



MULTI-INNO TECHNOLOGY CO., LTD.

LCD MODULE SPECIFICATION

Model : MI128128F

Revision	
Engineering	
Date	
Our Reference	



GENERAL SPECIFICATION

128 X 128 dot display

Toshiba LCD controller: T6963C

Toshiba LCD driver: T6A39 and T6A40

Interface with 8-bit MPU (directly connected to Z80 serial MPU)

Display Specification

Display Mode: graphic, text and combination text-graphic mode

Display Dot: 128 X 128

Display type: STN and FSTN

Display color-Display background color: Black-Yellow Green; Blue-Gray; Black-White

Polarizer mode: Positive,Negative; Reflective , Transflective,Transmissive

Viewing angle: 6:00 and 12:00

Display duty: 1/128

Driving bias: 1/12

Selectable character font: 6 X 8 or 8 X 8

Memory and External Memory:

A standard 128-word character generator ROM

8KB or 32KB external display RAM (static RAM)

Mechanical characteristics (Unit:mm)

Extenal dimension: 92.0 X 106.0 X 12.0

92.0 X 106.0 X 16.5 for Bottom LED Backlight

View area: 73.0 X 73.0

Dot size: 0.5 X 0.5

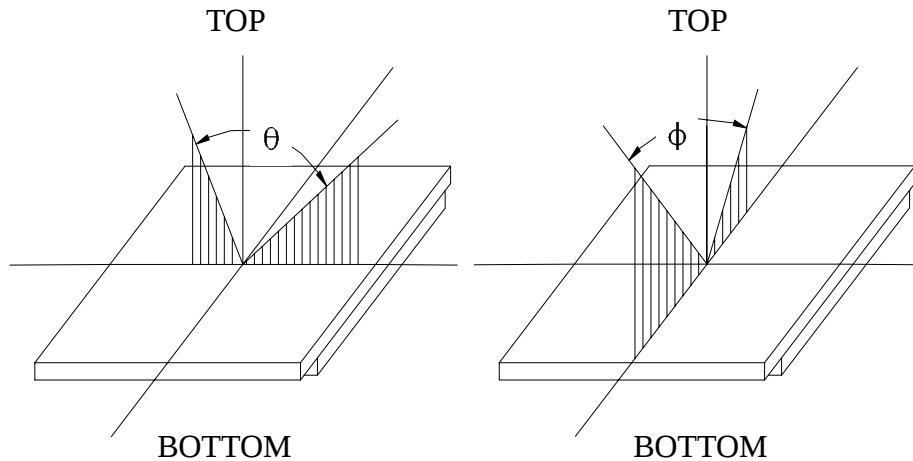
Dot pitch: 0.55 X 0.55

Weight:95g(127g for Bottom LED Backlight)

POWER: negative power, +5V power or single +5V power(negative power on module)

● Optical Characteristics

(1) Definition of viewing Angle



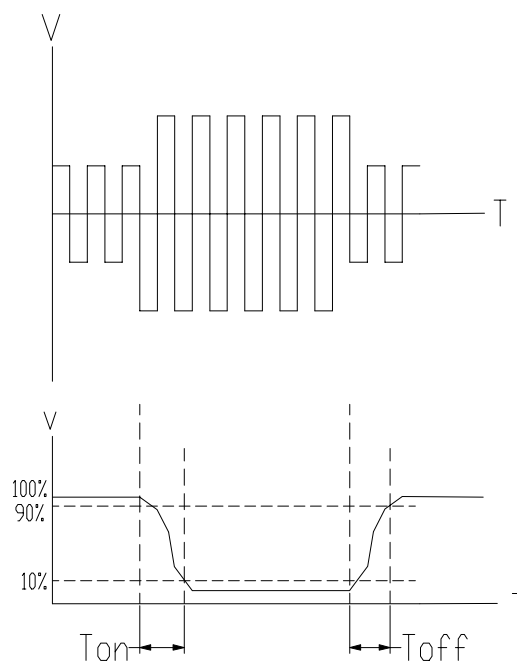
(2) Definition of Contrast Ratio:

$$\text{Contrast Ratio} = \frac{\text{Reflectance value of non-selected state brightness}}{\text{Reflectance value of selected state brightness}}$$

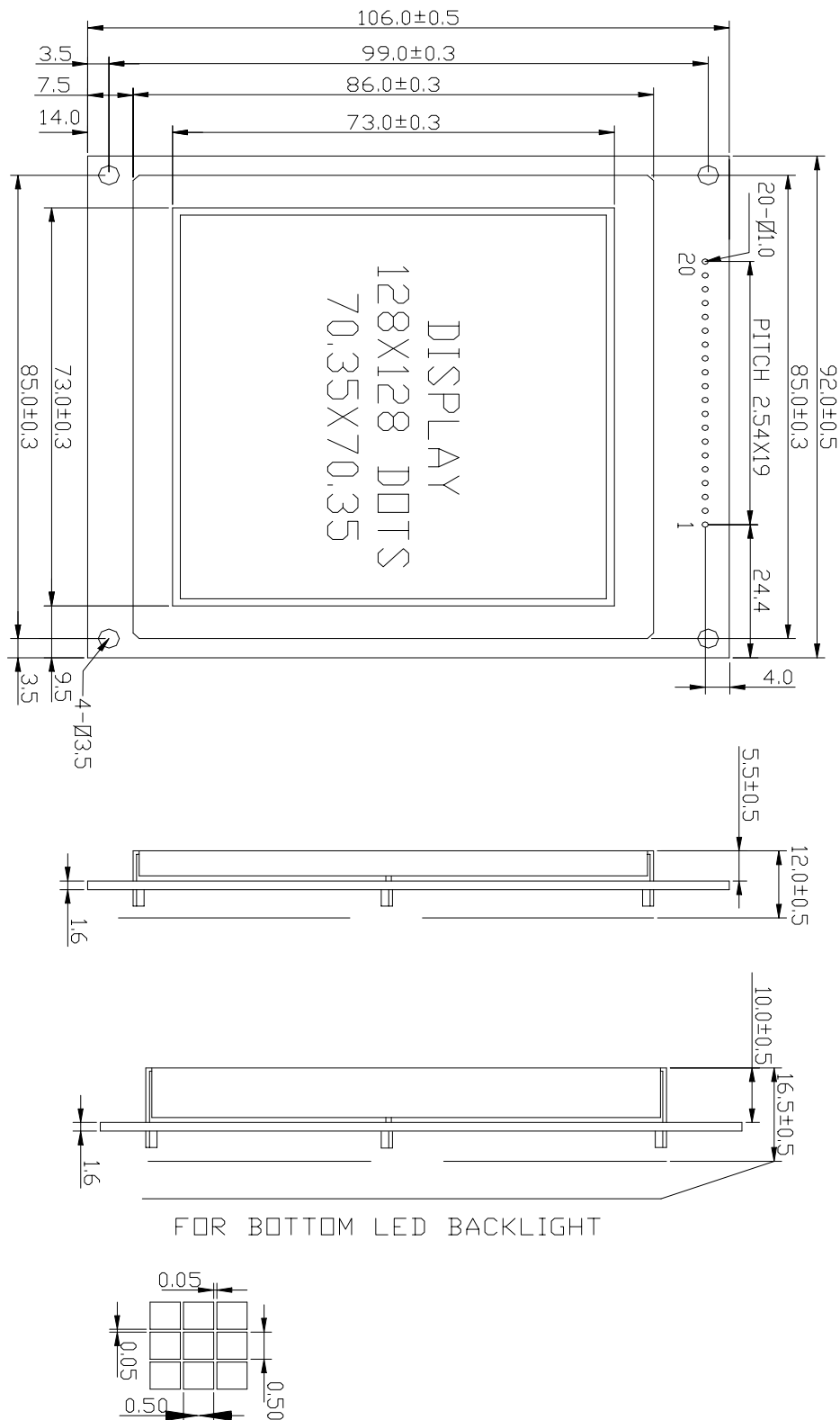
Test condition : standard A light source

