Industrial Networks
TABLE OF CONTENTS

Wireless wherever ................................................................. 4
A new dawn for fieldbus ....................................................... 14
Introduction to Modbus ......................................................... 16
MQTT and open SCADA ....................................................... 26
Expanding field networks ..................................................... 28

AD INDEX

Newark .............................................................................. 3
Wago ................................................................................. 13
Discover Over a Million Engineering Products from Suppliers You Know and Trust

900+ new products each week

Custom services such as kitting, panel meters, enclosures, and many more!

Market-leading online community of over 600,000 engineers
Wireless wherever

Self-healing mesh, WirelessHART, LoRaWAN and other innovations push boundaries to reliably deliver data from remote, hard-to-reach locations

by Jim Montague

It's well known that reach often exceeds grasp, but wireless lets process applications and users grasp as far as they want. Plus, updated wireless standards and devices are letting them do it more easily and at less cost than ever.

"Oil and gas wells have been in California’s Central Valley for 100 years, and wireless was used in the form of radios for 40 years in multiple locations. This is because we have to cover thousands of pump jacks over 100 square miles, and wireless is the only way. The radios were point-to-multipoint, and now we have wireless Ethernet and licensed, point-to-point, private microwave communications," says Jonathan Polly, solutions operations architect at Chevron’s (www.chevron.com) San Joaquin Valley business unit (SJVBU), which operates six oil and gas fields in the valley. “The pump jacks are instrumented with wireless to transmit data from automated testing of groups of wells and managed steam flooding needed by the heavy oil applications.” (Figure 1)

However, even though SJVBU used radios for decades, these older wireless methods were more costly and less flexible, making it impractical to get all the data they need. Besides learning about and adopting new wireless technologies, SJVBU also retained radio frequency (RF) procedures and skills that were still useful even as they sought to modernize.
LORAWAN IN THE VALLEY

“We have 18,000 wells and devices in the field to operate and maintain, and many data points were still manually collected. However, even though there was a lot of ‘windshield time’ spent driving around collecting data, it would have been prohibitively expensive to add traditional radios to all of them,” explains Polly.

“We also used our RF infrastructure for management of thousands of 250-gallon tanks that support well health, corrosion prevention, and adding treatments to the wells and flowlines to optimize production. These tanks used to be checked with dipsticks, but we recently added ‘smart lids,’ similar to what you might find at Disney World or in smart cities to monitor trash. Our smart lids take level readings, and transmit data over a Long Range Wide Area Network (LoRaWAN) using nodes with 10-year batteries that can transmit over 10 miles in ideal conditions, which is more cost-effective than cellular alone. In the future, we’re going to see if we can use these nodes and LoRaWAN to monitor temperature, pressure, vibration and soil samples.”

One reason LoRaWAN is inexpensive is because up to 2,000 nodes can transmit to a centralized gateway, which enables SJVBU to cover its six fields with just a couple of dozen gateways. Polly reports this transition from legacy radios to LoRaWAN was enabled by an interdisciplinary team that broke down former technical and organizational silos by combining legacy RF know-how with newer application and code development capabilities.

“We needed everyone’s perspective to succeed,” says Polly. “In the end, the economics of these wireless, low-power, WAN opportunities were a no-brainer, so
we just did it. We achieved a significant return on investment (ROI) and reduced drive time, while optimizing the supply chain and using easily deployable technology. In general, LoRaWAN is an order of magnitude less costly than traditional radios and wireless, so we’re looking at other use cases for it, such as routine operating duties.”

FEWER LEVELS, LOWER COSTS
Robert Ward, business development director at MultiTech (www.multitech.com), reports that SJVBU’s wells and equipment are using its MultiConnect wireless nodes, Conduit programmable LoRaWAN and LTE cellular gateways, and LoRa Enterprise Network Server (LENS). Though it started in telecommunications, building automation, healthcare, irrigation and life sciences, MultiTech has been getting into oil and gas applications in recent years because LoRaWAN and cellular connections are generally less costly than other wireless links and provide point access to disparate assets that are miles apart.

“We used to focus on embedded components, such as fax modems and LTE cell modems in devices like defibrillators,” says Ward. “We’re now providing solutions encompassing LTE connectivity from our LoRa gateways, as well as cloud components for device management and data connectivity to clouds like Microsoft Azure and Amazon Web Services (AWS).” Unlike the seven-layer Purdue Open System Interconnect (OSI) model for system and control hierarchies, which requires communications to go through multiple firewalls between field devices, operations, enterprise and public networks, Ward adds that LENS is more secure and efficient because it simplifies these levels and reduces how many firewalls are needed by plugging into this new process data. This is done by connecting a process application and its network at OSI Level 3.5 via a cloud service through potentially one firewall.

“Users like SJVBU are mainly using wireless like LoRaWAN to communicate non-critical information,” adds Ward. “However, even though this is low- or non-consequence data, a lot of good value and better decisions can be gained from it.”

