

Protocol Gateways

The Better Solution for Protocol
Conversion

Author: Jack Hsu

Table of Contents

Introduction	1
Communication between Different Industrial Protocols	1
Tradition Solution	1
Using an Industrial Protocol Gateway	2
What are the benefits of using protocol gateway?	3
Advantech Protocol Gateway Solution	4

Protocol Gateways: The Better Solution for Protocol Conversion

Abstract

In recent years, the trend in industrial automation is for all equipment, computers, and controllers to be able to communicate with each other to increase productivity, efficiency, and manufacturing quality. In this white paper, we will show different ways of how these industrial devices communicate with each other as well as offering a better solution.

Introduction

Industrial communication protocols are used to establish a client-server or master-slave communication between industrial devices, such as Programmable Logic Controller (PLC) and Remote Terminal Unit (RTU). For industrial automated applications, there are many industrial communication protocols, such as Modbus TCP and PROFINET. Each industrial communication protocol was invented by different vendors and not really designed to communicate with each other. In today's industrial environment, however, we want these devices to talk to each other, so how can we achieve this?

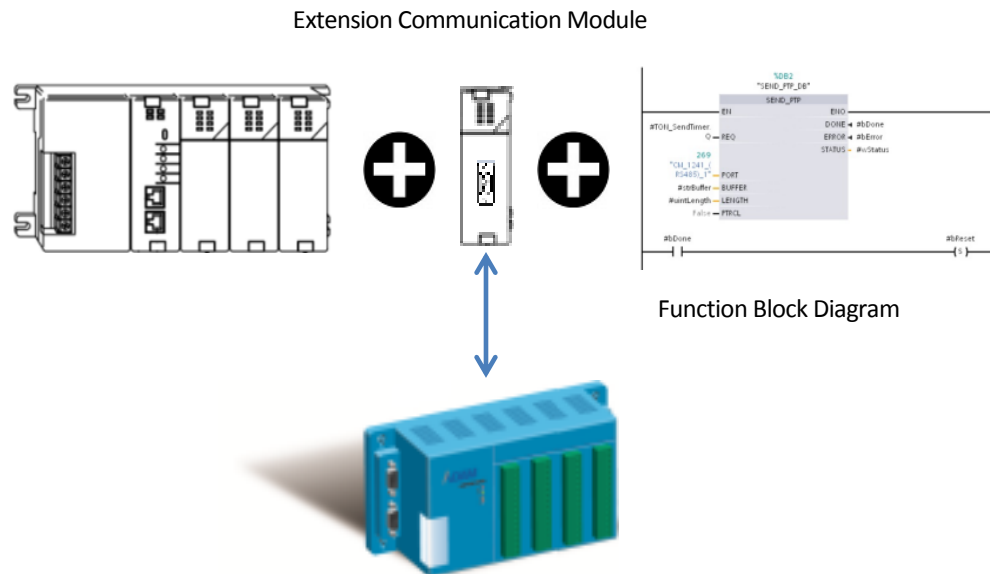
Communication between Different Industrial Protocols

With the progress of technology and the growth of Ethernet networks, industrial communication protocols have followed a steady evolution, such as Siemens PROFIBUS evolved to PROFINET, and Modbus RTU/ASCII evolved to Modbus TCP. From time to time, in order to improve production efficiency and stability, managers may adopt new industrial communication protocols with new automation equipment. But new industrial devices support specific protocols, which mean they may be incompatible with existing legacy equipment and protocols. So how can we make different devices communicate with each other, using an inexpensive solution which is also easy to implement—all of which are important issues for users.

Tradition Solution

In most cases, managers use a PLC to communicate with an RTU, to acquire data, and control the manufacturing process. But, because PLCs often only support vendor defined proprietary protocols, if you want a PLC to support other protocols, you need to use extension communication modules in the PLC. Siemens PLCs for example need extension communication modules to support Modbus RTUs. In this case, a user must input the

function block diagram data of the new protocol into the PLC and enter the correct parameters for the data.



The main advantage of using PLC extension communication modules is reliability. However, the disadvantages of using PLC extension communication modules are:

