

# 基于 VC++6.0 的读表软件设计

易 驰,文小玲,邹艳华,周 洵

(武汉工程大学 电气信息学院,湖北 武汉 430205)

**摘 要:** 利用 VC++6.0 设计了一种基于串口通信原理的直读表软件,实现了 PC 上位机与 RS485 型/Mbus 型水表、RS485 型气表和电表的串行数据通信。同时,该软件能查抄表的实时数据,完成重新设置表的地址和密码等功能。实际的测试结果证明了软件设计的正确性、有效性。

**关键词:** 串口通信;VC++;抄表;数据

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2014)05-0016-04

## The design of meter reading software based on VC++6.0

Yi Chi, Wen Xiaoling, Zou Yanhua, Zhou Xun

(School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430205, China)

**Abstract:** On the principle of serial communication, the meter reading software based on VC++ 6.0 is programmed, and the serial data communication between PC and RS485/Mbus water meter, RS485 gas meter, and electric meter is implemented. And operators can not only check the real-time data of the meter, but also reset the meter address and password by the using of this software. Test results proved the correctness and validity of the software.

**Key words:** serial communication; VC++; meter reading; data

随着城市化进程的加快、物业管理智能化的推进普及,城市的供水、供气单位越来越多地投入使用新型智能远传水表、气表。目前,RS485、M-bus 远传表已经在居民小区广泛地投入运用。但由于起步晚、技术不成熟等种种因素,安装了自动抄表系统的小区,都或多或少会出现一些问题,如抄不到表读数、表头坏死等。同时,新表的更替和使用也涉及到表地址的改写,表头密码的设置等问题。因此,在抄表维护工作人员电脑上设计一种基于串口通信原理的直读表软件成为一种必然。这将极大地方便水、气表表头模块的检测和抄表系统的日常维护等工作。本文运用 VC++6.0 设计一种能够读取表读数和表地址、设置表地址和表密码,并能测试串口通信情况的直读表软件。

### 1 硬件系统组成原理

硬件系统组成原理框图如图 1 所示,主要由表头、RS232-RS485 转换器、PC 机组成。RS232-RS485 转换器内部包涵了 485 接口电路、光电隔离和 232 接口电路。通过该转换器可以将 485 总线制设备(光电直读表模块)与计算机的串口相连,从而实现计算机与 485 总线制设备

的通信。表头模块内部除了包含单片机、光电隔离电路外,同时还集成了功能很强的可实现全双工通信的 485 接口电路,可以同时接收和发送数据,能够方便地和远端上位机进行通信。



图 1 硬件系统组成示意图

硬件系统各组成部分之间的接线很方便,将 485 型表的 4 根线准确地接在 RS232-RS485 转换器对应的 485 端口上,即 A、B 信号线和 P、G 电源线分别正确连接;同时 232 串口线一端连接转换器,另一端连接电脑,转换器电源口接上 12 V 直流电源,RS485 转换器可由 P、G 线给表头单片机模块上电,指令信号则由 A、B 信号线传递,因此在计算机中用直读表通信软件即可读得表数据。

### 2 软件设计

读表软件的主要功能是通过软件界面窗口按钮的操作,设置串口通信参数、查询表的通信情况和表的

数据信息等,包括上位机与表之间发送、接收指令的显示;查询表的类型,实时读取表的数据、密码、地址编码;修改表的地址编码、密码和开关阀门状态。

本软件依据硬件设备厂家配套的通信协议进行相应开发,此数据通信协议适用于 RS485 型/Mbus 型直读水表、RS485 型气表、RS485 型电表和总线制光电直读直饮水表,采用异步串行方式(一位起始位/八位数据位/两位停止位/无奇偶校验/波特率 2 400),采用 CRC-CCITT(循环冗余码)检验技术,保证通信数据准确、可靠。通信用数据包以“数据包头”开头,“数据包尾”结尾。从上位机到表的发送指令以及从表到上位机的返回指令格式如表 1、表 2 所示。其中,CRC 校验码为有效数据的校验码。CRC 校验数据长度为 2 B,CRC-CCITT 是一个 17 位生成多项式  $G(x)=x^{16}+x^{12}+x^5+1$ ,由它产生的检验码 R 的二进制位数是 16 bit(2 B)。

