Research of EtherNet/IP and Development of Its Network Node

Lizhong Zhang[1], Niansuo Xie[2]
1: School of Electrical Engineering,Shanxi University of Technology, Hanzhong , China
2: School of Materials Science and Engineering,Shanxi University of Technology, Hanzhong , China
goudarlin@126.com

***Abstract*—EtherNet/IP network node was developed on field bus for the upper EtherNet-EtherNet/IP on the NetLinx system of Rockwell Automation to be achieved. EtherNet/IP and its structure and encapsulation format were analyzed, and CIP and its model and object model were explained. The method and process of UDP multicast network programming were given and illustrated under linux, and some actual problems in using UDP socket programming encountered were explained and the corresponding solutions were given. EtherNet/IP network data was collected by using CIPAlyzrODVA and Ethernet TCP/UDP network packets were captured and analyzed by using TCP DUMP, and the problems should be paid attention to in process of actual network data collectted. The elements of network node developed were explained and software design flow charts of network node achieved were illustrated. On this basis, EtherNet/IP network node was developed by using Socket programming and C language in a PC under Linux on EtherNet/IP network. The results show that the developed network node is Capable of network communication and EtherNet/IP is achieved. The process and results of EtherNet/IP network node developed show that the method in which EtherNet/IP network node was developed on field bus in this development can overcome the limitations of development and debugging in the embedded microprocessor in the process of EtherNet/IP network node developed or EtherNet/IP achieved.**

Keywords-EtherNet/IP;networknode;CIP;socket;linux;UDP multicast; TCP/ IP; UDP/IP; explicit message; I/O message; UCMM

I. 引言

EtherNet/IP（Ethernet Industrial Protocol，以太网工业协议）是目前较为流行的一种工业以太网协议[1]。该协议由以太网标准的物理层和数据链路层、以太网 TCP/IP 协议以及应用层 CIP 协议三部分组成，它和 DeviceNet 以及 ControlNet 一样，它们都是基于 CIP(ControlandInformation Protoco1)协议的网络。其中 CIP（Common Industrial Protocol，or Control and Information Protocol，通用工业协议或控制与信息协议）是 EtherNet/IP 的核心组成部分。它具有完全标准化和完全开放性的特征[2]。CIP 通用工业协议是 Ethernet/IP、Devicenet 和 Controlnet 3 种网络的交叉点，从而使 3 种网络之间实现共享,并借由工业路由器连接起来[3]。与生产者/消费者通信模式相结合，CIP 允许主从、对等、一对多以及广播通信。由于它支持所有对自动化有用的主要的通信关系，它真正是一个最灵活和完全的协议。Ethernet/IP 以一个固定的应用接口将设备从总线层连接到控制层以及企业的层面上。EtherNet/IP 首次实现了传感器级网络到控制器和企业级网络的无缝集成[4]。

实现节点的网络功能和 EtherNet/IP 协议有着重要意义。本文以 Rockwell Automation 的 NetLinx 网络体系的上层以太网 EtherNet/IP 为研究对象，借助罗克韦尔三层网络平台，利用套接字接口编程技术，在 Linux 下 PC 机中用 C 语言开发了 EtherNet/IP 网络上的网络节点。由于 Linux 操作系统的可移植性，可在开发完成后移植到嵌入式系统中。和嵌入式系统相比 PC 机有着更多的开发和调试工具。

II. CIP 协议

A. *CIP 协议模型*

控制和信息协议（The Contorl and Information Protocol）简称 CIP 协议。CIP 是专为工业控制设计的应用层协议。 CIP 协议是一个端到端的面向对象并提供了工业设备和高级设备之间的连接的协议，它独立于物理层和数据链路层之上。CIP 提供了访问数据和控制设备操作的服务集。它采用用户数据协议/网际协议（UDP/IP）和传输控制协议/网际协议（TCP/IP）作为 Ethernet 网上的控制和信息协议，允许发送显式（信息）和隐式（控制）报文。其中，隐式报文是对时间有苛刻要求的 I/O 信息（事件触发、控制器互锁等），通过 UDP/IP 完成的隐式报文中数据区包含实时 I/O 数据（CIP 的控制部分）；显式报文是无时间苛刻要求的点对点信息，可由 TCP/IP 完成（CIP 的信息部分）。CIP 向终端用户提供了自动化系统必不可少的控制、组态、数据采集服务功能。为面向自动化领域提供了 Ethernet 网上的工业自动化设备的互操作性和可换性。

