Communication Tools & Protocols for Opto 22 Products

Introduction

Because Opto 22 products—both current and legacy—are based on open standards, you can use them with a variety of programming tools and communication protocols. In addition to monitoring and controlling sensors, machines, and equipment, you can exchange data with devices, databases, systems, software applications, and web services.

Thanks to open standards, Opto 22 products can make data available for integration, storage, analysis, and Industrial Internet of Things (IIoT) applications.

This technical note describes the programming tools, protocols, and options you can use and shows you which Opto 22 products work with each. Some require free Opto 22 software development kits (SDKs) and integration kits, which are available for SNAP PAC and legacy products.

- APIs, SDKs, and integration kits are briefly described starting on page 6.
- Protocols and other communication options are described starting on page 11.
- See developer.opto22.com for complete information on RESTful APIs, Node-RED nodes, and code samples.
- You can also see and contribute additional methods of communication developed by our user community in the OptoForum at www.opto22.com.

Most of the tools and protocols in this tech note apply to these Opto 22 products—follow the links to see descriptions and specifications:

- PAC Project™ automation software suite
- SNAP PAC controllers (SNAP PAC S-series, R-series, and SoftPAC)
- SNAP PAC brains (I/O processors)
- *groov* mobile operator interface tool

If you have questions on tools, protocols, or the use of Opto 22 products, contact **Pre-Sales Engineering** by phone or email. Pre-Sales Engineering is free, and we're glad to help.

Phone: 800-321-6786 (toll-free in the U.S.) or 951-695-3000
Email: systemseng@opto22.com

After you purchase Opto 22 products, you'll also receive free product support from the experienced engineers in our Temecula, California headquarters.

Overview: Tools, Protocols, and Products

NOTE: Shaded cells in the table indicate legacy products, which are not recommended for new development. For information on most legacy equipment, see Legacy and Current SNAP Product Comparison Charts, form 1693.

Development Environment and/or Protocol	Purpose	Opto 22 Products	Notes
Development Environments			
C, C++, C#, Java, JavaScript, Perl, Python, or any other JSON-compatible programming language	Access I/O point and variable data in a SNAP PAC controller, using the PAC's REST API	PAC Project and SNAP-PAC R-series or S-series controller	See developer.Opto22.com for complete API documentation and steps for getting started. Requires PAC firmware R9.5a or higher.
	Access data in a *groov*® Data Store using the *groov* API	*groov*	See *groov* Build and View User's Guide (form 2027). See API documentation within groov Build (choose Help > *groov* Public API). Requires *groov* R3.3a or higher.
C, C++, Java, JavaScript, Perl, Python, or any other Raspberry Pi-compatible programming language	Use a Raspberry Pi to monitor and control industrial digital sensors & devices	Digital I/O Carrier Board for Raspberry Pi G4PB8H or G4PB16H rack and G4 digital I/O modules SNAP-D4M rack and SNAP digital I/O modules	See developer.Opto22.com for getting started steps and code samples.
C++	Read or write to memory map addresses in the brain or controller	SNAP PAC controllers SNAP PAC brains G4EB2 brain E1 and E2 brain boards	PAC-DEV-OPTOMMP-CPLUS
		SNAP Ethernet I/O SNAP Simple I/O SNAP Ultimate I/O SNAP-LCE controller	
.NET	Read or write to I/O point and variable data in the PAC's control program (PAC Control™ strategy)	PAC Project and SNAP PAC controller	PAC-DEV-CONTROLLER-DOTNET
	Read or write to memory map addresses in the brain or controller	SNAP PAC controllers SNAP PAC brains G4EB2 brain E1 and E2 brain boards	PAC-DEV-OPTOMMP-DOTNET
	Communicate with current loop devices using the HART® protocol	HART SNAP I/O module(s)	PAC-DEV-HART-DOTNET Requires HART SNAP I/O module(s): SNAP-AIMA-iH, SNAP-AOA-23iH
Node-RED	Wire together hardware devices, databases, web services, APIs	SNAP-PAC R-series SNAP-PAC-S series *groov*	Node-RED nodes for SNAP PACs (requires PAC firmware R9.5a or higher). Node-RED node for *groov*. GROOV-AR1 *groov* Box includes Node-RED (requires *groov* Admin 1.570.44 or higher). See developer.opto22.com.