表 1 发送数据包帧格式

(发送数据包头)	(有效数据)	(CRC 校验码)	(数据包尾)
----------	--------	-----------	--------

表 2 返回数据包帧格式

(返回数据包头)	(有效数据)	(CRC 校验码)	(数据包尾)
----------	--------	-----------	--------

软件采用 Microsoft Communications Control(以下简称 MSComm)控件实现表模块单元与上位 PC 机间的串口数据通信。MSComm 是 Microsoft 公司提供的简化 Windows 下串行通信编程的 ActiveX 控件,它为应用程序提供了通过串行接口收发数据的简便方法<sup>[1]</sup>。MSComm 提供了一系列标准通信命令的使用接口,利用它可以建立与串口的连接,并可以通过串口连接到其他通信设备、发出命令、交换数据以及监视和响应串行连接中发生的事件和错误。而且,MSComm 控件在 VC++、VB、Delphi 等语言中均可使用,利用 MSComm 控件编写串口通信程序时,程序员不必花时间去了解较为复杂的 API 函数。

本软件的设计流程如图 2 所示。

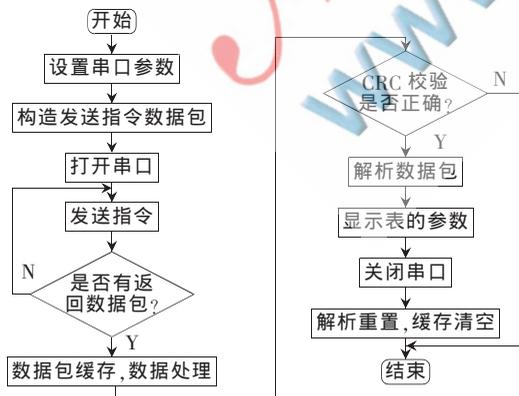


图 2 程序设计流程图

根据流程图,具体设计步骤如下:

(1) 建立项目,打开 VC++6.0,建立一个基于对话框

的 MFC 应用程序 COMTOOL,在项目中插入 MSComm 控件,对于软件界面的设计,在 VC 下运用 dialog 对话框,可以拖拽控件,对控件布局,极为方便<sup>[2]</sup>;

(2) 在对话框中添加发送和接收数据框。向主对话框中添加两个编辑框,一个用于显示接收数据包, ID 为 IDC\_ReceiveEdit; 另一个用于输入发送数据, ID 为 IDC\_SendEdit; 再添加一个按钮,功能是按一次就把发送编辑框中的内容发送一次,将其 ID 设为 IDC\_Send。再打开 ClassWizard -> Member Variables 选项卡,选择 CCommTestDlg 类,为 IDC\_ReceiveEdit 添加 CString 变量 m\_strReceive,为 IDC\_SendEdit 添加 CString 变量 m\_strSend。说明:m\_strReceive 和 m\_strSend 分别用来放入接收和发送的字符数据。

(3) 设置串口参数和打开串口。串口参数设置区的设计是关键环节,在操作软件时,只有正确的设置好串口的配置参数,才能保证串口的正常通信。同时需要设计一个按钮来打开串口,在界面做一个“打开串口”按钮,点击此按钮打开串口时,同时会显示串口参数的配置信息。

串口参数设置区的设计主要是向主对话框界面中添加以上参数相应的编辑框,并将编辑框命名相应的参数,把串口选择、波特率、数据位、停止位和校验位等编辑框拖拽在一个显示框内,命名为“串口设置”,这样方便串口参数的设置和显示,“串口设置”区的界面设计包括:串口选择、波特率、数据位、停止位和校验位的设置。向主对话框界面中添加以上参数相应的编辑框,在工程 COMTOOL 的主程序文件中对串口选择、波特率、数据位、停止位和校验位进行初值定义,程序如下:

```
int BaudRate []={300,600,1 200,2 400,4 800,9 600,14 400,19 200,38 400,56 000,57 600,115 200,230 400,460 800,921 600};
```

```
int ParitySelNum=5;
```

```
CString Parity[]={_T("None"),_T("Odd"),_T("Even"),_T("Mark"),_T("Space")};
```

```
int DataBits[]={5,6,7,8};
```

```
int StopBits[]={1,2};
```

在指令发送、接收编辑框和串口参数配置设计完成后,在软件界面上设计一个用来决定串口通信开和关的按钮,即 Push Button 控制键,将其命名为“打开串口/关闭串口”按钮。在该按钮的处理函数中打开串口,即在相应的处理函数 OnOpenClose() 中添代码即可实现按钮功能。