B. *CIP 协议对象模型*

CIP 协议用面向对象语言描述。CIP 协议对象模型：未连接报文管理器(UCMM)的主要作用是建立 I/O 连接和显性连接。当一设备要与网络上另一设备建立连接时，先给对方设备的 UCMM 发送连接请求，对方设备若答应请求则创建、初始化连接对象，并向请求设备的 UCMM 返回响应信息。然后原来发出请求的设备创建、初始化自己的连接对象。连接完成之后，就可以通过连接对象传送显性

978-1-4577-1415-3/12/$26.00 ©2012 IEEE

报和 I/O 报文。报文路由的主要作用是对报文打包、解包，将连接对象接收过来的报文解包，分发给各个对象。然后将各对象返回的数据打包，交给连接对象发送。

从 CIP 对象模型可以看到 CIP 协议采用未连接管理器和连接管理器来处理网络上的信息。EtherNet/ IP 协议是基于高层网络连接的协议，一个连接为多种应用之间提供传送信息的通道。未连接管理器为尚未连接的设备创立连接。每一个连接被建立时，这个连接就被赋予一个连接 ID，如果连接包括双向的数据交换，那它就被赋予两个连接 ID。EtherNet/ IP 使用 TCP/ IP 和 UDP/ IP 来封装网络上的信息，包括显式信息连接和隐式信息连接等。显式报文适用于两个设备间多用途的点对点报文传递，是典型的请求/应答通信方式，常用于节点的配置、问题诊断等。显式报文通常使用优先级低的连接 ID，并且该报文的相关报文包含在显式报文数据帧的数据区中，包括要执行的服务和相关对象的属性及地址。隐式报文适用于对实时性要求较高和面向控制的数据如 I/ O 数据等。I/O 报文为一个生产应用和一个或多个消费应用之间提供适当的专用的通信路径。I/O 报文通常使用优先级高的连接 ID，通过一点或多点连接进行报文交换。报文的含义由连接 ID 指示，在 I/O 报文利用连接 ID 发送之前，报文的发送和接受设备都必须先进行配置，配置的内容包括源和目的对象的属性，以及数据生产者和消费者的地址。UCMM 用来可靠的处理未连接的外部请求和回答。它包括建立双方外部信息和 I/O 连接。网络预定义了 UCMM 是如何访问和允许限制出现在 UCMM 中的信息。当建立 UCMM 外部信息连接时，对象请求目标是对象信息的路由。外部信息连接是无条件的点对点的进行连接。点对点连接仅存在两个设备之间。请求设备是连接的起始端（客户机），接收和回答模型（服务器）是请求的另一个终端。

III. ETHERNET/IP 协议网络节点的开发

A. EtherNet/IP 协议的体系结构与封装格式

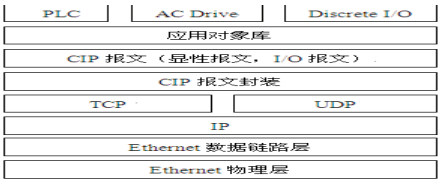

图 1 EtherNet/IP 封装体系结构

Figure 1Architecture of EtherNet/IP encapsulation

图 1 是 EtherNet/IP 协议的体系结构。EtherNet/IP 协议的体系结构应用层采用 CIP 协议，其底层完全采用了现有以太网的传输层、网络层、数据链路层、物理层，未作任何修改。因此,应用 Linux 的 Socket 编程接口，实现 EtherNet/IP 协议主要就是实现应用层的 CIP 协议。EtherNet/IP 协议不仅支持点对点的通讯模式,还支持生产者/消费者通讯模式，这种一对一个或多个的通讯模式是通过 IP 多播的技术实现的。组播是指组播源向一组接收者发送数据的方法。组播方式用于网络中一个设备向网络中多个指定设备进行数据传输，即服务器可以将一个数据包通过网络硬件设备复制的方法同时分组发送给多个需要接收的客户端。Ethernet/IP 网络的生产者/消费者模式要求实现一对多的通讯模式，在 Linux 下是通过 UDP 组播实现的。CIP 报文封装的格式如表 1 所示。将 CIP 报文正确封装好，通过 Socket 编程接口发送到网络上的其他节点，就可以实现与网络上其他节点的通讯。

