

# LM24064D

## 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	2005-11-15	周侃侃

## 目录

1 简介.....	3
2 应用.....	3
2.1 接口 .....	3
2.2 指令操作.....	4
2.3 指令集说明.....	4
2.4 显示屏幕.....	5
2.5 软件流程图.....	5
2.6 应用举例.....	6

## 1 简介

本公司产品 LM24064D 为 240x64 的全点阵图形液晶显示模块，内置 Toshiba T6963C 控制器。模块不仅可以显示单一的文本、图形，而且可以实现文本与图形的合成显示（“或”、“异或”、“与”三种逻辑关系）。此外，可以选择 6x8 和 8x8 两种字体来显示文本，而且还能够实现文本属性的特殊显示。使显示画面更加丰富，大大提高了效率，节约了产品的开发时间。

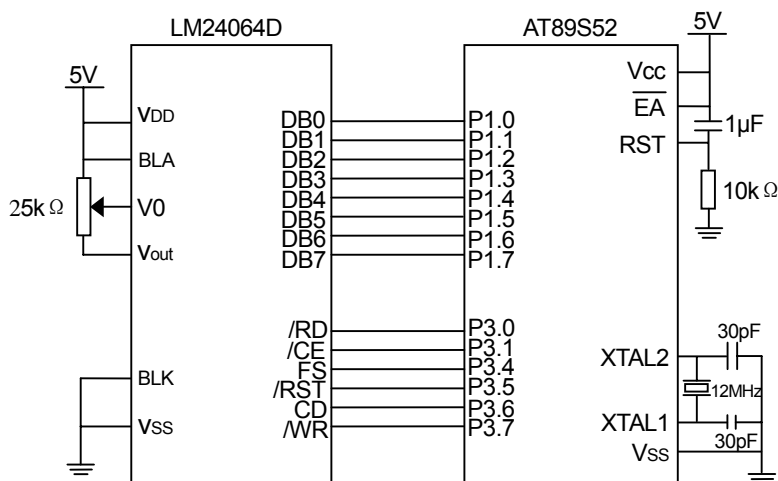
该模块的特点有：

- \* STN-LCD
- \* 内置 VLCD 负压电路
- \* Toshiba T6963C 液晶控制器
- \* 32K×8 显示内存
- \* 8 位单片机接口
- \* 自定义 8x8 字符 CGRAM

## 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。



模块引脚	名称
1	FG
2	Vss
3	VDD
4	V0
5	/WR
6	/RD
7	/CE
8	C/D
9	NC
10	/RST
11	DB0
...	...
18	DB7
19	FS1
20	Vout
21	BLA
22	BLK

