

# LM6029

## 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	2005-11-22	PanFeng
0.2	修改 4 节指令集说明: LCD bias select 的指令描述	2006-02-10	PanFeng



## 目录

1. 简介 .....	3
2. 电路连接原理简图 .....	3
3. 指令操作 .....	3
4. 指令集说明 .....	4
5. 显示屏与显示区位置关系 .....	5
6. 软件编程流程图 .....	5
附:参考程序 .....	6

## 1. 简介

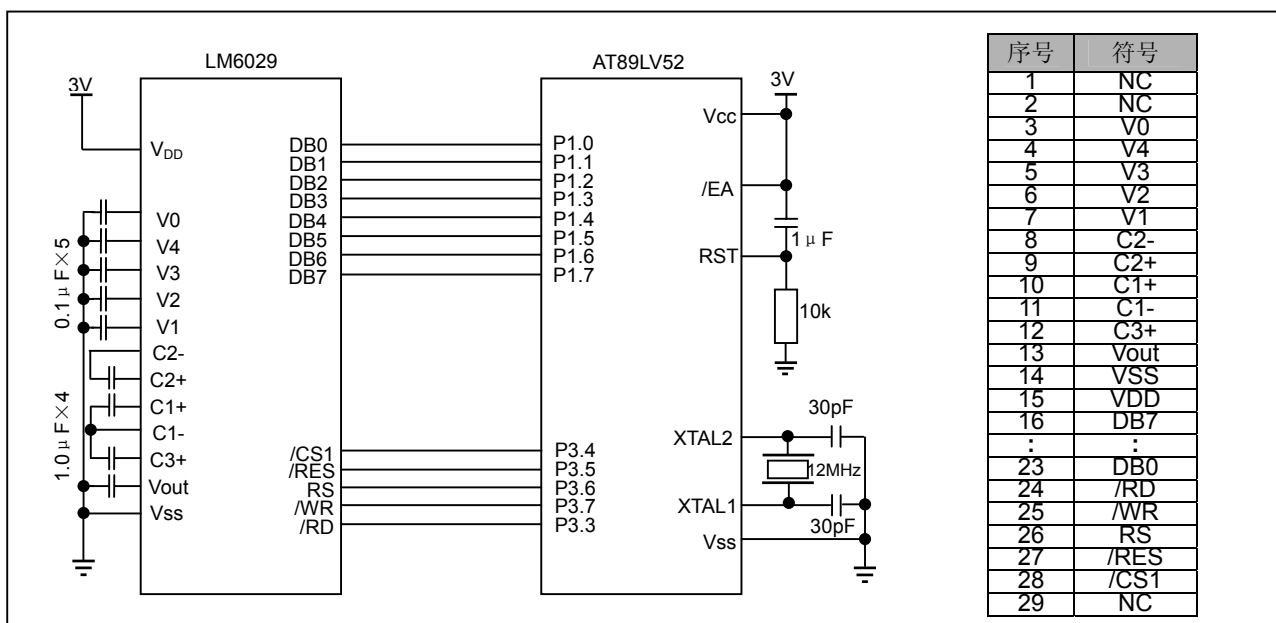
深圳市拓普微科技开发有限公司生产的 LM6029 点阵图形式液晶显示模块，采用 S6B0724 控制器，模块大小 57.6 × 42.0 × 2.0mm，点阵数 128 × 64dot。轻薄设计。适配 Intel8080 操作时序电路。8 bits 并口数据传输方式，软件调节对比度等功能。

基本特性：

- 电源电压：3.0V
- 显示分辨率：128 × 64dot
- 显示方式：FSTN，正显，半透

## 2. 电路连接原理简图

采用 AT89LV52 单片机同 LM6029 液晶模块通过 I/O 口相连为例。采用间接控制方式。



## 3. 指令操作

Intel8080 时序接口信号组合功能表如下：

操作	RS	/WR	/RD	功能说明
写寄存器命令	0	0	1	写指令到指令寄存器
读寄存器命令	0	1	0	读状态字 (READ STATUS)
写数据操作	1	0	1	写显示数据
读数据操作	1	1	0	读显示数据

