PDHonline Course E497 (3 PDH)

Industrial Communications and Control Protocols

By Michael J. Hamill, P.E.

PDH Online | PDH Center

5272 Meadows Estate Drive
Fairfax, VA. 22030-6658
Phone: 703-988-0088
www.PDHonline.com
www.PDHcenter.org

An Approved Continuing Education Provider

Index

Introduction

This course is intended to benefit readers by:

- Explaining what protocols are
- Explaining what communications and control protocols do
- Explaining how use of protocols makes it easier for controllers to operate
- Describing some commonly-used network topologies
- Describing some hardware and cables used for digital communications
- Detailing some different ways of transmitting data over networks
- Condensing often confusing information about protocols
- Explaining how protocols interrelate with networks and the Internet
- Describing some commonly used protocols and their uses

Readers who have experience with, or some knowledge about one or more of the following technologies will benefit most from this course: Programmable Logic Controllers (**PLCs**); Distributed Control Systems (**DCSs**); or Programmable Automatic Controllers (**PACs**).

This course also discusses networks in some detail. Protocols are best explained along with a discussion about some basic features of networks.

Protocol: A definition

It's worthwhile to begin by defining **Protocol**. The author's definition is: *a method for digital data communications between two or more devices in different locations, or on a network*.

There are many protocols in use around the world. This course focuses on protocols used by PLCs, DCSs, PACs, and devices existing on the same network as industrial controllers. Some protocols used with controllers are for data communications only, usually by sensors and transmitters. Others are used for both data communications and control applications.

Digital Data Basics

Digital data is a natural choice for communications. The smallest unit of digital data is a **Bit**, or binary digit, and it has just two states: Off, represented by a 0; and On, represented by a 1. Since the 1960s, most computerized devices have relied on miniaturized, two-state transistors that are either off or on. And a voltage associated with the transistor is either low (0) or high (1). A bit's value is represented using base 2, and can only be 0 or 1. Digital data can be generated and transmitted very quickly by electronic equipment.

Almost all transmitted data is at least one byte long. A **byte** consists of 8 consecutive bits, or binary digits. A byte can have up to 256 (2^8) values. In reality, data is frequently communicated in 2, 4 or 8 byte units. A data unit with 2 bytes (16 bits) is often called a **Word**.

Measurements that mean something in the real world - levels, pressures, temperatures, etc. - can easily be represented with 2 or 4 bytes. So can the ranges of set points. Likewise the On or Off state of a device like a motor. Within computers and microprocessors, arithmetic operations on data that has a digital equivalent is readily done. They can easily manipulate data in binary, or base 2.

Example: Assume a tank's level transmitter is calibrated so 0% is 0 feet, and 100% signal is 10 feet. If a 2-byte word is used to represent this data, than at 5 feet (50% level) that will be the same as 32,767, or 0111111111111111 in binary (and 7FFF in hexadecimal, a.k.a., hex - base 16). At 100% level, that will equate to 65,535 (hex: FFFF).

Before microprocessor-based controllers became the dominant control technology in the early 1980s, controllers for applications like refineries, chemical plants, and power plants mostly used analog electronic control systems. Analog signals vary in a range. These analog systems had some serious limitations. One was that analog signals were susceptible to being corrupted by electrical noise and unintentional grounds. Another was that settings and calculated values in control loops tended to drift over time, especially as components heated up.

Digital data communications through protocols has the advantages that *it is inherently more stable, reliable, and less susceptible to electrical noise than analog signals*.

Another advantage of digital communications is that a lot of data can be communicated on a single network or fieldbus cable. This reduces end users' needs for installing controller input and output modules, wiring, conduit, etc. It also lets users connect different types of devices to the same communications cable, such as transmitters and actuators. And it also makes it easier find the sources of problem conditions readily.

Fieldbus is defined as "*a family of industrial computer network protocols used for real-time distributed control, standardized as IEC 61158*."[1]

This course is progressing to a discussion of protocols. But worldwide, plants are continuing to use sensors, transmitters and actuators that transmit or respond to analog signals. Controllers handle that by embedding Analog-to-Digital (A/D) and Digital-to-Analog (D/A) converters in input and output modules. For example, most incoming analog signals are converted from a voltage[1] to a 16-bit integer within a controller. That makes it easy for a controller to use the data.

Differences in Controller Types

The two primary types of controllers in use are **PLCs** and **DCSs** (see page 3). The two technologies were developed for different markets in the 1970s. PLCs were meant for mostly for manufacturing applications where most devices were turned on and off; raised; lowered; rotated; moved sideways; and where users wanted to compare the state of something at the end of an operation to its beginning state. They also found many uses in applications where a process occurred in a repeated cycle. PLC functions were (and still are) mostly implemented using ladder logic symbols, which technicians understand. DCSs, on the other hand, replaced earlier systems and equipment in applications that involved mostly continuous measurement of process variables and ongoing control of processes – such as at oil refineries and power plants. DCSs replaced pneumatic controls; analog single loop controllers; and analog control systems. DCSs predominantly use function block programming. DCSs were (and remain) more expensive than PLCs because the reliability requirements are more demanding, in most cases, than in PLC applications. Also, in DCS systems, there are usually 2 or more layers of control, in which a master controller sends set points to sub-controllers. Sub-controllers sometimes send data back to a master controller. DCSs often interface with thousands of data points. DCS requirements are more difficult to meet. Downtime of some DCS components would have a major impact on lost production.

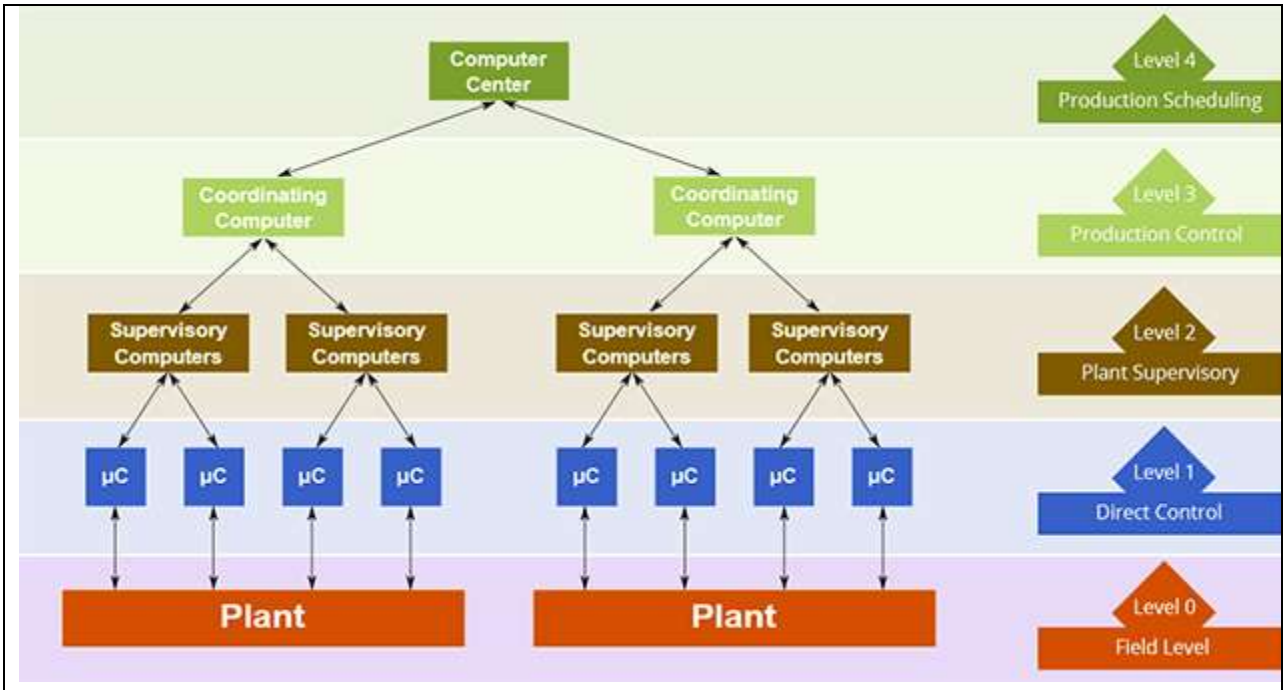
Over time, the two technologies became more alike. That has led some manufacturers to call their industrial controllers Programmable Automation Controllers, or **PACs**. There are many applications which can be met by either a PLC or DCS, but usually a PLC will be less costly.

Diversity in the PLC/ DCS/ PAC market and its problems

Since the 1970s, many Original Equipment Manufacturers (OEMs) have offered similar systems for control equipment end users. As time passed, technology improved. Processor speeds increased, sizes of components used by controller OEMS - processors, memory modules, etc. - decreased, and new products could be offered

[1] Typically the voltage across a 250 ohm resistor built into each analog input module channel.

which were smaller and more powerful. In some cases OEMs had difficulty getting some components for previous generation products from their suppliers. Competitive pressures forced controller OEMs to focus on new products to match competitors' offerings. But most companies in the PLC/DCS/PAC market were going their own way - developing their own unique equipment, and especially, using their own unique programming software and equipment.



