

目 录

- 1 . 修订记录
- 2 . 概述
- 3 . 外形尺寸
- 4 . 硬件方框图
- 5 . 电气特性
- 6 . 接口说明
- 7 . 指令说明
- 8 . 操作时序说明
- 9 . 应用例程
- 10 . 注意事项

1. 修订记录

版本	发行日期	新制/修订内容
V1.0	2020-8-12	新制

注：升级版本向下兼容，不做另行通知，如遇兼容问题影响性能请联系本公司解决

2.概述

1602-7 字符型液晶显示模块是专门用于显示字母、数字元、符号 等的点阵型液晶显示模块。分 4 位和 8 位数据传输方式。提供 5×7 点阵+光标的显示模式。提供显示数据缓冲区 DDRAM、字符发生器 CGROM 和字符发生器 CGRAM，可以使用 CGRAM 来存储自己定义的最多 8 个 5×8 点阵的 图形字符的字模数据。提供了丰富的指令设置：清显示；光标回原点；显示开/关；光标开/关；显示字符闪烁；游标移位；显示移位元等。提供内部上电自动复位电路，当外加电源时，自动对模块进行初始化操作，将模块设置为默认的显示工作状态。

显示字符数: 16 字符 X 2 行

字符点阵:5X7字阵+光标

显示颜色及背光颜色: STN 蓝,黄绿,灰; 背光 黑,白,黄绿

偏光膜:全透/半透

观察角度: 6:00

显示占空比: 1/16 **驱动偏压:** 1/5

控制芯片: SPLC780D或兼容IC (如AIP31066)

字符发生器 ROM (CGROM): 10880 bits (192 character 5*8 dots) 或
(64 character 5*11 dots)

字符发生器 RAM (CGRAM): 64X8 bits (8 characters 5*8 dots)或
(4 characters 5*11 dots)

显示数据 RAM (DDRAM) :80X8 bits (80 characters max)

尺寸 (Unit: mm)

外形尺寸: 80X36X11.3

可视区域 : 64.4X16

字符字体: 5X7 dots + 光标

字符尺寸:55.45X11

点尺寸:0.55X0.615

字符间距: 3.5X5.73

重量: g

对比度: V0外部调节或内部固定对比度

工作电压: +3.3V或+5V 默认5V

5.电气特性

5.1极限参数

参数名称	符号	条件	典型值		单位
			最小值	最大值	
电源电压	Vdd		-0.3	7.0	V
LCD驱动电压	V5		Vdd-10.0	Vdd+0.3	V
输入电压	Vi		-0.3	Vdd+0.3	V
工作温度(T)	Top	-	-20	70	°C
储存温度(T)	Tstg	-	-30	80	°C

5.2.1 直流参数1(Ta=25°C,Vdd=4.5V~5.5V)

参数名称	符号	条件	标称值			单位
			最小	典型	最大	
电源电压	Vdd-GND	-	4.5	5.0	5.5	V
工作电流 (不包括背光)	Idd	Vdd=5V	0.9	1.5	1.7	mA
LCD驱动电流	Iee		-	0.6	-	mA
LCD驱动电压	Vdd-V5		4.2	4.5	4.8	V
LED背光工作电流	If	Vf=3.0~3.2V	17	18	20	mA
LED背光功耗	Pd		90	100	110	mW
输入高电平	Vih		2.5	-	Vdd	V
输入低电平	Vil		-0.3	-	0.6	V
输出高电平	Voh	Ioh=-0.205mA	2.4	-	-	V
输出低电平	Vol	Iol=1.2mA	-	-	0.4	V

5.2.2 直流参数2(Ta=25°C,Vdd=2.7V~4.5V)

参数名称	符号	条件	标称值			单位
			最小	典型	最大	
电源电压	Vdd-GND	-	2.7	3.3	4.5	V
工作电流 (不包括背光)	Idd	Vdd=3.3V	0.45	0.9	1.0	mA
LCD驱动电流	Iee		-	0.6	-	mA
LCD驱动电压	Vdd-V5		4.2	4.5	4.8	V
LED背光工作电流	If	Vf=3.0~3.2V	17	18	20	mA
LED背光功耗	Pd		55	60	66	mW
输入高电平	Vih		0.7Vdd	-	Vdd	V
输入低电平	Vil		-0.3	-	0.55	V
输出高电平	Voh	Ioh=-0.1mA	0.75Vdd	-	-	V
输出低电平	Vol	Iol=0.1mA	-	-	0.2Vdd	V

