OPTOMMP PROTOCOL GUIDE

Used with:
SNAP PAC R-Series Controllers
SNAP PAC S-Series Controllers
SNAP PAC EB Brains
SNAP PAC SB Brains
SNAP Simple I/O™
SNAP Ethernet I/O™
SNAP Ultimate I/O™
SNAP-LCE Controller
E1 Brain Board
E2 Brain Board
G4EB2 Brain Boards

Form 1465-130906—September 2013

OPTO 22
Automation made simple.

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

OptoMMP Protocol Guide
Form 1465-130906—September 2013

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or newer are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

Opto 22
Automation Made Simple

Table of Contents

1: Introduction

Welcome to OptoMMP

OptoMMP is a memory-mapped protocol based on the IEEE 1394 standard. This protocol is used to create custom software applications for remote monitoring, industrial control, and data acquisition using the following hardware products from Opto 22:

Device type	Part numbers	
SNAP PAC R-series on-the-rack controllers	SNAP-PAC-R1 SNAP-PAC-R1-FM SNAP-PAC-R1-W	SNAP-PAC-R2 SNAP-PAC-R2-FM SNAP-PAC-R2-W
SNAP PAC S-series standalone controllers	SNAP-PAC-S1 SNAP-PAC-S1-FM SNAP-PAC-S1-W	SNAP-PAC-S2 SNAP-PAC-S2-W
SNAP PAC EB brains	SNAP-PAC-EB1 SNAP-PAC-EB1-FM SNAP-PAC-EB1-W	SNAP-PAC-EB2 SNAP-PAC-EB2-FM SNAP-PAC-EB2-W
SNAP PAC SB brains	SNAP-PAC-SB1	SNAP-PAC-SB2
SNAP Simple I/O	SNAP-ENET-S64	
SNAP Ethernet I/O	SNAP-B3000-ENET SNAP-ENET-RTC	SNAP-ENET-D64
SNAP Ultimate I/O	SNAP-UP1-ADS SNAP-UP1-M64	SNAP-UP1-D64
SNAP-LCE standalone controller	SNAP-LCE	
E1 digital brain board	E1	
E2 analog brain board	E2	
G4EB2 brain boards	G4EB2 G4D32EB2	G4D32EB2-UPG

Communication Options

Custom applications (Microsoft® Windows®-based)—If you need to write your own custom applications in Visual Basic® or C++®, this guide contains documentation for the OptoMMP

Communication Toolkit with ActiveX components and C++ classes. The toolkit is included on the CD that came with your Opto 22 hardware.

For Visual Studio 2005, 2008, and 2010, a toolkit for .NET developers is available for free download on our website, www.opto22.com. This toolkit provides an easy-to-use interface so you can quickly create applications for Opto 22 products that use the OptoMMP protocol. The toolkit is built using Microsoft's .NET 3.5 Framework.

Custom applications (non-Windows)—If you are programming for Linux® or another operating system other than Microsoft Windows, the OptoMMP protocol for communicating with Opto 22 memory-mapped devices is open and documented in this guide.

PAC Control™—Most of the devices that use the OptoMMP can be used with PAC Control, Opto 22's flowchart-based control development software. PAC Control commands communicate with the device's memory map. PAC Control Basic is included in your purchase of a SNAP PAC controller and is also available for free download from our website at www.opto22.com. PAC Control Professional adds more features and is available for purchase on our website or through an Opto 22 distributor. *If you are using PAC Control to communicate with these devices, you probably do not need this guide.*

EtherNet/IP™—SNAP PAC controllers and SNAP PAC EB brains with firmware 8.3 and higher can also communicate with Allen-Bradley® Logix-based PLCs and other systems that use EtherNet/IP. For details, see Opto 22 form #1770, the *EtherNet/IP for SNAP PAC Protocol Guide*, available on our website. The easiest way to find it is to search on the form number.

Modbus®/TCP—Ethernet-based devices that use the OptoMMP protocol can also communicate using Modbus/TCP. The Modbus memory map is not included in this guide; see Opto 22 form #1678, the *Modbus/TCP Protocol Guide*. This form is available on our website; you can find it by searching on the form number.

Optomux protocol—E1 and E2 brain boards can also communicate using the Optomux protocol over Ethernet or serial networks. The Optomux Protocol Driver is documented in form #1572, the *Optomux Protocol Guide*; the driver and the document are included on the CD that came with the brain board and are also available from our website at www.opto22.com.

About this Guide

This guide shows you how to use the OptoMMP Communication Toolkit to configure and communicate with Opto 22 memory-mapped hardware. If you are not using the OptoMMP Toolkit or protocol, but are communicating with these devices using EtherNet/IP, Modbus/TCP, flowcharts developed in PAC Control, or the Optomux protocol, you do not need this guide.

This guide assumes that you are already familiar with programming in the format you've chosen to use. It also assumes that you have already installed the Opto 22 hardware. If you have not, see the hardware user's guide for instructions.

The following sections are included in this user's guide:

Chapter 1: Introduction—information about the guide and how to reach Opto 22 Product Support.

Chapter 2: Overview of Programming—basic information you need for programming applications using the OptoMMP protocol.

Chapter 3: Using the OptoMMP Communication Toolkit—programming your own Windows applications using our ActiveX components or C++ classes to hide the details of the memory map and the protocol.

Chapter 4: Using the OptoMMP Protocol—details of the OptoMMP protocol, including examples, for non-Windows programmers.

Appendix A: Opto 22 Hardware Memory Map— the complete memory map for Opto 22 devices using the OptoMMP protocol.

Information Key

This guide includes information that applies to some types of Opto 22 memory-mapped products but not to others. Sections are marked as follows to indicate the products that support them:

This text	Indicates support by this hardware
PAC-R	SNAP PAC R-series controllers
PAC-S	SNAP PAC S-series controllers
EB	SNAP PAC EB brains
SB	SNAP PAC SB brains
UIO	SNAP Ultimate I/O
EIO	SNAP Ethernet I/O
SIO	SNAP Simple I/O
LCE	SNAP-LCE controllers
E1	E1 brain boards
E2	E2 brain boards
G4EB2	G4EB2 brain boards

Other Documents You May Need

See the following additional guides for the information listed. Most guides are available on our website, www.opto22.com; those that are not on the website are available with product purchase. The easiest way to find a guide is to search on its form number.

Some of these guides are included on the CD that came with your Opto 22 hardware, but the most recent versions are always on our website.

For this information	See this guide	Form #
Installing and using a SNAP PAC R-series controller	*SNAP PAC R-Series Controller User's Guide*	1595
Installing and using a SNAP PAC S-series controller	*SNAP PAC S-Series Controller User's Guide*	1592
Installing and using a SNAP PAC EB or SB brain	*SNAP PAC Brain User's Guide*	1690
Installing and using a SNAP Ultimate, SNAP Ethernet, or SNAP Simple I/O unit	*SNAP Ethernet-Based I/O Units User's Guide*	1460
Installing and using a SNAP-LCE controller	*SNAP-LCE User's Guide*	1475
Installing and using an E1 or E2 brain board	*EI and E2 User's Guide*	1563

For this information	See this guide	Form #
Assigning IP addresses to all hardware, configuring communications, doing real-time reads and writes, updating firmware. For PAC-R, EB, SB, UIO, EIO, SIO, and G4EB2 units, configuring I/O points and system functions.	*PAC Manager User's Guide, Legacy Edition*	1714
Designing flowchart-based control programs for the system (requires a SNAP PAC controller; SNAP-LCE and SNAP Ultimate controllers are partially supported)	*PAC Control User's Guide*	1700
	PAC Control Command Reference	1701
	PAC Control Commands Quick Reference	1703
Communicating with the system using OPC	*OptoOPCServer User's Guide*	1439
Exchanging system data with popular databases	*OptoDataLink User's Guide*	1705
Communicating with Allen-Bradley Logix-based PLCs or other systems using EtherNet/IP	*EtherNet/IP for SNAP PAC Protocol Guide*	1770
Communicating with systems using Modbus/TCP	*Modbus/TCP Protocol Guide*	1678
Communicating with an E1 or E2 I/O unit using the Optomux protocol	*Optomux Protocol Guide*	1572

For Help

If you have problems using the OptoMMP protocol and cannot find the help you need in this guide or on our website, contact Opto 22 Product Support.

Phone: 800-TEK-OPTO (800-835-6786)
951-695-3080
(Hours are Monday through Friday, 7 a.m. to 5 p.m. Pacific Time)

NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.

Fax: 951-695-3017

Email: support@opto22.com

Opto 22 website: www.opto22.com

When calling for technical support, be prepared to provide the following information about your system to the Product Support engineer:

- Software and version being used
- Firmware and loader versions for brains, brain boards, and controllers used (available through PAC Manager; see the Troubleshooting chapter in the device's user's guide)
- PC configuration (type of processor, speed, memory, and operating system)
- A complete description of your hardware and operating systems, including:
  - jumper configuration (if applicable)
  - IP addresses and net masks for devices on the system
  - accessories installed (such as expansion cards)
  - third-party devices installed (for example, barcode readers)
  - type of power supply
- Specific error messages seen

Rack and Module Compatibility

The following table shows rack and module compatibility for Opto 22 memory-mapped I/O processors (brains, brain boards, and on-the-rack controllers).

Processor Family	Part Number	Mounting rack	Modules per mounting rack	Module family	Maximum modules allowed per I/O unit (largest rack)
SNAP PAC R-series	SNAP-PAC-R1 SNAP-PAC-R1-FM SNAP-PAC-R1-W	SNAP-PAC-RCK4[1] SNAP-PAC-RCK4-FM SNAP-PAC-RCK8 SNAP-PAC-RCK8-FM SNAP-PAC-RCK12 SNAP-PAC-RCK12-FM SNAP-PAC-RCK16 SNAP-PAC-RCK16-FM	4, 8, 12, or 16	SNAP I/O	16 4-channel digital 16 analog 8 serial 16 high-density digital
	SNAP-PAC-R2 SNAP-PAC-R2-FM SNAP-PAC-R2-W				16 4-channel digital[2] 16 analog 8 serial 16 high-density digital
SNAP PAC EB brains	SNAP-PAC-EB1 SNAP-PAC-EB1-FM SNAP-PAC-EB1-W	SNAP-PAC-RCK4[1] SNAP-PAC-RCK4-FM SNAP-PAC-RCK8 SNAP-PAC-RCK8-FM SNAP-PAC-RCK12 SNAP-PAC-RCK12-FM SNAP-PAC-RCK16 SNAP-PAC-RCK16-FM	4, 8, 12, or 16	SNAP I/O	16 4-channel digital 16 analog 8 serial 16 high-density digital
	SNAP-PAC-EB2 SNAP-PAC-EB2-FM SNAP-PAC-EB2-W				16 4-channel digital[2] 16 analog 8 serial 16 high-density digital
SNAP PAC SB brains	SNAP-PAC-SB1 SNAP-PAC-SB1-FM	SNAP-PAC-RCK4[1] SNAP-PAC-RCK4-FM SNAP-PAC-RCK8 SNAP-PAC-RCK8-FM SNAP-PAC-RCK12 SNAP-PAC-RCK12-FM SNAP-PAC-RCK16 SNAP-PAC-RCK16-FM	4, 8, 12, or 16	SNAP I/O	16 4-channel digital 16 analog 16 high-density digital
	SNAP-PAC-SB2				16 4-channel digital[2] 16 analog 16 high-density digital
SNAP Simple I/O	SNAP-ENET-S64	SNAP-M16[1] SNAP-M32 SNAP-M48 SNAP-M64	4, 8, 12, or 16	SNAP I/O	16 4-channel digital[2] 16 analog 8 serial 16 high-density digital
SNAP Ethernet I/O	SNAP-B3000-ENET SNAP-ENET-RTC	SNAP-B racks	4, 8, 12, or 16	SNAP I/O	8 4-ch digital (first 8 slots) 16 analog 8 serial 16 high-density digital
	SNAP-ENET-D64	SNAP-D64RS	16	SNAP I/O	16 4-channel digital[2]
SNAP Ultimate I/O	SNAP-UP1-ADS	SNAP-B racks	4, 8, 12, or 16	SNAP I/O	8 4-ch digital (first 8 slots) 16 analog 8 serial 16 high-density digital
	SNAP-UP1-M64	SNAP-M16[1] SNAP-M32 SNAP-M48 SNAP-M64	4, 8, 12, or 16	SNAP I/O	16 4-channel digital[2] 16 analog 8 serial 16 high-density digital
	SNAP-UP1-D64	SNAP-D64RS	16	SNAP I/O	16 4-channel digital[2]
E1 Brain Board	E1	Digital G4PB series[3] Digital PB series[3] for G1 or Quad Pak	G4:16 G1:16 Quad Pak: 4	G4 G1 Quad Pak	G4: 16 digital G1: 16 digital Quad Pak: 4
E2 Brain Board	E2	Analog PB series	G1:16	G1	G1: 16 analog

Processor Family	Part Number	Mounting rack	Modules per mounting rack	Module family	Maximum modules allowed per I/O unit (largest rack)
G4EB2 brains	G4EB2	G4PB32H PB32HQ	G4: 32 Quad Pak: 4	G4 Quad Pak	G4: 32 digital Quad Pak: 4
	G4D32EB2	(Included)	32	G4	32 digital
	G4D32EB2-UPG	G4D32RS digital brick	32	G4	32 digital

1 SNAP PAC racks and M-series racks can be used interchangeably.
2 Digital point features are simplified.
3 E1 brain boards also work with PB and G4PB integral racks.

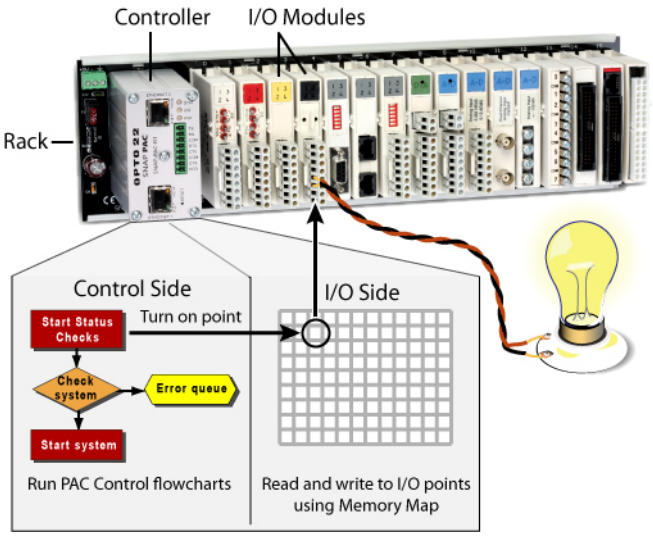
For additional compatibility information, such as the specific modules supported by each processor, see form #1693, Legacy and Current Product Comparison and Compatibility Charts.

Accessing Opto 22 Ethernet-Based Devices over the Internet

Since Opto 22 Ethernet-based devices are just like any other hardware on the Ethernet network, you can access them over the Internet in exactly the same way you would access a computer. The details depend on your network and Internet connection. Consult your system or network administrator or your Internet Service Provider (ISP) for more information.

Special Notes for SNAP PAC R-Series and SNAP Ultimate I/O Units

SNAP PAC R-series and SNAP Ultimate I/O system architecture is explained in the Architecture chapter of their user's guides. (If you haven't already read this chapter, it's a good idea to read it first.) As their user's guides explain, these controllers handle input/output processing and flowchart-based control on two "sides" of the controller, as illustrated below:



This example rack is shown with the SNAP PAC R controller on the left, and the input/output modules on the right.

The modules on the rack connect with devices such as the light bulb to monitor or control them.

The combination of rack, controller, and modules shown in the illustration is referred to as an *I/O unit*.

The I/O Side

The SNAP PAC R-series or SNAP Ultimate controller reads and writes to the I/O points on the rack using its I/O side memory map. The complete list of memory map addresses is in Appendix A. On its I/O side, the controller has features similar to a SNAP PAC EB brain or a SNAP Ethernet brain.

The Control Side

In the control side, the SNAP PAC R-series or SNAP Ultimate controller can run a control strategy, a control program you can develop in PAC Control. The strategy provides the logic that controls processes through the system. Commands within the strategy read from and write to the memory map in the I/O side of the controller in order to monitor and control the I/O points. On its control side, the controller has capabilities very much like a traditional Opto 22 industrial controller, such as the SNAP PAC S-series or SNAP-LCE.

Using the Two Sides

Many customers who use SNAP PAC R-series or SNAP Ultimate I/O units never need to think about the I/O side and its memory map. While the controller has features on its I/O side that are like a brain, most of these features are easier to use in a PAC Control strategy running on the control side. If your only communication with the controller is through a PAC Control strategy, you do not need to use this programming guide or the memory map at all. You should use PAC Control to configure the system and develop your monitoring and control strategy. The PAC Control guides will provide the information you need.

However, if you are using another method to communicate with the controller, either instead of or in addition to a PAC Control strategy, you need the additional information provided in this programming guide.

Remember that because these I/O systems are so versatile, you can use several methods at once to do several things. For example, the system might be running a PAC Control strategy, providing data to a peer on the network, and communicating with a third-party HMI, all at once.

Any time you are communicating with both the control side and the I/O side at once, plan carefully and use caution. The control side reads from and writes to the I/O side just as other methods do directly. Make sure that data being written directly to the I/O side does not conflict with control logic being executed in the PAC Control strategy.

Opto 22 Processor Comparison Chart

Some of the features mentioned in this guide apply to some models and not others. (Note that E1 and E2 brain boards have additional features if they are used with the Optomux protocol. See the *E1 and E2 User's Guide*, form #1563, for more information.)

Feature		SW	Current Hardware														Legacy Hardware								
			SNAP PAC Controller								SNAP PAC Brain														
		SoftPAC	SNAP-PAC-S1 SNAP-PAC-S1-FM	SNAP-PAC-S1-W	SNAP-PAC-S2	SNAP-PAC-S2-W	SNAP-PAC-R1 SNAP-PAC-R1-FM	SNAP-PAC-R1-W	SNAP-PAC-R2 SNAP-PAC-R2-FM	SNAP-PAC-R2-W	SNAP-PAC-EB1 SNAP-PAC-EB1-FM	SNAP-PAC-EB1-W	SNAP-PAC-EB2 SNAP-PAC-EB2-FM	SNAP-PAC-EB2-W	SNAP-PAC-SB1	SNAP-PAC-SB2	G4EB2	E1 Brain Board	E2 Brain Board	SNAP Simple SNAP-ENET-S64	SNAP Ethernet SNAP-B3000-ENET SNAP-ENET-RTC	SNAP Ethernet SNAP-ENET-D64	SNAP Ultimate SNAP-UP1-ADS	SNAP Ultimate SNAP-UP1-D64	SNAP Ultimate SNAP-UP1-M64
Digital I/O Points	Input latching	n/a	n/a				●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●
	Watchdog timer						●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●
	On/off status						●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●
	High-speed counting (up to 20 kHz)[1]						●	●			●	●			●						●		●		
	Quadrature counting[2]						●	●			●	●			●						●		●		
	On-pulse and off-pulse measurement[1]						●	●			●	●			●						●		●		
	Frequency and period measurement						●	●			●	●			●										
	TPO (time-proportional output)[3]						●	●	●	●	●	●	●	●	●	●	●				●		●		
	Pulse generation (N pulses, continuous square wave, on-pulse, off-pulse)[3]						●	●	●	●	●	●	●	●	●	●	●				●		●		
	Digital totalizing						●	●	●	●	●	●	●	●	●	●	●								

Feature		SW	Current Hardware														Legacy Hardware							
			SNAP PAC Controller								SNAP PAC Brain										SNAP Simple	SNAP Ethernet		SNAP Ultimate		
		SoftPAC	SNAP-PAC-S1 SNAP-PAC-S1-FM	SNAP-PAC-S1-W	SNAP-PAC-S2	SNAP-PAC-S2-W	SNAP-PAC-R1 SNAP-PAC-R1-FM	SNAP-PAC-R1-W	SNAP-PAC-R2 SNAP-PAC-R2-FM	SNAP-PAC-R2-W	SNAP-PAC-EB1 SNAP-PAC-EB1-FM	SNAP-PAC-EB1-W	SNAP-PAC-EB2 SNAP-PAC-EB2-FM	SNAP-PAC-EB2-W	SNAP-PAC-SB1	SNAP-PAC-SB2	G4EB2	E1 Brain Board	E2 Brain Board	SNAP-ENET-S64	SNAP-B3000-ENET SNAP-ENET-RTC	SNAP-ENET-D64	SNAP-UP1-ADS	SNAP-UP1-D64	SNAP-UP1-M64
Analog I/O points	Thermocouple linearization (32-bit floating point for linearized values)	n/a	n/a				●	●	●	●	●	●	●	●	●	●			●	●	●		●		●
	Minimum/maximum values						●	●	●	●	●	●	●	●	●	●			●	●	●		●		●
	Offset and gain						●	●	●	●	●	●	●	●	●	●			●	●	●		●		●
	Scaling						●	●	●	●	●	●	●	●	●	●			●	●	●		●		●
	Time-proportional output[4]						●	●	●	●	●	●	●	●	●	●				●	●		●		●
	Output clamping						●	●	●	●	●	●	●	●	●	●			●	●	●		●		●
	Filter weight						●	●	●	●	●	●	●	●	●	●				●	●		●		●
	Watchdog timer						●	●	●	●	●	●	●	●	●	●				●	●		●		●
	Ramping[3]						●	●	●	●	●	●	●	●	●	●			●		●		●		
	Analog totalizing[3]						●	●	●	●	●	●	●	●	●	●									
SNAP high-density digital modules			n/a				●	●	●	●	●	●	●	●	●	●				●	●		●		●
SNAP analog modules with more than 4 channels							●	●	●	●	●	●	●	●	●	●				●			●		●
Serial communication modules							●	●	●	●	●	●	●	●						●	●		●		●
Ethernet network		●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●
Two independent Ethernet interfaces (two IP addresses)		10	●	●	●	●	●	●	●	●															
Two switched Ethernet interfaces (one IP address)											●	●	●	●			●								
One wireless LAN interface (separate IP address)				●		●		●		●		●		●											
Serial network (RS-485/422)			●	●	●	●									●	●		●	●						
Serial ports (RS-232) for PPP or serial devices			●	●	●	●	●	●	●	●														●	●	●
OptoMMP protocol		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Optomux protocol (over Ethernet or serial)																		●	●						
Modbus/TCP			●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●
EtherNet/IP (Allen-Bradley® Logix PLCs)			●	●	●	●	●	●	●	●	●	●	●	●			●								
Runs PAC Control strategies		●	●	●	●	●	●	●	●	●														●[5]	●[5]	●[5]

Feature		SW	Current Hardware															Legacy Hardware								
			SNAP PAC Controller								SNAP PAC Brain									SNAP Simple	SNAP Ethernet		SNAP Ultimate		
		SoftPAC	SNAP-PAC-S1 SNAP-PAC-S1-FM	SNAP-PAC-S1-W	SNAP-PAC-S2	SNAP-PAC-S2-W	SNAP-PAC-R1 SNAP-PAC-R1-FM	SNAP-PAC-R1-W	SNAP-PAC-R2 SNAP-PAC-R2-FM	SNAP-PAC-R2-W	SNAP-PAC-EB1 SNAP-PAC-EB1-FM	SNAP-PAC-EB1-W	SNAP-PAC-EB2 SNAP-PAC-EB2-FM	SNAP-PAC-EB2-W	SNAP-PAC-SB1	SNAP-PAC-SB2	G4EB2	E1 Brain Board	E2 Brain Board	SNAP-ENET-S64	SNAP-B3000-ENET SNAP-ENET-RTC	SNAP-ENET-D64	SNAP-UP1-ADS	SNAP-UP1-D64	SNAP-UP1-M64
Compatible with PAC Control (using SNAP PAC controller)		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●[5]	●[5]	●[5]	●[5]	●[5]	●[5]
Legacy software support	Runs ioControl strategies	●	●	●	●	●	●	●	●	●													●	●	●
	Compatible with ioControl (through SNAP PAC, SNAP-LCE, or SNAP Ultimate controller running ioControl)	●	●	●	●	●	●	●	●	●								●	●	●	●	●	●	●	●
	Compatible with OptoControl (through Opto 22 controller with Ethernet card)		6	6	6	6	6	6	6	6										●	●	●			
UDP Streaming			●	●	●	●	●	●	●	●	●	●	●	●			●			●	●	●	●	●	●
SNMP (network management)			●	●	●	●	●	●	●	●	●	●	●	●			●				●	●	●	●	●
PPP (dial-up modems)			●	●	●	●	●	●	●	●											●	●	●	●	●
FTP server			●	●	●	●	●	●	●	●	●	●	●	●			●						●	●	●
FTP client[7]		●					●	●	●	●													●	●	●
Email (SMTP client)		●					●	●	●	●	●	●	●	●			●				●	●	●	●	●
OPC driver		●					●	●	●	●	●	●	●	●	●[9]	●[9]	●	●	●	●	●	●	●	●	●
Security for wireless network (WPA2-AES, WPA-TKIP, WEP)		10		●		●		●		●		●		●											
Security for wired Ethernet network (IP filtering, port access)		10	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●
PID logic on the brain Number of PIDs available		n/a	n/a				● 96	● 96	● 96	● 96	● 96	● 96	● 96	● 96	● 96	● 96					● 16		● 32		● 32
Digital events							●	●	●	●	●	●	●	●	●	●	●				●[1]	●[1]	●[1]	●[1]	●[1]
Alarm events							●	●	●	●	●	●	●	●	●	●					●		●		●
Serial events							●	●	●	●	●	●	●	●							●		●		●
Event messages							●	●	●	●	●	●	●	●			●				●	●	●	●	●
Data logging in the brain							●	●	●	●	●	●	●	●			●				●		●		●
I/O point data mirroring							●	●	●	●	●	●	●	●			●				●		●		●
Memory map data copying							●	●	●	●	●	●	●	●			●				●		●		●

Feature		Current Hardware															Legacy Hardware								
	SW	SNAP PAC Controller								SNAP PAC Brain									SNAP Simple	SNAP Ethernet		SNAP Ultimate		
	SoftPAC	SNAP-PAC-S1 SNAP-PAC-S1-FM	SNAP-PAC-S1-W	SNAP-PAC-S2	SNAP-PAC-S2-W	SNAP-PAC-R1 SNAP-PAC-R1-FM	SNAP-PAC-R1-W	SNAP-PAC-R2 SNAP-PAC-R2-FM	SNAP-PAC-R2-W	SNAP-PAC-EB1 SNAP-PAC-EB1-FM	SNAP-PAC-EB1-W	SNAP-PAC-EB2 SNAP-PAC-EB2-FM	SNAP-PAC-EB2-W	SNAP-PAC-SB1	SNAP-PAC-SB2	G4EB2	E1 Brain Board	E2 Brain Board	SNAP-ENET-S64	SNAP-B3000-ENET SNAP-ENET-RTC	SNAP-ENET-D64	SNAP-UP1-ADS	SNAP-UP1-D64	SNAP-UP1-M64
Scratch Pad area (peer-to-peer communication)		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
Bits	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●	●	●	●	●
Floats	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●	●
Strings	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●	●
Integers (32 bit)	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●	●
Integers (64 bit)	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
Realtime clock (RTC)	10	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				● 8		●	●	●

1 Four-channel SNAP digital modules only; not available on high-density digital modules.
2 Requires a SNAP quadrature input module (SNAP-IDC5Q).
3 Available when used with PAC Control Pro 8.2 or newer, or PAC Control Basic 9.0 or newer, and a SNAP PAC controller.
4 Requires a SNAP analog TPO module (SNAP-AOD-29).
5 Compatible with PAC Control using firmware 7.1 or newer; however, several 8.x features are not available.
6 Converts OptoControl strategies to PAC Control, when used with PAC Control Professional.
7 FTP client provided by PAC Control strategy.
8 Applies to SNAP-ENET-RTC, not to SNAP-B3000-ENET.
9 Available when used with OptoOPCServer, PAC Control, and a SNAP PAC controller.
10 As provided by the Microsoft Windows computer the software runs on..

2: Overview of Programming

Introduction

Details of programming steps depend on the language or method you use for developing your custom application, but some basic information is common to all languages and methods. This chapter includes the following basic information you need in order to program for Opto 22 memory-mapped hardware:

Except for E1 and E2 brain boards, you can configure points and functions in your program or in PAC Manager (see the *PAC Manager User's Guide* for instructions). For E1 or E2 brain boards, you must follow instructions in form #1576, *I/O Configuration for E1 and E2 Brain Boards*.

Programming Steps for Opto 22 Devices

PAC-R
PAC-S
EB
SB
UIO
EIO
SIO
LCE
E1
E2
G4EB2

In general, writing programs for Opto 22 memory-mapped devices using OptoMMP involves six steps:

1. Connect to the device (not required for UDP).
2. Send a powerup clear (PUC) to the device.
3. Configure points and point features.
4. (Optional) Configure event/reactions, security, streaming, and other functions.

   NOTE: Store configurations to flash memory if you want them to remain through power cycles.

5. Read and write to points.
6. Disconnect from the device (not required for UDP). This step occurs only after all communication is finished. The connection is left open during normal communications.

Understanding the Memory Map

PAC-R
PAC-S
EB
SB
UIO
EIO
SIO
LCE
E1
E2
G4EB2

Opto 22 memory-mapped devices use the IEEE 1394 specification to provide a standard for reading and writing data. This standard specifies a memory-mapped model for devices on a network. Basically, each node (such as a SNAP Ethernet brain) appears logically as a 48-bit address space. To communicate with a device, you read from and write to specific memory addresses in that space.

You can think of the memory map as a grid of mailboxes, with each mailbox having its own memory address. Each mailbox address has a specific purpose. For example, one address is used to turn on a single digital point; another address turns the same point off. There's an address that stores the device's firmware version, and one that contains the minimum value of a specific analog point.

All these addresses and many more are detailed in Appendix A. This memory map applies to all SNAP PAC controllers and brains; SNAP Ultimate, Ethernet, and Simple brains; SNAP-LCE controllers; E1 and E2 brain boards, and G4EB2 brains. The memory map also applies to all programming methods except EtherNet/IP, Modbus, and Optomux. (For EtherNet/IP, see form #1770, the *EtherNet/IP for SNAP PAC Protocol Guide*. For Modbus, see form #1678, the *Modbus/TCP Protocol Guide*. For Optomux, see Opto 22 form #1572, the *Optomux Protocol Guide*.)

However, devices vary in how much of the memory map they support. For example, the SNAP PAC S-series standalone controller does not support reading or writing to points, since it is not directly connected to points. Similarly, a digital-only brain does not support analog sections of the memory map. Appendix A shows the devices each section supports.

Also note that features included in the memory map may not always be the same as features available through another protocol. For example, waveform generation is supported on an E2 using the Optomux protocol, but it is not available on an E2 in OptoMMP. For a list of features supported by each device, see "Opto 22 Processor Comparison Chart" on page 9.

Communication Using the Memory Map

To change the configuration or status of an I/O point or to write other data, the host (computer or controller) sends the memory map address and the new data to be written. The memory-mapped device responds by returning a packet indicating success or failure.

The host can also access the status of I/O points and obtain other data by reading the appropriate addresses from the memory map. The host simply asks for data from those addresses, and the memory-mapped device returns the data.

To find the memory map address you need in order to read or write to points, you need to know how to reference I/O modules and the points on them. Referencing depends on the type of brain and I/O you're using:

For SNAP I/O, see the next section.

For E1s and E2s, see page 18.

For G4EB2s, see page 19.

Referencing I/O Points

To find the memory map address you need in order to read or write to points, you need to know the module and point numbers. This section illustrates the module and point numbers used with specific racks and processors (see "Rack and Module Compatibility" on page 5).

For serial modules, also see page 134.

NOTE: This section describes I/O point referencing for all methods of communication except Modbus/TCP and Optomux. For Modbus I/O point referencing, see the Modbus chapter in form #1678. For Optomux, see the Optomux Protocol Guide, form #1572.

See the illustration for your rack:

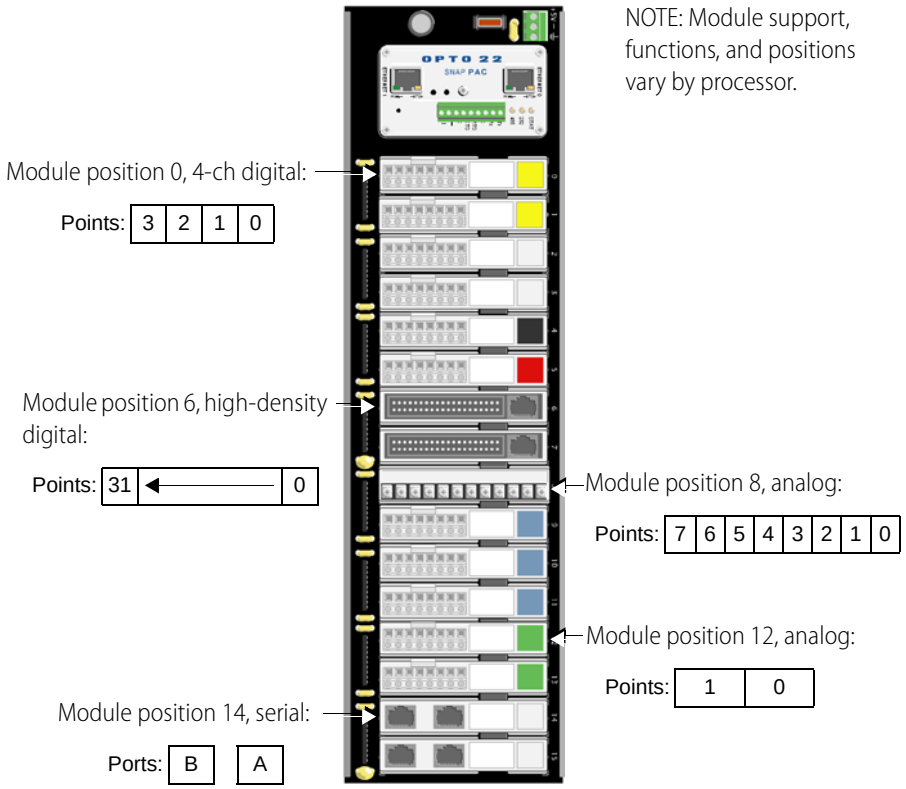
PAC-R
EB
SB
UIO
EIO
SIO

SNAP PAC Racks

Use any of these racks	With any of these processors
SNAP-PAC-RCK4 (or SNAP-M16) SNAP-PAC-RCK4-FM SNAP-PAC-RCK8 (or SNAP-M32) SNAP-PAC-RCK8-FM SNAP-PAC-RCK12 (or SNAP-M48) SNAP-PAC-RCK12-FM SNAP-PAC-RCK16 (or SNAP-M64) SNAP-PAC-RCK16-FM	SNAP-PAC-R1, SNAP-PAC-R1-FM, SNAP-PAC-R1-W SNAP-PAC-R2, SNAP-PAC-R2-FM, SNAP-PAC-R2-W SNAP-PAC-EB1, SNAP-PAC-EB1-FM, SNAP-PAC-EB1-W SNAP-PAC-EB2, SNAP-PAC-EB2-FM, SNAP-PAC-EB2-W SNAP-PAC-SB1 SNAP-PAC-SB2 SNAP-ENET-S64 SNAP-UP1-M64

SNAP PAC mounting racks can hold up to 4, 8, 12, or 16 Opto 22 SNAP I/O modules. Point features and modules supported vary by processor; see the processor's data sheet for specifications.

