View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Xiamen University Institutional Repository

基于 PIC18F6622 的人机界面设计

Design of Human Machine Interface Based on PIC18F6622

陈国鹏 （厦门大学信息科学与技术学院，福建 厦门 361005）

摘 要

介绍了如何使用 PIC 单片机 PIC18F6622 和液晶模块 MTG-S32240 进行显示控制系统的设计，分析了其硬件平台的设计要点，给出了清晰的人机界面软件设计架构，介绍了其中最主要的多级菜单的设计方法。设计的显示控制系统在实际系统中运行稳定可靠，证明了设计方法的正确性。

关键词：PIC 单片机，MTG-S32240 液晶模块，人机界面，多级菜单

Abstract

This paper presents how to design Display-Control system based on PIC microcontroller PIC18F6622 and LCM MTG-S32240.This paper analyzes the hardware platform of their design features,and gives a clear HMI software design architecture.This paper also introduces the most important multi-level menu design method.The Display-Control system designed in this paper is running steadily and reliably in the real system,which proves the correctness of this method.

Keywords:PIC Microcontroller,MTG-S32240 LCM,HMI, Multi-level menu

本文所论述的设计即是空压机变频控制系统的人机交互子系统——终端显示控制系统部分。采用 PIC 系列的PIC18F6622 作为系统的显示控制芯片。采用 SED1335 控制芯片的大型图形点阵式液晶模块 MTG-S32240，介绍了 PIC 单片机与液晶模块相结合实现终端显示的硬件接口电路，并在介绍按键与液晶菜单的调用关系的基础上，完整论述了人机界面的软件设计方案。

1 系统硬件设计

1.1 PIC18F6622 及液晶模块 MTG-S32240 简介

PIC18F6622 是 Microchip 公司生产的一款低功耗，高性价比的 8 位单片机。该型号的单片机硬件系统设计简洁，采用了指令总线和数据总线分开的哈佛双总线结构以及精简指令集 RISC 技术。

MTG-S32240 是台湾微端科技股份有限公司生产的大型图形点阵液晶模块，由 320×240 点阵构成。模块内置的 SED1335 控制器是整个显示模块的核心，是一种能够分层显示文本和图形的、功能强大的 LCD 控制器，具有自身的时钟信号，并提供了一套完整的指令系统，实现对液晶的驱动和控制，使得单片机与模块的接口电路设计更加的简单方便。

1.2 系统硬件组成

系统硬件平台主要由 MCU、液晶模块、键盘、通信接口，实时时钟等几个部分组成，系统框图如图 1 所示。

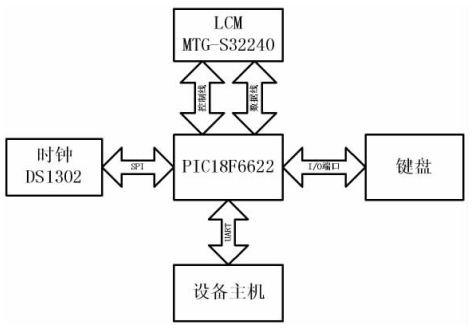

图 1 系统硬件框图

系统实现的人机交互功能：用户通过键盘操作、液晶菜单显示提示，设置设备主机运行所需的参数，再通过 UART 通信将这些参数传送到设备主机上；同时通过 UART 通信，终端显示系统可以查询、获得设备主机运行的工况参数以及故障、警告等信息并将其显示在液晶屏上。实时时钟用于系统的时间统计以及记录，如易损配件的使用时间，系统的运行时间以及故障的发生时间等等。

在整个硬件平台中，单片机与液晶模块的接口电路是整个终端显示系统的核心。由于 MTG-S32240 液晶模块内置了功能较为强大的 SED1335 控制器，MCU 访问 SED1335 时不需判其“忙”，SED1335 随时准备接收 MCU 的访问并在内部时序下及时的把 MCU 发来的指令、数据传输就位，这样配合 PIC 单片机强大的 I/O 口驱动能力，可以直接将单片机的 I/O 口线直接与液晶模块的数据线和控制线相连，无需另加驱动芯片。但有几点需要说明：由于 SED1335 支持 8080 系列和 6800 系列两种类型的单片机，模块的引脚 18 即用于选择哪种型号的单片机，而在本设计中 PIC 单片机属于 8080 系列单片机，所以在电路设计中要将引脚 18 置为低电平。引脚 19 至引脚 26 等 8 个引脚属于触摸屏的接口引脚，本设计在硬件电路上也预留了接口，方便今后的功能扩展。单片机与液晶模块的接口电路图如图 2 所示。

