2022 The 22nd International Conference on Control, Automation and Systems (ICCAS 2022)
BEXCO, Busan, Korea, Nov. 27~Dec. 01, 2022

Development of Ethernet/IP Adapter for Explicit Messaging in Cooperative Robot Communication

Shaily Bajpai[1], Rahul Ranjan[2], Prof. Sungchul Lee[3], Prof. Kyung oh Lee[3*], Prof. Joongeup Kye[3],

[1] Department of Computer Convergence and Electronics Engineering, Intelligent Robot Research Institute
Sun moon University, Asan-si,Korea

[2] Department of Computer Convergence and Electronics Engineering, Intelligent Robot Research Institute
Sun moon University, Asan-si,Korea

[3] Department of Computer Convergence and Electronics Engineering, Intelligent Robot Research Institute
Sun moon University, Asan-si,Korea

[3*]Department of Computer Convergence and Electronics Engineering, Intelligent Robot Research Institute
Sun moon University, Asan-si,Korea
(leeko@sunmoon.ac.kr) * Corresponding author

[3] Department of Mechanical Engineering, Intelligent Robot Research Institute
Sun moon University, Asan-si, Korea

Abstract: There are many robot industries in the world, but most of them only support Modbus communication. However, Ethernet/IP (EIP) is only supported by some robotics industries. EIP is more flexible than Modbus due to the amount of information exchanged which is wide in range. This paper is study of development an efficient and highly scalable EIP adapter for cooperative robots for the robotics industry. This module is developed to ensure EIP-based intercommunication for explicit messaging among Industrial Automation Machines. Explicit messaging enables acyclic data transfer for communication, and our architecture works without requiring physical components. Further, Ethernet/IP adapters are implemented and tested using EEIP DLL, EADK EDITT, etc. Thus, the architecture is reliable and flexible. Using our EIP module, the scalability and both cyclic-acyclic data transfer ability of the robot system are enhanced. The development time of the communication system can be shortened based on the reuse of the software module. In the future, through effective EIP communication, robot failure diagnosis will be performed based on AI, enabling effective maintenance.

Keywords: Ethernet/IP, Adapter, CIP, Explicit Messaging, PLC

1. INTRODUCTION

In today's era, where connectivity and informatization are seamless, its distribution requires enterprises to integrate it with a better-performing industrial control system, and thus Ethernet/IP has become a savior. As the demand for various industrial communication technologies has emerged, Ethernet/IP has been into vast acceptance due to its simple architecture, compatibility and flexibility. However, EIP is supported only by some large industry, although the standard protocol is open, it is not easy for robot industry to actually implement EIP.

In this paper, we describe the process of the Ethernet/IP adapter development on a virtual system using C++. We can use LS PLC and RB Series Collaborative robot to construct the hardware of Ethernet/IP (Ethernet/Industrial Protocol) adapter. For software development, Windows-based Virtual Box supporting Debian as Virtual Environment to implement I/O interface can be used.

2. ETHERNET/IP

The best interoperability with the present intranet and internet protocols is offered by Ethernet/IP. The network, transport, and application layers of CIP are utilized by Ethernet/IP, Control Net, and Device Net (Common Industrial Protocol. In order to carry CIP communication packets, Ethernet/IP then makes the most of conventional Ethernet and TCP/IP technology. The end result is a single open application layer that sits on top of the freely available, widely used Ethernet and TCP/IP protocols.

3. IMPLICIT & EXPLICIT MESSAGING

Both explicit and implicit communications are supported by Ethernet/IP. Time-sensitive messages are implicit messages. I/O and produced/consumed tags are included in these messages.

The following are some instances of implicit applications:

• Motion control data; functional safety data; and real-time I/O data implicit messages can be unicast or multicast and use the User Datagram Protocol (UDP).

Explicit messages are requests/replies in nature and are not time-critical.

