EATON | Cutler-Hammer

Canvas™ and ePro™ PS Communications to Allen-Bradley® PLCs Over Ethernet

Product Application AP04801006E

Effective June 2008

Introduction

There are two different protocols used with Allen-Bradley® PLCs over Ethernet. The first is commonly referred to as Allen-Bradley Ethernet, which is used to communicate to older PLC-5 and SLC 500 series of controllers. This protocol is essentially a TCP/IP encapsulation of the DF1 serial protocol and is a fairly basic communications protocol. The KEPServer_ePro OPC server that comes with Canvas software includes this protocol in its list of drivers and it fully supports all features of the protocol.

The second protocol is Ethernet/IP, where the IP stands for "Industrial Protocol" (not to be confused with the IP of TCP/IP which stands for Internet Protocol). Ethernet/IP, Modbus® TCP and ProfiNet are the three most common Ethernet protocols used in industrial applications. Ethernet/IP is a modern, full-featured protocol, is the fastest growing Ethernet protocol in North America, and is often described as DeviceNet over Ethernet TCP/IP and UDP (User Datagram Protocol) transport protocols. In truth it is really CIP, Common Industrial Protocol, over Ethernet TCP/IP and UDP; the DeviceNet protocol is a subset of CIP. This application note will focus on how Canvas and the PanelMate® ePro PS communicate with Allen-Bradley PLCs over Ethernet using this protocol. This includes the ControlLogix®, CompactLogix, FlexLogix and MicroLogix™ PLC families.

When you compare the features, capabilities and efficiency of Canvas and ePro communications to the Allen-Bradley Logix family of PLCs, it is clearly the best total solution available in the marketplace and superior to Rockwell's own drivers contained in the RSLinx® and FactoryTalk® products.

Ethernet/IP — Scheduled vs. Unscheduled Messaging

There are two distinctly different ways Ethernet/IP communications are implemented. They are commonly referred to as implicit, or scheduled messaging, and explicit, or unscheduled messaging. Implicit messaging is used in a Master/Slave mode where the master, almost always the PLC, scans a predefined list of I/O devices, reading inputs and writing outputs to the slave devices for high-speed control needs. This part of the protocol closely matches the DeviceNet command set and uses UDP as the transport mechanism because it requires less communications overhead and is therefore better suited for high-speed I/O.

Explicit Messaging is used in a peer-to-peer mode where data reads and writes are not pre-determined, but occur asynchronously as needed. This is commonly used for higher-level information needs in SCADA (Supervisory Control and Data Acquisition), Operator Interface and MRP/ERP (Manufacturing Resource Planning/Enterprise Resource Planning) applications, and uses TCP as the delivery mechanism to guarantee message receipt. **Figure 1** shows this graphically.