本设计的键盘总共有九个按键，即上、下、右、确认、返回、启动、停机、加载、卸载，前五个按键通过相互配合及功能复用，用于液晶显示菜单的操作以及数据参数的修改等，后四个按键则功能比较单一，用于对设备主机的操作。由于 PIC18F6622 单片机的 I/O 资源比较丰富，且每个 I/O 均可以编程设置其输入或输出，因此设计采用的键盘，直接将按键连到 I/O 口上面。实时时钟采用 DALLAS 公司生产的串行接口实时时钟芯片 DSI302。该芯片具有体积小、功耗低、占用 CPU 的 I/O 口线少等主要特点，还可对时钟芯片备份电池进行涓流充电，应用广泛。单片机与设备主机之间使用标准的 UART 串口通信，硬件接口电路比较简单成熟，加一片 TTL 到 RS232 的电平转换芯片即可。

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

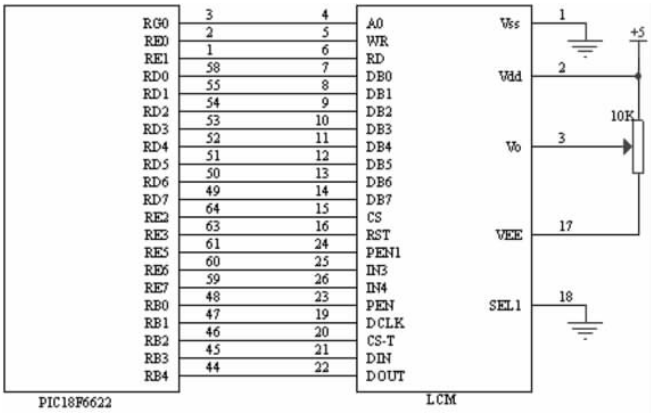

图 2 单片机与液晶模块接口电路

2 系统的软件设计

2.1 初始化液晶显示屏

向液晶模块发送数据和命令的驱动程序是系统显示软件的基础。相比于单片机与液晶模块简单的硬件接口电路,单片机对液晶模块进行驱动的软件程序则稍显复杂。在设计发送数据和命令的驱动程序时,必须严格遵循 SED1335 控制器对 8080 系列 MCU 的操作时序,通过加合适的延时来完成时序上的匹配。发送命令和数据的驱动程序如下。

```
void WRCmd(uchar lcd_cmd)
 {
    LATD=lcd_cmd;
        LCD_A0=1;
LCD_WR=0;
    Nop();
    LCD_WR=1;
Nop();
 }
void WRData(uchar lcd_data)
 {
    LATD=lcd_data;
LCD_A0=0;
    LCD_WR=0;
        Nop();
        LCD_WR=1;
    Nop();
 }
```

为了对液晶模块进行进一步的操作,依据 SED1335 控制器的指令集,向液晶模块发送合适的参数命令和数据,对液晶模块进行初始化设置,使液晶屏按照正确的方式进行显示。液晶屏初始化的流程图如图 3 所示。

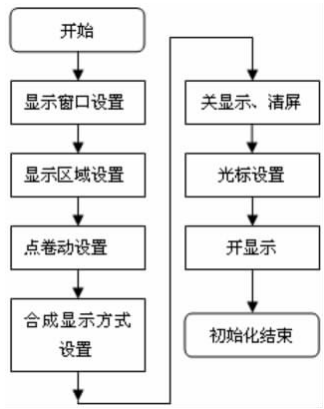

图 3 液晶屏初始化流程图

本设计中,将汉字、数字、英文字母的显示都看成是图形的显示处理,所以显示之前需要通过字模提取软件获得相关“图形”的信息,建立字库,再通过调用以下函数即可以完成任意“图形”在液晶屏上指定位置的显示。

```
void WRCHAR (uchar Layer, uchar X, uchar Y,
uchar rom * HZ,  uchar W, uchar H)
{// Layer:层数; X, Y:坐标位置;
//*HZ:字模指针;W, H:宽、高
uchar i, counter, col;
 col=W/8;
for(i=0;i<col; i++)
      {      //设定光标位置
SetCursor(Layer, X+i, Y);
          WRCmd(0x4F);
          WRCmd(0x42);
          for(counter=0;counter<H;counter++)
                WRData(HZ[i+counter*col]);
      }
}
```

