

图形点阵液晶显示模块使用手册

CM24064-18SLYB

深圳市彩晶科技有限公司

地址：深圳市南山区沙河西路茶光建兴3栋东四楼

TEL: 0755-26137169 26622598 FAX: 0755-26736698

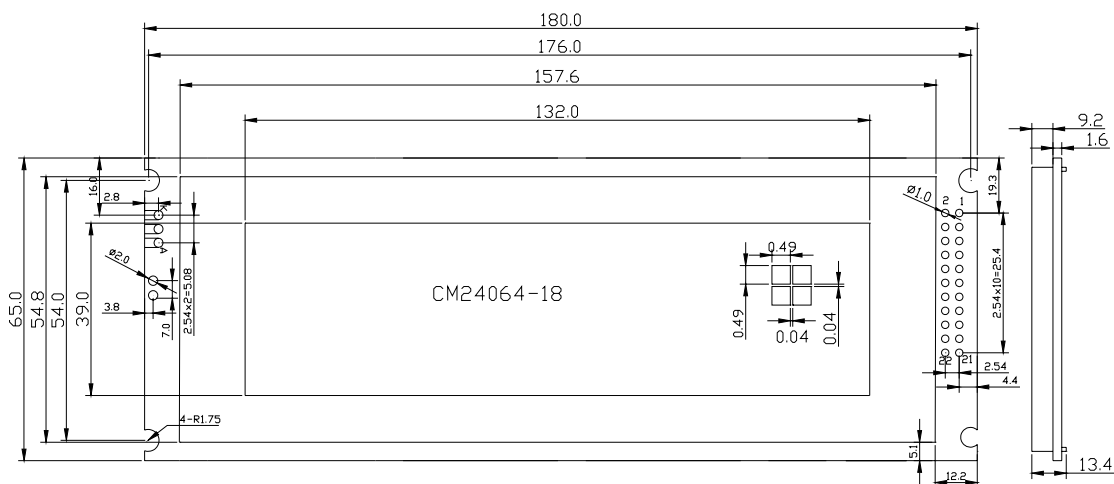
<http://www.szcm-lcd.com> E-MAIL: szcm@szcm-lcd.com

<http://www.szlcm.com> E-MAIL: Lcm@szLcm.com

CM24064-18 是一种图形点阵液晶显示器, 它由控制器 T6963C、行驱动器/列驱动器及 240×64 全点阵液晶显示器组成. 可完成图形显示, 也可以显示 15×4 个(16×16 点阵)汉字主要技术参数和性能:

1. 电源: VDD: +5V±10%; 模块内自带-15V 负压, 用于 LCD 的驱动电压。
2. 显示内容: 240(列)×64(行)点
3. 全屏幕点阵
4. 带 8K 外部数据存储器(其地址由软件设定)
5. 其接口适配 8080 系列和 Z80 系列 MPU 的控制时序
6. 驱动方式: 1/64 DUTY, 1/9 BIAS
7. 工作温度: -10℃~+60℃, 存储温度: -20℃~+70℃
8. 显示模式: 黄绿膜、灰膜、蓝膜、黑白膜
9. 背光特性: LED 或 EL 背光(黄绿色、蓝色、白色、红色)
10. 模块封装方式: SMT
11. 视角方向: 6:00
12. 功耗: 模块自带负压

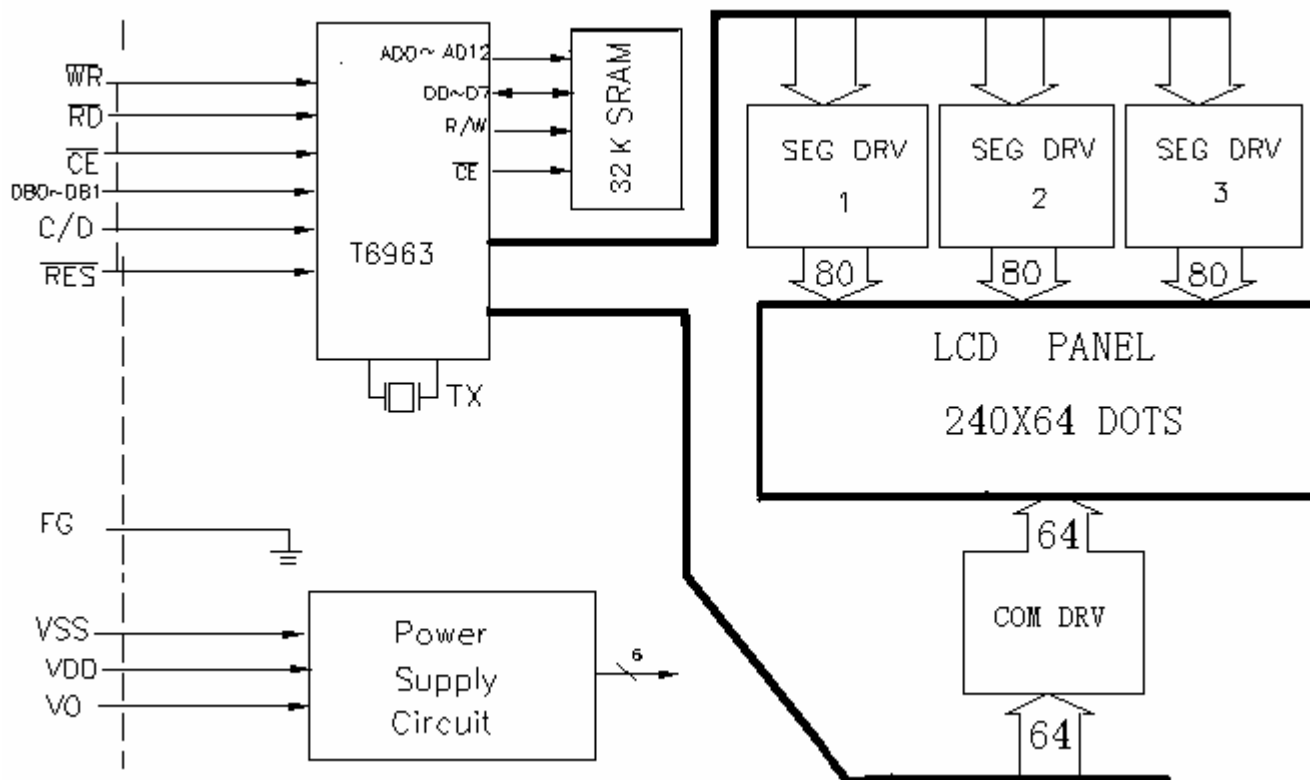
二、外形尺寸图



外形尺寸一览

ITEM	NOMINAL DIMEN	UNIT
模块体积	180×65×13.4	mm
视域	132×39	mm
行列点阵数	240×64	DOTS
点距离	0.04×0.04	mm
点大小	0.49×0.49	mm

三、硬件结构图



四、模块的外部接口

Pin no	Symbol	Level	Function
1	FG	0V	Frame ground
2	Vss	0V	Ground
3	VDD	5.0V	Supply voltage for logic
4	VO	-	Operating voltage for LCD drive (-10~15V)
5	/WR	L	Write signal
6	/RD	L	Read signal
7	/CE	L	Chip enable signal
8	C/D	H/L	H: Instruction code L: Data
9	NC	--	No connection
10	/RESET	L	Reset signal
11	DB0	H/L	Data bit 0
12	DB1	H/L	Data bit 1
13	DB2	H/L	Data bit 2
14	DB3	H/L	Data bit 3
15	DB4	H/L	Data bit 4
16	DB5	H/L	Data bit 5
17	DB6	H/L	Data bit 6
18	DB7	H/L	Data bit 7
19	FS	H/L	Font select signal (H:5X8 dots; L:8X8 dots)
20	VEE	--	-15V
21	LED+	5V	LED Backlight
22	LED-	--	LED Backlight

五、IC 说明及指令表

T6963C is LCD controller designed to be used for control LCD driver LCD driver LSIs and display data Memory, It has an 8 bit parallel data bus

And control lines for reading or writing through a MPU I/F.

