

液晶驱动控制器 SED 1335 在 320 × 240LCD 显示中的应用

王 战 海

(沈阳大学 网络中心, 辽宁 沈阳 110044)

摘 要: 介绍了液晶驱动控制器 SED 1335 以及与 MCS-51 系列单片机组成的 320 × 240LCD (DM F- 50174) 显示控制系统的硬件实际应用电路原理图和 C51 语言源程序。

关键词: SED 1335; 单片机系统; 应用程序

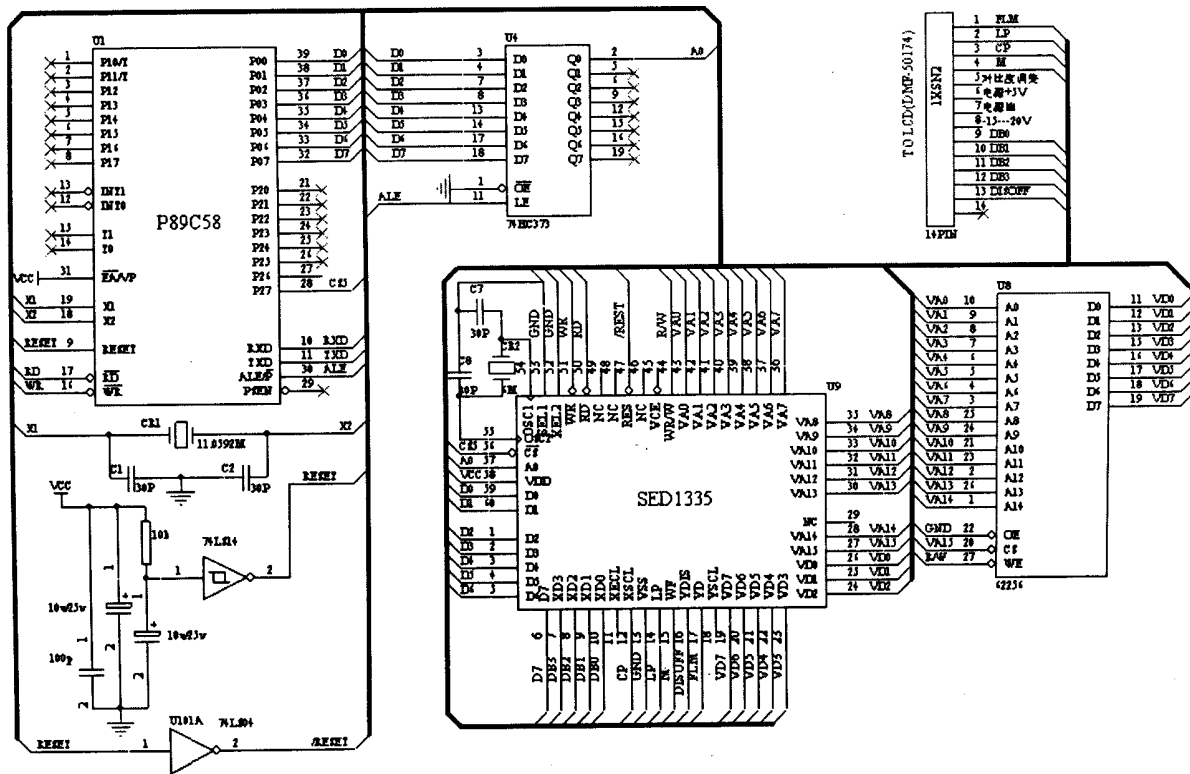
中图分类号: TP311

文献标识码: B

文章编号: 1007-533X (2004) 03-0264-04

1 SED 1335 与 MCS-51 系列单片机组成显示控制系统

下图是控制部分原理图(略去了系统中的其它部分): SED 1335 在进行显示驱动设计时需要为其设计一个显示数据存储器, 这里选择了随机存储器 RAM 芯片 62256, 该芯片具有 32K × 8 的存储空间; 图中的



收稿日期: 2004-03-12

作者简介: 王战海(1971-), 男, 助理工程师, 从事自动化控制研究

74LS373 在本系统中还另有其他用途, 所以保留下来。SED 1335 接口部分具有功能较强的 I/O 缓冲器, 这使得 μ PU 访问 SED 1335 不需要判断其是否忙, SED 1335 随时准备接收 μ PU 的访问并在内部时序控制下及时地把 μ PU 发来的指令数据传输就位。

