

LM6800

液晶显示模块应用参考

深圳市拓普微科技发展有限公司

版本	描述	日期	编者
0.1	新版本	2006-01-19	淮俊霞



目 录

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

1 简介

本公司生产的 LM6800 系列产品为 256×64 全图形点阵的液晶显示模块,采用 4 片 S6B0108 芯片控制 LCD 显示屏上四个区域的显示,指令简单,易于操作。模块适配 8 位的 MCU 接口,外形尺寸为 137.0×39.6×12.8mm,适用于 5.25" 驱动器槽的位置,视角方向为 6H。

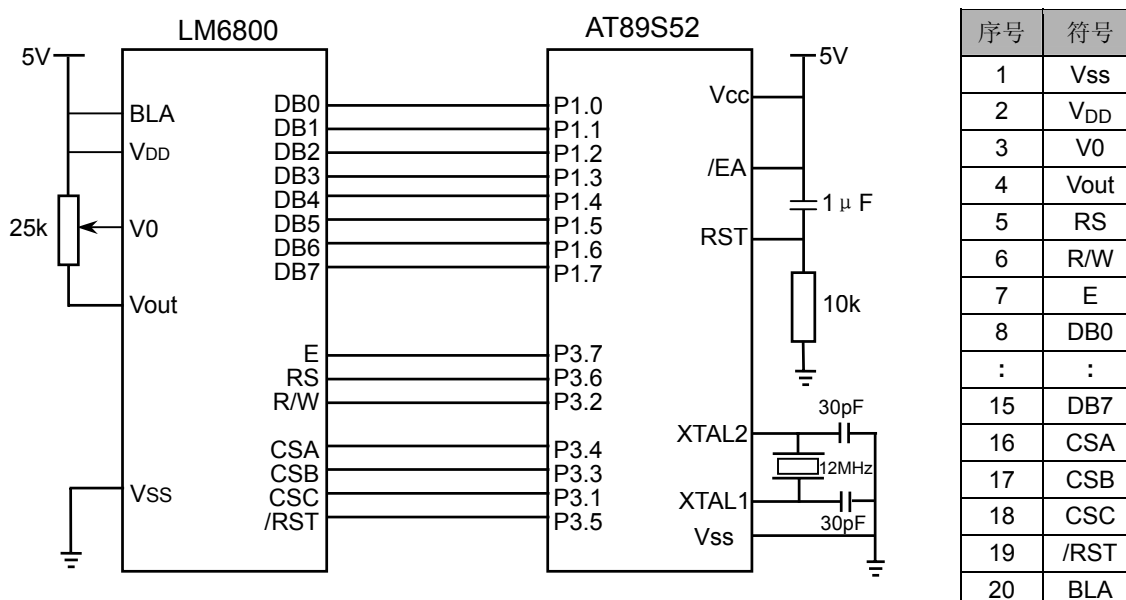
模块特点:

- ✧ 单电源 5V 供电
- ✧ 内建升压电路
- ✧ 多种 LCD 屏供选择
- ✧ 白色 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。



片选信号组合如下:

CSC	CSB	CSA	功能
0	0	0	选中芯片 1, 控制显示屏上 1—64 列的显示区域
0	0	1	选中芯片 2, 控制显示屏上 65—128 列的显示区域
0	1	0	选中芯片 3, 控制显示屏上 129—192 列的显示区域
0	1	1	选中芯片 4, 控制显示屏上 193—256 列的显示区域
1	×	×	禁止访问芯片

2.2 指令表

指令	代码										功能	
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
显示开关	0	0	0	0	1	1	1	1	1	1	on/off	显示开关的控制位。关显示状态下，RAM中的显示数据不受影响 On/off=1, 显示开 On/off=0, 显示关
列(Y)地址设置	0	0	0	1	Y 地址(0-63)						设置 Y 地址寄存器的内容	
页(X)地址设置	0	0	1	0	1	1	1	X 地址(0-7)			设置 X 地址寄存器的内容	
显示起始行设置	0	0	1	1	Z 地址(0-63)						设置显示屏上最顶一行所对应的显示存储器的行地址	
读取状态字	0	1	Busy	0	on/off	Reset	0	0	0	0		读状态 Busy=0,控制器准备好 Busy=1,控制器忙 On/off=0, 显示开 On/off=1, 显示关 Reset=0,正常运行 Reset=1,复位
写显示数据	1	0	数据									写显示数据到显示数据存储器 写操作完成后，Y 地址计数器自动加 1
读显示数据	1	1	数据									从显示数据存储器中读取显示数据

