

A GUIDE TO

# Connecting Motion Control in the Factory

Implementing Communication Protocols

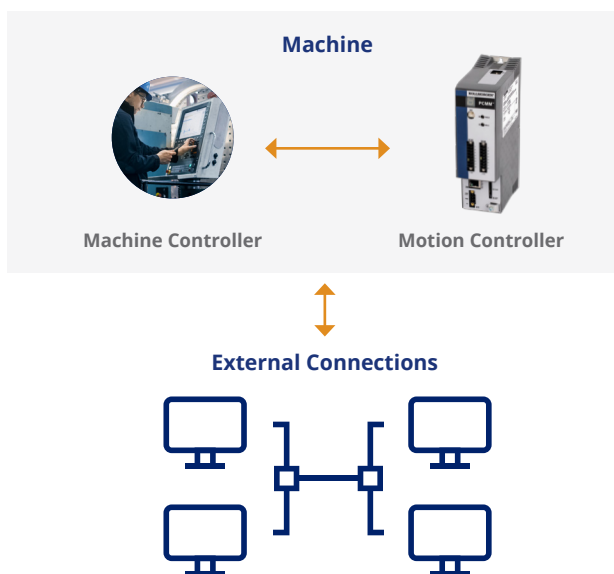
**KOLLMORGEN**

The need for Ethernet-based communication within the factory has grown over the last 10 years. Detailed information about the tasks and performance of machines at every stage is critical to productivity. As capabilities have expanded from performance reporting to predictive maintenance warnings and beyond, the need to connect multiple controllers within the same machine and elsewhere in the factory has also grown.

This paper reviews options that are available for Ethernet and non-Ethernet connections within the motion controller. We will begin with a summary of what information can be transferred in and out of the motion controller followed by network and non-network examples for communicating with controllers. Finally, we will present the factors that need to be considered when choosing the right solutions for your factory.

## MACHINE CONTROL FORMATS

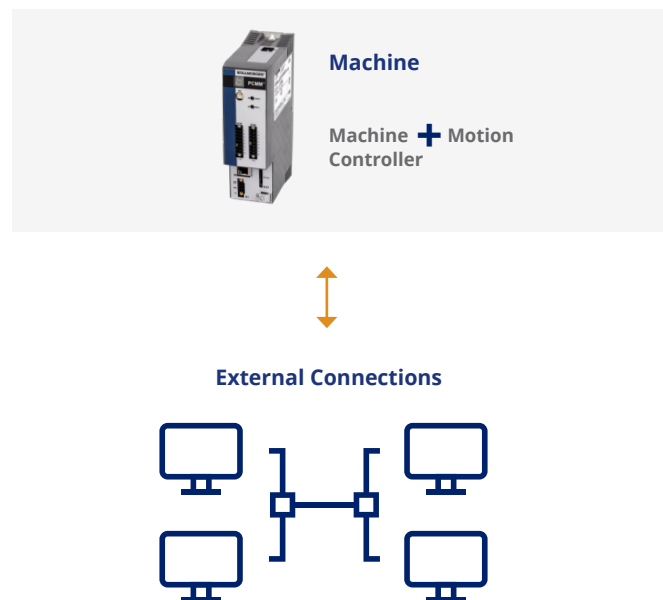
There are two formats when incorporating high-performance servo motion control in a machine.



In the example above, the motion and machine controllers are separate. The motion controller focuses on motion, and the machine control—a resident programmable logic controller (PLC) or personal computer (PC)—handles the rest of the machine control. The advantage of having a separate motion controller is that it will often have more functionality and processing power dedicated to motion performance. More motion types such as camming and more operational modes such as torque speed, gearing, and more time-critical input/output (I/O) are a benefit of standalone motion controllers. In addition, with motion

control as the central focus, higher motion performance and precision is often possible when motion and machine controllers are separate.

The second type of machine control combines both the machine and motion control into one controller as shown in the example below.



Thanks to increasingly powerful processors, there are now motion controllers that include PLC functionality. This can eliminate the need for the PLC or a PC-based machine controller, potentially lowering total machine control cost.





