

文章编号: 2095-4980(2019)01-0141-05

一种工程项目中的网络实现

程西娜, 蔡文斋

(中国电子科技集团公司 第三十九研究所, 陕西 西安 710065)

摘要:介绍了一种工程项目中整个网络系统的编程实现方法。采用Indy10组件实现了网络通信;使用用户消息机制完成了协议级置码、解码;使用初始化(INI)文件技术实现了网络参数配置的编辑与装载;使用内存对齐技术实现了协议的快速内存交换;开发出工程使用的统一网络配置、网络调试、网络通信与管理函数;设计了整个工程中网络模块与业务逻辑模块的通信机制与数据交换方法,制作并开发出整套工程用的监控系统。目前整套工程样机已参加了大系统联试,并通过总体单位验收。

关键词:传感器;消息;网络通信;初始化文件

中图分类号: TN914.42

文献标志码: A

doi: 10.11805/TKYDA201901.0141

Network design and implementation in an engineering project

CHENG Xi'na, CAI Wenzhai

(The 39th Research Institute of China Electronics Technology Group Corporation, Xi'an Shaanxi 710065, China)

Abstract: A programming method of the network in a project is introduced. The net communication is realized by INDY10 packages. The protocol coding and decoding are fulfilled with windows user messages. The network parameters configuration is realized with the INItialization(INI) files. The protocol data exchanging is realized by the memory alignment technique. Functions of net configuration, net debugging and net communication are developed. The method of data exchange between the network module and the logic module is designed. The whole monitor system is fulfilled and put into practice.

Keywords: sensor; message; net comunication; INItialization file

某单位承研了多套天线类仿真训练系统工程机,整个系统由十几台电脑组成,每3台电脑组成一组天线类仿真训练系统。该天线仿真训练系统分为高仿系统和普通仿真系统,其中高仿系统带一组硬件传感器,普仿系统需设计一组与高仿系统同样功能的软件传感器,要求图文界面与原真实硬件操作界面一致。主仿真计算机与外协单位计算机使用组播方式通信,主仿真计算机与硬件传感器使用网络用户数据报协议(User Datagram Protocol, UDP)方式通信。

1 总体需求

整个仿真系统的计算机使用网络交换机连接。普仿系统与高仿系统结构几乎相同,不同部分为普仿系统改用软件传感器代替高仿系统的硬件传感器,图1为天线主控机外置接口示意图。网络通信需求为:研发出工程多路网络UDP通信、多路组播方式通信的程序;具备配置信息装载与编辑存储功能、自环测试功能、调试支持功能。软件任务书要求为:网络通信模块应与具体控制工程项目无关,可用于不同的工程

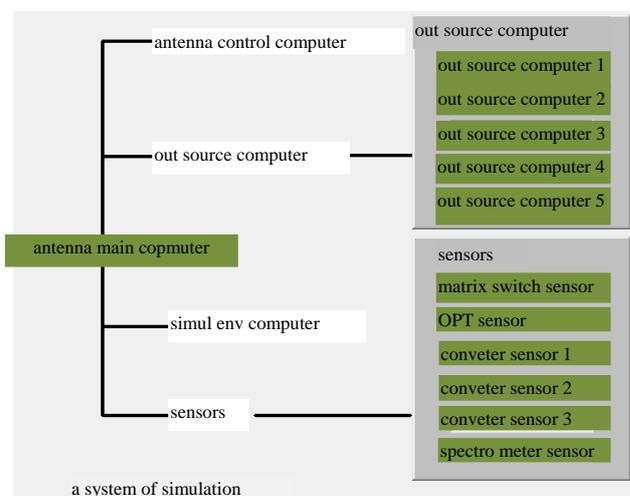


Fig.1 Antenna simulation trill system
图1 某套天线仿真训练系统

项目, 为整个系统研发出基础的网络通信部分。

工程应用软件运行于 Windows7 V64 系统, 开发工具为 Rad studio XE4, 网络通信使用 Indy10 组件集开发。

2 Windows 系统下网络编程方法

Windows 系统下通常采用的网络编程方法:

1) 采用纯应用程序编程接口函数(Application Programming Interface, API), 使用操作系统提供的 win32 API 中的网络通信库。网络通信库有 2 种形式: winsock v1.1 和 winsock v2.2, 这 2 套不同的库, 需要初始化的参数较多^[1-2]。