## 2.2 指令操作

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

## 2.3 指令集说明

指令名称	参数*	C/D	/RD	/WR	Code								指令描述	
					DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
读状态字 Status Read		1	0	1	S7	S6	S5	S4	S3	S2	S1	S0	S0(指令读写状态):1(准备好),0(忙); S1(数据读写状态):1(准备好),0(忙); *读写数据/指令时要求S0和S1同时有效。 S2(数据自动读状态):1(准备好),0(忙); S3(数据自动写状态):1(准备好),0(忙); S5(控制器运行检测可能性):1(可能),0(不能); S6(屏读/屏拷贝出错状态):1(出错),0(正确); S7(闪烁状态检测):1(显示),0(关显示).	
寄存器设置 Register Setting		1	1	0	0	0	1	0	0	N2	N1	N0	N0,N1,N2 不能同时为 1	
	D1	0	1	0	x	X 坐标地址							光标地址设置 (N2,N1,N0=001)	
	D2	0	1	0	x	x	x	Y 坐标地址					CGRAM 偏置地址设置(N2,N1,N0=010)	
	D1	0	1	0	x	x	x	偏置地址						
	D2	0	1	0	0	0	0	0	0	0	0	0	显示地址设置(N2,N1,N0=100)	
控制字设置 Set Control Word		1	1	0	0	1	0	0	0	0	N1	N0	N1,N0=00; 文本显示区首地址	
	D1	0	1	0	地址数据 (低 8 位)									
	D2	0	1	0	地址数据 (高 8 位)									
	D1	0	1	0	每行列数								N1,N0=01; 文本显示区宽度	
	D2	0	1	0	0	0	0	0	0	0	0	0	N1,N0=10; 图形显示区首地址	
	D1	0	1	0	地址数据 (低 8 位)									
	D2	0	1	0	地址数据 (高 8 位)									
显示模式设置 Mode Set		1	1	0	0	0	0	0	0	CG	N2	N1	N0	CG:字符发生器选择; N2,N1,N0:显示方式设置位 N2,N1,N0=000; 逻辑"或" N2,N1,N0=001; 逻辑"异或" N2,N1,N0=011; 逻辑"与" N2,N1,N0=100; 文本属性*
	D1	0	1	0	每行列数								N1,N0=11; 图形显示区宽度	
	D2	0	1	0	0	0	0	0	0	0	0	0	N0:光标闪烁设置 1:开;0:关 N1:光标显示设置 1:开;0:关(在文本开启时有效) N2:文本显示设置 1:开;0:关 N3:图形显示设置 1:开;0:关	
D1	0	1	0	1	0	0	1	N3	N2	N1	N0			
光标设置 Cursor Pattern		1	1	0	1	0	1	0	0	N2	N1	N0	N2,N1,N0=000~111; 光标 1~8 行	
自动读写命令 Data Auto Read/Write		1	1	0	1	0	1	1	0	0	N1	N0	N1,N0=00; 进入自动写方式 N1,N0=01; 进入自动读方式 N1,N0=10/11; 退出自动读/写方式	
数据读写命令 Data Read/Write		1	1	0	1	1	0	0	0	N2	N1	N0	N2,N1,N0=000; 数据写,地址加一 N2,N1,N0=001; 数据读,地址加一 N2,N1,N0=010; 数据写,地址减一 N2,N1,N0=011; 数据读,地址减一 N2,N1,N0=100; 数据写,地址不变 N2,N1,N0=101; 数据读,地址不变	
	D1	0	1	0	数据									
	D1	0	1	0	数据									
	D1	0	1	0	数据									
	D1	0	1	0	数据									
读屏幕 Screen Peek		1	1	0	1	1	1	0	0	0	0	0	读出当前一个字节的数据显示数据	
屏幕拷贝 Screen Copy		1	1	0	1	1	1	0	1	0	0	0	将当前一行数个字节的数据显示数据读取出来作为图形显示数据返回到图形显示区相应的显示单元内	
位设置/复位 Bit Set/Reset		1	1	0	1	1	1	1	N3	N2	N1	N0	N3=0; 位清零 N3=1; 位置一 N2,N1,N0=000~111; Bit0~Bit7	
写数据/参数 Data Write		0	1	0	数据/参数								写显示数据/指令参数	
读数据 Data Read		0	0	1	数据								读显示数据	

\* 参数按“参数1→参数2→指令”的顺序以数据的形式输入

\* 显示文本属性时, 属性代码储存在文本属性区 (图形显示区转化而来)

## 2.4 显示屏幕

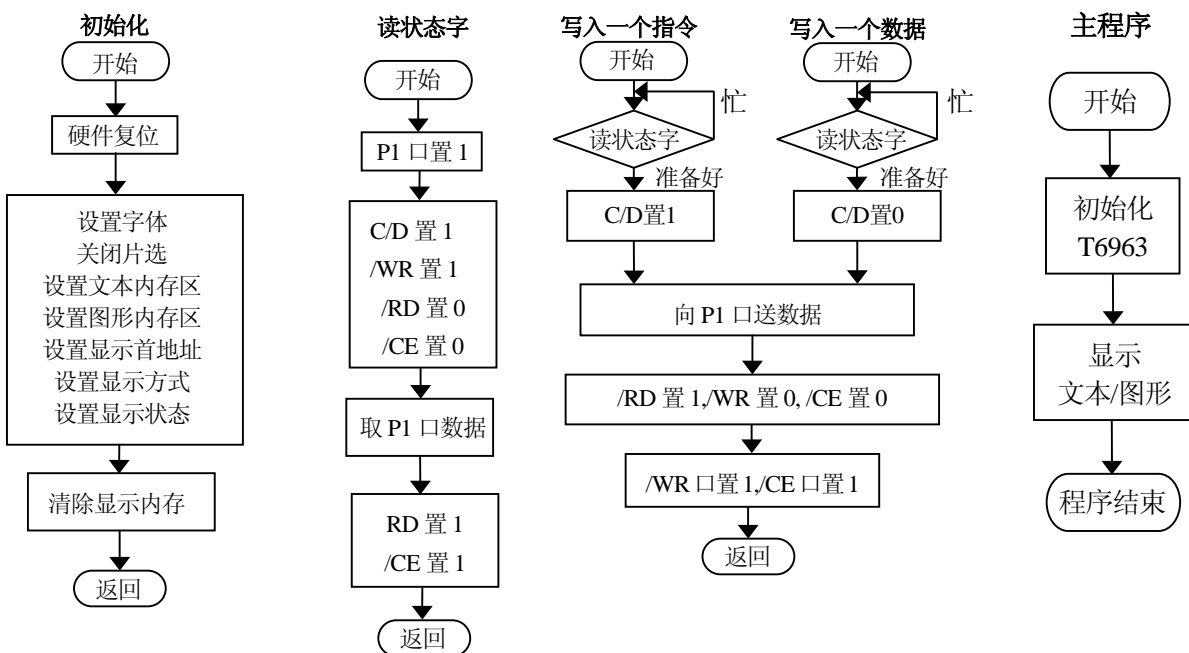
文本显示单元与显示屏上字符位的对应关系（以 FS=0 <8×8 字体>，首地址 0000H 为例）：

