INTERNET

DATA
VISUALIZATION
INPUT/OUTPUT
LOGIC

THINGS

SNAP PAC REST API and the Internet of Things

For OT Professionals

OPTO 22
www.opto22.com

Opto 22

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: **800-321-6786** or **951-695-3000**
Pre-sales Engineering is free.
Product Support is free.
www.opto22.com

SNAP PAC REST API and the Internet of Things

For OT Professionals

Introduction

We've all heard about the Internet of Things (IoT) and its promises: bringing useful data directly to people who make business decisions, and enabling machines to communicate with each other and make decisions for human benefit.

But how does the IoT actually work? What's the pathway, from the ground up, that all these machines use to connect to each other using cloud technologies?

Millions of sensors, machines, devices, and actuators exist now in the physical world. You monitor them, use them to control processes, and get data from them. But very few of these existing sensors and devices have any built-in capability to communicate natively with computer systems.

Many of them connect to programmable logic controllers (PLCs), programmable automation controllers (PACs), or distributed control systems (DCSs). But those systems are purpose-built, and they were not designed to communicate on the IoT.

To get data from these systems into company computer systems or the IoT currently requires a complex chain of conversion hardware and software: PLCs, proprietary drivers, protocol gateways. It's a complex path that requires time, money, and domain expertise at every level to install and maintain. Even if installed, its sheer complexity makes it difficult to establish security and maintain data integrity.

Valuable data in control systems

Why is getting data from these systems so important? This data is primarily used to monitor and control processes and machines in your automation application. But some of it is also highly useful outside the automation system. For example:

Managers may need to:

- See how many widgets were produced in the last hour
- Compare that figure to yesterday at the same time
- Track yield to identify production quality issues
- Know who just accessed the security door

Facilities managers may need to:

- Log temperatures in all refrigeration units at multiple locations
- Turn the pump (or chiller, or light, or production line) on or off
- Change the setpoint on an HVAC unit
- Revise a maintenance schedule based on machine use over time or current electrical usage

Accountants may need data sent to their business database, such as:

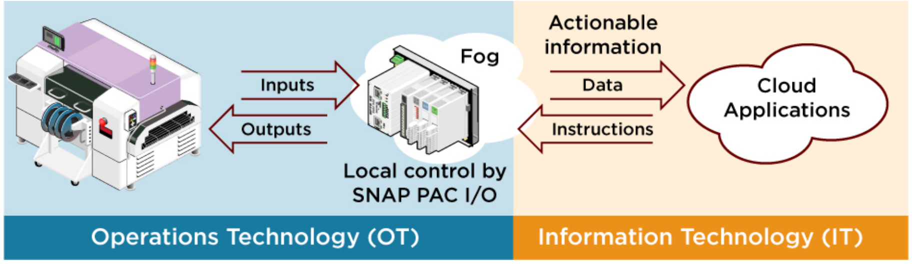
- Inventory levels
- How many products were shipped or received
- The cost or time involved in running product A versus product B through the production line
- How much electricity was used by each tenant in buildings the company owns

Data in controllers would be useful for these and many other reasons, but it's difficult to get this data into company computer systems.

Difficulties of accessing controller data

PACs and PLCs often monitor and control processes and equipment that cannot be interrupted, because an interruption would cause product or business losses, damage machinery, or endanger human safety.

Because their functions are so crucial, traditional industrial control networks are protected from unauthorized access and network outage by being physically isolated from other company networks, even when control networks use Ethernet. Automation engineers and technicians—also called Operational


Inputs
Outputs
Fog
Local control by SNAP PAC I/O
Actionable information
Data
Instructions
Cloud Applications
Operations Technology (OT)
Information Technology (IT)

Technology (OT) personnel—are understandably reluctant to open up control networks to company Information Technology (IT) networks.

In addition, industrial control networks typically use proprietary or industrial-only protocols, such as PROFINET, EtherNet/IP, or Modbus, rather than the standard TCP/IP used in the IT world.

But to realize the promise of the Internet of Things, the two groups—OT and IT—must work together to achieve two goals: first, keep critical control networks safe, and second, provide useful data where and when it is needed.

What can cut through this complexity—and as a result can help you realize your IoT goals now—is an Opto 22 SNAP PAC controller with unique features.

SNAP PACs and I/O

Like all industrial controllers, Opto 22 SNAP PACs contain and control a lot of data—data from the I/O points connected to the PAC, and data in variables within the controller's logic.

