



REAL TIME AUTOMATION



The Valve and Valve Positioner Ethernet/IP Implementation Guide

PREPARED BY REAL TIME AUTOMATION
WWW.RTAAUTOMATION.COM



Valve EtherNet/IP Implementation Guide

If you're a product manager, software developer or hardware designer investigating **EtherNet/IP™** or **Modbus TCP/IP** for a Valve or Valve Positioner product, I've written this guide just for you. It's everything that you need to know before you get started; what development path options are open to you and what some of the pitfalls are in the development process for each of those development paths.

My name is John Rinaldi and I, occasionally with tongue planted firmly in cheek, call myself the Doctor of Industrial Networking. I've been doing this kind of stuff now for going on 30 years and working with EtherNet/IP for almost 12 years.

My company is Real Time Automation. We provide EtherNet/IP and Modbus TCP/IP technology to developers building valves, barcode readers, weigh scales, robots, I/O devices, Controllers of all sorts of devices. You'll find our technology in Pharmaceuticals, Food and Beverage, Aerospace, Automotive and Material Handling. You name an industry and application and I'll find a solution that uses our EtherNet/IP or Modbus TCP/IP technology. A lot of the name brand EtherNet/IP and Modbus TCP/IP devices are powered by our EtherNet/IP software. So, if you're thinking about implementing EtherNet/IP or Modbus TCP/IP for your Valve or Valve Positioner you've come to the right place. I've been working with automation people and software developers on these solutions for a long time and I know all the possible solutions, all the pitfalls and all the arguments.

In this paper I would like to give you the background information that you have to know before you get started. You'll probably find that you know most of it but like anything else, it's the one thing that you don't know that can hurt you. After that we'll go through the checklist of what you're going to need. After that you'll learn the kinds of things you need to consider as you select a technology provider.

Important Things You Should Know Before Getting Started

1. What Is A TCP/IP Stack and Where Do I Get It?

A TCP/IP Stack is software that provides the basic software communication link on the Ethernet. Every single Ethernet device must have some sort of TCP/IP stack that can correctly transfer low level messages between nodes on the Ethernet. A TCP/IP stack consists of many communications protocols. You do need a TCP/IP stack if you are going to implement Ethernet/IP or Modbus TCP/IP. That stack will not usually come from your Ethernet/IP or Modbus TCP/IP provider.

There are a few TCP/IP protocols that are very important to Ethernet/IP or Modbus TCP/IP and should probably already be familiar to you:



IP (Internet Protocol) – The Internet Protocol is the most basic protocol. It simply moves a packet directly from one Ethernet device to another Ethernet device. Almost everything moves across the internet using IP.

TCP (Transport Control Protocol) – TCP is a protocol that creates a connection between two nodes over Ethernet. TCP maintains a connection and uses acknowledged messaging to transfer messages between two devices. That means that all TCP messages generate a reply. If there's no reply, there's an error.

UDP (User Datagram Protocol) – UDP is a protocol that is connection-less. Using UDP an application just sends a message into Ethernet. It might get there or it might not. You only use UDP if you don't really care.

Where can you get a TCP/IP stack? TCP/IP stacks are included if you're using an RTOS. If not, a lot of the hardware vendors provide one. If yours doesn't you may have to look for a vendor but note this warning. There are a number of disreputable vendors selling stuff they didn't build and providing little to no support.

*NOTE: There is a list of features that you should consider when choosing a stack. If you'd like the list go to my home page, <http://www.rtaautomation.com/> and hit the Contact Us button. I'll send it to you in a day or so.

2. What if I don't currently have Ethernet on my valve?

Well, Ethernet is obviously required to implement any of the Ethernet-based application layer protocols (EtherNet/IP, Modbus TCP/IP and Profinet IO). You really have just two choices. Redesign your valve controller and use an Ethernet equipped micro controller or add a PCB to your valve to handle the Ethernet. Contact Real Time Automation for a detailed discussion of the best approach for your product line.

2. What is an Application Layer Protocol?

Ethernet/IP or Modbus TCP/IP are application layer protocols. An application layer protocol is a well-understood way of decoding the data that crosses a wire. On Ethernet, we typically move packets using TCP/IP (Transport Control Protocol) or UDP (User Datagram Protocol). These are protocols that move bytes from one node to another node in a well-defined way that every device in the Ethernet universe understands.

