

SED1335 控制器的液晶显示模块与单片机接口应用

1 引言

液晶显示(Liquid CRYSTAL Display)简称LCD, 以其独特的低压、微功耗特性广泛应用于便携式电子产品如移动通信和笔记本电脑中。SED1335 是日本 Seiko Epson 公司生产的液晶显示控制器, 是同类产品中功能较强的一款产品。320240 模块以其优良的品质广泛应用于各类高级仪器仪表、POS 机、彩屏手机、车载产品、军工产品等, 本文以此模块为例, 介绍 SED1335 的编程控制。

2 液晶显示模块

2.1 模块结构和外部接口信号

ZX320240 模块的结构框图如图 1 所示。该模块内置 SED1335 控制器, 由 LED 背光、SED1335 控制器、32 K*8 SRAM、液晶板等部分组成。行、列驱动均由 SPLC086A 控制, 液晶点阵数 320×240。

ZX320240 液晶显示模块有两个插头与外部连接, 其外部连接信号及功能如表 1 所列。

从表 1 和图 1 可见, 外部信号中除了电源外, 其余控制信号都是用于控制 SED1335 控制器的。其中包括数据线 7 位、地址线 A0、片选信号及读写信号。

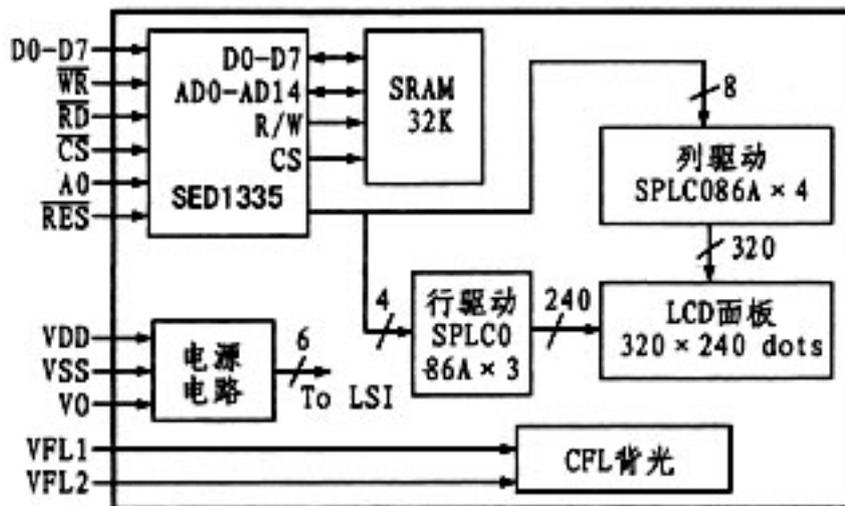


图 1 MSP-G320240DBC 与 AT89C51 接口电路



表1 MSP - G320240D 外部连接信号及功能

| 插头 | 引脚序号 | 信号名称 | 功能 |
|-------------|------|-------------------------|---|
| 插 头 1 | 1 | VSS | 地(0V) |
| | 2 | VDD | 供电电源(5V) |
| | 3 | VO | 阴极电源 |
| | 4 | $\overline{\text{WR}}$ | 写信号 |
| | 5 | $\overline{\text{RD}}$ | 读信号 |
| | 6 | CS | 片选 |
| 插 头 2 | 7 | A0 | A0 高, RD低, $\overline{\text{WR}}$ 高: 读显示数据和光标地址; A0 高, RD高, $\overline{\text{WR}}$ 低: 写命令字; A0 低, RD低, $\overline{\text{WR}}$ 高: 读状态字; A0 低, RD高, $\overline{\text{WR}}$ 低: 写显示数据和参数 |
| | 8 | $\overline{\text{RES}}$ | 复位 |
| | 9~16 | D0~D7 | 数据位 |
| 插 头 2 | 1 | VFL1 | 背光电源 |
| | 5 | VFL2 | 背光电源 |
| | 2~4 | NC | 空 |

SED1335 具有较强的 I / O 缓冲器, 单片机访问 SED1335 不需要判断其是否为“忙”状态, SED1335 可随时准备接收单片机的访问。并及时地传输单片机发来的指令和数据。另外, SED1335 具有较强的管理显示存储器的性能, 内置一个字符发生器, 具有 160 种 57 点阵字体的字符, 并能分区管理 64 K 的显示存储器, 可以同时管理 3 个或 4 个显示区, 并能同时管理用户自定义字符发生器。

SED1335 显示区根据其数据的性质可以分为文本显示区和图形显示区。文本显示区单元的数据被认为是字符代码。图形显示区单元的数据被认为是 8 点位的显示数据。

2. 2 ZX320240 模块与 AT89C51 接口设计

根据 SED1335 的电路特性, ZX320240 模块与 AT89C51 单片机采用直接接口方式, 电路如图 2 所示。单片机的 P0 口接至 SED1335 的 7 位数据线(D0~D7), SED1335 的读写控制信号 RD 的反、WR 的反分别由 AT89C51 的读写控制信号 RD 的反、WR 的反控制。注意, SED1335 的 A0 接单片机 P20 引脚, 片选 CS 的反接单片机 P27 引脚。CS 的反低电平有效, 与 A0 信号一起确定电路连接的 SED1335 的命令口地址为 0x7100, 数据口地址为 0x7000。单片机通过此命令口地址和数据口地址访问 SED1335。首先, 将指令代码写入命令口缓冲器内(A0=1), 然后, 将指令所需的参数按顺序通过数据口写入相应的功能寄存器内(A0=0)。



