ActiveX 技术在组态监控系统串行通信中的应用①

Application of ActiveX Technology on Serial Communication of Configuration Monitoring System

罗 培　马 茜　黄辉先　（湘潭大学信息工程学院 湖南 湘潭 411105）

摘要：以基于单片机的多路绝缘耐压测试系统为背景，针对力控组态软件与单片机的串行通信问题，采用 VC + + 开发了用于串行通信的 ActiveX 控件，在力控组态环境下采用规范的 OLE 接口挂接实现了数据传输和处理。实践表明，该控件很好地完成了力控与单片机系统的串行通信，提高了系统的开发效率和运行效率。

关键词：组态监控 ActiveX 串行通信

1 引言

随着计算机技术的迅猛发展，监控组态软件从数据采集与计量、数据分析、远程监控到过程控制，几乎无处不用[1]。虽然现在主流监控组态软件一般都支持几十甚至几百种外部设备，但在很多情况下，需要自行研发现场测控设备。这样就面临现场测控设备与监控组态软件通信的问题。现场测控设备大多采用标准的通信接口，而串行通信是最常用的一种通信方式。本文以基于单片机的多路绝缘耐压测试系统为背景，提出了一种使用 VC + + 开发组态软件串口设备通信模块的具体实现方法。

2 系统硬件构成及工作原理

多路绝缘耐压测试系统主要由单路绝缘耐压测试仪、高速单片机系统、CPLD、高压继电器组和上位 PC 机组成，见图 1。

单片机将单路测试仪的 250V ~ 1000V 交/直流测试信号通过 CPLD 控制的干簧继电器切换回路加载到相应的测试支路上，单路测试仪把测试到的绝缘耐压值通过 RS232 串行通讯发送到单片机，根据上位 PC 机的要求，单片机再通过 RS232 串口与上位 PC 机通讯。如果被测对象为分路检测，则干簧继电器组切换电路作为 50 路独立输出电路，分别对各路进行检测；如被测对象需要两两相互检测，如电缆、接头等，则干簧继电器组切换电路工作在循环测量状态，测量各路之间的参数；如果需要搜寻被测对象的故障点，则干簧继电器组切换电路工作在故障搜索状态，快速定位故障点。上位机监控系统采用力控组态软件开发，由于力控中没有针对单片机的 I/O 驱动程序，所以需要自行开发串行通信驱动程序。

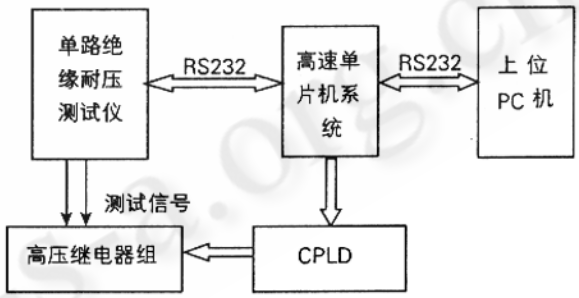

图 1　系统框图

3 驱动程序设计

组态软件一般采用脚本语言编程，但如果程序比较复杂，例如特定的数据处理和计算，则无法用脚本语言实现。这时可以采用 ActiveX 控件或 OLE 控件实现特定的算法功能。具体来说就是：采用 Microsoft VB、VC + + 或其他第三方应用程序开发工具生成一个自定义的 ActiveX 控件，在这个控件中根据通信协议编写特定的算法程序以获得有效的数据，这样当自定义控件注册后，用户就可以在组态软件中使用该控件了。

① 基金项目：湖南省自然科学基金资助项目（06JJ5112）；湖南省教育厅基金资助项目（03C463）

3.1 通信协议和数据格式

本系统使用如下串行通信参数：数据传送率9600bit/s，数据位8bit，停止位1bit，无奇偶校验。数据帧格式如表1。

表1 数据帧格式

帧头1	帧头2	指令	数据段	校验和
AA	55			

指令及数据段：表示本帧数据要求被叫方执行的指令和参数，数据段的长度依指令的不同而异，传输的数据为16进制；对于多字节数据，高字节在前，低字节在后；传送一帧数据为38个字节。