5.3.1 交流参数1(Ta=25°C,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	t _c	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 交流参数2(Ta=25°C,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	t _c	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	-	-	

6.接口说明

脚号	符号	功能	备注
1	Vss	电源供应	0V
2	Vdd		+5V
3	Vo		LCD偏压调节
4	RS	数据/指令选择(H:数据 L: 指令)	
5	R/W	读/写选择 (H:读 L:写)	
6	E	使能信号	
7	DB0	数据位 0	
8	DB1	数据位1	
9	DB2	数据位2	
10	DB3	数据位3	
11	DB4	数据位4	
12	DB5	数据位5	
13	DB6	数据位6	
14	DB7	数据位7	
15	A	LED 背光正	
16	K	LED 背光负	

7.指令说明

模块具有4位/8位MCU并行通讯模式，4位/8位总线通过指令寄存器的DL位进行选定。

在读写操作时，使用到2个8位寄存器，一个是数据寄存器DR，另一个是指令寄存器IR。数据寄存器DR作为写入和读出DDRAM/CGRAM数据的临时存放地，目标RAM通过RAM地址设定指令进行选定，任何读写RAM的内部操作都是自动完成的，当MCU读出DR内数据，DDRAM/CGRAM数据自动传输到DR，同样，当MCU写入数据到DR，DR中的数据自动传输出DDRAM/CGRAM。指令寄存器IR用于存储来自MCU的指令代码，MCU不能读出指令数据，可通过RS引脚切换选取寄存器。

通过设置RS/RW位的各种操作：

RS	RW	操作
L	L	写指令操作 (MCU写指令代码至IR)
L	H	读忙标志 (DB7) 和地址计数器 (DB0~DB6)
H	L	写数据操作 (MCU写数据至DR)
H	H	读数据操作 (MCU从DR读出数据)

7.1 忙标志 (BF)

BF为高，表示内部操作正在进行，所以在这个时间内，下一条指令将不能被执行。当RS="0"且R/W="1"（读指令操作时），BF的值可以从DB7口读出，在执行下一条指令时，必须确认BF不为"1"。

7.2 地址计数器 (AC)

从指令寄存器过来的DDRAM/CGRAM地址存储在地址计数器时，地址计数器内的数据在写入或读出DDRAM/CGRAM后递增或递减，当RS="0"且R/W="1"时，地址计数器中的数据可以从DB0~DB6读出。

7.3 显示数据RAM (DDRAM)

模块的DDRAM地址范围为00H~27H和40H~67H。

7.4 字符生成ROM (CGROM)

字符生成ROM具有5X8点阵、192字符，和5x11点阵、64字符模式

7.5 字符生成RAM (CGRAM)

字符生成RAM具有5*8点阵、8个字符空间，每个字符空间由8个5位字节空间组成，通过写入自定义字符数据至CGRAM，此8个字符空间用户可通过写入DDRAM数据00H~07H以调用自定义字符。

DDRAM/CGRAM地址对应图：

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	x	0	0	0	0	0	0	0	0	0	x	x	x	0	1	1	1	0	pattern1
											0	0	1				0	0				
											0	1	0				0	0				
											0	1	1				0	0				
											1	0	0				0	0				
											1	0	1				0	0				
											1	1	0				0	0				
											1	1	1				0	0				
0	0	0	0	x	0	0	1	0	0	1	0	0	0	x	x	x	0	1	1	1	0	pattern2
											0	0	1				0	1				
											0	1	0				0	0				
											0	1	1				0	0				
											1	0	0				0	0				
											1	0	1				0	0				
											1	1	0				0	0				
											1	1	1				0	0				
*				*				*														
0	0	0	0	x	1	1	1	1	1	1	0	0	0	x	x	x	1	1	1	1	1	pattern8
											0	0	1				0	0				
											0	1	0				0	0				
											0	1	1				0	0				
											1	0	0				0	0				
											1	0	1				0	0				
											1	1	0				0	1				
											1	1	1				0	0				

