

点阵式 LCD 显示模块在数控机床中的应用

许宜中¹ 朱欣华² 许向华¹

1. 苏州大学(215006) 2. 东南大学(210096)

Abstract

The application of dot LCD display module in NC machine is discussed in this paper. The elementary principle and technical characteristics of SED1335 are briefly introduced, and material application of SED1335 in controlling LCD display module of the NC machine is presented.

Keywords: LCD, SED1335 controller, NC machine, MCU

摘要

本文讨论了点阵式 LCD 显示模块在数控机床中的应用。简要地介绍了 SED1335 控制器的基本原理及其技术特性。给出了该控制器在控制数控机床液晶显示屏时的具体应用, 包括数控机床中液晶显示的实现思路、相关的硬件原理、软件的设计及字库的生成等。

关键词: LCD, SED1335 控制器, 数控机床, 单片机

1 SED1335 的基本原理及其特性

点阵式 LCD 液晶显示模块是一种集显示、控制与驱动为一体的显示器件。设计中, 为数控机床配置了 SHARP 公司生产的 LM32019T 点阵式液晶显示屏。该屏为 320×240 点阵, 像素大小为 0.36×0.36mm。为该屏配置的控制器是 SEIKO EPSON 公司的 SED1335。它是一种高性能 LCD 控制器, 用于 MCU 与液晶显示屏之间的接口控制。它接收来自 MCU 的指令和数据, 并产生相应的时序及数据控制液晶显示屏的显示。硬件结构可分为 MCU 接口部、内部控制部和驱动 LCM 的驱动部。SED1335 接口部具有功能较强的 I/O 缓冲器。MCU 访问 SED1335 不需要判断其“忙”, SED1335 随时准备接收 MCU 的访问并在内部时序控制下及时地把 MCU 发来的指令、数据传输到位。SED1335 在控制部设置了适配 8080 系列和 M6800 系列 MCU 的两种操作时序电路, 通过引脚的电平设置, 可以选择两者之一。控制部是 SED1335 的核心。由振荡器、功能逻辑电路、显示 RAM 管理电路、字符库及其管理电路以及产生驱动时序的时序发生器组成。振荡器工作频率可在 1M~10M 范围内选择。SED1335 能在很高的工作频率下迅速地解译 MCU 发来的指令代码, 将参数置入相应的寄存器内, 并触发相应的逻辑功能电路运行。控制部可以管理 64K 的显示存储器, 内藏的字符发生器及外扩的字符发生器 CGRAM 或 EXCGRAM。SED1335 将 64K 的显示 RAM 分成三个显示区。显示区可以根据数据的性质设置为文本方式或图形方式。在文本方式下, 该显示区的数据被认为是字符代码。这个字符代码作为字符发生器地址的一部分实现对字符发生器的寻址, 以取得相应的字模数据, 送到液晶显示驱动系统显示。在图形方式下, 该显示区的数据被认为是 8 位的显示数据而直接送到显示驱动系统显示。SED1335 驱动部具有各显示区的合成显示能力, 传输数据的组织功能及产生液晶显示模块所需要的时序。SED1335 向液晶显示模块传输数据的方式为 4 位并行方式。^[1]

2 数控机床中液晶显示的实现思路及具体方法

2.1 液晶显示的实现思路及硬件原理

为了加快系统的开发速度, 避免原系统程序的结构性改动, 采用了如下的设计思路: 给液晶显示模块配置了一个 MCU, 该 MCU 通过并行方式接收原系统中用于信号输入并处理 LED 显

示的 MCU 送来的显示控制命令, 并根据控制命令实现对 SED1335 的控制从而完成相应的显示任务。基于这种思路的液晶显示模块的硬件原理如图 1 所示。图 1 中的 MCU1 和 MCU2 均为 51 系列的单片机, 其中 MCU1 (即原系统中的 MCU) 负责处理外部输入的信号, 如: 键盘扫描信号, 急停信号, 暂停信号, X、Z 轴的 +1、-1 信号等。将这些信号的处理结果, 以 ASCII 码的形式, 按照与 MCU2 约定的通信协议传送给 MCU2。MCU2 负责处理 MCU1 传送过来的控制命令, 控制 SED1335 在液晶显示屏上显示相应内容。采用并行通信方式, 数据传输速度快, 硬件结构相对简单。MCU2 接收控制命令时, 采用中断方式, 低电平触发。

