Product Specification

Number: L-KLS9-1602E-YG-K/A

Name: Character LCD Module

Customer: _____

Date: 2025-03-26

Customer Signature:

KLS electronic

WWW.KLSELE.COM

NINGBO KLS ELECTRONIC CO; LTD

Tel : 0086-574-86828566

Fax : 0086-574-86824882

ADD : NO. 8-1, RONGXIA RD. XIAPU SHANQIAN INDUSTRIAL ZONE BEILUN NINGBO ZHEJIANG.

Compi	Check	Review	Approva
Jenny	Jack.C		

catalogue

1 Revised records

2. summary

3. outline dimension

4. Hardware box diagram

5. electrical character

6. interface specification

7 Instructions

8. Description of the operation timing

9. Application routines

10. matters need attention

1. Revised records

Edition	Date of issue	New system / revised content
V1.0	2020-8-12	Brand-new

2. summary

character LCD display module is a point formation LCD display module specialized for displaying letters, digital elements, symbols, etc. In a 4-bit and 8-bit data transmission mode. Provide 57 dot + cursor display mode. Provide display data buffer DDRAM, character generator CGROM and character, generator CGRAM, you can use CGRAM to store your own defined character model data of up to 858, dot matrix of graphical characters. Provide rich instruction settings: clear display; cursor back to the origin; display on / off; cursor on / off; display character flashing; cursor shift; display shift element, etc. Provide an internal power-on automatic reset circuit, automatically initialize the module, and set the module to the default display working state.

Number of characters displayed: 16 characters in X 2 lines

Character dot array: 5X7 word array + cursor

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

Polarized film: fully permeable / semi-permeable

Observation Angle: 6:00

Show duty cycle: 1 / 16 drive bias: 1 / 5

Control chip: SPLC780D or compatible IC (e. g. 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)
Overall dimensions: 84X44X11.7

Visual area: 64.5X16

Character font: 5X7 dots + light-mark size:

Character size: 55.45X11

0.54X0.6

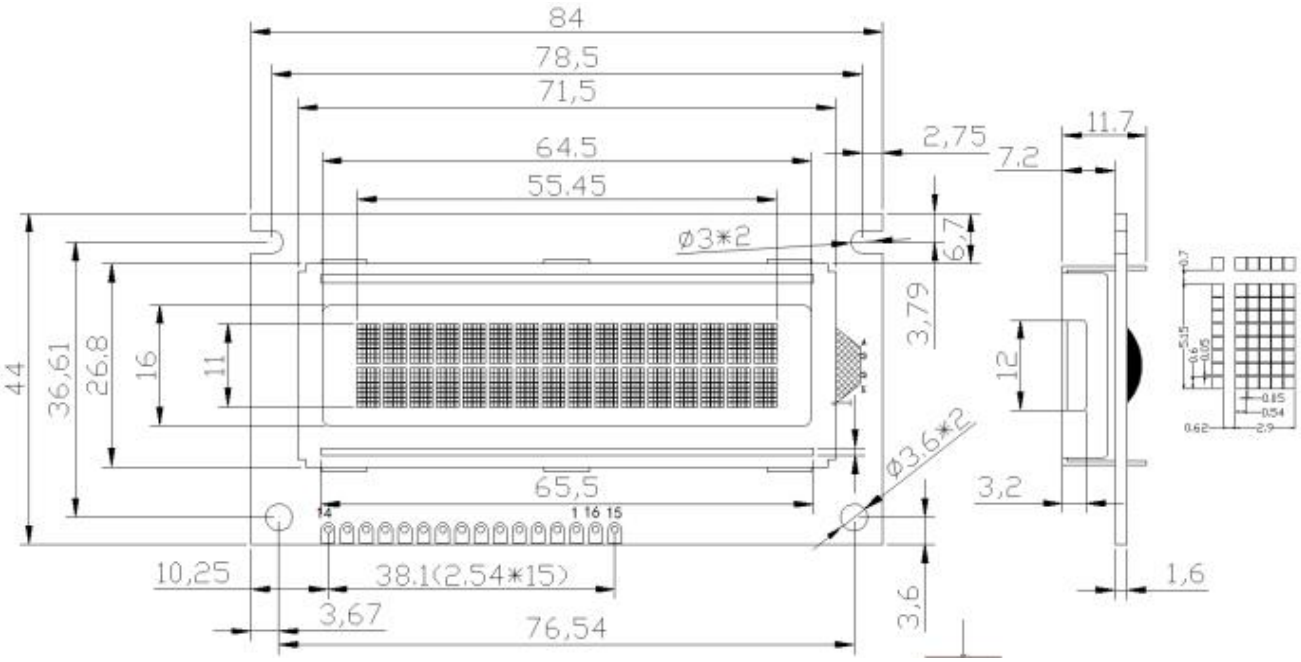
Character Spacing: 3.52X5.85

Contrast: V0 external adjustment or internal fixed contrast

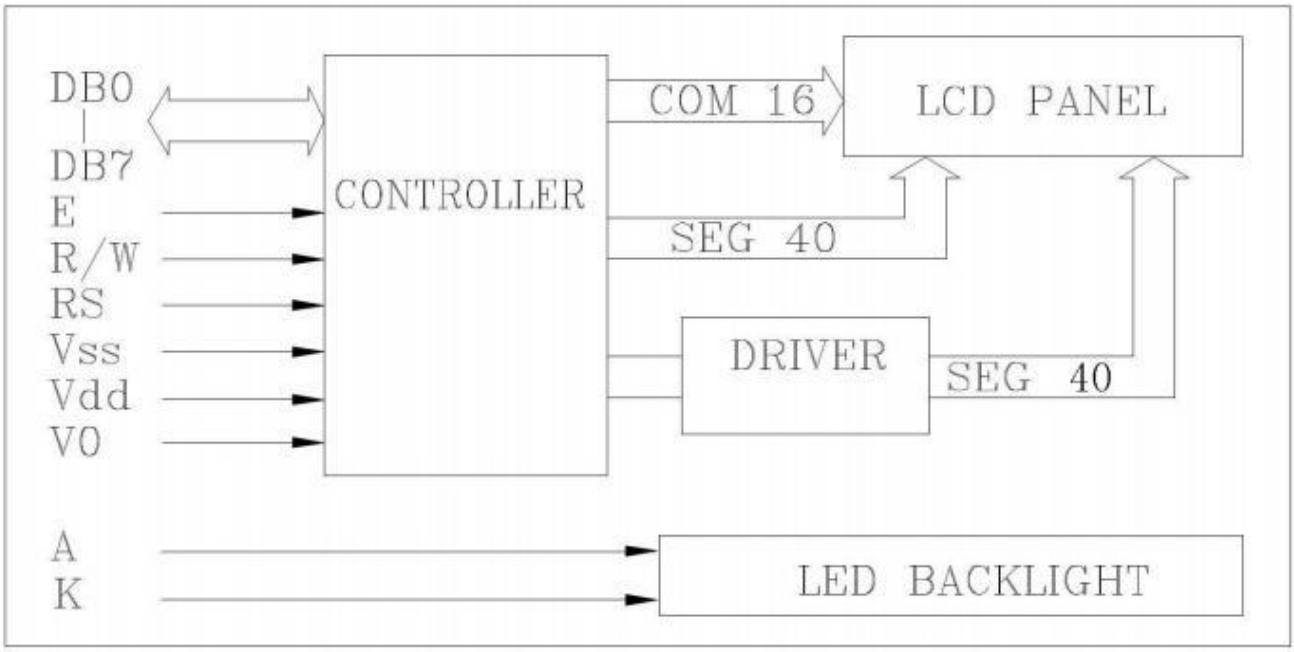
weight: e

Operating voltage: + 3.3V or + 5 default 5V

3. outline dimension:



4. Hardware block diagram:



5. electrical character

5.1 absolute rating

The parameter name	Symbol	Condition	Representative value		Unit
			Least value	Crest value	
Supply voltage	Vdd		-0.3	7.0	V
LCD driving voltage	V5		Vdd-10.0	Vdd+0.3	V
Input voltage	Vi		-0.3	Vdd+0.3	V
Working temperature (T)	Top	-	-20	70	℃
Storage temperature (T)	Tstg	-	-30	80	℃

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

The parameter name	Symbol	Condition	Nominal value			Unit
			Minimum	Typical case	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 driving 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 consumption	Pd		90	100	110	mW
Enter a high level	Vih		2.5	-	Vdd	V
Enter a low level	Vil		-0.3	-	0.6	V
Output high level	Voh	Ioh=-0.205mA	2.4	-	-	V
Output low level	olV	Io1=1.2mA	-	-	0.4	V

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

The parameter name	Symbol	Condition	Nominal value			Unit
			Minimum	Typical case	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 driving 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 consumption	Pd		55	60	66	mW

Enter a high level	Vih		0.7Vdd	-	Vdd	V
Enter a low level	Vil		-0.3	-	0.55	V
Output high level	Voh	Ioh=-0.1mA	0.75Vdd	-	-	V
Output low level	olV	Iol=0.1mA	-	-	0.2Vdd	V

