR	E	; E=0
	JB	ACC.7,PRR01	; 判"忙"标志为"0"否,否再读
	CLR	R/W	; R/W=0
	MOV	P1,COM	;写指令代码
	SETB	E	; E=1
	CLR	E	; E=0
	RET		

8、写显示数据子程序(右)

PRR1:	SETB	CSA	; 片选设置为"10"
	CLR	CSB	
	CLR	D/I	; D/I=0
	SETB	R/W	; R/W=1
PRR11:	MOV	P1,#0FFH	; P1 口置"1"
	SETB	E	; E=1
	MOV	A,P1	; 读状态字
	CLR	E	; E=0
	JB	ACC.7,PRR11	; 判"忙"标志为"0"否,否再读
	SETB	D/I	; D/I=1
	CLR	R/W	; R/W=0
	MOV	P1,DAT	; 写数据
	SETB	E	; E=1
	CLR	E	; E=0

RET

9、读显示数据子程序(右)

```

PRR2:  SETB      CSA           ; 片选设置为"10"
        CLR      CSB
        CLR      D/I          ; D/I=0
        SETB     R/W          ; R/W=1

PRR21: MOV      P1,#0FFH      ; P1 口置"1"
        SETB     E            ; E=1
        MOV      A,P1         ; 读状态字
        CLR      E            ; E=0
        JB       ACC.7,PRR21   ; 判"忙"标志为"0"否,否再读
        SETB     D/I          ; D/I=1
        MOV      P1,#0FFH      ; P1 口置"0"
        SETB     E            ; E=1
        MOV      DAT,P1       ; 读数据
        CLR      E            ; E=0
        RET

```

三、应用子程序

在使用该程序之前应根据使用的系统调用相应的驱动子程序，修改口地址，若使用 ZX--S12864, 则将(左)PRLi 驱动子程序屏蔽。

1、初始化子程序

```

INT:    MOV      COM,#0C0H      ; 设置显示起始行为第一行
        LCALL    PRL0
        LCALL    PRM0
        LCALL    PRR0
        MOV      COM,#3FH      ; 开显示设置
        LCALL    PRL0
        LCALL    PRM0
        LCALL    PRR0
        RET

```

2、清显示 RAM 区(清屏)子程序

```

CLEAR:  MOV      R4,#00H        ; 页面地址暂存器设置
CLEAR1: MOV      A,R4
        ORL      A,#0B8H        ; "或"页面地址设置代码
        MOV      COM,A          ; 页面地址设置

```

```

        LCALL    PRL0
        LCALL    PRM0
        LCALL    PRR0
        MOV      COM,#40H      ; 列地址设置为"0"
        LCALL    PRL0
        LCALL    PRM0
        LCALL    PRR0
        MOV      R3,#40H      ; 一页清 64 个字节

CLEAR2:  MOV      DAT,#00H      ; 显示数据为"0"
        LCALL    PRL1
        LCALL    PRM1
        LCALL    PRR1
        DJNZ     R3,CLEAR2      ; 页内字节清零循环
        INC      R4            ; 页地址暂存器加 1
        CJNE     R4,#08H,CLEAR1 ; RAM 区清零循环
        RET

```

示例程序:

```

MAIN:  MOV      SP,#60H
        ANL      P3,#0E0H
        LCALL    INT
        LCALL    CLEAR

```

3、西文字符写入子程序

```

COLUMN    EQU     30H      ; 列地址寄存器(0-191)
PAGE      EQU     31H      ; 页地址寄存器 D2,D1,D0:页地址
                                ; D7: 字符体 D7=0 为 6x8 点阵
                                ; D7=1 为 8x8 点阵
CODE      EQU     32H      ; 字符代码寄存器
COUNT    EQU     33H      ; 计数器

CW_PR:  MOV      DPTR,#CTAB      ; 确定字符字模块首地址
        MOV      A,CODE          ; 取代码
        MOV      B,#08H          ; 字模块宽度为 8 个字节
        MUL      AB              ; 代码 x8
        ADD      A,DPL           ; 字符字模块首地址
        MOV      DPL,A          ; = 字模库首地址 + 代码 x 8
        MOV      A,B
        ADDC     A,DPH
        MOV      DPH,A
        MOV      CODE,#00H      ; 借用为间址寄存器