7.6 指令描述

指令表

指令	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	执行时间	描述
清除显示	0	0	0	0	0	0	0	0	0	1	1.53ms	将20H写入DDRAM，将地址计数器地址设为00H
返回	0	0	0	0	0	0	0	0	1	-	1.53ms	将地址计数器地址设为00H，并将光标恢复至初始位置，DDRAM内容保持不变
输入模式设置	0	0	0	0	0	0	0	1	I/D	SH	39us	设置光标移动方向，并允许整个显示移动
显示开/关	0	0	0	0	0	0	1	D	C	B	39us	设置显示、光标，光标的闪烁控制位

移位	0	0	0	0	0	1	S/C	R/L	-	-	39us	设置光标移动、显示移动方向的控制位, DDRAM数据保持不变
功能设置	0	0	0	0	1	DL	N	F	-	-	39us	设置接口数据宽度 (DL:8位/4位), 显示行数 (N:2行/1行), 显示字体 (F:5x11点阵/5x8点阵)
设置CGRAM地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	39us	在地址计数器内设置CGRAM地址
设置DDRAM地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	39us	在地址计数器内设置DDRAM地址
读忙标志&地址	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	0us	读取忙标志位BF, 地址计数器中的内容同时被读出
写数据	1	0	D7	D6	D5	D4	D3	D2	D1	D0	43us	写数据至内部RAM (DDRAM/CGRAM)
读数据	1	1	D7	D6	D5	D4	D3	D2	D1	D0	43us	从内部RAM(DDRAM/CGRAM)中读取数据

注:“-”为不考虑

7.6.1 清除显示

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

通过写入20H (空格代码) 至所有的DDRAM地址和设置地址计数器为00H,可以清除显示数据, 将光标放在初始状态位置, 设置输入模式为递增 (I/D为高)。

7.6.2 返回

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

“-”为不考虑

返回指令是将光标回到起始位置, 将DDRAM地址设置为00H写入地址计数器, 并将显示改为初始状态, DDRAM中的数据保持不变。

7.6.3 输入模式

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

设置光标和显示的移动方向

I/D: DDRAM地址 (光标或闪烁) 的递增或递减

当I/D为1时, 光标闪烁向右移, DDRAM地址为递增; 当I/D为0时, 光标闪烁向左移, DDRAM地址递减, 当读出或写入CGRAM时, 移动方式与DDRAM一致。

SH: 显示移位

当SH为0时, 对DDRAM或CGRAM读写操作时, 整个显示不会位移。当SH为1时, 对DDRAM写操作是, 整个显示的移位将根据I/D设定的方向位移。

7.6.4 显示开关控制

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

显示/光标及闪烁控制

D: 显示开关控制位

当D为1时, 显示开启; 当D为0时, 显示关闭, 但DDRAM中显示数据保持不变。

C: 光标开关控制位

当C为1时，光标开；当C为0时，光标消失，但I/D寄存器保存它的数据。

B: 光标闪烁开关控制位

当B为1时，光标闪烁开，当B为0时，光标闪烁关。

7.6.5 光标闪烁/移位

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

不读写显示数据，将光标位置或显示左移或右移，这种指令用于纠正和寻找显示数据，当2行显示模式时，在第1行的第40个字符后，光标移向第2行。注意在所有行中，显示移位是同时进行的当显示数据重复移位时，每一位是独立移位的，当显示移位时，地址计数器中的内容保持不变

移位格式表

S/C	R/L	操作
0	0	光标向左移，地址计数器递减1
0	1	光标向右移，地址计数器递增1
1	0	所以显示向左移，光标跟随显示移位
1	1	所有显示向右移，光标跟随显示移位

