

图形点阵液晶显示模块使用手册

TG19264A (L)

广州铜铎电子有限公司
TINSHARP ELECTRONICS CO., LTD

公司地址：广州市天河区天河路龙苑大厦 A2 座 17 楼 1703

电话：(020-85512381, 87541571 传真:020-85517797

工厂地址：番禺大石金科工业园

网址：HTTP:WWW.TINSHARP.COM

E-mail:TINSHARP@public.Guangzhou.gd.cn

SALES@TINSHARP.COM

目 录

(一) 概述----- (1)

(二) 外形尺寸图----- (1)

(三) 模块主要硬件构成说明----- (1)

(四) 模块的外部接口----- (2)

(五) 指令说明----- (3)

(六) 读写操作时序----- (4)

(七) 应用举例 ----- (5)

一. 概述

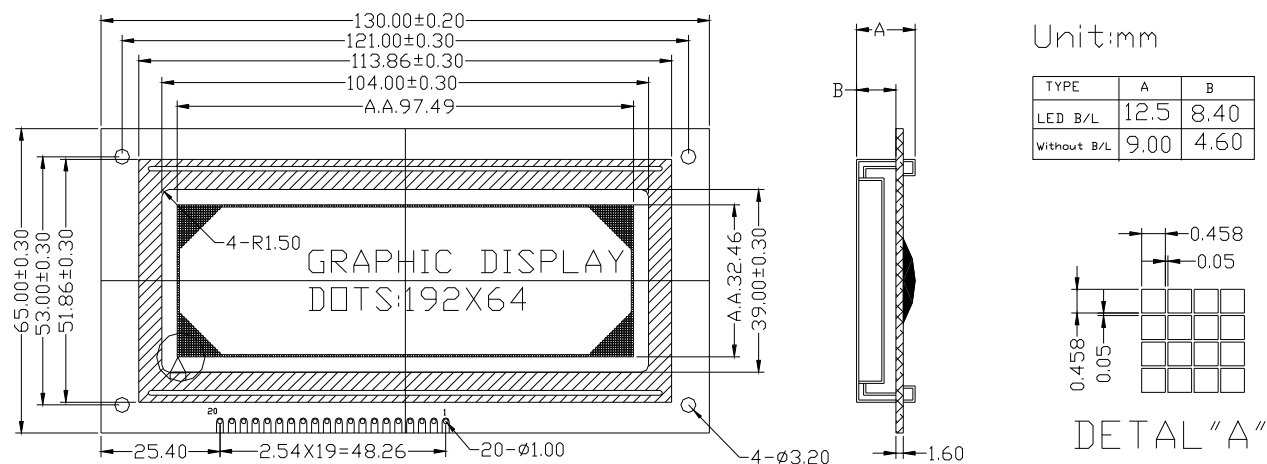
TG19264A 是一种图形点阵液晶显示器,它主要由行驱动器/列驱动器及格 192×64 全点阵液晶显示器组成。可完成图形显示,也可以显示 12×4 个(16×16 点阵)汉字。

主要技术参数和性能:

1. 电源: VDD: +5V;
2. 显示内容: 192(列)×64(行)点
3. 全屏幕点阵
4. 七种指令
5. 与 CPU 接口采用 8 位数据总线并行输入输出和 8 条控制线
6. 占空比 1/64
7. 工作温度: -10℃~+60℃, 存储温度: -20℃~+70℃

二. 外形尺寸图

1. 外形尺寸图



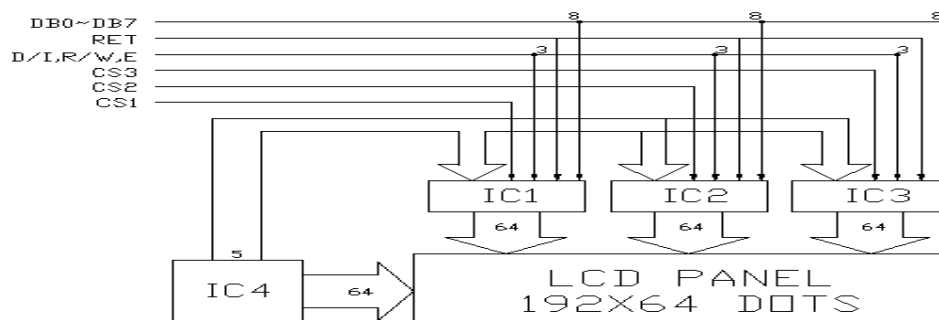
2. 外形尺寸

表 1

ITEM	NOMINAL DIMEN	UNIT
模块体积	130×65×12.5	mm
视域	104×39	mm
行列点阵数	192×64	dots
点距离	0.508×0.508	mm
点大小	0.458×0.458	mm

