

## HT48 & HT46 MCU 对 HT1621 的读写

### --使用 C 语言

文件编码：HA0103S

#### 介绍：

HT1621 是一款 128 个位元的 LCD 控制器件，内部 RAM 直接对应 LCD 的显示单元。相应的软件使它适用于包括 LCD 模块和显示子系统在内的多功能应用。主控制器与 HT1621 接口只需 4 到 5 根线。内置的省电模式极大的降低了功耗。本文介绍 HT48R05A-1 单片机来控制 HT1621，并介绍如何 turn ON 及 turn OFF LCD 所有位元。

#### 应用说明：

对于 HT1621，操作之前应该给它发送标志码，表明要求工作在何种状态。标志的定义如下表：

| 操作     | 状态 | 标志码 |
|--------|----|-----|
| 读      | 数据 | 110 |
| 写      | 数据 | 101 |
| 读-修改-写 | 数据 | 101 |
| 控制     | 命令 | 100 |

为了点亮 LCD，必须先给出两个控制指令：SYSTEM ENABLE 和 LCD ON。SYSTEM ENABLE 指令码是 :1000000001X( X 为 Don't care bit )，LCD ON 指令码是 :1000000011X ( X 为 Don't care bit )。操作结束可以用 SYSTEM DISABLE 来关闭 LCD。

由于是串行通信，数据应该先出现在 DATA INPUT 脚，然后给出一个写允许信号(WR)，输入一位数据，接着输入第二位...直到全部写入。

对 RAM 区不连续写数据过程如下：

|   |   |   |    |    |    |    |    |    |    |    |    |    |     |      |
|---|---|---|----|----|----|----|----|----|----|----|----|----|-----|------|
| 1 | 0 | 1 | A5 | A4 | A3 | A2 | A1 | A0 | D0 | D1 | D2 | D3 | 结束位 | 下一过程 |
|---|---|---|----|----|----|----|----|----|----|----|----|----|-----|------|

先发送标志码 101，表明下面要进行写操作。然后发送地址码 A5~A0，用 D0~D3 指定对应的位，就可以对 LCD 相应的位元操作了。读的过程除了标志位不同，其余类似。

连续读写时，给出起始地址，操作结束地址自动加一。

#### 例程：

本例介绍如何 turn ON 及 turn OFF LCD 全部位元。程序流程如下：

系统初始化 1621 启动 点亮 LCD 全部位元 关闭 HT1621

电路图：参照 HT1621 的规格书

```
#include <ht48r05a-1.h>
#define uchar unsigned char

#define BIAS 0x24
#define SYSEN 0x01
#define LCDOFF 0x02
#define LCDON 0x03

#define cs _pa0
#define wr _pa2
#define dat _pa3

void SendBit_1621(uchar data,uchar cnt) //data 的高 cnt 位写入 HT1621 , 高位在前
{
    uchar i;
    for(i =0; i <cnt; i ++ )
    {
        if((data&0x80)==0) dat=0;
        else dat=1;
        wr=0;
        _nop();
        wr=1;
        data<<=1;
    }
}

void SendDataBit_1621(uchar data,uchar cnt) //data 的低 cnt 位写入 HT1621 , 低位在前
{
    uchar i;
    for(i =0; i <cnt; i ++ )
    {
        if((data&0x01)==0) dat=0;
        else dat=1;
        wr=0;
        _nop();
        wr=1;
        data>>=1;
    }
}
```

```

void SendCmd(uchar command)
{
    cs=0;
    SendBit_1621(0x80,3);    //写入标志码"100"
    SendBit_1621(command,9); //写入 9 位数据,其中前 8 位为 command 命令,最后 1 位任意
    cs=1;
}

void Write_1621(uchar addr,uchar data)
{
    cs=0;
    SendBit_1621(0xa0,3);    //写入标志码"101"
    SendBit_1621(addr<<2,6); //写入 6 位 addr
    SendDataBit_1621(data,4); //写入 data 的低 4 位
    cs=1;
}

void WriteAll_1621(uchar addr,uchar *p,uchar cnt)
{
    uchar i;
    cs=0;
    SendBit_1621(0xa0,3);    //写入标志码"101"
    SendBit_1621(addr<<2,6); //写入 6 位 addr
    for(i =0; i <cnt; i ++,p++) //连续写入数据
    {
        SendDataBit_1621(*p,8);
    }
    cs=1;
}

void main()
{
    uchar a[10];
    a[0]=0x21;a[1]=0x43;a[2]=0x65;a[3]=0x87;a[4]=0xa9;
    _pac=0;                //将端口设为输出
    SendCmd(BIAS);         //设置偏压和占空比
    SendCmd(SYSEN);        //打开系统振荡器
    SendCmd(LCDON);        //打开 LCD 偏压发生器
    Write_1621(0x09,0x01); //在地址 0x09 处写入数据 0x01
    WriteAll_1621(0,a,5);  //在起始地址为 0 处连续写入 5 个字节数据
                           //a 为写入数据的起始地址,5 为写入的字节数
    SendCmd(LCDOFF);       //关闭 LCD 显示
}
    
```