

SED1520 控制器的使用方法

第一章 SED1520 功能介绍

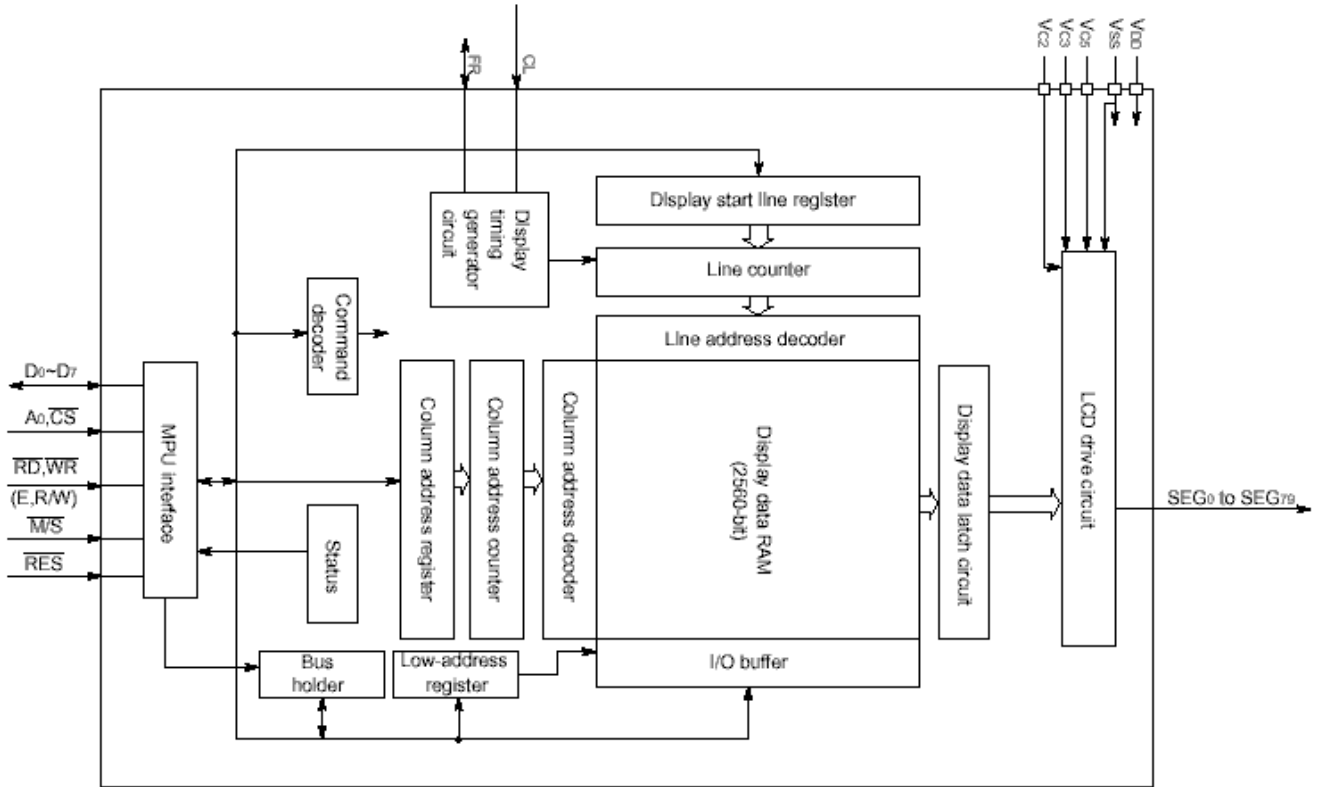
一、SED1520 的特性

- (1) 内置 2560 位显示 RAM 区。RAM 中的 1 位数据控制液晶屏上一个像素的亮、暗状态：“1”表示亮，“0”表示暗；
- (2) 具有 16 个行驱动输出和 61 个列驱动输出；
- (3) 可直接与 80 系列为微处理器相连，亦可直接与 68 系列微处理器相连；
- (4) 驱动占空比为 1/16 或 1/32；
- (5) 可以与 SED1520 级联使用，以便扩展行、列驱动能力。

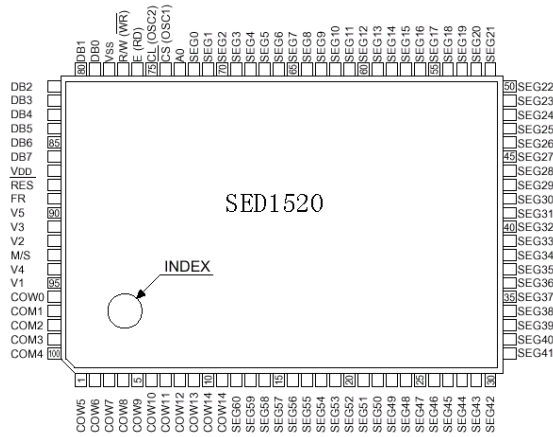
二、SED1520 的管脚功能介绍

SED1520 功能原理图和管脚图分别见图一和图二。

SED1520 有两种类型，即 SED1520FAA 和 SED1520FOA，其引脚定义基本相同，区别在于后者内部具有 18K Hz 时钟发生器，因此不需要外部提供时钟发生器即可工作。



