

# 基于有源滤波器的串口通信设计与实现

蔺海艳<sup>1</sup>, 刘 海<sup>1</sup>, 王银照<sup>2</sup>

(1. 山东大学威海分校, 山东 威海 264209;

2. 山东滨州供电公司, 山东 滨州 256610)

**摘 要:** 介绍了基于 DSP 的并联有源滤波器与上位机的 RS-232 串口通信设计, 给出了 DSP 串行通信的硬件接口电路和通信协议, 并完成了串口通信的软件编程。下位机使用 C 语言和汇编语言混合编程, 上位机采用 Visual C++ 设计界面, 并用串口控件 MSComm 编写串口程序, 实现上位机对有源滤波器数据的采集、显示、处理和存储。调试结果表明, 该串口通信设计可以实现上位机对有源滤波器的监控。

**关键词:** 有源滤波器; 串口通信; Visual C++; MSComm

中图分类号: TP29

文献标识码: A

文章编号: 1674-7720(2012)09-0032-03

## Design of serial communication based on APF

Lin Haiyan<sup>1</sup>, Liu Hai<sup>1</sup>, Wang Yinzhao<sup>2</sup>

(1. Shandong University at Weihai, Weihai 264209, China;

2. Power Supply Company at Binzhou, Binzhou 256610, China)

**Abstract:** The paper introduces the serial communication between active power filter based on DSP and PC. The hardware interface circuit of serial communication, software program and communication protocol are given. C and assembly language are used in lower computer program. Host computer adopts Visual C++ to design man-machine interface and uses MSComm ActiveX to realize serial communication program, which completes the data collection, display, processing and storage. The results of test show that PC can monitor the active power filter completely by the communication program.

**Key words:** active power filter; serial communication; Visual C++; MSComm

在工矿企业中,非线性电力负荷的大量增加使电网谐波污染日益加剧。而有源滤波器 APF (Active Power Filter) 能够有效地滤除电网谐波,提高电能质量,已成为谐波补偿的一种发展趋势。随着电网智能化的提高,仅仅完成谐波的滤除是不够的,还需要将有源滤波器所处理监控的一些动态参数(如电压、电流、谐波等)上传给上位机,以实现电网数据的实时监测、分析和处理。串口通信是上位机与下位机数据交换常使用的方法之一,它具有数据传输可靠、连线少、成本低和抗干扰强等优点<sup>[1]</sup>。因此,在本监控系统中本文采用 RS-232 串行通信方式完成上下位机的数据交换。如果需要长距离传输,可以加 Modem 使用电话网或是直接使用无线通信网络。

### 1 有源滤波器与上位机通信的硬件设计

有源滤波器设计采用并联型拓扑结构,电网电流经

电流互感器按比例转换成低电流,经 A/D 转换器采样后读入 DSP 控制模块,DSP 运用瞬时无功功率理论计算出补偿电流的指令信号,再利用该信号和实际的补偿电流设计滞环比较器,产生 PWM 信号控制主电路生成合适的补偿电流回馈给电网,以消除电网谐波。

为了完成数据的实时传输,DSP 控制模块通过扩展异步通信芯片 TL16C750 实现系统的高速串行通信。TL16C750 具有集成度高、使用方便及兼容性好等特点,可由软件设定 16 B 或 64 B 的 FIFO,波特率最高可达 1 Mb/s,具有可编程的串行数据发送格式<sup>[2]</sup>。它的片内寄存器选择信号线 A0~A2 和数据线 D0~D7 直接与 DSP 相连;片选信号、读写信号、接收准备好和发送准备好信号等控制信号通过 CPLD 逻辑电路与 DSP 相连,以便完成对 TL16C750 的控制。TL16C750 接收上位机传来的命

令,并将下位机并行数据转换成串行数据上传给上位机。

由于 RS-232 端口的电气特性规定逻辑 1 是  $-3\text{ V} \sim -15\text{ V}$ , 逻辑 0 是  $+3\text{ V} \sim +15\text{ V}$ , 与 TTL 电平不兼容。因此,本设计采用电平转换芯片 MAX3232,完成 TTL 电平到 RS-232 电平的转换。然后将 MAX3232 的 T1out 和 R1in 接到 PC 的串口上。有源滤波器与上位机通信的硬件原理框图如图 1 所示。