It has 128 words character generator ROM with the capability to control External display RAM of up to 128K bytes. Allocation of text, graphics And external generator RAM can be easily made and the display window can Be freely moved within the allocated memory range.

It supports a very board range of LCD formats by selecting different Combinations on a set of programmable inputs. It can be used in text,graphic

Modes and has various attribute functions.

指令表:

COMMAND	CODE	D1	D2	FUNCTION
Register Set	00100001	X address	Y address	Cursor pointer set
	00100010	Data	00H	Off register
	00100100	Low address	High address	Address pointer set
Control Word set	01000000	Low address	High address	Text home address set
	01000001	Columns	00H	Text area set
	01000010	Low address	High address	Graphic home address set
	01000011	Columns	00H	Graphic area set
Mode set	1000x000	-	-	"OR" mode
	1000x001	-	-	"EXOR" mode
	1000x011	-	-	"AND" mode
	1000x100	-	-	"Text attribute" mode
	10000xxx	-	-	Internal CGROM mode
	10001xxx	-	-	External CGRAM mode
Display Mode	10010000	-	-	Display off
	1001xx10	-	-	Cursor on, blink off
	1001xx11	-	-	Cursor on, blink on
	100101xx	-	-	Text on, graphic off
	100110xx	-	-	Text off, graphic on
	100111xx	-	-	Text on, graphic on
Cursor Pattern Select	10100000	-	-	1 line cursor
	10100001	-	-	2 line cursor
	10100010	-	-	3 line cursor
	10100011	-	-	4 line cursor
	10100100	-	-	5 line cursor
	10100101	-	-	6 line cursor
	10100110	-	-	7 line cursor
	10100111	-	-	8 line cursor
Data auto Read/write	10110000	-	-	Data auto write set
	10110001	-	-	Data auto read set
	10110010	-	-	Auto reset
Data read Write	11000000	Data	-	Data write and ADP increment
	11000001	-	-	Data read and ADP increment
	11000010	Data	-	Data write and ADP decrement
	11000011	-	-	Data read and ADP decrement
	11000100	Data	-	Data write and ADP no variable
	11000101	-	-	Data read and ADP no variable
Screen peek	11100000	-	-	Screen peek
Screen copy	11101000	-	-	Screen copy
Bit Set/Reset	11110XX	-	-	Bit reset
	11111XXX	-	-	Bit set
	1111X000	-	-	Bit 0(LSB)
	1111X001	-	-	Bit 1
	1111X010	-	-	Bit 2
	1111X011	-	-	Bit 3
	1111X100	-	-	Bit 4
	1111X101	-	-	Bit 5
	1111X110	-	-	Bit 6
	1111X111	-	-	Bit 7(MSB)

六、电气参数

1. ABSOLUTE MAXIMUM RATING

Item	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage(logic)	VDD-VSS	-	-0.3	-	7	V
Supply Voltage(LCD Drive)	VDD-VO	-	VDD+0.3	-	VDD-0.3	V
Input Voltage	VI	-	-0.3	-	VDD+0.3	V
Operating Temperature	Topr	-	-10	-	+55	° C
Storage Temperature	Tstg	-	-20	-	+60	° C

2. OPTICAL DATA Ta=25° C

Item	Symbol	Condition	Standard Value			Unit	
			min	typ	max		
Supply voltage(Logic)	VDD-VSS	-	4.75	5	5.25	V	
Supply voltage(LCD Drive)	VDD-VO	-	-	-	-	V	
Supply current	IDD	-	-	12.0	17.0	mA	
	IO	-	-	1.8	2.5	mA	
EL Backlight current	IEL	-	-	100	-	mA	
Input high voltage	VIH	High level	0.7VDD	-	VDD	V	
Input low voltage	VIL	Low level	0	-	0.3VDD	V	
Supply voltage for LCD Drive (1/80 duty)	VDD-VO	Ta=0° C	14.2	14.5	14.8	V	
		Ta=25° C	13.3	13.6	13.9	V	
		Ta=50° C	12.3	12.6	12.9	V	
Contrast Ratio	CR		-	4	-	-	
Viewing Angle	-	CR ≥ 2	θ	-10	-	20	deg
			θ	60	-	120	deg
Response Time (rise)	Tr	Note 1	Ta=25°	-	130	200	ms
Response Time (delay)	Td	Note 2	Ta=25°	-	150	230	ms

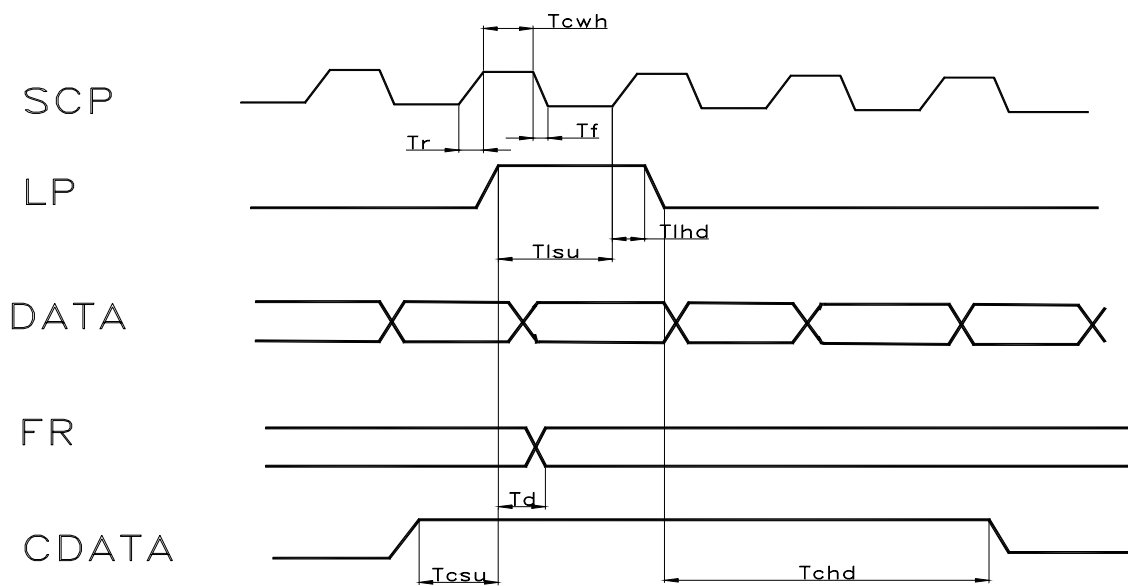
NOTE 1: Required time for blackening ratio of segment goes up from 0% to 90% when Wave from is switched from one selected one (θ =10° , φ =90°)

Note 2: Required time for blackening ratio of segment goes down from 100% to 10% When wave from is switched from one selected one (θ =10° , φ =90°)

3. TIMING CHARACTERISTICS

Item	Symbol	Min	Max	Unit
Operating frequency	fSCP	-	2.75	MHZ
SCP pulse width	Tcwh, Tcwl	150		ns
SCP rise/fall time	Tr	-	30	ns
LP set up time	Tlsu	150	290	ns
LP hold time	Tlhd	5	40	ns
Data set up time	Tdsu	170	-	ns
Data hold time	Tdhhd	80	-	ns
FR delay time	Td	0	90	ns
CDATA set up time	Tcsu	450	850	ns
CDATA hold time	Tchhd	450	950	ns

