PLC HANDBOOK

Practical Guide to Programmable Logic Controllers

from **AUTOMATIONDIRECT**.com

PLC Handbook

TABLE OF CONTENTS

What is a PLC

Programmable Logic Controllers (PLC) are often defined as miniature industrial computers that contain hardware and software used to perform control functions. More specifically, a PLC would be used for the automation of industrial electromechanical processes, such as control of machinery on factory assembly lines, amusement rides, or food processing. They are designed for multiple arrangements of digital and analog inputs and outputs with extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. A PLC will consist of two basic sections: the central processing unit (CPU) and the Input/Output (I/O) interface system.
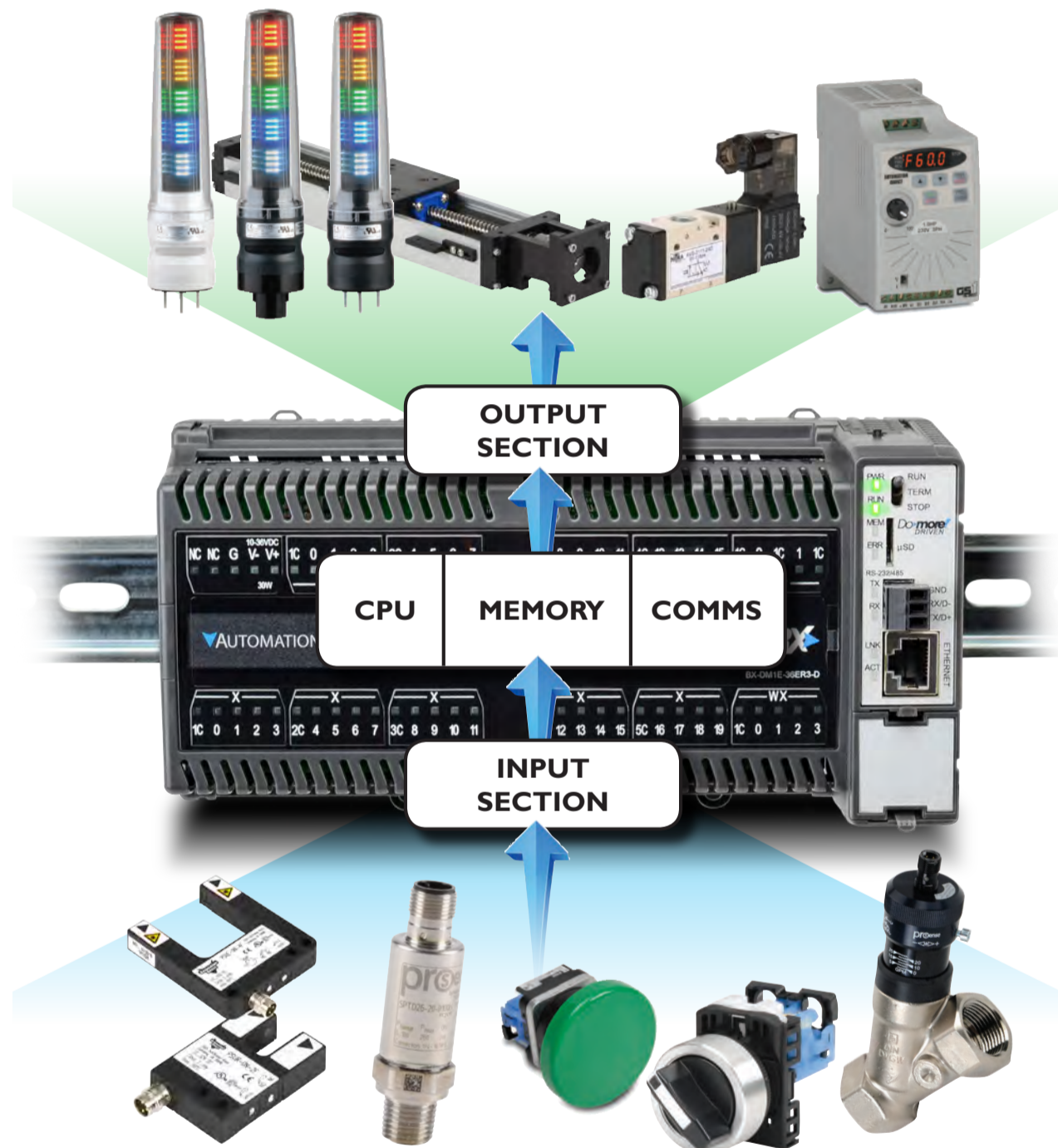

Figure 1A: Programmable Controller

The CPU controls all system activity primarily through its processor and memory system. The CPU consists of a microprocessor, memory chip and other integrated circuits to control logic, monitoring and communications. The CPU has different operating modes. In programming mode the CPU will accept changes to the downloaded logic from a PC. When the CPU is placed in run mode it will execute the program and operate the process. Input data from connected field devices (e.g., switches, sensors, etc.) is processed, and then the CPU "executes" or performs the control program that has been stored in its memory system. Since a PLC is a dedicated controller it will process this one program over and over again. The time it takes for one cycle through the program is called scan time and happens very quickly (in the range of 1/1000th of a second, depending on your program). The memory in the CPU stores the program while also holding the status of the I/O and providing a means to store values.

The input/output system is physically connected to field devices and provides the interface between the CPU and its information providers (inputs) and controllable devices (outputs). After the CPU processes the input data (input scan), it will then make any needed output changes after executing the user program (output scan). There are four basic steps in the operation of all PLCs: Input Scan, Program Scan, Output Scan, and Housekeeping. These steps continually take place in a repeating loop.

- Input Scan – Detects the state of all input devices that are connected to the PLC
- Program Scan – Executes the user created program logic
- Output Scan – Energizes or de-energizes all output devices that are connected to the PLC
- Housekeeping – Includes communicating with programming devices and performing internal diagnostics

Typical PLCs have a wide range of I/O modules available to accommodate all kinds of sensors and output devices. For example, discrete input modules can be used to detect object presence or events with devices such as proximity or photoelectric sensors, limit switches and pushbuttons. Discrete output modules can control "ON/OFF" loads such as motors, lights, and solenoid valves. Analog input modules can accept signals from process instrumentation such as flow, pressure, temperature and level transmitters. These modules can interpret the signal and present a value within a range determined by the devices' electrical specifications. Analog outputs will command loads that require a varying control signal, such as panel meters, variable frequency drives or analog flow valves. Many PLCs also offer specialized modules such as high-speed I/O or motion control, and serial or Ethernet communications.

The greatest benefit of automating with a Programmable Logic Controller is the ability to repeat or change and replicate the operation or process while collecting and communicating vital information. Those making the buying decisions for Programmable Controller applications can have very different needs. Cost, power, speed, and communication are a few of the many considerations when choosing the right PLC for the job.

Learn More...Watch The Video!
What is a PLC?

http://n2adc.com/96lzw

History of the PLC

The PLC or Programmable Logic Controller has revolutionized the automation industry. Today PLCs can be found in everything from factory equipment to vending machines, but prior to New Year's Day 1968 the programmable controller didn't even exist. Instead what existed was a unique set of challenges that needed a solution. In order to understand the history of the PLC we must first take some time to understand the problems that existed before programmable controllers.

Before the Programmable Controller

Before the days of the PLC the only way to control machinery was through the use of relays. Relays work by utilizing a coil that, when energized, creates a magnetic force to effectively pull a switch to the ON or OFF position. When the relay is de-energized, the switch releases and returns the device to its standard ON or OFF position. So, for example, if I wanted to control whether a motor was ON or OFF, I could attach a relay between the power source and the motor. Then I can control whether the motor is getting power by either energizing or de-energizing the relay. Without power, of course, the motor would not run, thus I am controlling the motor. This type of relay is known as a power relay. There could be several motors in one factory that need to be controlled, so what do you do? You add lots of power relays. So factories started to amass electrical cabinets full of power relays. But wait, what switches the coil in the power relays ON and OFF before the power relay turns the motor ON, and what if I want to control that? What do you do? More relays. These relays are known as control relays because they control the relays that control the switch that turns the motor ON and OFF. I could keep going, but I think you get the picture of how machines were controlled pre-PLC, and, more importantly, I think you start to see some of the problems with this system of electromechanical control via relays.
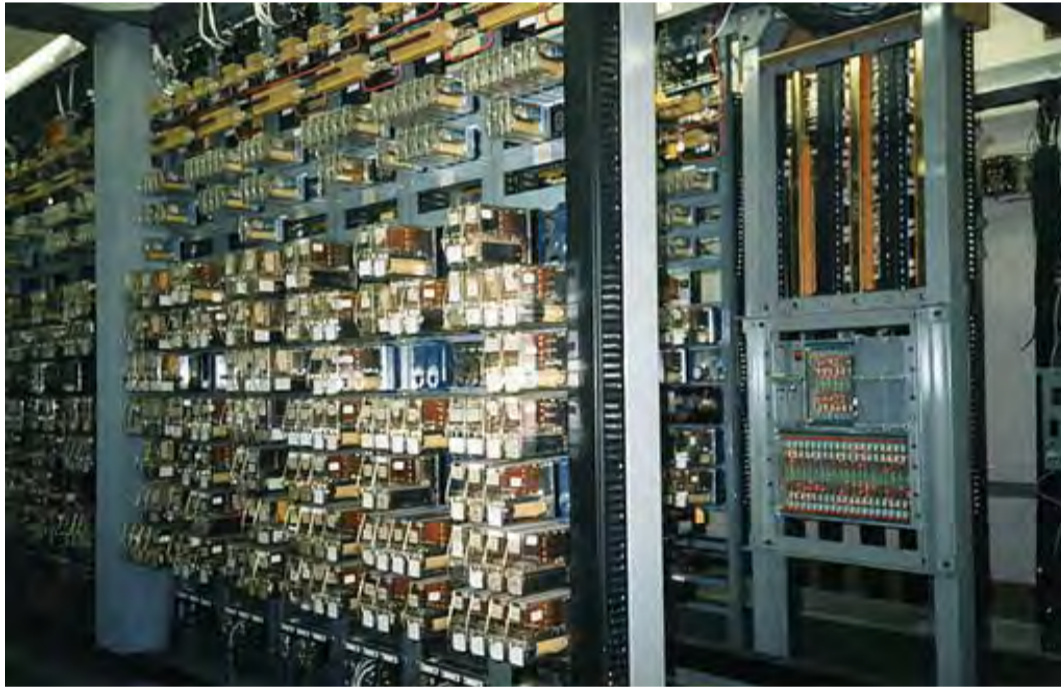

Figure 2A: Courtesy of Signalhead via Wikimedia Commons

The Problem with Relays

Think about modern factories, and how many motors and ON/OFF power switches you would need to control just one machine. Then add on all the control relays you need and what you get is… Yes, machine control, but you also get a logistical nightmare. All these relays had to be hardwired in a very specific order for the machine to work properly, and heaven forbid if one relay would have an issue, the system as a whole would not work. Troubleshooting would take hours, and because coils would fail and contacts would wear out, there was need for lots of troubleshooting. These machines had to follow a strict maintenance schedule and they took up a lot of space. Then what if you wanted to change something? You would basically have to redo the entire system. It soon became clear that there were problems installing and maintaining these large relay control systems.

Let's hear from a controls designer in the thick of things in the early '70s

"Upon graduating from technical college in 1970, I began working as a controls designer, automating metal working machinery and equipment with industrial relays, pneumatic plunger timers, and electro-mechanical counters. Also included were fuses, control transformers, motor starters, overload relays, pushbuttons, selector switches, limit switches, rotary drum sequencers, pilot lights, solenoid valves, etc.

The relay based control systems I created included anywhere from 50 to well over 100 relays. The electrical enclosures to house the controls would typically be six feet wide by four feet high, mounted near the machinery. Picture lots of wires bundled and laced together, connecting the relays, timers, counters, terminals, and other components, all nice and tidy. Then picture after a few months or years the same wiring, after many engineering changes and troubleshooting, being out of the wire duct or unlaced; in many cases wires were added in a crisscross point to point pattern to take the shortest route and amount of time to make the change. We referred to the condition of these control enclosures as a rat's nest; reliability suffered, along with an increase in difficulty during troubleshooting, or making additional operational engineering changes."

Tom, Controls Designer

Birth of the PLC Solution

So what was the solution? I am sure this is the exact question that engineers at the Hydra-Matic division of General Motors were struggling with every day. Fortunately, at that time, the concept of computer control had started to make its way into conversations at large corporations such as GM. According to Dick Morley, the undisputed father of the PLC, "The programmable controller was detailed on New Year's Day, 1968."

The popular forum **PLCDEV.com** outlines a list of requirements that GM engineers put out for a "standard machine controller." It is this request that Dick Morley and his company, Bedford and Associates, were responding to when the first PLC was envisioned. Besides replacing the relay system, the requirements listed by GM for this controller included:

1. A solid-state system that was flexible like a computer but priced competitively with a like kind relay logic system.
2. Easily maintained and programmed in line with the already accepted relay ladder logic way of doing things.
3. It had to work in an industrial environment with all its dirt, moisture, electromagnetism and vibration.
4. It had to be modular in form to allow for easy exchange of components and expandability.

Figure 2B: Image of Dick Morley
Courtesy of AutomationWorld.com

The programming look of the PLC required that it be easily understood and used by maintenance electricians and plant engineers. As relay-based control systems evolved and became more complicated, the use of physical component location wiring diagrams also evolved into the relay logic being shown in a ladder fashion. The control power hot wire would be the left rail, with the control power neutral as the right rail. The various relay contacts, pushbuttons, selector switches, limit switches, relay coils, motor starter coils, solenoid valves, etc., shown in their logical order would form the ladder's rungs. It was requested that the PLC be programmed in this Ladder Logic fashion.

As Dick Morley laments in his memoirs , the process from idea to actual controller wasn't all smooth sailing.

"The initial machine, which was never delivered, only had 125 words of memory, and speed was not a criteria as mentioned earlier. You can imagine what happened! First, we immediately ran out of memory, and second, the machine was much too slow to perform any function anywhere near the relay response time. Relay response times exist on the order of 1/60th of a second, and the topology formed by many cabinets full of relays transformed to code is significantly more than 125 words. We expanded the memory to 1K and thence to 4K. At 4K, it stood the test of time for quite a while."

Tom, our controls designer, recounts, "My experience in creating relay-based control systems, at that time, put me in the perfect position to be one of the first control system designers to use some of the very first programmable controllers to replace relay-based control systems. My first experience with a PLC happened to be with one of Bedford Associates competitor's solid state devices. The unit was programmed with a suitcase-sized programming device that required setting the instruction type and line address and then pressing a button to burn a fuse link open in a memory chip to set the logic path. Once the programming was completed and tested, the PLC was able to perform the machine cycle operation in a very reliable manner. Unfortunately the PLC card rack was open in the rear with a mixture of 24 VDC and 120 VAC power and signals. It didn't take much for an electrician checking signals during troubleshooting to accidentally short the 120 VAC to the 24 VDC and take out the entire PLC system. Being the first use of a PLC in a large corporation, the failure doomed the use of PLCs at this manufacturing facility for a couple of years."

Eventually Dick Morely spun off a new company named Modicon and started to sell those first PLCs, the Modicon 084 (named because it was prototype #84). It was the Modicon 084 that was presented to GM to meet its criteria for its "standard machine controller." Modicon started to sell the 084 with very limited success. As Dick Morley puts it, "Our sales in the first four years were abysmal." But nevertheless the company continued to learn and develop. Eventually, Modicon would bring to life the controller that would change the industry forever, the Modicon 184. Dick Morley writes this about the 184:

"The thing that made the Modicon Company and the programmable controller really take off was not the 084, but the 184. The 184 was done in design cycle by Michael Greenberg, one of the best engineers I have ever met. He, and Lee Rousseau, president and marketer, came up with a specification and a design that revolutionized the automation business. They built the 184 over the objections of yours truly. I was a purist and felt that all those bells and whistles and stuff weren't "pure", and somehow they were contaminating my "glorious design". Dead wrong again, Morley! They were specifically right on! The 184 was a walloping success, and it—not the 084, not the invention of the programmable controller—but a product designed to meet the needs of the marketplace and the customer, called the 184, took off and made Modicon and the programmable controller the company and industry it is today."

Figure 2C: Image courtesy of RepairZone.com

The PLC in its teenage years

The first PLCs had the ability to work with input and output signals, relay coil/contact internal logic, timers and counters. Timers and counters made use of word size internal registers, so it wasn't too long before simple four-function math became available. The PLC continued to evolve with the addition of one-shots, analog input and output signals, enhanced timers and counters, floating point math, drum sequencers and mathematic functions. Having built-in PID (Proportional-Integral-Derivative) functionality was a huge advantage for PLCs being used in the process industry. Common sets of instructions evolved into fill-in-the-blank data boxes that have made programming more efficient. The ability to use meaningful Tag Names in place of non-descriptive labels has allowed the end user to more clearly define their application, and the ability to import/export the Tag Names to other devices eliminates errors that result when entering information into each device by hand.

As the functionality of the PLC evolved, programming devices and communications also saw rapid growth. The first programming devices were dedicated, but unfortunately the size of suitcases. Later, hand-held programming devices came into the picture, but soon were replaced with proprietary programming software running on a personal computer. AutomationDirect's DirectSOFT, developed by Host Engineering, was the first Windows-based PLC programming software package. Having a PC communicating with a PLC provided the ability to not only program, but also allowed easier testing and troubleshooting. Communications started with the MODBUS protocol using RS-232 serial communications. The addition of various automation protocols communicating over RS-485, DeviceNet, Profibus, and other serial communication architectures have followed. The use of serial communications and the various PLC protocols also allowed PLCs to be networked with other PLCs, motor drives, and human to machine interfaces (HMI). Most recently Ethernet and protocols such as EtherNet/IP (for Industrial Protocol) have gained tremendous popularity.

Sources

http://www.barn.org/FILES/historyofplc.html

http://www.machine-information-systems.com/PLC_History.html

http://www.plcs.net/chapters/history2.htm

http://www.controldesign.com/articles/2005/264/

http://www.plcdev.com/plc_timeline, http://www.plcdev.com/the_birth_of_the_plc

How to Choose a Controller

Choosing the most effective controller for your application depends on a number of factors. The worksheet below serves as a checklist of things to consider when determining programmable controller requirements. It lists the most important areas to consider when choosing a system, as well as provides space for recording determinations of your system needs. **To print a copy, here is the PDF version of the Worksheet for Choosing a Controller.**

Step 1:

Consideration	Information to Record	
1. Proposed System	_____ New system	_____ Existing system

Determine whether your system is new or existing: Will your system be installed from scratch or are there existing products already installed? The rest of your system will need to be compatible with new components.

Why this is important: Certain controller products may not be compatible with others. Making sure your existing products are compatible with any new products you are researching will save you time and money. Check appropriate entry.

Step 2:

Consideration	Information to Record	
2. Environmental Issues	_____ Codes/ environmental issues to consider	_____ No codes or environmental issues to consider

Consider any environmental issues that will affect your application (temperature, dust, vibration, codes specific to your facility, etc.).

Why this is important: Certain environments may affect the operation of a controller. For example, typical controllers have an operating temperature of 0-55 degrees Celsius (32-130 degrees F). If your application will include any extreme environmental conditions, or you have specific codes at your facility that must be met, you will need to either research products that meet those specifications or design the installation to meet requirements. Check appropriate entry.

Step 3:

Consideration	Information to Record	
3. Discrete Devices	____ Total inputs ____ AC ____ DC	____ Total outputs ____ AC ____ DC

Determine how many discrete devices your system will have. Which types (AC, DC, etc.) are needed?

Why this is important: The number and type of devices your system will include is directly linked to the amount of I/O that will be necessary for your system. You will need to choose a controller that supports your I/O count requirements and has modules that support your signal types. Enter quantities and type based on corresponding field devices.

Step 4:

Consideration	Information to Record	
4. Analog Devices	____ Total inputs ____ Voltage ____ Current ____ Thermo ____ RTD	____ Total outputs ____ Voltage ____ Current

Determine how many analog devices your system will have. Which types (voltage, current, temperature, etc.) are needed?

Why this is important: The number and type of devices your system will include is directly linked to the amount of I/O that will be necessary for your system. You will need to choose a controller that supports your I/O count requirements and has modules that support your signal types. Enter quantities and type based on corresponding field devices.

Step 5:

Consideration	Information to Record
5. Specialty Modules or Features (application-specific)	____ High speed counter ____ Positioning ____ Servo/stepper ____ Real-time clock ____ Others (list)

Determine whether your system will require any specialty features: Will your application require high-speed counting or positioning? What about a real-time clock or other specialty feature?

Why this is important: Specialty functions are not necessarily available in a controller CPU or in standard I/O modules. Understanding the special functions your system may perform will help you determine whether or not you will need to purchase additional specialty modules. Check all features required.

Step 6:

Consideration	Information to Record
6. CPU Required	**Hardware requirements:** _____ K program memory required (estimated) _____ K data memory required (estimated) _____ Fast scan time required? _____ Battery backup required? **Software/special function requirements:** _____ PID _____ Floating Point Math _____ Bult-in Data Logging Others (**see Programming section below**)

Determine the type of CPU you will need: How much memory will your system require? How many devices will your system have (determines data memory)? How large is your program, and what types of instructions will your program include (determines program memory)? How fast a scan time do you need?

Why this is important: Data memory refers to the amount of memory needed for dynamic data manipulation and storage in the system. For example, counter and timer instructions typically use data memory to store setpoints, current values, and other internal flags. If the application requires historical data retention, such as measured device values over a long period of time, the size of the data tables required may determine the CPU model you choose. Program memory is the amount of memory needed to store the sequence of program instructions that have been selected to perform the application. Each type of instruction requires a specific amount of program memory, typically defined in a programming manual. Applications that are basically sequential in nature can rely on the I/O device rule of thumb to estimate program memory (five words of memory for each I/O device); complex applications will be more difficult to judge.

If scan time is important in your application, consider the CPU processor speed as well as instruction execution speed. Some CPUs are faster at boolean logic but slower with data handling instructions. If special functions such as PID are required, the CPU you select may make those functions easier to perform. Check or calculate all requirements that apply.

For program memory required, follow this rule of thumb:
5 words of program memory for each discrete device
and 25 words for each analog device.

Step 7:

Determine where your I/O will be located: Will your system require only local I/O, or both local and remote I/O locations?

Why this is important: If subsystems will be needed at long distances from the CPU, you will need a controller that supports remote I/O. You will also have to determine if the remote distances and speeds supported will be adequate for your application. Serial and Ethernet-based I/O hardware are two typical choices available for most systems. This I/O may also be referred to as distributed I/O, and may require a particular protocol, such as Modbus.

Consideration	Information to Record	
7. I/O Locations	_____ Local only	_____ Remote Locations Specific remote I/O protocol required? Which one?

Enter number of physical locations needed, and if/what specific protocol may be required.

Step 8:

Determine your communication requirements: Will your system be communicating to other networks, systems or field devices?

Why this is important: Communication ports (other than the programming port) are not always included with a controller. Knowing your system communication requirements will help you choose a CPU that supports your communication requirements, or additional communication modules if necessary. Check any/all communications functions required.

Consideration	Information to Record
8. Communications	_____ Ethernet _____ PLC to PLC _____ Modbus RTU _____ ASCII (interface to serial devices) _____ Other

Step 9:

Determine your programming requirements: Does your application require only traditional programming instructions, or are special instructions necessary?

Why this is important: Certain controllers may not support every type of instruction. You will need to choose a model that supports all instructions that you may need for a specific application. For example, built-in PID functions are much easier to use than writing your own code to perform closed-loop process control. Typical instructions such as timers, counters, etc., are available in most controllers; note any other special instructions required here. Check any/all programming functions required.

Consideration	Information to Record	
9. Programming	_____ Floating point math _____ Drum sequencer	_____ PID loops _____ Number of loops needed _____ Subroutines _____ Direct interrupts _____ Others (list)

Need help deciding which of our PLCs is right for you?

Check out our interactive PLC Selector **click here**, or **watch** a quick tutorial on how it can be used to put you on the right path to the right PLC.

PLC Hardware

What controls you? No, I'm not talking about your boss, your spouse or the government. Instead, think of your body as an industrial control system. Now, what controls you? Hopefully, you said your brain, although the heart is an acceptable answer sometimes. But for now, let's focus on the brain and the nervous system as well. Your body receives signals from your senses or nerves that travel through the spinal cord to the brain. Well, a PLC system operates the same way, with a brain (the CPU), a spine (the base or backplane), and the senses (I/O modules). Now, let's take a closer look at each and discuss functions, options and considerations.

The Base-ics

We'll start with the spine. In a PLC system, you have different options when it comes to connecting I/O modules to the CPU. Some systems have a fixed style where the CPU comes with a fixed set of I/O points already installed and dedicated connections already determined.

Other PLC systems have a stackable style where the CPU and I/O modules are separate, but come with connectors built in that are used to attach the components together. These connections form one continuous data bus throughout the system. This internal data bus is often referred to as a backplane.

Another option with PLC hardware is the modular base configuration. With this style of PLC, a separate base unit that holds the modular components is needed. Each module will seat into a slot on the base unit that is used to connect the module to the backplane already installed in the base. These bases come with different numbers of slots and some with power supplies built in. Typically, the first two leftmost slots are dedicated for the power supply and CPU.

One question to keep in mind with internal PLC connections is, "How easy are they to change?" In other words, after all of the modules are installed, how easy is it to get them out? With the fixed style, you are just that, fixed, so no help there. Stackable PLC components can be changed, but you have to break the backplane connection for all the modules downstream of the module you wish to remove. That could be an issue. The modular base configuration is by far the easiest to change. Simply slide the module out and slide the new one in. Some of these PLCs even offer hot swapping, which means that the module change can be done with the PLC powered and with no interruption to the control process.

The Ins and Outs

Now, let's look at the senses, the information providers, or, in the PLC's case, the I/O modules. The I/O modules, and their respective end devices, allow the PLC to know and affect the current state of the process being controlled. There are many types of input and output modules available, but they can all be classified as analog, discrete or specialty I/O.

Discrete I/O is the simplest of the bunch and provides the PLC with ON/OFF control. Used with both AC and DC voltage ranges, they provide the CPU with a yes/no, true/false indication and allow simple full ON or full OFF responses. On the input side, you would use discrete I/O for simple questions like, "Is the box there?", "Is the tank full?", "Can I start this motor?" These input signals are provided by devices such as photoeyes, proximity switches, E-stop pushbuttons, float switches, etc. For discrete outputs, your command choices are either ON or OFF with nothing in between and are commonly used for stack lights, alarms, relays, solenoids, etc.

One thing to consider with discrete I/O is whether you need a sink, source or relay configuration. With sinking inputs/outputs, the PLC will provide the reference voltage (typically 0V) when completing the circuit. Sourcing inputs/outputs are the opposite and the PLC will provide the source voltage, be it 12VDC, 24VDC, 240VAC, etc. Relay types don't provide either. They function just as a relay contact would, using an external source and connecting it to a load once activated. You can read more about sink/source concepts in **this blog**.
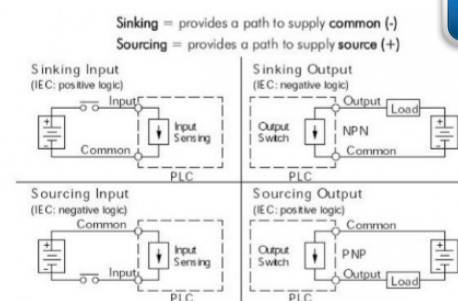
Sinking and Sourcing Concepts

Make the right choice the first time when selecting the type of I/O points for your application by reviewing these sinking and sourcing concepts.

To learn more **click here**

Next up is analog. Analog I/O deals with the gray area between full ON and full OFF that discrete I/O ignores. It provides the PLC with the data it needs for precision control of a process. If you want to know the exact liquid level in a tank or you want to open a valve 1/3 of the way, analog is the answer. Analog signals come in a variety of ranges including: 0-20mA, 4-20mA, 0-10V, etc. RTD and thermocouple modules are two analog modules that specialize in converting low voltage signals from temperature probes into usable data. One important factor to remember with analog modules is the resolution they provide. The higher the resolution, the higher the accuracy of the input measurement or output response. To read more about discrete and analog I/O, refer to **this blog**.

Serial Communication Module

High-speed Counter module
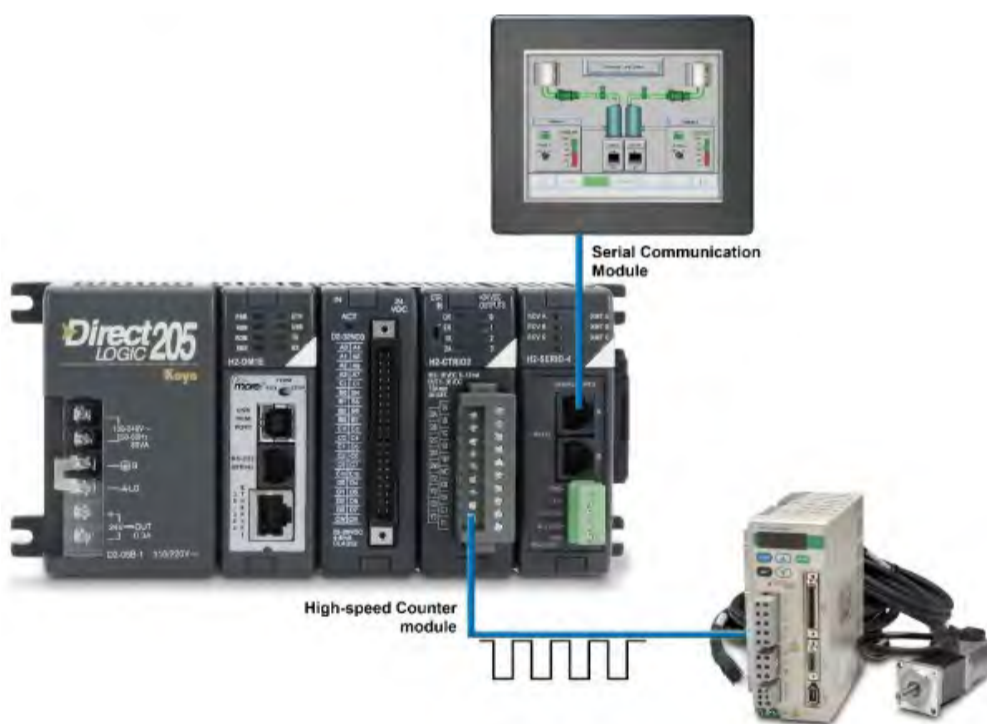
That leaves us with specialty I/O. This type of I/O includes special functions like high-speed and communication. High-speed modules are needed when the input/output data is comprised of high-frequency pulses. These modules can track input data, such as encoder signals, independent of th CPU scan, guaranteeing a more accurate pulse count. And high-speed outputs can provide precision control with stepper motors used in motion or positioning applications. Communication modules provide additional communication ports/protocols that a system may require: RS232, RS485, Ethernet, etc. They can also allow a local PLC system to be expanded remotely, if the need arises.
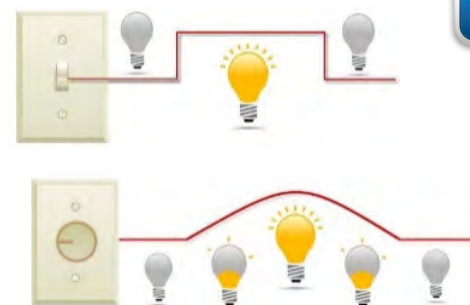
Understanding Discrete and Analog I/O

Shed some light on the differences between discrete and analog I/O for your PLC applications

To learn more **click here**

PLC Power

Now before we touch on the brain of the system, I'd like to detour a bit and discuss the Power Supply. As mentioned earlier, the PLC power supply can be included with the base or it can be a separate unit. They come in a variety of voltage ranges, including 12-24VDC and 110/220 VAC, and supply a limited amount of amperage. It is very important that the power supply chosen is capable of supplying the CPU and I/O modules with the power they need. To ensure this, perform a power budget analysis and calculate the required current for your application. An example is shown below.

Power Budget

When determining the types and quantity of I/O modules you will be using, it is important to remember there is a defined amount of power available from the base power supply.

1. Using a chart similar to the one below, fill in column 2.

2. Enter the current supplied and used by each device (columns 3 and 4). Devices which fall into the "Other" category (Row D) are devices such as the operator interface which also has power requirements, but do not directly plug into the base.

3. Add the current used by the system devices (columns 3 and 4) starting with the CPU slot and put the total in the row labeled "Maximum Current Required" (Row E).

4. Subtract the row labeled "Maximum Current Required" (Row E), from the row labeled "Current Supplied" (Row B). Place the difference in the row labeled "Remaining Current Available" (Row F).

5. If "Maximum Current Required" is greater than "Current Supplied" in either column 3 or 4, the power budget will be exceeded. It will be unsafe to use this configuration, and you will need to restructure your I/O configuration. Note the auxiliary power supply does not need to supply all the external power. If you need more than the 300mA supplied, you can add an external 24V power supply. This will help keep you within your power budget for external power.

A	Column 1	Column 2	Column 3	Column 4
		Device Type	5VDC (mA)	External Power 24 VDC (mA)
B	CURRENT SUPPLIED			
	Base	9 slot	2,600	300
C	CURRENT REQUIRED			
	CPU SLOT	H2-DM1E	275	0
	SLOT 0	D2-16ND3-2	100	0
	SLOT 1	D2-16ND3-2	100	0
	SLOT 2	D2-16NA	100	0
	SLOT 3	D2-08NA-1	50	0
	SLOT 4	D2-16TD1-2	200	80
	SLOT 5	D2-08TA	250	0
	SLOT 6	D2-08TA	250	0
	SLOT 7			
D	OTHER			
	Operator interface	EA1-S3ML	90	0
E	Maximum Current Required		1415	80
F	Remaining Current Available		2600-1415=1185	300-80=220

The CPU

So we have the connections made and the external data, now we need to know what to do with it. That is the job of the CPU (central processing unit). The CPU contains a microprocessor, memory storage and other integrated circuits that are used to execute the control program, store logic data, and communicate to external devices. So what should the CPU hardware include? Well, for starters, the CPU will need some way to communicate and this is typically done using Ethernet, serial or USB communication ports. Serial ports are important since many existing networks use these standards but Ethernet has become the favored method to communicate in today's industrial applications. USB is a recent addition and is extremely useful when connecting to a CPU to program or monitor its logic. Just as important as the ports available are the protocols the CPU can support: EtherNet/IP, Modbus TCP, etc... Some PLCs use proprietary communication and others use open standards. Either way, make sure the CPU you select has the communication capabilities you need. For a detailed look into PLC communication protocols, see the topic titled **"PLC Communications - Coming of Age".**

Another important aspect of CPU hardware is the memory size. Your CPU needs enough storage space to handle the amount of tasks you are going to assign and it doesn't hurt to have a little extra for future needs. CPU memory has expanded with the times and some CPUs today can have 50MB or more of memory available to the user. The large memory capacity in these CPUs allows for almost limitless programming, ample space for program documentation and impressively quick scan times. Removable memory card support is another feature that is popular today. Adding an additional 1 to 32GB of memory storage, these memory cards allow for extensive data logging and easy program downloads, without a PC. Other hardware related options available for CPUs include a battery backup, built-in I/O and status indicators/OLED message displays.

One Last Thing…

Now that we have discussed the individual hardware components of a PLC, there are a few other hardware considerations I'd like to mention for the PLC as a whole.

• **Real Estate –** size can be an issue with PLC installations and if it is a concern, look into compact or slim form factor designs. Also look for high I/O density. Check the maximum number of points per module and modules per base. More I/O points per module equals less modules and therefore more space.

• **Wiring Options –** screw type and spring clamp connectors are most common with I/O modules. Wiring connection systems like the **ZipLinks** wiring solution may also be available.

• **Mounting Options –** See what options are available and which you require: DIN rail, panel mount or both.

• **Environmental Certifications –** make sure the hardware you are using is certified for the environment it will reside in. For instance, hazardous locations may require extra precautions.

• **Quality Control Certifications –** see what kind of testing was done to verify the hardware's durability. Extreme temperature, electric shock and vibration testing are a few examples. To learn more about product certification and testing check out the following section on **Industrial Certifications**. By the way, if you have ever wondered what vibration testing was all about, wonder no more. See the Productivity2000's vibration test in action below or **click here**:
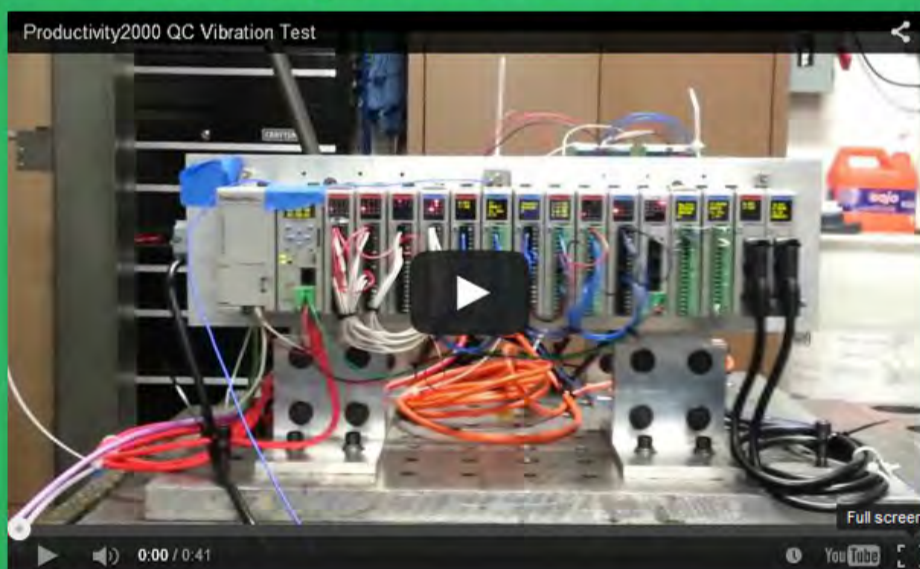
Industrial Certifications

What are they and why are they important?

True story: a guy buys new LED light fixtures and hangs them in his garage workshop. He's really impressed with the light intensity, but soon discovers they create so much radio-frequency interference (RFI) that the only music his radio can play is a loud buzz. The lights are imported, and in retrospect, he concludes their extremely low cost might be the result of cutting a few corners on the construction. So what other potential problems may lurk? Should he be worried about some inadequately-sized component in the power supply overheating? Could they become a shock or fire hazard?

These days we really don't concern ourselves with such questions all that often. Our homes and plants are full of electronic equipment, and under normal circumstances we don't worry about a refrigerator, computer or electric motor causing a fire. We take it for granted that these products are built well and tested to ensure safe performance because most of them are (Figure 4A). At least this was the working assumption, but things are changing for the worse with some components, particularly for industrial use.

Figure 4A: Quality-built home and industrial electronic devices have marks indicating they have passed a variety of tests. The UL mark shown here indicates extensive evaluation and testing.

For the last 20 years and more, the bulk of the electrical components used in industrial facilities has come from companies that are known for their reliability and quality, much of which is assured through careful product design and thorough testing. Users might assume this is the case for all electrical components. Unfortunately, it isn't.

Some high-quality products are counterfeited outright. This is a serious problem, but a topic for a different article. Here, we want to discuss what happens when inferior products are selected either unknowingly or as a means to reduce cost. Looking at the larger world of manu- facturing these days, this type of situation is happening more often for a couple reasons: the growing interest in devices from the "maker" community, and the growing ease of purchasing inexpensive electronic components from no-name suppliers.

Understanding Quality Differentiators

So why do our friend's lights mess up his radio? It could be a lack of shielding combined with a down-and-dirty circuit design using fast switching and no filtration, all done to save cost.

There are probably better light fixtures available at a higher price capable of operating with acceptable levels of RFI. They work because the manufacturer understands how these side effects must be minimized in the design. They are tested according to standards issued by the U.S. Federal Communications Commission (FCC) to ensure they do not emit excessive RFI, and they carry a mark to show they have passed the testing. But not all products get that kind of testing because it is expensive to perform and requires more sophisticated circuit design.

For someone working on a basic industrial project and needing a standard electronic automation device, say a ¼-DIN temperature loop controller, there are many potential suppliers with a wide range of prices. Some searching on eBay can turn up un-branded units which cost barely one-quarter the price of even a high-value supplier, with other name-brand units offered at far higher prices.

We'll assume the un-branded unit works, but it will likely have few if any marks showing it has passed any certifications or tests. The customer in this case has no assurance the unit has ever been checked for any type of manufacturing quality outside of powering-up correctly. The more expensive choices will undoubtedly come with an assortment of testing marks such as UL, CE, CSA and possibly others (Table 4A), determined by the country of origin and intended markets.