Development Environment and/or Protocol	Purpose	Opto 22 Products	Notes
Protocols and Data Exchange Options			
BACnet/IP	Communicate with devices on a BACnet/IP network	PAC Project and SNAP PAC controller	PAC-INT-BAC-IP Requires PAC Project R9.4a or higher and PAC firmware R9.4a or higher.
BACnet MS/TP	Communicate with systems using the BACnet protocol over RS-485 (serial)	PAC Project and SNAP PAC S-series controller	PAC-INT-BAC Requires PAC Project R8.2a or higher.
CAN	Receive data from systems using the CAN protocol	PAC Project and SNAP PAC controller and SNAP-SCM-CAN2B	PAC-INT-CAN-RX (Receive only) Requires SNAP-SCM-CAN2B CAN communication module with firmware R1.0d or lower and PAC Control R9.2a or higher.
	Transmit to and receive data from devices using the CAN protocol	PAC Project and SNAP PAC controller and SNAP-SCM-CAN2B	PAC-INT-CAN-RXTX (Transmit & receive) Requires SNAP-SCM-CAN2B CAN communication module with firmware R2.0d or higher and PAC Control R9.2a or higher.
DF1	Communicate with older Allen-Bradley® drivers or PLCs using the DF1 protocol	PAC Project and SNAP PAC controller	PAC-INT-ABDF1
		OptoControl and SNAP-LCM4 or M4 controller	OPTOINTAB
DNP3	Communicate with systems using the DNP3 protocol	PAC Project and SNAP PAC controller	PAC-INT-DNP3 (Requires PAC Control Professional R8.1 or higher).
EtherNet/IP™	Allow Allen-Bradley PLCs to use SNAP-PAC I/O, using the EtherNet/IP protocol	SNAP PAC controllers SNAP PAC EB brains G4EB2 brain	Direct communication over EtherNet/IP.
FTP (client)	Work with data files	SNAP PAC controllers SNAP PAC EB brains	Direct communication over Ethernet.
FTP (server)	Work with data files	SNAP PAC controllers SNAP PAC EB brains G4EB2 brain	Direct communication over Ethernet.
		SNAP Ultimate I/O SNAP Ethernet I/O	Direct communication over Ethernet.
HART	Communicate with current loop devices using the HART protocol	PAC Project and SNAP PAC controller and HART SNAP I/O module(s)	Requires HART SNAP I/O module(s).
		HART SNAP I/O module(s) and SNAP PAC R-series controller or SNAP PAC brain	Custom program built using PAC-DEV-HART-DOTNET. Requires HART SNAP I/O module(s).
HTTP HTTPS	Communicate with other devices & software	SNAP PAC S-series SNAP PAC R-series *groov*	HTTP/HTTPS server is built in. Requires PAC firmware R9.5a or higher.

