

# 数据采集系统串口通信的设计与实现

蒋萍花<sup>1</sup> 张楠<sup>2</sup>

(1. 青岛远洋船员职业学院 图文信息中心 青岛 266071; 2. 青岛大学 自动化工程学院 青岛 266071)

**摘要:** 串口通信因其简单、可靠和高标准化在工业自动化和电子仪器控制中得到广泛的应用。介绍数据采集系统串口通信的设计和实现。首先以 Visual C++ 的编程环境为例,给出了用 MSComm 控件实现计算机与下位机之间的串口通信关键代码设计,介绍了 MSComm 控件的创建和属性的设置后,又从数据的发送和接收 2 个方面详细阐述了其关键代码的实例。然后,进行了通信传输协议的设计,以图形和代码示例详细展示了如何实现数据块的传输。接着给出了通信握手协议的设计,以流程图的形式讲述了上位机与下位机之间如何实现握手的。最后展示了系统运行的结果。还做了思路扩展,提出了实现计算机与下位机之间远程串口通信的 2 种方案。

**关键词:** 串口通信; MSComm 控件; 通信传输协议; 通信握手协议; 远程串口通信

**中图分类号:** TP31 **文献标识码:** A **国家标准学科分类代码:** 510.1050

## The design and implementation of serial port communication of data collection system

Jiang Pinghua<sup>1</sup> Zhang Nan<sup>2</sup>

(1. Qingdao Ocean Shipping Mariners College, Center of library and information, Qingdao 266071, China;

2. Qingdao University, Automation engineering college, Qingdao 266071, China)

**Abstract:** As a simple, reliable and highly standard way of communication, serial port communication is widely used in industrial automation and electrical equipment controller. The paper presents the way to implement serial port communication of a data collection system. Take Visual C++ programming as example, the key code for MSComm control is presented to implement serial port communications between computer and lower computer. It presents how to create and set up the MSComm control properties. Detailed key code design is discussed from sending perspective and receiving perspective. The communication transfer protocol is designed in details with figure illustration and code examples. The communication handshake protocol is designed in details with process presented to show how the handshake does between computer and lower compute. At the end, the running result for the system is shown. The paper also extends its strategy to long distance communications between computer and lower computer. Two designs are mentioned.

**Keywords:** serial port communication; MSComm control; communication transfer protocol; communication handshake protocol; long distance serial port communication

## 1 引言

串口通信因为其简单、便宜、可靠以及控制函数的高度标准化,在一些领域内得到广泛应用。这些领域包括工业自动化系统、科学测量仪器系统、医疗仪器系统等。要实现串口通信,方法很多。一是采用 Windows 提供的标准通信 API 函数,用 CreateFile 来打开串口,用 SetCommSate 函数来设置串口,用 ReadFile 和 WriteFile 函数来对串口进行读写,实现数据传

输<sup>[1]</sup>。二是可采用直接嵌入汇编法,直接对串口进行操作<sup>[2]</sup>。但这种方法容易对计算机的操作系统安全构成威胁,不建议采用。三是利用现有的 ActiveX 控件实现。

介绍的是一个数据采集系统,计算机为上位机,连接一个或多个下位机,下位机连接传感器进行数据采集。上位机是系统的主要控制部分,并显示和分析下位机传回的数据,上位机与下位机的串口通信通过控件实现。首先介绍以 Microsoft Communications Control(MSComm)控件实现

串口通信的编程要点,然后介绍了系统采用的通信传输和握手协议,给出了系统运行的结果,最后就系统的远程拓展应用进行了方案设计。

## 2 MSComm 控件的创建和设置

MSComm 控件在 Visual C++、Visual Basic、Delphi 等语言中均可以使用。

本文 Visual C++ 编程环境为例,具体描述用 MSComm 控件实现串口通信的过程。

### 2.1 创建 MSComm 控件

创建 MSComm 控件一般分为 2 步:

1) 导入 MSComm 控件源代码

可通过菜单 Project → Add to Project → Components and Control,选中 MSComm 插入 Project 即可,可设置控件的 ID 为 ID\_COMMCTRL。此时 Project 中自动添加了 2 个描述该控件的文件 mscomm.h 和 mscomm.cpp。然后为该控件添加相关联的变量,比如 CMSComm m\_Com。

2) 在容器中创建控件

容器是指包含此控件的对话框、文档框等视图窗口。以对话框为例,可在对话框的 OnInitDialog() 函数中下列创建控件的语句。

```
DWORD style=WS_VISIBLE|WS_CHILD;
if (! m_Com.Create(NULL,style,CRect(0,0,0,0),
this,ID_COMMCTRL)) {
TRACE0("Failed to create OLE Communications
Control\n");
return -1; //fail to create
}
```