图一 SED1520 原理框图



1520F0A与FAA不同点

型号	管脚	
	74	75
SED1520F0A	OSC1	OSC2
SED1520FAA	CS	CL

图二 SED1520 管脚图

三、SED1520 电特性

绝对最大范围 (VDD=0V, VDD ≥ V1 ≥ V2 ≥ V3 ≥ V4 ≥ V5)

参数	符号	范围	单位
电源电压 1	V _{SS}	- 8.0~0.3	V
电源电压 2	V ₅	-16.5~0.3	V
电源电压 3	V1-V4	V ₅ - 0.3	V
输入电压	V _i	V _{SS} - 0.3~0.3	V
输出电压	V _o	V _{SS} - 0.3~0.3	V
功耗	PD	250	mW
工作温度	TOPR	- 30~85	°C
储存温度	TSTG	- 65~85	°C
焊接温度	TSOL	- 260°C/10 s	

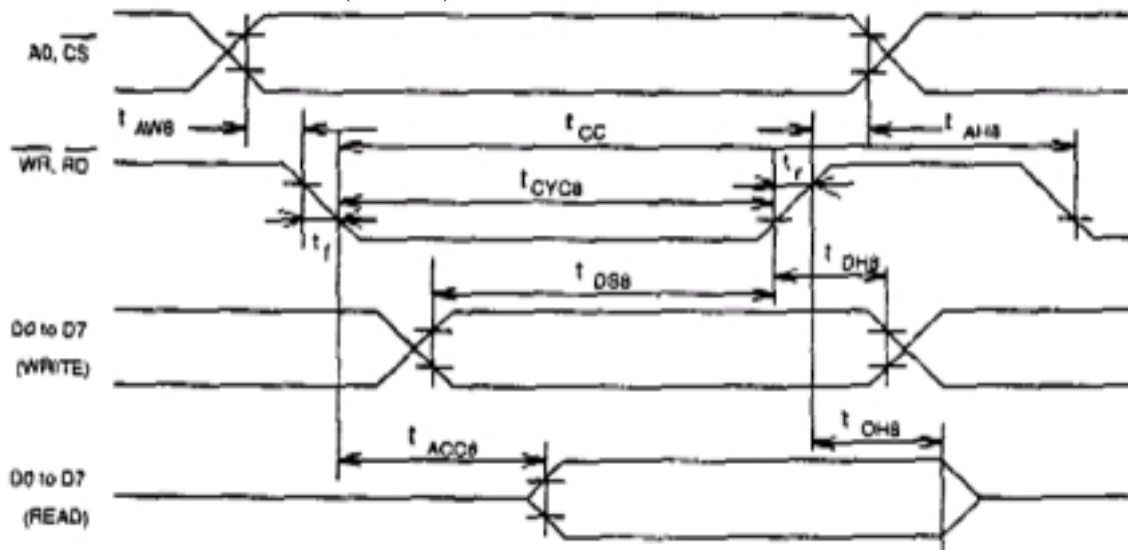
电特性 (VDD = 0 v Ta = -20~75°C)

参数	符号	最小	典型	最大	单位
工作电压 1	V _{SS}	- 5.5	- 5.0	- 4.5	V
工作电压 2	V ₅	-13.0		- 3.5	V
输入电压(H)	V _{iH}	V _{SS} +2.0		VDD	V
输入电压(L)	V _{iL}	V _{SS}		V _{SS} +0.8	V
输出电压(H)	V _{oH}	V _{SS} +2.4		VDD	V
输出电压(L)	V _{oL}			V _{SS} +0.4	V
输入漏电流	I _{Li}	- 1.0		1.0	uA
输出漏电流	I _{Lo}	- 3.0		3.0	uA
无驱动耗电	I _{DDO}		0.05	1.0	uA
工作耗电流	I _{DD}		5.0	10.0	uA
振荡频率	F _{osc}	15	18	21	KHz
复位时间	T _R	1.0		1000	us

