Product Instructions

Product Type: Character LCD display module

Product Model: 1602A

client:

Customer

Number: Date:

Confirm (stamp)

Manufacturer	client

Table of contents

1 . Revision History

Version	issue date	New/revised content
V1.0	2020-8-12	New system

Note: The upgraded version is backward compatible without further notice.

If compatibility issues affect performance, please contact our company for resolution

2. Overview

1602A character LCD module is a dot matrix LCD module specially used to display letters, numbers, symbols, etc. It has 4-bit and 8-bit data transmission modes. It provides 5×7 dot matrix + cursor display mode. It provides display data buffer.
DDRAM, character generator CGROM and character generator CGRAM, CGRAM can be used to store the font data of up to 8 5×8 dot matrix graphic characters defined by yourself. Provides a rich command setting: clear display; cursor return to origin; display on/off; cursor on/off; display character flashing; cursor shift; display shift element, etc. Provides internal power-on automatic reset circuit, when external power is applied, the module is automatically initialized and the module is set to the default display working state.

Display characters: 16 characters x 2 lines **Character matrix: 5X7 character matrix + cursor**

Display color and backlight color: STN blue, yellow-green, gray; backlight black, white, yellow-green

Polarizing film: fully transparent/

semi-transparent

Observation angle: 6:00 Display duty cycle: 1/16 Drive

bias: 1/5 Control chip: SPLC780D or compatible IC (such as AIP31066)

Character generator ROM (CGROM): 10880 bits (192 character 5*8 dots) or (64 character 5*11 dots) Character

generator RAM (CGRAM): 64X8 bits
(8 characters 5*8 dots) or (4 characters 5*11 dots) Display data RAM (DDRAM): 80X8 bits (80 characters

max) **Dimensions (Unit: mm)**

Dimensions: 80X36X11

Viewable area: 64.5X13.8 Character font: 5X7 dots + cursor

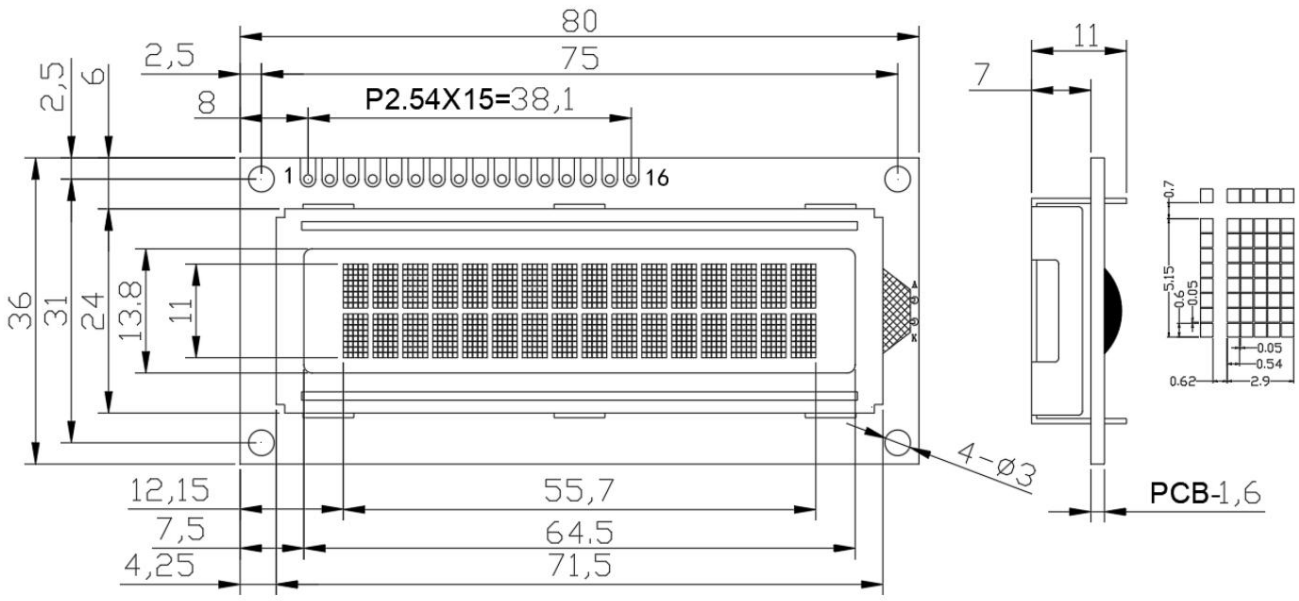
Character size: 55.7X11 Point size: 0.54X0.6

Character spacing: 3.52X5.85

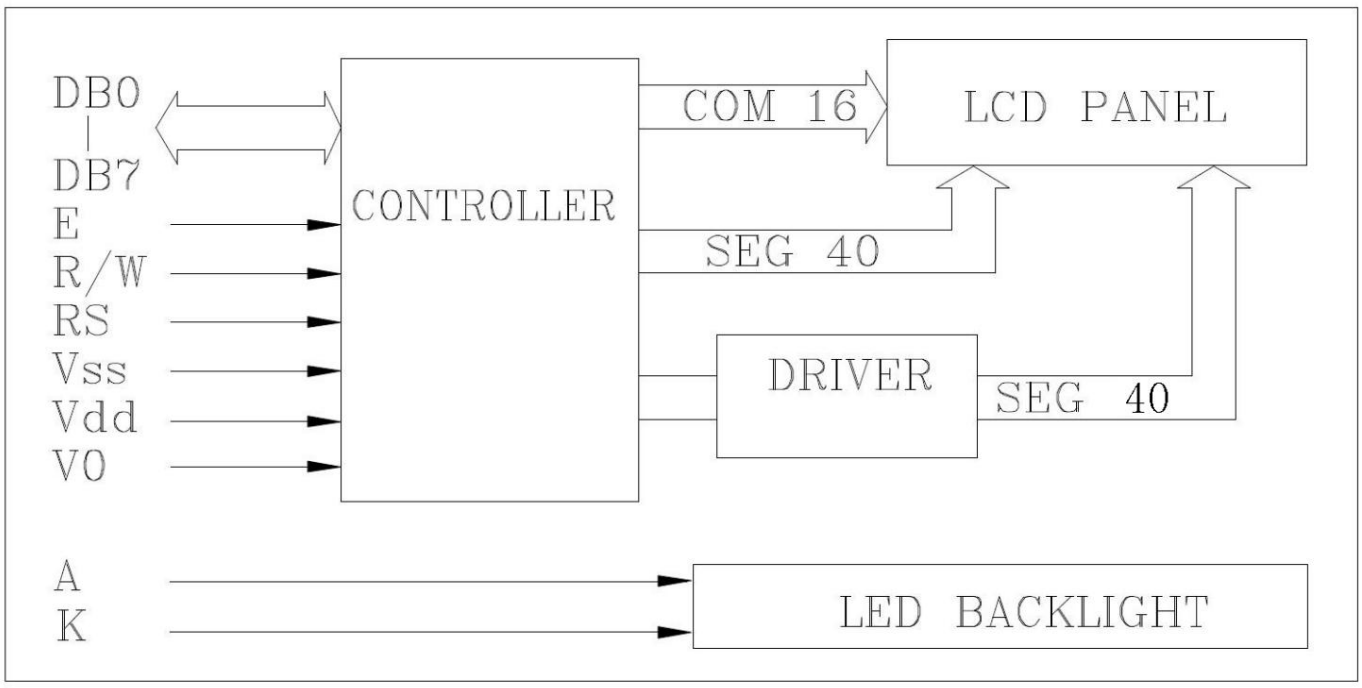
Weight: **g** Contrast: V0 external adjustment or internal fixed contrast