### 2.2 设置 MSComm 控件属性

MSComm 控件共有 27 个属性,属性的含义在很多参考书中可找到<sup>[3]</sup>。

程序中可通过类 CMSComm 的对象 m\_Com 便可以对串口属性进行设置,以下是典型的控件属性对串口进行初始化的实例:

```
m_Com.SetCommPort(1); //选择 COM1
m_Com.SetInBufferSize(1024); //设置输入缓冲区
的大小,Bytes
m_Com.SetOutBufferSize(512); //设置输出缓冲
区的大小,Bytes
m_Com.SetPortOpen(TRUE); //打开串口
m_Com.SetInputMode(1); //设置输入方式为二进
制方式
m_Com.SetSettings("9600,n,8,1"); //波特率
9600,无奇偶校验,8bit 数据,1bit 停止
m_Com.SetRThreshold(1); //为 1 表示有一个字
符引发一个事件
m_Com.SetInputLen(0); //控件读取缓冲区全部内容
```

## 3 串口数据的发送

任何类型数据都可以用一个二进制字节组来存储。可采用 CbyteArray 类型来存储将要发送的数据,定义一个 CbyteArray 的临时变量,然后采用 SetSize() 函数设置其大小并将相应的字节值存入该数组,最后利用 COleVariant() 转换为 COleVariant 类型,即可利用 SetOutput() 函数发送出去。

代码示例:

```
CByteArray mByteArray;
mByteArray.SetSize(2);
mByteArray.SetAt(0,0x4a);
mByteArray.SetAt(1,0x4b);
m_Com.SetOutput(COleVariant(mByteArray));
```

## 4 串口数据的接收

### 4.1 串口事件的捕捉

MSComm 控件可以采用查询或事件驱动的方法从串行端口获取数据。事件驱动占用系统的资源少,应用比较广泛<sup>[4]</sup>。本文介绍事件驱动法的实现。

捕捉串口事件一般分为 2 步:

1) 添加事件处理函数声明

例如接收串口数据的对话框的头函数为 CrecieveDlg.h,则在其中增加下列语句:

```
protected:
afx_msg void OnCommMscomm();
DECLARE_EVENTSINK_MAP()
```

2) 添加事件与事件处理函数的映射

例如接收串口数据的对话框的源文件为 CRecieveDlg.cpp,则在其中增加下列语句:

```
BEGIN_EVENTSINK_MAP(CRecieveDlg, CDialog)
ON_EVENT(CRecieveDlg, ID_COMMCTRL, 1,
OnCommMscomm, VTS_NONE)
END_EVENTSINK_MAP()
```

这样当有事件(如接收到数据)发生时,程序会调用 OnCommMscomm 函数进行处理<sup>[5]</sup>。

### 4.2 数据接收关键代码

接收数据时采用 GetInput() 函数,返回值是 VARIANT 类型。VARIANT 是结构,它的主要成员包括一个联合体和一个变量。联合体由各种类型的数据成员构成,而该变量则用来指明联合体中目前起作用的数据类型。接收时,先定义了一个 VARIANT 类型的变量 vResponse,利用它读取串口数据后,再通过它获取 parray 的 pvData 指针,就可以访问所需数据。以下的关键代码可以在系统发现接收缓冲区收到大于 RTHreshold 个字符时,函数 RecieveData() 进行调用。

```
VARIANT vResponse;
vResponse=m_Com.GetInput();
BYTE * pB;
```

pB= (BYTE \*) vResponse. parray->pvData;

## 5 通信协议的设计

### 5.1 通信传输协议设计

信息的传输需要可靠性和实时性。对于传感器的数据,更注重它的实时性。为了提高传输的速度和效率,可采用一定的缓冲机制,并采用采用数据块为单元进行传输<sup>[6]</sup>。