表 1 CIP 协议封装格式

Chart 1 CIP Encapsulation Format

结构	字段	数据类型	意义
Encapsulation header	Command	UINT	封装命令
	Length	UINT	数据长度
	Session handle	UDINT	Session ID
	Status	UDINT	状态代码
	Sender Context	ARRAY of octet	仅与发送方相关的封装命令长度为 8 字节
	Options	UDINT	选项标志
Command Specific data	Encapsulated data	ARRAY of 0 to 65511 USINT	特定命令相关的数据

CIP 协议报文经封装之后，在通讯时又逐层被封装上 TCP、IP、Ethernet 报头。EtherNet/IP 协议的封装也规定了支持所有的 EtherNet/IP 设备的保留的 UDP 端口号。所有的设备将在 UDP 端口号为 0xAF12 接收 UDP 数据包。所有经过 0xAF12 端口发送的 TCP 的数据都会有固定长度为 24 字节的报头放在数据前，作为数据的一部分。但这个封装信息的总长度不能超过 65535 个字节[5]。CIP 传送类 0 和类 1 数据包时将会发送共同格式的自带寻址信息的 UDP 数据包， 自带寻址信息 UDP 的一部分数据信息是为了 CIP 传送类 0 和类 1 数据包。

B. Linux 下的网络编程

Linux 下的网络编程包括 TCP 套接字编程、UDP 套接字编程、UDP 广播、UDP 组播、多线程编程以及 CIP 应用层软件设计等工作。Linux 下超文本编辑软件 Vi 作为 C 语言程序的编辑环境，采用 C 语言编译软件 Gcc、调试软件 Gdb 进行程序的编译和调试。

TCP 套接字的实现过程、UDP 套接字的实现过程及具体编程和其他软件设计由于篇幅有限略。这里需要重点说明强调的两个问题：使用 UDP 套接字的一些问题及措施，Linux 下 UDP 组播的软件设计。

使用 UDP 套接字的一些问题及措施：①因为 UDP 套接字是一种不可靠的协议，在应用中难免会遇到一些问题。UDP 协议不保证数据的可靠到达，如果程序要求传送的 UDP 数据报可靠的到达，就必须在程序中处理这个问题。通常是采用确认重传的方法来实现。②用超时和重传来处理丢失的数据报。当程序接收到一个 UDP 数据报之后，必须向发送者返回一个确认信号。③用数据报序列号来区分重复数据报。当使用超时重传机制时，接收者可能多次接收到同一数据报。为了使接收者能够区分重传数据报，对发送的每个数据报编号。④UDP 协议不保证数据报顺序到达。若对数据报处理的先后顺序有要求，就必须在程序中处理数据报的顺序问题。⑤UDP 协议没有流量控制，如果程序要保证流量控制，应采用 TCP 协议。

Linux 下 UDP 组播。以太网中的信息传播主要采取三种方式：单播、广播和组播。Ethernet/IP 网络的生产者/消费者模式要求实现一对多的通讯模式，在 Linux 下是通过

UDP 组播实现的。因此在 Linux 下的网络编程中，Linux 下 UDP 组播的实现至关重要。下面给出了 Linux 下 UDP 组播的实现的软件设计方法。

组播是指组播源向一组接收者发送数据的方法。组播功能的实现需要满足以下三个基本要求：一是有唯一标识一个组播组的机制。在 IP 网上更加精确的标识组播组，常采用(组播地址，组地址)地址对的方式，即(S，G)地址对的方式，对某一组播组进行唯一识别。二是需要有组成员加入和推出组的机制。一台主机可以自由地加入一个组播成为其中一个成员，一个组播组的成员也可以自由地退出该组。三是需要有一个能够使路由器在 IP 网上高效传送组播流量到各个成员的路由协议。

