Software for Programmable Automation Controllers (PACs)

Programmable automation controllers (PACs) meet the complex demands of modern industrial automation applications because they combine features of more traditional automation technologies: programmable logic controllers (PLCs), distributed control systems (DCSs), remote terminal units (RTUs), and personal computers (PCs).

If you're looking at PACs for your monitoring, automation, and data acquisition applications, you may be wondering how PACs are programmed. This white paper explores some of the most important features of software for PACs.

According to the ARC Advisory Group, generally credited with coining the term *PAC*, among a PAC's defining characteristics are three elements directly related to software:

- **Tightly integrated controller hardware and software**. In other words, the software used with a programmable automation controller is designed specifically for the PAC.
- **A single development platform**, using common tagging and a single database for development tasks across a range of disciplines.
- Programmability using software tools capable of designing control programs to **support a process that "flows**" across several machines or units.

Let's take a closer look at what these three characteristics mean as you develop your system.

Tightly Integrated Hardware and Software

When hardware and software are designed together, systems become easier and faster to build. Because no drivers are required, and the parts of the system are built to work together, it's not necessary to debug driver problems or fix incompatibilities.

If problems occur, you have just one company to call for product support or one website to visit for information. Documentation is often more complete, as well.

In some cases lower software cost may be another plus, as some PAC manufacturers, such as Opto 22, include surprisingly capable automation software with a PAC purchase. (See "PAC Project Software Suite from Opto 22" on page 3 for more information.)

A Single Development Platform

The software built for a PAC is not just integrated with the hardware it runs on. It's also internally integrated: it provides not only an integrated development environment (IDE) for programming but also a suite of related programs for HMI (human machine interface) development and other purposes.

The IDE is a single software program that handles everything related to control programming, such as editing, compiling, debugging. A software suite, made up of two or more software applications, offers a similar look and feel in all of them, so that familiarity with one helps you use the others more easily.

More importantly, the software applications in the suite work together behind the scenes in ways that significantly reduce development time.

Common tagging means that names and definitions you set up in one of the suite's applications are also used in the others. For example, if you define a string variable in the control development software, that same definition will be used in the human-machine interface (HMI) development software. If you name a digital I/O point in the control software, that name will automatically appear when you're configuring OPC data communications.

Because all these common tags you define are kept in a *single database* that all the applications in the software suite use, you don't have to reenter tags or maintain and reconcile lists of them. As a result, development tasks can be finished more quickly and easily.

Supporting a Process that Flows

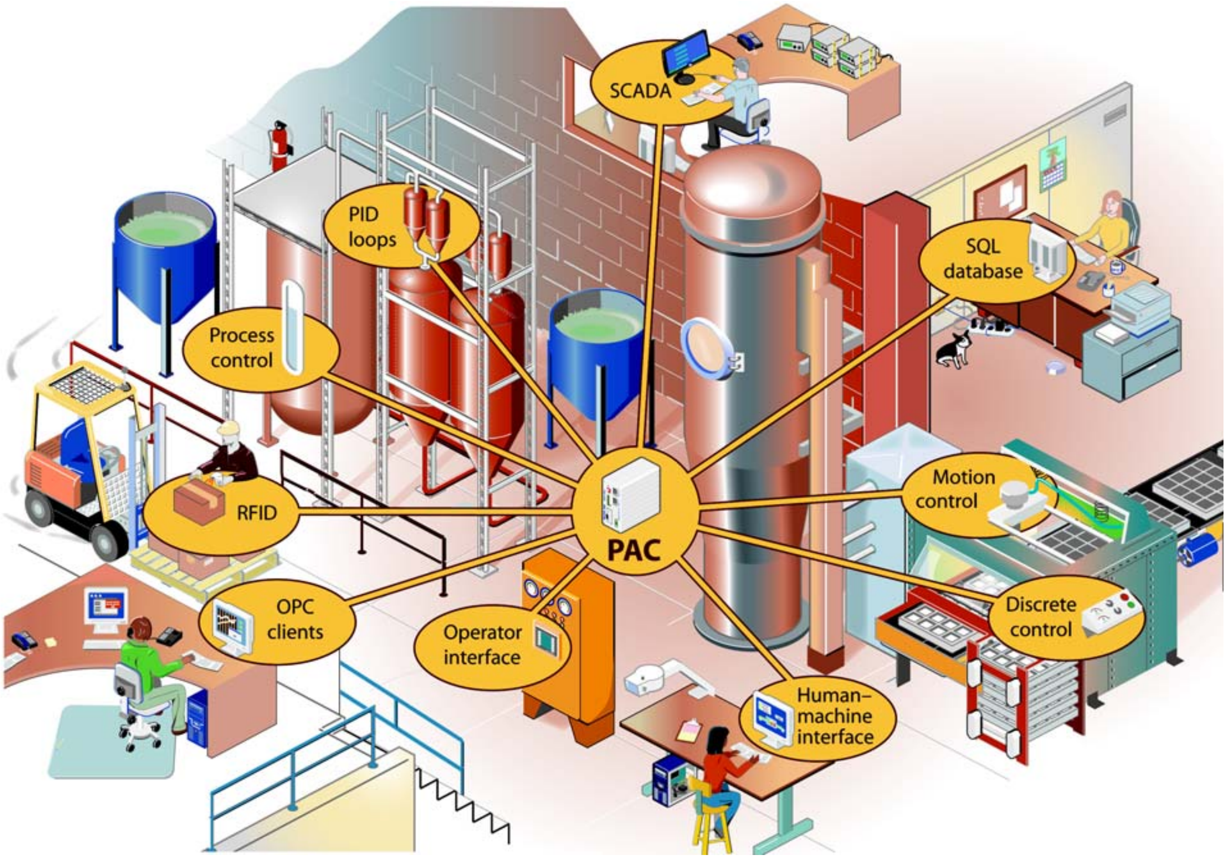
Because a key defining characteristic of PACs is that the same hardware can be used in multiple domains, including logic, motion, drives, and process control, it follows that the software must be capable of programming all control and monitoring tasks that must be done in multiple domains.