Working voltage: +3.3V or +5V, default is 5V

3. Dimensions:



4. Hardware block diagram:



5. Electrical characteristics

5.1 Limit parameters

Parameter name	symbol	condition	Typical Value		unit
			Minimum	Maximum	
Supply voltage	Vdd		-0.3	7.0	V
LCD drive voltage Input	V5		Vdd-10.0	Vdd+0.3	V
voltage	We		-0.3	Vdd+0.3	V
Operating temperature	Top		-20	70	ÿ
(T) Storage temperature (T)	Tstg		-30	80	ÿ

5.2.1 DC parameters 1 (Ta=25oC, Vdd=4.5V~5.5V)

Parameter name	symbol	condition	Nominal value			unit
			Minimum	Typical	Maximum	
Supply voltage	Vdd-GND		4.5	5.0	5.5 V	
Operating current (excluding backlight)	Idd	Vdd=5V	0.9	1.5	1.7 mA	
LCD drive current	Iee			0.6	- mA	
LCD drive voltage	Vdd-V5		4.2	4.5	4.8 V	
LED backlight operating current	If	Vf=3.0~3.2V	17	18	20 mA	
LED backlight power	Pd		90	100	110 mW	
consumption Input	HIV		2.5	- Vdd	V	
high level Input low	Will		-0.3		0.6 V	
level Output high	Voh	Ioh=-0.205mA	2.4			V
level Output low level	Vol	Io1=1.2mA			0.4 V	

5.2.2 DC parameter 2 (Ta=25oC, Vdd=2.7V~4.5V)

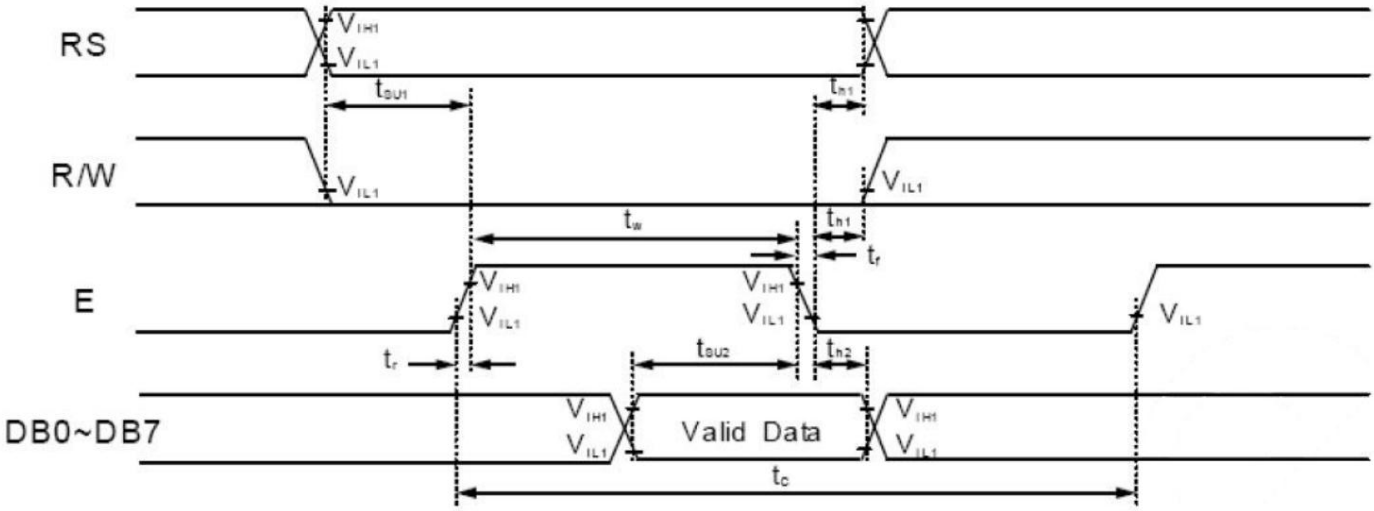
Parameter name	symbol	condition	Nominal value			unit
			Minimum	Typical	Maximum	
Supply voltage	Vdd-GND		2.7	3.3	4.5 V	
Operating current (excluding backlight)	Idd	Vdd=3.3V	0.45	0.9	1.0 mA	
LCD drive current	Iee			0.6	- mA	
LCD drive voltage	Vdd-V5		4.2	4.5	4.8 V	
LED backlight operating current	If	Vf=3.0~3.2V	17	18	20 mA	
LED backlight power	Pd		55	60	66 mW	
consumption Input	HIV		0.7Vdd	- Vdd	V	
high level Input low	Will		-0.3		0.55 V	
level Output high	Voh	Ioh=-0.1mA	0.75Vdd	-		V
level Output low level	Vol	Iol=0.1mA		- 0.2Vdd	V	