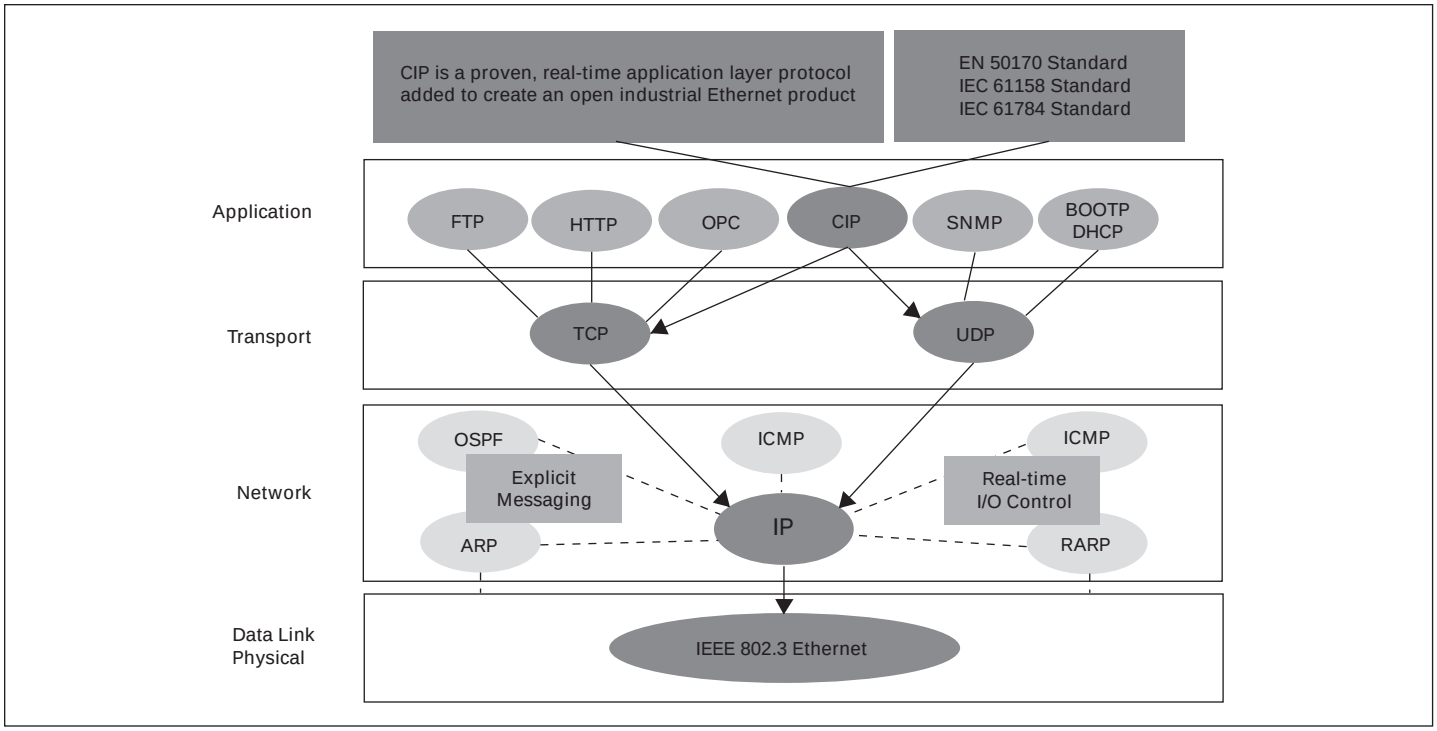

FIGURE 1.

Input/Output devices and simpler, more basic operator interface devices, such as text displays or simple annunciators that communicate on Ethernet/IP, typically support just the scheduled messaging portions of the protocol. Eaton products like the SVX9000 drives and EZ relays support Ethernet/IP in this manner, as does the D77D-EIP communications network adapter. This allows the Master PLC to read all I/O from those devices and write outputs to control them through logic in the PLC.

Canvas and PanelMate ePro PS use the KEPServer_ePro OPC server for Ethernet/IP communications. The ControlLogix Ethernet driver supports all unscheduled messaging commands for all Allen-Bradley Logix family PLCs over Ethernet/IP, as well as a number of special Allen-Bradley PLC commands that optimize communications and simplify configuration. The ControlLogix Ethernet driver does not support scheduled messaging, which means that the ePro PS cannot replace the PLC as the Master for I/O polling, instead it reads information from the PLC and writes operator commands to the PLC using unscheduled messaging.

There are three critical areas of differentiation when it comes to communicating with Logix PLCs. The first is how well the software is able to easily create or populate a list of tags from the PLC. The second is whether the driver supports all of the possible datatypes and tag syntax, and the third critical area is efficiency and speed of communications.

Interfacing to Tags in the PLC

There are four ways operator interface manufacturers provide tags when editing applications with Allen-Bradley Logix family of PLCs. They are:

1. Use the older DF1 protocol and map Logix tags in RSLogix 5000 software to SLC or PLC-5 file numbers. By creating a cross-reference table in the RSLogix ladder logic software, you can use the old DF1 protocol and use the old addressing structure to read values from the PLC. This method was implemented when the ControlLogix PLC was first introduced to the market to provide a simple option to connect using the legacy DF1 protocol. After more than eight years of ControlLogix availability, most operator interface manufacturers have created a true Ethernet/IP driver and tag interface to avoid this cumbersome method.
2. Import tags from an Excel® compatible CSV file format created by RSLogix 5000 software. RSLogix 5000 is the logic editing software for all Logix PLCs, and it supports an export to a tag file in CSV format function through the **Tools>Export** menu. Unfortunately, the resulting format leaves much to be desired for importing tags into an application. This is because it doesn't expand array variables, and it doesn't expose the structure components of either pre-defined or user-defined (UDT) structures. A lot of work would have to take place after the CSV export to provide a complete pick-list of tags for easy operator interface editing.
3. Import tags from an L5K formatted ladder logic file. RSLogix 5000 programs are saved in a binary coded file with an ACD file extension. The user, however, can use **File>Save As** and select the alternate ASCII file type, which has an L5K file extension. This method is preferred over the first two because all tags, all datatypes, and all structures are described in the L5K file. By using the L5K file, the software can be configured to accurately populate a pick-list of all tags from the controller. Plus the operator interface developer doesn't have to be physically connected to the PLC to generate a tag list; the L5K file is all that is needed. The downside of this method is that the user has to ensure that the correct date/version of the ladder logic program's L5K is used. And when tags are added, modified or removed from the logic program, the new file must be generated and re-imported to keep both the operator interface application and the ladder logic program in sync. The KEPServer_ePro OPC server uses this as an alternate way of importing tags.
4. The simplest way for a developer to populate a tag list is for the software to send a special command to the Logix PLC requesting it to transmit its current tag list over an Ethernet connection. The software must then parse through the response to create the tag list. This method ensures that the list is current and accurate. The challenge is that these special Logix commands are not documented or shared by Rockwell® and not considered part of the open feature set of Ethernet/IP. Fortunately, Kepware's software engineers were able to determine the correct command and response syntax to be able to upload the entire tag database map from the Logix family of PLCs, making it the easiest and most accurate means of providing tags to an operator interface application such as Canvas. The Canvas software simply browses the KEPServer_ePro drive for available tags. As an alternative the L5K file method is also supported when a physical connection to the PLC isn't available.

Tag Datatype Support

There are two categories of tags in an RSLogix 5000 tag database, Controller tags and Program tags. Controller tags are common (global) to all programs in the PLC and Program tags are local to a specific program. Within these two categories there are many datatypes permitted in Logix PLCs:

- Atomic types — These are the basic elemental data types that most PLCs support. They include Integer (INT), Double-Integer (DINT), Signed-Integer (SINT), Floating Point (REAL) and logical (BOOL) types.
- Multi-dimensional array types — All of the atomic types may also be used in arrays of 1, 2 or 3 dimensions.
- Structures — There are many pre-defined structures available, including Timers, Counters, Strings and PID, plus over 80 motion and function specific structures that greatly simplify the logic developer's job of creating a tag database for a wide range of applications.
- Used-Defined Types (UDTs) — Users can create their own structures within a program. These can be very useful and can greatly simplify database definition and tag organization.

The KEPServer_ePro ControlLogix Ethernet driver supports all of the possible data types and all tag syntax with no tag count restriction. Basically, it can read and write to any tag in the Logix family of products and seamlessly provide a complete pick list of tags for use in the Canvas editing environment.

Communications Efficiency

There are several aspects of communications efficiency where the KEPServer_ePro ControlLogix Ethernet driver is clearly superior to competitive communications drivers.

Connection Consumption: There are 64 simultaneous connections available on a ControlLogix or CompactLogix Ethernet module. By default, KEPServer_ePro consumes only one connection per Canvas/ePro PS device. It is possible to configure multiple connections to increase update performance, but it is seldom necessary to do so unless the number of continuously read tags is very large. This is typically only needed for very large data archiving application requirements. By contrast, a PanelView Plus unit or a FactoryTalkView (formerly called RSView) ME or SE session running on a Windows®-based PC uses FactoryTalk and RSLinx and consumes six connections. Rockwell's documentation warns that no more than six PanelView Plus units or FactoryTalkView applications should be communicating to a ControlLogix PLC at one time. This is a serious limitation in many industrial information applications. One PanelMate ePro PS customer is currently in the middle of integrating a large US military project where 24 ePro PS units are simultaneously communicating to a single ControlLogix PLC.

