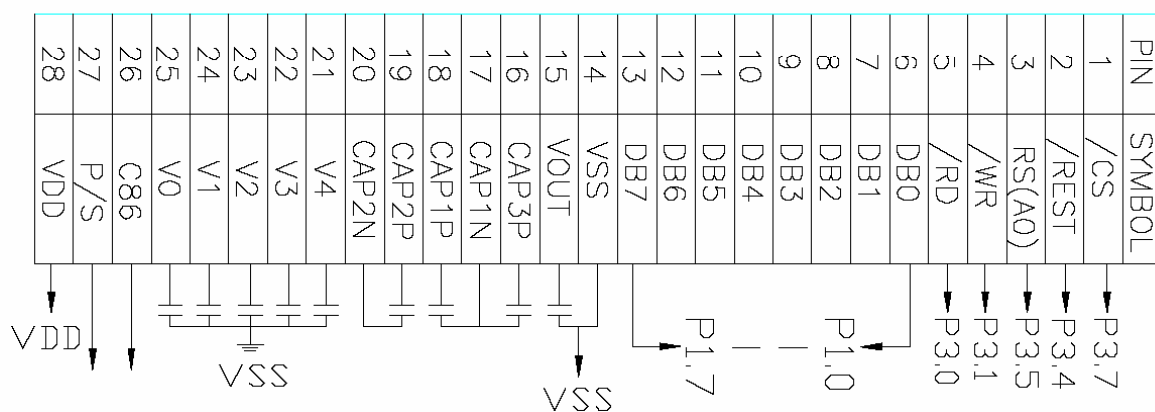


引脚序号	引脚名称	功能说明
1	/CS1	片选脚，底电平选通液晶屏工作
2	/RES	液晶屏复位脚.MCU 开机,先将/RES 拉低,等待 10us 后再拉高
3	A0(DC,RS)	数据/指令选择脚。高：写入的是数据；低：写入的是指令
4	/WR(RW)	8080条件下：数据锁存信号脚。 /RD=1：/WR由低到高锁存数据 /RD=0：/WR由低到高读取数据 6800条件下：为读、写选择信号脚。低：写数据；高：读数据
5	/RD(R/D,E)	8080条件下：为读、写选择信号脚。高：写数据；低：读数据 6800条件下：数据锁存信号脚， /WR=0：/RD由高到低锁存数据 /WR=1：/RD由高到低读取数据
6~13	D0-D7	8 位数据总线端。于串口工作时:D6 是时钟 SCL,D7 为数据 SDA
14	VSS	液晶工作电压地(负极)输入脚
15	VOOUT(VEE)	液晶内部产生驱动电压： 12.0V左右
16~20	CAP3P,CAP1N,CAP1P,CAP2P,CAP2N	液晶内部升压电路电容脚
21~25	V4,V3,V2,V1,V0	液晶偏压驱动，各脚与地之间接一个 1uF/25 无极性电容
26	C86	通讯时序选择。高：6800；低：8080
27	P/S	并、串口选择脚。高：并口；低：串口
28	VDD	液晶工作电压(正极)输入脚（3.3V）

1.5. 引脚电路接法



P/S	C86	
高	高	6800
高	低	8080
低	低/高	串口

电容均为1uF/25V

如果 MCU 速度很快(单周期指令时间小于 200ns)的，液晶显示屏于串口工作时，我们建议 DB0~DB5 要下拉 1K 电阻，并且在每次送数据之前保持时钟线为高电平状态，以确保显示屏工作可靠。

如果用 SPI 串口，/WR、/RD/、DB0~DB5，这些不需要用到的 PIN 需接 VSS。

指令说明:

顺序.	功能	代码											功能描述
		A0	/RD	/WR	D7	D6	D5	D4	D3	D2	D1	D0	
1	显示开、关	0	1	0	1	0	1	0	1	1	1	DON	DON=0,关闭显示 DON=1,打开显示
2	显示初始行设置	0	1	0	0	1	Display start address					设置显示 RAM 起始行地址	
3	设置页地址	0	1	0	1	0	1	1	Page address				设置显示 RAM 页面地址
4	设置列地址高 4 位	0	1	0	0	0	0	1	Col. Add				设置列地址高 4 位寄存器
	设置列地址低 4 位	0	1	0	0	0	0	0	Col. Add				设置列地址低 4 位寄存器
5	读显示屏状态	0	0	1	Status			0	0	0	0	0	读取显示屏当前状态
6	写入显示数据	1	1	0	Write Data							写入数据到显示 RAM	
7	读取显示数据	1	0	1	Read Data							从显示 RAM 中读取数据	
8	列镜象设置	0	1	0	1	0	1	0	0	0	0	ADC	设置显示 RAM 地址寄存器扫描方向 ADC = 0,正常. ADC = 1,地址左右颠倒
9	正常显示与反现设置	0	1	0	1	0	1	0	0	1	1	REV	REV = 0, 正常显示 REV = 1, 反显
10	全部显示开、关	0	1	0	1	0	1	0	0	1	0	EON	EON = 0, 全部显示功能关闭 EON = 1, 全部显示功能打开
11	显示屏偏压设置	0	1	0	1	0	1	0	0	0	1	BIAS	Bias = 0, 1/9 Bias 选择 IC 要驱动显示的行数 64 行 Bias = 1, 1/7 选择 IC 要驱动显示的行数 48 行以下
12	设置读、修改、写	0	1	0	1	1	1	0	0	0	0	0	
13	复位读、修改、写	0	1	0	1	1	1	0	1	1	1	0	清除读、修改、写模式
14	复位指令	0	1	0	1	1	1	0	0	0	1	0	复位指令存储器
15	页（行）镜象设置	0	1	0	1	1	0	0	SHL	*	*	*	设置行扫描方向 SHL = 0, 正常扫描 SHL = 1, 上下静象扫描 * 表示不确定
16	电源控制设置	0	1	0	0	0	1	0	1	VC	VR	VF	设置电源电路工作模式 VF : LCD 电压跟随器 VR : LCD 电压寄存器 VF : LCD 电压转换器 (1 = ON, 0 = OFF)
17	内部对比度可调电阻值设置(高)	0	1	0	0	0	1	0	0	Ratio			调整内部电阻(Rb/Ra), 实际是为调节对比度的高位
18	电压值设置模式	0	1	0	1	0	0	0	0	0	0	1	设置参考电压模式的指令
	内部对比度可调电阻值设置(低)	0	1	0	*	*	Electronic Control value					设置参考电压值, 实际为调节对比度的低位	
19	省电		-	-	-	-	-	-	-	-	-	-	复合指令 显示关 + 全部显示开
20	空操作	0	1	0	1	1	1	0	0	0	1	1	空操作指令

如果接口是并口，脚位定义如下：

```
sbit R_S= P3^5;//指令/数据选择信号
sbit RES= P3^4;//复位信号
sbit W_R= P3^1;//写信号
sbit R_D= P3^0;//读信号
sbit C_S= P3^7;//片选
bit busy_bit=0;
bit rev_bit=0;
uchar state=0;
```

如果接口是串口，脚位定义则如下：

```
sbit R_S= P3^5;//指令/数据选择信号
sbit RES= P3^4;//复位信号
sbit SCK= P1^6;//时钟
sbit SDA= P1^7;//数据
sbit C_S= P3^7;//片选
```

//根据接口方式选择以下三个时序中的一个

- (1) 8080 接口
- (2) 6800 接口
- (3) 串行接口 (SPI)

(1)

/******8080 时序******/

```
void WrateC(uchar cmd) //写入指令到 LCD
{
    C_S=0; //片选拉低使能显示屏接受数据
    R_S=0; //选择指令寄存器
    R_D=1; //准备写操作
    W_R=0; //执行写操作
    P1=cmd; //送数据到端口
    W_R=1; //锁存数据到显示屏中
    C_S=1; //片选拉高禁止显示屏工接受数据
}
```

void WrateD(uchar dat) //写入数据到 LCD

```
{
    C_S=0;
    R_S=1; //选择数据寄存器
    R_D=1;
    W_R=0;
    P1=dat;
    W_R=1;
    C_S=1;
}
```

```
bit CheckBusy() //读取显示屏状态
{
    P1=0xff;
    R_S=1;
    W_R=1;
    R_D=0;
    R_D=1;
    state=P1;
    busy_bit=(bit)(state>>7);
    return(busy_bit);
}
```

uchar ReadData() //读取显示屏显示 RAM 数据

```
{
    uchar dat;
    CS1=0;
    P1=0xff;
    while(CheckBusy());//显示屏忙则等待
    P1=0xFF;
    W_R=1; //执行读操作
    R_S=1; //读取的是数据
    R_D=0; //准备读
    R_D=1; //将显示屏 RAM 数据锁存到端口
    dat=P1; //读取显示屏数据端口数据给 dat
    CS1=1;
    return(dat); //返回所读取到的数据
}
/******8080 时序结束******/
```

```

(2)
/*****6800时序*****/
void WroteC(uchar cmd) //写入指令到 LCD
{
    C_S=0;        //片选拉低使能显示屏接受数据
    R_S=0;        //选择指令寄存器
    W_R=0;        //执行写操作
    R_D=1;        //准备写操作
    P1=cmd;       //送数据到端口
    R_D=0;        //锁存数据到显示屏中
    C_S=1;        //片选拉高禁止显示屏工接受数据
}

void WroteD(uchar dat) //写入数据到 LCD
{
    C_S=0;
    R_S=1;        //选择数据寄存器
    W_R=0;
    R_D=1;
    P1=dat;
    R_D=1;
    C_S=1;
}

bit CheckBusy() //读取显示屏状态
{
    P1=0xff;
    R_S=1;
    W_R=1;
    R_D=1;
    R_D=0;
    state=P1;
    busy_bit=(bit)(state>>7);
    return(busy_bit);
}

uchar ReadData() //读取显示屏显示 RAM 数据
{
    uchar dat;
    CS1=0;
    P1=0xff;
    while(CheckBusy()); //显示屏忙则等待
    P1=0xFF;
    W_R=1;
    R_S=1;
    R_D=1;
    R_D=0;
    dat=P1;

