

LM3121

液晶显示模块应用参考

深圳市拓普微科技开发有限公司

版本	描述	日期	编者
0.1	新版本	2006-01-22	淮俊霞

目 录

1 简介.....	3
2 应用.....	3
2.1 接口	3
2.2 显示内存映射图	3
2.3 指令表.....	4
2.4 流程图	4
参考程序.....	5

1 简介

本公司生产的 LM3121 为 202×32 全图形点阵的液晶显示模块,采用 2 片 AX6120 芯片、1 片 AX6121 芯片控制 LCD 显示屏上三个区域的显示,指令简单,易于操作。模块适配 8 位的 MCU 接口,外形尺寸为 231.0×53.6×12.5mm。

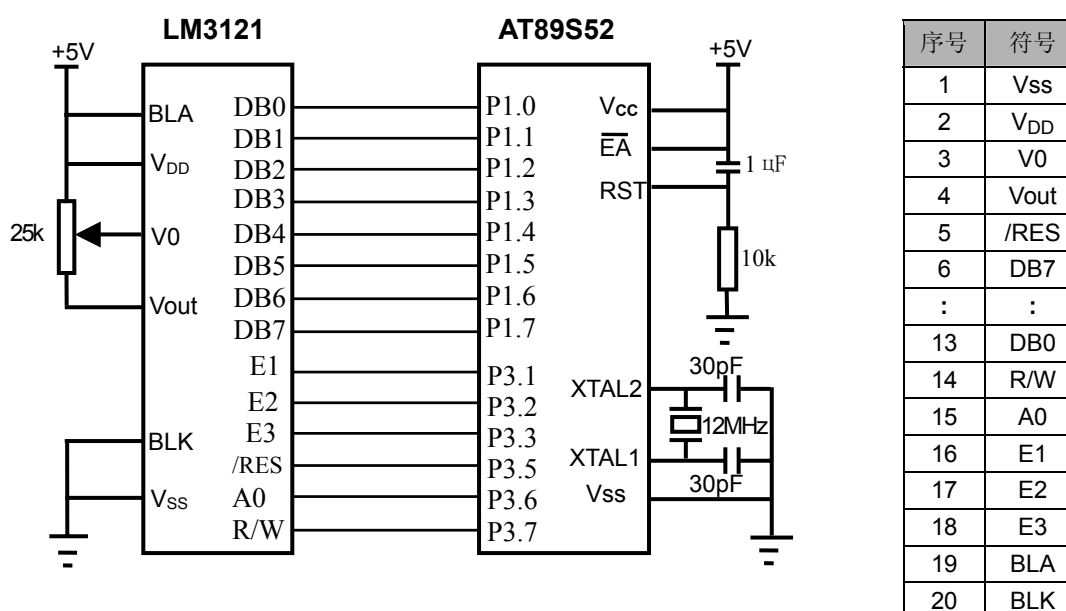
模块特点:

- ✧ 单电源 5V 供电
- ✧ 内建升压电路
- ✧ 视角方向为 6H
- ✧ FSTN, 负性, 全透
- ✧ 绿色 LED 背光

2 应用

2.1 接口

以 8 位并行通信为例,模块与单片机 AT89S52 的接口 (M6800 时序) 如下图所示。



2.2 显示内存映射图

页地址	数据	LCD 显示屏		
0	DB0 ⋮ DB7			
1	DB0 ⋮ DB7		202×32 像素	
2	DB0 ⋮ DB7			
3	DB0 ⋮ DB7			
列地址		00H→3CH	00H→4FH	00H→3CH
驱动控制		E1	E2	E3

