The Design of Industrial Ethernet Adapter Based On Ethernet/IP*

T.D.Liu W.Tang and H.Jiang
Department of Automation
University of Xiamen
Xiamen, Fujian Province, China
ltd@xmu.edu.com

***Abstract*—Aiming at application of industrial network, this paper adopted ARM9 microprocessor to construct the hardware system of an industrial Ethernet adapter, developed I/O interface driver under Windows CE.NET operating system and completed the application software of the industrial Ethernet adapter based on EADK software platform. In addition, with the help of EIPScan, EDITT, PCU-ETHIO Ethernet/IP scanner, etc, a complete test was given to the communication adapter. Test results demonstrate that the industrial Ethernet adapter has not only good performance in stability, real-time property of data transmission but also bright prospects in application.**

***Index Terms*—Industrial Ethernet, communication adapter, Ethernet/IP**

I. INTRODUCTION

The distribution, intelligence and informatization of industrial control system require enterprises to be integrated from control layer to management layer seamlessly. Industrial Ethernet satisfies this requirement and turns to become a main direction of the control network. As a mainstream kind of industrial Ethernet, Ethernet/IP gains wide acceptance. Currently, more than 100 million Ethernet/IP nodes have been installed all over the world. Ethernet/industrial protocol has been transformed into China Machinery Enterprise standards [1].

In this paper, an industrial Ethernet adapter was developed. The adapter integrated industrial control devices into Ethernet/IP control network, and improved the flexibility and compatibility of control system. Currently, Ethernet/IP devices are mainly supplied by foreign companies, which mean the domestic enterprises have to pay high fees for those devices. So the industrial Ethernet adapter provided a high-effective kind of industrial control device and filled domestic blank.

II. SUMMARY OF ETHERNET/IP

Ethernet/IP (Ethernet/industrial protocol) is an open protocol system suitable to industrial environment application [2-3]. It integrated standard TCP/IP Ethernet with CIP (control and information protocol) so as to extend standard TCP/IP Ethernet to industrial real-time control. Ethernet/IP introduced the CIP protocol to supply a unified application layer protocol. This enabled each network to realize the seamless integration in application layer. In addition, the Ethernet/IP network could be easily integrated to Internet/Intranet.

III. HARDWARE DESIGN OF INDUSTRIAL ETHERNET ADAPTER

Ethernet/IP has worked out device profile specification to categorize the industrial devices completing the same function. Device profile favored different companies to go into a unified action. Communication adapter is one kind of device class defined by the device profile regulations. (Class number is 0Chex).

Aiming at the development of industrial Ethernet adapter, this paper adopted S3C2410 microprocessor to construct the hardware system of industrial Ethernet adapter, and it architecture is as shown in Fig.1.

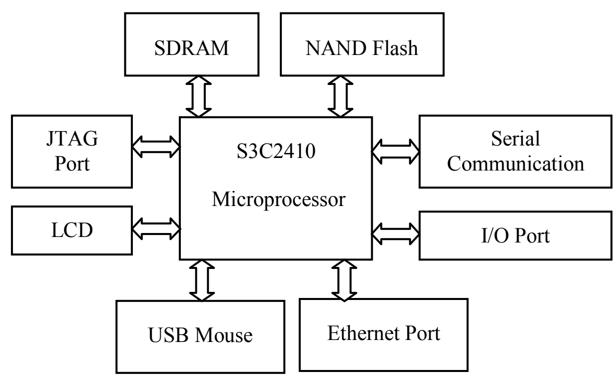

Fig.1 Hardware architecture of S3C2410 based communication adapter

This paper adopted large-capacity NAND Flash and SDRAM so as to assure the system software operate stably and fast. Due to S3C2410 itself not including Ethernet port, Network card chip CS8900 was adopted to realize the Ethernet function. The hardware system of industrial Ethernet adapter also included other parts, such as power, reset circuit, serial bus port, LCD port, JTAG program real-time simulation port and so on.

IV. I/O INTERFACE DRIVER DESIGN OF INDUSTRIAL ETHERNET ADAPTER

Having completed the design and manufacture of the hardware circuit, it is needed to write both stable and reliable I/O port driver so as to provide a software platform independent of hardware. I/O interface driver adopted flow interface driver model and processed data in way of interrupt.