(3) Response Time

Response time is measured as the shortest period of time possible between the change in state of an LCD segment as demonstrated below



● External Dimension



● Absolute Maximum Ratings For Bottom LED Backlight

Parameter	Symbol	Test condition	Min	Type	Max	Unit
LED Forward Consumption Current	I_f	$T_a=25^{\circ}\text{C}$ $V_f=4.1\text{V}$	-	530	-	mA
LED Allowable Dissipation	P_d		-	2200	-	mW

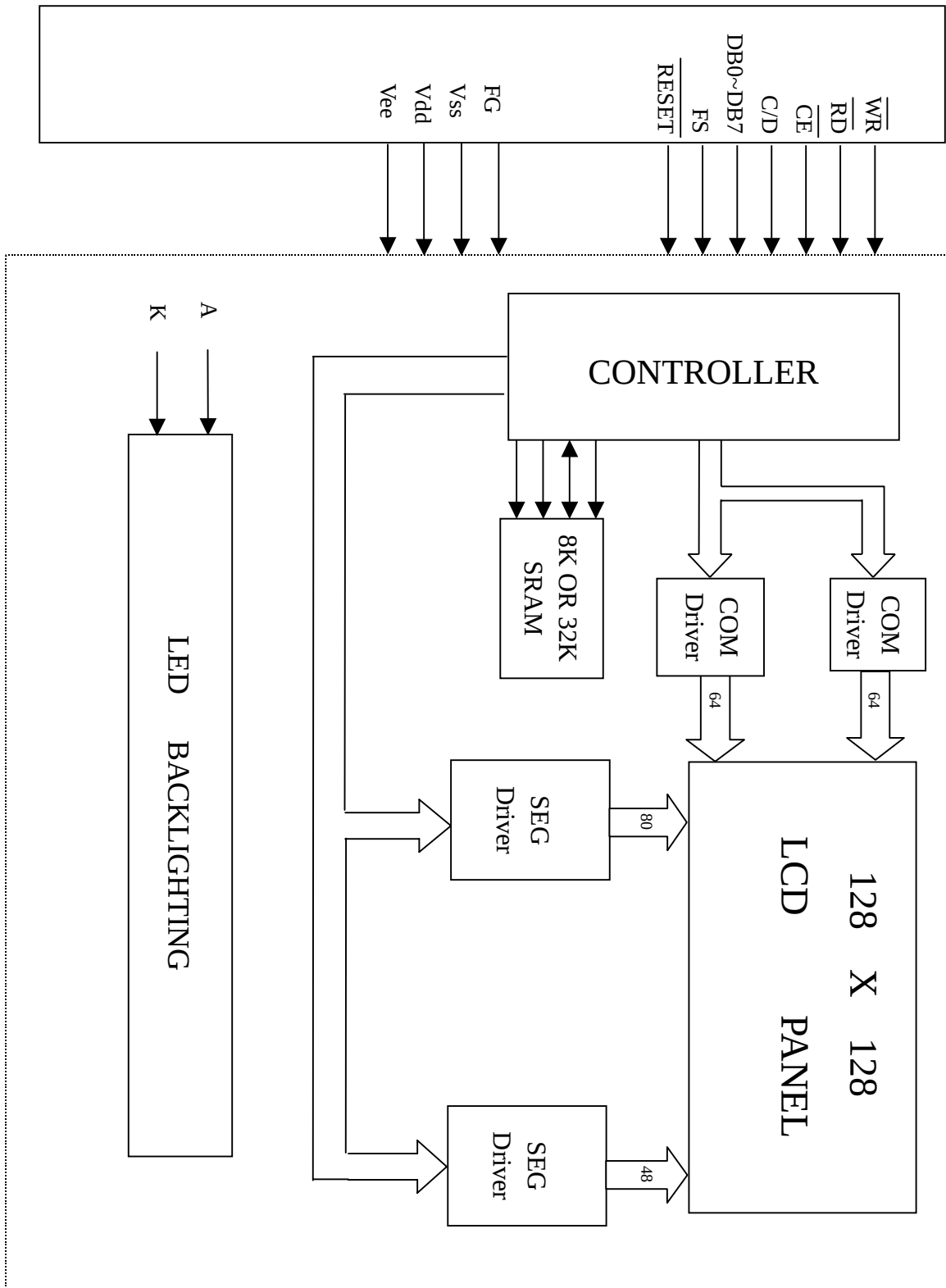
● Absolute Maximum Ratings

Item	Symbol	Condition	Standard Value		Unit
			Min	Max	
Supply Voltage for logic	Vdd	$T_a=25^{\circ}\text{C}$	-0.3	7.0	V
Supply Voltage for LCD	Vee		Vdd-30.0	Vdd+0.3	V
Input Voltage	V_i		-0.3	Vdd+0.3	V
Operating Temp(T)	Top	-	0	50	$^{\circ}\text{C}$
Storage Temp(T)	Tstg	-	-20	70	$^{\circ}\text{C}$
Operating Temp(HT)	HTop	-	-20	70	$^{\circ}\text{C}$
Storage Temp(HT)	HTstg	-	-30	80	$^{\circ}\text{C}$
Operating Temp(EHT)	EHTop	-	-30	80	$^{\circ}\text{C}$
Storage Temp(EHT)	EHTstg	-	-40	80	$^{\circ}\text{C}$