2) 采用操作系统提供的对象类别扩充组件(Object Linking and Embedding(OLE) Control Extension, OCX)库^[3], 该库需要注册才能使用。

3) 采用与网络通信协议无关的 winpcap(windows packet capture)网络包, 该包提供了访问网络底层的功能, 在普通的网络编程中较少使用^[4]。

开发工具本身提供网络组件, 如 rad studio 开发工具中的 overbyte 网络组件库、Clever.Suite.Internet 库、Indy(Internet direct socket)库等。Indy 库为 Indy 开源组织维护开发, 内置于 rad studio xe4 工具中, 它提供了几十种网络组件, 支持各种网络协议编程。在权衡多种方法后, 确定使用 rad studio 开发工具中 Indy 包。总体设计方法: 设计 16 路网络通信模块; 8 路组播网络通信模块; 网络访问数据结构; 网络通信部分与具体业务逻辑模块的通信方法。

3 网络配置参数编程方法

网络通信涉及 2 个重要参数: 网络地址与端口。为给其他程序开发者提供统一服务, 将本系统 16 路网络通信模块设计为可编辑、可装载函数。网络配置窗口在创建时的 FormCreate 函数中调用配置装载函数, 将上次的更改项装入。如果不再编辑, 则一次性初始化网络; 如果在界面上更改, 则按保存按钮存为 INI 文件形式^[5]。再编写 NetApply 函数, 负责本次更改生效并开放网络环境。

INI 文件的存储是 windows 系统其中之一功能, 例如 system1.ini 文件内容可能为:

```
[com 参数配置, 波特率, 9600];  
[Net 网络参数配置, IP 地址, 192.168.2.32];
```

下面给出关键的使用 C++Builder XE4 开发的代码段。

```
str1=m_Dir.GetCurrentDirectoryA();  
str2=m_Dir.GetParent(str1);  
str3=m_Dir.GetParent(str2);
```

MyIniFils=new TIniFile(str3+"\\NetConfigIniFile"+"\\NetConfig.ini");//将 INI 文件创建到自己定义的文件夹。如果不取当前文件夹, 则 INI 文件存储在 windows\system32 文件夹, 这样不利于查找。

```
MyIniFils->WriteInteger(ConfigSelection,"LocalPort",LocalPort);  
delete MyIniFils;
```

为使界面输入具备输入检查功能, 使用 Raize 组件库的 RzNumberEdit 组件设计端口输入、RzMaskEdit 组件设计 IP 地址输入检查。为进一步在网络初始化中检查 IP 地址输入的合理性, 使用 Indy 的 IsIp 函数检查发送地址的合理性, 避免其他程序员不小心写错地址引起网络通信异常。这种保护性设计在程序中至关重要, 否则当用户将参数配置错误时, 很可能引起当机。

将网络参数保存、装载、设置设计为函数形式, 通过界面按钮调用, 在应用逻辑模块中, 也可以调用, 这样无论是网络调试窗口还是其他应用窗口调用后效果一样。

4 网络读写

网络 UDP 协议是一种面向非连接的数据包协议。在 Indy 中, 分为服务器端与客户端, 使用 UDPClient 组件实现网络发送功能、UDPServer 组件实现网络读入功能, 这与通常的 win32 网络方法不一样。下面给出 3 个小函数(网络读、网络写、网络数据的分发)完成网络收发:

UDPServer 组件负责网络收, 需要重载 IdUDPServer1UDPRead 函数:

```
void __fastcall TNetConfigForm::IdUDPServer1UDPRead(TIdUDPListenerThread *AThread,const TIdBytes
AData,TIdSocketHandle *ABinding)
```

Indy 网络通信中, IdUDPServer1UDPRead 函数为自动回调函数, 当网络有数据时, 该函数自动被调用, 不需要程序员再写网络读函数; AData 为网络读入数据缓冲区; TIdSocketHandle 为发送信息, 使用时开发者将 AData 信息读取到自己使用的数据缓冲区即可。

为简捷十几路的数据分发, 开发了自创的数据分发函数, DecodeAnyNet 形式为:

```
DecodeAnyNet(const TIdBytes AData,TIdSocketHandle*ABinding, int NetIndex)
```