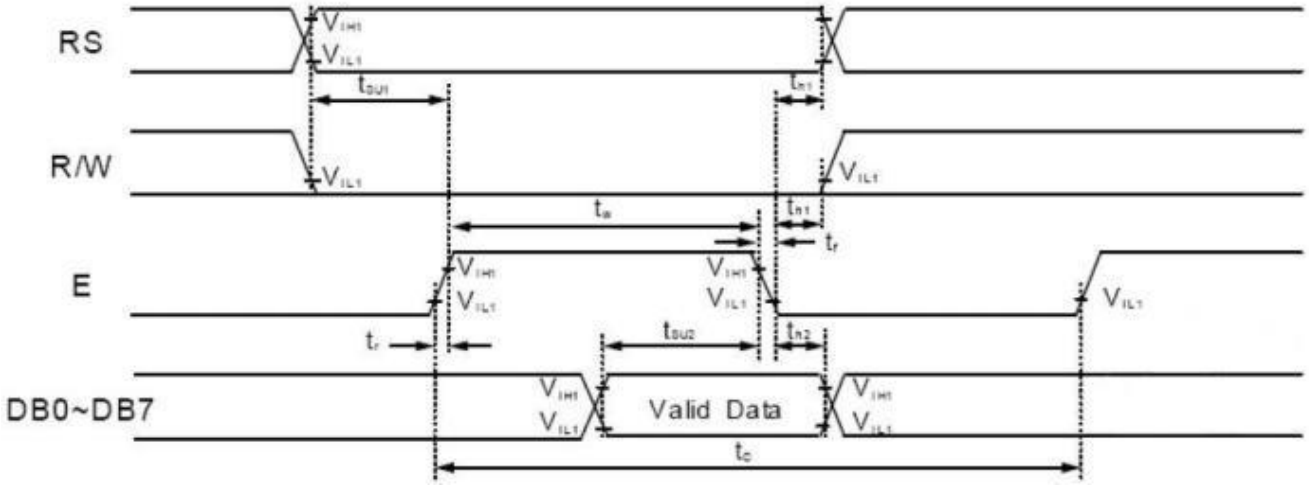
5.3.1 AC parameter 1(Ta=25oC,Vdd=4.5V~5.5V)

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	-	-	

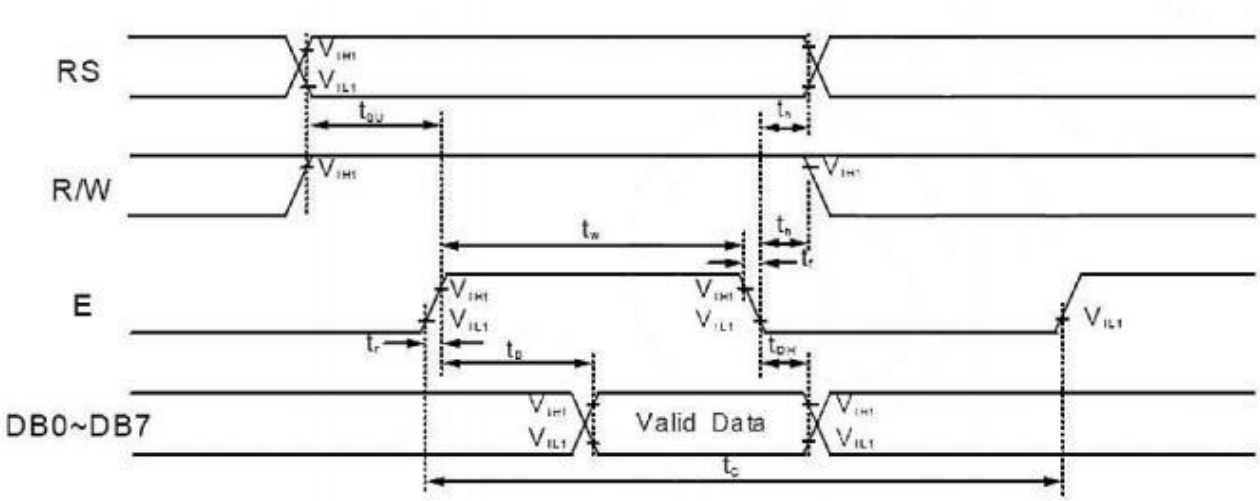
5.3.2 AC parameter 2(Ta=25oC,Vdd=2.7V~4.5V)

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	-	-	

The AC test waveform map



Write the pattern



Reading mode

6. interface specification

Subscript	Symbol	Function		Remarks
1	Vss	Power suppl	0V	
2	Vdd		+5V	
3	Vo		LCD bias regulation	Contrast co-ntrol
4	RS	Data / instruction selection (H: Data L: instruction)		
5	R/W	Read / Write Selection (H: Read L: wri - te)		
6	E	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	K	LED, backlight negative		
16	A	LED, backlight positive		