ACCEPTANCE, INVISIBILITY, IIOT
It’s ironic that as soon as innovative technologies like wireless succeed and become accepted, they also start to become transparent—and wireless is no exception, even though it’s largely invisible to begin with. This is a problem because gaining critical mass and going mainstream also means being taken for granted, which can stifle future inquiry and innovation in that area, just as calculators supposedly blunt mathematical thinking skills. This evolution happened to fieldbuses and Ethernet, and now this fade-out is happening to wireless.
This is doubly unfortunate because thoroughly understanding and applying wireless is one of the best ways to multiply the reach and effectiveness of other industrial networks. However, it’s never too late to look under the table, and see—or be reminded—how underpinnings like wireless can improve performance. It always bears repeating that wireless doesn’t just save cable and materials, but it allows signals and data to be gathered that were previously too remote, costly or hard to reach. Of course, this makes wireless a perfect enabler for the Industrial Internet of Things (IIoT) and other digitalization projects that are getting more attention lately.

“Wireless in industrial applications has, for the most part, become well-accepted and even commonplace, which can make it a bit invisible. We typically don’t have to convince users to try it because they already know its benefits. All we have to do is match each wireless technology to its appropriate application,” says Garrett Schmidt, senior product manager for communication interfaces at Phoenix Contact (www.phoenixcontact.com). “This means understanding each application’s individual needs; checking on the approved frequency band for that country; and determining the range that needs to be covered by the wireless network, how many nodes are needed in the network, and what kind of data it’s communicating, such as I/O signals, Ethernet or a fieldbus.

“For example, if a client needs level data from a tank that’s two miles away, then they might not be able to use WirelessHART technology. It’s not meant to cover long distances, but is meant to build a mesh network inside of a plant or within a few hundred square meters. In this case, we can create a simple, point-to-point wireless connection with our Trusted Wireless 900 MHz radios. Increasingly, thought must be put into wireless network designs. Users in operations technology (OT) may want wireless, but they have to make sure it won’t stomp on the corporate network. Since these are usually 2.4 GHz and owned by information technology (IT), OT may need to find wireless products that run at a different frequency, like 900 MHz.”

SITE SURVEYS STILL ESSENTIAL

Despite all the advances wireless has made in recent years, it’s universally agreed that each potential wireless application and location must have a site survey and assessment to identify any unique issues or problems, and determine the most appropriate solution that will best meet its individual needs.

Schmidt adds that wireless is accepted because everyone has WiFi routers at home, but even though users are comfy with it, that’s not enough for it to run reliably in industrial environments. “There are always obstructions and interference that wasn’t expected, whether it’s from steel structures, unrelated microwave signals, vehicles or
“Whatever. This is why it’s still important to do site studies,” says Schmidt. “When we consult with clients, we bring in nodes and gateways, set them up, run them, see how they work, and solve any issues we may find in that particular application. Data types include serial, analog and digital I/O, Modbus, Profibus, Ethernet and others, and they all have different behaviors and requirements. For instance, all Ethernet protocols aren’t created equal: Profinet may have some quality of service (QoS) requirements; while some EtherNet/IP may broadcast a lot of traffic that can clog up a wireless network. Many users want to use IP-based cameras, and send video over wireless, which can hog data and bandwidth. This is where expert support can test devices and integrate them properly.”

“In a site survey, communication tests with actual equipment are carried out at proposed installation points to check the reliability of wireless communication paths,” says Masahito Endo, transmitter division manager, Operational Technology Center, Yokogawa Electric Corp. (www.yokogawa.com). “Yokogawa uses an installation design method called Sky Mesh, in which robust communication paths are secured by using a group of repeaters that communicate with field wireless devices installed in the pipe jungles typically found in process plants (Figure 2). Yokogawa has been successfully applying this method to many plants to create stable wireless networks. When applying Sky Mesh, it’s crucial that communication paths through the repeaters can be fixed. Yokogawa’s field wireless system supports automatic mesh networks and fixed

PATHS WITH REPEATERS

*Figure 2: To design stable wireless networks, Yokogawa uses its Sky Mesh installation design method, which uses repeaters to secure robust communication paths. These repeaters communicate with field devices installed among the pipelines typically found in process plants. Source: Yokogawa*
path networks, allowing Sky Mesh to be applied effectively.”

**IMMEDIATE DATA, SMOOTH OPERATIONS**

Along with bringing in signals from remote and formerly inaccessible locations, wireless typically delivers data much faster, which enables quicker decisions and other operational efficiencies. In the wake of damage from hurricane Irene in 2011 and super-storm Sandy in 2012, engineers in Wallkill, N.Y., knew their water system needed an upgrade. With three pumping stations spread over a 20-mile radius and one of the most complex water supply systems in Orange County, monitoring its water levels was a manual, time-consuming process, while the more recent extreme weather added to existing problems.

“We used to rely on landline telephones, and we had frequent breakdowns and lost communications,” says Tim Grogan, water and sewer administrator, Town of Wallkill (www.townofwallkill.com). “We had to go to each location to find out what was going on. There were discrepancies between different locations. And when we got an alarm, we’d have to go to a location, assess the problem there, and relay information back to whoever needed to fix it.”