UL	**Underwriter's Laboratories**
FM	**Factory Mutual**
ANSI	**American National Standards Institute**
ISO	**International Organization for Standards**
CE	**Conformité Européene**
FCC	**U.S. Federal Communications Commission**
VDE	**Verband der Elektrotechnik**
TÜV	**Technischer Übervachungsverein**
IEC	**International Electrotechnical Commission**
NFPA	**National Fire Protection Association**
CSA	**Canadian Standards Association**
DIN	**Deutsches Institut für Normung**
ABS	**American Bureau of Shipping**
CCC	**China Compulsory Certificate**
CGMP	**Current Good Manufacturing Practices, U.S. FDA**
ATEX	**Atmosphères Explosibles EU Directive**
Baseefa	**British Approval Service for Electrical Equipment in Flammable Atmospheres**

Table 4A: Standards issuers and testing organizations used for industrial electronic components and devices

Similarly, someone wanting to use "maker" devices in a project has to be careful as to the source of the products. For example, the basic Arduino UNO, if made by the Arduino company, is provided with the CE (Conformité Européene) and FCC marks. Additionally, it is ROHS compliant. These together designate it meets the most basic requirements for levels of potentially hazardous chemical content such as lead and cadmium necessary for legal sale in the EU countries. The FCC mark indicates it doesn't cause excessive amounts of RFI.

Those are fine as far as they go, but they don't mean nearly as much as the kind of analysis and testing done by a nationally recognized testing laboratory (NRTL) such as UL. Moreover, it only applies to products actually made by Arduino. The bigger problem is an equivalent board can be purchased from multiple manufacturers because the UNO design is open source. Anyone can make and sell a functionally-equivalent board (Figure 4B). It may operate exactly as it should, but it may not have any certifications at all. Let the buyer beware.
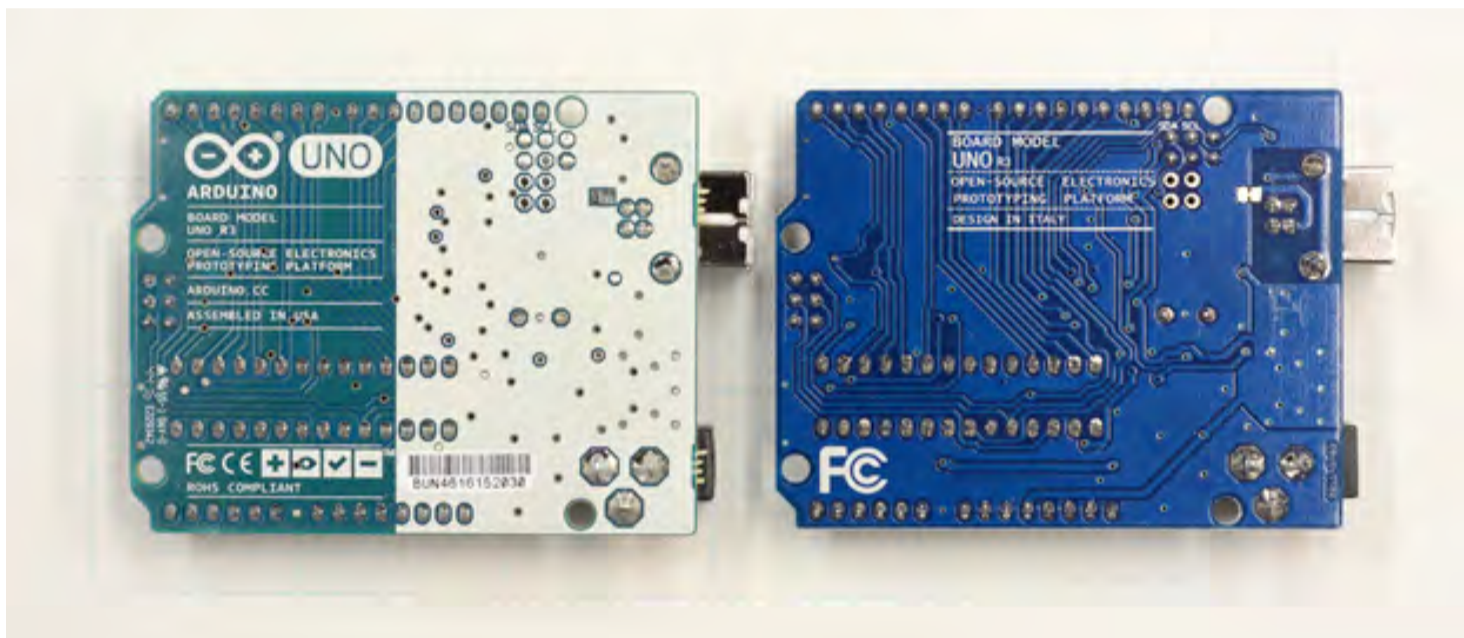
Figure 4B: Two functionally equivalent "Uno" boards, one from Arduino (left) and the other a clone, do not carry the same certifications.

Why Testing Matters

Product testing and certifications do not create quality, but they ensure a product is designed and manufactured with quality as a primary objective. Testing always requires using some standard as the yardstick (See Table 4A). For example, making sure a PLC can operate without getting excessively hot can be examined under IEC 61131-2. It covers more performance areas and goes into much greater detail, but basically it defines circumstances of normal operation, the potential amount of time involved, and what temperature qualifies as excessive. The more complex the standard, the more complex the testing.

Standards may be imposed externally or internally. Manufacturing a product so it can carry the UL mark means it has to be tested to the relevant UL standard, verified by UL. In other cases, another lab certified by the standard-issuing organization can perform the test, but it can't be done by the manufacturer. On the other hand, a manufacturer may perform its own tests. It can follow an existing standard, such as a MIL-STD, and say that the testing has been carried out in conformance with the standard.

If an existing standard is not available or doesn't pertain to the product as desired, a company can impose its own standard on a product. For example, let's say it wants to advertise a new device designed for use in cold climates, able to perform when it is -40 degrees F. The manufacturer may hire an external testing lab to carry out the test, but it may do it internally since it is an in-house standard. It will write its own procedure saying the unit must be placed in a freezer at -42 degrees F for 24 hours before powering up, etc. The objective is figuring out a way to prove the point being claimed in a repeatable and documentable manner. Prospective purchasers can decide if they accept its validity or not.

What Standards Include

Standards go well beyond simply describing testing procedures. Testing is certainly part of the process, but it goes much deeper to include how the product is designed and manufactured. A new product going through initial evaluation by an NRTL like UL will need to be studied in great detail. The first step is deciding which standards might apply for the type of product and its applications. Once established, the examiners begin looking for a variety of characteristics, including but not limited to:

• Components must be appropriate for the application and procured from reliable suppliers. UL maintains lists of recognized suppliers of individual components considered acceptable. Those not on the list have to be tested.

• The circuit board and internal component layout (Figure 4C) have to be analyzed to ensure conductors and board traces are wide enough for expected current levels, components must have sufficient space between them, internal spaces must be sufficient to allow heat dissipation, etc.
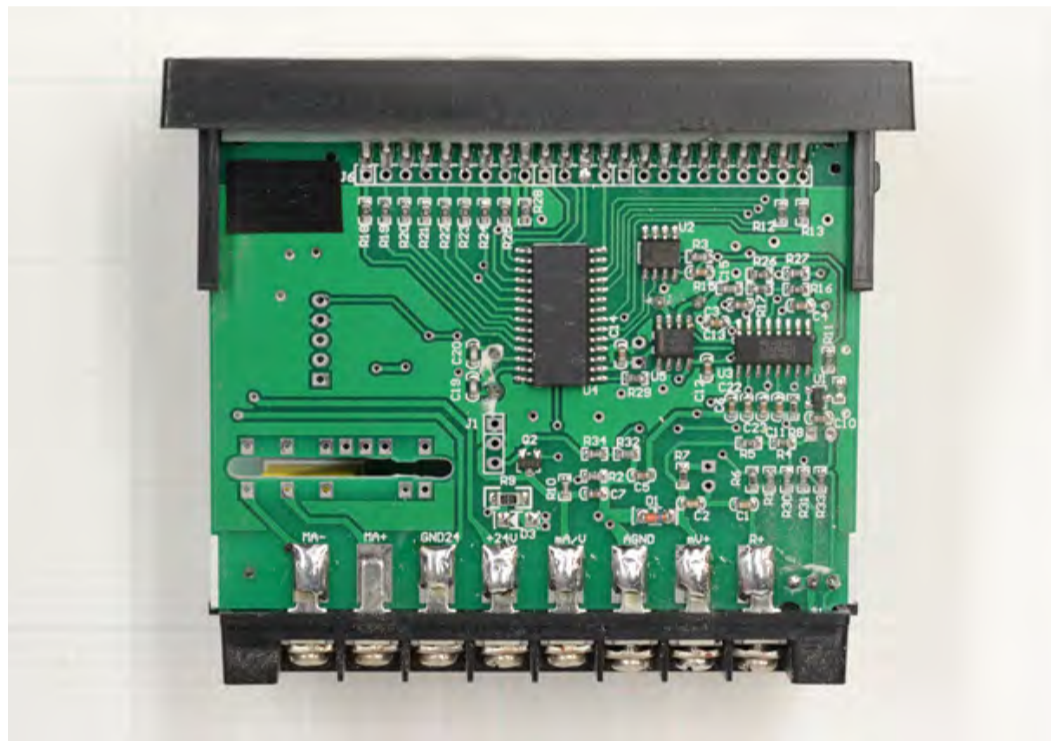
Figure 4C: The internals of an unbranded, uncertified temperature display. It carries no marks at all, so there is no assurance it was designed well, uses quality components, or was assembled properly. For example, the MA+ connection (second from left) does not appear to be soldered, and the adjacent connections are hand soldered and inconsistent.

• The power supply must be stable and capable of delivering sufficient current for all operating conditions without creating excessive RFI or heat.

• Design and construction files detailing the thought processes used for overall configuration, component selection and assembly must be examined to verify correct thinking and analysis.

The list could go on at great length. Suffice it to say there are many elements typically entering into the discussion which are dictated by the nature of the product and its applications.

The actual testing process pushes the product to any limits spelled out in the standard. If the standard requires it to survive a drop of 15 feet to a concrete floor, it will be dropped 15 feet to a concrete floor. If a switch has to be capable of surviving 50,000 cycles, it will be switched 50,000 times. Testing labs take these things very literally.

How Much Testing is Enough?

Most people who explore this topic in any depth are astonished by the number of testing organizations around the world and the mind boggling number of standards. These stem from the days before globalization when most industrial countries had their own locally manufactured products. Circuit breakers made in Germany were used there as well and their own NRTL networks grew up to service those producers and users, and the same happened in most other industrially advanced countries.

Nowadays, products can be produced anywhere and used anywhere, but the national concept of NRTLs persists. To a large extent, a user in Germany still has to have products tested and certified locally, but this is beginning to break down. For example, the CE mark in Europe applies to all the EU countries. Similarly, UL can do one test that applies equally in Canada and the U.S., but these are still exceptions to the larger regional rules.

Consequently, an automation product manufacturer has to launch a testing program for every major country. This creates a lot of duplicated effort, but at least most countries have relatively similar testing criteria. There are differences from country to country and some are stricter than others, but a truly well-designed and manufactured product can pass most examinations. Nonetheless, paying for all that testing can run up costs in six figures and delay introduction of a product for months.

To further complicate matters, NRTLs usually insist on testing actual production models of a product rather than prototypes. This means a manufacturer has to gear up and produce months before they can have the necessary marks applied. This also adds cost.

The number of certifications required by a given customer will depend on a variety of factors. Some are regionally based and required by law. Specific types of plants, particularly those with hazardous environments such as oil refining or chemical manufacturing, may have requirements outlined by a regulatory agency or industry group. Similarly, an insurance provider can impose minimum requirements needed to keep policies in force.

A Means, Not an End

Product designers who know their efforts will be dissected under a microscope by a dispassionate evaluator realize they can't cut corners. Everything has to be executed correctly at every phase of the process. Test and certification are simply a verification of the correct process. Passing the test is important, but not as important as a desire to create and build quality products.

Host Engineering: Design Verification Testing-Functional and Firmware Testing Example

All performance specifications must be verified. Additional tests must be done to verify conformance at operating temperature limits (See Temperature Cycling). Actual product usage conditions should be achieved or simulated as closely as possible. All function modes and configurations must be verified (i.e. every specification in the product manual should be verified by type testing). Interfaces and compatibility with all allowable system components must be tested.

Continue on next page.

Continued...Host Engineering: Design Verification Testing-Functional and Firmware Testing Example

Climatic Verification

Test limits are defined by IEC 61131-2, Programmable Controllers - Part 2: Equipment Requirements and Tests. This standard takes precedence over any test limit stated in the following:

- Temperature Cycling—Product must be monitored for continual conformance to specifications during several cycles from minimum to maximum allowable operating temperature.

- High and Low Temperature Storage—Product must conform to specifications after 4 days exposure to minimum and maximum non-operating temperatures. This test is intended to verify that the product is unaffected by worst case temperatures during shipment.

- Thermal Shock—Operating product is rapidly transferred between minimum to maximum temperatures. This test verifies withstand of worst case thermal stress.

- Humidity—Product must be monitored for continual conformance to specifications during 95% relative humidity at maximum operating temperature. Power cycling 4 hours ON 4 hours OFF is recommended.

- Moisture Resistance—Non-operating product is subjected to 10-day test similar MIL-STD-202F, 98% non-condensing relative humidity. Test is intended to check corrosion susceptibility of components.

- Vibration—Operating product is monitored for conformance during random vibration levels per NAVMAT P9492 Navy Manufacturing Screening Program. Likewise, the product must be immune to sinusoidal vibrations at the 2.0 g acceleration levels over the 20 Hz to 20 KHz range.

- Impact Shock—Operating product is monitored for conformance during a free fall rapid deceleration. This test checks for the mechanical robustness of the product.

- Drop Test—Non-packaged non-operating product is subjected to a one-inch drop onto a hard table surface. This is to verify mechanical robustness to mishandling during installation.

- Packaged Drop Test—In its shipping configuration the product is dropped from a height of 48 inches onto each surface, three edges, and one corner. This test verifies the pack material to sufficiently protect the product from transit damage.

Electromagnetic Verification

Test Limits are defined by IEC 61131-2, Programmable controllers - Part 2: Equipment requirements and tests. This standard takes precedence over any test limit stated in the following:

- Electrostatic Discharge—Product must be monitored for continual conformance to specifications during discharges over the range of 1 to 15kV.

- Conducted Susceptibility—Product must be monitored for conformance during electrical pulse disruptions to all wiring points as appropriate. Applicable referencing standards are IEEE 472 and MIL-STD-461 CS01, CS02, and CS06. These tests simulate power line lightning strikes, power line surges/brownouts, coupled noise from other industrial equipment, etc.

Radiated Susceptibility

Product must be monitored for conformance while immersed in an electric field of strength 10 V/m through the frequency range of 200 KHz to 1 GHz.

Emissions

Product must be monitored during normal operation for its generation of electrical noise in excess of the applicable environmental allowances. This is to confirm that the product will not adversely affect the operation of other electrical equipment in its nearby vicinity.

PLC Software

One of the biggest factors in any PLC buying decision is the capability of the programming software. How easy is it to use? What does it offer? PLC hardware is only half of the equation, and who wants to buy a controller only to find out the software is lacking the basics? Been there, done that! So what should you expect or want in a PLC programming software?

Since each user has his/her own needs and preferences when programming a PLC, that's a hard question to answer. However, here are some thoughts on features and tools that we believe are helpful and that you'll want to consider in your next PLC software package.

Built-in Simulator

A built-in simulator is extremely useful in PLC project development. The simulator provides a virtual PLC that you can use to test your logic without downloading to an actual PLC. A good simulator will emulate analog and discrete I/O and provide access to logic elements such as timers, counters, control bits, etc.
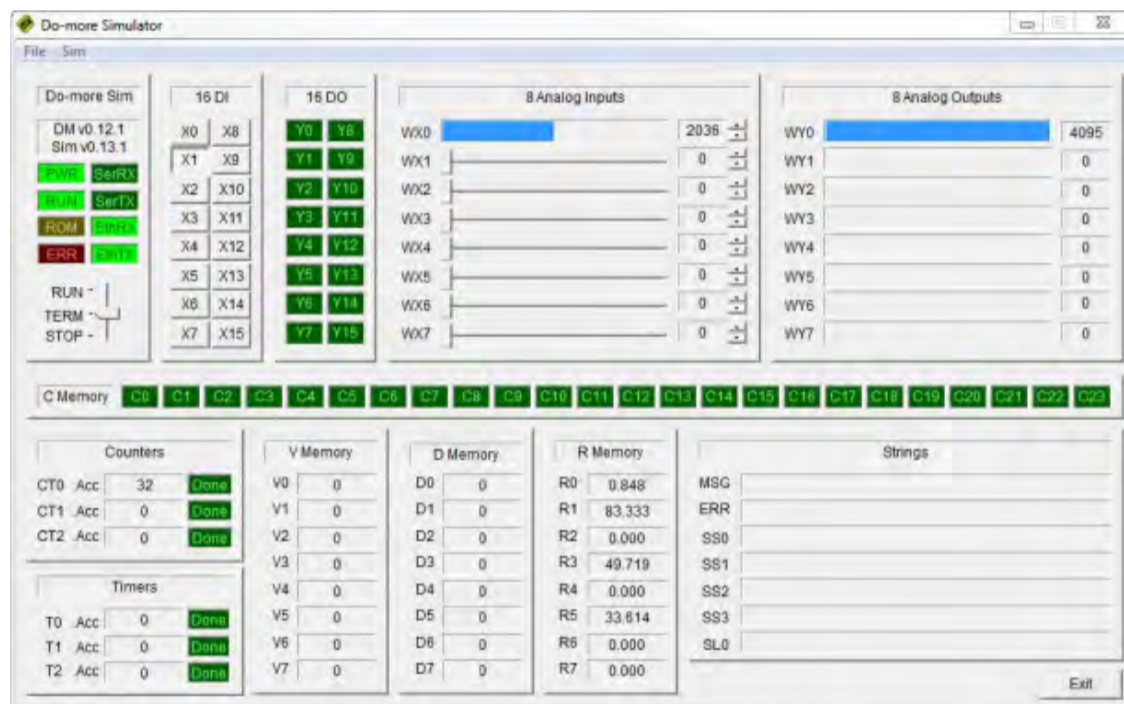
Hot Swapping and Run Time Transfers

Once you have your project up and running, system modifications can become much harder. Imagine having to schedule a shutdown at your plant in order to replace an I/O module or change a rung of code. If your software allows for hot swapping and run time edits, then don't worry. Hot swapping means replacing system modules while the system is HOT or powered, and run time transfers allow the user to transfer project edits into the CPU without stopping the CPU scan. These two software features are important because they prevent costly shutdowns and production losses whenever hardware and/or software changes are needed.
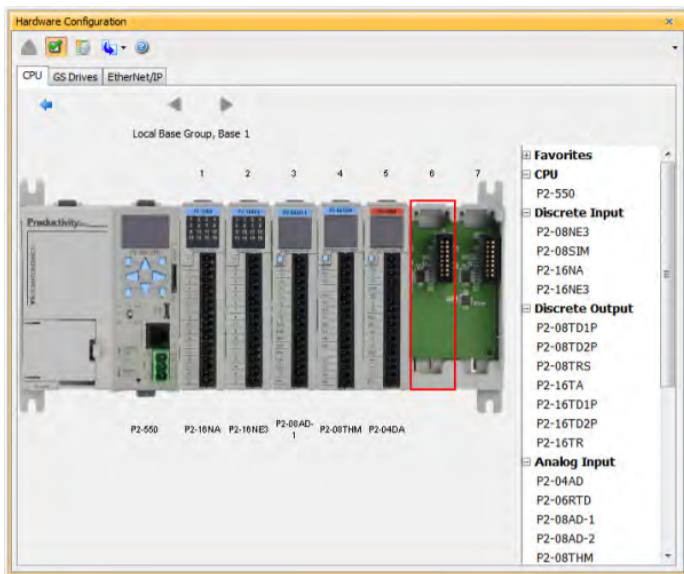
Auto Discovery

A huge timesaver! When software offers automatic hardware detection, everybody wins! With this feature, in order to configure the hardware setup of your PLC, all you do is install each I/O module in the base and power it up. The software will automatically discover the installed modules and assign physical I/O tags based on each module's position in the base. From there you are ready to program with the auto-configured settings or you can reconfigure the setup and assign new tags manually. Some software can also auto detect other connected devices, such as VFDs, to make your job even easier.

Data View and Histograms

Data View windows allow users to monitor and manipulate PLC logic values in real time. Although most PLC software will have a way to do this, there are some advanced capabilities, graphical trend charts for instance, that you may want to keep an eye out for. Also, bit and word histograms allow you to sample and plot logic values over time. These can be set up to sample data as often as you choose and can be very helpful when troubleshooting.
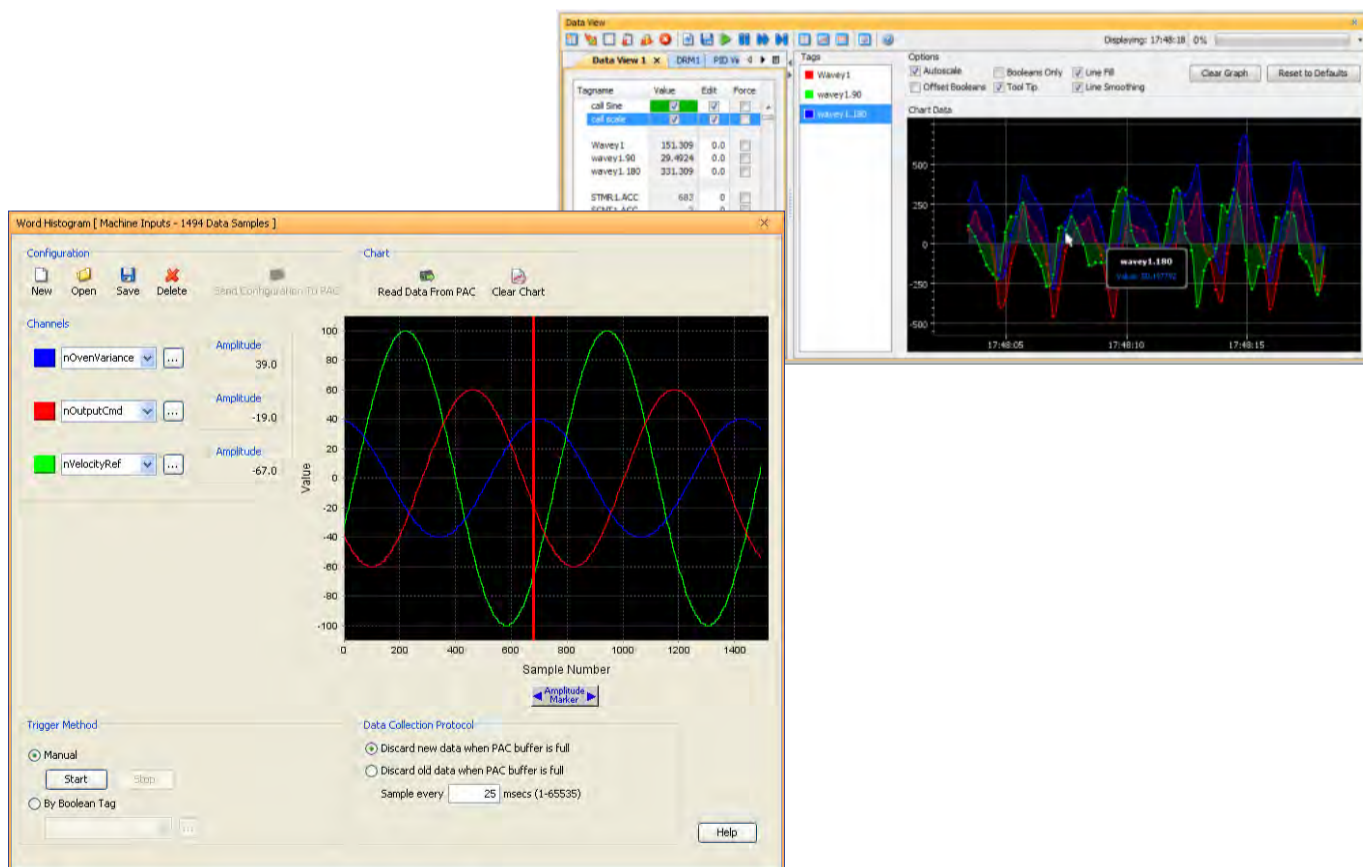
Security

Software user accounts allow you to control who has access to your PLC and what they are allowed to do with that access. System security is an important feature that good PLC software should have.
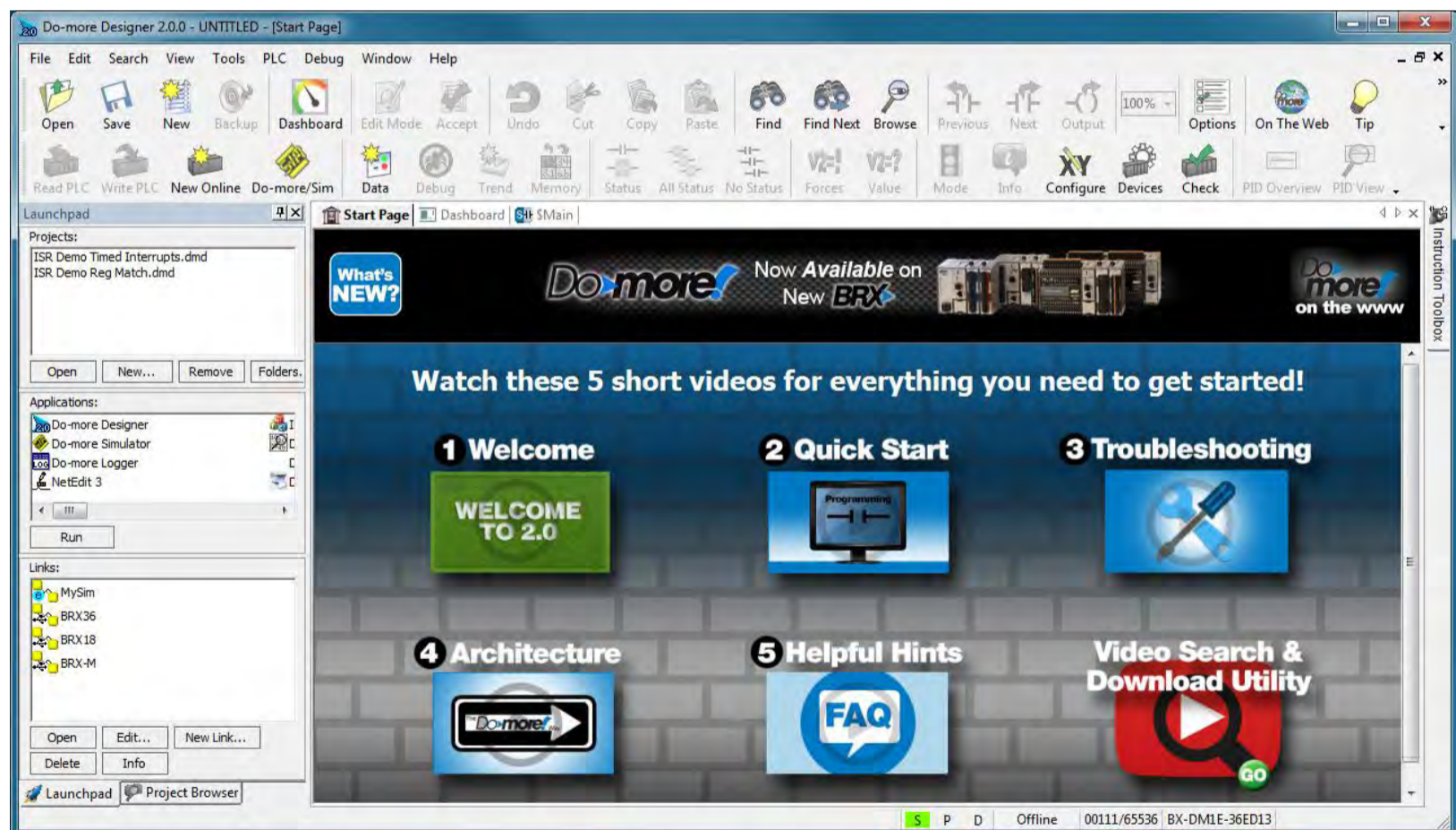
Search and Cross Reference

Three PLC software functions you will use often are the Search, Search and Replace, and Cross Reference. These functions will tell you if an address or variable is being used, where it is located, and allow you to easily make changes, if needed. One thing to consider is how well the PLC software performs these functions and if it offers the usability you need. See what kind of searches are allowed (tag, address, comment, instruction or partial) and what filtering options are available.

Help Files/Videos

Good PLC programming software should have good help files. Embedded help files are a great asset when programming in a new environment or when troubleshooting. Help files should provide information in a way that is easy to comprehend with visuals, screenshots and application examples. They should also include the ability to search for items and be printable.

Some newer PLCs even have videos embedded or include a video browser with the software to help when needed. These videos can provide step by step instructions on how to configure hardware or how to use logic commands.

Connectivity

Another thing to consider is what connection options are available when downloading or working online. A lot of PLCs today offer plug-and-play USB connections for programming, which are fast and easy to use but require direct connections between the PC and CPU. With networked PLCs, Ethernet is typically the connection method (unless you are using a serial network) but you may want to verify if any additional communication software is required with these connections. That software may cause unwanted headaches if it can't recognize connected controllers. One last option you may find helpful when it comes to downloading is USB project transfers. See if your PLC allows projects to be downloaded via an onboard USB port. This can be important if you have remote PLCs that aren't easily accessible with a PC.

Customizable Layouts

No matter what the PLC software has to offer, none of it matters if it isn't user friendly. Layouts that are fully customizable with dockability, display options, view options, accessibility options, and text options allow you to personalize your programming experience to fit your needs.

Project Compare

"Why is this message telling me my project is different from the one on the CPU?!" We've all been there. That's what makes the project compare feature so valuable. It allows you to perform project-to-project comparisons and get detailed information on the differences. This can be very useful when going online or when trying to remember what you changed in the last revision.
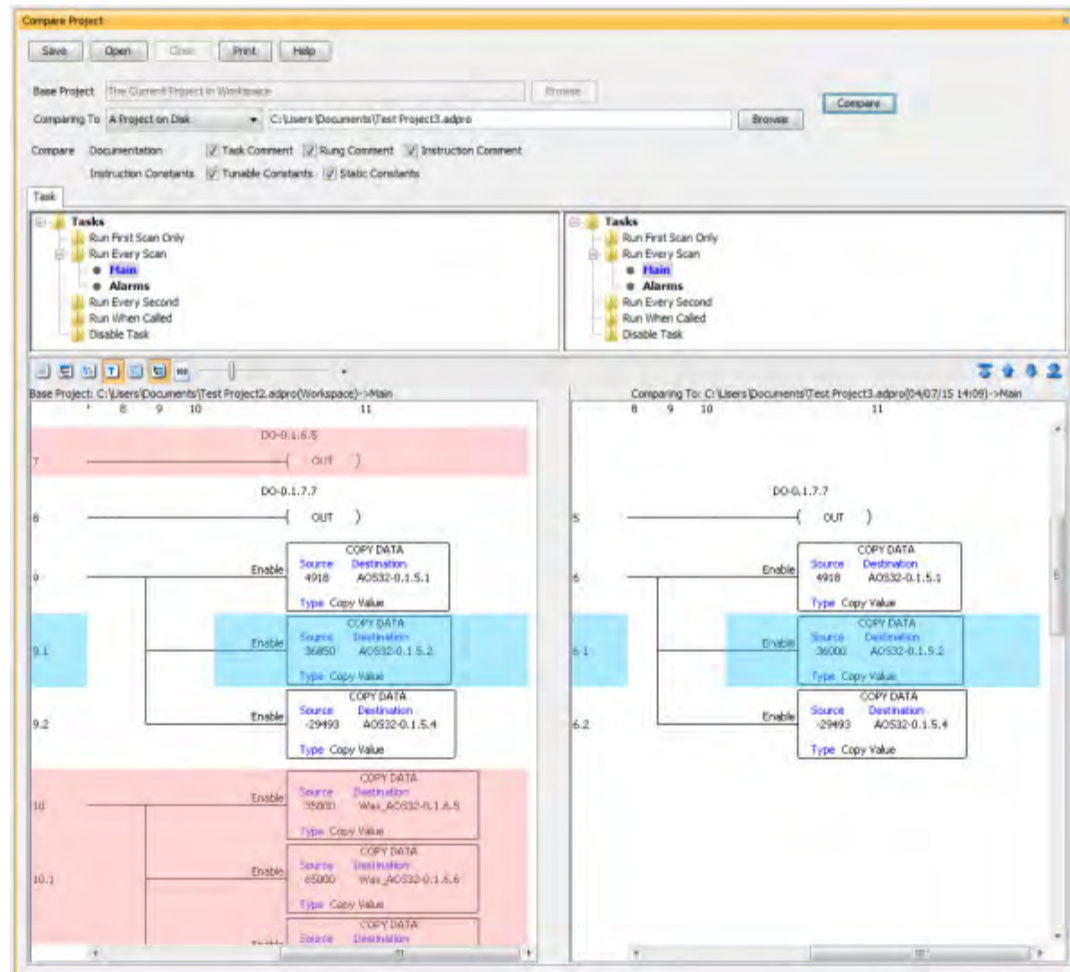
Debugging Tools

Debugging tools assist with locating and repairing bugs in your code by allowing you to slowly step through the execution of your logic. Step from one rung to the next, execute multiple rungs, run one complete scan or pause the execution all together. Some debugging tools also allow you to suspend selected tasks or routines and force watchdog errors.
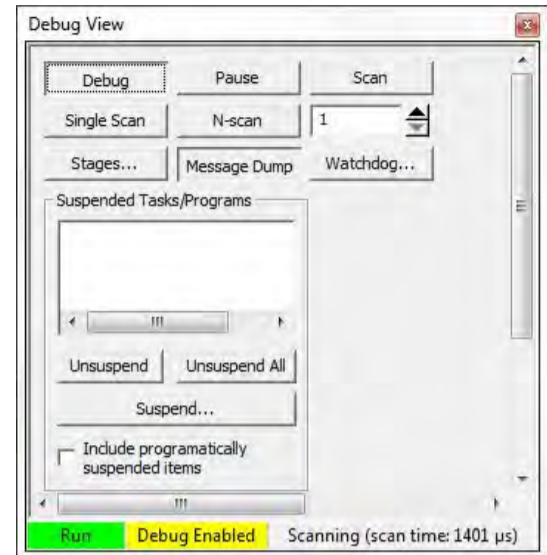
Web Server and Mobile Apps

With Web server functionality, you can access a PLC remotely using a standard Web browser and the configured IP address of your CPU. View system diagnostic data or process values offsite. Mobile apps are also very popular and you may want to see if the PLC you are interested in has mobile capabilities. This way you can monitor system operations on the go from anywhere using a Wi-Fi or cellular network connection.

Email

Integrated email capabilities allow your PLC to notify you and any other needed personnel when critical events occur at your facility. This way you can stay informed about any possible failures and can take immediate corrective action.

If secure emails are required, verify the PLC offers that capability and also take note if the email feature allows data attachments as some newer PLCs do.
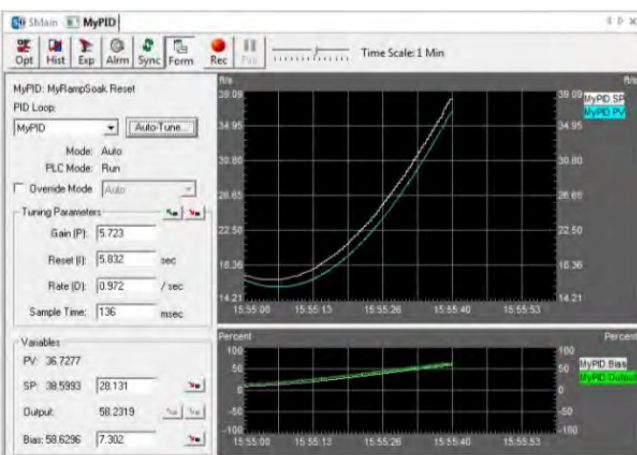
PID Options

For all the process control engineers out there, this is definitely one you'll want to watch for. Check to see how easy it is to implement PID loops in your program. Are there integrated PID instructions and how easy is it to monitor and tune the process? Some PLC software provides auto tuning, PID simulators and other helpful tools dedicated to PID loops.
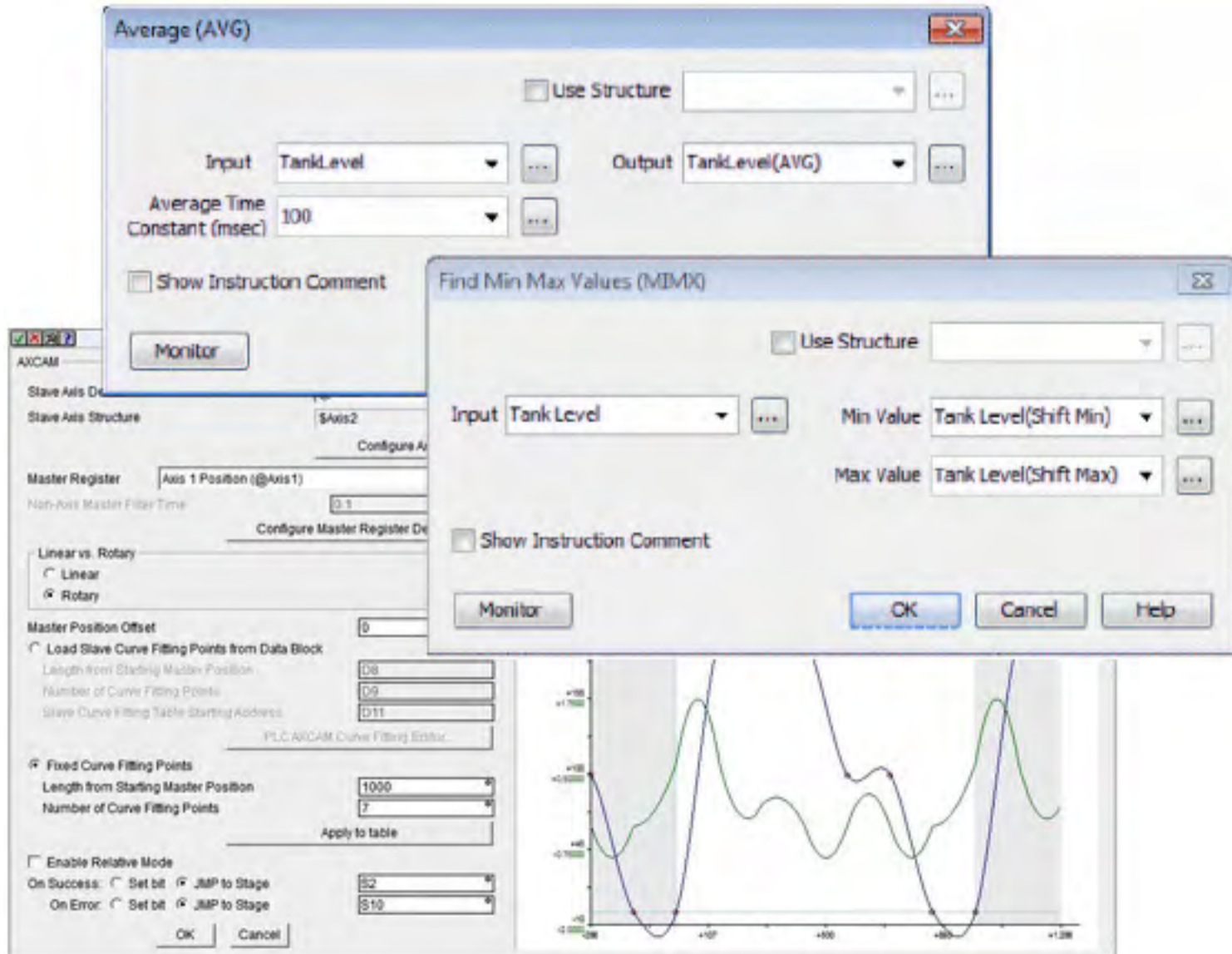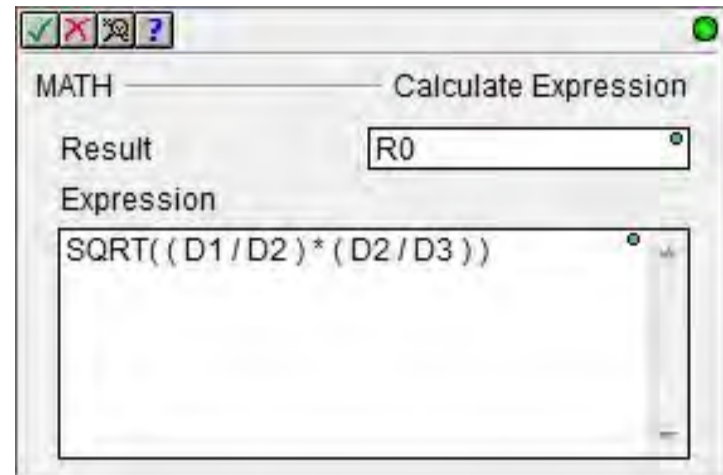
Powerful Math Functions

Powerful math functions that are easy to use are a definite plus to any PLC software. The ability to enter complex mathematical expressions (Ex. SQRT((D1/D2)*(D2/D3))) with floating points and/or integers directly into an editor, without having to load accumulators or use multiple lines of code, is a big timesaver.