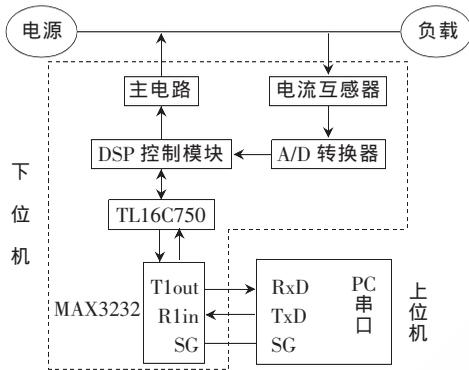


图 1 有源滤波器与上位机通信的硬件原理框图

## 2 通信协议设计

通信协议是指通信双方的一种约定,它规定了数据接收方和数据发送方需共同遵守的某种数据格式、同步方式、传送速度、纠错方式等,是串口通信的基础<sup>[3]</sup>。本设计采用主从式、半双工通信方式,波特率选择  $9\ 600\text{ b/s}$ , 1 bit 偶校验, 8 bit 数据位, 1 bit 停止位, 并采用十六进制数据格式来表示指令代码和数据,格式如下:

起始字节 地址码 功能码 参数代号 操作数 结束字节

指令代码和数据都采用定长格式,起始字节为 A5,结束字节为 5A,没有参数的字节用 FF 填充。其中,操作数为 2 B,其他为 1 B。

地址码为 1 B,在多机通信中,每个下位机有唯一的地址码,以响应与该地址相应的数据信息。由于本机还处于试验调试阶段,本文采用最简单的单机通信,即只有一台下位机,该下位机地址码为 01H。

功能码包括查询命令、读命令和写命令,每个功能码是 1 B,由上位机发送给下位机,告诉下位机执行何种动作。本通信中查询命令、读命令和写命令功能码分别为 25H、26H 和 27H,分别告诉下位机执行查询从机、读数据和写数据操作。

首先,上位机向下位机发送查询从机命令,命令格式为:A5+地址码+查询命令(25H)+FF FF FF 5A。下位机接收到命令后,上传特定数据给上位机,以完成上下位机的握手。

通信指令包括一条读指令和一条写指令。

读指令格式为:A5+地址码+读命令(26H)+FF FF FF 5A。

返回数据为电压、电流、谐波电流、小数点位置。

写指令格式为:A5+地址码+写命令(27H)+参数代号+被写入参数值+5A。

当上位机将读指令发送给下位机时,下位机将采集的电压、电流、谐波电流和小数点位置上传给上位机。小数点位置用 1 B 表示,其他参数由 2 B 组成,低字节在前,高字节在后。

写指令中的参数代号是指有源滤波器中需要修改的参数,如 01H 表示波特率,02H 表示小数点位置。被写入参数值用 2 B 表示,低字节在前,高字节在后。比如,写指令为 A5 01 27 01 80 25 5A,则表示要将下位机的波特率改为  $9\ 600\text{ b/s}$ 。第一个 01H 表示从机地址,27H 表示写命令,第二个 01H 表示需要修改波特率,要修改成的数据是 2580H,十进制为  $9\ 600$ 。

## 3 串行通信软件设计

串口通信软件设计包括下位机的 DSP 串口编程和上位机的程序设计两部分。下位机的串口编程主要由串口初始化和通信协议组成,它接收上位机传来的指令,并根据指令作出相应处理。上位机软件设计主要包括串口配置和后续数据处理,它向下位机发送命令,并将接收到的数据进行处理、运算和保存。

### 3.1 下位机串口软件设计

下位机的串口编程首先要完成串口的初始化,也就是设置操作所需要的参数,如设定串行通信中字符的格式、波特率和工作方式等。在本系统中,异步通信芯片采用 TL16C750,它内部有 11 个寄存器,占用 7 个 I/O 口地址。DSP 工作在微处理器模式,选择 TL16C750 端口地址为  $4000\text{H} \sim 4006\text{H}$  ( $A15=0, A14=1$ ),并通过控制地址线 A0、A1、A2 和线路控制寄存器的除数寄存器选通位 D7 来选通不同寄存器,以便进行读写操作。