Opto 22 SNAP PACs are industrially hardened, small-footprint programmable automation controllers (PACs) used in applications worldwide. SNAP PACs control analog and digital SNAP I/O modules, which come in a wide range of signal inputs and contain from 1 to 32 I/O points.

In both automation and the Internet of Things, I/O serves a key function: translating between the physical world and the digital world. As automation engineers know, sensors and machines typically understand electrical signals like voltage and current; PLCs, PACs, DCSs, and computers do not. This translation can go both ways:

- For monitoring, I/O takes the electrical signals from physical things and translates them into the ones and zeros understood by the digital world.
- For control, I/O translates digital ones and zeros into electrical signals and sends them to act on physical things.

That works for automation, but not for the IoT. The stumbling block is getting the data that last step—either into or out of the networks and protocols used by the IT world. As we've seen, it takes a variety of hardware and software—protocol converters, middleware, drivers—to move the data from PLCs, PACs, and DCSs to IT systems. But without this last step, the Internet of Things is impossible.

Opto 22's S-series and R-series SNAP PAC controllers make that last step simple, bridging the gap between OT and IT systems.

Accessing SNAP PAC data

Here's how SNAP PACs bridge the gap. First, they run on **standard Ethernet networks** and communicate over the **standard Internet Protocol (IP)**, so networking and protocols are already compatible with IT technologies.

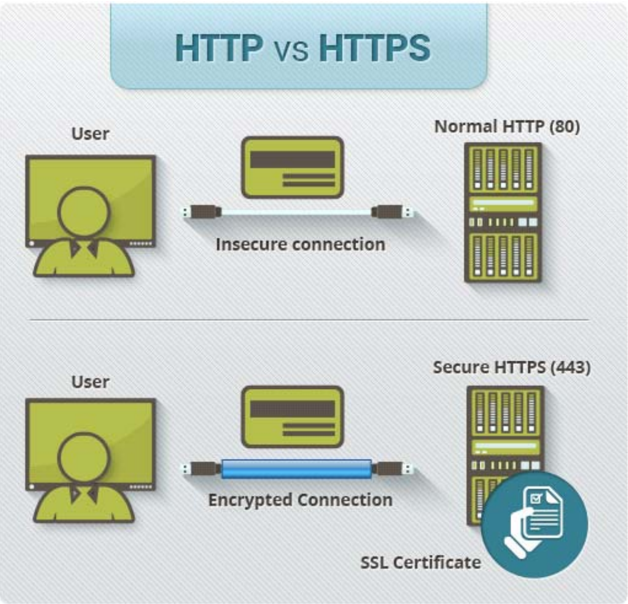
And second, as of firmware version R9.5a, SNAP PACs also include two critical building blocks of the IoT:

- a built-in **HTTP/HTTPS server** for communication
- a **RESTful API** (application program interface based on the REST architecture, which is a standard developer's format for writing programs)

Let's take a look at what these terms mean and what they do for you and the people in your company who need data currently locked in control systems and equipment.

HTTP and HTTPS

HTTP (Hypertext Transfer Protocol) is the application protocol that underlies data communication over the Internet. However, HTTP by itself does not provide secure



communication; all data going across the connection is wide open to anyone on the network.

HTTP used with TLS/SSL (Transport Layer Security and the older Secure Sockets Layer) becomes *HTTPS*, which is secure communication. What makes it secure is that HTTPS provides *encryption:* all data going over that connection is encrypted.

Encryption vs. authentication

It's easy to confuse encryption and authentication, but they are different, and both are important to understand and use for security, whether on your local network or over the Internet.

Encryption means turning data into nonsense text that means nothing unless you have the correct key—established in advance—to decrypt it. Encrypted data is protected; if it is intercepted, it is worthless without the key. There are various ways to encrypt and decrypt data, but fundamentally encryption is a way to make data understandable only by the people and systems that should have access to it.

Authentication means proof that you are who you say you are. Transactions over HTTPS are often blind, and it's important to make sure that the person or system on the other end is the right one, not a fake. Often just a username and a password are enough to establish authenticity, but some systems require more proof, like a token or a fingerprint.

Often authentication is added to encryption (and performed through an encrypted channel), so you have a secure connection and you also have to prove your identity. For example, when you buy a book on Amazon, you see the HTTPS and its padlock symbol in the URL bar. But you also have to put in your username and password, so Amazon knows it's really you and it's all right to process the sale.

