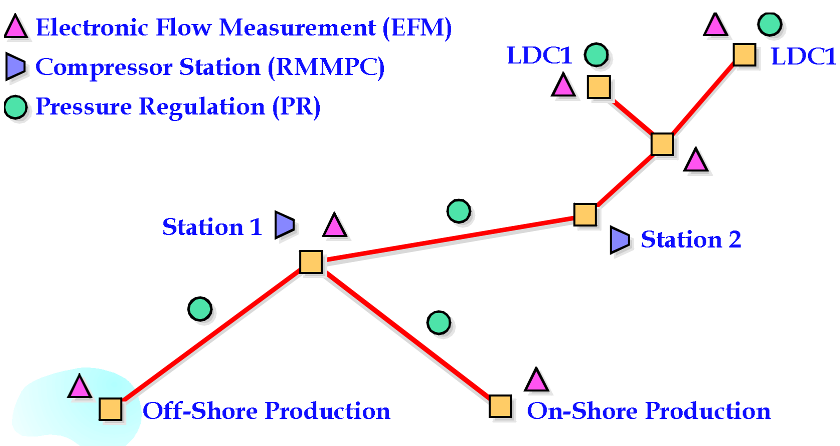
PC-Based Automation in the Pipeline Industry

James C. Bowman, Opto 22 Industry Manager, Energy
Austin, TX 78759

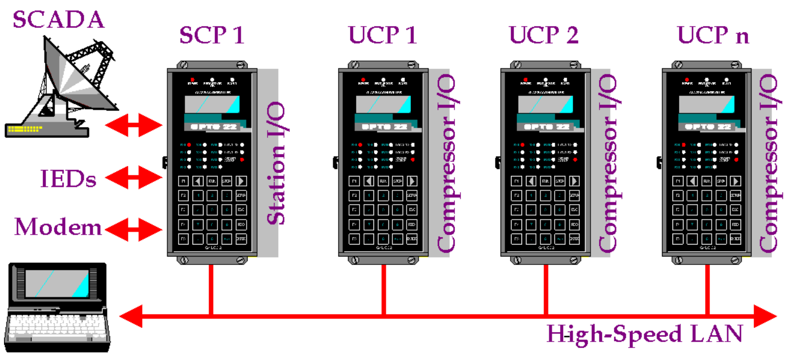
The benefits of pipeline automation have been known for years. But recently, competitive pressures and deregulation have forced pipeline operating companies to re-address their commitment to automation. At the same time, the personal computer (PC) has advanced from just being a useful office machine to an important control automation tool used in all aspects of the pipeline industry. Advances in PC operating system robustness, processor speed, and memory capacity have resulted in its wide use in industrial automation. The term PC-based Automation System (PCAS) has emerged from these advances. Packaged in a ruggedized form, PCAS hardware and software also includes integral electronics for processing instrumentation signal inputs and outputs (I/O). This technology can be used to form highly-intelligent solutions that will satisfy the most challenging pipeline automation requirements. Since most pipelines handle combustible fluids, there are several important aspects that need to be considered when searching for a PCAS solution. This paper assumes that any PCAS would be manufactured, packaged, and installed in such a way that all electrical and hazardous area requirements (Class I, Division 2, etc.) are satisfied. This paper examines four important PCAS aspects: connectivity, scalability, reliability, and supportability. While the following discussion focuses on natural gas pipeline applications, these general principles can be applied to any pipeline.



Typical Pipeline System
Figure 1

Introduction

Of all energy industry installations, pipelines require the most diversity in PCAS architecture. Figure 1 illustrates a small, single-node PCAS application that only requires a few I/O. This architecture is usually found in electronic flow measurement (EFM) systems which are typically located at production, sales, and exchange HUB sites, and pressure regulation (PR) systems, which are located at large end-user and local distribution company (LDC) transfer points. Other PCAS applications, such as those found at engine, turbine, or electric motor-driven compressor stations, require a large number of I/Os. Figure 2 shows how a multi-unit compressor station can be configured with multiple PCAS nodes working peer-to-peer on a high-speed local area network (LAN).