That means that the PAC software must handle discrete control, process control, motion control, remote monitoring, and data acquisition. And the software must let the developer mix and incorporate these as needed into control programs, so these programs can "flow" as the requirements of the application dictate.

Example: Microbrewery

For example, suppose your company is a microbrewery. Here are just some of the requirements for producing your end product:

- Water is piped in from a spring a couple of miles away, so you need to **monitor the pressure and flow** of that water as well as security at the spring (remote monitoring using analog and digital devices).
- You measure water quality as it enters your facility, **track this data** over time, **and store it** in your company database (data acquisition, database connectivity).
- You make more than one microbrew, so **recipes**, temperatures, and processing must vary (batch process control, **PID loop control**, distributed control).
- **Operator interfaces** mimic the process, providing secure interactive controls for technicians and operators.
- Quality control is essential to your reputation, so you test all products at several stages. **Quality data is kept as required** by government health authorities (monitoring, more data acquisition and database connectivity).
- In another building, the bottling line requires **discrete control**. As bottles come off the line, they are boxed and identified with radio-frequency identification (**RFID)** tags, then sent on to shipping.



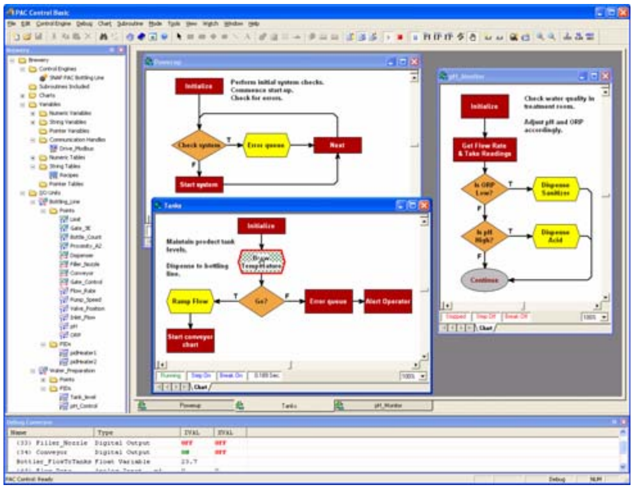
- In the separate shipping area, boxed stock automatically moves via conveyors (discrete control) based on RFID tags (**serial device connectivity**).
- Temperatures in the storage area are controlled and monitored. Energy usage is monitored and building systems controlled throughout all buildings (**remote I/O, distributed intelligence**).
- **Production and inventory data** go directly from machines and barcode readers to company computers; customer and shipping data flows in the opposite direction (**database connectivity**).

This microbrewery is just an example of how several different types of control in several different *domains*, to use ARC's term, are required by a modern industrial automation application.

Most industrial applications today are similarly varied. While the number of PACs needed will depend on application requirements, each PAC can be used in any domain or in multiple domains. Because the application requires processes that flow into each other over space and time, the PAC software accommodates that flow and integrates these multiple domains into one system.

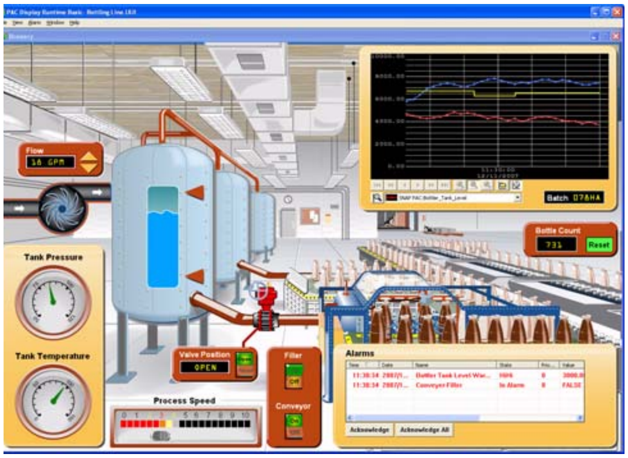
PAC Project Software Suite from Opto 22

Opto 22's PAC Project software suite is an example of PAC software. Fully integrated with SNAP PAC controllers, the suite includes control programming and HMI development software, plus optional OPC server and database connectivity software.