## 4. 指令集说明

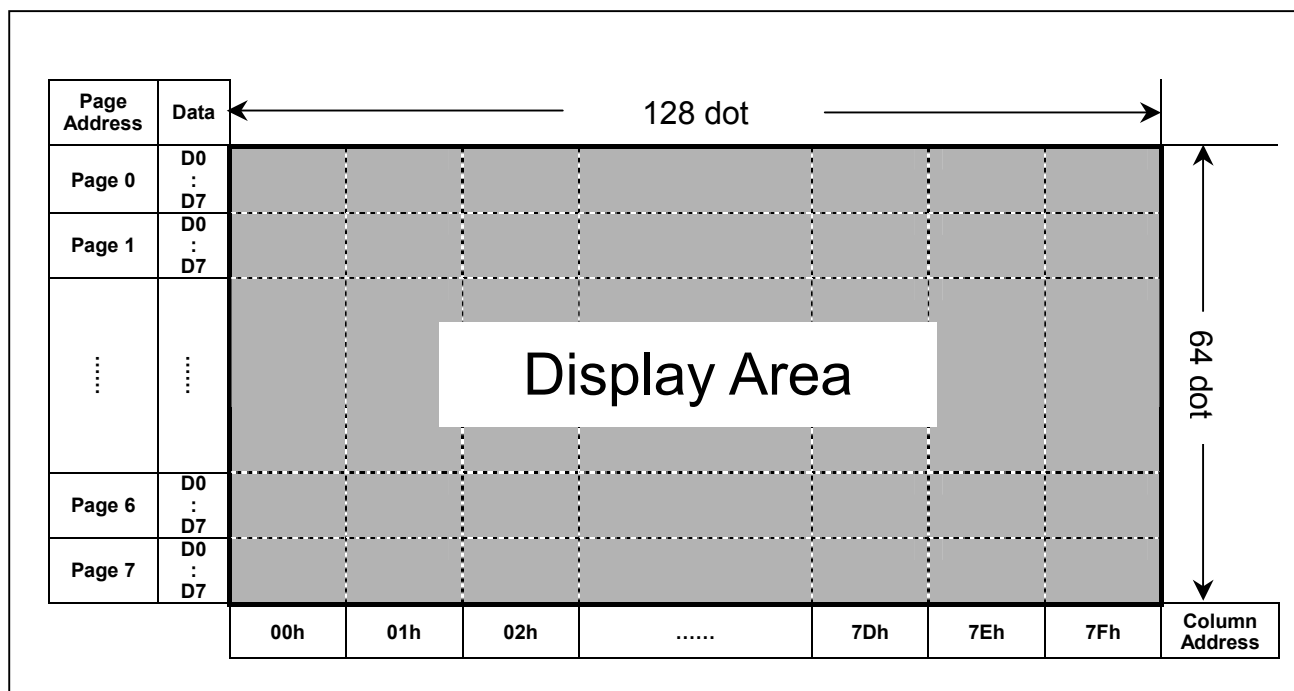
指令名称	RS	/WR	/RD	Code								指令描述	
				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Display ON/OFF	0	0	1	1	0	1	0	1	1	1	1	DON	LCD 显示开关设置 DON=0:关显示 DON=1:开显示
Initial display line	0	0	1	0	1	ST5	ST4	ST3	ST2	ST1	ST0		DDRAM 显示起始行地址设置
Set Page address	0	0	1	1	0	1	1	P3	P2	P1	P0		DDRAM 显示页地址设置
Set column address MSB	0	0	1	0	0	0	1	Y7	Y6	Y5	Y4		DDRAM 显示列地址高 4 位设置
Set column address LSB	0	0	1	0	0	0	0	Y3	Y2	Y1	Y0		DDRAM 显示列地址低 4 位设置
Read status	0	1	0	Busy	ADC	ON/OFF	RESETB	0	0	0	0		读状态字 Busy=0:控制器准备好 Busy=1:控制器忙 ADC=0:列地址反向 ADC=1:列地址正向 ON/OFF=0:开显示状态 ON/OFF=1:关显示状态 RESETB=0:正常状态 RESETB=1:复位状态
Write display data	1	0	1	Write data								写显示数据	
Read display data	1	1	0	Read data								读显示数据	
ADC select	0	0	1	1	0	1	0	0	0	0	ADC		选择 SEG 输出方向 ADC=0:正向(SEG0→SEG131) ADC=1:反向(SEG131→SEG0)
Reverse display ON/OFF	0	0	1	1	0	1	0	0	1	1	REV		LCD 正/反显开关设置 REV=0:正显 REV=1:反显
Entire display ON/OFF	0	0	1	1	0	1	0	0	1	0	EON		显示设置 EON=0:正常显示 EON=1:全亮显示
LCD bias select	0	0	1	1	0	1	0	0	0	1	BIAS		LCD 驱动电压的偏压比设置 BIAS=0:1/9 bias BIAS=1:1/7 bias
Set modify-read	0	0	1	1	1	1	0	0	0	0	0		修改-读指令设置
Reset modify-read	0	0	1	1	1	1	0	1	1	1	0		修改-读指令结束设置
Reset	0	0	1	1	1	1	0	0	0	1	0		内部重起
SHL select	0	0	1	1	1	0	0	SHL	x	x	x		选择 COM 输出方向 SHL=0:正向(COM0→COM63) SHL=1:反向(COM63→COM0)
Power control	0	0	1	0	0	1	0	1	VC	VR	VF		内部电源操作设置
Regulator resistor select	0	0	1	0	0	1	0	0	R2	R1	R0		内部电阻率(Rb/Ra)设置
Set reference voltage mode	0	0	1	1	0	0	0	0	0	0	1		对比度电流调节设置
Set reference voltage register	0	0	1	x	x	SV5	SV4	SV3	SV2	SV1	SV0		对比度电流大小设置 SV5~SV0:=000000→111111 small→large
Set static indicator mode	0	0	1	1	0	1	0	1	1	0	SM		指针状态开关设置 SM=0:开设置 SM=1:关设置
Set static indicator register	0	0	1	x	x	x	x	x	x	S1	S0		指针状态设置 S1,S0=00:关 S1,S0=01:开(约闪烁 1 秒) S1,S0=10:开(约闪烁 0.5 秒) S1,S0=11:开(常亮)
NOP	0	0	1	1	1	1	0	0	0	1	1		空操作