Consequently, Wallkill’s engineers worked with Phoenix Contact and Kapsch (www.kapsch.net) to design a synchronized,
wireless network that could deliver more timely data to the utility’s SCADA system, and provide immediate access to water levels and other information. The solution included Phoenix Contact’s TWE with I/O functions, which can transmit data several miles; its FL mGuard routers to serve as firewalls and provide Layer 3 routing for the new system’s cellular modems; and Kapsch’s Dynac traffic and incident reporting software. These devices and software give Wallkill’s staff a monitoring and control platform with historical and event recording, trending and reporting to better oversee their water-supply infrastructure.

“Since we began using Phoenix Contact’s wireless components, we haven’t had the breakdowns, lost communications or discrepancies that we had before,” adds Grogan. “Now, we receive information instantaneously displayed on a centralized SCADA system that we can view on our smart phones, laptops or on a monitor in

**PUNCHLIST FOR WIRELESS**

While the to-do list for any sizeable project is never complete or finished, there are still many common tasks required to design and implement wireless in process applications, multi-plant complexes and widely scattered facilities. Here are most of the high points:

- Recruit in-house, cross-functional advocates, and outside advice and expert assistance;
- Gain enterprise-level, administrative IT and plant-floor buy-in;
- Locate, list and profile all of a site’s existing production equipment and processes;
- Continue site study to identify environmental and artificial impediments to wireless signals;
- Account for each kind of information produced, how it’s delivered, and what new pathways are needed;
- Allow data requirements, equipment and application locations to help dictate the most suitable wireless solution;
- Examine non-wireless networking methods and devices onsite, and learn if they require any connections, interface or adjustments to cooperate with wireless devices being added;
- Once appropriate wireless solutions are selected, research applicable node, gateway, antenna, power delivery, and other equipment and software options;
- Calculate costs and return on investment of wireless solutions;
- Perform site tests of planned wireless components and network design to make sure they’ll fulfill application requirements;
- Deploy wireless devices, monitor performance, and adjust and update as operations needs change, and as better technologies become available.
“We’ve also become more efficient because now we know instantaneously what problems we have and where. All the information is at our fingertips, and within seconds, we know what’s happening and why it’s happening, and we have a team on the way to repair it.”

**BREATH OF LIFE FOR BROWNFIELDS**

Beyond streamlining operations, wireless can also revive legacy equipment and applications, and give many new leases on life—or at least extend their relevance—by delivering data that used to be stranded.

For instance, Empresas Publicas de Medellin (www.epm.com.co), a multi-utility company focused on power generation, transmission and distribution, natural gas distribution, wastewater treatment, aqueducts and telecommunications, recently sought a flexible way to monitor and interact with legacy equipment across its distributed network infrastructure. EPM worked with electrical contractor Facelco (www.facelco.com.co) and Red Lion (www.redlion.net) to enable remote connectivity for power distribution to its reclosers, and implemented remote
connectivity by combining Facelco’s network analyzer with Red Lion’s Sixnet series industrial RTUs and IndustrialPro cellular routers that deliver wireless DNP-3 connectivity to RTUs that act as controllers.

This upgrade let EPM remotely connect, communicate, monitor and control its recloser and power distribution networks, and reduce total cost of ownership (TCO) by extending the life of its infrastructure and preserving capital expenditures for other projects. In addition, Facelco’s network analyzers provide real-time diagnostics that allow EPM to proactively respond to conditions to ensure system integrity and network uptime (Figure 3).

“The primary benefit for EPM was implementing a solution that would extend the life of its existing recloser network,” says Alejandro Arango, engineering director at Facelco. “This remote connectivity solution let them extend the life of existing legacy equipment, which can last 15-20 years, as well as increase control of the network. By working with Red Lion and Facelco, EPM was able to cost-effectively improve network control, maximize availability and protect its network investment.”
MORE PORTS.
MORE FEATURES.
MORE ETHERNET SWITCHES.

UNMANAGED, MANAGED, PoE+

PoE+
- Eliminate the need for separate power cables with Power over ETHERNET+
- 30 W per port
- Wide temperature range

STANDARD
- Robust and versatile to extend application needs
- Alarm contact
- DIP switch configuration
- Up to 16 GIGABIT ports

MANAGED
- Performance and security built-in with Firewall, PoE+, Ring redundancy, VLAN, web-based configuration and monitoring via alarm contact or RS-232

ECO
- Compact and economical for everyday requirements
- Fast and GIGABIT speed
- Wide temperature range

Download our ETHERNET switch application guide:
www.wago.us/switches
It was 2017, and a small team at the recently-merged (HART + Fieldbus Foundation) Field-Comm Group unpacked an old demo from the days when the Fieldbus Foundation’s marketing committee would host workshops around North America. It was a simple, level-cascaded-to-flow, “live” system assembled with donated instruments from the committee’s sponsors.

It included a small DeltaV DCS that had been returned when the demo was mothballed. Resurrected for an overseas conference, the small team was shocked and pleasantly surprised when the demo fired up and controlled level before the DCS showed up. No DCS—it had retained its configuration in the Pelican case for more than two years, including all the function blocks that had been downloaded to the instruments.