四、SED1520 的读/写操作时序

SED1520 有两种读/写操作时序，分别对应于 6800 系列 MPU 和 8080 系列 MPU。

(1) 8080 系列 MPU 操作时序(RES=0)如下图。



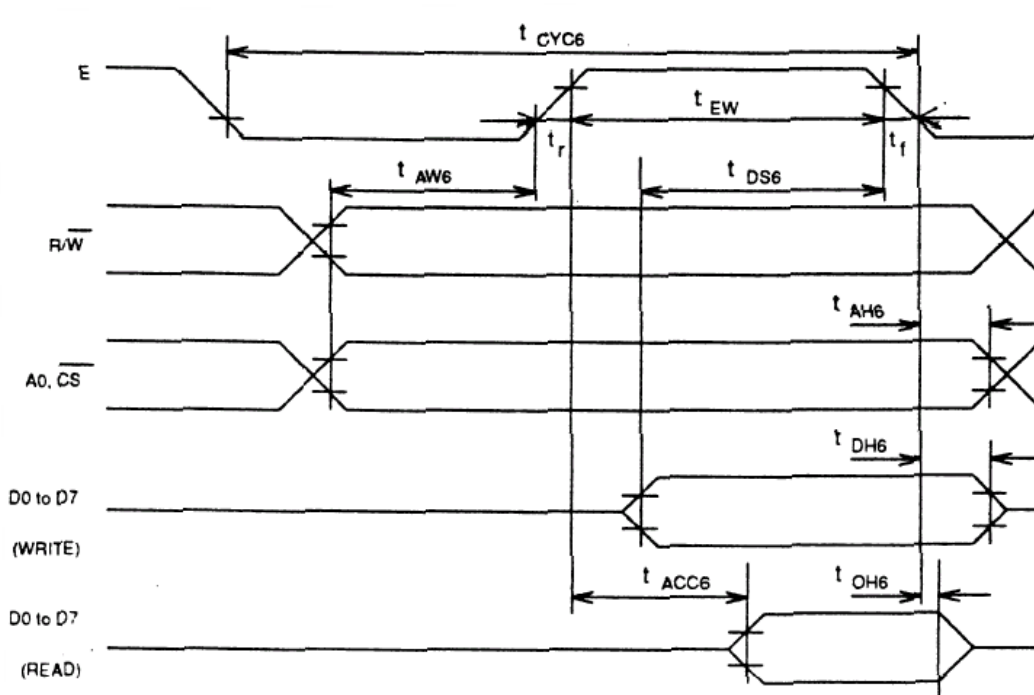
图三 SED1520 时序图一

在 WR 或 RD 后沿处，数据被写进或读出。

时序特性：

参数	符号	最小值	典型值	最大值	单位
地址保持时间	t_{AH8}	10	-	-	ns
地址建立时间	t_{AW8}	20	-	-	ns
系统周期时间	t_{cyc8}	1000	-	-	ns
控制脉冲宽度	t_{CC}	200	-	-	ns
数据建立时间（写）	t_{DS8}	80	-	-	ns
数据保持时间（写）	t_{DH8}	10	-	-	ns
数据读取时间（读）	t_{ACC8}	-	-	90	ns
数据保持时间（读）	t_{OH8}	10	-	60	Ns

(2) 6800 系列 MPU 操作时序(RES=1)如下图。



图四 SED1520 时序图二

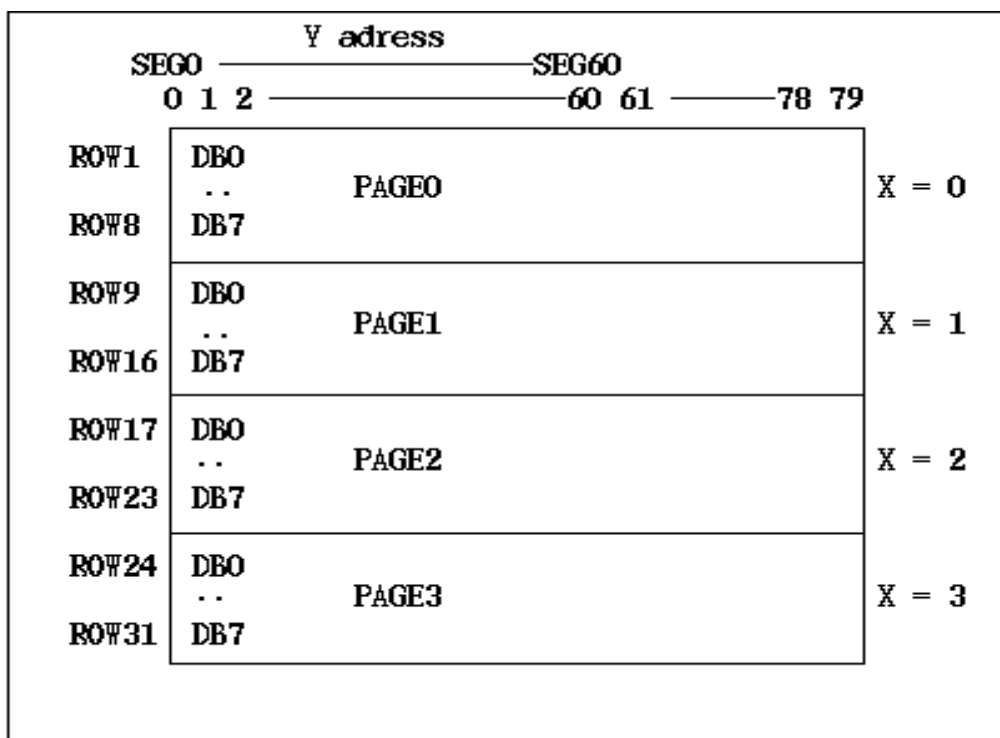
在 E 下降沿处，数据或指令被锁存进 SED1520 内部的数据或指令寄存器。在 E 高电平期间，数据或状态字被读到数据总线之上。

时序特性:

参数	符号	最小值	典型值	最大值	单位
系统周期时间	t _{cy6}	1000	-	-	ns
地址建立时间	t _{AW6}	20	-	-	ns
地址保持时间	t _{AH6}	10	-	-	ns
数据建立时间 (写)	t _{DS6}	80	-	-	ns
数据保持时间 (写)	t _{DH6}	10	-	-	ns
数据保持时间 (读)	t _{OH6}	10	-	60	ns
数据读取时间 (读)	t _{ACC6}	-	-	90	ns
使能脉冲宽度 (读)	t _{EW}	100	-	-	ns
使能脉冲宽度 (写)	t _{EW}	80	-	-	ns

五、SED1520 显示 RAM 的结构

SED1520 中的显示 RAM 共有 32 行 80 列，其结构如图五所示：



图五 SED1520 显示 RAM 结构图

第二章 指令系统

SED1520 液晶显示驱动器共有 13 种显示指令, 下面以与 68 系列 MPU 接口为例(RES=1), 分别介绍一下这 13 种指令。

1、读状态字

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	BUSY	ADC	OFF/ON	RESET	0	0	0	0

