BASICS OF PROGRAMMABLE LOGIC CONTROLLERS

Programmable logic controllers provide dependable, high-speed control and monitoring demanded by a wide variety of automated applications.

Before the automotive industry discovered the advantages of PLC's, the process of modifying relay circuitry was a headache-inducing endeavour. In the past, annual car model changes forced plant engineers to constantly modify production equipment managed by relay circuitry. In some cases, the engineers had to scrap entire relay-controlled panels and replace them with completely redesigned systems. Now, PLC's allow engineers to implement numerous manufacturing changes with relative ease, which reduces changeover costs and downtime.

Prior to PLC's, contactor or relay controls solved many of these control tasks. This is often referred to as hard-wired control. Electricians had to design circuit diagrams, specify and install electrical components, and create wiring lists before wiring the components necessary to perform a specific task. Design errors would force the electrician to trace the wires to identify the problem and then reconnect the wires. A change in function or a system expansion required extensive component changes and rewiring.

Now, PLC software programming makes wiring changes between devices and relay contacts easier. Although hard wiring is still necessary for connecting field devices, it's less intensive than before.

What is a PLC?

The National Electrical Manufacturers Association (NEMA) defines a PLC as a "digitally operating electronic apparatus which uses a programmable memory for the internal storage of instructions by implementing specific functions, such as logic, sequencing, timing, counting, and arithmetic to control through digital or analogue I/O modules various types of machines or processes."

Basically, it's a solid-state, programmable electrical/electronic interface that can manipulate, execute, and/or monitor, at a very fast rate, the state of a process or communication system. It operates on the basis of programmable data contained in an integral microprocessor-based system.

A PLC is able to receive (input) and transmit (output) various types of electrical and electronic signals and can control and monitor practically any kind of mechanical and/or electrical system. Therefore, it has enormous flexibility in interfacing with computers, machines, and many other peripheral systems or devices.

It's usually programmed in relay ladder logic and is designed to operate in an industrial environment.

How does a PLC work?

To know how the PLC works, it is essential that we have an understanding of its central processing unit's (CPU's) scan sequence. The methodology basically is the same for all PLC's. However, as special hardware modules are added into the system, additional scanning cycles are required.

A PLC interfaces numerous types of external electrical and electronic signals. These signals can be AC or DC currents or voltages. Typically, they range from 4 to 20 milliamperes (mA) or 0 to 120VAC, and 0 to 48VDC. These signals are referred to as I/O (input/output) points. Their total is called the PLC's I/O capability. From an electronic point-of-view, this number is based on how many points the PLC's CPU is able to look at, or scan, in a specified amount of time. This performance characteristic is called scan time. From the practical perspective of the user, however, the number of I/O modules needed as well as the number of I/O points contained on each I/O module will drive what the system's I/O capability should be.

It's important to have sufficient I/O capability in your PLC system. It's better to have more than less so that, when more I/O points are required at a future time, it's easier to write the existing spare I/O points into the software (since the hardware is already there). There's no harm to the operating system in having spare I/O points; the software can be programmed to ignore them, and these points will have a negligible effect on the PLC's scan time.

The PLC's software program

The software program is the heart of a PLC and is written by a programmer who uses elements, functions, and instructions to design the system that the PLC is to control or monitor. These elements are placed on individually numbered rungs in the relay ladder logic (RLL). The software's RLL is executed by the processor in the CPU module.

There are many types of PLC software design packages available. One frequently selected software package is of the RLL format and includes contacts, coils, timers, counters, registers, digital comparison blocks, and other types of special data handling functions. Using these elements, the programmer designs the control system. The external devices and components are then wired into the system identical to that of the programmer's software ladder logic. Not all of the software elements will have a hard-wired, physical counterpart, however.

As the PLC's processor scans (top down) through the software program (rung-by-rung), each rung of RLL is executed. The hard-wired device that the software is mirroring then becomes active. The software is thus the controlling device and provides the programmer or technician the flexibility to either "force a state" or "block a device" from the system operation. For example, a coil or contact can be made to operate directly from the software (independent of the control cabinet's hard-wiring to source or field input devices). Or, a device can be made to appear invisible (removed from the system's operation), even though it's electrically hard-wired and physically in place.

Peripheral devices

Peripheral devices to the PLC and its I/O base(s) can be anything from a host computer and control console to a motor drive unit or field limit switch. Printers and industrial terminals used for programming are also peripheral devices.

These external operating devices, with their sometimes harsh and/or fast signal characteristics, must be able to interface with the PLC's sensitive microprocessor. Various types of I/O modules are available to do this job.

Input module

The input module has two functions: reception of an external signal and status display of that input point. In other words, it receives the peripheral sensing unit's signal and provides signal conditioning, termination, isolation and/or indication for that signal's state.

The input to an input module is in either a discrete or analogue form. If the input is an ON-OFF type, such as with a push button or limit switch, the signal is considered to be of a discrete nature. If, on the other hand, the input varies, such as with temperature, pressure, or level, the signal is analogue in nature.

Peripheral devices sending signals to input modules that describe external conditions can be switches (limit, proximity, pressure, or temperature) push buttons, or logic, binary coded decimal (BCD) or digital-to-digital (A/D) circuits. These input signal points are scanned, and their status is communicated through the interface module or circuitry within each individual PLC and I/O base.

Output module

The output module transmits discrete or analogue signals to activate various devices such as hydraulic actuators, solenoids, motor starters, and displays the status (through the use of LEDs) of the connected output points. Signal conditioning, termination, and isolation are also part of the output module's functions. The output module is treated in the same manner as the input module by the processor.