LM3123

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	2006-01-19	杨军

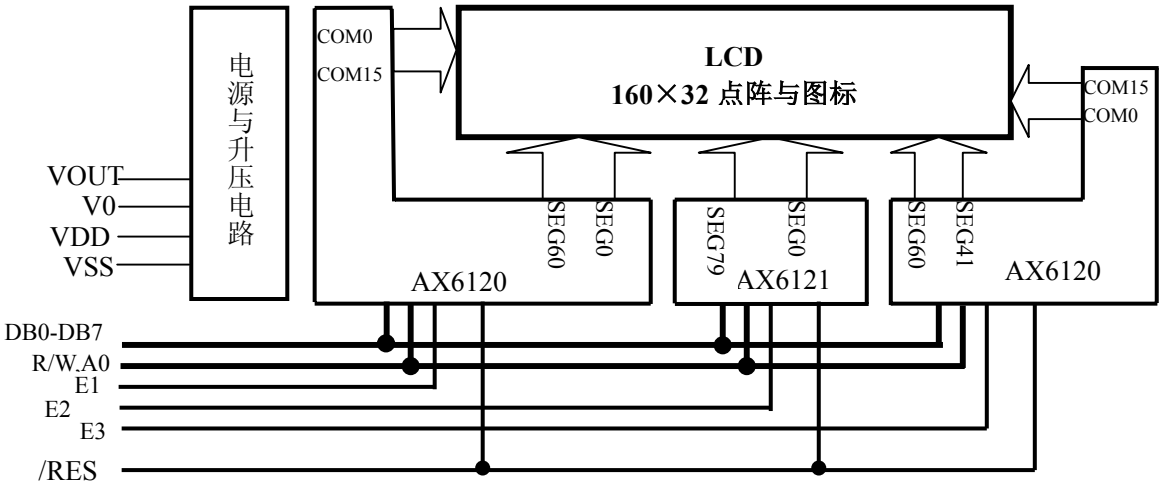
目 录

1、概述

LM3123 是本公司生产的全点阵图型液晶显示模块。采用的控制驱动器是 AX6120。其显示分辨率为 160×32 点。主要技术参数和显示特性:

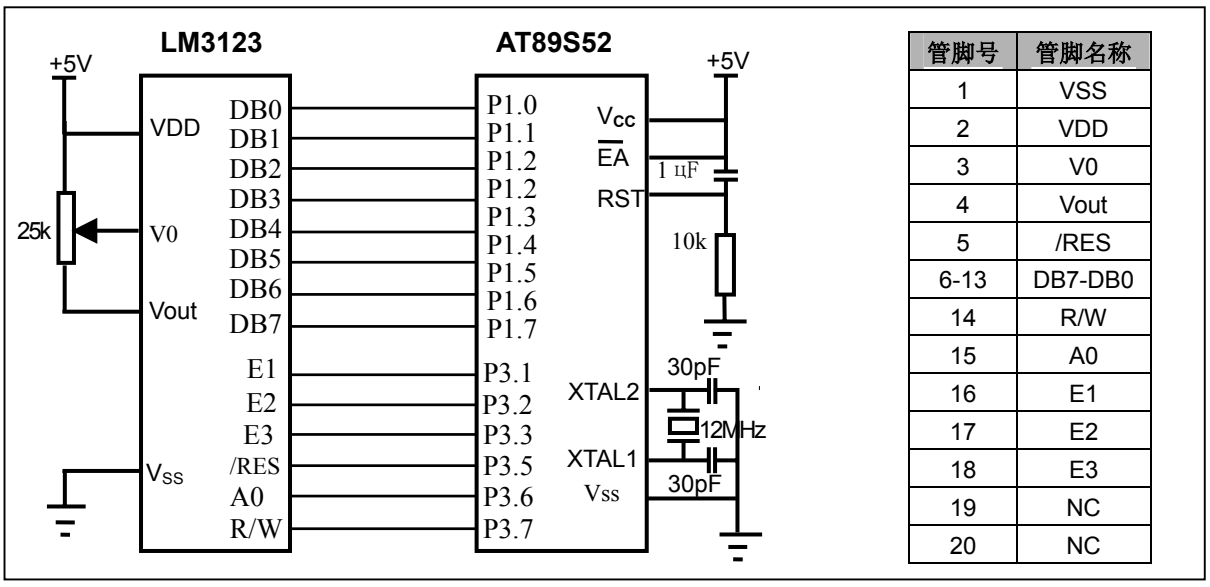
电源:单电源供电+5V，内有 LCD 负压电路
显示内容:160(列)×32(行)点和图标
显示颜色:黄绿底兰字
外形尺寸：99.6×31.5×10.7MAX
驱动方式:1/32 Duty,1/5 Bias

