

用 Delphi 实现 Windows 环境下的串口通信

宋宏宇 丁铁夫 庄文 (中国科学院长春物理所六室 130021)

摘要: 阐述 Windows 环境下串口通信的基本原理, 给出 Delphi 实现串口通信的编程实例。实现了单片机与计算机 Windows 环境下的串口通信。

关键词: 串口通信 Delphi 编程

串口通信, 就是通过串行口完成微机与外部设备或微机与微机之间的信息传递。串口程序开发相对简单, 在传输数据量不大、要求速率不高的通信场合得到广泛应用。在 DOS 环境下开发串口应用程序, 一般来说要对 UART(串行通信芯片)进行编程, 调用 INT14 BIOS 中断来实现串口通信。Windows 为用户提供了一种具有设备无关性的操作平台, 使用户对系统硬件资源的操作变得完全透明, 底层操作则由系统完成, 既把硬件资源封装成系统对象。Windows 环境下的串口通信的开发可以通过系统提供的 API 函数来实现。面向对象的开发平台 Borland Delphi 为 Windows 串口程序的开发提供了一个良好的开发环境。

本文以单片机与计算机之间的通信为例来说明 Windows 环境下开发串口通信的基本方法。

1. 与串口通信有关的几个 Windows API 函数

· 打开端口函数

```
function OpenComm ( ComName: PChar; InQueue,
OutQueue: Word): Integer;
```

OpenComm 函数用于打开一个通信设备, 并返回通信设备句柄。

· 端口配置函数

```
function GetCommState(Cid: Integer; var DCB; TD-
CB): Integer;
```

GetCommState 函数用于读取通信端口当前状态。其中 DCB 是一个记录类型变量, 包含与端口通信有关的一些设置, 如通信波特率、数据格式、校验发式等参数。

```
function SetCommState(var DCB; TDCB): Integer;
```

SetCommState 函数用于设置通信端口状态, 设置通信端口的一般过程为, 先读入当前端口设置数据, 然后根据需要修改其中部分参量。

· 端口输入函数

```
function ReadComm(Cid: Integer; Buf: PChar; Size:
Integer): Integer;
```

ReadComm 函数用于从通信端口接收缓冲区读取数据。函数返回值为读入字节数。用户可依此值判断是否接收到数据。

· 端口输出函数

```
function WriteComm(Cid: Integer; Buf: PChar; Size:
Integer): Integer;
```

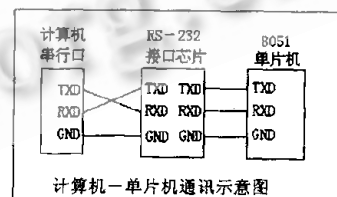
WriteComm 函数用于向通信端口写入数据。

· 关闭端口函数

```
function CloseComm(Cid: Integer): Integer;
```

CloseComm 函数用于关闭通信端口并释放端口所占用的资源。

2. 单片机 - 计算机串行通信电路结构框图



这里使用的单片机为 8051 单片机, 利用单片机自带串口与计算机进行通信, 其中 RS-231 接口芯片完成 8051 与计算机串口之间的电平转换。

3. Delphi 程序实现过程

系统通信计算机一方采用查询工作方式, 单片机一方采用中断工作方式。具体工作过程如下:

(1) 计算机需要与单片机进行通信时, 向单片机发送特定码字;

(2) 单片机收到信号后, 触发中断服务程序, 判断是

否收到指定码字；

(3)如未收到指定码字,返回主程序;收到指定码字,则向计算机发送数据,发送完指定数量数据后返回主程序;

(4)计算机接收数据并将数据存盘。完成一次发送接收周期。

具体程序如下:

```
unit communication ;
interface
//interface 部分程序略///
implementation
{$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject);
var      //定义局部变量
comhandle: Integer;
  retcode: Integer;
  readbyte: Integer;
  COM: Pchar;
  writebuf: Pchar;
  readbuf: Pchar;
  dcb : TDCB;
  F1: TextFile;
temp, readstring: String;
begin
COM:= 'COM2';           //指定串行口 2 为通信端口
writebuf:= 'ABCDE'; //设置请求发送控制字
comhandle:= OpenComm(COM, 1024, 1024); //打开通信端口
retcode := GetCommState(comhandle, dcb); //获取当前端口状态
  dcb.BaudRate := 9600; //设置通信波特率为 9600bit/s
  dcb.ByteSize := 8; //8 位数据位;1 位停止位
  dcb.StopBits := ONESTOPBIT;
  dcb.Parity := NOPARITY; //无校验
if retcode < 0 then MessageDlg('GetCommState error!',
mtInformation, [mbOk], 0);
retcode := SetCommState(dcb);
```

```
if retcode < 0 then MessageDlg('SetCommState error!',
mtInformation, [mbOk], 0);
```

//如果获取设置或者设置串口未成功,返回错误信息

```
WriteComm(comhandle, writebuf, 5); //向单片机发送请求发送控制字
```

```
for i:= 1 to 1000 do begin temp:= ''; end; //延时 20ms
```

```
readbyte:= ReadComm(comhandle, readbuf, 1024);
```

//读入端口数据到 readbuf,并返回读入字节数

```
readstring:= readbuf; //将接收得到数据存入字符串变量
```

```
for i:= 1 to readbyte do
```

```
begin
```

```
readstring:= readstring + (readbuf + i)^;
```

```
end;
```

```
temp:= 'v.dat'; //打开文本文件,将读入数据存盘
```

```
AssignFile(F1, temp);
```

```
FileMode := 1;
```

```
Rewrite(F1);
```

```
Write(F1, readstring);
```

```
CloseFile(F1); //关闭文件
```

```
end;end.
```

4. 实现 Windows 环境下串口通信的其他方法

Windows 环境下的软件开发工具除 Delphi 之外,还有 Visual C++、Visual Basic、Visual Foxpro、Microsoft Java 等可视化开发工具。其中有些软件更加提供了串行通信控件,将串口通信资源封装成独立对象,使设置参数和通信控制变得更加容易。不过,无论使用哪一种工具,开发 Windows 环境下的串行口通信基本都是通过调用 Windows API 函数来实现的。与 DOS 软件相比较,Windows 环境下的串口通信更具设备无关性,用户对串行口的控制变得相对容易,大大加快了软件开发周期。

参考文献

- [1] “RS-232 串口通信协议”
- [2] “Delphi 程序设计”,黄建华等,北京,清华大学 1996.6

(来稿时间:1997年11月)