由于 SED 1335 的复位端为低有效, 所以加入一个 74LS04 对 μ PU 的复位信号进行翻转以获得所需复位电平信号。SED 1335 的数据端口地址为 7FFE_H, 指令端口地址为 7FFF_H。

2 显示驱动程序

关于该控制芯片的文章和介绍也有不少(包括液晶生产厂家的说明书), 但很多介绍都过于简单, 使得在实际开发中仍然有许多地方存在难点, 所以给出以下程序是希望让开发者节省显示驱动程序的编写时间, 更好地提高整个系统的其他应用功能。本人在应用中整个程序是使用 C51 与 ASM 混合编程的, 主程序为 C 程序, 显示部分也是 C51 的程序, 为了更加清晰明确地说明显示功能的软件实现过程, 特别进行了删改, 只给出了相关部分并做了简化处理。

```

;      指令端口地址 7FFFH: 数据端口地址: 7FFEH
;      320 * 240 点阵整屏的点像素为 76800, 也就有 9600(2580H) 个字节, 单屏图形方式
; 所以第一区显示缓存地址范围为: 0000H - 257FH 第二区显示缓存地址范围为: 2580H - 4AFFH
; 第三区显示缓存地址范围为: 4B00H - 707FH 320 * 240 点阵整屏一行占 40 个字节的显示缓存
# include < AT89x51.h >
# include < Intrins.h >
# include < absacc.h >
# define uchar unsigned char
# define uint unsigned int
# define MyCommand          XBYTE[0x7FFF]
# define MyWriteData        XBYTE[0x7FFE]
uchar code ZKWU [] = {0x00, 0x40, 0x30, 0x17, 0x84, 0x68, 0x20, 0x09, // 沈
                     0x11, 0x21, 0xE1, 0x22, 0x22, 0x24, 0x28, 0x30,
                     0x80, 0x80, 0x80, 0xFC, 0x84, 0x88, 0xC0, 0x40,
                     0x40, 0x40, 0x40, 0x40, 0x42, 0x42, 0x3E, 0x00};
uchar code ZKL N [] = {0x00, 0x7D, 0x45, 0x49, 0x49, 0x51, 0x49, 0x49, // 阳
                      0x45, 0x45, 0x45, 0x69, 0x51, 0x41, 0x41, 0x41,
                      0x04, 0xFE, 0x04, 0x04, 0x04, 0x04, 0x04, 0xFC,
                      0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0xFC, 0x04}

void InitLCD (void);
void MyDelay (uint);
void SetCursor (uchar, uchar);
void WriteOneHZ (uchar, uchar, uchar *);
void main (void)
{
    uchar i;
    SP = 0x60;
    EA = 0;
    MyDelay (0x30);
    InitLCD ();
    MyDelay (0x9);
    MyCommand = 0x4f; /* 光标地址自动向下增一 */
    WriteOneHZ (0, 0, ZKWU);