(4) 构造发送指令。在串行通信的情况下,由于表的基本数据信息会以数据包的形式传送,为了上位机能获取表的数据信息,因此在程序设计起初阶段必须构造相应的发送指令,使表头模块返回对应的数据信息包(即返回指令)。先为发送按钮添加一个单击消息即 BN\_CLICKED 处理函数,打开 ClassWizard -> Message Maps,选择类 CCommTestDlg 选择 IDC\_BUTTON\_MANUALSEND,双击

BN\_CLICKED 添加 OnSend() 函数,并在函数中添加发送数据功能代码,现在可以连接设备测试软件是否能发送和接收指令,配置好串口参数,点击发送按钮,如数据通信成功,进行下一步<sup>[3]</sup>。

不同的功能按钮对应不同的发送指令,将每个按钮赋予一个变量参数,当按钮操作一次,设定的发送指令字符串便赋给变量参数,同时发送指令数据包就发送出去,因此才能完成软件的读取表读数、改写表地址等不同功能。例如“手动读表”按钮的功能代码如下:

```
CString temp;
rec_w=1;
UpdateData(true);
    m_strSend = _T("");
m_strReceive = _T("");
if(m_strStatus=="关闭")
{
    AfxMessageBox("请首先打开串口");
    return;
}
m_strSend=_T("FE FE FE 42 42 42 42 53 FF FF
FF FF 52 2D 3E 45");
ClearBuffer();
temp=m_strSend;
temp=ChangeCharstr2Hexstr(temp);
m_SerialPort.WriteToPort(temp.GetBuffer(temp.GetLength()),temp.GetLength());
m_nSendBytes+=temp.GetLength();
m_strSendBytes.Format("%d",m_nSendBytes);
UpdateData(false);
```

为了实现对表不同的数据信息的读取,即为了发送不同的指令从而获取相应不同的返回数据包,软件界面上会设计不同的功能按钮,如“手工读表”、“地址设置”、“密码设置”等,每个功能按钮会有不同的发送指令,这样就能实现不同的功能。每个按钮的处理函数中会设计相应的发送指令。例如:m\_strSend=\_T("FE FE FE 42 42 42 42 53 FF FF FF FF 52 2D 3E 45")这条广播读表指令以字符串形式赋值给“手动读表”按钮的变量参数 m\_strSend,这样每当操作“手动读表”或其他按钮,就会发送一条指令。

(5)CRC 校验。点击发送指令按钮,每发送一条指令,正常情况下表模块会返回一条指令。在主体程序的设计中,这里添加一个接收返回数据包的功能区,将返回的指令数据包暂时存储,以便对其进行 CRC 校验,判断返回的数据包是否是有效数据。CRC 校验环节是为了测试返回数据包是否是所需要的有效返回指令,即是否是发送指令对应的返回指令,实现 CRC 校验环节的主要程序段如下:

```
CrcA=s[0];
```

```
CrcB=s[1];
CrcC=s[2];
for(i=2;i<j;i++)
{
    CRC(&CrcA,&CrcB,&CrcC);
    CrcC=s[i+1];
}
s[j]=CrcA;
s[j+1]=CrcB;
```

(6)解析数据包程序。在 CRC 校验无误后,才对返回的数据包进行解析,故实现解析功能的程序设计是本设计的重要一环,解析的目的是将返回数据包中的有效数据段转换成表的相应信息,并正确地显示在软件界面上。如实现地址改写的主要解析程序如下所示:

```
CString temp=_T(""),temp2;
GetDlgItemText(IDC_AddrEdit,temp);
UpdateData(true);
rec_w=2;
m_strSend=_T("");
m_strReceive=_T("");
ChangeHexstr(temp,New_data);
for(int j=0;j<4;j++)
    NewSend[13+j]=New_data[3-j]; //获取新地址
for(j=0;j<4;j++)
    NewSend[8+j]=Rec_data[j+5]; //获取旧地址
getCRCCITT(&NewSend[8],result,9); //获取 CRC
NewSend[17]=result[0];
NewSend[18]=result[1];
for(j=0;j<20;j++)
    m_strSend+=DevideHexChar(NewSend[j])+_T(" ");
temp2=m_strSend;
temp2=ChangeCharstr2Hexstr(temp2);
ClearBuffer();
m_SerialPort.WriteToPort(temp2.GetBuffer(temp2.GetLength()),temp2.GetLength());
UpdateData(false);
```