Clients and servers

HTTP and HTTPS use a stateless, client-server method of communication that consists of individual request/response pairs on a network. The client requests data; the server responds.

Client	Server
Can only talk Cannot listen or respond Asks for data Sends data	Can only listen Cannot talk Responds to requests Acknowledges data received
> Like Modbus master	> Like Modbus slave

- A *client* can only talk; it can never listen or respond. The client asks for data or sends data at intervals it determines.
- A *server* can never initiate a conversation. A server is constantly listening but cannot talk; it can only listen and respond. The server responds to requests for data or acknowledges data sent to it.

The client must be able to reach the server on a TCP/IP network (more about this later).

You'll notice that this client-server method is similar to the Modbus master-slave communication method:

- The *client* is the Modbus *master:* it makes requests and sends data.
- The *server* is the Modbus *slave:* it listens, sends the requested data, and acknowledges data sent to it.

Notice that the SNAP PAC HTTPS **server** serves the data the PAC contains. The HTTPS server cannot "send" the data anywhere.[1] A client must make a request to the PAC, and then the PAC will respond. (Remember, a server can't talk; it can only listen.)

The client can use a standard HTTP GET to request data from the PAC and a standard HTTP POST to send data to the PAC. The format of the request is determined by the SNAP PAC REST API.

SNAP PAC REST API

An API is like a user interface for computer programs. It's a way for one computer program to access data or resources from another program. The SNAP PAC REST API uses the HTTP/S protocol. There are other APIs that use a different

[1] There are other ways the PAC can send data—as a client rather than a server, of course—but they're outside the scope of this document.

protocol, for example MQTT, but HTTP/S is the most common today.

The complete SNAP PAC REST API is documented on developer.opto22.com. It includes all possible calls you can make to read data from or write data to the PAC, using a REST-compliant programming language. There are many of these, all familiar to web or IT programmers: PHP, Python, .NET, JavaScript, and more.

The API shows you how to access I/O point data as well as numeric and string data from variables in the PAC's logic.

Data is returned as JSON (JavaScript object notation), a standard data interchange format that humans can easily read and write and computers can easily parse and generate.

What do you need to access SNAP PAC data?

You don't need protocol converters and software drivers. You don't even need OPC. All you need is a SNAP PAC with firmware R9.5a or higher and I/O.

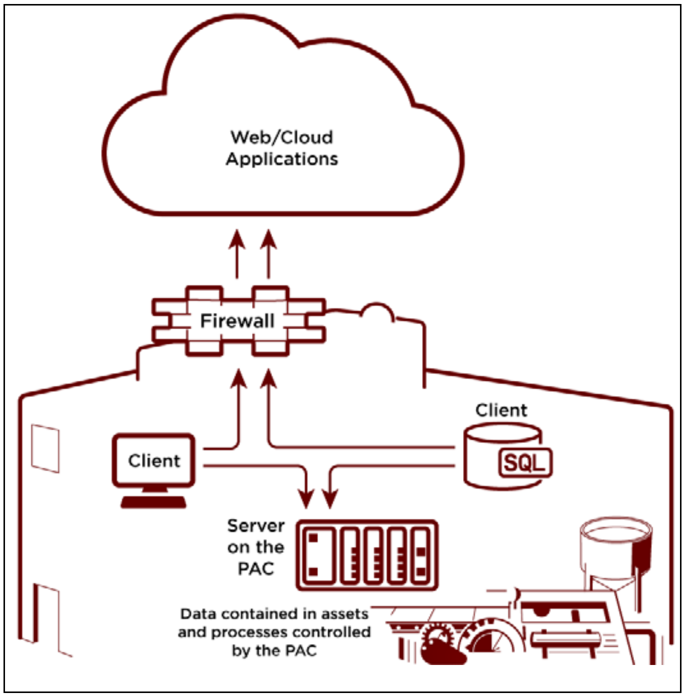
You can securely access data from a SNAP PAC's server over HTTPS (with encryption and authentication) using standard IT networking, protocols, and programming tools. To get started:

- Have someone who knows a programming language that works with a RESTful API. Or use Node-RED (see page 5).
- Become familiar with JSON formatting
- Make sure the client can access the PAC over a TCP/IP network on port 443
- Decide whether access to data should be read-only or read-write

Due to the often critical nature of data on a SNAP PAC, read-write access should be carefully controlled. Consider all possible consequences before granting read-write access to any client—person or system.