三. 模块主要硬件构成说明

(结构框图)



IC4 为行驱动器。IC1, IC2, IC3 为列驱动器。IC1, IC2, IC3, IC4 含有以下主要功能器件。了解如下器件有利于对 LCD 模块之编程。

1. 指令寄存器(IR)

IR 是用于寄存指令码,与数据寄存器数据相对应。当 D/I=0 时,在 E 信号下降沿的作用下,指令码写入 IR。

2. 数据寄存器(DR)

DR 是用于寄存数据的，与指令寄存器寄存指令相对应。当 D/I=1 时，在下降沿作用下，图形显示数据写入 DR，或在 E 信号高电平作用下由 DR 读到 DB7~DB0 数据总线。DR 和 DDRAM 之间的数据传输是模块内部自动执行的。

3. 忙标志：BF

BF 标志提供内部工作情况。BF=1 表示模块在内部操作，此时模块不接受外部指令和数据。BF=0 时，模块为准备状态，随时可接受外部指令和数据。

利用 STATUS READ 指令，可以将 BF 读到 DB7 总线，从检验模块之工作状态。

4. 显示控制触发器 DFF

此触发器是用于模块屏幕显示开和关的控制。DFF=1 为开显示 (DISPLAY ON)，DDRAM 的内容就显示在屏幕上，DFF=0 为关显示 (DISPLAY OFF)。

DDF 的状态是指令 DISPLAY ON/OFF 和 RST 信号控制的。

5. XY 地址计数器

XY 地址计数器是一个 9 位计数器。高 3 位是 X 地址计数器，低 6 位为 Y 地址计数器，XY 地址计数器实际上是作为 DDRAM 的地址指针，X 地址计数器为 DDRAM 的页指针，Y 地址计数器为 DDRAM 的 Y 地址指针。

X 地址计数器是没有记数功能的，只能用指令设置。

Y 地址计数器具有循环记数功能，各显示数据写入后，Y 地址自动加 1，Y 地址指针从 0 到 63。

6. 显示数据 RAM (DDRAM)

DDRAM 是存储图形显示数据的。数据为 1 表示显示选择，数据为 0 表示显示非选择。DDRAM 与地址和显示位置的关系见 DDRAM 地址表（见第 6 页）。

7. Z 地址计数器

Z 地址计数器是一个 6 位计数器，此计数器具备循环记数功能，它是用于显示行扫描同步。当一行扫描完成，此地址计数器自动加 1，指向下一行扫描数据，RST 复位后 Z 地址计数器为 0。

Z 地址计数器可以用指令 DISPLAY START LINE 预置。因此，显示屏幕的起始行就由此指令控制，即 DDRAM 的数据从哪一行开始显示在屏幕的第一行。此模块的 DDRAM 共 64 行，屏幕可以循环滚动显示 64 行。

四. 模块的外部接口

外部接口信号如下表 2 所示：

表 2

管脚号	管脚名称	LEVER	管脚功能描述
1	VSS	0	电源地
2	VDD	5.0V	电源电压
3	V0	--	液晶显示器驱动电压
4	D/I	H/L	D/I=“H”，表示 DB7~DB0 为显示数据 D/I=“L”，表示 DB7~DB0 为显示指令数据
5	R/W	H/L	R/W=“H”，E=“H” 数据被读到 DB7~DB0 R/W=“L”，E=“H→L” 数据被写到 IR 或 DR
6	E	H/L	R/W=“L”，E 信号下降沿锁存 DB7~DB0 R/W=“H”，E=“H” DDRAM 数据读到 DB7~DB0
7	DB0	H/L	数据线
8	DB1	H/L	数据线
9	DB2	H/L	数据线
10	DB3	H/L	数据线
11	DB4	H/L	数据线
12	DB5	H/L	数据线
13	DB6	H/L	数据线
14	DB7	H/L	数据线
15	CS1	L	选择 IC1，即（左）64 列
16	SET	L	复位控制信号，RST=0 有效
17	CS2	L	选择 IC2，即（中）64 列
18	CS3	L	选择 IC3，即（右）64 列