Task Manager

The task manager is a great way to organize and control the execution cycle of your program without any extra lines of code. Breaking your program into subroutines that execute only when needed (every scan, every second, first scan only, or when called) can be a lifesaver when scan time is a concern.

Integrated Function Blocks

Integrated function blocks make programming a lot faster. Instead of coding a large amount of ladder logic to handle a complex task, function blocks can do the work for you. PID loops, motion commands, communication functions, etc., can be handled quickly and easily with configurable function blocks.

Other Things to Consider…

Programming Languages

Although ladder logic is still by far the preferred programming language in our industry, there are other languages that you may find appealing. Structured text, instruction lists, sequential function charts and function block diagrams are all options that are available. If you come from a computer programming world you might prefer the "If…Then" loops that structured text offers, or if you are experienced with assembly language, instruction list programming might be for you. Each language offers something different, but choose the one that is easiest for you to program, troubleshoot and maintain.

Tag Name vs. Fixed Memory

There are many preferences and objections to both but you should definitely consider which is most useful to you and your particular application. Tag name databases can be easily integrated with HMIs and other database software but fixed memory controllers may offer better searchability. If you are unsure, feel free to download our tag name based **Productivity Suite software** or our fixed memory **Do-more Designer software** to get a feel for the differences. They are both free!

This list is in no way intended to encompass every aspect of "good" PLC software but it provides you with some features and tools that we think are important and that we think would greatly help with your PLC project development. If you would like to get more information on the PLC software we offer, take them for a test run, or watch informative videos on how to use our software, check out the links below:

• CLICK

• Do-more Designer

• Productivity Suite

Learn More...Watch The Video!
Productivity2000
Distance Sensing
http://www.automationdirect.com/VID-P2-0006

Learn More...Watch The Video!
Do-more
How To Do-more Quick Start
http://www.automationdirect.com/VID-DM-0002

# Understanding Ladder Logic

The Structure

The structure behind ladder logic is based on the electrical ladder diagrams that were used with relay logic. These diagrams documented how connections between devices were made on relay panels. They are called "ladder" diagrams because they are constructed in a way that resembles a ladder with two vertical rails and rungs between them. The positive power rail (on the left) flows to the negative power rail (on the right) through the physical devices connected on the rung. In this example, "PB" stands for "pushbutton", "CR" stands for "control relay", "L1" is a light and "M1" would be a motor.
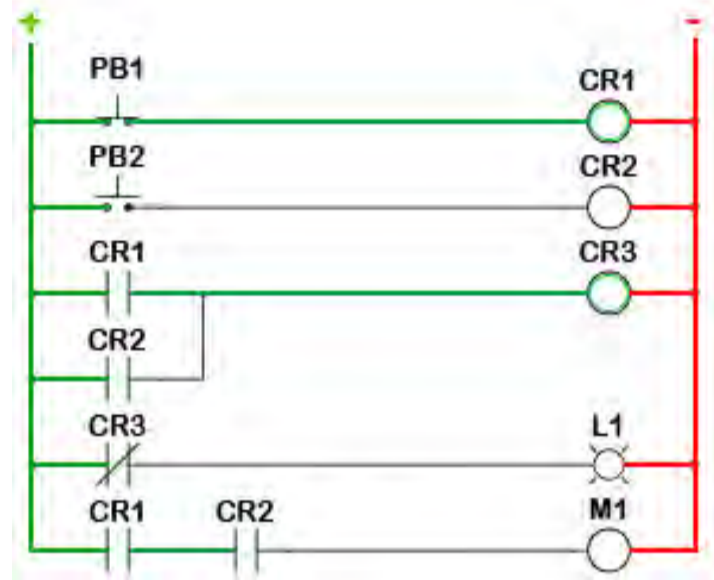
The Ins and Outs

Ladder logic was designed to have the same look and feel as ladder diagrams, but with ladder logic the physical contacts and coils are replaced with memory bits. Let's take a look.

For this program, we took the relay logic's ladder diagram and duplicated it with ladder logic. No more hard-wired logic, but memory locations instead. Some of these memory locations are used internally and others are used with external inputs and outputs. To monitor and control real world devices, they will need to be wired to I/O modules. For this particular PLC, these inputs and outputs are assigned to X and Y memory addresses like the X001 seen with PB1. This normally open contact's state is read from the input on the I/O module where the physical pushbutton is connected. On the other hand, each Y bit will have an output device wired to it as seen with the light controlled by Y001. Besides the I/O points, all other contacts/coils are assigned to internal bits that we can use as needed. One side note, today's PLC CPUs offer many types of functions, not just simple contacts and coils. Math, Shift Registers, Drum Sequencers, etc., are available to aid in your programming but for now, we'll keep it simple.
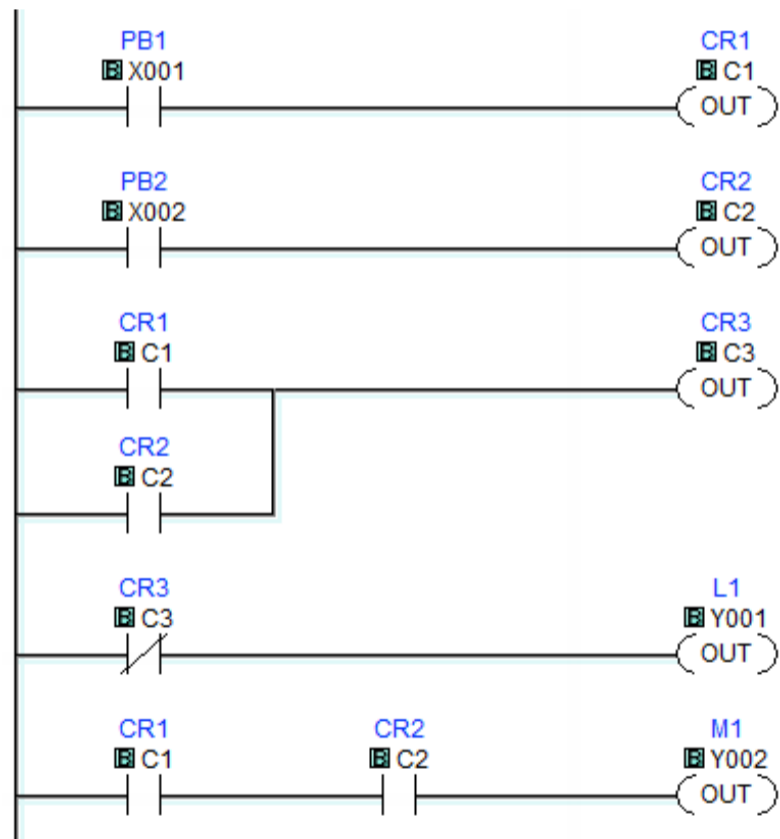
The Execution

The CPU will interpret the logic in a sequential order. Starting at the top left of the program, the CPU will work its way down the rail executing each rung or sub rung from left to right. So if PB1 is pressed, the CPU will turn ON CR1. Since CR1 has changed states, in rung 3 the CPU will activate CR3. CR3's normally-closed state is used in rung 4, so the CPU will then turn OFF L1. Even though we still refer to coils and contacts in ladder logic, remember that they are memory representations, not actual devices. Once the CPU reaches the last rung it will loop back to the start of the program and run it all over. This process will continue as long as the CPU is powered and in the RUN mode. The time it takes the CPU to loop back to the beginning is known as scan time. Scan time also includes the time it takes the CPU to update the outputs and read the inputs and it can be important to applications where timing is critical. Subroutines and special purpose I/O modules can be used to help reduce the scan time if needed.

The Logic Behind the Ladder

So what logic can ladder logic actually perform? With the increasing demand for functionality and ease of use, many of today's PLCs incorporate function blocks with ladder logic. The structure of the program is still ladder with the more complex instructions being function blocks. So to answer the question, let's look at a few examples:

1. Boolean Logic: The ON/OFF, TRUE/FALSE algebra of binary systems. The basics of which are AND, OR, and NOT operators. To put it simply, rung 5 in our code needs CR1(C1) AND CR2(C2) to turn ON motor M1 (Y002).

2. Timing: Timer instructions are available to allow for on-delayed or off-delayed events. Once triggered, the timer will turn its associated output ON (on-delay) or OFF (off-delay) after the set time has elapsed.

3. Counting: Count-up and count-down functions increase or decrease the counter value on every transition of the input.

4. Comparisons: Compare instructions are available to determine if values are less than, equal to, or greater than each other.

5. Math: These instructions not only allow for the simple addition and subtraction but also for more complex operations like tangents, square roots, etc.

6. Special functions: PID loops, communication instructions, shift registers, drum sequencers, ramp generators, etc.

Basic Instructions in Ladder Logic

Now that we have a better understanding of what Ladder Logic is we can dig a little deeper into how it works. And to do that, we need to first understand Boolean math and logic gates. Now, this is not intended to be a Digital Systems class so put away your Karnaugh maps, you won't need them. Instead, we'll just look at a few logic gates and how they work.

The two logic gates that you will see represented most often in Ladder Logic are the AND and OR gates. As the truth tables below show, the AND gate needs all inputs (A and B) to be true before the output becomes true, whereas the OR gate needs just one input (A or B) to be true to make the output true. These truth tables can be duplicated with relay contact logic by connecting normally open contacts in series (AND gate) or parallel (OR gate) as seen below.
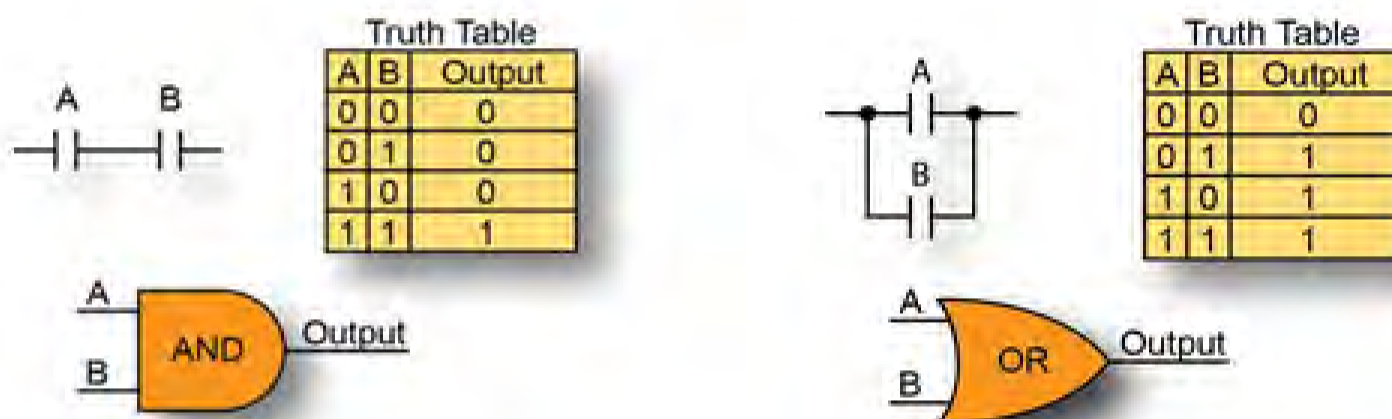
To equate this to a control circuit, let's say Input A and Input B (often referred to as permissive bits) are the Door Open and Motion Detected inputs of a home alarm system. When wired in series, as with the AND gate, both conditions will have to be met before the alarm is activated. The door will have to be opened and the motion detector tripped before the alarm is triggered. With the OR gate, the inputs are wired in parallel and only one of the conditions is needed. In this case, the door opening or the motion detection will trigger the alarm. The option you choose is dependent on your application and on how the system is expected to perform. Pretty simple, right? And in its very basic form, that's what Ladder Logic is. It's the arrangement of permissive bits or contacts into a Boolean expression that determines whether an output should be ON or OFF.

Today's Ladder Logic programming has advanced to include more than just simple contacts and coils but the same underlying principle holds true. The true or false state of each input element in a rung and how it is connected will determine the output's state. So let's take a look at some of the instructions available in current PLCs. And for that we will use the CLICK PLC since it has a simple, easy-to-use instruction set. The table below contains all of the available Ladder Logic elements in the CLICK programming software with descriptions of their functions.
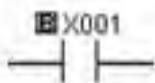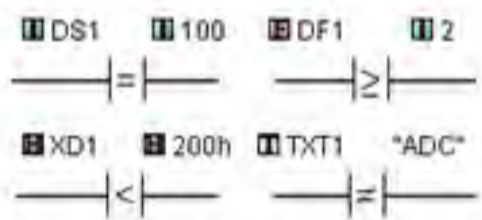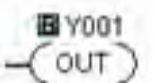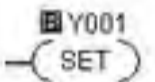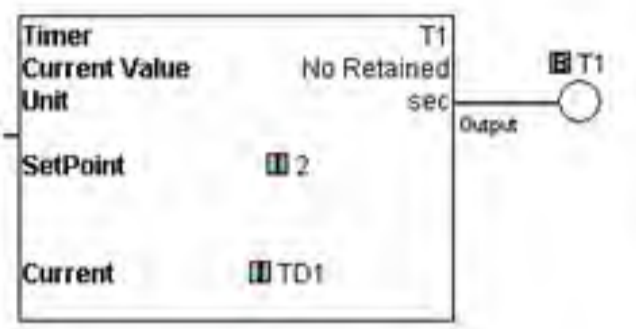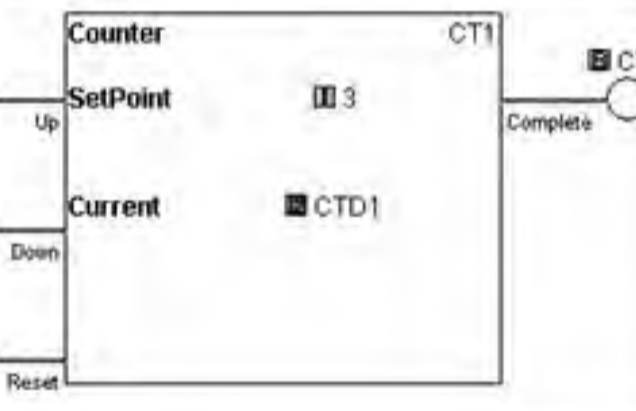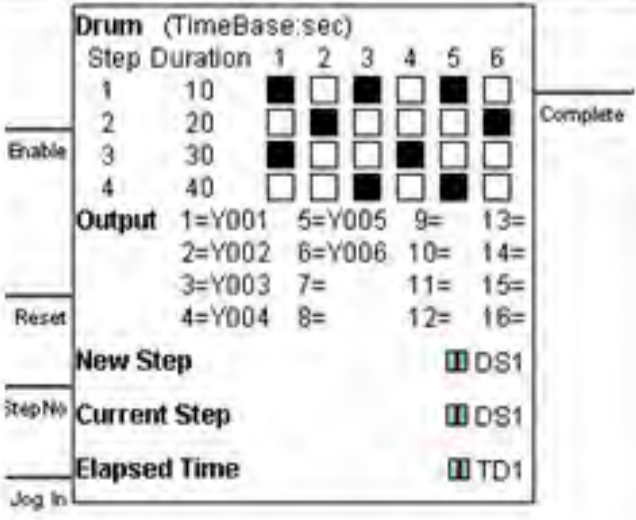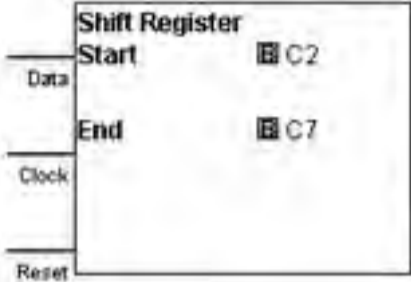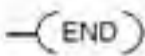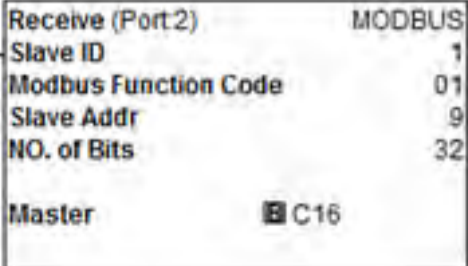
CLICK PLC Instruction Set

Ladder Symbol	Title	Type	Description
▣ X001 —| |—	Normally Open Contact	Bit Instruction	The Normally Open Contact mimics the behavior of a physical contact and changes in response to the status of a Bit Memory Address. The Normally Open Contact is ON when the related bit is ON.
▣ X001 —|/|—	Normally Closed Contact	Bit Instruction	The Normally Closed Contact mimics the behavior of a physical contact and changes in response to the status of a Bit Memory Address. The Normally Closed Contact is ON when the related bit is OFF.
▣ X001 —|↑|— ▣ X001 —|↓|—	Edge Contact	Bit Instruction	The Edge Contact turns ON when the related bit transitions from OFF to ON (Rising Edge) or ON to OFF (Falling Edge).
▥ DS1 ▥ 100 ▣ DF1 ▥ 2 —| = |— —| ≥ |— ▣ XD1 ▣ 200h ▥ TXT1 "ADC" —| < |— —| = |—	Compare Contact	Word Instruction	The Compare instruction uses a Mathematical Operator as a basis for comparison of two data values. When the data values satisfy the selected mathematical relationship (>, <, =, etc.) the Compare Contact turns ON.
▣ Y001 —(OUT)	Out Coil	Bit Instruction	An Out instruction turns ON its associated Bit Memory when the status of the rung is true. The Out instruction turns OFF its associated Bit Memory when the status of the rung is false.
▣ Y001 —(SET)	Set Coil	Bit Instruction	The Set instruction turns ON the associated Bit Memory when the status of the rung is true. The Bit Memory stays on after the rung becomes false.
▣ Y001 —(RST)	Reset Coil	Bit Instruction	The Reset instruction turns OFF the associated bit memory when the status of the rung is true. The Bit Memory stays OFF after the rung becomes false.

Timer Current Value No Retained Unit sec SetPoint 2 Current TD1	Timer	Word Instruction	An ON Delay Timer measures the time duration that begins with a transition of the enable rung from OFF to ON. Beyond this transition point, the Timer increases the Current Value; when it reaches the SetPoint, the Timer Bit is turned ON. An OFF Delay Timer measures the time duration that begins with a transition of the enable rung from ON to OFF. Beyond this transition point, the Timer increases the Current Value; when it reaches the SetPoint, the Timer Bit is turned OFF.
Counter CT1 SetPoint 3 Up Current CTD1 Down Reset	Counter	Word Instruction	When enabled, a Counter instruction counts up or down (depending on user settings) until it reaches the SetPoint. The Counter counts in response to the transition from OFF to ON of the enabling rung (up or down rung).
Math (PI * DS2 ^ 2) * (DS3 - SQRT (DF5)) + (5 MOD DS8) Result	Math	Word Instruction	The Math instruction solves a user-defined formula during the execution of the Ladder Program. Once the enable rung transitions from OFF to ON the formula will be solved and the result will be stored in the data format and location selected for the Result.
Drum (TimeBase:sec) Step Duration 1 2 3 4 5 6 1 10 2 20 Complete Enable 3 30 4 40 Output 1=Y001 5=Y005 9= 13= 2=Y002 6=Y006 10= 14= 3=Y003 7= 11= 15= Reset 4=Y004 8= 12= 16= New Step DS1 Step No Current Step DS1 Elapsed Time TD1 Jog In	Drum Instruction	Special Instruction	The Drum instruction simulates an electromechanical drum sequencer, using either a Time Based or an Event Base sequencing strategy. Each Drum instruction is capable of sequencing through 1 to 16 steps and turning ON as many as 16 outputs in a user defined pattern. Outputs can be either physical outputs or internal control relays.

Shift Register Start C2 Data End C7 Clock Reset	Shift Register	Bit Instruction	The Shift Register instruction shifts a range of control bits one memory location with each OFF-to-ON transition of the Clock pulse. If the Starting Address is lower than the Ending Address, the Shift register will Shift from the Starting Address to the Ending Address. If the Ending Address is lower than Starting Address then Shift Register will Shift from Ending Address to the Starting Address.
Copy Single Src 1234 Des DS100	Copy Instruction	File Instruction	The Copy instruction is used to copy a data or text value from its Source location to a specified Destination register.
Search Continuous Search ON Result Search > 2 Range DS101 DS110 Found	Search Instruction	File Instruction	The Search instruction is used to search for a data value that meets the specified condition and that is located within a specified range of data registers.
Call Subroutine01	Call Instruction	Program Control	The Call instruction is required to call or execute a Subroutine Program from the Main Program.
For DS1	For Instruction	Program Control	The For instruction indicates the starting point of a For-Next loop, and based on its user setting, determines how many times the For-Next loop will be executed in one program scan. Between the For instruction rung and the Next instruction rung, place the rungs of logic that should be repeated multiple times.
NEXT	Next Instruction	Program Control	The Next instruction indicates the end of a For-Next loop.
—(END)	End Instruction	Program Control	The End instruction marks the termination point of the normal program scan. The End instruction tells the CPU that there are no more rungs to be processed.
Receive (Port 2) MODBUS Slave ID 1 Modbus Function Code 01 Slave Addr 9 NO. of Bits 32 Master C16	Receive Instruction	Communication	The Receive instruction allows you to use the available communication ports on the CLICK CPU modules as a network master and read data from external devices.

Send (Port:2) MODBUS Slave ID 3 Modbus Function Code 05 Slave Addr 1 Master ☰ C54	Send Instruction	Communication	The Send instruction allows you to use the available communication ports on the CLICK CPU modules as a network master and write data to external devices.

As you can see, there are many more options available than just normally open contacts and many function blocks have been incorporated into Ladder Logic. For instance, the Drum and communication instructions are blocks of code that are stored and available for you to configure and use in your program. Now, we aren't going to discuss every instruction available but one of the more frequently used instructions is the Compare contact.

The Compare contact can be programmed to look at two different numerical values; either or both values can be a variable or fixed value. The Compare instruction will evaluate if the two values are:

• Equal

• Not Equal

• One Greater Than the other

• One Greater Than or Equal To the other

• One Less Than the other

• One Less Than or Equal To the other

If the evaluation condition is met, then the result will be true, passing the logic path along to the next instruction or turning the rung output ON.

So, how can we use the Compare contact in an application? Imagine having a freezer full of frozen fish. The key word there is frozen! We don't want to risk a smelly disaster if the fish were to thaw, so we've installed a sensor inside the freezer to monitor the temperature. The temperature sensor's output produces an analog signal that is wired into a PLC. The PLC is programmed to take the analog signal and convert it into degrees Fahrenheit. If the temperature reads greater than 32 degrees, we can sound an alarm horn to alert us that the frozen fish will soon thaw and spoil.

Using the Compare contact instruction dialog box as seen below, we can compare our actual Freezer Temperature (stored in memory location DF16) to the constant 32, which represents 32 degrees Fahrenheit.
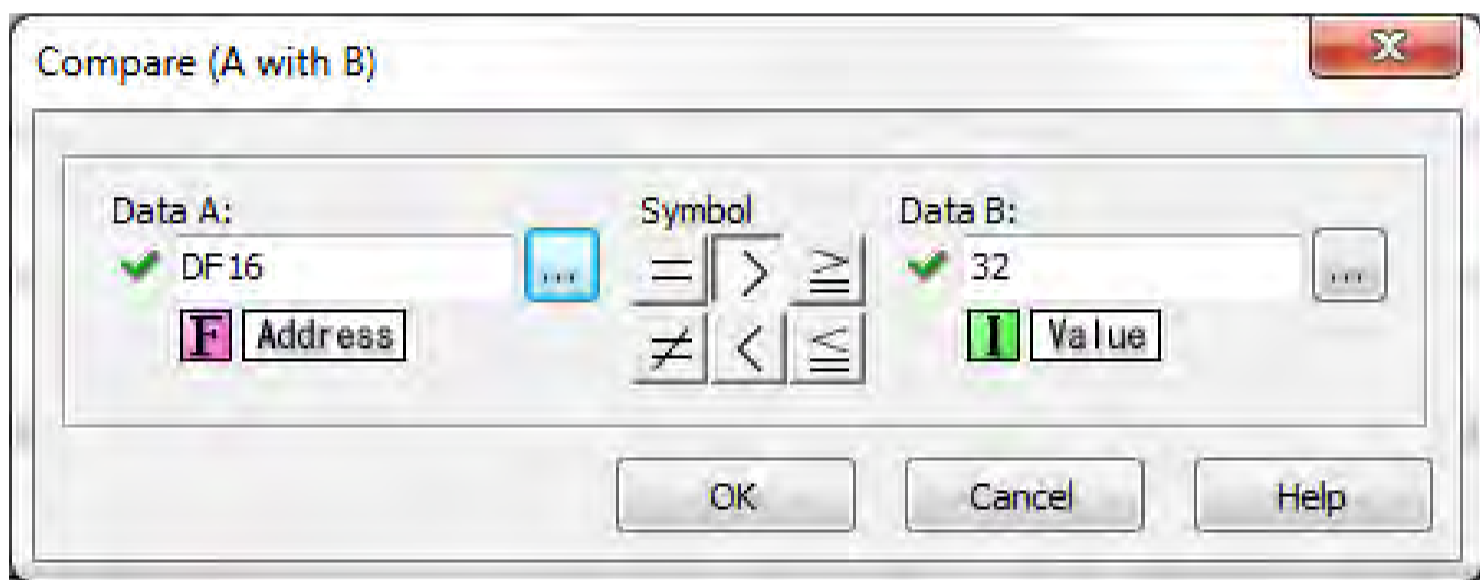
The Compare contact dialog box allows us to select any of the six different comparison methods. For this example, we have selected Greater Than so that if our Freezer Temperature ever becomes Greater Than the number 32, our Compare contact will become true.

An Out coil is then programmed from our Compare contact, as seen below, and will provide the signal that is wired to the external Alarm Horn, which alerts us to a problem with our freezer and hopefully giving us time to avoid disaster.
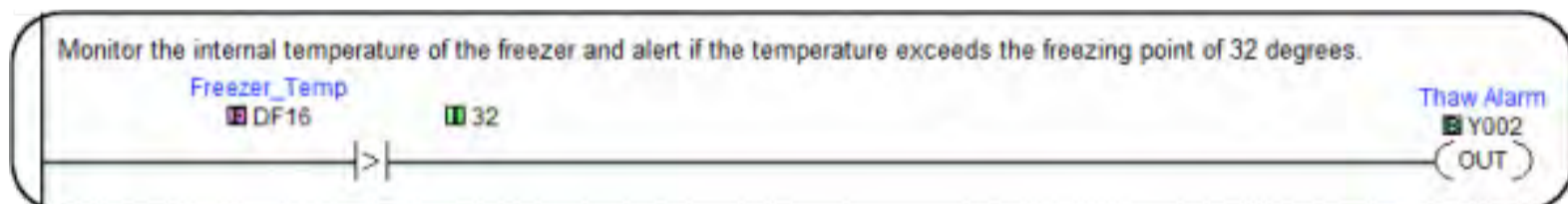
Ladder Logic is the most widely used programming language in industrial automation today. Its ease of use, traceability, and visual representation of physical components make it the favored programming method of many engineers. If you are new to PLC programming and would like to try Ladder Logic for yourself, download any of our programming software packages mentioned below for free and see what you think. After all, experience is the best teacher.

• **CLICK**
• **Do-more Designer**
• **Productivity Suite**

Give it a Try…

If you would like to try an exercise in ladder logic, AutomationDirect has created a beginner's programming exercise. This exercise was actually created for the Boy Scouts of America to help teach ladder logic to future PLC programmers. It uses the simulator included with the Do-more Designer programming software for our Do-more PLC series. The software is free and you do not need any hardware, so try it. **Download the free software** and watch the video below.

Learn More...Watch The Video!
BSA Programming Merit Badge
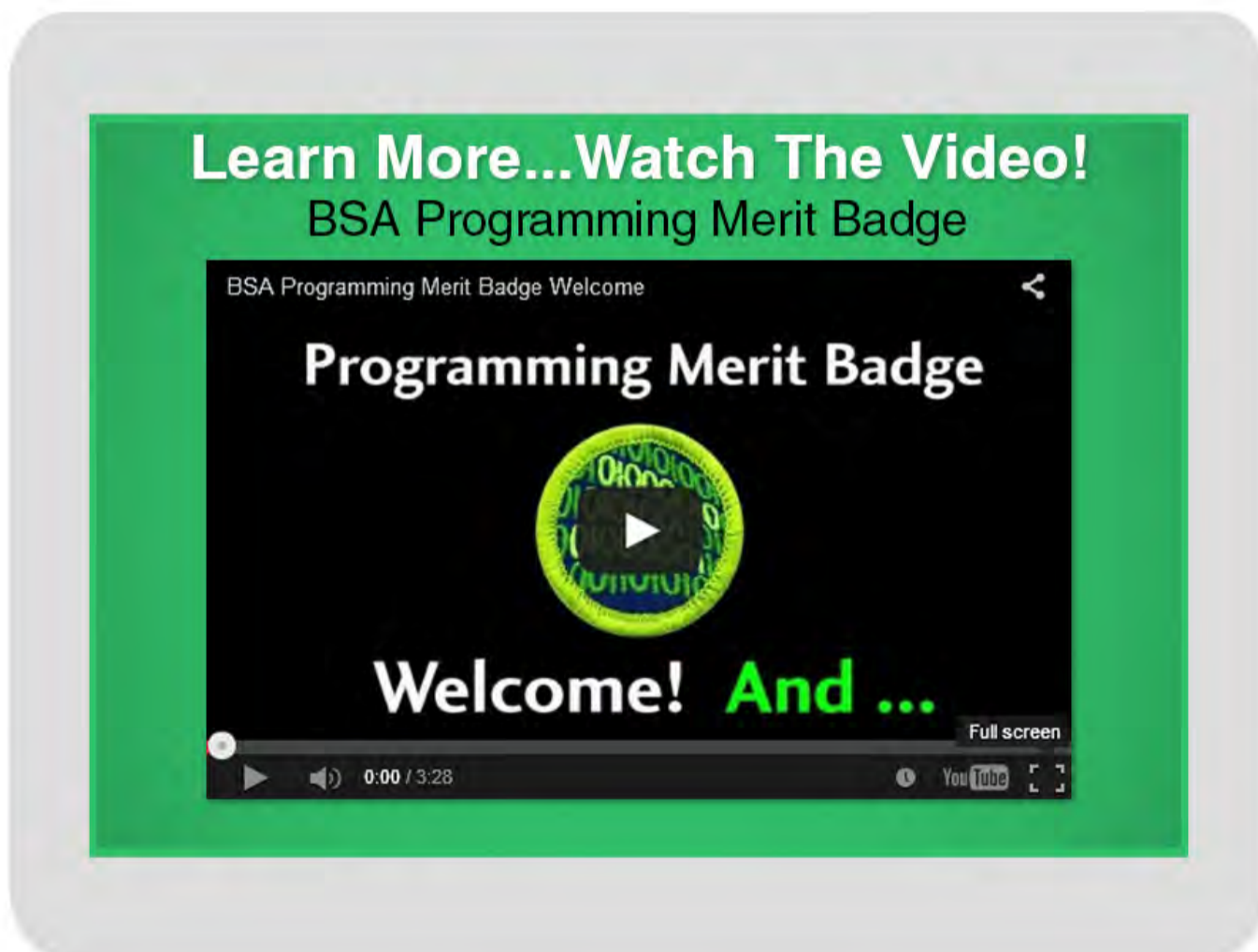
Ladder Logic in Action

Ladder logic programming for industrial controllers has evolved significantly over the past 30 years, and now supports advanced functionality such as process control, motion control, data manipulation, networking and data acquisition. However, when you start with a blank page to design a new control system, large or small, there are many basic functions this tried-and-true language is still asked to perform, and in fact, perform effortlessly.

Most every controls designer has implemented at least one of these functions in every system they have commissioned. Even when moving from one supplier's platform to another, the ladder code design can be copied and adjusted for the particular controller, becoming tested "building blocks" of logic to speed up programming and reduce troubleshooting time. Let's look at some of the most common control system functions and some practical ways to implement them.

Object Detection

One of the most pervasive functions of a control system is detecting the presence of an object. Whether you are detecting the presence of an object passing by on a conveyor, the closure of a gate, or the presence of a machine part as it goes through its motions, object detection is a staple of the automation industry. There are a myriad of object sensing technologies, such as mechanical; inductive, capacitive and ultrasonic devices that detect nearness (proximity); and photoelectric sensing using light beams.

Limit Switch

The most basic sensor is a **limit switch**, an electromechanical device used to detect the presence or absence of an object. The switch operates its set of contacts when its actuator comes into physical contact with the sensed object. Actuator styles offer application-specific means of contact – rollers, levers, springs, wands, plungers, etc. However, since they consist of moving parts, they are prone to wear and damage; making physical contact with the sensed object is not always desirable or possible.

Inductive Proximity Switches

Inductive proximity switches are the most common and inexpensive non-contact sensing technology, used to detect the presence of metallic objects without actually touching it. Their high-speed switching and small size make them indispensable in automation applications. Inductive proximity switches consist of a coil driven by an oscillator. The oscillator creates an electromagnetic field that appears at the active face of the switch. If a metal target enters this area, the electromagnetic field is reduced and the switch turns on or off. Some typical inductive sensor applications are counting metallic objects, monitoring the position of elements in a machine, sensing the presence of metallic parts like screws, etc., and measuring rotational speed.

Proximity Sensors

Proximity sensors typically come in shielded and unshielded styles. With a shielded proximity sensor, the face of the sensor can be mounted flush with metal, whereas an unshielded sensor should NOT be mounted flush with metal (otherwise the sensor will always be ON). In many applications, flush mounting is a requirement. Note that unshielded proximity sensors allow for greater sensing distances. Typical sensing ranges for inductive proximity sensors vary from 1 to 35 mm.

Capacitive Sensors

Capacitive sensors detect objects with a dielectric constant different from air, which makes them ideal for a much wider range of material detection, such as wood, liquids and plastic. Their operation is similar to inductive sensors, but instead creating an electrostatic field (vs. electromagnetic) that is changed by the presence of the object. Capacitive sensing ranges can typically reach up to 40mm.

Ultrasonic Proximity Sensors

Ultrasonic proximity sensors are based on the emission of a sound impulse and the measurement of the time elapsing of the return echo signal reflected by the detected object. The ultrasonic beam is well reflected by almost all materials (metal, wood, plastic, glass, liquid, etc.) and is not affected by colored, transparent, or shiny objects. This allows the user to standardize on one sensor for many materials without any extra setup or sensing concerns. This sensing technology typically offers ranges up to 6 meters.

Photoelectric Sensors

Photoelectric sensors use a variety of sensing technologies that address diverse application configurations, all using light beams as the detecting medium. The three most popular are diffuse, reflective and through-beam styles. The light source used – visible, infrared, LED, laser – will affect the sensing distance.

Diffuse Sensors

In **diffuse sensors**, the emitter and receiver form part of the same unit. The optical beams are either parallel or slightly converging. The presence of an object in the optical field causes diffused reflection of the luminous beam. The receiver detects the reflection from the object itself. The reflective properties of the object are important. It is generally possible to reliably detect the presence of any object unless it is perfectly reflective or black. Diffuse sensors offer a typical sensing range of up to approximately 2 meters.

Reflective Sensors

A **reflective sensor** also houses the emitter and receiver in the same unit, with the optical beams being parallel. The emitter's luminous beam hits a reflector mounted on the opposite side of the object's travel path and is redirected toward the receiver. Detection occurs when the path of the beam is interrupted by the presence of an opaque object. Operating distance mainly depends on the quality of the reflector used and on the optical-beam angle; ranges are typically up to 15 meters.

Through-beam Sensors

With **through-beam sensors**, the emitter and receiver are housed in two separate units and are installed in line with each other on either side of the object path. Detection occurs when the beam is interrupted by the presence of an opaque object. This style is the most accurate, with the farthest sensing ranges, up to 30 meters, and 50 meters or longer when using laser sensors.

Object Detection + Ladder Logic

So how does this look in Ladder Logic? Well, once the sensor has been wired to the input module and the module has been configured in the software's hardware configuration (done automatically with the auto-discover feature), the PLC CPU will assign a memory location to it. In the image on the next page, you can see that the Productivity2000, which is a tag name-based controller, has assigned the names DI-0.1.1.1, DI-0.1.1.2, DI-0.1.1.5 and DI-0.1.1.6 to the sensor inputs. These tag names identify the type of module (DI = discrete input), base group (DI-0 = local base, DI-1 = remote base), expansion rack number (DI-0.1 = 1st rack in local base group, DI-1.2 would be 2nd rack in the remote base group), slot of base the module is in (DI-0.1.1 = slot 1, DI-0.1.5 = slot 5, etc.) and the point on the module the sensor is wired to (for our example the sensors are wired to points 1, 2, 5, and 6). The programmer can easily change these to more meaningful names.

Four normally-open contacts are then added to four individual rungs in the main program of the ladder code. These contacts are assigned the tag names that were configured for our object detection sensors. Normally-closed contacts and edge-driven contacts could have also been chosen. As you can see in rung 4, when the sensor detects an object, the contact will change state and the rung will activate the C3 output coil. This coil is an internal bit but if a discrete output module is installed, this coil can be programmed as a physical output used to activate an alarm, turn on a light, fire a diverter arm, etc.
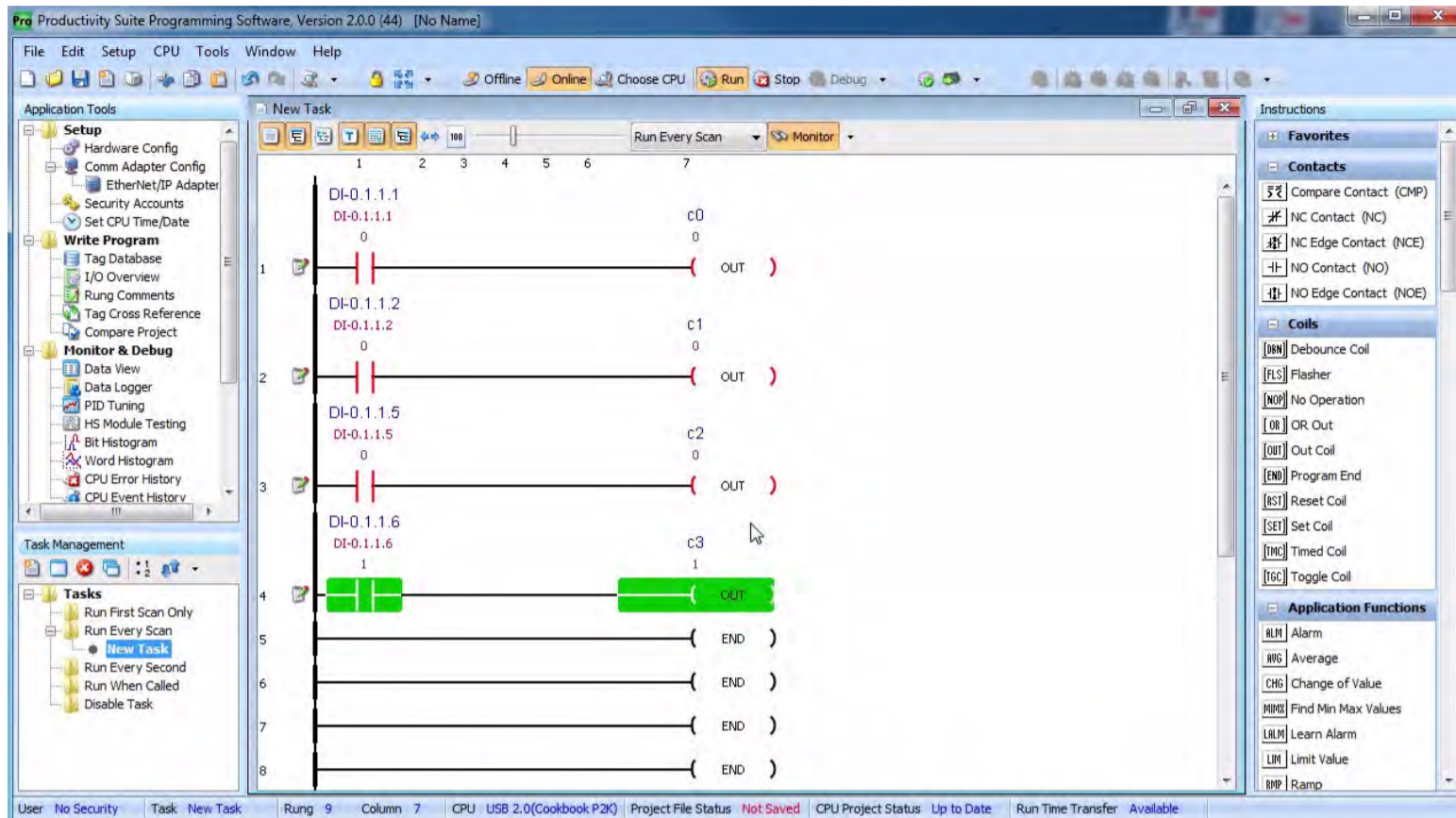
Learn More…

To learn more, see this **series of videos** on the different types of sensing technologies and their suitable application spaces; see typical ladder logic programming for detecting objects with CLICK, Do-more and Productivity series PLCs.