A representation of a **DCS (Distributed Control System)**

By the late 1980s it was evident there was too much diversity in this market. Devices from one OEM could be used with the same OEM's equipment, but not with comparable equipment from other companies. The need to standardize similar controllers and field devices as much as possible was apparent. Some common digital communication protocols and fieldbuses were needed. This was what customers would ultimately want. The BMBF (Department of Education and Research) agency of the German government recognized this need. It responded by spearheading the development of what became the PROFIBUS (Process Field Bus) group of standards. Around the same time, the Instrumentation Society of America (ISA)[2] formed a task group to develop common standards for use by transmitters, actuators, and control systems. This eventually led to the formation of the Fieldbus Foundation, which developed specifications for “open” protocols.

[2] ISA is now the Instrumentation, Systems and Automation society.

Better data communications methods were also developed. Ethernet, first developed in 1973, is a good example. As networks grew in size and the Internet spread in the 1990s, the advantages of using Ethernet and its native protocols, including TCP/IP, with industrial controllers became apparent. So Ethernet and TCP/IP variations are now widely used by industrial controller OEMs.

Networks, Nodes, and Topologies

This course diverges for a time to an explanation of networks, hardware, and cables.

A **Network** is an interconnected group of computers and/or controllers, and devices that interact with computers and controllers. A **Node** is a computer or other device in a network. Networks are interconnected by different types of conversion devices, cables, and sometimes, by radio transceivers.

3 common **topologies**, or arrangements for networks, are discussed below:

- **Bus**
- **Star**
- **Ring**

Figures 1A, 1B and 1C are diagrams which show each topology.

The **Bus** topology is the simplest. Figure 1A shows a simple bus network. Note the presence of resistors at the ends of the bus. Each node is exposed to data traffic on the bus, but it will only respond if data is directed to it. Otherwise the data is ignored. A bus topology has the disadvantage that failure of the bus cable will stop communications. End resistors with identical resistances are used to improve signal quality on the bus.

A **Point-to-Point** connection is the simplest example of the bus topology. Point-to-point connections are used, for example, to connect a PC and a single printer.

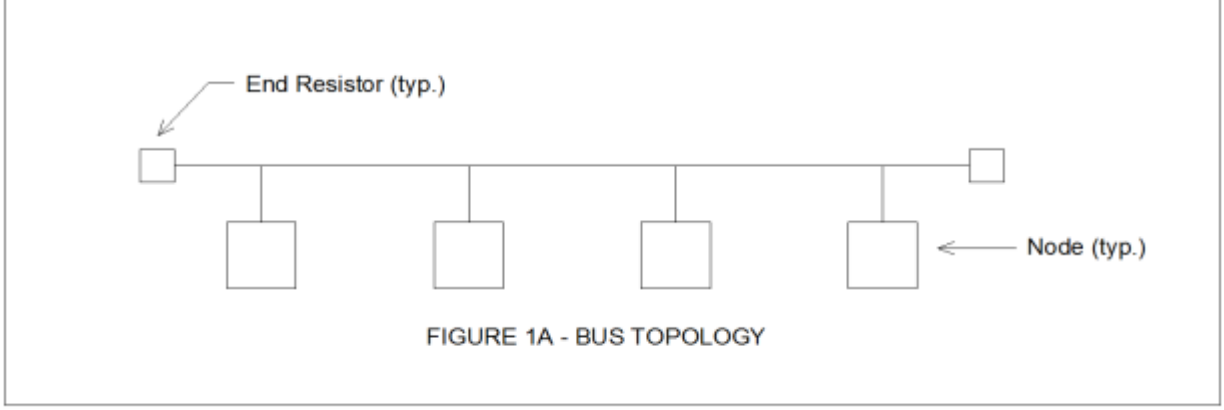


FIGURE 1A - BUS TOPOLOGY

In a **Star** topology (Figure 1B), individual nodes are connected to a central node. Very often the central node is a **Switch**. Switches allow temporary pathways to be made so any node on the network can communicate with any other node. In a star topology, an individual node can be disconnected without affecting communications on the rest of the network. It's more reliable than a bus topology. All data traffic stops if the central node fails. However, switches are built for high reliability, and often, Uninterruptible Power Supplies are connected to provide temporary backup power in event of loss of line power. Sometimes redundant switches are used for improved reliability. In that case, each node has two ports, with separate cables attached to each distributed node.

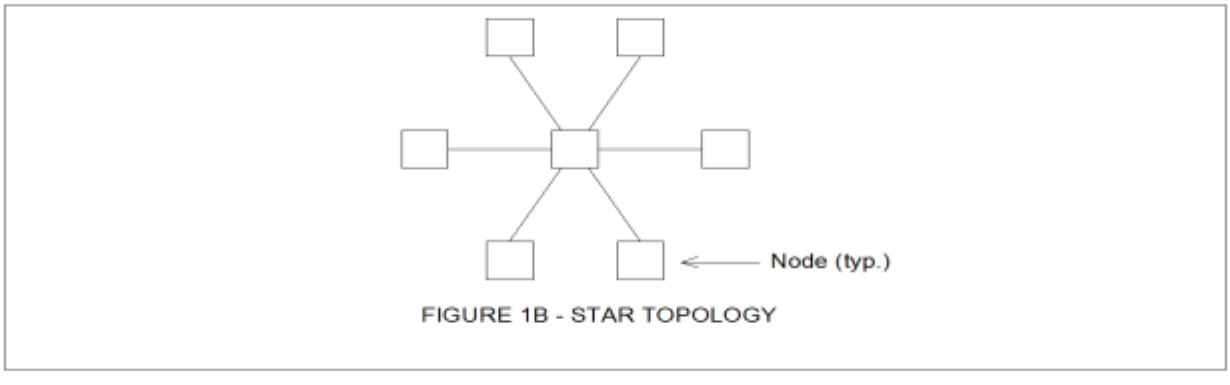


FIGURE 1B - STAR TOPOLOGY

Figure 1C on page 9 illustrates the **Ring** topology. The ring doesn't have a master device. Each node can both send and transmit data. Data sent from one node to another is forwarded around the ring from the originating node to the destination node to which it is addressed. If a segment fails, data can be sent in the reverse direction.

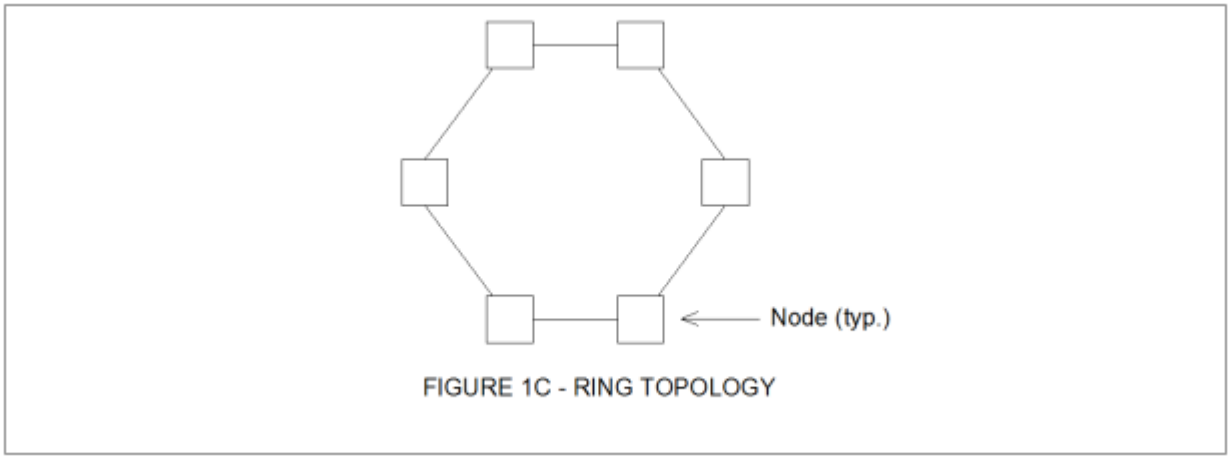


FIGURE 1C - RING TOPOLOGY

Many different types of devices besides computers, controllers and switches can be can be part of a network – such as printers, scanners, barcode readers, TV camera, etc.

In reality, *networks are interconnected in many different ways*. Some networks consist of combinations of one or more of the three basic networks. In a properly set-up network, data can get from one node to another as long as a path for data transmission exists. No doubt many readers have seen or worked with networks much more complex than those shown in this course. See References 5 and 6 for information on other types of networks.