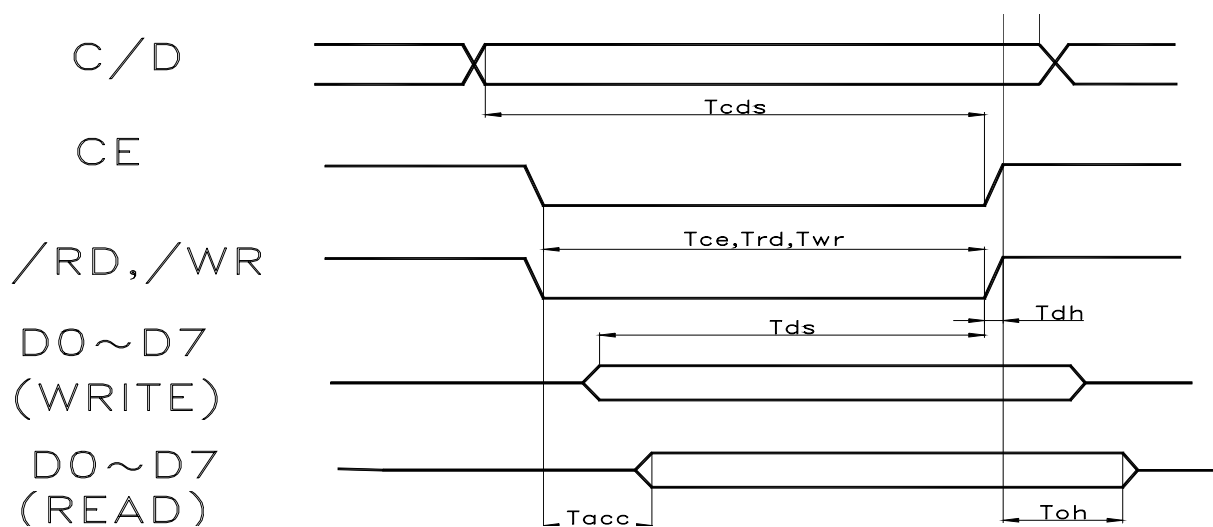
Condition: DV=+5.0V ± 10%, Ta=-10~+70° C



驱动波形图

4. INTERFACE TIMING

Item	Symbol	Min	Max	Unit
C/D set up time	Tcds	100	-	ns
C/D hold time	Tcdh	10	-	ns
CE,RD,WR pulse width	Tce,Trd,Twr	80	-	ns
DATA set up time	Tds	80	-	ns
DATA hold time	Tdh	40	-	ns
Access time	Tacc	-	150	ns
Output hold time	Toh	10	50	ns



读写时序图

5. EL Backlight Electrical Characteristics

Ta=25° C VDD=5V

Symbol	Parameter	Min	Type	Max	Units
IIN	VDD supply current	10		150	mA
VA-B	Output voltage across lamp	37	40	43	V
		75	80	85	V
FEL	VA-B output drive frequency	600	800	1000	V
VDD	Supply voltage	4.5		12	V
CL	Load capacitance	0		25	nF
TA	Operating temperature	0		50	° C

Note: EL Backlight with white lamp.

七、功能描述

7.1 STATUS READ

Before sending data(read/write),command it is necessary to check the Status.

Status check

Status of T6963c can read from data lines.

/RD L
 /WR H
 /CE L
 C/D H

D0 D7 Status word

T6963C status word format is following

MSB				LSB			
STA7	STA6	STA5	STA4	STA3	STA2	STA1	STA0
D7	D6	D5	D4	D3	D2	D1	D0

STA0	Check capability of command execution	0: Disable 1: Enable
STA1	Check capability of data read/write	0: Disable 1: Enable
STA2	Check capability of auto mode data read	0: Disable 1: Enable
STA3	Check capability of auto mode data write	0: Disable 1: Enable
STA4	Not use	
STA5	Check capability of controller operation	0: Disable 1: Enable
STA6	Error flag. Using screen peek/copy command	0: No error 1: error
STA7	Check the condition blink	0: Disable 1: Enable

Note 1: It is necessary to check STA0 and STA1 at the same time, The error is happened by sending data at executing command.

2: The status check will be enough to check STA0/STA1.

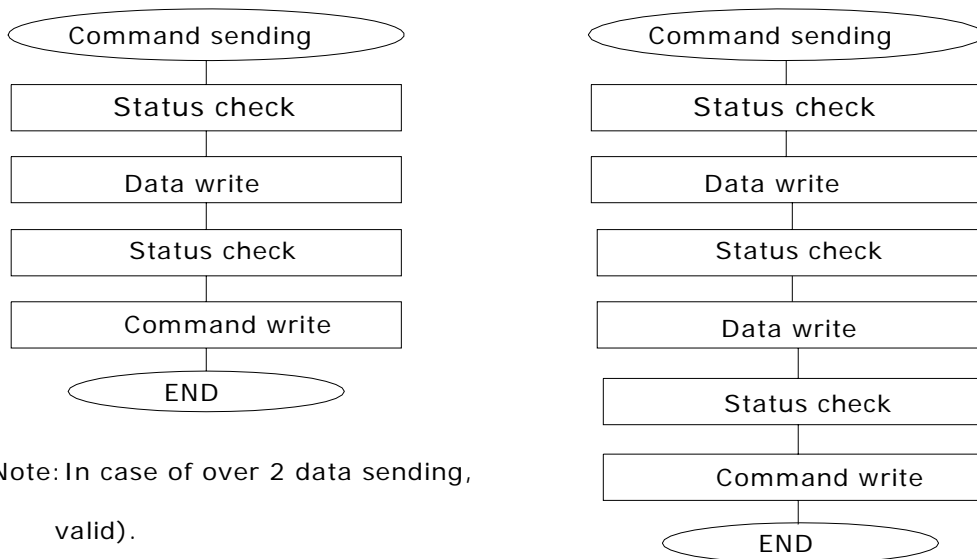
3: STA2/STA3 are valid in auto mode STA0/STA1 are invalid.

Status checking flow:



7.2 DATA SET

In T6963C, the data have been set and command executes.
The order of procedure of command sending



7.31 Description of command

1. Register set

CODE	HEX	FUNCTION	D1	D2
00100001	21H	Cursor pointer set	X address	Y address
00100010	22H	Offset register set	Data	00H
00100100	24H	Address pointer set	Low address	High address

(1) Cursor pointer set

The position of cursor is specified by X address. The cursor position is moved only by this command. The cursor pointer doesn't have the function of increment and decrement. The shift of cursor set by this command. X address, Y address are specified following.

X address 00H - - - - 4FH(Low 7bits are valid)

Y address 00H - - - - 1FH(Low 5bits are valid)

1 Screen drive

X address 00H - - - - 4FH

Y address 00H - - - - 0FH

(2) Offset register set

The offset register is used to determine external character generator RAM area.

T6963C has 16 bits address lines as follow:

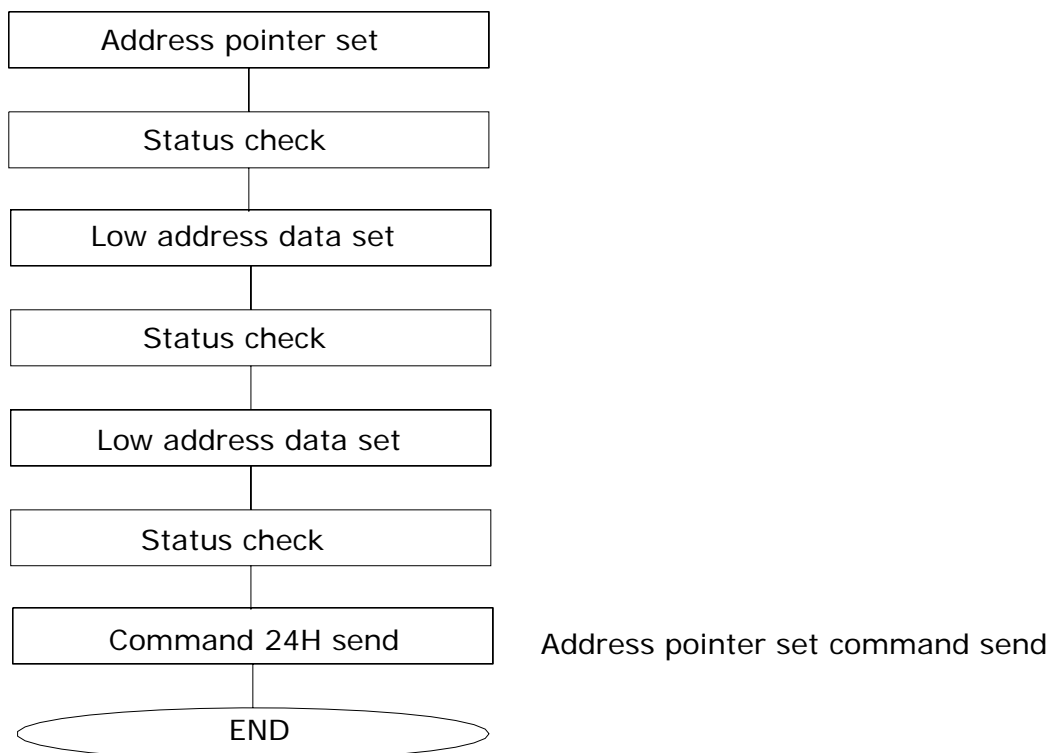
MSB											LSB				
Ad15	Ad14	Ad13	Ad12	Ad11	Ad10	Ad9	Ad8	Ad7	Ad6	Ad5	Ad4	Ad3	Ad2	Ad1	Ad0

The upper 5 bits (ad15 - ad11) are determined by offset register. The Middle 8 bits (ad10 - ad3) are determined by character code. The Lower 3 bit (ad2 - ad0) are determined by vertical counter. The Lower 5 bit of D1(data) are valid. The data format of external character Generator RAM.

(3) Address pointer set

The address pointer set command is used to indicate the start address for writing (or reading) to external RAM.

The flow chart address pointer set command



7.32 Control word set

CODE	HEX	FUNCTION	D1	D2
01000000	40H	Text home address set	Low address	High address
01000001	41H	Text area set	Columns	00H
01000010	42H	Graphic home address set	Low address	High address
01000011	43H	Graphic area set	Columns	00H

The home address and column size are defined by this command

(1) Text home address and area set

The starting address of external display RAM for text display is Defined by this command. The text home address shows the left end and Most upper position .

The relationship of external display RAM address and display position

Example:

Text home address: 0000H
 Text area: 00A0H
 MD2=0, MD3=0: 80 COLUMN
 DUAL=0, MDS=1, MD0=1, MD1=0: 28 LINES
 Display plane:

0000H	0001H	- - - - -	004EH	004FH	1 Line
00A0H	00A1H	- - - - -	00EEH	00EFH	2 Line
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
10E0H	10E1H	- - - - -	112EH	112FH	28 Lines

(2) Graphic home address and area set

The starting address of external display RAM for Graphic display is

Defined by this command. The graphic home address shows the left end most Upper line.

The relationship of external display RAM address and display position.

Example:

Graphic home address: 0000H
 Graphic area: 0020H
 MD2=H, MD3=H: 32 COLUMNS
 DUAL=H, MDS=L, MDO=H, MD1=H: 2 LINES

Example:
 Display plane:

0000H	0001H	- - - - -	001EH	001FH
0020H	0021H	- - - - -	003EH	003FH
⋮	⋮	⋮	⋮	⋮
01E0H	01E1H		01FEH	01FFH

7.33 MODE SET

The display mode is defined by this command. The display mode don't have changed until to send next this command. Logically "OR", "EXOR", "AND" of text and graphic display can be displayed.

When internal character generator mode is selected, character code 00H - 7FH are selected from built-in character generator ROM. The character code 80H-FFH are automatically selected external character generator RAM.

NOTE: Only text display is attributed, because attributed data is located.

Attribute function

"Reverse display", "Character blink" and "Inhibit" are called "Attribute". The attribute data is written in the graphic area defined by control word set command. The mode set command selects text display only and graphic the mode set command selects text display only and graphic display cannot be displayed. The attribute data of the 1st character in text area

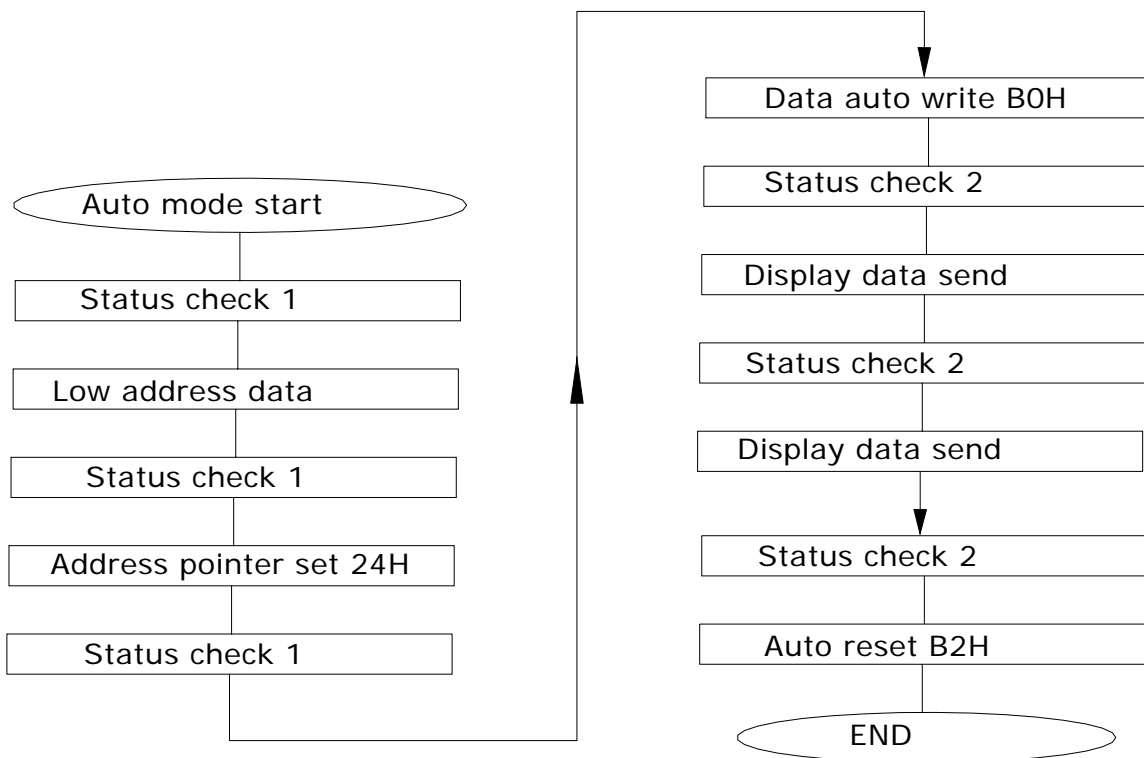
Is written at the 1st byte in graphic area, and attribute data of n-th 1byte in graphic area. Attribute function is defined as follow.