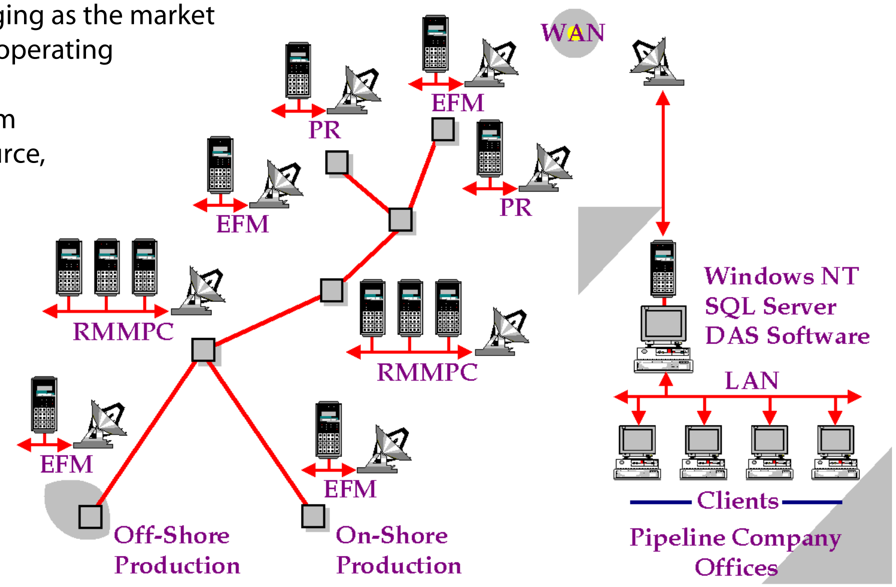


Typical Compressor Station PCAS
Figure 2

A compressor station PCAS often includes an additional PC located in the compressor station office or control room. This PC provides local supervisory control and data acquisition (SCADA) functions and a means for software support. Regardless of the field site architecture, the PCAS is usually connected, by some form of telecommunications, to a central SCADA master. The SCADA master, which may or may not be PC-based, is used by the Gas Control Department to control and optimize the overall pipeline.

As this subject involves the use of PC software, now is a good time to broach the subject of PC operating systems and Microsoft® Windows NT®. This commercial off-the-shelf, multitasking operating system has matured. Combined with Microsoft's suite of "back-office" software, Windows NT promises to be an excellent choice for any enterprise-wide PCAS. There are other multi-tasking operating systems available, and all of them have merit, but it seems as though Windows NT is emerging as the market leader in multitasking PC operating systems.

All PC operating system software, regardless of source, is designed to support other PC application software. Consequently, PCAS software should be written in accordance with the operating system vendors' guidelines. If the operating system is Windows NT, then PCAS software should be written to take advantage of dynamic link libraries (DLLs) and Object Linking and Embedding (OLE). If this is done properly, then information can easily flow back and forth between any PCAS node and any other DLL or OLE-compliant PC application software (SQL database management, word processing, spreadsheet, process modeling, etc.).