- 注: 1) 显示起始行为 0
 2) ADC=0
 3) E1、E2、E3 分别控制着 LCD 屏幕的左、中、右三区显示数据的访问

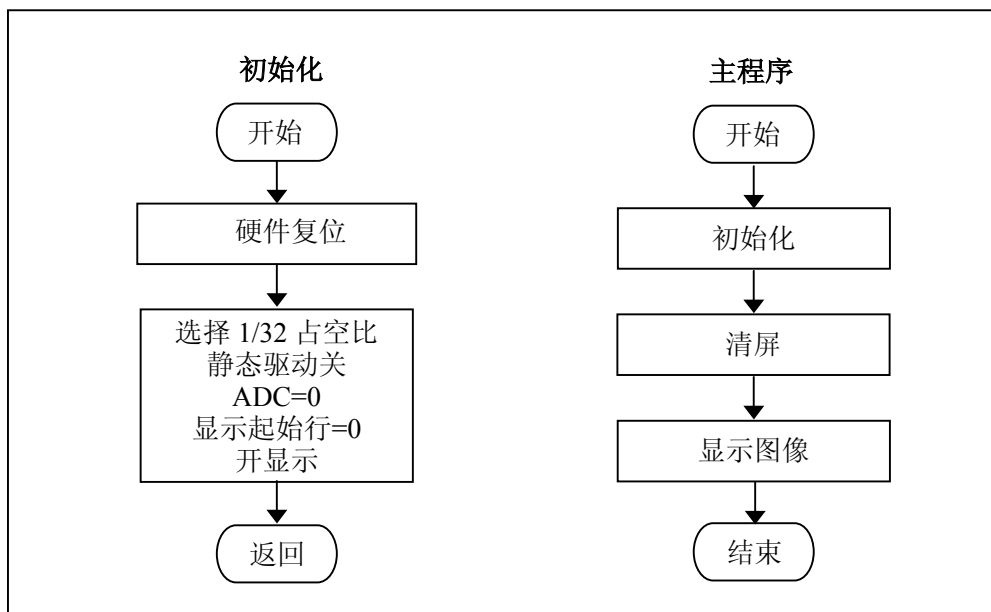
2.3 指令表

指令名称	控制信号		指令代码								功能
	R/W	A0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
显示开/关	0	0	1	0	1	0	1	1	1	0/1	0=关 (不影响显示存储器的内容) 1=开
显示起始行设定	0	0	1	1	0	L4	L3	L2	L1	L0	设置显示屏上最顶一行所对应的显示存储器的行地址
页地址设定	0	0	1	0	1	1	1	0	P1	P0	设置页地址寄存器
列地址设定	0	0	0	C6	C5	C4	C3	C2	C1	C0	设置列地址计数器
读状态字	1	0	Busy	ADC	ON/OFF	Reset	0	0	0	0	Busy=1 忙 Busy=0 准备好 ADC=1 列地址正向输出 ADC=0 列地址反向输出 ON/OFF=1 显示关 ON/OFF=0 显示开 Reset=1 复位状态 Reset=0 正常工作状态
写显示数据	0	1	数据								写显示数据到显示数据存储器
读显示数据	1	1	数据								从显示数据存储器中读取显示数据
ADC 选择	0	0	1	0	1	0	0	0	0	0/1	ADC=0 列地址正向输出 ADC=1 列地址反向输出
静态驱动开/关	0	0	1	0	1	0	0	1	0	0/1	0=正常驱动 1=静态驱动
选择占空比	0	0	1	0	1	0	1	0	0	0/1	0=1/16 占空比 1=1/32 占空比
启动改写方式	0	0	1	1	1	0	0	0	0	0	启动 Read-modify-write
结束改写方式	0	0	1	1	1	0	1	1	1	0	结束 Read-modify-write
复位	0	0	1	1	1	0	0	0	1	0	软件复位

注：1) 更详细的指令说明请参考 AX6120 手册

2) 指令和数据可以同时写入 3 个驱动/控制器，但读状态字或者读数据不能同时进行，否则出现数据冲突。

2.4 流程图




```

//-----
//写指令 (E2 控制)
//-----
void wrcmd2(uchar cmdcode)
{
    A0=0;
    R_W=0;
    E2=1;
    P1=cmdcode;
    _nop_();_nop_();_nop_();
    E2=0;
    _nop_();_nop_();_nop_();
}
//-----
//写数据 (E2 控制)
//-----
void wrdata2(uchar dispdata)
{
    A0=1;
    R_W=0;
    E2=0;
    P1=dispdata;
    E2=1;
    _nop_();_nop_();_nop_();
    E2=0;
}
//-----
//写指令 (E3 控制)
//-----
void wrcmd3(uchar cmdcode)
{
    A0=0;
    R_W=0;
    E3=1;
    P1=cmdcode;
    _nop_();_nop_();_nop_();
    E3=0;
    _nop_();_nop_();_nop_();
}
//-----
//写数据 (E3 控制)
//-----
void wrdata3(uchar dispdata)
{
    A0=1;
    R_W=0;
    E3=0;
    P1=dispdata;
    E3=1;
    _nop_();_nop_();_nop_();
    E3=0;
}
//-----
//清屏
//-----
void ClrDisp()
{
    uchar i,j;
    for(i=0;i<4;i++)
    {
        wrcmd1(0xb8+i);          //页地址
        wrcmd2(0xb8+i);
        wrcmd3(0xb8+i);

        wrcmd1(0x00);          //列地址
        wrcmd2(0x00);
        wrcmd3(0x00);
        for(j=0;j<80;j++)
        {
            wrdata1(0x00);
            wrdata2(0x00);
            wrdata3(0x00);
        }
    }
}

//-----
//显示图形
//-----
void Disp_Img(uchar code *Img)
{
    uchar i,j;
    uint n=0;

    for(j=0;j<4;j++)
    {
        wrcmd1(0xb8+j);          //左区页地址
        wrcmd1(0x00);          //左区列地址
        for(i=0;i<61;i++)
            wrdata1(Img[n++]);

        wrcmd2(0xb8+j);          //中区页地址
        wrcmd2(0x00);          //中区列地址
        for(i=61;i<141;i++)
            wrdata2(Img[n++]);

        wrcmd3(0xb8+j);          //右区页地址
        wrcmd3(0x00);          //右区列地址
        for(i=141;i<202;i++)
            wrdata3(Img[n++]);
    }
}
//-----
//程序初始化
//-----
initial()
{
    RES=1;
    delay(1);
    RES=0;          //硬件复位
    delay(1);
    RES=1;
    delay(1);

    wrcmd1(0xa9);          //1/32 占空比
    wrcmd2(0xa9);
    wrcmd3(0xa9);

    wrcmd1(0xa4);          //正常驱动
    wrcmd2(0xa4);
    wrcmd3(0xa4);

    wrcmd1(0xa0);          //ADC=0
    wrcmd2(0xa0);
    wrcmd3(0xa0);

    wrcmd1(0xc0);          //显示起始行
    wrcmd2(0xc0);
    wrcmd3(0xc0);

    wrcmd1(0xaf);          //开显示
    wrcmd2(0xaf);
    wrcmd3(0xaf);
}
//-----
//主程序
//-----
main()
{
    P1=0xff;
    E1=0;
    E2=0;
    E3=0;

    initial();
    ClrDisp();
    Disp_Img(BMP);
    while(1)
    {}
}
//end of program

```