



论文编号:1001-3954(2005)01-0064-02

# PC机与单片机串行通信的一种新方法

徐兴元\* 沈宪章 瓮嘉民

郑州大学电气工程学院 河南郑州 450002

**目** 前关于PC机和单片机串行通信的实现方法,主要由PC机加单片机硬件来完成。在没有单片机的条件下无法完成PC机和单片机的串行通信实验,因此本文提出了一种基于VC#6.0开发环境,利用MSComm控件和Keil uVision2.0串行窗口相结合实现PC机和单片机串行通信的软件仿真的新方法。

## 1 建立基于对话框的项目,并利用MSComm控件编程

### 1.1 建立项目

打开VC#6.0,建立一个基于对话框的MFC应用程序CommTest。

### 1.2 在项目中插入MSComm控件

选择Project菜单下Add To Project子菜单中的Components and Controls...选项,将ActiveX控件中Microsoft Communications Control, version 6.0控件添加到控件工具箱中,如图1所示。

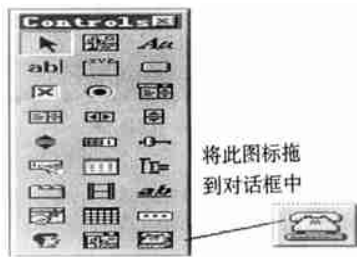


图1 添加通讯控件图

### 1.3 利用ClassWizard定义CMSComm类控制对象

打开ClassWizard -> Member Variables选项卡,选择CSCommTestDlg类,为IDC\_MSCOMM1添加控制变量:mCtrlComm,这时,在对话框头文件中自动加入了//{{AFX\_INCLUDES() #include "mscomm.h"}}AFX\_INCLUDES,然后运行程序。

### 1.4 在对话框中添加控件

向主对话框中添加两个编辑框,一个用于接收显示数据, ID为IDC\_EDIT\_RXDATA,另一个用于输入发送数据, ID为IDC\_EDIT\_TXDATA。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\_BUTTON\_MANUALSEND。再打开ClassWizard -> Member Variables选项卡,选择CSCommTestDlg类,为IDC\_EDIT\_RXDATA添加CString变量m-strRXData,为IDC\_EDIT\_TXDATA添加CString变量m-strTXData。说明:m-strRXData和m-strTXData分别用来放入接收和发送的字符数据。

### 1.5 添加串口事件消息处理函数OnComm()

打开ClassWizard -> Message Maps,选择类CSComm

TestDlg,选择IDC\_MSCOMM1,双击消息OnComm,在弹出的对话框中将函数名改为OnComm,确认。这个函数是用来处理串口消息事件的,每当串口接收到数据,就会触发Oncomm事件,只需在OnComm()函数加入相应的处理代码就能实现串口通讯功能。Oncomm函数的代码如下:

```
void CSCommTestDlg::OnComm()
{
    // TODO: Add your control notification handler code here
    VARIANT variant -inp;
    COleSafeArray safearray -inp;
    LONG len, k;
    BYTE rxdata2048; // 设置BYTE数组 An 8-bit integer that is not signed.
    CString strtemp;
    if (m -ctrlComm. GetCommEvent() == 2) // 事件值为2表示接收缓冲区内有字符
    {
        // 以下你可以根据自己的通信协议加入处理代码
        variant -inp = mCtrlComm. GetInput(); // 读缓冲区
        safearray -inp = variant -inp; // VARIANT型变量转换为 COleSafeArray 型变量
        len = safearray -inp. GetOneDimSize(); // 得到有效数据长度
        for(k = 0; k < len; k++) safearray -inp. GetElement(&k, rxdata + k); // 转换为BYTE型数组
        for(k = 0; k < len; k++) // 将数组转换为CString型变量
        {
            BYTE bt = *(char*)(rxdata + k); // 字符型
            strtemp. Format("%c", bt); // 将字符送入临时变量 strtemp 存放
            m -strRXData + = strtemp; // 加入接收编辑框对应字符串
        }
        UpdateData(FALSE); // 更新编辑框内容
    }
}
```

### 1.6 打开串口和设置串口参数

在主对话框的CSCommTestDlg::OnInitDialog()打开串口,加入一些程序代码,主要完成选择串口,设置串口波特率,有无数据检验,数据位和停止位等。

### 1.7 发送数据

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CSCommTestDlg,选择IDC\_BUTTON\_MANUALSEND,双

通  
用

\*徐兴元,男,1972年12月出生,河南孟津人,郑州大学电气工程学院硕士研究生,研究方向:计算机智能控制与模式识别。



击 BN - CLICKED 添加 OnButtonManualsend() 函数,并在函数中添加如下代码:

```
void CCommTestDlg::OnButtonManualsend()
{ // TODO: Add your control notification handler
code here

UpdateData(TRUE); // 读取编辑框内容
m_ctrlComm.SetOutput(COLEVariant m_str_TXData);
// 发送数据
}
```