Each module contains 1 to 32 points (channels), depending on the module. Examples of modules are shown in the following diagram.

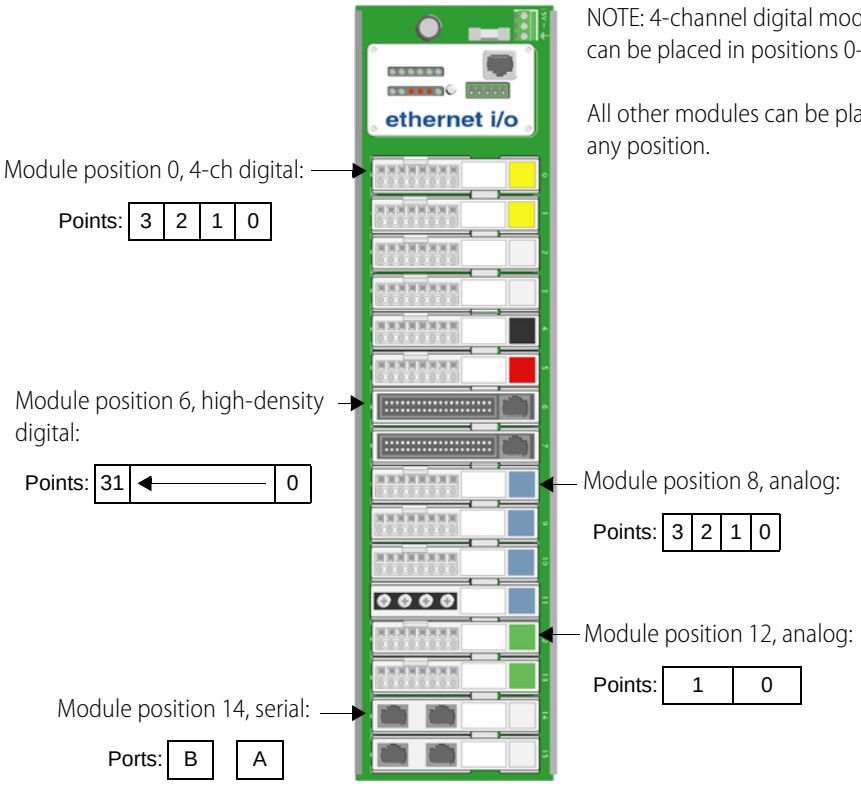


SNAP B-Series Racks

Use any of these racks			With any of these processors
SNAP-B4 SNAP-B8 SNAP-B12 SNAP-B16	SNAP-B8MC SNAP-B12MC SNAP-B16MC	SNAP-B8MC-P SNAP-B12MC-P SNAP-B16MC-P	SNAP-UP1-ADS SNAP-B3000-ENET SNAP-ENET-RTC

NOTE: SNAP B-series racks and the processors compatible with them are not recommended for new development. Use SNAP PAC racks and processors instead.

SNAP B-series mounting racks can hold up to 4, 8, 12, or 16 Opto 22 SNAP I/O modules. (Not all modules are supported by these processors; see form #1693 for details.) Analog, serial, and high-density digital modules (digital modules with more than four points) can be placed in any position. For the larger racks, 4-channel digital modules can be placed in positions 0–7 only. Each module contains 1 to 32 points (channels), depending on the module. Examples of modules are shown in the following diagram.



SNAP Digital-Only Rack

NOTE: Digital-only racks and processors are not recommended for new development. Use SNAP PAC racks and processors instead.

The SNAP-D64RS mounting rack is compatible with SNAP-UP1-D64 and SNAP-ENET-D64 processors. The rack holds up to16 4-channel SNAP digital I/O modules. Analog, serial, and high-density digital modules are not supported. Module position 0 is the position closest to the processor.

E1
E2

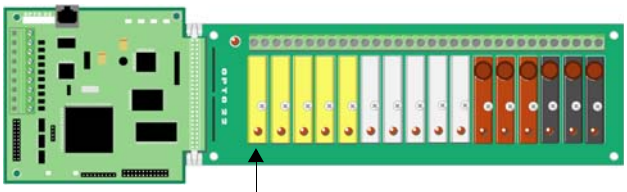
E1- and E2-Compatible Racks

When you use PAC Manager or the memory map with E1s and E2s, each module on the E1 or E2 corresponds to the *first point* on a similar SNAP module.

E1 brain boards can be used with a variety of modules. The maximum number of points is 16 on the largest rack, as shown below.

E1 shown with G4 modules.

Since each module has just one point, use only the first point for each module in the memory map.



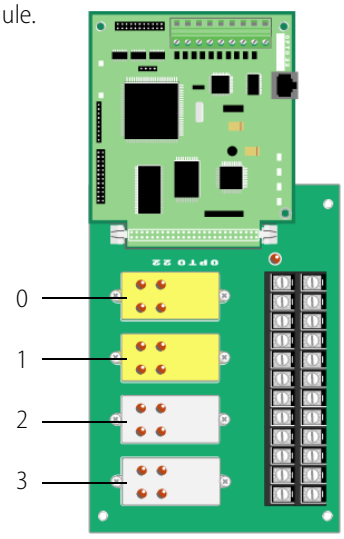
Module position 0

Module #	Point #
0 ↓ 15	0 ↓ 0

E1 with Quad Pak modules.

Quad Pak modules have four input or four output points, but each point is treated as if it were a separate one-point module.

Module position on Quad Pak rack	Module number	Point number
0	0 1 2 3	0 0 0 0
1	4 5 6 7	0 0 0 0
2	8 9 10 11	0 0 0 0
3	12 13 14 15	0 0 0 0



E2 brain boards are used with analog G1 modules only. The largest rack (shown below) holds 16 single-point modules.

E2 with G1 modules.

Since each module has just one point, use only the first point for each module in the memory map.

Module position 0

Module #	Point #
0 ↓ 15	0 ↓ 0

G4EB2

Racks for G4EB2 Brains

G4EB2 brains include part numbers G4EB2, G4D32EB2-UPG, and G4D32EB2.

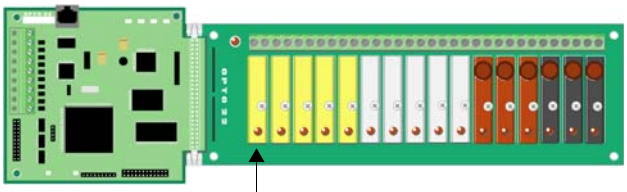
- **G4EB2** and **G4D32EB2-UPG** replace a 32-channel digital brain in a serial *mistic* or Pamux system with an Ethernet-based brain that uses the OptoMMP protocol.
- **G4D32EB2** is simply a new Ethernet-based brick that includes a G4EB2 brain and a G4D32RS rack.

These G4EB2 brains can be used with either G4 or Quad Pak modules on a suitable rack as shown:

Rack	Modules	Old brain PN	New brain PN	Old protocol	New protocol
G4PB32H	G4	B4	G4EB2	Pamux	Ethernet
PB32HQ	Quad Pak	B4	G4EB2	Pamux	Ethernet
G4D32RS	G4	G4RS	G4D32EB2-UPG	mistic	Ethernet
G4D32RS	G4	--	G4D32EB2	--	Ethernet

G4EB2 brains with G4 modules

Each module has just one point. However, in the memory map they are addressed like Quad Pak racks (see page 21). The difference is that with G4s, you can mix input and output modules within the same group of four points.



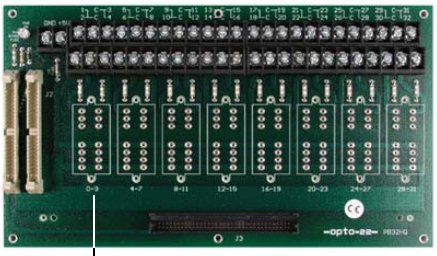
Module position 0

The following table shows Point Configuration addresses as an example of how to reference points on a rack with G4 modules.

G4 rack	Memory Map equivalent		
Module position	Module number	Point number	Starting address for point config
0	0	0	F010 0000
1	0	1	F010 00C0
2	0	2	F010 0180
3	0	3	F010 0240
4	1	0	F010 3000
5	1	1	F010 30C0
6	1	2	F010 3180
7	1	3	F010 3240
8	2	0	F010 6000
9	2	1	F010 60C0
10	2	2	F010 6180
11	2	3	F010 6240
12	3	0	F010 9000
13	3	1	F010 90C0
14	3	2	F010 9180
15	3	3	F010 9240
16	4	0	F010 C000
17	4	1	F010 C0C0
18	4	2	F010 C180
19	4	3	F010 C240
20	5	0	F010 F000
21	5	1	F010 F0C0
22	5	2	F010 F180
23	5	3	F010 F240
24	6	0	F011 2000
25	6	1	F011 20C0
26	6	2	F011 2180
27	6	3	F011 2240
28	7	0	F011 5000
29	7	1	F011 50C0
30	7	2	F011 5180
31	7	3	F011 5240

G4EB2 brains with Quad Pak modules

Quad Pak modules have four input or four output points per module, so each group of four points must be configured as either inputs or outputs.



Module position 0 (points 0–3)

The following table shows Point Configuration addresses as an example of how to reference points on a PB32HQ rack with Quad Pak modules.

Quad Pak rack		Memory Map equivalent		
Module position	**Point number**	**Module number**	**Point number**	**Starting address for point config**
0	0	0	0	F010 0000
	1	0	1	F010 00C0
	2	0	2	F010 0180
	3	0	3	F010 0240
1	4	1	0	F010 3000
	5	1	1	F010 30C0
	6	1	2	F010 3180
	7	1	3	F010 3240
2	8	2	0	F010 6000
	9	2	1	F010 60C0
	10	2	2	F010 6180
	11	2	3	F010 6240
3	12	3	0	F010 9000
	13	3	1	F010 90C0
	14	3	2	F010 9180
	15	3	3	F010 9240
4	16	4	0	F010 C000
	17	4	1	F010 C0C0
	18	4	2	F010 C180
	19	4	3	F010 C240
5	20	5	0	F010 F000
	21	5	1	F010 F0C0
	22	5	2	F010 F180
	23	5	3	F010 F240
6	24	6	0	F011 2000
	25	6	1	F011 20C0
	26	6	2	F011 2180
	27	6	3	F011 2240
7	28	7	0	F011 5000
	29	7	1	F011 50C0
	30	7	2	F011 5180
	31	7	3	F011 5240

Configuring I/O Points

Before you can read or write to I/O points, you must make sure point types and point features are configured as necessary.

Configuring Point Types

For point types, configuration requirements vary based on the type of I/O unit.

PAC-R
EB
SB
UIO
EIO
SIO

SNAP I/O units with analog capability. The I/O unit can recognize analog, serial, special-purpose, and high-density digital modules (digital modules with more than four points). The I/O unit assumes a default configuration for all points on those modules. Any module position not occupied by an analog, serial, special-purpose, or high-density digital module is assumed to be a digital input. (Note that SB brains do not support serial modules.)

For these I/O units, you must configure the following point types (see "(Expanded) Analog & Digital Point Configuration—Read/Write" on page 114):

- All digital output points on 4-channel digital modules. Use point type 180. (Not necessary for points on high-density digital modules, as they are automatically recognized. High-density digital modules are digital modules with more than four channels.)
- Analog points that do not use the default point type for the module. For example, if the points on a SNAP-AIRTD module are 120 Ohm Nickel 3-wire RTDs (–80 to +260 °C), they must be configured, because the default for that module is 100 Ohm Platinum 3-wire RTDs (–200 to +850 °C). Point types for analog modules are shown in the tables beginning on page 23. Default point types are indicated.

UIO
EIO
G4EB2

Digital-only SNAP and G4EB2 I/O units. The I/O unit assumes that all points are digital input points. You must configure all digital output points with the point type 180.

See "(Expanded) Analog & Digital Point Configuration—Read/Write" on page 114.

E1
E2

E1 and E2 brain boards. IMPORTANT! You must use PAC Manager to configure the I/O unit and point types for all points on the rack before trying to read or write to points. Follow the instructions in Opto 22 form #1576, *I/O Configuration for E1 and E2 Brain Boards*. If points are not configured following the steps in form #1576, points on E2s will return only a `~NAN` (not a number value).

PAC-R
EB
SB
UIO
EIO
SIO
E1
E2
G4EB2

Configuring Point Features

Point features vary based on the I/O processor (brain, brain board, or on-the-rack controller) and the module. The following point features are not automatic and must be configured for each point that uses them:

- Digital input counters and quadrature counters (Exception: Counters on high-density digital modules are automatic and don't need configuring unless you are using PAC Control.)
- Digital and analog watchdogs
- Analog scaling, clamping, offset and gain, and average filter weight

For SNAP and G4EB2 I/O units, see "Using I/O Point Features" on page 29 for a list of features and how to configure them. For E1 and E2 brain boards, see form #1576, *I/O Configuration for E1 and E2 Brain Boards*.

PAC-R
EB
SB
UIO
EIO
SIO
G4EB2

Point Type Configuration Tables

NOTE: For E1 and E2 brain boards, see form #1576, I/O Configuration for E1 and E2 Brain Boards.

The following tables help you configure points by showing the part number, the point type in decimal and in hex, and the module type in hex (module type is read-only). For analog modules, tables also include the number of points per module, the unit of measurement for the module, and its range.

Digital Input and Output Modules

Module & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)
4-channel digital input module*	256	100	00
4-channel digital output module*	384	180	00

* High-density digital modules are automatically recognized; points do not require configuration.

Analog Input Modules

NOTE: For E1 and E2 brain boards, see form #1576, I/O Configuration for E1 and E2 Brain Boards.

Use this data for configuring point types and features (see page 22). If a module has multiple listings, the default point type is shaded.

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low Scale	Full Scale	Overrange
SNAP-AIARMS: 0 - 10 A AC/DC	71	47	71	2	A	0.0	0.0	10.0	11.0
SNAP-AIARMS-i: 0 - 10 A AC/DC	71	47	29	2	A	0.0	0.0	10.0	11.0
SNAP-AIARMS-i-FM: 0 - 10 A AC/DC	71	47	29	2	A	0.0	0.0	10.0	11.0
SNAP-AICTD: ICTD Temp. Probe	4	4	04	2	Degrees C	-273.0	-40.0	150.0	150.0
SNAP-AICTD-4: ICTD Temp. Probe	4	4	42	4	Degrees C	-273.0	-40.0	150.0	150.0
SNAP-AICTD-8: ICTD Temp. Probe	4	4	4C	8	Degrees C	-273.0	-40.0	150.0	150.0

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low Scale	Full Scale	Overrange
SNAP-AILC: -2 - +2 mV/V Fast	34	22	0B	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC: -2 - +2 mV/V Slow	36	24	0B	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC: -3 - +3 mV/V Fast	35	23	0B	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC: -3 - +3 mV/V Slow	37	25	0B	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC: Filter of 1st channel	0	0	0B	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC-2: -3 - +3 mV/V Fast	35	23	0C	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC-2: -3 - +3 mV/V Slow	37	25	0C	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC-2: -4 - +4 mV/V Fast	34	22	0C	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC-2: -4 - +4 mV/V Slow	36	24	0C	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AILC-2: Filter of 1st channel	0	0	0C	2	Percent	-110.0	-100.0	100.0	110.0
SNAP-AIMA: -20 - +20 mA	64	40	64	2	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA: 0 - +20 mA	2	2	64	2	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA: 4 - +20 mA	3	3	64	2	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA2-i: -1 to +1 mA	85	55	27	2	mA	-1.1	-1.0	1.0	1.1
SNAP-AIMA-i: -20 - +20 mA	64	40	22	2	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-i: 0 - +20 mA	2	2	22	2	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-i: 4 - +20 mA	3	3	22	2	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA-iSRC: -20 - +20 mA	64	40	26	2	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-iSRC: 0 - +20 mA	2	2	26	2	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-iSRC: 4 - +20 mA	3	3	26	2	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA-iSRC-FM: -20 - +20 mA	64	40	26	2	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-iSRC-FM: 0 - +20 mA	2	2	26	2	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-iSRC-FM: 4 - +20 mA	3	3	26	2	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA-4: -20 - +20 mA	64	40	40	4	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-4: 0 - +20 mA	2	2	40	4	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-4: 4 - +20 mA	3	3	40	4	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA-8: -20 - +20 mA	64	40	4A	8	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-8: 0 - +20 mA	2	2	4A	8	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-8: 4 - +20 mA	3	3	4A	8	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA-32: -20 to +20 mA	64	40	4D	32	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-32: 0 - +20 mA	2	2	4D	32	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-32: 4 - +20 mA	3	3	4D	32	mA	-22.0	4.0	20.0	22.0
SNAP-AIMA-32-FM: -20 to +20 mA	64	40	4D	32	mA	-22.0	-20.0	20.0	22.0
SNAP-AIMA-32-FM: 0 - +20 mA	2	2	4D	32	mA	-22.0	0.0	20.0	22.0
SNAP-AIMA-32-FM: 4 - +20 mA	3	3	4D	32	mA	-22.0	4.0	20.0	22.0
SNAP-AIMV-4: -150 - +150 mV	66	42	44	4	mV	-165.0	-150.0	150.0	165.0
SNAP-AIMV-4: -75 - +75 mV	68	44	44	4	mV	-82.5	-75.0	75.0	82.5

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low Scale	Full Scale	Overrange
SNAP-AIMV2-4: -50 - +50 mV	9	9	45	4	mV	-55.0	-50.0	50.0	55.0
SNAP-AIMV2-4: -25 - +25 mV	67	43	45	4	mV	-27.5	-25.0	25.0	27.5
SNAP-AIPM (point 0 only)	70	46	0A	*	AC VRMS	0.0	0	250	275
SNAP-AIPM (point 1 only)	71	47	0A	*	AC ARMS	0.0	0	10	11.0
SNAP-AIPM (point 2 only)	82	52	0A	*	True power	n/a	n/a	n/a	n/a
SNAP-AIPM (point 3 only)	83	53	0A	*	Volt/Amps	n/a	n/a	n/a	n/a
SNAP-AIPM-3 (points 0, 4, & 8)	70	46	49	*	AC VRMS	0.0	0	300	330
SNAP-AIPM-3 (points 1, 5, & 9)	71	47	49	*	AC ARMS	0.0	0	5	5.5
SNAP-AIPM-3 (points 2, 6, & 10)	82	52	49	*	True power	n/a	n/a	n/a	n/a
SNAP-AIPM-3 (points 3, 7, & 11)	83	53	49	*	Volt/Amps	n/a	n/a	n/a	n/a
SNAP-AIPM-3 (points 12 & 13)	86	56	49	*	True power	n/a	n/a	n/a	n/a
SNAP-AIPM-3V (points 0, 4, & 8)	70	46	48	*	AC VRMS	0.0	0	300	330
SNAP-AIPM-3V (points 1, 5, & 9)	71	47	48	*	VAC from CT	0.0	0	0.333	0.366
SNAP-AIPM-3V (points 2, 6, & 10)	82	52	48	*	True power	n/a	n/a	n/a	n/a
SNAP-AIPM-3V (points 3, 7, & 11)	83	53	48	*	Volt/Amps	n/a	n/a	n/a	n/a
SNAP-AIPM-3V (points 12 & 13)	86	56	48	*	True power	n/a	n/a	n/a	n/a
SNAP-AIRATE: Rate (Frequency)	69	45	69	2	Hz	0.0	0.0	25000.0	27500.0
SNAP-AIRATE-HFi: Rate (0.1 s data freshness)	68	44	2B	2	Hz	2	2	500,000	500,000
SNAP-AIRATE-HFi: Rate (1 s data freshness)	69	45	2B	2	Hz	20	20	500,000	500,000
SNAP-AIRTD: 100 Ohm Pt 3-wire	10	0A	10	2	Degrees C	-200.0	-200.0	850.0	850.0
SNAP-AIRTD: 100 Ohm Ni 3-wire	46	2E	10	2	Degrees C	-60.0	-60.0	250.0	250.0
SNAP-AIRTD: 120 Ohm Ni 3-wire	48	30	10	2	Degrees C	-80.0	-80.0	260.0	260.0
SNAP-AIRTD-10: 10 Ohm Cu 3-wire	14	0E	0E	2	Degrees C	-180.0	-180.0	260.0	260.0
SNAP-AIRTD-1K: 1000 Ohm Pt 3-wire	92	5C	0F	2	Degrees C	-200.0	-200.0	850.0	850.0
SNAP-AIRTD-1K: 1000 Ohm Ni 3-wire	93	5D	0F	2	Degrees C	-60.0	-60.0	250.0	250.0
SNAP-AIRTD-1K: 1000 Ohm Ni 3-wire	94	5E	0F	2	Degrees F	-50.0	-50.0	275.0	275.0
SNAP-AITM: -150 - +150 mV	66	42	66	2	mV	-165.0	-150.0	150.0	165.0
SNAP-AITM: -75 - +75 mV	68	44	66	2	mV	-82.5	-75.0	75.0	82.5
SNAP-AITM: Type E Thermocouple	19	13	66	2	Degrees C	-270.0	-270.0	1000.0	1000.0
SNAP-AITM: Type J Thermocouple	5	5	66	2	Degrees C	-210.0	-210.0	1200.0	1200.0
SNAP-AITM: Type K Thermocouple	8	8	66	2	Degrees C	-270.0	-270.0	1372.0	1372.0
SNAP-AITM-i: -150 - +150 mV	66	42	20	2	mV	-165.0	-150.0	150.0	165.0
SNAP-AITM-i: -75 - +75 mV	68	44	20	2	mV	-82.5	-75.0	75.0	82.5
SNAP-AITM-i: Type E Thermocouple	19	13	20	2	Degrees C	-270.0	-270.0	1000.0	1000.0
SNAP-AITM-i: Type J Thermocouple	5	5	20	2	Degrees C	-210.0	-210.0	1200.0	1200.0
SNAP-AITM-i: Type K Thermocouple	8	8	20	2	Degrees C	-270.0	-270.0	1372.0	1372.0

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low Scale	Full Scale	Overrange
SNAP-AITM-4i: -150 - +150 mV	66	42	32	4	mV	-165.0	-150.0	150.0	165.0
SNAP-AITM-4i: -75 - +75 mV	68	44	32	4	mV	-82.5	-75.0	75	82.5
SNAP-AITM-4i: -50 - +50 mV	9	9	32	4	mV	-55.0	-50.0	50.0	55.0
SNAP-AITM-4i: -25 - +25 mV	67	43	32	4	mV	-27.5	-25.0	25.0	27.5
SNAP-AITM-4i: Type B Thermocouple	24	18	32	4	Degrees C	42.0	42.0	1820.0	1820.0
SNAP-AITM-4i: Type C Thermocouple	32	20	32	4	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-4i: Type D Thermocouple	33	21	32	4	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-4i: Type E Thermocouple	19	13	32	4	Degrees C	-270.0	-270.0	1000.0	1000.0
SNAP-AITM-4i: Type G Thermocouple	31	1F	32	4	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-4i: Type J Thermocouple	5	5	32	4	Degrees C	-210.0	-210.0	1200.0	1200.0
SNAP-AITM-4i: Type K Thermocouple	8	8	32	4	Degrees C	-270.0	-270.0	1372.0	1372.0
SNAP-AITM-4i: Type N Thermocouple	30	1E	32	4	Degrees C	-270.0	-270.0	1300.0	1300.0
SNAP-AITM-4i: Type R Thermocouple	17	11	32	4	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM-4i: Type S Thermocouple	23	17	32	4	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM-4i: Type T Thermocouple	18	12	32	4	Degrees C	-270.0	-270.0	400.0	400.0
SNAP-AITM-8: -75 - +75 mV	68	44	4F	8	mV	-82.5	-75.0	75.0	82.5
SNAP-AITM-8: -50 - +50 mV	9	9	4F	8	mV	-55.0	-50.0	50.0	55.0
SNAP-AITM-8: -25 - +25 mV	67	43	4F	8	mV	-27.5	-25.0	25.0	27.5
SNAP-AITM-8: Type B Thermocouple	24	18	4F	8	Degrees C	42.0	42.0	1820.0	1820.0
SNAP-AITM-8: Type C Thermocouple	32	20	4F	8	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-8: Type D Thermocouple	33	21	4F	8	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-8: Type E Thermocouple	19	13	4F	8	Degrees C	-270.0	-270.0	1000.0	1000.0
SNAP-AITM-8: Type G Thermocouple	31	1F	4F	8	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-8: Type J Thermocouple	5	5	4F	8	Degrees C	-210.0	-210.0	1200.0	1200.0
SNAP-AITM-8: Type K Thermocouple	8	8	4F	8	Degrees C	-270.0	-270.0	1372.0	1372.0
SNAP-AITM-8: Type N Thermocouple	30	1E	4F	8	Degrees C	-270.0	-270.0	1300.0	1300.0
SNAP-AITM-8: Type R Thermocouple	17	11	4F	8	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM-8: Type S Thermocouple	23	17	4F	8	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM-8: Type T Thermocouple	18	12	4F	8	Degrees C	-270.0	-270.0	400.0	400.0
SNAP-AITM-8-FM: -75 - +75 mV	68	44	4F	8	mV	-82.5	-75.0	75.0	82.5
SNAP-AITM-8-FM: -50 - +50 mV	9	9	4F	8	mV	-55.0	-50.0	50.0	55.0
SNAP-AITM-8-FM: -25 - +25 mV	67	43	4F	8	mV	-27.5	-25.0	25.0	27.5
SNAP-AITM-8-FM: Type B Thermocple	24	18	4F	8	Degrees C	42.0	42.0	1820.0	1820.0
SNAP-AITM-8-FM: Type C Thermocple	32	20	4F	8	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-8-FM: Type D Thermocple	33	21	4F	8	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM-8-FM: Type E Thermocple	19	13	4F	8	Degrees C	-270.0	-270.0	1000.0	1000.0
SNAP-AITM-8-FM: Type G Thermocple	31	1F	4F	8	Degrees C	0.0	0.0	2320.0	2320.0

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low Scale	Full Scale	Overrange
SNAP-AITM-8-FM: Type J Thermocple	5	5	4F	8	Degrees C	-210.0	-210.0	1200.0	1200.0
SNAP-AITM-8-FM: Type K Thermocple	8	8	4F	8	Degrees C	-270.0	-270.0	1372.0	1372.0
SNAP-AITM-8-FM: Type N Thermocple	30	1E	4F	8	Degrees C	-270.0	-270.0	1300.0	1300.0
SNAP-AITM-8-FM: Type R Thermocple	17	11	4F	8	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM-8-FM: Type S Thermocple	23	17	4F	8	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM-8-FM: Type T Thermocple	18	12	4F	8	Degrees C	-270.0	-270.0	400.0	400.0
SNAP-AITM2: -50 - +50 mV	9	9	09	2	mV	-55.0	-50.0	50.0	55.0
SNAP-AITM2: -25 - +25 mV	67	43	09	2	mV	-27.5	-25.0	25.0	27.5
SNAP-AITM2: Type B Thermocouple	24	18	09	2	Degrees C	42.0	42.0	1820.0	1820.0
SNAP-AITM2: Type C Thermocouple	32	20	09	2	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM2: Type D Thermocouple	33	21	09	2	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM2: Type G Thermocouple	31	1F	09	2	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM2: Type N Thermocouple	30	1E	09	2	Degrees C	-270.0	-270.0	1300.0	1300.0
SNAP-AITM2: Type R Thermocouple	17	11	09	2	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM2: Type S Thermocouple	23	17	09	2	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM2: Type T Thermocouple	18	12	09	2	Degrees C	-270.0	-270.0	400.0	400.0
SNAP-AITM2-i: -50 - +50 mV	9	9	21	2	mV	-55.0	-50.0	50.0	55.0
SNAP-AITM2-i: -25 - +25 mV	67	43	21	2	mV	-27.5	-25.0	25.0	27.5
SNAP-AITM2-i: Type B Thermocouple	24	18	21	2	Degrees C	42.0	42.0	1820.0	1820.0
SNAP-AITM2-i: Type C Thermocouple	32	20	21	2	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM2-i: Type D Thermocouple	33	21	21	2	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM2-i: Type G Thermocouple	31	1F	21	2	Degrees C	0.0	0.0	2320.0	2320.0
SNAP-AITM2-i: Type N Thermocouple	30	1E	21	2	Degrees C	-270.0	-270.0	1300.0	1300.0
SNAP-AITM2-i: Type R Thermocouple	17	11	21	2	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM2-i: Type S Thermocouple	23	17	21	2	Degrees C	-50.0	-50.0	1768.0	1768.0
SNAP-AITM2-i: Type T Thermocouple	18	12	21	2	Degrees C	-270.0	-270.0	400.0	400.0
SNAP-AIV: -10 - +10 VDC	12	C	12	2	VDC	-11.0	-10.0	10.0	11.0
SNAP-AIV: -5 - +5 VDC	11	B	12	2	VDC	-5.5	-5.0	5.0	5.5
SNAP-AIV-i: -10 - +10 VDC	12	C	23	2	VDC	-11.0	-10.0	10.0	11.0
SNAP-AIV-i: -5 - +5 VDC	11	B	23	2	VDC	-5.5	-5.0	5.0	5.5
SNAP-AIV-4: -10 - +10 VDC	12	C	41	4	VDC	-11.0	-10.0	10.0	11.0
SNAP-AIV-4: -5 - +5 VDC	11	B	41	4	VDC	-5.5	-5.0	5.0	5.5
SNAP-AIV-8: -10 - +10 VDC	12	C	4B	8	VDC	-11.0	-10.0	10.0	11.0
SNAP-AIV-8: -5 - +5 VDC	11	B	4B	8	VDC	-5.5	-5.0	5.0	5.5
SNAP-AIV-32: -10 - +10 VDC	12	C	4E	32	VDC	-11.0	-10.0	10.0	11.0
SNAP-AIV-32: -5 - +5 VDC	11	B	4E	32	VDC	-5.5	-5.0	5.0	5.5

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low Scale	Full Scale	Overrange
SNAP-AIV-32-FM: -10 - +10 VDC	12	C	4E	32	VDC	-11.0	-10.0	10.0	11.0
SNAP-AIV-32-FM: -5 - +5 VDC	11	B	4E	32	VDC	-5.5	-5.0	5.0	5.5
SNAP-AIV2-i: -100 - +100 VDC	72	48	24	2	VDC	-110.0	-100.0	100.0	110.0
SNAP-AIV2-i: -50 - +50 VDC	73	49	24	2	VDC	-55.0	-50.0	50.0	55.0
SNAP-AIVRMS: 0 - 250 VAC/VDC	70	46	70	2	VAC/VDC	0.0	0.0	250.0	275.0
SNAP-AIVRMS-i: 0 - 250 VAC/VDC	70	46	28	2	VAC/VDC	0.0	0.0	250.0	275.0
SNAP-AIVRMS-i-FM: 0–250 VAC/VDC	70	46	28	2	VAC/VDC	0.0	0.0	250.0	275.0
SNAP-AIR40K-4: 0 to 40K Ohms	74	4A	43	4	Ohms	0	0	40000.0	44000.0
SNAP-AIR40K-4: 0 to 20K Ohms	75	4B	43	4	Ohms	0	0	20000.0	22000.0
SNAP-AIR40K-4: 0 to 10K Ohms	76	4C	43	4	Ohms	0	0	10000.0	11000.0
SNAP-AIR40K-4: 0 to 5K Ohms	77	4D	43	4	Ohms	0	0	5000.0	5500.0
SNAP-pH/ORP: -1 - +1 VDC	78	4E	25	2	VDC	-1.1	-1.0	1.0	1.1
SNAP-pH/ORP: 0 - 14 pH	79	4F	25	2	pH	-1.4	0.0	14.0	15.4
SNAP-pH/ORP: -0.5 - +0.5 VDC	80	50	25	2	VDC	-0.55	-0.5	0.5	0.55
SNAP-PID-V	99	63	D0	4	Percent	0	0	100.0	110.0

* The SNAP-AIPM module monitors one device from point 0 (volts) and point 1 (amps). Points 2 and 3 return calculated values. The SNAP-AIPM-3 and SNAP-AIPM-3V monitor three phases from points 0,4, & 8 (volts) and points 1,5, & 9 (amps). All other points return calculated values. See form #1453, the *SNAP AIPM Modules Data Sheet*, for details.

Analog Output Modules

NOTE: For E1 and E2 brain boards, see form #1576, I/O Configuration for E1 and E2 Brain Boards.

Use this data for configuring point types and features (see page 22).

Part Number & Description	Point Type (Dec)	Point Type (Hex)	Module Type (Hex)	Points per Module	Default Unit of Measurement	Underrange	Low scale	Full scale	Overrange
SNAP-AOA-3: 4 - 20 mA	131	83	83	1	mA	4.0	4.0	20.0	20.0
SNAP-AOV-5: 0 - 10 VDC	133	85	85	1	VDC	0.0	0.0	10.0	10.0
SNAP-AOA-23: 4 - 20 mA	163	A3	A3	2	mA	4.0	4.0	20.0	20.0
SNAP-AOA-23-iSRC: 4 - 20 mA	163	A3	B3	2	mA	4.0	4.0	20.0	20.0
SNAP-AOA-23-iSRC-FM: 4 - 20 mA	163	A3	B3	2	mA	4.0	4.0	20.0	20.0
SNAP-AOV-25: 0 - 10 VDC	165	A5	A5	2	VDC	0.0	0.0	10.0	10.0
SNAP-AOV-27: -10 - +10 VDC	167	A7	A7	2	VDC	-10.0	-10.0	10.0	10.0
SNAP-AOA-28: 0 - 20 mA	168	A8	A8	2	mA	0.0	0.0	20.0	20.0
SNAP-AOD-29: TPO 5 - 60 VDC	169	A9	A9	2	percent	n/a	0.0	100.0	n/a
SNAP-AOD-29-HFi: TPO 2.5-24 VDC	131	83	B9	2	percent	n/a	0.0	100.0	n/a

Using I/O Point Features

The I/O point features available on I/O units depend on the combined capabilities of the I/O processor, the module, and in some cases, the protocol used. See page 9 to determine which features are available for the processor you are using.

See the referenced pages for more information on using them in your application.