## INFORMATION TRANSFERRED

Motion controllers are used in a variety of industries including medical, laboratory automation, robotics, printing, labeling, material forming, pharmaceutical, packaging, food and beverage, tire and rubber, and postal sorting.

The primary data out of the motion controller is performance-related. The data can include how efficiently the machine is making product, how much product has been made, warnings about machine errors or limitations, and unexpected variations such as a heater in the machine operating at the incorrect temperature.

While knowing that a machine is operating properly and at peak efficiency provides peace of mind, knowing when it is not running properly can save money, time and resources.

Suppose you have a motor drawing more current than it should, indicating a mechanical problem, or a cut-to-length machine making improper cuts. Having a system programmed to provide real-time machine performance data is critical to identifying and fixing these issues quickly.

Information moving from the machine controller to the motion controller is command-focused. These commands could be instructions or recipes for machine setup—for instance, part numbers—or the parameters for making a specific widget. They could also include motion specs such as distance, speed, camming motion points and other operational details like periodic reminders for machine operators to perform routine maintenance.

## Three Types of Information Transfer Mediums:

**Ethernet - PLC Oriented:** These are the common industrial Ethernet fieldbuses incorporated into traditional PLCs such as Ethernet/IP, Profinet, EtherCAT®, SERCOSIII®, vendor-specific and others.

**Ethernet-PC Oriented:** These are traditional PC-based networks such as TCP, UDP and HTTP

**Non-Ethernet Methods:** Web servers, SD cards, remote memory, FTP and VPN/remote web access.

## ETHERNET COMMUNICATIONS BASICS

- » Transportation medium is copper wire, CAT5 or CAT6 cable with RJ45 termination. Cables can be up to 100 meters in length between each node.
- » Connection topology can be a line, a star or a ring, although many network implementations only support a subset of the three.
- » Built-in electrical isolation is needed in many motion or machine applications to help eliminate part variances and maintain precision motion control.
- » Transfer of information can be deterministic or non-deterministic, depending on the network and how it is configured.
- » Update times can be 500 milliseconds or higher, and down to 250 microseconds depending on the applications and the networks.
- » Automatic network configuration and integrity checks are available to ensure the quality of the network transmission.
- » Information passed between the motion controller and the machine controller or external controller is often called parameters, variables, or tags, in the form of single objects or in the form of data array or structure.

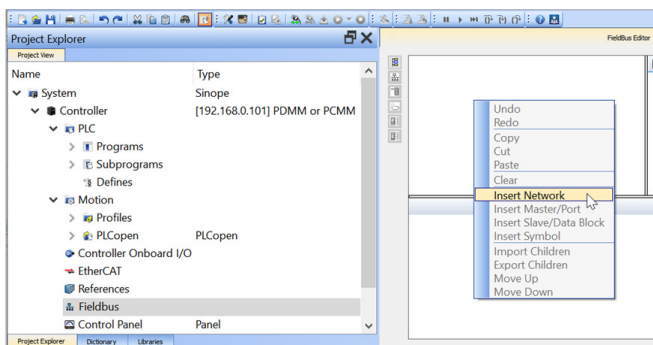
## MODBUS TCP/IP

The Ethernet-based Modbus® TCP/IP is built on the Modbus RTU protocol. It is an industrial standard used with and supported by a variety of devices and control products. Modbus TCP/IP has a standard address block scheme for transferring binary and non-binary data in 32-bit or 16-bit formats. It is non-deterministic, and the data update time can vary. In general, the performance range for update times between two devices is 20 to 200 milliseconds.

To integrate a Modbus TCP/IP interface into a motion controller, there will typically be a configuration setup in the motion controller's programming software. In the example to the lower left, there is a selection for inserting a network into a Fieldbus editor.