ZXLCD

中显液晶 北京中显电子有限公司, 现货库存, 欢迎光临!www.zxlcd.com,010-52926620,52926621

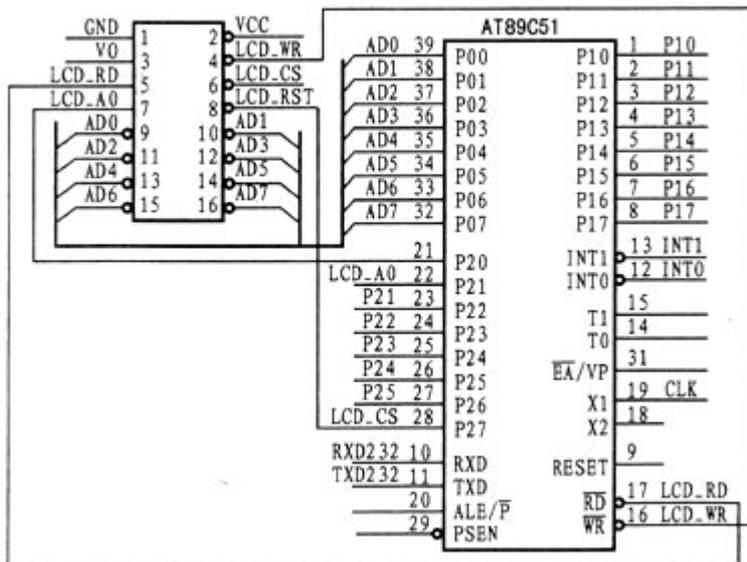


图 2 MSP-G320240D 液晶显示模块框图

3 SED1335 软件编程

SED1335 软件功能非常强大，命令控制操作简单，共有 14 条指令，用户只需给不同指令输入控制参数，就可得到所需的显示内容。

3. 1 SED1335 初始化

SED1335 初始化包括设置显示区域、字符显示大小、显示方式等。以下给出初始化的相关程序。程序中指令代码的具体含义见参考文献[1]。



```
// P2.7---CS P2.0 --- A0
char xdata LcmCmdPort _at_ 0x7100; // 写命令口
char xdata LcmWdataPort _at_ 0x7000; // 写数据口
#define LcmRdataPort LcmCmdPort //定义符号常量, 读数据口
#define LcmStatusPort LcmWdataPort //定义符号常量, 状态口
#define SystemSet 0x40 //初始化设置指令代码 0x40
#define Scroll 0x44 //显示域设置指令代码 0x44
#define CsrForm 0x5d //光标形状设置代码 0x5d
#define Overlay 0x5b //显示合成设置代码 0x5b
#define CsrW 0x46 //光标指针设置代码 0x46
#define DispOn 0x59 //显示状态设置代码 0x59
Uchar code ParaSysTable8 []={0x30,0x87,0x0f,0x27,0x42,0xf0,
0x28,0x00};
// 初始化指令参数 P1~P8 ,8 个参数
#define BasePart1 0x00
#define BasePart2 0x40
#define BasePart3 0x80
#define BasePart4 0x00
#define HdotScr 0x5a //点位移设置指令代码 0x5a
Uchar code ParaScrTableA []={0x00,BasePart1,0xf0,0x00,
BasePart2,0xf0,0x00,BasePart3,0x00,BasePart4 };
//显示域指令参数,10 个参数
void LcmInition( void ) {
    Uchar i;
    WriteCommand( SystemSet ); // 系统参数设置
    for (i=0;i<8;i++) {
```



```

    WriteData( ParaSysTable8[i] );
}
WriteCommand( Scroll );      //设定显示区域起始地址
for (i=0;i<10;i++) {
    WriteData( ParaScrTableA[i] );
}
WriteCommand( HdotScr );     // 写入点位移指令代码
WriteData( 0 );              // 写入 P1 参数
WriteCommand( Ovlay );      // 显示合成方式设置
WriteData( 4 );              // 0000 0100 显示一区图形,三区文本属性,二重"或"合成
WriteCommand( CsrW );       //CSRW 该指令地址设置了光标地址
WriteData( 0 );
WriteData( 0 );
WriteCommand( CsrForm);     //光标形状设置
WriteData( 0x07 );          //宽度
WriteData( 0x8f );          //高度
WriteCommand( DispOn );     // 显示状态指令代码
WriteData( 0x55);           // 显示 1~4 区开显示,光标开显示 01
初始化时光标不闪烁