Development Environment and/or Protocol	Purpose	Opto 22 Products	Notes
IEC-60870	Communicate with systems using the 60870-5-101 and 60870-5-104 protocols	PAC Project and SNAP PAC controller	PAC-INT-IEC60870-5 Requires PAC Control Professional R9.3a or higher.
microSD card	Local data storage	SNAP PAC S-series SNAP PAC R-series	Built-in microSD card slot (to 32 GB)[1]
mistic	Communicate with legacy *mistic* I/O	SNAP PAC S-series and PAC Project Professional	Use legacy versions of the PAC Control, PAC Display™, and PAC Manager™ User's Guides. Also see the FactoryFloor to PAC Project Migration Technical Note.
Modbus®/TCP	Exchange data with Modbus devices over Ethernet	*groov*	Direct Modbus master.
		PAC Project and SNAP PAC controller	For memory map addresses, direct Modbus slave-only communication. For PAC Control data, use PAC-INT-MB (Includes master and slave for both Ethernet and serial networks).
		SNAP PAC EB brains G4EB2 brain	Direct Modbus slave-only communication.
		E1 and E2 brain boards	Direct Modbus slave-only communication.
		ioProject and SNAP Ultimate I/O	IOP-INT-MBTCP
Modbus RTU Modbus ASCII	Exchange data with Modbus devices over a serial network	PAC Project and SNAP PAC controller	PAC-INT-MB (Includes master and slave for both Ethernet and serial networks.)
		ioProject and SNAP Ultimate I/O	IOP-INT-MBSER
		OptoControl and SNAP-LCM4 or M4 controller[2]	OPTOINTMB
	Exchange data with Basler DGC-2020 Digital Genset Controller	PAC Project and SNAP PAC controller	PAC-INT-MBBAS Requires PAC Control R9.0a or higher.
OPC UA	Communicate as a client with systems and equipment compatible with OPC UA	*groov*	Requires Enterprise platform on either *groov* Box or *groov* Server for Microsoft® Windows®.
OPC DA 2.0	Communicate as a server with OPC DA 2.0-compatible systems and equipment	OptoOPCServer™	Included in PAC Project Professional or available for separate purchase.
OPC DA 1.0	Communicate as a server with OPC DA 1.0-compatible systems and equipment	OptoServer	Included in legacy FactoryFloor software suite.

Development Environment and/or Protocol	Purpose	Opto 22 Products	Notes
OptoMMP	Read and write memory map data in Opto 22's OptoMMP-based brains and controllers	SNAP PAC controllers SNAP PAC brains G4EB2 brain E1 and E2 brain boards	PAC-DEV-OPTOMMP-DOTNET or PAC-DEV-OPTOMMP-CPLUS
		SNAP Ultimate I/O SNAP Ethernet I/O SNAP Simple I/O SNAP-LCE controller	PAC-DEV-OPTOMMP-CPLUS
Optomux (serial) Optomux (Ethernet)	Use Optomux protocol with I/O	E1 & E2 brain boards	Use Optomux Protocol Drivers & Utilities (compatible with Win XP/2000 only).
Pamux	High-speed control of digital I/O	PCI-AC51 or PCIe-AC51 adapter card and B4, B5, or SNAP-B4 brain	PC-PAMUX-SDK (compatible with Windows 8/7/Vista/XP/2000 only).
Peer-to-peer	Communicate between controllers	SNAP PAC controllers	See PAC Control User's Guide for Scratch Pad method. See developer.Opto22.com for using PAC's RESTful API.
		SNAP Ultimate I/O SNAP-LCE controller	See PAC Control User's Guide for Scratch Pad information.
PPP	Communicate over a dial-up modem	SNAP PAC S-series SNAP PAC R-series	Direct communication.
PROFIBUS® DP	Communicate with PROFIBUS DP systems	PAC Project and SNAP PAC controller and SNAP-SCM-PROFI	PROFIBUS-DP Master Software Requires SNAP-SCM-PROFI module.
SMTP	Send email	SNAP PAC controllers PAC Display™ SNAP PAC EB brains G4EB2 brain	SMTP client with authentication and attachments.
		SNAP Ultimate I/O SNAP Ethernet I/O	SMTP client.
SNMP	Allow network management systems to communicate with non-IT devices, including controllers, brains, and their connected I/O, through an SNMP management console	SNAP PAC controllers SNAP PAC EB brains G4EB2 brain	Direct communication. • Use OPTOMIB1 for SNMP v1-compliant Management Systems. • Use OPTOMIB2 for SNMP v2-compliant Management Systems (Requires firmware version R8.2a or higher, or R9.4b or higher for high-density digital modules).
		SNAP Ultimate I/O SNAP Ethernet I/O	Direct communication. Use OPTOMIB1 (SNMP v1-compliant Management Systems). Requires firmware version R5.0 or higher on SNAP Ultimate I/O.