19	VEE	-10V	LCD 驱动负电压
20	LED+	+5.0V	LED 背光板电源

五. 指令说明 指令表 表 3

指令	指令码										功能
	R/W	D/I	D7	D6	D5	D4	D3	D2	D1	D0	
显示 ON/OFF	0	0	0	0	1	1	1	1	1	1/0	控制显示器的开关，不影响 DDRAM 中数据和内部状态
显示起始行	0	0	1	1	显示起始行 (0 ••• 63)						指定显示屏从 DDRAM 中哪一行开始显示数据
设置 X 地址	0	0	1	0	1	1	1	X: 0 •• 7			设置 DDRAM 中的页地址(X 地址)
设置 Y 地址	0	0	0	1	Y 地址 (0 •• 63)						设置地址(Y 地址)
读状态	1	0	B U S Y	0	ON / OFF	R S T	0	0	0	0	读取状态 RST 1:复位 0:正常 ON/OFF 1:显示开 0:显示关 BUSY 0:READY 1:IN OPERATION
写显示数据	0	1	显示数据								将数据线上的数据 DB7~DB0 写入 DDRAM
读显示数据	1	1	显示数据								将数据线上的数据 DB7~DB0 写入 DDRAM

1. 显示开关控制(DISPLAY ON/OFF)

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

D=1:开显示(DISPLAY ON)意即显示器可以进行各种显示操作

D=0:关显示(DISPLAY OFF)意即不能对显示器可以进行各种显示操作

2. 设置显示起始行

代码形式	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	0	0	1	1	A5	A4	A3	A2	A1	A0

前面在 Z 地址计数器一节已经描述了显示起始行是由 Z 地址计数器控制的。A5~A0 的 6 位地址自动送入 Z 地址计数器, 起始行的地址可以是 0~63 的任意一行。

例如: 选择 A5~A0 是 62, 则起始行与 DDRAM 行的对应关系如下:

DDRAM 行: 62 63 0 1 2 3 28 29

屏幕显示行: 1 2 3 4 5 6 31 32

3. 设置页地址

代码形式	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	0	0	1	0	1	1	1	A2	A1	A0

所谓页地址就是 DDRAM 的行地址, 8 行为一页, 模块共 64 行即 8 页, A2~A0 表示 0~7 页。读写数据对地址没有影响, 页地址由本指令或 RST 信号改变复位后页地址为 0。页地址与 DDRAM 的对应关系见 DDRAM 地址表。

4. 设置 Y 地址(SET Y ADDRESS)

代码形式	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	0	0	0	1	A5	A4	A3	A2	A1	A0

此指令的作用是将 A5~A0 送入 Y 地址计数器, 作为 DDRAM 的 Y 地址指针。在对 DDRAM 进行读写操作后, Y 地址指针自动加 1, 指向下一个 DDRAM 单元。

DDRAM 地址表:

表 4

CS1=1						CS2=1					CS3=1					行号
Y=	0	1	••	62	63	0	1	••	62	63	0	1	••	62	63	
X=0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	0
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	7
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	8
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	55
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	56
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	63
↓	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	

5. 读状态 (STATUS READ)

代码形式	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	0	1	BUSY	0	ON/OFF	RET	0	0	0	0

当 R/W=1 D/I=0 时, 在 E 信号为 “H” 的作用下, 状态分别输出到数据总线 (DB7~DB0) 的相应位。

BF: 前面已叙述过 (见 BF 标志位一节)。

ON/OFF: 表示 DFF 触发器的状态 (见 DFF 触发器一节)。

RST: RST=1 表示内部正在初始化, 此时组件不接受任何指令和数据。

6. 写显示数据 (WRITE DISPLAY DATE)

代码形式	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	0	1	D7	D6	D5	D4	D3	D2	D1	D0

D7~D0 为显示数据, 此指令把 D7~D0 写入相应的 DDRAM 单元, Y 地址指针自动加 1。

7. 读显示数据 (READ DISPLAY DATE)

代码形式	R/W	D/I	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	1	1	D7	D6	D5	D4	D3	D2	D1	D0

