

PC机与单片机之间的串口通信程序的设计与实现

摘要：本文主要利用MSComm控件实现单片机与PC机的串口通信，重点是软件部分的设计。采用VC++ 6.0软件开发平台，利用MFC编程的思想方法，建立友好的人机交互界面。PC机向单片机发送信号，将单片机接收到信号后向PC机返回的数据在界面中实现动态显示、保存等。该系统具有操作简单，实用性强，可视化程度高等优点，为以后的软件设计打下了一定的基础。

1.引言 在众多集成可视化开发环境中，VisualC++是编程环境的选择，VC++应用程序的开发主要有两种模式，一种是WIN API方式，另一种是MFC方式，传统的WIN API方式比较繁琐，而MFC是对WIN API再次封装，MFC相对于

摘要：本文主要利用MSComm控件实现单片机与PC机的串口通信，重点是软件部分的设计。采用VC++ 6.0软件开发平台，利用MFC编程的思想方法，建立友好的人机交互界面。PC机向单片机发送信号，将单片机接收到信号后向PC机返回的数据在界面中实现动态显示、保存等。该系统具有操作简单，实用性强，可视化程度高等优点，为以后的软件设计打下了一定的基础。

1.引言

在众多集成可视化开发环境中，VisualC++是编程环境的选择，VC++应用程序的开发主要有两种模式，一种是WIN API方式，另一种是MFC方式，传统的WIN API方式比较繁琐，而MFC是对WIN API再次封装，MFC相对于WIN API开发更具备效率优势，开发者只要在局部做简单地修改，即可处理大部分窗口事物。MFC封装了200多个类并且提供了友好的界面环境，具有类管理灵活方便、代码编写和界面设计集成交互操作、可移植性强等优点，与各种数学手段和数据处理、分析算法相结合还可以实现海量数据分析，智能、快速、灵活数据处理，使便携现场检测仪器功能强，灵活度高，速度快，扩展性好。

串口通信具有连续简单、使用灵活方便、数据传递可靠等优点，在数据采集系统、监控系统和通信系统中，得到广泛的应用。在Win32下，用来实现串口通信的编程方式有两种：其一可以调用Windows的API函数，其二可以使用ActiveX控件。调用Windows提供的API函数，可以实现用户自定义功能，自由度高，缺点是这种方法编写复杂，需要清楚地掌握串口通信机制；使用ActiveX控件可以屏蔽底层的通信协议，只需对控件的属性进行设置并添加响应的响应处理函数即可实现串口通信中的数据收发，缺点是不支持用户对属性的修改，只能使用控件提供的功能，在使用时有一定限制本文要实现RS232协议的串口通信，采用ActiveX控件即可满足需求。

2.通信原理

PC机给单片机发送一个55（发其他的数据单片机不响应），单片机通过串口将返回三个数据传给PC:个是AA（作为数据包开包的用的）；第二个数据是一个随机数，第三个数据时EE（作为数据包结束的标志）。

点击接收数据就会在PC机上以16进制、ASCII码、10进制方式显示出来。指示灯LED:红色的是电源灯，单片机处于待机状态时：绿灯亮，黄灯不亮；到单片机收到55数据时，单片机要返回数据的过程中：绿灯亮，黄灯会一闪（指示在发送数据）。

3.硬件

单片机和PC机的波特率都设置为9600kbps.上位机的串口是COM1,其电平是RS232电平，而下位机的电平是TTL电平，所以需要电平转换芯片MAX3232进行电平转换，实现PC机和单片机的通信。在MAX3232芯片上，选用其中一路发送/接收，R2OUT接PIC16f877的RX,T2IN接PIC16f877的TX,T2OUT接PC机的RD,R2IN接PC机的TD,这样就能实现单片机与计算机互联通信。图1为单片机与PC的连接示意图。

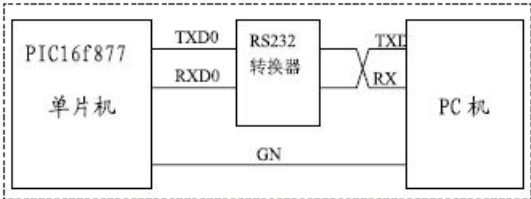


图1 单片机与PC机的连接

4.软件

MSComm通信控件是常用的ActiveX串口控件之一，它提供了一系列标准协议的通信接口，可以连接到其他通信设备（如单片机，以及本系统使用的FPGA），而且在VC++、VB等语言中使用[8].通过建立串行通信，完成命令发送、数据发送/接收、监视和响应在通信过程中可能发生的各种事件和错误。本文选用MSComm控件采用事件驱动机制。一般情况下，事件发生时将产生通知，MSComm控件提供的OnComm事件可以用来捕获并处理这些通信事件。可以通过在OnComm事件处理函数中加入读取串口程序代码和处理数据代码来提高程序响应度及可靠性。图2为本文设计的软件界面。