Development Environment and/or Protocol	Purpose	Opto 22 Products	Notes
SQL database	Exchange data between Opto 22 system and SQL databases	PAC Display or OptoDataLink™ and SNAP PAC controller	
		PAC Project and SNAP PAC controller	Custom programs using the PAC's RESTful API. Requires PAC firmware R9.5a or higher.
		groov	Custom programs using the *groov* Data Store REST API. Requires *groov* R3.3a or higher.
SSI	Acquire data from linear or rotary transducers using the serial synchronous interface (SSI) protocol	SNAP-SCM-SSI communication module SNAP PAC R-series controller or SNAP PAC brain	See the SNAP SSI Module User's Guide.
TL1	Integrate PAC Control with systems using the TL1 protocol	PAC Project and SNAP PAC controller	PAC-INT-TL1 Requires PAC Project R8.2a or higher.
TCP/IP UDP/IP	Communicate over this Internet layer	*groov* SNAP PAC controllers SNAP PAC EB brains G4EB2 brain E1 and E2 brain boards	Built-in communication.
		SNAP-LCM4 and M4 controllers[2] SNAP Ultimate I/O SNAP Ethernet I/O	Built-in communication.

[1]PACs manufactured after November 2008 have a card slot; firmware R9.4a and loader R6.1a or higher required for microSD cards above 2 GB.
[2]Requires an M4SENET-100 Ethernet card

Opto 22 APIs, SDKs, and integration kits

RESTful APIs

For software developers and Internet of Things (IoT) applications, a RESTful API and HTTP/HTTPS server allow secure access to data in SNAP PAC hardware controllers and *groov* mobile operator interfaces. These RESTful APIs are compatible with any programming language that supports JavaScript Object Notation (JSON).

SNAP PAC REST API—The SNAP PAC REST API gives you access to PAC Control variable and I/O data in any SNAP PAC S-series or R-series programmable automation controller with firmware R9.5a or higher. Each controller includes a

RESTful API and server (for secure access, always use HTTPS). Data is returned in JSON. For more information, see the SNAP PAC REST API Resources page.

Complete API documentation and steps to get started are on developer.opto22.com.

***groov* Data Store REST API—**You can create a Data Store in *groov* to hold data from databases, online services, and other software. This data can then be used in your *groov* operator interface. For example, data from a database can be placed in the interface for your users to see. Or an operator can change data in the interface, which is then read and changed in the database.

API documentation is available from within *groov* Build (choose Help > *groov* Public API).

Software development kits (SDKs)

C++ OptoMMP Software Development Kit for SNAP PAC (Part number PAC-DEV-OPTOMMP-CPLUS)—This SDK is for C++ developers and works with Windows (includes ActiveX controls), Linux®, and OS X. This SDK provides access to the OptoMMP memory map areas of SNAP PAC brains and controllers. It does not provide access to objects in a PAC Control strategy running on the controller.

.NET OptoMMP Software Development Kit for SNAP PAC (Part number PAC-DEV-OPTOMMP-DOTNET)—This 100% managed SDK is for .NET developers and supports Microsoft's .NET frameworks 3.5 through 4.5, Visual Studio 2005-2013, and Mono 3.0 for Linux. This SDK provides access to the OptoMMP memory map areas of SNAP PAC brains and controllers. It does not provide access to objects in a PAC Control strategy running on the controller.

.NET Control SDK for SNAP PAC (Part number PAC-DEV-CONTROLLER-DOTNET)—This SDK helps you build .NET applications to access objects in a PAC Control strategy running on a SNAP PAC controller. These objects include integer, float, and string variables and tables as well as analog and discrete input and output points.

This SDK does not provide access to the memory map areas of the controller; it provides access only to strategy tags.