5.3.1 AC Parameters 1 (Ta=25oC, Vdd=4.5V~5.5V)

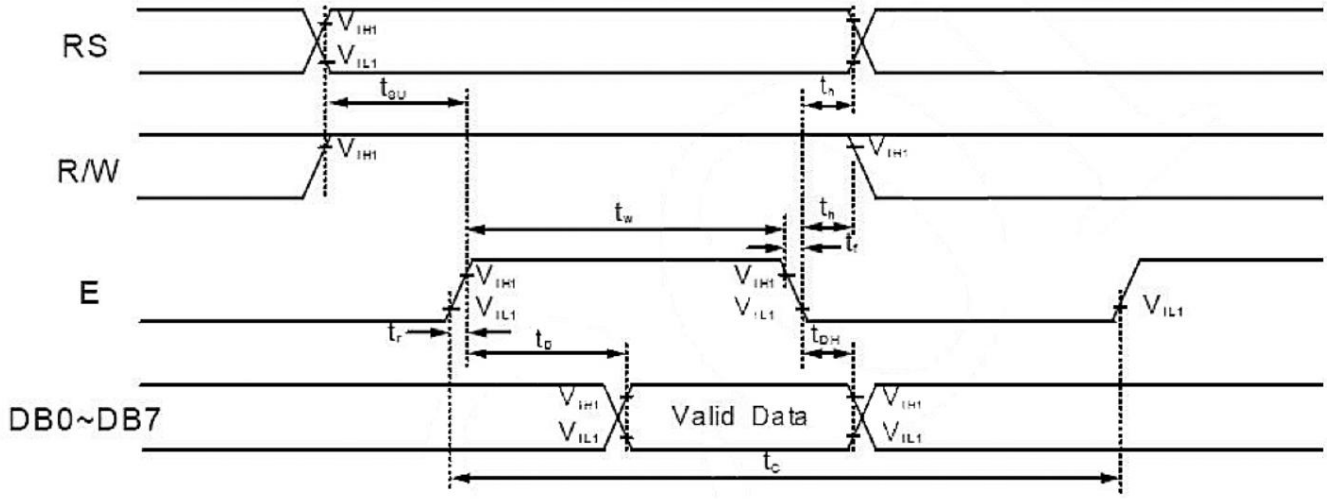
参 数名称	符号	测试条件	最小	典型	最大	单位
E周期	tc	写模式	500	-	-	ns
E上升/下降时间	t_R,t_F		-	-	20	
E脉冲宽度(1,0)	t_w		230	-	-	
R/W和RS建立时间	t_{su1}		40	-	-	
R/W和RS保持时间	t_{H1}		10	-	-	
数据建立时间	t_{su2}		80	-	-	
数据保持时间	t_{H2}		10	-	-	
E周期	t_c	读模式	500	-	-	ns
E上升/下降时间	t_R, t_F		-	-	20	
E脉冲宽度(1,0)	t_w		230	-	-	
R/W和RS建立时间	t_{su}		40	-	-	
R/W和RS保持时间	t_H		10	-	-	
数据输出延迟时间	t_D		-	-	120	
数据保持时间	t_{DH}		5	-	-	

5.3.2 AC parameters 2 (Ta=25oC, Vdd=2.7V~4.5V)

参 数名称	符号	测试条件	最小	典型	最大	单位
E周期	tc	写模式	1000	-	-	ns
E上升/下降时间	t_R,t_F		-	-	25	
E脉冲宽度(1,0)	t_w		450	-	-	
R/W和RS建立时间	t_{su1}		60	-	-	
R/W和RS保持时间	t_{H1}		20	-	-	
数据建立时间	t_{su2}		195	-	-	
数据保持时间	t_{H2}		10	-	-	
E周期	t_c	读模式	1000	-	-	ns
E上升/下降时间	t_R, t_F		-	-	25	
E脉冲宽度(1,0)	t_w		450	-	-	
R/W和RS建立时间	t_{su}		60	-	-	
R/W和RS保持时间	t_H		20	-	-	
数据输出延迟时间	t_D		-	-	360	
数据保持时间	t_{DH}		5	-	-	



Write Mode



Read Mode

6. Interface Description

Foot number	symbol	Function		Remark
1	Vss	power supply	0V	
2	Vdd		+5V	
3	Vo		LCD bias adjustment contrast adjustment	
4	RS	Data/command selection (H: data L: command)		
5	R/W	Read/write selection (H: read L: write)		
6	AND	Enable signal		
7	DB0	Data bit 0		
8	DB1	Data bit 1		
9	DB2	Data bit 2		
10	DB3	Data bit 3		
11	DB4	Data bit 4		
12	DB5	Data bit 5		
13	DB6	Data bit 6		
14	DB7	Data bit 7		
15	A	LED backlight positive		
16	K	LED backlight negative		

7. Instructions

The module has 4-bit/8-bit MCU parallel communication mode, and the 4-bit/8-bit bus is selected by the DL bit of the instruction register.

During read and write operations, two 8-bit registers are used, one is the data register DR and the other is the instruction register IR.

Register DR is used as a temporary storage for writing and reading DDRAM/CGRAM data. The target RAM is set by the RAM address instruction.

After selecting, any internal operation of reading and writing RAM is completed automatically. When the MCU reads the data in DR, DDRAM/CGRAM

Data is automatically transferred to DR. Similarly, when MCU writes data to DR, the data in DR is automatically transferred out of DDRAM/CGRAM.

The IR register is used to store the instruction code from the MCU. The MCU cannot read the instruction data. The RS pin can be used to switch the register.

device.

Various operations by setting the RS/RW bit:

RS	RW	operate
L	L	Write instruction operation (MCU writes instruction code to IR)
L	H	Read busy flag (DB7) and address counter (DB0~DB6)
H	L	Write data operation (MCU writes data to DR)
H	H	Read data operation (MCU reads data from DR)

7.1 Busy Flag (BF)

BF is high, indicating that the internal operation is in progress, so during this time, the next instruction will not be executed. When RS = "0"

When R/W=“1” (read instruction operation), the value of BF can be read from port DB7. When executing the next instruction, it is necessary to confirm that BF is not “

1”ÿ

7.2 Address Counter (AC)

When the DDRAM/CGRAM address from the instruction register is stored in the address counter, the data in the address counter is written

When RS = "0" and R/W = "1", the data in the address counter can be read from DDRAM/CGRAM and then incremented or decremented.