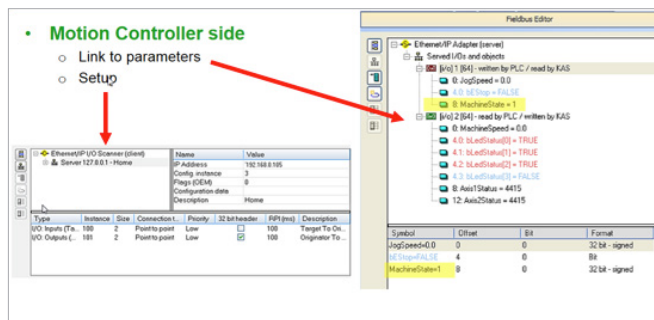
Linking Modbus addresses to preset tags or variables is straightforward as there is no text programming required. Links are established either by the drag-and-drop method or by adding them via a dropdown dialog box.

Setting up the Modbus connection on the machine controller side requires the configuration of an IP address. Then, by importing a tag file, motion parameters are available for use in the machine controller's tag or variable dictionary and application program.



## ETHERNET/IP

Ethernet/IP™ is widely used because of its flexibility, scalability and ease of integration. Possible configurations include Polled I/O, Flex I/O and Explicit Messaging. It can be built into motion controllers and can provide direct access to the motion controller's machine and process parameters. Transmission update rates can go as low as 10 milliseconds with the Kollmorgen PDMM or PCMM controller, although many applications work well with update rates in the 30 to 100 milliseconds range.

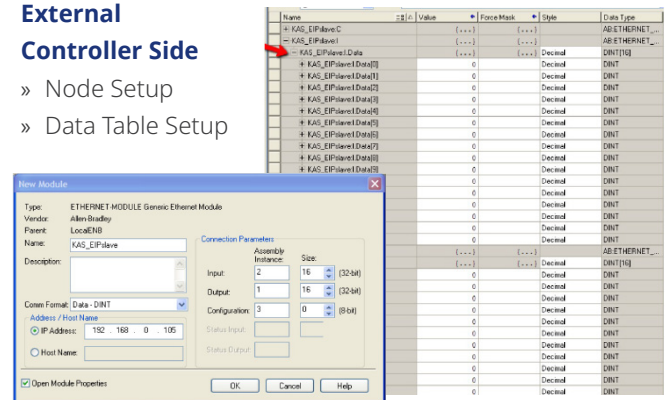


Implementation on the motion control side is accomplished through a setup screen on which a user sets parameters pertinent to Ethernet/IP, and easily ties in the application program.

Built-in screens on the machine controller's side allow for node and data table setup.

### External Controller Side

- » Node Setup
- » Data Table Setup



## UDP (USER DATAGRAM PROTOCOL)

UDP is gaining acceptance within the industrial automation world due to the simplicity and speed it delivers to machine control applications. UDP is often used in applications developed with Visual Basic (VB), Visual Studio (VS) C++, C# and others. These languages have typically been used outside the machine automation world, but the low communication overhead makes the protocol appealing to automation applications. It can provide very fast update rates as low as 1–4 milliseconds.

Function	Description
<a href="#">udpAddrMake</a>	Build an address buffer for UDP functions
<a href="#">udpClose</a>	Close a socket
<a href="#">udpCreate</a>	Create a UDP socket
<a href="#">udpIsValid</a>	Test if a socket is valid
<a href="#">udpRcvFrom</a>	Receive a telegram
<a href="#">udpRcvFromArray</a>	Receive a byte array through UDP
<a href="#">udpRcvFromVar</a>	Receives the contents of a variable through UDP
<a href="#">udpSendTo</a>	Send a telegram
<a href="#">udpSendToArray</a>	Send a byte array through UDP
<a href="#">udpSendToVar</a>	Sends the contents of a local variable through UDP

Unlike in an Ethernet/IP setup, a UDP connection in a Kollmorgen PDMM motion controller is implemented without a fieldbus screen. Communication is established via PLC function blocks directly in the application program. Incoming information is read and parsed into motion control parameters.