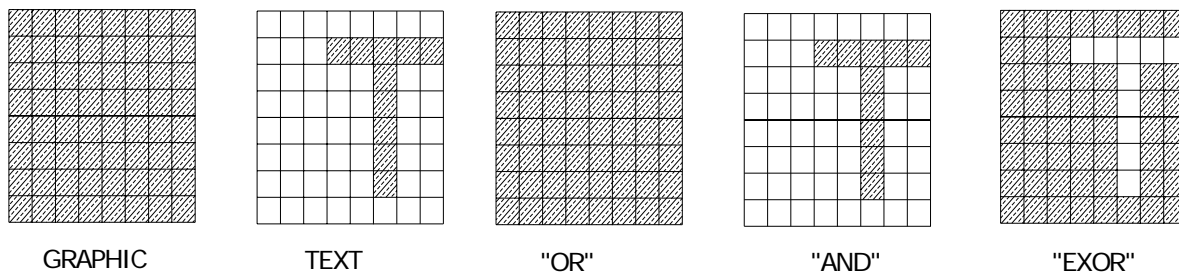
Attribute RAM 1byte

X	X	X	X	D3	D2	D1	D0
---	---	---	---	----	----	----	----

X: don't care

D3	D2	D1	D0	FUNCTION
0	0	0	0	Normal display
0	1	0	1	Reverse display
0	0	1	1	Inhibit display
1	0	0	0	Blink of normal display
1	1	0	0	Blink of reverse display
1	0	1	1	Blink of inhibit display





7.34 DATA AUTO READ/WRITE

CODE	HEX	FUNCTION	OPERAND
10110000	B0H	Data auto write set	-
10110001	B1H	Data auto read set	-
10110010	B2H	Auto reset	-

This command is convenient to send full screen data from external display RAM. After setting auto mode, "Data write(or read)" command is not necessary between each data. "Data auto write (or read)" command should follow the "Address pointer set" and address pointer is automatically increment by + 1 after each data. After sending (or receiving) all data "Auto reset" is necessary to return normal operation because all data is regarded "Display data" and no command can be accepted in the auto mode.

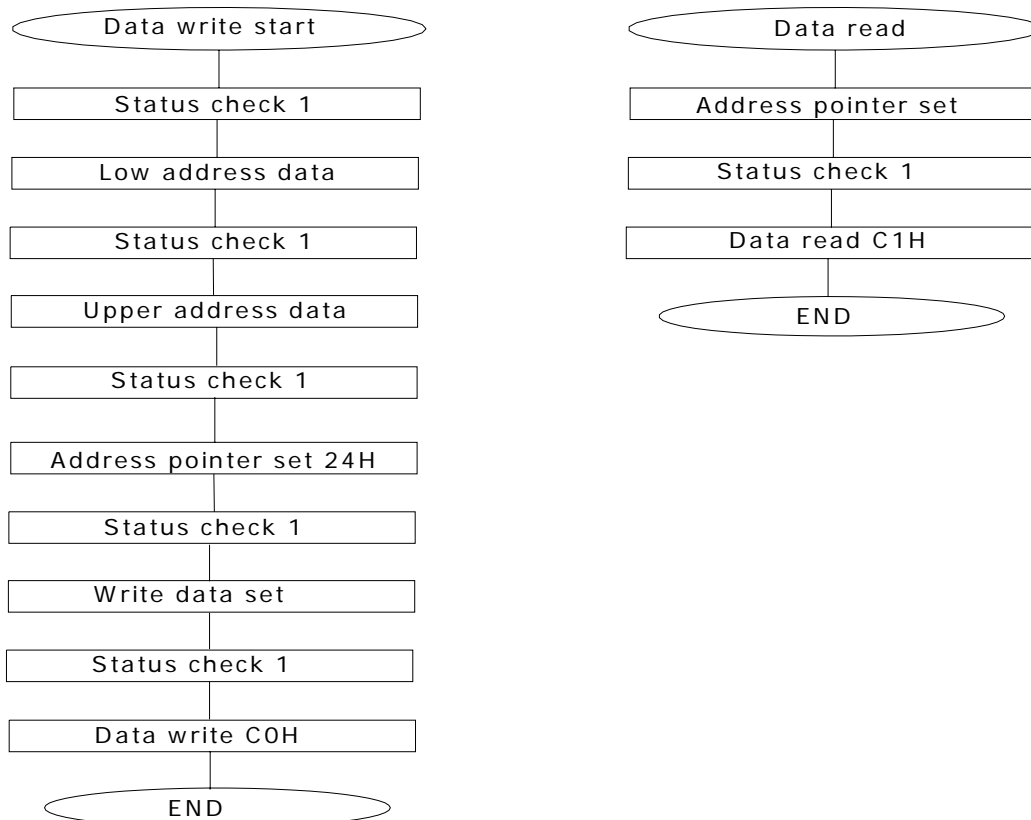
Note: status check for auto mode(STA2, STA3 should be checked between each Data. Auto reset should be performed after checking (STA3=1 STA2=1)

7.35 DATA READ WRITE

CODE	HEX	FUNCTION	OPERAND
11000000	C0H	Data write and ADP increment	Data
11000001	C1H	Data read and ADP increment	Data
11000010	C2H	Data write and ADP decrement	Data
11000011	C3H	Data read and ADP decrement	Data
11000100	C4H	Data write and ADP nonvariable	Data
11000101	C5H	Data read and ADP nonvariable	Data

This command is used for data write from MPU to external display RAM, AND data read external display RAM to MPU. Data write/data read should be executed after setting address by address pointer set command. Address Pointer can be automatically increment by setting this command.

Note: this command is necessary for each 1 byte data. Please refer following flow chart.



NOTE:

- (1) After power on, it is necessary to reset. /RESET is kept "L" between 5 CLOCK up(oscillation clock).
- (2) When /HALT has been "L", the oscillation is stopped. It is necessary To turn off power supply for LCD, because LCD goes down by DC bias.
- (3) The HALF function contains the RESET function.
- (4) After state of RESET/HALT.

TERMI NAL	HALT	RESET
DO-D7	F	F
DO-d7	F	F
R/w	H	H
/ce	H (NOTE 1)	H (NOTE 1)
Ad0-ad15	H (NOTE 2)	H (NOTE 2)
/ce0, /ce1	H (NOTE 1)	H (NOTE 1)
ED, HOD	Final Data	Final Data
HSCP	L	L
LP	L	L
CDATA	H	H
FR	H	H
CH1	L	K0
CH2	L	VEND
DSPON	L	L
X0	H	OSC CLOCK

H: Level H

L: Level L

F: Floating (High impedance)

K0: Internal state (TEXT data access) normally open

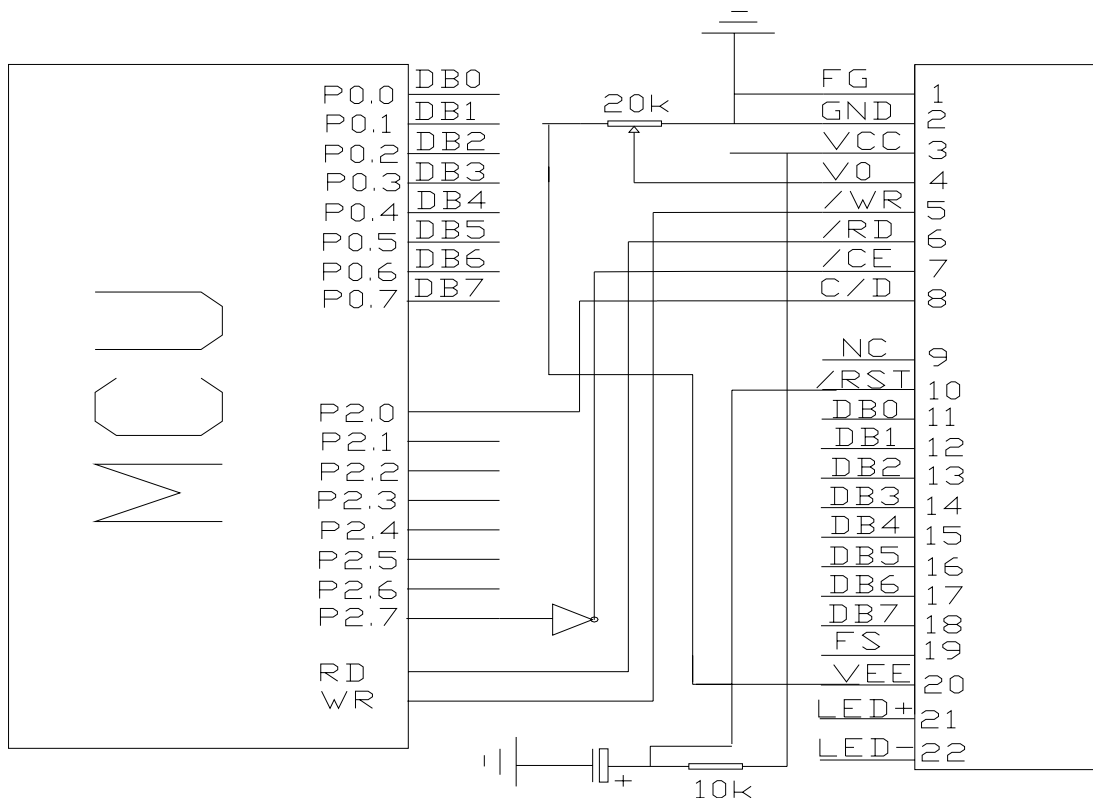
VEND: End signal of V-counter(Line count) if MDS=H, T2=L,
HEND (end signal of H-count) normally open.