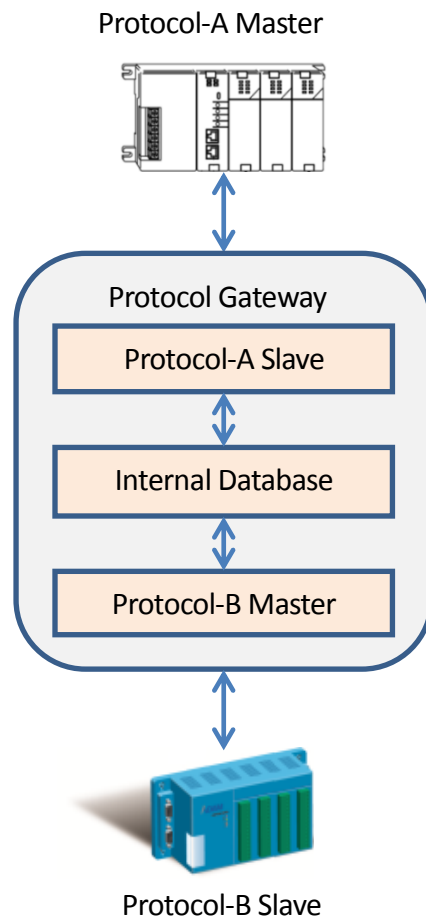
- The high cost of PLC extension communication modules.
- Each PLC function block diagram occupies PLC memory and PLC memory is expensive.
- It can be difficult to input a PLC function block diagram, and users must be familiar with the characteristics of different protocols and settings in order to correctly implement it.
- There may not even be a suitable PLC extension communication module that supports your desired protocol.

Using an Industrial Protocol Gateway

Currently, we have another suitable solution called a protocol gateway. A protocol gateway is a device that converts from one protocol to another to allow communication between devices.

The general architecture of protocol gateway includes a protocol-A slave, a protocol-B master and an internal database. One normal scenario for a protocol gateway is the protocol-B master communicates with a remote device, gets that device's data, and keeps it in the internal database. When remote protocol-A master, such as a PLC, asks the gateway about the data, gateway protocol-A's slave obtains the data from its internal database, and sends it to remote protocol-A master.

Another typical scenario for a protocol gateway is the remote protocol-A master sends data or command to gateway's protocol-A slave. Protocol-A slave passes the data or command to protocol-B master, and triggers protocol-B master to send the data or command to a remote device.



With the different protocols, the physical interface of protocol gateways may be different, such as the physical interface of DeviceNet is CAN bus, however the physical interface of Modbus RTU is RS-232/422/485.

What are the benefits of using protocol gateway?

As mentioned earlier, when using PLC extension communication modules, the user must program the PLC for the new industrial protocol. And in general, this is difficult for normal users. But if a protocol gateway is chosen, the user will not need to worry about how to write PLC programs for the new protocol. The market has many protocol gateways, which support

a variety of industrial protocol conversions so a user can easily find a suitable protocol gateway to meet their conversion requirement.

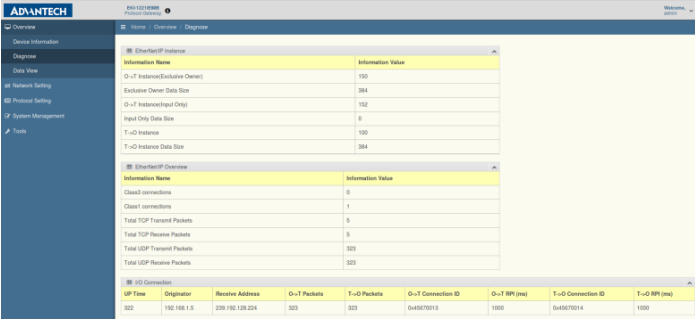
There is also another important benefit of using a protocol gateway. Good protocol gateways have diagnosis and troubleshooting tools, which can help users to easily install industrial

devices and help them with problem solving. For example, when an industrial device status or data is incorrect, you can use the diagnosis tools to confirm connection status, and through the data monitoring and analysis tool you can capture and analyze data sent from the device to figure out what the problem is.

Advantech Protocol Gateway Solution

Advantech provides a variety of Protocol Gateway solutions that support all major industrial protocols, such as PROFINET, EtherNet/IP, EtherCAT and Modbus TCP/RTU. Advantech protocol gateways use the highest quality components to enable them to operate in temperatures of between -40 and 75°C. To optimize protocol conversion, and reduce maintenance cost, Advantech protocol gateways also support these features below.

- High speed conversion of data between different protocols provides runtime protocol information, such as connection status of peer-to-peer industrial devices to ensure everything is running smoothly.



The screenshot displays the Advantech protocol gateway software interface. It features a navigation menu on the left with options like 'Overview', 'Device Information', 'Diagnosis', 'Data View', 'Network Setting', 'Protocol Setting', 'System Management', and 'Tools'. The main content area is divided into sections for 'EtherNet/IP Instance' and 'EtherNet/IP Overview'. The 'EtherNet/IP Instance' section shows a table with columns for 'Information Name' and 'Information Value'. The 'EtherNet/IP Overview' section shows a table with columns for 'Information Name' and 'Information Value'. At the bottom, there is a table for 'ICD Connections' with columns for 'UP Time', 'Originator', 'Receiver Address', 'O-T Packets', 'T-O Packets', 'O-T Connection ID', 'O-T RPR (ms)', 'T-O Connection ID', and 'T-O RPR (ms)'. The table shows one connection with an originator of '192.168.1.5' and a receiver address of '209.160.128.224'.