7. Instructions

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

In read and write operations, two 8-bit registers are used, one is data register DR and the other is instruction register IR. The data register DR is stored as a temporary place for writing and reading DDRAM / CGRAM data, the target RAM is selected through the RAM address setting instruction, and any internal operation of reading and writing the RAM is automatically completed. When the MCU reads the data in the DR, the DDRAM / CGRAM data is automatically transferred to the DR. Similarly, when the MCU writes the data to the DR, the data in the DR is automatically transmitted to DDRAM / CGRAM. Instruction register IR is used to store the instruction code from MCU. MCU cannot read the instruction data and can switch the register through the RS pin.

Various actions by setting the RS / RW bit:

RS	RW	Operate
L	L	Write instruction operation (MCU write instruction code to IR)
L	H	Read the busy flag (DB7) and the address counter (DB 0 to DB 6)
H	L	Write data operation (MCU write data to DR)
H	H	Read data operation (MCU reads out data from the DR)

7.1 Busy Sign (BF)

Bishigh,indicatingthattheinternaloperationisongoing,soatthistime thenextinstructionwillnotbeexecuted.WhenRS="0"andR/W="1"(read instructionoperation), thevalueofBFcanbereadfromtheDB7port,andthenext instructionmustbeconfirmedthattheBFisnot"1".

7.2 Address counter (AC)

WhentheDDRAM/CGRAMaddressfromtheinstructionregisterisstoredinthe address counter, the data in the address counter increases or decreases after writing or reading out DDRAM / CGRAM, and when RS= "0" and R / W= "1", the data in the address counter can be read out from DB0 to DB6.

7.3 Display Data RAM (DDRAM)

Instructions for use of the LCD display module

The DDRAM address of the module ranges from 00 H to 27 H and 40 H to 67 H.

7.4 Character Generation ROM (CGROM)

The character generation ROM has 5X8 dot matrix, 192 characters, and 5x11 dot matrix, 64 character patterns

7.5 Character Generation RAM (CGRAM)

Character generation RAM has a 5 * 8 dot matrix, 8 character space, each consisting of 85-bit bytes, by writing custom character data to CGRAM, this 8 character space users can write to DDRAM data 00 H to 07 H to call custom characters.

DDRAM / CGRAM address map:

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 Instructions described

code repertory

Instruct	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	E-time	Description
Clean up Show	0	0	0	0	0	0	0	0	0	1	1.53ms	Write 20H to DDRAM and set the address counter address to 00H
Return	0	0	0	0	0	0	0	0	1	-	1.53ms	Set the address counter address to 00H and restore the cursor to the initial position, leaving the DDRAM content unchanged
Input mode Type sett-	0	0	0	0	0	0	0	1	I/D	SH	39us	Set the cursor movement direction, and allow the

ing												entire display to move
Show open / close	0	0	0	0	0	0	1	D	C	B	39us	Set the display, the cursor, and the flicker control bit of the cursor
Gression	0	0	0	0	0	1	S/C	R/L	-	-	39us	Set the cursor movement and display the movement direction

Instructions for use

												Control bit, the DDRAM data remained unchanged
Function Set up	0	0	0	0	1	DL	N	F	-	-	39us	Set the interface data width (DL: 8 bit / 4 bit), Displays the number of rows (N: 2 rows / 1 row), and displays the rows Font (F: 5x11 dot matrix / 5x8 dot matrix)
Set up the CGRAM address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	39us	Set the CGRAM address within the address counter
Set up the DDRAM address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	39us	Set the DDRAM address in the address counter
Read the busy flag & address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	0 us	Read the busy flag bit BF, and the content in the address counter is read out simultaneously
WD	1	0	D7	D6	D5	D4	D3	D2	D1	D0	43us	Write the data to the internal RAM (DDRAM/CGRAM)
Read data	1	1	D7	D6	D5	D4	D3	D2	D1	D0	43us	Data were read from the internal RAM (DDRAM / CGRAM)

Note: "-" is 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

By writing 20H (space code) to all DDRAM addresses and setting the address counter to 00H, you can clear the display data, place the cursor in 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	-

"-" is not an exam consider

The return instruction is to return the cursor back to the starting position, set the DDRAM address to 00H and write to the address counter, and change the display to the initial state, leaving the data in DDRAM unchanged.