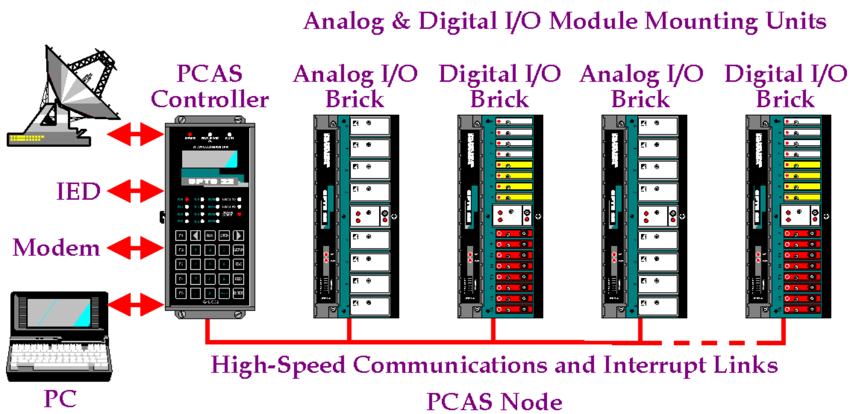
Enterprise-Wide PCAS
Figure 3

PCs are everywhere in today's pipeline enterprise. Figure 3 illustrates how this is feasible, using a wide area network (WAN), to interconnect each PCAS node in the field to a PC located in the pipeline's regional and central business offices, thereby forming an enterprise-wide PCAS. The benefits of an enterprise-wide PCAS solution are obvious. Gas control uses spot-flow rates from all EFM sites, as well as unit and gate valve statuses from all compressor stations, to effectively supervise pipeline operation. Gas accounting needs a traceable history from many of the same EFM sites to facilitate sales invoicing and gas purchase requirements. Engineering and maintenance personnel can use the more detailed operational and historical information available from each compressor station's PCAS. With the enterprise-wide PCAS providing the right real-time information to field and office personnel, more informed decision making should result in maximized profits and minimized operating costs.

Connectivity

If the enterprise-wide PCAS is to meet expectations, each field site's PCAS node must reliably communicate with the rest of the system. Therefore, connectivity is the ability of any PCAS node to communicate with the equipment at its own site and with the rest of the enterprise using the WAN.

In the field, the PCAS node consists of its controller and its I/O bricks. Figure 4 shows how the PCAS controller is responsible for connectivity at the site. In addition to its own I/O bricks, the PCAS must communicate with other site-based intelligent electronic devices (IEDs), and with the WAN through telecommunications equipment residing at the site. The IEDs may be a simple printer or display terminal, or a complex gas chromatograph. The gas chromatograph, using a serial communications port on its own controller, can provide the details of gas quality at a particular site. An IED may also be an electronic temperature scanner that monitors engine and compressor temperatures and has serial communications capabilities. As stated earlier, there will always be the need to communicate with an ordinary PC. This may be a portable notebook PC used exclusively for system support, or it may be a desktop PC that is used for both system support and local SCADA interface.



PCAS Node
Figure 4

Connectivity also means reliable information. The labels in Figure 5 indicate the connectivity features inherent in the PCAS controller. The PCAS's ability to perform CRC and LRC error detection checks on all communications transactions is of particular

- Full 32 bit CPU
- Math Coprocessor
- ROM .5 to 1MB
- RAM .5MB to 4MB
- Battery-Backed RAM
- Flash-Based ROM

- RS-232 Ports (2 ea.)
- RS-485 ports (2 ea.)
- LAN Port
- Two and Four Wire
- CRC Error Detection
- LRC Error Detection

- CPU Interrupt Link

- Alpha-Numeric LCD
- Network Activity LEDs

- Factory Mutual approval for Class 1: Div II Areas

- Built in keypad for easy operator interface

- Three level password protection for maximum security

Remote Site PCAS Controller
Figure 5

importance. This feature assures the integrity of the information to be shared (using the enterprise-wide PCAS) with field and office personnel.

Scalability

Scalability of a PCAS is related to its capability to be effectively scaled from a small to a large system without suffering from performance degradation. This capability is the result of the PCAS controller's modularity, its intelligent components, and in particular, its intelligent analog and digital I/O bricks.

Modularity

As illustrated in Figure 4, the typical PCAS always uses one or more single-point analog, or digital I/O bricks and I/O modules to interface the PCAS with the instrument and control signals at the site. The I/O modules are plug-in devices used to condition and adapt input signals (from transmitters, thermocouples, RTDs, limit switches, magnetic pickups, etc.) and output signals (to control devices such as I/P converters, solenoids, etc.) to and from the PCAS. I/O bricks are devices with multiple slots where I/O modules can be plugged in and field signal wiring can be terminated. Single point I/O bricks allow input and output modules to be plugged into the same device. Figures 6 and 7 show both types of I/O bricks. The labels emphasis the modularity of each.