### Initialize Communications

```

3: //Create Socket
SocketNumber := udpCreate(20);
TestState := TestState + 1;
if 0 = SocketNumber then
    bTestFailed := true;
    TestState := 100;
    printmessage(LEVEL_ERROR, 'Create Socket Failed');
end if;

1: //Validate Socket
if true = udpIsValid(SocketNumber) then
    TestState := TestState + 1;
else
    bTestFailed := true;
    TestState := 100;
    printmessage(LEVEL_ERROR, 'Socket is invalid');
end if;

2: //Indicate that ready to receive
if true = udpAddrMake('10.156.238.207'(*STRING*), 20(*DINT*), A,
bSendStatus := udpSendTo(SocketNumber, 20, Address, 'I am ready t
if true = bSendStatus then
    TestState := TestState + 1;
else
    bTestFailed := true;
    TestState := 100;
    printmessage(LEVEL_ERROR, 'Failed to send acknowledgement');
end if;
  
```

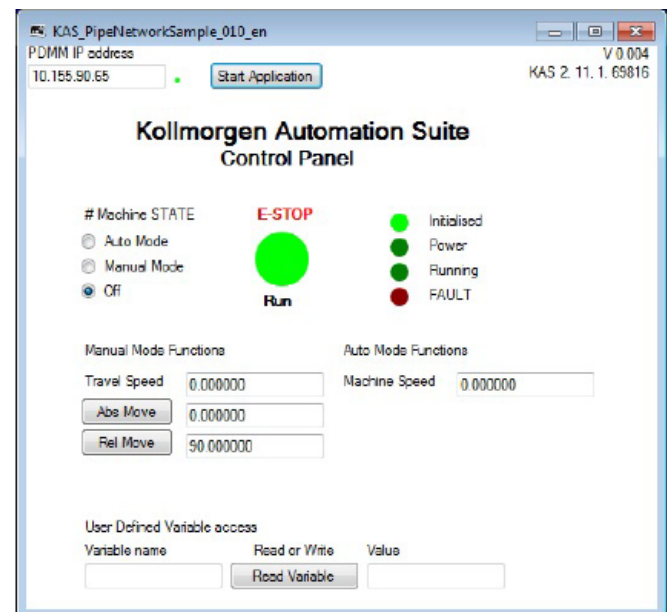
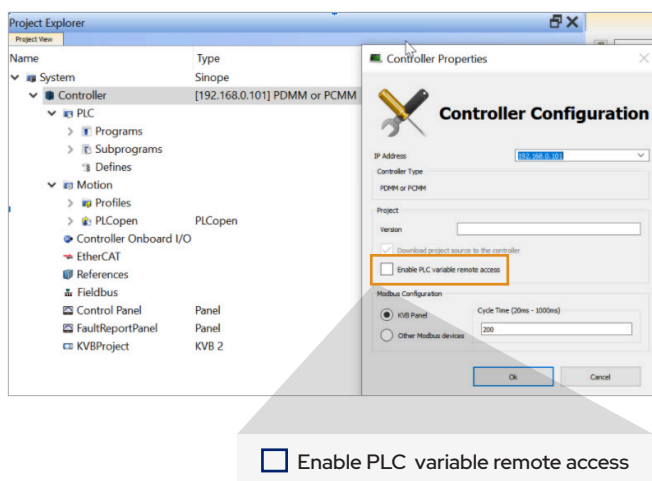


## HTTP (HYPERTEXT TRANSFER PROTOCOL)

HTTP has been called the protocol that makes the web work. Although not specifically developed for machine automation, HTTP can also be used in the factory automation world. Languages used in PC-based controllers such as Visual Basic (VB), Visual Studio (VS), Excel, C#, C++ and Java offer HTTP communications support. HTTP is non-deterministic, with communication rates of 50 to 300 milliseconds. This works well in applications that don't need to pass time-critical information at the servo update rate, such as when communicating machine setup information.

In a motion controller, implementing the HTTP interface is done through the controller setup screen, making all of the variables in the dictionary available through the HTTP network. HTTP setup and communication of external controllers is simple, with transmissions only requiring the IP address of the motion controller and the parameter name.