当 SED1520 处于“忙”状态时, 除了读状态指令, 其它指令均不起任何作用, 因此在访问 SED1520, 都要先读一下状态, 判断是否“忙”。

BUSY: 1:忙状态 0:准备好状态

ADC: 1:正常输出(右向) 0:反向输出(左向)(具体见 ADC 选择)

OFF/ON: 1:显示关闭 0:显示打开

RESET: 1:复位状态 0:正常状态

2、复位

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	1	0	0	0	1	0

该指令为软件复位指令, 执行该指令后, 使显示起始行置为第 0 行, 列地址置为 0, 页地址置为 3。

3、占空比选择

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	1	0	0	0/1

D0 = 0 占空比为 1/16, D0 = 1 为 1/32。
 驱动 32 行液晶显示时, 使 D0 为 1; 驱动 16 行时 D0 为 0。

4、显示起始行设置

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	显示起始行(0~31)				

该指令设置了对应显示屏上首行的显示 RAM 中的行号。有规律地修改该行号, 可实现滚屏功能。

5、休闲工作状态设置

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	0	1	0	0/1

该指令用软件停止 SED1520 的 LCD 驱动的输出。使系统处于低功耗休闲状态。休闲指令须在关显示状态下输入。

D0 = 1 为休闲工作状态, D0 = 0 为正常工作状态。

6、ADC 选择指令

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	0	0	0	0/1

该指令用来设置列驱动输出端与液晶显示屏的列驱动线的连接方式。应根据厂方提供的模块实际接线设置，一般设为 ADC = 0。

7、显示开/关指令

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	1	1	1	1/0

D0 = 1 为开显示；D0 = 0 为关显示。
该指令不影响显示 RAM 内容。

8、设置页地址

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	1	1	0	页地址(0-3)	

9、设置列地址

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0 列地址 (0~79)							

由上图看出，显示 RAM 被分成四页每页 80 个字节，当设置了页地址和列地址后就确定了显示 RAM 中的唯一单元，该单元由低到高的各个数据位对应于显示屏上某一列的 8 行数据位。

10、改写方式设置指令

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	1	0	0	0	0	0

该指令发出后，使得每次写数据后列地址自动增 1，而读数据后列地址仍保持原值不变。这种称为“改写模式”(Read Modify Write)的方式为逐个读取像素点状态进行修改提供了方便。

在改写方式中，SED1520 不接受任何指令代码（除改写方式结束指令）。

11、改写方式结束指令

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	1	0	1	1	1	0

该指令执行后，将结束改写方式，以后无论读或写数据后，列地址都增 1。

12、写数据

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	显示数据							

13、读数据

R/W	A0	D7	D6	D5	D4	D3	D2	D1	D0
1	1	显示数据							

正常状态下，写数据或读数据后，列地址将自动增 1。

第三章 液晶模块 GXM-12232-1SL 和 GXM-12232-4SL 的电路结构特点

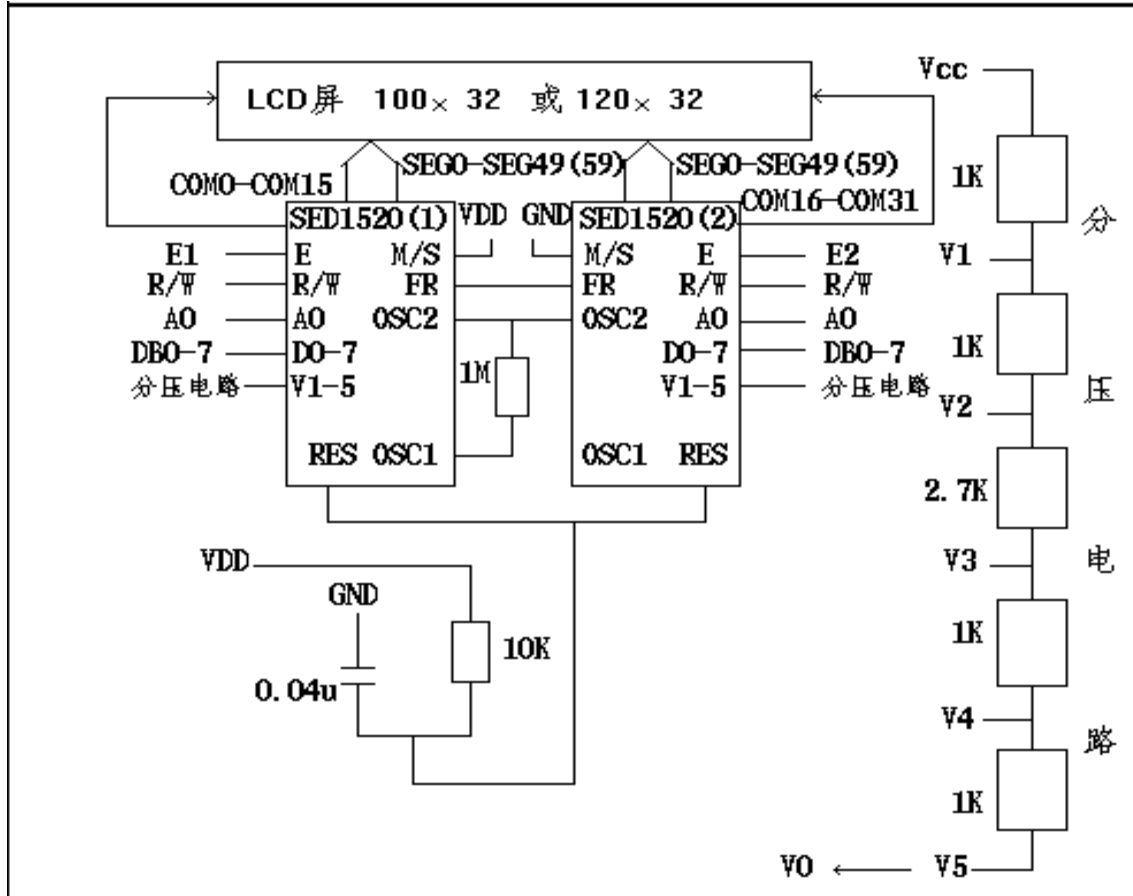
在 SED1520 应用的液晶模块中，都采用了 SED1520FOA，所以其时钟信号由内部产生。GM-12321SILY 和 GXM12232-4SL这两种液晶模块都是由两片 SED1520 来驱动的。