Symbolic Addressing vs. Physical Addressing: The average length of a tag name in a typical ControlLogix application exceeds 40 characters. When a communications driver creates a request to read data, the packet length limitation of a TCP/IP request becomes a factor that can limit how many values can be specified in a single read. This means that if the driver only supported symbolic addressing and wanted to continuously read 50 tags, it may need to generate two or three requests to get all the tags read. On a medium or large sized application, the cumulative effect can significantly bog down network communications.

While symbolic addressing is supported, physical addressing is the default mode of the KEPServer_ePro ControlLogix driver. This is accomplished by reading the tag/memory map from the PLC when the driver starts up (another of the non-published commands that Kepware's engineers were able to decode). Once the tag/memory map is read, all tags are converted to a 4 byte physical memory address, for example 2F8043A2. The ability to use physical addressing means that on average it can read 9 to 10 times more data in a single request than a driver that uses symbolic addressing. The other benefit to using physical addressing is that it removes the burden of tag to memory address lookup at the PLC. The extra CPU cycles saved by reading memory directly allows the PLC to spend that time reading I/O and solving logic rather than supporting external communications. The only other driver available on the market that supports physical addressing is Rockwell's FactoryTalk/RSLinx.

Communications Optimization: It is impractical and unnecessary for an operator interface or SCADA application to continuously read all the tags from a PLC. In many ControlLogix applications, the tag table can have 15,000 or more tags. There are many large applications with well over 100,000 tags in a network of one or more ControlLogix PLCs. Typically, less than 10% of the total available PLC tags are used by an operator interface or SCADA application. For the sake of communications efficiency, an OI/SCADA application reads only those tags that are required to support foreground and background functions. Foreground functions are based on visual data requirements and are limited to tags that support the current page/screen of information that the operator is viewing. Examples of background functions are alarm and event monitoring, real-time trending and historical trending and archiving. Tags that are required for background functions are read at all times, but page related tag requirements change when the user changes pages on the operator interface or SCADA system.

For maximum network communications speed and efficiency, operator interface and SCADA communications drivers incorporate algorithms that optimize the way read and write packets are constructed to minimize the number of packets needed at any one time. In modern drivers, these algorithms are designed to run whenever data requirements change. This happens when the operator changes pages or when a new local (same PC) or remote (networked PC) application connects to the driver (OPC server) and requests data. The speed with which the optimization algorithm resolves the optimal way to satisfy the new data needs determines how quickly data begins to update when an operator changes pages. This is where KEPServer_ePro really outperforms competitive drivers. On applications with a small number of tags the difference is negligible, but on large applications with many tags, the time to get fresh data after a page change can be dramatic.

Summary

When selecting an operator interface or SCADA system to communicate to the Allen-Bradley Logix family of PLCs, make sure you evaluate all aspects of the communications architecture, including:

- **Communications Mode:** Explicit, unscheduled messaging incorporated in the KEPServer_ePro driver is much easier for the operator interface and logic developer to set up and doesn't require any programming in the PLC. If the driver only supports implicit, scheduled messaging this must be configured in the PLC, much like the older Allen-Bradley Remote I/O, and other Master/Slave protocols where the PLC is the master of the I/O network.
- **Tag Interface:** The KEPServer_ePro and Canvas tag interface is easy and seamless, which saves development and debug time. If a development software package doesn't have a clean mechanism for populating tags, it can cost the user many hours of development time and lead to mistakes that lengthen the startup and debug phases on a project.
- **Datatype Support:** KEPServer_ePro supports all Logix family data types. This means that no support is required in ladder logic to move data to operator interface-specific tags. If a driver doesn't support all data types, the developer is left to create operator interface-specific tags and program data moves from unsupported types to supported types in ladder logic, which slows down the development process.
- **Communications Efficiency:** KEPServer_ePro is the most efficient driver for Ethernet/IP communications. An inefficient driver can bog down the network, reduce data updates, and negatively impact the overall project success.

For more information on the PanelMate ePro PS and Canvas Software, please visit www.eaton.com/electrical.

PowerChain Management®

Eaton Corporation
Electrical Group
1000 Cherrington Parkway
Moon Township, PA 15108
United States
877-ETN-CARE (877-386-2273)
Eaton.com

Printed in USA
Publication No. AP04801006E / Z7206
June 2008

EATON | Cutler-Hammer