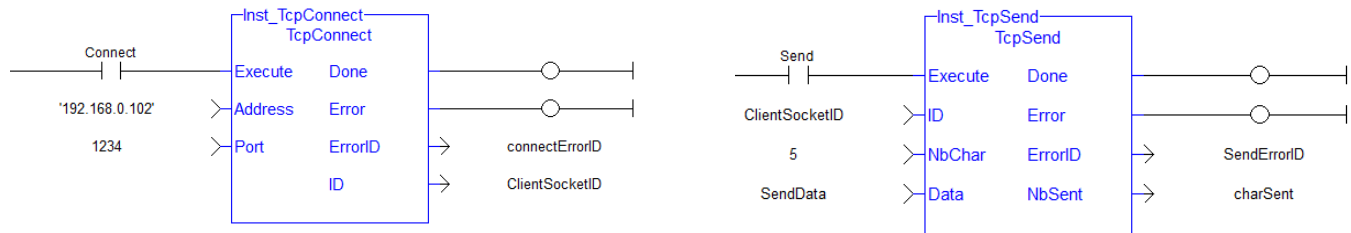
The following is an example of HTTP communication from a VB2008 platform to the motion controller. It is a control screen/panel with controls to both read and write machine parameters. The data from the motion controller is read using a cyclic read command. Data transferred or written to the motion controller is event-driven when a user clicks on a button or enters a motion parameter value.





## TCP TRANSMISSION CONTROL PROTOCOL

TCP/IP is one of the most popular networks in the world, commonly used in the office environment but also usable in the factory. In the motion controller, TCP/IP communications are implemented using dedicated function blocks.



## ETHERCAT

The speed and accuracy of EtherCAT makes it ideal for connecting devices such as remote I/O or drives to the motion controller. A common format in industrial automation applications is CAN Over EtherCAT (CoE), which enables data to be sent deterministically with update times as fast as one-fourth millisecond, or 250 microseconds. Data such as the type of part being made doesn't need to be transferred during every cycle, so a non-deterministic side of the protocol, called SDO or Mailbox, handles those data transfers.

On the device side, a predefined ESI (EtherCAT Slave Information) file from the device vendor defines a set of parameters that can be transferred. Some parameters, called process data objects (PDO), are updated cyclically. Other parameters that are transferred at a slower rate in the background (SDO or Mailbox channel) can also be defined. Additionally, the motion controller, containing the EtherCAT master, can set device parameters when the network is initialized to configure the device for how it will be used in the application, as well as to scan for devices on the network.

**Device\_2 (EL2521) PDO Selection/Mapping**

Output (Rx) PDOs

Select Output (Rx) PDOs

Index	Subindex	Object Name	Size [bit]	PLC Variable
0x7010	1	Control_Frequency select	1	(Global)/ControlFreqSelector
0x7010	2	Control_Disable ramp	1	(Global)/ControlDisableRamp
0x7010	3	Control_Go counter	1	(Global)/ControlGoCounter
0x7010	17	Frequency value	16	(Global)/FreqValue

**0x1602 ENC Control compact**

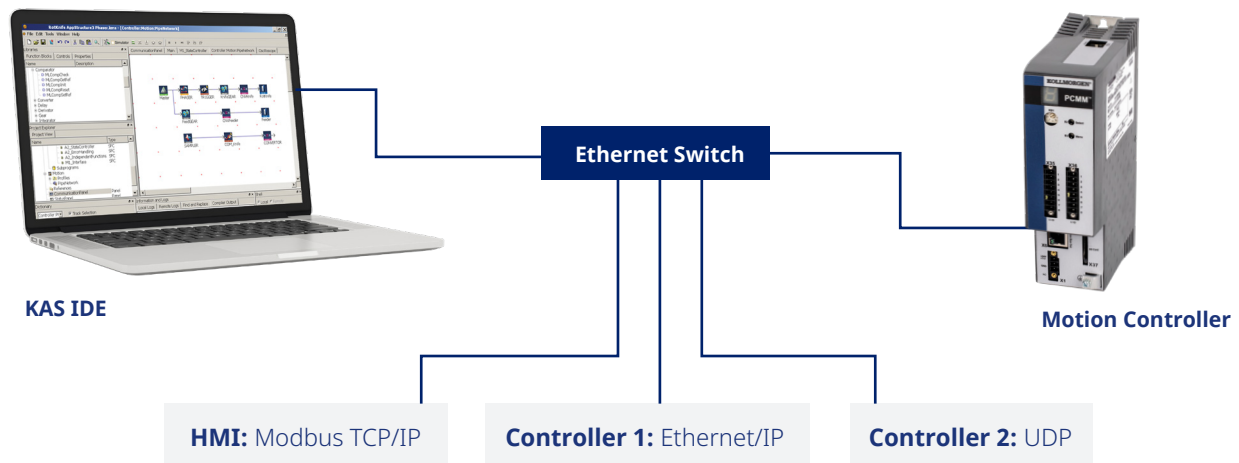
Index	Subindex	Object Name	Size [bit]	PLC Variable
0x7020	3	Control_Set counter	1	(Global)/ControlSetCounter
0x7020	17	Set counter value	16	(Global)/SetCounterValue