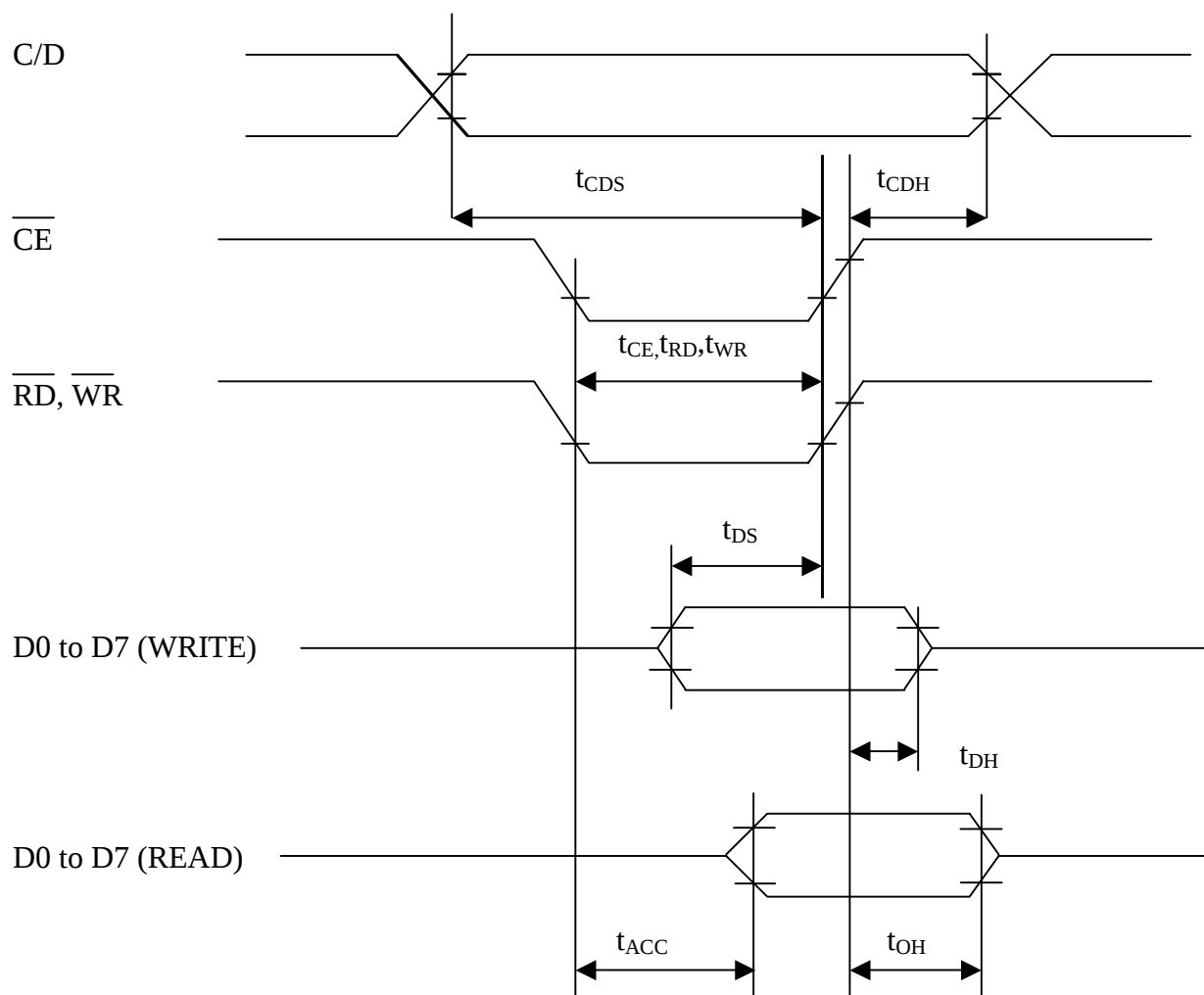
● Electrical Characteristics ($T_a=25^{\circ}\text{C}$, Vdd= 5.0V)

Item	Symbol	Condition	Standard Value			Unit
			min	Type	max	
Supply Voltage for logic	Vdd-Vss	-	4.5	5.0	5.5	V
Supply Current for logic	Idd	Vdd= 5.0	-	9.0	-	mA
Driving Current for LCD	Iee	Vee= -14	-	3.5	-	mA
Driving Voltage for LCD	Vdd-Vee	25°C	-	17.8	-	V
Input Voltage “H” level	V_{IH}	H	Vdd-2.2	-	Vdd	V
Input Voltage “L” level	V_{IL}	L	0	-	0.8	V

● Block Diagram



● Bus Timing



TEST CONDITIONS (Unless otherwise noted, $V_{DD}=5.0V \pm 10\%$, $V_{SS}=0V$, $T_a=-20$ to $75^\circ C$)

ITEM	SYMBOL	MIN	MAX	UNIT
C/D Set-up Time	t_{CDS}	100	-	ns
C/D Hold Time	t_{CDH}	10	-	ns
$\overline{CE}$, $\overline{RD}$, $\overline{WR}$ Pulse Width	t_{CE}, t_{RD}, t_{WR}	80	-	ns
Data Set-up Time	t_{DS}	80	-	ns
Data Hold Time	t_{DH}	40	-	ns
Access Time	t_{ACC}	-	150	ns
Output Hold Time	t_{OH}	10	50	ns

● Pin assignment

Pin NO.	Symbol	Function		Remark
1	FG	Module Frame Ground		
2	V _{ss}	Power Supply	0V	
3	V _{dd}		+5V	
4	V _{ee}		For LCD	NC when single +5V in use
5	$\overline{\text{WR}}$	Data Write		
6	$\overline{\text{RD}}$	Data Read		
7	$\overline{\text{CE}}$	Chip Enable		
8	$\overline{\text{C/D}}$	Command/Data Select		
9	RESET	Reset Signal		
10	DB0	Data Bit 0		
11	DB1	Data Bit 1		
12	DB2	Data Bit 2		
13	DB3	Data Bit 3		
14	DB4	Data Bit 4		
15	DB5	Data Bit 5		
16	DB6	Data Bit 6		
17	DB7	Data Bit 7		
18	FS	Font Selection		
19	A	Anode of LED Unit		
20	K	Cathode of LED Unit		

● Flowchart of communications with MPU

Status Word

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

STA0	Check command execution capability	0:Disable 1:Enable
STA1	Check data read/write capability	0:Disable 1:Enable
STA2	Check Auto mode data read capability	0:Disable 1:Enable
STA3	Check Auto mode data write capability	0:Disable 1:Enable
STA4	Not used	
STA5	Check controller operation capability	0:Disable 1:Enable
STA6	Error flag. Used for Screen Peek and Screen copy commands	0:No error 1>Error
STA7	Check the blink condition	0:Display off 1:Normal display

Note1: A status check must be performed before data is read or written.

Note2: It is necessary to check STA0 and STA1 at the same time.

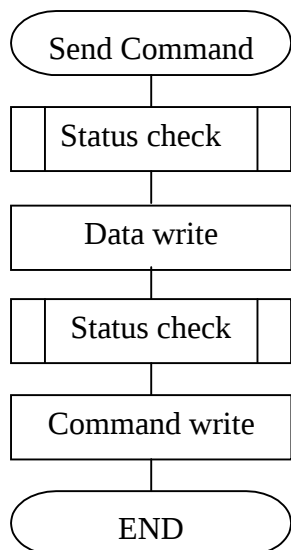
There is a possibility of erroneous operation due to a hardware interrupt.

Note3: For most modes STA0/STA1 are used as a status check.

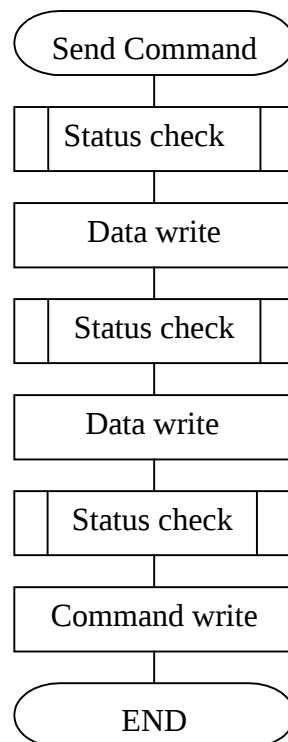
Note4: STA2 and STA3 are valid in Auto mode; STA0 and STA1 are invalid.