2、结构原理图



3、硬件接口

LM3123 与单片机的连接，我们采用的是 8 位并行接口，下面是 LM3123 的管脚定义和与单片机连接的硬件接口图。



管脚号	管脚名称
1	VSS
2	VDD
3	V0
4	Vout
5	/RES
6-13	DB7-DB0
14	R/W
15	A0
16	E1
17	E2
18	E3
19	NC
20	NC

4、软件特性（指令表）

指令名称	控制状态			指令代码								功能
	E	R/W	A0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
显示开/关	1	0	0	1	0	1	0	1	1	1	0/1	DB0=1（0AFH）：开显示 DB0=0（0AEH）：关显示
显示起始行	1	0	0	1	1	0	L4	L3	L2	L1	L0	0C0H~0DFH 对应显示存储器的 0~31 行。 定时修改起始行地址可产生上下滚动的效果
页地址设定	1	0	0	1	0	1	1	1	0	P1	P0	0B8H~0BBH将显示存储器分为4个页面0~3， 每页80个字节
列地址设定	1	0	0	0	C6	C5	C4	C3	C2	C1	C0	00H~4FH对应0~79个列单元地址
读取状态字	×	1	0	B	A	ON	R	0	0	0	0	B：BUSY=1 表示忙不接受计算机访问 BUSY=0 准备接受访问
												A：ADC 表示显示列地址与列驱动器关系 ADC=1：正向顺序 ADC=0：反向
												ON：ON/OFF 表示当前显示状态 ON/OFF=1显示关 ON/OFF=0开
												R：RESET复位状态 RESET=1正复位 RESET=0正工作
写数据	1	0	1	数据								将8位数据写入先确定的显示存储器地址，结束列地址加1
读数据	1	1	1	数据								从显示存储器读取数据
ADC选择	1	0	0	1	0	1	0	0	0	0	A	A=0（0A0H）：正向排列 A=1（0A1H）：反向排列
静态驱动	1	0	0	1	0	1	0	0	1	0	S	S=0（0A4H）：正常状态 S=1（0A5H）：休闲状态
选择占空比	1	0	0	1	0	1	0	1	0	0	DU	DU=0（0A8H）：为1/16占空比 DU=1（0A9H）：为1/32占空比
读修改写入	1	0	0	1	1	1	0	0	0	0	0	0E0H 启动改写方式
结束	1	0	0	1	1	1	0	1	1	1	0	0EEH 结束改写方式
复位	1	0	0	1	1	1	0	0	0	1	0	0E2H 软件复位

注：详细的指令说明请参考控制驱动器 AX6120

5、显示数据存储器的地址

页地址	数据			
0	D0 : D7	160 列×32 行		
1	D0 : D7			
2	D0 : D7			
3	D0 : D7			
列地址		14h→4Fh	00h→4Fh	13h→26h
驱动		E1	E2	E3