In the motion controller interface, a window shows which parameters are PDO parameters and can be sent cyclically. In the setup, the user can link the cyclical parameters to PLC variables in an application. Parameters coming through the EtherCAT network can also be linked to PLC programming variables through specific controller function blocks. For example, if you need to read the position of a servo axis, a standard PLC open function block can be used, such as MC\_ReadActPosition. On the motion controller side, a wide range of variables can be passed non-cyclically.

## MULTIPLE CONNECTIONS

Some applications require multiple Ethernet-based network interfaces to connect to the motion controller. One way to accomplish this is through an external Ethernet switch connected to a single RJ45 port on the motion controller. In the example below, there are three connections: Modbus TCP/IP, Ethernet/IP and UDP. Modbus TCP/IP goes to the HMI. External controller #1 is connected via Ethernet IP, and a third connection uses UDP to connect to external

controller #2 for SCADA implementation. One practical question to ask is how using three networks will affect performance. Users should carefully plan the networks in use, optimizing the update rate for each and the information transferred, as well as the update rates of the programs in the motion controller, to minimize any negative effects on performance.



New technologies such as 5G fast connections between factory devices are becoming commonplace. A motion controller that supports multiple networks helps users fit it in the Internet of Things (IoT) with their choice of network, allowing the motion controller information to travel around the factory and around the world.

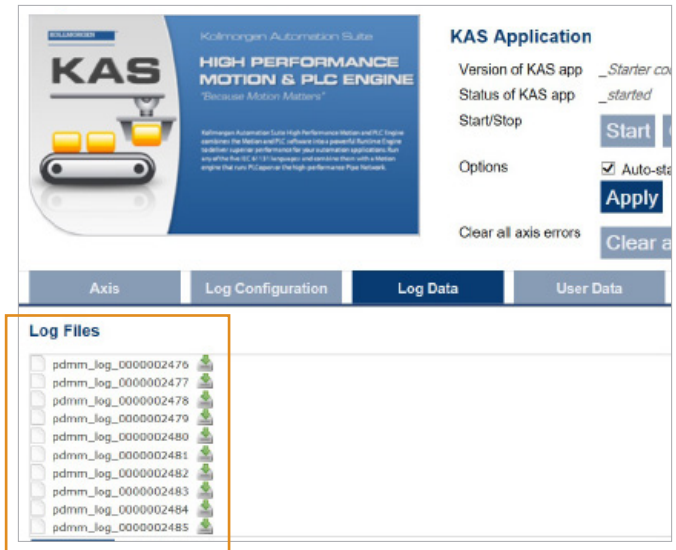


## WEB SERVER

A web server can also be used to process information in and out of motion controllers simply by entering the IP address. Once connected, machine operation information is available as well as a certain amount of remote control.

The screenshot to the right is from a web server on a PDMM series controller showing log files that can be opened to provide information on the state of the controller and how it's running. These log files can help a controls engineer troubleshoot issues with the controller.

Operational data generated by the motion controller during machine operation can also be exported through the web server.



## MEMORY CARD

Another common way to access information is with a memory card. This method is not network-based, but a memory card allows users to import data into the motion controller and move operational data from the controller to a file. In addition, an SD (Secure Digital) card can be used to transfer controller, system and drive configuration information from one motion controller to another. This provides a time-efficient setup method for duplication of the controller's firmware, application software and parameters from one machine to the next.