组播套接字接口编程。对组播选项所进行的操作只需五个新的套接字操作。函数 setsockopt()及 getsockopt()用来建立和读取这五个选项的值。IP_ADD_MEMBERSHIP struct ip_mreq 加入到组播组； IP_ROP_MEMBERSHIP struct ip_mreq 从组播组中退出；IP_MULTICAST_IF struct ip_mreq 指定提交组播报文的接口；IP_MULTICAST_TTL u_char 指提交组播报文的 TTL；IP_MULTICAST_LOOP u_char 使报文有效或无效。

以“IP_ADD_MEMBERSHIP struct ip_mreq 加入到组播组”为例：IP 表示 IPv4，_ADD_MEMBERSHIP 表示选项，struct ip_mreq 表示数据类型，加入到组播组表示描述，其余同。

IP_ADD_MEMBERSHIP：若进程要加入到一个组播组中，用 setsockopt()函数发送该选项。该类型是 ip_mreq 结构，它的第一个字段 imr_multiaddr 指定了组播组的地址，第二个字段 imr_interface 指定了接口的 IPv4 地址。IP_DROP_MEMBERSHIP：该选项用来从某个组播组中退出。IP_MULTICAST_IF 该选项可以修改网络接口，在结构 ip_mreq 中定义新的接口。IP_MULTICAST_TTL：设置组播报文的数据包的 TTL（生存时间）。默认值是 1，表示数据包只能在本地的子网中传送。IP_MULTICAST_LOOP： 组播组中的成员自己也会收到它向本组发送的报文。这个选项用于选择是否激活这种状态。如果一个进程要加入到组播组中，要使用 setsockopt()函数在 IP 层设置 IP_ADD_MEMBERSHIP。然后用函数 ip_mc_leave_group()从一个组播组中退出。

C. EtherNet/IP 网络数据的采集及分析

为了深入的理解 EteherNet/IP 协议，验证对 EteherNet/IP 协议及 CIP 协议理解的正确性，并且为 EteherNet/IP 网络节点的开发打下基础，必须通过采集网络上的数据并对采集的数据进行分析实现。这里需要注意的是掌握设备上电后 RSlinx 命令序列，因为它有助于对协议的理解和网络节点的开发，本文给出了主站设备上电后 RSlinx 的通讯命令序列步骤：①定时扫描网络设备。②利用 UCMM 建立连接列表。③读取设备 IP 地址及设备类型。④建立会话的 Session Handle。⑤读取设备的硬件设备信息 。⑥建立 I/O 连接。⑦等待事件的发生。

数据的采集的主要过程是通过 CIPAlyzrODVA 软件。CIPAlyzrODVA 软件是专门用于网络数据通信的数据包采集。在主站上先打开 CIPAlyzrODVA 软件，再打开 RSlinx 或 RSNetWorx；RSlinx 或 RSNetWorks 自动扫描以太网上的硬件设备，整个扫描的数据包被 CIPAlyzrODVA 软件所读取。ControlLogix 与 Flex I/O 的通信不通过主机，在网络上直接通信的。因此 ControlLogix 与 Flex I/O 的数据采集就不能像 RSlinx 或 RSNetWorx 与 Flex I/O 的数据采集一样。可以在 ControlLogix 与 Flex I/O 通信的线路上安装了一个 HUB，并在 Liux 下把网卡设为混杂模式，再输入 tcpdump 命令，这样 Linux 就可以监听所有 HUB 上通过的数据。EtherNet/IP 的网络数据采集完成后，然后使用 TCPDUMP 软件抓获并分析 Ethernet 的 TCP/UDP 网络报文。

报文数据分析内容包括： RSlinx 与 Flex I/O 的数据分析、RSNetWorx 与 Flex I/O 的数据分析、ControlLogix 与 Flex I/O 的数据分析。后两者具体分析略。