DB0~DB6 read out.

7.3 Display Data RAM (DDRAM)

The DDRAM address range of the module is 00H~27H and 40H~67H.

7.4 Character Generation ROM (CGROM)

The character generation ROM has 5x8 dot matrix, 192 characters, and 5x11 dot matrix, 64 characters mode

7.5 Character Generation RAM (CGRAM)

The character generation RAM has a 5*8 dot matrix and 8 character spaces. Each character space consists of 8 5-bit byte spaces.

Custom character data is stored in CGRAM. Users can call custom character data by writing DDRAM data 00H~07H to this 8 character space.

Meaning character.

DDRAM/CGRAM address correspondence diagram:

Character Code (DDRAM data)								CGRAM address						CGRAM Data								Pattern number
D7	D6	D5	D4	D3	D2	D1	D0	A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0	
0	0	0	0	×	0	0	0	0	0	0	0	0	0	×	×	×	0	1	1	1	0	pattern1
											0	0	1				0	0	1	0	0	
											0	1	0				0	0	1	0	0	
											0	1	1				0	0	1	0	0	
											1	0	0				0	0	1	0	0	
											1	0	1				0	0	1	0	0	
											1	1	0				0	1	1	1	0	
											1	1	1				0	0	0	0	0	
0	0	0	0	×	0	0	1	0	0	1	0	0	0	×	×	×	0	1	1	1	0	pattern2
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	0	0	0	
											0	1	1				1	0	0	0	0	
											1	0	0				1	0	0	0	0	
											1	0	1				1	0	0	0	1	
											1	1	0				0	1	1	1	0	
											1	1	1				0	0	0	0	0	
				*						*						*						
				*						*						*						
				*						*						*						
				*						*						*						
0	0	0	0	×	1	1	1	1	1	1	0	0	0	×	×	×	1	1	1	1	1	pattern8
											0	0	1				1	0	0	0	0	
											0	1	0				1	0	0	0	0	
											0	1	1				1	1	1	1	0	
											1	0	0				1	0	0	0	0	
											1	0	1				1	0	0	0	1	
											1	1	0				1	1	1	1	0	
											1	1	1				0	0	0	0	0	

7.6 Command Description

Instruction List

Command RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 Execution time clear												describe
show	0	0	0	0	0	0	0	0	0	1	1.53ms	Write 20H to DDRAM and set the address count The device address is set to 00H
Return 0		0	0	0	0	0	0	0	1		1.53ms	Set the address counter to 00H, and To restore the cursor to its original position, DDRAM contents remain unchanged
Input module Mode setting	0	0	0	0	0	0	0	1	I/D SH		39us	Set the cursor movement direction and allow the entire Display Mobile
display switch	0	0	0	0	0	0	1 D C B				39us	Set the display, cursor, and cursor blinking control Positioning

Shift 0		0	0	0	0	1	S/C R/L				39us	Set the cursor movement and display movement direction Control bit, DDRAM data remains unchanged
Function set up	0	0	0	0	1 DL N F						39us	Set the interface data width (DL: 8 bits/4 bits), display number of lines (N: 2 lines/1 line), display Font (F: 5x11 dots/5x8 dots)
set up CGRAM address	0	0	0	1 AC5 AC4 AC3 AC2 AC1 AC0							39us	Set CGRAM address in address counter
set up DDRA M-Address	0	0	1 AC6 AC5 AC4 AC3 AC2 AC1 AC0								39us	Set the DDRAM ground in the address counter site
Read Busy Logo & address	0	1	BF AC6 AC5 AC4 AC3 AC2 AC1 AC0								0us	Read the busy flag BF, the address counter The contents are read out at the same time
Write data 1		0	D7 D6 D5 D4 D3 D2 D1 D0								43us	Write data to internal RAM ÿDDRAM/CGRAMÿ
Read data 1		1	D7 D6 D5 D4 D3 D2 D1 D0								43us	From internal RAM (DDRAM/CGRAM) Read data from

Note: “-” means not considered

7.6.1 Clear Display

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

0 0 By writing 20H (space code) to all DDRAM addresses and setting the address counter to 00H, the display data can be cleared.

Place the cursor at the initial state position and set the input mode to increment (I/D is high).

7.6.2 Return

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

“-” means not considered

The return instruction returns the cursor to the starting position, sets the DDRAM address to 00H, writes the address counter, and changes the display to the initial

The data in DDRAM remains unchanged.

7.6.3 Input Mode

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	Set the cursor and		0	0	0	1	I/D SH

display movement direction

I/D: Increment or decrement of DDRAM address (cursor or blinking)

When I/D is 1, the cursor flashes and moves to the right, and the DDRAM address is incremented; when I/D is 0, the cursor flashes and moves to the left, and the DDRAM

The address is decremented, and when reading or writing to CGRAM, the movement is consistent with DDRAM.

SH: Display Shift

When SH is 0, the entire display will not shift when reading or writing to DDRAM or CGRAM.

During a write operation, the entire display will be shifted in the direction set by I/D.

7.6.4 Display switch control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	Display/		0	0	0	0	1 D C B	

cursor and blink control

D: Display switch control bit

When D is 1, the display is turned on; when D is 0, the display is turned off, but the display data in DDRAM remains unchanged.

C: Cursor switch control bit

When C is 1, the cursor is on; when C is 0, the cursor disappears, but the I/D register saves its data.

B: Cursor flashing switch control bit

When B is 1, the cursor flashes on, and when B is 0, the cursor flashes off.

7.6.5 Cursor Blinking/Shifting

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L		

Do not read or write display data, but move the cursor position or display left or right. This instruction is used to correct and find display data.

In display mode, after the 40th character in line 1, the cursor moves to line 2. Note that display shifting is done simultaneously in all lines.

When the display data is repeatedly shifted, each bit is shifted independently. When the display is shifted, the content in the address counter remains unchanged.