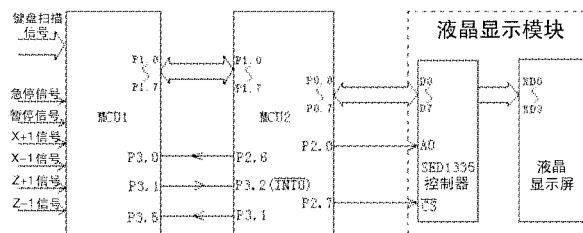


图 1 液晶显示模块的硬件原理框图

两者之间通信还需要三根握手信号线进行控制。MCU1 发送数据前首先要判断 MCU2 是否允许发送, 即 P3.5 是否为 1。当 P3.5=1 时, 可以发送数据; 当 P3.5=0 时, 等待允许发送信号。MCU2 允许发送数据时, MCU1 将待发送的数据写入 P1 口, 并使 P3.1=0, 从而 MCU2 进入中断, 读入 P1 口上的数据。MCU2 进入中断服务程序以后关闭中断, 并使 P3.5=0, 不允许 MCU1 发送数据, 直到读取 P1 口的数据处理完毕, 再次开中断允许 MCU1 发送数据。若 MCU2 接收到的数据串错误, 则往 P2.6 引脚上写 1。当 MCU1 检测到 P3.0 为高时, 则将原来的数据串重新发送一遍, 直到 MCU2 接收正确。相关的操作可参考文献[1]。

MCU2 根据接收到的数据, 将要求显示的内容送到液晶显示模块。用 MCU2 的 P2.7 作为片选信号线, 配合地址信号线 A0 实现对 SED1335 的操作。

2.2 液晶显示模块的软件设计

2.2.1 MCU1 与 MCU2 之间的通信协议

对于数控系统而言,显示方式主要有加工状态下的显示和非加工状态下的显示两种。非加工状态下的显示,实时性要求不高。加工状态下,要求显示坐标值,还要显示暂停、急停及故障提示信息提示等,实时性强。因此,通信协议分为两种情况:编辑状态下的显示和运行状态下的显示。编辑状态下,又分为单个字符的显示与数据块的显示。运行状态下,在运行之前,X、Z 的坐标值已经被赋以初值;运行过程中,要求及时响应 X、Z 坐标值的+1、-1 命令。对此,通信协议中设置了专门的特征命令字。具体的通信协议如下:①编辑状态下,单个字符显示时的通信协议为:EE、显示类型、显示字符的 ASCII 码校验和;②编辑状态下,数据块显示时的通信协议为:DE、显示字符 1 的 ASCII 码、显示字符 2 的 ASCII 码、……校验和、DF;③运行状态下,X、Z 坐标值的+1、-1 显示时的通信协议为:F0,表示 X 坐标值+1,Z 坐标值+1;F1,表示 X 坐标值+1,Z 坐标值-1;F2,表示 X 坐标值+1,Z 坐标值不变;F3,表示 X 坐标值-1,Z 坐标值+1;F4,表示 X 坐标值-1,Z 坐标值-1;F5,表示 X 坐标值-1,Z 坐标值不变;F6,表示 X 坐标值不变,Z 坐标值+1;F7,表示 X 坐标值不变,Z 坐标值-1。

2.2.2 相关软件的设计

要实现正确显示,首先必须根据需要对 SED1335 控制器的各个参数进行设定,以规定 LCD 显示器的显示类型、点阵数目、字符点阵大小等。这些参数一经正确设定,LCD 显示系统便可进入工作状态。初始化程序如下:

```
#define wrcoaddr XBYTE[0x7f00] //定义写指令代码地址
#define wrdaaddr XBYTE[0x7e00] //定义'j'参数及数据地址
#define rddaaddr XBYTE[0x7f00] //定义读数据地址
uchar code systab[]={0x32,0x87,0x07,0x27,0x42,0xef,0x28,0x00};
uchar code scrtab []={0x00,0x00,0xef,0x00,0x30,0xef,0x00,0x00,0x00,0x00};
void initlcd (void) //初始化液晶屏
{uchar ii;
 wrcoaddr=0x40; //设定 LCD 的系统参数
 for(ii=0;ii<8;ii++)
 {wrdaaddr=systab[ii];}
 wrcoaddr=0x44;
 for(ii=0;ii<10;ii++)
 {wrdaaddr=scrtab[ii];}
 wrcoaddr=0x5a;wrdaaddr=0x00; //设置水平点位移量
 wrcoaddr=0x5b;wrdaaddr=0x0d; //OVLAY 指令二重,L1 和 L3 文本
 clear(); //清屏即清除显示 RAM
 wrcoaddr=0x59;wrdaaddr=0x17;
 wrcoaddr=0x5d;wrdaaddr=0x07;
 wrdaaddr=0x8f;}
```

