

lcd1602资料

一、功能介绍

1. 主要用来显示数字、字母、图形以及少量自定义字符
2. 可以显示 2行16个字符
3. 拥有16个引脚
 - 8位数据总线D0-D7
 - RS、R/W、EN三个控制端口

二、工作原理

液晶：液态晶体，是一种几乎透明的物质，是不能发光的

光源：lcd屏幕的光来自于屏幕最下面的背光板发出的光源

偏光片：一种镜片，只有特定方向的光线才能通过，而其它方向的光线不能通过

液晶的物理特性：液晶控制光线通过的程度是由加在液晶上的电压强弱来控制的

通电导通时：排列变得有秩序，使光线容易通过，光的路径不改变

不通电时：排列混乱，阻止光线通过，改变光的路径

也就是说：

lcd屏幕不显示：背光源的光不穿过上偏光片即可

lcd屏幕显示字符：背光源的光穿过上偏光片

也就是说，我们只需要控制每个像素点的电压，就可以达到lcd屏幕显示的效果，这样就可以通过不同区域的电压控制液体水晶重新排列来显示出各种图形图像以及字符

1.引脚介绍

lcd1602共有16个管脚

1. 一个VCC和GND用于给1602供电
2. 一个VCC和GND用于给背光源供电
3. D0~D7分别为8位双向数据线，用于传输数据
4. RS（数据命令选择端）：高电平传输的是数据，低电平传输的是指令
5. R/W（读写选择端）：高电平进行读操作，低电平进行写操作
6. E（使能信号）

2.LCD1602的RAM地址及标准字库表

LCD1602，共显示16行2列，对应着32个RAM地址，在使用时，需要在哪个位置显示，就写入对应的RAM地址，然后再写入需要的字符，对应就会显示该字符

在执行每条指令之前一定要确认模块的忙标志为低电平，表示lcd此时不忙，这时才能写指令或数据，否则此指令失效。

要显示字符时要先输入显示字符地址（写指令），也就是告诉模块，在哪里显示字符，然后再写入需要显示的字符（写数据），才能正常显示字符

DDRAM地址与显示位置的对应关系

00H 01H 02H 03H 04H 05H 06H 07H 08H 09H 0AH 0BH 0CH 0DH 0EH 0FH
40H 41H 42H 43H 44H 45H 46H 47H 48H 49H 4AH 4BH 4CH 4DH 4EH 4FH

3.LCD1602读写操作

读操作：读状态 和 读数据

写操作：写指令 和 写数据

读状态：读取LCD引脚状态，返回状态字，D0~D6为当前LCD数据指针的地址，D7为是否允许读写操作（即检查LCD是否处于忙状态）

读数据：读取D0~D7内的数据

写指令：写入LCD的控制指令

写数据：写入需要显示的数据

读状态

引脚电平：RS = 0(指令) R/W = 1(读操作) E = 1(使能)

读数据

引脚电平：RS = 1(数据) R/W = 1(读操作) E = 1(使能)

写指令

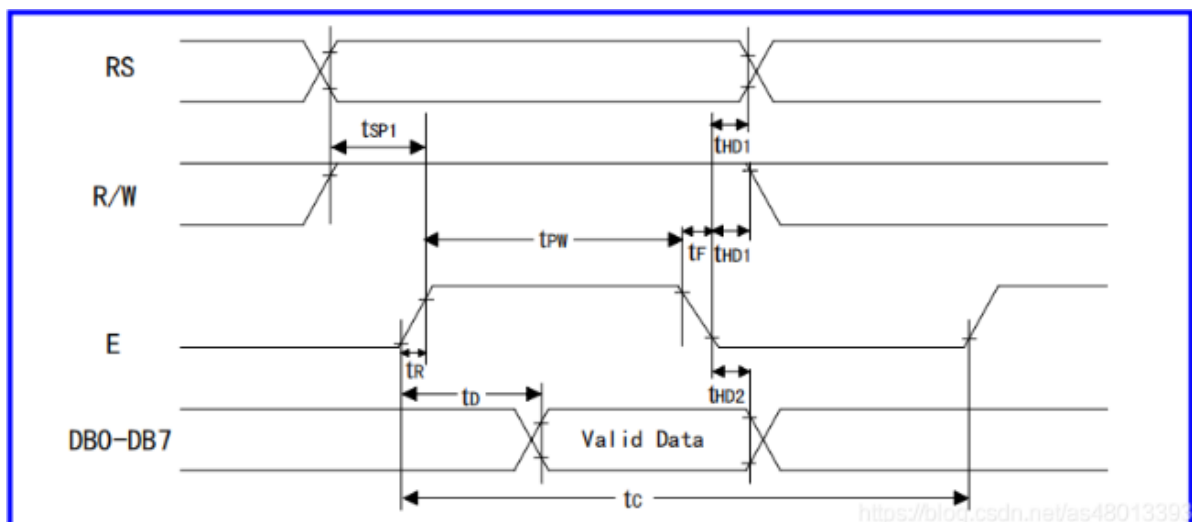
引脚电平：RS = 0(指令) R/W = 0(写操作) D0~D7(指令码) E = 高脉冲