Most networks have a central computer known as a **Server**. Servers are computers which meet higher standards for dependability, durability, and speed of access to data than ordinary desktop computers. Typically they also have far larger data storage capacity. Servers are frequently set up in redundant pairs. Computers in a network which act only as "dumb terminals" - which operators can use for monitoring and control, but don't directly interact with controllers or perform processing tasks - are referred to as **Clients**. A server processes requests from its clients and interacts with controllers. Servers are often referred to as **Thick Clients**, and client PCs are sometimes called **Thin Clients**. Such networks are usually called **Client-Server Networks**.

Many plants and facilities have multiple client PCs and controllers in various locations. So use of servers in process control applications is sensible for 4 reasons: first, to centralize key databases used to monitor the site in one reliable computer (or redundant pair of computers). Second, historical data can be stored on servers. Third, use of servers simplifies access to shared resources such as printers. Last, a Client-Server network allows appropriate delegation of roles and tasks to different entities. Each local

controller - a PLC, DCS, or PAC - can execute control over systems in a specific area of a site, and report the status of its inputs and outputs to the server. And client operator terminals that are networked with a server can access the server's database(s) for several purposes:

- To allow operators to monitor operations at a plant's subsystems.
- To let operators start and stop equipment as needed, and adjust set points for automatic control.[3]
- To view alarm screens.
- To access trend screens which show how key levels, pressures, temperatures, etc., have varied over time.
- To view accumulated values such as total flow in a day, and numbers of equipment starts.
- To view archived data.

A diagram of a sample network follows on the next page. This network is an Ethernet network. Note how it shows a variety of components. Brief explanations of some network components follow in this course.

Some communication cables have one group of conductors or fibers for communication in one direction, and another group of conductors or fibers for communication in the other direction. Regarding such cables, two communications terms sometimes used are **Half-duplex** and **Full-duplex**. **Half-duplex** refers to the transmission of data in only one direction at a time on a cable or other data link. **Full-duplex** refers to the transmission of data in two directions simultaneously. Typical copper Ethernet cable has separate pairs of conductors for data transfer in opposite directions.

[3] For these purposes, the server passes on commands from client PCs to the appropriate local controller.

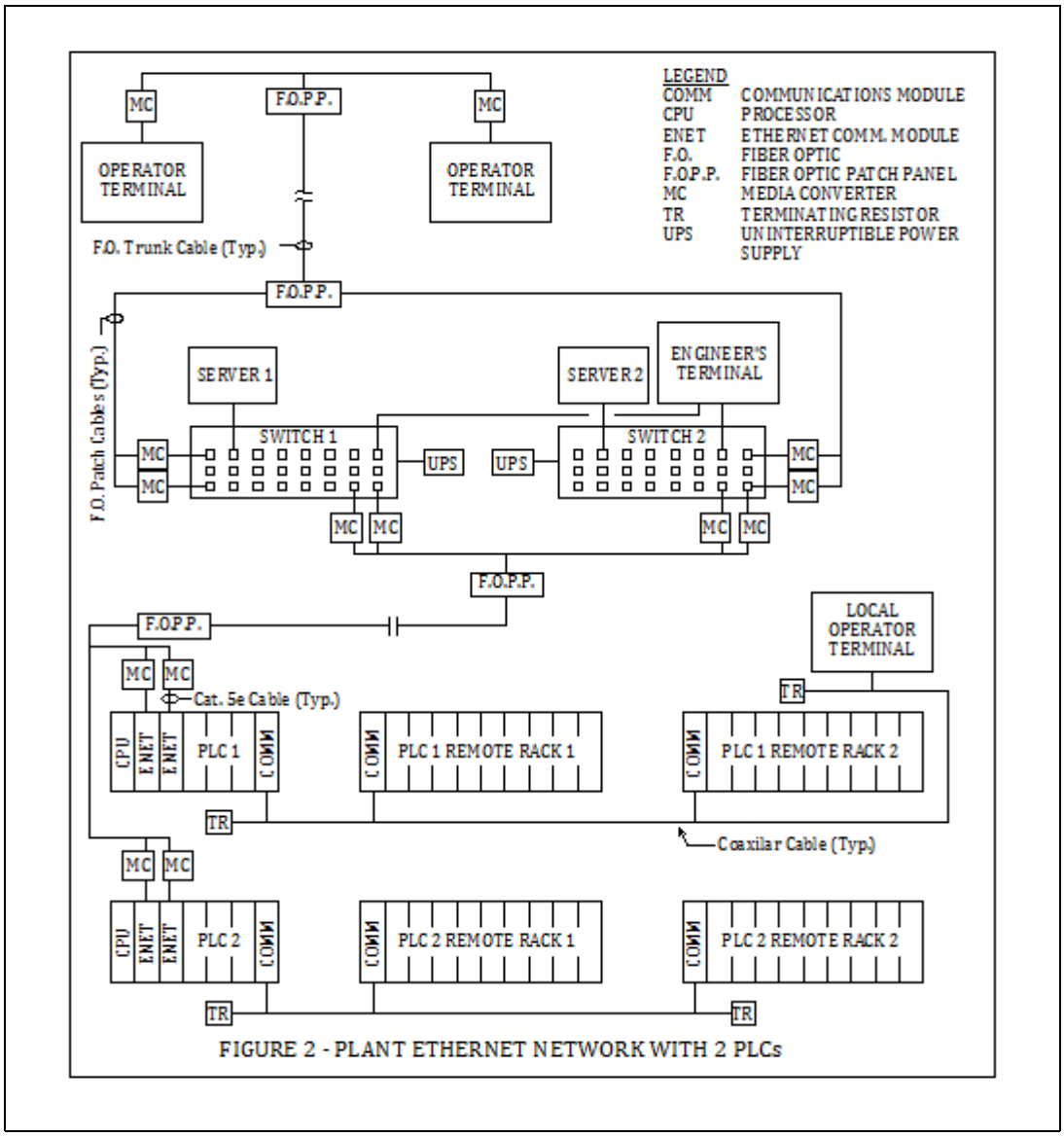


FIGURE 2 - PLANT ETHERNET NETWORK WITH 2 PLCs

The OSI Model and its importance

The **OSI**, or **Open Systems Interconnect** model, is the next topic. The OSI model is a theoretical model of how communications occur on a network. It has 7 layers. It's helpful to refer to the OSI model to explain features of protocols, hardware and networks. The layers are:

Layer 1 (Physical): This layer considers only the physical aspects of a network; the cables, converters, interconnecting devices, etc.

Layer 2 (Data-link): This layer concerns itself with how Layers 1 and 3 work together.

Layer 3 (Network): This layer provides an addressing scheme for routing of data and messages.

Layer 4 (Transport): This layer makes sure that messages get to their correct destination.

Layer 5 (Session): This layer handles the actual connections between systems.

Layer 6 (Presentation): This layer deals with the way different systems represent data.

Layer 7 (Application): This layer concerns itself chiefly with the software applications used on a computer screen.

Use of protocols involves both software and hardware, and it's hard, but sometimes necessary, **to differentiate between functions performed by hardware devices,** and **functions performed by software**. So it's helpful to refer to the OSI model sometimes.

Hardware and Connecting Cables

Communication and control protocols operating on a network need hardware devices and connecting cables to work. Some common hardware devices are built into computers and controllers - or attach directly to them. Many do not. A few of each sort are listed below in Tables 1 and 2. Table 3 lists some network communications cables.

TABLE 1: Computer and Controller Network Devices

Device	Description
Serial ports	Data is transmitted one bit at a time through serial ports. RS-232 serial ports usually either have 9, 15 or 25 pins. Universal Serial Bus (USB) ports, which allow data at much higher speeds than RS-232 ports, have become the dominant type of serial port.
RJ-45 receptacles	These are receptacles for cable with RJ-45 connectors. Industry-standard RJ-45 connectors are applied to the ends of standard twisted-pair copper Ethernet cables with 8 conductors (e.g., Cat. 6 cable.)
Network Interface Cards (NICs)	NICs allow a computer to be connected to a network. Most NICs have an RJ-45 receptacle. These cards also have a built-in MAC (Media Access Card) card for the computer-to-network interface.

Small Form-factor Pluggage (SFP) Transceivers	SFP transceivers are plug-in devices which are inserted into dedicated slots of switches, routers, and some Network Interface Cards. Most SFPs are used for interface with fiber-optic cables. Another less-often used type permits interface with copper Ethernet cable.

TABLE 2: Standalone Network Devices

Device	Description
Switches	Switches provide connectivity for many computers on a network, and connectivity to other network devices, including other switches. Switches commonly have as few as 4, and as many as 48 ports for Ethernet cables. Switches usually have at least one port for a cable to connect to another switch or a router. Switches are usually (but not always) **managed** devices, that is, configurable by the end user. In Ethernet networks, they reduce chances that "data collisions" will happen if simultaneous data transmissions occur.
Media converters	Media converters provide bi-directional communications by interfacing copper Ethernet cable and fiber optic cable. Media converters are often rack or DIN-rail mounted.
Wireless transceivers	These devices facilitate wireless communication on a network.

