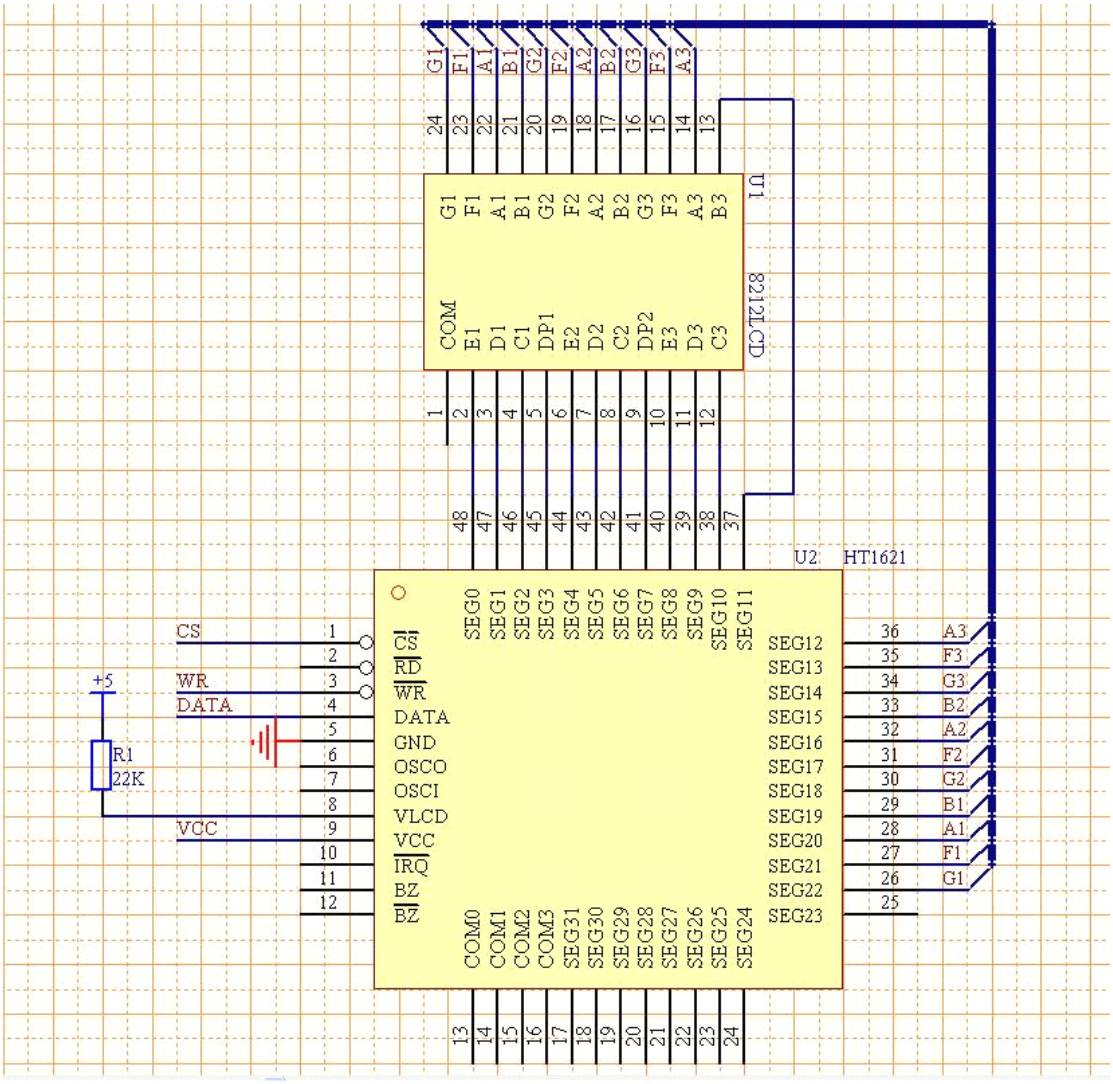
Ht1621 液晶显示详细驱动使用说明以及程序

1. 概述

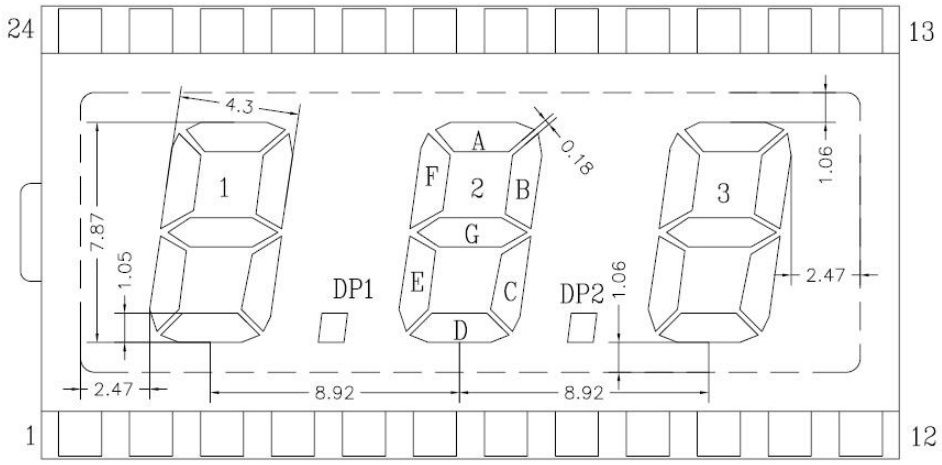
HT1621是128点内存映象和多功能的LCD驱动器，HT1621的软件配置特性使它适用于多种LCD应用场合，包括LCD模块和显示子系统。用于连接主控制器和HT1621的管脚只有4或5条，HT1621还有一个节电命令用于降低系统功耗。

在虎风所做的这个系统中ht1621用于驱动一个静态的LCD液晶显示器。液晶显示的方式分为静态显示和动态显示。静态与动态的区别在于静态显示是持续供电的，而动态显示是利用人的视觉停留效果，快速扫描数码管各个段，让人在视觉上感觉到数码管是同时显示的。

2. HT1621 接线原理图



3. 静态 LCD 结构图



PIN	1	2	3	4	5	6	7	8
SEG	COM	E1	D1	C1	DP1	E2	D2	C2
PIN	9	10	11	12	13	14	15	16
SEG	DP2	E3	D3	C3	B3	A3	F3	G3
PIN	17	18	19	20	21	22	23	24
SEG	B2	A2	F2	G2	B1	A1	F1	G1

4. 几个曾经纠结的概念

Time base：时基，即时间基准，可以用来输出，作为外部时钟的时间基准。