The content of these packages is the Application Layer protocol. It's like me dialing somebody in Thailand. I can get the connection (similar to a TCP/IP or UDP connection) and we can exchange sounds (packets) but it's meaningless unless we have a common way of understanding that data. An application layer protocol is the common way of understanding the data in a TCP/IP or UDP packet. EtherNet/IP is one way of interpreting the contents of those packets.



There are lots of other Application Layer Protocols. Here are three that you are probably familiar with:

FTP (File Transfer Protocol) - Moves files across the Ethernet

SMTP (Simple Mail Transfer Protocol) – Moves mail across the Ethernet

3. What is CIP?

CIP is the Common Industrial Protocol. It is the basic component that forms the basis for DeviceNet, CompoNet, ControlNet and EtherNet/IP. There are two parts to CIP, an Object/Attribute Library and a Messaging structure.

CIP devices use a very specific data model based on Objects (related collections of data) and Attributes (Data Identifiers). From a CIP network, every CIP device is assumed to consist of a set of Objects and Attributes. There are required Objects and Application Objects. Required objects are in every device. Application objects are specific to a specific target application.

CIP devices use a very specific data model based on Objects (related collections of data) and Attributes (Data Identifiers). From a CIP network, every CIP device is assumed to consist of a set of Objects and Attributes. There are required. Objects and Application Objects. Required objects are in every device. Application objects are specific to a specific target application.

Objects are composed of related attributes. An attribute is some sort of data name and value. The Identify Object is a required object that contains the attributes associated with device identity. The Identify Object appears in ALL CIP devices. You'll find it in DeviceNet devices, ControlNet devices, EtherNet/IP devices and CompoNet devices. The Identity object is composed of identity attributes like vendor name, product name, catalog number and more.

CIP devices have a very specific set of messages that they support. There are Explicit Messages and Implicit Messages. Explicit messages are sporadic, one-time messages requesting data values or setting parameters. Implicit messages are ongoing I/O messages that are exchanged between two CIP devices.

4. What is a Scanner (Client) and what is an Adapter (Server)?

In the Modbus TCP/IP world, Client devices are devices that create connections with one or more Server devices. The Client devices send read and write messages to the Server devices. Clients can connect to multiple Server devices and request data from all of them. Server devices usually support a connection with a single Client but may offer multiple Client connections. When multiple Client connections are supported, only one connection is allowed to write data points in the Server.

In a CIP network, an EtherNet/IP Adapter, sometimes known as a Server, is an end-device in an EtherNet/IP network. I/O blocks, valves, barcode readers, photoeyes, even sophisticated devices like cameras and drives are all Adapter devices. Adapter devices do nothing until a Scanner device requests a connection. Your Valve or Valve Positioner will be an EtherNet/IP Adapter device.



5. What is the Difference Between EtherNet/IP and Modbus TCP/IP?

First let's look at the similarities. Both EtherNet/IP and Modbus TCP/IP are Ethernet Application Layer protocols that use the TCP and IP protocols of the TCP/IP stack. They are both highly structured ways of moving data between automation devices in an industrial application.

The difference is in how data is represented to the network. EtherNet/IP represents data using the CIP object structure. Modbus TCP/IP represents data as a series of registers (16-bit unsigned values) and coils (bits). The other major difference is in the support from major vendors.

6. How Does ControlLogix and CompactLogix use EtherNet/IP?

A Logix processor has an EtherNet/IP Scanner module that manages Explicit and I/O connections to Adapter devices. How the Logix processor interfaces with the Adapter module is dependent on whether Rockwell or a 3rd party makes the Adapter. For RA Adapters, the Logix processor knows a lot about the Adapter device and is capable of understanding the size of the Input and Output areas of that device.

For non-RA Adapter devices, the Logix processor must be configured with the IP Address of the Adapter and the number of bytes of transferred from the device (inputs) and the number of bytes to send to the device (outputs). This configuration is done with RSLogix Software.