Setting Data

a) The case of 1 data



b) The case of 2 data



● COMMAND DEFINITIONS

COMMAND	CODE	D1	D2	FUNCTION
REGISTERS SETTING	00100001	X address	Y address	Set Cursor Pointer
	00100010	Data	00H	Set Offset Register
	00100100	Low address	High address	Set Address Pointer
SET CONTROL WORD	01000000	Low address	High address	Set Text Home Address
	01000001	Columns	00H	Set Text Area
	01000010	Low address	High address	Set Graphic Home Address
	01000011	Columns	00H	Set Graphic Area



COMMAND	CODE	D1	D2	FUNCTION
MODE SET	1000X000	—	—	OR mode
	1000X001	—	—	EXOR mode
	1000X011	—	—	AND mode
	1000X100	—	—	Text Attribute mode
	10000XXX	—	—	Internal CG ROM mode
	10001XXX	—	—	External CG RAM 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	—	—	Set Data Auto Write
	10110001	—	—	Set Data Auto Read
	10110010	—	—	Auto Reset
DATA READ/WRITE	11000000	Data	—	Data Write and Increment ADP
	11000001	—	—	Data Read and Increment ADP
	11000010	Data	—	Data Write and Decrement ADP
	11000011	—	—	Data Read and Decrement ADP
	11000100	Data	—	Data Write and Non-variable ADP
	11000101	—	—	Data Read and Non-variable ADP
SCREEN PEEK	11100000	—	—	Screen Peek
SCREEN COPY	11101000	—	—	Screen Copy
BIT SET/RESET	11110XXX	—	—	Bit Reset
	11111XXX	—	—	Bit Set
	1111X000	—	—	Bit 0 (LSB)
	1111X001	—	—	Bit1
	1111X010	—	—	Bit2
	1111X011	—	—	Bit3
	1111X100	—	—	Bit4
	1111X101	—	—	Bit5
	1111X110	—	—	Bit6
	1111X111	—	—	Bit 7 (MSB)

Note: First set the data, then set the command.

● Description of Command

Setting Registers

(1) Set Cursor Pointer

The position of the cursor is specified by X ADRS and Y ADRS. The cursor position can only be moved by this command. Data read/write from the MPU never changes the cursor pointer.

X ADRS: 00H to 4FH(lower 7 bits are valid); Y ADRS: 00H to 1FH(lower 5 bits are valid)

a) Single-Scan

X ADRS 00 to 4FH

Y ADRS 00H to 0FH

b) Dual-Scan

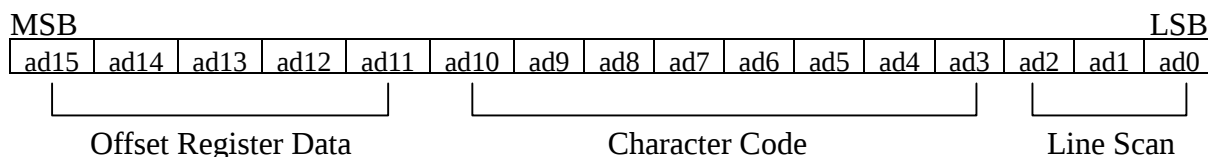
X ADRS 00H to 4FH

Y ADRS 00H to 0FH
Upper screen

Y ADRS 10H to 1FH
Lower screen

(2) Set Offset Register

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



The senior five bits define the start address in external memory of the CG RAM area. The next eight bits represent the character code of the character. In internal CG ROM mode, character codes 00H to 7FH represent the predefined “internal” CG ROM characters, and codes 80H to FFH represent the user’s own “external” characters. In external CG RAM mode, all 256 codes from 00H to FFH can be used to represent the user’s own characters.

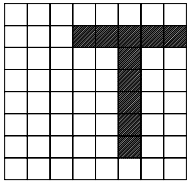
The three least significant bits indicate one of the eight rows of eight dots that define the character’s shape.

The relationship between display RAM address and offset register

Offset register data	CG RAM hex. Address (start to end)
00000	0000 to 07FFH
00001	0800 to 0FFFH
00010	1000 to 17FFH
11100	E000 to E7FFH
11101	E800 to EFFFH
11110	F000 to F7FFH
11111	F800 to FFFFH

Example1:

Offset register	02H
Character code	80H
Character generator RAM start address	0001 0100 0000 0000
	1 4 0 0 H

	(address)	(data)
	1400H	00H
	1401H	1FH
	1402H	04H
	1403H	04H
	1404H	04H
	1405H	04H
	1406H	04H
	1407H	00H

Example 2:

The relationship between display RAM data and display characters:

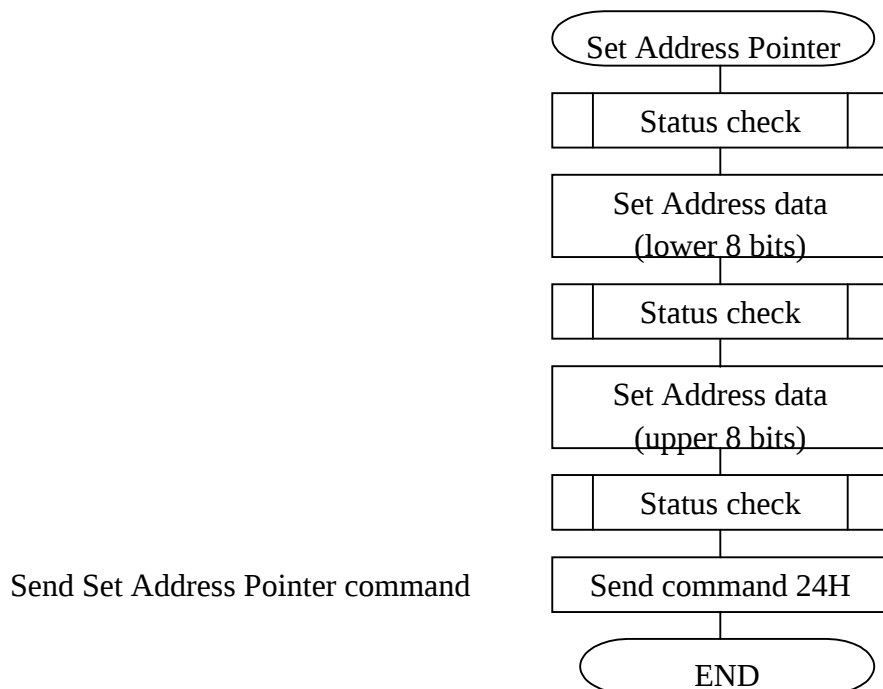
	(RAM DATA)	(Character)
AB γ DE ζ GHUKLM	21H	A
	22H	B
	83H	γ
	24H	D
Display character	25H	E
	86H	ζ

* γ and ζ are displayed by character generator RAM

(3) Set Address Pointer

The Set Address Pointer command is used to indicate the start address for writing to (or reading from) external RAM.

The Flowchart for set Address Pointer command



Set Control Word

The home address and column size are defined by this command.

(1) Set Text Home Address

The starting address in the external display RAM for text display is defined by this command. The text home address indicates the leftmost and uppermost position.

The relationship between external display RAM address and display position.