```

	MOV	A,PAGE	; 读页地址寄存器
	JB	ACC.7,CW_1	; 判字符体
	MOV	COUNT,#06H	; 6x8 点阵
	LJMP	CW_2	
CW_1:	MOV	COUNT,#08H	; 8x8 点阵
CW_2:	ANL	A,#07H	; 取页地址值
	ORL	A,#0B8H	; "或"页地址指令代码
	MOV	COM,A	; 写页地址指针
	LCALL	PRL0	
	LCALL	PRM0	
	LCALL	PRR0	
	MOV	A,COLUMN	; 读列地址寄存器
	CLR	C	
	SUBB	A,#40H	; 列地址-64
	JC	CW_3	; <0 为左屏显示区域
	MOV	COLUMN,A	
	SUBB	A,#40H	; 列地址-64
	JC	CW_21	; <0 为中屏显示区域
	MOV	COLUMN,A	; ≥0 为右屏显示区域
	MOV	A,PAGE	
	SETB	ACC.5	; 设置区域标志位.
	MOV	PAGE,A	; "00"为左,"01"为中,"10"为右
	LJMP	CW_3	
CW_21:	MOV	A,PAGE	
	SETB	ACC.4	;设置区域标志位
	MOV	PAGE,A	
CW_3:	MOV	COM,COLUMN	; 设置列地址值
	ORL	COM,#40H	; "或"列地址指令标志位
	MOV	A,PAGE	; 判区域标志以确定设置哪个控制器
	ANL	A,#30H	
	CJNE	A,#10H,CW_31	; "01"为中区
	LCALL	PRM0	
	LJMP	CW_4	
CW_31:	CJNE	A,#20H,CW_32	; "10"为右区
	LCALL	PRR0	
	LJMP	CW_4	
CW_32:	LCALL	PRL0	; "00"为左区

```

CW_4:  MOV      A,CODE           ; 取间址寄存器值
        MOVC     A,@A+DPTR       ; 取字符字模数据
        MOV      DAT,A          ; 写数据
        MOV      A,PAGE          ; 判区域标志
        ANL      A,#30H
        CJNE     A,#10H,CW_41    ; "01"为中区
        LCALL    PRM1
        LJMP     CW_5

CW_41:  CJNE     A,#20H,CW_42    ; "10"为右区
        LCALL    PRR1
        LJMP     CW_5

CW_42:  LCALL    PRL1           ; "00"为左区
CW_5:   INC      CODE           ; 间址加 1
        INC      COLUMN         ; 列地址加 1
        MOV      A,COLUMN        ; 判列地址是否超出区域范围
        CJNE     A,#40H,CW_6
CW_6:   JC       CW_9           ; 未超出则继续
        MOV      COLUMN,#00H
        MOV      A,PAGE          ; 超出则判在何区域
        JB       ACC.5,CW_9      ; 在右区域则退出
        JB       ACC.4,CW_61     ; 判在左或中区
        SETB     ACC.4          ; 在左区则转中区
        MOV      PAGE,A
        MOV      COM,#40H       ; 设置中区列地址为"0"
        LCALL    PRM0
        LJMP     CW_9

CW_61:  SETB     ACC.5          ; 在中区则转右区
        CLR      ACC.4
        MOV      PAGE,A
        MOV      COM,#40H       ; 设置右区列地址为"0"
        LCALL    PRR0

CW_9:   DJNZ     COUNT,CW_4     ; 循环
        RET

```

西文显示演示程序段

```

        MOV      PAGE,#05H      ; 6x8 点阵字体, 第 4 页
        MOV      COLUMN,#30H    ; 起始列为第 4 列
        MOV      CODE,#34H      ; 字符代码
        LCALL    CW_PR
        MOV      PAGE,#05H