```

```

    WriteOneHZ(0x2, 0x80, ZKL N);
    while(1); }
void InitLCD(void) // 初始化LCD 函数
{
    uint i;
    MyCommand= 0x40;          /* - - - - LCD SYSTEM SET 系统设置令, 8 个参数- - - - */
    MyWriteData= 0x30;
    MyWriteData= 0x87;
    MyWriteData= 0x07;
    MyWriteData= 39;          /* 显示域长度为 320dot */
    MyWriteData= 0x36;       /* 确定液晶工作频率 */
    MyWriteData= 239;        /* 显示屏高度为 240dot */
    MyWriteData= 40;         /* 显示屏一行所占显示缓冲区字节数(L) */
    MyWriteData= 0;          /* 显示屏一行所占显示缓冲区字节数(H) */
    MyDelay(0x9);
    MyCommand= 0x44;        /* - - - LCD SCROLL 显示区设置命令, 最多 10 个参数- - - */
    MyWriteData= 0x00;       /* 显示 1 区对应的显示 RAM 起始高 8 位地址 */
    MyWriteData= 0x00;       /* 显示 1 区对应的显示 RAM 起始低 8 位地址 */
    MyWriteData= 240;        /* 显示 1 区占用 240 个 dot 行 */
    MyWriteData= 0x80;       /* 显示 2 区对应的显示 RAM 起始低 8 位地址 */
    MyWriteData= 0x25;       /* 显示 2 区对应的显示 RAM 起始高 8 位地址 */
    MyWriteData= 240;        /* 显示 3 区占用 240 个 dot 行 */
    MyWriteData= 0x00;       /* 显示 3 区对应的显示 RAM 起始低 8 位地址 */
    MyWriteData= 0x4b;       /* 显示 3 区对应的显示 RAM 起始高 8 位地址 */
    MyDelay(0x9);
    MyCommand= 0x5a;         /* 水平卷动, 初始化时必须清零 */
    MyWriteData= 0x00;
    MyDelay(0x9);
    MyCommand= 0x5b;         /* 各个显示区的合成方式, 1 个参数 */
    MyWriteData= 0x1c;       /* 参数: 显示方式为三屏图形叠加, 逻辑或 */
    MyDelay(0x9);
    MyCommand= 0x59;         /* 打开显示, 1 个参数 */
    MyWriteData= 0x04;       /* 只显示第一个区并关闪烁 */
    MyDelay(0x9);
    MyCommand= 0x4c;         /* 光标自动向右增一 */
    MyDelay(0x9);
    MyCommand= 0x46;         /* 设定光标位置命令 */
    MyWriteData= 0;          /* 从 0000 位置开始 */
    MyWriteData= 0;
    MyDelay(0x9);
    MyCommand= 0x42;         /* 允许 uPU 连续地把显示数据写入显示区内命令 */
    for(i= 0; i< 32760; i+ + )
        {MyWriteData= 0;      /* 将 32K 显缓全部清零, 即LCD 不显示任何东西 */}
void SetCursor(uchar CursorHIGH, uchar CursorLOW) /* 设定光标, 允许显示 */
{
    MyDelay(0x9);

```

```

MyCommand= 0x46;          /* 设定光标位置命令 */
MyWriteData= CursorLOW;   /* 从 0000 位置开始 */
MyWriteData= CursorHIGH;
MyDelay(0x9);
MyCommand= 0x42;          /* 允许 MPU 连续地把显示数据写入显示区内命令 */
void WriteOneHZ(uchar CurHIGH, uchar CurLOW, uchar * ZiKu) /* 在指定位置显示一个汉字 */
{
    uchar i;
    SetCursor(CurHIGH, CurLOW); /* 设定光标, 允许显示 */
    i= 0;
    do {MyWriteData= ZiKu[i];
        i= i+ 2; }
    while(i< 0x1e);
    SetCursor(CurHIGH, CurLOW + 1);
    i= 1;
    while(i< 0x1f)
    {MyWriteData= ZiKu[i];
    i= i+ 2; }}
void MyDelay(uint delx) /* 延时子程序 */
{
    uint i= 0;
    while(i< delx)
    {i+ + ; }}

```

3 结束语

在实际应用中,对单片机的编程可以采用 C51 来做,也可以采用 ASM 来做;这里给出的驱动程序是 C51 的源程序,可以供参考。对于 320 × 240 这样的 LCD 显示屏,在实际应用中属于显示区域比较大的 LCD 显示屏,对于显示部分的软件编程工作量是比较大的,尤其想要使得显示界面丰富多彩(该显示屏可以选择四种不同的背景颜色)、图文并茂。以上给出的例子仅仅是完成最基本的但也是最重要的显示功能,以供借鉴。

参考文献:

- [1]李华 MCS-51 系列单片机实用接口技术[M].北京:航空航天大学出版社,1999.8
- [2]王建校,杨建国,宁改娣,等.51 系列单片机及 C51 程序设计[M].北京:科学出版社,2002
- [3]马忠梅,马岩,张凯,等.单片机的 C 语言应用程序设计[M].北京:航空航天大学出版社,1997.3
- [4]杨振江,杜铁军,李群.流行单片机实用子程序及应用实例[M].西安:西安电子科技大学出版社,2002.7

Application of LCD drive controller SED 1335 in 320X240 LCD display

WANG Zhan-hai

(Networks Center, Shenyang University, Shenyang 110044, China)

Abstract: Introduction to LCD drive controller SED 1335, principal circuit diagram, C51 programme and hardware for practical application of the 320X240 LCD (DMF-50174) display control system are dealt with in this paper.

Key Words: SED 1335; microprocessor system; application programme