Feature	Description	See
States	(digital inputs and outputs)—A digital point is either on or off. You can read the current state of a digital input or write an on/off state to a digital output.	page 31
Latches	(digital inputs)—When the value of a digital input point changes from off to on, an on-latch is automatically set. While the value of the point may return to off, the on-latch remains set, as a record of the change, until you clear it. Similarly, an off-latch is set when the value of a digital point changes from on to off, and it remains set until cleared.	page 31
Counters	(digital inputs)—A counter keeps track of the number of times a digital input changes from off to on. The count accumulates until it reaches the maximum count available in the I/O unit or until you reset the counter to zero. For example, to count the number of widgets produced per shift, you would clear the counter at the start of each shift and read it at the end of each shift. The speed of the counter depends upon the I/O processor's capabilities and the speed of the module used.	page 31

Feature	Description	See
Quadrature counters	(digital input)—A quadrature counter requires a SNAP quadrature input module, which is attached to the encoder device. The module sends a pulse to the I/O unit upon each change in quadrature state, and the I/O unit counts the pulses and keeps track of the direction and rotation.	page 32
Frequency measurement	(digital input)—Frequency is the speed with which a digital point changes state and is usually measured in counts per second. For example, reading the frequency can help you determine the speed of rotating machinery. Frequency measurement can be one-time or continuous.	page 33
Period measurement	(digital input)—Period refers to the elapsed time for a complete on-off-on transition on a digital point. Measurement starts on the first transition (either off-to-on or on-to-off) and stops on the next transition of the same type. Period measurement can be one-time or continuous.	page 33
Pulse measurement	(digital input)—Measures the duration of a pulse, either an on-pulse or an off-pulse.	page 33
Digital Totalizer	(digital input)—A digital totalizer accumulates the total amount of time that a digital input is on (or off). The on-time totalizer shows how long the point has been on; the off-time totalizer shows how long the point has been off. Totalizers are often used to determine maintenance or use schedules.	page 34
Watchdog	(digital and analog points)—A watchdog monitors communication on the OptoMMP port (port 2001, unless you have changed it). If nothing accesses the port for the length of time set in the watchdog, the I/O unit automatically sets designated digital and analog I/O points to the values you have determined. A watchdog helps make sure that a communication failure doesn't result in disaster. If communication fails between the host and the I/O unit controlling a process, the watchdog makes sure the process is automatically brought to a safe state. For example, a valve could automatically close to avoid completely emptying a tank.	page 34
Scaling	(analog points)—Analog input and output points can be scaled as needed. For example, you can scale a -5 V to +5 V input point to reflect 0% to 100%	page 34
Minimum and maximum values	(analog inputs)—Minimum and maximum values are sometimes called peaks and valleys. You can read these values at any time, for example, to record minimum and maximum temperatures. You can also reset min/max values. For example, if you want to record the maximum temperature at point 2 in each 24-hour period, you must reset the values after they are read each day.	page 35
Offset and gain	(analog inputs) Offset and gain calculations are used to calibrate analog points. If a -50 mV to +50 mV input receives signals that are slightly off (not exactly -50 mV at the lowest point, for example), the offset and gain can be calculated so that values will appear accurately when read.	page 35
Clamping	(analog outputs)—Clamping limits values that can be sent to analog output points so they do not go above or below a specific value. For example, if you are using a 0–10 VDC output module, but the device attached to one of its points can only handle a maximum of 5 VDC, you can set an upper clamp of 5 VDC for that point. The values for upper and lower clamp are set in engineering units.	page 35
Average filter weight	(analog inputs)—A filter weight smooths analog input signals that are erratic or change suddenly.	page 36

I/O point features are discussed in the following sections.

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

States (Digital Points)

You can read the ON or OFF state of a digital input point or write to a digital output point to turn it on or off. This feature is automatic and needs no configuration.

For E1 brain boards, each point on the unit is treated like the first point on a SNAP module; that is, only the first of every four points contains data. For more information on interpreting data formats, see page 58.

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Latches (Digital Points)

SNAP and G4EB2—Latching is available on both 4-channel and high-density digital points. It is automatic and needs no configuration. Using memory map values, you can read the on-latch or off-latch state of a digital point, and you can clear latches.

E1—Latching is available on all modules used with the E1. Note that latching is different on an E1 depending on the protocol used with the brain board. When the E1 is used with the Optomux protocol, only one latch is available and you must configure it to be an on-to-off latch or an off-to-on latch. When you use an E1 with PAC Control or OptoMMP, however, both types of latches are automatically available for each point, and no configuration is required.

To read and/or clear latches, remember that each point on an E1 is treated like the first point on a SNAP module, and each point on a G4EB2 falls within the first four points of a SNAP module. For example, to read latches for E1 or G4EB2 points using the Digital Point Read area of the memory map, read the following addresses:

This module position on E1:	0	1	2	3
Or this point on G4EB2:	0	4	8	12
Is like this module,point on SNAP:	0,0	1,0	2,0	3,0
Address for on-latch state:	F080 0004	F080 0104	F080 0204	F080 0304
Address for off-latch state:	F080 0008	F080 0108	F080 0208	F080 0308

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Counters (Digital Points)

SNAP—Any SNAP digital input can be used as a counter. Note the differences in counting between 4-channel and high-density digital modules:

	SNAP 4-channel digital counters	SNAP high-density digital counters
Processor compatibility	SNAP-PAC-R1 (-FM, -W) SNAP-PAC-EB1 (-FM, -W) SNAP-PAC-SB1 SNAP-B3000-ENET SNAP-ENET-RTC SNAP-UP1-ADS	SNAP-PAC-R1 (-FM, -W) SNAP-PAC-R2 (-FM, -W) SNAP-PAC-EB1 (-FM, -W) SNAP-PAC-EB2 (-FM, -W) SNAP-PAC-SB1 SNAP-PAC-SB2 SNAP-B3000-ENET SNAP-ENET-RTC SNAP-ENET-S64 SNAP-UP1-ADS SNAP-UP1-M64
Counting is done on...	...the brain	...the module

	SNAP 4-channel digital counters	SNAP high-density digital counters
Counting speed	High speed (depends on speed of module; modules available up to 20 KHz)	Low speed (up to 50 Hz)
Configuration and Use	• Each point to be used as a counter must be configured. • Counters start as soon as configured. • Counters can be Started, Stopped, Read, and Read & Cleared.	• Configure points only if using PAC Control. • Counters are always counting. • Counters can be Read or Read & Cleared. • Counters cannot be Started or Stopped.

G4EB2—Any digital input can be used as a counter. Counters must be configured. They start as soon as they are configured and can be Started, Stopped, Read, and Read & Cleared.

E1—Any digital input can be used as a counter. Follow the steps in form #1576, *I/O Configuration for E1 and E2 Brain Boards*, to configure counters before using them.

Use memory map values in the Point Configuration area to work with counters.

PAC-R
EB
SB
UIO
EIO

Quadrature Counters (Digital Inputs)

I/O units with the following I/O processors support quadrature counters for quadrature encoder devices:

- SNAP-PAC-R1
- SNAP-PAC-EB1
- SNAP-PAC-SB1
- SNAP-UP1-ADS
- SNAP-B3000-ENET
- SNAP-ENET-RTC

A quadrature counter requires a SNAP quadrature input module (SNAP-IDC5Q), which is attached to the encoder device. The module sends a pulse to the processor upon each change in quadrature state, and the processor counts the pulses and keeps track of the direction and rotation. For each axis, the counter counts up if Phase A leads Phase B; it counts down if Phase A lags behind Phase B. Each axis can have counts from 0 to 2,147,483,647.

If your encoder device has an index feature, you can use two separate digital input points as indexes, one for each axis. The index automatically resets the count, and it shows what the count was when the index was triggered. Counts are sometimes lost, due to noise or encoder problems, for example; with the index, you can see whether the count varies too much.

See form #1823, the *Using Quadrature Counters* technical note, for details on programming quadrature counters.

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Pulse Measurement (Digital Inputs)

You can measure the duration of a pulse on any digital input. Pulse measurement must be configured by writing to the Point Feature memory map address as follows (for example, for module 0 point 0, you would write one of these values to address F010 0008):

- On-pulse: 0x00000009
- Off-pulse: 0x0000000A

Measurement begins at the next leading edge and ends at the following trailing edge. When the measurement is complete, the feature number is cleared, the counter stops, and a completion bit is set for the point (for example, for module 0 point 0, in address F040 0024). When the Point Feature is reset, the completion bit is cleared.

Read the pulse measurement in the Feature Value field (see "Digital Point Read—Read Only" on page 149)—for example, for module 0 point 0, in address F080 0010. This value is a 32-bit unsigned integer. Units and resolution are in increments of 100 µsec (0.1 msec or 0.0001 seconds). For example, a 60 Hz frequency can be counted with a resolution of 166.6, which is calculated as follows:

```
Resolution = 1/(Frequency * Resolution) = 1/(60 * 0.0001) = 166.6
```

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Frequency or Period Measurement (Digital Inputs)

You can measure frequency or period, either one-time or continuously, on digital inputs. You must configure frequency or period measurement for each point.

After you've configured the point feature, you can read the frequency or period measurement in the Feature Value field (see "Digital Point Read—Read Only" on page 149)—for example, for module 0 point 0, in address F080 0010. This value is a 32-bit unsigned integer. Units and resolution are in increments of 100 µsec. For example, a 60 Hz frequency can be counted with a resolution of 166.6, which is calculated as follows:

```
Resolution = 1/(Frequency * Resolution) = 1/(60 * 0.0001) = 166.6
```

One-Time Measurement

Write to the Point Feature memory map address as follows (for example, for module 0 point 0, you would write one of these values to address F010 0008):

- For one-time period measurement: 0x0000000B
- For one-time frequency measurement: 0x0000000C

When the measurement is complete, the feature number is cleared, the counter stops, and a completion bit is set for the point (for example, for module 0 point 0, in address F040 0024). When the Point Feature is reset, the completion bit is cleared.

Continuous Measurement

Write to the Point Feature memory map address as follows (for example, for module 0 point 0, you would write one of these values to address F010 0008):

- For continuous period measurement: 0x00000003
- For continuous frequency measurement: 0x00000005

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Digital Totalizer

A digital totalizer can be configured for any digital point. Write to the Point Feature memory map address as follows (for example, for module 0 point 0, you would write one of these values to address F010 0008):

- For on-time totalizer: 0x00000002
- For off-time totalizer: 0x00000012

Read the totalizer value in the Feature Value field (see "Digital Point Read—Read Only" on page 149)—for example, for module 0 point 0, in address F080 0010.

PAC-R
EB
SB
UIO
EIO
SIO
E1
E2
G4EB2

Watchdog (Digital and Analog Points)

To configure a watchdog, set the watchdog when configuring the *I/O unit*. Then when you configure a digitalor analog output point, you can choose the status or value the point should be set to if the watchdog timer expires. Some older I/O units do not include watchdogs on high-density digital points. See the comparison chart for details.

For E1 and E2 I/O units, follow the steps in form #1576, *I/O Configuration for E1 and E2 Brain Boards*.

PAC-R
EB
SB
UIO
EIO
SIO
E2

Scaling (Analog Points)

You can scale analog input or output points to match your needs. For example, you can scale a -5 V to +5 V input point to reflect 0% to 100%. Point types may be unipolar or bipolar.

Examples of Unipolar Points	Examples of Bipolar Points
4–20 mA analog output	-25 mV to +25 mV analog input
0–10 A RMS analog input	-10 to +10 VDC analog output

Unipolar and bipolar points are scaled in the same way, with the lowest reading reflecting the low scale and the highest reading reflecting the high scale. Here are two examples:

	Unipolar Input Point		Bipolar Input Point		
	Low scale	High scale	Low scale		High scale
Actual reading	0 mA	20 mA	-5 V	0 V	+5 V
Scaled for percentage	0%	100%	0%	50%	100%
Scaled for counts*	0	+25,000	-25,000	0	+25,000

*Counts for input points always range -25,000 to +25,000.

	Unipolar Output Point		Bipolar Output Point		
	Low scale	High scale	Low scale		High scale
Actual reading	4 mA	20 mA	-10 VDC	0 VDC	+10 VDC
Scaled for percentage	0%	100%	0%	50%	100%
Scaled for counts*	0	4,095	0	2,047.5	4,095

*Counts for output points always range 0–4,095.

NOTE: With firmware version 8.1 and higher, you can also use inverted scaling. For example:

0 mA 20 mA
742 fpm -27 fpm

To scale an analog point, see the instructions in the *PAC Manager User's Guide*. EXCEPTION: For an E2 I/O unit, follow the steps in form #1576, *I/O Configuration for E1 and E2 Brain Boards*.

PAC-R EB SB UIO EIO SIO E2

Minimum and Maximum Values (Analog Points)

All memory-mapped I/O units with analog capability automatically keep track of minimum and maximum values on analog points. You can read these values using the memory map (see "(Expanded) Analog Point Read—Read" on page 116. You can also read and clear them at the same time (see "(Expanded) Analog Point Read & Clear—Read/Write" on page 115).

PAC-R EB SB UIO EIO SIO E2

Offset and Gain (Analog Points)

Memory-mapped I/O units with analog capability can calculate offset and gain for analog input points. Calculate offset first, and then calculate gain. See the *PAC Manager User's Guide* for instructions.

NOTE: If you are using Modbus/TCP, you will need to calculate the offset and gain yourself. Then you can write offset and gain values to the I/O unit. See the PAC Manager User's Guide for more information.

PAC-R EB SB UIO EIO SIO E2

Clamping (Analog Points)

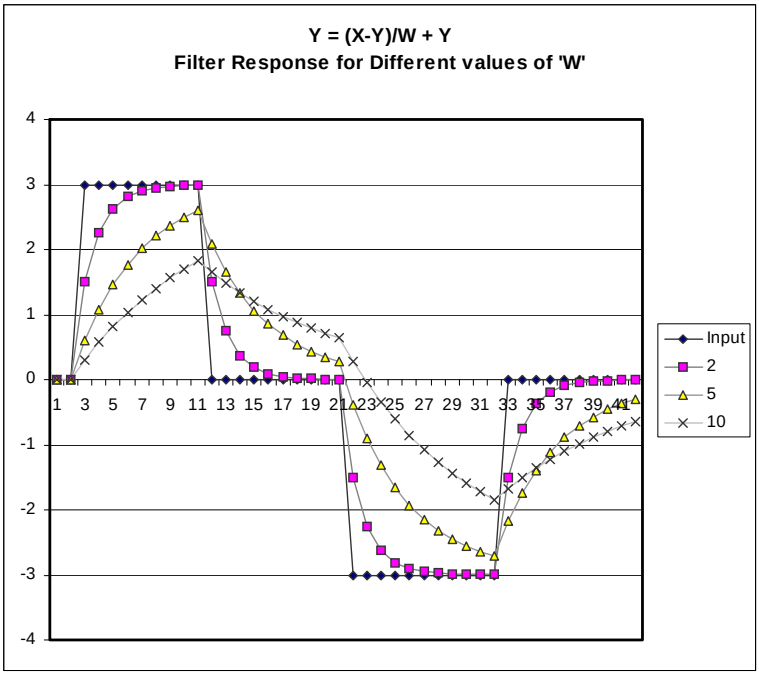
SNAP Ethernet-based I/O units with analog capability can clamp values sent to analog output points so they do not go above or below a specific limit.

For example, if you are using a 0–10 VDC output module, but the device attached to one of its points can only handle a maximum of 5 VDC, you can set an upper clamp of 5 VDC for that point. The values for upper and lower clamp are set in engineering units. Set upper and lower clamps when configuring the point.

PAC-R
EB
SB
UIO
EIO
SIO

Average Filter Weight (Analog Points)

SNAP Ethernet-based I/O units can use a filter weight to smooth analog input signals that are erratic or change suddenly. The formula used for filtering is Y = (X - Y)/W + Y, where Y is the filtered value, X is the new unfiltered value, and W is the filter weight. The following graph shows the effect of filter weights (W) 2, 5, and 10 on a step input signal:



As this graph shows, the larger the filter weight (W) you use, the smoother the analog signal.

A filter weight of zero turns off the calculation. Values less than or equal to 0.5 are changed to zero, since those values would cause an unstable signal.

Filtering is applied to values that are in engineering units, including minimum and maximum values. Filtering does not apply to values that are in counts. Set filter weight when configuring the analog point.

Using Event/Reactions

PAC-R
EB
SB
UIO
EIO

Event/reactions are available on SNAP PAC R-series, SNAP PAC EB and SB brains, SNAP Ultimate, and SNAP Ethernet I/O units. SNAP Simple I/O units and E1 and E2 brain boards do not have event/reaction capability.

CAUTION: Event/reactions occur on the I/O side of a SNAP PAC R-series or SNAP Ultimate controller, independently of any PAC Control strategy running on the control side. If you are using PAC Control, it is best to use flowchart logic to handle reactions to events. If you do set up event/reactions, be very careful that they do not conflict with PAC Control logic.

PAC-R
PAC-S
EB
SB
UIO
EIO
LCE

Understanding the Scratch Pad

SNAP PAC S-series and SNAP-LCE standalone controllers, SNAP PAC R-series and SNAP Ultimate on-the-rack controllers, and SNAP PAC and SNAP Ethernet brains contain Scratch Pad areas within their memory maps. (SNAP Simple brains and E1 and E2 brain boards do not contain a Scratch Pad.) Scratch Pad areas can be used for two main purposes:

- as a place to hold data being transferred from one peer to another on the network (SNAP PAC S-series and R-series, SNAP-LCE, and SNAP Ultimate only)
- as a virtual notebook for keeping track of events and alarms (SNAP PAC R-series, SNAP PAC brains, SNAP Ultimate, and SNAP Ethernet only)

The Scratch Pad is user-defined, meaning that you define and use its addresses to fit your needs, and you can redefine them whenever necessary. The Scratch Pad area includes up to four sections, depending on device type, to accommodate different types of data: bits, strings, floats, and integers.

- The Scratch Pad bits section is a 64-bit mask.
- The Scratch Pad strings section is a table of 64 elements. Each element can hold 128 characters or 128 bytes of binary data.
- The Scratch Pad float section is a table of 10,240 elements; each float is four bytes.
- The Scratch Pad 32-bit integer section is also a table of 10,240 four-byte elements.
- The Scratch Pad 64-bit integer section is a table of 1024 eight-byte elements.

NOTE: Scratch Pad float and 32-bit integer tables are not made up of contiguous addresses in the memory map; each table is in two address sections. You won't notice this if you are using PAC Control Scratch Pad commands, but if you are addressing these tables in another application, check the memory map appendix to make sure you have the correct addresses for the table elements you want.

For more information on using the Scratch Pad for peer-to-peer data transfer, see "I/O Unit—Scratch Pad Commands" in Chapter 10 of the *PAC Control User's Guide*. (You can also use PAC Manager for one-time reads and writes.) The rest of this chapter shows you how to use the Scratch Pad for tracking events and alarms.

Scratch Pad strings, floats, and integers are available for SNAP PAC R-series, S-series, and SNAP Ultimate I/O and are primarily used to transfer data from one peer to another on the network. For more information on using the Scratch Pad in this way, see "I/O Units—Scratch Pad Commands" in Chapter 10 of the *PAC Control User's Guide*. (You can also use PAC Manager for one-time reads and writes.)

Scratch Pad bits are available for both standalone and on-the-rack controllers and for SNAP Ethernet I/O units. Controllers and Ultimate I/O units usually use them in the same way as strings, floats, and integers—they're just another data format—but in Ethernet I/O units, Scratch Pad bits are primarily used for tracking events and alarms.

PAC-R
EB
SB
UIO
EIO

Using Scratch Pad Bits for Events and Alarms

When Scratch Pad bits are used to track events and alarms, the 64 bits in the mask do not represent point numbers. Instead, they represent whatever you decide they should be. For example, you might decide that bit 1 in the Scratch Pad will indicate a temperature level in Vat #12 (if the temperature reaches 48 °C, bit 1 is turned on). Bit 2 might indicate the status of Pump A (if the pump is off, the bit is off; if the pump is on, the bit is on).

Because you can use Scratch Pad bits to keep track of events and alarms, you can set up reactions based on a variety of conditions. In the example above, you could set up a reaction on an EB brain that sends a stream packet if bit 1 is on and bit 2 is off.

Cascading Events, Alarms, and Reactions

Scratch Pad bits are really a way to set up cascading events and reactions (that is, a series of events and reactions dependent on each other). For example, the first event in the cascade could be the temperature in Vat #12 reaching 40 degrees, and the reaction to it is setting Scratch Pad bit 1. The second event in the cascade is that Scratch Pad bit #1 is set, and the reaction to that is some other action. A cascade of any number of events and reactions can be configured, as needed.

PAC-R
EB
SB
UIO
EIO

Types of Events, Alarms, and Reactions

NOTE: SB brains do not support serial events and reactions nor reactions requiring an Ethernet network, such as sending email.

Effect of Firmware on Events and Reactions

The following table shows the types of events and reactions available, depending on your processor and the firmware version you are using. The event or reaction can consist of one or a combination of the following. The reaction can take place immediately or after a delay.

	Firmware ≥ 8.1		Firmware ≤ 8.0	
	PAC-R, EB	SB	PAC-R, EB	UIO, EIO
Events				
On/off state of digital point on 4-channel module	●	●	●	●
State of on-latch or off-latch for digital point on 4-ch mod	●	●		
On/off state of digital point on HDD module	●	●		
State of on-latch or off-latch for digital point on HDD mod	●	●		
High or low value of analog point (in EU)	●	●	●	●
Number on a digital counter or high or low number on quadrature counter	●	●	●	●
Analog point value or quadrature counter that is outside allowable range	●	●	●	●
State of a bit in the Scratch Pad bits area	●	●	●	●
State of a bit in the Scratch Pad integer 64 area	●	●		
Specific string received by serial module	●		●	●
Reactions				
Turn on/off digital point on 4-channel module	●	●	●	●
Turn on/off digital point on HDD module	●	●		
Clear on-latch or off-latch on 4-channel or HDD module	●	●		
Copy data from one memmap location to another	●	●	●	●

	Firmware ≥ 8.1		Firmware ≤ 8.0	
	PAC-R, EB	SB	PAC-R, EB	UIO, EIO
Log data	●	●	●	●
Turn on or off a bit in the Scratch Pad bits area	●	●	●	●
Turn on or off a bit in the Scratch Pad integer 64 area	●	●		
Send stream packet	●		●	●
Send email message	●		●	●
Send string through a serial module to a serial device	●		●	●
Send SNMP trap	●		●	●

The following table shows the number and type of events available, depending on the processor and the firmware version.

Event Type	Firmware ≥ 8.1		Firmware ≤ 8.0	
	PAC-R, EB	SB	PAC-R, EB	UIO, EIO
Digital events–Expanded (formerly called Timers)	512	512	64	64
Digital events–Old	128	128	128	128
Alarm events	64	64	64	64
Serial events	32	n/a	32	32

Note that the memory map section formerly called Timers, which provided digital events with a delay between an event and the reaction to it, has been expanded in firmware 8.1 to include additional options such as latches and HDD modules. All new digital events should be configured in Digital Events - Expanded to take advantage of the new flexibility.

Digital events you already configured still exist in Digital Events - Old. Timers you already configured still exist in Digital Events - Expanded.

Steps for Configuring Events and Reactions—Firmware 8.1 and Higher

The following table shows steps you would use to configure possible events and reactions if you are using firmware 8.1 or higher with SNAP PAC I/O units. (For older firmware, see "Steps for Configuring Events and Reactions—Firmware 8.0 and Lower" on page 41.) Page numbers refer you to the memory map addresses in Appendix A that you would use for configuration.

See explanations starting on page 44 for important information you'll need to set up the different kinds of events and reactions.

Event	Reaction	Configuration Steps	See page
If digital point is on/off OR If on-latch or off-latch is set	Turn digital point on/off on same I/O unit *or* clear on-latch or off-latch on same I/O unit	Configure Expanded Digital Events, with or without delay	158
	Turn digital point on/off on different I/O unit *or* clear on-latch or off-latch on different I/O unit *or* log data *or* copy memory map data *or* send message(stream, email, serial, or SNMP trap).	1. Configure Expanded Digital Events—set Scratch Pad bit 2. (Email message only) Configure Email (Streaming only) Configure Streaming (SNMP only) Configure SNMP 3. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	158 175 146 141 172 187 175
If analog point value (Engineering Units) goes above or below a specified value OR If digital counter reaches a specified value	Turn digital point on/off on same I/O unit *or* clear on-latch or off-latch on same I/O unit	1. Configure Alarm Events (high alarm or low alarm)—set Scratch Pad bit 2. Configure Expanded Digital Events—turn on/off point or clear latch	171 158
	Turn digital point on/off on different I/O unit *or* clear on-latch or off-latch on different I/O unit *or* log data *or* copy memory map data *or* send message(stream, email, serial, or SNMP trap).	1. Configure Alarm Events (high alarm or low alarm)—set Scratch Pad bit 2. (Delay only) Configure Expanded Digital Events—set time delay and set a Scratch Pad bit after the delay 3. (Email message only) Configure Email (Streaming only) Configure Streaming (SNMP only) Configure SNMP 4. (Except data logging) Configure Event Messages—send message or data based on timer-expired bit (Data logging) Configure Data Logging based on timer-expired bit and configure Email (optional)	171 158 175 146 141 172 187 175
If analog point value (Engineering Units) or quadrature counter goes outside an allowable range	Turn digital point on/off on same I/O unit *or* clear on-latch or off-latch on same I/O unit	1. Configure Alarm Events (deviation alarm)—set Scratch Pad bit 2. Configure Expanded Digital Events—turn on/off point	171 158
	Turn digital point on/off on different I/O unit *or* clear on-latch or off-latch on different I/O unit *or* log data *or* copy memory map data *or* send message(stream, email, serial, or SNMP trap).	1. Configure Alarm Events (deviation alarm)—set Scratch Pad bit 2. (Delay only) Configure Expanded Digital Events—set time delay and set a Scratch Pad bit after the delay 3. (Email message only) Configure Email (Streaming only) Configure Streaming (SNMP only) Configure SNMP 4. (Except data logging) Configure Event Messages—send message or data based on timer-expired bit (Data logging) Configure Data Logging based on timer-expired bit and configure Email (optional)	171 158 175 146 141 172 187 175

Event	Reaction	Configuration Steps	See page
If a specific string is received by a serial module	Turn digital point on/off on same I/O unit *or* clear on-latch or off-latch on same I/O unit	1. Configure Serial Events—set Scratch Pad bit 2. Configure Expanded Digital Events—turn on/off point or clear latch	175 158
	Send SNMP trap	1. Configure Serial Events—set Scratch Pad bit 2. (Delay only) Configure Expanded Digital Events—set time delay and set a Scratch Pad bit after the delay 3. Configure SNMP 4. Configure Event Messages—send trap based on timer-expired bit	175 158 141 172
	Send one-time email	1. Configure Serial Events—send email 2. Configure Email	175 175
	Turn digital point on/off on different I/O unit *or* clear on-latch or off-latch on different I/O unit *or* log data *or* copy memory map data *or* send message(stream, email, serial, or SNMP trap).	1. Configure Serial Events—set Scratch Pad bit 2. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	175 172 187 175
		1. Configure Serial Events—set Scratch Pad bit 2. (Delay only) Configure Expanded Digital Events—set time delay and set a Scratch Pad bit after the delay 3. (Email message only) Configure Email (Streaming only) Configure Streaming 4. (Except data logging) Configure Event Messages—send message or data based on timer-expired bit (Data logging) Configure Data Logging based on timer-expired bit and configure Email (optional)	175 158 175 146 172 187 175

Steps for Configuring Events and Reactions—Firmware 8.0 and Lower

The following table shows steps you would use to configure possible events and reactions if you are using firmware 8.0 or lower with PAC Manager 8.0 or 8.1 (see "Types of Events, Alarms, and Reactions" on page 38 for more details). Page numbers refer you to the memory map addresses in Appendix A that you would use for configuration.

See explanations starting on page 44 for important information you'll need to set up the different kinds of events and reactions.

Event	Reaction	When?	Configuration Steps	See page
If 4-channel digital point is on/off	Turn 4-channel digital point on/off (on same I/O unit)	Now	Configure Digital Events	157
		After a delay	Configure Timers	158
	Turn 4-channel digital point on/off (on different I/O unit) OR Log data OR Copy memory map data OR Send message (stream, email, serial, or SNMP trap).	Now	1. Configure Digital Events—set Scratch Pad bit 2. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	157 172 187 175
		After a delay	1. Configure Timers—set Scratch Pad bit 2. (Email message only) Configure Email (Streaming only) Configure Streaming (SNMP only) Configure SNMP 3. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	158 175 146 141 172 187 175
If analog point value (Engineering Units) goes above or below a specified value OR If digital counter reaches a specified value	Turn 4-channel digital point on/off (on same I/O unit).	Now	1. Configure Alarm Events (high alarm or low alarm)—set Scratch Pad bit 2. Configure Digital Events—turn on/off point	171 157
		After a delay	1. Configure Alarm Events (high alarm or low alarm)—set Scratch Pad bit 2. Configure Timers—turn on/off point	171 158
	Turn 4-channel digital point on/off (on different I/O unit) OR Copy memory map data OR Log data OR Send message (stream, email, serial, or SNMP trap).	Now	1. Configure Alarm Events (high alarm or low alarm)—set Scratch Pad bit 2. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	171 172 187 175
		After a delay	1. Configure Alarm Events (high alarm or low alarm)—set Scratch Pad bit 2. Configure Timers—set time delay and set a Scratch Pad bit after timer expires 3. (Email message only) Configure Email (Streaming only) Configure Streaming (SNMP only) Configure SNMP 4. (Except data logging) Configure Event Messages—send message or data based on timer-expired bit (Data logging) Configure Data Logging based on timer-expired bit and configure Email (optional)	171 158 175 146 141 172 187 175

Event	Reaction	When?	Configuration Steps	See page
If analog point value (Engineering Units) or quadrature counter goes outside an allowable range	Turn 4-channel digital point on/off (on same I/O unit)	Now	1. Configure Alarm Events (deviation alarm)—set Scratch Pad bit 2. Configure Digital Events—turn on/off point	171 157
		After a delay	1. Configure Alarm Events (deviation alarm)—set Scratch Pad bit 2. Configure Timers—turn on/off point	171 158
	Turn 4-channel digital point on/off (on different I/O unit) OR Copy memory map data OR Log data OR Send message (stream, email, serial, or SNMP trap).	Now	1. Configure Alarm Events (deviation alarm)—set Scratch Pad bit 2. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	171 172 187 175
		After a delay	1. Configure Alarm Events (deviation alarm)—set Scratch Pad bit 2. Configure Timers—set time delay and set a Scratch Pad bit after timer expires 3. (Email message only) Configure Email (Streaming only) Configure Streaming (SNMP only) Configure SNMP 4. (Except data logging) Configure Event Messages—send message or data based on timer-expired bit (Data logging) Configure Data Logging based on timer-expired bit and configure Email (optional)	171 158 175 146 141 172 187 175
If a specific string is received by a serial module	Turn4-channel digital point on/off (on same I/O unit)	Now	1. Configure Serial Events—set Scratch Pad bit 2. Configure Digital Events—turn on/off point	175 157
		After a delay	1. Configure Serial Events—set Scratch Pad bit 2. Configure Timers—turn on/off point	175 158
	Send SNMP trap	Now	1. Configure Serial Events—send SNMP trap 2. Configure SNMP	175 141
		After a delay	1. Configure Serial Events—set Scratch Pad bit 2. Configure Timers—set time delay and set a Scratch Pad bit after timer expires 3. Configure SNMP 4. Configure Event Messages—send trap based on timer-expired bit	175 158 141 172
	Send one-time email	Now	1. Configure Serial Events—send email 2. Configure Email	175 175
	Turn 4-channel digital point on/off (on different I/O unit) OR Copy memory map data OR Log data OR Send message (stream, serial, or multiple e-mails)	Now	1. Configure Serial Events—set Scratch Pad bit 2. (Except data logging) Configure Event Messages—send message or data (Data logging) Configure Data Logging and configure Email (optional)	175 172 187 175
		After a delay	1. Configure Serial Events—set Scratch Pad bit 2. Configure Timers—set time delay and set a Scratch Pad bit after timer expires 3. (Email message only) Configure Email (Streaming only) Configure Streaming 4. (Except data logging) Configure Event Messages—send message or data based on timer-expired bit (Data logging) Configure Data Logging based on timer-expired bit and configure Email (optional)	175 158 175 146 172 187 175

PAC-R
EB
SB
UIO
EIO

Using Digital Events and Reactions

NOTE: Availability varies depending on I/O processor, firmware, and module. See "Types of Events, Alarms, and Reactions" on page 38.

In a digital event, the I/O unit monitors one or more inputs, outputs, and Scratch Pad bits for a match to a specific pattern (the event). When the pattern is matched, the I/O unit reacts in a predetermined way. The reaction can turn digital points on or off and can also set bits in the Scratch Pad. You can configure up to 128 digital events and reactions.

Digital event/reactions can be as simple as turning on a light (reaction) when a door opens (event). They can also be very complex, depending on your needs. For example, suppose you need to monitor a critical group of switches. If switches 1, 2, and 3 are all off at the same time, you want to turn on an emergency light and sound an alarm. You can set up a digital event for the state of the three switches, and a reaction that automatically turns on the emergency light and alarm.

In addition to digital states, events can include alarm or other conditions noted in the Scratch Pad. For instance, to regulate the temperature of a room, you might set up an alarm event that turns on a bit in the Scratch Pad when the temperature reaches 78° F (see "Using Alarms and Reactions" on page 48). Then you would set up a digital event/reaction to turn on a fan when that Scratch Pad bit is on.

NOTE: If you want to turn on or off digital points that are located on a different I/O unit, you can do so by using the memory map copying feature when setting up event messages (see page 49).

Digital On/Off and Scratch Pad Masks

Both events and reactions are in the form of a mask. Digital point masks represent 64 possible digital states; you choose whether these represent point states or on-latch or off-latch states. Scratch Pad masks represent whatever you decide each bit should be.

For each digital event/reaction, you set up two to eight masks (up to four for the event and up to four for the reaction), as shown below.

For the event: The table below shows possible triggers for the event, in the form of four masks. You can configure only Trigger #1, only Trigger #2, or both. If you configure both, both must be true for the event to be true. Choose the trigger(s) you want to use; then set up the masks.

Trigger #1		Trigger #2	
On mask	**Off mask**	**On mask**	**Off mask**
Digital point state Digital point on-latch Digital point off-latch HDD point state HDD on-latches HDD off-latches Scratch Pad bits	Digital point state Digital point on-latch Digital point off-latch HDD point state HDD on-latches HDD off-latches Scratch Pad bits	Scratch Pad bits Scratch Pad Integer 64 Digital point state Digital point on-latch Digital point off-latch	Scratch Pad bits Scratch Pad Integer 64 Digital point state Digital point on-latch Digital point off-latch

For the reaction: This table shows possible reactions, again in the form of four masks. You can configure only Reaction #1, only Reaction #2, or both. When the event occurs, all configured reactions will take place. Choose the reaction(s) you want to occur, and then set up the masks.