Once powered up, the backup Link Active Scheduler (LAS) took over and resumed controlling the little process. You can watch the drama at https://t.co/HGmpb1pZhY—a success that portends a lesson about Foundation fieldbus’ troubles and potential (inevitable?) sunset.

A brief dialog sprang up on the mostly-quiet ISA Controls list serve last month, framed by the question, “Is fieldbus dead?”

Is Foundation fieldbus (FF) dead? It’s a valid question. EPC firms that have some fieldbus expertise are seeing fewer end users who insist on FF for new projects, retrofits or upgrades.

A new dawn for fieldbus?
Foundation fieldbus has fundamental flaws, but they’re not in the technology.

by John Rezabek
Even big oil companies that were vocal FF advocates five years ago are no longer requiring new projects to use it. Though their FF projects were nearly all great successes, rough edges like cumbersome device replacement took its toll on the thinning ranks of FF advocates. So, despite thoughtful standards, training, recruiting and so on, even big companies struggled to maintain a healthy nucleus of fieldbus practitioners. And as the ranks of I&C people have dwindled through headcount cuts, transfers, retirements and other attrition, any appetite for discretionary challenges like fieldbus has become proportionately less.

One fieldbus culture change was especially irritating: it fomented an awkward interplay between DCS (systems) people and instrument (field) people who previously lived in almost separate worlds. Because field devices now participated in control schemes, and had function blocks configured in the DCS and stored in its database, DCS engineers were intruding into the instrument department’s bailiwick. And because field device service often had systems-level impacts, the instrument person’s muddy footprints were appearing in the sacrosanct DCS room.

The remarkable resurrected demo dramatized a unique property of FF many of us didn’t comprehend: except for an operator interface, engineering interface or alarm system, FF was a fully functional distributed control system, operating on its own self-sufficient protocol, network and operating system. What this translates to is, any DCS supplier that sought to support FF was really constructing an interface to a foreign system—FF had its own network with its own rules, function blocks, baud rate, physical layer, etc.

Out of respect for their installed base, most capable suppliers of DCS systems were reluctant to leave legacy systems in the dust to fully integrate this new protocol. Hence, many end users received products with clunky or incomplete support for FF. Early on, it was probably as important to be able to say, “We support FF” as it was to produce a system that seamlessly exploited all its features and capabilities.

Choosing fieldbus remains fundamentally a systems decision. It was envisioned to become an integral part or extension of the DCS, and it needs to be tended by systems-capable individuals, whether they wear steel-toe shoes or sneakers. Meanwhile, end users are just trying to find people who will show up for work.

Will future systems decisions ever turn on FF capability again? That will be a big factor in whether there’s a new dawn for the protocol. FF retains numerous strong selling points for the future—but will we find enough suppliers and end users who know how to sell it?
Introduction to Modbus

Why the grandfather of modern fieldbus is still on the job after 40 years.

by William L. Mostia, Jr., P.E.

MODBUS is a communication protocol for transmitting information between electronic devices over serial lines (original version) or via Ethernet, and is commonly used in process and factory automation. While it’s an open protocol and anybody can use it, “Modbus” is a registered trademark of Schneider Electric USA Inc., which owns the Modicon brand. The Modbus Organization (www.modbus.org) was created to further use of Modbus, and Schneider Electric has been a partner in it. While this article is an introduction to Modbus and its basic functions, the organization’s website has extensive coverage of Modbus, specifications for the various Modbus types, software, testing, interface code and more. The Internet also has available tutorials and specific information on individual device Modbus implementations.

Modbus serial protocol (again, original version) is a master/slave protocol with one master device that controls data transactions with multiple slaves that respond to the master’s requests to read from or write data to the slaves. Modbus TCP, also known as Modbus TCP/IP, uses a client/server architecture. These network architectures are shown Figures 1 and 2.

In a standard Modbus serial network, there is one master and up to 247 slaves, each with a unique slave address. Modbus TCP is typically implemented on an Ethernet network, and data transactions from a Modbus client are directed toward a Modbus server via an IP address.
Modbus comes in several varieties including Serial RTU, Serial ASCII, TCP/IP and UDP/IP. Modbus dialects, such as Enron, Daniel and Pemex Modbus, have arisen due to people modifying standard Modbus to handle floating-point data, long integer data and other data requirements. Reading the Modbus interface and slave documentation is key to understanding and implementing these types of Modbus networks and mixing different manufacturers’ devices in the same Modbus network, which should be carefully done.

**THE ORIGINAL FIELDBUS**

Modbus is the grandfather of modern fieldbuses. Its popularity is due to its simplicity, its openness and ubiquitous nature—it’s used everywhere. It’s withstood the test of time, and is still kicking after almost four decades. Modbus was originally published by Modicon in 1979, primarily for use with its own PLCs. When industrial Ethernet appeared, Modbus TCP was developed, retaining much of Modbus’ simplicity in a TCP/IP wrapper.