RSlinx 与 Flex I/O 的数据分析的具体步骤如下：1、RSLinx 扫描 Flex I/O：设备上电后，首先检查硬件是否已连接，可通过 PC 机进行网络检测来完成。上电后 RSLinx 扫描 Flex I/O 的通信命令。通过观察数据会发现 RSLinx 会发送三条首代码为 01，04，64 的命令来来扫描设备，如果有三条相映的回答指令，则表示连接成功；在这里建立设备显性连接的 Session Handle，Session Handle 在同一个网络上是唯一的；读取设备信息，建立隐性报文连接（I/O 连接），等待事件发生进行数据的交换[6]。在此基础上，分析这三条指令及其应答信号，具体分析略。2、RSLinx 与 Flex I/O 建立一个会话。3、RSLinx 与 Flex I/O 通信。通过这三类报文的分析可以了解 CIP 协议、EtherNet/IP 协议的具体实现过程。

D. Ethernet/IP 协议网络节点的设计

网络节点的设计在 Linux 下的 PC 机中进行，和嵌入式系统相比 PC 机有着更多的开发和调试工具。本次网络节点的设计的核心思想是：①重点在于实现统一的网络通信协议，关键是程序的分层结构模型以及合理调用接口的处理。②使整个网络协议体系既能与通用协议一致，又兼顾到嵌入式系统本身空间小、结构紧凑的特点。此外要考虑整个代码的精简与否、结构安排是否有效。采用多线程编程来优化代码，节省系统调度的时间，提高了系统数据通信的实时性。首先完成的工作是分析 CIP 协议、CIP 协议模型、CIP 协议对象模型。分析 Ethernet/IP 封装体系结构及封装格式，对 TCP 协议、CIP 协议等协议进行封装。之后需要完成 Linux 下的网络编程。必须实现下面三个功能：一是设计的节点和 RSLinx 的通信。主要实现的功能是：确认设备的端口号、IP 地址、供应商、名称及所支持的具体的通信协议等信息。二是设计的节点和 RSNetWorx 通信：主要实现的功能是：把 Flex I/O 的硬件信息读取到网络上，以供其他节点能与之通信。三是设计的节点和 ControlLogix 通信：要实现的功能是：读取 Flex I/O 的硬件信息，并建立 I/O 连接。通过和 Flex I/O 通信，可以得

到 Flex I/O 的部分属性值和能执行的命令等。他们的通信成功标志着节点和服务器的连接成功。

E. 节点实现的软件设计流程图及实验结果

节点在和 RSLinx 的通信时整个过程通过询问和应答信号主要涉及的是 Identity Class 的内容。即连接发送者可能会用到状况列表命令来查找和辨别可能的目的。这个命令将用 UDP 广播的形式发送和不需要确定的会话请求。而每个应答命令接收时，回答都是一个标准的封装报头和数据，一部分数据信息将规定对象列表的一部分信息。回答以广播的形式发送接收者的 IP 地址。整个过程的软件流程图基本框架如图 2 所示。具体通信流程图如 3 所示。