Reaction #1		Reaction #2	
On mask	**Off mask**	**On mask**	**Off mask**
Set digital point state Clear digital point latch Set HDD point state Clear HDD latch Set Scratch Pad bits	Set digital point state Clear digital point latch Set HDD point state Clear HDD latch Set Scratch Pad bits	Set Scratch Pad bits Set Scratch Pad Integer 64 Set digital point state Clear digital point latch	Set Scratch Pad bits Set Scratch Pad Integer 64 Set digital point state Clear digital point latch

NOTE: Trigger #1 does NOT control Reaction #1; Trigger #2 does not control Reaction #2. Instead, all the masks work as a group. All the event masks must be a match for the I/O unit to set the reaction(s), and if the event occurs, any and all reactions will be set. If it doesn't matter whether a specific point or bit is on or off, leave its value at zero in both the on mask and the off mask.

To choose the triggers and reactions from the tables above, you also set up another mask: the event detail mask (see "Event Detail Mask," below, for examples).

When you configure events and reactions, the masks are in hex notation. If you are setting up a Digital On mask for points on the first two modules, for example, you might do so as follows:

Module position:	1				0			
Point number:	3	2	1	0	3	2	1	0
State:	On	--	On	On	--	--	--	On
Binary notation:	1	0	1	1	0	0	0	1
Hex notation:	B				1			

(For more information on mask data format, see page 58.)

You can also configure the I/O unit to send a message as a reaction to digital events. See page 49.

Event Detail Mask

In addition to the two to eight on/off bitmasks mentioned above, there's also a separate bitmask that indicates the details for the event—which triggers to use and which reactions should occur. For Event 0, for example, this mask goes in memory map address FFFF F0D4 0044.

The table on the following page shows which bits to set in this detail mask to achieve the triggers and reactions you want. In the table the triggers and reactions are separated for clarity, but you build only one mask that includes all the elements you need. See the example below the table.

Remember that bit numbering starts at 0.

Event Detail Mask Bits

	For this	Set these bits	Binary example	Hex example
Event	Trigger #1			
	Digital Point State	None	0000 0000 0000 0000 0000 0000 0000 0000	0x00000000
	Digital Point On-Latch	1, 2	0000 0000 0000 0000 0000 0000 0000 0110	0x00000006
	Digital Point Off-Latch	1	0000 0000 0000 0000 0000 0000 0000 0010	0x00000002
	HDD Point State	4	0000 0000 0000 0000 0000 0000 0001 0000	0x00000010
	HDD Point On-Latch	1, 2, 4	0000 0000 0000 0000 0000 0000 0001 0110	0x00000016
	HDD Point Off-Latch	1, 4	0000 0000 0000 0000 0000 0000 0001 0010	0x00000012
	Scratch Pad Bits	15	0000 0000 0000 0000 1000 0000 0000 0000	0x00008000
	Trigger #2			
	Scratch Pad Bits	None	0000 0000 0000 0000 0000 0000 0000 0000	0x00000000
	Scratch Pad Integer 64	6	0000 0000 0000 0000 0000 0000 0100 0000	0x00000040
	Digital Point State	16	0000 0000 0000 0001 0000 0000 0000 0000	0x00010000
	Digital Point On-Latch	16, 17, 18	0000 0000 0000 0111 0000 0000 0000 0000	0x00070000
	Digital Point Off-Latch	16, 17	0000 0000 0000 0011 0000 0000 0000 0000	0x00030000
Reaction	Reaction #1			
	Digital Point State	None	0000 0000 0000 0000 0000 0000 0000 0000	0x00000000
	Clear Digital Point Latches	3	0000 0000 0000 0000 0000 0000 0000 1000	0x00000008
	HDD Point State	5	0000 0000 0000 0000 0000 0000 0010 0000	0x00000020
	Clear HDD Latches	3, 5	0000 0000 0000 0000 0000 0000 0010 1000	0x00000028
	Scratch Pad Bits	20	0000 0000 0001 0000 0000 0000 0000 0000	0x00100000
	Reaction #2			
	Scratch Pad Bits	None	0000 0000 0000 0000 0000 0000 0000 0000	0x00000000
	Scratch Pad Integer 64	7	0000 0000 0000 0000 0000 0000 1000 0000	0x00000080
	Digital Point State	19	0000 0000 0000 1000 0000 0000 0000 0000	0x00080000
	Clear Digital Point Latches	21	0000 0000 0010 0000 0000 0000 0000 0000	0x00200000
	How reaction occurs (applies to both Reaction #1 and Reaction #2)			
	Reaction occurs once*	8	0000 0000 0000 0000 0000 0001 0000 0000	0x00000100
	* By default, the reaction occurs continuously. See "How Digital Events Trigger Reactions" on page 47).			

Event Detail Mask Example

In the table above, the triggers and reactions are separated for clarity, but in practice you build only one mask that includes all the elements you need.

For example, suppose you want to use two triggers and one reaction for Event 0, like this:

Trigger #1 = Digital point on-latch
Trigger #2 = Scratch Pad bit
Reaction #1 = [not used]

Reaction #2 = Scratch Pad Integer 64
And you want the reaction to occur only once, not continuously.

Looking at the table, you see you must set the following bits (don't forget, bit numbering starts at 0):

Trigger #1 = Digital point on-latch = bits 1 and 2
Trigger #2 = Scratch Pad bit = [none]
Reaction #1 = [not used] = [none]
Reaction #2 = Scratch Pad Integer 64 = bit 7
Reaction occurs once = bit 8

So in memory map address FFFF F0D4 0044, you enter a hex mask you build like this:

Bit #:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0
Hex:	0				0				0				0				0				1				8				6			

How Digital Events Trigger Reactions

Reactions to digital events are level-triggered, not edge-triggered, by default. The I/O unit continually checks the digital state to see if it matches the event. The I/O unit sends the reaction as soon as the state matches the event, and the I/O unit continues to send the reaction until the state changes. On a SNAP PAC I/O unit with 8.1 firmware or higher, however, you can set the event to trigger the reaction just once, rather than continuously.

In either case, if the state changes so that it no longer matches the event, the I/O unit does NOT reverse the reaction.

Digital Event/Reaction Example

For example, suppose you have set up an event/reaction to turn on a light when a door is open. As soon as the event occurs (the door opens), the I/O unit sends the reaction (turn on the light). Unless you have set the reaction to be triggered just once, the reaction continues to be sent as long as the door is open.

When the door is shut, the I/O unit does NOT turn the light off. To turn off the light when the door is shut, you need to set up a second event/reaction.

Suppose the input for the door's status is on point 0, and the output for the light is on point 5. Here are the two event/reactions to turn on the light when the door is open, and turn off the light when the door is shut:

Event #0:	IF	Mod 0 Pt 0 (Door) is	OFF (Open)
Reaction #0:	THEN	Turn Mod 1 Pt 1 (Light)	ON

Event #1:	IF	Mod 0 Pt 0 (Door) is	ON (Closed)
Reaction #1:	THEN	Turn Mod 1 Pt 1 (Light)	OFF

PAC-R EB SB UIO EIO

Using Alarms and Reactions

A reaction can also be set up as a response to an alarm. You can configure alarms for analog points or digital counters. (See "Alarm Event Settings—Read/Write" on page 171). For example, you could trigger an alarm when the pressure in a tank rises above a certain level, or when a specific number of boxes on a conveyor have passed through a beam sensor. For each alarm, you configure a suitable reaction.

For analog points, alarms are based on the analog input value. For digital points, alarms are based on the counter value. For each point, you can configure any or all of the following alarms:

- **Deviation alarm**—sets a range on either side of the current value that is acceptable; beyond that range, the reaction occurs. For example, suppose you are monitoring temperature. If the current value is 80 and you set a deviation limit of 6, the reaction will not occur unless the value drops below 74 or rises above 86.

  NOTE: When a reaction occurs, the deviation limit stays the same, but the value that set off the reaction becomes the new deviation value. In this example, if the temperature drops to 73, the reaction occurs. Six is still the deviation limit, but now 73 is the deviation value; another reaction will not occur unless the value drops below 67 or rises above 79.

- **High-limit alarm**—sets a fixed upper limit. If the analog value or counter is higher than the high limit, the reaction occurs.
- **Low-limit alarm**—sets a fixed lower limit. If the analog value or counter is lower than the low limit, the reaction occurs.

How Alarms Trigger Reactions

Reactions to alarms are edge-triggered, not level-triggered, and when the alarm state changes, the reaction is automatically reversed. The I/O unit sends the reaction just once, as soon as the alarm occurs (at the "edge" of the alarm). The I/O unit does not send the reaction again until the alarm occurs again. If the alarm stops, however, the I/O unit *reverses* the reaction.

For example, suppose you set up a high-limit alarm that turns on a Scratch Pad bit that will turn on a fan if the temperature goes over 70°. As soon as the alarm state occurs (the temperature goes over 70°), the I/O unit sends the reaction (turns on the bit to turn on the fan). If the temperature remains above 70°, the I/O unit does not continue to turn on the fan bit; the bit just stays on.

When the temperature falls back below the high limit (70° minus whatever deadband you have set), the I/O unit automatically reverses the reaction by turning the Scratch Pad bit off. (To turn the fan off, you would have to set up a reaction for the off bit, turning the fan off.)

Notice that the reaction and its reversal are absolute; they do not depend on the pre-alarm condition. For example, if the bit to turn on the fan was already on at the time the temperature rose above 70°, the reaction would turn the bit on even though it was already on. When the temperature fell back below 70°, the I/O unit would *not* return the fan bit to its pre-alarm condition (on); it would turn the bit off.

PAC-R EB UIO EIO

Using Serial Events and Reactions

(Not available on SB brains) If you are using Opto 22 serial communication modules with a SNAP PAC R-series, SNAP PAC EB, SNAP Ultimate, or SNAP Ethernet I/O unit, you can configure a serial event to send a serial message, to send an SNMP trap, or to turn bits in the Scratch Pad on or off

when a specific string is received from one or more modules. See "Serial Event Configuration—Read/Write" on page 175 or "Wiegand Serial Event Configuration—Read/Write" on page 177.

Before you configure serial events and reactions, make sure you have configured the serial modules. (See page 134 or page 136.)

PAC-R EB UIO EIO

Using SNMP in Reactions

To send an SNMP trap as a reaction to an event, you must also tell the I/O unit information about the SNMP agent and access privileges for hosts on the network.

SNMP Access Privileges

Community groups control access to information from the SNAP Ethernet-based I/O unit. The first community group, public, is set up for you. All hosts on the network are part of the public group; all can read and write I/O unit data but cannot receive traps. You can change or delete this public group if necessary.

In order to receive traps, a host must be a registered *management host* and be part of a community group that does have access privileges for traps. Once a registered management host becomes part of a community group, that group is no longer available to non-registered hosts. It includes only the hosts registered to it.

You must set up the additional community groups you need, either in PAC Manager or in your program. See "SNMP Configuration—Read/Write" on page 141. Note that SNMP configuration must be stored to flash memory and the I/O unit restarted for it to take effect.

SNMP Traps

The SNAP Ethernet-based I/O unit can send three kinds of traps:

- Authentication trap—sent when a host requests data that is outside its access permissions
- Cold start trap—sent whenever the I/O unit is turned on
- Exception trap—sent in reaction to an event; an exception trap is a type of event message.

Authentication and cold start traps require no configuration and can simply be enabled. Exception traps must be configured when you set up event messages.

PAC-R EB UIO EIO

Setting Up Event Messages

You may need to send a message—via email, data streaming, SNMP, or a serial module—from the SNAP Ethernet-based I/O unit when a specific event occurs. For example, you could send a message if a digital point is on, if an analog point reaches a certain value, or if a specific string is received through a serial module. You can send one type of message or more. Your program monitors the event and triggers the message you have configured.

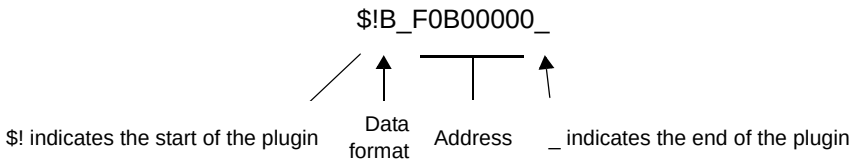
You can configure up to 128 messages, either in PAC Manager or in your program. See "Event Message Configuration—Read/Write" on page 172 for memory map addresses to use.

Copying Binary or Memory Map Data

You can use memory map copying to do the following:

- Copy data on the same I/O unit.
- Copy data to a memory map location on another unit.

Set up memory map copying when you configure event messages. For Message Text, enter a plugin containing a memory map address to write *from* (the source address), in the following format:



or a four-byte constant, in this format:

```
&#x00000001_
```

NOTE: Constants must be written in exactly four bytes (8 hex characters).

While the data format indicator in the plugin can be other types (D=integer, F=float), B is typically used for memory map copying. The other types copy a *string representation* of the data, because plugins are primarily used for generating messages and emails. For more about plugins, see "Using Plugins" on page 50.

PAC-R EB UIO EIO

Using Email

You can send an email message in response to an event. In addition to setting up the email message when you configure event messages, you also need to tell the I/O unit where to send the email. See "Email Configuration—Read/Write" on page 175.

PAC-R EB UIO EIO

Using Plugins

Several plugins are available for use in event/reactions and messages.

To do this	Use this plugin	In these places
Include the pattern string from a serial communication module.	`$!_str_`	Serial event/reactions
Show which serial port sent the pattern string.	`$!_port_`	Serial event/reactions
Include data from a memory map address. X = type of data (S=string representation of the data, D=integer, F=float, P=IP address, B=4 binary bytes) YYYYYYYY = memory map address (see examples below)	`$!X_YYYYYYYY_`	Serial event/reactions Event messages Memory map copying Email
Number emails with a sequence ID.	`$!_seqid_`	Email
Turn digital points on or off using a bit mask.	`&#x00000000_`	Memory map copying Event messages

NOTE: For email messages, message text including plugins must be 126 bytes (characters) or less. The message length after all plugins have been expanded into their data values must be 255 bytes or less.

Examples: Including Data from Memory Map Addresses

Memory map addresses are shown in PAC Manager, or see Appendix A for the complete memory map. Here are a couple of examples:

To include the on/off state of a switch on module 0 point 3, you would put this in the message:

```
$!D_F08000C0_
```

To include the temperature of an ICTD input on module 4 point 0, you would use:

```
$!F_F0A00400_
```

Sending Binary Data in Event Messages

To send binary data in the text of an event message, begin with `&#x` and end with `_`. You can include any number of ASCII hex digits up to the 126-byte limit for the message field. You can also include multiple &#x plugins. This plugin is resolved after all other plugins have been resolved, and only just before sending the contents of the message field. Examples:

To include an embedded null (one binary character):

```
&#x00_
```

To include a number of binary characters:

```
&#x0a0dCF0034_
```

Streaming Data

PAC-REB UIO EIO SIO

Most communication involves the two-step process of request and response. A faster way of getting information from a SNAP Ethernet-based I/O unit, however, is by streaming data. Streaming does not use TCP/IP; it uses the User Datagram Protocol (UDP) instead.

NOTE: Because Modbus/TCP runs on TCP, not UDP, streaming data via Modbus/TCP is not possible. However, you can stream to a non-Modbus host at the same time you are using the Modbus/TCP protocol for another purpose.

Streaming is a fast way to get continuous information about I/O from the SNAP Ethernet-based I/O unit and is ideal for data acquisition applications. When it streams, the I/O unit sends data at regular intervals to specified IP addresses. You set up the interval, the IP addresses to receive the data, and (optionally) the port number. The I/O unit sends the data at the specified interval. The communication is one-way; the I/O unit does not wait for a response.

CAUTION: If you stream to multiple IP addresses, and one or more of the streaming targets is either offline or not running the application that receives the stream, delays may occur. If a target is offline, the I/O unit will stop streaming while it tries to resolve the IP address. If the application is not running on the PC that receives the stream, the PC will send the I/O unit an error message; if the stream occurs frequently, the additional error messages can slow down the network.

Streaming involves two steps: configuring parameters on the I/O unit for streaming, and receiving data in your application.

Configuring Parameters for Streaming

To set up the I/O unit for streaming data, you can use PAC Manager or you can write to the memory map area for configuring streaming (see page 146).

- Write how often in milliseconds you want to receive the streamed data. If you are configuring streaming to use only as a reaction to a digital event or an analog alarm condition, set the streaming interval to 0 (send once).
- The data that is streamed is normally the whole Streaming section of the memory map (see page 168). To stream only a portion of the Streaming section, or to stream a different part of the memory map, write the starting address and size of the data to stream.

  Note that high-density digital module data is not included in the Streaming section of the memory map. It can be streamed separately by writing the starting address and size of data you need.

  You can also use the Packed areas (see "Analog EU or Digital Counter Packed Data—Read" on page 169 and "Digital Packed Data—Read/Write" on page 170) to get a large section of useful data all at once.
- Write the IP port number on the PCs or devices that will receive streamed data. Your application must refer to this port number.
- Write the IP addresses of the hosts that should receive the data (the target addresses).
- To turn streaming on, write anything but a zero to the Streaming On/Off address. To turn streaming off, write a zero.

Receiving Streamed Data

As soon as you've configured parameters for streaming, the I/O unit starts sending the data you requested. Your application does not need to respond; it only needs to process the data.

Stream Packet Format

The stream packet consists of an IEEE 1394 header and data. Addresses will be zero-filled in areas that don't apply. For example, addresses that show analog data will be filled with zeros for points that are digital.

Streaming Section Packet. The following table shows the format for the stream packet based on receiving the entire Streaming section of the memory map (see page 168). (If you are streaming data from other sections of the memory map, see "Other Memory Map Addresses Streaming Packet." on page 53.)

Area	Number of Bytes	Description
Packet header	2	Total length of packet
	2	First byte is zero-filled; second byte contains transaction code 0x0A for an isochronous data block (4 bits) and synchronization code for Opto 22 use (4 bits).

Area	Number of Bytes	Description
Useful data	256	Analog values in Engineering Units (IEEE floats). Contains 4 bytes of data for each of 64 points on 16 modules, starting with point 0 on module 0. If the analog module contains more than 4 points, only the first 4 points are included.
	256	Point feature data (counter data) (unsigned 32-bit integers).
	8	On/off state of all digital points on 4-channel modules (mask). (Streaming does not provide data for high-density digital modules.)
	8	On-latch state (mask)
	8	Off-latch state (mask)
	8	Active counters (mask)
Unused data	56	Reserved for future data; zero-filled

For example, the first bytes of a stream packet might look like this:

These packet bytes:	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	→
Contain this hex data:	header				41	77	33	33	41	3F	AC	66	→
For these points:	--				0				1				→
On module at position:	--				0								→

In the body of the stream packet, IEEE floats are arranged in low point/low address order. See page 60 for an example. All masks in the stream packet are in Big Endian format, however, with higher-numbered points in the lower-addressed byte. See page 58 for more information about how data in a mask is formatted.

Other Memory Map Addresses Streaming Packet. The following table shows the format for the streaming packet if you are receiving streamed data from memory map addresses other than the Streaming area.

Area	Number of Bytes	Description
Packet length	2	Number of bytes in this packet
Zero byte	1	Always zero
Transaction byte	1	Upper nibble: A Lower nibble increases with each transmission.
Source memmap address	4	Address you configured in the streaming configuration area
Data	Number of bytes you requested	Data format and organization of the memory map areas you requested. See Appendix A for details.

Logging Data

PAC-R
EB
SB
UIO
EIO

Your SNAP PAC R-series, SNAP PAC EB or SB, SNAP Ultimate, or SNAP Ethernet I/O unit includes a feature that allows data from memory map addresses to be recorded in a log file. The data from up to 64 memory map addresses can be logged, and all logged data is recorded in one file. The log file holds up to 300 lines of data; when it is filled, new entries replace the oldest ones.

Logging data requires two steps:

- Configure the events (Scratch Pad masks) that trigger logging and the memory map addresses to log data from.
- Read the data from the data log.

Configuring the Event and Memory Map Addresses to Log

Use either PAC Manager or your own application to configure the events that trigger logging (the Scratch Pad masks) and to tell the I/O unit which memory map addresses to log data from. "Data Logging Configuration—Read/Write" on page 187 shows the memory map addresses used to configure this information. Remember that the Scratch Pad masks work together: both masks must be a match to trigger logging. If it doesn't matter whether a specific bit is on or off, leave its value at zero in both the on mask and the off mask.

Reading the Data from the Data Log

The composite log file can be viewed through PAC Manager or emailed. (If the log will be emailed, remember to configure email.) The log file can also be accessed by a software application you develop. "Data Log—Read/Write" on page 188 shows the memory map addresses in which the data log is stored.

Each address in the log file consists of the date and time stamp, the memory map address the data is coming from, the format of the data, and the data itself. For example, address FFFFF3020000, the first data log address, might contain the following information:

Information in log file address (in hex):	07D0	06	1E	0E	23	2A	07	F0A0008C	00000066	41773333
Meaning:	2001	06	30	14	35	42	07	point 2 analog max value	float	15.45
Description:	year	month	day	hour	min-ute	sec-ond	100th of a sec.	memory map address the data comes from	data format*	data
Number of bytes:	2	1	1	1	1	1	1	4	4	4

* Data format indicators (in hex) may be any of the following:

- 66 (f) for float
- 64 (d) for signed value
- 78 (x) for unsigned value.

In this example, the data is date/time stamped for 42.07 seconds after 2:35 P.M. on June 30, 2001. The data shows that the analog maximum value for point 2 is 15.45 in Engineering Units (a float).

Using PID Loops

PAC-R
EB
SB
UIO
EIO

What is a PID?

A proportional integral derivative (PID) control system (often referred to as a PID loop) monitors an input or process variable, compares the variable's current value to a desired value (a setpoint), and calculates an output to correct error between the setpoint and the variable. Because the calculation is complex, it is done by a mathematical formula that is adjusted (tuned) for each PID loop. The mathematical formulas vary, but all PID systems share these fundamental concepts:

- They evaluate an input or process variable against its setpoint.
- They control an output to correct the variable.
- The controller output consists of proportional, integral, and derivative calculations.
- The effect of proportional, integral, and derivative calculations is modified by user-determined P, I, and D constants.
- The P, I, and D constants need to be tuned for each system.

PID Loops on SNAP Ethernet-based I/O Units

PID loop control is provided on the following I/O units:

I/O Processor	Number of PID loops
SNAP-PAC-R1 SNAP-PAC-R1-FM SNAP-PAC-R2 SNAP-PAC-R2-FM	96
SNAP-PAC-EB1 SNAP-PAC-EB1-FM SNAP-PAC-EB2 SNAP-PAC-EB2-FM SNAP-PAC-SB1 SNAP-PAC-SB2	96
SNAP-UP1-ADS SNAP-UP1-M64	32
SNAP-B3000-ENET SNAP-ENET-RTC	16

NOTE: PID capabilities in these I/O units are compatible with PAC Control, but not with OptoControl.

The simplest way to use these PIDs is with PAC Control, which provides easy-to-use configuration and tuning tools. For more information, see Opto 22 form #1700, the *PAC Control User's Guide*.

If you are not using PAC Control, however, it is possible to configure and tune PIDs through the I/O unit's memory map. Memory map addresses start on page 111.

You can configure each PID loop with unique settings for a large number of parameters. For a simple PID loop, you must configure at least the following:

- Input (the process variable being monitored)

- Setpoint (the desired value)
- Output (the I/O point that effects change in the system)
- Scan time (how often the input is sampled, the calculation performed, and the output updated)
- PID algorithm used (Four algorithms are available; see "Algorithm Choices," below.)

You can also configure the following parameters:

- Valid range for input
- Upper and lower clamps for output
- Minimum and maximum change for output
- Forced output value or use of manual mode if input goes out of range
- Feed forward gain
- Square root of input (typcially used with differential pressure cells)

In these PID loops, the derivative is applied only to the process variable (the input) and not to the setpoint. This means you can change the setpoint without causing spikes in the derivative term. Non-velocity PIDs also prevent integral windup by back calculating the integral without the derivative term. The feed forward term ("bias") is added before output clamping and has a tuning factor.

If desired, you can cascade PIDs by simply using the output point of one PID loop as the input point for another.

PAC-R
EB
SB
UIO
EIO

Algorithm Choices

When you configure a PID loop, choose one of the following algorithms:

- Velocity - Type C (available only for PAC-R, EB, and SB with firmware version 8.5e and higher)
- Velocity - Type B
- ISA
- Parallel
- Interacting

The only difference between Velocity - Type C and Velocity - Type B is the Term P equation (see equations below). The ISA, Parallel and Interacting algorithms are functionally equivalent; the only difference is the way the tuning constants are factored. The identical and differing equations for all algorithms are shown in the following sections.

Key to Terms Used in Equations

PV	Process variable; the input to the PID
SP	Setpoint
InLo, InHi	Range of the input
OutLo, OutHi	Range of the output
Gain	Proportional tuning parameter. Unitless. May be negative.
TuneI	Integral tuning parameter. In units of seconds. Increasing magnitude increases influence on output.

TuneD	Derivative tuning parameter. In units of seconds. Increasing magnitude increases influence on output.
Output	Output from the PID
Err_1	The Error (PV – SP) from the previous scan
Integral	Integrator. Anti-windup is applied after the output is determined to be within bounds.
PV1, PV2	PV from the previous scan and the scan before that.
ScanTime	Actual scan time (time since previous scan)

Equations Common to All Algorithms

```
Err = PV - SP
Span = (OutHi - OutLo) / (InHi - InLo)
Output = Output + FeedForward * TuneFF
```

Velocity - Type C Algorithm

```
TermP = ( PV - PV1 )
TermI = TuneI * ScanTime * Err
TermD = TuneD / ScanTime * ( PV - 2 * PV1 + PV2 )
ΔOutput = Span * Gain * ( ΔTermP + ΔTermI + ΔTermD )
```

Velocity - Type B Algorithm

This algorithm is similar to the algorithm used on legacy mistic I/O units, except that the derivative does not act on setpoint changes.

```
TermP = ( Err - Err_1 )
TermI = TuneI * ScanTime * Err
TermD = TuneD / ScanTime * ( PV - 2 * PV1 + PV2 )
ΔOutput = Span * Gain * ( ΔTermP + ΔTermI + ΔTermD )
```

Non-velocity Algorithms

These equations were derived from the article "A Comparison of PID Control Algorithms" by John P. Gerry in *Control Engineering* (March 1987). These three equations are the same except for the tuning coefficients; converting from one equation to another is merely a matter of converting the tuning coefficients.

Equations common to all but the velocity algorithm:

```
Integral = Integral + Err
TermP = Err
TermI = TuneI * ScanTime * Integral
TermD = (TuneD / ScanTime) * ( PV - PV1 )
```

"Ideal" or ISA Algorithm

```
Output = Span * Gain * ( TermP + TermI + TermD )
```

"Parallel" Algorithm:

```
Output = Span * ( Gain * TermP + TermI + TermD )
```

"Interacting" Algorithm:

```
Output = Span * Gain * ( TermP + TermI ) * ( 1 + TermD )
```

Formatting and Interpreting Data

PAC-R
PAC-S
EB
SB
UIO
EIO
SIO
LCE
E1
E2

Data is formatted differently for different addresses in the memory map. The memory map tables in Appendix A (page 111) show whether the data in each address is a mask, a signed or unsigned integer, a float, and so on. This section shows how to format and interpret various types of data when you are reading or writing to a memory-mapped device.

Mask Data

Some data is in the form of a 32-bit or 64-bit mask—four or eight addresses, each holding eight bits. Each bit in the mask contains the data for one thing in a group: one point, one module, one Scratch Pad bit, etc.

Mask Data for SNAP

For example, most digital bank data (as well as high-density digital module data) is in this form. To read the state of digital points in a bank, you would read the eight bytes starting at FFFFF0400000. Here's how the data would be returned:

At address:	FFFFF0400000								→	FFFFF0400007							
These bit numbers:	7	6	5	4	3	2	1	0	→	7	6	5	4	3	2	1	0
Show data for these points:	3	2	1	0	3	2	1	0	→	3	2	1	0	3	2	1	0
On SNAP modules in these positions in the rack:	15				14				→	1				0			

Therefore, at address FFFFF0400000:

This hex data:	B				1			
Equals this binary data:	1	0	1	1	0	0	0	1
Showing the states:	On	Off	On	On	Off	Off	Off	On
Of these points:	3	2	1	0	3	2	1	0
On these modules:	15				14			

Data from other addresses marked as masks is formatted in a similar way.

Mask Data for E1s

The bank area of the memory map is based on a four-point SNAP module. For I/O units with E1 brain boards, each point is treated as the first point on a SNAP module. That means that when you read a bank of digital points on an E1, data appears only in the first of every four points, like this:

At address:	FFFFF0400000								→	FFFFF0400007							
These bit numbers:	7	6	5	4	3	2	1	0	→	7	6	5	4	3	2	1	0
Show data for these points:	--	--	--	0	--	--	--	0	→	--	--	--	0	--	--	--	0
On G1 or G4 modules in these positions in the rack:	15				14				→	1				0			

So, at address FFFF04000000:

This hex data:	1				0			
Equals this binary data:	0	0	0	1	0	0	0	0
Showing the states:	--	--	--	On	--	--	--	Off
Of these points:	--	--	--	0	--	--	--	0
On these modules:				1				0

These memory map addresses apply not only to G1 and G4 modules, but also to integral racks and even to Quad Pak modules. Points on all E1 I/O units are treated the same way, no matter how they are physically placed on the rack.

Unsigned 32-bit Integer Data

Much of the data in the memory map is in the form of unsigned integers, either one byte, two bytes, or four bytes. With multiple bytes, since the memory-mapped devices use a Big Endian architecture, the high order byte is in the low order address.

For example, digital bank counter data is in 4-byte unsigned integers. It takes four bytes to contain the data for one point. To read digital bank counter data for point 0 on module 0, you would start with address FFFFF0400100. The following table shows the pattern of bank counter data for the first few points on a SNAP rack:

Bytes at these addresses:	FFFFF0400100 FFFFF0400101 FFFFF0400102 FFFFF0400103	FFFFF0400104 FFFFF0400105 FFFFF0400106 FFFFF0400107	FFFFF0400108 FFFFF0400109 FFFFF040010A FFFFF040010B	FFFFF040010C FFFFF040010D FFFFF040010E FFFFF040010F	FFFFF0400110 FFFFF0400111 FFFFF0400112 FFFFF0400113	FFFFF0400114 FFFFF0400115 FFFFF0400116 FFFFF0400117	→
Show data for this point:	0	1	2	3	0	1	→
On the module in this position on the rack:	0				1		→

The most significant byte is at the lowest address. For module 0, point 0, for example, you might receive the following data:

At this address	This binary data	Equals this hex data
FFFF F040 0100	0001 0110	16
FFFF F040 0101	1011 1011	BB
FFFF F040 0102	0001 1000	18
FFFF F040 0103	1000 0111	87

16 BB 18 87

The 32-bit integer for this reading would be **16 BB 18 87** (most significant byte at lowest address). This hex figure correlates to the decimal value 381,360,263.

Remember that if you are processing this data using a Little Endian computer (such as an Intel®-based PC), you must convert the data from the Big Endian format in order to use it. Little

Endian format is the opposite of Big Endian; Little Endian places the most significant byte at the highest address.

Digital Point Data (4-Channel Modules)

NOTE: For high-density digital modules, see "Mask Data" on page 58.

For consistency in starting addresses, data for individual digital points has a length of four bytes. However, only the least significant bit contains the data you're looking for.

For example, to read the state of point 0 on module 0, you would start with address FFFFF0800000. Data would be returned as follows:

To read this information:	Point 0 on Module 0: Point State							
Use these addresses:	FFFFF0800000		FFFFF0800001		FFFFF0800002		FFFFF0800003	
These bits:	7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0	
Contain this data (binary):	0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 1	
(hex):	0	0	0	0	0	0	0	1
	Ignore these							Point state is ON.

Digital Point Data for E1s

If you are using I/O units with E1 brain boards, remember that the memory map is based on a four-point SNAP module. For an E1, point data appears in the addresses that correspond to the first of each group of four points in the memory map, like this:

Addresses for point state:	FFFFF0800F00	FFFFF0800E00	FFFFF0800D00	FFFFF0800C00	FFFFF0800B00	FFFFF0800A00	FFFFF0800900	FFFFF0800800	FFFFF0800700	FFFFF0800600	FFFFF0800500	FFFFF0800400	FFFFF0800300	FFFFF0800200	FFFFF0800100	FFFFF0800000
E1 module position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Point number:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Point data appears this way for all module types used with an E1: G1, G4, Quad Pak, and integral racks.

IEEE Float Data

For individual analog points, values, counts, and minimum and maximum values for one point are located next to each other in the memory map. All are four bytes and are IEEE 754 floats.

For example, individual analog point data for points 0 and 1 on module 0 appears in these addresses:

Module	Point	Data	Beginning Address	Ending Address
0	0	Scaled units (E.U.*)	FFFF F026 0000	FFFF F026 0003
		Counts	FFFF F026 0004	FFFF F026 0007
		Minimum value (E.U.*)	FFFF F026 0008	FFFF F026 000B
		Maximum value (E.U.*)	FFFF F026 000C	FFFF F026 000F
	1	Scaled units (E.U.*)	FFFF F026 0040	FFFF F026 0043
		Counts	FFFF F026 0044	FFFF F026 0047
		Minimum value (E.U.*)	FFFF F026 0048	FFFF F026 004B
		Maximum value (E.U.*)	FFFF F026 004C	FFFF F026 004F

* Engineering Units

IEEE 754 float format is as follows:

1 bit	8 bits	23 bits
x	xxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx
Sign	Exponent	Significand

Float calculation: $(-1)^{Sign} \times [1 + Significand/2^{23}] \times 2^{(Exponent-127)}$

Example for Opto 22 memory map

At this address:	base address		base address + 1		base address + 2		base address + 3	
This hex data:	41		77		33		33	
In binary:	0	100 0001	0	111 0111	0011	0011	0011	0011
In these bits:	31	30 . . . 23	22 . . . 0					
Equals (in decimal):	0	130	7,811,891					
Representing:	Sign	Exponent	Significand					

Decimal $= (-1)^0 \times [1 + 7{,}811{,}891/2^{23}] \times 2^{(130-127)}$

$= 1 \times [1.931] \times 8$

$= 15.45$ (rounded to 2 decimal places)

For more information on floats and issues that may arise in their use, see form #1755, *Using Floats Technical Note*, available on our website, www.opto22.com.

Analog Bank Data

Remember that the bank area of the memory map is set up for four points per module. Analog modules with more than four points (channels) will show data for points 0–3 only. If the analog modules you are using have only one or two points, the addresses for the upper two or three points in each module will contain the following: for output modules, 0; for input modules, FFFFFFFF.