Note 1: In Attribute mode, H or L by state of Graphinc pointer.

Note 2: In Attribute mode, DATA of Graphinc pointer.

八、应用举例

CM24064-18 与单片机 8031 的一种接口如下图. 所示:
 注:V0 为液晶驱动电压。此图为模块内自带负压的示例



```

        ORG      0000H

        AJMP    MAIN

        ORG      0003H

        AJMP    ZHONGDUAN

        ORG      0035H

ZHONGDUAN:

        HERE:   SJMP  HERE

        RET

;*****
DATA1   EQU    30H      ;第一参数单元
DATA2   EQU    31H      ;第二参数/数据单元
    
```

```

COMMAND    EQU    32H    ;指令代码单元
    C_ADD    EQU    8100H ;指令通道地址
    D_ADD    EQU    8000H ;数据通道地址
    LI1     EQU    33H
    LI2     EQU    34H

```

;*****

BUSY1:

```

    PUSH    DPH
    PUSH    DPL
    MOV     DPTR,#C_ADD
    MOVX    A,@DPTR
    POP     DPL
    POP     DPH
    RET

```

;*****

```

BUSY: LCALL    BUSY1
    JNB     ACC.0, BUSY1
    JNB     ACC.1, BUSY1
    RET

```

;*****

```

WRITE_COMMAND: PUSH    DPH
                PUSH    DPL
                LCALL    BUSY
                MOV     A,COMMAND
                MOV     DPTR,#C_ADD
                MOVX    @DPTR,A
                POP     DPL
                POP     DPH
                RET

```

WRITE_DATA:

```

        PUSH    DPH
        PUSH    DPL
        LCALL   BUSY
        MOV     A,DATA2
        MOV     DPTR,#D_ADD
        MOVX    @DPTR,A
        POP     DPL
        POP     DPH
        RET
    
```

```

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

```

DELAY:  MOV     R5,#05H
DELAY1: LCALL   MS40
        DJNZ   R5,DELAY1
        RET
    
```

```

MAIN:
        MOV     SP,#20H
        MOV     P3,#0FFH
        SETB   EA
        SETB   EX0
        SETB   IT0
        MOV     COMMAND,#90H
        LCALL   WRITE_COMMAND
        LCALL   MS40
        MOV     DATA2,#00H
    
```



```
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#40H ;文本显示区首地址设置
LCALL WRITE_COMMAND
MOV DATA2,#20H
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#41H ;文本显示区宽度设置
LCALL WRITE_COMMAND

MOV DATA2,#00H
LCALL WRITE_DATA
MOV DATA2,#08H
LCALL WRITE_DATA
MOV COMMAND,#42H ;图形显示区首地址设置
LCALL WRITE_COMMAND

MOV DATA2,#1EH
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#43H ;图形显示宽度设置
LCALL WRITE_COMMAND

MOV COMMAND,#80H ;逻辑或,用 CGROM
LCALL WRITE_COMMAND
```

```
MOV    COMMAND,#9CH    ;    启动文本.图形.光标显示
LCALL  WRITE_COMMAND
```

```
;*****888
```

```
;显示全屏
```

```
MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    COMMAND,#24H
LCALL  WRITE_COMMAND
MOV    R3,#00H
MOV    COMMAND,#0B0H
LCALL  WRITE_COMMAND
MOV    R4,#20H
```

```
YY1:
```

```
NOP
```

```
YY:   MOV    A,#0FFH
MOV    DATA2,A
LCALL  WRITE_DATA
DJNZ   R3,YY
DJNZ   R4,YY1
MOV    COMMAND,#0B2H
LCALL  WRITE_COMMAND
LCALL  MS40
```

```
;*****
```

```
CLEAR1:
```

```
MOV    DATA2,#00H
LCALL  WRITE_DATA
```

```

MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    COMMAND,#24H      ;显示地址设置
LCALL  WRITE_COMMAND
MOV    R3,#00H
MOV    R4,#20H
MOV    COMMAND,#0B0H    ;自动写入
LCALL  WRITE_COMMAND

M1:
NOP
M: MOV  A,#00H
MOV  DATA2,A
LCALL WRITE_DATA
DJNZ R3,M
DJNZ R4,M1
MOV  COMMAND,#0B2H      ;推出自动写入
LCALL WRITE_COMMAND

; *****

;显示字库

MOV  R3,#00H
MOV  R4,#7FH
MOV  COMMAND,#0B0H      ;自动写入
LCALL WRITE_COMMAND
MOV  A,#0FFH            ;写入英文字库
L: INC  A
MOV  DATA2,A

```

```

LCALL WRITE_DATA
DJNZ R4,L
MOV COMMAND,#0B2H ;退出自动写入
LCALL WRITE_COMMAND
LCALL DELAY

```

```

;#####

```

```

; 写 CGROM 80H

```

```

MOV DATA2,#03H
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#22H
LCALL WRITE_COMMAND

```

```

MOV DATA2,#00H
LCALL WRITE_DATA
MOV DATA2,#1CH
LCALL WRITE_DATA
MOV COMMAND,#24H
LCALL WRITE_COMMAND

```

```

MOV R2,#2
MOV R0,#0FFH
MOV COMMAND,#0B0H
LCALL WRITE_COMMAND

```

```

        MOV    DPTR,#TEST
LL2:    NOP
LL:     CLR    A
        MOVC   A, @A+DPTR
        MOV    DATA2,A
        LCALL  WRITE_DATA
        INC    DPTR
        SETB   P1.5
        DJNZ   R0,LL
        DJNZ   R2,LL2
        MOV    DATA2,#0B2H
        LCALL  WRITE_DATA
        SETB   P1.4

;#####

        MOV    DATA2,#00H
        LCALL  WRITE_DATA
        MOV    DATA2,#00H
        LCALL  WRITE_DATA
        MOV    COMMAND,#24H      ;显示地址设置€
        LCALL  WRITE_COMMAND

        MOV    COMMAND,#0B0H
        LCALL  WRITE_COMMAND
        MOV    R2,#2
TIAO1: MOV    R1,#255
MMM:    MOV    DATA2,#80H
        LCALL  WRITE_DATA

```

```
DJNZ R1,MMM
DJNZ R2,TIAO1
MOV COMMAND,#0B2H
LCALL WRITE_COMMAND
LCALL DELAY

MOV COMMAND,#0B0H
LCALL WRITE_COMMAND
MOV DATA2,#00H
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#24H ;显示地址设置€
LCALL WRITE_COMMAND
MOV R2,#2
TIAO2: MOV R1,#255
MMM1: MOV DATA2,#81H
LCALL WRITE_DATA
DJNZ R1,MMM1
DJNZ R2,TIAO2
MOV COMMAND,#0B2H
LCALL WRITE_COMMAND
LCALL DELAY