占空比：将所有公共电极（COM）各施加一次扫描电压的时间叫一帧,单位时间内扫描多少帧的频率叫帧频,将扫描公共电极（COM）选通的时间与帧周期之比叫占空比。通常占空比等于公共电极数N的倒数,即1/N。这就是说假如你要驱动4个液晶，就需要4个COM，那么你的占空比就要设定为1/4。

偏压比：指的是液晶的偏压系数，可以看看专业技术文章，偏压目的是克服交叉效应，通过把半选择点与非选择点的电压平均，适度提高非选择点的电压来抵消半选择点上的一部分电压,使半选择点上的电压下降,从而提高显示对比度;最终行半选择点和非选择点上的电压均为显示电压的1/a，1/a就称为偏压系数,也称为偏压。此方法称为1/a偏压的平均电压法，简称为1/a偏压法。

VLCD(LCD 驱动电压): LCD 的驱动电压为加在点亮部分的段电压与公共电压之差(峰-峰值)。

5. 关于 RAM 地址映射的概念

为了这个问题困扰了很久，虎风太愚钝啦……

Ht1621 有一个 32*4 的 LCD 驱动，映射到 32*4 的 RAM 地址。

命令名称	命 令 代 码	D/C	功 能 描 述	上电时复位缺省
READ	**110** a5 a4 a3 a2 a1 a0 d0 d1 d2 d3	D	读 RAM 数据	
WRITE	**101** a5 a4 a3 a2 a1 a0 d0 d1 d2 d3	D	写数据到 RAM	
READ-MODIFY-WRITE	**101** a5 a4 a3 a2 a1 a0 d0 d1 d2 d3	D	读和写数据	