可采用如图1所示的格式发送数据块。



图1 通信传输协议

其中字节数为字节1到校验字节的字节数目,用WORD型(2个字节)表示,校验字节为字节数高8位到字节n的所有字节模2加的结果。

例:通信过程中最常用的数据类型就是float型(4个字节大小),下面的程序发送和接收7个字节的数据块,其中头2个字节为此数据块的长度(字节数),中间4个字节存放了一个float的数值,最后一个字节为校验字节。关键代码如下:

```
union {
    WORD Wa;
    BYTE pBb[2];
}WtoB; //利用共用体实现 WORD 向 BYTE 转换
union{
    float fa;
    BYTE pBa[4];
}ftoB; //利用共用体实现 float 向 BYTE 转换
WtoB. Wa=7; //数据块字节数,设为7
CByteArray mByteArray; //存放数据块
//头2个字节为数据块长度7
mByteArray. SetAt(0, WtoB. pBb[0]);
mByteArray. SetAt(1, WtoB. pBb[1]);
ftoB. fa=0.56;
//第2,3,4,5字节存放了一个float数值
mByteArray. SetAt(2, ftoB. pBa[0]);
mByteArray. SetAt(3, ftoB. pBa[1]);
mByteArray. SetAt(4, ftoB. pBa[2]);
mByteArray. SetAt(5, ftoB. pBa[3]);
BYTE CheckByte=0x00; //校验字节先置0;
for(int i=0; i<6; i++)
    CheckByte=CheckByte^(mByteArray. GetAt(i)); //校验字节为前面6个字节的模2加
mByteArray. SetAt(6, CheckByte); //校验字节放在最后
//利用 Mscomm 控件发送
```

m\_Com. SetOutput(ColeVariant(mByteArray));

### 5.2 通信握手协议设计

握手协议是为了提高数据传输的可靠性。特别是在数据块传输之初。

通信双方要设置一些数据流中不常用的字节作为握手信号,方案中采用十六进制中的06、00、15、F0、0F作为握手信号字节。

计算机端在 MSCOMM 控件初试化时,设置一个准备好标志 Ready\_Flag 为假,当端口有数据时,控件自动调用 OnCommMscomm()。若接收到大于 RTHreshold 个字符,则调用 RecieveData(),在 RecieveData()函数中,先判断 Ready\_Flag 是否为真,当 Ready\_Flag 为假时,调用 GetInput()接收数据。此时,如果计算机端收到的第一个字节为06H,即为双方约定好的准备发数据的握手标志时,向下位机端发送00H,同时设置 Ready\_Flag 为真,表示可以开始接收有用数据了。下位机端收到00H后,发送数据块到计算机端口。计算机的 MSComm 控件调用 OnCommMscomm(),当在 RecieveData()中检测到 Ready\_Flag 为真时,直接接收数据块。根据字节数确定数据是否接收完,把接收到的数据进行模2加,与收到的的校验字节比较,相同说明传输无误,可把数据保存为硬盘上的文件,然后把 Ready\_Flag 置为假,向下位机发送0FH,告诉下位机接收正确,等待下一次的接收。若校验出传输有误把 Ready\_Flag 置为假,向下位机发送 F0H,下位机可进行重发。通信握手协议流程如图2所示。

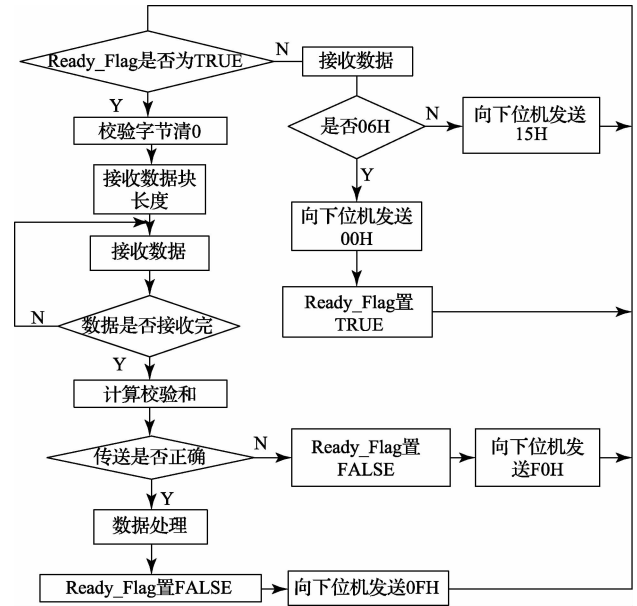


图2 通信握手协议

## 6 系统运行结果

运用上述的通信方法,系统可正确接收到下位机传来的数据,如图3所示。系统可绘制接收到的数据图形,如图4所示。

	T(Seconds)	E(Volts)
1	0.00000	0.23864
2	0.00005	0.23829
3	0.00010	0.23627
4	0.00015	0.23265
5	0.00020	0.22887
6	0.00025	0.22536
7	0.00030	0.22209
8	0.00035	0.21902