此指令把 DDRAM 的内容 D7~D0 读到数据总线 DB7~DB0, Y 地址指针自动加 1。

六. 读写操作时序

1. 写操作时序

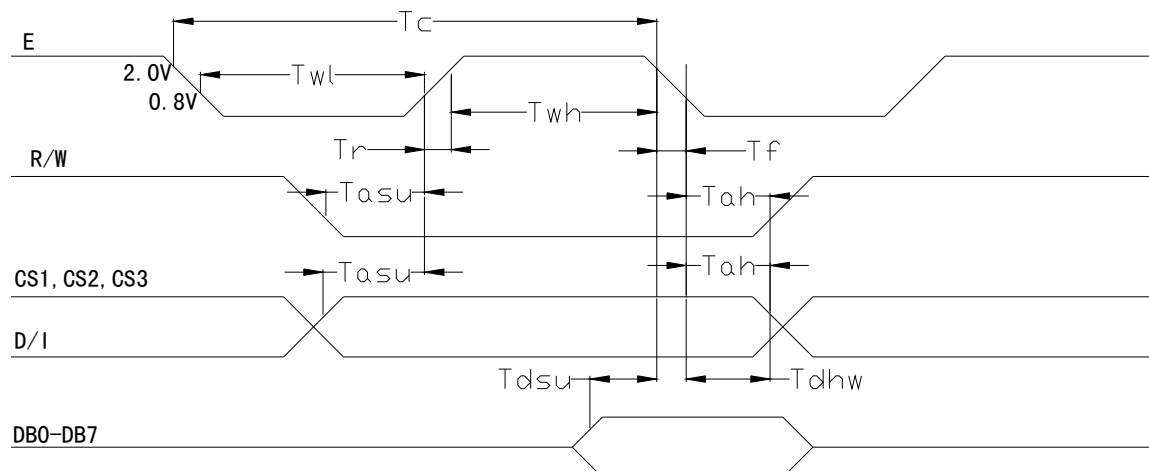
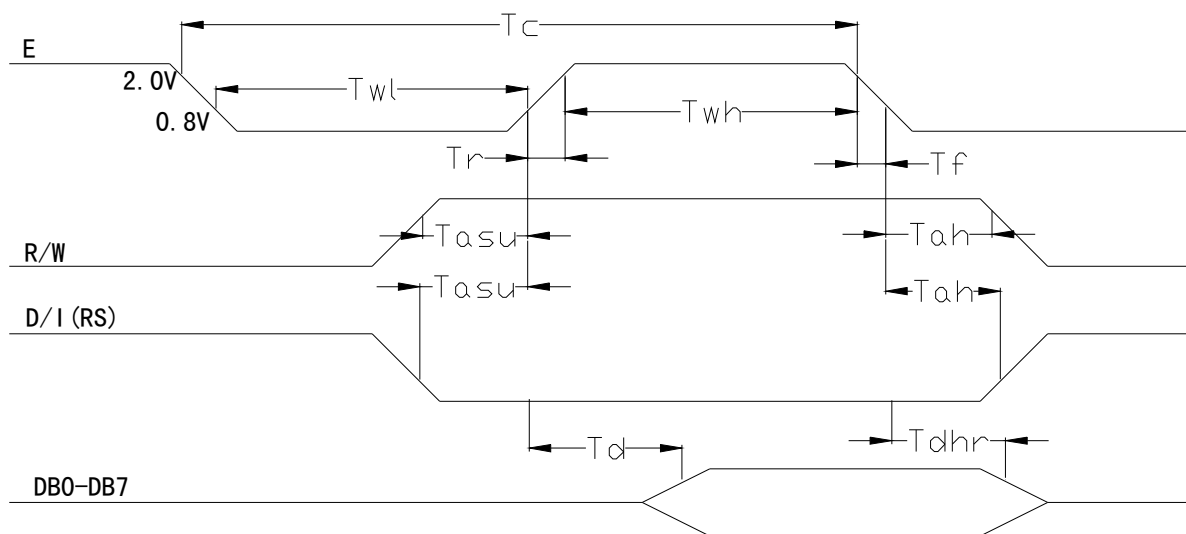