说明：1.表中 x 为任意值。

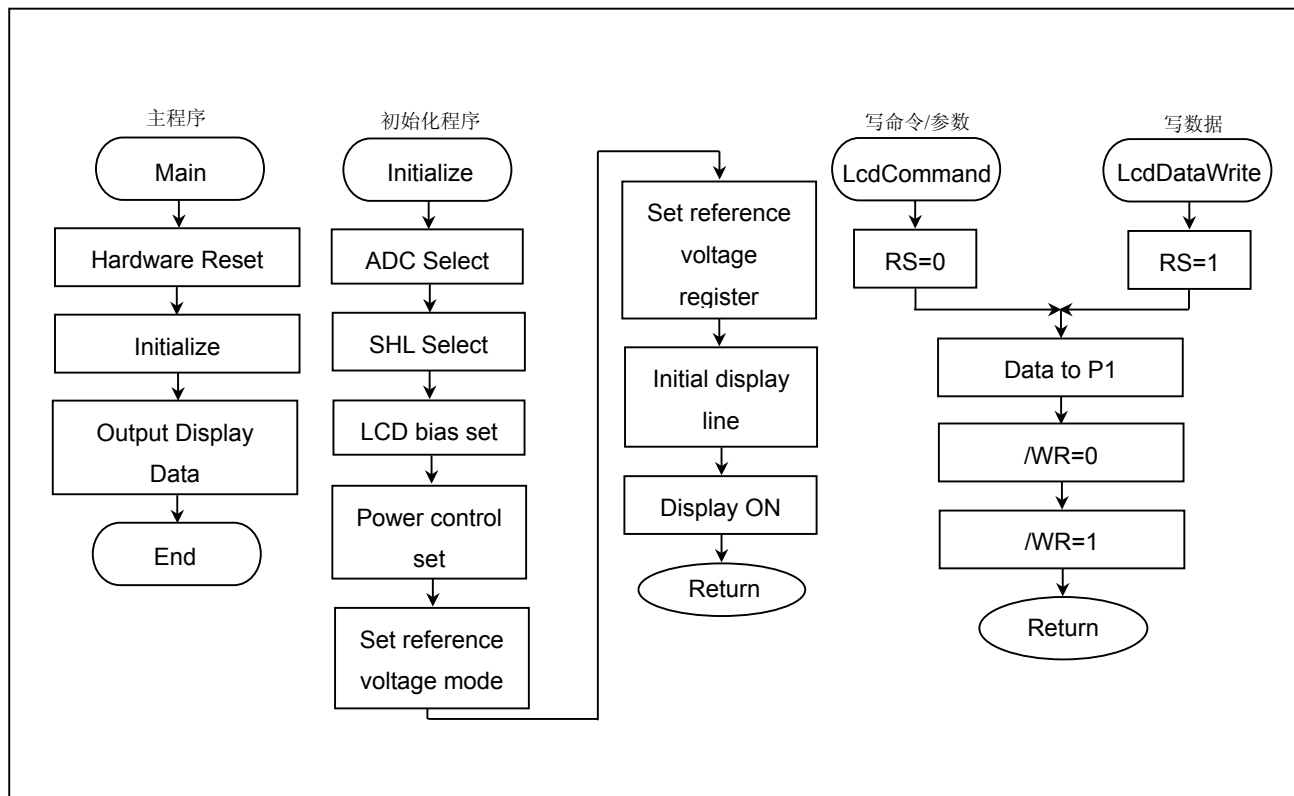
2.更多详细指令参数设置请参考 Samsung S6B0724 data sheet。