7.6 3 input pattern

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

Sets the movement direction of the cursor and the display

I / D: Increase or decrease of the DDRAM address (cursor or flicker)

When I / D is 1, the cursor flashes to the right and the DDRAM address is increasing; when I / D is 0, the cursor flashes to the left and the DDRAM address decreases when read out or written to CGRAM.

SH: Display the shift

When SH is 0, the entire display does not shift for the DDRAM or CGRAM read / write operations. When SH is 1, the operation to DDRAM write is that the shift of the entire display will shift according to 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	0	0	0	0	1	D	C	B

Display / cursor and flicker control Display the switch control position

D Display is on when D is 1, and off when D is 0, but the display data in DDRAM remains unchanged.

C: Cursor switch control position

Instructions for use

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

B: the cursor flicker switch control position

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

7.6.5 The cursor flashes / shifts

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

Without read or write the display data, move the cursor position or display left or right. This instruction is used to correct and find the display data, and move the cursor after the 40th character in row 1 to line 2. Note that in all rows, the display shift is performed simultaneously when the data is displayed repeatedly, each bit is independently shifted, and when the display shift, 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 follows the display shift

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

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

N: Display row number control bits

1 line when N is 0; 2 lines when N is 1.

F: Display the font settings

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

7.6.7 Set the CGRAM address

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

Place the CGRAM addresses into the address counter to make the CGRAM data from the MCU valid.

7.6.8 Set the 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 DDRAM data from the MCU valid, with 1 row address 00 H to 4 FH, row 1 DDRAM address from 00 H to 27 H and row 2 DDRAM address from 40 H to 67 H.

7.6.9 Read the busy logo and the 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 the BF is 1, the internal work is in progress, and you need to wait until the next instruction is set 0 before the BF can be executed. In this instruction, the value in the address counter can also be read.

7.6.10 Write the data to the RAM

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

Write the 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), where the address counter increases or decreases according to the RAM setting instruction.

7.6.11 Read the data from the RAM

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

8-bit data were read from DDRAM / CGRAM, The choice between DDRAM and CGRAM is determined by the address setting instruction, If the address setting instruction is not written before reading the data, The data readout is invalid; If the RAM address setting instruction is not written before the read operation, And read out the data multiple times, Starting with the second data is valid, The first data is incorrect, Because there is no timing coupled with the RAM data output, In DDRAM, The cursor transfer instruction plays the same role as the DDRAM address setting instruction, Also send RAM data to the output register, After reading the operation, The address counter increments or decreases according to the input mode instruction, After the CGRAM reading operation, Show that the shift may not perform correctly.

7.6.12 initialise

When powered on, the module will be initialized, during which the following instructions will be executed, and the busy flag bit will remain high until the end of the initialization.

1. Clear display instruction: All DDRAM is written to 20H

2. Set function instructions: DL=1,8-bit bus mode
   N=0, 1-row display mode
   F=0, 5x8 font

3. Display function instructions: D=0, display off
   C=0, Cursor Off
   B=0, blinking off

4. Set return command: I/D=1, incremental
   SH=0, display does not shift

7.7 Display the location table corresponding to the RAM addresses

Display position	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 position	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	11	12	13
Display position	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 position	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 position	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 position	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 position	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 position	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

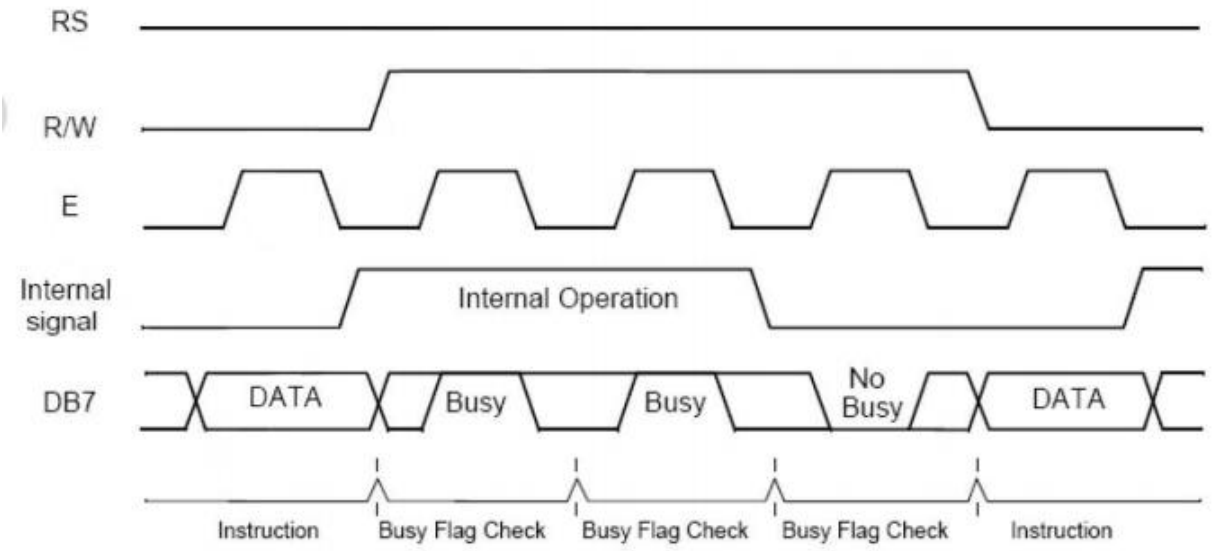
1-1 indicates the first character of the first line