MOV COMMAND,#0B0H
LCALL WRITE_COMMAND
MOV DATA2,#00H
```

```
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#24H ;显示地址设置€
LCALL WRITE_COMMAND
MOV R2,#2
TIAO3: MOV R1,#255
MMM2: MOV DATA2,#82H
LCALL WRITE_DATA
DJNZ R1,MMM2
DJNZ R2,TIAO3
MOV COMMAND,#0B2H
LCALL WRITE_COMMAND
LCALL DELAY

MOV COMMAND,#0B0H
LCALL WRITE_COMMAND
MOV DATA2,#00H
LCALL WRITE_DATA
MOV DATA2,#00H
LCALL WRITE_DATA
MOV COMMAND,#24H ;显示地址设置€
LCALL WRITE_COMMAND
MOV R2,#2
TIAO4: MOV R1,#255
NN: MOV DATA2,#83H
LCALL WRITE_DATA
DJNZ R1,NN
DJNZ R2,TIAO4
```

```
MOV    COMMAND,#0B2H
LCALL  WRITE_COMMAND
LCALL  DELAY

MOV    COMMAND,#0B0H
LCALL  WRITE_COMMAND
MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    COMMAND,#24H    ;显示地址设置€
LCALL  WRITE_COMMAND
MOV    R2,#2
TIAO5: MOV    R1,#255
NN1:   MOV    DATA2,#84H
LCALL  WRITE_DATA
DJNZ  R1,NN1
DJNZ  R2,TIAO5
MOV    COMMAND,#0B2H
LCALL  WRITE_COMMAND
LCALL  DELAY

MOV    COMMAND,#0B0H
LCALL  WRITE_COMMAND
MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    DATA2,#00H
LCALL  WRITE_DATA
MOV    COMMAND,#24H    ;显示地址设置€
```



```
        LCALL  WRITE_COMMAND
        MOV   R2,#2
TIAO6: MOV   R1,#255
NN2:   MOV   DATA2,#85H
        LCALL WRITE_DATA
        DJNZ  R1,NN2
        DJNZ  R2,TIAO6
        MOV   COMMAND,#0B2H
        LCALL WRITE_COMMAND
        LCALL DELAY

        MOV   DATA2,#00H
        LCALL WRITE_DATA
        MOV   DATA2,#00H
        LCALL WRITE_DATA
        MOV   COMMAND,#24H      ;显示地址设置
        LCALL WRITE_COMMAND
        MOV   R3,#00H
        MOV   R4,#20H
        MOV   COMMAND,#0B0H    ;自动写入
        LCALL WRITE_COMMAND
MM1:   MOV   R4,#20H
MM:    MOV   A,#00H
        MOV   DATA2,A
        LCALL WRITE_DATA
        DJNZ  R4,MM
        DJNZ  R3,MM1
        MOV   COMMAND,#0B2H    ;推出自动写入
        LCALL WRITE_COMMAND
```

#####

TIAN:

MOV DATA2,#00H ;显示地址

LCALL WRITE_DATA

MOV DATA2,#08H

LCALL WRITE_DATA

MOV COMMAND,#24H

LCALL WRITE_COMMAND

MOV COMMAND,#0B0H

LCALL WRITE_COMMAND

MOV R2,#10H

MOV R1,#000H

MOV DPTR,#TIANSHI

SSS: NOP

SSS1: CLR A

MOVC A,@A+DPTR

MOV DATA2,A

LCALL WRITE_DATA

INC DPTR

```

DJNZ  R1,SSS1
DJNZ  R2,SSS
MOV   COMMAND,#0B2H
LCALL WRITE_COMMAND
LCALL DELAY
LJMP  MAIN

```

```

;#####

```

```

TEST: DB 0FFH,00H,0FFH,00H,0FFH,00H,0FFH,00H
      DB 00H,0FFH,00H,0FFH,00H,0FFH,00H,0FFH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 55H,55H,55H,55H,55H,55H,55H,55H
      DB 0AAH,55H,0AAH,55H,0AAH,55H,0AAH,55H
      DB 55H,0AAH,55H,0AAH,55H,0AAH,55H,0AAH

```

TIANSHI :