在网络回调函数中再调用这个数据分发函数, 十几路的网络回调函数书写就非常简单, 只需要改网络索引号即可。该函数由配置索引号分解到各自的缓冲区, 网络缓冲区定义为一共用的数据结构, 使用一维数组引用这个结构, 十几路网络数据可以使用循环方法访问, 将极大减少代码量。

DecodeAnyNet 函数内关键的形式化代码为:

```
CopyMemory(@AnyCommPackedNetN[Index].AnyReadBuffer[0],AData,Len);
```

```
FromIP[Index]=ABinding.PeerIP;
```

```
PostMessage(.....);
```

DecodeAnyNet 将网络数据分配好, 得到对方地址后, 再根据索引号调用发送消息函数, 通知业务逻辑窗口取数据, PostMessage(业务逻辑窗口句柄, 用户定义的某网络读完成消息号, 0,0)^[6]; 在业务逻辑模块中, 加入对该消息的处理函数, 网络窗口读到数据, 通过用户消息 PostMessage 通知业务逻辑模块处理数据, 业务逻辑模块不需要与网络通信模块直接通信。

网络读函数为自动回调函数, 一旦网络有数据到来, 在函数的 AData 参数中直接得到数据, Length(AData) 为数据长度, 在 Abinding 参数中得到对方的 IP 地址。

设计专用的网络发送函数, 方便业务逻辑层程序员调用。网络层设计为通用函数形式, 哑元参数使用网络组件类型、网络地址等。如第 6 路网络发送函数的调用为:

```
UDPWrite(m_IdUDPClient[6], //indy 组件;
```

```
m_UdpNetObj[6].WriteAddr, //网络 IP 地址;
```

```
m_UdpNetObj[6].WritePort, //网络端口;
```

```
m_UdpNetObj[6].WriteBuffer, //写缓冲区地址指针;
```

```
m_UdpNetObj[6].WriteLength); //写长度;
```

使用 GStack->IsIP(str2) 函数^[7]检查该函数内 IP 地址的正确性, 需要在程序开始部分引入非可视类组件 #include<IdStack.hpp>,

在其他应用模块中(业务逻辑模块)真正调用网络发送函数时, 不需要直接调网络发送函数, 而是调用自己私有的填写缓冲区函数 FillBuffer, 将私有的缓冲区数据赋值, 然后调用内存复制函数 CopyMemory 完成从私有变量到公有变量的赋值, 再调用发送消息函数, 形式化的伪代码为:

```
FillNetBuffer; //该函数在调用前可能是具体业务部分的频域处理, 也可能是信号分析等;
```

```
CopyMemory(&PublicNetBuffer[1],&MyStructure,SizeOf(MyStructure));
```

```
PostMessage(网络配置窗口句柄, 用户发送消息号, 0,0);
```

CopyMemory 函数功能为: 为自己私有的协议数据结构到内存通信缓冲区的内存赋值。使用该函数的条件是自己定义的通信结构内容与通信缓冲区每字节数据一致, 即内存对齐技术。在定义结构时使用 C 语言的条件编译指令(#pragma pack(1))即可实现。

```
#pragma pack(push) //保存内存对齐状态
```

```
#pragma pack(1)
```

```
typedef struct _ACUTOMAC{
```

```
    BYTE    CurCtrlMode;
```

```
    BYTE    CurAEWorkmode; //Az EI 工作方式
```

```
.....
```

```
ACUTOMAC,*PACUTOMAC;
```

```
#pragma pack(pop) //恢复内存对齐状态
```

使用这种技术后, 结构变量在 WINDOWS 操作系统中将按照一字节分配内存, 保证了 CopyMemory 函数从网络缓冲区到用户缓冲区的直接复制。

在网络配置窗口或者业务逻辑窗口内, 设计了对读入与写出用户消息的处理函数(其他所有消息同理处置)。网络配置窗口消息函数才真正调用网络发送函数, 业务逻辑模块的读消息已经得到了网络数据, 不再关心网络通信部分, 网络通信部分由网络配置一个窗口类统一处理。

