

# LM12864L

## 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	2006-2-9	周侃侃



## 目 录

<b>1 简介</b> .....	3
<b>2 应用</b> .....	3
<b>2.1 接口</b> .....	3
<b>2.2 操作指令说明</b> .....	4
<b>2.3 显示屏幕</b> .....	4
<b>2.4 软件流程图</b> .....	5
<b>2.5 应用举例</b> .....	6

## 1 简介

本公司产品 LM12864L 为  $128 \times 64$  的全点阵图形液晶显示模块 LM12864 中的一个系列。该系列模块内置三星公司 S6B0108 控制器（两片）和 S6B0107 驱动器，能完成图形方式的各种内容的显示。该模块硬件电路简单，功能强大，性能稳定，是性价比极高的一款液晶显示模块。

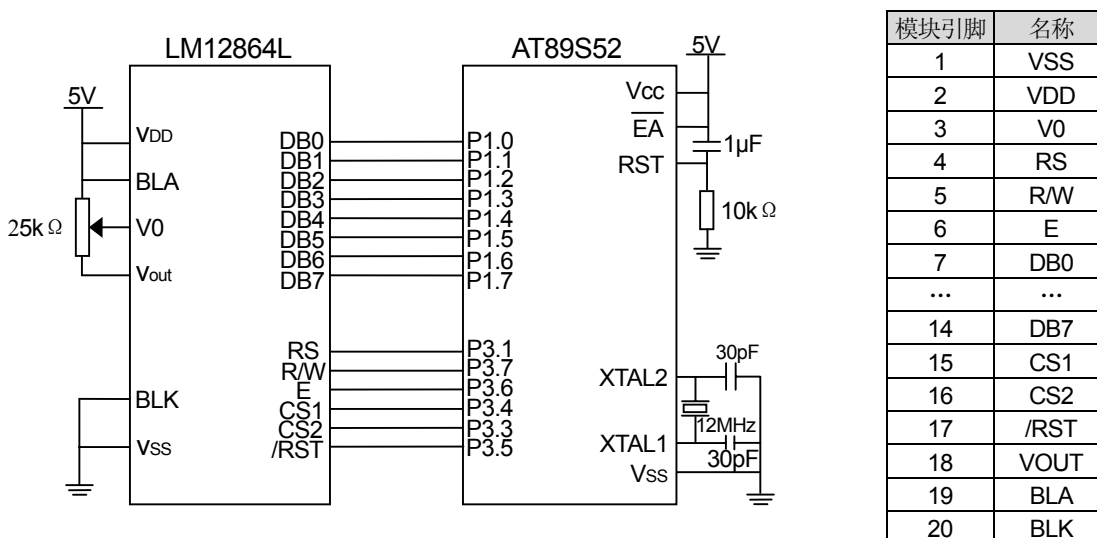
该模块的特点有：

- \* STN-LCD
- \* 内置 VLCD 负压电路
- \* Sumsang S6B0107/S6B0108 液晶驱动/控制器
- \* 8 位单片机接口

## 2 应用

### 2.1 接口

模块与单片机 AT89S52 的接口，采用 I/O 方式的 8 位并行通信，如下图所示。



## 2.2 操作指令说明

指令名称	控制信号		控制代码								功能	备注	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
显示开关	0	0	0	0	1	1	1	1	1	1/0	控制屏幕开关。不影响显示内存的状态和显示数据 0=关, 1=开	*2	
选择列(Y)地址	0	0	0	1	Y 地址 (0-63)					设置列地址送入 Y 地址计数器		*2	
选择页(X)地址	0	0	1	0	1	1	1	X 地址 (0-7)			设置页地址送入 X 地址存储器		*2
设置显示起始行(Z地址)	0	0	1	1	Z 地址 (0-63)					设置屏幕上最顶一行所对应的显示存储器的行地址		*2	
读状态字	0	1	忙位	0	开关	复位	0	0	0	0	忙位=0, 准备好; 忙位=1, 忙 开关=0, 显示开; 开关=1, 显示关 复位=0, 正常工作状态; 复位=1, 复位	*1	
写数据	1	0	写数据								将显示数据写入显示数据存储器, Y 地址计数器自动加 1		*2
读数据	1	1	读数据								从显示数据存储器读取显示数据		*1