```

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  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, 0C0H, 000H, 000H, 000H
DB  000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB  000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H, 0C0H, 000H, 000H, 000H, 000H
DB  000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB  000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H, 0C0H, 000H, 000H, 000H, 000H, 000H, 000H
DB  000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB  000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H, 0C8H, 000H, 000H, 000H, 000H, 080H, 003H, 0C0H, 000H, 000H
DB  001H, 082H, 004H, 008H, 000H, 040H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 080H, 080H, 000H, 000H

```

DB 000H, 000H, 000H, 003H, 0C4H, 0FFH, 0C2H, 040H, 080H, 040H, 03EH, 002H, 003H, 0FCH, 01EH, 022H
 DB 004H, 008H, 000H, 040H, 03EH, 0FEH, 002H, 020H, 03FH, 0FCH, 040H, 040H, 01FH, 0E0H, 03FH, 0E0H
 DB 07FH, 0E3H, 0C4H, 080H, 082H, 048H, 0BFH, 0FFH, 084H, 043H, 002H, 004H, 002H, 012H, 004H, 008H
 DB 01FH, 0FFH, 0A4H, 082H, 003H, 020H, 000H, 004H, 04FH, 0FCH, 010H, 020H, 020H, 020H, 000H, 003H
 DB 0C0H, 024H, 002H, 048H, 080H, 040H, 012H, 064H, 003H, 0FCH, 002H, 012H, 03FH, 07FH, 080H, 080H
 DB 024H, 0FEH, 002H, 020H, 000H, 004H, 012H, 040H, 01FH, 0E0H, 03FH, 0E0H, 000H, 003H, 0D0H, 023H
 DB 01FH, 0C8H, 080H, 040H, 00AH, 088H, 002H, 004H, 03FH, 082H, 004H, 008H, 001H, 000H, 028H, 082H
 DB 004H, 010H, 07FH, 0E5H, 012H, 040H, 010H, 020H, 020H, 020H, 000H, 003H, 0CAH, 0C1H, 002H, 048H
 DB 087H, 0FEH, 009H, 011H, 003H, 0FCH, 006H, 022H, 005H, 008H, 003H, 0FCH, 028H, 082H, 004H, 008H
 DB 000H, 004H, 0A4H, 0F8H, 01FH, 0E0H, 03FH, 0E1H, 0FFH, 0FBH, 0CAH, 008H, 002H, 048H, 084H, 042H
 DB 002H, 001H, 082H, 004H, 007H, 012H, 006H, 07FH, 006H, 004H, 024H, 0FEH, 008H, 00CH, 000H, 004H
 DB 0A4H, 088H, 010H, 020H, 008H, 080H, 004H, 003H, 0C2H, 008H, 002H, 048H, 084H, 042H, 03FH, 0E2H
 DB 000H, 000H, 00AH, 092H, 00CH, 042H, 00AH, 004H, 022H, 0A0H, 011H, 007H, 03FH, 0C4H, 02DH, 050H
 DB 000H, 000H, 008H, 080H, 004H, 003H, 0C4H, 0FFH, 082H, 048H, 084H, 042H, 002H, 005H, 01FH, 09FH
 DB 08AH, 003H, 0B4H, 022H, 013H, 0FCH, 022H, 092H, 021H, 082H, 020H, 044H, 057H, 030H, 0FCH, 0FCH
 DB 088H, 090H, 044H, 083H, 0C4H, 008H, 002H, 048H, 084H, 042H, 007H, 009H, 090H, 090H, 092H, 01EH
 DB 004H, 024H, 022H, 004H, 02AH, 094H, 001H, 000H, 020H, 044H, 044H, 0A0H, 084H, 084H, 048H, 090H
 DB 044H, 043H, 0D8H, 01CH, 007H, 048H, 084H, 042H, 006H, 081H, 01FH, 09FH, 0A2H, 0E2H, 004H, 018H
 DB 002H, 004H, 024H, 088H, 002H, 010H, 03FH, 0C5H, 0C4H, 040H, 0FCH, 0FCH, 028H, 0A0H, 084H, 023H
 DB 0C8H, 02BH, 018H, 088H, 084H, 042H, 00AH, 062H, 010H, 090H, 082H, 002H, 004H, 018H, 003H, 0FCH
 DB 020H, 088H, 004H, 008H, 020H, 044H, 044H, 060H, 084H, 084H, 028H, 0C1H, 004H, 033H, 0C8H, 0C9H
 DB 0C0H, 088H, 084H, 04AH, 012H, 044H, 010H, 090H, 082H, 002H, 004H, 024H, 002H, 004H, 020H, 084H
 DB 008H, 0FCH, 020H, 004H, 044H, 0A0H, 084H, 084H, 008H, 082H, 004H, 013H, 0CBH, 008H, 081H, 000H
 DB 084H, 044H, 022H, 018H, 01FH, 09FH, 082H, 002H, 004H, 043H, 082H, 024H, 020H, 0A3H, 09FH, 08CH
 DB 000H, 014H, 045H, 010H, 0FCH, 0FDH, 0FFH, 0FCH, 004H, 003H, 0C8H, 008H, 002H, 000H, 080H, 040H
 DB 002H, 060H, 010H, 090H, 082H, 002H, 015H, 081H, 002H, 01CH, 020H, 0C1H, 008H, 008H, 000H, 008H
 DB 046H, 01CH, 084H, 084H, 000H, 000H, 014H, 003H, 0C8H, 008H, 000H, 000H, 000H, 040H, 002H, 000H
 DB 000H, 000H, 002H, 002H, 008H, 000H, 002H, 008H, 020H, 080H, 000H, 000H, 000H, 000H, 044H, 008H
 DB 000H, 000H, 000H, 000H, 008H, 003H, 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
 DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H

DB 000H, 000H, 014H, 003H, 0C8H, 008H, 000H, 000H, 000H, 040H, 002H, 000H, 000H, 000H, 002H, 002H

DB 008H, 000H, 002H, 008H, 020H, 080H, 000H, 000H, 000H, 000H, 044H, 008H, 000H, 000H, 000H, 000H

DB 008H, 003H, 0C0H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H

DB 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

DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H, 0C8H, 000H

DB 000H, 000H, 000H, 080H, 003H, 0C0H, 000H, 000H, 001H, 082H, 004H, 008H, 000H, 040H, 000H, 000H

DB 000H, 000H, 000H, 000H, 080H, 080H, 000H, 000H, 000H, 000H, 000H, 003H, 0C4H, 0FFH, 0C2H, 040H

DB 080H, 040H, 03EH, 002H, 003H, 0FCH, 01EH, 022H, 004H, 008H, 000H, 040H, 03EH, 0FEH, 002H, 020H

DB 03FH, 0FCH, 040H, 040H, 01FH, 0E0H, 03FH, 0E0H, 07FH, 0E3H, 0C4H, 080H, 082H, 048H, 0BFH, 0FFH

DB 084H, 043H, 002H, 004H, 002H, 012H, 004H, 008H, 01FH, 0FFH, 0A4H, 082H, 003H, 020H, 000H, 004H

DB 04FH, 0FCH, 010H, 020H, 020H, 020H, 000H, 003H, 0C0H, 024H, 002H, 048H, 080H, 040H, 012H, 064H

DB 003H, 0FCH, 002H, 012H, 03FH, 07FH, 080H, 080H, 024H, 0FEH, 002H, 020H, 000H, 004H, 012H, 040H

DB 01FH, 0E0H, 03FH, 0E0H, 000H, 003H, 0C0H, 023H, 01FH, 0C8H, 080H, 040H, 00AH, 088H, 002H, 004H

DB 03FH, 082H, 004H, 008H, 001H, 000H, 028H, 082H, 004H, 010H, 07FH, 0E5H, 012H, 040H, 010H, 020H

DB 020H, 020H, 000H, 003H, 0CAH, 0C1H, 002H, 048H, 087H, 0FEH, 009H, 011H, 003H, 0FCH, 006H, 022H

DB 005H, 008H, 003H, 0FCH, 028H, 082H, 004H, 008H, 000H, 004H, 0A4H, 0F8H, 01FH, 0E0H, 03FH, 0E1H

DB 0FFH, 0FBH, 0CAH, 008H, 002H, 048H, 084H, 042H, 002H, 001H, 082H, 004H, 007H, 012H, 006H, 07FH

DB 006H, 004H, 024H, 0FEH, 008H, 00CH, 000H, 004H, 0A4H, 088H, 010H, 020H, 008H, 080H, 004H, 003H

DB 0C2H, 008H, 002H, 048H, 084H, 042H, 03FH, 0E2H, 000H, 000H, 00AH, 092H, 00CH, 042H, 00AH, 004H

DB 022H, 0A0H, 011H, 007H, 03FH, 0C4H, 02DH, 050H, 000H, 000H, 008H, 080H, 004H, 003H, 0C4H, 0FFH

DB 082H, 048H, 084H, 042H, 002H, 005H, 01FH, 09FH, 08AH, 003H, 0B4H, 022H, 013H, 0FCH, 022H, 092H

DB 021H, 082H, 020H, 044H, 057H, 030H, 0FCH, 0FCH, 088H, 090H, 044H, 083H, 0C4H, 008H, 002H, 048H

DB 084H, 042H, 007H, 009H, 090H, 090H, 092H, 01EH, 004H, 024H, 022H, 004H, 02AH, 094H, 001H, 000H

DB 020H, 044H, 044H, 0A0H, 084H, 084H, 048H, 090H, 044H, 043H, 0C8H, 01CH, 007H, 048H, 084H, 042H

DB 006H, 081H, 01FH, 09FH, 0A2H, 0E2H, 004H, 018H, 002H, 004H, 024H, 088H, 002H, 010H, 03FH, 0C5H

DB 0C4H, 040H, 0FCH, 0FCH, 028H, 0A0H, 084H, 023H, 0C8H, 02BH, 018H, 088H, 084H, 042H, 00AH, 062H

DB 010H, 090H, 082H, 002H, 004H, 018H, 003H, 0FCH, 020H, 088H, 004H, 008H, 020H, 044H, 044H, 060H

DB 084H, 084H, 028H, 0C1H, 004H, 033H, 0C8H, 0C9H, 0C0H, 088H, 084H, 04AH, 012H, 044H, 010H, 090H

DB 082H, 002H, 004H, 024H, 002H, 004H, 020H, 084H, 008H, 0FCH, 020H, 004H, 044H, 0A0H, 084H, 084H

DB 008H, 082H, 004H, 013H, 0CBH, 008H, 081H, 000H, 084H, 044H, 022H, 018H, 01FH, 09FH, 082H, 002H
DB 004H, 043H, 082H, 024H, 020H, 0A3H, 09FH, 08CH, 000H, 014H, 045H, 010H, 0FCH, 0FDH, 0FFH, 0FCH
DB 004H, 003H, 0C8H, 008H, 002H, 000H, 080H, 040H, 002H, 060H, 010H, 090H, 082H, 002H, 015H, 081H
DB 002H, 01CH, 020H, 0C1H, 008H, 008H, 000H, 008H, 046H, 01CH, 084H, 084H, 000H, 000H, 014H, 003H
DB 0C8H, 008H, 000H, 000H, 000H, 040H, 002H, 000H, 000H, 000H, 002H, 002H, 008H, 000H, 002H, 008H
DB 020H, 080H, 000H, 000H, 000H, 000H, 044H, 008H, 000H, 000H, 000H, 000H, 008H, 003H, 0C0H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 003H, 0C0H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 003H, 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 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 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H
DB 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H, 000H