图 3

2. 读操作时序

图 4



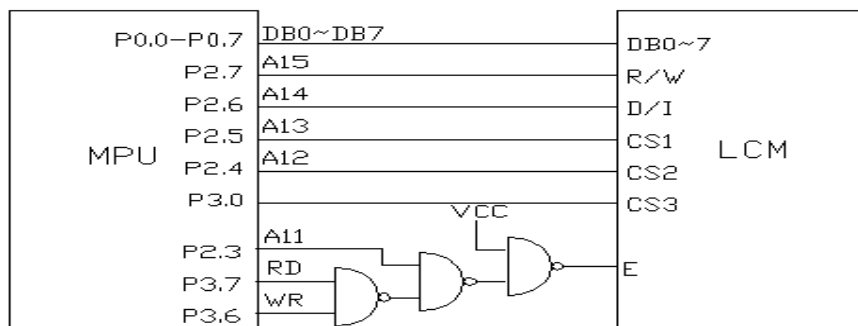
3. 读写时序参数表

表 5

名 称	符 号	最小值	典型值	最大值	单位
E 周期时间	T_{cyc}	1000	---	---	ns
E 高电平宽度	P_{weh}	450	---	---	ns
E 低电平宽度	P_{wel}	450	---	---	ns
E 上升时间	T_r	---	---	25	ns
E 下降时间	T_f	---	---	25	ns
地址建立时间	T_{as}	140	---	---	ns
地址保持时间	T_{ah}	10	---	---	ns
数据建立时间	T_{dsw}	200	---	---	ns
数据延迟时间	T_{ddr}	---	---	320	ns
写数据保持时间	T_{dhw}	10	---	---	ns
读数据保持时间	T_{dhw}	20	---	---	ns

七. 应用举例

图 5



TG19264 与单片机 8031 的一种接口如图 5. 所示:

利用图 5 举例介绍几编程实例

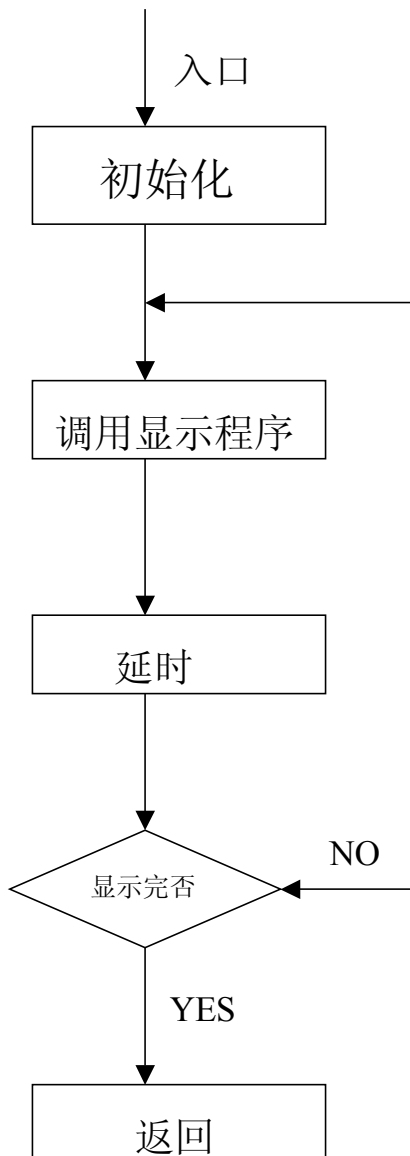
1. 显示开/关控制 (DISPLAY ON/OFF)

所谓显示开与关, 意即当显示器置于“开”状态时, 可以对显示进行各种操作。当显示器置于“关”状态时, 不能对显示器显示进行各种显示操作。

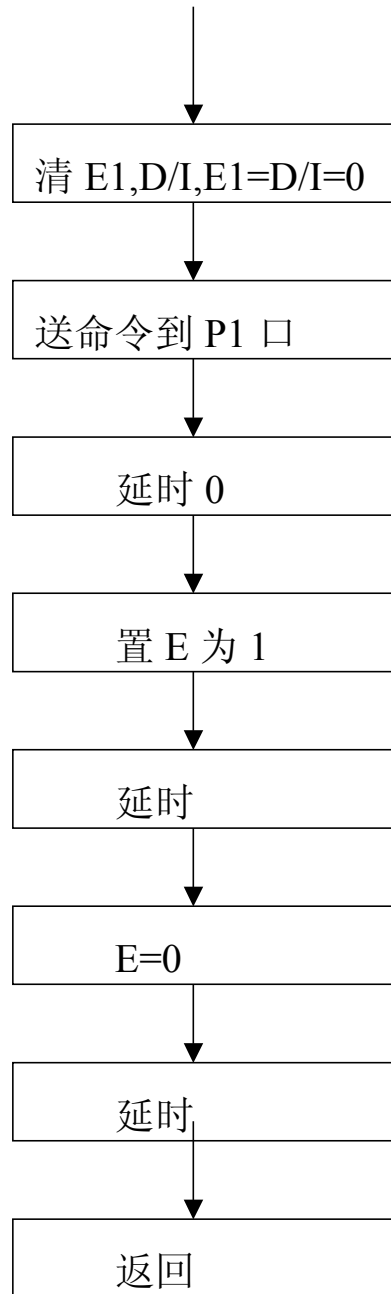
```
MOV A, #3EH      ; 关显示代码为 3EH, 开显示操作只需将代码 3EH 换为 3FH 即可
SETB P3.0        ; Select Chip3
MOV DPTR, #3800H ; R/W For Write, And Select Chip1 and Chip2
MOVX @DPTR, A    ; Output the Instruction
CLR P3.0
```