Instructions for use of the LCD display module

8. time sequential routine

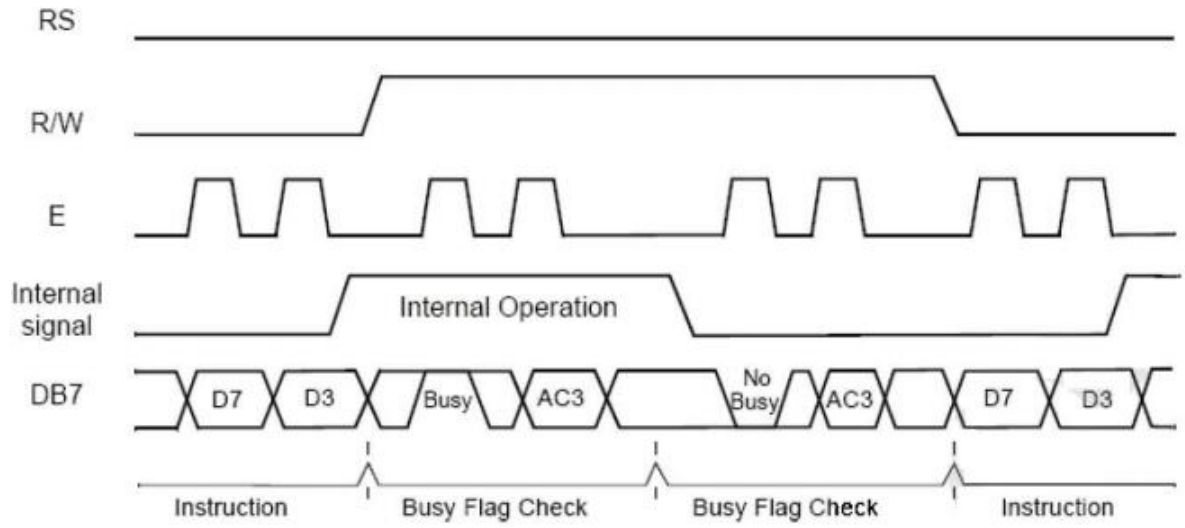
8.1 The 8-bit MCU interface

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



8.2 The 4-bit MCU interface

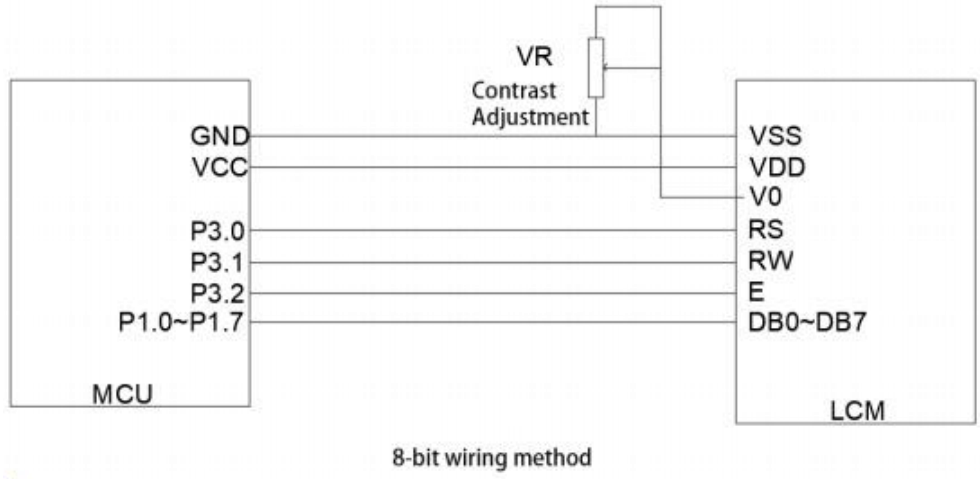
When the interface data width is set to 4 bits, the data is only read and written from the high 4 bits (DB4~DB7) of the 8-bit port, first transmitting high 4 bits and then transmitting low 4 bits. The timing is shown in the figure below:



Instructions for use

9. Application routines

9.1 8-bit wiring diagram



8-bit wiring method

9.2 The 8-bit C51 routine

```
#include <STC15.H>
#inc-lude <string.h>
#include<INTRINS.H>
#define uchar unsigned char
#defineuint unsigned int
#defineDB07 P1
sbit RS=P3^0;
sbit RW=P3^1;
sbit E=P3^2;
////////////////////////////////////
DB zone uchar code border _ inf [16] = {


0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,
0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,
};
uchar code string[]={ 0xa0,0xa1,0xa2,0xb0,0xb1,0xb2,0xc0,0xc1,0xc2,0xd0,0xd1,0xd
2,0xe0,0xe1,0xe2,0xf0, 0xf1,0xf2,0xa3,0xa4,0xa5,0xa6,0xb3,0xb4,0xb5,0xb6,0xc3,0xc
4,0xc5,0xc6,0xd3,0xd4, 0xd5,0xd6,0xe3,0xe4,0xe5,0xf3,0xf4,0xf5
};
```

```
void    delay(unsigned int m)          //time-delayed   program
                {
                  unsigned int i,j;
                  for(i=0;i<m;i++)
                     for(j=0;j<20;j++);
                }

void    delayms(unsigned int n)          //latency10×n millisecond   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=dat;
    E=1;
    delay(10);
    E=0;

}
```

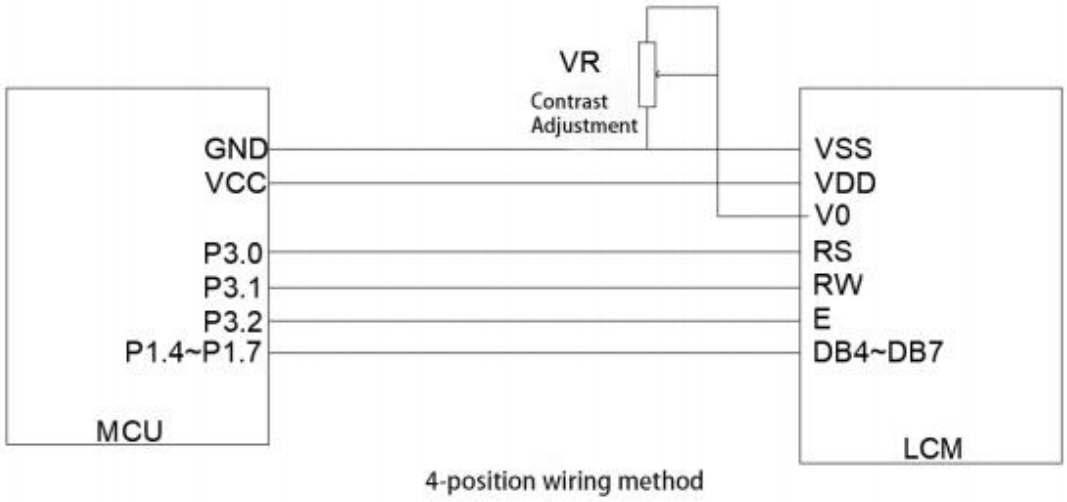
```
void LcdInit()
{
    LcdWriteCom(0x38);
    LcdWriteCom(0x0c);
    LcdWriteCom(0x06);
    LcdWriteCom(0x01);
    delayms(5);
}
void Show_string1(uchar a[])
{
    uchar i;
    LcdWriteCom(0x80);
    for(i=0;a[i]!='\0';i++)
    { LcdWriteData(a[i]);
    }
}

void Show_string2(uchar *a)
{
    uchar i;
    LcdWriteCom(0x80+0X40);
    for(i=0;i<16;i++)
    { LcdWriteData(a[i]);
    }
}

/*****************************************************************
* Name: Main ()
* Function: Main function
* Input: None
* Output:
no *****************************************************************
***/
void   main(void)
    {
```

```
                uchar i,j;
              LcdInit();
    LcdWriteCom(0x40);
       for(i=0;i<16;i++)
        { LcdWriteData(border_inf
        [i]); }
                while(1)
        {
                    for(j=0;j<2;j++)
      { LcdWriteCom(0x
   80);
   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-position wiring method

9.4 4-bit C51 routine

```
#include <STC15.H> #inc-
lude <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;