```

```

MOV        COLUMN,#3CH
MOV        CODE,#45H
LCALL     CW_PR
MOV        PAGE,#05H
MOV        COLUMN,#48H
MOV        CODE,#4CH
LCALL     CW_PR
MOV        PAGE,#05H
MOV        COLUMN,#54H
MOV        CODE,#1AH
LCALL     CW_PR
MOV        R7,#00H
MOV        R6,#60H

LOOP: MOV   A,R7
MOV        DPTR,#TAB1
MOVC       A,@A+DPTR
MOV        CODE,A
MOV        PAGE,#85H      ; 8X8 点阵字体, 第 4 页
MOV        COLUMN,R6
LCALL     CW_PR
INC        R7
MOV        A,#06H
ADD        A,R6
MOV        R6,A
CJNE       R7,#08H,LOOP
SJMP      $

TAB1: DB 16H,12H,17H,18H,10H,18H,16H,16H

```

西文字符库

```

CTAB:DB 000H,000H,000H,000H,000H,000H,000H,000H      ;" "=00H
      DB 000H,000H,000H,04FH,000H,000H,000H,000H      ;"! "=01H
      DB 000H,000H,007H,000H,007H,000H,000H,000H      ;"'"=02H
      DB 000H,014H,07FH,014H,07FH,014H,000H,000H      ;"# "=03H
      DB 000H,024H,02AH,07FH,02AH,012H,000H,000H      ;"$ "=04H
      DB 000H,023H,013H,008H,064H,062H,000H,000H      ;"% "=05H
      DB 000H,036H,049H,055H,022H,050H,000H,000H      ;"& "=06H
      DB 000H,000H,005H,003H,000H,000H,000H,000H      ;"'"=07H

```

DB 000H,000H,01CH,022H,041H,000H,000H,000H	;"("=08H
DB 000H,000H,041H,022H,01CH,000H,000H,000H	;"")=09H
DB 000H,014H,008H,03EH,008H,014H,000H,000H	;"*=0AH
DB 000H,008H,008H,03EH,008H,008H,000H,000H	;"+=0BH
DB 000H,000H,050H,030H,000H,000H,000H,000H	;" =0CH
DB 000H,008H,008H,008H,008H,008H,000H,000H	;"-=0DH
DB 000H,000H,060H,060H,000H,000H,000H,000H	;"."=0EH
DB 000H,020H,010H,008H,004H,002H,000H,000H	;"/=0FH
DB 000H,03EH,051H,049H,045H,03EH,000H,000H	;"0"=10H
DB 000H,000H,042H,07FH,040H,000H,000H,000H	;"1"=11H
DB 000H,042H,061H,051H,049H,046H,000H,000H	;"2"=12H
DB 000H,021H,041H,045H,04BH,031H,000H,000H	;"3"=13H
DB 000H,018H,014H,012H,07FH,010H,000H,000H	;"4"=14H
DB 000H,027H,045H,045H,045H,039H,000H,000H	;"5"=15H
DB 000H,03CH,04AH,049H,049H,030H,000H,000H	;"6"=16H
DB 000H,001H,001H,079H,005H,003H,000H,000H	;"7"=17H
DB 000H,036H,049H,049H,049H,036H,000H,000H	;"8"=18H
DB 000H,006H,049H,049H,029H,01EH,000H,000H	;"9"=19H
DB 000H,000H,036H,036H,000H,000H,000H,000H	;":"=1AH
DB 000H,000H,056H,036H,000H,000H,000H,000H	;" ="=1BH
DB 000H,008H,014H,022H,041H,000H,000H,000H	;"<"=1CH
DB 000H,014H,014H,014H,014H,014H,000H,000H	;"="=1DH
DB 000H,000H,041H,022H,014H,008H,000H,000H	;">"=1EH
DB 000H,002H,001H,051H,009H,006H,000H,000H	;"?"=1FH
DB 000H,032H,049H,079H,041H,03EH,000H,000H	;"@ "=20H
DB 000H,07EH,011H,011H,011H,07EH,000H,000H	;"A"=21H
DB 000H,041H,07FH,049H,049H,036H,000H,000H	;"B"=22H
DB 000H,03EH,041H,041H,041H,022H,000H,000H	;"C"=23H
DB 000H,041H,07FH,041H,041H,03EH,000H,000H	;"D"=24H
DB 000H,07FH,049H,049H,049H,049H,000H,000H	;"E"=25H
DB 000H,07FH,009H,009H,009H,001H,000H,000H	;"F"=26H
DB 000H,03EH,041H,041H,049H,07AH,000H,000H	;"G"=27H
DB 000H,07FH,008H,008H,008H,07FH,000H,000H	;"H"=28H
DB 000H,000H,041H,07FH,041H,000H,000H,000H	;"I"=29H
DB 000H,020H,040H,041H,03FH,001H,000H,000H	;"J"=2AH
DB 000H,07FH,008H,014H,022H,041H,000H,000H	;"K"=2BH
DB 000H,07FH,040H,040H,040H,040H,000H,000H	;"L"=2CH
DB 000H,07FH,002H,00CH,002H,07FH,000H,000H	;"M"=2DH
DB 000H,07FH,006H,008H,030H,07FH,000H,000H	;"N"=2EH
DB 000H,03EH,041H,041H,041H,03EH,000H,000H	;"O"=2FH
DB 000H,07FH,009H,009H,009H,006H,000H,000H	;"P"=30H