Intelligence

The best way to think of the PCAS controller (Figure 4) is to visualize it as an industrially-hardened computer with 32-bit processing power, multiple communications ports, a display, and a keyboard for local operator interface. It functions as a network controller for all attached I/O bricks, as a communications hub for various site-based IEDs, as a SCADA node, and as a depository for important measured and calculated information.

To do all these tasks simultaneously requires a multitasking operating system. The PCAS controller can simultaneously execute up to 30 complex tasks, including an I/O-driven interrupt task. The PCAS controller also has single precision, IEEE floating-point math capabilities to assist with heavy-duty number crunching. This feature is essential in gas flow calculations, plant or component efficiency calculations, and other similar mathematical exercises that are required when implementing advanced compressor station optimization strategies.

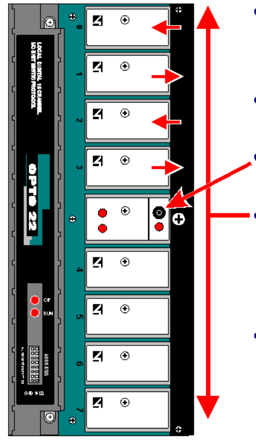
When gathering control information, such as operational history files (flow trace, alarm logs, process variables, etc.), the PCAS controller has 500KB of random access memory (RAM), which can be expanded to 4MB. In addition to historical files, the site-specific applications program also runs in RAM. Its operating system resides in "flash" RAM, which is a non-volatile memory ranging from 500KB to 1MB. The "flash" memory allows PCAS operating system enhancements to be loaded directly from a PC without the annoying process of changing EPROMs in the field.

Regardless of how powerful the PCAS controller is, the only way to achieve scalability without losing performance is by adding processing power as the PCAS is being scaled up. This is accomplished with brain boards that are an integral part of each analog and digital brick (Figures 6 and 7). The I/O brick brain board is built around a 16-bit micro-processor with enough local memory to handle a variety of I/O control tasks. In fact, all I/O control in the PCAS is handled at the individual I/O brick level. Table 1 provides a list of some of the important intelligent functions that can be configured for execution by any 16-channel analog or digital I/O brick in the PCAS.

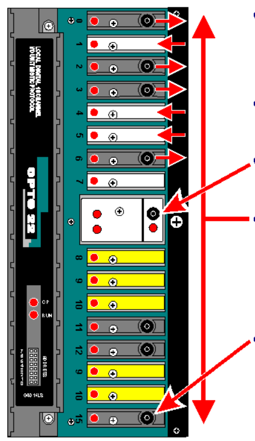
Reliability

The PCAS controller has RAM with a five-year battery backup. To provide further reliability, the automation strategy can be stored in the PCAS controller's "flash" read-only memory.

All I/O modules provide 4,000 VRMs of optical isolation between the instrument or control end device and the I/O brick channel to which it terminates. It also provides 4,000 VRMs of optical isolation between the I/O brick channel and all other I/O brick channels. This feature eliminates damage to the I/O brick if a short occurs at the end



Analog I/O Brick
Figure 6



Digital I/O Brick
Figure 7

Table 1- Functions that are executable on the I/O Brick

Single Point, Analog I/O Bricj Single, Point Digital I/O Brick	Single Point, Digital I/O Brick
Engineering Unit Conversions Input Latching	Input Latching
PID Looping Control Input Timing	Input Timing
HI/LO Limit Monitoring Input Counting (0-20 kHz)	nput Counting (0-20 kHz)
Thermocouple Linearization Input Totaling	Input Totaling
Filtering Output Timing	Output Timing
Input Totaling Pulse/Waveform Generation	Pulse/Waveform Generation
Running Average Calculation Time Proportional Output	ime Proportional Output
Ramping and Waveform Generation Built in Diagnostics	Built in Diagnostics
Square Root Extraction PCAS Controller Interrupt Capability	PCAS Controller Interrupt Capability
Programmable Offset and Gain Communications Watchdog Timer	Communications Watchdog Timer
Precalibrated Modules	------
Host Interrupt Capability	------
PCAS Controller Interrupt Capability	------
Communications Watchdog Timer	------

device. It also eliminates end device damage if a short occurs in the I/O brick. Therefore, an end device failure will not cause serious damage to the PCAS and reduce its availability. All analog I/O modules have the same optical isolation, plus 750 VRMs of transformer isolation. That way, ground loops are not a problem when the PCAS is used.