## 5. 显示屏与显示区位置关系

显示屏的显示位置与显示缓冲区 DDRAM 单元的一一对应关系如下图。



## 6. 软件编程流程图







```
void Delay(uint Delx)
{
    uint i=0;
    while(i<Delx)
        i++;
}
//-----
//写指令函数
//-----
void LcdCommand(uchar Com)
{
    RS=0;
    LcdDataPort=Com;
    _nop_();_nop_();_nop_();
    WR=0;
    _nop_();_nop_();_nop_();
    WR=1;
}
//-----
//写数据函数
//-----
void LcdDataWrite(uchar Data)
{
    RS=1;
    LcdDataPort=Data;
    _nop_();_nop_();_nop_();
    WR=0;
    _nop_();_nop_();_nop_();
    WR=1;
}
//-----
//初始化函数
//-----
void Initialize()
{
    CS1=0;
    RES=0;
    Delay(1000);
    RES=1;

    LcdCommand(0xa0);
    LcdCommand(0xc8);
    LcdCommand(0xa2);
    LcdCommand(0x2f);
    LcdCommand(0x81);
    LcdCommand(0x29);
    LcdCommand(0x40);
    LcdCommand(0xaf);
}
//-----
//设置显示位置
//-----
void SetPage(uchar Page)
{
    Page=Page & 0x0f;
    Page=Page | 0xb0;
    LcdCommand(Page);
}

}
void SetColumn(uchar Column)
{
    uchar temp;
    temp=Column;
    Column=Column & 0x0f;
    Column=Column | 0x00;
    LcdCommand(Column);
    temp=temp>>4;
    Column=temp & 0x0f;
    Column=Column | 0x10;
    LcdCommand(Column);
}
//-----
//清屏
//-----
void ClearScr()
{
    uchar i,j;
    for(i=0;i<8;i++)
    {
        SetColumn(0);
        SetPage(i);
        for(j=0;j<128;j++)
            LcdDataWrite(0x00);
    }
}
//-----
//显示子函数
//-----
void DisplayPic(uchar Page,uchar
Column,uchar *PicKu)
{
    uchar i,j;
    for(j=0;j<8;j++)
    {
        SetPage(Page+j);
        SetColumn(Column);
        for(i=0;i<128;i++)
            LcdDataWrite(PicKu[i+j*128]);
    }
}
//-----
//主函数
//-----
void Main()
{
    Initialize();
    Delay(1000);
    ClearScr();

    DisplayPic(0,0,Pic);
    while(1)
    {}
}
//-----End of Program-----
```