Visit: www.AutomationDirect.com/Cookbook

Distance Measuring

A more advanced object detection function involves measuring the actual distance of an object from the sensing point. Ultrasonic sensors and laser distance sensors are commonly used to measure how far away an object is, out to 100 meters and beyond. Laser devices can be incredibly accurate, with resolutions down to 8um.
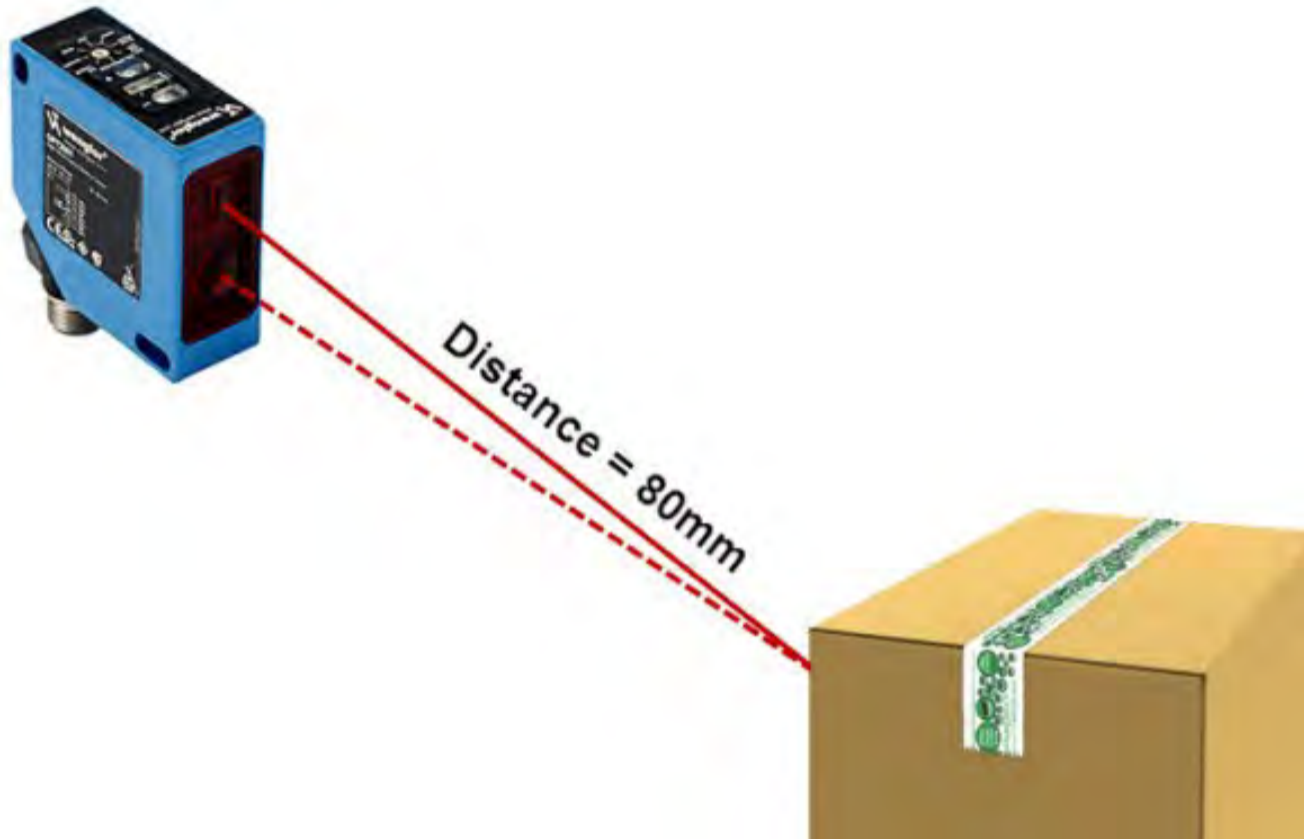
Ultrasonic Sensors

Ultrasonic sensors use sound waves to measure distance, sending out a burst of audio energy and waiting for it to return. Given the rate sound travels through air, the distance is just the time it takes for the energy to get to the object and back to the sensor, divided by two. Using sonic energy, these sensors do not have issues with things that plague optical sensors like smoke, dust, fog and steam. Conversely, they can be less effective in environments with high vibration.

Ultrasonic sensors fill the gap between laser distance sensors, which have long ranges but are more expensive, and proximity sensors, which have very short ranges and are typically inexpensive. Ultrasonic sensors can measure distances up to several meters, which is all that is needed in many applications such as bulk material distance measurements.

Laser Sensors

Laser sensors use highly-focused (coherent) light beams to perform distance measurement. Techniques include triangulation, time of flight, phase-shift and frequency modulation. Laser sensors for industrial use typically employ time of flight (transit time), similar to the ultrasonic method in that they measure the time it takes the laser pulse from the emitting device to reach the target and return.

Typical laser distance sensors come in two power ranges referred to as Class-I and Class-II lasers. The main difference is Class-I lasers have a maximum output of 0.5 mW, where Class-II lasers can have up to 1mW of output. In an industrial environment with subdued lighting, the Class-I laser devices will work just fine. But, if in a brightly lit area the more powerful Class-II laser may warrant consideration. Keep in mind that the safety requirements are different for each. Class-I lasers are considered safe from creating any potential hazards. Class-II lasers are also considered safe because of the assumption that your eye's blink reflex will protect you. You probably don't want to be the one that tests that, so you will want to be sure the Class-II laser is not used where potential eye contact can be made either directly or via reflection.

The extreme resolution and fast response time of a laser sensor make it ideal for precision measurements, such as thin film thickness, or even detecting the diameter of rolls of film.

One issue with laser sensors is since they depend on visible light, they have a tough time detecting transparent objects like glass, plastic bottles, etc. Laser sensors will also suffer in environments with lots of dust and particulates floating in the air, since they have to 'see' the object, and they have different responses depending on the color of the object. Some objects reflect laser light better than others.

Distance Measurement + Ladder Logic

Now, let's look at some distance values in Ladder Logic. Laser and ultrasonic distance sensors provide an analog value to the PLC; therefore, they will need to be wired to an analog input module. For this example, the CLICK PLC will be used. The CLICK PLC does offer the auto-discover feature and once the modules have been installed, the PLC CPU will recognize all of the I/O modules available. Since these inputs are analog, we will need to scale them appropriately. With the CLICK PLC, analog scaling can be done in the system configuration as seen below. Our ultrasonic sensor is connected to channel one of the C0-04AD-1 analog input card. The raw analog (4-20mA) input value from this channel is scaled to a range of 3 to 12 inches and is stored in memory location DF1.
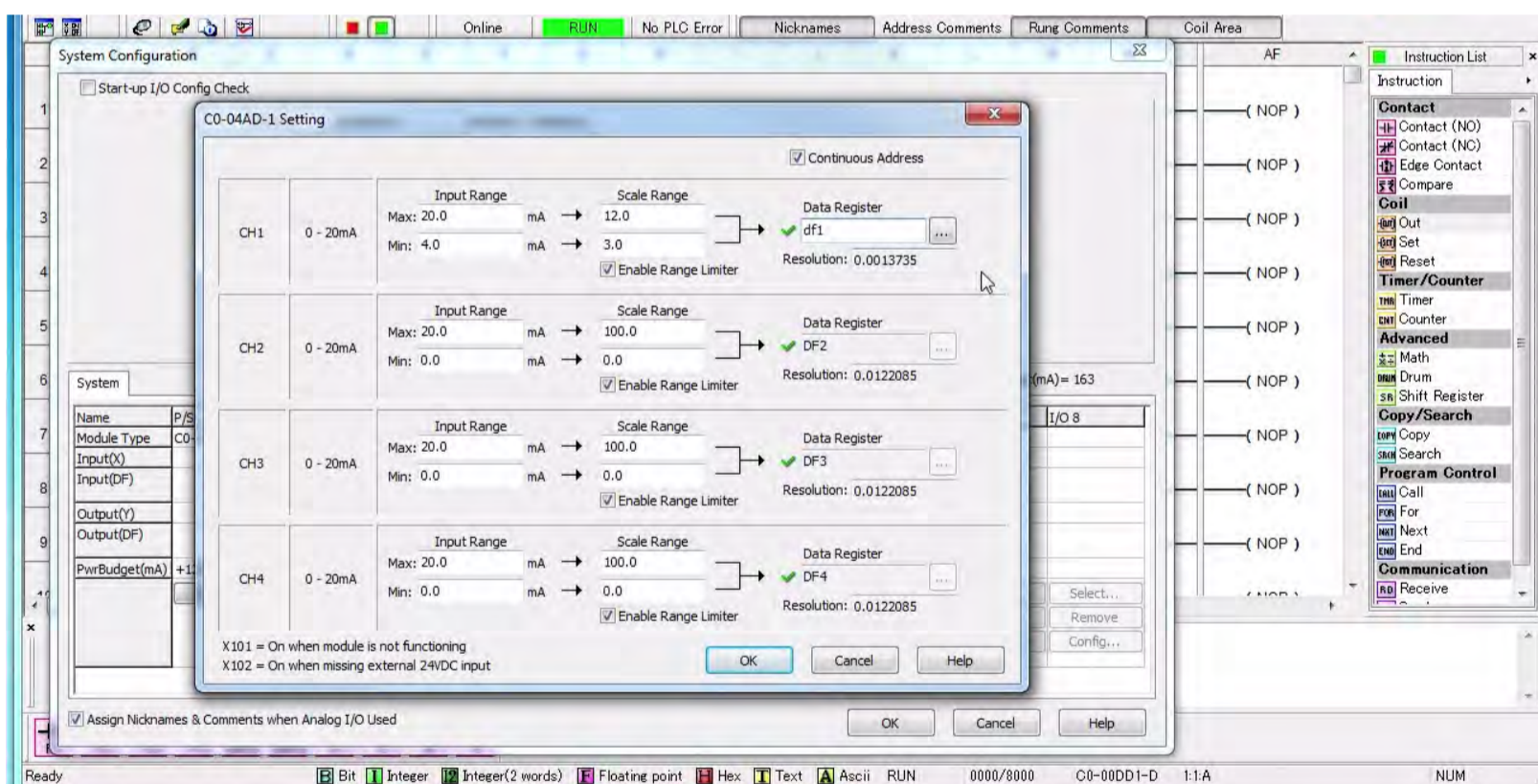
Now that the input has been wired, configured, and scaled, it can be used in the ladder code. Below is the ladder program and you can see that the ultrasonic sensor input is being compared to a static value (8 inches) on rung one. The compare contact is provided by the software and the output of the comparison is connected to a physical output coil. As mentioned previously, this output can be used to turn on a light, activate an alarm, open a valve, etc.
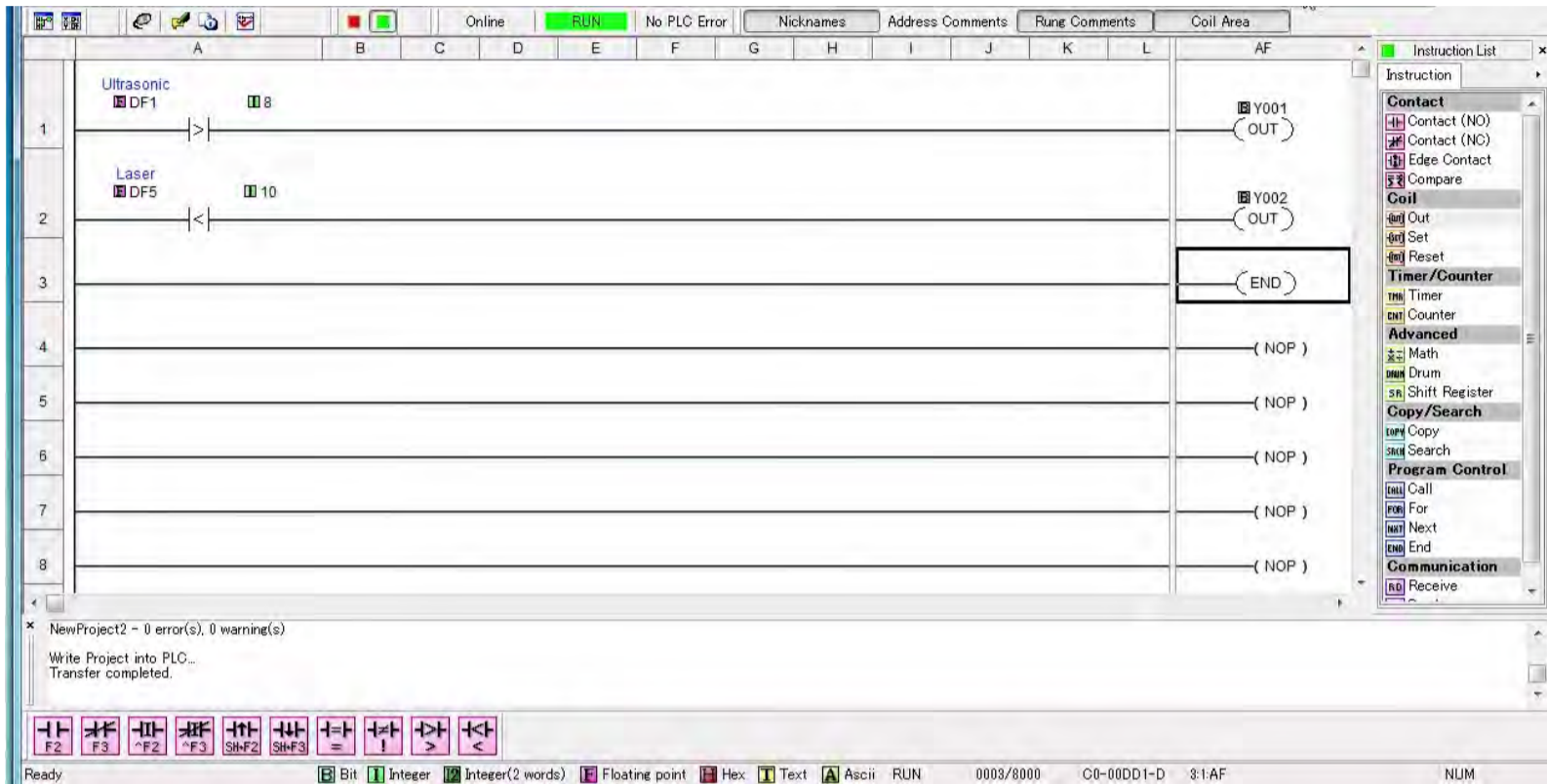
Learn More…

With this **series of videos**, you will learn more about distance measuring technologies including ultrasonic and laser sensors; selecting the correct measuring style depending on the environment and object types; how to set up and use distance sensors, see typical ladder logic programming for distance measuring with CLICK, Do-more and Productivity series PLCs.

Visit: **www.AutomationDirect.com/Cookbook**

A/C Motors (ON/OFF Control)

Most control systems have to make things move, and that usually involves motors. Lifting, pumping, robotics, conveyors, fans – pretty much everything uses a motor of some kind. General purpose three-phase AC motors are great for simple on/off systems, and Inverter-duty motors are specifically designed for operation with variable frequency drives, or VFDs.

General Purpose Motors

General purpose motors are typically connected to the main power circuit with a master circuit breaker or fuses, and use contactors to enable and disable the power to the motor; overloads protect your equipment from unexpected overcurrent/overheating that can be caused by jams or breakdowns.
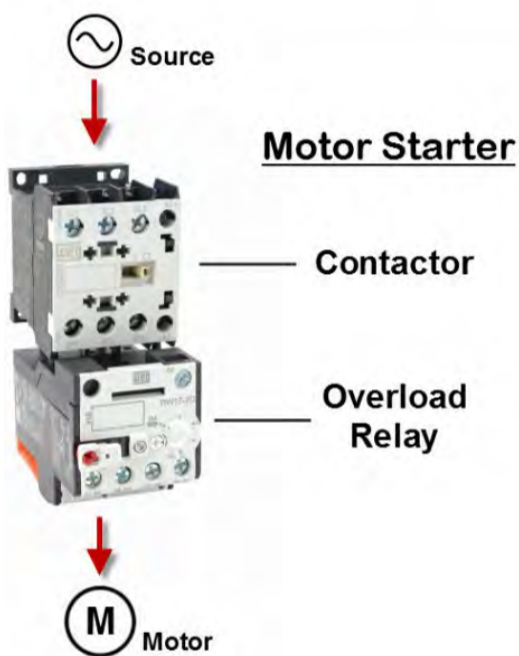
Motor Starters

A **motor starter** is a combination of devices used to start, run, and stop an induction motor based on commands from an operator or a controller. The motor starter must have at least two components to operate: a contactor to open or close the flow of energy to the motor, and an overload relay to protect the motor against thermal overload.

Contactors

A **contactor** is a 3-pole electromechanical switch whose contacts are closed by applying voltage to a coil. When the coil is energized, the contacts are closed, and remain closed, until the coil is de-energized. The contactor is specifically designed for motor control, but can be used for other purposes such as resistive and lighting loads. Since a motor has inductance, breaking the current is more difficult, so the contactor has both a horsepower and current rating that needs to be adhered to.

Overload Relays

The **overload relay** is a device that has three current sensing elements and protects the motor from an overcurrent. Each phase going from the contactor to the motor passes through an overload relay current-sensing element. The overload relay has a selectable current setting based on the full load amp rating of the motor. If the overload current exceeds the setting of the relay for a sufficient length of time, a set of contacts opens to protect the motor from damage.

Surge Protection

Since the signals between the contactor and PLC are inductive, it is incredibly important to protect your control system from power surges and inductive kickback. Surge protection comes in several different types – Diode, Transorb, RC networks. Which one you use depends on your application. Diodes are great if you have a DC circuit – they will offer the maximum protection BUT increase the time it takes to turn the motor off. Transorbs are great for AC or DC circuits, are inexpensive, and have a medium to short delay but only attenuate surges above their rated value, which can be above the threshold allowed by the PLC's I/O. RC circuits mostly filter noise and are not terribly effective for true surge protection.

Reversing Kits

Some control systems require that the motor be reversible. On a three-phase motor that simply involves swapping any two of the three wires. Reversing kits are bus bars that are pre-wired to perform the wire swapping function and some even include extra wiring to help protect against trying to drive the motor in both directions at the same time.

AC Motor On/Off + Ladder Logic

In its simplest form, the ladder code to start an AC motor uses a start input to activate an output wired to a motor contactor. The image below is from a Do-more PLC and it shows how a start contact is used to activate the forward motor contactor. This start contact is controlled by a C-more HMI. The HMI will close the normally open start contact in rung one whenever the Start button on the HMI screen is pressed, and the motor will start in the forward direction.

In reality, most motor starter code is not that simple. There are a lot of other factors involved with starting a motor. The next image shows some of the possible permissive bits that can be used to start/stop a motor. You can see that a MOTOR_COASTING bit has been added. This bit will block the start command when the motor is still coasting to a stop. The MOTOR_STOP_BTN contact is added to stop the motor from starting when it is being requested to stop. A REVERSE_ACTIVE bit is added so the motor does not start in the forward direction when already started in reverse. The OVERLOAD_STATUS contact will stop the motor if an overload condition is detected. The CB_STATUS contact doesn't allow the CPU to attempt a start when there is no power applied to the contactor. A "seal-in" or "latching" bit is also added below the rung to keep the motor running once the contactor is closed. The seal can be broken by any of the contacts to the right of the MOTOR_COASTING contact. Other possible additions include: an emergency stop pushbutton contact, a rung for the reversing contactor, a shunt trip contact, an alarm-on-start function, etc.
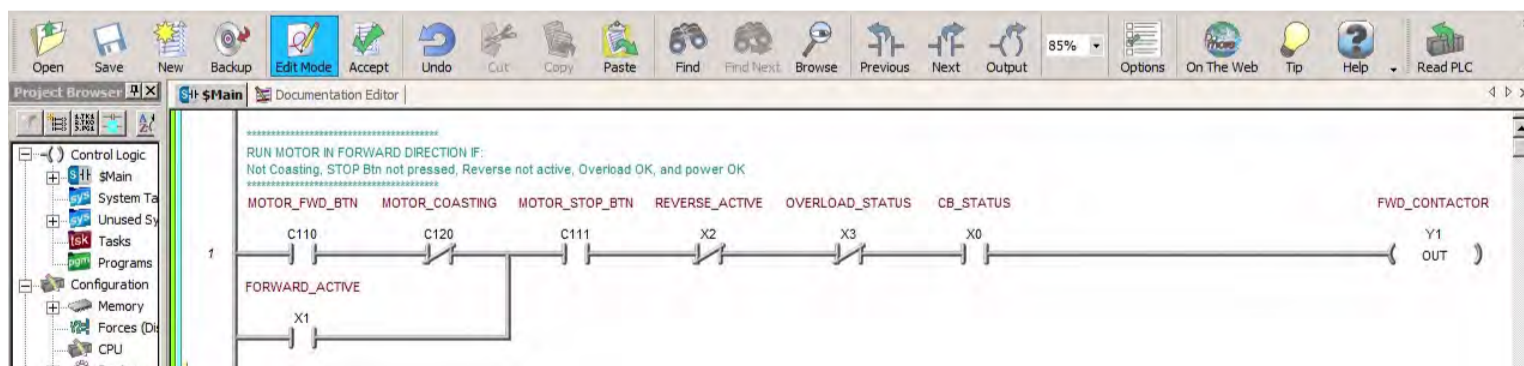
Learn More…

Learn more about AC motor control in this **series of videos**, from a simple switch to full reversing contactor configurations; selecting and using contactors and associated circuit breakers, overload protectors, and surge protectors; see typical hardware components and ladder logic programming for motor control with CLICK, Do-more and Productivity series PLCs.

Visit: www.AutomationDirect.com/Cookbook

Temperature Sensing

From curing ovens to milk pasteurization, temperature sensing and control has become an important part of automation. Two of the most commonly used temperature sensors in the industry are the Resistance Temperature Detector (RTD) and the thermocouple.

Resistance Temperature Detectors

RTDs are devices with an internal resistance that changes with temperature in a predictable, linear way. They are typically made from a very fine wire wrapped around a ceramic or glass core, or they can be created using thin film technology. Traditionally, platinum, copper, or nickel wire is used and is wound to achieve a specific resistance value. This resistance value will change as temperature does and by supplying a constant current, the measured voltage drop across the resistor can be used to determine the new resistance, and thus the temperature.

RTDs come in a variety of types, with the most common being a Pt100. It's made from platinum that has been calibrated to be 100 ohms at 0 degrees C. Platinum is an ideal metal for RTDs because of its stability, resistance to corrosion, and higher melting point. RTDs are great for applications up to around 600 degrees Fahrenheit. They provide accurate, repeatable results over the long term, but they are also fragile, have a long response time and are bigger than thermocouples.

Thermocouples

For wider temperature ranges up to a couple thousand degrees, a thermocouple is a better fit. **Thermocouples** work based on the Seebeck effect, a phenomenon where a small voltage is produced across a junction of two dissimilar metals when exposed to a temperature gradient. This is believed to be a result of heated electrons in the warmer metal traveling toward the colder area because of their thermal energy.

The amount of voltage produced within the thermocouple is very small, usually millivolts, and is directly related to the difference in temperature across the junction. There are numerous types of thermocouples and each is represented by a letter. The most common ones are J, K and T, which designate a specific temperature range and the wires of each type are color coded to identify them.

Most PLCs today have I/O modules designed to accept RTD or thermocouple signals directly. If using a temperature transmitter, I/O modules that accept 4-20 mA or 0-10V are often used to bring the signals into the controller.

Temperature Sensing + Ladder Logic

For temperature sensing using Ladder Logic, we can look back at an earlier example. Remember our freezer full of frozen fish from the Basic Instructions in Ladder Logic chapter? If not, no biggie, we were just using a temperature sensor to monitor the temperature of our fish freezer. To do so, we used a CLICK PLC and a compare contact. The temperature sensor's output produces an analog signal that is wired into the thermocouple module of the PLC. The CPU will convert this raw signal into degrees Fahrenheit and store it into DF16. Using the Compare contact instruction below, we compare our actual Freezer Temperature (DF16) to the constant 32, which represents 32 degrees Fahrenheit.
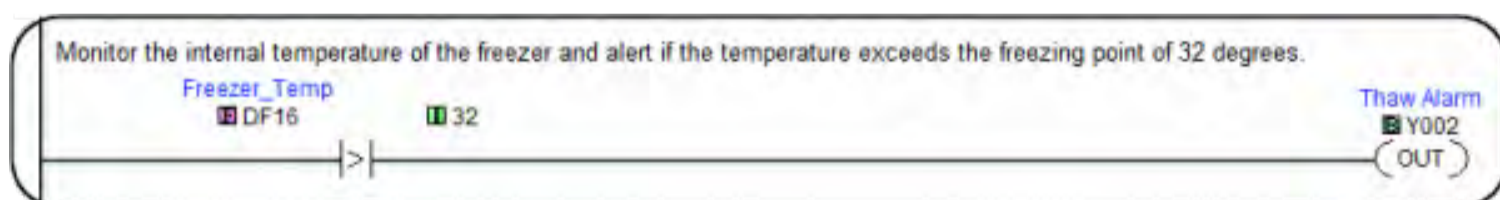
An Out coil is then programmed from our Compare contact and will provide the signal that is wired to the external Alarm Horn. If the temperature of the freezer reads greater than 32 degrees, an alarm will sound to alert us that the frozen fish will soon thaw and spoil.

Learn More…

Learn more about types of temperature sensors and how they work in this **series of videos**; hardware review of temperature probe styles and transmitters; how to select the right temperature sensor for your application; see typical ladder logic programming for temperature sensing with CLICK, Do-more and Productivity series PLCs.
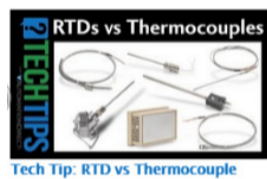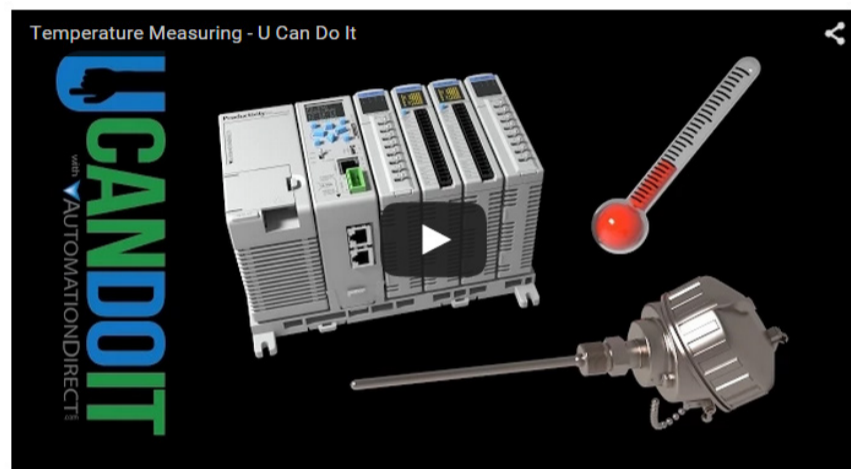
Visit: **www.AutomationDirect.com/Cookbook**

Automation Cookbook

Automation Cookbook is a collection of short video chapters covering a wide variety of automation application topics. Each chapter starts with a "U can do it!" overview, which explains the capabilities and usefulness of a specific automation task. There is also a group of "Tech Tips" videos in each chapter with technical nuggets and the gotcha's that can slow down experts and novices alike when specifying and implementing the topic in question. The practical "How To" videos show step-by-step demonstrations of exactly how to connect and troubleshoot, along with sample programming shown for the basics of the task. The Cookbook includes chapters on Temperature Sensing, Object Detection, A/C Motors (On/Off Control), Distance Measuring, Ethernet, Linear Motion and more.

Visit: **www.AutomationDirect.com/Cookbook**

Ladder Logic vs Object-Oriented Programming

Understanding Object-Oriented Programming (OOP)

OOP has a history longer than ladder logic, appearing in the early 1960s or before. Current hardware and software available in the "maker" world is adding to its popularity. Leading microcontrollers, such as Arduino and Raspberry Pi, are examples (Figure 5A). Arduino microcontroller hardware typically uses a basic form of C programming. Raspberry Pi is a Linux-based system, with many OOP programming platforms, such as Python and Java. These low-cost microcontrollers and Microsoft's .NET platform have made OOP easily available to those who want to learn it.
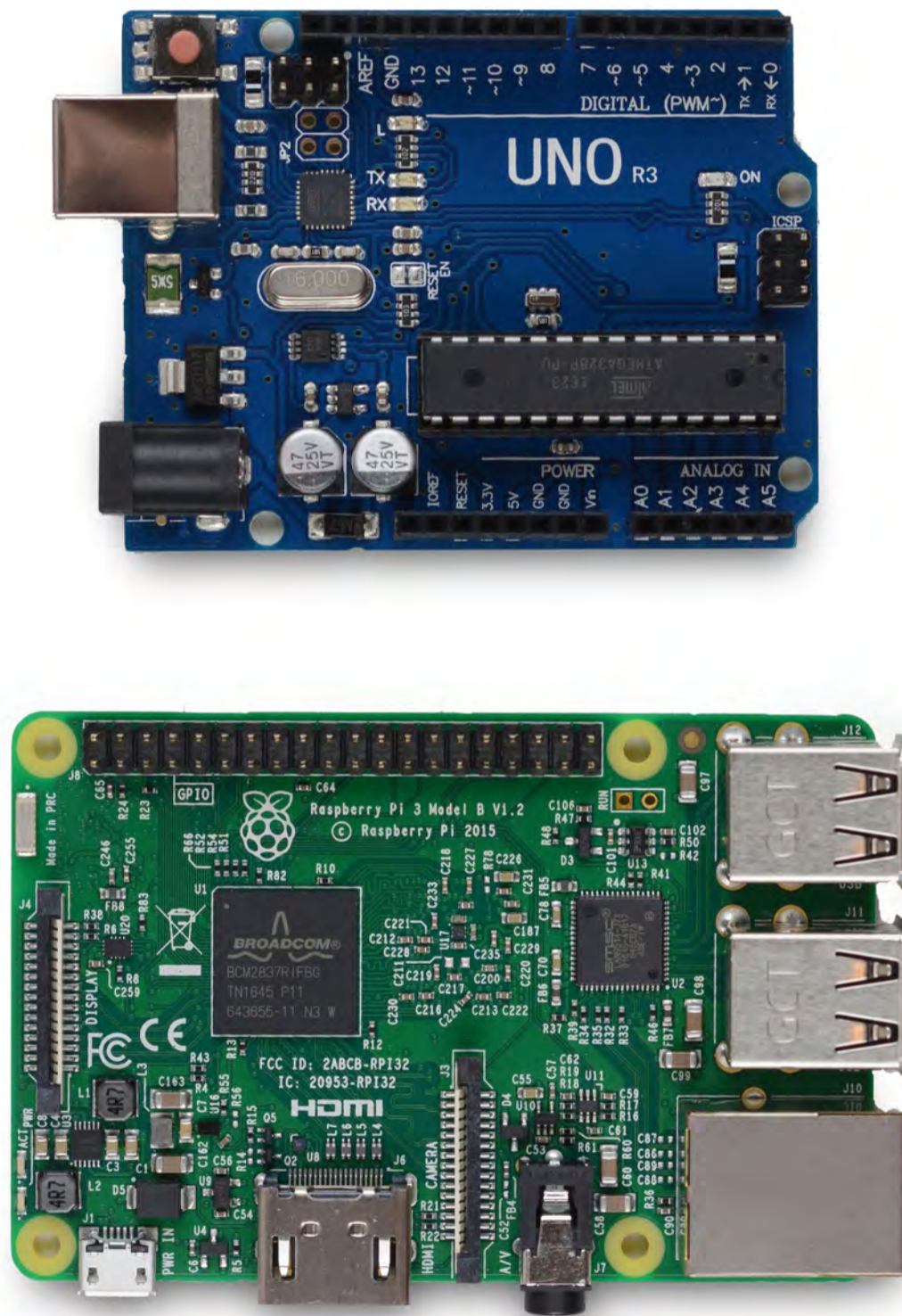
Figure 5A: Arduino microcontroller hardware is usually programmed with an object-oriented language, typically a basic form of C language.

OOP Advantages

Compared to ladder logic, OOP:

- Code is portable and easy to reuse
- Is easy to work with math, loops, etc.
- Is taught in nearly every computer programming course
- Code can be run on various hardware platforms

OOP uses objects, and it's necessary to understand the concepts of objects and their use to grasp OOP. Once an object or modular class has been written, it is easy to reuse and call it multiple times. For example, when an object is created to control a motor—it takes care of all inputs, outputs, and faults. This single object can be used multiple times as instances when several motors need to be controlled. It's called when needed and creates an instance when used. Each instance, for each motor, has its own characteristics such as motor off, motor running, motor RPM, motor overload, etc. Much of the programming work is done when the object is first created. That's a different way of thinking than ladder logic, and it is more powerful because once an object is built, it's easy to use and reuse. OOP makes it much easier to do complex math, loop computations, arrays, and nested subroutines. It is also taught in nearly every computer programming course in high school, college, and online. The code created is portable and can be run on various hardware platforms.

OOP Drawbacks

Compared to ladder logic, OOP drawbacks are:

- Higher overhead
- Steeper learning curve
- Not as easy to troubleshoot for maintenance personnel
- Usually must be compiled so source code can be uploaded to the processor

OOP can have higher overhead than ladder logic because it tends to need more memory and processor power. The learning curve for an OOP programming language can be longer. Significant time, practice, testing and application, likely in a classroom, may be needed to grasp concepts. OOP must often be closely studied by a programmer to follow the code, with a tracer or debugger needed to follow the logic. With this type of high-level programming, it can also be difficult to perform monitoring in real-time. The source code must be compiled before it is downloaded to the controller. Often, the source code is not present in the processor memory. This means the source code must be carefully backed up because the compiled code usually is not editable. With OOP, libraries need to be connected to other resources used during program compiling. If the connections and resources are not understood, it will be difficult to get the program to run.

Learning Ladder Logic

Ladder logic is a very simplistic and self-documenting method of coding, and some even debate whether it is even a programming language. It was the only industrial-strength programming language for decades in the machine automation arena, and it's still the main programming language used in the automation industry (Figure 5B). Over time, as different people enter the industry from different areas and backgrounds, different languages have been added to the industrial automation toolbox. These include function block programming, structured text, state programming and sequential function chart. These four languages and ladder logic constitute the IEC 61131-3 standard for programming languages from the International Electrotechnical Commission (IEC). The idea behind IEC 61131 is that if each vendor follows the standard, at least to some extent, a person can learn the five languages once, and then easily move among different platforms from different vendors. However, the reality is different (see the following section for a comparison of **Ladder Logic and the IEC standard**).

Basic ladder logic—such as the use of relay contacts and coils—is the same, but the syntax and the user experience when programming, and the details of how to use the programming platform, must be learned for each vendor. Despite this lack of standardization, ladder logic advantages over OOP include:

- Well suited for both machine and process control
- Easy to understand because it's naturally self-documenting
- Easy to use for troubleshooting the controlled system
- Easy to debug
- Source code that can often be stored in the processor

Ladder logic is well suited for both machine and process control, particularly for automating systems with lots of discrete I/O. Ladder logic has been modified over the years to also handle analog I/O, making it a good fit for many process control applications, which tend to have a higher proportion of analog I/O as compared to machine control applications. A wide range of skilled, technical, and engineering personnel can learn ladder logic quickly because it's easier to use than OOP. The logic is very systematic and orderly, making it simple to follow and understand due to its self-documenting nature. Each line of code must be true before it turns something on. If there are five motors, there are typically at least five lines of code, providing a high degree of simplification.
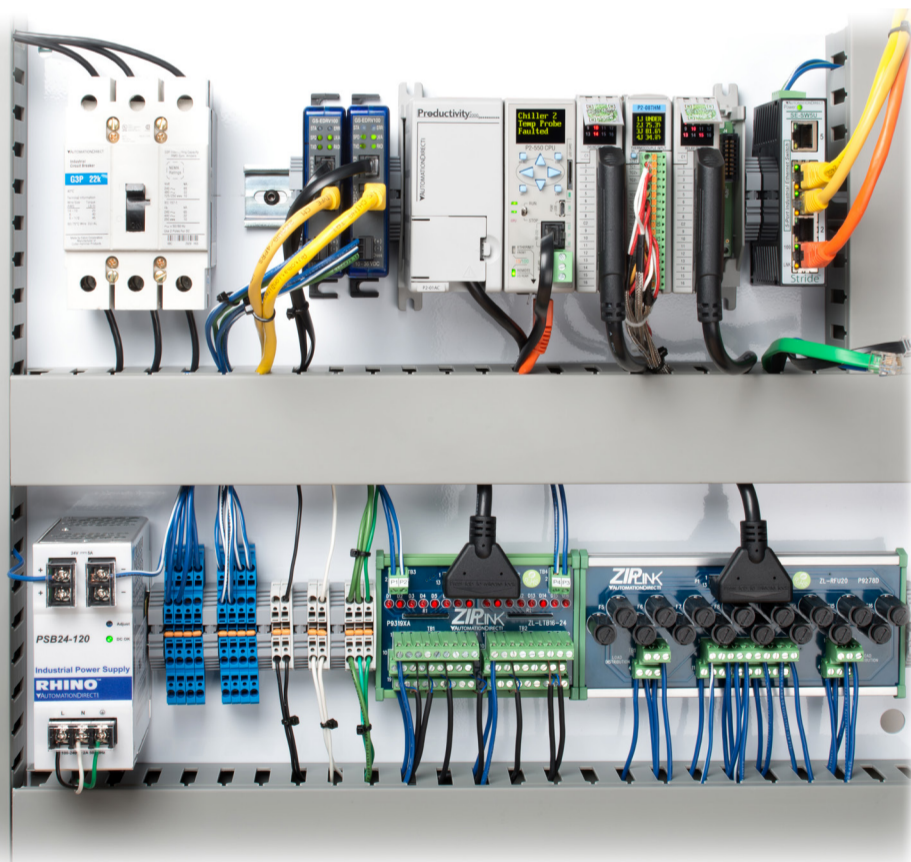
Figure 5B: The Productivity2000 PLC provides an industrial strength hardware platform for executing ladder logic.

Easier to Learn

Ladder logic is intuitive for electrical engineers and maintenance personnel. While ladder logic requires a different mindset than OOP, it can be learned quickly, and it takes less time to understand code written by others. It is very clear when the logic is true.

Contacts on or off, energized coils, comparing variables and common math functions are easy to follow—even by someone with limited programming experience (Figure 5C). Its ease-of-use simplifies troubleshooting and debugging as well. When monitoring the logic, it is straightforward and easy to see what is happening. No software degree or advanced programming skills are necessary. With ladder logic, it's easy for maintenance and engineering personnel to follow the flow and understand what is going on. As discussed previously, ladder logic can be thought of as a truth table. If the logic on the left is true, the logic on the right is solved or turned on. Ladder logic source code and descriptors are often stored in the controller. This can eliminate the frustration a programmer experiences when trying to understand a compiled program without access to the source code, a common issue with OOP.

However, when compared to OOP, ladder logic can:

- Be unfamiliar to computer programmers, IT, etc.
- Be difficult for math, text, and data handling
- Be dependent on scan time
- Require specialized hardware to execute, such as a PLC.

Figure 5C: The intuitive nature of ladder logic programming makes it easier to comprehend even for those with limited experience.