For example, to read all bank analog point values in scaled units, you would read 256 bytes starting at address FFFFF0600000. Here's how data for two-channel input modules in positions 0 and 1 would appear:

Beginning Address	Ending Address	Data Format	Module	Point
FFFF F060 0000	FFFF F060 0003	four bytes—IEEE float	0	0
FFFF F060 0004	FFFF F060 0007	four bytes—IEEE float		1
FFFF F060 0008	FFFF F060 000B	FFFFFFFF		2
FFFF F060 000C	FFFF F060 000F	FFFFFFFF		3
FFFF F060 0010	FFFF F060 0013	four bytes—IEEE float	1	0
FFFF F060 0014	FFFF F060 0017	four bytes—IEEE float		1
FFFF F060 0018	FFFF F060 001B	FFFFFFFF		2
FFFF F060 001C	FFFF F060 001F	FFFFFFFF		3

On an I/O unit with an E2 brain board, all modules have only one point, so the upper three points would contain 0 (outputs) or FFFFFFFF (inputs).

3: Using the OptoMMP Communication Toolkit

Introduction

This chapter shows you how to use the OptoMMP Communication Toolkit for communication between a computer and an Opto 22 Ethernet-based memory-mapped device. (Exception: The Toolkit cannot currently be used for direct communication between a PC and an SB brain.)

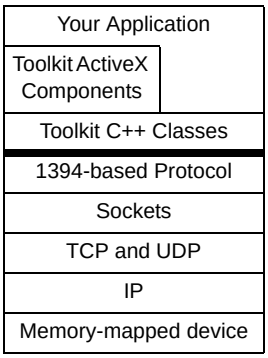
The Toolkit includes two ActiveX components and two C++ classes. The components and classes have nearly identical programming interfaces and are used in the same manner. They hide the details of Ethernet communications and the OptoMMP protocol, making it easy to communicate with the memory-mapped device.

This chapter assumes that you already have an understanding of ActiveX and C++ programming.

The toolkit's two components and two classes communicate differently with Opto 22 memory-mapped devices:

- **OptoSnapIoMemMap component or class**—Communicates with the device's memory map via TCP or UDP. You can use it to read and write to analog and digital points, configuration settings, and other sections of the memory map.
- **OptoSnapIoStreaming component or class**—Listens to and processes stream packets received from multiple devices, hiding the details of receiving and processing UDP streaming packets.

The C++ classes handle all the details of Ethernet sockets and the OptoMMP protocol. They provide functions like OptoEnet(), GetDigPtState(), and SetAnaPtValue(), making programming easier and more intuitive.The ActiveX components are essentially wrappers around the C++ classes, making the efficient C++ code available to many types of ActiveX clients (see "Important ActiveX Client Issues" on page 65). The diagram at right shows how layers are related.



The toolkit includes all the code for the C++ classes, ActiveX components, and examples. You are free to use this code at no charge in your own application, with or without modification. However, Opto 22 cannot support modification of the code.

NOTE: Because the OptoMMP Communication Toolkit was originally developed for SNAP Ethernet I/O, it contains some methods and structures that do not apply to other Opto 22 memory-mapped devices. See page 9 for features that apply to each hardware product. For

example, the SNAP-PAC-EB2 brain does not do high-speed digital counting. Methods such as GetStreamReadArea normally read and write data on active counters; you can still use these methods for the EB2, but disregard any data in the areas that don't apply.

A Note on Multithreading

The components and classes in this toolkit do not have internal synchronization. If you wish to access a single instance of an object from multiple threads, you must provide your own external synchronization.

Installing the Toolkit

The OptoMMP Communication Toolkit consists of examples and all source code. It is included on the CD that came with the device and is also available from our website at www.opto22.com.

Two installations are available:

- For Windows, use SnapENETDTK_R20a.exe. It will copy all files and register the ActiveX components.
- For Linux, use snapenetdtk_R20a.tar. It contains just the files needed for a Linux system.

Overview of the ActiveX Components

You can use the OptoSnaploMemMapX and OptoSnaploStreamX ActiveX components from Windows programming environments that support ActiveX components, such as Borland Delphi, Microsoft Visual Basic®, Visual Basic for Applications, VBScript, and Visual C++. See "Important ActiveX Client Issues" on page 65 for more details on suitable clients.

ActiveX is an umbrella term for many technologies based on COM (Component Object Model, a Microsoft technology allowing binary code reuse). Different programming environments use different terminology; in this chapter, we use terms as follows:

- An *interface* is a set of related functions (not a user interface).
- A *component* is an implementation of one or more interfaces.
- A *control* is a component that also has user interface elements.
- An *object* is an instance of a component.
- A *type library* is a description of a library's exposed classes, interfaces, methods, and properties. In this toolkit, the type library is contained in the OptoSnaploMemMapX.dll file.
- A *UUID* (Universally Unique Identifier, also called a GUID) is a a 128-bit value that uniquely identifies something.

ActiveX Names and IDs

NOTE: Different programming environments may require different information.

File Name: OptoSnaploMemMapX.dll

Type Library UUID: 54D2FA40-E34F-11D2-9707-080009ABC65D

Type Library Name: OptoSnapIoMemMapXLib

MemMap ProgID: OptoSnapIoMemMapX.O22SnapIoMemMapX.1

MemMap Class UUID: 54D2FA50-E34F-11D2-9707-080009ABC65D

MemMap Class Name: OptoSnapIoMemMapX

MemMap Interface UUID: 54D2FA4F-E34F-11D2-9707-080009ABC65D

MemMap Interface Name: IO22SnapIoMemMapX

Streaming ProgID: OptoSnapIoMemMapX.OptoSnapIoStreamX.1

Streaming Class UUID: 54D2F150-E34F-11D2-9707-080009ABC65D

Streaming Class Name: OptoSnapIoStreamX

Streaming Interface UUID: 54D2F14F-E34F-11D2-9707-080009ABC65D

Streaming Interface Name: IO22SnapIoStreamX

Streaming Event Interface UUID: 54D2F148-E34F-11D2-9707-080009ABC65D

Streaming Event Interface Name: _IOptoSnapIoStreamXEvents

Important ActiveX Client Issues

Before deciding to use these components in a particular language or tool, please keep the following in mind.

Some types of ActiveX clients are better suited for use with these components than others. The components are designed to be used in procedural languages, such as Visual Basic and C++. They are not recommended for use in other types of languages, such as graphical or dataflow languages (for exmaple, LabVIEW). Opto 22 recommends Visual Basic 6.0 and Visual Basic for Applications as the best environments for using these components.

Opto 22 is able to support the use of the ActiveX components only in the environments for which examples are provided: Access 2000, Word 2000, Visual Basic 6.0, Internet Explorer 6.0, Visual C++ 6.0, and Borland Delphi 5. See page 69 for more information on these examples.

Also, please note these are ActiveX components, not controls. They do not have any user interface elements or design-time properties. They use methods and events but not properties.

OptoSnapIoMemMapX General Instructions

Each programming environment has a different way of using ActiveX components. Also, environments use different and conflicting terminology. See the environment's documentation for more information on how to use ActiveX components. In general, however, you will need to do the following:

1. Add the component to the project, so the project knows about it.
   - In some environments, you must add a reference to the type library, class, or interface.
   - In others, you must import the type library, class, or interface into a native format. For example, if you are using the component in Visual C++, it will build a C++ wrapper class for accessing the component.
2. Create an instance of the component.

3. Open a connection to the Opto 22 memory-mapped device.
4. Use the methods of the component to communicate with the connected device and check for errors.
5. When done, disconnect from the device .
6. If needed, destroy the instance that was created in step 2.

See "MemMap Example Code" on page 71 for examples of these procedures in Visual Basic.

OptoSnapIoStreamX General Instructions

Common procedures for using the OptoSnapIoStreamX ActiveX component in a program are:

1. Add the component to the project, so the project knows about it.
   - In some environments, you must add a reference to the type library, class, or interface.
   - In others, you must import the type library, class, or interface into a native format. For example, if you are using the component in Visual C++, it will build a C++ wrapper class for accessing the component.
2. Create an instance of the component.
3. Call OpenStreaming() to initialize the stream type, length, and port.
4. Call StartStreamListening() for each I/O unit.

   A new thread is created to listen for incoming UDP packets. Every time a stream packet from a registered device is received, the OnStreamEvent will be called.
5. Depending on the type set in step #3., call the function GetLastStreamStandardBlock(), GetLastStreamStandardBlockEx(), or GetLastStreamCustomBlockEx() every time OnStreamEvent is called.
6. To stop listening for a specified I/O unit, call StopStreamListening() at any time.
7. To add I/O units, call StartStreamListening() at any time.
8. When done, call CloseStreaming().
9. If needed, destroy the instance that was created in step 2.

Using Visual Basic 6.0 or Higher

When you add the component to a Visual Basic project, you must reference the component before using it. To do so, choose Project?References. In the list box, check OptoSnapIoMemMapX 2.0 Type Library. This action lets Visual Basic know that you intend to use the component from within the project.

"MemMap Example Code" on page 71 shows how to create and use the component from within Visual Basic. Several sample programs are also provided, including one (Demo Center.vbp) that demonstrates all aspects of the ActiveX component. The Visual Basic examples use the file OptoENET-IO\DriverToolkit\Examples\ActiveX\VB6\Common\O22SIOMMXUtils.bas, which includes several useful general-purpose functions and error codes.

Using VBA and Microsoft Office

Some Microsoft Office programs include a macro language, Visual Basic for Applications (VBA), which is a subset of Visual Basic. VBA can use the OptoSnapIoMemMapX component in the same way that Visual Basic does. Examples are included of using VBA in Word and Access.

Using VBScript

Visual Basic Scripting Edition, or VBScript, is a subset of the Visual Basic programming language. VBScript is an interpreted language used in World Wide Web browsers and other applications.

Microsoft FrontPage® or Microsoft Visual InterDev® can be used to write VBScript code that runs in an HTML page located on a PC on the same network as the Opto 22 memory-mapped device. An example is included that illustrates using VBScript in a Web page to access the ActiveX component.

If you are programming with VBScript, use the OptoSnapIoMemMap ActiveX component functions that begin with the letter V. These functions use only the variant data type, which is the only data type supported by VBScript. See "Variant Methods" on page 76 for more information.

Using Borland Delphi 5

Borland Delphi 5, a version of the Pascal programming language, may be used with the ActiveX components. To access them, the type library must be imported into a project. See the readme.txt file in the included Delphi example for more information.

Using the ActiveX Components without the Toolkit

The ActiveX component must be installed on a computer before it can be used. If you want to run the examples or other ActiveX files you create on a computer that doesn't have the toolkit installed, you need to register the ActiveX DLL. To do so, follow these steps:

1. Copy the OptoSnapIoMemMapX.dll file to the new computer.
2. Register the OptoSnapIoMemMapX.dll file using the regsvr32.exe program.

   Regsvr32.exe can be found on most Windows systems. For instance, if you copied the ActiveX file to "c:\windows\system", you could register it by executing the command "regsvr32 c:\windows\system\OptoSnapIoMemMapX.dll" from a Command Prompt.

Building the ActiveX Component with Visual C++ 6.0

All Visual C++ 6.0 source code for the ActiveX component is included in the OptoENET-IO\DriverToolkit\Dev\Source\ActiveX directory. This source code may be modified and built by using Visual C++ 6.0.

If the component's interface or functionality is changed, consider changing the library and interface's universally unique identifiers (UUID). These values can be found in the files OptoSnapIoMemMapX.idl and O22SnapIoMemMapX.rgs. Use the GUIDGEN.EXE program included with Visual C++ to generate new UUID values.

Overview of the C++ Classes

The OptoSnapIoMemMap and O22SnapIoStream C++ classes should be usable by any program written in C++ . In fact, these classes are the core of the ActiveX components. The ActiveX components are merely wrappers around the C++ classes.

Examples of using the C++ classes in Windows and Linux are included. These classes should also be usable on other systems with a few modifications. The class uses generic Berkeley-style sockets, which are fairly universal, but some code is still Windows- and Linux-specific. To find system-dependent code, you can search the source code for the strings _WIN32 and _LINUX.

OptoSnapIoMemMap Procedures

Common procedures for using the OptoSnapIoMemMap C++ class in a program are:

1. Add the class to a program by adding the O22SIOMM.cpp and O22SIOUT.cpp files and including the file O22SIOMM.h for the class definition.
2. Either statically or dynamically create an instance of the class.
3. Open a connection to the Opto 22 memory-mapped device.
4. Use the public member functions of the class to communicate with the connected device and check for errors.
5. When done, disconnect from the device.
6. If the instance was created dynamically in step 1, destroy it.

See "MemMap Example Code" on page 71 for examples of these procedures in Visual Basic.

O22SnapIoStream Procedures

Common procedures for using the OptoSnapIoStream C++ class in a program are:

1. Add the class to a program by adding the O22SIOST.cpp and O22SIOUT.cpp files and including the file O22SIOST.h for the class definition.
2. Create an instance of the O22SnapIoStream class.
3. Call SetCallbackFuntions() to initialize several callback functions.
4. Call OpenStreaming() to initialize the stream type, length, and port.
5. Call StartStreamListening() for each I/O unit.

   A new thread is created to listen for incoming UDP packets. Every time a stream packet from a registered I/O unit is received, the Stream Event callback function will be called.
6. Depending on the type set in step #3, call the function GetLastStreamStandardBlockEx() or GetLastStreamCustomBlockEx() every time this Stream Event is called.
7. To stop listening for a specified I/O unit, call StopStreamListening() at any time.
8. To add I/O units, call StartStreamListening() at any time.
9. When done, call CloseStreaming()

Using Visual C++ with the Core C++ Code

The core C++ classes are found in the OptoENET-IO\DriverToolkit\Dev\Source\C++ directory. To use them in a Visual C++ project, you'll need to do the following:

1. Add the appropriate files to the projects. Turn off precompiled headers for these files by adjusting the settings in the compiler.
2. Link Windows Sockets into your project. Add the library file ws2_32.lib for WinSock 2.0 to your project linker settings. These files should be included with your Visual C++ compiler.

Using Linux g++ with the Core C++ Code

The Red Hat 7.2 distribution of Linux for Intel and the GNU g++ compiler were used to test the core C++ classes.

The class files are found in the OptoENET-IO\DriverToolkit\Dev\Source\C++ directory. Simply include them in your project. Add a "-D_LINUX" command to the g++ prompt to make sure _LINUX is defined.

Examples Included

O22SnapIoMemMapX ActiveX Examples

The OptoMMP Communication Toolkit includes the following ActiveX examples of using the O22SnapIoMemMapX ActiveX component.

Environment	Example Name	Use with	Description
Access 2000	Demo Center	Demo Center	Uses Visual Basic for Applications to update entries in an Access database.
Word 2000	Demo Center	Demo Center	Uses Visual Basic for Applications to update a report about the Demo Center.
	Digital Bank	Any rack with digital points	Uses Visual Basic for Applications to update a letter with the values from the device's digital bank area.
Visual Basic 6	Demo Center	Demo Center	Uses virtually every aspect of OptoSnapIoMemMapX component. It shows how Visual Basic (or any similar programming environment, such as Visual C++) can be used to create a Human Machine Interface (HMI) that communicates with an Opto 22 memory-mapped device. It also shows how Visual Basic can be used to provide control logic and send e-mail reports.
	Digital Bank	Any rack with digital points	Reads and writes to the bank of digital points.
	Scatch Pad	Any SNAP Ultimate I/O unit	Reads and writes values to the entire Scratch Pad area. Useful as a standalone tool.

Environment	Example Name	Use with	Description
Internet Explorer 6.0	Digital Bank	Any rack with digital points	Uses VBScript and a timer to dynamically update a Web page located on a PC on the same network as the memory-mapped device. If the user's machine does not have the ActiveX component installed, Internet Explorer automatically installs it.
Visual C++ 6	Digital Bank	Any rack with digital points	Uses Microsoft Foundation Classes (MFC) to use the OptoSnapIoMemMapX component. It is very similar to the Visual Basic Digital Bank example.
Borland Delphi 5	Digital Bank	Any rack with digital points	Reads and writes to the bank of digital points.

O22SnapIoStreamX ActiveX Example

The toolkit includes the following ActiveX example of using the O22SnapIoStreamX ActiveX component.

Environment	Example Name	Use with	Description
Visual Basic 6	Streaming	Any memory-mapped device with streaming turned on	Captures and displays standard or custom stream packets.

O22SnapIoMemMap C++ Class Examples

The toolkit also includes the following examples, which use the core C++ code. See "Using Visual C++ with the Core C++ Code" on page 69 for more information.

Environment	Example Name	Use with	Description
Visual C++ 6	Digital Bank	Any rack with digital points	Identical to the Visual C++ 6.0 ActiveX Digital Bank example, except that it uses the core C++ class, O22SnapIoMemMap, instead of the OptoSnapIoMemMapX ActiveX component.
	Digital Bank Console	Any rack with digital points	A simple Win32 console application. It takes one command-line parameter, an IP address, and prints out the state of the digital bank area of the device at that address.
	Eiocon	Any rack	Creates a small shell environment that lets a user issue commands to read and write to a memory-mapped device.
	Eiocl	Any rack	A command line tool for reading or writing values to a device. Can also process a file of commands. Useful as a standalone tool.
Linux	Eiocon	Any rack	Creates a small shell environment that lets a user issue commands to read and write to a device.
	Eiocl	Any rack	A command line tool for reading or writing values to a device. Can also process a file of commands. Useful as a standalone tool.

O22SnapIoStream C++ Class Examples

The toolkit also includes the following examples, which use the core C++ code. See "Using Visual C++ with the Core C++ Code" on page 69 for more information.

Environ-ment	Example Name	Use with	Description
Visual C++ 6	Eiocon	Any memory-mapped device with streaming turned on	Identical to the Eiocon MemMap example shown above. Includes a mode (the "ls" command) for catching and printing stream packets.
Linux	Eiocon	Any memory-mapped device with streaming turned on	Identical to the Eiocon MemMap example shown above. Includes a mode (the "ls" command) for catching and printing stream packets.

MemMap Example Code

This section gives a few examples of how to use the OptoSnapIoMemMapX ActiveX component to connect to and communicate with an Opto 22 memory-mapped device. The same principles would apply to the OptoSnapIoMemMap C++ class. All example code is written with Microsoft Visual Basic. The interesting code is in **boldface**. For more example code, see the example files listed in the previous section.

Create the OptoSnapIoMemMapX Component

Define a global instance of the component and create it in a function. For example:

```
Public gBrain As OptoSnapIoMemMapX

Sub Main()
  Set gBrain = New OptoSnapIoMemMapX
End Sub
```

Open a Connection to the Device

Use methods OpenEnet() and IsOpenDone() to create a connection to the device. For example:

```
Dim nResult As Long
  ' Attempt to open the brain...
  nResult = gBrain.OpenEnet("10.192.54.0", 2001, 10000, 1)

  ' Check the result
  If (nResult = SIOMM_OK) Then

    ' Keep calling IsOpenDone() until we succeed, time-out, or error
    nResult = gBrain.IsOpenDone()
    While (nResult = SIOMM_ERROR_NOT_CONNECTED_YET)

      ' let other tasks get some CPU
      DoEvents

      ' Check the status of the open operation
```

```
    nResult = gBrain.IsOpenDone() ' keep trying
  Wend
End If
```

Close the Connection to the Device

Use the Close () method. For example:

```
gBrain.Close ()
```

Configure Points

Use the SetPtConfiguration() method to configure points. See the tables in the section "Configuring I/O Points" on page 22 for point type values.

```
Dim nResult As Long

' Digital In
nResult = gBrain.SetPtConfiguration(0, &H100&, 1, 0, 0, 0, 0)
If (nResult <> SIOMM_OK) Then
  HandleError (nResult)
  Exit Sub
End If

' Digital Out
nResult = gBrain.SetPtConfiguration(4, &H180&, 0, 0, 0, 0, 0)
If (nResult <> SIOMM_OK) Then
  HandleError (nResult)
  Exit Sub
End If

' Analog Out (SNAP-AOV25)
nResult = gBrain.SetPtConfiguration(8, &HA5&, 0, 0, 0, 0, 10)  ' AOV25
If (nResult <> SIOMM_OK) Then
  HandleError (nResult)
  Exit Sub
End If

' Analog In (SNAP-AIV)
nResult = gBrain.SetPtConfiguration(12, &H6&, 0, 0, 0, 0, 20)  ' AIV
If (nResult <> SIOMM_OK) Then
  HandleError (nResult)
  Exit Sub
End If

' Update the error display
HandleError (nResult)
```

Set an Output Based on an Input

This example shows how to read and write analog and digital points. It sets a digital output based on a digital input point. It does the same for analog points.

```
Dim nResult As Long
  Dim nPtState As Long
  Dim nPtValue As Single

  ' Get the state of digital input
  nResult = gBrain.GetDigPtState(nDigInPoint, nPtState)

  ' if no error, set the digital output
  If (nResult = SIOMM_OK) Then
    nResult = gBrain.SetDigPtState(nDigOutPoint, nPtState)
  End If

  ' Deal with any errors and update the error display
  Call HandleError(nResult)
' Get the value of the analog input
  nResult = gBrain.GetAnaPtValue(nAnaInPoint, nPtValue)

  ' if no error, set the analog output
  If (nResult = SIOMM_OK) Then
    nResult = gBrain.SetAnaPtValue(nAnaOutPoint, nPtValue)
  End If

  ' Deal with any errors and update the error display
  Call HandleError(nResult)
```

Streaming Example Code

This section gives a few examples of how to use the OptoSnapIoStreamX ActiveX component to listen for and process stream packets from an Opto 22 memory-mapped device. The same principles apply to the OptoSnapIoStream C++ class. All example code is written with Microsoft Visual Basic. The interesting code is in **boldface**. For more example code, see the files listed in "Examples Included" on page 69.

Create the OptoSnapIoStreamX Component

Define a global instance of the component and create it in a function. For example:

```
Public WithEvents gBrain As OptoSnapIoStreamX

Sub Main()
  Set gBrain = New OptoSnapIoStreamX
End Sub
```

Open a Connection to the Device

Use the OpenStreaming() method to prepare for listening to stream packets. For example:

```
Dim nResult As Long
```

```
' Open for standard streaming
nResult = gBrain.OpenStreaming(1, 0, PortNumberEdit.Text)

' Check for error
If (nResult <> SIOMM_OK) Then
  HandleError (nResult)
End If
```

Start Listening to a Device

Use the StartStreamingListening() method to start listening for stream packets from the specified device. For example:

```
' Start listening
If (nResult = SIOMM_OK) Then
  nResult = gBrain.StartStreamListening("10.20.30.40", 2000)
End If
```

Handle Stream Events

Use the OnStreamEvent() event to handle caught stream packets. For example:

```
Private Sub gBrain_OnStreamEvent(ByVal nIpAddress As Long,
                                 ByVal nStatus As Long)
  Dim nResult As Long

  ' Check the status of the event
  If (nStatus <> SIOMM_OK) Then
    ' A timeout is the most likely error
    HandleError (nStatus)
  Else
    Dim StreamBlock As SIOMM_StreamStandardBlock

    ' Get the last stream block
    nResult = gBrain.GetLastStreamStandardBlockEx(StreamBlock)

    If (nResult = SIOMM_OK) Then
      ' If no error, do something with the data
    Else
      ' Check the result
      HandleError (nResult)
    End If

  End If
End Sub
```

Stop Listening and Clean Up

Use the StopStreamListening() and CloseStreaming() methods to clean up. For example:

```
' Stop listening
nResult = gBrain.StopStreamListening("10.20.30.40")

' Close the connection
nResult = gBrain.CloseStreaming()
```

Interface Introduction

Here are the interfaces for the ActiveX components and C++ classes. Since the interfaces are very similiar and the components may be used from different languages, the information is presented in a general format.

Basic Data Types

The following data types are used. They may have different names in different languages.

- `byte:` 8-bit signed integer
- `short:` 16-bit signed integer
- `long:` 32-bit signed integer
- `string:` string
- `float:` 32-bit float
- `by_array:` an array of bytes
- `l_array:` an array of long integers
- `f_array:` an array of floats

Structures

Since some ActiveX containers, such as Visual Basic, support predefined structures rather than just the basic data types, several methods also support structures. These structures can be easier to use. The names of these methods all end with Ex, and they are similar to other methods.

For example, in the table on page 80, compare GetDigPtReadArea to GetDigPtReadAreaEx. The former takes individual parameters, whereas the latter takes the following structure:

```
struct SIOMM_DigPointReadArea
{
  long  nState;          // bool
  long  nOnLatch;        // bool
  long  nOffLatch;       // bool
  long  nCounterState;   // bool
  long  nCounts;         // unsigned int
};
```

See the file OptoENET-IO\DriverToolkit\Source\C++\O22STRCT.h for structure definitions. These definitions are included in the ActiveX type library.

[In] and [Out] Parameter Types

Parameters marked as [in] are not changed by the method. Parameters marked as [out] can be changed by the method. The exact parameter mechanism depends upon the programming environment being used. For instance, in C++, all [out] parameters use pointers.

Variant Methods

A variant is a special data type that can represent values of many different types, such as integers, floats, Booleans, strings, or pointers. A variant represents only one type at a time. Some languages, such as Visual Basic, make extensive use of the variant data type. VBScript supports only variants.

The OptoSnapIoMemMapX interface includes two versions of the most commonly used methods. The regular versions have parameters and return values that are strongly typed, with types such as long integer and float. The variant versions have parameters and return values of the variant type only, because some languages, such as VBScript, support only the variant data type. All variant-only methods start with the letter V.

We recommend you use the non-variant methods whenever possible to provide the strong typing.

Even though a parameter is a variant, the variant type must match that of the corresponding regular method. For example, the method VSetStatusOperation (VARIANT nOpCode) takes one parameter, nOpCode. The variable nOpCode must be a long integer at the time when VSetStatusOperation is called. In VBScript, use CLng() to force a variant to be a long integer.

Example:

```
gBrain2.VSetStatusOperation(CLng(OpCode))
```

Return Values

All methods return a long integer value. The following table lists possible return values.

Return Value Name	Code	Indicates
SIOMM_OK	1	The method succeeded.
SIOMM_ERROR	-1	A general error occurred.
SIOMM_TIME_OUT	-2	The method call timed out.
SIOMM_ERROR_NO_SOCKETS	-3	Sockets interface (wsock32.dll) could not be found.
SIOMM_ERROR_CREATING_SOCKET	-4	A socket could not be created.
SIOMM_ERROR_CONNECTING_SOCKET	-5	A socket connection could not be made. Not used at this time.
SIOMM_ERROR_RESPONSE_BAD	-6	The response from the SNAP device was bad. The most likely reason is that communication responses are out of sync. This may indicatef a problem with the network
SIOMM_ERROR_NOT_CONNECTED_YET	-7	Returned by IsOpenDone() to indicate the open process isn't done yet but has not timed out.
SIOMM_ERROR_OUT_OF_MEMORY	-8	Out of memory.
SIOMM_ERROR_NOT_CONNECTED	-9	No successful connection exists to a device via the OptoEnet() or OptoEnet2() methods.

The Opto 22 device itself may also generate errors. If a command to the device is unsuccessful, the method attempts to read the error generated by the device.

The return values in the following table correspond with the error codes shown on page 109.

Return Value Name	Code (Hex)	Indicates
SIOMM_BRAIN_ERROR_UNDEFINED_CMD	E001	Undefined command
SIOMM_BRAIN_ERROR_INVALID_PT_TYPE	E002	Invalid point type
SIOMM_BRAIN_ERROR_INVALID_FLOAT	E003	Invalid float
SIOMM_BRAIN_ERROR_PUC_EXPECTED	E004	A Powerup Clear is expected.
SIOMM_BRAIN_ERROR_INVALID_ADDRESS	E005	A command attempted to read/write an invalid memory address.
SIOMM_BRAIN_ERROR_INVALID_CMD_LENGTH	E006	Invalid command length
SIOMM_BRAIN_ERROR_RESERVED	E007	Reserved
SIOMM_BRAIN_ERROR_BUSY	E008	The device is busy.
SIOMM_BRAIN_ERROR_CANT_ERASE_FLASH	E009	Flash memory cannot be erased.
SIOMM_BRAIN_ERROR_CANT_PROG_FLASH	E00A	Flash memory cannot be programmed.
SIOMM_BRAIN_ERROR_IMAGE_TOO_SMALL	E00B	Downloaded image is too small.
SIOMM_BRAIN_ERROR_IMAGE_CRC_MISMATCH	E00C	Image CRC mismatch
SIOMM_BRAIN_ERROR_IMAGE_LEN_MISMATCH	E00D	Image length mismatch
SIOMM_BRAIN_ERROR_FEATURE_NOT_IMPL	E00E	Feature not implemented
SIOMM_BRAIN_ERROR_WATCHDOG_TIMEOUT	E00F	Communications watchdog timeout

OptoSnapIoMemMap Interface

Connection Methods

Use the following methods to connect to an Opto 22 memory-mapped device.

Method	Description

OpenEnet[V]
`[in] string strIpAddressArg`
`[in] long nPort`
`[in] long nOpenTimeOutMS`
`[in] long nAutoPUC`
	Starts the connection process to a device via TCP/IP. Use the IsOpenDone() method to check whether the open process is completed.
	bstrIpAddressArg is a string representing the IP address of the memory-mapped device, such as 10.192.54.1. The format of the string is not verified. NOTE: You must use the IP address, not the host name, of the device.
	nPort is the IP port used to connect to the device. Default is 2001.
	nOpenTimeOutMS is the timeout period in milliseconds for the open process. A different timeout is used for normal communications. See SetCommOptions below.
	If nAutoPUC is a non-zero value, the Powerup Clear flag on the device is automatically set. If zero, you must send a Powerup Clear by using the SetStatusOperation(1) method. (See the Status Write Methods table on page 86.)

Method	Description

OpenEnet2
`[in] string strIpAddressArg`
`[in] long nPort`
`[in] long nOpenTimeOutMS`
`[in] long nAutoPUC`
`[in] long nConnectionType`
	Same as OpenEnet() above, except it allows TCP or UDP to be specified.
	nConnectionType is 1 for TCP and 2 for UDP.

IsOpenDone[V]	Use after a call to OpenEnet() or OpenEnet2() to check whether the connection process is completed. Will return `SIOMM_ERROR_NOT_CONNECTED_YET` (-7) until the connection is completed, a timeout occurs, or another error occurs. Using this function in a program provides a way for the user to cancel the connection process. See the Visual Basic examples for more information.
Close[V]	Closes the connection to the device.

SetCommOptions[V]
`[in] long nTimeOutMS`
`[in] long nReserved`
	Sets options for communication with the device.
	nTimeOutMS is the timeout value in milliseconds for communications to the device. **nReserved** is not currently used and should be set to zero.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Return values are shown on page 76.

Point Configuration Methods

Use the following methods to configure I/O points. For more information, see "Referencing I/O Points" on page 15 and "Configuring I/O Points" on page 22.

Method	Description

ConfigurePoint[V]
`[in] long nPoint`
`[in] long nPointType`
	Configures a point with the specified point type (see page 15 for point numbers and page 22 for point types).

SetDigPtConfiguration
`[in] long nPoint`
`[in] long nPointType`
`[in] long nFeature`
	Configures a digital point with the specified point type and feature.
	For digital feature values, see page 114.

SetAnaPtConfiguration
`[in] long nPoint`
`[in] long nPointType`
`[in] float fOffset`
`[in] float fGain`
`[in] float fHiScale`
`[in] float fLoScale`
	Configures an analog point with the specified point type, offset, gain, and scale. See "Configuring I/O Points" on page 22.

SetPtConfiguration[V] `(ActiveX Only)`
`[in] long nPoint`
`[in] long nPointType`
`[in] long nFeature`
`[in] float fOffset`
`[in] float fGain`
`[in] float fLoScale`
`[in] float fHiScale`
`[in] float fWatchdogValue`
`[in] long nWatchdogEnabled`
	Configures a point with the specified point type, feature, offset, gain, scale, and watchdog.
	See "Configuring I/O Points" on page 22. For digital feature values, see page 114.

Method	Description
SetPtConfiguration2 (ActiveX Only) [in] long nPoint [in] long nPointType [in] long nFeature [in] float fOffset [in] float fGain [in] float fLoScale [in] float fHiScale [in] float fWatchdogValue [in] long nWatchdogEnabled [in] float fFilterWeight [in] string bstrName	Same as SetPtConfiguration, except it adds filter weight and name parameters.
SetPtConfigurationEx (C++ Only) [in] long nPoint [in] SIOMM_PointConfigArea Data	Same as SetPtConfiguration, except this method uses a structure.
SetPtConfigurationEx2 (C++ Only) [in] long nPoint [out] SIOMM_PointConfigArea2 pData	Same as SetPtConfiguration2, except this method uses a structure.
SetPtWatchdog [in] long nPoint [in] float fWatchdogValue [in] long nWatchdogEnabled	Sets the watchdog value and enable flag for the given point. See page 34 for more information on watchdogs.
GetPtConfiguration[V] (ActiveX Only) [in] long nPoint [out] long pnModuleType [out] long pnPointType [out] long pnFeature [out] float pfOffset [out] float pfGain [out] float pfLoScale [out] float pfHiScale [out] float pfWatchdogValue [out] long pnWatchdogEnabled	Retrieves a point's module type, point type, feature, offset, gain, scale, and watchdog.
GetPtConfiguration2 (ActiveX Only) [in] long nPoint [out] long pnModuleType [out] long pnPointType [out] long pnFeature [out] float pfOffset [out] float pfGain [out] float pfLoScale [out] float pfHiScale [out] float pfWatchdogValue [out] long pnWatchdogEnabled [out] float pfFilterWeight [out] string pstrName	Same as GetPtConfiguration, except it adds filter weight and name parameters.
GetPtConfigurationEx (C++ Only) [in] long nPoint [out] SIOMM_PointConfigArea pData	Same as GetPtConfiguration, except this method uses a structure.
GetPtConfigurationEx2 (C++ Only) [in] long nPoint [out] SIOMM_PointConfigArea2 pData	Same as GetPtConfiguration2, except this method uses a structure.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Return values are shown on page 76.

Digital Point Methods

Use the following methods to read and write information about individual digital points. For more information on digital point features, see page 29, "Digital Point Read—Read Only" on page 149, and "Digital Point Write—Read/Write" on page 150.

Read Methods	Description
GetDigPtState[V] [in] long nPoint, [out] long pnState	Retrieves the specified digital point's state. **pnState** values: 0 = off, non-zero = on
GetDigPtOnLatch[V] [in] long nPoint [out] long pnState	Retrieves the specified digital point's on-latch state.
GetDigPtOffLatch[V] [in] long nPoint [out] long pnState	Retrieves the specified digital point's off-latch state.
GetDigPtCounterState[V] [in] long nPoint [out] long pnState	Retrieves the specified digital point's counter state.
GetDigPtCounts[V] [in] long nPoint [out] long pnValue	Retrieves the specified digital point's counter value.
GetDigPtReadArea[V] (ActiveX Only) [in] long nPoint [out] long pnState [out] long pnOnLatchState [out] long pnOffLatchState [out] long pnCounterActive [out] long pnCounterData	Retrieves the specified digital point's state, on-latch state, off-latch state, counter state, and counter value.
GetDigPtReadAreaEx [in] long nPoint [out] SIOMM_DigPointReadArea pPointData	Uses a structure to retrieve the specified digital point's state, on-latch state, off-latch state, counter state, and counter value.
ReadClearDigPtCounts[V] [in] long nPoint [out] long pnValue	Reads and clears the specified digital point's counter value.
ReadClearDigPtOnLatch[V] [in] long nPoint [out] long pnState	Reads and clears the specified digital point's on-latch state.
ReadClearDigPtOffLatch[V] [in] long nPoint [out] long pnState	Reads and clears the specified digital point's off-latch state.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Write Methods	Description
SetDigPtState[V] [in] long nPoint [in] long nState	Sets the specified digital point's state. **pnState** values: 0 = off, non-zero = on
SetDigPtCounterState[V] [in] long nPoint [in] long nState	Sets the specified digital point's counter state.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Return values are shown on page 76.