网络收端与网络发端初始化函数稍有区别, 收端只开放端口, 激活 UDPServer 组件即可; 发端需要设置对方 IP 地址与端口, 再激活 UDPClient 组件。使用形式化参数编程, 将这些初始化函数写成统一的初始化函数。

Indy 组件使用时, 有些非可视组件中的函数功能强大, 需要引入对应的 *.hpp 文件, 如 #include <IdGlobal.hpp>。

5 网络消息机制

网络模块与其他业务逻辑模块交互部分设计为用户消息机制, 网络读与写函数完成后(该函数位于网络配置窗口类), 分别设计为用户定制的读、写完成消息, 在网络通信完成后, 发送这些完成消息给用户业务逻辑窗口。用户定制的消息机制是 windows 操作系统具备的功能, 开发者在使用时如果定义了自己的用户消息, 并提供了消息映射方法, 则自定义消息与 windows 操作系统消息由操作系统统一调度处理。

为了使网络通信部分可以适应所有需要使用的业务逻辑开发者, 将该网络初始化及调试部分设计为通用的 Form 界面, 与具体某台计算机运行的程序无关。该部分可以在 3 台计算机的任一台上运行, 为独立窗口、带网络配置与存储、十六进制源码显示、附加的网络调试功能。该界面在任一业务逻辑程序中都可以通过菜单加载, 也可以通过业务逻辑模块的窗口创建函数调用网络初始化函数, 完成网络后台模块的加载。

网络读数据功能在 Indy10 组件收到数据时自动触发, 该组件在后台以线程方式运行。一旦网络端口有数据到来, 则自动执行 UDPServer 的回调函数^[8], 开发者从该函数的参数中直接得到数据, 在网络配置窗口中, 实现了网络的读写, 与具体工程项目的通信协议无关。网络通信部分是其他所有业务逻辑模块都要使用的共同的、最基础部分。当网络读完成后, 由该网络 Form 发送用户消息, 通知其他业务逻辑模块读取数据, 然后在业务逻辑模块的消息处理函数中完成数据处理等工作, 这样网络通信部分与整个业务逻辑模块隔离开来。当业务逻辑模块需要发送数据时, 首先调私有缓冲区填写函数完成发送缓冲区按照协议的填写, 发写用户消息给后台的网络配置窗口, 由统一的网络通信模块完成网络数据发送。这样任一业务逻辑模块的开发者不用关心网络通信部分, 业务逻辑程序员看到的仅仅是用户消息, 这样一种机制下各程序员仅编写与自己有关的模块即可。

用户消息的定制方法: 首先定义一个全局变量, 代表用户消息, 注意消息中应带 user 常量, 这是 windows 系统预留给用户的常量, 用户消息都应大于该数。用户消息定义:

```
#define WM_ReadNet1 WM_USER+221
```

设置消息映射机制, 在窗口类说明(*.h 文件)中引入私有消息处理过程^[9], 如:

```
private://User declarations
```

```
BEGIN_MESSAGE_MAP[7] MESSAGE_HANDLER(WM_ReadNet1,TMessage,ReadNetProcedure1);
```

```
MESSAGE_HANDLER(WM_ReadNet2,TMessage,ReadNetProcedure2);
```

```
END_MESSAGE_MAP(TForm)
```

编写消息处理函数, 该函数放置在对模块的窗口类中, 消息接收者需要写用户消息处理函数, 如网络配置窗口类需要处理由其他计算机传来的网络读消息, 则函数形式为:

```
void__fastcall ReadNetProcedure1(TMessage & Msg);
```

```
void__fastcall ReadNetProcedure2(TMessage & Msg);
```

编写完这些机制代码后, 在网络通信读回调函数中调用 win32API 提供的 PostMessage 函数, 其中参数 1 为接收消息的窗口句柄, 参数 2 为消息号, 参数 3,4 分别对应消息参数本身。如果消息不传参数, 设为 0 即可。一般用户程序内自己需要交换的数据已定义为某结构缓冲区类型^[10], 在消息参数中直接使用 0 即可。如, 网络读函数完成后, 调用 PostMessage(业务逻辑窗口句柄, 用户消息, 0,0); 通知业务逻辑窗口模块处理该消息。