TABLE 3: Some Network Communications Cables

Device	Description
Twisted-pair wiring	Twisted-pair copper cables with varying numbers of conductors are used in some networks, e.g., RS-485. Resistors are applied to the far ends of some networks with twisted-pair cables. (Twisted-pair wiring may or may not have a shield around it – and/or individual pairs may be shielded.)
Coaxial cable	Coaxial cables are also widely used. Appropriate connectors and receptacles, and use of terminating resistors are essential with coax.
Copper Ethernet cable	Copper Ethernet cables have 4 Unshielded Twisted-Pairs (UTPs). RJ-45 connectors are usually attached to each end. The different types of Ethernet cables (Cat. 5, Cat. 5e, Cat. 6, etc.) have subtle differences. The maximum rated length recommended for Ethernet cables varies with the type used.
Crossover cable (copper type)	A crossover cable is an Ethernet cable in which the RJ-45 end connectors are applied so the Tx (transmit) conductors of

	one receptacle are connected to the Rx (receive) conductors of the other receptacle. Normally, Tx conductors on one end connect to Tx conductors on the opposite end, and likewise for Rx conductors. These cables are sometimes used for direct computer-to-computer communications without an intermediate switch. They're usually fairly short.
Fiber optic cable	Fiber optic cable is used to transmit data over much longer distances than copper Ethernet cable. It can carry a great amount of data. Fiber optic cables are either "single mode" or "multimode", and different types of cables pass light at different wavelengths. Individual fibers have very small diameters, and the ends of fiber optic cables must be prepared by a specialist before end connectors (there are about 6 common types) are applied. They are available as either trunk (long) cables, or patch (short) cables. Patch cables come with end connectors applied.

Communication methods

The last topic to be covered before interface standards and protocols are discussed in detail is communication methods. Three common methods for communications between networks are **Master-Slave**, **Token-Ring**, and **Ethernet**.

Master-slave communications is quite simple. It's also used often in industrial controllers. In this technique, a master device issues commands to one or more slaves, and the slave(s) respond. For example, a PLC which is connected to 3 external (or remote) racks may be the master. Every 40 milliseconds, the PLC's processor first checks for any altered inputs in the local rack (the rack where the processor is located); then it sends a query to the 1st external rack to see if any inputs have changed, or hardware failures have occurred. Any changes are reported back to the processor. The process is repeated for the 2nd external rack after querying of the 1st rack is done; and the 3rd rack after the 2nd rack has been checked. Then the processor runs its program, changes any outputs if necessary, then it sends, or "writes" new data to the output registers of modules in the local rack and 3 external racks. Usually, the slave devices are *not* controllers – but sometimes they are. In the latter case, a master controller might send a set point to a slave controller.

Token-Ring networks rely on token passing for data communications. As the name suggests, token-ring communications is used on ring networks. "A special message, called the token, is passed from one machine to another around the ring, and each machine can transmit only while it is holding the token."[2] Token-ring networks never

became prevalent in business and industry. However, *token-passing doesn't require a ring network for use* – it can be, and is, used on networks of other topologies.

So, some protocols rely on token-passing but don't use a ring network.

Since the Internet became widespread in the 1990s, the term **Ethernet** has become common. Ethernet is essentially a method for communications between computers and other devices on a network. Ethernet makes use of Carrier Sense - Multiple Access with Collision Detection, or CSMA/CD. On an Ethernet network, any node can communicate with any other node. It works like this: first, before a computer or controller sends data to another (or more than one) device, it listens to see if the network is busy. If the network is active, it will wait and try to send data later. If it senses the network is available, it will transmit the data. However, due to network time delays, a node may start sending data before it senses data release by another network node. In that case, a **data collision** occurs, and neither node will be able to successfully transmit data. Afterwards, each device will wait a random amount of time before attempting data transfer again. Ethernet works better than any other method for large networks. That's why the Internet uses Ethernet. And use of the Ethernet method has become common among industrial controllers. Furthermore, communication speeds on Ethernet networks are fast – typically 100 megabits per second (Mbps).

Peer-to-peer communications

When any network can transmit data to any other node on the network, the network is called a peer-to-peer network. Peer-to-peer capability is only possible if the application layer protocol supports it. Peer-to-peer networks can be implemented on networks with many different topologies. A master-slave network is *not* a peer-to-peer network.

Deterministic communications

Communications that occur within predictable and repeated time intervals are referred to as **deterministic**. In controllers, deterministic communications capability is vitally important. Most programmable controllers check and see if any inputs from field devices have changed many times per second. If any have, the controller responds rapidly.

Deterministic communications is also referred to as **cyclic communications** – notably in the PROFIBUS standards.

Ethernet is "nondeterministic" - communications occur on an as-needed basis. But some industrial communications protocols use deterministic adaptations of Ethernet[4].

[4] For example, EtherNet/IP and Modbus TCP/IP.

Use of managed switches effectively makes most networks deterministic[3]. However, switches used in industrial Ethernet networks should be segregated from office networks to ensure determinism.

Interface standards and devices

Interface standards are discussed below to minimize confusion. That's because the same type of interface, or interfacing device, is often used by more than one protocol for communications. Two interface standards and one interface device are mentioned below:

RS-232: **RS-232** is a Telecommunications Industry Association (TIA) standard for data transmission at speeds up to 20,000 bits per second (bps). It is used for serial data transmission. RS-232 is used only for data transmission from one device to another.

RS-232 interfaces were built into many legacy computers, and devices like controllers and printers. RS-232 ports most often have D-shell 9-pin, 15-pin, or 25-pin receptacles. Most RS-232 cables have male ends. In cases where the numbers of pins differ between connected devices, an appropriate cable will be needed. (These can be obtained from electronics suppliers.) RS-232 cables can be no longer than 50 feet.

RS-232 ports have been replaced by USB ports in contemporary use. USB allows far faster data transmissions with a simpler cable. However, many controllers are so reliable that they're used for 15 years or longer. So the typical reader may have seen RS-232 ports and cables.

RS-485: **RS-485** is an interface standard for data communications that allows much higher data transmission speeds than RS-232 - up to 1 megabit per second (1 Mbps) on a 100 meter cable. Data communications speed varies inversely with the total length of an RS-485 cable; the shorter the total length, the faster the maximum speed and vice-versa. At RS-485 terminals and on conductors, a small negative voltage represents an off bit, and a small positive voltage represents an on bit.

The technically correct term for this standard is TIA-485. But it is still referred to as RS-485 because the standard has existed for decades and has usually been called RS-485.

Most often RS-485 networks consist of point-to-point connections to nodes from a bus cable. Twisted-pair wiring is used for the bus cable. It offers high immunity to electrical noise when implemented properly. Terminating resistors are used on each end of the network to minimize communications problems. In most RS-485 networks, one master device communicates with slaves. It is a relative inexpensive network to supply and

install into controllers. This makes it an attractive option for relatively simple networks and applications. RS-485 is widely-used.

RS-485 is on the physical layer of the 7-layer of the OSI model. It allows use of only 1 master node, and supports up to 32 nodes on a bus. Additional nodes can be used if repeaters are used. More than one protocol can operate on RS-485 networks.

Figures 3A & 3B depict RS-485 setups. Tx indicates data transmission, Rx indicates data reception.

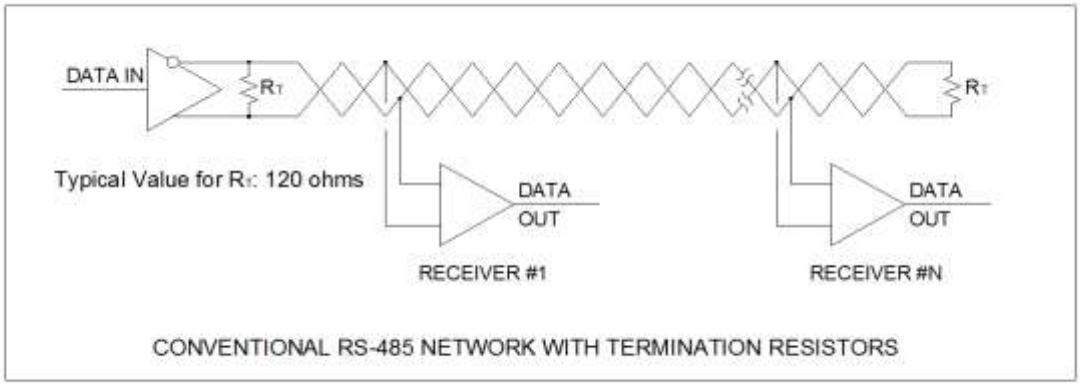


Figure 3A – Typical 2-wire RS-485 Network

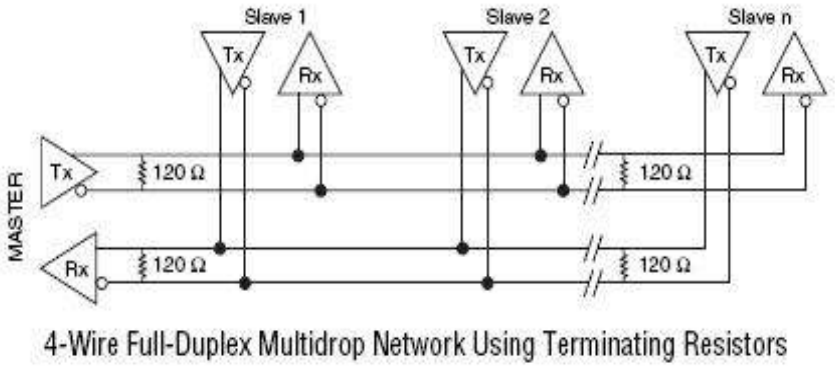


Figure 3B – 4-Wire RS-485 Network

Figure 3C shows how an RS-485 cable segments can be joined to a terminal strip near an RS-485 node. In this arrangement, it's possible to temporarily disconnect an RS-485 node without interrupting network communications.