```

3.2 清屏例程

清屏就是将显示内存的内容全部置 0 或全部置 1, ZX320240 模块在全部置 0 时底色为蓝色, 全部置 1 时底色为白色。

清屏的方法是, 光标从地址 0x0000 开始, 向显示缓冲区连续写入 0x00(或 0xff)。下面给出把 32K 显示内存的内容全部清 0 的清屏例程。

```

void LcmClear( void ) {
    Uint i1=32768;
    WriteCommand( CsrDirR ); // 定义光标自动右移
    WriteCommand( CsrW );    // 光标 Locate,定位
    WriteData( 0 );          // 设置参数光标指针低 8 位
    WriteData( 0 );          // 设置参数光标指针高 8 位
    WriteCommand( mWrite );  // 数据写入指令,代码 0x42
    while(i1-- ) {
        WriteData( 0x0 );    // 写入数据 0
    }
}

```

3.3 汉字显示例程



本文设置汉字显示方式为图形方式, 图形方式下显示汉字, 是根据已建立好的汉字字模库 (实际是一个汉字字模矩阵), 根据矩阵中汉字的位号选取要显示的汉字。

本文给出的汉字显示函数有 4 个参数, 分别是位置横坐标、位置纵坐标、字库位号、字符点阵格式 16×16 或 24×24 。ZXP-G320240 模块显示屏为 320×240 , 故对于 16×16 点阵汉字, 横向可显示 20 个, 纵向 15 个。分别对应横坐标 (0, 2, 4, 6, 8, ...18), 纵坐标为 (0, 16, 32, 48, ...224)。

为 (0,16,32,48,...224)。

```
void WriteOneWord ( Uint Ox,  Uchar Oy,  Uchar Cnumber,
Uchar DotWidth ) {
    Uint tempPtr,Optr;
    Uint tempCount1,tempCount2,nByte;
    Optr = (Uint)Oy * paraP9 + Ox;
    nByte = DotWidth/8;
    WriteCommand(CsrDirD); // 自动右移。
    tempPtr = (Uint)Cnumber * DotWidth*nByte;
    for (tempCount1=0;tempCount1<nByte;tempCount1++) {
        WriteCommand( CsrW ); // 光标定位指令
        WriteData( (Uchar)(Optr &0xff) ); // 设置光标地址 CSR
        WriteData( (Uchar)(Optr /256 ) );
        WriteCommand( mWrite ); // ram 写指令
        switch(DotWidth){
            case 16

                for(tempCount2=0;tempCount2<DotWidth;tempCount2++) {
                    WriteData ( Cdotlib [Cnumber].Msk [temp-
Count2*2+tempCount1] ); // 写入数据
                }

            case 24
                break;
            for(tempCount2=0;tempCount2<DotWidth;tempCount2++)
                { WriteData( Cdot24lib[tempPtr] ); // 写入数据
                    tempPtr += nByte;
                }
                break;
        }
        tempPtr = Cnumber * DotWidth*nByte + tempCount1+1;
        Optr++;
    }
}
```



例如要从屏幕左上角开始在第一行显示汉字“设置航向”。设字模库为:

```
struct typFNT_GB16 code Cdotlib[]={
0x40,0x00,0x21,0xF0,0x31,0x10,0x21,0x10,0x01,0x10,0x01,
0x10,0xE2,0x0E,0x25,0xF8,0x21,0x08,0x21,0x08,0x20,0x90,
0x20,0x90,0x28,0x60,0x30,0x90,0x23,0x0E,0x0C,0x04, //0 设
0x3F,0xF8,0x24,0x48,0x24,0x48,0x3F,0xF8,0x01,0x00,0x7F,
0xFC,0x02,0x00,0x1F,0xF0,0x10,0x10,0x1F,0xF0,0x10,0x10,
0x1F,0xF0,0x10,0x10,0x1F,0xF0,0x10,0x10,0xFF,0xFE, //1 置
0x08,0x40,0x10,0x20,0x3C,0x00,0x25,0xFC,0x34,0x00,0x2C,
0xF0,0x24,0x90,0xFC,0x90,0x24,0x90,0x34,0x90,0x2C,0x90,
0x24,0x90,0x24,0x92,0x45,0x12,0x55,0x0E,0x8A,0x00, //2 航
0x02,0x00,0x04,0x00,0x3F,0xFC,0x20,0x04,0x20,0x04,0x27,
0xE4,0x24,0x24,0x24,0x24,0x24,0x24,0x24,0x27,0xE4,
0x24,0x24,0x20,0x04,0x20,0x14,0x20,0x08,0x00,0x00, //3 向
WriteOneWord ( 0, 16, 0, 16 ); // 设
WriteOneWord ( 2, 16, 1, 16 ); // 置
WriteOneWord ( 4, 16, 2, 16 ); // 航
WriteOneWord ( 8, 16, 3, 16 ); // 向
```

4 结束语

本文根据 SED1335 的硬件和软件特性, 以 ZXP-G320240 模块为例。对 SED1335 控制器和单片机的接口应用和软件设计进行分析。该设计已成功应用于实际项目的液晶界面显示。本文所介绍的单片机和液晶显示控制器的接口技术对其他型号的显示控制器(如 SED1330、SED1336、ZXM6255 等)也有很好的参考价值。