Modbus is an application-layer protocol, independent of the data transmission medium. Data transactions are based on the master/client requesting data from or writing data to the slave/server. The data transactions are controlled by the master/client and there is no data-by-exception transmitted in standard Modbus. Data is based on 16-bit registers that can contain discrete on/off or 16-bit integer values. Some implementations use two or more integer registers to represent floating data or long integer values. Diagnostic data can be requested by a Modbus serial master from the slave, and the slave/server can send error codes to the master/client if they perceive there is something wrong with the
request they received. Modbus data transactions only contain a function code, register addresses and data, and it is up to the master/client and the slave/server to make sense of the data.

There are two types of serial Modbus, RTU and ASCII. RTU and ASCII transmission modes determine the way in which Modbus messages are encoded. In Modbus RTU, bytes are sent consecutively with no space in between them, with a 3½-character space between messages as a delimiter. This allows Modbus interface software to know when a new message is starting. For each eight-bit byte, one start bit, eight data bits, one bit for parity, and one stop bit are sent, for a total of 11 bits per byte. Each Modbus RTU message is terminated with an error checksum called a cyclic redundancy check (CRC).

Modbus ASCII marks the start of each message with an ASCII colon character “:” and the end of each message is terminated with ASCII carriage return/line feed (CR/LF) characters. This allows the spacing between bytes in the message to be variable, which makes it suitable for transmission through some modems. The data in a Mod-
bus ASCII message uses ASCII characters. For each eight-bit byte, one start bit, seven data bits, one bit for parity, and one stop bit are sent, for a total of 10 bits. Modbus ASCII messages are terminated with an error checksum called a longitudinal redundancy check (LRC).

The tradeoff between the two types is that Modbus ASCII is easier to read if you look at the message, but the RTU messages are smaller, which allows more data exchange in an identical time period. All devices on a Modbus serial link must be of the same type, either RTU or ASCII. Modbus RTU is by far the more common.

Modbus TCP or TCP/IP is basically Modbus RTU wrapped in an Ethernet (IEEE 802.3) package with the destination address as an IP address using the TCP/IP transaction protocol. In addition, TCP port 502 is reserved for Modbus, while the new Modbus/TCP Security uses Port 802. For more information, visit “Modbus Messaging on TCP/IP Implementation Guide,” V1.0b (www.modbus.org).

ADDRESSING AND MESSAGING

Modbus memory addressing is generally organized around 16-bit registers that contain 16 coils or on/off (0/1) states or integer values in 16-bit registers (input/output or holding registers). While some devices will use their own Modbus addressing, typical Modbus addressing can be seen in Figure 3. Modbus messaging is based on what’s called an application data unit (ADU) and a protocol data unit (PDU). The Modbus message includes the slave/server address for the slave/server involved, a function code, data start addresses, and the data being sent to (written) or to be sent back (read) to the master/client, with an error checksum at the end (CRC/LRC/Checksum).

The size of the serial Modbus PDU is limited by the size constraint that was inherited from the first Modbus serial network implementation of 256 bytes. Modbus slave addresses are limited to 1-255. Addresses 1-247 are available to the user and addresses 248-255 are reserved.

A typical Modbus serial data transaction is also shown in Figure 3. The Modbus TCP data transactions are essentially the same except the server address is an IP address, there is some Ethernet overhead, and the error checksum is different. Modbus data can include starting data addresses, data quantity or count, and actual data that’s read or is to be written. If the Modbus slave/server has a problem with the master/client request, the slave/server will issue an error response back to the master/client.

In the Modbus TCP/IP message format, the Modbus PDU is typically wrapped into the Ethernet package, and consists of the Modbus function code and the Modbus data request. The slave address and error code
(CRC) are typically not needed as the Modbus TCP/IP packet is routed by the network to the desired IP address (unless there's to be a connection into a serial network), and the error check is done as part of the Ethernet packet. See the “Modbus Messaging on TCP/IP Implementation Guide,” V1.0b (www.modbus.org) for further details.

Modbus data transactions are also function code-based, which tells the Modbus slave/server what type of data transaction is taking place. Function codes can be divided into public codes, user codes and reserved codes. Public function codes are well-defined and guaranteed to be unique and validated by the Modbus.org community. User function codes can be implemented, and aren’t supported by the Modbus specification. There's no assurance that the user function code will be unique. Reserved function codes are function codes currently
used by some companies for legacy products, and aren’t available for public use. Refer to the “Modbus Application Protocol Specification,” V1.1b3 (www.modbus.org) for more information on function codes.

**SERIAL WIRING AND SPEED**

Modbus serial traditionally uses RS-232 and RS-485 wiring for collecting data from Modbus slaves. The original Modbus was RS-232-based, but the need to connect up multiple slaves soon expanded Modbus to use multidrop wiring methods. RS-422 has been used but RS-485 is more capable and is more commonly used. A Modbus master can communicate with up to 247 Modbus slaves (limited by the RS-422/485 driver capability), which can be PLCs, smart devices, DCS and remote telemetry units (RTUs), which is where Modbus RTU got its name. “Modbus over Serial Line Specification and Implementation Guide” (www.modbus.org) provides more information on wiring Modbus serial links.