写数据

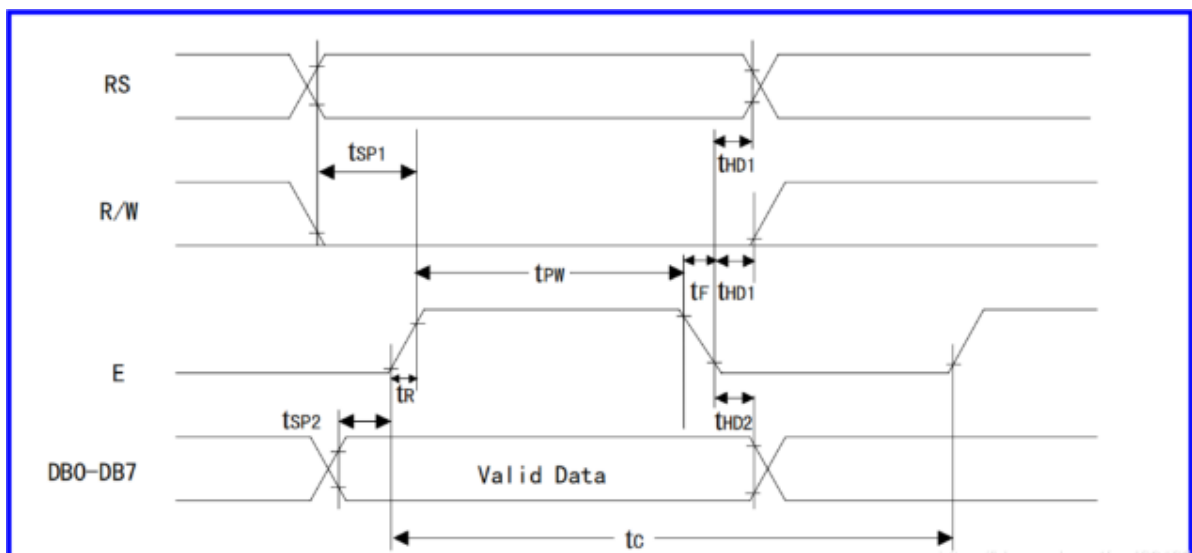
引脚电平：RS = 1(数据) R/W = 0(写操作) D0~D7(数据) E = 高脉冲