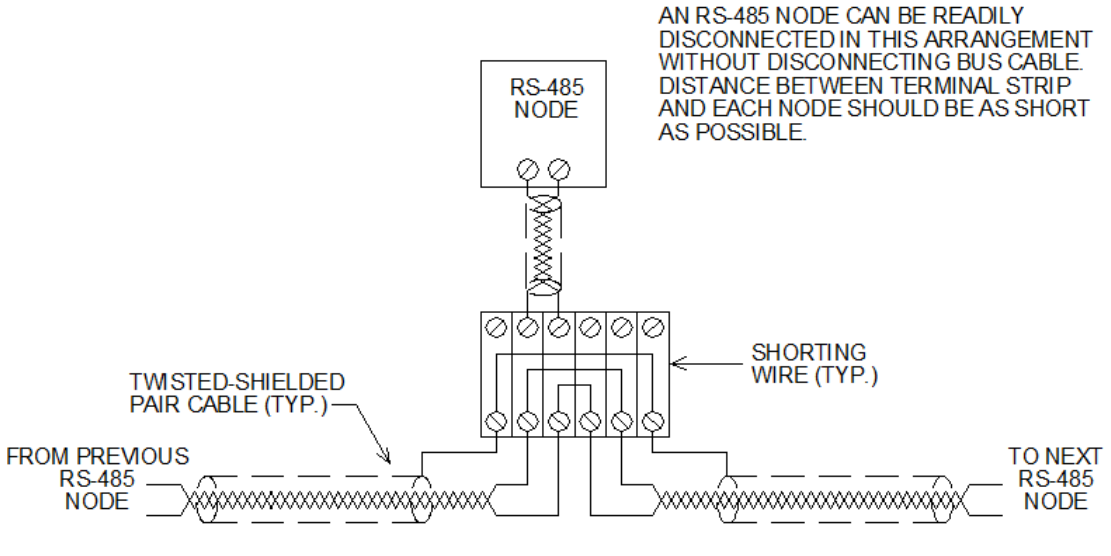


FIGURE 3C
RS-485 BUS-TO-NODE DROP EXAMPLE

RS-485 networks often are used in applications like building HVAC (Heating, Ventilation and Air Conditioning) systems, where system parameters change slowly and fast communication is not essential.

RJ-45: RJ-45 ports[5] and cable connectors have 8 conductors, and RJ-45 connectors are attached to twisted-pair copper Ethernet cables, e.g. Category 6 cables. RJ-45 ports are built into Network Interface Cards (NICs) in computers, controllers, Ethernet switches, media converters and other devices. Communication speeds of 100 Mbps and higher are often realized on Ethernet networks.

Converters: Converters can be bought to allow different types of serial devices to interface. One example is an RS-485 to USB converter. Sometimes they're helpful.

Common Features in Protocols

Many different protocols are used for data communications, but most that have found widespread acceptance share some features. Some are mentioned below.

Data transmissions are usually done in consecutive clusters of many bytes. Each cluster consists of parts that have either fixed or variable length, depending on the

[5] Technically, an RJ-45 port is one type of 8P8C (eight-piece eight contact) connector.

protocol and particular situation. Many protocols have specific ways to mark the start and end of each transmission.

In TCP, or Transmission Control Protocol, which is used on the Internet and in most Ethernet networks; these consecutive clusters are called **frames**. **Frames**, in turn, are broken down into smaller units called **packets** that are handled by routers.

Individual data transmissions on a network also typically include the address of the source node, and address(es) of the destination node(s).

Each protocol also has its own method (or user-selectable method) for **error checking**. **Error checking** is used in protocols to make sure the data received at a destination node is identical to the transmitted data. Electrical noise, connection problems, or a failed part may interfere with effective data communications. If the sending node gets a response from the receiving node that the data was received as intended, the task is done. But if a mismatch was detected, or no acknowledgement that the data has been received properly is returned, than either the data will be resent, an informative message will be generated, or another appropriate action will be taken.

Protocols exist for communicating data. But different protocols are used for different purposes. Yet, in plant control applications, much of the data falls into four categories:

- Analog inputs
- Analog outputs
- Discrete inputs
- Discrete outputs

Sometimes other types of data, like character data, are also transmitted. Log-on names, passwords, text messages, and product barcodes are examples of character data[6].

Each protocol used in plant control applications has its own unique way of characterizing different types of data.

Most people using communications and control protocols won't have to be concerned about these details - but one never knows. It might, for example, become necessary to choose between two or more error-checking methods.

Some Notable Automation Companies

[6] ASCII, extended ASCII, and Unicode are some widely-used standards used for representing character data.

Some of the most widely-used control protocols in use were developed by important suppliers of automation equipment. Three suppliers are mentioned below.

The first is **Modicon**. Modicon built the first PLC ever used. It was 1 of 3 functionally similar controllers installed at a transmission plant of a General Motors subsidiary in 1969. Modicon was an independent company for about 8 years before it was purchased by Gould in 1977, then by AEG in 1994. Modicon has been owned by Schneider Electric since 1996. Modicon remains active in the PLC/ PAC market around the world. It has maintained a significant market presence for over 40 years.

The second company is **Allen-Bradley**. Allen-Bradley, a prominent manufacturer of electrical equipment, was the dominant PLC supplier in the United States by the mid-1980s. It was by far the main PLC supplier to car and truck manufacturers. Like Modicon, it has a large international presence. In the late 1990s, Rockwell Automation bought Allen-Bradley. It still has a sizeable market position.

The last is the German multinational company **Siemens**. Siemens had a large market position in Europe by 1990. It now has a world-wide presence. Siemens collaborated with other German companies and a bureau of the German government to develop the PROFIBUS protocols.

Proprietary and Open Protocols

Protocols subdivide between **Proprietary** and **Open** types. Allen Bradley's DH+ (Data Highway Plus) is an example of a proprietary protocol. It could be used with many Allen-Bradley PLC-5 family PLCs, but not with PLCs made by other OEMs. Open protocols are (or were) developed by more than one organization acting in collaboration. Most open protocols developed as a result of long, iterative processes of proposals, consultations, and adjustments to proposed standards prior to release of a standard. The people involved in development of open protocols developed them to increase standardization; to provide reliable communications; to reduce end users' costs; and to increase end users' convenience. Some open protocols are revised from time to time. FOUNDATION Fieldbus H1 protocol is an example of an open protocol.

The HART Protocol

In the 1970s, end users starting preferring transmitters with 4-20 milliampere (mA) signal outputs over other options. Over 40 years later, 4-20 mA output transmitters remains in widespread use. A typical transmitter is connected to a DC power supply, and sends a 4 to 20 mA signal to a controller that's linear (or adjusted to be nearly linear) in a range between the low and high limits of a transmitter's calibrated range.

Rosemount, Inc. was the most notable supplier of transmitters in the 20th century for a wide variety of measurement devices. In the late-2010s, it still has a large market presence. (Rosemount was bought by Emerson Electric in 1976.)

In 1982, Rosemount introduced its **HART** (Highway Addressable Remote Transducer) protocol.

A simplified discussion of HART follows. HART transmitters place a wave atop the 4-20 mA signal that isn't needed or sensed by a controller, and doesn't affect the output of the transmitter. The wave's frequency is continuously varied depending on whether a bit is 0 or 1. The wave contains data about the circuit, measurement, and transmitter. When a HART Handheld Terminal[7] is connected to the 4-20 mA circuit[8] with a HART-compatible transmitter, the user can monitor the circuit and adjust key transmitter aspects from the terminal. This simplifies the tasks of checking for problems, and changing a transmitter's range without removing the transmitter's cover. The superimposed wave is immune to electrical noise and ground loop currents, which can affect a purely analog 4-20 mA signal.