Digital Bank Methods

Use the following methods to read and write information in the digital bank area of the IEEE 1394-based protocol. For more information, see "Mask Data" on page 58, "Digital Bank Read—Read Only" on page 147, and "Digital Bank Write—Read/Write" on page 147.

Read Methods	Description
GetDigBankPointStates[V] [out] long pnPts63to32 [out] long pnPts31to0	Retrieves the state of the entire digital bank as a mask.
GetDigBankOnLatchStates[V] [out] long pnPts63to32 [out] long pnPts31to0	Retrieves the on-latch state of the entire digital bank as a mask.
GetDigBankOffLatchStates[V] [out] long pnPts63to32 [out] long pnPts31to0	Retrieves the off-latch state of the entire digital bank as a mask.
GetDigBankActCounterStates[V] [out] long pnPts63to32 [out] long pnPts31to0	Retrieves the active counter state of the entire digital bank as a mask.
GetDigBankReadArea[V] (ActiveX Only) [out] long pnStatePts63to32 [out] long pnStatePts31to0 [out] long pnOnLatchStatePts63to32 [out] long pnOnLatchStatePts31to0 [out] long pnOffLatchStatePts63to32 [out] long pnOffLatchStatePts31to0 [out] long pnActiveCountersPts63to32 [out] long pnActiveCountersPts31to0	Retrieves the point state, on-latch state, off-latch state, and active counter state of the entire digital bank.
GetDigBankReadAreaEx [out] SIOMM_DigBankReadArea pBankData	Uses a structure to retrieve the point state, on-latch state, off-latch state, and active counter state of the entire digital bank.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Write Methods	Description
SetDigBankOnMask[V] [in] long nPts63to32 [in] long nPts31to0	Uses the bitmask to turn digital points on.
SetDigBankOffMask[V] [in] long nPts63to32 [in] long nPts31to0	Uses the bitmask to turn digital points off.
SetDigBankPointStates[V] [in] long nPts63to32 [in] long nPts31to0 [in] long nMask63to32 [in] long nMask31to0	Sets the state of the digital bank from the **nPts63to32** and **nPts31to0** parameters. Only those points set in the **nMask63to32** and **nMask31to0** parameters are affected.
SetDigBankActCounterMask[V] [in] long nPts63to32 [in] long nPts31to0	Uses the bitmask to activate counters.
SetDigBankDeactCounterMask[V] [in] long nPts63to32 [in] long nPts31to0	Uses the bitmask to deactivate counters.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Return values are shown on page 76.

Analog Point Methods

Use the following methods to read and write information about individual analog points. For more information on analog point features, see page 29, "(Old) Analog Point Read—Read Only" on page 151, "(Old) Analog Point Write—Read/Write" on page 152, "(Old) Analog Point Calculation and Set—Read Only" on page 165, and "(Old) Analog Read and Clear/Restart—Read Only" on page 167.

Read Methods	Description
GetAnaPtValue[V] [in] long nPoint [out] float pfValue	Retrieves the value of the specified analog point.
GetAnaPtCounts[V] [in] long nPoint [out] float pfValue	Retrieves the counts of the specified analog point.
GetAnaPtMinValue[V] [in] long nPoint [out] float pfValue	Retrieves the minimum value of the specified point.
GetAnaPtMaxValue[V] [in] long nPoint [out] float pfValue	Retrieves the maximum value of the specified point.
GetAnaPtTpoPeriod [in] long nPoint [out] float pfValue	Retrieves the TPO period for the specified point. See the SetAnaPtTpoPeriod() method below for more information.

Read Methods	Description
GetAnaPtReadArea[V] (ActiveX Only) [in] long nPoint [out] float pfValue [out] float pfCounts [out] float pfMinValue [out] float pfMaxValue	Retrieves the specified point's value, counts, minimum value, and maximum value.
GetAnaPtReadAreaEx [in] long nPoint [out] SIOMM_AnaPointReadArea pPointData	Uses a structure to retrieve the specified point's value, counts, minimum value, and maximum value.
ReadClearAnaPtMinValue[V] [in] long nPoint [out] float pfValue	Reads and clears the specified point's minimum value.
ReadClearAnaPtMaxValue[V] [in] long nPoint [out] float pfValue	Reads and clears the specified analog point's maximum value.
CalcSetAnaPtOffset[V] [in] long nPoint [out] float pfValue	Calculates, sets, and retrieves the specified analog point's offset.
CalcSetAnaPtGain[V] [in] long nPoint [out] float pfValue	Calculates, sets, and retrieves the specified analog point's gain.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Write Methods	Description
SetAnaPtValue[V] [in] long nPoint [in] float fValue	Sets the specified analog point's value.
SetAnaPtCounts[V] [in] long nPoint [in] float fValue	Sets the specified analog point's counts.
SetAnaPtTpoPeriod [in] long nPoint [in] float fValue	Sets the specified analog point's TPO period in units of time in seconds. Valid range: 0.25 to 64 seconds, in 0.25 steps.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Return values are shown on page 76.

Analog Bank Methods

Use the following methods to read and write information in the analog bank area of the memory map. For more information, see "Analog Bank Read—Read Only" on page 148, "Analog Bank Write—Read/Write" on page 148, and "IEEE Float Data" on page 60.

Read Methods	Description
GetAnaBankValuesEx [out] SIOMM_AnaBank pBankData	Retrieves the values of the entire analog bank.

Read Methods	Description
GetAnaBankCountsEx [out] SIOMM_AnaBank pBankData	Retrieves the counts of the entire analog bank.
GetAnaBankMinValuesEx [out] SIOMM_AnaBank pBankData	Retrieves the minimum values of the entire analog bank.
GetAnaBankMaxValuesEx [out] SIOMM_AnaBank pBankData	Retrieves the maximum values of the entire analog bank.

Write Methods	Description
SetAnaBankValuesEx [in] SIOMM_AnaBank pBankData	Sets the values of the entire analog bank.
SetAnaBankCountsEx [in] SIOMM_AnaBank pBankData	Sets the counts of the entire analog bank.

Return values are shown on page 76.

Status Methods

Use the following methods to read from the status area of the device. For more information about the status area of the memory map, see "Status Area Read—Read Only" on page 119.

Read Methods	Description
GetStatusVersion[V] (ActiveX Only) [out] long pnMapVer [out] long pnLoaderVersion [out] long pnKernelVersion	Retrieves version information about the device's memory map, loader, and kernel.
GetStatusVersionEx (C++ Class Only) [out] SIOMM_StatusVersion pVersionData	Same as GetStatusVersion, except this method uses a structure.
GetStatusHardware[V] (ActiveX Only) [out] long pnIoUnitType [out] byte pbyHwdVerMonth [out] byte pbyHwdVerDay [out] short pwHwdVerYear [out] long pnRamSize	Retrieves hardware information about the device's unit type, hardware revision date, and quantity of bytes of installed RAM.
GetStatusHardware2 (ActiveX Only) [out] long pnIoUnitType [out] byte pbyHwdVerMonth [out] byte pbyHwdVerDay [out] short pwHwdVerYear [out] long pnRamSize [out] string pbstrPartNumber	Same as GetStatusHardware, except it adds a part number parameter.
GetStatusHardwareEx (C++ Only) [out] SIOMM_StatusHardware pHardwareData	Same as GetStatusHardware, except this method uses a structure.
GetStatusHardwareEx2 (C++ Only) [out] SIOMM_StatusHardware2 pHardwareData	Same as GetStatusHardware2, except this method uses a structure.

Read Methods	Description

`GetStatusNetwork`[V] `(ActiveX Only)`
`[out] short pwMACAddress0`
`[out] short pwMACAddress1`
`[out] short pwMACAddress2`
`[out] long pnTCPIPAddress`
`[out] long pnSubnetMask`
`[out] long pnDefGateway`
	Retrieves network information about the device's MAC address, TCP/IP address, subnet mask, and default gateway.

`GetStatusNetwork2V` `(ActiveX Only)`
`[out] short pwMACAddress0`
`[out] short pwMACAddress1`
`[out] short pwMACAddress2`
`[out] long pnTCPIPAddress`
`[out] long pnSubnetMask`
`[out] long pnDefGateway`
`[out] long pnTcpIpMinRtoMS`
`[out] long pnInitialRtoMS`
`[out] long pnTcpRetries`
`[out] long pnTcpIdleTimeout`
`[out] long pnEnetLateCol`
`[out] long pnEnetExcessiveCol`
`[out] long pnEnetOtherErrors`
	Same as GetStatusNetwork, expect it adds parameters for TCP/IP minimum RTO, initial RTO, TCP retries, TCP idle session timeout, and Ethernet errors.

`GetStatusNetworkEx` `(C++ Only)` `[out] SIOMM_StatusNetwork pNetworkData`	Same as GetStatusNetwork, except this method uses a structure.
`GetStatusNetworkEx2` `(C++ Only)` `[out] SIOMM_StatusNetwork2 pNetworkData`	Same as GetStatusNetwork2, except this method uses a structure.
`GetStatusPUC`[V] `[out] long pnPUCFlag`	Retrieves the Powerup Clear flag. 0=okay, anything else means a Powerup Clear is needed. See SetStatusOperation() for how to send a Powerup Clear command.
`GetStatusLastError`[V] `[out] long pnErrorCode`	Retrieves the last error. See "Error Codes" on page 109 for more information.
`GetStatusBootpAlways`[V] `[out] long pnBootpAlways`	Retrieves the BootP request flag. 0 = Send BootP only if device's IP address is 0.0.0.0 1 = Send BootP always
`GetStatusDegrees`[V] `[out] long pnDegrees`	Retrieves the temperature degrees unit flag. 1 = degrees F 0 = degrees C
`GetStatusWatchdogTime` `[out] long pnTimeMS`	Retrieves the watchdog time in milliseconds.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Use the following methods to write to the status area of the device. For more information about the status area of the memory map, see "Status Area Write—Read/Write" on page 127.

Write Methods	Description
SetStatusOperation[V] [in] long nOpCode	Sends an operation code to the memory-mapped device. 0x0001 - Send Powerup Clear 0x0002 - Reset to defaults 0x0003 - Store to flash 0x0004 - Erase flash configuration 0x0005 - Reset hardware See address FFFFF0380000 on page 127 for more information.
SetStatusBootpRequest[V] [in] long nFlag	Sets the device to send BootP request when turned on. 0 (default) = Send BootP request only if device's IP address is 0.0.0.0. 1 = Always send BootP request when device is turned on.
SetStatusDegrees[V] [in] long nDegFlag	Sets the temperature degrees unit flag. 1 = degrees F; 0 = degrees C
SetStatusWatchdogTime[V] [in] long nTimeMS	Sets watchdog time in milliseconds. 0 disables the watchdog.

[V] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Return values are shown on page 76.

Time/Date Methods

Use the following methods to read from or write to the time and date area of the memory-mapped device. For more information about the time and date area of the memory map, see "Date and Time Configuration—Read/Write" on page 127.

Read Methods	Description
GetDateTime [out] string pbstrDateTime	Retrieves the device's time and date as a string in the format YYYY-MM-DD HH:MM:SS.00
Write Methods	**Description**
SetDateTime [in] string bstrDateTime	Sets the device's time and date as a string in the format YYYY-MM-DD HH:MM:SS.00

Serial Module Configuration Methods

Use the following methods to read and write information in the serial module configuration area of the memory map. For more information, see "Serial Module Configuration—Read/Write" on page 134.

Read Methods	Description
GetSerialModuleConfigurationEx [in] long nSerialPort [out] O22_SIOMM_SerialModuleConfigArea pConfigData	Retrieves the configuration for the given serial port.

Read Methods	Description
Write Methods	**Description**
SetSerialModuleConfigurationEx [in] long nSerialPort [in] O22_SIOMM_SerialModuleConfigArea pConfigData	Sets the configuration for the given serial port.

Return values are shown on page 76.

Streaming Methods

Use the following methods to read and write information in the stream configuration and data areas of the memory map. For more information, see "Streaming Configuration—Read/Write" on page 146 and "Streaming—Read Only" on page 168. For information on receiving and processing stream packets, see the General Instructions on page 66 and also see page 52.

Read Methods	Description
GetStreamConfiguration [out] long pnOnFlag [out] long pnIntervalMS [out] long pnPort [out] long pnIoMirroringEnabled [out] long pnStartAddress [out] long pnDataSize	Retrieves current streaming configuration information. Also see GetStreamTarget bellow.
GetStreamTarget [in] long nTarget [out] long pnIpAddressArg	Retrieves the IP address for the given target. There are eight possible targets.
GetStreamReadArea [in] f_array fAnalogValues [in] l_array nDigPointFeature [out] long pnStatePts63to32 [out] long pnStatePts31to0 [out] long pnOnLatchStatePts63to32 [out] long pnOnLatchStatePts31to0 [out] long pnOffLatchStatePts63to32 [out] long pnOffLatchStatePts31to0 [out] long pnActiveCountersPts63to32 [out] long pnActiveCountersPts31to0	Reads the stream data area. This is an excellent function for reading all the basic I/O point information in one method call.
GetStreamReadAreaEx [out] O22_SIOMM_StreamStandardBlock pData	Same as GetStreamReadArea, except this method uses a structure.

Write Methods	Description
SetStreamConfiguration [in] long nOnFlag [in] long nIntervalMS [in] long nPort [in] long nIoMirroringEnabled [in] long nStartAddress [in] long nDataSize	Sets streaming configuration information. Also see SetStreamTarget below.

Write Methods	Description
SetStreamTarget [in] long nTarget [in] BSTR bstrIpAddressArg	Sets the IP address for the given target. There are eight possible targets.

Return values are shown on page 76.

Scratch Pad Read/Write Methods

Use the following methods to read or write data in the Scratch Pad area of the device. SNAP PAC R-series and S-series, SNAP-LCE, and SNAP Ultimate controllers contain all Scratch Pad areas; SNAP Ethernet brains contain only Scratch Pad bits; SNAP Simple brains and E1/E2 brain boards do not have any Scratch Pad areas. For more information about the Scratch Pad area of the memory map, see page 163.

Note that some methods differ for Active X and C++.

Read Methods	Description
GetScratchPadBitArea [out] long pnBits63to32 [out] long pnBits31to0	Reads the state of the Scratch Pad bit area.
GetScratchPadIntegerArea (ActiveX only) [in] long nStartIndex [out] O22_SIOMM_ScratchPadIntegerBlock pBlock	Reads a block of 256 integers from the Scratch Pad integer area.
GetScratchPadIntegerArea (C++ only) [in] long nStartIndex [in] long nLength [out] l_array pnData	Reads an array of up to 256 integers from the Scratch Pad integer area. **nLength** must be between 1 and 256.
GetScratchPadFloatArea (ActiveX only) [in] long nStartIndex [out] O22_SIOMM_ScratchPadFloatBlock pBlock	Reads a block of 256 floats from the Scratch Pad float area.
GetScratchPadFloatArea (C++ only) [in] long nStartIndex [in] long nLength [out] f_array * pfData	Reads an array of up to 256 floats from the Scratch Pad float area. **nLength** must be between 1 and 256.
GetScratchPadStringArea (ActiveX only) [in] long nStartIndex [out] O22_SIOMM_ScratchPadStringBlock pBlock	Reads a block of 8 strings from the Scratch Pad string area.
GetScratchPadStringArea (C++ only) [in] long nStartIndex [in] long nLength, [out] SIOMM_ScratchPadString pStringData	Reads a block of up to 8 strings from the Scratch Pad string area. **nLength** must be between 1 and 8.

Write Methods	Description
SetScratchPadBitArea [in] long nBits63to32 [in] long nBits31to0	Sets the state of the Scratch Pad bit area.

Write Methods	Description
SetScratchPadBitAreaMask [in] long nOnMask63to32 [in] long nOnMaskPts31to0 [in] long nOffMask63to32 [in] long nOffMaskPts31to0	Sets the state of the Scratch Pad bit area. Bits in the on mask are turned on and bits in the off mask are turned off. All other bits are unaffected
SetScratchPadIntegerArea (ActiveX only) [in] short nStartIndex [in] short nLength [in] O22_SIOMM_ScratchPadIntegerBlock pBlock	Writes a block of up to 256 integers to the Scratch Pad integer area. **nLength** is the number of integers to write and must be between 1 and 256.
SetScratchPadIntegerArea (C++ only) [in] long nStartIndex [in] long nLength [in] l_array pnData	Writes a block of up to 256 integers to the Scratch Pad integer area. **nLength** is the number of integers to write and must be between 1 and 256.
SetScratchPadFloatArea (ActiveX only) [in] short nStartIndex [in] short nLength [in] O22_SIOMM_ScratchPadFloatBlock pBlock	Writes a block of up to 256 floats to the Scratch Pad float area. **nLength** is the number of floats to write and must be between 1 and 256.
SetScratchPadFloatArea (C++ only) [in] long nStartIndex [in] long nLength [in] f_array pfData	Writes a block of up to 256 floats to the Scratch Pad float area. **nLength** is the number of floats to write and must be between 1 and 256.
SetScratchPadStringArea [in] short nStartIndex [in] short nLength [in] O22_SIOMM_ScratchPadStringBlock pBlock	Writes a block of up to 8 strings to the Scratch Pad string area. In the ActiveX version, use this method when the strings contain text. **nLength** is the number of strings to write and must be between 1 and 8.
SetScratchPadStringAreaBin (ActiveX only) [in] short nStartIndex [in] short nLength [in] O22_SIOMM_ScratchPadStringBlock pBlock	Writes a block of 8 strings to the Scratch Pad string area. For use when the strings contain binary data. **nLength** is the number of strings to write and must be between 1 and 8.

Return values are shown on page 76.

Memory Map Read/Write Methods

If you need to read or write to areas of the memory map that are not yet supported by the toolkit, for example to configure timers or event messages, use the following methods. For more information on the memory map, see Appendix A.

For the address offset, ignore the first four F's in the memory map address (for example, use F03FFFD0, not FFFFF03FFFD0).

Read Methods	Description
ReadLong[V] [in] long nOffset [out] long pnValue	Reads a long integer value from a memory address offset in the device.
ReadFloat[V] [in] long nOffset [out] float fValue	Reads a float value from a memory address offset in the device.

Read Methods	Description
ReadBlock [in] long nOffset [out] by_array byBlock [in] short nLength	Reads a block of bytes from a memory address offset in the device. Use the GetLongAtBlockIndex() and GetFloatAtBlockIndex() methods below to help read data from the block of bytes.

[v] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

Write Methods	Description
WriteLong[V] [in] long nOffset [in] long nValue	Writes a long integer value to a memory address offset in the device.
WriteFloat[V] [in] long nOffset [in] float fValue	Writes a float value to a memory address offset in the device.
WriteBlock [in] long nOffset [in] by_block byBlock [in] short nLength	Writes a block of bytes to a memory address offset in the device. Use the SetLongAtBlockIndex() and SetFloatAtBlockIndex() methods below to help fill in the block of bytes with data.

[v] The OptoSnapIoMemMapX ActiveX component has a variant-only version of this method.

The following methods help to get and set integer and float values in byte blocks. They are helpful when using the WriteBlock and ReadBlock methods described in the previous table.

Block Helper Methods	Description
GetLongAtBlockIndex [in] by_array byBlock [in] short nIndex [out] long pnValue	Retrieves a long value at the given index in the given byte array.
GetFloatAtBlockIndex [in] by_array byBlock [in] short nIndex [out] float pfValue	Retrieves a float value at the given index in the given byte array.
SetLongAtBlockIndex [in] by_array byBlock [in] short nIndex [in] long nValue	Sets a long value at the given index in the given byte array.
SetFloatAtBlockIndex [in] by_array byBlock [in] short nIndex [in] float fValue	Sets a float value at the given index in the given byte array.

Return values are shown on page 76.

OptoSnaploStream Interface

For more information on configuring a device's streaming feature, see "Streaming Configuration—Read/Write" on page 146 and "Streaming Methods" on page 87.

For more information about streaming, see "Streaming Data" on page 51.

Stream Methods and Events

Use the following methods and events for listening to and processing stream packets.

Methods	Description
OpenStreaming [in] long nType [in] long nLength [in] long nPort	Initializes the stream listener. **nType** specifies the type of stream being sent by the devices: 1 = standard, 2 = custom. All devices being listened to must be using the same type of streaming. **nLength** is the length, in bytes, of a custom stream. Length is ignored for standard streams. **nPort** is the port the streams are being sent to.
CloseStreaming	Stops all listening and closes the port
StartStreamListening [in] string bstrIpAddressArg [in] long nTimeoutMS	Starts listening for streams from the given IP address, **bstrIpAddressArg**. Use multiple calls to this function to listen to multiple devices. The timeout value, **nTimeoutMS**, in milliseconds, is used to generate timeout events if a packet from the given IP address has been received in the timeout period. Different devices can have different timeout values.
StopStreamListening [in] string bstrIpAddressArg	Stops listening to the given IP address.
GetLastStreamStandardBlock (ActiveX only) [out] long pnIpAddressArg [out] f_array fAnalogValues [out] l_array nDigPointFeature [out] long pnStatePts63to32 [out] long pnStatePts31to0 [out] long pnOnLatchStatePts63to32 [out] long pnOnLatchStatePts31to0 [out] long pnOffLatchStatePts63to32 [out] long pnOffLatchStatePts31to0 [out] long pnActiveCountersPts63to32 [out] long pnActiveCountersPts31to0	Retrieves the last standard stream block that was received. The IP address of the device that sent the packet is in the **pnIpAddressArg** parameter. The other parameters contain the analog and digital point values. This method should only be called if the standard type was set in the OptoStreaming() method. This method should be called immediately after receiving an OnStreamEvent ActiveX event (see below).

Methods	Description
GetLastStreamStandardBlockEx [out] O22_SIOMM_StreamStandardBlock pData	Retrieves the last standard stream block that was received. The IP address of the device that sent the packet is in the pnIpAddressArg field. The other parameters contain the analog and digital point value. This method should only be called if the standard type was set in the OptoStreaming() method. **ActiveX:** This method should be called immediately after receiving an OnStreamEvent event . **C++:** This method should be called immediately after receiving a packet event via the pStreamEventCallbackFunc callback function set in StartStreamListening().
GetLastStreamCustomBlockEx [out] O22_SIOMM_StreamCustomBlock pData	Retrieves the last custom stream block that was received. The IP address of the device that sent the packet is in the pnIpAddressArg field. The other parameters contain binary data and the starting memory map address of the data. This method should only be called if the custom type was set in the OptoStreaming() method. **ActiveX:** This method should be called immediately after receiving an OnStreamEvent event . **C++:** This method should be called immediately after receiving a packet event via the pStreamEventCallbackFunc callback function set in StartStreamListening().
SetCallbackFunctions (C++ Only) [in] STREAM_CALLBACK_PROC pStartThreadCallbackFunc [in] void * pStartThreadParam [in] STREAM_EVENT_CALLBACK_PROC pStreamEventCallbackFunc [in] void * pStreamEventParam, [in] STREAM_CALLBACK_PROC pStopThreadCallbackFunc, [in] void * pStopThreadParam	Sets the three callback functions for the C++ class. The **pStreamEventCallbackFunc** callback is called whenever a stream packet of interest is received. See the comments for the method in the O22SIOST.H and O22SIOST.CPP files for more information.

ActiveX Event	Description
OnStreamEvent long nIPAddress long nStatus	This event is called whenever a stream packet is received or an error occurs. The **nIPAddress** parameter contains the IP address of the device that this event concerns. The **nStatus** parameter contains a status code from the Return Values table on page 76. Possible values are: **SIOMM_OK (0):** means that the stream packet is good and should be retrieved with one of the methods above. **SIOMM_TIME_OUT (-2):** means that a stream packet has not been received within the amount of time specified in the StartStreamListening() method. This is not necessarily a serious problem. Depending upon the application, it can perhaps ignore this event and continue waiting for the next packet.

4: Using the OptoMMP Protocol

Introduction

This chapter shows you how to use the OptoMMP protocol to write your own applications for direct communication between a PC and an Opto 22 Ethernet-based memory-mapped device, for example, if you are writing your own driver. (Information on the Opto 22 OptoMMP Communication Toolkit, which makes programming easier for Microsoft Windows users, is in Chapter 3.)

This chapter assumes the following:

- Your Ethernet network—including a PC, hubs if needed, and one or more Opto 22 devices—is already installed. (For help installing and troubleshooting your hardware, see the user's guides listed on page 3.
- Unique, appropriate IP addresses have been assigned to the devices.
- Each device can be reached by the host PC using the PING program.

This chapter also assumes that you are familiar with programming, TCP/IP or UDP/IP, and Ethernet networking. If you are not familiar with these subjects, we strongly suggest you consult commercially available resources to learn about them before attempting to program applications for memory-mapped hardware.

The complete memory map is in Appendix A, starting on page 111. This memory map covers all possible addresses; some may not apply to the hardware you are using. For detailed information on hardware models and features, see "Opto 22 Processor Comparison Chart" on page 9.

Memory Mapping

PAC-R
PAC-S
EB
UIO
EIO
SIO
LCE
E1
E2

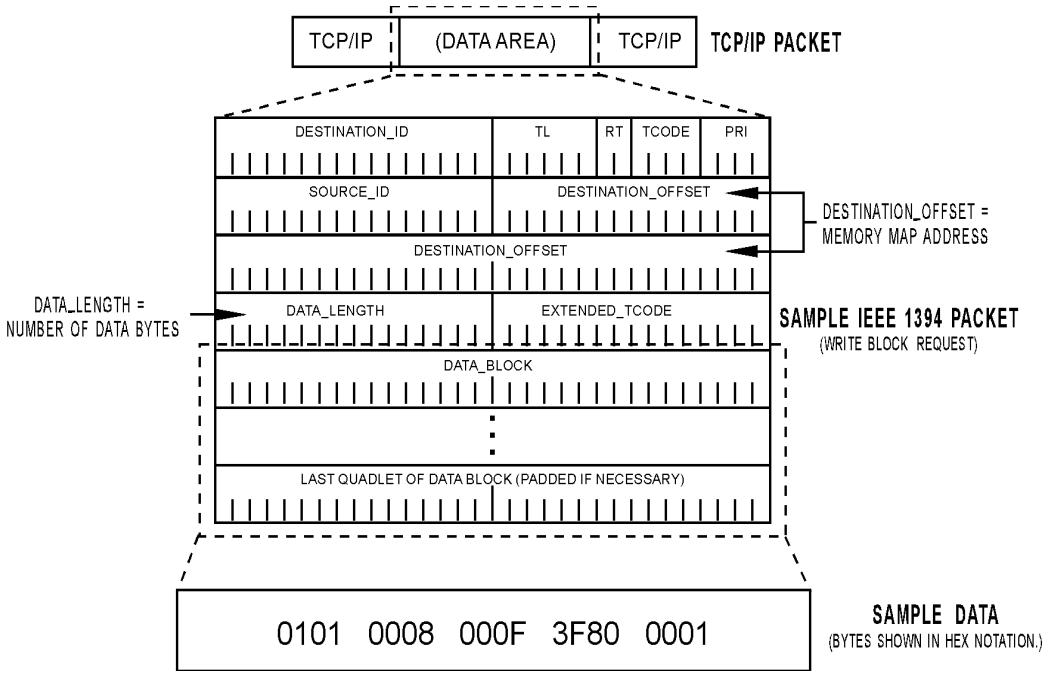
Opto 22 memory-mapped devices use the OptoMMP protocol, based on the IEEE 1394 specification, to provide a standard for reading and writing data.

IEEE 1394 specifies a memory-mapped model for devices on a serial network. For asynchronous transfers, it also specifies a request-response protocol for read/write operations. Basically, each IEEE 1394 node appears logically as a 48-bit address space. To communicate with a device, you read from and write to specific memory addresses in that space.

The memory map is similar to a grid of mailboxes, with each mailbox having its own memory address. See Appendix A for the detailed memory map.

Communication Packets

Communication using the OptoMMP protocol basically involves an IEEE 1394 packet placed inside a TCP/IP or UDP/IP packet. These nested packets look like this TCP example:



The Opto 22 memory-mapped device uses the following types of request packets specified by the IEEE 1394 standard:

- Read Quadlet: reads four bytes starting at an address
- Read Block: reads N bytes starting at an address
- Write Quadlet: writes four bytes starting at an address
- Write Block: writes N bytes starting at an address.

To start communication with the memory-mapped device, the host computer sends one of these four packets via TCP/IP or UDP/IP. To complete each transaction, the device returns a Read Response packet or a Write Response packet. The structure and parameters of the request and response packets are shown beginning on page 105.

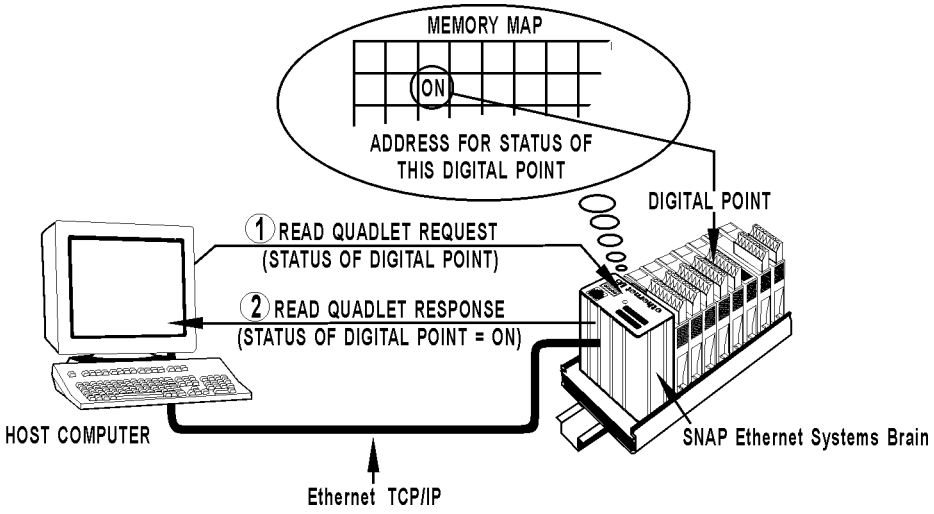
Writing Data

To change the configuration or status of an I/O point, to enable a counter, or to write other data, the host sends a Write Request packet containing the destination address and the new data to be written. The device responds by returning a Write Response packet indicating success or failure.

Reading Data

The host can also access the status of I/O modules, counter values, and other data by reading the appropriate memory locations from the memory map. The host computer simply sends a Read Request packet asking for data from those memory locations, and the device returns the data in a Read Response packet.

The following diagram shows a specific example of a host computer reading data from a SNAP Ethernet-based I/O unit:



Streaming Data

Most communication involves the two-step process of request and response. But some Opto 22 memory-mapped devices can also stream data, as explained on page 51. Streaming uses UDP and does not require a response.

For more information on using streaming, see page 100.

Overview of Custom Application Programming

PAC-R
PAC-S
EB
UIO
EIO
SIO
LCE
E1
E2

If you are not using the OptoMMP Communication Tookits for Windows or Linux (see Chapter 3) but need to develop custom applications using the protocol itself, this section shows you how to build packets to communicate with Opto 22 memory-mapped hardware.

Programming requires five basic steps: connect, send Powerup Clear, configure, read/write, and disconnect. This overview section leads you through these steps.

Connecting

To connect with the device, you can use a basic socket interface, such as Microsoft Winsock control. Assign the IP address and port. Note that the OptoMMP port defaults to 2001 for the device (You can change this port number using address F03A0004 (see page 131). If the variable name for the device is tcpIOUnit, connection might look like this:

```
tcpIOUnit.RemoteHost = "10.192.0.69" 'IP address of device
tcpIOUnit.RemotePort = 2001
tcpIOUnit.Connect
```

Sending Powerup Clear

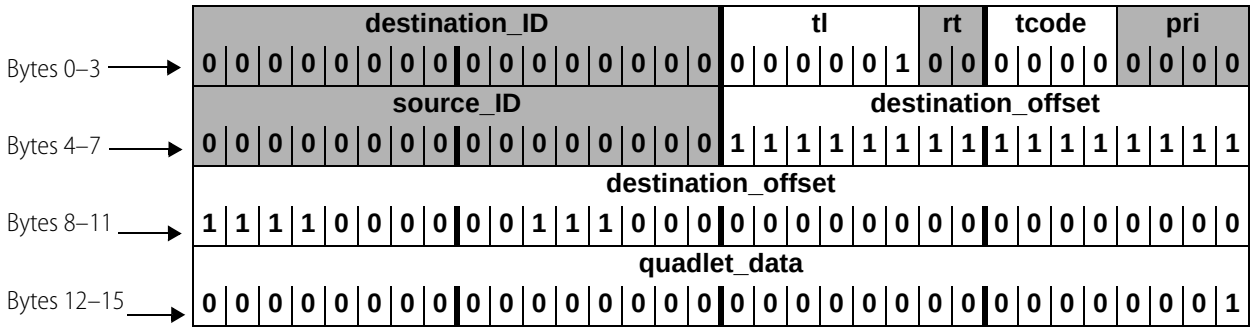
Once a connection has been established, the host must send a Powerup Clear message (PUC) to the memory-mapped device. **You can't do anything except read the memory map's Status area until the Powerup Clear is sent.** Other requests will return a negative acknowledgment (NAK), and the error Powerup Clear Expected will appear in the Status area. (See page 127.)

After the initial PUC is sent, you do not need to send another unless the device has been turned off or restarted. To check whether a Powerup Clear is needed, you can read the PUC flag in the Status area. A zero means the PUC has been sent; anything else means you must send a PUC.

To send a Powerup Clear, build a Write Quadlet Request packet with data00000001 written to offset FFFFF0380000, which is the memory map location for sending a Powerup Clear. (The complete memory map is shown in Appendix A, starting on page 111.)

Write Quadlet Request Packet (PC→Device)

The packet should appear as follows in binary notation. (For more information on communication packets, see "Read and Write Packet Structure" on page 105).