* Supported by Fujian province key scientific and technological project (2008H0060)

978-1-4244-3521-0/09/$25.00 ©2009 IEEE

A key step of this driver development was the implementation of interrupt processing. Interrupt processing was divided into two stages, Interrupt Service Routine (ISR) and Interrupt Service Thread (IST). When hardware happened to interrupt, OS kernel called OAL function OEMinterruptDisable to inform hardware of closing this specific interrupt until certain necessary measures ended. Then, OS kernel called ISR to map the physical interrupt into logical interrupt value, and triggered an event to invoke the suspended IST. Structure of IST is as follows.

```
DIO_Initialize Address(); //virtual memory mapping
DIO_EnableInterrupt();   //relevant register's configuration
while(g_fRun)            //circulate until informed to exit
{……
   dwStatus=WaitForSingleObject(g_hevInterrupt,
   INFINITE);    //wait for synchronized event
   if(dwStatus==WAIT_OBJECT_0)
   {   // undertake interrupt handling at this time
       ……
      // inform inner core that interrupt handling had been
      // accomplished
       InterruptDone(g_dwSysInt);
   }
}
```

When had been started, IST needed to accomplish initialization work, including virtual memory mapping and relevant register's configuration. Then IST entered the status of circularly waiting for interrupt event until the triggering event returned from OS kernel. Once the interrupt happened, IST would communicate with I/O interfaces to read all necessary data. Last IST adopted relevant logical interrupt to call 'InterruptDone' function, and informed OS kernel that the interrupt handling had been accomplished.

Having finished programming of the driver, we also needed to enclose the compiled driver file to operating system image. So far, application program could achieve access and control of the devices through visiting a particular file.

V. APPLICATION SOFTWARE DEVELOPMENT OF INDUSTRIAL ETHERNET ADAPTER

The communication adapter defined in CIP device profile should include identity object, message routing object, connection object and the objects relevant to network connection. As for Ethernet/IP, the objects relevant to network connection included TCP/IP interface object and Ethernet connection object [4-5]. According to the device object model defined by Ethernet/IP, this paper designed application software to accomplish the various functions of Ethernet/IP Communication adapter on the basis of EADK (Ethernet/IP adapter developers kit). EADK is a kind of software protocol stack, and it can make the developers to add Ethernet/IP adapter function to product quickly.

A. Overall Software Architecture of Adapter

Ethernet/IP communication adapter should serve as the client-side and server of UCMM (unconnected manager), explicit message server and I/O server. Among them, UCMM was responsible to process unconnected information and establish connection for the unconnected devices in the Ethernet/IP control network. Explicit message server and I/O server mainly undertook to connect with Ethernet/IP scanners and exchange real-time I/O data. The overall software architecture of this communication adapter is as shown in Fig. 2.

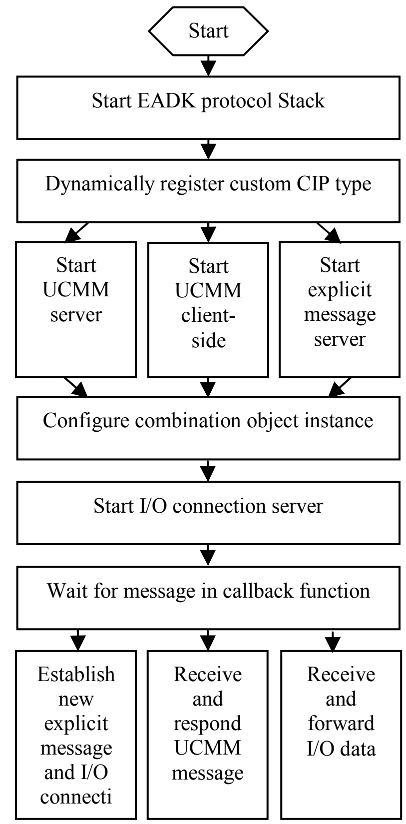

Fig.2 Application software functional structure

When program had begun to run, firstly, initialize EADK protocol stack and dynamically register custom CIP class. Secondly, start server and client-side of UCMM as well as explicit message server so as to send, receive UCMM message and explicit message. Thirdly, according to I/O module connected to this communication adapter, configure matching combination object instance and start I/O server. Finally, make the execution thread of application program wait in the callback function so as to receive the right connection request to establish explicit-message connection and I/O connection with remote Ethernet/IP scanner, Then exchange explicit message and real-time I/O data.

B. Implementation of Application Program

CIP was a connection-oriented protocol, so connection object played a fairly important role in Ethernet/IP. Connection object was mainly responsible for recording every connection's information such as setting value, connection status, etc. In each node, there were at least two connection instances in order to separately process I/O connection and explicit message connection. Each CIP connection can be expressed by a connection object instance. Connection establishment could be achieved by using Create service of connection object or Forward Open service of connection

978-1-4244-3521-0/09/$25.00 ©2009 IEEE

manager object. The both establishment of explicit message connection and I/O connection were accomplished with the help of Forward Open service of connection manager object [6]. Part of header file definition of connection manager object is as follows.

```
// connection manager object class ID
#define CONNMGR_CLASS6
// connection request service
#define FWD_OPEN_CMD_CODE 0x54
……
// initialize connection manager object
void connmgrInit(void);
// process Forward Open connected request
void connmgrIncomingConnection( REQUEST* pRequest);
// send Forward Open response.
void connmgrPrepareFwdOpenReply( SESSION*pSession, CONNECTION* pConnection, REQUEST*pRequest);
// analyze unconnected request (UCMM)
void connmgrProcessUnconnectedSend( REQUEST* pRequest);
……
```

Usually, Ethernet/IP scanner sent Get_Attributes_All service to a certain device's identity object in the network to gain the device's information. I/O module connected to the communication adapter itself did not support Ethernet/IP protocol. Besides, under different occasions, both the number and class of I/O modules connected to the communication adapter were uncertain. So registering a custom CIP class was needed so that combination tool can obtained the information of connected I/O module. The code of this class was defined as 78hex, belonging to vendor specific. This CIP class returned each I/O module's external ID to configuration software by responding to Get_Attributes_All request. In this way, users can make the right configuration and choice on configuration software.