TH	-----	TH+CL
TH+TA	-----	TH+TA+CL
(TH+TA)+TA	-----	TH+2TA+CL
-----	-----	-----
TH+(N-1)TA	-----	TH+(N-1)TA+CL

TH: Text home address TA: Text area number (columns)

CL: Columns are fixed by hardware (pin-programmable).

Example:

Text home address: 0000H Text area: 0020H

MD2=H, MD3=H: 32 columns DUAL=H, MDS=L, MD0=L, MD1=H: 4lines

0000H	0001H	-----	001EH	001FH
0020H	0021H	-----	003EH	003FH
0040H	0041H	-----	005EH	005FH
0060H	0061H	-----	007EH	007FH

(2) Set Graphic Home Address

The starting address of the external display RAM used for graphic display is defined by this command. The graphic home address indicates the leftmost and uppermost position. The relationship between external display RAM address and display position

GH	-----	GH+CL
GH+GA	-----	GH+GA+CL
(GH+GA)+GA	-----	GH+2GA+CL
-----	-----	-----
GH+(N-1)GA	-----	GH+(N-1)GA+CL

GH: Graphic home address GA: Graphic area number (columns)

CL: Columns are fixed by hardware (pin-programmable).

Example:

Graphic home address: 0000H Graphic area: 0020H

MD2=H, MD3=H: 32 columns DUAL=H, MDS=L, MD0=H, MD1=H: 2lines

0000H	0001H	-----	001EH	001FH
0020H	0021H	-----	003EH	003FH
0040H	0041H	-----	005EH	005FH
0060H	0061H	-----	007EH	007FH
0080H	0081H	-----	009EH	009FH
00A0H	00A1H	-----	00BEH	00BFH
00C0H	00C1H	-----	00DEH	00DFH
00E0H	00E1H	-----	00FEH	00FFH
0100H	0101H	-----	011EH	011FH
0120H	0121H	-----	013EH	013FH
0140H	0141H	-----	015EH	015FH
0160H	0161H	-----	017EH	017FH
0180H	0181H	-----	019EH	019FH
01A0H	01A1H	-----	01BEH	01BFH
01C0H	01C1H	-----	01DEH	01DFH
01E0H	01E1H	-----	01FEH	01FFH

(3) Set Text Area

This command can be used to define the columns of the test display.

Example:

LCD size: 20 columns, 4lines Text home address: 0000H

Text area: 0014H MD2=H, MD3=H: 32 columns

DUAL=H, MDS=L, MD0=L, MD1=H: 4lines

0000	0001	-----	0013	0014	-----	001F
0014	0015	-----	0027	0028	-----	0033
0028	0029	-----	003B	003C	-----	0047
003C	003D	-----	004F	0050	-----	005B



(4) Set Graphic Area

This command can be used define the columns of the graphic display.

Example: LCD size: 20 columns, 2lines Graphic home address: 0000H
Graphic area: 0014H MD2=H, MD3=H: 0014H
DUAL=H, MDS=L, MD0=H, MD1=H: 2lines

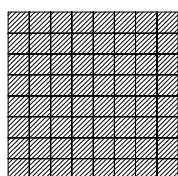
0000	0001	-----	0013	0014	-----	001F
0014	0015	-----	0027	0028	-----	0033
0028	0029	-----	003B	003C	-----	0047
003C	003D	-----	004F	0050	-----	005B
0050	0051	-----	0063	0064	-----	006F
0064	0065	-----	0077	0078	-----	0083
0078	0079	-----	008B	008C	-----	0097
008C	008D	-----	009F	00A0	-----	00AB
00A0	00A1	-----	00B3	00B4	-----	00BF
00B4	00B5	-----	00C7	00C8	-----	00D3
00C8	00C9	-----	00DB	00DC	-----	00E7
00DC	00DD	-----	00EF	00F0	-----	00FD
00F0	00F1	-----	0103	0104	-----	011F
0104	0105	-----	0127	0128	-----	0123
0128	0129	-----	013B	013C	-----	0147
013C	013D	-----	014F	0150	-----	015B



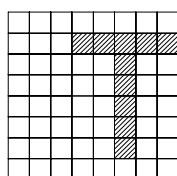
Mode Set

The display mode does not change until the next command is sent. In Internal Character Generator mode, character codes 00H to 7FH are assigned to the built-in character generator RAM. The character codes 80H to FFH are automatically assigned to the external character generator RAM.

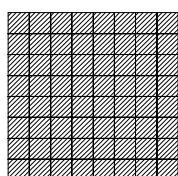
Example: (Note: Attribute functions can only be applied to text display, since the attribute data is placed in the graphic RAM area.)



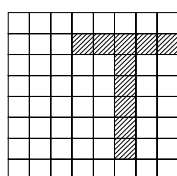
GRAPHIC



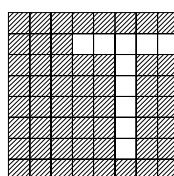
TEXT



“OR”



“AND”



“EXOR”

Attribute Function

The attribute operations are Reverse display, Character blink and inhibit. The attribute data is written into the graphic area which was defined by the Set Control Word command. Only text display is possible in Attribute function mode; graphic display is automatically disabled. However, the Display Mode command must be used to turn both Text and Graphic on in order for the Attribute function to be available.

Attribute RAM 1byte

X	X	X	X	d3	d2	d1	d0
---	---	---	---	----	----	----	----

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	1	Blink of reverse display
1	0	1	1	Blink of inhibit of display

X: invalid

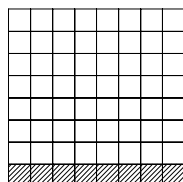
Display Mode

It is necessary to turn on “Text display” and “Graphic display” in the following cases.

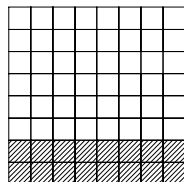
- Combination of text/graphic display
- Attribute function

Cursor Pattern Select

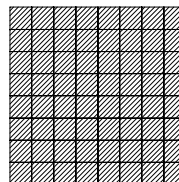
When cursor display is ON, this command selects the cursor pattern in the range 1 line to 8lines. The cursor address is defined by the Cursor Pointer Set command.



1-line cursor



2-line cursor



8-line cursor

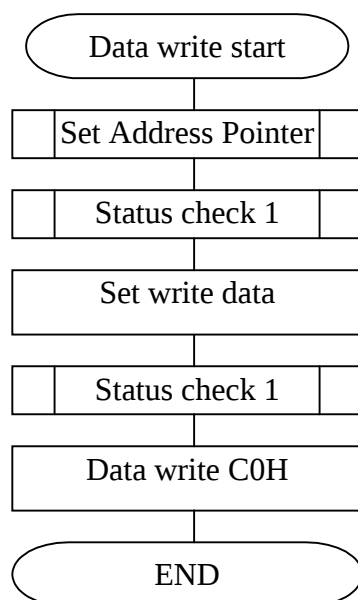
Data Auto Read/Write

This command is convenient for sending a full screen of data from the external display RAM. After setting Auto mode, a Data Write (or Read) command is need not be sent between each datum. A Data Auto Write (or Read) command must be sent after a Set Address Pointer command. After this command, the address pointer is automatically incremented by 1 after each datum. In Auto mode, the LCM cannot accept any other commands. The Auto Rest command must be sent to the LCM after all data has been sent, to clear Auto mode.