.NET HART Software Development Kit for SNAP PAC (Part number PAC-DEV-HART-DOTNET)—This SNAP PAC .NET HART SDK helps you develop or add to a .NET or Mono custom program communicating with Opto 22 HART SNAP I/O modules (SNAP-AIMA-iH current input module and SNAP-AOA-23-iH current output module). These modules talk with other Highway Addressable Remote Transducer (HART) current loop devices. The SDK supports Visual Studio 2005, 2008, and 2010, as well as Mono 3.2.7 or newer.

Note that if you are controlling HART SNAP I/O modules through a SNAP PAC controller running a PAC Control strategy, you do not need this SDK.

Optomux Protocol drivers and utilities— Optomux Protocol drivers and utilities is a download that installs Optomux drivers for E1, E2, B1, B2, and serial B3000 and B3000-B brains. This download is recommended for all new Optomux-based applications, because it supports Optomux communications over both Ethernet and serial networks. It is compatible with Windows 2000/XP but does not support .NET.

PAMUX Systems SDK for PCI Adapter Cards (Part number PC-PAMUX-SDK)—This SDK is for high-speed control of multiple digital and/or analog I/O points, using a Pamux brain (I/O processor) and the Pamux protocol. The SDK requires a PCI-AC51 or a PCIe-AC51 adapter card.

Integration kits for PAC Project

The PAC Project industrial automation software suite includes PAC Control for programming a SNAP PAC controller and PAC Display for HMI development and runtime. The integration kits in this section help you use I/O and variable data from the PAC Control *strategy* (control program) with other systems and equipment.

Allen-Bradley DF1 Integration Kit for PAC Control (Part number PAC-INT-ABDF1)—The Allen-Bradley DF1 Integration Kit provides users of Opto 22's PAC Control software an easy method of communicating with Allen-Bradley devices or PLCs that support the DF1 protocol.

The Allen-Bradley DF1 Integration Kit consists of a set of PAC Control subroutines, which can be called from any flowchart in the PAC Control strategy.

BACnet IP Integration Kit for PAC Control (Part number PAC-INT-BAC-IP)—The BACnet IP Integration Kit enables your SNAP PAC controller to communicate with devices on a BACnet/IP network. BACnet is a communications protocol for building automation and control networks. The integration kit contains BACnet_IP_Protocol, BACnet_Master, and VAV_Poll_Example charts. The kit is a full BACnet/IP master with slave functions.

BACnet MS/TP Integration Kit for SNAP-PAC-S (Part number PAC-INT-BAC)—The BACnet MS/TP Integration Kit for PAC Control enables your SNAP PAC S-series controller to communicate with a BACnet MS/TP network via RS-485. BACnet is a communications protocol for building automation and control networks. MS/TP is a Master-Slave / Token-Passing specification of BACnet. The integration kit contains the BACnet_Protocol and Master_Sub charts, which contain everything you need to use the BACnet MS/TP protocol in your own PAC Control strategy.

Basler DGC-2020 Digital Genset Controller Integration Kit for PAC Control (Part number PAC-INT-MBBAS)—The Basler Integration Kit provides an interface to use a SNAP PAC controller to read and write Modbus parameters to a DGC-2020 Digital Genset Controller by Basler Electric. The PAC Control and PAC Display files take advantage of an optional feature of the DGC-2020 that performs Modbus

communications by emulating a subset of the Modicon 984 Programmable Controller.

CAN Integration Kits for PAC Project)—The CAN Integration Kits provide sample PAC Control subroutines and charts so that your Opto 22 SNAP PAC System equipped with one or more SNAP-SCM-CAN2B modules can communicate with devices on a Controller Area Network (CAN). The SNAP-SCM-CAN2B is a high-speed serial communications module that can transmit to and receive data from CAN devices. Two integration kits are available:

- PAC-INT-CAN-RXTX (Transmit & receive)—Requires SNAP-SCM-CAN2B CAN module, module firmware R2.0d or higher, and PAC Control R9.2a or higher.
- PAC-INT-CAN-RX (Receive only)—Requires SNAP-SCM-CAN2B CAN module, module firmware R1.0d or lower, and PAC Control R9.2a or higher.