HART communications can sometimes be used to represent two or more signals. For example, the air flow measurement inferred from a differential pressure (D.P.) transmitter depends on air temperature. If a HART-capable D.P. transmitter also measures temperature, both measurements can be transmitted on the same cable.

HART was originally proprietary, but now it is an open protocol. HART-compatible transmitters have been offered by many OEMs besides Rosemount for some time.

These days, some HART devices have either IEEE 802.3 radio wireless compatible receivers, or receivers that can be used with Bluetooth, a communications technology that relies on short-range transmission of signals. These types of devices are used with handheld HART interface devices. An IP compatible version of HART also has been developed.

TCP/IP

TCP (Transmission Control Protocol) & **IP** (Internet Protocol) – known as **TCP/IP** - are key protocols. They are used together on the Internet and in Ethernet networks in commercial and governmental organizations. "TCP/IP is a standard for transmitting data in packets from one computer to another .... The two parts are TCP, which deals with

[7] Configured personal computers can also interface with HART transmitters.

[8] Most controller analog input modules have a built-in dropping resistor for each input. Typically it follows the negative circuit terminal. Both HART communicator connections to a 4-20 mA circuit must precede the resistor.

construction of data packets, and IP, which routes them from machine to machine."[4] IP uses 4 "octets" for a sum of 32 address bits. The decimal equivalents of each octet range from 0 to 255, and periods separate octets in an IP address, e.g., 192.168.41.23. There are restrictions regarding IP addresses. Some can't be used, and Internet Protocol version 4 (IPv4) is quickly running out of available addresses for public access.

TCP/IP is mentioned here because some protocols used in control applications are adaptations of TCP/IP. For example, in Modbus TCP/IP, TCP/IP is like an envelope for data transmitted per the Modbus protocol. In these adaptations of TCP/IP, data is sent from one node to another on a network using the same node addressing scheme that IP uses.

In the late 2010s, IPv4 is being used around the world. IPv6 (Internet Protocol version 6), with 128 address bits, will eventually replace IPv4.

Control protocols

Many control protocols are used with controllers. Some of the most widely-used ones are explained in the remainder of this course.

Modbus and some of its variants: The original, proprietary **Modbus** protocol was developed by Modicon in 1979 for use with Modicon PLCs. It is one of the earliest control protocols ever developed. Data transmission in Modbus is serial – one bit follows another. In 2004 Modbus became an open protocol managed by the Modbus Foundation. Modbus is a relatively simple protocol that uses numbering schemes that are familiar to those who have programmed Modicon PLCs. Some commercially available instruments, meters, indicators, and actuators are Modbus-compatible.

It's correct to refer to Modbus as a control protocol, but it also can be used for data communications applications that don't require control capability. Many commercially available transmitters and indicators are Modbus compatible. The same is true for FOUNDATION Fieldbus H1 and HSE, and PROFIBUS PA, which are examined later.

Modbus Characteristics

Master-slave network	
Application layer protocol	It isn't hardware dependent.
Serial data communications	The maximum obtainable speed varies with the variant of Modbus used, and equipment on the Modbus network.
Maximum nodes allowed	247 slaves possible with repeaters but 32 is

	standard.

Modbus Variants

Modbus is used on different types of networks for different applications. Here are two of its variants:

- **Modbus RTU:** This variation is very widely used. It's typically used on RS-485 networks. Master-slave communications are used. It includes error checking. All devices on a Modbus RTU network must be set to the same communications speed.
- **Modbus TCP/IP:** This is Modbus adapted for use on an Ethernet network. It is also known as Modbus IP. Process data in messages communicated using Modbus TCP/IP are "embedded" in a TCP frame. Nodes in a network using Modbus TCP/IP (and other protocols that use TCP/IP) use IP addresses.

The Appendix has an overview of the Modbus RTU protocol. It is meant to help readers understand how a communications protocol works.

Modbus Plus

Modbus Plus has similarities to Modbus but also some differences. It's an application layer protocol that uses token-passing, network nodes use peer-to-peer communications, and allows use of multiple masters. It was developed as, and remains, a proprietary network. It is managed by Schneider Electric. Network nodes may be PLCs, PCs, terminals, and drives.

Rockwell / Allen-Bradley Protocols

Some Rockwell/ Allen-Bradley protocols deserve attention because of the company's large market presence. The first two discussed are proprietary.

a. Data Highway Plus (DH+)

DH+ was probably used for communications between nodes more often in Allen-Bradley's PLC-5 product group[9] than any other protocol. DH+ is a token-passing protocol. Twisted-shielded pair wiring is used for communications between nodes. Maximum communications speed is 57.6 kilobytes per second (kbps). Rockwell/ Allen-Bradley no longer supports the PLC-5 family, which has been superseded by other products. DH+ is not used on new installations. A protocol converter, or **gateway**, is available from at least one firm$_5$ for communications between DH+ networks, and more

[9] The longevity of PLC-5 systems makes it worthwhile to mention DH+.

recent networks using the underlying Control and Information Protocol (CIP) applied in later Rockwell/ Allen-Bradley products. DH+ networks use end-of-line resistors.

c. Common Industrial Protocol (CIP)

CIP[10] is a protocol used in industrial applications which "*underlies*" and is common to the next 2 next protocols discussed. CIP's standards are now open protocols, managed by the non-profit organization Open DeviceNet Vendors Association (ODVA), Inc.

d. ControlNet

ControlNet is an open protocol that's notable for its use in Rockwell/ Allen-Bradley's ControlLogix group of controllers. Its functions are distributed in several layers in the OSI model. Allen-Bradley participated in development of ControlNet (and EtherNet/IP & DeviceNet). It's an open protocol whose standards are managed by ODVA. Rated communications speed is 500 kbps. Nodes in a ControlNet network are usually connected by RG-6 coaxial cables and connectors. Each end of the network must have a terminating resistor. It can support up to 99 nodes. The physical length of a ControlNet network can be extended using repeaters and/or media converters which interface copper and fiber optic cable.

Slave devices in a ControlNet (or EtherNet/IP) network are not limited to controller remote racks or other controllers. The author once participated in a project where 10 Variable Frequency Drives (VFDs) were linked to a branch of a ControlNet network. The use of many Input and Output (I/O) modules was avoided.

Use of ControlNet is becoming less common as EtherNet/IP gradually replaces it.

e. EtherNet/IP

EtherNet/IP is an open protocol which uses CIP and implements Ethernet communications between nodes for communications and control. It is managed by ODVA. ODVA will only certify EtherNet/IP hardware for use if it will work in industrial environments. Ethernet is non-deterministic – communications occur only as needed – but EtherNet/IP achieves deterministic communications for input & output updates, which makes it suitable for plant control uses. Processor to remote rack and processor-to-processor communications are possible with EtherNet/IP. Drives and Motor Control Centers (MCCs) which interface with EtherNet/IP are also commercially available. Such interconnections reduce the need to install conduits, cable trays, wiring, etc.

[10] CIP is also used by DeviceNet, but DeviceNet is not discussed in this course.

Ethernet-based protocols like EtherNet/IP have been gaining acceptance in applications like motion control that require fast response times. (Ethernet communications speeds are much higher compared to other network speeds.)

According to John Rinaldi of Real Time Automation, EtherNet/IP will become an even more important protocol in the next 5-10 years. Rinaldi says EtherNet/IP's high data transfer speed and deterministic communications make it an ideal candidate for acting as a sort of master protocol for use in large networks.

Some Important Open Protocols

The Fieldbus Foundation and its work

Early work on what became the FOUNDATION Fieldbus group of protocols was done by the ISA. In 1994, two industry groups working on similar goals merged to form the Fieldbus Foundation. This non-profit group, which became the Fieldbus Foundation, issued the first standards in 1996. They have since been updated (and expanded). The foundation's main goals all along were to develop an open protocol that users would want; to reduce end users' reliance on proprietary protocols; to provide standards for high reliability; and to test manufacturers' products to verify conformance to its standards. Regarding the last point, if two different field devices from different manufacturers both conform to FOUNDATION standards, either can be used in a FOUNDATION network.

The FOUNDATION protocols are discussed below. Both are targeted mostly to the DCS market.

FOUNDATION Fieldbus H1

The Fieldbus Foundation participated in development of an international field bus standard, IEC 61158. **FOUNDATION Fieldbus H1** complies with IEC 61158-1. It is for use for communications between a controller and field devices on a network - or between field devices. All communications on Fieldbus H1 networks are only digital. Each Fieldbus H1 communications network requires a fieldbus linking device, a power supply, and terminating resistors. Linking devices interface the fieldbuses with a DCS. Twisted-pair wiring that meets certain specifications is normally used for cables, but the standard also supports use of fiber-optic cables. Fieldbus H1 compatible transmitters and actuators have been offered by OEMs for some time now. Some of its other characteristics are listed below:

- 31.25 kbps communication speed.
- 32 network nodes supported.
- A trunkline (main) bus with branches to individual devices is the most common arrangement for Fieldbus H1.