上图中写命令 101 后面跟 6 位 RAM 地址，那么这个地址是如何确定的呢？其实说白了也很简单，RAM 地址就是 SEG 的序号。我们要点亮一段液晶管需要给他提供一个电平，而这个电平是由 SEG 管脚提供的，SEG 管脚电平的输出又取决于对应 RAM 地址中的值。

驱动一个 8 段数码管的顺序是 A,B,C,D,E,F,G,DP，我们认为前面 LCD 结构图中的数码管 3 为我们要显示的低位，那么连接原理图中 A3 的是 SEG12，我们就说此时的 RAM 地址为 0b001100，连接 B3 的是 SEG11,对应的 RAM 地址为 0b001011,依次类推，第一个数码管的所有地址为：

0b001100,0b001011,0b001010,0b001001,0b001000,0b001101,0b001110,0b000111//DP2;

其余地址类似，在此不再解释。

6. 程序

```
Unsigned                                                    char
LCD_ADD[]={0b00001100,0b00110100,0b00010100,0b00100100,0b00000100,0b0
0101100,0b00011100,0b00111000,0b00000010,0b00111100,0b00011000,0b0010
1000,0b00001000,0b00100010,0b00010010,0b00110000,0b00001010,0b0011001
0,0b00010000,0b00100000,0b00000000,0b00101010,0b00011010};


void HT1621_Dis_Char(unsigned char d,unsigned char d_loca)
{
    unsigned char j;
   unsigned char seg_cnt=8 ;
   if (d_loca==(HT1621_DATA_NUM-1))
     seg_cnt=7;
                        //送 3 位模式码 101 及 6 位首地址 000000 0b000000101
```

```
        for (j=0; j<seg_cnt; j++)  //送 cache_size 组数据
        { HT1621_PORT&=~_BV(HT1621_CS);
          HT1621_Send_Bits(0x05,3);                    //送写命令代码
101，高到低
          HT1621_Send_Bits(LCD_ADD[d_loca*8+j],6);   //送每段对应的地
址
          HT1621_Send_Bits(d&0x01,4);  //送数据，com0 对应的低位，低到
高
          d=d>>1;//右移位
          HT1621_PORT|=_BV(HT1621_CS);
        }
      }

    /*=========显示小数点=========
       输入参数：Poi_loca,为小数点的位置（0 是第一位，1 是第二位）
     */
    void HT1621_Pri_Point(unsigned char Poi_loca)
    {
       unsigned char seg_cnt=8;
        if (Poi_loca<2)
        { HT1621_PORT&=~_BV(HT1621_CS);
          HT1621_Send_Bits(0x05,3);                    //送写命令代码
101
          HT1621_Send_Bits(LCD_ADD[Poi_loca*seg_cnt+7],6);   //送点对
应的地址
          //送数据
          HT1621_Send_Bits(1,4);
          HT1621_PORT|=_BV(HT1621_CS);
        }
```

```
    }

    /*******************************************************
    =======显示 INT 型数据
    dat--待显示数据
    *******************************************************/
    void HT1621_Print_onlyInt(unsigned long dat)
    {
        unsigned char i;
        unsigned char data[HT1621_DATA_NUM];
        if (dat>=HT1621_DATA_max)
         dat=dat%HT1621_DATA_max;
       for(i=0;i<HT1621_DATA_NUM;i++)
data[i]=LCD_CODE[HT1621_NoPrint_Loc];//初始化为不显示
       if(dat<10)     data[0]  =  LCD_CODE[(unsigned  char)  (dat)];
//个位数
        else if(dat<100) {data[1] =LCD_CODE[(unsigned char) (dat/10)];
                          data[0] =LCD_CODE[(unsigned char) (dat%10)] ;}
//十位数
        else {
             data[2] =LCD_CODE[(unsigned char) (dat/100%10)];
             data[1] =LCD_CODE[(unsigned char) (dat/10%10)] ;
             data[0] =LCD_CODE[(unsigned char) (dat%10)];