4.读写操作时序图

读操作时序：



写操作时序：



当我们要 写指令 的时候：

1. RS = 0, R/W = 0, EN = 0
2. 将指令送到D0~D7
3. 延时tsp1, 让1602准备接收指令
4. 将EN拉高, 产生一个上升沿, 这时候 指令 就开始写入lcd
5. 延时一段时间
6. 将EN置低电平

当我们要 写数据 的时候

1. RS = 1, R/W = 0, EN = 0
2. 将数据送到D0~D7
3. 延时tsp1, 让1602准备接收数据
4. 将EN拉高, 产生一个上升沿, 这时候 数据 就开始写入lcd
5. 延时一段时间
6. 将EN置低电平

写指令代码`

```
sbit INFO = 0xA0;

void LcdwriteCmd(unsigned char cmd)
{
    //1.RS = 0, R/W = 0, EN = 0
    RS = 0; //选择写指令
    RW = 0; //选择写
    EN = 0; //使能拉低

    //2.将指令送到D0~D7
    INFO = cmd; //把指令送到D0~D7

    //3.延时tsp1, 让1602准备接收指令
    delay_ms(2);

    //4.将EN拉高, 产生一个上升沿, 这时候 指令 就开始写入lcd
    EN = 1;

    //5.延时一段时间
    delay_ms(2);

    //6.将EN置低电平
    EN = 0;
}
```

写数据代码

```
sbit INFO = 0xA0;

void LcdwriteData(unsigned char data)
{
    //1.RS = 1, R/W = 0, EN = 0
    RS = 1; //选择写数据
    RW = 0; //选择写
    EN = 0; //使能拉低

    //2.将指令送到D0~D7
    INFO = data;
```

```

//3.延时tsp1，让1602准备接收数据
delay_ms(2);

//4.将EN拉高，产生一个上升沿，这时候 数据 就开始写入lcd
EN = 1;

//5.延时一段时间
delay_ms(2);

//6.将EN置低电平
EN = 0;
}

```

三、LCD1602的指令操作

LCD1602一共有11条指令

1.清显示（01H）

光标复位到00H

LCD显示DDRAM的内容全部写入 空白符的ASCII码20H

2.光标复位（00H）

光标复位到00H

LCD显示DDRAM的内容不变

3.光标和显示位置设置

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	I/D	S

I/D: 写入新数据后，光标移动方向

= 1, 右移

= 0, 左移

S: 写入新数据后，显示屏字符是否整体左移 或右移一个字符

= 1, 表示有效

= 0, 表示无效

4.显示开关控制

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	D	C	B

D(Display): 控制整体的显示开与关

= 1, 表示开显示屏

= 0, 表示关显示屏

C(Cursor): 控制光标的开与关

= 1, 表示有光标

= 0, 表示无光标

B(Blink): 控制光标是否闪烁

= 1, 闪烁

= 0, 不闪烁

5.光标或显示移位

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	S/C	R/L	X	X

SC	RL	功能
----	----	----

0	0	光标左移
---	---	------

0	1	光标右移
1	0	字符和光标都左移
1	1	字符和光标都右移

6. 功能设置命令

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	DL	N	F	X	X

DL: 控制数据长度

= 1, 代表数据长度为8位

= 0, 代表数据长度为4位

N: 控制显示行数

= 1, 两行都可以显示

= 0, 只有一行可以显示

F: 控制显示字符的大小

= 1, 一个字符大小为5 x 10的点阵字符

= 0, 一个字符大小为5 x 7的点阵字符

四、代码实现

1.LCD初始化

1. 写指令38H（指令6）

8位数据口，两行都显示，5*7点阵字符

2. 写指令0CH（指令4）

开启显示屏，无光标，光标不闪烁

3. 写指令06H（指令3）

写一个数据后，显示位置右移一位

4. 写指令01H（指令1）

显示清屏

```
void LcdInit()
{
    LcdwriteCmd(0x38);
    LcdwriteCmd(0x0C);
    LcdwriteCmd(0x06);
    LcdwriteCmd(0x01);
}
```

2.LCD写地址

本质就是指令8，不过我们再写的时候，最高位D7要求必须要为1，也就是实际是7位地址

要写入第一行的第column列

0x80 + column

要写入第二行的第column列

0x80 + 0x40 + column

```
void LcdGoToXY(unsigned char line, unsigned char column)
{
    if(line == 0)//第一行
    {
        LcdwriteCmd(0x80 + column);
    }
    if(line == 1)//第二行
```

```
    {  
        LcdWriteCmd(0x80 + 0x40 + column);  
    }  
}
```

3.LCD写数据

直接用一个指针，判断字符是否结束，然后写入数据就可以

```
void LcdPrintStr(unsigned char *str)  
{  
    while(*str != '\0')  
    {  
        LcdWriteData(*str++);  
    }  
}
```