### 1.8 设置自动发送

用 SetTimer() 函数设定自动发送周期。这种设置在数据采集和传送控制指令时经常用到。方法是:在 Class Wizard 中选上 Message Map 卡,然后在 Objects IDs 选中 CCommTestDlg 类,再在 Messages 框中选上 WM\_TIMER 消息,单击 ADD - FUNCTION 加入 void CCommTestDlg::OnTimer(UINT nIDEvent) 函数,这个函数是放入“时间到”后要处理的代码:

```
void CCommTestDlg::OnTimer(UINT nIDEvent)
{ // TODO: Add your message handler code here and/or
call default

OnButtonManualsend();
CDialog::OnTimer(nIDEvent); }
```

再在主对话框中加入一个复选按钮, ID 为 IDC\_CHECK\_AUTOSEND Caption: 自动发送(周期 1 s),再利用 Class Wizard 为其添加 BN\_CLICK 消息处理函数 void CCommTestDlg::OnCheckAutosend():

```
void CCommTestDlg::OnCheckAutosend()
{ // TODO: Add your control notification handler
code here

m_bAutoSend = !m_bAutoSend;
if(m_bAutoSend) { SetTimer(1, 1000,
NULL); } // 时间为 1 000 ms
else{ KillTimer(1); } // 取消定时
}
```

其中:m\_bAutoSend 为 BOOL 型变量,在 CLASSVIEW 中为 CCommTestDlg 类加入,并在构造函数中初始化:m\_bAutoSend = FALSE;编译、连接、调试程序。

## 2 集成开发环境 Keil uVision2.0 设置单片机的串口通讯

### 2.1 Keil 的串行窗口

Keil 的串行窗口除可以模拟串口的输入和输出功能外,还可以与 PC 机上实际的串口相连,接收串口输入的内容,并将输出送到串口。这需要在 Keil 中进行设置,方法是:首先在 Debug 环境下,在输出窗口的 Command 页用 MODE 命令设置串口的工作方式,然后用 ASSIGN 命令将串口窗口与实际的串口相关联。程序如下:

```
#include <reg51.h>

void main(void) { unsigned char a;
TMOD = 0x20; // 方式 2,波特率 19 200
TL1 = 0xfd;
TH1 = 0xfd;
SCON = 0x50; // 串行口方式 1
PCON = 0x80;
TR1 = 1; // 启动定时器 T1
while(1) { while(RI == 0);
```

```
RI = 0;
a = SBUF; // 接收到的字节立即发送出去
SBUF = a;
while(TI == 0);
TI = 0; }
}
```

在这段程序中使用中断方式编写串行口输入/输出程序,其功能是实现将串行口收到的字符回送,再通过串行口发送出去。

正确输入源程序,建立工程、编译连接、调试。

### 2.2 连线

为简单实用,让 PC 机上的 COM1 发送、COM2 接收,而 COM2 发送则由 COM1 接收,找两个可以插入 PC 机串口的 DIN9 插座(母),然后用一根 3 芯线将它们连接起来,连线的方法是:2 - 3, 3 - 2, 7 - 7, 接好线把两个插头分别插入 PC 机上的 COM1 和 COM2。

### 2.3 设置通信协议

进入 Debug 模式,然后在 Keil 调试窗口的 Command 页中输入:

```
> mode com1 19200,0,8,1
> assign com1 < sin > sout
```

每输完一行后,应回车。其中:第 1 行命令定义串口 1 的波特率为 19 200,无奇偶校验,8 位数据,1 位停止位,第 2 行是将串口 1 分配给串行窗口。

## 3 通信软件仿真测试

在 Keil uVision2.0 中全速运行单片机的程序,然后切换到 VC++ 6.0 环境,运行 SCommTest 程序,在 SCommTest 发送框中输入 00 01 02 03 AA BB CC 等 16 进制字符,并选上以 16 进制发送及 16 进制显示,单击手动发送,会看到发送后的数据会立即回显到 SCommTest 窗口的接收框中,说明 SCommTest 已接收到了 Keil 发送过来的数据 00 01 02 03 A1 CC 了。若选择自动发送,则会看到连续的回送字符。

最后,切换到 Keil uVision2.0,查看串行窗口 1(注意:点击鼠标右键,选 Hexmode 即 16 进制显示),会看到这里的确接收到了 SCommTest 送来的内容。

## 4 结论

通过利用 MComm 控件和 Keil uVision 2.0 串行窗口,很方便地实现了基于 VC++ 6.0 的 PC 机和单片机间的串行通信的软件仿真,解决了缺乏硬件无法完成 VC++ 6.0 和单片机通信实验的困难。这是本人经过实验验证的一种较好的方法,可以为从事这方面研究的人员提供一条新的思路。

### 参考文献

- 1 李现勇. Visual C++ 串口通信技术与工程实践. 北京:人民邮电出版社, 2002.

(收稿日期:2004-08-19)