Flowchart-based PAC Control development software from Opto 22 is fully integrated with SNAP PAC programmable automation controllers.

I/O points and variables you define while building a control program (called a control *strategy*) are stored in a single tagname database. When you open the HMI development, OPC server, or database connectivity software, those defined items are immediately available for use.



PAC Display HMI development software uses the same database of I/O points and variables as PAC Control.

Control programming incorporates multiple domains by including commands for digital and analog control; PID loop control; logical and mathematical operators; string handling; time/date and timers; event/reactions; communication with distributed I/O, peers, and computers on the network; testing and error handling. Motion control commands are also available. All these commands are in plain English and can be used in the control strategy you build, both in flowcharts and in scripting blocks within a flowchart.

For more information about PAC Project software, visit the Opto 22 website, www.opto22.com.

More About Opto 22

Products

Opto 22 develops and manufactures reliable, flexible, easy-to-use hardware and software products for industrial automation, remote monitoring, and data acquisition applications.

SNAP PAC System

Designed to simplify the typically complex process of understanding, selecting, buying, and applying an automation system, the SNAP PAC System consists of four integrated components:

- SNAP PAC controllers
- PAC Project™ Software Suite
- SNAP PAC brains
- SNAP I/O™

SNAP PAC Controllers

Programmable automation controllers (PACs) are multifunctional, multidomain, modular controllers based on open standards and providing an integrated development environment.

Opto 22 has been manufacturing PACs for many years. The latest models include the standalone SNAP PAC S-series and the rack-mounted SNAP PAC R-series. Both handle a wide range of digital, analog, and serial functions and are equally suited to data collection, remote monitoring, process control, and discrete and hybrid manufacturing.

SNAP PACs are based on open Ethernet and Internet Protocol (IP) standards, so you can build or extend a system without the expense and limitations of proprietary networks and protocols.

PAC Project Software Suite

Opto 22's PAC Project Software Suite provides full-featured and cost-effective control programming, HMI (human machine interface) development and runtime, OPC server, and database connectivity software to power your SNAP PAC System.

These fully integrated software applications share a single tagname database, so the data points you configure in PAC Control™ are immediately available for use in PAC Display™, OptoOPCServer™, and OptoDataLink™. Commands are in plain English; variables and I/O point names are fully descriptive.

PAC Project Basic offers control and HMI tools and is free for download on our website, www.opto22.com. PAC Project Professional, available for separate purchase, adds OptoOPCServer, OptoDataLink, options for Ethernet link redundancy or segmented networking, and support for legacy Opto 22 serial *mistic*™ I/O units.

SNAP PAC Brains

While SNAP PAC controllers provide central control and data distribution, SNAP PAC brains provide distributed intelligence for I/O processing and communications. Brains offer analog, digital, and serial functions, including thermocouple linearization; PID loop control; and optional high-speed digital counting (up to 20 kHz), quadrature counting, TPO, and pulse generation and measurement.

SNAP I/O

I/O provides the local connection to sensors and equipment. Opto 22 SNAP I/O offers 1 to 32 points of reliable I/O per module, depending on the type of module and your needs. Analog, digital, serial, and special-purpose modules are all mixed on the same mounting rack and controlled by the same processor (SNAP PAC brain or rack-mounted controller).

Quality

Founded in 1974 and with over 85 million devices sold, Opto 22 has established a worldwide reputation for high-quality products. All are made in the U.S.A. at our manufacturing facility in Temecula, California. Because we do no statistical testing and each part is tested twice before leaving our factory, we can guarantee most solid-state relays and optically isolated I/O modules for life.

Free Product Support

Opto 22's Product Support Group offers free, comprehensive technical support for Opto 22 products. Our staff of support engineers represents decades of training and experience. Product support is available in English and Spanish, by phone or email, Monday through Friday, 7 a.m. to 5 p.m. PST.

Free Customer Training

Hands-on training classes for the SNAP PAC System are offered at our headquarters in Temecula, California. Each student has his or her own learning station; classes are limited to nine students. Registration for the free training class is on a first-come, first-served basis. See our website, www.opto22.com, for more information or email training@opto22.com.

Purchasing Opto 22 Products

Opto 22 products are sold directly and through a worldwide network of distributors, partners, and system integrators. For more information, contact Opto 22 headquarters at 800-321-6786 or 951-695-3000, or visit our website at www.opto22.com.