■ GXM12232-01SL PIN CONNECTIONS

Pin NO.	Symbol	Level	Description
1	VDD	5.0V	Supply voltage for logic
2	VSS	0V	Ground
3	VLCD	---	Operating voltage for LCD
4	RST	L → H	Reset the system
5	E1	H	Read/Write enable signal(Master)
6	E2	H	Read/Write enable signal(Slave)
7	R/W	H/L	H : read singal, L : write singal
8	A0	H/L	H : Data singal, L : Instruction singal
9	DB0	H/L	Data bit 0
10	DB1	H/L	Data bit 1
11	DB2	H/L	Data bit 2
12	DB3	H/L	Data bit 3
13	DB4	H/L	Data bit 4
14	DB5	H/L	Data bit 5
15	DB6	H/L	Data bit 6
16	DB7	H/L	Data bit 7
17	ELA	AC	EL light anode
18	ELK	0V	EL light cathode

由图六可以看出，SED1520(1)为主方式工作，OSC1 和 OSC2 跨接一个电阻，并由 OSC2 输出时序信号；SED1520(2)为从方式工作，时钟由 SED1520FOA(1)提供，从 OSC2 输入。两个芯片的 RES 上电后经一定时间升为高电平，所以其控制时序为 68 系列 MPU 的时序。两个芯片的选通由 E1，E2 信号决定。

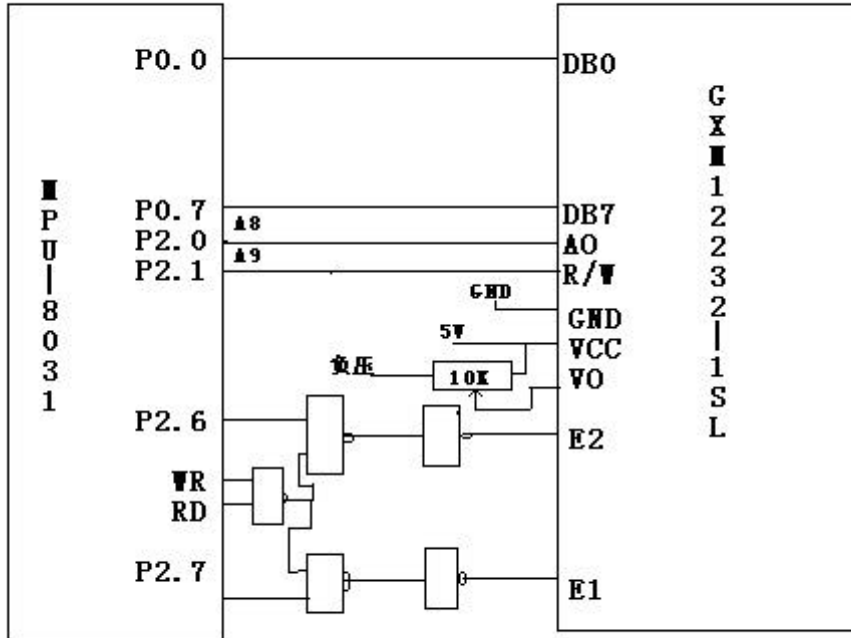
对于这几种模块，ADC 选择应设置为 0，占空比设置成 1/32。



图六 模块原理图

第四章 液晶模块 GXM12232SL和 GXM-12232-4的应用

1、直接访问方式接口电路及驱动程序



直接访问方式驱动子程序如下:

```

COM      EQU      20H          ; 指令寄存器
DAT      EQU      21H          ; 数据寄存器
CWADD1   EQU      8000H        ; 写指令代码地址(E1)
CRADD1   EQU      8200H        ; 读状态字地址(E1)
DWADD1   EQU      8100H        ; 写显示数据地址(E1)
DRADD1   EQU      8300H        ; 读显示数据地址(E1)
CWADD2   EQU      4000H        ; 写指令代码地址(E2)
CRADD2   EQU      4200H        ; 读状态字地址(E2)
DWADD2   EQU      4100H        ; 写显示数据地址(E2)
DRADD2   EQU      4300H        ; 读显示数据地址(E2)
    
```

(1) 写指令代码子程序(E1)

```

PR0:    PUSH     DPL
        PUSH     DPH
        MOV      DPTR,#CRADD1    ; 设置读状态字地址
PR01:   MOVX     A,@DPTR          ; 读状态字
        JB      ACC.7,PR01        ; 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(E1)