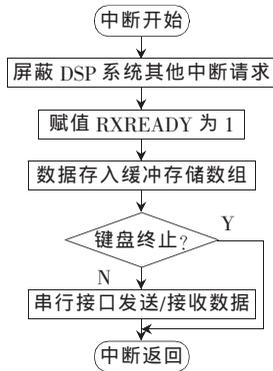
在异步串行通信中,一方的发送数据波特率一定要和对方的接收数据波特率相同<sup>[4]</sup>,本系统设置波特率为  $9\ 600\text{ b/s}$ ,又因 TL16C750 的晶振是  $14.745\ 6\text{ MHz}$ ,根据“发送波特率=输入时钟频率÷除数寄存器内容÷16”可得波特率除数寄存器(低位地址是  $4000\text{H}$ ,高位地址是  $4001\text{H}$ ,选择该地址时,线路控制寄存器的  $D7=1$ )内容为 96,转化成十六进制是  $0060\text{H}$ 。

线路控制寄存器地址为  $4003\text{H}$ ,它既是可写的也是可读的,该寄存器中的内容规定了通信中一帧的格式。本系统选择通信格式为 8 bit 数据位, 1 bit 停止位, 1 bit 偶校验,因此线路控制寄存器的值为 1B。

系统工作中断方式下,允许接收错、发送寄存器空和接收数据就绪中断,因此中断允许寄存器(地址为  $4001\text{H}$ ,选择该地址时线路控制寄存器的  $D7=0$ )的值为  $07\text{H}$ 。

将 TL16C750 的接收准备好信号(RXRDY)和发送准备好信号(TXRDY)与 DSP 系统的外部中断 1 ( $\overline{\text{INT1}}$ )和中断 3 ( $\overline{\text{INT3}}$ )相连。当 DSP 接到串口中断时,先屏蔽掉

系统的其他中断请求,然后将RXREADY赋值为1,以便用来判断DSP是否完成了此中断响应。如果键盘没有终止串口通信,则将存入缓冲存储器组的数据上传给上位机。串口通信的中断子程序流程图如图2所示。



### 3.2 上位机串口通信软件设计

上位机程序使用 Visual C++ 作为软件开发环境。Visual C++ 在串口通信方面功能很强,不仅操作简单、代码执行速度快,而且界面设计比较好<sup>[5]</sup>。使用 Visual C++ 进行串口开发有多种方法,如 Windows API 通信函数、MSComm ActiveX 串行通信控件、Visual C++ 运行库函数,第三方编写通信类都可以实现串口编程。本设计采用 MSComm ActiveX 串行通信控件来编写通信程序。

MSComm ActiveX 串行通信控件是 Microsoft 公司提供的简化 Windows 下串行通信编程的 ActiveX 控件<sup>[6]</sup>。它提供了事件驱动方式和查询方式两种处理通信的方式。查询方式是通过检查 CommEvent 属性值来查询事件和错误,然后进行相应的处理;而事件驱动方式是在串口发生事件或错误时,产生 OnComm 事件进行相应的处理,它响应及时,可靠性高,因此本文选用事件驱动方式。

首先在 VC 中建立一个基于对话框的应用程序,在该 MFC 项目中插入 Microsoft Communication Control 控件。在使用该控件前,先要对其进行初始化设置,主要包括选择 COM 口以及设置输入方式、波特率、数据位、检验位、停止位、发送缓冲区大小和接收缓冲区大小等。当然不要忘记打开串口,因为在 32 bit Windows 中,串口和其他通信设备都被作为文件进行处理,在使用前必须先将其打开。部分初始化程序如下:

```

m_ctrlComm.SetCommPort(1); //选择 COM1
m_ctrlComm.SetInputMode(1); //输入方式为二进制方式
m_ctrlComm.SetInBufferSize(1024); //设置输入缓冲区大小
m_ctrlComm.SetOutBufferSize(1024); //设置输出缓冲区大小
m_ctrlComm.SetSettings(bc1); //设置波特率、校验位、数据位、停止位。其中 bc1 是从串口参数设置组合框中得到并经 sprintf() 函数格式化后的字符串
if(! m_ctrlComm.GetPortOpen())
    m_ctrlComm.SetPortOpen(TRUE); //打开串口
m_ctrlComm.SetRThreshold(1); //参数 1 表示每当串口接收缓冲区中有多于或等于 1 个字符时将引发一个接收数据的 OnComm 事件
m_ctrlComm.SetSThreshold(1); //参数 1 表示当传输缓冲区完全空时将引发一个接收数据的 OnComm 事件
  
```

```

m_ctrlComm.SetInputLen(0); //设置当前接收区数据长度为 0
m_ctrlComm.GetInput(); //先预读缓冲区以清除残留数据
  