7.6.6 功能设置

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

DL: 接口数据宽度控制位

当DL为1时，表示8位总线连接至MCU

当DL为0时，表示4位总线连接至MCU，当为4位总线模式时，8位数据需要通过传输4位数据2次完成。

N: 显示行数控制位

当N为0时，为1行显示模式；当N为1时，为2行显示模式。

F: 显示字体设置

当F为0时，字体为5x8点阵模式；当F为1时，字体为5x11点阵模式。

7.6.7 设置CGRAM地址

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

将CGRAM地址置入地址计数器，使得来自MCU的CGRAM数据有效。

7.6.8 设置DDRAM地址

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

将DDRAM地址置入地址计数器，使得来自MCU的DDRAM数据有效，当1行显示模式时，DDRAM地址00H~4FH；当2行显示模式时，第1行的DDRAM地址从00H~27H，第2行DDRAM地址从40H~67H。

7.6.9 读忙标志和地址

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

该指令显示模块是否处于内部工作中，如果BF为1，内部工作在进行中，需要等待直到BF被置0时，下条指令才能被执行，在这条指令中，同样可以读到地址计数器内的值。

7.6.10 写数据到RAM

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

写入8位数据至DDRAM/CGRAM。DDRAM和CGRAM之间的选取由地址设置指令来决定（DDRAM地址设置指令和CGRAM地址设置指令），在写入操作后地址会根据RAM设置指令设置地址计数器递增或递减。

7.6.11 从RAM中读数据

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

从DDRAM/CGRAM中读取8位数据，DDRAM和CGRAM之间的选择由地址设置指令来决定，如果读数据之前没有写入地址设置指令，则读出的数据是无效的；如果读操作之前没有写入RAM地址设置指令，且读出多次数据，则从第二个数据开始是有效的，第一个数据是不对的，因为没有时序配合RAM数据输出，在DDRAM的读操作中，光标转移指令起到了DDRAM地址设置指令相同的作用，同样将RAM数据送至输出寄存器，在读操作后，地址计数器根据输入模式指令递增或递减，在CGRAM读操作后，显示移位可能不能正确执行。

7.6.12 初始化

上电时，模块会进行初始化，在这个过程中会执行以下指令，初始化结束前忙标志位保持为高。

1. 清除显示指令：所有DDRAM被写入20H
2. 设置功能指令：DL=1，8位总线模式
N=0，1行显示模式
F=0，5x8 字体
3. 显示开关指令：D=0，显示关
C=0，光标关
B=0，闪烁关
4. 设置返回指令：I/D=1，递增
SH=0，显示不移位

7.7显示位置与RAM地址对应表

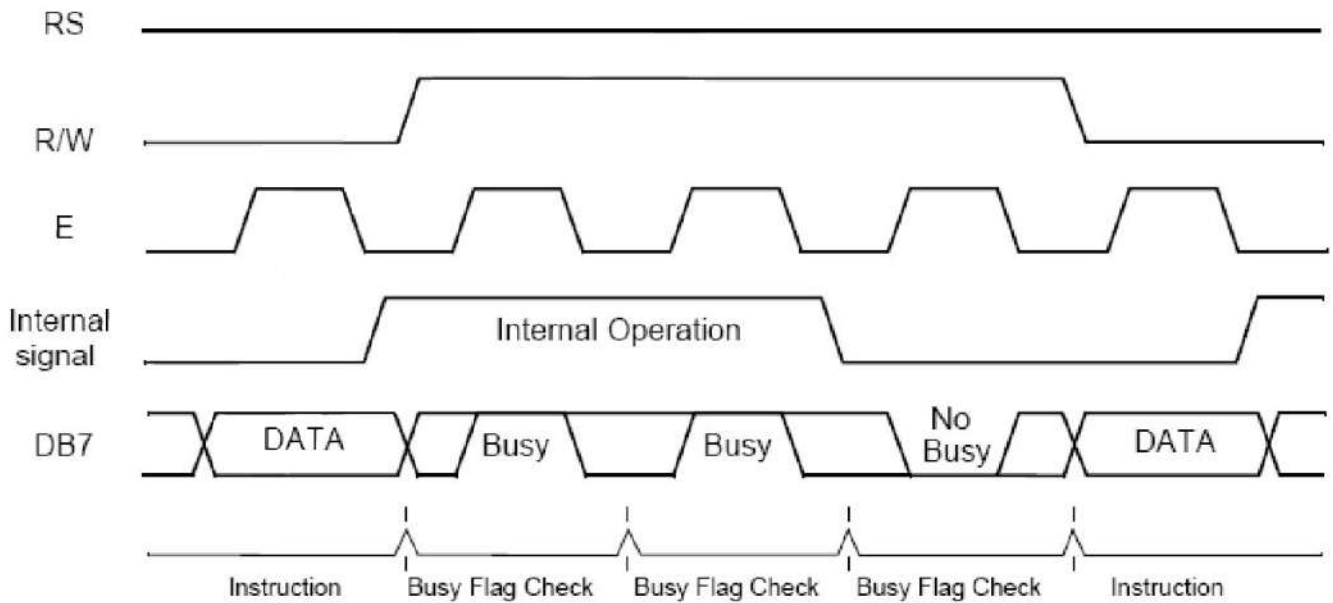
显示位置	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9	1-10
DDRAM 地址	00	01	02	03	04	05	06	07	08	09
显示位置	1-11	1-12	1-13	1-14	1-15	1-16	1-17	1-18	1-19	1-20
DDRAM地址	0A	0B	0C	0D	0E	0F	10	11	12	13
显示位置	1-21	1-22	1-23	1-24	1-25	1-26	1-27	1-28	1-29	1-30
DDRAM 地址	14	15	16	17	18	19	1A	1B	1C	1D
显示位置	1-31	1-32	1-33	1-34	1-35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