Data Read/Write

This command is used for writing data from the MPU to external display RAM, and reading data from external display RAM to the MPU. Data Write/Read should be executed after setting address using Set Address Pointer command. The address pointer can be automatically incremented or decrement using this command.

Note : This command is necessary for each 1-byte datum.
Refer to the following flowchart



Screen Peek

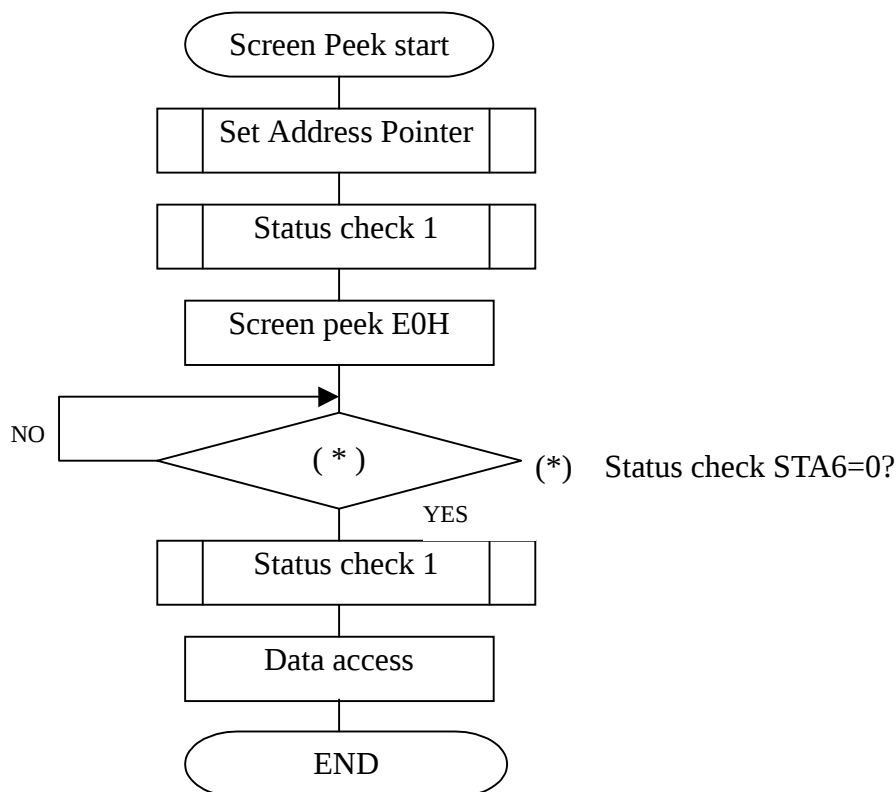
The command is used to transfer 1 byte of displayed data to the data stack ; this byte can then be read from the MPU by data access. The logical combination of text and graphic display data on the LCD screen can be read by this command.

The status (STA6) should be checked just after the Screen Peek command. If the address determined by the Set Address Pointer command is not in the graphic area, this command is ignored and a status flag (STA6) is set.

Refer to the following flowchart

Note: This command is available when hardware column number and software column number are the same. Hardware column number is related to MD2 and MD3 setting.

Software column number is related to Set Text Area and Set Graphic Area command.

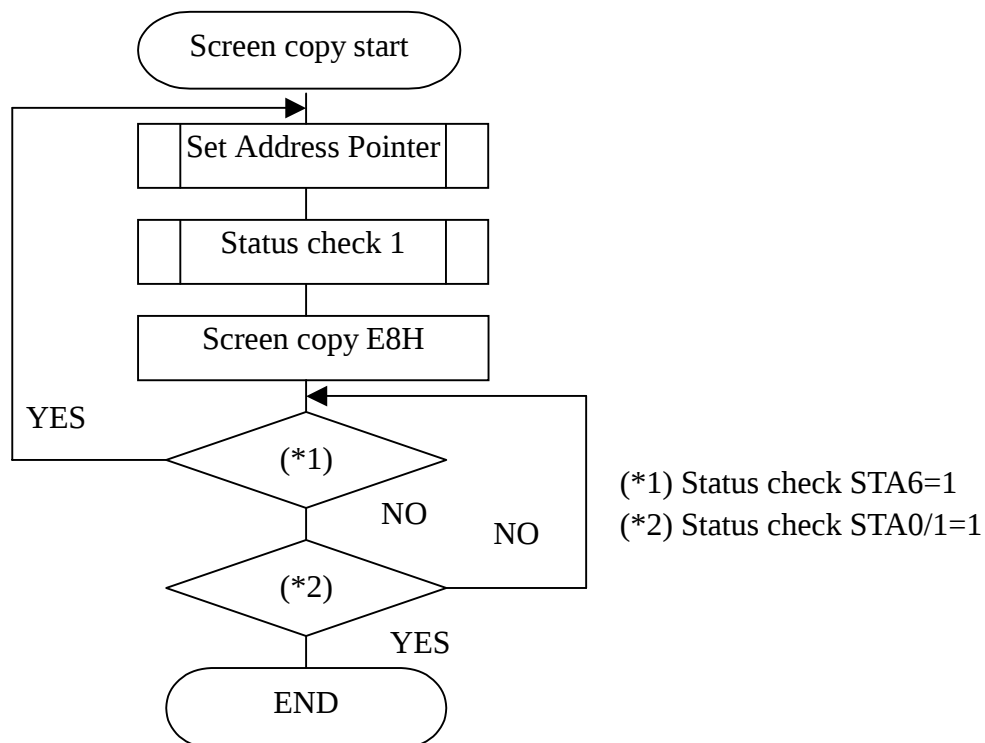


(Note) This command is available when hardware column number and software column number are the same. Hardware column number is related to MD2 and MD3 setting. Software column number is related to Set Text Area and Set Graphic Area command.

Screen Copy

This command copies a single raster line of data to the graphic area.
The start point must be set using the Set Address Pointer command.

Refer to the following flowchart :



Note1: If the attribute function is being used, this command is not available.(With Attribute data is graphic area data)

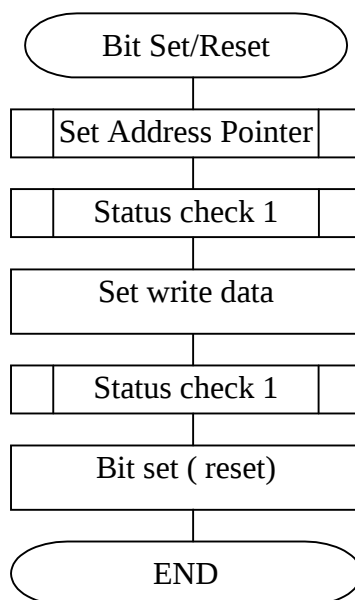
Note2: With Dual-Scan, this command cannot be used (because the LCM cannot separate the upper screen data and lower screen data).

Note3: This command is available when hardware column number and software column number are the same.