Shift Format Table

S/C	R/L	operate
0	0	The cursor moves to the left and the address counter decreases by 1
0	1	The cursor moves to the right and the address counter increases by 1
1	0	So the display moves to the left, and the cursor follows the display shift
1	1	All displays move to the right, and the cursor moves with the display

7.6.6 Function settings

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F		

DL: Interface data width control bit

When DL is 1, it indicates that the 8-bit bus is connected to the MCU

When DL is 0, it means that the 4-bit bus is connected to the MCU. When it is in 4-bit bus mode, 8-bit data needs to be transmitted by 4-bit data 2 times completed.

N: Display line number control bit

When N is 0, it is 1-line display mode; when N is 1, it is 2-line display mode.

F: Display font settings

When F is 0, the font is in 5x8 dot matrix mode; when F is 1, the font is in 5x11 dot matrix mode.

7.6.7 Setting CGRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

The CGRAM address is placed into the address counter, making the CGRAM data from the MCU valid.

7.6.8 Setting DDRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

The DDRAM address is placed into the address counter to make the DDRAM data from the MCU valid. When in 1-line display mode, DDRAM address 00H~4FH; when in 2-line display mode, the DDRAM address of the first line is from 00H~27H, and the DDRAM address of the second line is From 40H~67H.

7.6.9 Reading the busy flag and address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether the module is in internal operation. If BF is 1, the internal operation is in progress and it is necessary to wait until BF is set to 0 before the next instruction can be executed. In this instruction, the value in the address counter can also be read.

7.6.10 Writing data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write 8-bit data to DDRAM/CGRAM. The selection between DDRAM and CGRAM is determined by the address setting instruction (DDRAM address setting instruction and CGRAM address setting instruction). After the write operation, the address will be incremented or decremented according to the address counter set by the RAM setting instruction.

7.6.11 Reading Data from RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

Read 8-bit data from DDRAM/CGRAM. The selection between DDRAM and CGRAM is determined by the address setting instruction. If the address setting instruction is not written before reading the data, the read data is invalid. If the RAM address setting instruction is not written before the read operation, and the data is read multiple times, the second data is valid and the first data is incorrect because there is no timing to match the RAM data output. In the DDRAM read operation, the cursor transfer instruction plays the same role as the DDRAM address setting instruction, and also sends the RAM data to the output register. After the read operation, the address counter increases or decreases according to the input mode instruction. After the CGRAM read operation, the display shift may not be executed correctly.

7.6.12 Initialization

When powered on, the module will be initialized. During this process, the following instructions will be executed. The busy flag will remain high

before the initialization is completed. 1. Clear display instruction:

All DDRAMs are written to 20H 2. Set function instruction:

DL=1, 8-bit bus mode N=0, 1-

line display mode F=0, 5x8 font

3. Display switch command: D=0, display off

C=0, cursor off

B=0, flash off

4. Set the return command: I/D=1, increment

SH=0, display does not shift

7.7 Display position and RAM address correspondence table

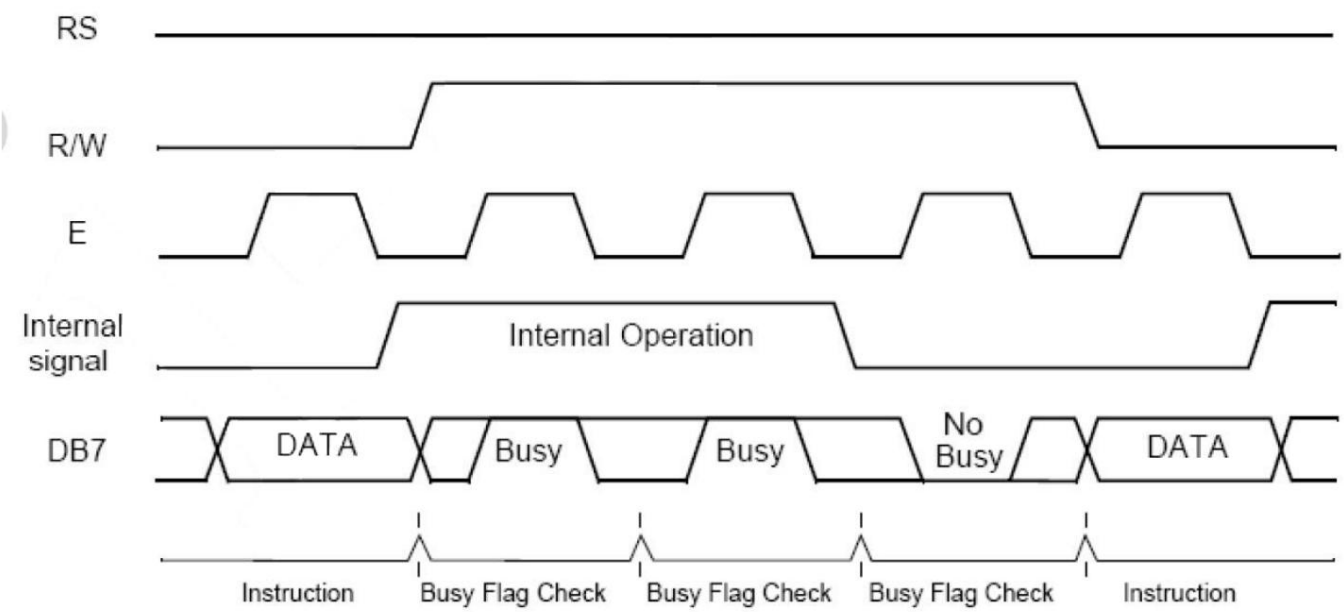
Display Location	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9	1-10
DDRAM Address	00	01	02	03	04	05	06	07	08	09
Display Location	1-11	1-12	1-13	1-14	1-15	1-16	1-17	1-18	1-19	1-20
DDRAM Address	0A	0B	0C	0D	0E	0F	10			
Display Location	1-21	1-22	1-23	1-24	1-25	1-26	1-27	1-28	1-29	1-30
DDRAM Address	14	15	16	17		18	19	1A	1B	1C 1D
Display Location	1-31	1-32	1-33	1-34	1-35	1-36	1-37	1-38	1-39	1-40
DDRAM Address	1E	1F	20	21	22	23	24	25	26	27
Display Location	2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9	2-10
DDRAM Address	40	41	42	43	44	45	46	47	48	49
Display Location	2-11	2-12	2-13	2-14	2-15	2-16	2-17	2-18	2-19	2-20
DDRAM Address	4A	4B	4C	4D	4E	4F	50	51	52	53
Display Location	2-21	2-22	2-23	2-24	2-25	2-26	2-27	2-28	2-29	2-30
DDRAM Address	54	55	56	57	58	59	5A	5B	5C	5D
Display Location	2-31	2-32	2-33	2-34	2-35	2-36	2-37	2-38	2-39	2-40
DDRAM Address	5E	5F	60	61	62	63	64	65	66	67