Event reactions are another intelligent feature inherent to I/O bricks. If a broken wire or system malfunction interrupts communications between the PCAS controller and an I/O brick, that bricks' communication watchdog timer can generate an event reaction, such as forcing the brick's outputs to a safe status. Event reactions can also be driven by analog I/O brick input values or by digital I/O brick input state changes. For example, consider an engine oil pressure switch input on the same digital I/O brick as the fuel supply solenoid output. An event reaction could be set that would automatically turn off the fuel supply solenoid anytime the oil pressure switch (fail safe) opens. The beauty of this is that it all occurs at the digital I/O brick without any intervention of the PCAS controller.

Analog and digital bricks also have the ability to activate an interrupt routine in the PCAS controller. This means that some PCAS controller routines can be automatically suspended in the case of certain events. Using the engine example again, consider a situation where a normal startup sequence is underway and a safety signal transitions to an unsafe condition. Instead of building a contingency into the startup sequence, an event reaction can suspend the startup, and at the same time, initiate a safety shutdown sequence.

Supportability

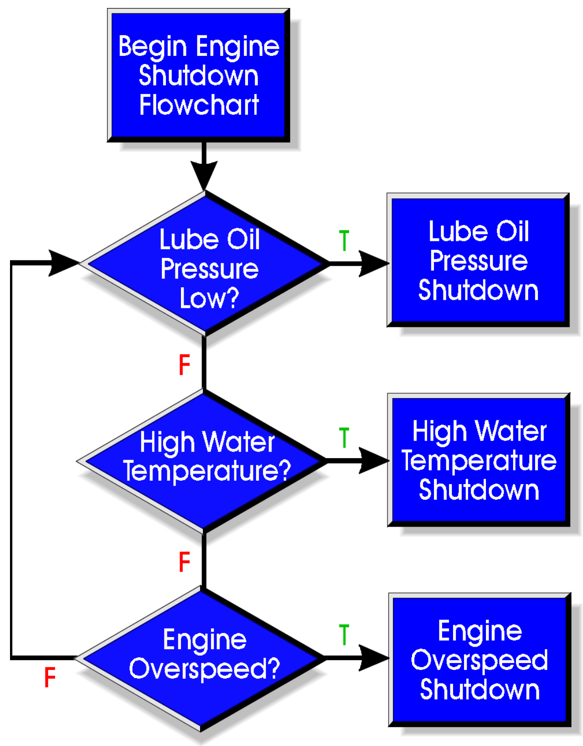
As pipeline companies attempt to downsize their field and administrative workforces, advancing technology continues to drive the demand for highly-skilled, technically-diverse operations personnel. The emergence of PC-based automation systems is driving the need for operating personnel who have the special skills needed to support them.

The dilemma is that operating budgets limit the number of people available in the operating departments. This results in existing technicians, mechanics, and operators who have to keep upgrading their skills to stay on top. The ideal solution is to deploy an automation system that can be easily understood and supported by existing personnel.

The PCAS has a modular design. It can be maintained on a modular basis. For example, if a digital or analog I/O module needs to be replaced, the service person need only remove one small screw, pull the module out of its socket, and replace it. Likewise, the I/O brick's brain board can be replaced by removing two screws and its cover, unplugging the old one, and plugging in a new one. Either of these replacements can be made while the rest of the system remains in service.

One area that provides the greatest support challenge is the software environment used to configure and support the system. Six months after an automation solution has been implemented, the person who programmed it should be able to come back and understand it as well. Any other system support personnel should be able to understand it too. One of the fundamental features of PCAS software is that it's easy to understand and use.