Ladder logic is a symbolic language, unfamiliar to computer programmers and IT personnel because it's not something they typically learn in school. Handling math, text strings, and data in ladder logic can be difficult because ladder logic was not originally conceived to deal with these functions. Ladder logic is scan time dependent. Larger programs take longer to scan and solve the logic. With ladder logic, the inputs are read, the logic is scanned, the data table and outputs are updated, the overhead communication is performed—and then the cycle repeats. Functions such as interrupts and other programming techniques can be implemented to ensure certain rungs are executed quicker. Although software-based PLCs with ladder logic can run on a PC, often the hardware, such as a PLC, is matched to the programming software, with both purchased from the same vendor. This ensures compatibility but hinders switching vendors.

In addition to comparing pros and cons of ladder logic and OOP, users should evaluate the existing use of these languages within the environment where they will be deployed. If the plant or facility has standardized on ladder logic, the use of OOP will be discouraged, even if it's a better fit for the application. But OOP use is growing, and it's expected to co-exist with ladder logic for decades to come, so it would behoove an aspiring automation professional to learn both languages.

Ladder Logic and the IEC 61131 Standard

Like many standards, IEC 61131 has its uses, but shouldn't be accepted verbatim. IEC 61131 covers five programming languages: ladder, function block, instruction list, structured text and sequential function chart—but for most applications, ladder logic will suffice, with function block added in only as required. The other three languages are generally not necessary for programming machines and most processes, as explained in a column titled "Keep it simple with ladder diagram" from the Control Design web site.

Here are some of the reasons why ladder logic, or ladder diagram as the author refers to it, is often the best choice for a wide range of applications.

"Although personal preference plays a role, using just ladder diagram programming for machine sequence control keeps things simple for an integrator's controls department and the customer's support and maintenance department, and that's what most machine builders want. The quicker the customer can support the equipment, the better. Sticking with ladder diagram helps, as the majority of programmers have little need or awareness of the IEC 61131 specification even though it has been around for more than 20 years.

Although the IEC 61131 standard is useful and has its place in more complex applications, most programmers do not save time or money by using multiple programming languages. Ladder diagram and maybe some function block programming is all that is needed. Mixing several different languages in a single program is less efficient from a programming and support standpoint.

It adds more time and costs during development, testing and maintenance of the program. It can also make it more difficult for the next guy to figure out the program when using multiple programming methods. The syntax is different between each type of program and must be learned before it can be supported, and some are just a waste of time."

If ladder logic is best for most applications, what are the main issues with the other languages? According to the author, "Sorry instruction list programming—the days of low-level programming are over, and I cannot think of anything but the most limited applications where you would want to write single instructions in a line-by-line format."

Instruction list programming is similar to many other lower-level computer languages such as Fortran, with many lines of code required to perform even the simplest function.

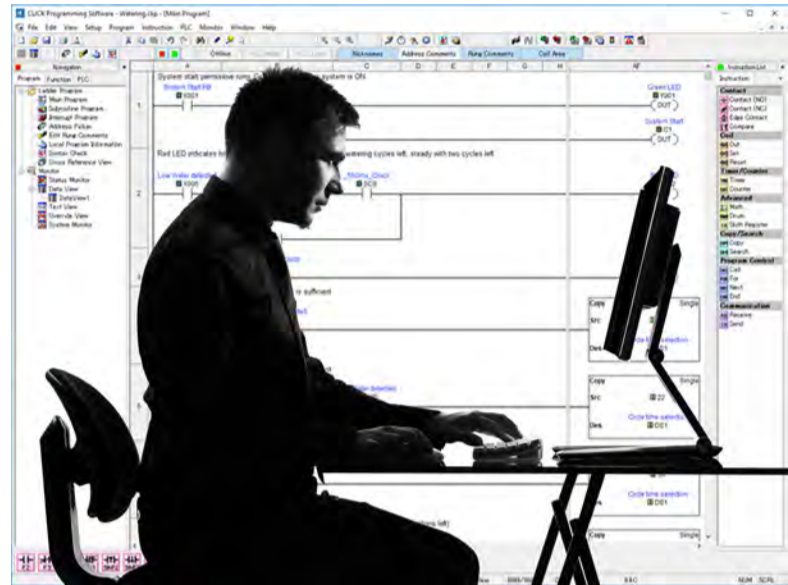**Instruction List:**
START:
LD X0
ANDN X1
ST C45

LD C45
OR X3
**ADD (5, PRESENT, RESULT)**
JMP END

The advantage of these lower-level languages, and the main reason for their use in the past, is their very efficient use of computer resources such as processing power and memory. But with today's powerful PLCs capable of easily handling execution of ladder logic and function block, the main reason for using instruction list is no longer valid.

How about structured text and sequential function chart?

Structured Text

```
INDEX := 1;
IF STOP OR ESTOP THEN
RUN := FALSE;
END_IF;
WHILE NOT(RUN) AND (INDEX <= 6) DO
MACHINE [INDEX] PERM := FALSE;
INDEX := INDEX + 1;
END_WHILE;
```

Sequential Function Chart
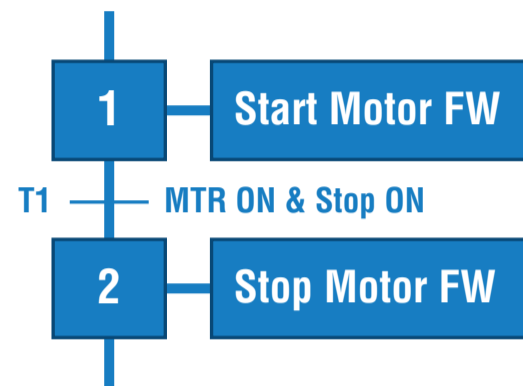
"Structured text has its uses for data handling, string functions and math, among others, but most can be done using ladder. It's also good for iteration loops, but doing it in ladder eliminates the need to understand another syntax and tie two different programs together.

The sequential function chart structure may help to define process or machine sequence flow, but each block in the chart has additional programming behind it that must be managed within the program. There are too many bits and pieces. Just reset a sequence in the middle, and there will be many small sections of logic that will need to be cleaned up, reset and initialized, and the chart structure hides it nicely."

Instead of spending time learning how to use all of the IEC 61131 languages—system integrators, machine builders and end users would be better served by creating internal standards for ladder logic programming to ensure consistency and adherence to best practices.
"A how-to standard can better help the average PLC programmer, technician and maintenance worker by creating well-thought-out ladder diagram programming techniques, as opposed to defining what ladder diagram, function block, structured text and instruction list programming languages and sequential function chart structure are. The programming environment typically doesn't need the 'what'—at least not in the United States—they need the 'how.' The what has been done already.

Often the reason software costs so much to develop and maintain is that the program is poorly written. Most often, I see what I call scatter code—the incredible wandering PLC program where rungs of logic are just randomly placed with no thought to purpose, function or organization. Focusing on proper programming structure and techniques is what is most important.

It's not the complexity of the sequence or process as much as it is the poorly written program that makes it difficult to support. It's really the programmer who sets the program architecture, and this is where improvements can really be made. A well-written and simple ladder-diagram program includes sensor back checking, mode and cycle control, step sequences, output logic and fault logic at a minimum. Keeping these functional blocks of code separate is a good starting point"

When internal standards are created and followed, the result is a program which is easy to follow and modify, whether by the original programmer or someone else years down the road.

"The key here is the ladder-diagram step sequence. When a program is written, started up and run for a year before anyone touches it again, there is nothing better than ladder diagram and a well-written and documented step sequence to quickly understand program function. Reading through the coil descriptors in the ladder-diagram step sequence should read like a sequence of operation. The steps define the operation chronologically and display the condition of the logic graphically.

Timer starts when Start PB is pressed. It will time for 12 hours then turn on T2. T2 is used to set the C2 Water On bit and also reset the timer on the next scan.

Once 12HR Water Timer has reached 12 hours, set the Water On bit so timer can be reset

When 12th hour is detected and Water On is set, allow watering for XX minutes depending on selector switch

At 12th hour, turn on pump for as long as T1 allows. Pump test button allows for testing pump system if sufficient water is in tank

Reset the Water On bit when watering cycle is complete
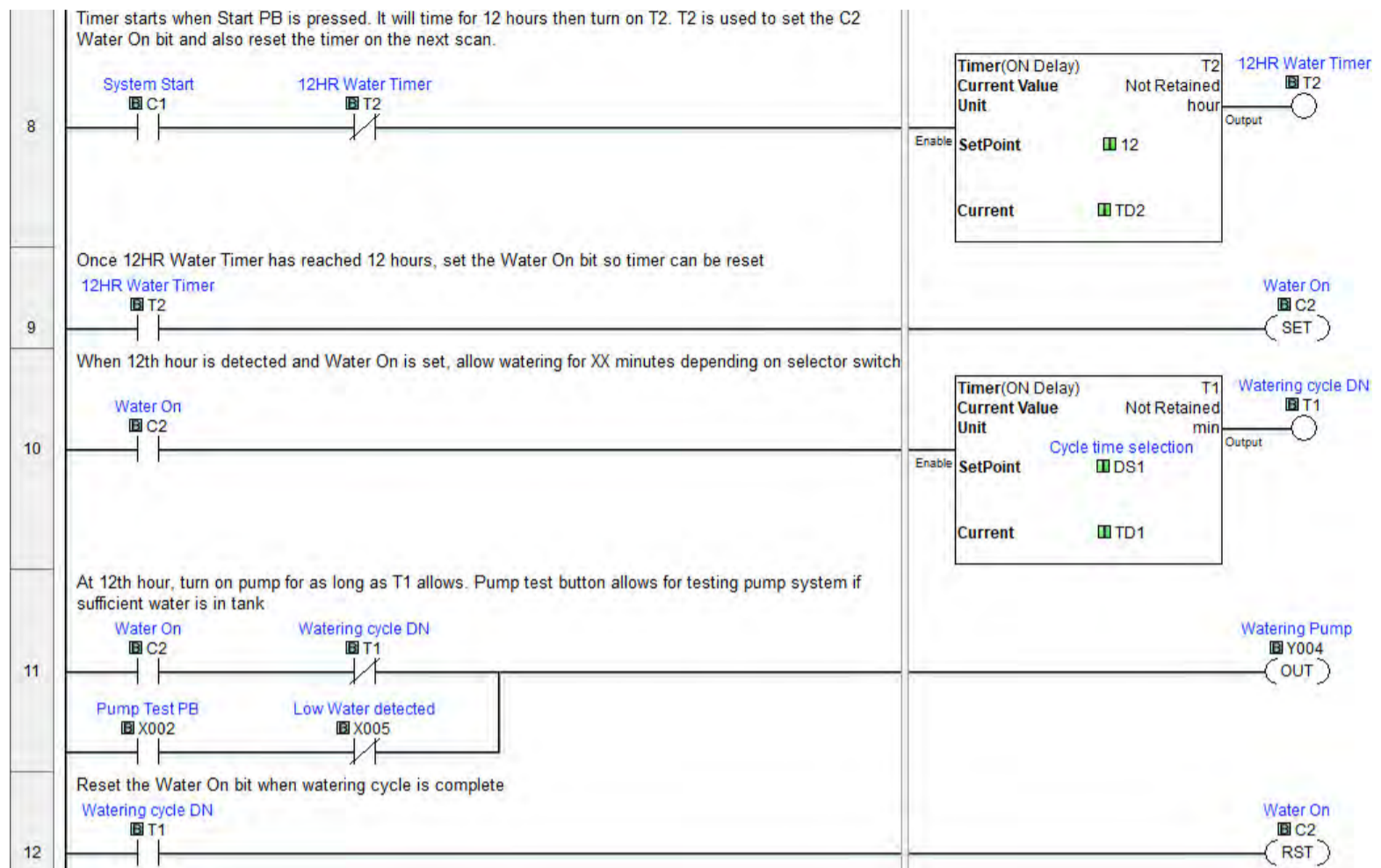
A well-documented ladder diagram can read like a sequence of operation bringing those unfamiliar with the program up to speed quickly

When there is a problem or an enhancement needed, the new guy can get up to speed quickly and not have to wade through multiple programming languages. If you don't do it every day, simpler is better. Opening a program and finding the main step-sequence ladder diagram, the programmer, technician or maintenance personnel can understand what's going on with the equipment with little study or review needed."

While its good to be aware of standards, they must be applied in a way that makes sense for your company, your customers and your applications. This is especially true with the IEC 61131 standard, where intelligent use of ladder logic, supplemented by function block only as needed, will result in a program which is simple to create, understand and modify.

PLC Communications

Coming of Age

PLC communications has grown and changed over the many years since the controller's inception decades ago. Industrial needs and technological advances have kept PLC communications evolving, and this section will explore some past and present communication standards, and what the future may hold.

PLC Partners in Communications

PLCs need to communicate to a number of different devices. First, there are devices which are part and parcel of the PLC, such as remote I/O. Next, are other PLCs and control devices, such as motor drives and servo controllers. Communication with some type of operator interface device is also often required, as is Internet connectivity. Finally, PLCs often communicate to server level PCs running various manufacturing related applications.

All of these communications require two things:

- a physical connection or layer, which is the wiring and connection components, and
- a shared protocol, which is the common language allowing each device to understand what the bits and bytes in the communication messages mean.

Back in its infancy, PLC communications were typically proprietary, with each supplier having their own closed connections and protocol. This made communications relatively simple among a single supplier's products, but devilishly complex if you chose to venture outside the realm.

Proprietary Has Its Place

These old, proprietary communication methods still have their place, as they allow quick and easy connections among a single supplier's products. For simple systems with no plans to expand or connect to other components, proprietary communication is an option and will continue to be for quite some time.

Many PLC suppliers still support these older physical, wired connections—most often based on RS-232C, RS-422 and RS-485 connectivity—and the related proprietary communication protocols.

DirectNET

For example, some AutomationDirect PLCs use a serial RS-232C or RS-422 physical layer network with the DirectNET protocol for communications. This protocol can be used for PLC-to-PLC communication over a point-to-point or multipoint network using standard PLC instructions, or to talk to AutomationDirect's C-more HMIs.

When the control system requires venturing outside a single supplier, communication standards come to the forefront, starting at the physical layer.

Physical Layer is both Mechanical and Electrical

Many times the physical layer is confused with the physical medium such as the cable, connectors, network interface cards and wireless transmission hardware. It's not physical in that sense, although the physical layer does define the required mechanical and electrical interface to the physical medium.

The physical layer defines how to connect the upper data link layer in the Open Systems Interconnection (OSI) communications model within a computer to physical devices. It is basically the hardware requirements with schematics and specifications for successful bit-level communication to different devices. The physical layer not only defines items such as bit rates, transmission of electrical, light or radio signals, and flow control in asynchronous serial communication, it also defines types of cables and the shape of connectors. For a further explanation of the OSI 7-layer model go to this **section**.
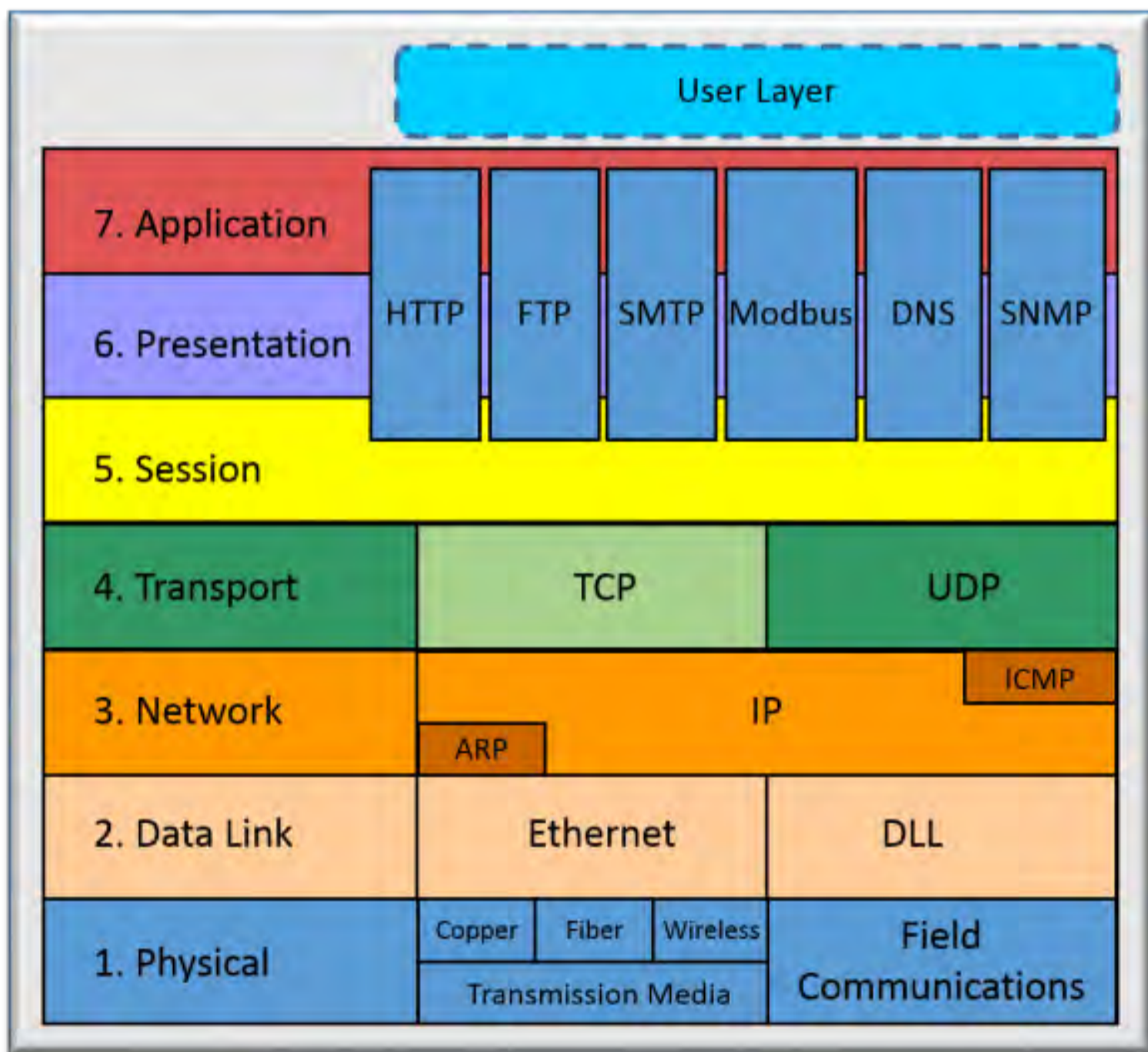
OSI model builds a protocol in layers

Once the physical layer's hardware requirements are defined, some examples of protocols that can reside within different physical layers include:

- Proprietary protocol for a barcode reader using an RS-232 point-to-point connection
- Motor drives protocol connecting multiple devices on an RS-422/485 multi-drop connection
- Printer protocol over a parallel interface
- Ethernet for connection to a plant or control network
- Universal Serial Bus (USB) for connection to keyboard
- Bluetooth for connection to a wireless microphone

RS-232, RS-422/485 and parallel interfaces have been around for decades and are still in widespread use. But Ethernet, USB and Bluetooth are the physical layer protocols most commonly specified in new products and applications, and represent the future direction of PLC communications.

The physical media and connections are affected by the type of physical layer chosen as it defines the cable type, topology, number of devices, maximum transmission speed, and maximum cable distance.

Protocols Ride on the Physical Layer

A protocol is a set of rules for communication among networked devices. Some common protocols used in the industrial arena include:

Modbus RTU

EtherNet/IP

Ethernet TCP/IP

Modbus TCP/IP

Profinet

Perhaps the most common industrial serial communication protocol is Modbus RTU, developed by Modicon, which usually runs on an RS-485 network. This and other popular serial protocols are supported by a wide variety of suppliers, and are very familiar to a wide group of automation professionals. But performance is limited, making serial protocols a poor choice for high-speed and other demanding applications.

Because of performance and other advantages, Ethernet has emerged as the dominant standard for the physical layer of many industrial protocols such as EtherNet/IP, Ethernet TCP/IP, Modbus TCP/IP and Profinet. And unlike serial protocols, multiple Ethernet protocols can run on the same Ethernet physical layer.

Using Ethernet, it's not too difficult to interconnect several devices such as PLCs, HMIs, field I/O and valve banks. Plus, the communication remains fast while talking to several dissimilar devices on the same cable, due to the very high speed of Ethernet as compared to older serial networks.

Here's an example of a typical Ethernet network connecting a wide variety of control systems and other components:
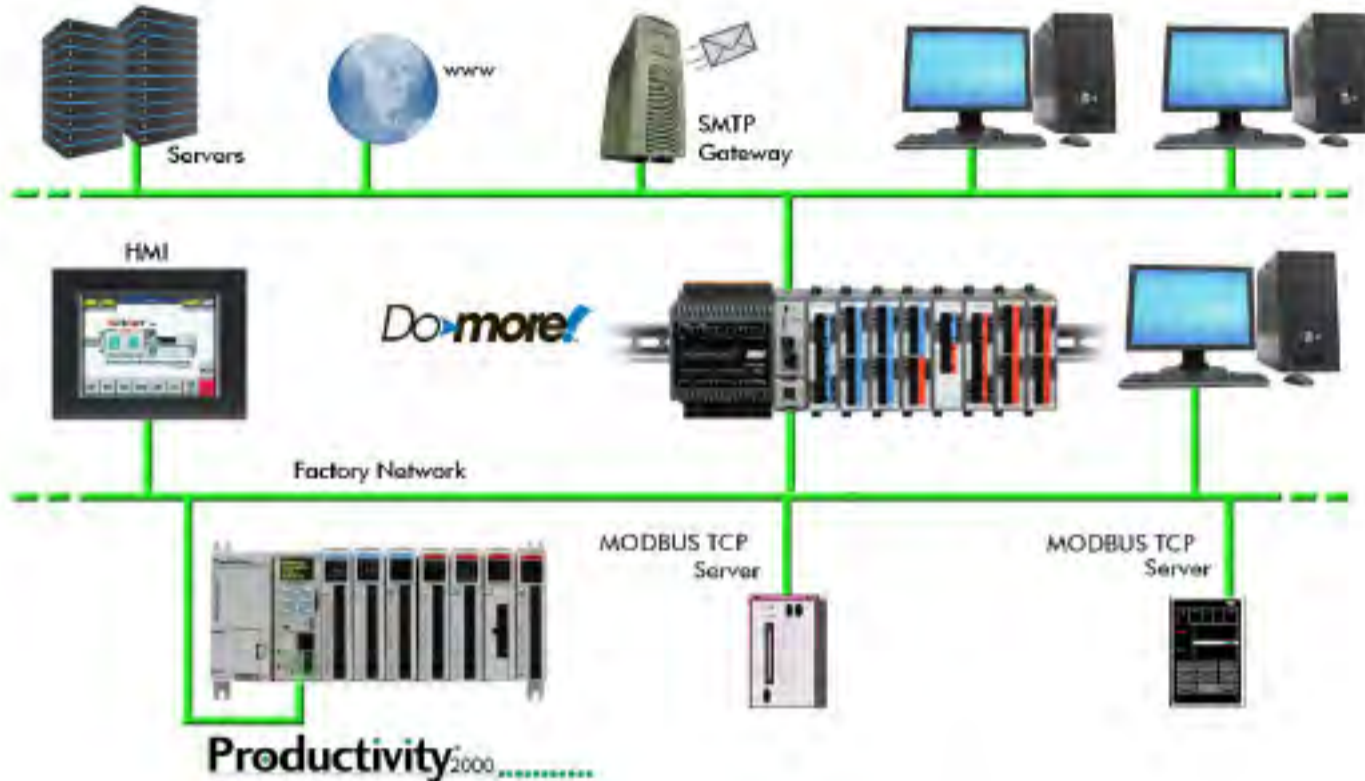
Ethernet Supports Multiple Protocols

Today's physical, wired layer is moving to Ethernet for most control system communications. Wiring, connectors and managed or unmanaged switches are now standard—with a wide variety of devices capable of connecting to the Ethernet hardware and industry-standard networking technologies.

Most users just need to interact with the physical Ethernet layer, plugging in the cables and letting the protocols worry about error-free communication.

Popular protocols like EtherNet/IP and a common network infrastructure allow consistent data and communication from the HMI and desktop PCs at the information level—through the PLCs, HMIs, servo controllers and drives at the control level—to the device level distributed I/O and field components.

And, of course, Ethernet is the portal to the Internet which allows a PLC to communicate worldwide. Some PLCs have built-in web server capability, allowing access to it from any web browser.

The prevalence of Ethernet communication in industrial automation could result in even more standardized communication protocols in the near future. But for now, if your system is simple, a proprietary protocol may be the best solution. When requirements grow in terms of speed, connectivity or expansion, an Ethernet protocol may be the way to go.

Ethernet Basics

Ethernet was first conceived and deployed in a research setting in the 1970s and was soon standardized in the early 1980s under IEEE 802.3. This quickly led to commercial adoption in office environments, usually to network business PCs. Since its evolution, Ethernet has undergone many upgrades and improvements, making it the world's leading wired network technology not only in offices, but also in industrial plants and facilities. The use of Ethernet in these industrial environments will be the focus of this section, although much of the information is also applicable to commercial Ethernet installations.

Ethernet Everywhere

Industrial sites today commonly use Ethernet to connect PCs, programmable logic controllers (PLCs), remote I/O, human machine interfaces (HMIs), smart instruments, motor drives and other intelligent automation components (see Figure 6A). PCs are commonly used to host HMI, asset management, enterprise resource planning, production planning and other applications.

Exponential speed improvements have been a major component of Ethernet's evolution. There are currently three versions of Ethernet: 10BaseT, 100BaseTX (also known as Fast Ethernet) and 1000BaseT (also known as Gigabit Ethernet because it transmits Ethernet frames at a rate of one gigabit per second). While some details vary among these versions, the most important differences are speed and cost, with cost increasing along with speed. While consumer hardware and PCs typically operate at Gigabit speeds, most modern industrial Ethernet installations and components use Fast Ethernet.

Figure 6A: In an industrial environment, Ethernet is used to network PCs, controllers and other intelligent industrial automation components. Lower level networks are often used to connect controllers to field instruments and sensors.

What are the Different Types of Ethernet Cabling

The first point to consider when deploying an Ethernet network is the type of cabling. Cat5e cable supports Gigabit speeds and is often the minimum specification recommended. Also available is Cat6, providing better performance at a higher cost, and also providing a degree of future-prooffng as Ethernet moves toward 10 Gigabit speeds in coming years. Although commercial installations use unshielded cables, shielded cable should be used in industrial environments to protect from electrical noise generated by motors, motor drives, welding, and other electrical equipment.

In years past, it was important to define whether cables should use straight-through patch or crossover connections (Figure 6B). Crossover cabling was typically required when directly connecting components to each other, and patch cabling was used when connecting components to Ethernet IT devices such as switches and routers.

However, most devices today support Auto MDI/MDIX which will accept either configuration, so most installations use patch cabling throughout. As shown in Figure 6B, RJ-45 connectors are used for most Ethernet connections.

Another point to consider for both performance and compatibility is communication speed, and half or full duplex capability. While duplex compatibility can be overcome by using a managed switch or router — discussed in more detail later—performance is limited to the slowest device.

As previously mentioned, Fast Ethernet is used in most industrial automation applications because its raw speed is more than sufficient, but this can be affected by the selection of half or full duplex. With half duplex, a device can only transmit or receive, but can't do both simultaneously. This reduces performance, and in the worst case half duplex devices drop packets under heavy load due to data collisions.

Most components and devices used in Fast and Gigabit Ethernet installations therefore use full duplex, meaning the node can transmit and receive simultaneously.
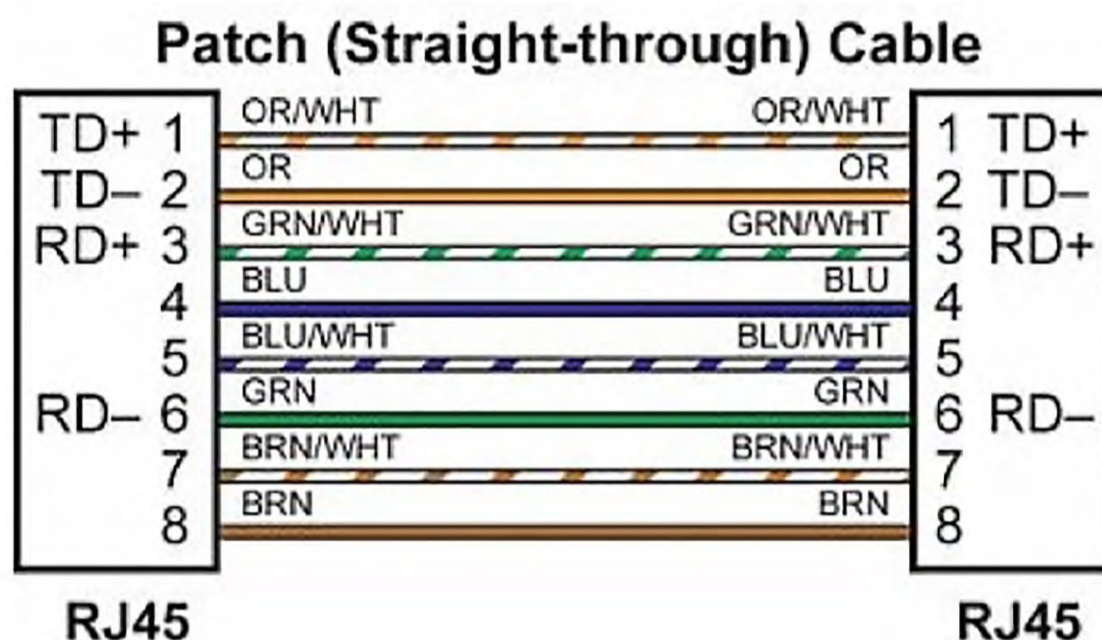
Figure 6B: Basic wiring standards must be considered when connecting components and devices to each other in an Ethernet network.

What are MAC Addresses and IP Addresses?

The media access control (MAC) address is the hard, physical address of a component or device. The address is set during manufacturing and should never change. Although there are exceptions to this, hard coding the MAC address was the original intent of the design. An Ethernet packet cannot enter a component or device without this address. However, it's rare that a communication setup or configuration will require entry of this address because another protocol, called address resolution protocol (ARP), usually automatically retrieves it and correlates it to an Internet Protocol (IP) address.

The IP address is the logical address of a component or device, used to identify its address and network. An IP address, such as 192.168.070.001, includes two identifiers: the network address and the host address. The network address helps switches and routers determine where to send messages. The host address identifies the specific component or device on that network. The subnet mask determines which part of an IP address is the network address and which part is the host address.

The default gateway address and a router enable connections to other networks. A component or a device can only send and receive Ethernet messages to other components and devices on its network as determined by the IP address and subnet mask. If a component or device needs to communicate with a component or a device on another network, a router is required. In order to get the message to the other network, the components or device will send its message to the default gateway address, which is the IP address of the router.

Local area network (LAN) and wide area network (WAN) definitions vary greatly, but in general LANs are networks encompassing switches and hubs. Once a network is divided into groups using a router, these networks are considered to be encompassed within a WAN.

What are Ethernet Hubs, Switches and Routers?

There are a variety of Ethernet network topologies including bus, ring and star. Bus and ring are similar to a daisy-chain connection, with a single cable hopping between each device. Bus topology is a bit obsolete, while the newer ring topology adds fault tolerance. However, most industrial Ethernet installations use star topology, with some ring connections to reduce cable run distances or improve resilience.

With the star topology, many devices connect to the central access point, the switch. Ethernet switches come in both unmanaged and managed configurations. There are applications and situations that make sense for unmanaged switches, and there are others where a managed switch is a better choice.

An Ethernet hub simply connects Ethernet devices together with no regard for MAC addresses, so all messages are sent to all devices (Figure 6C). An unmanaged switch is a little smarter, using the MAC addresses to determine what components and devices are connected to each of its ports, and then routing messages targeted to those components and devices out the appropriate port, preventing the collisions often occurring with an Ethernet hub.
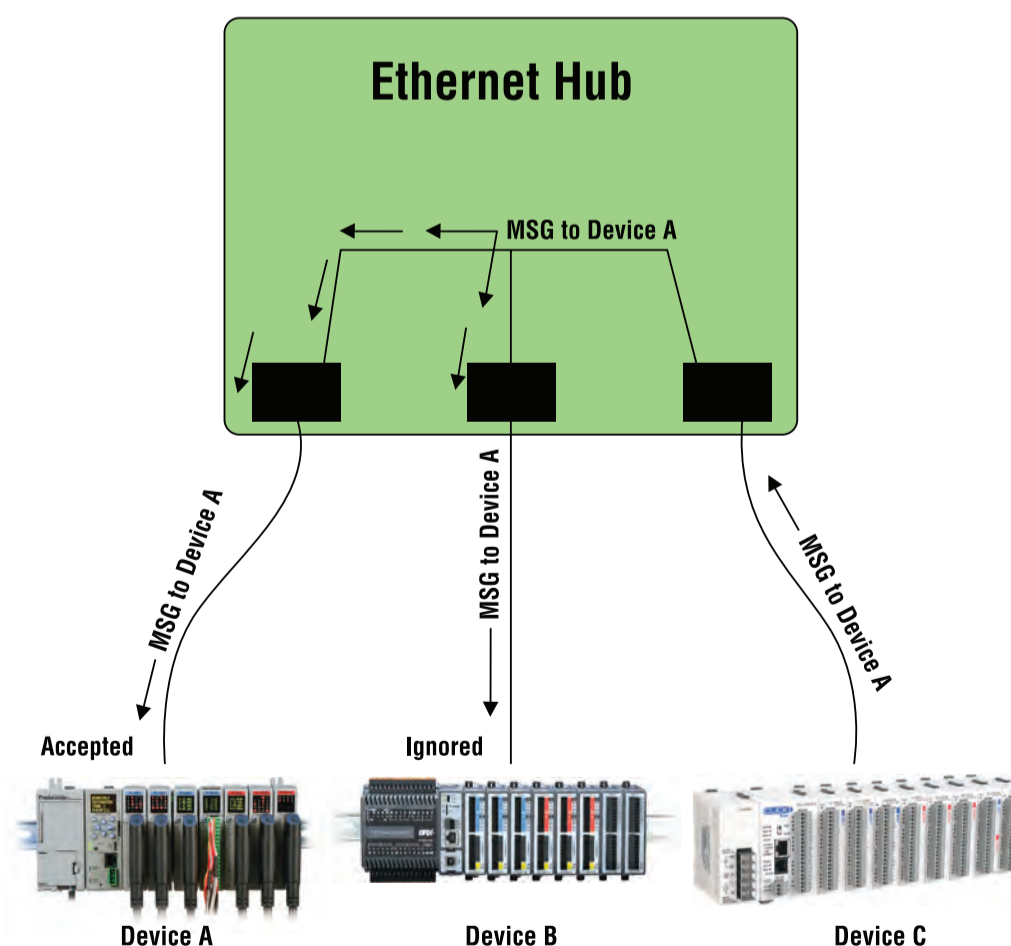
Figure 6C: Ethernet hubs are an older technology, sending all messages to all connected devices or components.
Switches are able to selectively route messages to appropriate ports, preventing collisions and improving network throughput.

A managed switch adds features to those offered by an unmanaged switch, with one of the most important being selection of speed and duplex mode. When connecting a component or device to a switch, it must auto negotiate to an agreed-upon speed and duplex mode.

Auto negotiation is common in commercial environments but can fail for industrial situations. In these applications, it is often better to turn off auto negotiation, and use a managed switch to set the speed and duplex mode to a known working setting for both sides. This can be particularly helpful when connecting together components and devices from different manufacturers.

For a simple network with five or fewer components or devices in a relatively small area, an unmanaged switch will usually work, and it will always be less expensive than its managed equivalent. For larger applications with more components and/or devices from several different manufacturers, a managed switch should be strongly considered. The benefits of a managed switch's configuration and other features will easily outweigh the cost premium over an unmanaged switch in most complex applications.

A router is a more powerful version of a managed switch, with the ability to create different groups of components and devices within an Ethernet network. Creating groups allows one part of a network to be separated from another, a useful feature for managing network traffic. A virtual private network (VPN) router should be used when connecting a network via the Internet to provide the required degree of cybersecurity.

What are Ethernet Protocols?

The discussion so far covered some of the physical or hardware aspects of industrial Ethernet, but this is only half the story, with the other half being the many different software protocols used in industrial Ethernet installations.

Most industrial Ethernet protocols can run on standard Ethernet hardware, but some require modified and proprietary versions of hardware to operate. This is usually done to improve performance in highly specialized applications.

Different industrial Ethernet protocols running on standard Ethernet hardware can all use the same network but can't communicate with each other. The two leading industrial Ethernet protocols are EtherNet/IP and Modbus TCP. A standard Ethernet network could in theory have both of these protocols running on it simultaneously, but in practice it's better to limit each network to one protocol.

To simplify, a protocol allows every device on the network to use a common language for communication. To understand this subject more deeply, it's necessary to look at the Open Systems Interconnection (OSI) 7 layer model for Ethernet communications.

What is the OSI 7 Layer Model?

Figure 6D shows the model divided into upper and lower layers. Most industrial Ethernet protocols will not use all 7 layers. Figure 6E shows how Modbus TCP fits into the model, using Layers 1-4, and Layer 7. The goal for protocols adhering to this model is for any network-connected device to be able to transmit a data packet or message to any other device.

As previously discussed, every Ethernet device carries a unique MAC address and must be assigned an IP address compatible with the local Ethernet network port it is connected to.

Messages initiate at a device connected to Physical Layer 1, which has already been discussed in terms of its electrical signals and cabling. The Data Link Layer 2 is the next one up, acting as a traffic cop in the form of Ethernet switches and other Layer 2 devices. These devices use the MAC address of each component or device to figure out which packets go to which ports, and strip off the MAC address in the process. Similarly, network layer 3 uses the IP address to route the data packets. Finally, transport layer 4 delivers the data payload, the actual Modbus message for Modbus TCP, up to layer 7.

Layer 7 is the Application Layer, and this is where a common language is important. Although other industrial Ethernet protocols may be identical to Modbus TCP in their use of layers 1-4, and can thus use the same network, they won't be able to communicate with each other because their application layer 7 is different.

Figure 6D: Most industrial Ethernet protocols don't use all 7 layers of the OSI 7 Layer Model.
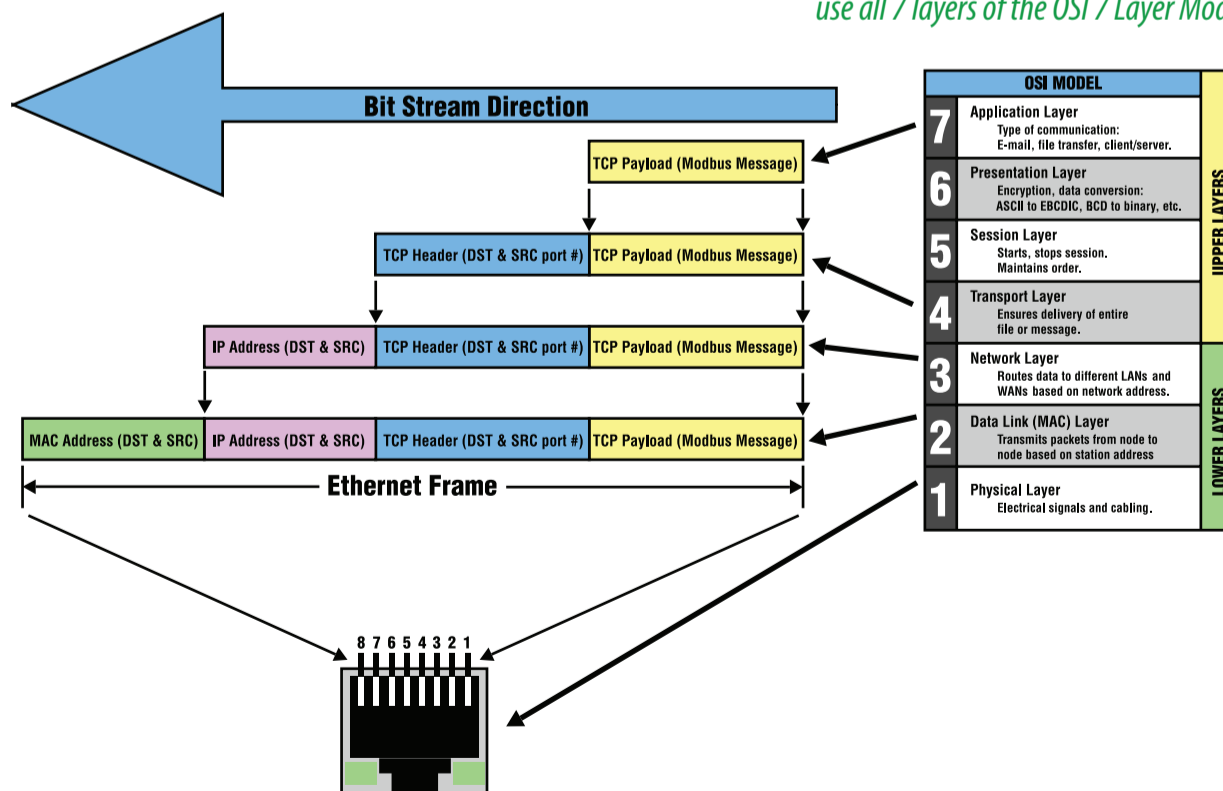
Figure 6E: This diagram shows how the popular industrial Ethernet protocol Modbus TCP fits into the model, using layers 1-4, and layer 7, but not layers 5 and 6.