1-1 means the first character of the first line

8. Operation sequence

8.1 8-bit MCU Interface

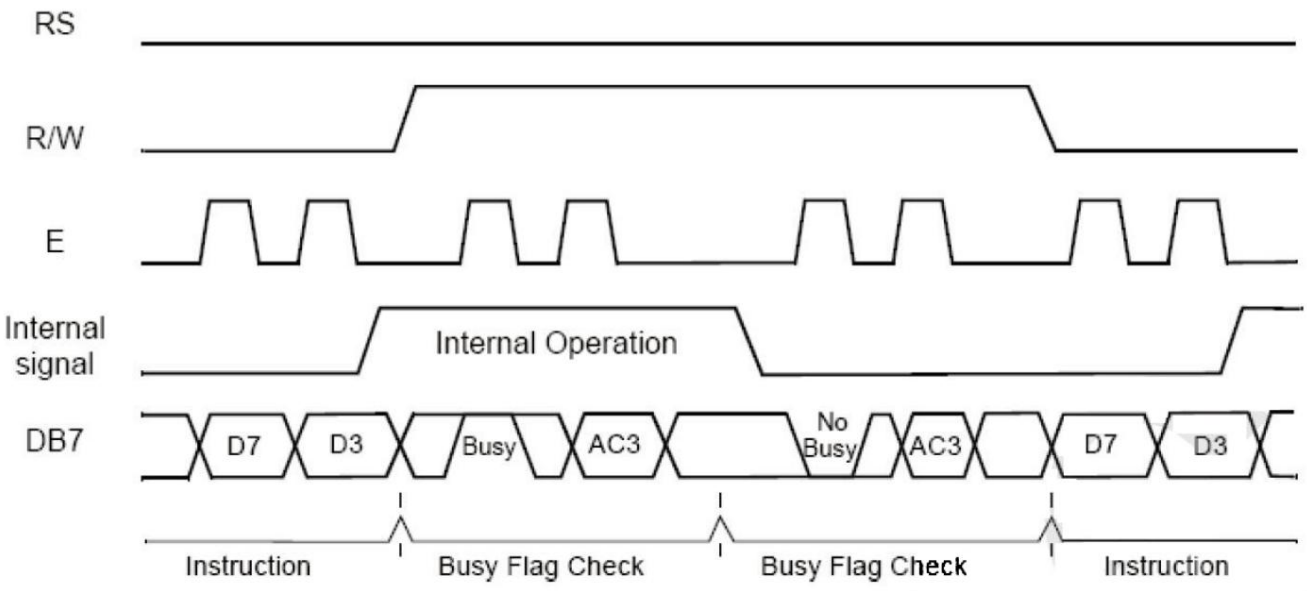
When the interface data width is set to 8 bits, data is read and written simultaneously from the 8-bit port (DB0~DB7), and the timing is shown in the following figure:



8.2 4-bit MCU interface

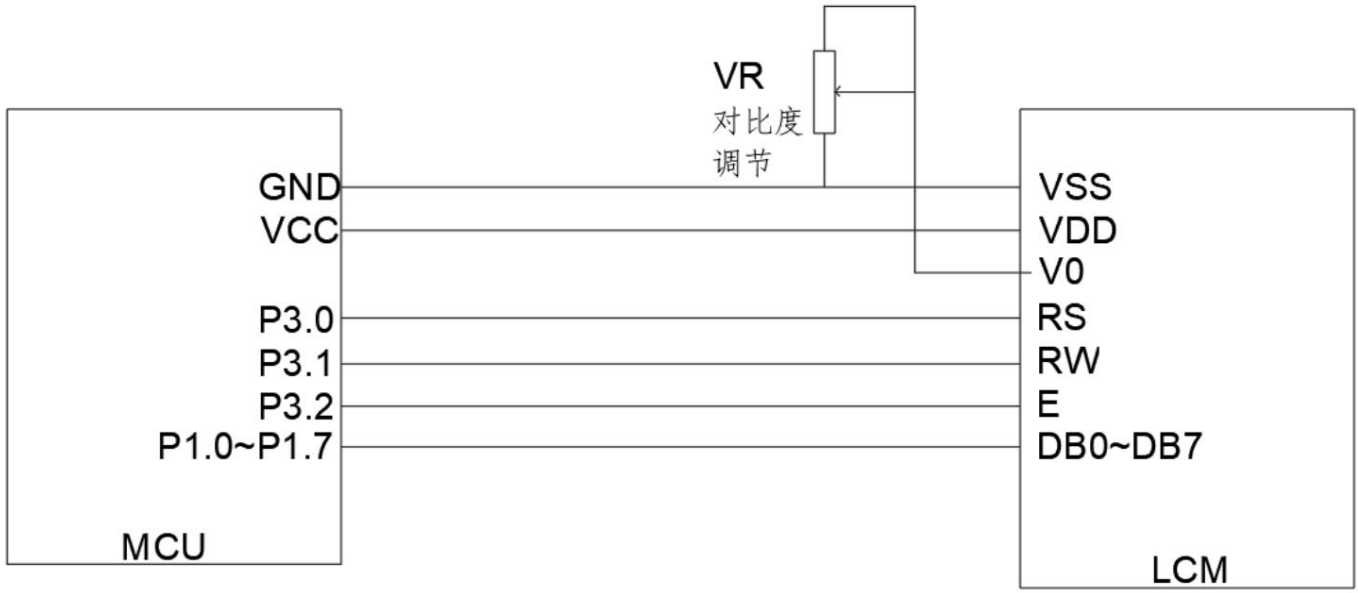
When the interface data width is set to 4 bits, data is read and written only from the upper 4 bits (DB4~DB7) of the 8-bit port, and the upper 4 bits are transmitted first.