(7)添加表数据显示框和设置按钮。为了清楚反映表的数据信息,故设在软件界面上添加不同的数据显示框,也是设计的一部分。基本通信成功后,和在对话框中添加发送和接收数据框类似,添加表的数据显示框,并对每个显示框设置函数变量,变量的值就是表的数据;添加表的设置按钮,如“地址设置”等按钮,并添加相应的处理函数,在处理函数中编写代码就能实现按钮的相应功能。

(8)解析重置和清空缓存。程序设计中,“解析重置,缓存清空”这一环节的设计是为了在一次通信结束,表的数据正确显示后,将缓存数据包的功能区清空,以便下一次通信返回数据包的缓存,从而使软件能够多次、反复地对表模块发送相同或不同指令,快捷地完成“读

表”、“修改地址”等各种功能。这里在“解析重置”的设计中参数 rec\_w 起到关键作用。例如,在“手动读表”中 rec\_w=1, 同时对应这条发送指令的解析代码中也将 rec\_w=1, 故不同的发送指令会有不同的解析函数段, 这样, 完成一次读表操作后需要将 rec\_w 的值重置为 0, 以便下次 rec\_w 重新赋值进行不同的操作。而对于“缓存清空”的代码设计如下所示:

```
void CCOMTOOLDlg::ClearBuffer() //接收清零
{
    for(int i=0;i<30;i++)
    {
        All_data[i]=0x00;
        Rec_data[i]=0x00;
    }
    m_strReceive="";
}
```

如程序设计流程图所示, 基于这种设计思想, 实际操作时软件的工作流程如下: 程序配置好通信串口参数后, 一直处于等待 OnComm 事件状态。上位机需要读取表的数据时, 通过读表软件发送一条读表指令数据包, 当接收到表模块返回的数据包后放入程序中的缓存功能函数中, 再经过 CRC 校验返回数据包是否正确, 校验成功后才调用解析子函数<sup>[4]</sup>, 对接收指令数据包进行解析, 然后将解析出的表的参数完整显示在读表软件界面上, 最后将解析函数重置并将缓存功能函数中数据清空, 结束本次数据通信。

### 3 测试结果分析

经过调试运行成功后, 设计好的界面如图 3 所示。

从图 3 中可以看到, 使用该软件调试串口通信、读取表数据非常方便、直观。当连接好表等相关设备时, 用户根据实际情况选择相应的串行端口, 设定所需的波特率(在此选用 2 400), 再设置奇偶校验位、数据位、停止位; 点击“串口打开”按钮串口便打开, 再点击“手工读表”便可以进行指令的传送和接收, 并能正确显示表的读数、地址等相关信息; 当在表与上位机通信正常情况下, 点击“地址设置”按钮便可以将表的地址更改; “密码设置”等按钮的操作与“地址设置”类似, 并且实现其相应的功能。同时, 可以通过“清空显示”按钮清除接收



图 3 完善后的软件界面

数据在接收文本框中的显示。

本软件经过测试后可以进行上位机与表之间数据包的发送与接收, 并能正确解析出返回指令中表的实时数据, 如表的读数、表的地址等表的信息清楚地显示在软件界面上; 同时, 表地址的重新设置, 表密码的修改等功能均能实现。软件适用于常见的 RS485 型/Mbus 型水表、RS485 型气表和 RS485 型电表, 对远传抄表系统的维护和安装工作有一定的现实意义。

#### 参考文献

- [1] 封亚斌. 采用串口通信技术实现 Modbus 数据通信[J]. 自动化仪表, 2004, 25(10): 56-58.
- [2] 张新村, 严殊. 基于 ARM 的 Linux 系统下 Qt 串口助手的设计[J]. 软件导刊, 2011, (8): 64-66.
- [3] 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, 2007.
- [4] 宋国清, 刘畅. 基于 BTF340 开发板的 Modbus 从机协议实现[J]. 鸡西大学学报, 2011, 11(1): 44-45.

(收稿日期: 2013-11-05)

#### 作者简介:

易驰, 男, 1988 年生, 硕士研究生, 主要研究方向: 检测技术与自动化装置。

文小玲, 女, 1962 年生, 教授, 硕导, 主要研究方向: 电力系统及其自动化。