What are Unicasting, Broadcasting and Multicasting?
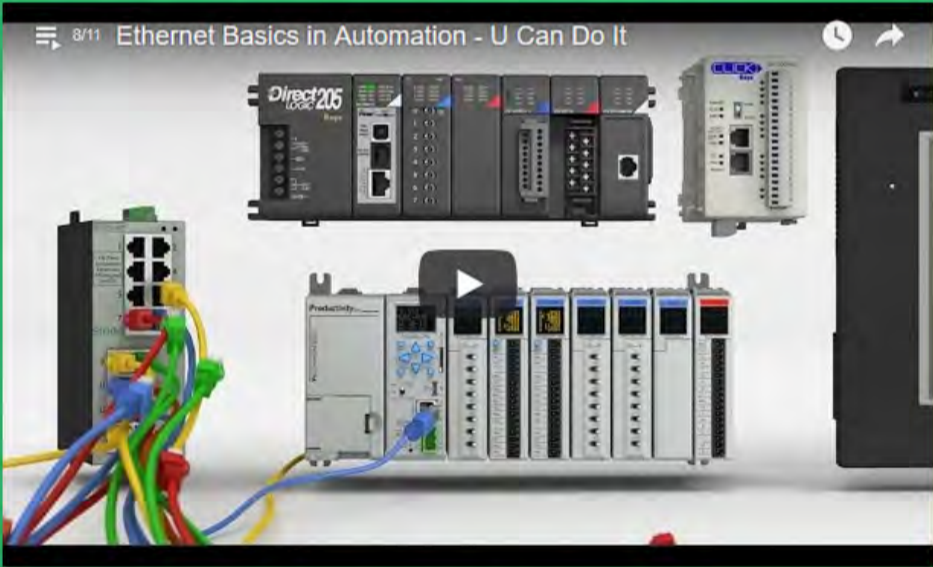
Unicast messages are sent from one specific device or component to another. Broadcast messages are sent from one device to all components or devices in a group. Multicast messages are sent to many different receivers at once, typically all components or devices on an Ethernet network. The groups associated with Ethernet networks are created by a router.

A router is needed to make sure a broadcast message only goes to members of a group, but managed switches can intelligently route multicast messages to the correct ports of all components or devices on a network. A managed switch will learn which ports should be receiving multicast messages, using a feature called IGMP Join, and only send messages to those ports, resulting in significant performance improvement.

Managed switches can also prevent a network from shutting down if someone accidently connects switches into a ring. If unmanaged switches are mistakenly connected together in a ring, then messages will continue to propagate, creating a broadcast storm which will overwhelm and shut down the network. Managed switches prevent this by shutting down one path of the ring. If something happens to the working path, the switch changes over to the other path, which also provides a level of redundancy.

Learn More...Watch The Video!
Ethernet Basics and to how to use Ethernet with AutomationDirect PLCs

Visit: http://n2adc.com/zvrxz

Ethernet Dos and Don'ts

If you are reading this article, it's very likely you have plugged in an RJ45 connector into a jack on your computer. This common Ethernet connector uses 8-pins to connect the 8-conductors in a CAT cable (8P8C). Check out the TIA/EIA-568-B standard for proper connections. RJ45 connectors are used in many industrial applications, but be sure to provide strain relief and connector shielding along with careful cable routing.

Don't use RJ45 connectors if excessive humidity, water, shock or vibration are present. An M12 connector is a better choice and works well as a bulkhead or line connection. Different M12 connectors are used for physical Ethernet connections depending on network speed.

- For speeds up to 100 Base-T Ethernet, a 4-pin, D-coded M12 connector is used with two twisted-wire pairs connected, one twisted-pair sends data and the other receives data.

- For Gigabit 1000Base-T Ethernet, an 8-pin, X-coded M12 connector is used with all four twisted-wire pairs connected.

Ethernet Cable Runs

In new or retrofit industrial Ethernet network installations, do specify Ethernet cable rated for Gigabit network speeds. Most quality Category 5e cable (CAT5e) will work well. CAT6 cable can handle 10 Gigabit speeds, which might be overkill for your application, but will future-proof the installation.

Do use stranded copper Ethernet patch cables for connections in a control panel or field devices. They are available in lengths up to about 50 feet. Longer cable runs often use bulk cable and RJ45 connectors that are manually assembled. Do run plenum-rated Ethernet cable in recirculating or air-handling ceilings.

Patch cables should include an overall metal foil shied to reduce the effects of electrical noise often present in industrial applications. This will help eliminate data loss in transmission and possible intermittent problems.

Since patch cables come in many different colors, do consider differentiating network applications using different color cables. Red for critical, blue for information only, orange for office, etc. While colors are not necessary, do label the cable ends to simplify troubleshooting and future changes.

Don't run these CAT5 or CAT6 Ethernet cables over 100 meters. For longer runs, do use an unmanaged or managed switch or Ethernet extender every 100 meters. For longer distances susceptible to electromagnetic interference, do use fiber optic cables.

Going the Distance

Use of fiber optic cable greatly extends Ethernet signal transmission distance. A managed or unmanaged switch or a media converter takes the electrical signal and converts it to light. Use multimode type fiber (larger diameter and less money) for transmission from 500m up to 4Km, and single-mode typefiber (smaller diameter and more expensive) for transmission from 10Km up to 120Km.

ST, SC or LC fiber optic connectors can be used on either multi-mode and single-mode fiber types. LC connectors are often used in small form pluggable (SFP) transceiver modules that can plug in to Ethernet switches. For a wide transmission range, use the LC connector and SFP for transmission distances from 500Km up to 120Km

Use Industrial Strength Ethernet Switches

An Ethernet switch needs to be tough enough to manage, monitor, optimize and secure communication in an industrial environment. Do specify the following features for an industrial strength switch.

- Industrial grade hardware not commercial
- Temperature range of at least -40 to 165°F
- Metal housing for EMI protection
- Control cabinet mounting
- Redundant dual power inputs
- Surge and spike protection
- RJ45 ports and fiber optic port options
- Small form pluggable transceivers

Ethernet Traffic and Security

Ethernet switches tie all the devices on a network together. For small networks, an unmanaged switch can filter MAC address and route messages through the appropriate port, and can step up the speed of a 10 Mbps device to 100 Mbps.

For enhanced filtering use a managed switch. A managed switch provides multicast filtering (IGMP snooping) to learn what should be sent where. It can also set network traffic priority (QoS/Cos) to give specific data a higher priority.

One of the first layers of security is a managed switch. Set up a good password for switch access, and use port control to limit unauthorized access, or disable ports. Configure port security MAC entries to limit which devices can transmit across the switch. Do use IPsec or some other protocol to encrypt data.

EtherNet/IP: Implicit vs. Explicit Messaging

Over the years, EtherNet/IP has grown into an impressive industrial Ethernet communication network for connecting automation equipment, processes and machines. Keys to its success are the network and transport layers where explicit and implicit messaging takes place; here's how to use these features to optimize your EtherNet/IP networks.

Please refer to the Table and Diagram below for clarification as we discuss explicit and implicit messaging.

Table: Explicit versus Implicit Messaging Cheat Sheet

EtherNet/IP Connection or Feature	Explicit Messaging – Requests and defines information	Implicit Messaging – Data only, no protocol info is included (aka I/O Messaging)
Originator (master)	Client (controller)	I/O Scanner (controller)
Target (slave)	Server (field device)	I/O Adapter (field device)
Form of messaging	Unconnected but can be connected	Connected
Typical use	Diagnostic/event/configuration data	Real-time control data

With explicit messaging, the controller is called the client and the field devices are called servers. With implicit messaging, the controller is called the I/O Scanner and the field devices are called I/O Adapters.

It is important to select a controller that will support explicit messaging as a client or server, and implicit (real I/O) messaging as a scanner or adapter.
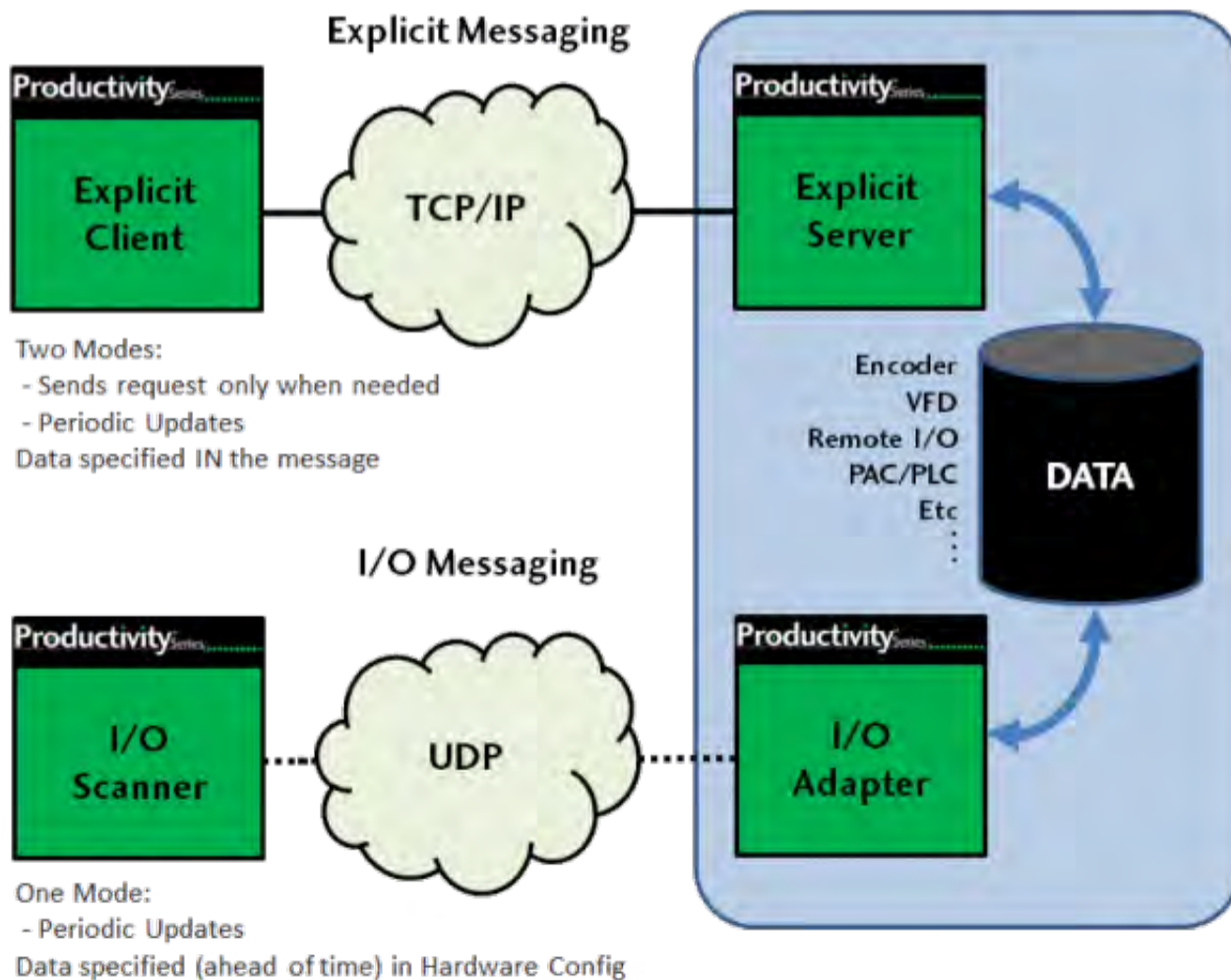
Chapter

6

PLC Communication - EtherNet/IP: Implicit vs. Explicit Messaging

Which Messaging Mode is Best For You?

The selection of explicit or implicit messaging often depends on the choice of field devices, as each may only support only one messaging mode. A controller, on the other hand, generally supports both modes as a client, server, scanner or adapter.

If your application requires large amounts of data, explicit messaging is the preferred choice because bandwidth is saved, as data is only requested when necessary.

For high speed, real-time applications, implicit messaging is the way to go. Think of the "i" in implicit as the "I" in I/O messaging, which of course requires high-speed as I/O is generally used for real-time control.

Explicit messaging requires programming in the controller for setup. You need to request the data, add handshaking, acknowledge the data, and move the data where it's needed in the controller.

Real-time implicit messaging can be quickly configured. Just set the controller up as a scanner to receive data, and configure the connection to a remote EtherNet/IP device by filling in the blanks in the controller programming software. This configuration is basically just defining what the data is and where it will be in the controller data table. Once configured, the data just shows up in the controller memory with no handshaking required.

For outgoing commands from the controller to field devices, just write to local memory and the data is automatically sent to the device. The data is transferred at the rate you specify, typically in the 5ms to 20ms range.

Explicit Messaging Details

Within EtherNet/IP, the explicit message connection can be thought of as a client/server relationship. The client, such as the PLC controller, asks or requests the information from a server, such as a VFD field device, and the server sends the requested info back to the controller.

Because the client requests the information from the server via TCP/IP services, the request has all the information needed to respond explicitly to the message. The Client basically says, "Server, I need this information, formatted exactly as specified in this message, please send it." The server responds with a message containing the formatted information, perhaps confirmation that the VFD speed set point has changed, to let the controller know everything is OK.

This configuring and monitoring ability, common to explicit messaging, works well for non-real time messaging as the client (controller) can send a message request anytime, and the server (field device) can respond when it's available. Explicit messaging is typically used for client/server communication which isn't time critical.

Chapter

6

PLC Communication - EtherNet/IP: Implicit vs. Explicit Messaging

Implicit Messaging Details

EtherNet/IP uses implicit messaging, sometimes called I/O messaging, for time-critical applications such as real-time control. Implicit messaging is often referred to as I/O messaging because it's frequently used for communication between a controller and remote I/O. It's a much more efficient communication connection than explicit messaging as both the client and server ends are pre-configured to know implicitly or exactly what to expect in terms of communication.

Real-time, implicit messaging basically copies data with minimal additional information built into the message. Each end of the communication link doesn't need to be told much about what the information is for, as the data has been predefined. The meaning of the data is "implied" or "implicit" so there is no extra baggage because both ends already know exactly what each bit and byte mean.

Making the Connection

There are two forms of messaging connections used with EtherNet/IP: unconnected and connected. Unconnected messaging is typically used for explicit messages. Connected messaging uses features built into each device and configured in advance for real-time I/O messaging, and it's commonly used with implicit messaging.

Not all controllers with EtherNet/IP support both explicit and implicit messaging. Be sure to verify that messaging capabilities of your controller match those required by your application.

Learn More...Watch The Video!
Productivity Series - EtherNet/IP - Overview

Productivity Series - EtherNet/IP - Overview

Productivity Series

EtherNet/IP™

Overview and Intro

0:00 / 4:46   Full screen

Visit: http://n2adc.com/hya06

Practical PLC Topics

Now that we have covered the basics, let's take a look at some of the more complex operations of a PLC system. In the next several sections we will discuss some of the most widely-used, yet often misunderstood topics in industrial automation. First, we take a deeper look into Feedback Loops, PID control and the method behind the madness. Feedback and PID control is essential in process control industries, such as oil and gas, and in this chapter we break down the mathematical method used in this type of control.

Cyber security is becoming a very real concern for the automation industry. With the industry advancing towards Big Data and the IIoT, securing network PLCs and other control devices is as important as ever. Our section on PLC Security takes a closer look at measures being implemented in newer PLCs to help combat this ever-present threat.

Lastly, we will explore the motion control functionality of PLCs. Servo/Stepper systems are important when precise positioning or movement is required, and PLCs can supply these systems with the needed pulse and direction signals, discrete signals (to an indexing drive) or fieldbus communication.

This Chapter Includes:

The Importance of Feedback

Closing the loop with feedback improves control, measuring and monitoring of packaging, process, and custom machinery. Checking the actual output condition and adjusting the commanded output helps machinery automatically adapt to changing conditions. Open-loop systems save money initially, but will almost always be less efficient and non-repeatable, resulting in a higher total cost of ownership.

Feedback loops can be configured in many different ways, but all have the same basic characteristics. An output signal from some type of controller drives a device which affects the controlled variable. Measurement of this variable is the feedback signal delivered to the controller. The controller compares the desired with the measured variable to determine the error, and adjusts its output accordingly to minimize the error, creating a closed loop system. This is in contrast to an open loop system where there is no measurement of the controlled variable, forcing the controller to operate blindly.

This section will review the advantages and basic design techniques for closing the feedback loop on a process, using practical examples wherever possible. It will also include installation tips and techniques to improve closed loop control in both new and retrofit feedback applications.

Closing the Loop

Simple discrete feedback systems, such as sensors to detect end-of-stroke on pneumatic cylinders, or the use of discrete handshaking signals between equipment instead of timers to adjust controller output, improves control and monitoring. The same is true for properly designed closed-loop feedback on a process.

Closed loop control—which includes measurement, computation and correction—is taught in engineering programs across the world. Laplace transforms and related functions are studied in detail to explain and improve upon closed loop control. The characteristics and advantages of closed-loop control systems are well documented, but closed loop control extends beyond the theoretical to include identification of the advantages of feedback, and the application of feedback in particular machine operations and processes.

Feedback Advantages

Using feedback in closed-loop systems improves control by automatically adjusting the controller output to reduce the error. This helps reduce the effects of dynamic disturbances. Feedback also adds stability to an unstable process, ensuring a repeatable and reliable control loop. Table 1 lists some of the main reasons to add closed-loop feedback to a machine or process.

Table 1: Why close the loop?

1.	Reduce human involvement and related error
2.	Provide 24/7 control with minimal required operator intervention
3.	Standardize machine operation and process control
4.	Improve process repeatability
5.	Cut labor costs.

Many processes have been manually "tweaked" for years, with operators adjusting the controller output to reduce the error. With today's sensor and controller technology, many of these open loops can make use of feedback and a controller to improve operation.

Reducing human involvement in the feedback loop greatly reduces process variations, and allows for continuous improvement as control loop parameters can be continually adjusted to optimize control. These adjustments can be made automatically by various loop tuning software algorithms and programs, or manually by experienced operators. In many cases, a combination of the two methods is used, with operators evaluating recommended changes from loop tuning software, and implementing recommendations judiciously.

Use of feedback in 24/7 operations can reduce process variations and changes that may occur at shift changes as different operators put their own spin on manual loop control. It can also reduce the number of operators needed, or allow operators and other plant personnel to concentrate on other areas such as optimizing operations.

With automatic control enabled by feedback and change control functionality enforced in the controller, process repeatability is improved along with output quality.

Designing Closed Loop System

There are many types of feedback devices available to help the actual output match the desired output in process control and measurement applications. Feedback sensors measure many variables such as temperature, flow, pressure, level, weight and position.

Each of these variables can be sensed or measured with a variety of transducers, transmitters and detectors. Because they are so many choices with widely varying costs and performance, the feedback device must be carefully selected. A check list to help with some common specifications is included in Table 2, and this can be used as a guide when selecting the proper feedback sensor.

Table 2: Feedback selection checklist

1.	Sufficient measurement range
2.	Appropriate sensor accuracy and resolution for the application
3.	Standardized signal types, generally 4-20 mA
4.	Consider digital or smart sensors
5.	Evaluate environmental concerns
6.	Design redundancy and fail-safe system where necessary
7.	Ensure the dynamic response is suitable.

Once the type of sensor is chosen, a temperature sensor for example, measurement range is often the driving requirement. It's important to leave enough headroom for unexpected changes or process upgrades, but not to excess as this will negatively impact accuracy, increase costs or both. If the sensor measurement range is properly specified, then most other specifications will fall into place.

Feedback sensor accuracy and resolution are other important requirements to carefully consider, whether the sensing element is part of a transmitter or a signal conditioner is used. There are two main types of sensing devices, analog and digital, also referred to as a smart sensor.

With an analog sensor, the resolution of the sensing device output and its corresponding analog input at the controller must be considered. Whenever possible, it's best to stick with the common 12-bit resolution as this cuts cost and promotes standardization. Upgrading to higher resolution 16- or 20-bit devices and inputs is of course possible, but it's often not necessary.

In fact, in cases where higher resolution is needed, it's often best to instead use digital or smart sensors. These devices connect to the controller over a high-speed and high-resolution digital data link. Although more expensive than their analog counterparts, they are often cheaper than very high-resolution analog sensors and input cards. But, even high-speed digital communication networks can't match the speed of analog, so loops requiring very quick response times are often better served with analog.

If a temperature transmitter with 0 to 100°C range and <0.02°C resolution works with the application, a 12-bit analog input card will be sufficient to maintain the desired accuracy. In another example, a pressure transmitter has 11-bit resolution. Dividing the total range of the transmitter by 2048 shows the pressure change required to see a corresponding change in the transmitter's analog output. With a common 0-100PSI pressure transducer, 100/2048=0.05, so the transducer will change its output by one step for every 0.05 PSI pressure change. This level of resolution is adequate for most applications, particularly given the vagaries of total system measurement and control.

If multiple feedback devices are present, the feedback output signal should stay consistent wherever possible. Mixing 4-20mA, 0-20mA, 0-5vdc, 0-10vdc, +/-10vdc, thermocouple and RTD signals can increase the complexity of a control system. Feedback output signals can be specified at a common signal level such as 4-20mA, or interposing devices can be used to convert other signals to a standard 4-20mA.

Specifying the appropriate feedback device for the environment is an important design consideration. Harsh chemicals can rapidly damage improperly specified devices, and caustic liquids and gasses will accelerate the deterioration of standard feedback devices. Checking the selection of the feedback device using chemical resistance charts is a good practice, along with questioning knowledgeable suppliers and comparing specifications to the environment. For example, a device carrying a NEMA 4X specification will be suitable for application in outdoor areas listed by NEMA.

Feedback on vital processes must be addressed and carefully specified, and redundancy and fail-safe feedback techniques are often necessary. Backup systems or duplicate feedback control methods should be considered if the machine or process must continue if something fails, and the same is true if the machine or process must automatically shutdown on a failure.

Dynamic response, or signal response time, should be reviewed. A typical dynamic response of .5 seconds to reach a 50% full-scale change works well for most applications. However, some sensors can be much slower or much faster to respond, affecting feedback loop performance. A slower dynamic change can smooth and improve system stability, but an overly fast response can make a loop unstable.

Feedback in Practice

Improving precision, increasing throughput and enhancing quality are possible with proper feedback. For example, maintaining a steady, reliable head pressure of product in a filling machine's hopper is achieved with feedback sensors, and is a simple solution to improve accuracy.

Providing a consistent and regulated supply of ingredients improves product quality. Careful, automatic control of material feed reduces spillage and increases productivity, and required changes are automatically enabled by proper feedback and design of closed loop control.

A PLC or PAC is the controller in many closed loop systems, and provides significant configuration options. Smart relays and low-cost PLCs can be configured to respond to feedback programmatically. Control of the percent on time of a contactor controlling a heating element is an example of simple closed loop control. Although this type of feedback and control works in many applications, it does not respond to input disturbances as well as more sophisticated methods such as PID control.

Higher end PLCs will have built-in PID control, which will provide a much higher degree of control, particularly if the PID parameters are set correctly. The PID control output is generally sent to an analog output channel, or may be used as a time proportion on a discrete output. With analog control, the output can also be used as a set point to another loop for cascade control.

When you're designing a control system, feedback and closed loop systems can greatly enhance performance. Identify areas where feedback can improve and optimize machine operation and process control. Carefully select from the wide variety of available process control and measurement sensors, and make sure each loop is programmed with the right parameters. Following these steps will improve operation of your machines and processes, while reducing labor requirements and enhancing quality.

Methods Behind PID Loop Control

Lock your doors, board up the windows, and prepare yourself for a deeper look into PID loops! Now, if you are still here, great! If you ran out of the room screaming like I did, you can catch up later. For some, PID loops can be scary. After all, they are pretty complex and involve CALCULUS. Hold on, I think I'm gonna turn on all the lights real quick… ok, that's better. PID loops shouldn't be something to fear, they can be confusing but they are not impossible.

First off, for the newbies, PID is an acronym for Proportional, Integral and Derivative. It is a control function that is frequently used in process control. You take a reference signal in from your process, compare it to your setpoint and then change the output to the control device accordingly to minimize the error. Think of the cruise control in your car. You set the speed you want to travel at, your car's computer will compare that to the current speed and either accelerate or decelerate to match that setpoint and remove the error. The bigger the error, the bigger the gain or response. For example, when your car on cruise control revs the engine when you start up a hill, the error between the setpoint and the current speed increased so much that the response was also greatly increased.

WARNING: Readers with sensitivity to math equations, including integral-induced headaches, differential skin rashes or anyone who has had prior calculus-related trauma of any kind, should avert their eyes and not proceed any further.

To help us better understand how a PID loop works, let's take a closer look at one PID equation:

$$M_n = K_c * e_n + K_i * \sum_{i=1}^{n} e_i + K_r * (e_n - e_{n-1}) + M_o$$
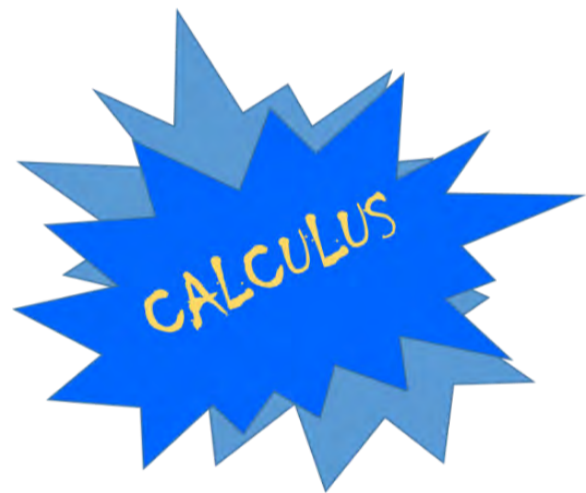
Stay with me, stay with me, it's not as bad as it looks. We'll take it in pieces:

M_n represents the control output at the moment of time "n". This is the amount of gain or response sent to the control device.

e_n is the calculated error at the moment of time "n". It is calculated by subtracting the current value from the set point.

$K_c * e_n$ is the Proportional Term (P) of the PID equation. Kc is the proportional gain coefficient and is a fixed value.

$K_i * \sum_{i=1}^{n} e_i$ is the Integral Term (I) of the PID equation and the most frightening. It is simply the sum of the calculated errors from the first sample (i=1) to the current moment "n". This sum is then multiplied by the Integral coefficient Ki which is calculated using the formula: Ki = Kc * (sample rate/integral time). Basically, the integral coefficient is the proportional coefficient times the ratio of sample rate to integral time. It will make sense once we start plugging in values….I hope.

PID Loop Tuning Helpful Hints:

1. Know Your Equipment

• Don't try to tune over an issue that is caused by faulty or erratic equipment.

• Make sure there are no external equipment problems causing issues in your PID control. For example: tuning a methane-powered generator with a dirty methane supply.

• Watch out for lag or deadtime in your equipment which can be caused by actuators or sensors that are slow to respond to dynamic changes.

• Keep an eye out for external noise on your feedback signal; use a filter whenever possible to be sure the feedback is pure.

$Kr * (e_n - e_{n-1})$ is the Derivative Term (D) of the PID equation. All you do here is subtract the error found at the previous sample (e_{n-1}) from the error now (e_n) and multiply it by the calculated derivative coefficient. The derivative coefficient is calculated using the formula: Kr = Kc * (derivative time/sample rate). Mathematically, the derivative term will look at the difference between the error now and the error before, determine how rapidly the process variable is deviating from or correcting to the setpoint and adjust the output accordingly. Simple, right?

Mo represents the initial control output when the loop is activated. If transitioning from a manual operation, this value is transferred when the loop transitions to auto mode.

So, now that you know all of the pieces and what they do, you need to know how they fit together. The best way to do that is with an example. So, let's apply the equation to a temperature control loop. In our example, the loop is maintaining the temperature of an oven, as seen below, and the desired temperature of the oven is 350°F.

For this particular application, we will set the equation's parameters as follows:
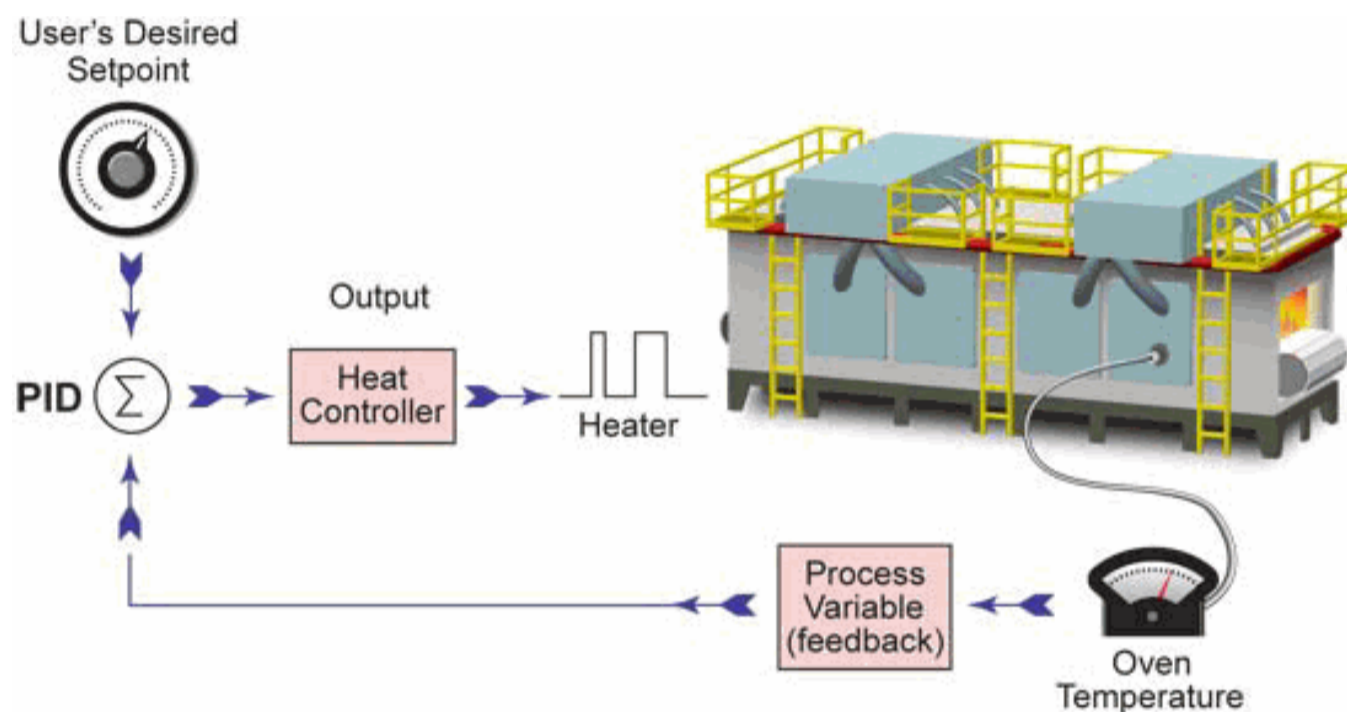
Kr = 0 (To achieve this, set the Derivative Time to zero. This will make the loop a PI controller; we will look at the derivative reaction later.)

Kc = 3 (Proportional Gain)

Ts = 60 seconds (Sample Rate)

Ki = 1 (To achieve this, set the Integral Time to 180 seconds, Ki = Kc * (sample rate/integral time) or Ki = 3*60/180 = 1)

M(0) = 30 (Initial Control Output)

PID Loop Tuning Helpful Hints:

$$M_n = K_c * e_n + K_i * \sum_{i=1}^{n} e_i + K_r * (e_n - e_{n-1}) + M_0$$

$$P - K_c * e_n$$

$$I - K_i * \sum_{i=1}^{n} e_i$$

$$D - K_r * (e_n - e_{n-1})$$

2. Know Your Algorithm

- All controllers are not created equal and most manufacturers have their own interpretation of the PID equation.
- Look through the help file or product documentation to determine which version of the PID equation your controller is using.
- This will help you determine the proper values for your coefficients.

The following graph shows the temperature fluctuation over the past 9 minutes. Let's see how the equation operates.
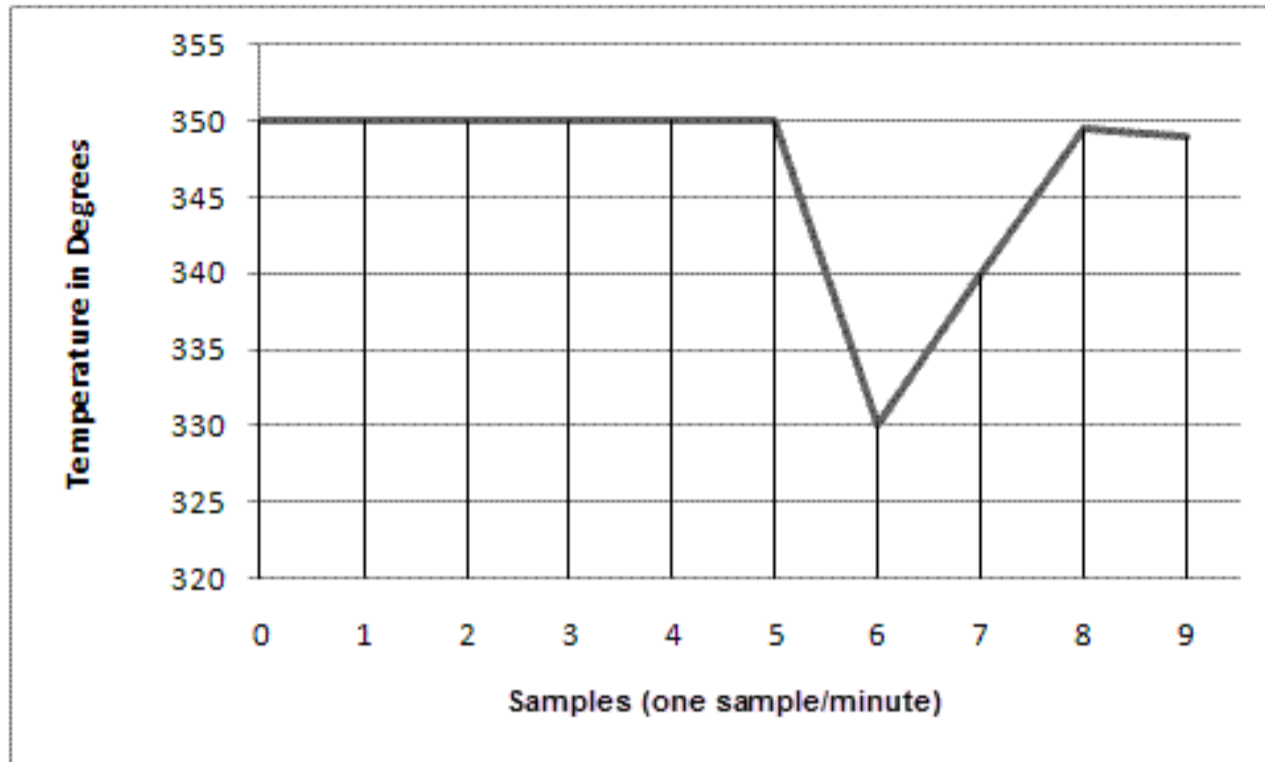
At sample zero, the loop has an initial control output (M(0)) of 30. Looking at the graph we can see that with this control output the loop has been stable at 350°F for the first five samples. After the fifth sample, something occurs to make the temperature drop 20°F. This is seen in our actual temperature reading at sample six. Therefore, the error in the system at sample six is calculated to be: e(6) = setpoint – current value = 350 – 330 = 20. Also, at this point we can easily calculate the sum of all the errors for the system:

$$\sum_{i=1}^{n} e_i = (e(1) + e(2) + e(3) + e(4) + e(5) + e(6)) = (0+0+0+0+0+20) = 20$$

So using the parameters and calculated values above, the equation for the loop at sample six would become: (colors represent proportional, integral and derivative terms)

$$M_n = K_c * e_n + K_i * \sum_{i=1}^{n} e_i + K_r * (e_n - e_{n-1}) + M_o$$

M(6) = (3 * 20) + (1 * 20) + (0 * (20-0)) + 30
M(6) = 60 + 20 + 0 + 30
M(6) = 110

PID Loop Tuning Helpful Hints:

3. Set Your Sample Rate
• PID sample rate identifies how often the instruction will calculate a new output value.
• Make sure the rate at which you are calculating is set appropriately:
  - Too low will cause your controller to miss important changes.
  - Too fast could cause instability in your integral and derivative terms.
• Try to shrink the sample rate as much as possible but more importantly, keep that value constant.

At this point, there is an increase in the control output of 80 from the initial control output and the proportional term is the main force driving this increase with a value of 60. This control output will be converted to an analog value (0-10V, 4-20mA, etc.) that is fed to an end device. In our example, that device is the heating element inside an industrial oven. As the control output drives the temperature toward the setpoint, we see, at sample seven, that the error has decreased to: $e(7) = SP\text{-}CV = 350 - 340 = 10$. Also at this point, the sum of all the sample errors will equal 30 $((e(1) + e(2) + e(3) + e(4) + e(5) + e(6) + e(7)) = (0+0+0+0+0+20+10) = 30)$ and the equation will become:

$M(7) = (3 * 10) + (1 * 30) + (0 * (10\text{-}20)) + 30$

$M(7) = 30 + 30 + 0 + 30$

$M(7) = 90$

Here we see that as the error is corrected, the control output will decrease due to the drop in the proportional term. However, notice that the integral term has risen slightly. At sample eight, the oven temperature has recovered to 349.5°F leaving a minimal error of: $e(8) = SP\text{-}CV = 350 - 349.5 = 0.5$. The sum of all sample errors is now 30.5 and the control output equals:

$M(8) = (3 * 0.5) + (1 * 30.5) + (0*(0.5\text{-}10)) + 30$

$M(8) = 1.5 + 30.5 + 0 + 30$

$M(8) = 61.5$

At this point, the proportional term has dropped off considerably and the integral bias has exceeded the proportional gain in controlling the output. You can see that across the three samples the proportional term has gone from 60 to 30 to 1.5, as the error decreases while the integral term has gone from 20 to 30 to 30.5. When tuning a loop, it is important to remember that with large errors the proportional term will drive the output and with small errors the integral term will take control.

Wow, you've made it this far and I am proud of you! For your reward we will now look at the derivative term and how it affects the equation. Wait, where you going?...The job of the derivative term is to keep stability in the loop. With our example, the rate of change is very rapid at sample six – a 20°F drop within one minute. With very rapid changes, there is a risk of instability and the derivative term attempts to alleviate this while aiding in the correction. As mentioned previously, the derivative term will look at the difference between the error now and the error before, determine how rapidly the process variable is deviating from or correcting to the setpoint and adjust the output accordingly.

PID Loop Tuning Helpful Hints:

4. Tune Your Process

- When tuning a PID loop, focus on one parameter at a time, i.e., cancel out the integral and derivative terms and focus on the proportional term.
- Basic Tuning Method:
  1. Negate the integral and derivative parameters, introduce an error, and get the proportional value to a point where the system is reacting quickly and correcting in the right direction.
  2. Add the integral term, introduce an error, and tune the integral until you get the desired response in the low error range.

If we add a derivative coefficient of Kr = 1 (set the Derivative Time (Td) to 20 seconds, Kr = Kc * (derivative time/sample rate) or Kr = 3*20/60 = 1) to the samples in our discussion they become:

$M(6) = (3 * 20) + (1 * 20) + (1 * (20-0)) + 30$

$M(6) = 60 + 20 + 20 + 30$

$M(6) = 130$

$M(7) = (3 * 10) + (1 * 30) + (1 * (10-20)) + 30$

$M(7) = 30 + 30 - 10 + 30$

$M(7) = 80$

$M(8) = (3 * 0.5) + (1 * 30.5) + (1 * (0.5-10)) + 30$

$M(8) = 1.5 + 30.5 - 9.5 + 30$

$M(8) = 52.5$

Learn More...Watch The Video!
Productivity3000 PID Loop Programming 11 Part Series

Productivity3000 PID Loop - What Is a PID and What Does It Do? (Part 1 of 11)

Visit: http://n2adc.com/2i3v7

Looking at the samples above you can see that after our dramatic temperature drop at sample six, the derivative term starts to lessen the control output at sample seven, going from a +20 gain in sample M(6) to a -10 in sample M(7). At this point, the derivative term determined that the process variable was approaching the setpoint at a very rapid rate and dampened the control output to avoid an overshoot of the setpoint.