In hex, the packet looks like this:

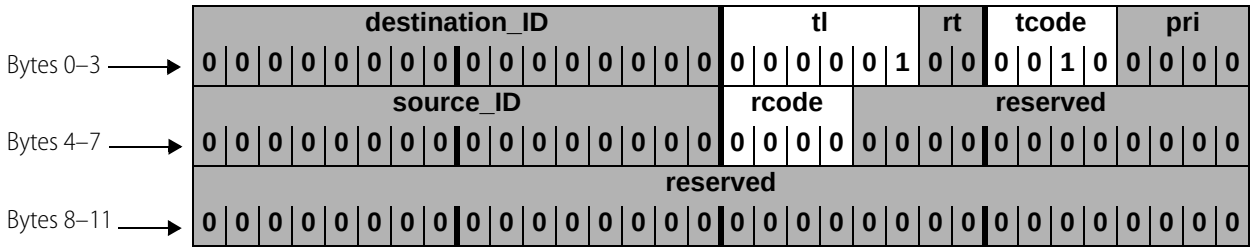
000004000000**FFFFF038000000000001**

quadlet_data

destination_offset

Write Response Packet (Device→PC)

When the device receives the Powerup Clear, it sends a Write Response packet back to the host PC acknowledging receipt. In binary notation, the response packet looks like this:



The *rcode* parameter contains the ACK or NAK for the transaction. The 0 in this example indicates an ACK.

In hex, the packet looks like this: 000004200000**0**00000000000. The 0 shown in bold type is the ACK in the *rcode*.

See "Error Codes" on page 109 for information on what to do if you receive a NAK.

Configuring

SNAP Ethernet-based I/O units (but not E2 I/O units) can recognize the presence and type of an analog module on the rack, but the values for each of the points must be configured if they do not match the default for that module type. For example, the I/O unit can report that a SNAP-AITM

module is in position 4, but if the points are anything other than the default value of ±150 mV, you must configure them by writing configuration codes to the points. Point types and default values are shown in the tables starting on page 23.

Digital modules and empty positions are reported the same by an Ethernet-based I/O unit or an E1 I/O unit: they are assumed to be digital input modules. If a position contains a digital output module, you must configure the points as outputs.

Serial and high-density digital modules do not require configuration for use with custom applications.

Configuring I/O Point Types—Write Quadlet Request Packet (PC?Device)

Suppose you have a digital output module in position 0 on the rack. Since the I/O unit cannot distinguish a digital module from an empty position, you need to configure the points (for a SNAP module, all points) as outputs. You configure them by writing to each one's Point Type address in the "(Expanded) Analog & Digital Point Configuration—Read/Write" area of the memory map. On page 114 you can see this area of the map.

The Write Quadlet Request for point 0 would look like this: 000004000000FFFF**F0100004**00000180

Write Quadlet Request for point 1 (hex): 000004000000FFFF**F01000C4**00000180

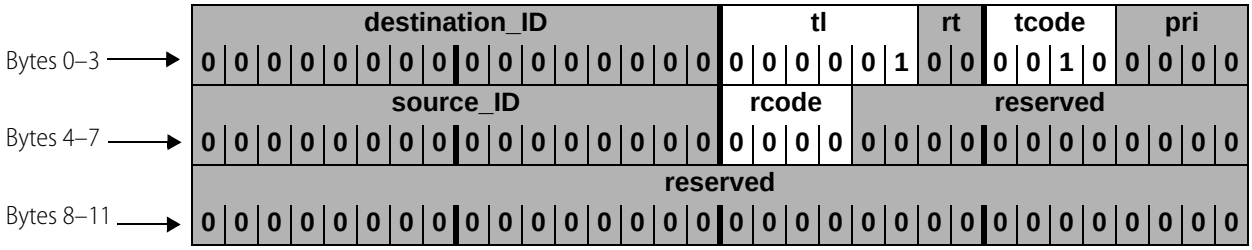
Write Quadlet Request for point 2 (hex): 000004000000FFFF**F0100184**00000180

Write Quadlet Request for point 3 (hex): 000004000000FFFF**F0100244**00000180

As you can see, the only difference in these packets is the memory map address for the point. For more information on module types and setting point types, see "Configuring I/O Points, Event/Reactions, and Security" on page 105.

Configuring I/O Point Types—Write Response Packet (Device?PC)

The response from the device is a simple acknowledgment, as you saw before, with the ACK or NAK appearing in the *rcode* parameter:



Reading and Writing

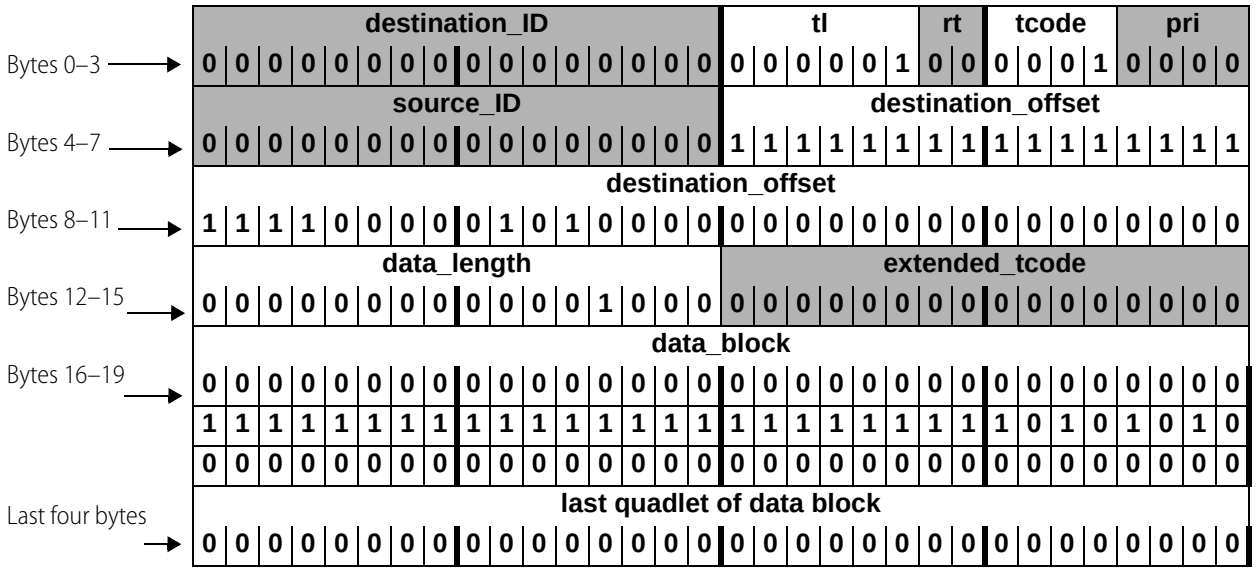
Now that the PC has successfully connected to the device, sent a Powerup Clear, and configured I/O points as necessary, you can read and write to the points.

Turn on Digital Points—Write Block Request (PC?Device)

Suppose you want to turn on multiple points on 4-channel digital modules. Using bank addresses in the memory map, you can turn them on all at once. As you can see on page 147, the starting address for Turn On (Digital Bank Write) is FFFFF0500000. This starting address goes into the *destination_offset* parameter.

Since this portion of the memory map is a mask, you need to use the entire length of 8 bytes (hex). The length goes into the *data_length* parameter. The *data_block* parameter contains the mask. (For more information on formatting data for a mask, see "Mask Data" on page 58.) The mask shown in this example would turn on points 1 and 3 on modules 0 and 1, and all four points on modules 2–7.

Here is the Write Block Request:



In hex, the request would look like this:

000004100000**FFFFF0500000**000800000000**FFFFFFAA**...

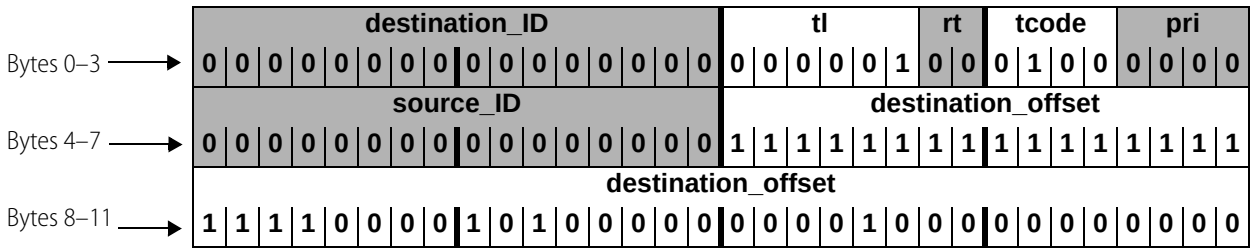
The starting address and the mask are shown in bold.

Turn on Digital Points—Write Block Response (Device?PC)

Again, look in the *rcode* parameter of the Write Block Response from the device to see an ACK (0) or a NAK (other than 0).

Read Analog Point Data—Read Quadlet Request (PC?Device)

Suppose you want to read the value of point 1 on module 0, which is an analog point. You can tell from the memory map ("(Expanded) Analog Point Read—Read" on page 116) that the value in Engineering Units for module 0, point 1 is at the address FFFFF0260040. The Read Quadlet Request would look like this:

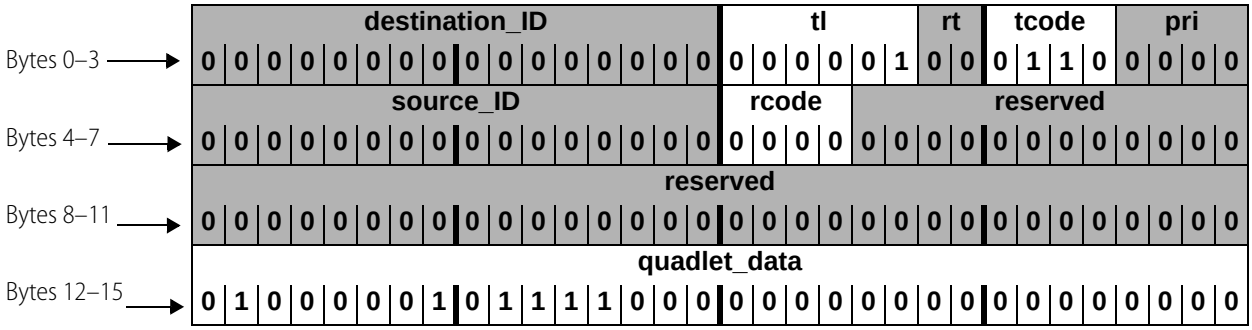


In hex, the request would be as follows (address is bolded):

000004400000**FFFFF0260040**

Read Analog Point Data—Read Quadlet Response (Device→PC)

The response to your request might look like this in binary notation:



In hex, it would look like this: 00000460000000000000000**41780000**

The *rcode* parameter shows an ACK (0), and the *quadlet_data* parameter, shown in bold type in the hex version, equals the IEEE float 15.5.

Disconnecting

The connection is kept open during normal communications. Disconnect only when all communication is complete. To disconnect, you can again use a basic socket interface, such as Microsoft Winsock control:

```
tcpIOUnit.Close
```

Streaming Data

PAC-REB UIO EIO SIO

Streaming is a fast way to get continuous information about I/O from some memory-mapped devices. See page 51 for information on streaming and other methods to configure it. Streaming involves two steps: configuring parameters for streaming, and receiving streamed data.

Configuring Parameters for Streaming

To configure parameters for streaming, use a Write Block Request to the memory map area "Streaming Configuration—Read/Write" on page 146.

- To FFFFF03FFFD4, write how often in milliseconds you want to receive the streamed data.
- To FFFFF03FFFD8, write the Ethernet port number that will receive data.
- To the Stream Target addresses, write the IP addresses of the hosts that should receive the data.
- To turn streaming on, write anything but a zero to the address FFFFF03FFFD0. To turn streaming off, write a zero to this address.

Receiving Streamed Data

As soon as you've configured parameters for streaming, the device starts sending the data you requested. The device sends data using a Write Block Request, with the *data_block* parameter

containing data as shown in the memory map area "Streaming—Read Only" on page 168. Your program does not need to respond to this Write Block Request; it only needs to process the data block.

Addresses will be zero-filled in areas that don't apply. For example, addresses FFFFF1000000 through FFFFF10000FF show analog data for 64 points in Engineering Units. If some of the points are digital, addresses corresponding to them will be filled with zeros.

See "Stream Packet Format" on page 52 for additional information.

Using I/O Point Features

General information on I/O point features is on page 29. This section provides specific information for the OptoMMP protocol.

PAC-R
EB
UIO
EIO
SIO
E1

Latches

Latching is automatic and needs no configuration. Using the Opto 22 protocol, you can read the on-latch or off-latch state of a digital input point. You can:

- Read latches for individual points and leave them set
- Read latches for a bank of digital points and leave them set
- latches for individual points.

To read latches for individual points on modules with more than 4 points, see "SNAP High-Density Digital—Read Only" on page 179.

To read latches for individual points on 4-channel modules, see the memory map area "Digital Point Read—Read Only" on page 149, and use the on-latch or off-latch state starting address for the point you want to read. For example, you would read off-latch status for module 0, point 0 starting at address FFFFF0800002.

To read a bank of points on 4-channel modules, see the memory map area "Digital Bank Read—Read Only" on page 147. The starting address for reading the state of on-latches is FFFFF0400008, and the starting address for off-latches is FFFFF0400010.

To latches for individual points (all digital modules), see the memory map area "(Expanded) Digital Point Read & Clear—Read/Write" on page 118. If you are reading and clearing the on-latch at module 0, point 1, for example, you would use the starting address FFFFF02E0018. Alternatively, for points on high-density digital modules, you can latches using "SNAP High-Density Digital Read and Clear—Read/Write" on page 180.

For help in understanding the data you read, see "Formatting and Interpreting Data" on page 58.

PAC-R
EB
UIO
EIO
SIO
E1

Counters

This feature applies to digital input points on I/O units with the following processors:

SNAP-PAC-R1
SNAP-UP1-ADS
SNAP-B3000-ENET
SNAP-ENET-RTC
E1

When configured, it will count the number of times the input changes from off to on. For most points, using counters involves two steps: configuring the counter and reading data.

NOTE: On SNAP high-density digital points, counting is automatic and requires no configuration for use with the OptoMMP protocol.

Configuring a Counter

(Not necessary for SNAP high-density digital points.) To configure a digital input as a counter, **first configure the point as an input**. Write to the "(Expanded) Analog & Digital Point Configuration—Read/Write" area of the memory map. (See page 114.) For example, to configure module 0, point 0 as a counter, you would write to the memory map address FFFFF0100004, using 00000100 as the data for a digital input.

Next, configure the point feature as a counter. In the same area of the memory map, for the same point, you would write to the address FFFFF0100008 and use 00000001 as the data for a counter.

Reading a Counter

You can:

- Read counters for individual points and leave them counting
- Read a bank of points and leave them counting
- individual point counters in one step, setting the counters back to zero

For points on SNAP high-density digital modules, to read counters, see "SNAP High-Density Digital—Read Only" on page 179 or "Analog EU or Digital Counter Packed Data—Read" on page 169. To counters, see "SNAP High-Density Digital Read and Clear—Read/Write" on page 180.

To read counters for individual points, see the memory map area "Digital Point Read—Read Only" on page 149, and use the counter data starting address for the point you want to read. For example, you would read counter data for module 0, point 1 starting at address FFFFF0800044.

To read a bank of points, see the memory map area "Digital Bank Read—Read Only" on page 147. The starting address for reading counter data is FFFFF0400100. For help in interpreting this data, see "Mask Data" on page 58.

To counters for individual points on all digital modules, see the memory map area "(Expanded) Digital Point Read & Clear—Read/Write" on page 118. If you are reading and clearing the counter at module 0, point 0, for example, you would use the starting address FFFFF02E0000.

PAC-R
EB
UIO
EIO
SIO

Quadrature Counters

I/O units with the following processors also support quadrature counters for quadrature encoder devices:

SNAP-PAC-R1	SNAP-ENET-RTC
SNAP-B3000-ENET	SNAP-UP1-ADS

See page 32 for detailed information on using quadrature counters.

PAC-R
EB
UIO
EIO
SIO
E1
E2

Watchdog

Using a watchdog involves three steps:

1. Setting up the watchdog time in milliseconds
2. Configuring the watchdog values for the critical points on digital and analog modules
3. Enabling the watchdog for those points

Set up the watchdog time by using the "Status Area Write—Read/Write" on page 127. Write the watchdog time in milliseconds starting at the address FFFFF0380010. This is the amount of time the I/O unit will wait for communication from the host device.

Configure the watchdog values for digital and analog points using the memory map area "(Expanded) Analog & Digital Point Configuration—Read/Write" on page 114. These are points you want to set to a certain state or value if the watchdog timeout occurs. For example, to close a valve at digital output point 1 on module 0, you would write a zero starting at the address FFFFF0100024. To set a value on analog output point 0 on module 1, you would write the EU float starting at the address FFFFF01000E4.

Enable the watchdog for the points for which you've set watchdog values, also using "(Expanded) Analog & Digital Point Configuration—Read/Write" on page 114. For the example of the digital output at point 1, you would write starting with the address FFFFF0100028. For the analog output at point 4, the starting address would be FFFFF01000E8.

PAC-R
EB
UIO
EIO
SIO
E2

Scaling

Scaling applies to analog points only. To scale a point, see the memory map area "(Expanded) Analog & Digital Point Configuration—Read/Write" on page 114. Write to the point's addresses for high scale and for low scale. For example, to scale module 0, point 0, you would write the high-scale float starting at the address FFFFF0100014 and the low-scale float starting at FFFFF0100018.

PAC-R
EB
UIO
EIO
SIO
E2

Minimum and Maximum Values

Memory-mapped I/O units with analog capability automatically keep track of minimum and maximum values on analog points. You can read the values at any time, for example, to record minimum and maximum temperatures. You can:

- Read min/max values for individual points
- Read a bank of points
- Read *and restart* min/max values for individual points.

To read min/max values for individual points, see the memory map area "(Expanded) Analog Point Read—Read" on page 116, and use the Min Value or Max Value starting address for the point you want to read. For example, you would read the minimum value for module 0, point 0 starting at address FFFFF0260008.

To read min/max values for a bank of points, see the memory map area "Analog Bank Read—Read Only" on page 148. The starting address for reading minimum values is FFFFF0600200. For help in interpreting this data, see "IEEE Float Data" on page 60.

To read and restart min/max values for individual points, see the memory map area "(Expanded) Analog Point Read & Clear—Read/Write" on page 115. For example, if you want to record the maximum temperature at module 0, point 1 in each 24-hour period, the values must be reset when they are read each day. You would read and restart the maximum value for module 0, point 0 using the starting address FFFFF01D4010.

PAC-R
EB
UIO
EIO
SIO
E2

Offset/Gain

Offset and gain apply to analog input points only. To have the I/O unit calculate offset and gain, use the memory map area "(Expanded) Analog Point Calc & Set—Read/Write" on page 115. Calculate offset first, then calculate gain.

For example, calculate offset for module 0, point 1 by reading addresses FFFFF01C0008 through FFFFF01C000B. Calculations are completed in the background, and the response gives the offset in counts. Next, calculate gain for the same point by reading addresses FFFFF01C000C through FFFFF01C000F. Response for gain is in percent.

Since the purpose of the read request is simply to have the offset or gain calculated so that values you read later will be accurate, you can normally ignore the response data.

If you want to save the response data—or if you want to calculate offset and gain by hand—you can write this data to the memory map area "(Expanded) Analog & Digital Point Configuration—Read/Write" on page 114.

Configuring I/O Points, Event/Reactions, and Security

See Chapter 2 for important information on configuring I/O points, I/O point features, event/reactions, and other system functions.

PAC-R
EB
UIO
EIO
SIO

Reading Module Types

Analog/digital/serial I/O units can recognize analog and special-purpose modules on the rack and report what they are. Digital modules (input and output) and empty slots are reported as the digital inputs by all I/O units.

To read a module type, use the "(Expanded) Analog & Digital Point Configuration—Read/Write" area of the memory map. (See page 114.) Module type values are shown in the table on page 156.

PAC-R
EB
UIO
EIO
SIO
E1
E2

Configuring I/O Point Types

Although some I/O units recognize many module types, they do not recognize 4-channel digital modules or individual analog point values. If the actual module type or point values differ from the defaults, you must assign the correct values by writing to the "(Expanded) Analog & Digital Point Configuration—Read/Write" area of the memory map. (See page 114.) See "Configuring I/O Points" on page 22 for more information. Point type values are shown in the tables starting on page 23.

Configuring I/O Point Features

See page 101 for information on counters and quadrature counters.

Read and Write Packet Structure

Parameters

The following table defines the parameters for all requests and responses:

Parameter	Full Name	Description
destination_id	Destination identifier	Not used by Opto 22 memory-mapped devices. Set this parameter to zero.
tl	Transaction label	A label specified by the requester and identifying this transaction. This value is returned in the response packet.
rt	Retry code	Not used by Opto 22 memory-mapped devices. Set this parameter to zero.

Parameter	Full Name	Description
tcode	Transaction code	Defines the type of packet: Write Quadlet Request = 0 Write Block Request = 1 Write Quadlet or Block Response = 2 Read Quadlet Request = 4 Read Block Request = 5 Read Quadlet Response = 6 Read Block Response = 7
pri	Priority	Not used by Opto 22 memory-mapped devices. Set this parameter to zero.
source_id	Source identifier	Optional parameter. If you are running two or more applications simultaneously, you can give each application a different ID in this parameter. If an error occurs, you can read the Source address in the memory map Status area to find out which application caused the error. See page 119.
rcode	Response code	Indicates whether the command was successful. Successful command (ACK) = 0 Unsuccessful command (NAK) = any number except zero. If you receive a NAK in this parameter, see “Error Codes” on page 109.
destination_offset	Destination offset	Specifies the address location in the target node.
data_length	Data length	Specifies the amount of data being sent in the data parameter of this packet. Maximum size is 2034 bytes for data sent via TCP, or 1480 bytes for data sent via UDP.
extended_tcode	Extended transaction code	Not used by Opto 22 memory-mapped devices. Set this parameter to zero.
quadlet_data	Quadlet data	Data being delivered to the target node. If it is not an even four bytes of data, pad with zeros at the end, not the beginning.
data_block	Data parameter	Data being transferred to the target device.

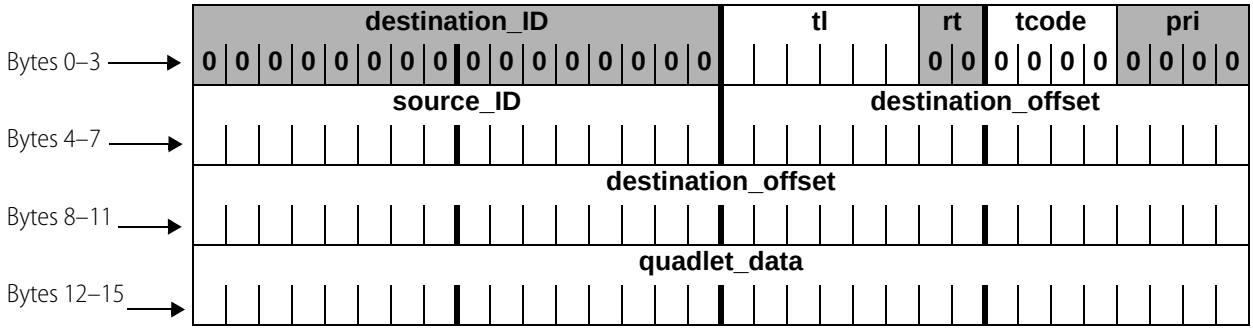
Packet Structure

The following pages show the structure for read and write request and response packets. Opto 22 memory-mapped devices do not use the parameters *destination_ID, rt,* or *pri* (or *reserved* areas). **These areas must be zero filled.** They are shown shaded. The *source_ID* parameter (described in the previous table) is optional. If you do not use it, fill it with zeros.

OptoMMP packets have boundaries at four bytes (a quadlet). When you send a Write Quadlet Request, if the data you enter in the *quadlet_data* parameter is less than four bytes, fill the remaining spaces with zeros to complete the quadlet. Zero fill at the end, not the beginning, of the data.

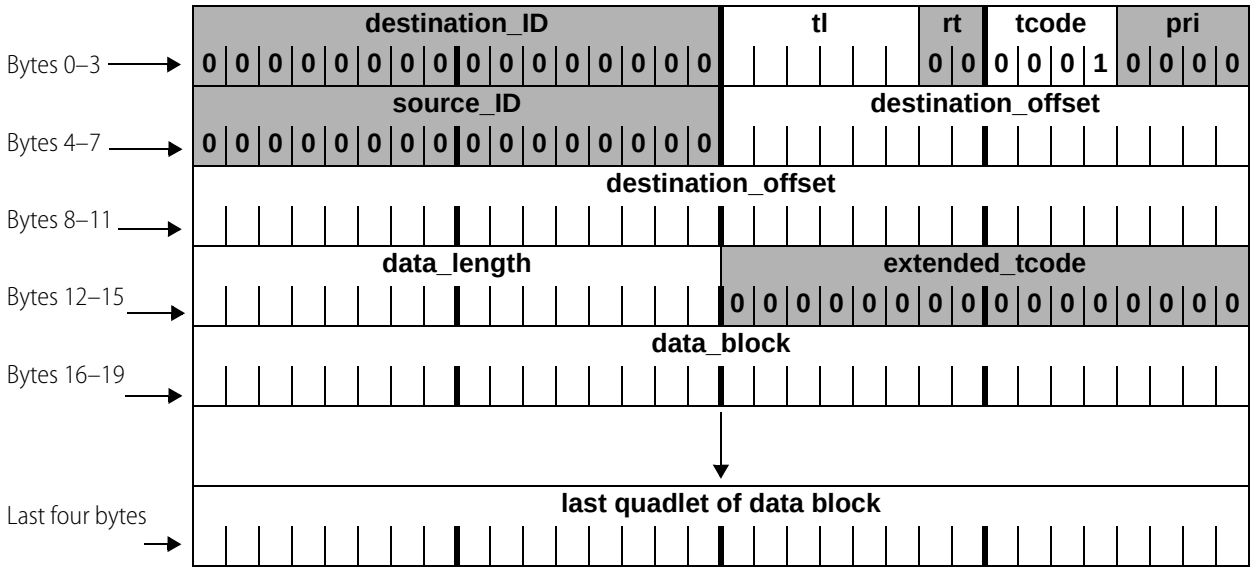
Write Quadlet Request

Tcode = 0



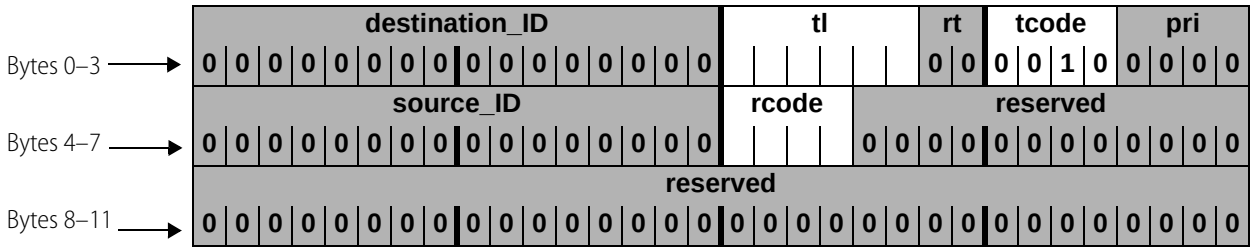
Write Block Request

Tcode = 1



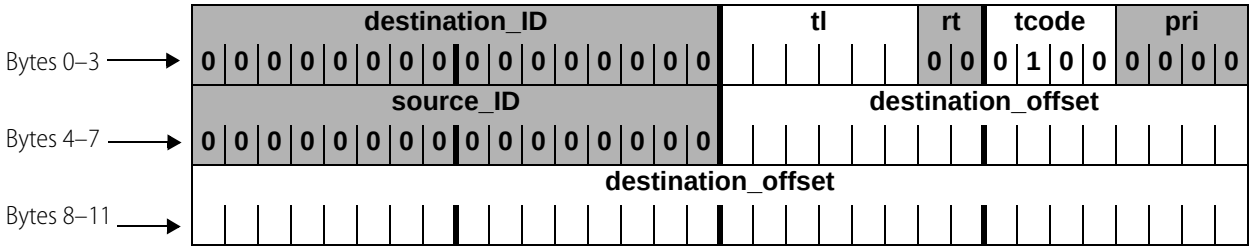
Write Quadlet or Block Response

T-code = 2



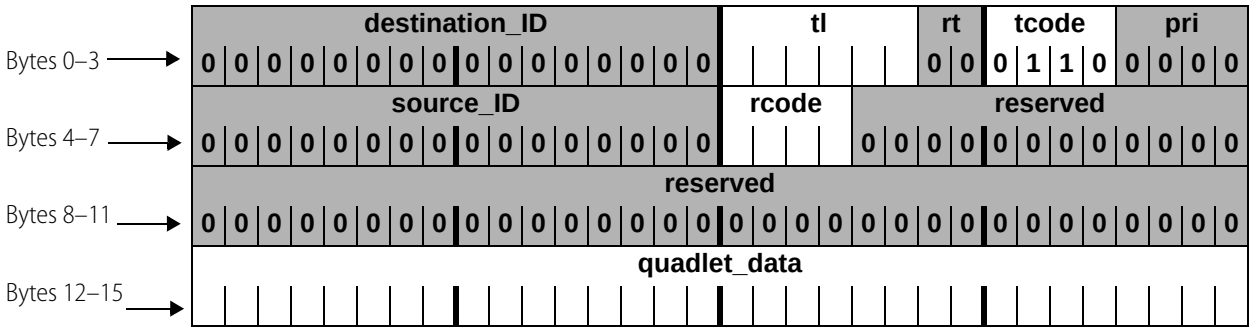
Read Quadlet Request

Tcode = 4



Read Quadlet Response

Tcode = 6

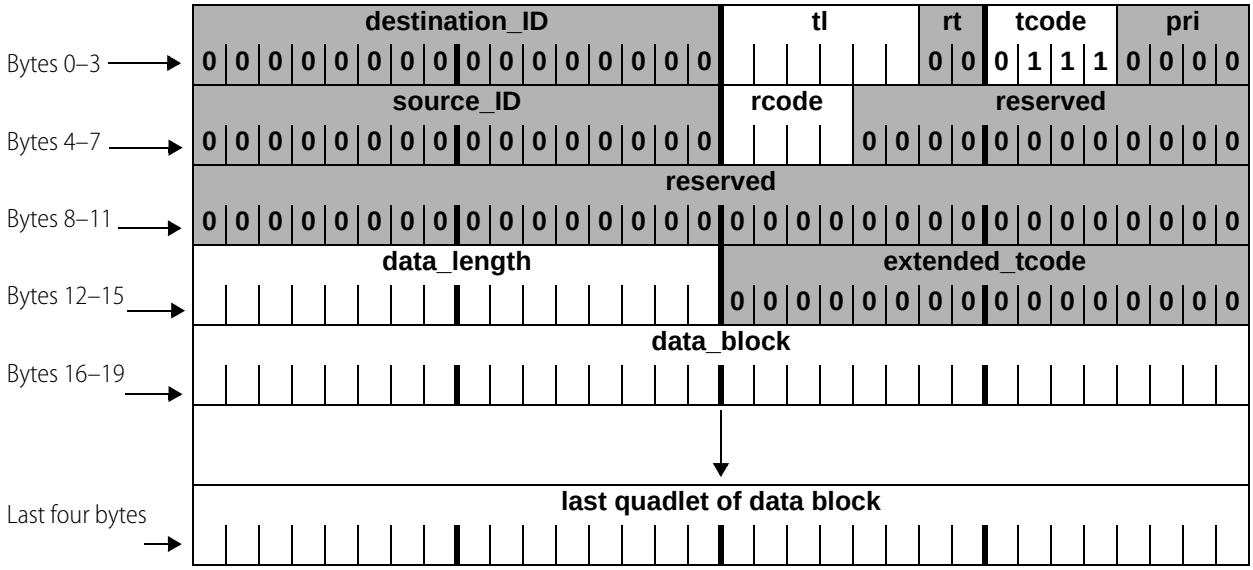


Read Block Request

Tcode = 5

Read Block Response

Tcode = 7



Error Codes

Response packets contain an *rcode* parameter, which shows whether the response is an ACK or a NAK. An *rcode* of 0 is an ACK; anything else is a NAK. If a NAK appears in the *rcode* parameter, check the Status area of the memory map (see page 119) to find out the reason for the NAK.

CAUTION: If more than one client is communicating with the memory map (for example, your program and PAC Manager), you may read an error caused by a different client.

The following table lists the error codes that may appear in the Status area of the memory map:

Code (Hex)	Meaning
0000	No error
E001	Undefined command
E002	Invalid point type
E003	Invalid float
E004	Powerup Clear expected
E005	Invalid memory address or invalid data for the memory address
E006	Invalid command length
E007	Reserved

Code (Hex)	Meaning
E008	Busy
E009	Cannot erase flash
E00A	Cannot program flash
E00B	Downloaded image too small
E00C	Image CRC mismatch
E00D	Image length mismatch
E00E	Feature is not yet implemented
E00F	Communications watchdog timout

A: Opto 22 Hardware Memory Map

Introduction

The tables on the following pages show all possible memory map locations for Opto 22 memory-mapped hardware devices. The addresses available on your device are determined by the device type and model. Device types that support each section of the memory map are indicated at the beginning of the section (and sometimes for specific addresses).

Features available for I/O processors (brains, brain boards, and on-the-rack controllers) are detailed in the "Opto 22 Processor Comparison Chart" on page 9. When using the memory map, **ignore any addresses that don't apply to your device**; for example, ignore all analog addresses if you are using a digital-only I/O unit.

This appendix does not apply to Modbus/TCP. For Modbus, see the device's user's guide.

Notes on Columns

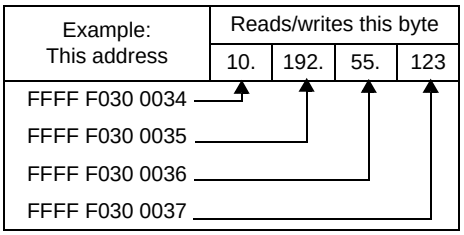
Starting Address—All memory map addresses are shown in hex notation. Note that addresses are shown with spaces for easier reading; when you use them, however, do not use spaces.

Length—The Length column shows the number of bytes in hex.

Type—Data type for the addresses is indicated as follows:

B	Boolean: Zero = false. Any non-zero value = true. (1-byte or 4-byte; see Length column for size)
F	Float (4-byte)
I	Signed integer (1-byte, 2-byte, or 4-byte; see Length column for size)
UI	Unsigned integer (1-byte, 2-byte, or 4-byte; see Length column for size)
IP	IP address format (four 1-byte integers; see example following this table)
M	Mask (32-bit or 64-bit; see Length column for size)
S-ZT	String (null-terminated). The maximum length string allowed is one less than the total length, since the last character is always zero.
S-PL	String (prepended length)

IP address data type consists of four 1-byte integers. As shown in the example below, the lower-numbered address reads or writes the first byte:



Byte Ordering and Data Ordering

All non-mask data for points in bank and point areas are arranged in low point/low address order, as follows:

- Point 0 at 0x....00
- Point 1 at 0x....04
- Point 2 at 0x....08
- Etc.

All masks and multi-byte values (floats, integers, and so on) are in Big-Endian format, which means that the higher-ordered byte is in the lower-ordered address.

See page 58 for more information and examples.