DB 000H,03EH,041H,051H,021H,05EH,000H,000H	;"Q"=31H
DB 000H,07FH,009H,019H,029H,046H,000H,000H	;"R"=32H
DB 000H,026H,049H,049H,049H,032H,000H,000H	;"S"=33H
DB 000H,001H,001H,07FH,001H,001H,000H,000H	;"T"=34H
DB 000H,03FH,040H,040H,040H,03FH,000H,000H	;"U"=35H
DB 000H,01FH,020H,040H,020H,01FH,000H,000H	;"V"=36H
DB 000H,07FH,020H,018H,020H,07FH,000H,000H	;"W"=37H
DB 000H,063H,014H,008H,014H,063H,000H,000H	;"X"=38H
DB 000H,007H,008H,070H,008H,007H,000H,000H	;"Y"=39H
DB 000H,061H,051H,049H,045H,043H,000H,000H	;"Z"=3AH
DB 000H,000H,07FH,041H,041H,000H,000H,000H	;"["=3BH
DB 000H,002H,004H,008H,010H,020H,000H,000H	;"\"=3CH
DB 000H,000H,041H,041H,07FH,000H,000H,000H	;"]="=3DH
DB 000H,004H,002H,001H,002H,004H,000H,000H	;"^"=3EH
DB 000H,040H,040H,040H,040H,040H,000H,000H	;"_ "=3FH
DB 000H,001H,002H,004H,000H,000H,000H,000H	;"`"=40H
DB 000H,020H,054H,054H,054H,078H,000H,000H	;"a"=41H
DB 000H,07FH,048H,044H,044H,038H,000H,000H	;"b"=42H
DB 000H,038H,044H,044H,044H,028H,000H,000H	;"c"=43H
DB 000H,038H,044H,044H,048H,07FH,000H,000H	;"d"=44H
DB 000H,038H,054H,054H,054H,018H,000H,000H	;"e"=45H
DB 000H,000H,008H,07EH,009H,002H,000H,000H	;"f"=46H
DB 000H,00CH,052H,052H,04CH,03EH,000H,000H	;"g"=47H
DB 000H,07FH,008H,004H,004H,078H,000H,000H	;"h"=48H
DB 000H,000H,044H,07DH,040H,000H,000H,000H	;"i"=49H
DB 000H,020H,040H,044H,03DH,000H,000H,000H	;"j"=4AH
DB 000H,000H,07FH,010H,028H,044H,000H,000H	;"k"=4BH
DB 000H,000H,041H,07FH,040H,000H,000H,000H	;"l"=4CH
DB 000H,07CH,004H,078H,004H,078H,000H,000H	;"m"=4DH
DB 000H,07CH,008H,004H,004H,078H,000H,000H	;"n"=4EH
DB 000H,038H,044H,044H,044H,038H,000H,000H	;"o"=4FH
DB 000H,07EH,00CH,012H,012H,00CH,000H,000H	;"p"=50H
DB 000H,00CH,012H,012H,00CH,07EH,000H,000H	;"q"=51H
DB 000H,07CH,008H,004H,004H,008H,000H,000H	;"r"=52H
DB 000H,058H,054H,054H,054H,064H,000H,000H	;"s"=53H
DB 000H,004H,03FH,044H,040H,020H,000H,000H	;"t"=54H
DB 000H,03CH,040H,040H,03CH,040H,000H,000H	;"u"=55H
DB 000H,01CH,020H,040H,020H,01CH,000H,000H	;"v"=56H
DB 000H,03CH,040H,030H,040H,03CH,000H,000H	;"w"=57H
DB 000H,044H,028H,010H,028H,044H,000H,000H	;"x"=58H
DB 000H,01CH,0A0H,0A0H,090H,07CH,000H,000H	;"y"=59H

```

DB 000H,044H,064H,054H,04CH,044H,000H,000H      ;"z"=5AH
DB 000H,000H,008H,036H,041H,000H,000H,000H      ;"{"=5BH
DB 000H,000H,000H,077H,000H,000H,000H,000H      ;"|"=5CH
DB 000H,000H,041H,036H,008H,000H,000H,000H      ;"}"=5DH
DB 000H,002H,001H,002H,004H,002H,000H,000H      ;"~"=5FH
DB 000H,0FFH,0FFH,0FFH,0FFH,0FFH,000H,000H      ;" "=60H