            }  //百位数
       for(i=0;i<HT1621_DATA_NUM;i++)//从低位到高位，调用显示每个数的
函数 HT1621_Dis_Char
         HT1621_Dis_Char(data[i],i);
    }
```

```
/*************************************************************/
//打印 INT 型数据
//dat--待显示数据
//chk--显示的数据位数： 0--关闭  1~3 时，显示对应的数据；大于 3 时只显示 3 位；（从左到右）
//point_loc--显示的小数点的位置：0 以及大于 2 时：不显示小数点；1-显示第 1 个，2-显示第 2 个；（从左到右）
/*************************************************************/
void HT1621_Print_Int(unsigned long dat,unsigned char chk,unsigned char point_loc)
{
    unsigned char i;
    unsigned char data[HT1621_DATA_NUM];

   //chk 为 0，则屏幕不显示
   for(i=0;i<HT1621_DATA_NUM;i++)                    data[i]=LCD_CODE[HT1621_NoPrint_Loc];

    if(chk>0) {data[0] =  LCD_CODE[(unsigned char) (dat%10)];
               dat=dat/10;    }
    if(chk>1) {data[1] =  LCD_CODE[(unsigned char) (dat%10)];
              dat=dat/10;    }
   if(chk>2) {data[2] =  LCD_CODE[(unsigned char) (dat%10)];
               }
  //显示小数点
   if (point_loc==1)
       data[0]|=0x80;
   else if (point_loc==2)
      data[1]|=0x80;
```

```
  //从低位到高位，调用显示每个数的函数 HT1621_Dis_Char
    for(i=0;i<HT1621_DATA_NUM;i++)
     HT1621_Dis_Char(data[i],i);


}

/*********************************************************/
//送 d_count 位数据 d;从低位开始送
/*********************************************************/
void HT1621_Send_Bits(unsigned char d, unsigned char d_count)
{
   unsigned char j;

   for (j=0; j<d_count; ++j)
   {

        HT1621_PORT&=~_BV(HT1621_WR);        //wr 置低;
        HT1621_Delay(50);

       if (d & 0x01==1)                    //置数据位;
       {  HT1621_PORT|=_BV(HT1621_DATA);  //data 置 0
       }
       else
       {   HT1621_PORT&=~_BV(HT1621_DATA); //data 置 1
       }
       HT1621_PORT|=_BV(HT1621_WR);         //wr 置高;
        HT1621_Delay(50);
       d =(d>>1);  //右移位
```

```
    }
}

/*************************************************************/
//HT1621 初始化
/*************************************************************/
void HT1621_Init()
{

    DDRC  |= _BV(4)|_BV(5)|_BV(7);          //定义端口 PC4-PC5-PC7 输出
    HT1621_PORT&=~_BV(HT1621_CS);           //CS 端清零
     HT1621_Send_Bits(0x01,3);                 //送 3 位命令模式码 100
0x80=0b10000000
    HT1621_Send_Bits(0x18,9);               //系统时钟选用片内 RC
    HT1621_Send_Bits(0x80,9);               //打开系统振荡器
    HT1621_Send_Bits(0x94,9);               //1/2 偏置,2 个公共口 1000
0100
     HT1621_Send_Bits(0xc0,9);               //打开 LCD 偏置发生器
     HT1621_PORT|=_BV(HT1621_CS);            //CS 端置 1
    HT1621_NoPrint( );                      //液晶不显示
}
```

7. 结束

HT1621 作为液晶驱动芯片，功能十分灵活，而液晶在我们的学习项目中会经常用到，所以学好它的操作是十分必要的。

如果你看了这篇文章还不明白，就加虎风的 QQ 吧，279204362，我喜欢交朋友，让我们共同进步，嘎嘎！