The lower 4 bits are transmitted one by one, and the timing is shown in the following figure:



Application Examples

9.1 8-bit wiring diagram



8位接线方法

9.2 8-bit C51 routines

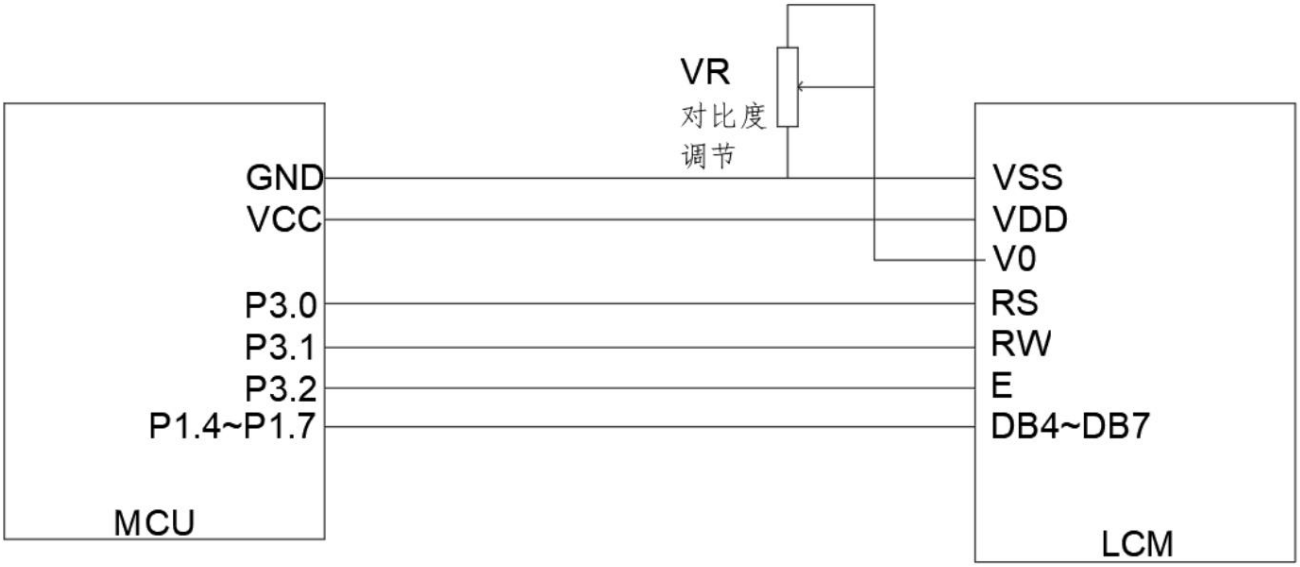
```
#include <STC15.H>
#include <string.h>
#include <INTRINS.H>
#define uchar unsigned char
#define uint unsigned int
#define DB07 P1
sbit RS=P3^0;
sbit RW=P3^1;
sbit E=P3^2;

 ////////////////////////////////////////////DB area
fly code border_inf[16] =
  { 0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,
  0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa, };

uchar code
 string[]={ 0xa0,0xa1,0xa2,0xb0,0xb1,0xb2,0xc0,0xc1,0xc2,0xd0,0xd1,0xd2,0xe0,0xe1,0xe2,0xf0,
 0xf1,0xf2,0xa3,0xa4,0xa5,0xa6,0xb3,0xb4,0xb5,0xb6,0xc3,0xc4,0xc5,0xc6,0xd3,0xd4,
 0xd5,0xd6,0xe3,0xe4,0xe5,0xf3,0xf4,0xf5 };
```

```
void delay(unsigned int m) //delay program

        { unsigned int i,j;
          for(i=0;i<m;i++)
            for(j=0;j<20;j++);
        }

void delayms(unsigned int n)          //Delay 10×n milliseconds program

        { unsigned int i,j;
          for(i=0;i<n;i++)

            { for(j=0;j<800;j++);
              }
        }

void LcdWriteCom(uchar com) {

    E=0;
    RW=0;
    RS=0;
    DB07=com;
    E=1;
  delay(10);
    E=0;

}

void LcdWriteData(uchar dat) {

    E=0;
    RW=0;
    RS=1;
    DB07=that;
    E=1;
    delay(10);
    E=0;

}
```

```
void LcdInit() {

    LcdWriteCom(0x38);
    LcdWriteCom(0x0c);
    LcdWriteCom(0x06);
    LcdWriteCom(0x01);
    delayms(5);
}

void Show_string1(uchar a[]) {

    uchar i;
    LcdWriteCom(0x80);
    for(i=0;a[i]!='\0';i++)

    { LcdWriteData(a[i]); }

}


void Show_string2(uchar *a) {

    flying i;
    LcdWriteCom(0x80+0X40);
    for(i=0;i<16;i++) {

       LcdWriteData(a[i]); }

}

/********************************************************************** * Name: Main() * Function: Main function
* Input: None *
Output: None


**********************************************************************
/
void main(void)
```

```
{
        flying i,j;
        LcdInit();
  LcdWriteCom(0x40);
    for(i=0;i<16;i++) {

     LcdWriteData(border_inf[i]); }

        while(1)
      {
            for(j=0;j<2;j++)
    {
  LcdWriteCom(0x80);
  for(i=0;i<16;i++) {

    LcdWriteData(j); }

LcdWriteCom(0xC0);
  for(i=0;i<16;i++) {

    LcdWriteData(j); }

  delayms(500); }

 Show_string1("12345abcefg!@#$%");
  Show_string2(string);
delayms(600);

  }}
```