1-1 表示第一行的第一个字符

8.操作时序

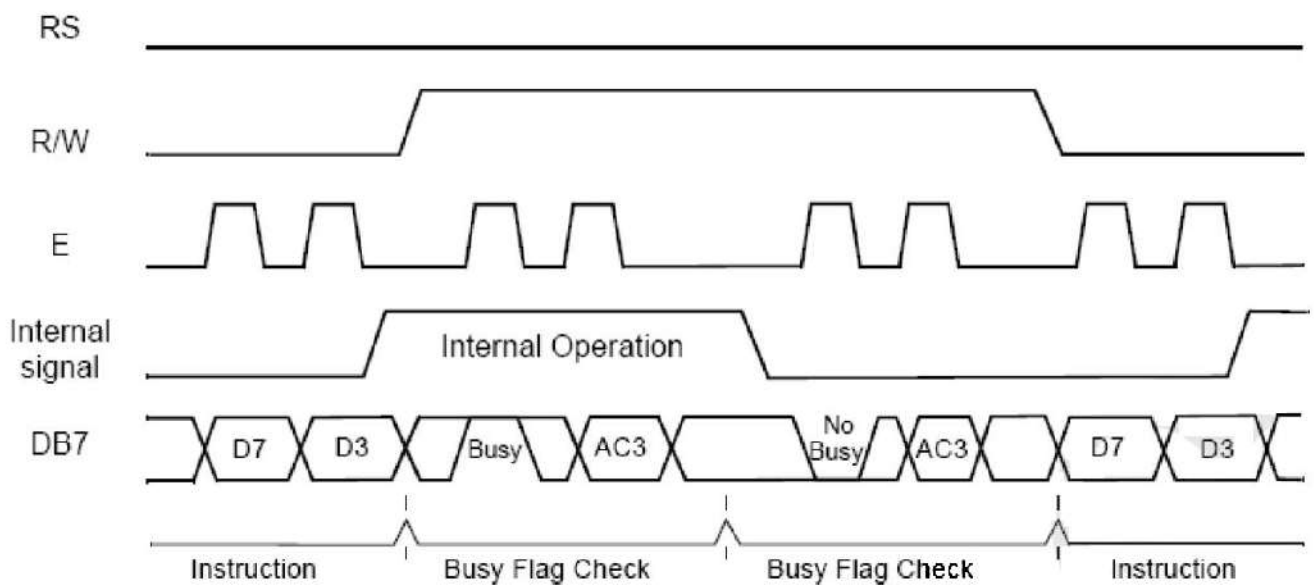
8.1 8位MCU接口

当接口数据宽度设置为8位时，数据从8位端口（DB0~DB7）同时读写，时序如下图所示：



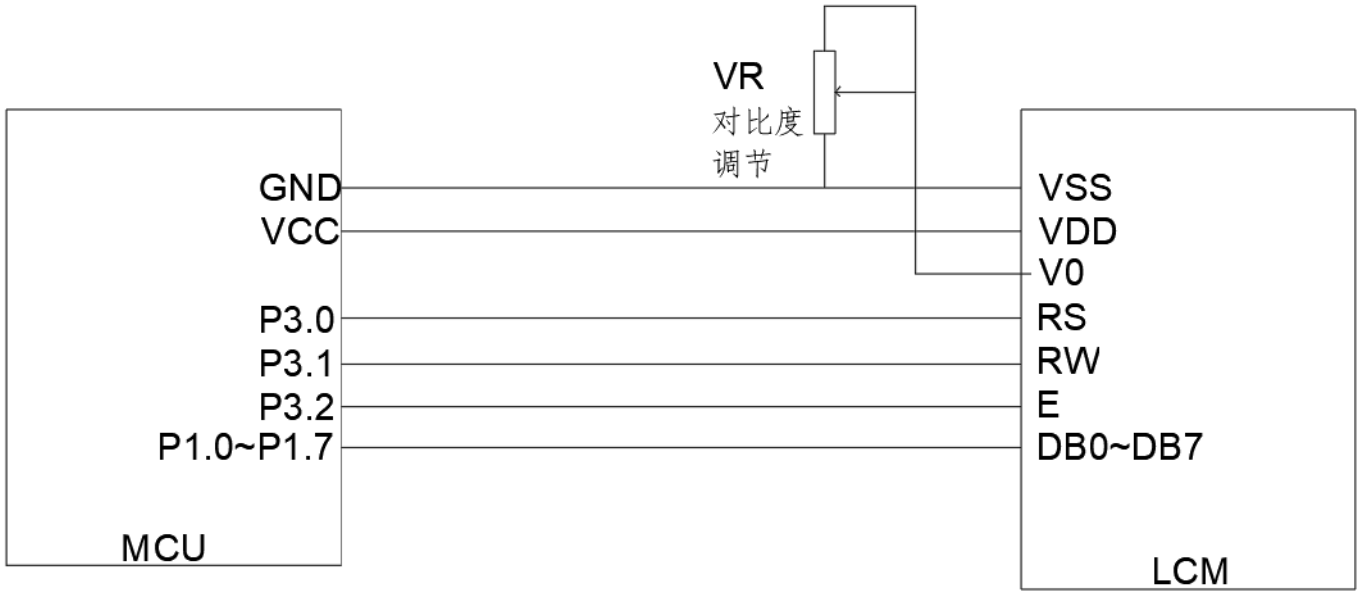
8.2 4位MCU接口

当接口数据宽度设置为4位时，数据仅从8位端口的高4位（DB4~DB7）进行读写，先传输高4位再传输低4位，时序如下图所示：



9.应用例程

9.1 8位接线图



8位接线方法

9.2 8位C51例程

```
#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区
uchar code border_inf[16] = {
    0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,
    0x55,0xaa,0x55,0xaa,0x55,0xaa,0x55,0xaa,
};
uchar code string[]={
    0xa0,0xa1,0xa2,0xb0,0xb1,0xb2,0xc0,0xc1,0xc2,0xd0,0xd1,0xd2,0xe0,0xe1,0xe2,0xf0,
    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) //延时程序
    {
        unsigned int i,j;
        for(i=0;i<m;i++)
            for(j=0;j<20;j++);
    }

void delays(unsigned int n) //延时10×n毫秒程序
    {
        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]);
    }
}