DNP3 Integration Kit for PAC Control (Part number PAC-INT-DNP3)—The DNP3 Integration Kit for PAC Control allows Opto 22 SNAP PAC controllers, using PAC Control, to connect via an Ethernet network or serial port and communicate using DNP3, the Distributed Network Protocol. The integration kit contains a set of PAC Control subroutines that are added to a strategy to add DNP events, an example DNP3 strategy containing the DNP3_protocol, and DNP3_CROB charts that are imported into a strategy to enable an Opto 22 controller to communicate as a DNP3 Level 2 Master or Outstation.

IEC60870-5 Integration Kit (Part number PAC-INT-IEC60870-5)—The IEC60870-5 Integration Kit for PAC Project allows Opto 22 SNAP PAC controllers using PAC Control to connect via an Ethernet network and serial port or serial module and communicate using the 60870-5-101 and 60870-5-104 protocol.

It supports 60870-5-101 outstation over serial and Ethernet and 60870-5-104 outstation over Ethernet. The 60870-5 outstation can connect to 1–4 masters using Ethernet and Serial ports.

Modbus Integration Kit for PAC Control (Part number PAC-INT-MB)—The Modbus Integration Kit allows Opto 22 SNAP PAC controllers using PAC Control to communicate using the Modbus Serial RTU, Modbus Serial ASCII, or Modbus/TCP protocol. The Integration Kit contains a set of PAC Control master subroutines, an example Modbus Master chart, a PAC Control slave subroutine, an example Modbus Slave that uses the slave subroutine, a demo data chart, and example initialization files.

PROFIBUS-DP Master Software—PROFIBUS-DP Master Software, though not officially an integration kit, is an example of how to use Opto 22's PAC Control (or legacy ioControl) software to monitor and control a PROFIBUS DP network through a SNAP-SCM-PROFI module. The example includes a configuration utility and a sample ioControl strategy.

SNMP MIB Files—Also not officially integration kits, these are management information base (MIB) files that describe the Opto 22 network objects, such as SNAP Ethernet-based brains or controllers, that can be managed using the Simple Network Management Protocol (SNMP):

- OPTOMIB1 is for systems compatible with SNMP version 1.
- OPTOMIB2 is for systems compatible with SNMP version 2.

TL1 Integration Kit for PAC Control (Part number PAC-INT-TL1)—The TL1 Integration Kit for PAC Control allows Opto 22 SNAP PAC controllers using PAC Control to manage telecommunication network elements (NEs) using the Transaction Language 1 (TL1) protocol, an element management protocol widely used in North America. The integration kit contains an example strategy, which contains everything you need to use the TL1 protocol in your own PAC Control strategy.

Legacy integration kits (ioControl, OptoControl)

These legacy integration kits work with the older control software ioControl and OptoControl, which were used with legacy controllers.

(Legacy) Allen-Bradley DF1 Integration Kit for OptoControl (Part number OPTOINTAB)—The Allen-Bradley DF1 Integration Kit for OptoControl allows communication between OptoControl controllers and Allen-Bradley DF1 protocol devices. The integration kit includes subroutines, example strategies, and documentation.

(Legacy) Modbus Serial Integration Kit for ioControl (Part number IOP-INT-MBSER)—The Modbus Serial Integration Kit for ioControl allows Opto 22 Ultimate I/O controllers, using ioControl, to connect via a serial network to any Modbus device and communicate using the Modbus RTU or ASCII protocols. The Integration Kit contains a set of ioControl master subroutines, an example Modbus slave strategy, and ioDisplay master and slave projects to show communication between the example strategies and a Modbus device.