图 3 系统接收到的数据

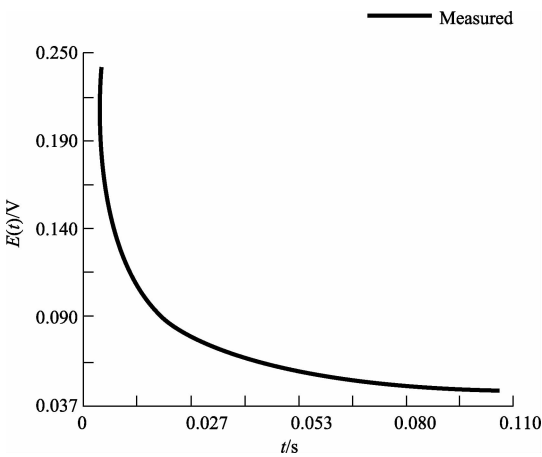


图 4 系统处理接收到的数据

## 7 远程通信方案设计

有时计算机与下位机距离比较远,这时要实现计算机与下位机的远程通信。可采用以下的 2 种方案。

### 7.1 Modem 远程控制

硬件设备可如图 5 连接。

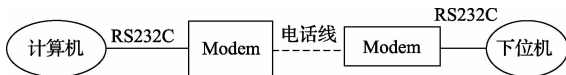


图 5 Modem 远程控制连接

软件设计可以仍然采用在 Visual C++ 中用 MSComm 控件实现。程序中可设计一个界面,让用户输入电话号码,在发送数据时要先进行拨号处理,让 MSComm 控件向串口发送“ATDT”命令,后面接上电话号码即可。如要挂断电话,让 MSComm 控件向串口发送“ATH0”命令即可。在接收数据时,要先进行自动摘机,让 MSComm 控件向串口发送“ATS0=1\n”命令即可。

### 7.2 Internet 远程控制

硬件设备可如图 6 连接。

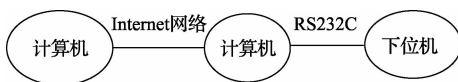


图 6 远程控制连接

软件设计如用 Visual C++ 实现,Internet 网上的 2 台计算机之间的数据传输可采用 Visual C++ 中 Csocket 类或 Microsoft Winsock 控件来实现。以 Csocket 为例,由用户输入对方计算机的 IP 地址,用 Csocket 类中的 Connect 函数建立连接,用 Send 函数传送数据,用 Receive 函数接收数据,用 Close 函数关闭连接<sup>[7]</sup>。Winsock 的编程流程与此类似<sup>[8-11]</sup>。计算机与下位机之间的串口通信继续采用 MSComm 控件实现。把两者结合在一起,就可实现远程计算机通过 Internet 网络对下位机进行控制。

## 8 结 论

介绍了实现串口通信数据采集系统的关键代码和通信协议,系统运行效果良好。这些代码和通信协议具有相似和通用性,很容易移植到别的系统中。还就远程串口通信做了探讨,有待于在具体项目上得到进一步的实践。

## 参 考 文 献

- [1] 丁国兴,高琴,高伟. 基于 Win32 API 的通用串口通信组件的开发[J]. 工业控制计算机, 2013(11): 33-35.
- [2] 吴来杰,李波,金远雄. 基于内嵌汇编语言的 PC 机与单片机串口通信[J]. 煤矿机械, 2005(3): 57-59.
- [3] 王明. Visual C++ 串口通信技术与工程实践[J]. 计算机与网络, 2003(17): 54-55.
- [4] 周永刚. 基于 MSComm 控件实现计算机与单片机间的串行通信[J]. 工业控制计算机, 2013(5): 80-81.
- [5] 章辉,叶建芳,叶建威. 基于 MSComm 控件串口通信的实现[J]. 电子测量技术, 2011, 34(8): 126-129.
- [6] 雪娟. 浅谈计算机通信传输控制技术[J]. 计算机光盘软件与应用, 2014, 17(14): 24-25.
- [7] 张红红,连卫民. 远程监控系统的设计与实现[J]. 计算机与数字工程, 2012, 40(6): 98-100.
- [8] 刘赞. Winsock 技术在网络通信系统中的应用[J]. 西南科技大学学报, 2013, 28(2): 88-91.
- [9] 卢海峰,江朝元,阳小光,等. 基于串口通信的在线监测系统关键技术研究[J]. 仪器仪表学报, 2006, 27(增刊 3): 2043-2044, 2053.
- [10] 李文海,刘勇,王怡莘,等. 串行总线合成仪器设计与实现[J]. 电子测量与仪器学报, 2011, 25(10): 917-923.
- [11] 吕向锋,高洪林,马亮,等. 基于 LabVIEW 串口通信的研究[J]. 国外电子测量技术, 2009, 28(12): 27-30, 42.

## 作者简介

蒋萍花,工程硕士、讲师。主要研究方向为电子,计算机通信、信号处理、软件设计、开发、验证、项目管理。  
E-mail: pjiang9@163.com

张楠,工程硕士,讲师。主要研究方向为电子,通信、计算机、密码学、信号处理、软件设计、医学工程等。  
E-mail: qduzhangnan@126.com