```

初始化完成后,按照通信协议在查询设备和开始采集按钮所对应的函数 Oncheck() 和 Onstart() 中编写发送命令的字符串。为了定时发送读指令,在 Onstart() 函数中添加计时器函数 SetTimer(), 当采集结束时,一定要清除该计时器。函数如下:

```

SetTimer(1, 1000, NULL); //创建一个计时器,每秒发送一次读指令
KillTimer(1); //销毁该计时器
  
```

当接收缓冲器中有数据时,将会触发 OnComm 事件,因此要在 MSComm 控件相应的消息响应函数 OnOnCommMscmm1() 中编写数据处理程序。由于串口通信接收和发送数据类型只能是 VARIANT, 因此要将数据进行格式转换。将接收到的数据转化成 CString 类,处理后将其保存到与之相连接的 SQL Server 2000 数据库的表中,以便在主界面中进行显示和相应数据的处理。

### 4 串口调试

串口属性设计完成后,运行主界面,单击串口设置按钮,进入串口通信界面,运行界面如图3所示。

首先单击串口参数设置按钮,设置串口通信的波特率、检验位、数据位和停止位。波特率选择 9600b/s, 1 bit 偶校验, 8 bit 数据位, 1 bit 停止位。设置完成后单击设备查询,上位机将会向下位机发送设备查询命令,下位机收到查询命令后,将返回相同的数据,如图3中的 a5 01 25 ff ff ff 5a, 就是下位机返回给上位机的数据。上位机接收到该数据,说明上下位机完成握手,数据能在两者之间进行实时传送。

单击开始采集按钮,上位机定时向下位机发送读指令,下位机接收到读命令后,将相关数据上传给上位机。为了更加清楚地查看数据的传输过程,本文在串口对话框中加一个编辑框,用来显示下位机传给上位机的数据,如图3中的 98 08 22 00 09 00 01 就是下位机上传的数据。其中,0898 是电压值,0022 是电流值,0009 是谐波电流值,01 表示保留一位小数,因此转换成十进制电压是 220 V, 电流是 3.4 A, 谐波电流是 0.9 A。

当需要修改下位机相关参数时,将写命令填写到图3最下边的编辑框中,然后单击写命令按钮,即会向下

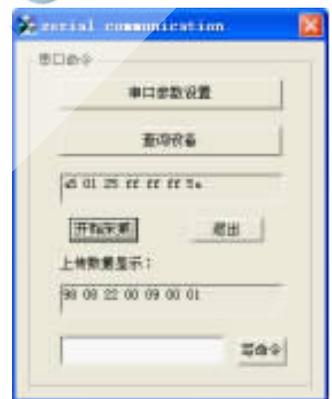


图3 上位机串口通信对话框

位机发送修改命令。

为了实现上位机对有源滤波器的监控,本文设计了有源滤波器与上位机的 RS-232 串口通信,并详细介绍了串口通信的硬件电路、通信协议和软件编程。下位机使用 C 和汇编语言混合编程,上位机使用 Visual C++ 完成界面设计,并编写 MSComm 通信控件。通过串口调试可知,该设计能够完成上位机对有源滤波器的实时监控。

#### 参考文献

- [1] 曹卫彬.C/C++串口通信典型应用实例编程实践[M].北京:电子工业出版社,2009.
- [2] TL16C750 data sheet[DB/OL].[2011-12-16].[http://www.icpdf.com/partnoview.asp?id=733855\\_128973](http://www.icpdf.com/partnoview.asp?id=733855_128973).
- [3] 王晓丽.基于串口通信的摩托车前照灯检测[J].小型内燃机与摩托车,2009,38(2):66-68.

[4] 刘乐善,欧阳星明,刘学清.微型计算机接口技术及应用[M].武汉:华中科技大学出版社,2000.

[5] 龚建伟,熊光明.Visual C++/Turbo C 串口通信编程实践[M].北京:电子工业出版社,2008.

[6] 王春晓,刘海.基于 VB 和 MSComm 的 APF 监控系统设计[J].工业控制计算机,2010,23(12):20-21.

(收稿日期:2012-01-07)

#### 作者简介:

蔺海艳,女,1985年生,硕士研究生,主要研究方向:计算机测控技术。

刘海,男,1964年生,硕士,教授,硕士生导师,主要研究方向:计算机测控技术。

王银照,男,1987年生,本科,助理工程师,主要研究方向:电力调度。

电子技术应用  
APPLICATION OF ELECTRONIC TECHNIQUE  
www.ChinaAET.com