(Legacy) Modbus/TCP Integration Kit for ioControl (Part number IOP-INT-MBTCP)—The Modbus/TCP Integration Kit for ioControl allows Opto 22 Ultimate controllers using ioControl to connect via Ethernet to Modbus/TCP devices. This Integration Kit contains a set of ioControl master subroutines, an example Modbus/TCP slave strategy, and ioDisplay master and slave projects to show communication between the example strategies and a Modbus/TCP device.

(Legacy) Modbus Serial Integration Kit for OptoControl (Part number OPTOINTMB)—The OPTOINTMB integration kit allows communication between OptoControl controllers and third-party serial Modbus RTU or ASCII protocol

devices. The integration kit includes subroutines, example strategies, and documentation.

Protocol and communication option definitions

Protocol types

Industrial Ethernet protocols run over an Ethernet network or a wireless LAN (local area network) and are used in industrial automation applications.

Internet protocols are open protocols originally designed for use on the Internet and in information technology (IT). All run over a wired Ethernet network or wireless LAN. Detailed information about Internet Protocols is available on (curiously enough) the Internet.

Serial protocols run over a standard serial physical network, either RS-485/422 or RS-232.

Communication options

In addition to protocols, an assortment of options exist for exchanging data among automation systems, equipment, devices, and databases.

Definitions (in alphabetical order)

- **BACnet**—BACnet is a communications protocol for building automation and control networks. BACnet MS/TP runs on a serial network. BACnet IP runs on Ethernet.
- **CAN**—Controller Area Network (serial protocol), originally developed for use in the automotive industry but now used more widely in automation.
- **DF1**—An Allen-Bradley serial protocol used to communicate with older A-B devices and PLCs over an RS-232 serial link.
- **DNP3**—Distributed Network Protocol (serial protocol), primarily used in the electrical and water utility industries. DNP3 is often used for communication between a SCADA master station and RTUs or intelligent electronic devices (IEDs).
- **EtherNet/IP**—An open industrial Ethernet protocol originally developed by Rockwell Automation and now managed by ODVA (Open Device Vendors' Association). To set up communication, see Opto 22 form 1770, *EtherNet/IP for SNAP PAC Protocol Guide*.

- **FTP**—File Transfer Protocol (Internet protocol). An application layer protocol used to manage files on the Internet, for example to move files from one device to another or add data to them.
- **HART**—Used to communicate with Highway Addressable Remote Transducer (HART) devices over analog wires.
- **HTTP**—Hypertext Transfer Protocol (Internet protocol). Another application layer protocol, HTTP is fundamental to the World Wide Web and used for communication between servers and clients.
- **HTTPS**—Hypertext Transfer Protocol Secure (or HTTP over TLS, an Internet protocol). HTTPS incorporates authentication of a website on the Internet and additional privacy and integrity for exchanged data, to make online communications more secure.
- **IEC60870-5**—Used for telecontrol and telecommunications in electric power systems.
- **Mass storage—microSD card**—A microSD card is a tiny flash memory card used to store data. The card can be physically moved between devices that have microSD card slots in order to exchange data between devices. SNAP PAC controllers manufactured after November 2008 contain a microSD card slot. For more information, see the PAC's user's guide.
- ***mistic***—An open serial protocol developed by Opto 22. Runs on RS-485/422.
- **Modbus/TCP**—Open industrial Ethernet protocol developed by Modicon, Inc., Industrial Automation Systems. See Opto 22 form 1678, the *Modbus/TCP Protocol Guide*, for details on setting up communication.
- **Modbus RTU**—Modbus is an open protocol originally developed by Modicon for its serial-based PLCs. Modbus RTU is the binary version of the serial protocol and often used for remote telemetry units (RTUs) in a supervisory control and data acquisition (SCADA) application.
- **Modbus ASCII**—Also the Modbus serial protocol, but this version presents numerical data in ASCII for easier reading. One major difference between the binary and ASCII versions of Modbus is the checksum calculation method.
- **OPC UA (Unified Architecture)—**Successor to OPC DA; also developed and managed by the OPC Foundation. OPC UA is a cross-platform standard for interoperability among industrial automation devices. The platform independence, multi-layered approach, and better security of OPC UA are utilized by Opto 22's *groov* tool for building and using mobile operator interfaces for virtually any automation system or equipment. (Learn more about *groov*.)
- **OPC DA (Data Access)**—Predecessor to OPC UA (see above). OLE (Object Linking and Embedding) for Process Control. Not a protocol, but a series of

standards for exchanging data among devices in industrial automation. These standards can be used over an Ethernet or serial physical layer. Managed by the OPC Foundation. Use the following Opto 22 software for the OPC versions shown:

  - **OPC DA 2.0—OptoOPCServer**—An Opto 22 OPC server for exchanging data between OPC clients and the SNAP PAC System or legacy SNAP Ethernet-based I/O units. OptoOPCServer is OPC 2.0 compliant.
  - **OPC DA 1.0—OptoServer**—An older Opto 22 OPC server used with legacy FactoryFloor systems. For OPC 1.0-compliant or DDE clients.

- **OptoMMP**—An open industrial Ethernet protocol developed by Opto 22 for accessing the memory map in its Ethernet-based controllers and brains. Documented in Opto 22 form 1465, *OptoMMP Protocol Guide*.
- **Optomux**—An open serial protocol developed by Opto 22, which became an industry standard. Optomux runs on RS-485/422 and is documented in form 1572, the *Optomux Protocol Guide*.
- **Pamux**—An open parallel protocol developed by Opto 22 for fast digital control over a parallel cable. Also supports analog I/O. See individual product documentation and form 726, the *Pamux Manual*.
- **Peer-to-peer—Scratch Pad**—For peer-to-peer communication, the Scratch Pad is an area in the memory map of SNAP PAC controllers and brains (and older Ethernet-based I/O units) that can be used to store and retrieve data. For details on types of data and how to use the Scratch Pad, see the *PAC Control User's Guide* (form 1700 for SNAP PAC controllers and brains, or form 1710 for legacy hardware).
- **PPP**—Point-to-Point Protocol (Internet protocol) is at the link layer, just above the physical layer. While technically an Internet Protocol, PPP requires a serial connection to a dial-up modem for communication.
- **PROFIBUS DP**—An open serial fieldbus protocol originally developed by an industry group in Germany and widely used in Europe today.
- **SMTP**—Simple Mail Transfer Protocol (Internet protocol). Also an application layer protocol, SMTP is used to send email.
- **SNMP**—Simple Network Management Protocol (Internet protocol). Runs at the application layer, over TCP or UDP. SNMP is primarily used by IT departments for managing devices on an Ethernet network.
- **SQL database—**One communication option for interacting with database data is OptoDataLink, an Opto 22 software application that lets you move data easily between the SNAP PAC System and popular databases such as Microsoft SQL Server, Microsoft Access, and MySQL. Another option is PAC Display HMI development and runtime software. You can also exchange data with

databases using the SNAP PAC REST API or *groov* API and your JSON-compatible programming language (for example, Transact-SQL with Microsoft SQL Server).

- **SSI**—Serial Synchronous Interface (serial protocol). Often used for communication between an absolute position sensor and a controller for motion control. Runs on RS-422.
- **TL1**—Transaction Language 1 (a serial protocol) is an element management protocol widely used in telecommunications in North America.
- **TCP/IP**—Transmission Control Protocol/Internet Protocol (Internet protocol), the first two networking protocols defined in the Internet Protocol Suite. IP is just above the link layer and provides for the delivery of data packets. TCP runs at the transport layer, above IP, and delivers data reliably by setting up initial data channels and verifying receipt of data. TCP also resends data if no response is received within a given time.
- **UDP/IP**—User Datagram Protocol (Internet protocol), also running at the transport layer over IP. UDP, in contrast to TCP, delivers data quickly by sending it without having to set up data channels in advance and without waiting for a response. UDP is useful for time-sensitive data because it is faster, but data may be lost. The application must verify data delivery.

Form 1820-170216 • 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