```

```

    CS1=1;
    return(dat);
}
/*****6800时序结束*****/

//MCU 系统时钟在 12MHz 条件下
//每个指令执行时间为 1us
void delay(uchar ms)
{
    uchar j;
    while(ms--)
        for(j=0;j<=20;j++); // 20us=0.02ms//
}

```

```

(3)
/*****串口时序*****/
void WrateC(Uchar cmd)
{
    unsigned char data i,j;
        SCK=1; //如果时钟很快, 请注意此处
        C_S=0;
        R_S=0;
    for(i=0;i<8;i++)
        {
            j=com;
            SCK=0;
            SDA=com&0x80;
            SCK=1;
            com=j<<1;
        }
        C_S=1;
        R_S=1;
}

void WrateD(Uchar dat)
{
    unsigned char data i,j;
        SCK=1; //如果时钟很快, 请注意此处
        C_S=0;
        R_S=1;

    for(i=0;i<8;i++)
        {
            j=dat;
            SCK=0;
            SDA=dat&0x80;
            SCK=1;
            dat=j<<1;
        }
        C_S=1;
        R_S=0;
}
/*****串口时序结束*****/

```

```

//LCD 初始化设定
void LcdInit()
{
    RES=0;
    delay(50);//最少 2um
    RES=1;
    delay(20);

    WrateC(0xe2);//软件复位
    WrateC(0xA2);//Bais set
    WrateC(0xA0);
    //ADC seg 镜像选择 0xa0 正常, 0xA1 左右镜像
    WrateC(0xc0);
    // com 镜像选择 0xc0 正常, 0xC8 上下镜像

    WrateC(0x40);//从首行开始显示

    WrateC(0x2c); //内部电源管理,
    WrateC(0x2e); //三条指令间隔时间 2um
    WrateC(0x2F);

    WrateC(0x81);
    WrateC(0x16);
    //电压调整寄存器低位 范围: 0x00-0x3f

    WrateC(0x25);
    //电压调整寄存器高位 范围: 0x21-0x27 , 值越大,
    显示效果越浓 (底影越浓)

    WrateC(0xAF);//显示开
    Clear();//清除屏幕
}

void Clear()
{
    unsigned char i,j,m=0;
    for(i=0;i<8;i++)
        {
            Setadd(0,i);
            for(j=0;j<128;j++)
                WrateD(0x00);
        }
}

```

```

/* LCD 数据指针位置程序*/
//带 入参 数: :xl, yl
//列地址范围: 0<xl<128
//页地址范围: 0<yl<8
void Setadd(uchar xl,uchar yl)
{
    uchar i;
    i=xl&0x0f;
    xl=(xl>>4)+0x10;
    WrateC(0xb0+yl);
    WrateC(xl);
    WrateC(i);
}

/*显示英文及 ASCII 码程序 */
//带 入参 数: col, pag, *ptr 字符串首地址
//列地址范围: 0<col<128
//页地址范围: 0<pag<8
void WriteEnglish(uchar col,uchar pag,uchar *ptr)
{
    uchar i,j;
    uchar c;
while(*ptr)
{
    c= *ptr; //获取字符代码值
    c -= 32; //减去空格之前的不可见字符
    for(i=0;i<2;i++)
        {
            Setadd(col,pag);
            If(rev_bit==0) //正常显示字符
            for(j=0;j<8;j++)
                WrateD(fontASCII8x16[c][i*8+j]);
            Else //反白显示字符
            for(j=0;j<8;j++)
                WrateD(~fontASCII8x16[c][i*8+j]);//反显
            pag++;
        }
        col+=8;
        if(col>0x7f)//超过显示屏列地址后,页面地址加一
            {
                col=0;
                pag+=2;
            }
        ptr++;
        pag-=2;
    }
}

```

```

void main()
{
    LcdInit();
    WriteEnglish(1,0,"HOT ICD");//显示标准英文字符
    while(1);
}

//取模软件: Zimo221
//取模方式: 8x16 字符, 纵向、字节到序、逗号保留
code unsigned char fontASCII8x16[96][16]=
{
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // - -
    0x00,0x00,0x38,0xFC,0xFC,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // -!-
    .
    . //中间省略, 标准 ASCII 码库
    .
    0x08,0x0C,0x04,0x0C,0x08,0x0C,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // ---
    0x80,0xC0,0x60,0x30,0x60,0xC0,0x80,0x00,0x07,0x07,0x04,0x04,0x04,0x07,0x07,0x00, // - -
};

```