		列补偿地址						
		00H	01H	02H	...	...	1BH	1CH
行首地址	0000H	<b>30字符×8行</b>						
	001EH							
	... ..							
	00D2H							

图形显示单元与显示屏上像素的对应关系：（以首地址 0800H 为例）：

		列补偿地址											
		00H			01H			.....			1DH		
		DB7	...	DB0	DB7	...	DB0	DB7	...	DB0	DB7	...	DB0
行首地址	0800H	<b>240×64像素</b>											
	081EH												
	... ..												
	0F62H												

## 2.5 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。



## 2.6 应用举例

参考程序:

```

//本程序为LM24064D显示程序
//程序结果显示为TOPWAYLOGO图形和
//"    TOPWAY    "
//"  LM24064DBC  "
//"www.topwaydispaly.com "

#include<stdio.h>
#include<reg52.h>
#include<intrins.h>                //头文件

#define uchar unsigned char        //0~255
#define uint unsigned int          //0~65535

#define lcd_bus    P1
sbit   _WR    =   P3^7;
sbit   _RD    =   P3^0;
sbit   _CE    =   P3^1;
sbit   CD     =   P3^6;
sbit   _RES   =   P3^5;
sbit   FS     =   P3^4;                //端口设置

uchar code TOPWAYLOGO[]={
/*-- 调入了一幅图像--*/
/*-- 宽度x高度=120x64 --*/
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,
0x00,0x00,0x00,0x03,0xFF,0xFF,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x07,0xFF,0xFF,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,
0xFF,0xFF,0xFF,0xC0,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFF,
0xFF,0xFC,0x3F,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0xFF,0xFF,0xFF,
0xE3,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFE,0x1F,
0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0xFF,0xFF,0xFF,0xF0,0xFF,0xFF,
0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xFF,0xFF,0xFF,0xFF,0xC7,0xFF,0xFF,0xE0,
0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFE,0x3F,0xFF,0xFF,0xE0,0x00,
0x00,0x00,0x00,0x7F,0xFF,0xFF,0xFF,0xFF,0xF8,0xFF,0xFF,0xFF,0xE0,0x00,0x00,
0x00,0x00,0x3F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE3,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,
0x1F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x8F,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x07,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFE,0x3F,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x1F,0xC0,0x7F,
0xFF,0xFF,0xFF,0xFF,0xF8,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x03,
0xFF,0xFF,0xFF,0xE3,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x1F,
0xFF,0xFF,0xC7,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0xFF,
0xFF,0x1F,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0xFF,0xFE,
0x3F,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xFF,0xF1,0xFF,0xFF,
0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0xFF,0xE3,0xFF,0xFF,0xFF,
0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0xFF,0xC7,0xFF,0xFF,0xFF,0xFF,
0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0xFF,0x9F,0xFF,0xFF,0xFF,0xFF,0xFF,
0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0xFF,0x3F,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,
0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xFC,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,
0x00,0x00,0x00,0x00,0x00,0x7F,0xF8,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,
0x00,0x00,0x00,0x00,0xFF,0x1,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,
0x00,0x00,0x00,0xFF,0xE3,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,
0x00,0x01,0xFF,0xE7,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,
0x01,0xFF,0xCF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x03,
0xFF,0x9F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x03,0xFF,
0x3F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x07,0xFE,0x7F,
0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0xFC,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xF9,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xF3,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0xF7,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x1F,0xEF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,