```

PR1:  PUSH    DPL
      PUSH    DPH
      MOV     DPTR,#CRADD1      ; 设置读状态字地址
      JB     ACC.7,PR11        ; 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(E1)

```

PR2:  PUSH    DPL
      PUSH    DPH
      MOV     DPTR,#CRADD1     ; 设置读状态字地址
PR21: MOVX    A,@DPTR         ; 读状态字
      JB     ACC.7,PR21       ; 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
      RET

```

(4) 写指令代码子程序(E2)

```

PR3:  PUSH    DPL
      PUSH    DPH
      MOV     DPTR,#CRADD2     ; 设置读状态字地址
PR31: MOVX    A,@DPTR         ; 读状态字
      JB     ACC.7,PR31       ; 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(E2)

```

PR4:  PUSH    DPL
      PUSH    DPH
      MOV     DPTR,#CRADD2     ; 设置读状态字地址
PR41: MOVX    A,@DPTR         ; 读状态字
      JB     ACC.7,PR41       ; 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

```

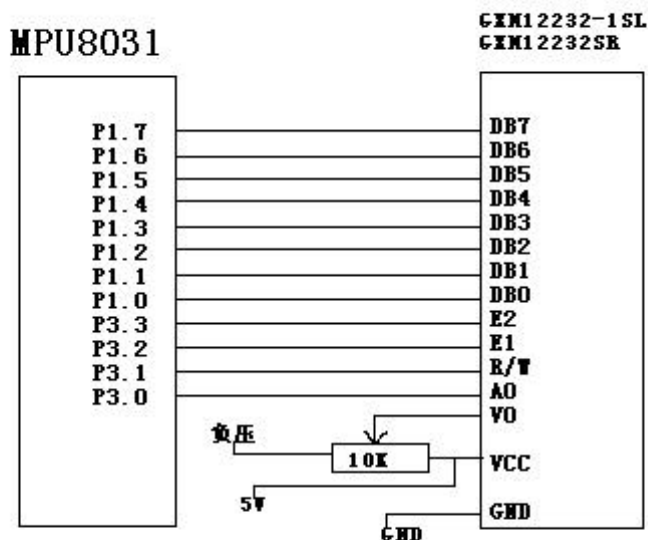
POP DPH
 POP DPL
 RET

(6) 读显示数据子程序(E2)

```

PR5:  PUSH   DPL
      PUSH   DPH
      MOV    DPTR,#CRADD2      ; 设置读状态字地址
PR51: MOVX   A,@DPTR          ; 读状态字
      JB    ACC.7,PR51        ; 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
    
```

2、间接控制方式接口电路及驱动程序



图八 间接控制方式原理图

间接控制方式驱动程序如下:

A0	EQU	P3.0	; 寄存器选择信号
R/W	EQU	P3.1	; 读/ 写选择信号
E1	EQU	P3.2	; 使能信号 E1
E2	EQU	P3.3	; 使能信号 E2

(1) 写指令代码子程序(E1)

```

PR0:  CLR    A0              ; A0=0
      SETB  R/W            ; R/W=1
    
```

```

PR01:  MOV    P1,#0FFH      ; P1 口置"1"
        SETB   E1           ; E1=1
        MOV    A,P1         ; 读状态字
        CLR    E1           ; E1=0
        JB     ACC.7,PR01    ; 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   E1           ; E1=1
        CLR    E1           ; E1=0
        RET

```

(2) 写显示数据子程序(E1)

```

PR1:    CLR    A0           ; A0=0
        SETB   R/W         ; R/W=1
PR11:   MOV    P1,#0FFH    ; P1 口置"1"
        SETB   E1           ; E1=1
        MOV    A,P1         ; 读状态字
        CLR    E1           ; E1=0
        JB     ACC.7,PR11   ; 判"忙"标志为"0"否, 否再读
        SETB   A0           ; A0=1
        CLR    R/W         ; R/W=0
        MOV    P1,DAT       ; 写数据
        SETB   E1           ; E1=1
        CLR    E1           ; E1=0
        RET

```

(3) 读显示数据子程序(E1)

```

PR2:    CLR    A0           ; A0=0
        SETB   R/W         ; R/W=1
PR21:   MOV    P1,#0FFH    ; P1 口置"1"
        SETB   E1           ; E1=1
        MOV    A,P1         ; 读状态字
        CLR    E1           ; E1=0
        JB     ACC.7,PR21   ; 判"忙"标志为"0"否, 否再读
        SETB   A0           ; A0=1
        MOV    P1,#0FFH    ; P1 口置"1"
        SETB   E1           ; E1=1
        MOV    DAT,P1       ; 读数据
        CLR    E1           ; E1=0
        RET

```

(4) 写指令代码子程序(E2)

```

PR3:    CLR    A0           ; A0=0
        SETB   R/W         ; R/W=1
PR31:   MOV    P1,#0FFH    ; P1 口置"1"
        SETB   E2           ; E2=1

```

```

MOV    A,P1                ; 读状态字
CLR    E2                  ; E2=0
JB     ACC.7,PR31          ; 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   E2                  ; E2=1
CLR    E2                  ; E2=0
RET

```

(5) 写显示数据子程序(E2)