7. Can a Logix Processor be used as an EtherNet/IP Adapter?

No. The Logix Processor supports all the required objects of an EtherNet/IP Adapter but no application objects. A Scanner device can open an Adapter-type connection to the Logix processor and send EtherNet/IP Explicit Messages to the processor to read any attribute from any of the required CIP objects. That means that you can open a connection, read the Identify information, read the attributes of the Ethernet Object or the TCP/IP object but you can't read anything having to do with the data table of a Logix processor.

A Logix processor also doesn't support any type of I/O messaging. There is no way to access the data table of a Logix processor using EtherNet/IP I/O messages.

And no, the Logix Processor doesn't directly support Modbus TCP/IP. You will need some sort of gateway device to incorporate a Logix Processor into a Modbus TCP/IP network.



8. How Are Legacy Rockwell processors used with EtherNet/IP or Modbus TCP/IP?

It is a common misconception that Legacy Rockwell processors (PLC5, SLC, MicroLogix) are EtherNet/IP devices. This is somewhat true. An Ethernet enabled legacy processor like a PLC5E, SLC 5/05 or MicroLogix can accept an EtherNet/IP connection but can't respond to CIP Explicit Messages.

Instead of CIP messages those legacy processors respond to messages containing the legacy communication protocol. Those are the same messages used in Data Highway (DH+) but carried over an EtherNet/IP connection.

And, just like ControlLogix Processor, legacy Rockwell processors can not be directly used on a Modbus TCP/IP network without some sort of gateway device.

9. How Are Devices Configured?

For EtherNet/IP

There are actually two answers to this question for EtherNet/IP. One answer for non-Rockwell devices and another answer for highly integrated products from Rockwell and their close partners.

EtherNet/IP is a very complex protocol but you can break it down into this; the Scanner (usually a PLC) exchanges data buffers with each Adapter. The Scanner sends Outputs to each Adapter device. Each Adapter device sends some Inputs back to the Scanner.

In most devices, there are really only a few simple things needed to configure this data exchange. One, the size of the Input data buffer (moving from Adapter to Scanner) and two, the size of the Output data buffer (moving from the Scanner to the Adapter). If a Scanner device has the TCP/IP Address of the Adapter and those two pieces of information, it is usually ready and able to initiate a connection and start exchanging data.

Most simple EtherNet/IP devices need very little configuration. A simple 16 channel valve device, for example, might have 16 bits of fixed input data and 16 bits of fixed output data. All it needs is a TCP/IP address. Other devices, like a Yaskawa drive, have a host of configuration data. There are all sorts of things to configure. These kinds of advanced devices can have their own configuration utility. That utility will configure the sizes of the data transfers over EtherNet/IP and the TCP/IP address.

These types of non-Rockwell devices can be configured in a myriad of ways. Some use a browser interface. Some have their own configuration utility. Some are configured from a front panel HMI.



Rockwell devices and devices from close partners may have lots of configuration data built right into the Rockwell Programmable Controllers and Tools. Of course, those kinds of devices are very easy to integrate with a Rockwell Controller.

For Modbus TCP/IP

Modbus TCP/IP devices on the other hand are relatively easy to configure. These devices simply need a TCP/IP address. The available function codes, number of registers and number of coils is usually predefined by the device manufacture.

10. What is an EDS File?

An EDS (Electronic Data Sheet) is an ASCII file. It is one of the files that are required for ODVA Certification Testing. The EDS file is an electronic catalog of all the CIP Objects and Attributes in a device. It can provide not only the list of Objects and Attributes but additional information for each attribute like minimum value, maximum value, a unit's designation and more though all that extra information is optional.

Most customers assume that the EDS file is used to configure devices over the network. Unfortunately, that's not true. It's one of the aspects of EtherNet/IP that customers have the most difficulty understanding.

The truth here is that the RA EtherNet/IP tools, RsLogix and RsNetworks, don't use the EDS file for very much. For 3rd Party devices, the Programmable Controller only needs to know the TCP/IP address and the number of bytes of cyclic data to exchange. Nothing else. Those tools may use the EDS file to check the version number of an Adapter or pull out other basic information but they don't function as generic configuration tools. And since most customers just buy the RA tools, there is no incentive for anyone else to build generic configuration tools that use the EDS file.

Most device developers simply provide a simplified EDS file with generic device information that satisfies the needs of the ODVA Certification test.