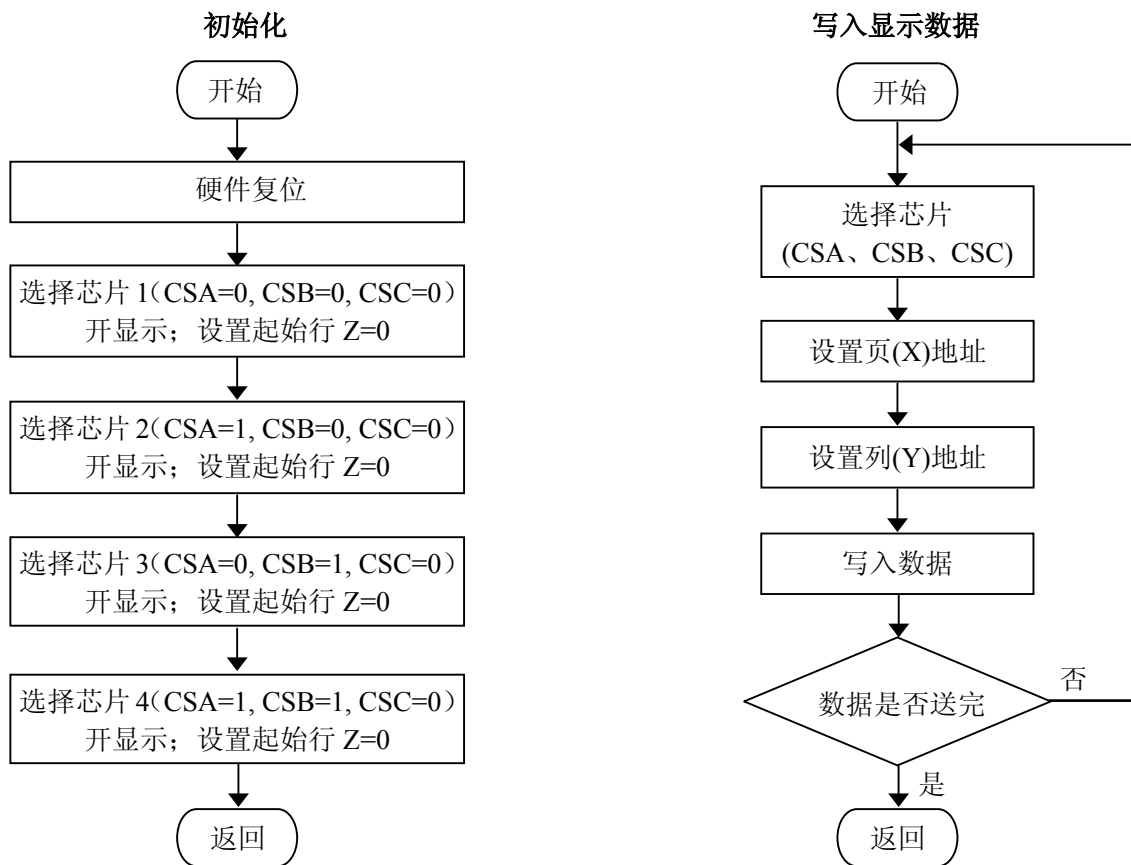
注：更详细的指令说明请参考三星的 S6B0108 手册

2.3 显示内存映射图

页(X)地址	数据	LCD 显示屏			
0	DB0 ⋮ DB7				
1	DB0 ⋮ DB7				
2	DB0 ⋮ DB7				
3	DB0 ⋮ DB7				
4	DB0 ⋮ DB7				
5	DB0 ⋮ DB7				
6	DB0 ⋮ DB7				
7	DB0 ⋮ DB7				
列(Y)地址		00H...→3FH	00H...→3FH	00H...→3FH	00H...→3FH
控制器芯片		芯片 1 (CSA=0,CSB=0,CSC=0)	芯片 2 (CSA=1,CSB=0,CSC=0)	芯片 3 (CSA=0,CSB=1,CSC=0)	芯片 4 (CSA=1,CSB=1,CSC=0)

- 注：1) 显示起始行(Z 地址)为 0
2) 显示数据分别存储在四个控制器里

2.4 流程图



参考程序

```

//-----
//本程序的演示结果为:
// "TOPWAY LM6800 LCD Module"
//-----
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
#define LCD_BUS P1 //MCU P1<-----> LCM

sbit CSA=P3^4;
sbit CSB=P3^3;
sbit CSC=P3^1;
sbit _RST=P3^5; //复位信号
sbit E=P3^7; //使能信号
sbit RS=P3^6; //数据/指令选择
sbit R_W=P3^2; //读写信号

uchar 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,
0x00,0x00,0x00,0x08,0x08,0x08,0xF8,0x08,
0x08,0x18,0x00,0xE0,0x10,0x08,0x08,0x08,
0x10,0xE0,0x00,0x08,0xF8,0x08,0x08,0x08,
0x08,0xF0,0x00,0xF8,0x08,0x00,0xF8,0x00,
0x08,0xF8,0x00,0x00,0x00,0xC0,0x38,0xE0,
0x00,0x00,0x00,0x08,0x38,0xC8,0x00,0xC8,
0x38,0x08,0x00,0x00,0x00,0x00,0x00,