```

PR4:   CLR    A0            ; A0=0
        SETB   R/W          ; R/W=1
PR41:  MOV    P1,#0FFH      ; P1 口置"1"
        SETB   E2            ; E2=1
        MOV    A,P1         ; 读状态字
        CLR    E2            ; E2=0
        JB     ACC.7,PR41    ; 判"忙"标志为"0"否, 否再读
        SETB   A0            ; A0=1
        CLR    R/W          ; R/W=0
        MOV    P1,DAT       ; 写数据
        SETB   E2            ; E2=1
        CLR    E2            ; E2=0
        RET

```

(6) 读显示数据子程序(E2)

```

PR5:   CLR    A0            ; A0=0
        SETB   R/W          ; R/W=1
PR51:  MOV    P1,#0FFH      ; P1 口置"1"
        SETB   E2            ; E2=1
        MOV    A,P1         ; 读状态字
        CLR    E2            ; E2=0
        JB     ACC.7,PR51    ; 判"忙"标志为"0"否, 否再读
        SETB   A0            ; A0=1
        MOV    P1,#0FFH      ; P1 口置"0"
        SETB   E2            ; E2=1
        MOV    DAT,P1       ; 读数据
        CLR    E2            ; E2=0
        RET

```

3、应用子程序

示例程序使用 8031 汇编语言，接口电路为上两节提供的实用电路，相应的驱动子程序也为上两节所提供。

示例一 初始化子程序

对模块的初始化实际上是对 SED1520 的初始化。程序如下：

```

INT:  MOV    COM,#0E2H          ; 复位
      LCALL  PR0
      LCALL  PR3
      MOV    COM,#0A4H          ; 关闭等待状态
      LCALL  PR0
      LCALL  PR3
      MOV    COM,#0A9H          ; 设置 1/32 占空比
      LCALL  PR0
      LCALL  PR3
      MOV    COM,#0A0H          ; 正向排序设置
      LCALL  PR0
      LCALL  PR3
      MOV    COM,#0C0H          ; 设置显示起始行为第一行
      LCALL  PR0
      LCALL  PR3
      MOV    COM,#0AFH          ; 开显示设置
      LCALL  PR0
      LCALL  PR3
      RET

```

示例二 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             ; 取页地址值
      ORL    A,#0B8H          ; "或"页面地址设置代码
      MOV    COM,A            ; 页面地址设置
      LCALL  PR0
      LCALL  PR3
      MOV    COM,#00H          ; 列地址设置为"0"
      LCALL  PR0
      LCALL  PR3
      MOV    R3,#50H          ; 一页清 80 个字节
CLEAR2: MOV   DAT,#00H        ; 显示数据为"0"
      LCALL  PR1
      LCALL  PR4
      DJNZ  R3,CLEAR2          ; 页内字节清零循环
      INC   R4                 ; 页地址暂存器加 1
      CJNE  R4,#04H,CLEAR1     ; RAM 区清零循环
      RET

```

示例三 西文字符写入子程序

由于水平为 100 点列，由两片 SED1520 平分各管理 50 路驱动，所以在程序中要有中间接合部字符的处理。程序把显示屏分为 4 个字符行，以页地址设定；列地址为任意设定。由于 SED1520 没有字符库，所以在程序后面提供了由北京清华蓬远公司编制的常用标准的 5

×7 点阵的字符字模。每个字符由 8 个字节的字模数据组成，其中字模的第一个字节和第七、八个字节作为字符间距设置的，该程序可以选择写入 6×8 点阵或 8×8 点阵的字符。程序的输入寄存器为列地址寄存器，页地址寄存器和代码寄存器。页地址寄存器 PAGE 的 D7 位即为字符体选择位。该程序设置了一个参数值，使其适应 10032 或 12032 等各种规格的内置 SED1520 液晶显示模块。子程序清单如下：

	PD1	EQU	32H	； 模块参数 MGLS10032
	PD1	EQU	3CH	； 模块参数 MGLS12032
	COLUMN	EQU	30H	； 列地址寄存器(0-63H)
	PAGE	EQU	31H	； 页地址寄存器 D1,D0: 页地址
				； D7: 字符体 D7=0 为 6×8 点阵
				； D7=1 为 8×8 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×8
	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		； 6×8 点阵
	LJMP	CW_2		
CW_1:	MOV	COUNT,#08H		； 8×8 点阵
CW_2:	ANL	A,#03H		； 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	PR0		
	LCALL	PR3		
	MOV	A,COLUMN		； 读列地址寄存器
	CLR	C		
	SUBB	A,#PD1		； 列地址-模块参数
	JC	CW_3		； <0 为左半屏显示区域(E1)
	MOV	COLUMN,A		； ≥0 为右半屏显示区域(E2)
	MOV	A,PAGE		
	SETB	ACC.3		； 设置区域标志位
	MOV	PAGE,A		； "0"为 E1,"1"为 E2
CW_3:	MOV	COM,COLUMN		； 设置列地址值
	MOV	A,PAGE		； 判区域标志以确定设置哪个控制器
	JNB	ACC.3,CW_4		
	LCALL	PR3		； 区域 E2


```

        LJMP      CW_5
CW_4:  LCALL    PR0          ; 区域 E1
CW_5:  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
        JNB    ACC.3,CW_6
        LCALL   PR4        ; 区域 E2
        LJMP   CW_7
CW_6:  LCALL    PR1        ; 区域 E1
CW_7:  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,#PD1,CW_8
CW_8:  JC      CW_9        ; 未超出则继续
        MOV    A,PAGE     ; 超出则判是否在区域 E2
        JB    ACC.3,CW_9   ; 在区域 E2 则退出
        SETB  ACC.3       ; 在区域 E1 则修改成区域 E2
        MOV   PAGE,A
        MOV   COM,#00H    ; 设置列地址为"0"
        LCALL PR3
CW_9:  DJNZ   COUNT,CW_5  ; 循环
        RET