2.2 菜单按键数据结构

人机界面系统中最核心的部分,就是多级菜单的软件设计实现。以往的菜单结构大多采用两个全局变量记录当前菜单的所在层数和所在列数,判断用户键值以修改层、列全局变量,用 switch/case 语句判断当前的层、列全局变量以执行相应的代码,这样的函数架构使得菜单函数代码庞大,结构复杂,程序可读性差,除了代码开发者,其他人很难看懂程序。更谈不上其后续的开发。

为了避免这些弊端,本系统软件对各个菜单项进行编号,将该编号作为菜单的状态索引号,只要根据该索引号就可以唯一确定菜单的当前状态,并根据菜单项与用户按键的相互关系设计了菜单按键的数据结构。

```
typedef struct
      {
      unsigned int Key_MenuIndex;          //当前状态索引号
 unsigned int Key_PressOk;          //按下"回车"键时转向的状态索引号
 unsigned int Key_PressEsc;          //按下"返回"键时转向的状态索引号
 unsigned int Key_PressDown;          //按下"向下"键时转向的状态索引号
 unsigned int Key_PressUp;          //按下"向上"键时转向的状态索引号
      unsigned int Arrow_X;          //ARROW X, 用于指示当前菜单项的箭头的显示坐标(X,Y)
      unsigned int Arrow_Y;          //ARROW Y
      }MenuKeyStruct;
```

该数据结构体现了菜单选项的嵌套以及菜单选项移动与按键的关系:首先对各个菜单项进行编号,并将该编号作为菜单的状态索引号,只要根据该索引号就可以唯一确定菜单的当前状态。此外,在多级菜单的界面中,要实现菜单的嵌套和菜单选项的移动,还需要与按键的相互配合。该数据结构设计了响应不同的按键后状态索引号的改变。该结构体的最后两个成员变量标识用于指示当前菜单项的箭头的显示坐标位置。

在定义了菜单按键数据结构的基础上,定义该结构体类型的数组变量 MenuKeyStruct MenuKey [MAX_ITEM],将全部的菜单选项的嵌套移动关系体现在这个数组中,再定义全局变量 CurMenuID 记录当前状态索引号,当有按键时只要查询结构体数组变量修改状态索引号即可。以图 4 为例说明:当前状态索引号 CurMenuID 指向“厂家参数”

图 4 菜单显示

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

这一菜单项，当用户有按键输入时，只要查询数组元素 MenuKey [CurMenuID]的对应按键的响应成员变量，然后相应地修改 CurMenuID 的值即可以实现菜单选项的移动和转向。这样在软件设计时就可以大大节省时间和精力，当有菜单项增加或减少时，只要修改结构体数组变量 MenuKey [MAX_ITEM]就可以了。由于菜单按键结构的设计，程序结构简单，具有很强的可读性。

2.3 显示的数据的数据结构

在人机界面中，除了菜单文字的显示外，还要有数据的显示，这包括反映主机设备信息的数据的显示、保存，以及可设置的参数的显示、保存及修改等，这是人机界面设计的另一个难点。为了达到数据显示与文字显示的分离，本文设计了显示数据的数据结构。

```
typedef struct
    {
    uchar X;                    //显示数据的 X 坐标
    uchar Y;                    //显示数据的 Y 坐标
    uchar NDigit;               //数据的位数
    uchar dotpos;               //数据小数点的位置
    unsigned int SourceAddr;    //数据存放的内存地址
}DataDispalyStruct;
```

本设计的数据流向图如图 5 所示。终端显示控制系统系统复位后，将保存在 EEPROM 中的数据复制到 RAM 中，并将参数通过 UART 发送给设备主机；系统通过 UART 查询设备主机的工况信息，并保存在 RAM 当中，若是故障信息，则一并写到 EEPROM 中；LCM 要显示数据时再直接从相应的 RAM 中读取显示；当参数设置修改时，按“确认”键修改参数后，将 RAM 中的参数写到 EEPROM 中，确保数据的同步性。采用这样的数据流向设计，主要是由于 RAM 的读写速度比较快，EEPROM 的读写速度较慢的缘故，同时由于 EEPROM 的使用寿命有限，须减少对 EEPROM 进行频繁的读写操作；这里 RAM 起到了一个缓冲区的作用。