11. Is There Some Sort of Data Representation File for Modbus TCP/IP?

No, there is nothing equivalent to an EDS file for a Modbus TCP/IP device. Generally, a device manufacturer simply publishes a couple of lists:

1. A list of the read and write register addresses that can be accessed
2. A list of the read and write coil addresses that can be accessed in the device and the meaning of each coil
3. A list of the Modbus function codes supported by the device



12. Is Certification Required?

For EtherNet/IP

When you sign the EtherNet/IP license agreement with the ODVA (Open Device Vendor Association) you agree to not sell a device that does not pass their certification test. That means that you agree to ship your device to an ODVA lab where they will send thousands and thousands of messages to it on a network with lots of other devices. When your device passes that test, you can use the ODVA Checkmark in your product literature to note that your device has passed certification testing.

For Modbus TCP/IP

Certification is not required for Modbus TCP/IP devices. Modbus TCP/IP devices can be self certified. The Modbus IDA publishes a list of tests that can be used to self certify a device. You can find the guide to self certification here: <http://www.modbus.org/certification.php>.

Getting It Done: Your EtherNet/IP Checklist

Know Your Data and Object Model

All CIP devices look to the network as a set of Objects composed of related data items called Attributes. Your first implementation step is to group your data into application objects that **make sense to the user**. Notice that last part. Make sense to the User. Many of us who work with our devices all day long are not paid PLC programmers. What makes sense to us may not make sense to these users.

Luckily that is where RTA comes in. We have years and years of experience presenting data to these programmers. Our services include assisting you with organizing your data to present it correctly to the network. And we have the standard template for this that will make your life easier. Click "[Info on RTA EtherNet/IP](#)" for more information.

Pick a TCP/IP Stack

This isn't always as easy as it sounds. If you are using a real-time operating system like Linux, QNX or VxWorks, you'll have the TCP/IP stack that comes with the OS. If not, your processor hardware vendor sometimes supplies one. Freescale, for example, provides OPENTCP/IP, a completely free TCP/IP stack. Failing that, you can get one on the internet, free or otherwise.

Be careful. The TCP/IP stack is the main link between your EtherNet/IP application and the network. If it's substandard you could be in for lots of trouble; either at the ODVA lab or in the field. The OPENTCP/IP stack for example, has some subtle issues that our engineers only discovered and fixed long after deploying our first product.

You can [Contact RTA](#) if you need help picking a stack or want to verify that your stack is adequate for an EtherNet/IP application.



Decide On How Your Device Will Be Configured

Even if your device is as simple as the 16 point valve, it will still need a TCP/IP address. What will your default be and how will a user set it? There are all sorts of ways including:

- Set it from the front panel if your device has one
- Use a software utility (most users will frown at this)
- Set it from the network using an EtherNet/IP Configuration Utility like RsNetworks
- Use a browser

Some people think that small processors can not have a browser. That's not true. Almost any device that can do Ethernet can have a browser. It is probably the most common way of setting an EtherNet/IP TCP/IP address.

You'll need to understand and your customers will need to understand that it's not a simple process. If your device ships with TCP/IP Address 192.168.0.100; address 100 on the 192.168.0 subnet, a computer accessing that device will also need to be on that subnet. That means that your user will have to change the address of his computer, access your EtherNet/IP device's web page, change the TCP/IP address and then restore the original TCP/IP settings on his computer. It's not rocket science but challenging for people not used to working with Ethernet.

Two ways that your user won't get a TCP/IP address: DHCP and BOOTP. Those protocols that provide addresses are just not used on the factory floor for all sorts of reasons not pertinent here.

Get An ID From the ODVA (EtherNet/IP Only)

Next you'll need to sign the license agreement with the ODVA and get a Vendor ID. The Vendor ID is Attribute 1 of Object Class 1 in EtherNet/IP. It describes to everyone on the network the manufacturer of this device.

You'll have to make the choice to either just buy the license or join the organization. The ODVA actually makes this easy for you. If you join you get a discount on the required certification test. If not, the certification test is much more expensive. It's always less expensive to join.

The ODVA is a non-profit, vendor-driven organization that owns the EtherNet/IP technology. In fact, it owns all CIP (Common Industrial Protocol) technology including EtherNet/IP, DeviceNet, ControlNet and CompoNet.