```

4、中文字符写入子程序

```

        COLUMN    EQU        30H      ; 列地址寄存器(0-191)
        PAGE      EQU        31H      ; 页地址寄存器 D1,D0: 页地址
        CODE      EQU        32H      ; 字符代码寄存器
        COUNT     EQU        33H      ; 计数器

CCW_PR:  MOV        DPTR,#CCTAB      ; 确定字符字模块首地址
        MOV        A,CODE            ; 取代码
        MOV        B,#20H            ; 字模块宽度为 32 个字节
        MUL        AB                ; 代码 x32
        ADD        A,DPL              ; 字符字模块首地址
        MOV        DPL,A              ; = 字模库首地址 + 代码 x 32
        MOV        A,B
        ADDC       A,DPH
        MOV        DPH,A
        PUSH       COLUMN             ; 列地址入栈
        PUSH       COLUMN             ; 列地址入栈
        MOV        CODE,#00H         ; 代码寄存器借用为间址寄存器

CCW_1:  MOV        COUNT,#10H         ; 计数器设置为 16
        MOV        A,PAGE             ; 读页地址寄存器
        ANL        A,#07H
        ORL        A,#0B8H           ; "或"页地址设置代码
        MOV        COM,A              ; 写页地址设置指令
        LCALL      PRL0
        LCALL      PRM0
        LCALL      PRR0
        POP        COLUMN             ; 取列地址值
        MOV        A,COLUMN           ; 读列地址寄存器
        CLR        C
        SUBB       A,#40H             ; 列地址-64
        JC         CCW_2              ; <0 为左屏显示区域
        MOV        COLUMN,A
        SUBB       A,#40H             ; 列地址-64