Information Name	Information Value
O-T Instance(Exclusive Owner)	100
Exclusive Owner Data Size	384
O-T Instance(Output Only)	192
Input Only Data Size	0
T-O Instance	100
T-O Instance Data Size	384

Information Name	Information Value
Class1 connections	0
Class1 connections	1
Total TCP Transmitted Packets	5
Total TCP Received Packets	5
Total UDP Transmitted Packets	323
Total UDP Received Packets	323

UP Time	Originator	Receiver Address	O-T Packets	T-O Packets	O-T Connection ID	O-T RPR (ms)	T-O Connection ID	T-O RPR (ms)
322	192.168.1.5	209.160.128.224	323	323	0405670013	1000	0405670014	1000

- Provides protocol traffic monitoring tools to instantly capture and clearly present protocol data. With this information, end users can ensure the accuracy of peer-to-peer industrial devices and discover problems when they arise.

Status	Time	Event	Capturing...																	
34	0h:0m:2s:867	3	Mapping 3	Send	10	3	12	00 00 00 00												
35	0h:0m:2s:868	3	Mapping 3	Receive	10	3	19	09 C1 00 00 00 0A 03 00 95 00 05												
36	0h:0m:2s:867	5	Mapping 5	Send	10	16	29	00 C2 00 00 17 0A 10 00 C7 00 04 10 00 00 00 00 00 00 00												
37	0h:0m:2s:868	5	Mapping 5	Receive	10	16	12	09 C2 00 00 06 0A 10 00 C7 00 08												
38	0h:0m:2s:967	6	Mapping 6	Send	10	3	12	09 C3 00 00 06 0A 03 01 2B 00 02												
39	0h:0m:2s:968	7	Mapping 7	Send	10	16	77	09 C4 00 00 47 0A 10 00 F9 00 20 40 00 00 00 00 00 00												
40	0h:0m:2s:969	6	Mapping 6	Receive	10	3	13	09 C3 00 00 07 0A 03 04 2B 07 00 00												
41	0h:0m:2s:970	7	Mapping 7	Receive	10	16	12	09 C4 00 00 06 0A 10 00 F9 00 20												
42	0h:0m:2s:967	1	Mapping 1	Send	10	4	12	09 C5 00 00 06 0A 04 00 63 00 0A												
43	0h:0m:2s:968	1	Mapping 1	Receive	10	4	29	09 C5 00 00 17 0A 04 14 00 00 00 00 00 00 00 00 00												
44	0h:0m:2s:012	4	Mapping 4	Send	10	3	12	09 C6 00 00 06 0A 03 00 45 00 08												
45	0h:0m:2s:013	4	Mapping 4	Receive	10	3	25	09 C6 00 00 13 0A 03 10 00 00 00 3A 02 00 00 00 00												

- User-friendly WEB design to help users quickly and easily establish a mapping table between different protocols, for example, users can duplicate existing protocol mapping to create a new mapping entry and shorten the time of deployment.

ADANTECH											
Modbus/TCP Setting											
Start-up Mode: Running											
Modbus/TCP unit: France Data											
Modbus Commands											
Allocated input size: 102 bytes, output size: 100 bytes											
Index	Name	Item ID	FC	Address/Quantity	Trigger	Scan Interval	Data Smap	Response Timeout	I/O Disconnect	Task Value	
0	Mapping 1	1	4	Read Address 100, Quantity 10	Cyclic	1000	None	Enabled	1000		
1	Mapping 2	2	16	Write Address 20, Quantity 10	Change	1000	None	Enabled	1000	France Data	
2	Mapping 3	3	16	Read Address 100, Quantity 5	Cyclic	1000	None	Enabled	1000		
3	Mapping 4	4	16	Read Address 70, Quantity 8	Cyclic	1000	None	Enabled	1000		

ADANTECH											
Mapping Overview											
Input Data Byte buffer view											
Modbus/TCP - Master											
Name	Function Code	Data Size(Byte)	Byte range								
Mapping 1	4	20	0 - 19								
Mapping 2	16	20	20 - 39								
Mapping 3	16	10	40 - 49								
Mapping 4	16	4	50 - 53								
Output Mapping Overview											
Output Data Byte buffer view											
Name	Function Code	Data Size(Byte)	Byte range								
Mapping 1	16	20	0 - 19								
Mapping 2	40	40	20 - 79								
Mapping 3	16	10	80 - 89								