2. 页地址设置
前面已对页地址进行定义，即 8 行为一页。LCD（或 DDRAM）共为 8 页。
MOV A, #0B8H ; 设置页地址为 0 页
SETB P3.0 ; Select Chip3
MOV DPTR, #3800H ; R/W For Write, And Select Chip1 and Chip2
MOVX @DPTR, A ; Output the Instruction
CLR P3.0
3. 写显示数据与对应 DDRAM
假设页地址，Y 地址已设置完毕，显示数据为 DATA。
MOV A, #DATA ; 设送显示数据为
SETB P3.0 ; Select Chip3
MOV DPTR, #7800H ; R/W For Write, And Select Chip1 and Chip2
MOVC @DPTR, A ; Output the Instruction
CLR P3.0

显示流程图

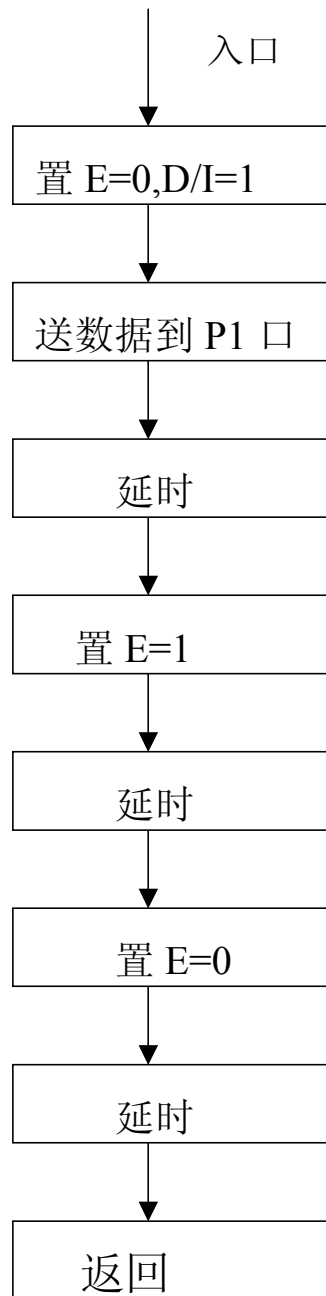


写指令程序



WRI:

```
CLR    E
CLR    DI
MOV     P1,R1
CALL    DELAY
SETB    E
CALL    DELAY
CLR     E
CALL    DELAY
CALL    DELAY
RET
```



WRD:

```
CLR      E
SETB     DI
MOV      P1,R1
CALL     DELAY
SETB     E
CALL     DELAY
CLR      E
CALL     DELAY
CALL     DELAY
RET
```