- Schedules control activity in, and communications activity between devices.
- Regularly polls devices for process data.
- The trunk cable requires a power source in the range of 9-32 VDC, with 24 VDC ± 2 VDC recommended for most uses. Individual 2-wire end devices (typically instruments or actuators) usually draw all their power from the trunk line to which each branch is connected. But some network devices are "4-wire", using supplemental power.
- Standard or Enhanced function blocks are used for interaction between control systems and network devices.
- FOUNDATION H1 networks can be used in applications requiring intrinsic safety if the end user abides by certain practices.
- All values are communicated as floating-point values in engineering units.[11]

Fieldbus H1 networks can be connected using a variety of topologies, but the most common one is the trunk (master cable) and spur (branch) one. The tree topology[12] is also used frequently. Also, redundant buses may be used for interfacing to the same set of field devices for greater reliability, if the network hardware supports redundant connections. Many DCSs are Fieldbus H1 compatible. Fieldbus H1 networks can be deployed to provide Intrinsic Safety (I.S.) for bus-connected devices located where an explosion hazard exists, if the end user abides by certain practices[13].

Anyone considering using Fieldbus H1 for the first time might contact the Fieldbus foundation, or talk with controller OEMs.

An image of a FOUNDATION Fieldbus H1 network is shown in Figure 4.

FOUNDATION Fieldbus HSE

FOUNDATION Fieldbus HSE (High Speed Ethernet) is a communications and control protocol which merges process control and Ethernet connectivity & communications. Deterministic communications are used. The high data transfer bandwidth of Ethernet is its main advantage. It uses redundant cabling. Communications over distances exceeding a mile can be obtained with Fieldbus HSE certified equipment. Approved hardware is manufactured for industrial environments. It supports up to 255 nodes, however, some of the nodes are reserved for special uses.

[11] Instrument Engineers' Handbook, Process Measurement and Analysis, Chapter 1.6. (4th Edition, 2003)

[12] See Reference 5 for an explanation of the tree topology.

[13] Intrinsically safe practices limit energy in circuits so explosions can't happen. They're applied in environments where explosion hazards exist.

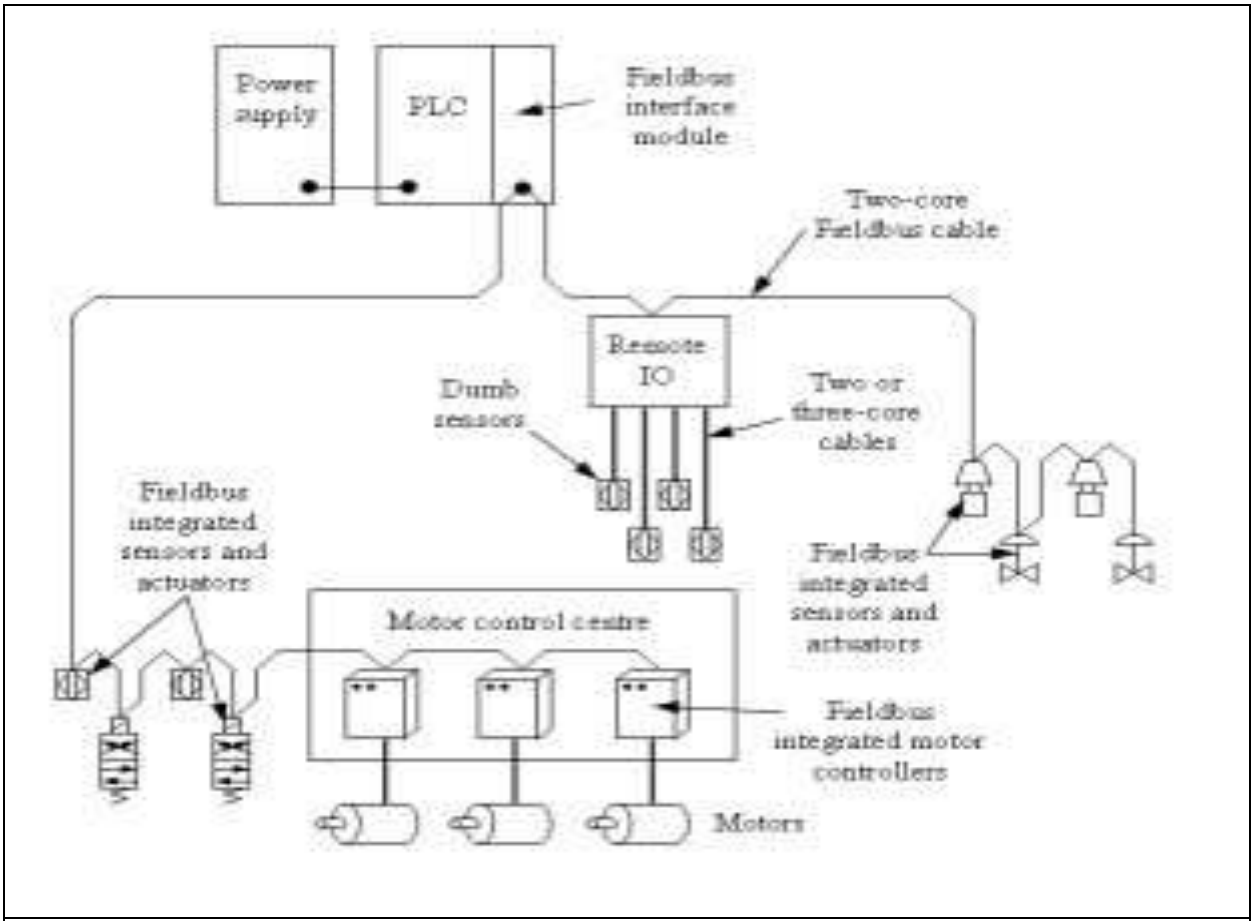


Figure 4 - A Fieldbus Network

The PROFIBUS STANDARDS

The PROFIBUS (Process Field Bus) standards arose from the efforts of a consortium of German companies including Siemens Corp., and the German government. These standards, and the FOUNDATION Fieldbus standards, have many similarities. The PROFIBUS standards are maintained by PROFIBUS & PROFINET International (PI). PI tests vendor equipment for conformance to the applicable standard(s). These protocols are all open.

The PROFIBUS protocols are not as widely-used in the United States as they are in Europe and Asia. The PROFIBUS standards conform to IEC 61158.

PROFIBUS PA

PROFIBUS PA (Process Automation) is a Profibus standard that applies to field devices and bus cables. It is interoperable with PROFIBUS DP. Its rated communication speed is 31.25 kbps. PROFIBUS PA and FOUNDATION Fieldbus H1 are similar. PROFIBUS PA compatibility is offered by many instrumentation and actuator OEMs. Some of its other characteristics are listed below:

- Implemented with either twisted-pair or fiber-optic cable. If twisted-pair cables are used, end of line resistors are required.
- RS-485 topology is standard.
- Star, bus, and ring topologies possible when fiber optic cable is used.
- Connectivity to PROFIBUS DP controllers is possible.
- Can support field devices used in environments where explosion hazard exists.
- 126 nodes can exist on a network when hubs and repeaters are used.

PROFIBUS DP

PROFIBUS DP (Decentralized Peripherals) is another PROFIBUS standard which applies to controllers. Profibus DP is marketed for PLC applications. Some of its features include:

- It supports cyclic (deterministic) communications.
- 126 network nodes supported.
- Can optionally be configured with redundant cabling.

PROFINET

PROFINET is for use on high speed Ethernet networks. It has similarities to EtherNet/IP and FOUNDATION Field Bus HSE. PROFINET devices are designed to be rugged enough to work in challenging factory and process control environments. Just as EtherNet/IP is slowly replacing ControlNet, PROFINET is likewise gradually supplanting PROFIBUS PA and DP.

Anyone considering using any variation(s) of PROFIBUS for the first time might talk with controller OEMs first.

Protocols used with HMIs

No course on control protocols would be complete without emphasizing that protocols are also needed for communications between controllers and **HMIs – Human-Machine Interfaces**. HMIs are also called Man-Machine Interfaces (MMIs) and Graphic User Interfaces (GUIs). For plant operators, a typical HMI is usually desktop computer with a monitor, keyboard and a mouse. Controllers, HMIs, and the networks which interconnect the devices are more broadly called **SCADAs** - Supervisory Control and Data Acquisition Systems. In reality, a plant with more than two HMI terminals will usually have a server computer, or redundant server pair.

People use HMIs to monitor what controllers are doing, provide key information on plant operations, and enable operators to operate equipment and systems from remote locations. Operators must be able to start up and shutdown equipment and systems; switch between automatic to manual modes; change set points of automatic control

loops; and intervene in other ways. Also, operators need to be able to view and respond to alarms; monitor trends in processes; and access historical information. This is the role of HMIs.