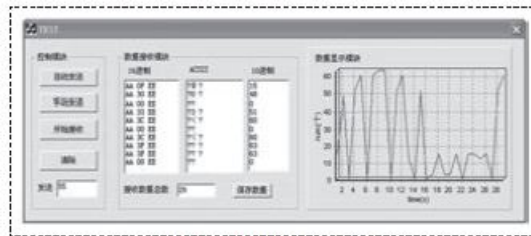


图2 可视化软件界面

(1) 建立应用工程

启动VC++6.0,新建工程名为“TEST”

的对话框应用程序，这时在Resource View的Dialog树状文件夹中会自动生成IDD_TEST_DIALOG对话框。

(2) 在当前的Workspace中插入MSComm控件

选择Project菜单下Add to Project子菜单中的Components and Controls选项，在弹出的对话框中双击Registered Active X Controls项，选择Microsoft Communications Control version 6.0,单击insert按钮即可将通信控件插入该工程，这时添加了类CMSComm及相应文件：MSComm.h和MSComm.cpp。

(3) 添加MSComm控件

在IDD_TEST_DIALOG对话框内添加MSComm控件，通过class wizard添加成员变量m_Com。

(4) 串口初始化

在初始化对话框进行串口初始化，包括端口号、波特率、输入方式等。

(5) 捕捉串口事件

双击IDD_TEST_DIALOG对话框内的MSComm控件，这时在TESTDlg.h中，添加了TESTDlg类消息响应、在TESTDlg.cpp中映射了MSComm控件事件。

(6) 串口读写

当用户输入“55”后，使用函数Set-Output () 向串口写入数据，点击“手动发送”按钮，点击发送。点击“自动发送”按钮，则设置一个Timer定时，1s发送。

```
void CTESTDlg::OnOnCommComm1()
{
    VARIANT variant_inp;
    COleSafeArray safearray_inp;
    LONG len, cnt_data; //存储9个十进制数:00~随机数*EE
    BYTE rxdata[2048];
    CString strtemp1, strtemp2, strtemp3; //存储16进制、字符、十进制数据
    if (m_Com.GetCommEvent() == 2)
    {
        variant_inp = m_Com.GetInput();
        safearray_inp = variant_inp;
        len = safearray_inp.GetOneDimSize();
        for (k = 0; k < len; k++)
            safearray_inp.GetElement(k, rxdata + k);
        for (k = 0; k < len; k++)
        {
            BYTE bt = (char)(rxdata[k]);
            strtemp1.Format("%02X ", bt); //以16进制显示结果
            m_strReceive1 = m_strReceive1 + strtemp1;
            strtemp2.Format("%c ", bt); //以字符显示结果
            m_strReceive2 = m_strReceive2 + strtemp2;
            if (k == len / 2)
            {
                cnt_data = (rxdata[k] / 16) * 16 + (rxdata[k] % 16);
                strtemp3.Format("%d ", cnt_data);
                m_strReceive3 = strtemp3;
                AllNum++;
                itoa(AllNum, AllNum1, 10);
                m_AllNum = AllNum1;
                Time++;
                m_Chart.Series(0).AddXY(Time, cnt_data, HRL, 0);
                UpdateData(FALSE);
            }
            m_strReceive1 += "\r\n";
            m_strReceive2 += "\r\n";
            m_strReceive3 += "\r\n";
        }
        UpdateData(FALSE);
    }
}
```

在进行读操作时，当接收缓冲区有数据时，使用GetInput () 将缓冲区的数据保存到COleSafeArray类型的变量里，以便后续的数据处理。

(7) 存储数据

使用MFC提供的文件操作，将数据保存在TestData.txt文本文件中。CStdioFile是CFile的派生类，适用于文本文件的读写，可以按行读取写入。同时在文件写入数据之前写入本次保存数据的时间，为数据的正确保存提供保障。如图3所示。



图3 数据保存结果

5.程序测试及结果

程序测试过程中，首先利用串口调试助手进行演示，确保单片机能正常工作，串口可以自由接收和发送。然后使用本文设计的软件实现数据的发送显示，结果如图2所示。

6.结论

本文利用VC++6.0中的MSComm控件便捷地实现了PC机与单片机之间的串口通信，显示了ActiveX技术充分的灵活性和易用性。

在Windows环境下开发上位机的串口通信程序，软件界面友好，操作简单，软件设计简单可靠，降低了工作量，准确、有效地实现了数据的采集显示保存。（作者：任晓敏，马龙）