9.3 4-bit Wiring Diagram



4位接线方法

9.4 4-bit C51 routines

```
#include <STC15.H>
#include <string.h>
#include <INTRINS.H>
#define uchar unsigned char
#define uint unsigned int
#define DB07 P1
sbit RS=P3^0;
sbit RW=P3^1;
sbit E=P3^2;

fly code border_inf[16] =
  { 0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,
  0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa, };


uchar code
 string[]={ 0xa0,0xa1,0xa2,0xb0,0xb1,0xb2,0xc0,0xc1,0xc2,0xd0,0xd1,0xd2,0xe0,0xe1,0xe2,0xf0,
 0xf1,0xf2,0xa3,0xa4,0xa5,0xa6,0xb3,0xb4,0xb5,0xb6,0xc3,0xc4,0xc5,0xc6,0xd3,0xd4,
 0xd5,0xd6,0xe3,0xe4,0xe5,0xf3,0xf4,0xf5 };


void delay(unsigned int m) //delay program
```

```
{ unsigned int i,j;
  for(i=0;i<m;i++)
    for(j=0;j<20;j++);
}

void delayms(unsigned int n)                    //Delay 10×n milliseconds program

{ unsigned int i,j;
  for(i=0;i<n;i++)

    { for(j=0;j<800;j++); }

}

void LcdWriteCom(uchar com) {

    uchar com1,com2;
    com1=com&0xf0;
    com2=(com<<4)&0xf0;
    E=0;
    RW=0;
    RS=0;
    DB07=com1;
    E=1;
  delay(10);
    E=0;
    RW=0;
    RS=0;
    DB07=com2;
    E=1;
  delay(10);
    E=0;
}

void LcdWriteData(uchar dat) {

    fly dat1,dat2;
    dat1=dat&0xf0;
```

```
    dat2=(dat<<4)&0xf0;
    E=0;
    RW=0;
    RS=1;
    DB07=dat1;
    E=1;
  delay(10);
    E=0;
    RW=0;
    RS=1;
    DB07=dat2;
    E=1;
  delay(10);
    E=0;
}

void LcdInit() {

    LcdWriteCom(0x28);
    LcdWriteCom(0x0c);
    LcdWriteCom(0x06);
    LcdWriteCom(0x01);
    delayms(5);
}

void Show_string1(uchar a[]) {

    uchar i;
    LcdWriteCom(0x80);
    for(i=0;a[i]!='\0';i++)

    { LcdWriteData(a[i]); }

}

void Show_string2(uchar *a) {

    flying i;
    LcdWriteCom(0x80+0X40);
    for(i=0;i<16;i++)
```

```
{
    LcdWriteData(a[i]); }

}

/*********************************************************************
* Name: Main() *
Function: Main
function *
Input: None * Output: None
*********************************************************************
/
void main(void) {

            flying i,j;
            LcdInit();
    LcdWriteCom(0x40);
        for(i=0;i<16;i++) {

          LcdWriteData(border_inf[i]); }

            while(1)
        {
                    for(j=0;j<2;j++)
      {
    LcdWriteCom(0x80);
    for(i=0;i<16;i++) {

      LcdWriteData(j); }

LcdWriteCom(0xC0);
    for(i=0;i<16;i++) {

      LcdWriteData(j); }

    delayms(500); }

  Show_string1("12345abcefg!@#$%");
    Show_string2(string);
```

```
delayms(600);

}}
```

9.5 Fonts

1. Liquid Crystal Display (LCD)

Liquid Crystal Display is made of glass, organic sealant, organic fluid, and polymer-based polarizer. The following precautions should be taken when handling:

(1) Keep the temperature within the use and storage range. Excessive temperature and humidity can cause polarization degradation, polarizer peeling, or bubbles. (2) Do not touch the exposed polarizer with anything harder than HB pencil lead. To remove dust from the display surface, gently wipe it with cotton. Wipe with a deerskin towel or other soft material soaked in cleaning oil.

(3) Wipe off saliva or water drops immediately. If ITO is in contact with water for too long, it will cause the surface of the LCD to deform or discolor.

(4) Glass can easily break due to rough handling. Especially at corners and edges. (5) Do not drive the LCD with DC voltage.

2. Liquid Crystal Display

Module 2.1 Mechanical

Precautions The assembly and adjustment of the LCM are performed with high precision. Avoid excessive vibration and do not make any changes or modifications. The following points should be noted. (1) Do not change the lugs on the metal frame in any way. (2) Do not modify the PCB by drilling additional holes, changing its contour, moving its components, or modifying its pattern. (3) Do not touch the elastomer connectors, especially those inserted into the backlight panel (e.g., EL). (4) When installing the LCM, make sure that the PCB board is not subjected to any stress such as bending or twisting. The elastomer contacts are very delicate and the slightest misalignment of any element may cause a missing pixel. (5) Avoid pressing on the metal baffle, otherwise the elastomer connector may deform and lose contact, resulting in a missing pixel. 2.2. Static Electricity The LCM contains a CMOS LSI and the same precautions should be taken for such devices, namely (1) When the operator comes into contact with the module, it should be grounded. Never touch any conductive parts such as LSI pads, copper wires on the PCB, and interface terminals with any part of the human body. (2) The module should be stored in an anti-static bag or other anti-static container. (3) Only a properly grounded soldering iron should be used. (4) If an electric screwdriver is used, it should be well grounded and prevent commutator sparks. (5) Normal anti-static measures should be followed for work clothes and workbenches; for the latter, a conductive (rubber) mat is recommended. (6) Since dry air can induce static electricity, a relative humidity of 50-60% is recommended. 2.3. Soldering (1) Solder only to I/O terminals. (2) Only a properly grounded and leak-free soldering iron should be used.

(3) Soldering temperature: 280ÿ±10ÿ (4) Soldering time: 3 to 4 seconds. (5) Use low-temperature solder filled with resin flux. (6) If flux is used, cover the LCD surface to avoid flux splashing. Flux residue should be removed after protection. 2.4. Operation (1) The viewing angle can be adjusted by changing the LCD drive voltage V0. (2) The drive voltage should be kept within the specified range. Excessive voltage will shorten the life of the display. (3) The response time increases Increases with decreasing temperature.

(4) At temperatures above its operating range, the display may turn black or dark blue; however, do not press the display area. domain) may cause some displayed segments to "break".

(5) Mechanical damage during operation (such as pressing the display area) may cause the line segments to "break". 2.5. Storage If liquid leaks from a damaged glass battery, rinse any exposed body parts with soap and water. Do not swallow the liquid. The toxicity is extremely low, but caution should be exercised at all times.

Warranty Unless

otherwise agreed with the customer, within one year from the date of shipment, any LCD and IC will be repaired or serviced when electrical and appearance defects are found when inspected according to the acceptance criteria. The confirmation of this date shall be based on the shipping documents. The warranty liability is limited to repairs and/or replacements made in accordance with the above terms. We will not be responsible for any subsequent or consequential events.