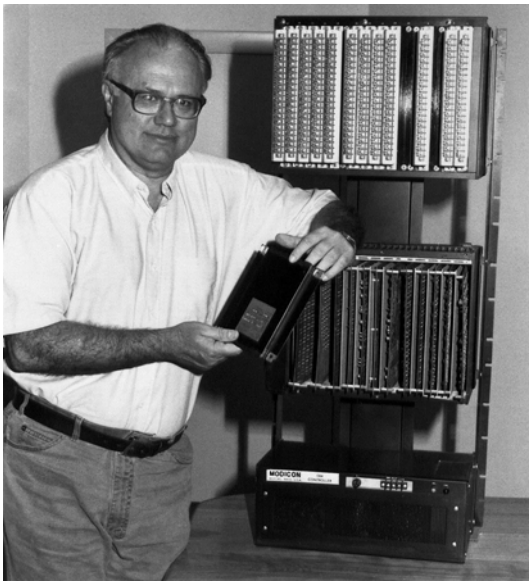
The Rise of the Programmable Automation Controller

Written by: John Wilhite, Advantech Corporation, Industrial Automation Group

So, What's a PAC?
It was only thirty years ago that most industrial processes were controlled either by hardwired relay logic or analog loop controllers. It was only thirty years ago that the Space Shuttle's three computer systems had less than 100 kilobytes of RAM and ran their entire complex programs in that space. It was only thirty years ago that there was no such thing as a PC.

In the last thirty years, we've come very far from hardwired relays and analog controllers, and computers have changed beyond the belief of anyone in the late 1970s.

First Came the PLC
In the 1970s, microprocessors were slow, expensive, and prone to failures. Memory was expensive. Richard Morley, at Bedford Associates (later Modicon) and Otto Struger of Allen-Bradley faced the challenge of how to eliminate the costs associated with redesigning and rewiring hardwired relay logic in the automotive industry by producing a device called the Programmable Logic Controller, or PLC. "It was always a computer," Morley recalls, "but we called it a controller so we didn't scare the operators." PLCs were able to easily replicate the logic of hardwired relays through the use of "ladder programming" that resembled the single-line wiring diagrams plant electricians were used to. With the automotive industry in the lead, PLCs became ubiquitous since their cost savings were enormous.



PLCs revolutionized control of the discrete manufacturing industries, at the same time that what was originally called the "personal computer" or PC was revolutionizing the control of the office and the enterprise.

Then Came the PC
Thirty years ago, office computing was done on minicomputers and mainframe systems. Most were standalone, but there were experiments ongoing toward networking computer systems together. Software to do simple word processing, or a spreadsheet cost in excess of $10,000 per seat license (as contrasted to today's Open Office, which is a no-charge download). Memory and storage were expensive and at a premium. Then Apple, IBM, and a host of other manufacturers created the simple and inexpensive "personal computer." By the end of the 1980s, the personal computer had invaded the enterprise space, and by the end of the 1990s, it was the standard for computing in business and manufacturing worldwide.

Then Came the PAC
Costs had come down, memory and processor power increased, and reliability had become significant. It became less expensive to build a PC single board computer than to build a PLC. This made companies and users investigate the usability of PCs to replace the traditional PLC. By the early 2000s, these experiments had coalesced into something called a PAC, for programmable automation controller. ARC Advisory Group's Craig Resnick is credited with coining this name for a personal computer-type architecture used as a replacement for a ladder-logic programmed PLC. In 2002, he wrote that a PAC included:

- "Multi-domain functionality, including logic, motion, drives and process on a single platform
- "Single multi-discipline development platform incorporating common tagging and a single database
- "Software tools that allow the design by process flow across several machines or process units
- "Open, modular architectures that mirror industry applications from machine layouts in factories to unit operations in process plants
- "[Employment of] de-facto standards for network interfaces, languages, etc., allowing data exchange as part of networked multi-vendor systems"

Where Do You Use a PAC?
PACs have significant advantages over the older, more limited PLC designs. These include:

- COTS (Commercial Off The Shelf)networking to higher-level platforms
- Wireless and fieldbus connectivity
- Interfaces through multiple protocols
- Modern, fast, COTS CPUs with outstanding processing speed and math coprocessors
- Use of COTS operating systems like DOS, Linux, Windows CE
- Memory and storage essentially unlimited
- HMI functionality in one platform
- Advanced control algorithms
- Extensive database manipulation
- Integrated custom control routines
- Complex process simulation

PACs, in fact, can be used in nearly every application where hardwired relays, analog controllers, or PLCs were and are used. The few exceptions are where very fast motion control is required, or for safety instrumented systems.