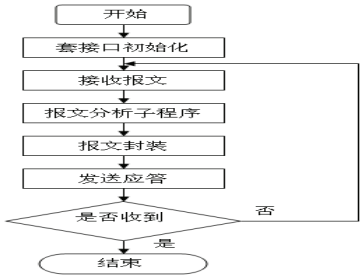

图2 节点和RSLinx通信流程图

Figure 2 communicating flow chart between the node and RSLinx

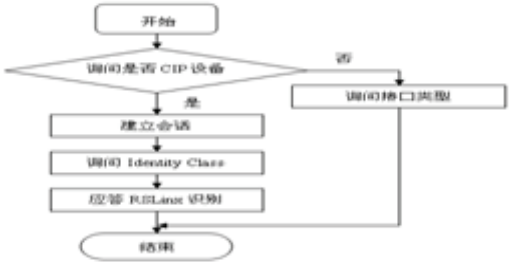

图3 节点和RSLinx具体通信流程图

Figure 3 Specific communicating flow chart between the node and RSLinx

节点与 RsNetWorx 比节点与 RSLinx 通信多一组数据通信，但他们的通信数据相同，并且是通过 UCMM 通信的，UCMM 信息包将会通过 TCP/IP 连接发送数据,主要涉及到的是组态信息。其软件流程图和节点与和 RSLinx 的通信的流程图类似。节点和 ControlLogix 通信的初始阶段也是要建立会话，即要询问设备的状态信息和设备信息，全过程通过指令的询问和应答完成，在和 ControlLogix 通信时主要涉及的是 I/O 连接。I/O 通信是改变数据参数的主要方式，并且 I/O 通信都是通过 UDP 实现的。其软件流程图和节点与和 RSLinx 的通信的流程图类似。本次实验通过 C 语言编程实现了节点和 RSLinx、 RSNetWorx 的通信。在 Linux 的 PC 机上运行开发的源程序，用 Rockwell 软件 RSLinx、 RSNetWorx 扫描网络，可扫描到开发的“192.168.1.30”I/O 节点，节点和 RSLinx 的通信如图 4 所示，节点与 RSNetWorx 通信如图 5 所示。

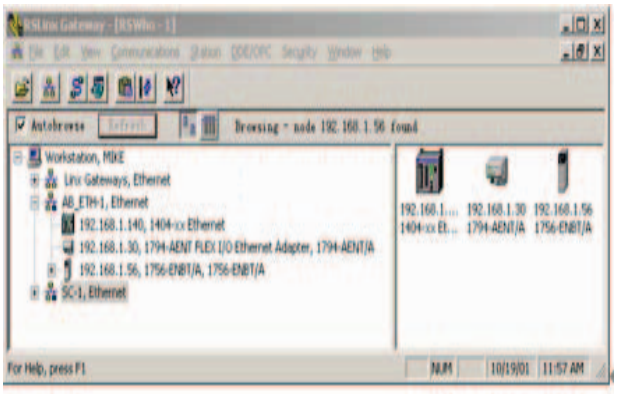

图 4 节点和 RSLinx 的通信

Figure 4 Communication between the node and RSLinx

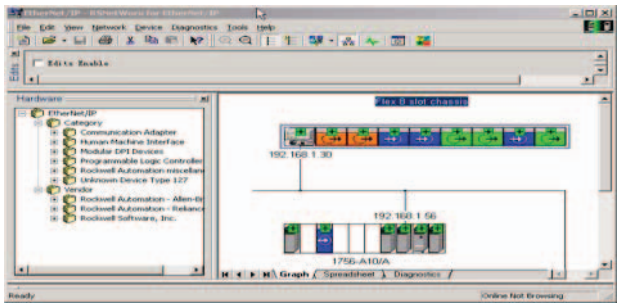

图 5 节点与 RSNetWorx 通信

Figure 5 Communication between the node and RSNetWorx

IV. 结论

给出了 EtherNet/IP 的网络节点开发的设计思想，实现了 EtherNet/IP 网络节点与网络通信的基本功能。本文设计使整个网络协议体系既能与通用协议一致，又兼顾到嵌入式系统本身空间小、结构紧凑的特点。EtherNet/IP 是标准 TCP/IP 以太网和通用工业协议（CIP）的结合，作为一种通信协议的标准，有着其它现场总线所不可比拟的优点。

REFERENCES

[1] HU Yong, ZHENG Lifang, LI Jitang, Research of EtherNet/IP under EPICS Environment. Nuclear Techniques, Vol 29, pp.805-807, November 2006 (In Chinese).

[2] QIU Hao, XUE Ji, XU Zhiqiong, Switch Tchnology and Its Application in Industrial EtherNet/IP. Low Voltage Apparatus, No 11, pp.23-25, November 2008 (In Chinese).

[3] LI Ping, Ethernet/IP Analysis. Journal of Yangtze University (Naturaal Science Edition)Sci&EngV, Vol 7, No 1, pp.254-255, March 2010 (In Chinese).

[4] XU Zhiqiong, EtherNet/IP—An Open Industrial EtherNet. Low Voltage Apparatus, No 10, pp.39-42, 2006 (In Chinese)

[5] ZHANG Lizhong, Design of EtherNet/IP Node. Control and Instruments in Chemical Industry, Vol. 37, pp.100-102, March 2010 (In Chinese).

[6] ZHANG Lizhong, LU Guojian, MA Yongxiang, Exploitation of Smart Node Based on Ethernet/IP Protocol. Instrument Technique and Sensor, No 5, pp.54-55, May 2010 (In Chinese)