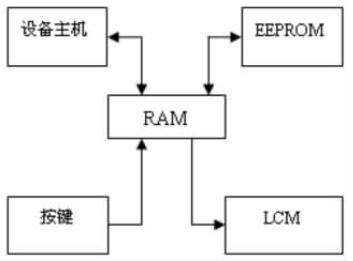

图 5 数据流向图

2.4 系统软件流程

人机界面设计的目标是用户能在菜单方式下进行交互。菜单按键的结构体设计使得按键响应与液晶显示的关系简单明了，显示数据的结构体设计则实现了程序与数据的分离。整个系统软件采用模块化设计，系统软件流程图如图 6 所示。

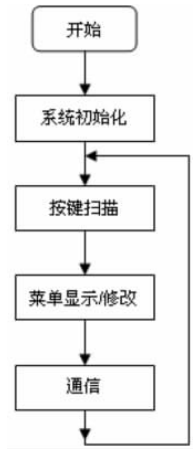

图 6 系统主程序流程图

3 结束语

采用菜单按键的数据结构以及程序和数据分离的思想，使得在修改原有的窗口或增加新的窗口时，不改变程序原有的结构，这对开发人员有一定的借鉴意义。据此设计的终端显示控制系统，在实际的空压机变频控制系统中运行稳定可靠，很好地完成了人机信息交互的任务。该硬件平台还预留了触摸屏的扩展接口，便于今后的功能扩展。

参考文献

[1]Microchip Technology Inc. PIC18F8722 Family Data Sheet. 2004
[2]Microtips Technology Inc. LIQUID CRYSTAL DISPLAY MODULE. 2006
[3]Seiko Epson Corporation. SED1335 Series LCD Controller ICs Technical Manual. 1999
[4]陆铮，罗嘉.单片机 C 语言下 LCD 多级菜单的一种实现方法[J].工矿自动化，2006(1)：14–15
[5]张皆喜.PIC 系列单片机 C 语言编程与应用实例[M].北京：电子工业出版社，2008

[收稿日期：2009.4.19]

(上接第 3 页)

```
{
    packet.TagList.Add( tag );
}
```

3)实现“写数据包”

重载 WriteTag(SampleTagPacket packet)方法。

注意：需要区分于 WriteTag(IoTag tag)方法。

该方法和设备通信协议相关，不给出具体代码实现。

```
protected override bool WriteTag( SampleTagPacket packet )
{
    return base.WriteIoTag( packet );
}
```

3.4 驱动程序的调试

最后在完成驱动程序的相关编写工作后，在 Microsoft Visual Studio 调试环境下首先生成解决方案，在 Debug 的模式下进行调试。也可以通过引用外部指定调试工具进行调试。

4 结束语

本文通过利用面向对象的编程技术，在.NET Framework框架操作平台下完成了一种通用组态软件驱动程序功能框架的开发工作，相关开发方法为接入组态软件的各种类型的 I/O 设备设计了一种通用的设备通信驱动程序框架，可以方便灵活地开发设备驱动程序，使系统拥有非常好的可扩展性，既满足了中小企业的特殊需求，也保证了企业过程控制系统的可靠性，分包数据处理技术提高了系统的通信效率。

参考文献

[1]Li G,Ying J,Wu M H.A configuration software system for industrial monitoring and controlling.Information Acquisition, 2004. [C] Proceedings.International Conference,Hang –zhou, 2004:466–470
[2]欧金成，欧世乐，林德杰，等.组态软件的现状与发展[J].工业控制计算机，2002，15(4)：58–61
[3]Karli Watson,Christian Nagel.C# 入门经典[M].北京：清华大学出版社，2006
[4]李贺斌.监控组态软件中设备驱动程序开发平台的研究与实现[D].秦皇岛：燕山大学，2006
[5]王亚民，郝建领.DCS 组态软件体系结构及其数据交换标准研究[J].计算机工程，2006，32(1)：110–112
[6]Christian Nagel.C# 高级编程[M].4 版.北京：清华大学出版社，2006

[收稿日期：2009.3.31]

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net