Well, there you have it, the basics of PID control. PID loops are very important in industrial automation and aren't something you should be intimidated by. Yes they are complex, and yes they use calculus, but they really aren't too bad. The built-in function blocks in some PLC ladder logic instruction sets can do the heavy calculation lifting for you; the most important tasks then are choosing the coefficient values and tuning the loop for maximum control and performance. Hopefully knowing what each term means and how they affect the equation will help you understand how to set them up. If you would like to see some informative videos on how to use PID loops, check out our **video library**.

PID Loop Tuning Helpful Hints:

4. Tune Your Process (cont.)

3. If PI control is not sufficient, carefully add the derivative component and test accordingly.
   * Remember that the derivative term is based on the <u>rate of change</u> in the error not the error itself. It determines the difference between the error now and the error before (how rapidly the process variable is deviating from or correcting to the setpoint) and adjusts the output accordingly.

4. Once selected parameters are providing adequate control, fine tuning can be done with small changes to one parameter at a time.

• For a more detailed discussion of PID loops and PID loop tuning, check out the following link: Whitepaper: How to Tune PID Loops - **http://bit.ly/pid-tuning**

PLC Security

The Harsh Reality

Let's face it, we are living in the age of the hacker and in the cat-and-mouse game of cyber warfare, hackers have racked up some pretty impressive scores. From WannaCry, to the Equifax breach, to exposing Ashley Madison, hackers have been busy exploiting vulnerabilities and gaining massive notoriety. A recent study by the **Identity Theft Resource Center** concluded that in 2016 U.S companies and government agencies were breached 1,093 times. That's a new record and a 40% increase from the 780 breaches in 2015. But it's not the number of attacks, instead it's the size of the score and the resulting damage that's so striking. In the 2014 Home Depot data breach, hackers were able to compromise store self-check-out terminals and steal email and credit card information from more than 50 million customers. Fifty million! It is estimated that this one attack will cost Home Depot at least $179 million in restitution, with some estimates much higher when legal fees and undisclosed payouts are accounted for. The Home Depot breach was one of many wake-up calls for the corporate world, but unfortunately some of the biggest government agencies were also asleep. The 2015 IRS data breach, where hackers infiltrated the Internal Revenue Service's "Get Transcript" system and gained access to over 700,000 accounts which contained sensitive financial information and Social Security numbers, is believed to have cost taxpayers $50 million in bogus claims.
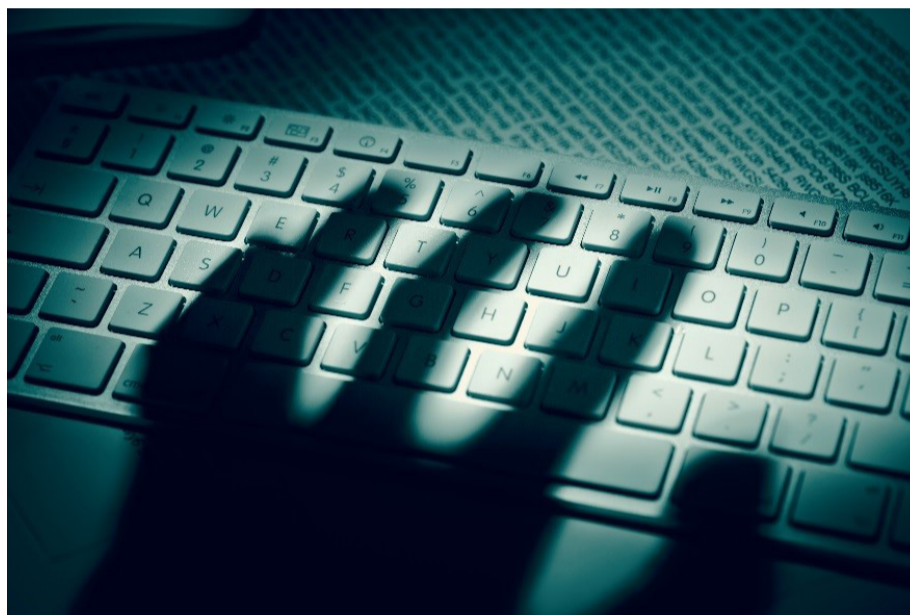
There's no way around it, hacking has become a very profitable business. That's why hackers are so persistent and why they are constantly improving their methods. Methods that range from simple phishing emails, which trick victims into providing personal information through a fraudulent email (phishing scams account for about 56 percent of all breaches last year according to the Identity Theft Resource Center), to Denial of Service (DOS) attacks, which attempt to overload a website with traffic causing a crash, to cookie theft. And while you are probably familiar with phishing since we all seem to have distant relatives in Nigeria who left us an inheritance, or like me, your PayPal account has a problem and is about to be frozen, hackers are constantly looking for new avenues and techniques that you are unaware of. Some of these techniques can even originate from unexpected places as was seen with the recent WannaCry ransomware attack.

WannaCry

This attack was made possible by a data breach at the National Security Agency (NSA). The NSA had recently discovered a vulnerability in the Windows® operating system that would provide an undetected "window" (pun intended) into computer systems worldwide. Instead of reporting the vulnerability, they filed it away as a means for future intelligence gathering. Unfortunately, a group of hackers, known as Shadow Brokers, stole a host of NSA documents and released them on the internet. These documents contained details about the Windows vulnerability which led to the creation of WannaCry, believed to have originated from North Korea. At the time of this article, WannaCry had compromised more than 200,000 computers in 150 countries around the globe, locking people and corporations out of their data and demanding a ransom to regain access.

What about PLCs?

Hackers, especially state-sponsored ones, are extremely active and are becoming more brazen, well-funded, and capable of pulling off even more sophisticated attacks. State-sponsored hackers aren't solely focused on the profits that organized crime hackers seek, with possible targets including a rival country's infrastructure or economy. A country's power grid, water treatment facilities, military systems or nuclear sites are some of the possible targets for these rogue states. Regrettably, one technique that can be used to facilitate these attacks has already been highly publicized and yes, it involved PLCs.

The Stuxnet virus, believed to be a joint venture between Washington D.C. and Tel Aviv, was used to attack Siemens PLCs at the Iranian nuclear plant in Natanz. The virus was believed to have entered the nuclear facility on a thumb drive. Once inside, Stuxnet compromised the facility's PLCs, collecting information and causing fast-spinning centrifuges to spin out of control and destroy themselves. Reportedly, Stuxnet ruined almost 20% of Iran's nuclear centrifuges. Although the Stuxnet attack was used ostensibly to deal a critical blow to the nuclear capabilities of a worrisome government, the attack consequently provided new avenues for those who wish to replicate such destruction.

A **study** conducted on the Stuxnet virus by the SAP research group in Germany concluded that Stuxnet successfully proved to the masses that a very targeted and highly sophisticated cyberwarfare attack was possible. The study also found that Stuxnet's design and architecture are not domain-specific and with some modifications could be tailored as a platform for attacking other automated systems e.g. automobile or power plants.

PLCs with added security

For many years, the industrial sector was isolated from the outside world, foregoing open communication for in-house proprietary means. This type of communication provided a level of inherent security from prying eyes and those wishing to do harm. But with the recent acceptance and explosion of Ethernet in the industrial world and with the current drive to connect as many "things" as possible through Ethernet, industry is more vulnerable than ever. It's imperative in today's connected world that industrial facilities guard their automated systems from possible cyber attacks. Using firewalls, backing up and encrypting data, keeping firmware/software updated, using antivirus software where possible are all ways to protect your facility from attack. But what about the PLC itself? Newer PLCs, like the Do-more! BRX, have multiple security measures built in to help guard against unwanted intrusion.

Guest Memory - Some of today's PLCs are designed with extra measures to help protect them from cyber-attacks. For instance, the Do-more! BRX PLC platform reserves a group of internal memory blocks specifically for external communication. These blocks are isolated to prevent external devices (operator panels, HMIs, etc.) from being able to directly control the PLC's I/O and memory. This method of communication secures the native memory from unwanted access through communication channels.

In practice, when operating as a Modbus server using either Modbus/TCP or Modbus/RTU, the Do-more! BRX PLC routes all Modbus read and write requests from external devices to the four isolated memory blocks that are reserved for external communication. The information stored in this guest memory is available to any ladder logic instruction and can be accessed in place by these instructions. When operating as a Modbus client, the MRX (Modbus Network Read) and MWX (Modbus Network Write) instructions can access all of the native memory as well as the guest memory for any outgoing message data.
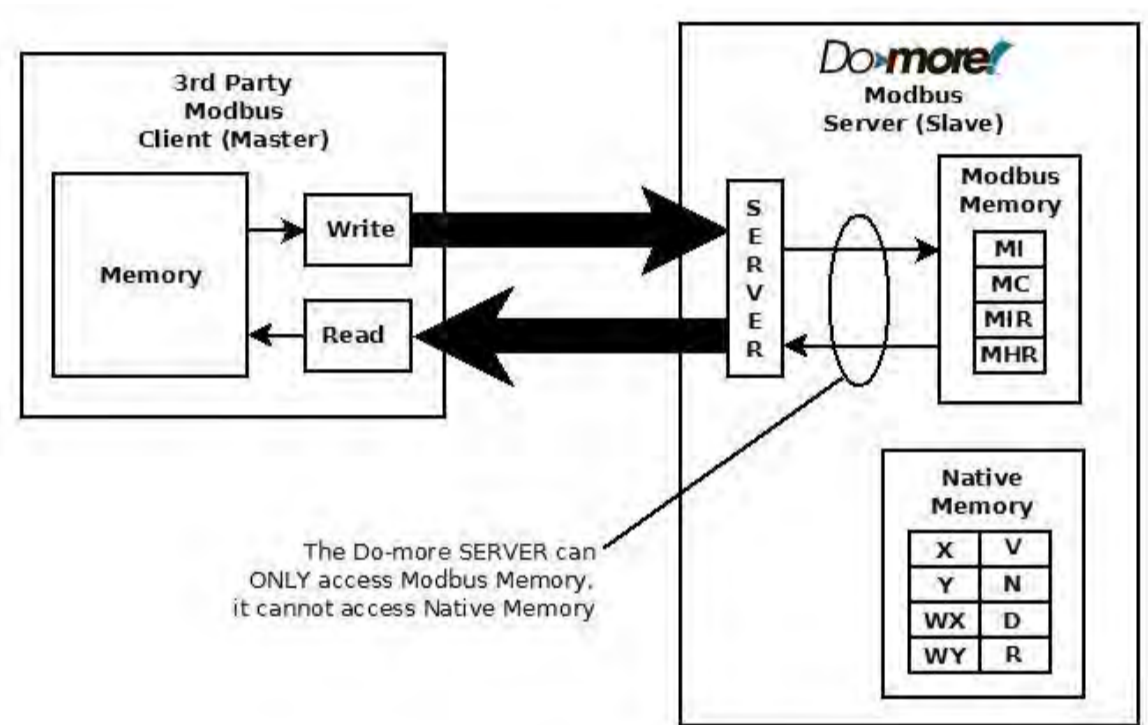
External read/write using guest memory for Server applications
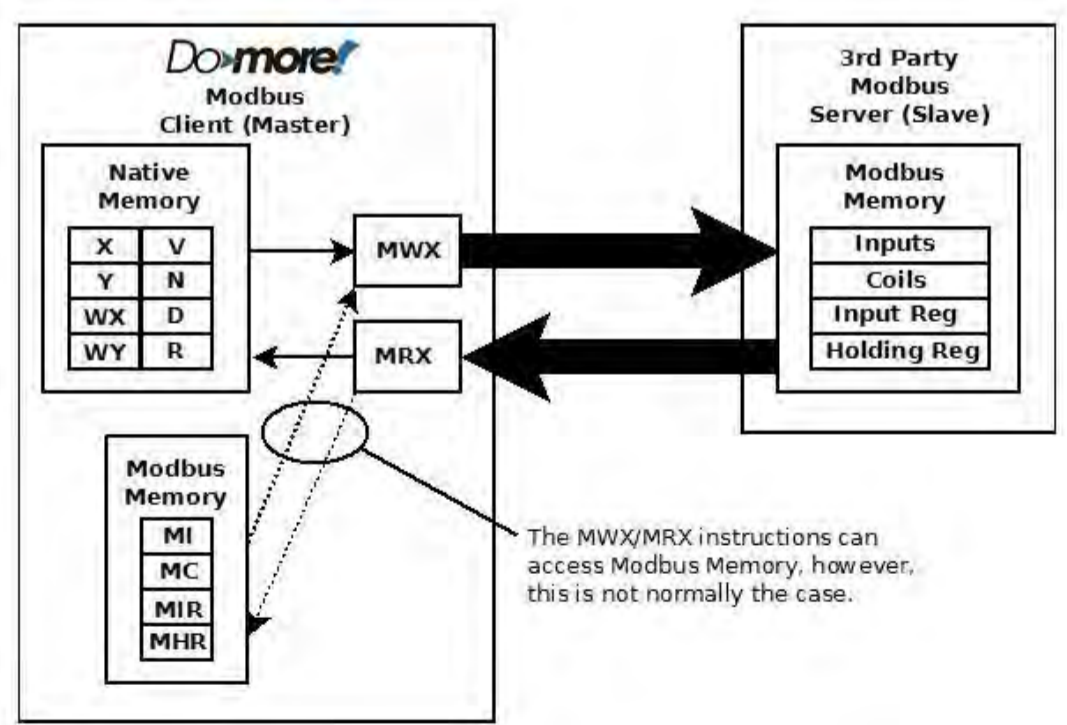
External read/write using ladder instructions for Client applications

Session-based Communication - Many PLCs will be installed on networks that have varying degrees of isolation. This can cause security concerns for programmers and OEMs who need to have communication with the controller be restricted to authorized personnel only. To this end, some PLC programming software, including Do-more Designer, use communication sessions any time that the software is online with the controller.

When communication sessions are established, they are done so with a unique ID, and all communication packets must contain that ID. Any packets received without that ID are discarded by the controller. This prevents unauthorized access of the controller, and also prevents other computers on the network from accidentally accessing the wrong controller. A timeout system is also utilized and will terminate a session after a period of time with no communication between the programming software and the controller. The session must be re-established before communication can continue.

Write Protect the Operating System - Further protection is achieved by simply preventing all downloads to the controller. The BRX PLC includes a block of 8 on-board DIP switches that are used to perform various debug and recovery operations. One of these switches enables/disables firmware downloads to the controller. Disabling firmware downloads will protect your CPU from unwanted operating system changes and keep you in control if and when these changes are made.

Passwords/User Accounts - System security is more than simply allowing or denying the ability to connect to a controller based on a user ID and password. System security also involves the creation of accounts that will allow or deny access to the different resources in the controller. By creating multiple accounts, each with different levels of access, you can efficiently limit who has access to the controller and what each of those users can and cannot do. With user accounts, you can also track the operations performed by each account. Many PLCs will have an event log that will record the active user account for each event that is logged so you know exactly who had access and when.

Advances in industrial communication have provided numerous improvements including faster responses and wider access. However, these advances have also opened up the industrial sector to increasingly active hackers and their growing arsenal of cyber threats. Because of this, it is very important to guard who has physical and online access to your system and its controls. There are many ways to keep your facilities and networks safe and some PLCs, including the Do-more! BRX, offer built-in security measures to help fend off those with bad intentions. If you would like more information on the BRX controller, please visit **www.BRXPLC.com**

Motion Control Explained

Motion control is generally understood to mean the use of servo and/or stepper systems as the "muscle" to move a given load. These motion control systems are capable of extremely precise speed, position, and torque control. Applications which require positioning of product, synchronization of separate elements, or rapid start/stop motion are all perfect candidates for the use of motion control. PLCs are very capable of providing the signals required to command these servo and stepper systems in a cost-effective and digital (noise-free) manner.
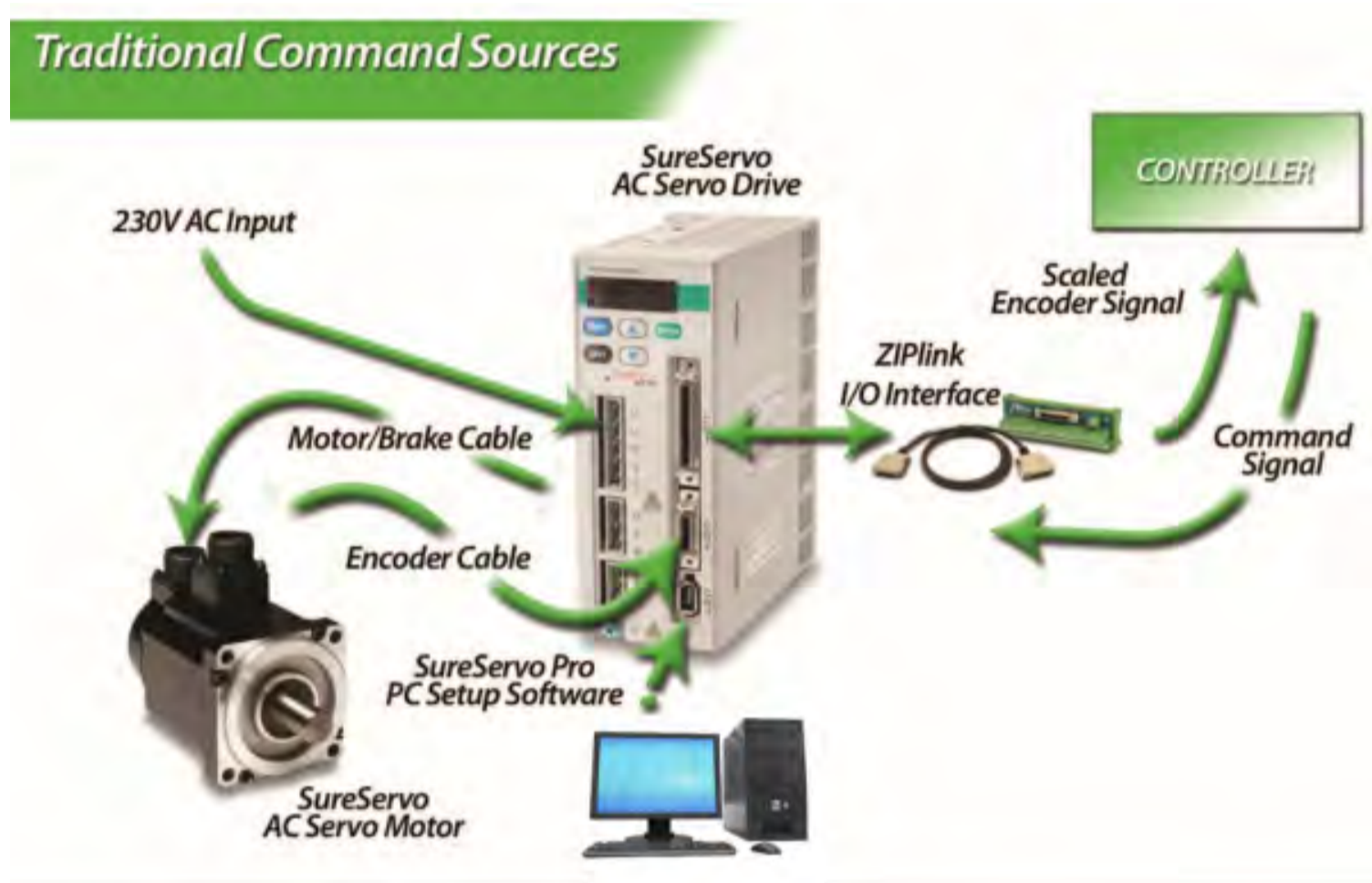
In a typical motion control system there are three basic components: the controller, the drive (sometimes referred to as an amplifier), and the motor. The path planning or trajectory calculations are performed in the controller, which sends low-voltage command signals to the drive, which in turn applies the necessary voltage and current to the motor, resulting in the desired motion. Sometimes feedback devices on the motor or the load are used to notify the drive or the controller with specific details about the actual movement of the motor shaft or the load. This feedback data is used to increase the accuracy of the motion, and can be used to compensate for dynamic changes that may occur at the load, such as changes in mass, friction or other disturbances. Servo systems operate in a closed-loop fashion and vary output torque to move into/stay at the commanded position, while most stepper systems typically provide open-loop position control (a stepper will drive at full force to get to the commanded position or fail trying).

The choice of open-loop versus closed-loop control depends on many factors and both are useful methods for controlling motion. PLC-based controllers can be used for either type of system. Applications that can be accomplished with a low-cost PLC and servo/stepper components include cut-to-length, indexing tables or conveyors, and x/y tables (plotter/cutter/router/placer).

The classic 'pulse and direction' signals that are widely used with PLCs provide an inexpensive, noise-free (digital) method for precision motion control. Extensions or function blocks within the PLC ladder logic are typically used for programming and are easy for factory personnel to understand and maintain. While typically limited to a few axes of control and where coordination between axes is limited, PLC controllers with pulse and direction capability are an excellent fit for many motion applications. Often, low-cost PLCs are already being used for logic control on the machinery and can also handle the motion tasks with the addition of a pulse output card and some additional programming. This can eliminate the need to integrate the logic controller with a separate motion controller. Machine builders can also save considerable time when implementing PLC-based systems, especially if they are already familiar with the PLC and its programming software.

In a typical PLC-based motion control system, high-speed pulse output cards are used in the PLC to generate a 'pulse train' for each servo or stepper drive. The drive receives the pulses and indexes the motor shaft by a pre-set amount for each pulse. Typical stepper systems might index 1/200 of a revolution per pulse, while micro-stepping or servo systems might be configured for as little as 1/10,000 of a revolution for each incoming pulse. The amount of motion dictated by a single pulse can be adjusted in the drive to obtain a desired resolution, or to achieve a required top speed (as a resolution/top speed trade-off may be required to accommodate the maximum pulse output frequency from some PLCs). A separate signal is used to determine the direction of travel. This control method is referred to as 'step and direction'. A similar but functionally equivalent method, 'clockwise/counterclockwise', uses a separate pulse train for each direction of travel. This method is somewhat less popular, but has advantages in some applications. Electronic gearing can usually be enabled in the drive to allow high-resolution moves at low speeds, as well as a high-speed mode for faster moves with lower resolution.

Encoder feedback, when used in these types of systems, is normally handled at the drive level. Two simple hardwired signals from the drive back to the PLC, drive fault and in-position, are often used to notify the PLC of exceptions and/or completion of each move.
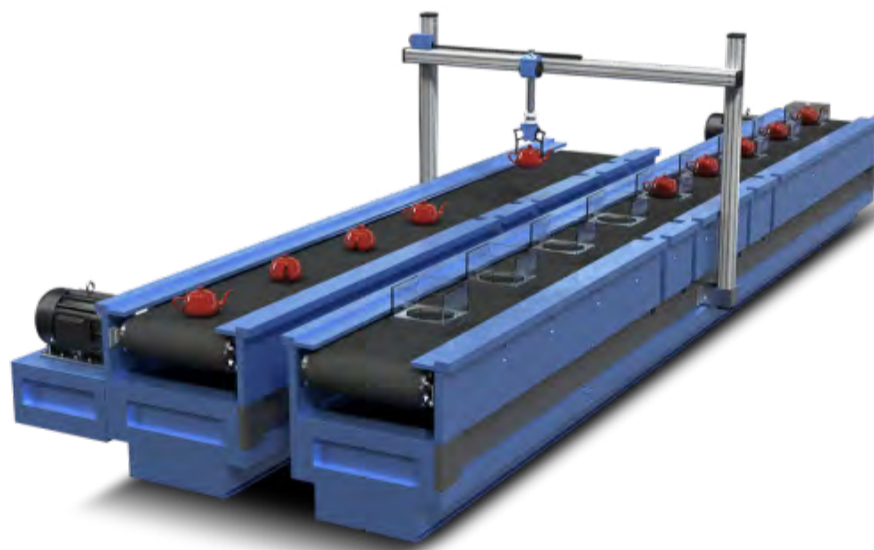
## Evolution of Electric Motion Control Technology

Electric motion control systems originated as alternatives to hydraulic motion systems. With most electric systems ranging in size from a few watts into the kilowatt range, electric motion control has become prevalent at these smaller sizes, while hydraulic systems continue to dominate the larger applications approximately 5kW and above, where electric power is impractical or unavailable and where the environment is harsh or extreme. The electric systems are more factory-friendly, less obtrusive, and are easier to install than the plumbing required by hydraulics. Early electric servo systems were usually operated in velocity or torque mode, accepted analog command signals, and were quite successful despite problems with electrical noise and drift. Early PLC-based controllers used analog output cards to provide the velocity or torque command signals.

While analog control signals are still used on some systems, most modern motion systems have migrated to some form of digital control. The advent of the digital servo drive, with the ability to close the position loop, was another major step forward. New types of signals between the controller and drive are now required to send position commands to these digital servo drives.

The three most common control signals used by today's PLC-based motion controllers are the pulse and direction signals detailed earlier, discrete signals to an intelligent or indexing drive, and fieldbus communications.

The pulse and direction interface that was originally developed for stepper systems has now become a standard feature on most servo products as well. A PLC with a high-speed output is one of the most cost-effective methods for controlling motion today. No intelligence is required in the drive and all programming is performed in the PLC. Almost all PLCs available from AutomationDirect offer some form of high-speed pulse output. For example, the smallest and least expensive DirectLOGIC® PLC, the DL05, includes a single 7kHz high-speed output which can be used for limited motion control applications. The DL05 will also accept an optional module which provides an additional high-speed output channel at up to 250kHz. The AutomationDirect SureStep stepping system can easily be controlled by this or newer PLCs such as the Do-more! BRX or the Productivity2000. In addition to the benefit of the low price for stepper components, all motion and logic instructions are programmed in the PLC's software for significant time savings.

Indexing drives offer another two options for PLC-based control. Indexing drives include a standalone controller which is built in to the servo or stepper drive. These single-axis devices have I/O capability, and can execute motion profiles based on a single PLC or real world input. This type of drive often includes a fieldbus connection and can perform moves based on commands and parameters received across such a connection.
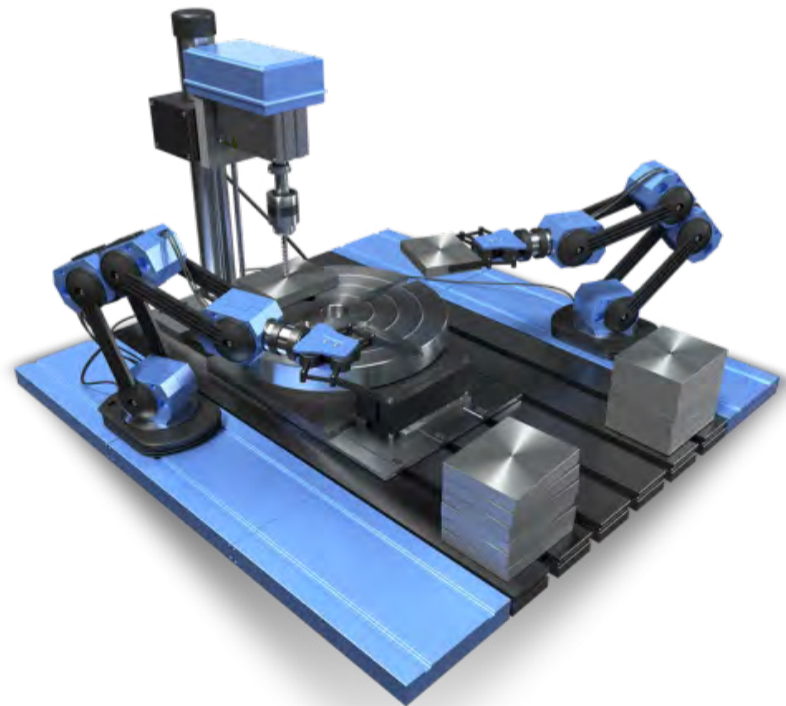
For example, the SureServo® line of servo products from AutomationDirect are indexing drives. The SureServo drives can be pre-programmed with parameters for up to 8 separate motion profiles, which can then be initiated via discrete inputs from a PLC or signals from other devices, even pushbuttons. These moves can consist of precise, user defined accelerations (ramps) to preset speeds, with accurate decelerations carefully timed to end at predefined positions. Dynamic velocities (with controlled ramps) and precise application of torque are also possible. Parameters for vibration suppression (notch filters), load inertia range, proportional and integral gain, and many others can also be customized for specific applications in the SureServo drive. The SureServo offers two adaptive auto-tuning modes, which continue to tune the system while it operates.

The AutomationDirect SureStep® line of stepper products also includes drive models with built-in indexers that offer similar capabilities.

The SureServo products also have a built-in Modbus interface. Modbus-enabled PLCs can initiate moves and download parameters to the SureServo drive across the Modbus link. The Modbus link can also supply information back to the controller about the performance and status of the servo motor and drive system. Multiple SureServo systems can be controlled via a single Modbus port on the PLC. The SureServo's ability to download custom motion profiles from a PLC on the fly, and execute these moves on command, allows the ultimate in flexibility and control with a PLC-based motion controller.

Trends in PLC-based Motion Control

Motion control modules for PLCs are increasing their capabilities year after year. High-speed output modules offered by AutomationDirect now have output frequencies up to 1 MHz. The Productivity2000™ high speed output module provides two 1MHz output channels that can allow extremely high resolution settings in the drives while also commanding very high (if not "top") motor speeds when required. These new high-speed output cards are designed to be low-cost "mini-coprocessors", capable of executing logic asynchronous to the PLC scan.
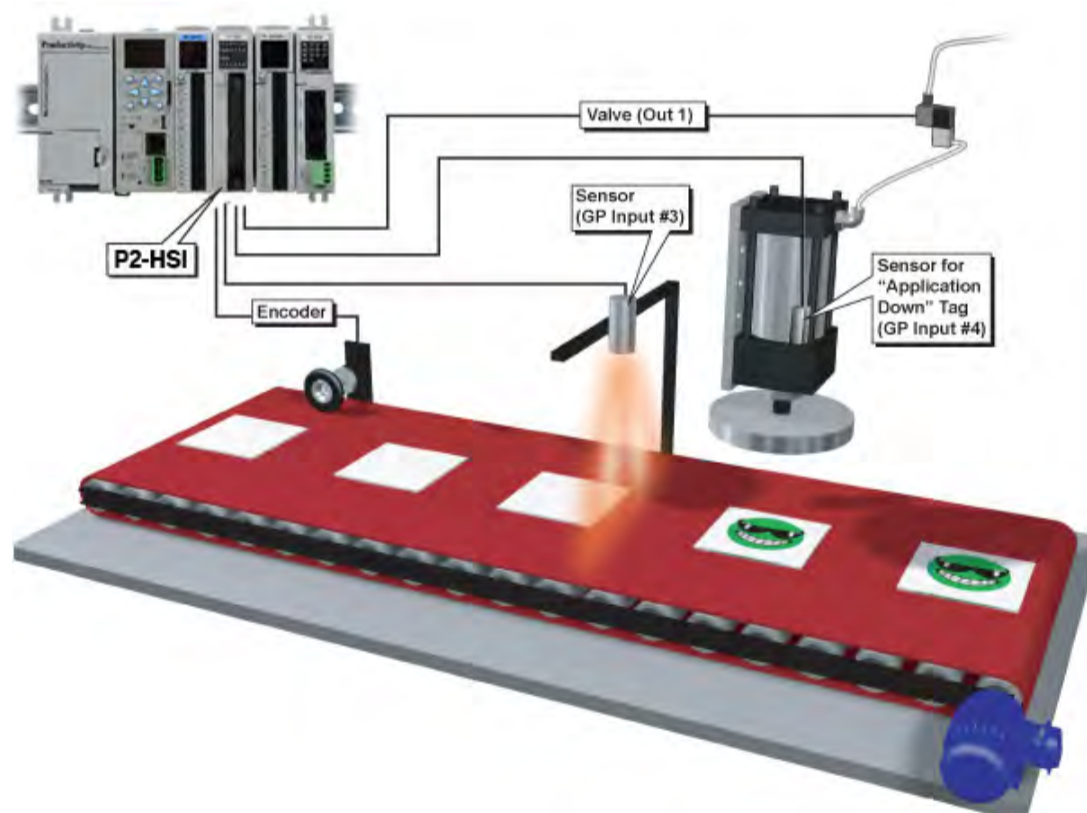
Software is also playing a bigger role in the evolution of PLC specialty modules for both discrete and hybrid control applications. Advancements in software that provide for intuitive graphical configuration are eliminating the need for large amounts of setup ladder logic previously needed in the beginning of the user's PLC program. Microsoft Windows-based point-and-click configuration utilities are becoming more common and provide substantial time savings for programmers and maintenance personnel. With these advancements in PLC technology, PLCs will continue to meet the needs of machine control applications for many years.

Collection of PLC Application Stories

Want some examples of what people are doing with PLCs? You're in luck, we have a full collection of Application stories on our website at **Library.AutomationDirect.com/application-stories**. For your convenience we took the time to list some of our favorites on the next few pages. You can see a preview in the table of contents below and then pick the ones that look the most interesting to you. ust remember that there is a lot more where these came from. And you can check out all of our **application stories on our website**.

PLCs Improve Controls of Radio Broadcasts

The Bulgarian Bat Detector – Simulating Bat Echolocation

The BRX PLC provides agricultural component manufacturer the proof they need

Wastewater treatment plants can be built for small-scale applications

By: Dennis Sloatman,
Vice President of Engineering and Information
Technology at iHeartMedia, Los Angeles

Figures all courtesy of iHeartMedia

This article reprinted from
NASA Tech Briefs, 2017

PLCs Improve Controls of Radio Broadcasts

Broadcasting is a high tech industry and much of it is cutting edge, such as Ethernet-based audio over IP. Our industry employs the latest technology in many areas, such as the use of coding algorithms for bit rate reduction and remote broadcasting. But when it comes to controlling the radio signals sent out by the transmitter, most stations are still doing things the same way they have for a long time.

Controlling the broadcast hardware connected to the antenna—and the related transmission power, source and facilities—is still low tech. Typically, the broadcast engineer gets in his or her car and drives to the unattended transmitter, translator or repeater site (Figure 8A) to flip switches and make other manual adjustments.

In radio and TV broadcasting, it's easy to get distracted by the acoustics, microphones and audio processing. But it's also important to focus on the transmission system because this is where all of the hard work in the studio is disseminated to listeners.

I love the broadcast industry and have been working in it for 46 years. I've been using programmable logic controllers (PLCs) to automate broadcasting functions that have been traditionally very manual, or custom and semi-automatic at best. This article shows how I automated the switching and backup functions of an antenna and transmitter system to improve operations, cut costs and improve reliability. It also provides a path for others in the industry to automate their broadcast and other facilities with PLCs, just as I've done.

Chapter 8

Serving Southern California

iHeartMedia oversees 10 radio stations in a geographical area the size of South Carolina. The LA metro area is enormous in terms of population, which presents opportunities and challenges. In addition to radio stations, iHeartMedia owns outdoor advertising and also has a large online radio presence in the form of the iHeartRADIO app. The station broadcasts in digital, HD radio and old-fashioned FM.

The iHeartMedia website (www.iheatmedia.com) claims over a quarter of a billion monthly listeners in the U.S. and has the largest reach of any radio or television outlet in America. It owns and operates 858 broadcast radio stations serving more than 150 markets throughout the U.S., so there are many transmitter sites controlling AM and FM radio signal transmission.

In the old days, controlling the audio, antenna and transmitter was strictly a manual process. In more recent times, broadcast-specific remote control systems have been deployed to provide rudimentary control via dialup modem. With these systems, alarms can be setup to notify personnel by phone or email if there is an issue. The systems can also have macros programmed to handle automatic functions. The problem is that the cost, reliability and simplicity of these systems don't come close to what the PLC can deliver.

Interfacing and Control with PLCs

The use of PLCs in broadcasting to control transmitter functions is a topic broadcast engineers don't know much about. Discussing a common piece of broadcast equipment, such as audio mixers and audio distribution amplifiers, is easily understood in the broadcast industry. The use of PLCs to perform broadcast functions is not. Ladder logic and tag databases are not well understood, so the learning curve is steep.

Although many broadcast engineers don't realize it, the hardware commonly used in industrial automation works well for control and monitoring of transmission facilities. 50,000 Watt transmitter sites consume about 150 kW of power and contain air conditioning, power distribution and backup power systems.

Over the years, broadcast engineers have upgraded manual monitoring of transmission facilities by applying custom electrical cabinets filled with relays, timers, diodes and capacitors. Some are even taking a Raspberry Pi and interfacing it to a motor or blower fan using contactors and relays, which is not the simplest, most cost-effective, or the most robust design. By contrast, a PLC is designed to reliably control contactors, relays and other components in an industrial setting.

Figure 8A: Many transmitter sites, such as this KLAC radio station facility in Los Angeles, are remote and unattended.

Many broadcast engineers spend a lot of time developing interfaces among different components and items of audio equipment. They take a transmitter manufactured by one company, an audio processor made by someone else, and a remote control system by yet another manufacturer and connect them together along with air conditioners, electrical distribution panels and exhaust blowers. They then figure out a way to integrate it all into a system, but the control hardware they use is often not suited to the task.

Automation Broadcast Functions

Automatic failure switchover and remote access are sorely needed in broadcast because many of the remote transmitter sites aren't readily accessible, particularly during periods of inclement weather. The discrete manufacturing and process control industries have been automating with PLCs for years to provide remote access and other advanced functionality, and we're now adapting this technology to broadcasting.

An application that highlights the use of PLCs in the broadcast industry is at our Dodgers Flagship station in Los Angeles, KLAC AM 570. In this application, we're using the AutomationDirect Productivity2000 controller (Figure 8B) as an antenna/transmitter controller, and we chose this PLC for a number of reasons.

We've used the AutomationDirect CLICK controller in the past for transmitter control and it worked well. However, for the KLAC antenna/transmitter controller, we wanted to add more features. This included email notifications anytime the system did a transmitter pattern change or an error occurred, which is a built-in function of the Productivity2000 PLC. It also has PID built-in, which will be needed for building environmental control when added in a future project.

KLAC is a directional, amplitude-modulated (AM) radio station. In the daytime, it operates using a single tower (antenna) in non-directional mode, which is common in broadcasting at AM stations. At night, it uses two towers to produce a more directional signal to avoid interference with other stations on the same frequency, which is needed because AM signals travel farther at night. This pattern change is required and defined in the station's license from the FCC.

The PLC automates the switching between daytime and nighttime patterns. Using the real-time clock in the controller, we built a table to control the switching time in compliance with FCC regulations. The PLC performs the automated pattern change and sends an email to confirm.

Figure 8B: An AutomationDirect Productivity2000 PLC was used to automate transmitter pattern and backup switching at a remote facility.

Reliable Broadcasting

The Productivity2000 PLC and an AutomationDirect C-more touch panel display are the main components of the automation system also used to control and monitor the status of the station's main and backup transmitters. When we are broadcasting a Dodgers game, the station absolutely has to be on the air, and the reliability of the PLC ensures a continuous transmission signal.

The PLC monitors the status of the main transmitter and will try to restart it if it fails. If unsuccessful, it automatically switches to the backup transmitter. If the backup fails, it switches to a third level transmitter and lowers power if necessary to keep at least a weak signal on the air.

The automation system duplicates what a human would do at the site, automatically handling antenna switching, pattern change, parameter monitoring and other functions. It also performs data logging per FCC guidelines, with this information available remotely. The PLC has a built-in webserver, so we can access logged data remotely via any browser. In our case, this is done through a Cisco firewall at the site.

PLCs have been very reliable compared to the custom control systems used previously. The PLC controls a wide variety of off-the-shelf audio switchers made by various companies. For the radio antenna switching, the PLC is controlling large RF contactors. We control these devices by mixing and matching the PLC's input and output modules including discrete, analog, relay contact and serial I/O. Using these I/O points, we can interface the PLC to all of our broadcast industry equipment.