由于 MCU1 和 MCU2 之间是通过中断方式实现信息交换的,因此,MCU2 的中断接收程序是显示系统的关键之一。中断接收程序中,要考虑单字符、数据块以及运行状态下 X、Z 坐标值+1、-1 的情况。具体中断接收程序如下:

```
void receive_char (void) interrupt 0 //中断接收程序,采用外部中断 0
{rec_data[rec_lth]=P1;
 txdlne=0; //不允许 MCU1 发送数据
 EX0=0; //关闭外部中断 0
 resendlne=0; //数据不需要重新发送
 if(rec_data[0]==0XEE)
 {sum11=sum11+rec_data[rec_lth];
```

```
if(rec_lth==3)
 {txdlne=0;EX0=0;
 sum11=sum11-rec_data[3];
 rec11_ok=1;} //数据类型 1 接收完毕
else
 {rec_lth++;EX0=1;txdlne=1;} //打开外部中断 0,允许
MCU1 继续发送数据
else if(rec_data[0]==0XDE)
 {sum22=sum22+rec_data[rec_lth];
 if(rec_data[rec_lth]==0XDF)
 {txdlne=0;EX0=0;
 sum22=sum22-rec_data[rec_lth-2]-0X76;
 rec22_ok=1;} //数据类型 2 接收完毕
else
 {rec_lth++;EX0=1;txdlne=1;}
else if((rec_data[0]>=0XF0)&&(rec_data[0]<=0XF7))
 {switch(rec_data[0])
 {case 0XF0:
 X_coor=X_coor+1;Z_coor=Z_coor+1;// X 坐标
值+1,Z 坐标值+1
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF1:
 X_coor=X_coor+1;Z_coor=Z_coor-1;// X 坐标
值+1,Z 坐标值-1
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF2:
 X_coor=X_coor+1; //X 坐标值+1,Z 坐标值不变
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF3:
 X_coor=X_coor-1;Z_coor=Z_coor+1; // X 坐标值-1,Z 坐标值+1
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF4:
 X_coor=X_coor-1;Z_coor=Z_coor-1;// X 坐标
值-1,Z 坐标值-1
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF5:
 X_coor=X_coor-1; //X 坐标值-1,Z 坐标值不变
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF6:
 Z_coor=Z_coor+1; // X 坐标值不变,Z 坐标值+1
 rec_lth=0;EX0=1;txdlne=1;break;
 case 0XF7:
 Z_coor=Z_coor-1; // X 坐标值不变,Z 坐标值-1
 rec_lth=0;EX0=1;txdlne=1;break;}}
else
 {resendlne=0;EX0=1;txdlne=1;}}
```

2.3 汉字、字符的显示及字库的生成

2.3.1 汉字或字符的显示

为方便控制 MCU2 完成各种显示功能,用 C 语言编写了字符或汉字显示的底层接口函数:

```
#define uchar unsigned char
void display_word (uchar w_width,uchar w_high,uchar px,uchar
py, uchar *word);
```

display_word 函数在液晶显示屏的横坐标为 px,纵坐标为 py 的地方开始显示一个宽为 w_width,高为 w_high 的字符或汉字,word 指针指向所要显示的文字或汉字的编码。其中液晶显示屏的坐标定义如下:以显示屏的左上角第一个像素为坐标

原点,横轴为向右水平方向,以字节为单位(最大为 40);竖轴正方向向下,以像素为单位(240 个像素)。通过该函数,实现了与显示屏底层操作的封装。具体的应用程序在需要显示时只需调用该函数即可。例如在显示屏水平显示“编辑”(每个字为 16×16 像素)时的程序如下。

```
display_word(2,16,6,20,BIAN);//在横轴第(6×8)像素,竖轴第 20 像素处开始显示 16×16 点阵的“编”(BIAN)
```

```
display_word(2,16,8,20,JI);//在横轴第(8×8)像素,竖轴第 20 像素处开始显示 16×16 点阵的“辑”(JI)
```

2.3.2 字库的生成

由于 SED1335 内含的点阵字符是 5×7 点阵的,显示的字太小,加之没有汉字字符,因此,必须建立系统所需显示汉字的字库(字模数据)。设计过程中,采用专用了字模提取软件来生成所要显示的文字的编码。由于编码以字节为单位,因此,当横向取模时,宽度(以像素为单位)也必须是 8 的整数倍;当纵向取模时,高度(以像素为单位)必须是 8 的整数倍。例如,汉字“编”的编码(横向取模)如下:

```
unsigned char code BIAN[ ]=  
{ /* -- 文字: 编 -- */  
/* -- 宋体 12; 此字体下对应的点阵为:宽×高=16×16 -- */
```

```
0x10,0x18,0x23,0x22,0x4A,0xFB,0x12,0x23,0x7B,0x03,0x05,0x35,  
0xC5,0x09,0x11,0x00,  
0x80,0x40,0xFC,0x04,0x04,0xFC,0x00,0xFC,0x54,0x54,0xFC,0x54,  
0x54,0x54,0x0C,0x00,};
```

在程序区内建立系统所需的汉字字库后,当需要在显示屏上某一位置显示汉字时,只要如上所例,调用显示子程序 display_word 即可,由程序逐字节地向液晶显示区相应单元写入汉字的编码,从而在显示屏上显示出汉字或字符。

3 结束语

根据文中所述的设计思路及相关的具体方法设计的液晶显示模块经调试,实现了下述功能:不同界面的切换显示;编辑状态下,单个字符及数据块的显示(每行可显示 40 个字符,可显示 12 行),编辑窗口的上下滚屏,光标在字母之间的左右移动;运行状态下,坐标值及速度等的实时显示;“暂停”,“急停”,错误标号的显示等等。这些功能的实现,大大提高了原数控系统的人机对话功能。

参考文献

- 1 李维,郭强.液晶显示应用技术.北京:电子工业出版社,2000

[收稿日期:2003.9.25]