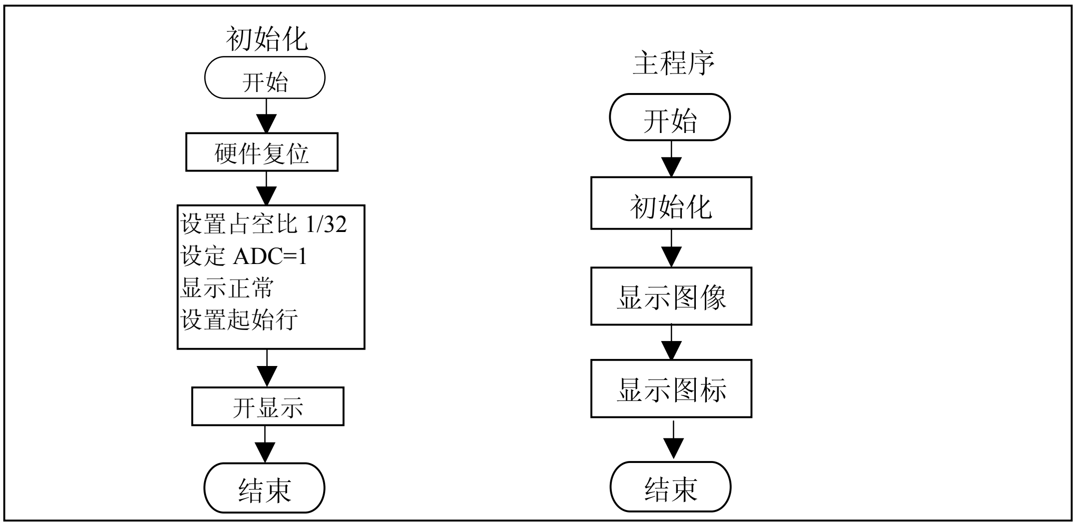
注：显示存储器首行为 0
ADC 必须等于 1

Icons 存储地址

图标	驱动	页地址	列地址	数据
	E1	**01h**	**13h**	D7
				D6
				D5
				D4

6、应用举例

6.1 程序流程图



6.2 参考程序

```
//--------------------------------------------
//演示结果为:“TOPWAY LM3123”
//          +图标
//--------------------------------------------

#include<reg52.h>
#include<intrins.h>
sbit    A0       =P3^6;
sbit    RW       =P3^7;
sbit    E1       =P3^1;
sbit    E2       =P3^2;
sbit    E3       =P3^3;
sbit    _REST    =P3^5;
#define uchar unsigned char
#define uint unsigned int
unsigned char CMDCODE,DAT;
unsigned char code BMP[]=
{
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x01,0x01,0x7F,0x01,0x01,0x00,0x3E,0x41,
0x41,0x41,0x3E,0x00,0x7F,0x09,0x09,0x09,0x06,0x00,
0x3F,0x40,0x38,0x40,0x3F,0x00,0x7C,0x0A,0x09,0x0A,
0x7C,0x00,0x03,0x04,0x78,0x04,0x03,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x7F,0x40,0x40,0x40,0x40,0x00,
0x7F,0x02,0x0C,0x02,0x7F,0x00,0x22,0x41,0x49,0x49,
0x36,0x00,0x00,0x42,0x7F,0x40,0x00,0x00,0x42,0x61,
0x51,0x49,0x46,0x00,0x22,0x41,0x49,0x49,0x36,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x26,0x49,
0x49,0x49,0x32,0x00,0x01,0x01,0x7F,0x01,0x01,0x00,
0x7F,0x04,0x08,0x10,0x7F,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x7F,0x40,0x40,0x40,0x40,0x00,0x3E,0x41,
0x41,0x41,0x22,0x00,0x7F,0x41,0x41,0x22,0x1C,0x00,
0x00,0x40,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x42,0x7F,0x40,0x00,0x00,0x3E,0x49,
0x49,0x49,0x30,0x00,0x3E,0x51,0x49,0x45,0x3E,0x00,
0x44,0x28,0x10,0x28,0x44,0x00,0x22,0x41,0x49,0x49,
0x36,0x00,0x42,0x61,0x51,0x49,0x46,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x7C,0x14,0x14,0x14,0x08,0x00,
0x00,0x44,0x7D,0x40,0x00,0x00,0x44,0x28,0x10,0x28,
0x44,0x00,0x38,0x54,0x54,0x54,0x18,0x00,0x00,0x41,
0x7F,0x40,0x00,0x00,0x48,0x54,0x54,0x54,0x20,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x42,0x7F,0x40,0x00,0x00,0x42,0x61,0x51,0x49,
0x46,0x00,0x22,0x41,0x49,0x49,0x36,0x00,0x18,0x14,
0x12,0x7F,0x10,0x00,0x27,0x45,0x45,0x45,0x39,0x00,
0x3E,0x49,0x49,0x49,0x30,0x00,0x01,0x01,0x79,0x05,
0x03,0x00,0x36,0x49,0x49,0x49,0x36,0x00,0x26,0x49,
0x49,0x49,0x3E,0x00,0x3E,0x51,0x49,0x45,0x3E,0x00,
0x00,0x42,0x7F,0x40,0x00,0x00,0x42,0x61,0x51,0x49,
0x46,0x00,0x22,0x41,0x49,0x49,0x36,0x00,0x18,0x14,
0x12,0x7F,0x10,0x00,0x27,0x45,0x45,0x45,0x39,0x00,
0x3E,0x49,0x49,0x49,0x30,0x00,0x01,0x01,0x79,0x05,
0x03,0x00,0x36,0x49,0x49,0x49,0x36,0x00,0x26,0x49,
0x49,0x49,0x3E,0x00,0x3E,0x51,0x49,0x45,0x3E,0x00,
0x00,0x42,0x7F,0x40,0x00,0x00,0x42,0x61,0x51,0x49,
0x46,0x00,0x22,0x41,0x49,0x49,0x36,0x00,0x18,0x14,
0x12,0x7F,0x10,0x00,0x27,0x45,0x45,0x45,0x39,0x00,