The subject of Windows NT was discussed earlier.



Engine Shutdown Flowchart
Figure 8

Figure 8 shows the easy use and understanding of flowchart programming.

Why use flowcharting? Because complex processes are easier to understand through pictures. Complexity is always one of the more challenging aspects of automation programming. Flowcharts were developed to graphically display and document processes in a simple, straightforward manner. It's easy to learn, and most engineers, technicians, and programmers are already familiar with it. The flowchart not only *represents* the automation solution, it *becomes* the solution.

First, PCAS software provides an intuitive user interface that graphically documents the control strategy. Flowcharts are created using a simple set of CAD-type drawing tools. Instructions can be inserted and edited using the standard Windows-type pull-downs, dialogs, and point-and-click commands. Using these tools, the programmer can choose from an extensive, plain English command set that handles many different types of data. This data can include integer and floating point variables, as well as ASCII strings that give the programmer complete control over all PCAS communications ports.

Then, using simple objects, the programmer lays out the logic of the automation solution by creating a flowchart. There are four constructs used in the layout.

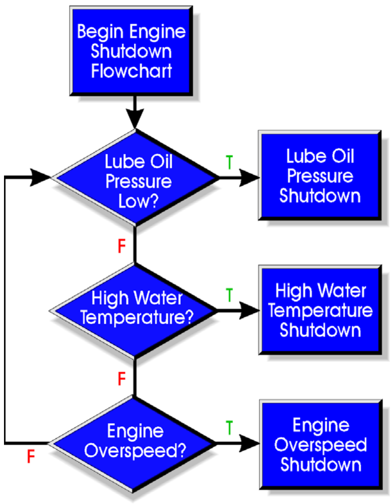
(1) Operation blocks (rectangles) contain a list of things to do.

(2) Condition blocks (diamonds) are used to test variable values or device states coming into an analog or digital I/O channel for true or false conditions.

(3) Connections (arrows) define the execution sequence of the operation and condition blocks.

(4) Continue blocks (ellipse) are used to jump to other areas of the flowchart without using any connections. The text tool allows the PCAS software to be a completely self-documenting application. This means that six months or six years later, the programmer or anyone else that needs to can understand the program logic and do what needs to be done.

In Figure 8, the simplicity of the PCAS software is readily apparent. In the case of the low oil pressure, notice the condition block labeled "Low Oil Pressure?." The true exit (T) is connected to an operation block labeled "Low Oil Pressure Shutdown." To make the flowchart become the solution, a condition command that asks the question, *"Is Low Oil Pressure Switch Off?"* is added to the condition block with the label "Low Oil Pressure?." An operations command "Turn Off *Engine Fuel Switch*," is added to the operations block labeled "Low Oil Pressure Shutdown." Once these commands are added to the condition and operation blocks, and the PCAS software downloads the commands, the strategy for low oil pressure is complete. The italicized tags Low Oil Pressure Switch and Engine Fuel Switch are simply the long, plain English names given to those I/O channels.



Engine Shutdown Flowchart
Figure 8

Once programming is complete, it's time to compile and download the strategy to the PCAS controller to test its functions. PCAS software has a unique debugging capability that, once downloaded, allows the flowchart to come to life. The debugger has single-stepping and auto-stepping functions that animate the flow-chart. It also has pull-downs and menus that allow the programmer to evaluate all I/O channels and variables at any step in the automation sequence. The integral debugging tool, coupled with the easy-to-understand flowchart, minimizes startup and personnel support.

In summary, enterprise-wide PC-based automation systems are today's reality. To gain the greatest benefit from this technology, ample attention should be given to enterprise-wide connectivity that allows all PCAS nodes to exchange information with any PC in the "back-office." Attention to scalability, reliability, and supportability will assure that PCAS technology will provide benefits far into the future. By embracing this technology, the pipeline industry can deliver better service to its customers and higher rates of return to its investors.