校验：采用校验和方式，即此前所有数据相加，包括帧头。

监控组态系统可向下位机发送测试参数设置命令，也可查询实时测试参数值。

3.2 ActiveX 控件的设计

由于C++编程的源代码效率高、程序运行速度快而且稳定，所以选用VC++来开发自定义的ActiveX控件。利用VC++6.0开发串口通信一般有两种方法，一种是直接使用Windows API函数进行编程，另一种是使用MSComm控件来进行串口编程，采用后一种方法在程序实现上相对第一种方法简单[2]。本设计采用MSComm控件来进行串口编程，具体步骤如下：

（1）建立新控件工程CommOcx。使用MFCActiveX ControlWizard建立控件工程CommOcx；

（2）创建控件包容器。因为本项目为非对话框容器，必须手动设置容器为控件包容器，在项目的CommOcx.cpp文件的InitInstance函数中加入对函数AfxEnableControlContainer()的调用，如下所示：

```
BOOL CCommOcxApp::InitInstance()
{ BOOL bInit = COleControlModule::InitInstance();
  AfxEnableControlContainer();
  ……
}
```

在项目的STDAFX.h头文件中加入头文件afxdisp.h，如下：

#include "afxdisp.h" 即可。

（3）创建MSComm控件。添加MSComm控件后，由于本容器是非对话框容器，所以需要使用CMSComm控件的包裹类的Create函数创建MSComm控件：

① 在Cole类的头文件CommOcxCtrl.h中类的定义前加上#include "mscomm.h"；

② 在CommOcxCtrl.h中CommOcxCtrl.h类的定义的protected部分加上控件包裹类类型的成员变量声明；

③ 用ClassWizard为类CCommOcxCtrl添加消息响应函数WM_CREATE的处理函数CCommOcxCtrl::OnCreat；

④ 添加串口事件消息处理函数OnComm()；

（4）添加消息映射。将有关的消息映射宏及相应的消息响应函数添加到控件中。另外，由于控件的消息映射机制和普通应用程序有所不同，它不会自动执行ON_UPDATE_COMMAND_UI消息响应的函数，需要手动添加工具栏的OnUpdateCmdUI函数。

（5）初始化串口。正常操作将自定义控件CommOcx.ocx插入组态软件中后，控件一般不能正常初始化，因此必须将OnCreate函数中对串口的初始化移至此函数体外触发[3]。在此添加一个Timer控件，使之在控件开始加载时完成所有初始化工作，之后使之无效。基本步骤如下：

① 为OnTimer控件建立消息映射：afx_msg void OnTimer(UINT nIDEvent)；

② 初始化Timer控件：在OnCreat中添加如下代码SetTimer(1,1000,NULL)；

③ 初始化串口：在OnTimer函数中添加代码如下：

```
void CTestCtrl::OnTimer(UINT nIDEvent)
{// TODO: Add your message handler code here and/or call default
  m_cMscomm.SetCommPort(1); //选择COM1
  m_cMscomm.SetInputMode(1); //输入方式为二进制方式
  m_cMscomm.SetInBufferSize(1024); //设置输入缓冲区大小
  m_cMscomm.SetOutBufferSize(512); //设置输出缓冲区大小
  m_cMscomm.SetSettings("9600,n,8,1"); //波特率9600,无校验,8个数据位,1个停止位
```

```
if(! m_cMscomm.GetPortOpen())
{ m_cMscomm.SetPortOpen(TRUE);//打开串口
}
//每当串口接收缓冲区中有多于等于1个字符时将引发一个数据的OnComm事件
m_cMscomm.SetRThreshold(1);
m_cMscomm.SetInputLen(0); //设置当前接收区数据长度为0
m_cMscomm.GetInput(); //先预读缓冲区以清除残留数据
KillTimer(1); //使Timer控件无效
COleControl::OnTimer(nIDEvent);
}
```

(6) 添加控件属性及其他成员变量。监控组态系统可向下位机发送测试参数设置命令,也可查询实时测试参数值。按照数据格式,传送一帧数据为38个字节。所以在控件中设置38个上行成员变量x0 – x37和38个下行成员变量y0 – y37。具体步骤如下:

① 打开MFC ClassWizard,选择Automation—Add Property,为控件添加自定义属性x0 – x37;