0x3E,0x49,0x49,0x49,0x30,0x00,0x01,0x01,0x79,0x05
};

//------------------------------------------
//延时程序
//------------------------------------------
void delay(uint t)
{
    uint i;
    uint j;
    for(j=0;j<t;j++)
        for(i=0;i<10;i++)
            _nop_();
}

//------------------------------------------
//写指令数据
//------------------------------------------
void wrcmd1(uchar CMDCODE)            //E1 写指令
{
    A0=0;RW=0;delay(5);
    E1=1;
    P1=CMDCODE; delay(5);
    E1=0; E1=1;
}
void wrdata1(uchar DAT)               // E1 写数据
{
    A0=1;RW=0;delay(5);
    E1=1;
    P1=DAT; delay(5);
    E1=0; E1=1;
}

void wrcmd2(uchar CMDCODE)            //E2 写指令
{
    A0=0;RW=0;delay(5);
    E2=1;
    P1=CMDCODE; delay(5);
    E2=0;E2=1;
}
void wrdata2(uchar DAT)               //E2 写数据
{
    A0=1;RW=0;delay(5);
    E2=1;
    P1=DAT; delay(5);
    E2=0; E2=1;
}

void wrcmd3(uchar CMDCODE)            //E3 写指令
{
    A0=0;RW=0;delay(5);
    E3=1;
    P1=CMDCODE; delay(5);
    E3=0; E3=1;
}
void wrdata3(uchar DAT)               //E3 写数据
{
    A0=1;RW=0;delay(5);
    E3=1;
    P1=DAT; delay(5);
    E3=0; E3=1;
}

//------------------------------------------
//程序初始化
//------------------------------------------
init_1520()
{
wrcmd1(0xa4);wrcmd2(0xa4);wrcmd3(0xa4);//正常显示
wrcmd1(0xa9);wrcmd2(0xa9);wrcmd3(0xa9);//占空比1/32
wrcmd1(0xa1);wrcmd2(0xa1);wrcmd3(0xa1);//选择 ADC
wrcmd1(0xc0);wrcmd2(0xc0);wrcmd3(0xc0);// 起 始 行
wrcmd1(0xaf);wrcmd2(0xaf);wrcmd3(0xaf);//显示开
}

//------------------------------------------
//清屏
//------------------------------------------
void ClrDisp()
{
    unsigned char i,j;
    for(i=0;i<4;i++)
    {
        wrcmd1(0xb8+i);
        wrcmd2(0xb8+i);
        wrcmd3(0xb8+i);

        wrcmd1(0x00);
        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)
{
    unsigned char i,j,k;
    unsigned int n=0;
    for(k=0;k<4;k++)
    {
        wrcmd1(0xb8+k);
        for(i=0;i<60;i++)
        {
            wrcmd1(0x14+i);
            wrdata1(Img[n++]);
        }
        wrcmd2(0xb8+k);
        for(i=0;i<80;i++)
        {
            wrcmd2(i);
            wrdata2(Img[n++]);
        }
        wrcmd3(0xb8+k);
        for(i=0;i<20;i++)
        {
            wrcmd3(0x13+i);
            wrdata3(Img[n++]);
        }
    }
}

//------------------------------------------
//主程序
//------------------------------------------
main()
{
    ClrDisp();
    delay(100);
    init_1520();
    delay(10);
    while(1)
    {
        Disp_Img(BMP);
        delay(1000);
        wrcmd1(0xb9);        //显示图标
        wrcmd1(0x13);
        wrdata1(0xf0);
    }
}                         //end of program
```