```

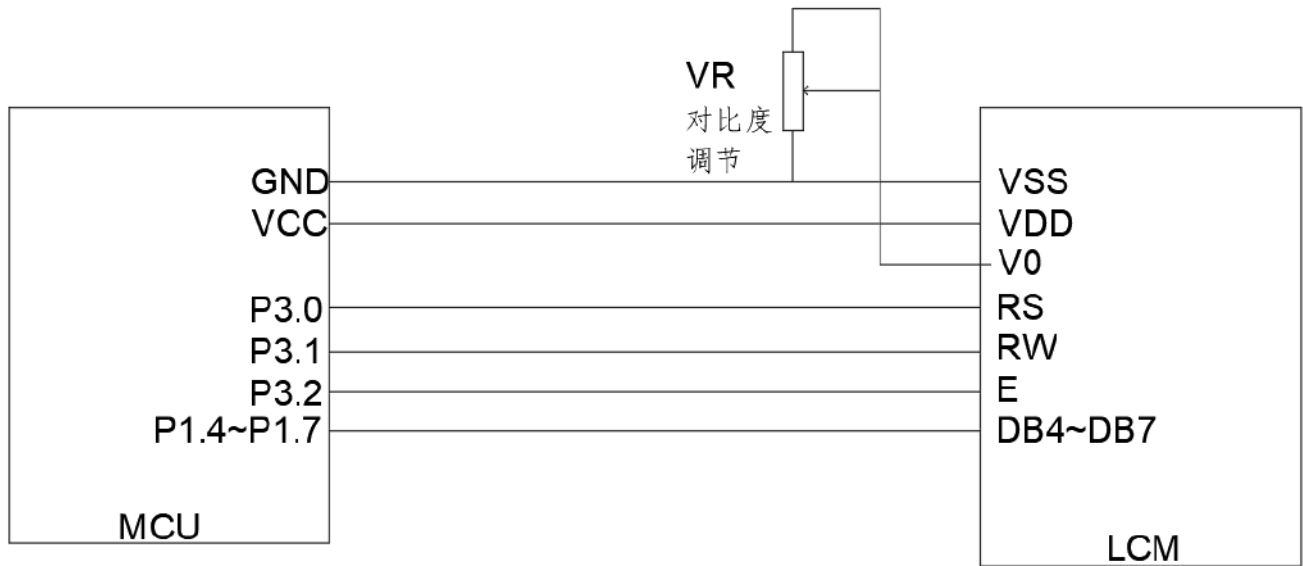
```

/*****
* 名称 : Main()
* 功能 : 主函数
* 输入 : 无
* 输出 : 无
*****/
/
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(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位接线图



4位接线方法

9.4 4位C51例程

```
#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;

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

uchar code string[]={
    0xa0,0xa1,0xa2,0xb0,0xb1,0xb2,0xc0,0xc1,0xc2,0xd0,0xd1,0xd2,0xe0,0xe1,0xe2,0xf0,
    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) //延时程序
```

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

```
void delayms(unsigned int n)    //延时10×n毫秒程序
```

```
    {
        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;
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++)
```

```

    {
        LcdWriteData(a[i]);
    }
}

/*****
* 名称 : Main()
* 功能 : 主函数
* 输入 : 无
* 输出 : 无
*****/
/
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(0x80);
            for(i=0;i<16;i++)
            {
                LcdWriteData(j);
            }
            LcdWriteCom(0xC0);
            for(i=0;i<16;i++)
            {
                LcdWriteData(j);
            }
            delayms(500);
        }
        Show_string1("12345abcef!@#%");
        Show_string2(string);
    }
}

```