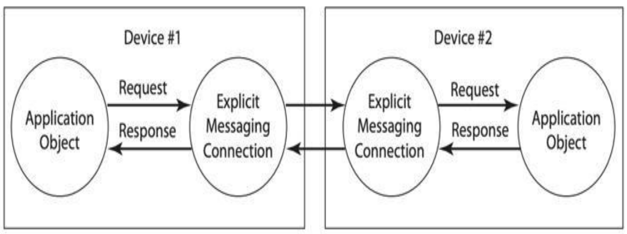

Fig. 1. Message (MSG) Transfer Flow

Examples of explicit connections include running a program upload or an MSG instruction. Explicit describes the fundamental data that is present in each communication (such as the source address, data type, or destination address)

The following are some illustrations of explicit applications:

• RSLinx connectors; HMI

• Program upload/download directions; Message (MSG) instructions TCP is used for explicit messages. Client-server interactions, explicit messages are utilized. It is possible to link or disconnect explicit communications.

By default, an explicit message expires after 30 seconds. The Message (MSG) instruction structure allows the user to alter this option.

4. HARDWARE ARCHITECTURE

For many automation applications, Ethernet/IP networks provide a full range of massages and services. This open network standard supports real-time I/O messaging, information exchange, and general messaging using products from the Ethernet communication standard.

Focusing on the development of Ethernet/IP Adapter, the hardware architecture included industrial components such as LS PLC of Ethernet module: XGL EFMTBS. The hardware architecture of the network system is shown in the below figure.

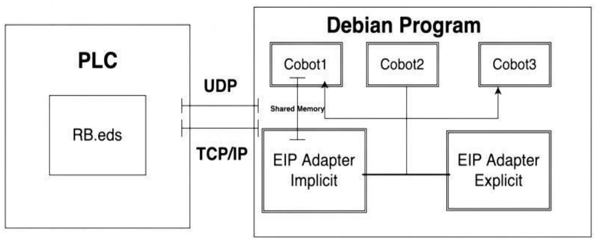

Fig. 2. Hardware Architecture of EIP Adapter

The hardware system of the Ethernet/IP adapter also consisted of other parts like Power, CPU, I/O ports and so on. This hardware architecture is the simplest to achieve and it tests the operability of the control system and field devices connected using Ethernet I/P communication types i.e., Implicit and Explicit.

5. STRUCTURE OF EIP ADAPTER FOR COOPERATIVE ROBOTS

An Ethernet/IP scanner device can access an adapter as a target. A device known as an adapter receives connections from scanners and analyzes explicit and implicit (cyclic) messages. However, the Scanner device, which may or may not be a PC Tool, does not know whether an Adapter Scanner is present at that IP address. It just understands that it wants to connect and obtain its identity information. The only way the target device can react to that explicit message if it is a Scanner is if it behaves like an Adapter. Therefore, it was decided to make all Scanners behave like Adapters, at least when it comes to explicit communications. In the EIP adapter for explicit messaging, the server-side is bound to the Message Router object and has access to all internal resources. The client-side is bound to a client application object and must generate requests to the server. Explicit messages use an explicit messaging protocol in the data portion of the message packet.

Each communication between devices is treated separately by explicit messaging as a query and response. Each request must have a source address, a destination address, and a CIP Connection ID. Each request contains explicit information that the receiving device, or node, decodes and reacts to. Because every explicit message contains destination, source, and connection information, it is less efficient than implicit messaging but enables a higher level of flexibility. Explicit messages are sent using TCP (Transmission Control Protocol).

The server, a field device like a servo drive, can respond when it is ready to receive explicit messages from the client (a controller or HMI, for example). Consequently, explicit messaging is only utilized for non-time-critical information, including diagnostic or configuration data. Devices that reply to messages are referred to as "servers" for explicit messaging, while those that initiate interactions are referred to as "clients."

Explicit messaging communication between the client and server is established using the EIP adapter module. The controller (PLC) is referred to as the client in real-world applications, whereas the field devices are referred to as servers.

6. SOFTWARE DEVELOPMENT OF EIP ADAPTER MODULE

Due to the lack of multiple devices, a virtual machine via Virtual Box is employed to set up a remote server. In that, the virtual environment of Debian is installed and its network configured as a DHCP server. Also, in the virtual box, another virtual environment of simulator program is added to support the simulation of Rainbow Robotics-based RB series robots. The IP addresses are the sample IP address entered to configure the connection of both Debian and other robot simulations.Debian runs free PLC programming software, which enables users to establish Ethernet/IP Communication, writes PLC code, and use the simulator.