More About OPTO 22

LATEST PRODUCTS
PRODUCT SUPPORT
COMPANY INFORMATION

Products

Opto 22 produces a broad array of reliable, flexible hardware and software products for industrial automation, remote monitoring, enterprise data acquisition, and machine-to-machine (M2M) applications.

SNAP Ethernet Systems

Based on the Internet Protocol (IP), SNAP Ethernet systems offer flexibility in their network connectivity and in the software applications they work with. The physical network may be a wired Ethernet network, a cellular wireless network, or a modem. A wide variety of software applications can exchange data with SNAP Ethernet systems, including:

- Opto 22's own ioProject™ suite of control and HMI software
- Manufacturing resource planning (MRP), enterprise management, and other enterprise systems
- Human-machine interfaces (HMIs)
- Databases
- Email systems
- OPC client software
- Custom applications
- Modbus/TCP software and hardware.

SNAP Ethernet system hardware consists of controllers and I/O units. Controllers provide central control and data distribution. I/O units provide local connection to sensors and equipment.

SNAP OEM Systems

Opto 22 SNAP OEM I/O systems are highly configurable, programmable processors intended for OEMs, IT professionals, and others who need to use custom software with Opto 22 SNAP I/O modules.

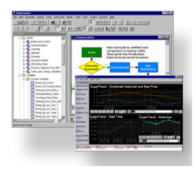
Linux® applications running on these systems can read and write to analog, simple digital, and serial I/O points on SNAP I/O modules using easily implemented file-based operations. Applications can be developed using several common development tools and environments, including C or C++, Java, and shell scripts.

M2M Systems

Machine-to-machine (M2M) systems connect your business computer systems to the machines, devices, and environments you want to monitor, control, or collect data from. M2M systems often use wireless cellular communications to link remote facilities to central systems over the Internet, or to provide monitoring and control capability via a cellular phone.

Opto 22's Nvio™ systems include everything you need for M2M—interface and communications hardware, data service plan, and Web portal—in one easy-to-use package. Visit nvio.opto22.com for more information.

Opto 22 Software



Opto 22's ioProject and FactoryFloor® software suites provide full-featured and cost-effective control, HMI, and OPC software to power your Opto 22 hardware. These software applications help you develop control automation solutions, build easy-to-use operator interfaces, and expand your manufacturing systems' connectivity.

Quality

In delivering hardware and software solutions for worldwide device management and control, Opto 22 retains the highest commitment to quality. We do no statistical testing; each product is made in the U.S.A. and is tested twice before leaving our 160,000 square-foot manufacturing facility in Temecula, California. That's why we can guarantee solid-state relays and optically-isolated I/O modules *for life*.

Product Support

Opto 22's Product Support Group offers comprehensive technical support for Opto 22 products. The staff of support engineers represents years of training and experience, and can assist with a variety of project implementation questions. Product support is available in English and Spanish from Monday through Friday, 7 a.m. to 5 p.m. PST.

Opto 22 Web Sites

- www.opto22.com
- nvio.opto22.com
- www.internetio.com (live Internet I/O demo)

Other Resources

- OptoInfo CDs
- Custom integration and development
- Hands-on customer training classes.

Opto 22 manufactures and develops hardware and software products for industrial automation, remote monitoring, enterprise data acquisition, and machine-to-machine (M2M) applications. Using standard, commercially available Internet, networking, and computer technologies, Opto 22's input/output and control systems allow customers to monitor, control, and acquire data from all of the mechanical, electrical, and electronic assets that are key to their business operations. Opto 22's products and services support automation end users, OEMs, and information technology and operations personnel.

Founded in 1974 and with over 85 million Opto 22-connected devices deployed worldwide, the company has an established reputation for quality and reliability.

Opto 22
43044 Business Park Drive
Temecula, CA 92590
Phone: (951) 695-3000
(800) 321-OPTO
Fax: (951) 695-3095
www.opto22.com

Inside Sales: (800) 452-OPTO
Product Support:
(800) TEK-OPTO
(951) 695-3080
Fax: (951) 695-3017
E-mail: sales@opto22.com