② 初始化属性对应的成员变量:在void CCommOcxCtrl::OnTimer(UINT nIDEvent)中添加代码如下:

```
void CCommOcxCtrl::OnTimer(UINT nIDEvent)
{. . .
m_x0 =0;
. . .
m_x037 =0;
m_y0 =0;
. . .
m_y037 =0;
KillTimer(1); //使Timer控件无效
COleControl::OnTimer(nIDEvent);
}
```

(7) 由于串口接收数据都是由CMSComm控件的串口事件消息处理函数OnComm()完成的。因此,我们把接收后的转化、分发等处理部分也置于此函数中,具体代码不在此敷述。

(8) 初始化时间处理。控件的初始化应在容器运行时进行,否则,如果在容器设计时初始化一次,而在容器运行时会再初始化一次,系统将会不断弹出提示窗口导致系统无法工作,为解决此问题,可在OnCreate()中初始化Timer控件时添加一个AmbientUserMode()函数作为判断,该函数可判断容器是否处于运行状态,若是,则返回值"1",如下:

```
if(AmbientUserMode()) //当容器处于运行状态时
SetTimer(1,1000,NULL);
```

当容器处于运行状态时,Timer控件只运行一次,之后便无效。

(9) 界面处理。使控件在嵌入使用显示为一矩形,运行时不可见,同时去掉椭圆,应在OnDraw函数中作如下修改:

```
void CCommOcxCtrl::OnDraw(CDC * pdc, const CRect& rcBounds, const CRect& rcInvalid)
{if(! AmbientUserMode()) //当容器处于非运行状态时
pdc -> FillRect(rcBounds, CBrush::FromHandle((HBRUSH)GetStockObject(WHITE_BRUSH)));
//pdc -> Ellipse(rcBounds);
}
```

选择工具条上"编译——构件CommOcx.ocx",生成CommOcx.ocx,控件制作完成。

经过上述步骤,即可用VC++编写的串口通信程序包装成ActiveX控件。

4 ActiveX控件应用

在力控组态程序中插入自定义ActiveX控件 – CommOcx.ocx,选择Draw菜单命令"查看/OLE控件方法/属性",可以看到此控件的所有属性。如图2所示。这样组态监控系统的变量与单片机中的变量就可以通过控件的属性变量联系起来了。

5 结论

本系统采用ActiveX技术实现了组态监控软件与单片机的串行通信,在多路绝缘耐压测试系统现场运行时性能稳定,效果良好。针对各工业应用系统不同的串口设备终端,开发人员只要在该控件中,根据不同的通信协议和监控需求对相应的函数进行更改,就可以快速实现串口通信编程。实际运行表明,该方法具有一定的推广实用性。

(下转第49页)

(上接第 45 页)

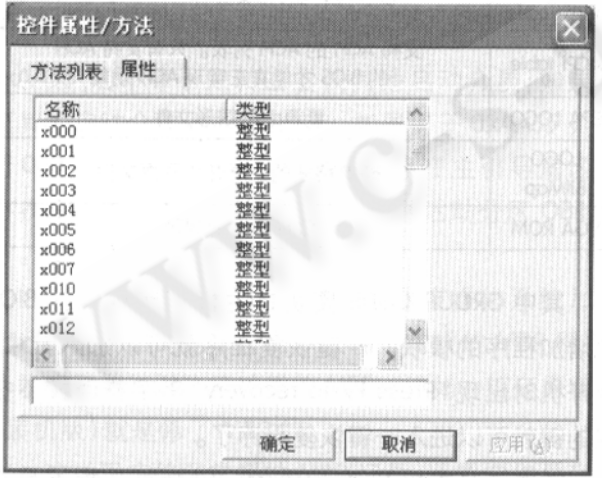

图 2 控件属性/方法

参考文献

1 马国华,监控组态软件及其应用[M],北京:清华大学出版社,2001,8.

2 徐世国、张全贵,组态软件中串口设备通信实现方法设计[J],微计算机信息,2004,20(8):25-26.

3 程锴、张楠,串口通讯技术在组态软件中应用[J],电子测量技术.2004(4):82-84.

4 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,2004,203-205.

5 李现勇,Visual C++串口通信技术与工程实践[M],北京:人民邮电出版社,2002,8.