Bit Set/Reset

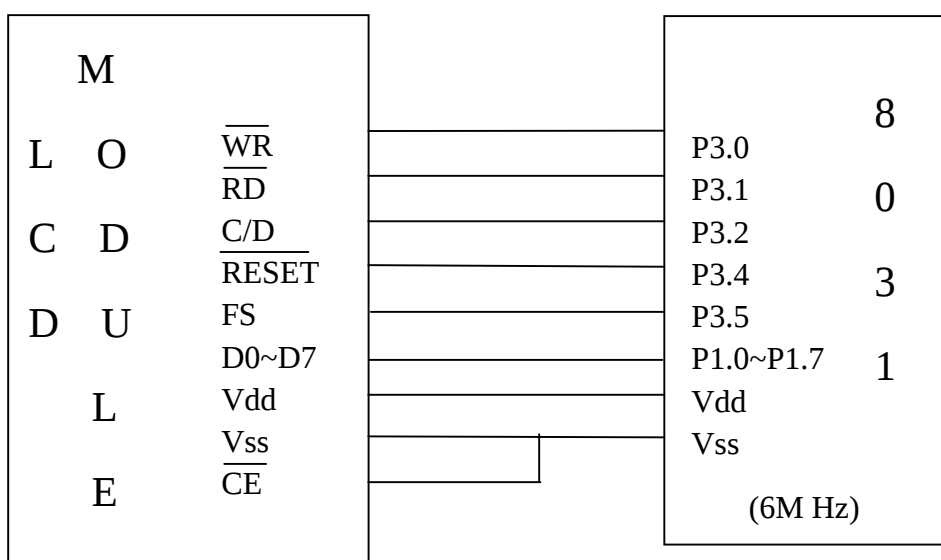
This command use to set or reset a bit of the byte specified by the address pointer. Only one bit can be set/rest at a time.

Refer to the following flowchart:



● Application Example

Application Circuit





Program Example

```
;*****THIS IS 51 PROGRAM(128128)*****  
; P3.0 P3.1 P3.2 P3.4 P3.5  
; WR RD C/D SET FS1  
;*****  
WR EQU P3.0  
RD EQU P3.1  
C/D EQU P3.2  
SET EQU P3.4  
FS1 EQU P3.5  
X EQU 30H  
Y EQU 31H  
;*****  
ORG 0000H  
  
SETB SET  
MOV R5,#01H  
REST: CLR SET  
LCALL T1  
SETB SET  
LCALL T1  
DJNZ R5,REST  
SETB WR  
SETB RD  
SETB C/D  
CLR FS1  
;-----INIT SET 0  
MAIN: LCALL AA3  
MOV P1,#00H  
LCALL DAT  
MOV P1,#00H  
LCALL DAT  
MOV P1,#42H  
LCALL INS  
MOV R6,#40H  
;-----CLEAR 00  
MOV P1,#98H  
LCALL INS  
MOV P1,#80H  
LCALL INS  
LCALL AA3
```

```
        MOV    R6,  #80H
LOOP1: MOV    DPTR, #TAB6
        LCALL   A5
        DJNZ    R6,  LOOP1
        MOV     P1, #00H
        LCALL   DAT
        MOV     P1, #00H
        LCALL   DAT
        MOV     P1, #24H
        LCALL   INS
        MOV     P1, #0F0H
        LCALL   INS
;-----FULL GRAP;
        MOV     P1, #98H
        LCALL   INS
        MOV     P1, #80H
        LCALL   INS
        LCALL   AA3
        MOV     R6,  #80H
LOOP2: MOV    DPTR, #TAB5
        LCALL   A5
        DJNZ    R6,  LOOP2
;-----ONE MEI HUA
        MOV     P1,  #98H
        LCALL   INS
        MOV     P1,  #80H
        LCALL   AA3
        MOV     R6,  #40H
LOP:    MOV    DPTR, #TAB7
        LCALL   AAA5
        DJNZ    R6,  LOP
;-----CLP ONE MEI HUA
        MOV     P1,  #98H
        LCALL   INS
        MOV     P1,  #80H
        LCALL   AA3
        MOV     R6,  #40H
LOP24: MOV    DPTR, #TAB8
        LCALL   AAA5
        DJNZ    R6,  LOP24
;-----FOUR MEI HUA
```



```

        CLR    FS1
        MOV    P1, #98H
        LCALL  INS
        MOV    P1, #80H
        LCALL  AA3
        MOV    R6, #20H
LOP23:  MOV    DPTR,#TAB9
        LCALL  AAA6
        DJNZ   R6, LOP23
;-----CLP FOUR MEI HUA
        CLR    FS1
        MOV    P1, #98H
        LCALL  INS
        MOV    P1, #80H
        LCALL  AA3
        MOV    R6, #20H
LOP22:  MOV    DPTR,#TB10
        LCALL  AAA6
        DJNZ   R6, LOP22
;-----DISP 1,2,3...
;-----SETB 1BYTE,CURSOR ON
        SETB   FS1
        MOV    P1,#97H
        LCALL  INS
        MOV    P1,#80H
        LCALL  INS
        MOV    P1,#0A7H
        LCALL  INS
        MOV    Y, #00H
        LCALL  AA1
        MOV    R6, #10H
LOOP3:  MOV    DPTR, #TAB1
        LCALL  AA5
        INC    Y
        DJNZ   R6, LOOP3
        MOV    P1,#27H
        LCALL  DAT
        MOV    P1,#0FH
        LCALL  DAT
        MOV    P1,#21H
        LCALL  INS
```



```
MOV P1,#00H
LCALL DAT
MOV P1,#10H
LCALL DAT
MOV P1,#24H
LCALL INS
MOV P1,#0FAH
LCALL INS
LCALL T3
;-----SET ATTRIBUTE,THREE MODE
MOV P1,#98H
LCALL INS
MOV P1,#80H
LCALL INS
LCALL AA3
MOV R6, #0C8H
LOOP4: MOV P1,#05H
LCALL DAT
MOV P1,#0C0H
LCALL INS
DJNZ R6, LOOP4
MOV R6, #0C8H
LOOP5: MOV P1,#08H
LCALL DAT
MOV P1,#0C0H
LCALL INS
DJNZ R6, LOOP5
MOV R6, #0F0H
LOOP6: MOV P1,#03H
LCALL DAT
MOV P1,#0C0H
LCALL INS
DJNZ R6, LOOP6
MOV P1,#9CH
LCALL INS
MOV P1,#84H
LCALL INS
LCALL T3
LCALL T3
;-----20 CHAR MOVE
MOV R6, #14H
```

```
        MOV  R3, #00H
LOOP7: MOV  P1,R3
        LCALL DAT
        MOV  P1,#18H
        LCALL DAT
        MOV  P1,#40H
        LCALL INS
        INC  R3
        DJNZ R6,  LOOP7
        LCALL T3
```

```
;-----
ZZZ:   CLR  A
        MOV  R5, #10H
ZZZZ:  CLR  C/D
        CLR  A
        MOVC A, @A+DPTR
        MOV  P1,A
        CLR  WR
        SETB WR
        LCALL T1
        INC  DPTR
        SETB C/D
        MOV  P1,#0C0H
        CLR  WR
        SETB WR
        DJNZ R5,  ZZZZ
        RET
```



```
,*****  
DAT:  CLR  C/D  
      CLR  WR  
      SETB WR  
      RET  
INS:  SETB  C/D  
      CLR  WR  
      SETB WR  
      LCALL T1  
      RET  
AA1:  MOV  P1,#0FFH ;SET CHAR MODE  
      LCALL DAT  
      MOV  P1,#17H  
      LCALL DAT  
      MOV  P1,#40H  
      LCALL INS  
      MOV  P1,#0FFH  
      LCALL DAT  
      MOV  P1,#17H  
      LCALL DAT  
      MOV  P1,#24H  
      LCALL INS  
      MOV  P1,#10H  
      LCALL DAT  
      MOV  P1,#00H  
      LCALL DAT  
      MOV  P1,#41H  
      LCALL INS  
      RET  
AA3:  MOV  P1,#00H ;SET GRAPHIC MODE  
      LCALL DAT  
      MOV  P1,#00H  
      LCALL DAT  
      MOV  P1,#42H  
      LCALL INS  
      MOV  P1,#00H  
      LCALL DAT  
      MOV  P1,#00H  
      LCALL DAT  
      MOV  P1,#24H  
      LCALL INS
```

```
MOV P1,#10H
LCALL DAT
MOV P1,#00H
LCALL DAT
MOV P1,#43H
LCALL INS
RET
AAAA: MOV P1,#98H
LCALL INS
MOV P1,#80H
LCALL INS
MOV P1,#00H
LCALL DAT
MOV P1,#00H
LCALL DAT
MOV P1,#24H
LCALL INS
MOV P1,#10H
LCALL DAT
MOV P1,#00H
LCALL DAT
MOV P1,#43H
LCALL INS
RET
AA5: CLR A ;MOV 40 BYTE
MOV R5,#10H
MOV X,#00H
CDJ2: CLR A
INC X
MOVC A,@A+DPTR
MOV P1,A
LCALL DAT
MOV P1,#0C0H
INC DPTR
LCALL INS
MOV P1,X
LCALL DAT
MOV P1,Y
LCALL DAT
MOV P1,#21H
LCALL INS
```

```
        DJNZ    R5,CDJ2
        RET
;*****
A5:     CLR     A
        MOV     R5,#10H
ADJ2:   CLR     C/D
        CLR     A
        MOVC    A,@A+DPTR
        MOV     P1,A
        CLR     WR
        SETB    WR
        LCALL   T1
        INC     DPTR
        SETB    C/D
        MOV     P1,#0C0H
        CLR     WR
        SETB    WR
        DJNZ    R5,ADJ2
        RET
;*****
AAA5:   CLR     A
        MOV     R5,  #20H
ZDJ2:   CLR     C/D
        CLR     A
        MOVC    A,  @A+DPTR
        MOV     P1,  A
        CLR     WR
        SETB    WR
        LCALL   T1
        INC     DPTR
        SETB    C/D
        MOV     P1,  #0C0H
        CLR     WR
        SETB    WR
        DJNZ    R5,  ZDJ2
        RET
;*****
AAA6:   CLR     A
        MOV     R5,  #040H
ZDJ6:   CLR     C/D
        CLR     A
```

```
        MOVC  A,    @A+DPTR
        MOV   P1,   A
        CLR   WR
        SETB  WR
        LCALL T1
        INC   DPTR
        SETB  C/D
        MOV   P1,   #0C0H
        CLR   WR
        SETB  WR
        DJNZ  R5,   ZDJ6
        RET

;*****
;
;                               DELAY 1MS
;*****
T2:  MOV  R7,#04H
TTT2: MOV  R2,#0FFH
TTT1: DJNZ R2,TTT1
      DJNZ R7,TTT2
      RET

;*****
;
;                               DELAY 64MS
;*****
T3:  MOV  R7,#0FFH
TTT3: MOV  R2,#0FFH
TTT4: DJNZ R2,TTT4
      DJNZ R7,TTT3
      RET

;*****
;
;                               DELAY 0.25MS
;*****
T1:  MOV  R7,#01H
TT2: MOV  R2, #0FH
TT1: DJNZ R2,TT1
      DJNZ R7,TT2
      RET

;*****
;
;                               DELAY 2S
;*****
T4:  MOV  R7, #20H
TTTT3:MOV R2,#0FFH
```



```
TTTT2:MOV R3,#0FFH
TTTT1:DJNZ R3,TTTT1
      DJNZ R2,TTTT2
      DJNZ R7,TTTT3
      RET
```

```
,*****
```

```
TAB1:DB 11H,12H,13H,14H,15H,16H,17H,18H
      DB 19H,21H,22H,23H,24H,25H,26H,27H
      DB 28H,29H,2AH,2BH,2CH,2DH,2EH,2FH
      DB 30H,31H,32H,33H,34H,35H,36H,37H
      DB 38H,39H,3AH,3BH,3CH,3DH,3EH,3FH
```

```
TAB5:DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
      DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
```

```
TAB6:DB 00H,00H,00H,00H,00H,00H,00H,00H
      DB 00H,00H,00H,00H,00H,00H,00H,00H
      DB 00H,00H,00H,00H,00H,00H,00H,00H
      DB 00H,00H,00H,00H,00H,00H,00H,00H
      DB 00H,00H,00H,00H,00H,00H,00H,00H
```