```

```

0x00,0x00,0x00,0x08,0xF8,0x08,0x00,0x00,
0x00,0x00,0x00,0x08,0xF8,0xF8,0x00,0xF8,
0xF8,0x08,0x00,0x00,0xE0,0x10,0x88,0x88,
0x18,0x00,0x00,0x00,0x70,0x88,0x08,0x08,
0x88,0x70,0x00,0x00,0xE0,0x10,0x08,0x08,
0x10,0xE0,0x00,0x00,0xE0,0x10,0x08,0x08,
0x10,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x08,0xF8,0x08,0x00,0x00,
0x00,0x00,0x00,0xC0,0x30,0x08,0x08,0x08,
0x08,0x38,0x00,0x08,0xF8,0x08,0x08,0x08,
0x10,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x08,0xF8,0xF8,0x00,0xF8,
0xF8,0x08,0x00,0x00,0x00,0x80,0x80,
0x80,0x00,0x00,0x00,0x00,0x00,0x80,0x80,
0x88,0xF8,0x00,0x80,0x80,0x00,0x00,0x00,
0x80,0x80,0x00,0x00,0x08,0x08,0xF8,0x00,
0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,
0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
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,
0x00,0x00,0x00,0x00,0x20,0x3F,0x20,
0x00,0x00,0x00,0x0F,0x10,0x20,0x20,0x20,
0x10,0x0F,0x00,0x20,0x3F,0x21,0x01,0x01,
0x01,0x00,0x00,0x03,0x3C,0x07,0x00,0x07,
0x3C,0x03,0x00,0x20,0x3C,0x23,0x02,0x02,
0x27,0x38,0x20,0x00,0x00,0x20,0x3F,0x20,

```

```

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x3F,0x20,0x20,0x20,
0x20,0x30,0x00,0x20,0x3F,0x00,0x3F,0x00,
0x3F,0x20,0x00,0x00,0x0F,0x11,0x20,0x20,
0x11,0x0E,0x00,0x00,0x1C,0x22,0x21,0x21,
0x22,0x1C,0x00,0x00,0x0F,0x10,0x20,0x20,
0x10,0x0F,0x00,0x00,0x0F,0x10,0x20,0x20,
0x10,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x3F,0x20,0x20,0x20,
0x20,0x30,0x00,0x07,0x18,0x20,0x20,0x20,
0x10,0x08,0x00,0x20,0x3F,0x00,0x20,0x20,
0x10,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x3F,0x00,0x3F,0x00,
0x3F,0x20,0x00,0x00,0x1F,0x20,0x20,0x20,
0x20,0x1F,0x00,0x00,0x0F,0x11,0x20,0x20,
0x10,0x3F,0x20,0x00,0x1F,0x20,0x20,0x20,
0x10,0x3F,0x20,0x00,0x20,0x20,0x3F,0x20,
0x20,0x00,0x00,0x00,0x1F,0x22,0x22,0x22,
0x22,0x13,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};

//-----
//延时子程序
//-----
void Delay(uint t)
{
    uchar i;
    uint j;
    for(j=0;j<t;j++)
        for(i=0;i<109;i++)
            _nop_();
}
//-----
//写命令到 LCD
//-----
void CmdWrite(uchar cmdcode)
{
    RS=0;
    R_W=0;
    LCD_BUS=cmdcode;
    E=1;
    E=0;
}
//-----
//写数据到 LCD
//-----
void DataWrite(uchar dispdata)
{
    RS=1;
    R_W=0;
    LCD_BUS=dispdata;
    E=1;
    E=0;
}
//-----
//芯片选择
//-----
void ChipSelect(uchar number)
{
    switch(number)
    {
        case 0: CSA=0;CSB=0;CSC=0;break;
        case 1: CSA=1;CSB=0;CSC=0;break;
        case 2: CSA=0;CSB=1;CSC=0;break;
        case 3: CSA=1;CSB=1;CSC=0;break;
        default:break;
    }
}

//-----
//清屏
//-----
void ClearScr()
{
    uchar i,j,k;
    for(k=0;k<4;k++)
    {
        ChipSelect(k); //芯片选择
        for(j=0;j<8;j++)
        {
            CmdWrite(0xb8+j); //页地址设置
            CmdWrite(0x40); //列地址设置
            for(i=0;i<64;i++)
                DataWrite(0x00);
        }
    }
}
//-----
//显示一幅图片(256×16)
//-----
void Disp_Bmp(uchar code *img)
{
    uchar i,j,k;
    uint n;
    for(k=0;k<4;k++)
    {
        ChipSelect(k); //芯片选择
        for(j=3;j<5;j++)
        {
            CmdWrite(0xb8+j); //页地址设置
            CmdWrite(0x40); //列地址设置
            for(i=0;i<64;i++)
            {
                n=i+256*(j-3)+64*k;
                DataWrite(img[n]);
            }
        }
    }
}
//-----
//初始化
//-----
void LCD_Initial()
{
    uchar i;
    RST=0;
    Delay(100);
    RST=1;
    Delay(10);
    for(i=0;i<4;i++)
    {
        ChipSelect(i); //芯片选择
        CmdWrite(0x3f); //开显示
        CmdWrite(0xc0); //首行地址设置
    }
}
//-----
//主程序
//-----
main()
{
    CSC=1;
    LCD_BUS=0xff;
    LCD_Initial();
    ClearScr();
    Disp_Bmp(BMP);
    while(1)
    {}
}
//end of program

```