```
delays(600);
```

```
  }
```

```
}
```

9.5 字库

b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)			0	@	P	\	F				-	9	3	o	p
0001	(2)	!	1	A	Q	a	q			.	7	7	6	ã	q	
0010	(3)	"	2	B	R	b	r			7	7	9	x	p	o	
0011	(4)	#	3	C	S	c	s			7	7	7	e	e	o	
0100	(5)	\$	4	D	T	d	t			7	7	7	p	p	a	
0101	(6)	%	5	E	U	e	u			.	7	7	1	o	o	
0110	(7)	&	6	F	V	f	v			7	7	7	o	p	z	
0111	(8)	*	7	G	W	g	w			7	7	7	7	g	π	
1000	(1)	(8	H	X	h	x			7	7	7	7	7	7	
1001	(2))	9	I	Y	i	y			o	7	7	7	7	7	
1010	(3)	*	:	J	Z	j	z			z	o	o	7	7	7	
1011	(4)	+	;	K	L	k	l			7	7	7	7	7	7	
1100	(5)	,	<	L	#	l	l			7	7	7	7	7	7	
1101	(6)	-	=	M	^	m	^			z	7	7	7	7	7	
1110	(7)	.	>	N	^	n	^			7	7	7	7	7	7	
1111	(8)	/	?	O	_	o	_			7	7	7	7	7	7	

10.注意事项

1.液晶显示器 (LCD)

液晶显示器是由玻璃, 有机密封胶, 有机流体, 和聚合物基偏光片。搬运时应注意以下事项:

- (1) .保持温度在使用和储存范围内。过高的温度和湿度会导致偏振退化、偏振器剥离或气泡。
- (2) .不要用比HB铅笔芯更硬的东西接触暴露的偏光镜。清除显示器表面的灰尘, 用棉花轻轻擦拭, 鹿皮巾或其他软材料浸泡在清洁油中。
- (3) 立即擦掉唾液或水滴。ITO与水接触时间过长, 会导致液晶显示器表面变形或变色。
- (4) 玻璃很容易因粗暴的操作而碎裂。尤其是在角落和边缘。
- (5) .不要用直流电压驱动液晶显示器。

2.液晶显示模块

2.1机械注意事项

LCM的装配和调整具有高精度。避免过度震动, 不要做任何改动或修改。应注意以下几点。

- (1) .不要以任何方式改变金属框架上的凸耳。
- (2) .请勿通过钻额外的孔、更改其轮廓、移动其组件或修改其图案来修改PCB。
- (3) .请勿触摸弹性体连接器, 尤其是插入背光面板(例如, EL)。
- (4) .安装LCM时, 请确保PCB板不受任何压力, 如弯曲或扭曲。弹性体接触非常精细, 任何元素的轻微错位都可能导致像素缺失。
- (5) .避免压在金属挡板上, 否则弹性体连接器可能会变形和失去接触, 从而导致丢失像素。

2.2.静电

LCM包含CMOS LSI, 对此类设备应采取相同的预防措施, 即

- (1) .当操作员与模块接触时, 应将其接地。切勿用人体任何部位接触任何导电部件, 如LSI焊盘、PCB上的铜导线和接口端子。
- (2) .模块应保存在防静电袋或其他防静电容器中储存。
- (3) .只能使用正确接地的烙铁。
- (4) .如果使用电动螺丝刀, 应良好接地, 并防止换向器火花。
- (5) .工作服和工作台应遵守正常的防静电措施; 对于后者, 建议使用导电(橡胶)垫。
- (6) .由于干燥空气会感应静电, 建议相对湿度为50-60%。

2.3.焊接

- (1) .仅焊接至I/O端子。
- (2) .只能使用接地正确且无漏电的烙铁。
- (3) .焊接温度: $280^{\circ}\text{C} \pm 10^{\circ}\text{C}$
- (4) .焊接时间: 3到4秒。
- (5) .使用树脂助焊剂填充的低温焊锡。
- (6) .如果使用助焊剂, 应覆盖LCD表面, 以避免焊剂飞溅。助焊剂残留物应在防护后清除。

2.4.操作

- (1) 观察角度可以通过改变LCD驱动电压V0来调节。
- (2) 驱动电压应保持在规定的范围内, 过高的电压会缩短显示器的寿命。(3) 响应时间随着温度的降低而增加。
- (4) .在高于其工作范围的温度下, 显示器可能会变成黑色或深蓝色; 这(但是不要按压显示区域)可能会导致部分显示线段“断裂”。
- (5) .操作过程中的机械损害(如按压显示区域)可能会导致线段出现“断裂”。

2.5.储存

如果有液体从损坏的玻璃电池中漏出, 将任何接触的人体部分用肥皂和水冲洗干净。切勿吞下液体。毒性极低, 但应始终小心。

2.6.有限保修

除非与客户另有约定，从装运日期起一年内，当根据验收标准检查时发现其电气和外观缺陷，将维修或修理其任何LCD和IC，该日期的确认应以货运单据为依据，保修责任仅限于根据上述条款进行的维修和/或更换。不会对任何后续或后果性事件负责。