An aspect of HMI software worth mentioning is that typically, it will request updated data from the controllers to which it is connected at a periodic rate, and in a sequence, even as each controller goes on executing its program. HMI data refreshes typically occur at longer time intervals than PLC or DCS cycle times.

Windows OS and OPC

Microsoft Corp. developed its Windows operating system (OS) so it could be used with programs developed by many outside firms for all sorts of applications. Companies that developed HMI programs wanted them to be interoperable with Windows. So, HMI developers like Intellution[14] and Wonderware[15] made their software interoperable with Windows. HMI developers also found it useful to use some Microsoft programs – notably its spreadsheet program, Excel.

Some key Microsoft application programs use Object Linking and Embedding, or **OLE**. OLE allows two applications to operate together and share data. HMI developers and controller OEMs alike sought to use OLE techniques so HMIs and controllers could work together. This led to the development of **OPC** - Object Linking and Embedding for Process Control. OPC is now known as **Open Platform Communications**.

OPC was developed for a key purpose: to provide a common set of standards, so any OPC-compliant HMI software program would be able to interface with any controller. OPC compliance is a very desirable feature and selling point for companies that offer HMI software.

In practice, a software subprogram called a **driver** is needed to interface an OPC compliant HMI with a particular controller. The development of drivers was – and remains - time-consuming for both for the HMI vendors and controller OEMs. Many technical issues come into play. HMI vendors usually have separate drivers, only one of which can be selected, for interfacing with a particular OEM's controller. (And many controller OEMs use different software for different controllers, too!)

Sometimes the best choice for a driver is not obvious. For example: GE Intelligent Platforms' Intellution HMI software can use the Modbus Ethernet (MBE) driver to interface with Rockwell/ Allen-Bradley ControlLogix processors.

[14] A division of GE Intelligent Platforms, as of 2016.

[15] A division of Schneider Electric, as of 2016.

Rockwell Automation has a somewhat different approach. It offers several different HMI software programs such as RSView32 and FactoryTalk. But all of them rely on an OPC-compliant applications program called RSLinx[16] that acts as a network communications manager. RSLinx also has drivers that can be selected to interface with programmable controllers provided by other OEMs besides Rockwell/ Allen-Bradley PLCs and PACs.

The topic of HMI software is quite complex. This course only provides an overview.

Local operator terminals

In many plant applications, the end user's requirements for operator monitoring or control are fairly simple. In such cases, a single operator interface terminal is located close to the equipment or system being controlled, and a SCADA network with multiple remote terminals is unnecessary. Local operator terminals may have touch screens, or a mouse with a built-in trackball (which moves an arrow on the screen) for operators. In such cases, a single cable usually connects the controller and terminal. In applications like these, a complex driver which requires interoperability between different manufacturers' equipment (such as Modbus MBE, mentioned above) is not needed for communications between the processor and terminal.

Transmitters, Actuators and Protocols

For some time now, manufacturers have offered transmitters and actuators that can be ordered that are compatible with one or more of the following protocols: HART, FOUNDATION Fieldbus H1, FOUNDATION Fieldbus HSE, Profibus PA, and others. They have done this because use of protocols has become more common over time and it is advantageous to all concerned to have such capabilities as an option.

Disadvantages of using protocols

Intelligent use of protocols has many advantages. But protocols, fieldbuses and networks shouldn't be used without recognizing the disadvantages.

The main disadvantage is that using protocols and fieldbuses is that in some cases, communications with multiple devices (instruments and/or actuators) can be lost if a single device (like a power supply or interface module) fails, or, for some networks, if electrical continuity is lost in a segment of a cable.

Using conventional transmitters with 4-20 mA signals and twisted-pair wiring has advantages over using a protocol and a communications bus. So does using typical

[16] RSLinx is available in different versions for different application requirements.

wiring for interfacing with discrete sensors, and controlled devices like motors. Technicians are more familiar with conventional devices than equipment operated over a bus - and if a problem arises, it's often easier to find the cause of a problem and fix it. Sometimes there is a high turnover rate among techs, so if a few key people leave and have to be replaced, then it's easier to find replacement staff capable of using proven technology.

Before choosing to use a new protocol, decision makers also need to judge the capabilities of the installers, and in-house people who will be maintaining the system, doing things like modifying programs, replacing failed modules, and expanding the system. A new technology, with unfamiliar programming software and a new protocol, shouldn't be used unless it represents a good choice. In particular, the installers, OEM, and software providers should have a good record of customer support.

Summary

This course discussed some widely-used industrial communications and control protocols. It explained how use of protocols has improved reliability in plant applications. It reviewed different types of networks, network cables, hardware devices and network topologies. And it emphasized how reliable plant control depends on reliable networks.

The reality is protocols often are very complex, many evolve over time, and some cease to be used over time as technology changes and better alternatives emerge. So, the author updated this course in late 2019 to make it more current.

Appendix: Overview of the Modbus RTU Protocol

The Modbus RTU protocol is a simple protocol that's widely used both for communications only and control applications. There is one master device and multiple slaves. Slave devices respond if a query issued by the master is directed to it.

Each slave on a Modbus RTU network must be set to the same communications speed, and use the same *parity* setting: either Even, Odd, or None. A parity bit is added by the master after each byte in the data component of each frame transmitted (unless "None" is selected).

Each Modbus RTU frame sent by a master consists of 7 components, as tabulated below. Addresses can be either read-only (r) or read/write (r/w), that is, they can be both read from and written to. The function code will determine the type of operation performed.

Frame Component	Description
Start	Minimum 28 consecutive bits of silence
ID	Slave number (1-247 decimal)
Address	The first address in the slave device for the data of interest. When more than 1 consecutive address is being read from or written to, the slave is told how many registers will be accessed. See table on the next page. (8 bits minimum)
Function Code	8 of the most often used function codes are tabulated below. (8 bits)
Data (for write operations)	N * 11 bits, where N is the number of addresses. This count includes start, parity, and stop bits for each byte
CRC	Cyclic Redundancy Check This is a generated bit stream that's a function of the packet sent to the slave. The CRC response from the slave device will let the master know whether or not the transmission & response occurred error-free. If not, the master will take appropriate action. 16 bits.
End	Minimum 28 consecutive bits of silence.

Function Codes

1	Read Coil (output bit) status
2	Read Input (input bit) status
3	Read Multiple Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Holding Register
15	Write Multiple Coils (bits)
16	Write Multiple Holding Registers

Notes on Function Codes

- Registers have 16-bit (2-byte) memory spaces that typically store AIs and AOs.
- Holding registers can hold any type of data.

Modbus PLC Address Ranges[17]

Address Range	Description
000001-009999	For control applications, these are the same as discrete outputs (DOs). 1 bit sent by master per address (r/w)
010001-019999	Same as discrete inputs (DIs). 1 bit sent per address. (r)
030001-039999	Same as analog inputs (AIs). 16 bits sent per address. (r)
040001-049999	Same as analog outputs (AOs). 16 bits sent per address. (r/w)

Modbus RTU data is stored in registers (or consecutive registers) that hold either single bits, 16 bits, 32 bits, or 64 bits.

OEMs offering Modbus-compatible devices have considerable latitude regarding what the address ranges may be for each class of register. However, there can be no more than 65,536 (2^{16}) registers of each type. This is far more registers than are needed by Modbus-compatible field devices like transmitters and actuators.

Replies from slave devices

The master needs to know if a frame it sent was received properly by the slave. 3 common responses by a slave are:

- If write data is received as transmitted, the frame will be echoed back to the master as it was issued.
- If a read command was successfully sent and received, data from the registers will be included with the reply message.
- In event the data received by the slave was corrupted, incomplete, or can't be processed, an exception code will be returned to the master.

[17] The address ranges shown in the table above apply to late-model Modicon PLCs.

References:

1. Web resource: https://en.wikipedia.org/wiki/Hexadecimal (Wikipedia)
2. Web resource: https://en.wikipedia.org/wiki/Fieldbus (Wikipedia)
3. Web resource: http://www.fieldbus.org/
4. Web resource: www.profibus.com/
5. “Introduction to Computer Networking” (Course E175, Dale E. Callahan, www.pdhcenter.com/ www.pdhonline.org)
6. "How to Develop High Reliability Ethernet Control Systems Using Media Redundancy" (Acromag, Inc. White Paper)
7. “An Introduction to Modbus® Serial Communication” (Precision Digital Corp.)

Endnotes:

[1] https://en.wikipedia.org/wiki/Fieldbus

[2] Dictionary of Computer and Internet Terms, Ninth Edition, Barron’s Educational Series, Inc.

[3] Introduction To Modbus TCP/IP, Publication 8500-765-A05C000, Acromag Inc., 2005

[4] Dictionary of Computer and Internet Terms, Ninth Edition, Barron’s Educational Series, Inc.

[5] Prosoft Technologies, Bakersfield, California.