利用图 5 举例介绍编程实例

DI	EQU	P3.7
E	EQU	P3.5
RST	EQU	P3.2
cs1	equ	p3.0
cs2	equ	p3.1

;本测试程序中 CS1,CS2=H ,RW =>GND

```
ORG 0000H
LJMP MAIN
```

```
ORG 0003H
LJMP LINTO
```

```
ORG 0100H
```

```
MAIN: NOP
```

```
clr ea
clr cs1
clr cs2
CLR E
lcall init
```

```
MOV 34H,#0ffH
LCALL DISPLAY_ALL
lcall dly100
```

```
MOV 34H,#00H
LCALL DISPLAY_ALL
```

```
lcall plybw
lcall DLY100
```

```
MOV 34H,#00H
LCALL DISPLAY_ALL
```

```
MOV R1,#0C0H ;START = 0
LCALL WRI
```

```
mov 31h,#0
mov 30h,#0

MOV 30H,#0          ;x
MOV 31H,#8          ;y
MOV DPTR,#HZI7      ;您
LCALL outhz

MOV 30H,#0
MOV 31H,#40
MOV DPTR,#HZI8      ;好
LCALL outhz

MOV 30H,#4
MOV 31H,#16
MOV DPTR,#HZI1      铜
LCALL outhz

MOV 30H,#4
MOV 31H,#32
MOV DPTR,#HZI2      ;铎
LCALL outhz

MOV 30H,#6
MOV 31H,#0
MOV DPTR,#HZI3      ;电
LCALL outhz

MOV 30H,#6
MOV 31H,#16
MOV DPTR,#HZI4      ;子
LCALL outhz

MOV 30H,#6
MOV 31H,#32
MOV DPTR,#HZI5      ;公
LCALL outhz

MOV 30H,#6
MOV 31H,#48
MOV DPTR,#HZI6      ;司
LCALL outhz

LCALL DLY100
LJMP  MAIN
```

```
LINTO: NOP
      NOP
      RETI

init:  CLR RST      ;RESET
      LCALL DLY50
      LCALL DLY50
      LCALL DLY50
      LCALL DLY50

      setb rst
      MOV R1,#3FH   ;DISPLAY ON
      LCALL WRI

      MOV R1,#0C0H  ;START = 0
      LCALL WRI

      MOV 34H,#00H
      LCALL DISPLAY_ALL
      ret

DISPLAY_ALL: MOV R1,#0C0H  ;START = 0
      LCALL WRI

      MOV R7,#8      ;DISPLAY DATA= 34H
      MOV R2,#0B8H
LOP1:  MOV A,R2
      MOV R1,A      ; X+1
      LCALL WRI
      MOV R1,#40H   ; Y
      LCALL WRI

      MOV R5,#64
LOP11: MOV R1,34H
      LCALL WRD
      DJNZ R5,LOP11

      INC R2
      DJNZ R7,LOP1

      ret

outhz:
```

```
                ;30h=x (0-6)  31h=y(0-63-16)
MOV R1,#0C0H    ;START = 0
LCALL WRI

MOV R7,#2
MOV A,#0B8H
ADD A,30H      ;B8+X
MOV 30H,A      ;A>>30H
HZ2:  ; MOV A,R2
      MOV R1,30H ; X+1
      LCALL WRI

      MOV A,#40H ; Y
      ADD A,31H
      ; MOV 31H,A ; 40+Y
      MOV R1,A
      LCALL WRI

      MOV R5,#16
HZ1:  MOV A,#00H
      MOVC A,@A+DPTR
      MOV R1,A
      LCALL WRD
      LCALL DLY10
      INC DPTR
      DJNZ R5,HZ1

      INC 30H
      DJNZ R7,HZ2
      RET

plybw:
MOV R1,#0C0H    ;START = 0
LCALL WRI

mov 30h,#0
mov 31h,#0
mov dptr,#bw
MOV R7,#8
MOV A,#0B8H
ADD A,30H      ;B8+X
MOV 30H,A      ;A>>30H
HZ2bw:  ; MOV A,R2
        MOV R1,30H ; X+1
        LCALL WRI
```

```
; MOV A,#40H    ; Y
; ADD A,31H
; MOV 31H,A      ; 40+Y
; MOV R1,A
mov R1,#40h
LCALL WRI
```

```
MOV R5,#64
HZ1bw:  MOV A,#00H
        MOVC A,@A+DPTR
        MOV R1,A
        LCALL WRD
        LCALL DLY10
        INC DPTR
        DJNZ R5,HZ1bw

        INC 30H
        DJNZ R7,HZ2bw
        RET
```

```
WRI:    ;写指令子程序
        CLR E
        CLR DI
        MOV P1,R1
        lcall DLY10
        SETB E
        LCALL DLY10
        CLR E
        LCALL DLY10
        LCALL DLY10
        LCALL DLY10
        RET
```

```
WRD:    ;写数据子程序
        CLR  E
        SETB DI
        MOV P1,R1
        LCALL DLY10
        SETB E
        LCALL DLY10
        CLR E
        LCALL DLY10
        LCALL DLY10
```