```

	JC	CCW_11	; <0 为中屏显示区域
	MOV	COLUMN,A	; ≥0 为右屏显示区域
	MOV	A,PAGE	
	SETB	ACC.5	; 设置区域标志位.
	MOV	PAGE,A	; "00"为左,"01"为中,"10"为右
	LJMP	CCW_2	
CCW_11:	MOV	A,PAGE	
	SETB	ACC.4	;设置区域标志位
	MOV	PAGE,A	
CCW_2:	MOV	COM,COLUMN	; 设置列地址值
	ORL	COM,#40H	; "或"列地址指令标志位
	MOV	A,PAGE	; 判区域标志以确定设置哪个控制器
	ANL	A,#30H	
	CJNE	A,#10H,CCW_31	; "01"为中区
	LCALL	PRM0	
	LJMP	CCW_4	
CCW_31:	CJNE	A,#20H,CCW_32	; "10"为右区
	LCALL	PRR0	
	LJMP	CCW_4	
CCW_32:	LCALL	PRL0	; "00"为左区
CCW_4:	MOV	A,CODE	; 取间址寄存器值
	MOVC	A,@A+DPTR	; 取汉字字模数据
	MOV	DAT,A	; 写数据
	MOV	A,PAGE	; 判区域标志
	ANL	A,#30H	
	CJNE	A,#10H,CCW_41	; "01"为中区
	LCALL	PRM1	
	LJMP	CCW_5	
CCW_41:	CJNE	A,#20H,CCW_42	; "10"为右区
	LCALL	PRR1	
	LJMP	CCW_5	
CCW_42:	LCALL	PRL1	; "00"为左区
CCW_5:	INC	CODE	; 间址寄存器加 1
	INC	COLUMN	; 列地址寄存器加 1
	MOV	A,COLUMN	; 判列地址是否超出区域范围
	CJNE	A,#40H,CCW_6	
CCW_6:	JC	CCW_7	; 未超出则继续
	MOV	COLUMN,#00H	
	MOV	A,PAGE	; 超出则判在何区域
	JB	ACC.5,CCW_9	; 在右区域则退出
	JB	ACC.4,CCW_61	; 判在左或中区

	SETB	ACC.4	; 在左区则转中区
	MOV	PAGE,A	
	MOV	COM,#40H	; 设置中区列地址为"0"
	LCALL	PRM0	
	LJMP	CCW_7	
CCW_61:	SETB	ACC.5	; 在中区则转右区
	CLR	ACC.4	
	MOV	PAGE,A	
	MOV	COM,#40H	; 设置右区列地址为"0"
	LCALL	PRR0	
CCW_7:	DJNZ	COUNT,CCW_4	; 当页循环
	MOV	A,PAGE	; 读页地址寄存器
	JB	ACC.7,CCW_9	; 判完成标志 D7 位, "1"则完成退出
	INC	A	; 否则页地址加 1
	SETB	ACC.7	; 置完成位为"1"
	ANL	A,#0CFH	; 清区域标志
	MOV	PAGE,A	
	MOV	CODE,#10H	; 间址寄存器设置为 16
	LJMP	CCW_1	; 大循环
CCW_9:	RET		

中文演示显示程序段

MOV	PAGE,#02H
MOV	COLUMN,#35H
MOV	CODE,#00H
LCALL	CCW_PR
MOV	PAGE,#02H
MOV	COLUMN,#4BH
MOV	CODE,#01H
LCALL	CCW_PR
MOV	PAGE,#02H
MOV	COLUMN,#63H
MOV	CODE,#02H
LCALL	CCW_PR
MOV	PAGE,#02H
MOV	COLUMN,#7BH
MOV	CODE,#03H
LCALL	CCW_PR
SJMP	\$

CCTAB: DB 082H,08AH,092H,0B2H,002H,0A7H,092H,05EH
DB 02AH,0AFH,02AH,05AH,04AH,083H,082H,000H
DB 000H,080H,040H,03FH,041H,090H,095H,095H
DB 095H,0FFH,095H,095H,095H,0D0H,040H,000H
DB 080H,064H,024H,024H,03FH,024H,0A6H,064H
DB 020H,020H,0FFH,020H,020H,0F0H,020H,000H
DB 000H,008H,049H,089H,07DH,00BH,089H,040H
DB 030H,00EH,041H,080H,040H,03FH,000H,000H
DB 040H,042H,044H,0CCH,000H,020H,022H,0E2H
DB 022H,022H,0E2H,023H,032H,020H,000H,000H
DB 000H,040H,020H,01FH,020H,050H,04CH,043H
DB 040H,040H,047H,048H,048H,06EH,020H,000H
DB 020H,020H,020H,020H,020H,020H,0A0H,07FH
DB 0A0H,020H,020H,020H,020H,030H,020H,000H
DB 000H,040H,040H,020H,010H,00CH,003H,000H
DB 001H,006H,008H,010H,020H,060H,020H,000H