We also installed a C-more touch panel display (Figure 8C), and created a main screen with buttons to change modes of operation or select a transmitter. The main screen also displays current time and transfer times for the month, along with a wide variety of analog parameters such as transmitter output, current to the antenna and system status. And if needed, a calibration screen can be accessed to scale the analog values to actual engineering units.
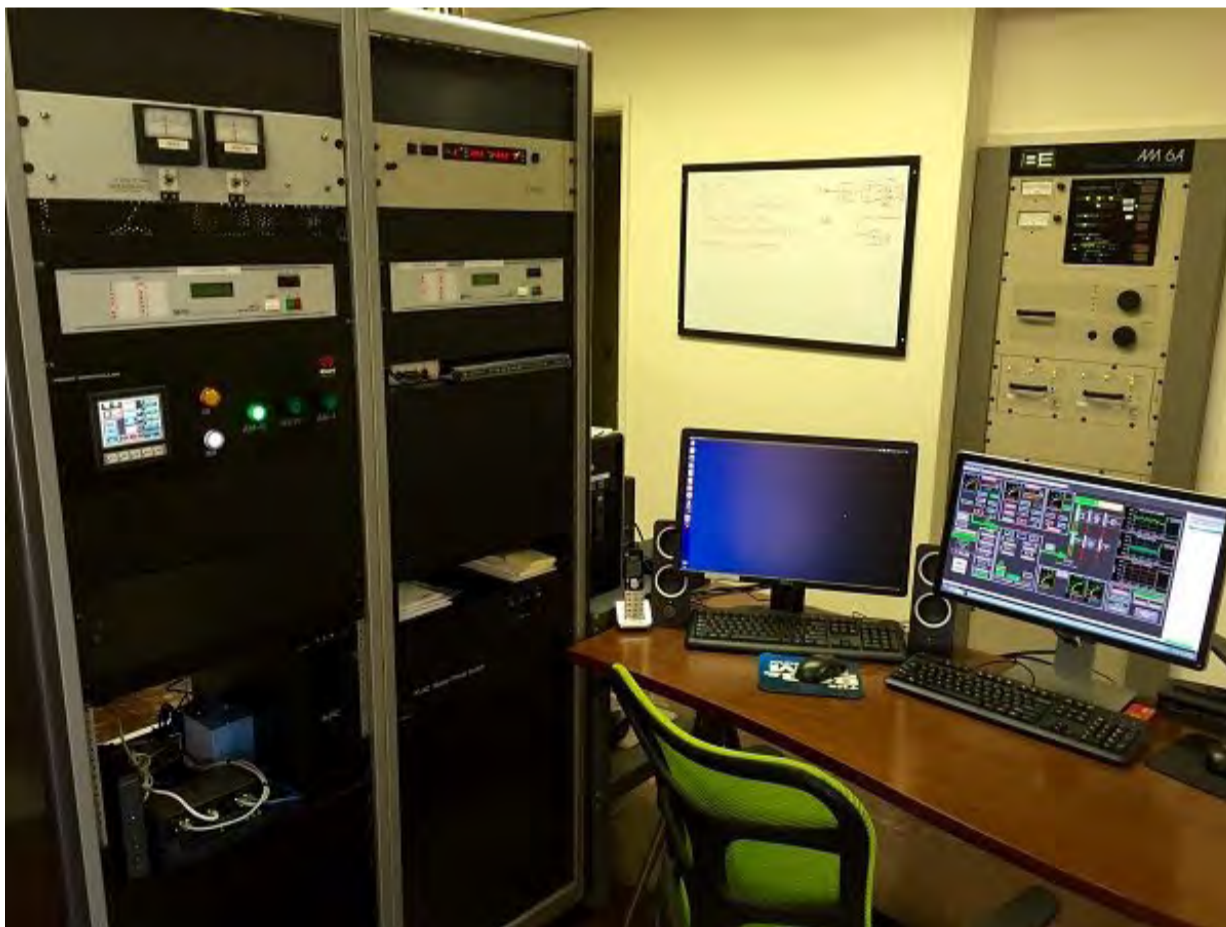
*Figure 8C: An AutomationDirect C-more touch panel display provides functional information
about the transmitter site, as well as manual control and calibration functions.*

Test, Installation and Startup

Installation and startup of the antenna/transmitter automation system took significant planning to minimize the impact on broadcasts. Several test jigs were built to verify operation of the automation system prior to installation, with simulated transmitters built to our specifications to make the test as close to the actual installation as possible.

For testing, we plugged in three simulated transmitters and connected RF contactors. We simulated the complete KLAC setup that the PLC would be controlling, and performed weeks of testing. After some tweaking and optimizing of the PLC and C-more programs, the resulting automation system worked far better than anything that had been installed at KLAC.

The installation went as planned. There were over 30 cables to install, and we used AutomationDirect ZIPLink modules to greatly reduce wiring time. To reduce off-air time to the absolute minimum, signals were bypassed with jumpers until tested so the station could continue transmitting while we installed the new automation system.

There were a few minor tweaks required to the program during onsite startup. When switching transmitter patterns from day to night, the RF signal from the transmitter must be muted to avoid putting 50,000 W through a contactor and damaging it. Tweaking was required to minimize the time that the station was off the air when switching. Actuation times of contactors and RF power-off delay of transmitters needed to be fine-tuned, and when this was completed the pattern switching was barely audible.

Working with AutomationDirect was straightforward and simple. They were chosen based on cost, quick delivery and ease of use. Shipping is free, and they provide emailed status updates. We have called for technical support a few times and were always provided with quick and comprehensive answers.

The AutomationDirect automation system is operating flawlessly at a high-power facility in a remote location. In the future, we plan to use AutomationDirect hardware to automate many more broadcast systems including automatic FM antenna switching, power control, and transmission line pressurization and fault detection.

By Jay Rees, PE,
Rees Engineering Corp.

The Bulgarian Bat Detector – Simulating Bat Echolocation

Over my years working as an Automation Systems Integrator, I occasionally get projects that are different from the normal industrial, process and manufacturing systems. I was recently asked to develop the controls for a museum exhibit to simulate a bat's echolocation. The museum, Children's Center of Sofia, in Sofia, Bulgaria, and I promptly named the project "The Bulgarian Bat Detector". In a near dark environment, the museum visitors rotate a viewfinder towards a display depicting an environment for bats. A couple of bats are located in the display and they are the targets for the bat detector. As the viewfinder approaches the bat's position, the sound level of an audio track increases in volume with a maximum level when pointed directly at a bat. The volume level decreases as the viewfinder leaves the bat's position.

The control solution devised does require the bats to actually be detected as the controls compare the direction of the viewfinder to the location of the bats in the display. The volume of the audio track is set based on the position of the viewfinder relative to the bats with volume increasing when approaching the bat and decreasing when leaving.

Figure 8D: Bat Exhibit under construction at the Children's Center of Sofia, Bulgaria.

Four main design aspects needed to be developed for this project:

      1. Viewfinder direction

      2. Audio playback

      3. Audio volume adjustment

      4. Setup adjustment controls

The biggest hurdle encountered was how to control the audio volume without excessive development and testing while also keeping costs low. As an experienced automation systems integrator, I did not have a volume control solution in my bag of tricks, and it quickly became apparent that some type of hybrid solution was needed. The position detection and process logic would be handled with an AutomationDirect CLICK® PLC, the audio playback would use a commercially available digital audio repeater, and the volume adjustment solution would need to be built using electronic components.

Viewfinder Position Detection

The viewfinder's position is determined using a touchless encoder. The encoder uses a magnet attached to the rotating shaft of the viewfinder and the encoder (mounted to the support) is a Hall Effect type sensor that determines the rotational position of the magnet. The output of the encoder is a 0.25V to 4.75V DC signal representing 0 to 360 degrees. The encoder's output range is perfectly suited for the CLICK's built in 0V to 5V analog input. The PLC also has a built-in scaling configuration for the analog input providing the position's 0 to 360 degree value and placing it directly into a floating point (real) PLC variable.

Figure 8E: Museum visitors will use viewfinders to locate bats in the near dark exhibit

Audio Playback

For the audio playback, a purchased solution was used. The QuikWave™ EM38A is a unit used by museum exhibit developers and allows MP3 audio files to be played in a variety of ways. For this application we used a playback script to run a single audio file in a continuous loop.

Audio Volume Adjustment

The most challenging part of the project was how to control the audio volume from a PLC output. I was hesitant to accept this project because I didn't have a clear cost effective solution at hand. During the quoting process, I remembered reading about a few technologies that might work. These included a digitally controlled potentiometer and using IC analog switches to control a voltage divider. Both of these solutions would require a lot of development and testing to prove them out for the application.

My customer, Andrew Tomasulo of Connect Interactives, who is in the museum exhibit business, came across a solution that used Light Dependent Resistors (LDR). The LDR consists of an LED and a photoresistor as a single device. The intensity of the LED regulates the resistance of the photoresistor. Because the only connection between the LED and the photoresistor is light, the photoresistor is an isolated and passive device. The PLC's 0-5V analog output signal is used to vary the device's resistance.

In order to use the LRD solution for volume control, two LDR units are set up as a voltage divider. Two separate 0-5V analog outputs from the PLC provided opposite voltages to the devices (0-5V and 5-0V). The resistance of the LDR's is from 1,200 ohms down to 60 ohms, which provides a full volume control range.

An issue was encountered when the LDR circuit was first connected to the PLC analog outputs. The PLC analog outputs did not provide enough current to drive the LED in the LDR over its full range. The PLC can only source 2.5 mA and the LED needs up to 25 mA. This prompted a quick trip to the "Shack" for an OP Amp. They only had one in stock (LM324) with rail-to-rail output that would work with my single ended power supply.

Setup Adjustment Controls

The unit uses four potentiometers to set the target locations for the bats in the display. The audio output volume increases and drops within +/- 10 degree range of the target. A 5V power supply voltage regulator (LM7805) was added to the circuit board to power the potentiometers. The outputs of the potentiometers are sent to an analog input module of the CLICK PLC.

Figure 8F: Target potentiometers adjust for the loudest output volume.

Assembly

All of the components were packed into a 12" x 10" x 5" non-metallic enclosure. Field connections are made using Micro (M12) connections and the power cord uses a European Schuko plug (remember this unit is going to Bulgaria). The custom circuit board was hand assembled and mounted on stand-offs. Almost all of the components were from AutomationDirect except the audio playback unit, custom circuit board, potentiometers and (of course) the power cord.
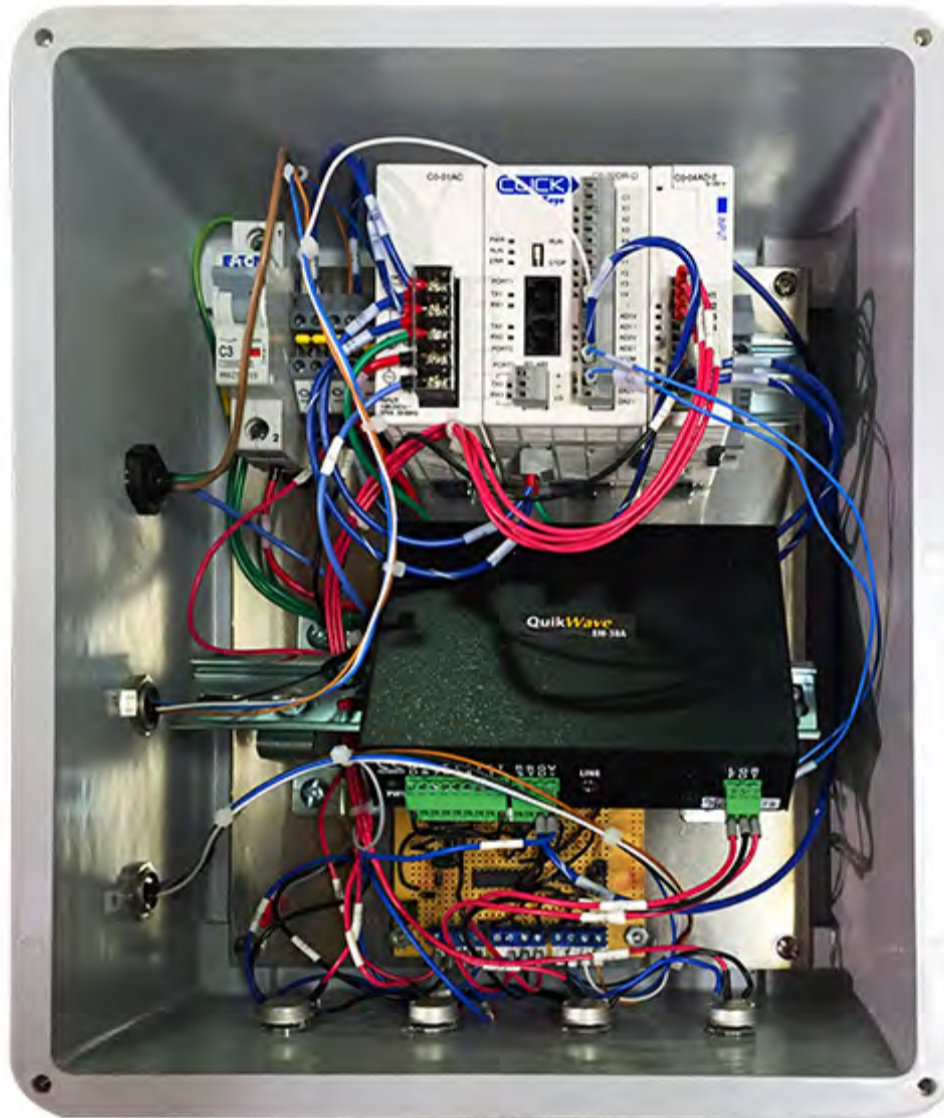

Figure 8G: The CLICK PLC controls position detection and process logic.

Testing

My responsibility only included factory testing. My customer, the museum exhibit manufacturer, would perform the final testing with the actual equipment. I needed to simulate the external functions using an analog simulator for the encoder signal, an MP3 file found on the Internet (bat noises) and a computer speaker.

To test the unit, simulated positions were set on the analog simulator (encoder). For each encoder position, a target potentiometer was adjusted for the loudest output volume. After all of the positions were set, the simulated encoder swept over the full range to confirm the volume changes at the targets.

Conclusion

This was a fun project with its share of challenges but in the end it all worked out (and no bats were hurt in the process). I'm thinking of adding a trip to Sofia, Bulgaria, to my bucket list so I can see and hear the unit in operation.

By James Catlett

Founder, owner and president of system integrator Sensible Automation Solutions, located in Forsyth County, Georgia. He started out as a machinist early in his career, and now has over 12 years of wide-ranging experience in automation, process engineering and manufacturing in the areas of SCADA, PLCs, SolidWorks, equipment maintenance, continuous improvement and predictive maintenance. He holds degrees in Computer Sciences and Industrial Systems Technology from Lanier Technical College, in Engineering from Georgia State University, and in Electromechanical Engineering from North Central State College.

The BRX PLC provides agricultural component manufacturer the proof they need

A local municipality wanted confirmation, and the data to prove it, that the wastewater released to the sewer system by a large agricultural component manufacturer was within the county standards, and they wanted it immediately. What made this request difficult was that the current wastewater treatment system in the plant used only manual data logging. What made it worse was that the system's process measurement instrumentation was uncalibrated and unvalidated, resulting in unreliable chemical dosing.

Application Stories - The BRX PLC provides agricultural component manufacturer the proof they need

Figure 8H: Acid and caustic tanks: These tanks hold the acid and caustic needed to balance the PH of the wastewater produced from a large agriculture components metal coating process.

System integrator Sensible Automation Solutions was brought in to quickly upgrade the wastewater treatment system. Although the original system operated well with an AutomationDirect Click PLC and C-more HMI, it couldn't be used to meet the advanced requirements now demanded by the municipality.

It was necessary to add automatic data logging in place of the manual data logging. To record process data better, high-speed inputs to the system were needed to read flowmeter pulse outputs. The agricultural component manufacturer also requested daily reports on the wastewater system process data, with the ability to email this information to its managers. None of this functionality was available with the Click PLC, so Sensible Automation Solutions chose the AutomationDirect BRX PLC to replace the existing controller and meet requirements.

Getting Control of an Existing Facility

The control system upgrade was performed on the existing wastewater treatment system, housed in a 200,000 square-feet facility with fabrication and metal coating processes for agriculture components used on large equipment. The system treats wastewater from the metal coating side of the plant, and it supports the power coating line, which uses a variety of chemicals. These chemicals are changed out or refreshed multiple times a day. This requires treatment of 25,000 to 30,000 gallons of wastewater per day so it can be safely released to the municipality's sewer system.

The BRX PLC was a cost effective and capable choice for this system, made easier because a couple of the technicians at the facility had prior experience with AutomationDirect. Sensible Automation Solutions has used a lot of PLCs in the past, but the BRX brings new capabilities for its form factor and cost. It's a small, brick-type PLC with analog inputs and outputs, configurable high-speed inputs, and high-speed outputs—filling the needs of many different processes. The PLC's Do-more programming software is powerful yet easy to use, adding to the appeal of the platform.

Another reason the BRX was chosen was because it fit in the panel space originally occupied by the small Click PLC. We simply unwired the existing Click controller and installed the BRX, along with ZIPLink connector modules and cables, and re-attached the existing power and field wiring. Everything fit with minimal required changes and wiring additions.

*Figure 8i: Control Panel: This new AutomationDirect BRX PLC replaced the existing Click controller
with minimal required changes or wiring additions.*

Some of the I/O that was already in place included a Rotex flowmeter, three analog pH sensors and an ultrasonic level sensor. Preexisting solenoids were used to run the air-operated diaphragm pumps.

Process Details

There are various tanks at the facility filled with chemicals used to treat and prepare the surface of the steel before it is powder coated. These chemicals are consumable. Once a tank's chemical gets dirty, for example, it must be removed and changed out with fresh chemicals. Used chemicals are dumped to a waste manifold that drains into a pit.

The system must treat caustic and acidic chemicals in the pit. Every time the chemicals are changed out, they are drained into this pit. The pit must be brought to a pH between 5.5 and 8.5 before release to the sewer system. The control system monitors fluid level in the pit, adding either an acid or a base depending on measured pH.

The system logs pH data throughout the day including how much acid or base is being added and when, and how much water has been discharged. Once the water is within specification, the system pumps it out and logs process data such as when the water was pumped out, how much was pumped out, the amount of chemicals needed to achieve the final pH level, and the pH level when it was released to the sewer system.

The coating line is almost three quarters of a mile long. Workers hang components such as racks, engine mounts and frames on to hooks connected to an overhead monorail-conveyor line. The line is powered, transporting the components and dipping them into the different tanks using conveyor elevation changes. There are also two or three robots inside a paint booth that apply the power coat to the components. Once coated, the components move through an oven where the coating is cured.

The monorail conveyor moves components through five different tanks. The first two are cleaning and neutralizing tanks. The next tank contains a surface modifier to help the power coat attach to the substrate. A rinse tank follows, and the final tank is a specialty tank, sometimes used or not, depending on the process.

Capable Control

The BRX PLC controls two 15-GPM diaphragm pumps used to feed the acidic or base solution to the pit based on the monitored pH levels. The amount of solution pumped varies depending on pit fill level. For example, if the pH is 5 and needs to be brought up to a pH of 7, the PLC will pump in the base while also running a recirculating pump that moves water from the bottom of the pit to the top for improved mixing. The PLC directs dosing of the pit, waits for 60 seconds and records the pH reading. If the pH level is acceptable, it will turn on the discharge pump to empty the pit to the sewer system.

The dosing needs vary. The pit could be holding 1000 to 5000 gallons, depending on the changeover process running. Because of this, a programmed routine in the BRX PLC, including a PID function, calculates the amount of acid or base to add to the pit depending on the measured pit level and measured pH in the pit, using this information to calculate the run time of the dosing diaphragm pumps.

Plant personnel knew about how much chemical was needed to treat the wastewater, but the tricky part was that it was just an estimate, depending on what the wastewater was composed of and the mixture of chemicals in the pit. Sometimes adding a gallon of base wouldn't necessarily change the pit pH as expected, requiring more dosing.

The flowmeter added to the system had high speed outputs proportional to flow that could be read by the BRX PLC. Since this flow data and other data was logged and sent to the municipality, it needed to be accurately date-stamped, and the PLC provided this capability with its built-in clock.

The program to control the system was written from scratch for use in the new BRX PLC using the Do-more Designer Software. The facility provided a sequence of operations defining how it wanted the system to work. Ladder logic was then used to satisfy the requirements of the provided process sequence.

Installation Aids

It was tricky to install the equipment—and time sensitive. The facility runs 24/7, so only six hours of shutdown time were available to remove the old controller, install the new one, power up, debug and test the system. Fortunately, another feature of the BRX helped with this quick upgrade, namely the on-board simulation, used extensively before installation to debug the program code.

Using simulation, it was possible to change pit level and pH process variable and see how the code responded. Using the simulation GUI interface, which is configurable, the process variables could be changed, and it made it easy to understand the program's response by providing visualization of the controller outputs. As the actual program was simulated running, it even logged data.

On the hardware side, there were about 60 wires to remove and re-install for the new controller. There were some discrete signals, such as manual override pushbuttons and selector switches on the front panel as well. It all went together quickly.

HMI, Trending and Logging

The new control system includes an AutomationDirect C-more EA9 Series 15-inch touch panel HMI. The old system only had three or four pushbuttons on the HMI, so a new graphical interface program was created from scratch to take advantage of the C-more's capabilities.

Figure 8J: HMI: An AutomationDirect C-more EA9 Series 15-inch touch panel HMI was added to the new control system to provide status, data trending, historical data, reporting and other functions.

Screens were added for data trending, allowing viewing of historical data for the last few hours. Screens were also added to view logged data from previous days. New operational screens were provided to view the level of the pit, dosing activity and general process status.

Data is logged in two places because of its criticality, in the BRX and the C-more touch panel. The C-more has memory onboard, and more can be added using a USB memory device or microSD card. Data is logged to the C-more's microSD card, from where it is pulled for reports and emails.

The automation system creates reports and sends them via emails. For example, every day at 12:01 am, or any configured time, the C-more panel is setup to transfer the data via email to company managers. These reports are set up so that every night it also sends a picture of the multi-variable trend graph, along with a comma-separated values file of all the data trended in the graph. The automation system also sends out other specialized reports such as how much of acid and base solution was used throughout the day.

Sensible Automation Solutions is currently working with the agricultural component manufacturer on three more projects, including automatic leveling of process tanks, and adding additional data logging on other C-more panels in the plant for additional plant processes.

The biggest challenge was doing the work quickly to satisfy the local authorities. To meet this requirement, Sensible Automation Solutions installed the new automation system within 10 days after the first call from the agricultural component manufacturer.

Innovative Treatment Products redesigns its process control system to handle a wide range of treatment throughput

By Gary H. Lucas,
Innovative Treatment Products

Wastewater treatment plants can be built for small-scale applications

For most people, treatment of wastewater from homes and businesses is one of those it-just-happens things taken care of by local governments. But any facility too isolated to connect to a larger sewer system—such as a school, office or retailer—will have to supply its own treatment for obvious practical reasons and to avoid a variety of fines from environmental agencies.

These isolated locations often have too much water to treat using a septic tank, so they must provide their own wastewater treatment plant (WWTP), usually a miniature version of city-scale counterparts. The main operational problem these small-scale WWTPs face is flow variability. A small-scale WWTP is designed for an anticipated flow rate to cover the maximum the supported facility expects to generate but can struggle to operate properly when the flow is toward the low end of the range. This can be mitigated somewhat by adding storage tanks, but a better solution is to change the process to make it more inherently scalable.

This task of redesigning the process has been the objective of Innovative Treatment Products (ITP) in Owing Mills, Maryland, from its inception. By starting with a clean slate and avoiding the pitfalls of simply making smaller versions of large plants, new pre-packaged plants using membrane-bioreactor technology can operate continuously at 5% of rated capacity, while still delivering high discharge quality in compliance with the highest state and municipal regulations (Figure 8K).

Chapter 8

Figure 8K: Single skid ready to run. By starting with a clean slate and avoiding the pitfalls of simply making smaller versions of large plants, new pre-packaged plants using membrane bio-reactor (MBR) technology can operate continuously at 5% of rated capacity, while still delivering high discharge quality in compliance with the highest state and municipal regulations. (Source: ITP)

ITP is an original equipment manufacturer (OEM) of wastewater-treatment equipment. It designs and manufactures complete, self-contained, packaged WWTPs for capacities ranging from 5,000 to 50,000 gallons per day. Additional products include:

- miniature membrane bio-reactor (the mini)
- Clarifier CleanSweep algae remediation system
- preScreen rotary-drum screen.

All miniMBR process equipment, with the exception of the reactor tanks, is contained on a single skid that arrives at the site ready to install and run.

Bugs at Work

An MBR breaks down biological waste using naturally occurring bacteria. This is done in multiple phases in separate tanks as the bacteria tends to be specialized. The first step uses one type of bacteria to break down the solid material, while the second step uses a different type to convert ammonia in the wastewater into a nitrogen compound suitable for release back into the atmosphere.

The final step is removing the inert solid byproducts while keeping the various types of bacteria in the tanks where they belong. This is done using an ultrafiltration membrane with 0.03-micron pores to remove virtually all solids. With this filtration method, wastewater at a concentration of 10,000 mg/l of solids, which is very dark and thick, is turned into crystal-clear water in one step.

The wastewater being processed must move from stage to stage at the appropriate rates to allow the bacteria sufficient time to do its work. Additives must be dosed into the wastewater in the right amounts for optimal processing to use the available time efficiently based on the throughput demand at any given flow rate. As a result, the automation system controlling the installation must be able to perform sophisticated analysis of the operating parameters and make adjustments to match demand. Achieving this internal scalability is far more complex than designing a unit running at a constant flow rate.

Variable Flow Rates

The description so far understates the complexity of what's going on inside the unit. Whether it's the largest or smallest MBR-equipped plant, the equipment and control strategy are very similar. A typical unit uses 14 automated valves, four pumps and three blowers (Figure 8L). Measuring progress of the process and the condition of the bacteria requires monitoring pH, oxidation-reduction potential (ORP) and dissolved oxygen (DO). Flowmeters and level instruments are also needed. All these elements are tied together by a programmable logic controller (PLC) automating the entire operation and by human-machine interfaces (HMIs) for monitoring and operator interface.

Figure 8L: Multiple control panels and valves. Whether it's the largest or smallest MBR-equipped plant, the equipment and control strategy are very similar. A typical unit uses 14 automated valves, four pumps and three blowers. (Source: ITP)

This PLC has a lot to do, so the application requires a versatile and sophisticated unit. ITP has been using AutomationDirect Do-more Terminator PLCs for some time, but it recently began using the new AutomationDirect BRX Series PLCs, a stackable micro brick design. It uses the same Do-more Designer Windows programming and documentation software as the other Do-more PLCs, so only a few simple changes were needed to integrate it into new systems in the production pipeline.

When wastewater begins to flow into the system, the process must be controlled precisely. For example, the pumps moving wastewater into the first part of the process are regulated by variable-frequency drives (VFDs) via a flow loop controlled by the PLC. The PID function in the PLC keeps the flow at exactly the rate needed by the system.

The DO level must be maintained within a very narrow range. Too much oxygen wastes energy and can stop the second step of the process, denitrification. The PLC reads data from a DO probe and controls an adjustable air valve from the blowers to keep the level constant.

The solids content level increases during the process, so sludge must be extracted periodically. The PLC calculates the amount to be taken out each day based on the flow of the previous day and automatically extracts the correct amount to keep the concentration constant.

Membrane-Filter Maintenance

One of the most critical elements of the MBR system is the filter membrane. Since it captures all solids down to 0.03 microns, bacteria and most viruses get removed. To keep the membrane from fouling, there is an automatic backwash process performed every 10 minutes. This step reverses the flow through the membrane using various pumps and valves.

The membrane also needs periodic chemical cleanings. This is a 36-step process that includes filling, flushing, dosing of two chemicals, soaking, circulating, heating, dumping, refilling and monitoring over a period of about eight hours, usually overnight. This was a complex sequence to automate, but the PLC's stage-programming functionality made it much easier.

Stage or state programming is an extension of normal relay ladder logic and works well for processes with clearly defined steps or states occurring multiple times during a sequence. The result is a robust process able to run time and time again without interruptions.

Performance with Affordability

For ITP, being the least expensive among WWTP producers does not assure success because potential users might not be able to afford a solution from any vendor. Therefore, ITP must find ways to keep costs down, well below other competitors, while making sure this objective doesn't sacrifice performance or reliability. The answer is to get the most from every dollar, and this takes a variety of forms.

System designers model everything, including electrical panels, using SolidWorks software. Equipment models are imported directly into the purchasing system which allows ITP to estimate, quote, purchase and assemble systems directly from the models. Material-receiver documents reference individual job and subassemblies, so incoming materials can be placed directly in the work-in-process bins dedicated to each job. This just-in-time approach to inventory keeps costs as low as possible.

ITP has a small but well-equipped shop to fabricate and machine unique system parts. It includes a CNC milling center that doubles as a CNC lathe, in addition to both plastic and metal welding capabilities. A core pillar of the company's philosophy is: "Inventory is the enemy of innovation." Therefore, it works toward processes allowing ITP to make one of any part quickly and economically. The large fiberglass tanks and buildings are supplied by a partner company.

Supplier Selection

Choosing the market leader as the primary supplier for critical automation components can have its drawbacks. Some suppliers are very responsive, while others exploit their dominance at the expense of customers. Within the world of automation, the latter is often the case. Without naming names, let's just say the market leader charges for tech support and has cumbersome distribution methods, which add cost and make it difficult to get critical parts in an emergency.

Fortunately, when it was time to design the control system for ITP's first small MBR WWTPs, the decision was to start with a clean slate and choose the best automation supplier, not just the most recognizable name. With a plan to sell these treatment units nationwide, support would have to be available anywhere. And, since most units would be purchased by schools, parks, convenience stores and other small companies, low prices were very important.

AutomationDirect's Do-more PLC was chosen for this application primarily for financial reasons, but it gets the job done well and is easy to support. Also, previous experience using AutomationDirect C-more HMIs and other components over six years made us very comfortable with the company's capabilities and the product reliability (Figure 8M)
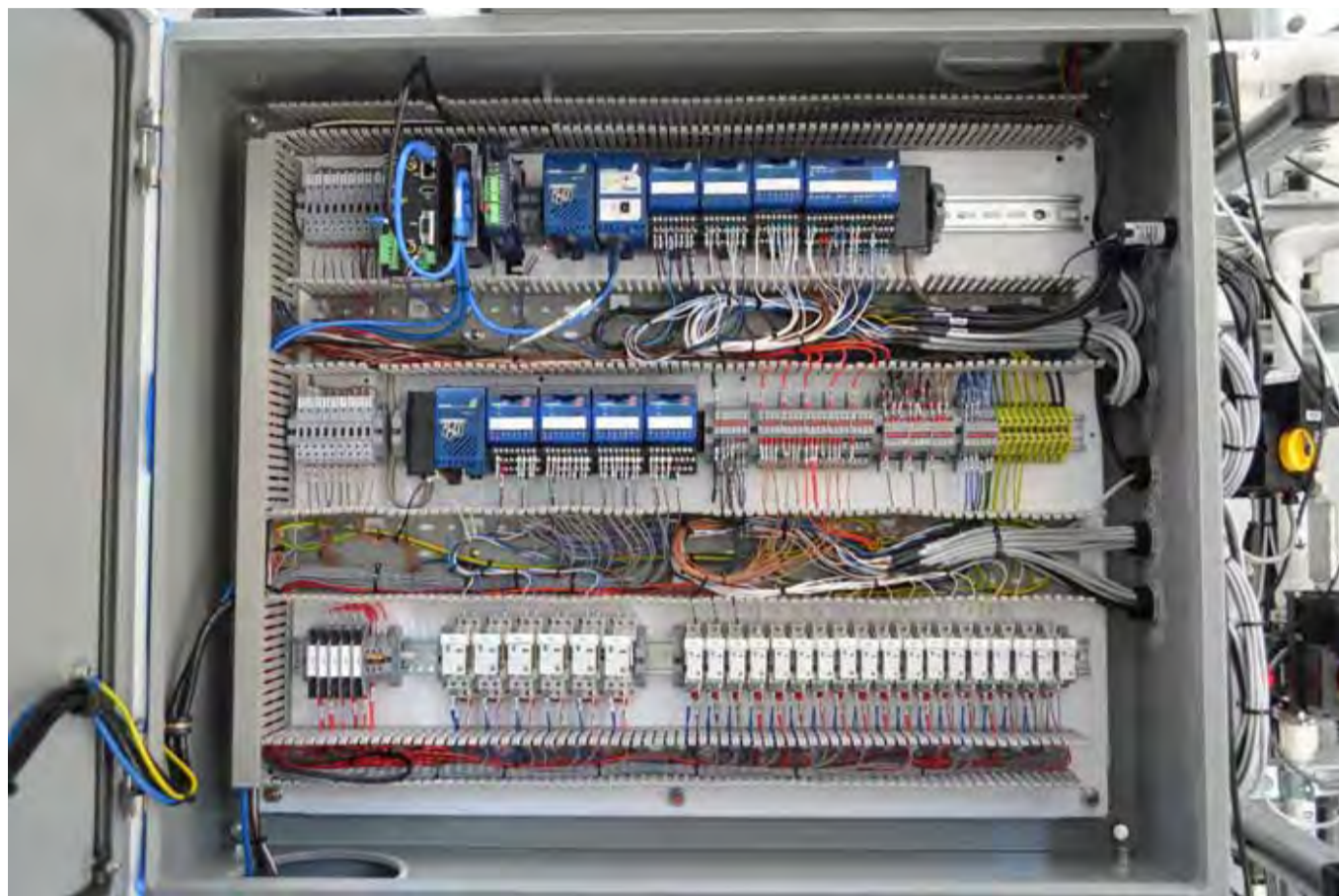

Figure 8M: Reliable control. Capable and reliable components provided opportunities to cut costs on flowmeters by using turbine designs and feeding the pulse-count output directly to a high-speed input on the PLC. (Source: ITP)

These components also provided opportunities to cut costs in clever ways. For example, we save costs on flowmeters by using turbine designs and feeding the pulse-count output directly to a high-speed input on the PLC. This eliminates the cost of a transmitter to perform a conversion the PLC can easily do itself. Similarly, high-speed outputs are used to drive chemical-dosing pumps directly. This approach saves nearly 10% of our control system cost. In addition, the free Do-more Designer Windows programming and documentation software allows us to supply operators with the means to service a unit on site at no cost, simplifying troubleshooting.

PLC Programming and HMI Configuration

The PLC programming strategy takes advantage of the Do-more architecture by using a main program with subprograms for each feature of the system. Using just one control program with descriptive tag names and clear comments ensures everyone will still understand how the program works years from now. Not every system uses every feature, so setup parameters simply disable the subprograms not required.

Operators in the field often struggle with calculating dosing requirements and flow rates, so the PLC does those using its math block. For example, tanks can be vertical cylinders, horizontal cylinders or rectangular. The PLC calculates tank capacity in gallons, along with the current volume, regardless of tank shape.

The HMI plays a significant role, but it is changing. Displays on the actual unit are now smaller than in years past, and we have settled on AutomationDirect 6- and 7-inch C-more EA9 series touch panels. This is practical because users tend to rely on the C-more's smartphone app for remote access to the equipment. By designing the graphics for a small display, everything is also easily readable remotely on a smartphone (Figure 8N).
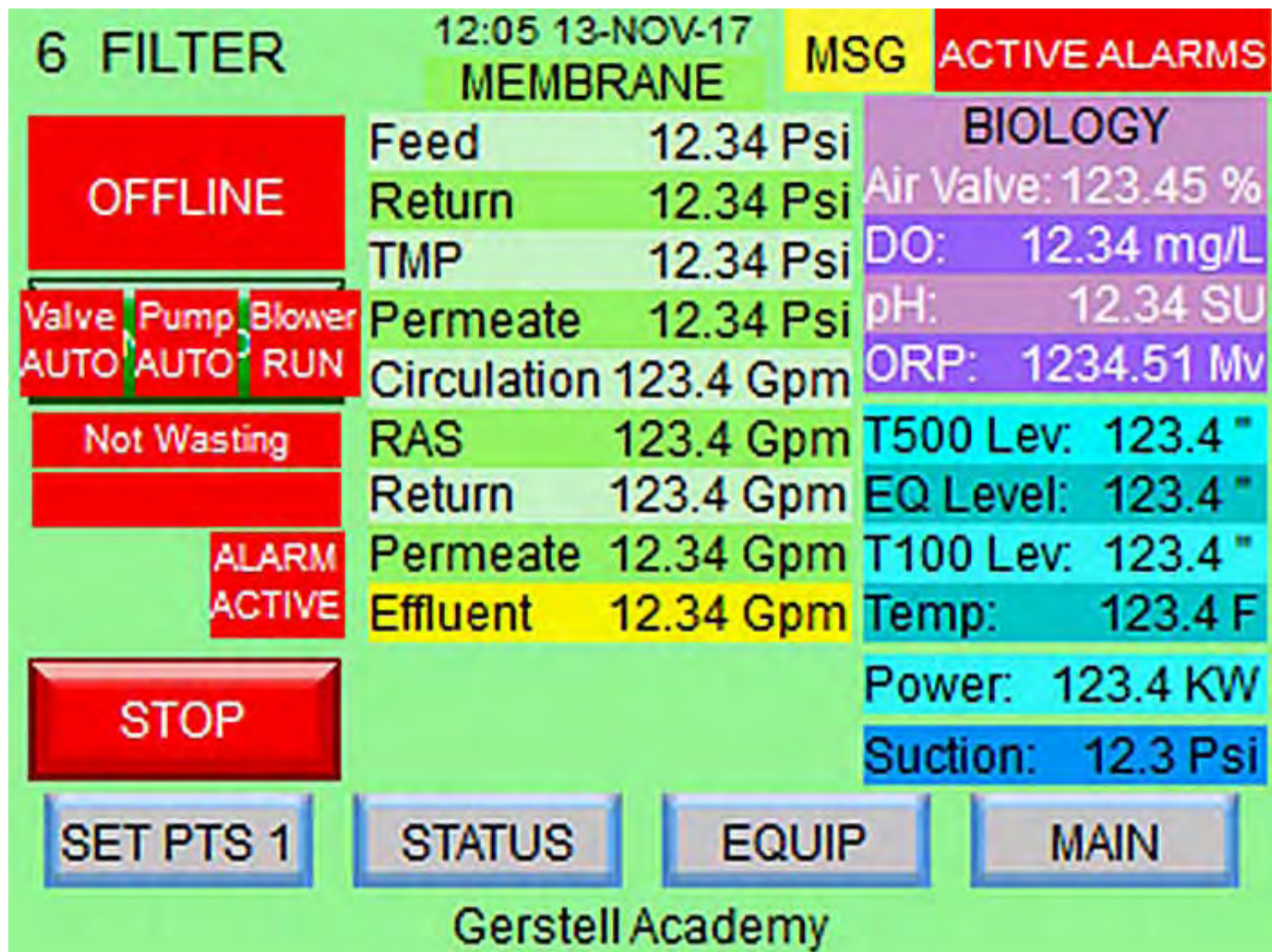

Figure 8N: Complete control via 130 HMI screens. By designing the graphics for a small display, everything is also easily readable remotely on a smartphone. (Source: ITP)

There are approximately 130 HMI screens in the system, and ITP has put a lot of effort into making them easy to access. We try to make them discoverable by asking, "When would operators need this function, and where would they tend to look for it?"

Every system has a built-in cellular modem, which gives technicians remote access from the first day, dramatically reducing startup costs. Every important parameter is trended, even a few the operators don't see. This data often helps to identify sources of problems. Data is logged continuously to a memory stick and is available online via any Web browser.

Results

The first WWTP built by ITP was installed to support an old industrial building that contained about 20 small businesses. The building owner said it cost half the price of the other units under consideration and brought an end to a consent decree with the state, which was imposing huge fines.

The second unit was supplied in a fiberglass building and tank structure for immediate installation and operation. It was ready to run except for adjusting the operating parameters at delivery. It replaced an old plant that had never worked properly in 15 years, so the owners were trucking wastewater away twice a week. The next two systems replace two more failed systems, each only five years old.

The reason so many conventional small plants fail to work is a lack of scalability. They are designed to handle full "nameplate" flow, which usually includes a huge safety margin. Once installed, most of them operate at about 10% of capacity much of the time, and the processes can't break down the effluent properly at such reduced rates.

ITP's process design and advanced control system produces high discharge quality while operating as low as 5% of capacity. AutomationDirect control hardware is a very large part of the success. While there are sometimes pressures to use the industry's market leader for the automation system—the added cost of equipment, software and support would make it impossible for many customers to afford our systems, so ITP plans to stick with what works for the company and its customers.

Results

The first WWTP built by ITP was installed to support an old industrial building that contained about 20 small businesses. The building owner said it cost half the price of the other units under consideration and brought an end to a consent decree with the state, which was imposing huge fines.

The second unit was supplied in a fiberglass building and tank structure for immediate installation and operation. It was ready to run except for adjusting the operating parameters at delivery. It replaced an old plant that had never worked properly in 15 years, so the owners were trucking wastewater away twice a week. The next two systems replace two more failed systems, each only five years old.

The reason so many conventional small plants fail to work is a lack of scalability. They are designed to handle full "nameplate" flow, which usually includes a huge safety margin. Once installed, most of them operate at about 10% of capacity much of the time, and the processes can't break down the effluent properly at such reduced rates.

ITP's process design and advanced control system produces high discharge quality while operating as low as 5% of capacity. AutomationDirect control hardware is a very large part of the success. While there are sometimes pressures to use the industry's market leader for the automation system—the added cost of equipment, software and support would make it impossible for many customers to afford our systems, so ITP plans to stick with what works for the company and its customers.