## REMOTE MEMORY

Connecting to an external hard drive through an Ethernet connection is another option to pass information. To make the connection, the motion controller's web server is used to set up the IP address and other information. An external hard drive can be accessed in the same building or from elsewhere, making it easier for the user to put information on a remote central memory location that stores factory-wide operation information. Also in the motion controller's program, the remote memory files with operational information can be read when running the machine.

## WHICH NETWORK TO CHOOSE?

With all the options available, which one makes the most sense for your application? Considerations include:

- ✔ **What network protocols** are on the present controllers you are using?
- ✔ **What is your personal experience with a particular network?** For example, experience with Ethernet/IP or HTTP will help minimize the time to get the network up and working.
- ✔ **What is your supplier's experience with a specific network?** Do they have the application experience that is often critical for a timely integration?
- ✔ **Can the network update at the rates the machine requires?** Look at your application specifics to determine what update times you need to pass the information. In many applications, there will be two levels—for example, information that needs to get there in 1 to 5 milliseconds and information that can get there in 50 to 200 milliseconds.
- ✔ **What is the network's effect on other areas of motion controller performance?** Will the network loading compromise other areas of machine performance?
- ✔ **Tools:** For a particular network, what tools and documentation does the product/vendor have to help establish communications and monitor information that is passed?
- ✔ **What technical support** (human interaction) does the supplier have in case you need it?
- ✔ **What third-party tools are available?** With Modbus or HTTP or UDP, for example, there are free tools available online for setting up an interface to communicate with the motion controller.
- ✔ **What security and safety measures** do they have in place?

## COMMISSIONING:

Steps to getting the network to work and achieve the desired performance goals.

- ✔ **Define the need.** What information needs to be passed through the network: Motion, process, IO, status, etc.?
- ✔ **What is the required update rate** of each parameter?
- ✔ **Use an industrial grade Ethernet cable.** The extra cost is well worth it to prevent any noise issues and machine downtime.
- ✔ **Set up communication parameters** such as update rate, data size and IP address on the controller to establish the connection.
- ✔ **Start small.** Get basic communication working first. Pick one or a few parameters to send and receive successfully. It is faster to make adjustments to a few parameters during the development process.
- ✔ **Add the rest of the parameters.**
- ✔ **Verify machine performance on both sides of the network.** Is all the information getting there and at the needed update rate? Are there any effects on motion or other aspects of the control?

## CONCLUDING THOUGHTS

Transferring information in and out of motion controllers is a function that is critical to the daily operation and overall productivity of today's factories. This paper has presented many of the setup options that are available and many of the factors that need to be considered when determining which one works best for you.

Ethernet-based communication within factories has been on the rise over the last 10 years, and as more capabilities are developed to measure and track machine performance it is sure to keep growing. The diversity of applications and information available can provide a competitive edge for those factories making use of the technology. Building the expertise of your staff to include the most current technologies will put you in position to capitalize on new advancements.



### About the Author

Carroll Wontrop is a senior systems engineer at Kollmorgen located in Radford, Virginia. He's a 1981 Engineering Graduate of Virginia Tech who has been engaged in the motion control industry since 1983. He can be reached at [carroll.wontrop@kollmorgen.com](mailto:carroll.wontrop@kollmorgen.com)

## For Answers, Partner With Kollmorgen

Kollmorgen is more than a supplier. We're a partner, dedicated to your success. We give you direct engineer-to-engineer access to the designers who create our motion systems and who understand how to address specialized machine requirements. Our self-guided design tools help you model, choose and optimize products online. And with our global footprint of manufacturing, design, application and service centers, you always have access to dependable supply, co-engineering expertise, and personalized support that no other partner can provide. Whether you're upgrading an existing machine or designing the next-generation machine that will define the state of the art for your customers, we can help you engineer the exceptional.

Ready to discover all your machine is capable of? Visit [www.kollmorgen.com](http://www.kollmorgen.com)