On the basis of having enclosed required CIP classes, application software needed to integrate server and client-side of UCMM, I/O connection server and explicit message connection server into a complete Ethernet/IP communication adapter. Owing to space constraints, here we only show part of implementation code, specific as follows.

```
EtIPAdapterStart();      // start EADK protocol stack
EtIPRegisterEventCallBack((LogEventCallbackType*)&fnCallback);      // register callback function
EtIPRegisterObjectsForClientProcessing((iAppClasses,1,NULL,0);      // dynamically register 0x78 class
……
if(slotsInfo[0].bUsed= =TRUE)
{ // create combination object instance for each I/O module
    EtIPAddAssemblyInstance(
  PRODUCING_ASSEMBLY_SLOT0,
  AssemblyStatic|AssemblyProducing,  INVALID_OFF-
  SET, slotsInfo[0].inputSize,  NULL);
  ……
  switch(nEvent )
  {  // receive I/O data
    case NM_NEW_MEMBER_DATA:
    switch(nParam1)
    {
      case CONSUMING_SLOT0:
          EtIPGetAssemblyInstanceData(
        CONSUMING_SLOT0,OtoTData,
        slotsInfo[0].outputSize);
    ……
```

VI. TESTING AND APPLICATION OF INDUSTRIAL ETHERNET ADAPTER

When having compiled application software of industrial Ethernet adapter, there was another critical job—testing of communication adapter. This paper undertook a complete communication test with the help of EIPScan(Ethernet/IP scanner simulation tool), EDITT(Ethernet/IP device interoperability test tool), Rockwell company's ControlLogix PLC and Woodhead company's PCU-ETHIO Ethernet/IP scanner, etc. Among this software, EIPScan could successfully establish explicit message connection and I/O connection and accurately exchange data with this communication adapter. EDITT was applied to test this adapter's interchangeability with other Ethernet/IP devices. Rockwell's ControlLogix series PLC was able to send Forward Open connection request and establish I/O connection with Ethernet/IP communication adapter. Woodhead Company's PCU-ETHIO CRD card was an integrated Ethernet/IP scanner, and it can give a complete test to communication adapter's function.

Lastly, in order to further verify the stability and reliability of data exchange, EIPScan, ConrolLogix PLC, PCU-ETHIO Ethernet/IP scanner all undertook a 72h-long data exchange with this Ethernet/IP communication adapter at the frequency of timing 20ms. During the test, all CIP connection ran stable with a packet loss rate less than 0.1%. Certainly, the product needs to experience the consistency test in lab appointed by ODVA before being put onto the market.

VII. CONCLUSION

Industrial Ethernet adapter developed by this paper perfectly integrated the embedded developing technique with Ethernet/IP technique. Stable operated I/O interface driver provided solid foundation for the development of application software, and at the same time, implemented CIP protocol on the basis of TCP/IP protocol stack and accomplished various function required by Ethernet/IP communication adapter. This industrial Ethernet adapter provided an open solution to promote industrial users to Ethernet/IP network.

ACKNOWLEDGMENT

The authors wish to express their gratitude to Wenbo Zhou, Changcai Wu and Qiqi Yu for all constructive discussions during the initial phase of this work and to the anonymous reviewers for their helpful comments.

REFERENCES

[1] CH.T.Li, D.Y.Gu and W.J.Meng: Study of Ethernet/IP using trends in e-enterprise (Chinese) [J]. Chinese Journal of Scientific Instrument, 2005, 26(Suppl.2), p.698-700.

[2] ODVA. Ethernet/IP: the proven and complete solution for manufacturing automation [M].2006.

[3] D.Y.Gu and F.H.He: Realization of Ethernet/IP protocol on Linux (Chinese) [J].Chinese Journal of Scientific Instrument, 2005, 26 (Suppl.1), p.441-444.

[4] W.J.Huang and J.X.Jin: Developing a field controller based on the EPA industrial Ethernet (Chinese) [J]. Chinese Journal of Scientific Instrument, 2006, 27(8), p.949-952.

[5] Y.X.Wang and L.D.Wang: Design of multicast and switch gateway between CAN field bus and Ethernet (Chinese) [J]. Journal of Electronic measurement and Instrument, 2007, 21(3). p.83-87.

[6] ODVA. Ethernet/IP specification [M]. Release 1.0, 2001.

 978-1-4244-3521-0/09/$25.00 ©2009 IEEE