备注:

- \*1. 在同一时间只有一片控制器的数据能被读取, 即只有下面的设置才是有效的操作:  
CS1=1, CS2=0 选择控制器 1 ; CS1=0, CS2=1 选择控制器 2
- \*2. 指令能被同时写入两片控制器。
- \*3. 关于显示控制功能的细节, 请查阅 Samsung S6B0108 手册。

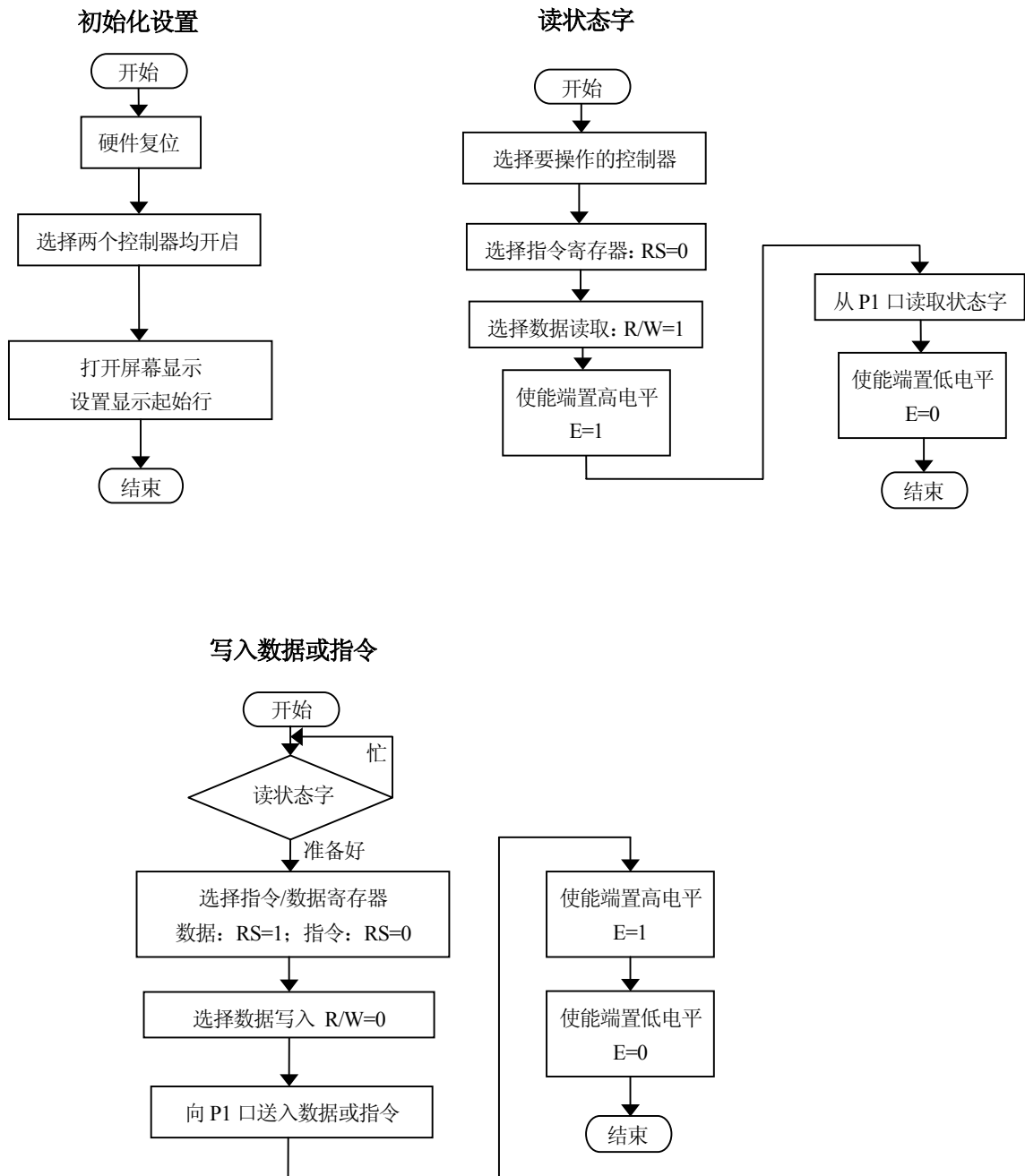
## 2.3 显示屏幕

页(X)地址	数据	屏幕显示区域	
0	D0 : D7	<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;">                     128 × 64 像素                 </div>	
1	D0 : D7		
2	D0 : D7		
3	D0 : D7		
4	D0 : D7		
5	D0 : D7		
6	D0 : D7		
7	D0 : D7		
列(Y)地址		00H → 3FH	00H → 3FH
控制器选择		CS1 = 1 , CS2 = 0	CS1 = 0 , CS2 = 1

备注:

- 1) 显示起始行(Z地址)=0
- 2) 显示数据被分别放在两个控制器中。
- 3) 左半屏的显示数据在 CS1=1 时写入; 右半屏的显示数据在 CS2=1 时写入。

## 2.4 软件流程图



\* 注: 在本手册的参考程序中, 为了简化程序, 用延时等待的方式替代了读状态字的过程。





```
RS      = 1;
RW      = 0;
lcd_bus = ddata;          //送入数据
E       = 1;
E       = 0;
}

//-----
//写指令
//-----
void SdCmd(uchar command)
{
    delay(0);              //延时等待
    RS      = 0;
    RW      = 0;
    lcd_bus = command;    //送入指令
    E       = 1;
    E       = 0;
}

//-----
//初始化
//-----
void Init()
{
    _RST    = 0;
    delay(150);
    _RST    = 1;          //硬件复位
    delay(100);
    CS1=1;
    CS2=1;                //控制器全开
    SdCmd(0x3f);         //开显示