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

uchar code string[]={ 0xa0,0xa1,0xa2,0xb0,0xb1,0xb2,0xc0,0xc1,0xc2,0xd0,0xd1,0xd
 2,0xe0,0xe1,0xe2,0xf0, 0xf1,0xf2,0xa3,0xa4,0xa5,0xa6,0xb3,0xb4,0xb5,0xb6,0xc3,0xc
 4,0xc5,0xc6,0xd3,0xd4, 0xd5,0xd6,0xe3,0xe4,0xe5,0xf3,0xf4,0xf5
};

void delay(unsigned int m) //time-delayed program
        {
```

Instructions for use

```
            unsigned int i,j;
            for(i=0;i<m;i++)
              for(j=0;j<20;j++);
          }
void    delayms(unsigned int n)          //latency10×n millisecond  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)
{
    uchar dat1,dat2;
    dat1=dat&0xf0;
    dat2=(dat<<4)&0xf0;
```

Instructions for use

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

Instructions for use

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

/*****************************************************************
* Name: Main ()
* Function: Main function
* Input: None
* Output:
no *****************************************************************
***/
void   main(void)
    {
              uchar i,j;
              LcdInit();
     LcdWriteCom(0x40);
       for(i=0;i<16;i++)
        {
        LcdWriteData(border_inf[i]);
        }
              while(1)
        {
                    for(j=0;j<2;j++)
      { LcdWriteCom(0x
    80);
    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);
```

Instructions for use

}

}

9.5

word stock

10. matters need attention

Instructions for use

1. LCD Display (LCD)

LCDs are made of glass, organic sealant, organic fluid, and polymer-based polarizer. The following matters should be paid attention to when handling:, (1). Keep the temperature within the use and storage range. Excessive temperature and humidity can cause polarization degradation, polarizer stripping, or air bubbles. (2). Do not touch the exposed polarizer with something harder than the HB pencil lead. Remove dust from the display surface and gently wipe with cotton and soak deer tamkins or other soft materials in cleaning oil.

（3） Immediately erase the saliva or water droplets. Excessive ITO contact with water can deform or change the liquid crystal display surface.

（4） Glass is easily broken by rough operation. Especially in the corners and at the edges.

（5） . Do not use a DC voltage to drive the LCD display.

2. LCD display module

2.1 Mechanical precautions

LCM assembly and adjustment with high accuracy. Avoid excessive vibration and make no 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 profile, moving its components, or modifying its pattern.

（3） . Do not touch the elastomer connector, especially insert the backlight panel (e. g., EL).

（4） . When installing the LCM, ensure that the PCB plate is not under any pressure, such as bending or twisting. The elastomer contact is very fine and any slight misalignment of any element can result in missing pixels.

（5） . Avoid pressing on the metal baffle, otherwise the elastomer connector may deform and lose contact, resulting in the loss of pixels.

2.2. static electricity

The LCM contains CMOS LSI and the same precautions should be taken for such equipment, i. e

（1） . When the operator contact the module. Do not use any body with any conductive part, such as the LSI pad, copper wires on the PCB, and interface terminals.

（2） . The modules shall be stored in antistatic bags or other antistatic containers.

（3） . Only a properly grounded flipping iron can be used.

（4） . If an electric screwdriver is used, be well grounded and prevent sparks from commutators.

（5） The work clothes and workstations shall be subject with normal anti-static measures; for the latter, conductive (rubber) pads are recommended.

（6） . Because dry air senses static electricity, the relative humidity is recommended to be 50-60%.

2.3. weld

（1） . Weld only to the I / O terminals.

（2） . Only use the correct grounding and no leakage of the flipping iron.

（3） . Welding temperature: 280℃± 10℃

（4） . Welding time: 3 to 4 seconds.

（5） . Low ogenic solder filled with resin flux.

（6） . If flux is used, the LCD surface shall be covered to avoid flux splash. The flux residues shall be removed after protection.

2.4. operate

（1） The viewing angle can be adjusted by changing the LCD drive voltage V0.

（2） The drive voltage shall be kept within the specified range, too high voltage will shorten the life of the display. (3) The response time increases with the decrease of the temperature.

（4） . At temperatures above its operating range, the display may turn black or dark blue; this (but do not press the display area) may cause part of the display segment to "break".

（5） . Mechanical damage during operation (such as pressing the display area) may cause "breakage" in the line segment.

2.5. lay in

If any liquid leaks from the damaged glass battery, rinse any contact body parts with soap and water. Do not swallow the liquid. The toxicity is extremely low, but you should always be careful.

2.6. Limited warranty

Unless otherwise agreed with the Customer, within one year from the date of shipment, when its electrical and cosmetic defects are found under the acceptance criteria, the confirmation of the date shall be based on freight documents and the warranty liability shall be limited to the above clause

Instructions for use

Repair and / or replacement performed. Will not be responsible for any subsequent or consequential events.