```

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

西文显示演示程序段

MOV	PAGE,#03H	; 6x8 点阵字体, 第 4 页
MOV	COLUMN,#04H	; 起始列为第 4 列
MOV	CODE,#34H	; 字符代码
LCALL	CW_PR	
MOV	PAGE,#03H	
MOV	COLUMN,#0CH	
MOV	CODE,#45H	
LCALL	CW_PR	
MOV	PAGE,#03H	
MOV	COLUMN,#14H	
MOV	CODE,#4CH	
LCALL	CW_PR	
MOV	PAGE,#03H	
MOV	COLUMN,#1CH	

```

MOV      CODE,#1AH
LCALL   CW_PR
MOV      R7,#00H
MOV      R6,#28H
LOOP:   MOV      A,R7
MOV      DPTR,#TAB1
MOVC    A,@A+DPTR
MOV      CODE,A
MOV      PAGE,#83H           ; 8×8 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,15H,12H,10H,18H,18H,13H
```

1. 中文字符写入子程序:

```

PD1      EQU      32H           ; 模块参数 MGLS10032
PD1      EQU      3CH           ; 模块参数 MGLS12032
COLUMN   EQU      30H
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,#03H	
	ORL	A,#0B8H	; "或"页地址设置代码
	MOV	COM,A	; 写页地址设置指令
	LCALL	PR0	
	LCALL	PR3	
	POP	COLUMN	; 取列地址值
	MOV	A,COLUMN	; 读列地址寄存器
	CLR	C	
	SUBB	A,#PD1	; 列地址-模块参数
	JC	CCW_2	; <0 为左半屏显示区域(E1)
	MOV	COLUMN,A	; ≥0 为右半屏显示区域(E2)
	MOV	A,PAGE	
	SETB	ACC.3	; 设置区域标志位.
	MOV	PAGE,A	; "0"为 E1,"1"为 E2
CCW_2:	MOV	COM,COLUMN	; 设置列地址值
	MOV	A,PAGE	; 判区域标志以确定设置哪个控制器
	JNB	ACC.3,CCW_3	
	LCALL	PR3	; 区域 E2
	LJMP	CCW_4	
CCW_3:	LCALL	PR0	; 区域 E1
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	
	JNB	ACC.3,CCW_5	
	LCALL	PR4	; 区域 E2
	LJMP	CCW_6	
CCW_5:	LCALL	PR1	; 区域 E1
CCW_6:	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,#PD1,CCW_7	
CCW_7:	JC	CCW_8	; 未超出则继续
	MOV	A,PAGE	; 超出则判是否在区域 E2
	JB	ACC.3,CCW_8	; 在区域 E2 则退出
	SETB	ACC.3	; 在区域 E1 则修改成区域 E2
	MOV	PAGE,A	
	MOV	COM,#00H	; 设置区域 E2 列地址为"0"
	LCALL	PR3	
CCW_8:	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"
CLR      ACC.3
MOV      PAGE,A
MOV      CODE,#10H      ; 间址寄存器设置为 16
LJMP    CCW_1           ; 大循环
CCW_9:   RET
    
```

2. 中文字符库:

```

CCTAB:   DB 001H,001H,0FFH,001H,001H,07FH,048H,044H; 南
          DB 05FH,041H,041H,05FH,041H,041H,041H,040H
          DB 000H,004H,0FEH,000H,004H,0FEH,024H,044H
          DB 0F4H,004H,004H,0F4H,004H,004H,014H,008H
          DB 002H,001H,001H,0FFH,000H,01FH,010H,010H; 京
          DB 010H,01FH,001H,009H,009H,011H,025H,002H
          DB 000H,000H,004H,0FEH,010H,0F8H,010H,010H
          DB 010H,0F0H,000H,040H,030H,018H,008H,000H
          DB 000H,07FH,040H,05FH,041H,041H,041H,04FH; 国
          DB 041H,041H,041H,041H,05FH,040H,07FH,040H
          DB 004H,0FEH,024H,0F4H,004H,004H,044H,0E4H
          DB 004H,044H,024H,004H,0F4H,004H,0FCH,004H
          DB 000H,01FH,010H,010H,01FH,010H,010H,01FH; 显
          DB 014H,044H,034H,014H,004H,004H,0FFH,000H
          DB 010H,0F8H,010H,010H,0F0H,010H,010H,0F0H
          DB 050H,044H,04CH,050H,040H,044H,0FEH,000H
    
```

```

MOV      - PAGE,#00H
MOV      COLUMN,#10H
MOV      CODE,#00H
LCALL   CCW_PR
MOV      PAGE,#00H
MOV      COLUMN,#20H
MOV      CODE,#01H
LCALL   CCW_PR
MOV      PAGE,#00H
MOV      COLUMN,#30H
MOV      CODE,#02H
LCALL   CCW_PR
MOV      PAGE,#00H
MOV      COLUMN,#40H
MOV      CODE,#03H
LCALL   CCW_PR
SJMP    $
    
```