And in the past decade, PACs have come to replace PLCs in many applications. Some PLC vendors make PACs, and some continue to label as PLCs devices which in fact meet the criteria for PACs. It's all in the name, these days.

The Embedded Computing Revolution

PACs are part of the revolution called "embedded computing" where nearly every device has some sort of computer device built into it, from tires to toasters. PACs are simply embedded computers used as industrial automation controllers.

PACs use commercial operating systems, either as-found or slightly customized for use in embedded computing. This means that PACs leverage off of the growth of the entire embedded computing market, not just the industrial automation market, and the improvements and development of embedded computing products are immediately reflected in the capabilities of PACs.

With a PAC, You Can Pick Your OS

PACs can be supplied with either DOS or Windows-based operating systems, as well, as in some cases, with a Linux OS. DOS, even though it is not new, still have some significant advantages over the others. In the first place, DOS is not being updated significantly, like Windows and Linux are, and therefore any code written for a DOS-based PAC will not be obsolete, or potentially damaged by updates and patches. DOS is more cost effective, too, and allows the end user or machine builder to operate with lower power requirements, and have a lower requirement for heat dissipation. Fanless PAC operation is easier to achieve with a DOS-based PAC.

On the other hand, Windows CE-based PACs can leverage most of the capabilities of the Windows universe for applications, peripherals, and networking. Windows permits the user to connect a touch panel or flat panel monitor to the VGA port of the controller. In fact, PACs are often built as a composite unit, with display and controller in the same housing.

Users can install panel software on a controller and use a less expensive flat panel instead of a touch screen, too.

The use of Windows CE for the PAC's operating system also allows the use of specialty modules that DOS does not allow, including Ethernet networking, and USB connectivity.

Picking a PAC

Selecting a PAC is much like selecting a PLC, or even a PC for your desktop. You need to select the hardware and operating system, and then you need to select the peripherals that go with them. Then you can select the applications you want to run on the PAC, or create them.

You need to start by selecting the PAC platform you want. Decide if you need low power operation, or if you are putting the PAC in a location where venting and fan cooling is not advisable—like on the factory floor in a dirty environment.

Decide if you want a DOS based PAC or if you want to go with Windows CE for the greater functionality and ability to use a touch screen or other Windows-based HMI.

You should also figure out what kinds of I/O you need, and how much. This will impact your choice of platform, as well as your choice of OS and peripherals. Select a PAC that will give you the correct number of digital inputs, digital outputs, analog inputs and analog outputs.

For example, items like push buttons, sensors, and switches should be counted and that's how many digital inputs are needed.

These digital inputs will cause the PAC to turn on items like fans, lights, motors, and should be counted and that's how many digital outputs are needed. When you measure voltage, current, or temperature, you need some form of an analog input. This can be a basic analog input or a module like a thermocouple or RTD module. To control an item like the speed of a motor or the speed of a conveyor, you will need analog outputs.

Make sure you have expansion capability in your system, too.

PACs have the ability to perform special operations, and have the ability to host specialty modules, too. Specialty modules for motion control, counter modules, additional memory and custom I/O modules are examples of this kind. This way, there is one controller and one operating system, controlling the entire operation.

Once You Have Hardware, Then Comes Software

There are three types of software in a PAC system. First, as we have discussed, is the operating system. Next is the programming software. Finally, there are the application software programs that can run on the PAC.

Most people who are familiar with PLCs are used to programming in either ladder or one of the IEC 61131-3 programming languages. Most people who are familiar with personal computers are used to programming in higher order programming languages like Visual Basic, C++, Visual C, and others. One of the significant advantages of the PAC over the PLC is that there are many thousands more programmers familiar with these higher order programming languages than can program in ladder. Users and machine builders can therefore leverage the much larger code base for Windows and Windows CE-based software by using PACs.

At the same time, it is useful to be able to use the PLC-centric programming languages like ladder, or statement list, or FBD. A PAC is capable of both kinds of programming.

PACs are also ready-made to operate as RTUs (remote terminal units) in a SCADA system. They can even run the HMI themselves, and operate as peer-to-peer nodes in a distributed network control system.

Windows CE-based PACs are designed for easy integration into standard Ethernet networks, using industrial grade Ethernet switches.

The other leverage that comes with using a Windows based PAC is the ability to run COTS (Commercial off the Shelf) programs and applications on the same machine that is operating as a process controller. Integrating control with data historian and HMI is really easy with a PAC, where it is considerably less so with a traditional PLC design.

In the past decade, programmable automation computers have proved themselves, leveraged themselves on COTS hardware and software, and become rugged, reliable, versatile, and ubiquitous. Often it is not possible to tell the difference between a PLC and a PAC, from the outside. It is only when you get to see the advantages built into a PAC from the inside out that you can tell.

###