To connect more than two devices and distances greater than 50 feet, a network of multidrop devices, RS-485 or RS-422 should be used. For multidrop networks, RS-485 is by far most popular and supports up to 32 nodes without repeaters over the range of up to 4,000 ft (1,200 m).

The speed that Modbus messages are sent is known as the baud rate (or bits per second) and all devices on the network must use the same baud rate, typically 9,600-19,200 Baud, but network speeds up to 115 kb/s are not unknown. Generally, the higher transmission speed, the shorter the cable and the more important to have the correct network end termination resistors to minimize reflections (equal to resistance component of the cable characteristic impedance).

**SAFETY AND CYBERSECURITY**

In control systems such as a DCS, any incorrect data can potentially lead to a safety incident, whether or not a safety instrumented system (SIS) is involved. So, we must be concerned with data integrity of the Modbus data transactions—the correct data, error-free, getting to and from where it’s supposed to go. Standard Modbus has error-detection capabilities such as parity, CRC/LRC and checksums, as well as diagnostics from the slave/server or master/client interfaces. These protections should be adequate for normal data transactions, but not against cybersecurity attacks or internal security breaches.

Modbus is commonly used to read data and status from SIS for the SIS HMI as required by the SIS standard IEC 61511, which is typically on the basic process control system (BPCS), commonly a DCS. Writing to the SIS via Modbus is less common and should be done cautiously. Failure of the Modbus communication link between the BPCS and the SIS should not compromise the safety integrity of the safety functions.
in the SIS. This also means there must be provisions to protect against writing bad data from the BPCS or from other sources into the SIS. This same logic should apply to any independent protection layer (IPL) that uses Modbus.

Modbus was designed to read and write data and not for safety-critical functionality. The use of Modbus for safety-critical actions should be risk-assessed, and if it’s used, protections should be put in place to ensure the safety integrity of the Modbus data that performs a safety-critical function. Use of communication links like Modbus in SIS is covered by IEC 61511-1. Essentially, failure of the Modbus link can’t be able to affect the SIS safety integrity. Potential failures include failures caused by a security breach that affect the Modbus data and the SIS safety integrity. ISA technical reports on fieldbus (TR84.00.07) and cybersecurity (TR84.00.09) provide additional guidance.

If the control system and networks comply to IEC 62443, the exposure of the Modbus networks to cybersecurity attacks is reduced. Using the new Modbus/TCP Security protocol can further reduce the cybersecurity exposures of your Modbus TCP data transactions. See “Modbus/TCP Security Protocol Specification, 2018” (www.modbus.org).

By its hardwired nature, Modbus serial has less cybersecurity exposure than Ethernet-based client/server communications using Modbus TCP, especially when Ethernet is connected to a larger process control network, which can be connected to the corporate enterprise network and/or to the Internet. Modbus serial can be exposed to a cybersecurity attack on the control system side of the master or if it’s connected to a Modbus TCP network. Using Ethernet-based Modbus TCP without additional protection can greatly increase the odds that a security breach may be possible.

Modbus TCP’s cybersecurity exposure can be further reduced by using security appliances or firewalls specifically designed for Modbus with deep-packet inspection, such as are available from Tofino Security (www.tofinosecurity.com). A security appliance/Modbus firewall should always be placed between a Modbus TCP network and a Modbus serial network, and between any remote access into a Modbus Ethernet network. (Here, the term “Modbus firewall” means a firewall or security appliance specifically designed for Modbus data transactions that has deep packet inspection, whitelisting and other security features related to Modbus.) Figure 4 shows a simple network example for Modbus TCP/Serial with SIS systems and broken down into zones and conduits per IEC 62443.

There are many different possible network configurations, and only the Modbus portion of the zones and conduits are shown in this example.
MODBUS TROUBLESHOOTING

Troubleshooting Modbus can be broken into two general pieces, Modbus TCP/IP and Modbus Serial. Troubleshooting Modbus TCP/IP can be broken down into network troubleshooting, such as making sure that the Modbus TCP/IP Ethernet packets with error-free data get to and from where they’re supposed to go to perform desired transactions. Troubleshooting Ethernet is typically an IT function, but may be done by other personnel if industrial Ethernet is used. Once inside the target device, troubleshooting the data transaction is typically the same as in a Modbus serial device and the troubleshooting is similar.

The first question to ask when troubleshooting a Modbus link is—unless the problem is new—did the Modbus link work in the past or it has never worked? If it worked before for a Modbus TCP link, this
means the Ethernet and device configurations are probably OK.

The same applies to a Modbus serial link—Modbus type, RTU/ASCII, baud rate, start characters, parity, end characters, timing between messages, termination resistors in place, etc. are probably correct. This allows the troubleshooter to concentrate on what’s changed on the link, such as new device(s), modified or changed wiring, new wiring near existing wiring, etc. If the installation is a new one, the wiring and the Modbus configuration for all the Modbus devices on the link should be double-checked, and made sure they line up with the Modbus link manual and documentation. A common problem for Modbus serial is to have signal wires reversed.