    SdCmd(0xc0);         //设置起始地址
}

//-----
//清空显示内存
//-----
void CleanScreen()
{
    uchar   i,j,k;
    for(i=0;i<8;i++)
    {
        CS1 = 1;
        CS2 = 1;          //控制器全开
        SdCmd(0x40);     //设置列地址

        SdCmd(0xb8+i);   //设置页地址
        CS1 = 1;
        CS2 = 0;          //选择控制器 1
        for(j=0;j<64;j++)
            SdData(0x00);
        CS1 = 0;
        CS2 = 1;          //选择控制器 2
        for(k=64;k<128;k++)
            SdData(0x00);
    }
}

//-----
//显示图象
//-----
void ShowPics(uchar code *bmp)
{
    uchar   i,j,k;
    for(i=0;i<8;i++)
    {
        CS1 = 1;
        CS2 = 1;          //控制器全开
        SdCmd(0x40);     //设置列地址
        SdCmd(0xb8+i);   //设置页地址
        CS1 = 1;
        CS2 = 0;          //选择控制器 1
        for(j=0;j<64;j++)
            SdData(bmp[i*128+j]); //写入图象编码
        CS1 = 0;
        CS2 = 1;          //选择控制器 2
        for(k=64;k<128;k++)
            SdData(bmp[i*128+k]); //写入图象编码
    }
}

//-----
//主程序
//-----
main()
{
    Init();                //初始化模块
    CleanScreen();         //清除显示内存
    ShowPics(LOGO);       //显示LOGO图形
    while(1)
    {}
}

//end of program
```