6.1. Code Flow for the module

The flow diagram below is an example and written in C#. It is subjected to modification in C++ programming with further enhancements. The code below uses the EEIP library to read/ write digital inputs from an IO-Device.

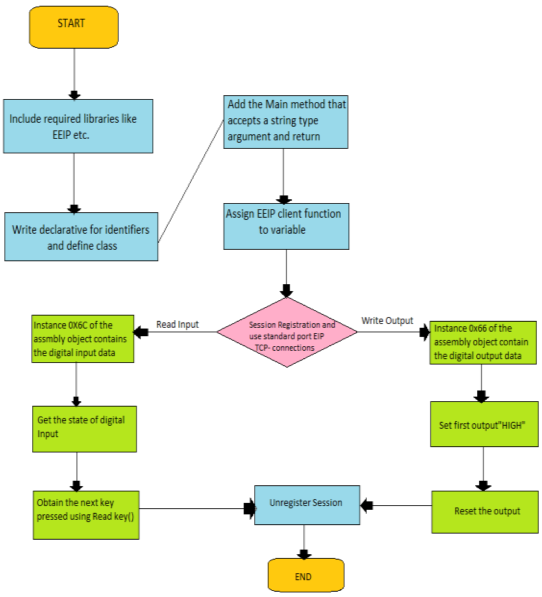

Fig. 3. Code Structure Flow

6.2. Role of the module

The module serves as the client, requesting certain data from the distant server. Client requests for explicit messaging contain all the information required to expressly respond to the message. The code employs the EEIP.NET library. This module thus solves the issues of the control system device performing the function of the adapter, which generally has scanner functionality. This module is developed using the EADK Protocol stack.

6.3. Implementation and testing of module

In order to use the Common Industrial Protocol (CIP) via TCP/IP, the EADK's Ethernet/IP Adapter Class Protocol Stack offers I/O Server, Message Server, Message Client, and Application Programming Interface (API) capabilities. The EADK stack communicates with the application software for your product and the TCP/IP stack that is compatible with your product platform at the socket level. Including assemblies are divided into distinct modules with corresponding functionality in the EADK Adapter Stack source code. This makes it simple to comprehend and debug the source code.Testing is done using EDITT. The PC/Windows-based software program known as the Ethernet/IPTM Device Interoperability Test Tool (EDITT) automates several steps in the Ethernet/IP interoperability Test Procedure.

EDITT offers the capability for Ethernet/IP I/O Server, I/O Client, Message Server, and Message Client. Depending on the connection configuration that the user has chosen, EDITT is capable of initiating a wide range of I/O connections.

7. CONCLUSION

Since the scanner can perform less or no functionality of adapter, the idea of this paper was to develop a specific module for EIP adapter to resolve the explicit messaging issue which the Scanner device couldn't perform. The module designed, not only resolves the scanner-to-be-adapter issue, but is also flexible, and low cost along with fewer resources. The majority of the work has been completed utilizing simulations, a virtual environment, and low-risk alternatives like remote servers and fictitious IP addresses. Future improvements to the module may be made in response to the Ethernet/IP protocol or other emerging technological developments, due to the open resources used in the module.

ACKNOWLEDGEMENT

This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-004)

REFERENCES

[1] Rockwell Automation Publication ENET-UM006B-EN-P - May 2022
[2] T.D. Liu, W. Tang and H. Jiang, "The Design of Industrial Ethernet Adapter Based On Ethernet/IP" 2009 4th International Conference on Computer Science & Education
[3] Paul Brooks, "Ethernet/IP-Industrial Protocol," in *Technology Adoption representing Open Device Net Vendor Association,* 2001-IEEE.
[4] Jiming Chen, Zhi Wang and Youxian Sun, "Real-time capacity analysis of ethernet in industrial application", IEEE Proceedings of the 4th World Congress on Intelligent Control and Automation, vol. 6, no. 2, pp. 227-223, 2002.