Networking considerations

We never recommend putting a SNAP PAC controller directly on the Internet. The PAC should be located behind a firewall, and in most cases the control network should be segmented from the company computer network to facilitate security. You can use the two built-in independent Ethernet network interfaces on the SNAP PAC to segment networks (see the controller user's guide for setup steps).



Because the SNAP PAC acts as an HTTPS server for data, the PAC must be accessible by the client over a TCP/IP network.

A client might be:

- A computer that displays the PAC's data
- A database or other system
- A smartphone on your wireless network that incorporates data into an app
- Other applications

What if you want to access the PAC's data from somewhere else, like on your phone outside the building or from a remote location? All you need to do is set up a client that accesses data from the PAC and then sends it forward, beyond the firewall.

Node-RED

If you're not ready to write your own program for the client using one of the languages compatible with the REST API, you can access data on the PAC by using Node-RED.

Node-RED was created by IBM and is a free, web-based, visual tool that can be used to "wire" together hardware devices, APIs, and online services.

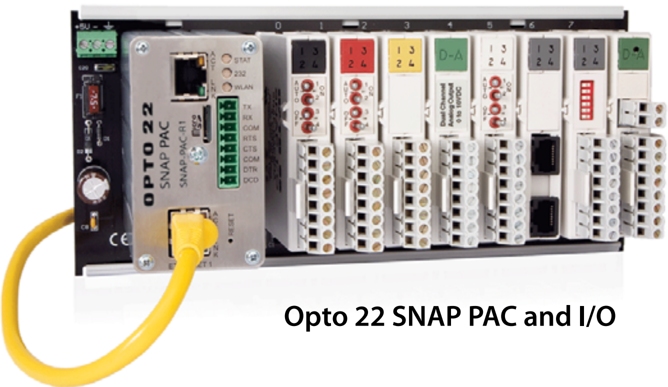
For example, you might use Node-RED to provide a facilities manager with a map showing local temperatures, chiller on/off times, and electrical usage for building sites. This "mashup" could use data from your PAC, a weather website such as Weather Underground, and Google maps.

Two nodes are available for SNAP PACs, one for reading and one for writing. Complete information and instructions are on developer.opto22.com. (Node-RED is based on Node.js, but you don't need to know Node.js in order to use it. If you know JavaScript, you can optionally add functions in the Node-RED editor, but again, you don't need to.)

Why Opto 22?

Three reasons: experience, reliable products, and personal attention.

Opto 22 started in 1974, when our engineer founder, Bob Engman, designed a better solid-state relay (SSR). With more than 40 years of automation experience, today we manufacture controllers, SSRs, and I/O known world-wide for reliability, plus easy-to-use software for developing control programs, visualizing data on any device, and integrating automation systems with computer networks.



Opto 22 SNAP PAC and I/O

All Opto 22 products are built on open standards. All products are designed, manufactured, and supported at our company headquarters in Temecula, California, U.S.A.

Most SSRs and I/O modules carry a lifetime warranty. Product Support is free. Pre-sales Engineering is also free, so you can call us with any questions about your application.

Getting Started

See developer.opto22.com for the REST API and Node-RED nodes for SNAP PAC controllers.

If you already have an Opto 22 SNAP PAC S-series or R-series controller and I/O, make sure you have firmware version R9.5a or higher in the PAC.

If you don't have a PAC and I/O, see them on our website, www.opto22.com:

- SNAP PAC S-series standalone controllers
- SNAP PAC R-series rack-mounted controllers
- SNAP I/O

All Opto 22 products are available through distributors worldwide:

- Regional distributors in the U.S. and Canada
- National distributors (U.S. and Canada)
- International distributors

Questions?

Contact your local distributor or Opto 22 Pre-sales Engineering with any questions about products, system architecture, or the IoT.

Phone: **1-800-321-6786** (toll-free in the U.S. and Canada) or **1+951-695-3000**

Or send a question to an engineer.

Form 2187-160719 • Opto 22 • 43044 Business Park Drive • Temecula, CA 92590-3614 • www.opto22.com
SALES 800-321-6786 • 951-695-3000 • FAX 951-695-3095 • sales@opto22.com • SUPPORT 800-835-6786 • 951-695-3080 • FAX 951-695-3017 • support@opto22.com
© 2016 Opto 22. All rights reserved. Dimensions and specifications are subject to change. Brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.
PAGE 5