Anyone that wishes to use EtherNet/IP by embedding it in a device must get a user license for that technology from the ODVA. You do that by applying for a vendor ID, a number that forever associates you with your Adapter and Scanner devices. By signing the license agreement you agree that you will not sell devices unless they are certified by an ODVA test lab.



Schedule the Certification Test for EtherNet/IP

You will want to contact the ODVA to schedule certification of your device well before the anticipated date. The Certification lab can operate with as much as a 6 to 8 week backlog.

The certification test is an all day affair where hundreds of thousands of messages are sent to your device. It is power cycled. It is made part of a huge network. Every conceivable message sequence is delivered to your device to make sure it will interoperate successfully in the field.

You will not need to be present at the certification test. There is really nothing you can do there except wait. If a failure occurs the tester will contact the device representatives with the failing sequence of messages. This is where it can get tricky. These message sequences are simply lists of hex bytes and generally require a trained technician to decode. The ODVA Tester will provide assistance but you will incur charges of up to several hundred dollars per hour for every hour that the tester is working with you.

That is one of the reasons that many device developers use the precertification assistance offered by RTA. RTA pretests your device using the same software as the ODVA. If a failure occurs the tester works directly with the experts at RTA to decipher the reason and decide on a course of action.

Once your device passes certification you can use the ODVA Certification checkmark on all your electronic and printed promotional material for the device.

Tools You'll Need for Development

There aren't a lot of tools you'll need for development and testing. Here are the major ones:

For EtherNet/IP

1. An EtherNet/IP Scanner – You can either use a real Rockwell Programmable Controller or a PC-based software scanner. Both Real Time Automation and Pyramid Controls have scanners that can be used for testing. The disadvantage of using a Rockwell controller is that you'll have less troubleshooting information and more complexity issuing Explicit Message instructions.
2. EtherNet/IP Test Tool – This is the package that the ODVA uses to test your device. You can use it to test your device after you make any changes. The problem that you'll have with it is that if there is a failure, it simply dumps out strings of hex bytes to describe the failure. It really takes an EtherNet/IP expert like the engineers at RTA to decipher these kinds of failures.
3. EDS File Checker – this is a tool that will validate your Electronic Data Sheet.



For Modbus TCP/IP

1. Modbus TCP/IP Client—if you are developing a Modbus TCP/IP Server device you will need a PC Test client. There are clients available at very reasonable costs. Contact RTA for more information.

Beta Customer

Like any other project you'll want to identify your first customer ahead of time. There is nothing better than doing a live

Choosing an EtherNet/IP or Modbus TCP/IP Technology Solution

There are quite a few alternatives for implementing these application layer protocols. You have a number of companies that offer ASIC solutions. These solutions implement EtherNet/IP in the ASIC with communications to your processor over some serial channel. Other vendors offer daughter card solutions or module solutions. Others, like RTA, provide 100% ANSI C Software solutions.

Every application is different. Each of these solutions is appropriate for certain applications and not appropriate for others. Here is a list of the important factors to consider when choosing a technology solution for your product:

Volume – How many Ethernet capable devices do you expect to ship? The smaller the number the more the daughter card or external module solutions make the most sense. You can buy these as needed and not bear any cost when you don't need Ethernet capability.

Response Time – How critical is it that your device responds quickly to an EtherNet/IP or Modbus TCP/IP Scanner? The faster response you need the more highly integrated a solution you will need. External module solutions are probably the slowest. These devices generally convert a protocol like Modbus RS485 to EtherNet/IP. With an external module solution and a volume of data you may not be able to update a scanner with new data more than every 100 msec at the very best.

Data Presentation – How do you wish to present your device to the network? Do you want to have highly configured objects that describe your data? For example, if you have a flow meter, a highly integrated software solution would provide a Flow Object with a Flow attribute in Meters per second and another in Feet per second. A non-highly integrated solution like a gateway or daughter card would simply provide a series of analog values.