The most common indications of Modbus problems are an I/O timeout (a message request was sent and a reply wasn’t received before the interface timed out), or an error code was received from the slave/server device. This would commonly show up as a bad process variable PV or an indication of an I/O timeout on the DCS or HMI. Most Modbus devices have communication lights that blink when the device is communicating, and some will use colored LEDs to provide some troubleshooting capability.

When a Modbus slave/server has detected a problem with a request, it will respond to the master/client with an exception code. The meanings of the exception codes can be found in “Modbus Application Protocol Specification,” V1.1b3 (www.modbus.org). It may not be possible to determine what Modbus exception code is causing the problem from the DCS, and you may have to look at the actual Modbus transaction. There are free and paid software packages available that will allow you see and generate Modbus messages for testing purposes. Once you get past the wiring and configuration problems, it’s almost always necessary to look at the actual Modbus data transaction to see what the problem is.

If you’re placing a different manufacturer’s hardware on a Modbus Link and having a problem, it probably revolves around the Modbus master interface software, and reading the device manual is a must. If you’re reading or writing data but not to the right memory locations, the interface memory addressing may not agree with the slave addressing, or the data types don’t agree. You may also be having problems with a memory offset—the original addressing in the Modbus message started at zero, while the memory location used in the PLC was started at one, such as Coil 1 has a message memory location of 0 and holding register address of zero in the Modbus message has a memory location of 40001. This offset is retained in many modern Modbus interfaces and devices. Also, Modbus uses big-endian data flow for addresses and data (the most significant data byte sent first, lower signifi-
If you are experiencing noise or intermittent or strange problems with a Modbus serial link, the problem is likely related to grounding, incorrect shielding, or power wiring next to Modbus wiring.

cant byte second), but some Modbus interfaces for vendor-specific equipment may use little-endian addresses (least significant data byte sent first, most significant byte second). Using programs that allow you to see and send messages to a slave can allow you see addressing, function codes, data sent, and what error codes are being returned from what slave/server. Modbus 07 and 08 function codes and sub-codes can be used to get additional diagnostic information on serial links. See “Modbus over Serial Line Specification and Implementation Guide,” V1.02 (www.modbus.org).

If you’re experiencing noise or intermittent or strange problems with a Modbus serial link, the problem is likely related to grounding, incorrect shielding, or power wiring next to Modbus wiring. RS-485 two-wire and four-wire require a common line grounded at one point only in addition to the two or four wires.

In summary, using Modbus in process control systems and factory automation is common primarily due to its simplicity and its availability. Modbus serial is based on a master/slave architecture, while Modbus TCP is client/server-based. Modbus’ primary function is to read or write data between the master/client and slaves/servers. Modbus data is based on 16-bit registers and it is up to the Modbus master/server or slave/client to understand what the data is and what it relates to.

Modbus does not have any inherent protections against inadvertent or malicious assaults on its data transactions from cybersecurity attacks, and requires additional protections. The use of Modbus for safety-critical actions should be risk-assessed, and if used in safety-critical applications, protections should be put in place to ensure the safety integrity of the Modbus data that performs a safety-critical function.

Frequent contributor William L. (Bill) Mostia, Jr., P.E., principle engineer, WLM Engineering Co., can be reached at wlmostia@msn.com.
Message queuing telemetry transport (MQTT), developed in 1999, is a publish/subscribe message lightweight protocol based on TCP that is now the most commonly used messaging protocol above HTTP. The reference architecture is very simple, and is based on client/server. The client is generally a sensor that “publishes” the information to the server (broker) that receives the information and dispatches it to the subscribers. MQTT protocol uses a many-to-many paradigm, and the broker decouples the publisher to the subscriber and acts as a message router with every message a discrete chunk of data, opaque to the broker. MQTT’s publisher/subscriber model enables clients to communicate one-to-one, one-to-many and many-to-one.

Every message is published to an address, known as a topic. Clients may subscribe to multiple topics. Every client subscribed to a topic receives every message published to the topic. The MQTT specification doesn’t dictate any particular Topic Namespace, nor does it dictate any particular payload data encoding, though MQTT topics are hierarchical, like a filing system (e.g. sales volume/flow/corrected). Wildcards are allowed when registering a subscription (but not when publishing), thus allowing whole hierarchies to be observed by clients.

MQTT supports three quality of service levels: “fire and forget,” “delivered at least once,” and “delivered exactly once.” To prevent excess traffic, MQTT clients can register a custom “last will and testament” message, sent by the broker when a device knowingly disconnects.
MQTT also has support for persistent messages stored on the broker. When publishing messages, clients may request that the broker preserve the message. Only the most recent persistent message is stored. When a client subscribes to a topic, any persistent message will be sent to the client. However, unlike a message queue, MQTT brokers don’t allow persistent messages to back up inside the server.

MQTT brokers can require username and password authentication from clients to connect, and to ensure privacy, the TCP connection may be encrypted with SSL/TLS.