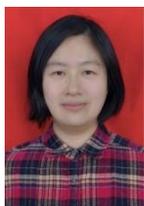
6 结论

介绍网络 Indy 组件使用方法, 保护性设计; 开发整个控制工程项目中底层的网络通信部分, 设计为统一服务形式, 在业务逻辑层节约了大量代码段; 电脑运行软件业务需求各异, 但共性部分设计为灵活的可配置选择, 提高了效率。网络组播编程与 UDP 编程相仿, 需要注意的是组播客户组件完成网络收数据功能, 组播服务器组件完成网络发数据功能。有了可靠的后台网络通信保障, 业务逻辑层开发者在开发具体工程用监控系统时容易了许多。目前整套工程样机已经参加了大系统联试, 并通过总体单位验收。

参考文献:

- [1] 王艳平. Windows 网络与通信程序设计[M]. 2 版. 北京:人民邮电出版社, 2009. (WANG Yanping. Windows network communication and programming[M]. 2nd ed. Beijing:Posts and Telecom Press, 2009.)
- [2] JEFFREY Richter. Window高级编程指南[M]. 王书洪,刘光明,译. 北京:清华大学出版社, 1999. (JEFFREY Richter. Advanced Windows[M]. Translated by WANG Shuhong,LIU Guangming. Beijing:Tsinghua University Press, 1999.)
- [3] 姜文喜,吴玉新,张学成,等. 伺服系统多测试平台测试数据管理软件的设计[J]. 微电机, 2016,49(8):92-95. (JIANG Wenxi,WU Yuxin,ZHANG Xuecheng,et al. The servo system multi-test platform test data management software design[J]. Micromotors, 2016,49(8):92-95.)
- [4] 李国奇. 地球站职工测评系统(客户端)的设计与实现[J]. 广播电视信息, 2016(12):95-96,99. (LI Guoqi. Design and implementation of the earth station staff assessment system(client)[J]. Broadcast Television Information, 2016(12):95-96,99.)
- [5] 拉西诺维奇,所罗门. 深入解析 Windows 操作系统[M]. 5 版. 北京:人民邮电出版社, 2009. (RUSSINOVICH M E,SOLOMON D A. In-depth analysis the Windows operating system[M]. 5th ed. Beijing:Posts and Telecom Press, 2009.)
- [6] 周爱民. Delphi 源代码分析[M]. 北京:电子工业出版社, 2004. (ZHOU Aimin. Delphi source code analysis[M]. Beijing: Electronic Industry Press, 2004.)
- [7] 范文庆,周彬彬,安靖. 精通 Windows API——函数、接口、编程实例[M]. 北京:人民邮电出版社, 2009. (FAN Wenqing, ZHOU Binbin,AN Jing. Master Windows API—function,interface,program sample[M]. Beijing:Posts and Telecom Press, 2009.)
- [8] 苏春晓,王鹏,杨存榜,等. 基于网络的数字示波器远程集中软件平台[J]. 太赫兹科学与电子信息学报, 2014,12(6): 884-889. (SU Chunxiao,WANG Peng,YANG Cunbang,et al. Remote centralized software platform for digital oscilloscope based on network[J]. Journal of Terahertz Science and Electronic Information Technology, 2014,12(6):884-889.)
- [9] MYRERS S. Effective C++55 个改善编程与设计的有效方法[M]. 3 版. 侯捷,译. 北京:电子工业出版社, 2017. (MYRERS S. Effective C++:55 specific ways to improve your programs and designs[M]. 3rd ed. Translated by HOU Jie. Beijing:Publishing House of Eleectronics Industry, 2017.)
- [10] DOV bulka,DAVID Mayhew. 提高 C++性能的编程技术[M]. 北京:电子工业出版社, 2017. (DOV bulka,DAVID Mayhew. Efficient C++:performance programming techniques[M]. Beijing:Publishing House of Eleectronics Industry, 2017.)

作者简介:



程西娜(1983-),女,湖南省湘潭市人,硕士,工程师,主要研究方向为航天测控工程开发、控制类应用软件开发、软件测试 .email: 157989374@qq.com.

蔡文斋(1962-),男,西安市人,硕士,高级工程师,主要研究方向为航天测控工程开发、工程控制类应用开发、软件测试、新技术在工程化中的应用研究等.