LDCALL DLY10
RET

DLY10:MOV R3,#01H
DL1:MOV R4,#01H
DL2:MOV R6,#0FH
DL3:DJNZ R6,DL3
DJNZ R4,DL2
DJNZ R3,DL1
RET

DLY50:MOV R3,#04H
DL111:MOV R4,#0FH
DL222:MOV R6,#0AFH
DL333:DJNZ R6,DL333
DJNZ R4,DL222
DJNZ R3,DL111
RET

DLY100:MOV R3,#06H
DL11:MOV R4,#0FFH
DL22:MOV R6,#0FFH
DL33:DJNZ R6,DL33
DJNZ R4,DL22
DJNZ R3,DL11
RET

ORG 0300H
WELCOM: DB 57H,65H,6CH,63H,6FH,6DH,65H,20H,20H,20H,20H,20H,20H,20H
DB 20H,20H,20H,20H,20H,20H
ORG 0320H
TINSHARP: DB 54H,49H,4EH,53H,48H,41H,52H,50H,20H,4CH,43H,4DH,20H,20H
DB 20H,20H,20H,20H,20H,20H

ORG 0340H
HZI1: ; 铜
DB 080H,080H,070H,0C8H,056H,048H,0F8H,010H,010H,048H,048H,028H,0F8H,008H,000H,000H
DB 000H,000H,002H,02FH,012H,008H,03FH,000H,00FH,009H,027H,040H,07FH,000H,000H,000H
HZI2: ; 铎
DB 000H,080H,060H,090H,05EH,050H,030H,010H,0FCH,044H,0FEH,010H,008H,080H,000H,000H
DB 000H,000H,002H,03FH,012H,00AH,008H,008H,00BH,048H,07CH,009H,009H,005H,000H,000H
HZI3: ; 电

DB 00H,0f8H, 48H, 48H, 48H, 48H,0ffH, 48H, 48H, 48H, 48H,0fcH, 08H, 00H, 00H, 00H
DB 00H, 07H, 02H, 02H, 02H, 02H, 3fH, 42H, 42H, 42H, 42H, 47H, 40H, 70H, 00H, 00H
HZI4: ;子
DB 80H, 80H, 82H, 82H, 82H, 82H, 82H,0e2H,0a2H, 92H, 8aH, 86H, 80H,0c0H, 80H, 00H
DB 00H, 00H, 00H, 00H, 00H, 40H, 80H, 7fH, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H
HZI5: ;公
DB 00H, 00H, 80H, 40H, 30H, 0cH, 00H,0c0H, 06H, 18H, 20H, 40H, 80H, 80H, 80H, 00H
DB 01H, 01H, 00H, 30H, 28H, 24H, 23H, 20H, 20H, 28H, 30H, 60H, 00H, 01H, 00H, 00H
HZI6: ;司
DB 10H, 10H, 92H, 92H, 92H, 92H, 92H, 92H,0d2H, 9aH, 12H, 02H,0ffH, 02H, 00H, 00H
DB 00H, 00H, 3fH, 10H, 10H, 10H, 10H, 10H, 3fH, 00H, 40H, 80H, 7fH, 00H, 00H, 00H
HZI7: ;您
DB 40H, 20H, 10H,0fcH, 23H, 10H, 8cH, 67H, 04H,0f4H, 04H, 44H, 94H, 8cH, 00H, 00H
DB 40H, 30H, 00H, 77H, 80H, 81H, 88H, 92H,0b4H, 83H, 80H,0e0H, 00H, 11H, 60H, 00H
HZI8: ;好
DB 10H, 10H,0f0H, 1fH, 10H,0f0H, 80H, 82H, 82H, 82H,0e2H, 92H, 8aH,0c6H, 80H, 00H
DB 40H, 22H, 15H, 08H, 14H, 63H, 00H, 00H, 40H, 80H, 7fH, 00H, 00H, 00H, 00H, 00H

bw:

DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffh,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH
DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH
DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffh,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH
DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH

DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffh,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH
DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH
DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffh,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH
DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH

DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB 0ffH,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H

DB 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H,0ffh,0ffH,0ffH,0ffH,0ffH,0ffH,0ffH

```
DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH
DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffh, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH
DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH

DB  0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB  0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB  0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H
DB  0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 00H, 00H, 00h, 00H, 00H, 00H, 00H, 00H

DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffh, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH
DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH
DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffh, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH
DB  00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH, 0ffH
END
```