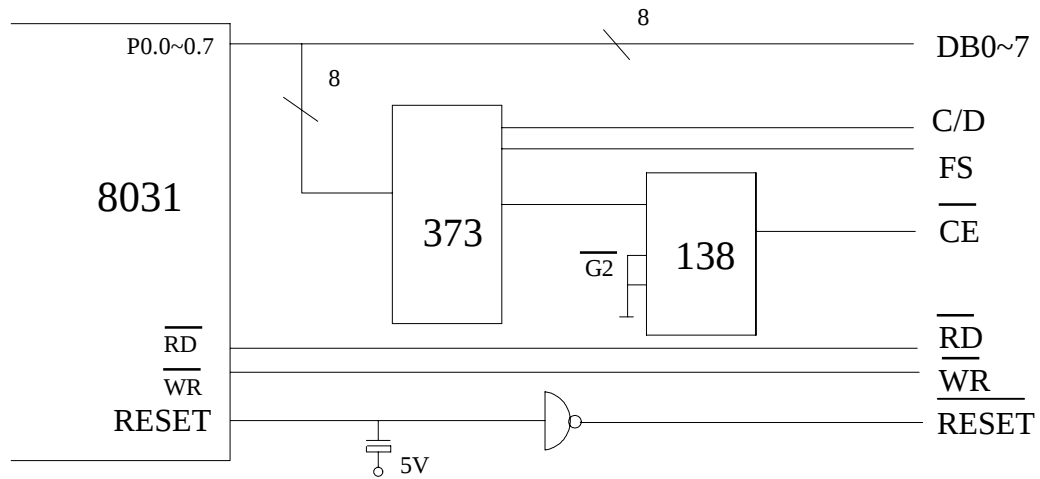
```
TAB7:DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
```

```
TAB8:DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 55H,55H,55H,55H,55H,55H,55H,55H,55H,55H
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
      DB 0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH,0AAH
```

```
TAB9:DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
      DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
      DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
      DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
```

```
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
TB10:DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 33H,33H,33H,33H,33H,33H,33H,33H,33H,33H
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
DB 0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH
END
```

● Application Circuit



● Character Code Map

ROM Code 0101

LSB MSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
1	0	1	2	3	4	5	6	7	8	9	:	;	=	>	?@	
2	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
3	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	
4	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
6	0	1	2	3	4	5	6	7	8	9	:	;	=	>	?@	
7	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P