Expense Allocation – You have two choices on paying for this technology. One, you can take the one-time hit of a software license fee. Costs are not outrageous but it will be a hit on the monthly budget. Or you can choose to take the BOM hit by adding hardware. Hardware solutions can cost you anywhere from \$10 to \$300 a device. What makes sense for your device is very dependent on how your company finances projects and what kind of expected volume you have.

*Note that one doesn't preclude the other. You **MAY** (it's not guaranteed) find that you can deploy the hardware solution really quickly and then if you see enough volume take that part off and switch to a software solution.



Multiple Sources – If you do choose the hardware solution be conscious that all those solutions are pretty much sole source. If that company gets bought, goes out of business or obsoletes your part, you're going to have a problem.

Support for Recommended Functionality (EtherNet/IP Only) – The ODVA has created a list of recommended features that they suggest that vendors support. Items like DHCP or BOOTP support are on the recommended functionality list. You should discuss with your EtherNet/IP technology provider what is on the recommended functionality list and if their technology supports the EtherNet/IP Recommended Functionality.

Access to Software Source – How important is access to source code to you? Many companies want source. These companies have found that when they do have a problem with some piece of technology it's a huge benefit to be able to investigate the source code themselves.

Certification Assistance (EtherNet/IP Only) – You will want to make sure that your technology provider is going to support you during the required ODVA Certification test. If you are going to sell an EtherNet/IP device you must complete certification testing.

You should know that if you are going to just use EtherNet/IP in a closed system of your own, you do not need to complete the certification test.

Browser Support – You will want to make certain that your technology solution does not preclude having a web server in your product. A web server can provide you with all sorts of additional functionality that can be a winner with your end customer.

Vendor Capabilities – When evaluating vendors for an EtherNet/IP technology solution be certain that you get Object Model Development Support including templates, EDS (Electronic Data Sheet) development assistance and ongoing support for serious customer problems.

What is important here is to have assistance available if you have a significant customer problem. These problems can be complicated involving switches, routers, Programmable Controllers and Internet infrastructure. You will want to know that a talented, experienced professional is available to you



Special Requirements for the EtherNet/IP Physical Layer Implementation

If you are going to go about [developing an EtherNet/IP device](#), what are the physical requirements that you must meet? Is it enough to just use off-the-shelf components? Can you just add some software and an RJ45 jack to your device and “TA DA”, you’re EtherNet/IP capable?

If your device simply exists in a clean, lab environment, or enclosed cabinet with no requirements for high speed operation (>100mb) a Commercial implementation will be adequate. In this type of implementation, you can use off-the-shelf components, support 10mbps and 100mbps and use a standard, non-sealed RJ45 or Fiber connector. Requirements for these kinds of devices can be found in the ANSI 802.3 and TIA 568 standard.

Devices meeting these commercial standards are often found in industrial environments. They usually perform well if the physical environment is not subject to temperature, vibration, shock, noise and other environmental extremes. The ODVA Physical Layer specification recognizes devices meeting commercial implementation standards as acceptable for use within the guidelines of the EtherNet/IP specification. However, products meeting these commercial standards are not eligible for the industrial conformance checkmark. These products are eligible for the commercial conformance checkmark.

The ODVA specification goes to great pains to warn EtherNet/IP developers away from Commercial implementations. They prefer that developers follow the industrial standards and protect devices to the maximum against shock, vibration and other environmental extremes. They go to great pains in the specification to warn that commercial components and a commercial implementation can “degrade system performance” however the vast majority of EtherNet/IP device developers have followed the commercial implementation without problems.

Product developers, with requirements that exceed the 802.3 or 568 standards, should follow the Industrial implementation standard. Volume 2, Chapter 8 of the EtherNet/IP Physical Layer standard defines this standard. The standard defines the vibration, shock, humidity, temperature and EMI requirements for an Industrial Implementation. These standards are all based on published standards. Vibration is based on IEC 60068-2-6. EMI is based on the IEC 61000-6 and IEC 61000-4. Temperature and Humidity standards are based on the IEC 60068-2 standard.

The ODVA Physical Layer specification also details the kinds of cabling that meet the Commercial and Industrial standards. The types of shielded and unshielded cables, the typical impedance, the wire color codes and the acceptable losses are all detailed by this standard. Sealed and unsealed RJ45 housings are both acceptable under the Industrial standard. Coding for the sealed housings are specified in the standard.