```



```

0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0xCF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x9F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,
0x00,0x00,0x00,0x00,0x00,0x3F,0xBF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,
0x00,0x00,0x00,0x00,0x7F,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,
0x00,0x00,0x00,0x7E,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,
0x00,0x00,0x7C,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,
0x00,0xFD,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,
0xFB,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xE0,0x00,0x00,0x00,0x01,0xF7,0xC0,
0x38,0x1E,0x03,0x38,0xE7,0x87,0x1C,0x60,0x00,0x00,0x00,0x00,0x01,0xEF,0xF9,0xF3,
0xCE,0x39,0x98,0x67,0xB7,0x89,0xE0,0x00,0x00,0x00,0x00,0x01,0xDF,0xF9,0xF3,0xCE,0x3F,
0x38,0x1E,0x03,0x38,0xE7,0x87,0x1C,0x60,0x00,0x00,0x00,0x00,0x01,0xDF,0xF9,0xF3,0xCE,0x3F,
0xC2,0x4E,0x01,0xE3,0xE0,0x00,0x00,0x00,0x00,0x03,0xBF,0xF9,0xF8,0x1E,0x3F,0xC7,
0x1E,0x79,0xE3,0xE0,0x00,0x00,0x00,0x00,0x03,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xE0,0x00,0x00,0x00,0x00,0x03,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xE0,0x00,0x00,0x00,0x00,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
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,
};
//-----
//延时子程序
//-----
void delay(uint m) //在时钟频率为 12MHz 下，延时约为 1ms
{
    uint i;
    uint j;
    for(i=0;i<m;i++)
        for(j=0;j<109;j++);
}
//-----
//指令写入子程序
//-----
void SdCmd(uchar command) //从P1 口送入指令
{
    delay(0);
    lcd_bus = command;
    CD = 1;
    _RD = 1;
    _WR = 0;
    _CE = 0;
    _WR = 1;
    _CE = 1;
}
//-----
//数据写入子程序
//-----
void SdData(uchar ddata) //从P1 口送入数据
{
    delay(0);
    lcd_bus = ddata;
    CD = 0;
    _RD = 1;
    _WR = 0;
    _CE = 0;
    _WR = 1;
    _CE = 1;
}
//-----
//显示文本子程序
//-----
void writeTextScreen(uint startaddr,uchar line,uchar column,uchar code *s)
{
    startaddr=startaddr+line*0x1e+column; //确定显示首地址
    SdData((uchar)(startaddr)); //强制类型转换，将 16 位地址
    SdData((uchar)(startaddr>>8)); //转换为两个 8 位，并送入
    SdCmd(0x24);
    while(*s>0)
    {
        SdData((*s)-0x20); //T6963 的字符库代码比计算机
        SdCmd(0xc0); //低 0x20
        s++;
    }
}

```

```

//-----
//显示图形子程序
//-----
void writeGraphicScreen(uint startaddr,uchar line,uchar column,uchar *gddata)
{
    uchar i,j;
    startaddr=startaddr+line*0x1e+column;           //确定显示首地址
    SdData((uchar)(startaddr));                     //强制类型转换,将16位地址
    SdData((uchar)(startaddr>>8));                 //转换为两个8位,并送入
    SdCmd(0x24);
    for(i=0;i<64;i++);
        {
            for(j=0;j<15;j++)
                {
                    SdData(gddata[i*15+j]);
                    SdCmd(0xc0);                     //将图形编码送入,地址加一
                }
            startaddr=startaddr+30;                  //由于图象宽度小于屏幕宽度,所以
            SdData((uchar)(startaddr));             //显示完一行,换行时地址须加30
            SdData((uchar)(startaddr>>8));
            SdData(0x24);
        }
}
//-----
//清理显示内存子程序
//-----
void cleanscreen()
{
    uint i;
    SdData(0x00);
    SdData(0x00);
    SdCmd(0x24);                                     //确定清屏的起始位置
    delay(0);
    SdCmd(0xb0);                                     //开启自动写数据
    for(i=0x0000;i<0x2000;i++)
        SdData(0x00);                               //写入0x00清理屏幕
    SdCmd(0xb2);                                     //关闭自动写
}
//-----
//初始化子程序
//-----
void initLCDM()
{
    _RES= 0;
    delay(250);
    _RES= 1;
    delay(100);                                     //硬件复位
    FS      = 0;                                    //选择8x8字体
    _CE     = 1;                                    //关闭芯片
    SdData(0x00);SdData(0x00);SdCmd(0x40);          //设置文本显示区首地址
    SdData(0x1e);SdData(0x00);SdCmd(0x41);          //设置文本显示区宽度
    SdData(0x00);SdData(0x08);SdCmd(0x42);          //设置图像显示区首地址
    SdData(0x1e);SdData(0x00);SdCmd(0x43);          //设置图像显示区宽度
    SdData(0x00);SdData(0x00);SdCmd(0x24);          //设置显示首地址
    SdCmd(0x80);                                     //设置显示方式
    SdCmd(0x9c);                                     //设置显示状态
    cleanscreen();                                   //清屏
}
//-----
//主程序
//-----
void main()
{
    initLCDM();                                     //初始化
    writeGraphicScreen(0x0800,0x00,0x07,TOPWAYLOGO); //显示图象
    delay(2000);
    cleanscreen();
    writeTextScreen(0x0000,0x02,0x0c,"TOPWAY");     //显示文本内容
    writeTextScreen(0x0000,0x03,0x0a,"LM24064DBC");
    writeTextScreen(0x0000,0x05,0x04,"www.topwaydisplay.com");
    delay(2000);                                    //延时
    while(1);
}
//end of program
    
```