As a machine-to-machine (M2M)-oriented protocol, MQTT is designed to be lightweight, and it has two drawbacks for very constrained devices:

• Every MQTT client must support TCP and, therefore, always holds a connection open to the broker, which can be a problem in high-packet-loss/computing environment.
• MQTT topic names are often long strings, which make them impractical for 802.15.4 industrial wireless environments.

These shortcomings are addressed by MQTT-SN, which defines a UDP mapping of MQTT and adds broker support for indexing topic names.

MQTT is an OSI Application Layer (Layer 7) like HTTP, and as a carrier, requires tagging such as HTML or XML for web pages to represent the data. MQTT for SCADA applications have Sparkplug, which provides an open specification for how edge-of-network (EoN) gateways or native MQTT-enabled end devices and MQTT applications communicate bidirectionally.

The Sparkplug specification defines an MQTT Topic Namespace, payload, and session state management applicable to the SCADA/IoT market. To meet these requirements, the specification is based on a lightweight, bandwidth-efficient, low-latency payload encoding architecture.

The above standards are all available and open, with Sparkplug from Cirrus Link Solutions, while Eclipse Foundation (www.eclipse.org) has released an open-source implementation of MQTT called Mosquitto (https://mosquitto.org) while Oasis maintains advanced message queue protocol (AMQP) at www.amqp.org. Do they define a fully open solution? Not quite. We mere humans still need to interact with the data, which typically requires an HMI though several hardware and interface suppliers to support these protocols.

I’ve also heard rumors of an end user-driven group similar to OPAF being discussed to support an Open SCADA standard.
Expanding field networks

IEEE P802.3cg will define single-pair Ethernet with power for harsh environments.

by Ian Verhappen

Ethernet and IP-based systems have had limitations that prevent them from being used as field sensor networks for moving all the data packets around a facility. Now one of those limitations—the power-to-the-field hurdle—is about to be overcome. The IEEE P802.3cg 10 Mb/s Single Pair Ethernet Task Force expects to release a new standard describing power over Ethernet (PoE) twisted-pair wiring option within the year. This new standard is specifically designed for harsh environments, and will support both short-reach (15 m/50 ft.), high-noise installations (automobiles) and long hauls of up to 1,000 m (3,280 ft.) such as plant environments, while being able to survive certain fault conditions.

The long-haul, point-to-point configuration will support a trunk up to 1,000 m with 200-m spurs and a maximum of 12 connectors. Echo cancellation to 300 m is incorporated to provide increased reliability along the spurs. To ensure suitability for industrial and automotive environments, the standard also supports optional physical layer powering techniques, including addition of power injection partway along the network.

The estimated power dissipation is less than 50 mW, which means that with the addition of protective diodes, a 1.0 V peak-to-peak signal will be compatible with intrinsically safe (IS) operation. Additional energy management features include support for both quiet and standby modes. Target bit error rates (BER) are in the range of 10-9 with support for message failure mechanisms to detect when a packet fails to reach its destination.
Key variables proposed for the IS implementation include maximum output voltage $U_o$ 17.5 V; maximum allowed supply voltage 14.3 V; functional supply voltage $13.2 \pm 0.3$ V; short circuit current $I_o$ 380 mA; minimum output voltage 9.6 V; minimum device voltage 9.0 V; maximum device current 55.6 mA; and maximum available device power 500 mW (www.ieee802.org/3/10SPE/public/Nov2016/index.html).

These values are quite different from the normal 24 V for process instrumentation, 12 V for CAN bus (automotive), and 40 V for PoE in the IT environment. However, with 0.5 W/55 mA at the remote end of the network (a typical transmitter today is less than 20 mA and closer to 10 mA), it should be possible to power three or four devices per link, which coincidentally was what the original Foundation fieldbus IS spur was capable of supporting.

One challenge with any new system is the same problem faced by the various process fieldbus systems: the installed base of field devices. However, because the average home-run cable in a plant is approximately 300 m (1,000 ft.), especially with the trend towards using distributed interface rooms, it should be feasible to use this technology with existing infrastructure.

I can envision distributed I/O options for various protocols, similar to the connector blocks we used to replace terminals when migrating to Foundation fieldbus in 1996, that reuse the existing multiconductor cables for the home-run signals. ODVA is watching these developments closely, as are, I'm sure, many other protocol trade organizations.

The new IEEE 802.3cq standard makes it foreseeable that the death of the traditional field junction box will soon be upon us. With 10 Mb bandwidth for arguably four signals, there will not be any challenge in accessing the full range of device diagnostics or similar capabilities for an installed HART or fieldbus device, while also making it possible to use true Ethernet devices with access to faster updates or raw data for further analysis and understanding of the process than is possible today. Of course, Ethernet to the field device goes the multi-drop capability. However, for some applications (perhaps vibration or other high-frequency, high-value signals), this may be just what’s needed.

This is sure starting to sound like the Open Process Automation Forum’s concept of the distributed control node (DCN) without the local regulatory control capability—but that’s just software that could be added to the field junction block I/O connector microprocessor.