For more information on Physical Layer specifications you should contact the ODVA at its Ann Arbor, Michigan office. They can provide the most complete details on how to meet the Industrial standard.



Why Use The RTA EtherNet/IP Scanner and Adapter Royalty Free Source Stacks?

The Real Time Automation No Royalty EtherNet/IP Scanner and Adapter Source Code is **the best Technical Solution** and **the best Commercial Solution**. Our source code products are focused on making your development breathtakingly easy while quickly getting you into the fast growing EtherNet/IP market - often EtherNet/IP-enabling your product in as little as a few hours. We regularly work with companies all around the world that can't afford an 12 to 18 month EtherNet/IP development cycle.

Just like other industrial developers that have used our EtherNet/IP software over the last ten years, you will find that the RTA EtherNet/IP Software Development Kits **reduce your Engineering resources and speed your time to market**. Here is why:

- **NO RTOS Requirements** – All Real Time Operating Systems are supported. All we need is a 10msec Ticker.
- **NO RTOS Needed** – The RTA Source Stacks can run without an RTOS. Just put our code inline with your application code.
- **NO TCP/IP Stack Requirements** – ANY BSD capable TCP/IP stack can be supported. The entire stack interface is located in a single C source file. It can easily support a home grown TCP/IP Stack.
- **Low Complexity API** – There are a few configuration calls, a single Explicit Messaging interface handler, and two pointers. That is the ENTIRE API! You only have to write the EM Handler which maps Object/ Instance/Attribute data of EtherNet/IP to your internal memory structure. EtherNet/IP to your internal memory structure.
- **No Platform Requirements** – The ANSI C Source code will compile into any compiler for any hardware platform. Over the last ten years it's been deployed on 8, 16 and 32 bit platforms, Big Endean systems, Little Endean Systems, High End DSPs, and Low Speed 8-bit processors.
- **Small Footprint** – You won't need more than 20K of Flash to support the Adapter and a little more to support the Scanner. The Adapter RAM requires about 10K while the Scanner requires 8K with 2-3K per Adapter connection.
- **Single Task** – There's just one task to implement and you can implement that inline with your code if you'd like.
- **100% Developed In House** – The RTA Source code is the product of thousands of hours of development by RTA Engineers, all of it done in house at RTA, and 18 months of field trials in factories all over the world.
- **No Processor Requirements** – The RTA EtherNet/IP Source code solution that has been implemented over 45 times using fast processors running big, heavy duty operating systems like VxWorks or CE to small low end embedded processors with no OS like Rabbit from Zworld and low end Microchip processors. Our stack has no specific processor requirements.



If you have questions or to get more information please visit [Contact John Rinaldi](#) to contact me directly to discuss your application and how you can get Modbus TCP or EtherNet/IP running in 3 days or less.

GOOD LUCK!

John S Rinaldi

150 S. Sunny Slope Road

Brookfield, WI 53005

262-439-4999

www.rtaautomation.com

Real Time Automation – RTA is a 21 year old provider of industrial networking technology. RTA source code, daughter cards, modules and gateways support everything from ASCII to EtherNet/IP, BACnet and LON. Visit www.rtaautomation.com for more information.

Copyright Notice © 2012 Real Time Automation, Inc. All rights reserved. Printed in USA.

This document is copyrighted by Real Time Automation Inc.. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited.

Trademark Notices Allen-Bradley, ControlLogix, FactoryTalk, PLC-5, Rockwell Automation, Rockwell Software, RSLinx, RSView, the Rockwell Software logo, and VersaView are registered trademarks of Rockwell Automation, Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other

countries **QNX** and **Neutrino** are registered trademarks of QNX Software Systems Ltd. **VxWorks** is a registered trademark of Wind River Systems, Inc.

Wind River Systems and the **Wind River Systems** logo are trademarks of Wind River Systems, Inc.

ControlNet is a registered trademark of ControlNet International.

DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA). **EtherNet/IP** is a trademark of the ODVA. **Modbus TCP** is a trademark of Modbus IDA

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.

Trademarks Microsoft, Windows, Windows ME, Windows NT, Windows 2000, Windows Server 2003, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

All other trademarks are the property of their respective holders and are hereby acknowledged.