PAC-R
EB
SB

For Experienced OptoMMP Users

IMPORTANT: If you've used the memory map in the past, be aware that major changes occurred when firmware versions 8.0 and 8.1 were introduced for SNAP PAC controllers and brains. To accommodate new SNAP I/O modules with more than four points, new expanded memory map areas were developed. While the older areas still work, we recommend using the expanded areas for new development, as they offer greater flexibility for the future.

NOTE: Expanded areas apply to SNAP PAC R-series, SNAP PAC EB, and SNAP PAC SB I/O units only. For SNAP Ultimate, Ethernet, Simple, E1, and E2 I/O units, continue to use the old areas.

If your program must accommodate devices with different firmware versions, you can try reading an address in the expanded area first. If you receive a response, use addresses in the expanded area. If there is no response, then use addresses in the old area.

Expanded areas and old areas are clearly marked in the memory map sections in this appendix. Here's a brief list of them:

Function	Expanded area		Old area		Notes
	Address	Page	Address	Page	
Analog and Digital Point Configuration—Read/Write	F0100000	page 114	F0C00000	page 153	
Analog Point Calc & Set	F01C0000	page 115	F0E00000	page 165	
Analog Point Read & Clear	F01D4000	page 115	F0F80000	page 167	
Analog Point Read	F0260000	page 116	F0A00000	page 151	
Analog Point Write	F02A0000	page 117	F0B00000	page 152	
Digital Point Read & Clear	F02E0000	page 118	F0F00000	page 166	
Digital Events - Expanded	F0D40000	page 158	F0D00000	page 157	Replaces Digital Events - Old. Expansion of Timers.
Scratch Pad - 64-bit Integers	F0DE0000	page 163	--	--	New Scratch Pad section
Analog EU or Digital Counter Packed Data Read	F1001000	page 169	--	--	Not a replacement
Digital Packed Data Read/Write	F1001800	page 170	--	--	Not a replacement

General Notes

For these devices, you cannot read or write more than 2,034 bytes at a time via TCP. Via UDP, the limit is 1,480 bytes.

Within these limits, you can read or write to large areas within the memory map using a block read or write. (Each area of the map is shown under a separate heading in the following pages.) For example, using TCP, you could read or write up to 2034 bytes of data—about a third of the entire area—in the "(Expanded) Analog & Digital Point Configuration—Read/Write" area shown on page 114. If you are reading, just ignore any data in the reserved addresses between points.

If you read or write beyond the last valid address in any area, however, you may receive an error.

Reading or writing in multiples of four bytes is recommended and is generally faster than accessing a number of bytes that is not a multiple of four.

CAUTION: *In certain areas, if you read or write less than a quadlet, the data will be useless. For example, reading two bytes of a float won't give complete data. Even more important, if you read only two bytes of a float in an area such as the Read and Clear area of the map, not only will the data you receive be useless, but also the information in the memory map will be erased (cleared).*

(Expanded) Analog & Digital Point Configuration—Read/Write

PAC-R EB SB G4EB2

See page 22 for configuration information. For I/O units with firmware version 8.0 and newer, this area of the memory map replaces the F0C00000 area (page 153). To allow for future growth, this area has space for a total of 4096: an array of 64 points per module on 64 modules. (Current SNAP I/O units offer a maximum of 16 modules.)

Only the first point on the first module is shown in the table. Each successive point starts on an even C0 hex boundary and follows the same pattern.

Starting Address	Length (Hex)	Type	Description
FFFF F010 0000	4	UI	Module 0, Point 0 (0,0): Module Type (read only). Module type is the value reported by an analog, serial, or high-density digital module on a SNAP rack. Zero is returned for single or 4-channel digital modules, no module, or points that don't exist (for example, the upper 62 points on a two-channel analog module). See table on page 156 for values.
FFFF F010 0004	4	UI	0,0: Point Type • Use 0x0000 0100 for single or 4-channel digital inputs. • Use 0x0000 0180 for single or 4-channel digital outputs. • For analog types, see tables starting on page 23.
FFFF F010 0008	4	UI	0,0: Point Feature (unsigned integer) Digital feature values: • 0x0000 0001 for counter input (configures and starts the counter) • 0x0000 0002 for on-time totalizer input • 0x0000 0003 for period measurement (continuous) • 0x0000 0004 for simple quadrature counter input • 0x0000 0005 for frequency measurement (continuous) (Firmware lower than 8.1) • 0x0000 0008 for frequency measurement (continuous) (Firmware 8.1a and higher) • 0x0000 0009 for on-pulse duration measurement (one-time) • 0x0000 000A for off-pulse duration measurement (one-time) • 0x0000 000B for period measurement (one-time) • 0x0000 000C for frequency measurement (one-time) • 0x0000 0012 for off-time totalizer input • 0x0000 0041 quadrature counter input with index. For quadrature counter information, see page 32. Analog feature values: none at present To disable point features, use 0x0000 0000
FFFF F010 000C	4	F	0,0: Analog point offset
FFFF F010 0010	4	F	0,0: Analog point gain
FFFF F010 0014	4	F	0,0: Analog point high scale (Engineering Units)
FFFF F010 0018	4	F	0,0: Analog point low scale (EU)
FFFF F010 001C	4	--	0,0: Reserved
FFFF F010 0020	4	F	0,0: Average filter weight
FFFF F010 0024	4	F	0,0: Watchdog value. EU float for analog and digital (for digital, 0 = off; non-0 = on)
FFFF F010 0028	4	B	0,0: Enable watchdog. 0 = disable; non-0 = enable
FFFF F010 002C	4	--	0,0: Reserved

Starting Address	Length (Hex)	Type	Description
FFFF F010 0030	33	S-ZT	0,0: Point name (51 characters maximum)
FFFF F010 0063	55	--	0,0: Reserved
FFFF F010 00B8	4	F	0,0: Analog point lower clamp
FFFF F010 00BC	4	F	0,0: Analog point upper clamp
(Additional points follow in order on even C0 hex boundaries.) (Additional modules follow in order on even 3000 hex boundaries.)			
Last valid address for this area: FFFF F01B FFFF			

(Expanded) Analog Point Calc & Set—Read/Write

PAC-R EB SB

See page 35 for more information on setting offset and gain. For I/O units with firmware version 8.0 and newer, this area of the memory map replaces the F0E00000 area (page 165). To allow for future growth, this area has space for 4096 points, which is 64 points per module on 64 modules. (Current SNAP I/O units offer a maximum of 16 modules.)

See "IEEE Float Data" on page 60 for help in interpreting data.

Only the first two points on the first module are shown in the table. Successive points and modules follow the same pattern.

Starting Address	Length (Hex)	Type	Description
FFFF F01C 0000	4	F	Module 0, Point 0 (0,0): Offset
FFFF F01C 0004	4	F	0,0: Gain
FFFF F01C 0008	4	F	0,1: Offset
FFFF F01C 000C	4	F	0,1: Gain
(Additional points follow in order.) (Additional modules follow on even 200 hex boundares.)			
Last valid address for this area: FFFF F01C 7FFF			

(Expanded) Analog Point Read & Clear—Read/Write

PAC-R EB SB

See page 35 for more information on minimum and maximum values. For I/O units with firmware version 8.0 and newer, this area of the memory map replaces the F0F80000 area (page 167). To allow for future growth, this area has space for 4096 points, which is 64 points per module on 64 modules. (Current SNAP I/O units offer a maximum of 16 modules.)

When you read data from this area, the data is returned and then cleared or reset. See "IEEE Float Data" on page 60 for help in interpreting data.

CAUTION: If you read or write less than a quadlet in this area of the memory map, the returned data will be useless and the information will be erased (cleared).

Only the first two points on the first module are shown in the table. Successive points and modules follow the same pattern.

Starting Address	Length (Hex)	Type	Description
FFFF F01D 4000	4	F	Module 0, Point 0 (0,0): Minimum value
FFFF F01D 4004	4	F	0,0: Maximum value
FFFF F01D 4008	4	--	0,0: Reserved
FFFF F01D 400C	4	F	0,1: Minimum value
FFFF F01D 4010	4	F	0,1: Maximum value
FFFF F01D 4014	4	--	0,1: Reserved
(Additional points follow in order.) (Additional modules follow in order on 300 hex boundaries.)			
Last valid address for this area: FFFF F01D FFFF			

(Expanded) Analog Point Read—Read

PAC-R EB SB

For I/O units with firmware version 8.0 and newer, this area of the memory map replaces the F0A00000 area (page 151). To allow for future growth, this area has space for 4096 points, which is 64 points per module on 64 modules. (Current SNAP I/O units offer a maximum of 16 modules.)

See "IEEE Float Data" on page 60 for help in interpreting data.

Only the first point on the first module is shown in the table. Successive points and modules follow the same pattern.

Starting Address	Length (Hex)	Type	Description
FFFF F026 0000	4	F	Module 0, Point 0 (0,0): Analog point value (Engineering Units)
FFFF F026 0004	4	F	0,0: Analog point value (counts)
FFFF F026 0008	4	F	0,0: Minimum value (EU)
FFFF F026 000C	4	F	0,0: Maximum value (EU)
FFFF F026 0010	4	UI	0,0: Reserved
(Additional points follow in order on 40 hex boundaries.) (Additional modules follow in order on 1000 hex boundaries.)			
Last valid address for this area: FFFF F029 FFFF			

(Expanded) Analog Point Write—Read/Write

PAC-R
EB
SB

For I/O units with firmware version 8.0 and newer, this area of the memory map replaces the F0B00000 area (page 152). To allow for future growth, this area has space for 4096 points, which is 64 points per module on 64 modules. (Current SNAP I/O units offer a maximum of 16 modules.)

See "IEEE Float Data" on page 60 for help in interpreting data.

Only the first point on the first module is shown in the table. Successive points and modules follow the same pattern.

Starting Address	Length (Hex)	Type	Description
FFFF F02A 0000	4	F	Module 0, Point 0 (0,0): Analog point value (Engineering Units)
FFFF F02A 0004	4	F	0,0: Analog point value (counts)
FFFF F02A 0008	4	--	0,0: Reserved
FFFF F02A 000C	4	F	0,0: TPO period (units of time in seconds. Valid range: 0.25 to 64.0 seconds, in 0.25 steps.) (Not for E2s.)
FFFF F02A 0010	4	--	0,0: Reserved
FFFF F02A 0014	4	UI	0,0: Load cell fast settle level—For SNAP-AILC modules only. (Not for E2s.) Use with load cell filter weight (next address) to get filtered readings faster. Valid values: 0–32,767. 0 = disabled;
FFFF F02A 0018	4	UI	0,0: Load cell filter weight—For SNAP-AILC modules only. (Not for E2s.) Valid values: 0–255; default = 128. A larger value increases filtering. Use with F0B00014 to get the filtered reading faster. Note that 0, 1, or 255 value disables fast settle level (F0B00014). The second channel on the module is the filtered reading of the first channel.
FFFF F02A 001C	24	--	0,0: Reserved
(Additional points follow in order on 40 hex boundaries.) (Additional modules follow in order on 1000 hex boundaries.)			
Last valid address for this area: FFFF F02D FFFF			

(Expanded) Digital Point Read & Clear—Read/Write

PAC-R EB SB G4EB2

For I/O units with firmware version 8.0 and newer, this area of the memory map replaces the F0F00000 area (page 166). To allow for future growth, this area has space for 4096 points, which is 64 points per module on 64 modules. (Current SNAP I/O units offer a maximum of 16 modules.)

When you read data from this area, the data is returned and then cleared. You can use this section for all SNAP digital points; for high-density modules, you can also use "SNAP High-Density Digital Read and Clear—Read/Write" on page 180.

Only the first point on the first module is shown in the table. Successive points and modules follow the same pattern.

Starting Address	Length (Hex)	Type	Description
FFFF F02E 0000	4	UI	Module 0, Point 0 (0,0): Digital point feature data (counter value, quadrature counter value)
FFFF F02E 0004	4	UI	0,0: On latch
FFFF F02E 0008	4	UI	0,0: Off latch
FFFF F02E 000C	C	--	0,0: Reserved
(Additional points follow in order on 18 hex boundaries.) (Additional modules follow in order on 600 hex boundaries.)			
Last valid address for this area: FFFF F02F 7FFF			

Status Area Read—Read Only

PAC-R
PAC-S
EB
SB
UIO
EIO
SIO
LCE
E1
E2
G4EB2

This is the only area that can be read before sending a Powerup Clear message to the Opto 22 memory-mapped device. See "Sending Powerup Clear" on page 96 for more information.

Starting Address	Length (Hex)	Type	Description
FFFF F030 0000	4	UI	Memory Map revision number
FFFF F030 0004	4	UI	Powerup Clear flag (0 = OK; anything else means a Powerup Clear is needed)
FFFF F030 0008	4	UI	Busy flag (0 = not busy; anything else means the unit is busy and cannot process your request)
FFFF F030 000C	4	I	Last error code (see page 109)
FFFF F030 0010	2	UI	Transaction label for previous transaction (lower 6 bits)
FFFF F030 0012	2	UI	Source address of the unit that sent the request. Zero-filled unless you use the optional Source_ID parameter in the packet. See "Parameters" on page 105.
FFFF F030 0014	4	UI	Error address for last error (lower 32 bits of offset)
FFFF F030 0018	4	UI	Loader revision. 1st byte: major version number; 2nd byte: minor version number; 3rd byte: revision (0=A, 1=B, 2=R); 4th byte: letter (0=a, 1=b, 2=c, etc.)
FFFF F030 001C	4	IP	Firmware (kernel) revision. 1st byte: major version number; 2nd byte: minor version number; 3rd byte: revision (0=A, 1=B, 2=R); 4th byte: letter (0=a, 1=b, 2=c, etc.)
FFFF F030 0020	4	UI	Unit type of the device: 0x00000052 = OPTOEMU-SNR-DR2 0x00000056 = OPTOEMU-SNR-DR1 0x00000058 = G4EB2 0x0000005A = OPTOEMU-SNR-3V 0x0000005C = SNAP-PAC-SRA 0x00000062 = SNAP-PAC-SB2 0x00000064 = SNAP-PAC-SB1 0x00000066 = SNAP-PAC-R2-W 0x00000068 = SNAP-PAC-R1-W 0x0000006A = SNAP-PAC-S1-W 0x0000006C = SNAP-PAC-S2-W 0x00000070 = SNAP-PAC-EB2-W 0x00000072 = SNAP-PAC-EB1-W 0x00000074 = SNAP-PAC-EB2 0x00000076 = SNAP-PAC-EB1 0x00000078 = SNAP-PAC-R2 0x0000007A = SNAP-PAC-R1 0x0000007C = SNAP-PAC-S1 0x00000083 = SNAP-ENET-S64 0x0000008A = SNAP-UPN-ADS 0x0000008C = SNAP-UP1-M64 0x00000092 = SNAP-UP1-D64 0x00000093 = SNAP-UP1-ADS 0x00000094 = SNAP-WLAN-FH-ADS 0x00000097 = SNAP-ENET-D64 0x00000098 = SNAP-B3000-ENET or SNAP-ENET-RTC 0x000000E1 = E1 0x000000E2 = E2 0x00000193 = SNAP-LCE
FFFF F030 0024	1	UI	Hardware revision (Month)
FFFF F030 0025	1	UI	Hardware revision (Day)
FFFF F030 0026	2	UI	Hardware revision (Year)
FFFF F030 0028	4	UI	Number of bytes of installed RAM
FFFF F030 002C	2	--	Pad for alignment (Does not apply to SBs.)
FFFF F030 002E	6	UI	ENET1 MAC address. First three bytes of all Opto 22 Ethernet devices' MAC addresses are 00-A0-3D. (Does not apply to SBs.)

Starting Address	Length (Hex)	Type	Description
FFFF F030 0034	4	IP	ENET1 TCP/IP address (1.2.3.4 format: four 1-byte integers) (Not for SBs) (For ENET2, see FFFF F050.)
FFFF F030 0038	4	IP	ENET1 TCP/IP subnet mask (1.2.3.4 format: four 1-byte integers) (Not for SBs)
FFFF F030 003C	4	IP	ENET1 TCP/IP default gateway (1.2.3.4 format: four 1-byte integers) (Not for SBs)
FFFF F030 0040	4	IP	ENET1 DNS server address (1.2.3.4 format: four 1-byte integers) (Not for SBs)
FFFF F030 0044	4	--	Reserved
FFFF F030 0048	4	UI	Device sends BootP request (UIO, EIO, SIO, LCE) or DHCP request (E1, E2) when turned on: 0 = Only if device's IP address is 0.0.0.0. (Current IP address is static.) 1 = Whenever device is turned on. (Current IP address is dynamic.) (Does not apply to SBs.)
FFFF F030 004C	4	B	Degrees are in F or C (Valid only for I/O unit with analog capability) 0 = degrees C; non-0 = degrees F
FFFF F030 0050	4	--	Reserved
FFFF F030 0054	4	UI	Watchdog time in milliseconds (unsigned integer). 0 means watchdog is disabled.
FFFF F030 0058	4	UI	TCP/IP minimum Response Timeout (RTO) in milliseconds. TCP/IP calculates the RTO based on past network response times. The default is a minimum 3000 msec between retries. (Does not apply to E1s, E2s, SBs.)
FFFF F030 005C	4	UI	Digital (single or 4-channel) scan counter. Counts the number of scans of these digital modules; can be used for benchmarking. (Does not apply to E1s.)
FFFF F030 0060	4	UI	Analog and high-density digital scan counter. Counts the number of scans of analog and HDD modules; can be used for benchmarking.
FFFF F030 0064	4	UI	Initial RTO. (Does not apply to E1s, E2s, SBs.)
FFFF F030 0068	4	UI	TCP number of retries. (Does not apply to E1s, E2s, SBs.)
FFFF F030 006C	4	UI	TCP idle session timeout. (Does not apply to E1s, E2s, SBs.)
FFFF F030 0070	4	UI	Ethernet errors: late collisions. This and the next two sets of addresses indicate problems on the Ethernet network, often due to length of the segment or number of devices. Late collisions are not normal. Normally two Ethernet hosts trying to talk at once collide early in the packet and both back off, or the second host waits. (Does not apply to E1s, E2s, SBs.)
FFFF F030 0074	4	UI	Ethernet errors: excessive collisions. Indicate that all the backup attempts to communicate have been exhausted. (Does not apply to E1s, E2s, SBs.)
FFFF F030 0078	4	UI	Ethernet errors: other. (Does not apply to E1s, E2s, SBs.)
FFFF F030 007C	4	M	"Smart" modules present. Bitmask showing the presence of analog, serial, high-density digital, and PID modules. (Does not apply to E1s, E2s, G4EB2s.) Present = 1; not present = 0. Use as a troubleshooting tool to make sure modules are online.

Starting Address	Length (Hex)	Type	Description
FFFF F030 0080	20	S-ZT	Device's part number (string)
FFFF F030 00A0	10	S-ZT	Firmware version date
FFFF F030 00B0	10	S-ZT	Firmware version time
FFFF F030 0100	4	UI	ARCNET reconfigs detected. Indicates that a "smart" module (analog, serial, high-density digital) has been added, removed, or reset. This and the other addresses starting with ARCNET refer to the ARCNET bus used on the rack for communication between the I/O processor and I/O modules. (Does not apply to E1s, E2s, G4EB2s.)
FFFF F030 0104	4	UI	ARCNET reconfigs initiated by I/O unit. Error on the rack's ARCNET bus. (Does not apply to E1s, E2s, G4EB2s.) Not a concern unless it happens frequently. If there are no analog, serial, high-density digital, or PID modules on the rack, ignore it. If there is at least one of these modules on the rack, make sure the rack has adequate voltage.
FFFF F030 0108	4	UI	Number of times the device closed the session because it was idle. (Does not apply to E1s, E2s, SBs.)
FFFF F030 010C	4	UI	Milliseconds since powerup. Value rolls over after 4,294,967,295 ms, which is equal to 49.71 days. For longer periods of time, use address F030 0160 (in seconds) or F030 0228 (64-bit field, in milliseconds). (Does not apply to E1s or E2s.)
FFFF F030 0110	4	UI	Ethernet MAC resets since powerup. Caused by electrical noise. (Does not apply to E1s, E2s, SBs.)
FFFF F030 0114	4	UI	Digital output point resets since powerup. Caused by electrical noise. (Does not apply to E1s or E2s.)
FFFF F030 0118	4	UI	Digital interrupt failures since powerup. Digital counter may have missed counts. (Does not apply to E1s or E2s.)
FFFF F030 011C	4	UI	Total number of PID loops available on this I/O unit (96 for SNAP PAC R-series, 32 for UIO, 16 for EIO, none for other I/O units).
FFFF F030 0120	4	UI	ARCNET transmit attempts since powerup. (Not on E1s, E2s, G4EB2s.)
FFFF F030 0124	4	UI	ARCNET other (node not found, etc.). (Not on E1s, E2s, G4EB2s.)
FFFF F030 0128	4	UI	ARCNET ACKs. (Not on E1s, E2s, G4EB2s.)
FFFF F030 012C	4	UI	ARCNET delay between transmit attempt and ACK for most recent attempt. (Not on E1s, E2s, G4EB2s.)
FFFF F030 0130	4	UI	ARCNET timeout value (msec.). (Not on E1s, E2s, G4EB2s.)
FFFF F030 0134	4	UI	ARCNET timeouts. (Not on E1s, E2s, G4EB2s.)
FFFF F030 0138	4	UI	ARCNET receive interrupts. (Not on E1s, E2s, G4EB2s.)
FFFF F030 013C	4	--	Reserved
FFFF F030 0140	4	F	Milliseconds per analog/HDD scan. Value of -1 means scanner is not running.
FFFF F030 0144	4	F	Milliseconds per digital (4-ch) scan. Value of -1 means scanner is not running.

Starting Address	Length (Hex)	Type	Description
FFFF F030 0148	4	UI	Number of single or 4-channel digital modules supported (0, 8, or 16). Useful for SNAP-PAC-R1s; R1s with serial numbers below 600,000 support only 8 (in the first 8 rack positions); newer modules support 16. Added in firmware 8.1; older firmware doesn't include this address.
FFFF F030 014C	4	UI	(SB brains only) Brain's unique serial number (SB brains do not have a MAC address).
FFFF F030 0150	4	UI	(SB brains only) Multidrop address
FFFF F030 0154	4	UI	(SB brains only) Baud rate.
FFFF F030 0158	4	UI	(SB brains only) Number of framing errors during serial transmission (no stop bit detected). Should be zero under normal operation; if it is anything other than zero, the baud rate may be incorrect or there may be noise on the serial bus.
FFFF F030 015C	4	UI	(SB brains only) Number of FIFO overrun errors. Should be zero under normal operation. Contact Opto 22 Product Support if any other number appears in this field.
FFFF F030 0160	4	UI	Elapsed time since powerup (in seconds). Use instead of F030 010C for longer periods of time. Value rolls over after 4,294,967,295 seconds, which is equal to 49,710 days. (Does not apply to E1s or E2s.)
FFFF F030 0164	C4	--	Reserved
FFFF F030 0228	8	UI	Milliseconds since powerup. Use instead of F030 010C for longer periods of time in milliseconds. Value rolls over after 584,542,046 years. (Does not apply to E1s or E2s.)
FFFF F030 0230	4	UI	(PAC-R and PAC-S only, firmware 8.4a and higher) Current boot device: 0 = Flash memory; 1 = microSD card
FFFF F030 0234	14	--	Reserved
FFFF F030 0248	4	I	Result of some Status Area Write commands (F0380000 on page 127). Results apply only to commands 03, 04, 0D, 0E, 0F, and 10. 0 = Operation was successful. -109 = microSD card is read only. -110 = microSD card is not inserted. -111 = Controller doesn't support microSD cards.
Last valid address for this area: FFFF F030 024B			

Communications Port Configuration—Read/Write

PAC-R
PAC-S
UIO
LCE

Use this area of the memory map to configure the serial ports on a SNAP PAC, SNAP-LCE, or SNAP Ultimate controller. See the device's user guide for numbers of ports and port locations. See the *PAC Manager User's Guide* for configuration information.

Starting Address	Length (hex)	Type	Description
FFFF F031 0400	4	UI	Port 0: Control function for serial port. Default is 1. LCE, PAC-R, or UIO must be rebooted if value is changed; PAC-S does not have to be rebooted. See device's user guide for port locations and details. 0 = None (used with PAC Control to connect directly to serial I/O units or devices, depending on controller capabilities) 1 = PPP (for modem use). Only one port should be configured for PPP. 2 = Nokia - M2M (not valid for PAC-S1)
FFFF F031 0404	4	UI	Port 0: Enable logging. 0 = Disabled; 1 = Enabled. Default is 0. If control function is set to None (0), received characters are only logged if they've been retrieved by the PAC Control strategy. All data received and transmitted is logged to a binary file named Comm<port number>.log, stored in the device's file system. Each byte of data is stored as two bytes: the first is data direction (capital letter `I` = received data; capital letter `O` = transmitted data); the second is the data itself. Maximum size of the log file is 16,384 bytes; beyond this limit, new data replaces the oldest data. A null character (ASCII code 0) with no data direction prepended follows the most recent data logged. This value takes effect immediately. If you use this port to retrieve the log file from the file system, always set this value to 0 before retrieval, or the log will be filled with data generated by the file transfer.
FFFF F031 0408	4	UI	Port 1: Control function for serial port. Default is 0. Valid value for PAC-S1 is 0. See device's user guide for port locations and details. 0 = None (used with PAC Control to connect directly to serial I/O units or devices, depending on controller capabilities) 1 = PPP (for modem use). Only one port should be configured for PPP. 2 = Nokia - M2M.
FFFF F031 040C	4	UI	Port 1: Enable logging. 0 = Disabled; 1 = Enabled. Default is 0. Same as port 0 logging.
FFFF F031 0410	4	UI	Port 1: Control function for serial port. Default is 0. Valid value for PAC-S1 is 0. See device's user guide for port locations and details. 0 = None (used with PAC Control to connect directly to serial I/O units or devices, depending on controller capabilities) 1 = PPP (for modem use). Only one port should be configured for PPP. 2 = Nokia - M2M.
FFFF F031 0414	4	UI	Port 1: Control function for serial port. Default is 0. Valid value for PAC-S1 is 0. See device's user guide for port locations and details. 0 = None (used with PAC Control to connect directly to serial I/O units or devices, depending on controller capabilities) 1 = PPP (for modem use). Only one port should be configured for PPP. 2 = Nokia - M2M.
FFFF F031 0418		--	Reserved
FFFF F031 0800	4	UI	(PAC-S & PAC-R only) Port 0: Read only. Detects a failed link by checking CD signal. 0 = deasserted (failed); 1 = asserted.
FFFF F031 0804	4	UI	(PAC-S & PAC-R only) Port 0: Read/write. DTR; hangs up the modem (modem must be configured with &D2). 0 = deassert; 1 = assert.

<table>
<tr><th>Starting Address</th><th>Length (hex)</th><th>Type</th><th>Description</th></tr>
<tr><td>FFFF F031 0808</td><td>4</td><td>UI</td><td>(PAC-S & PAC-R only) Port 0: Read only. DSR. 0 = deasserted; 1 = asserted</td></tr>
<tr><td>FFFF F031 080C</td><td>4</td><td>UI</td><td>(PAC-S & PAC-R only) Port 0: Read/write. RTS. 0 = deassert; 1 = assert</td></tr>
<tr><td>FFFF F031 0810</td><td>4</td><td>UI</td><td>(PAC-S & PAC-R only) Port 0: Read only. CTS. 0 = deasserted; 1 = asserted</td></tr>
<tr><td>FFFF F031 0814</td><td>4</td><td>UI</td><td>(PAC-S & PAC-R only) Port 0: Read only. RI. 0 = deasserted; 1 = asserted</td></tr>
<tr><td>FFFF F031 0818</td><td>10</td><td>--</td><td>Reserved</td></tr>
<tr><td>FFFF F031 082C</td><td>4</td><td>UI</td><td>(PAC-S & PAC-R only) Port 1: Read/write. RTS. 0 = deassert; 1 = assert</td></tr>
<tr><td>FFFF F031 0830</td><td>4</td><td>UI</td><td>(PAC-S & PAC-R only) Port 1: Read only. CTS. 0 = deasserted; 1 = asserted</td></tr>
<tr><td>FFFF F031 0834</td><td>14</td><td>--</td><td>Reserved</td></tr>
<tr><td>FFFF F031 084C</td><td>4</td><td>UI</td><td>(PAC-S only) Port 2: Read/write. RTS. 0 = deassert; 1 = assert</td></tr>
<tr><td>FFFF F031 0850</td><td>--</td><td>--</td><td>Reserved</td></tr>
<tr><td>FFFF F031 0C00</td><td>4</td><td>UI</td><td>PAC-R and PAC-S only (each has one programmable LED, the PPP LED). Indicates what controls the PPP LED to indicate the current PPP state. Can be stored to flash.
0 = None
1 = Programmable LED state address (F0310E00) controls the LED.
2 = (Default) PPP service controls the LED; state is indicated as follows:
<table>
<tr><th>LED action</th><th>Indicates state</th></tr>
<tr><td>Off</td><td>Idle (0) or Disabled (9)</td></tr>
<tr><td>Slow blink green</td><td>Outgoing connecting (1) or
Incoming connecting (5)</td></tr>
<tr><td>Solid green</td><td>Outgoing connected (2) or
Incoming connected (6)</td></tr>
<tr><td>Slow blink orange</td><td>Outgoing disconnecting (3) or
Incoming disconnecting (7)</td></tr>
<tr><td>Solid orange</td><td>Listen (4)</td></tr>
<tr><td>Slow blink red</td><td>Shutting down (8)</td></tr>
</table></td></tr>
<tr><td>FFFF F031 0C04</td><td></td><td>--</td><td>Reserved</td></tr>
<tr><td>FFFF F031 0E00</td><td>4</td><td>UI</td><td>PAC-R and PAC-S only. State for programmable PPP LED.
0 = Off
1 = Green
2 = Red
3 = Orange (green and red at the same time)</td></tr>
<tr><td>FFFF F031 0E04</td><td>--</td><td>--</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Starting Address</th><th>Length (hex)</th><th>Type</th><th>Description</th></tr>
<tr><td>FFFF F031 1100</td><td>4</td><td>UI</td><td>(PAC-S2 only) Port 0 Mode<table>
<tr><td>0</td><td>None (all signals set to high impedance inputs)</td></tr>
<tr><td>232</td><td>RS-232 (default)</td></tr>
<tr><td>24850</td><td>2-wire RS-485, no bias, no termination</td></tr>
<tr><td>24852</td><td>2-wire RS-485, bias, no termination</td></tr>
<tr><td>24851</td><td>2-wire RS-485, no bias, termination</td></tr>
<tr><td>24853</td><td>2-wire RS-485, bias, termination</td></tr>
<tr><td>44850</td><td>4-wire RS-485, no bias, no termination</td></tr>
<tr><td>44852</td><td>4-wire RS-485, bias, no termination</td></tr>
<tr><td>44851</td><td>4-wire RS-485, no bias, termination</td></tr>
<tr><td>44853</td><td>4-wire RS-485, bias, termination</td></tr>
</table></td></tr>
<tr><td>FFFF F031 1104</td><td>4</td><td>UI</td><td>(PAC-S2 only) Port 1 Mode, same values as above</td></tr>
<tr><td>FFFF F031 1108</td><td>4</td><td>UI</td><td>(PAC-S2 only) Port 2 Mode, same as above except default is 24853</td></tr>
<tr><td>FFFF F031 110C</td><td>4</td><td>UI</td><td>(PAC-S2 only) Port 3 Mode, same as above except default is 24853</td></tr>
<tr><td colspan="4">Last valid address for this area: FFFF F031 110F</td></tr>
</table>

Serial Pass-Through—Read/Write

Use this area of the memory map only to talk to a SNAP PAC SB (serial) brain through an S-series or R-series controller's serial port (called *serial pass-through)*. This area contains a configuration section and a data section. Configure the port first; then write to the data section to send a memory map packet; finally, read the data section to read the response.

Configuration—For each serial port, write all fields in the configuration area at once, in one memory map packet. Once the port has been enabled, you can change only the Timeout field by writing to it individually.

Data—Write to send an OptoMMP packet out the serial port. Read to see the response to the packet you sent. You will not get an ACK back until the response is ready to be read.

Starting Address	Length (Hex)	Type	Description
Configuration Section (write all in one memmap packet)			
FFFF F032 9000	4	UI	Port 0: enable serial pass-through (0 = disabled; non-zero = enabled)
FFFF F032 9004	4	UI	Port 0: data rate (bps)
FFFF F032 9008	4	UI	Port 0: number of data bits

<table>
<tr><th>Starting Address</th><th>Length (Hex)</th><th>Type</th><th>Description</th></tr>
<tr><td>FFFF F032 900C</td><td>4</td><td>UI</td><td>Port 0: parity:
<table>
<tr><td>0x4E</td><td>None</td></tr>
<tr><td>0x45</td><td>Even</td></tr>
<tr><td>0x4F</td><td>Odd</td></tr>
</table>
<table>
<tr><td>0x4D</td><td>Mark</td></tr>
<tr><td>0x53</td><td>Space</td></tr>
<tr><td>0x44</td><td>9th bit multidrop</td></tr>
</table></td></tr>
<tr><td>FFFF F032 9010</td><td>4</td><td>UI</td><td>Port 0: number of stop bits</td></tr>
<tr><td>FFFF F032 9014</td><td>4</td><td>UI</td><td>Port 0: 2-wire or 4-wire (0x48 = 2-wire; 0x46 = 4-wire)</td></tr>
<tr><td>FFFF F032 9018</td><td>4</td><td>UI</td><td>Port 0: timeout in ms. Number of ms to wait for a response. This address can be written to individually after initial configuration.</td></tr>
<tr><td>FFFF F032 A000</td><td>--</td><td>--</td><td>Port 1 configuration (same pattern as port 0, above)</td></tr>
<tr><td>FFFF F032 B000</td><td>--</td><td>--</td><td>Port 2, same pattern</td></tr>
<tr><td>FFFF F032 C000</td><td>--</td><td>--</td><td>Port 3, same pattern</td></tr>
<tr><td>FFFF F032 D000</td><td>--</td><td>--</td><td>Port 4, same pattern</td></tr>
<tr><td>FFFF F032 E000</td><td>--</td><td>--</td><td>Port 5, same pattern</td></tr>
<tr><td colspan="4">Data Section (write to transmit; read to see response)</td></tr>
<tr><td>FFFF F032 9100</td><td>4</td><td>UI</td><td>Port 0: Write = 8-bit multidrop address of the serial brain
Read = 16-bit length of the response field (F0329104)</td></tr>
<tr><td>FFFF F032 9104</td><td>4</td><td>UI</td><td>Port 0: Write = 8-bit packet type identifier (OptoMMP packet = 0x02)
Read = OptoMMP response.</td></tr>
<tr><td>FFFF F032 9108</td><td>4</td><td>UI</td><td>Port 0 (Write only): 16-bit length of the packet (F032910C)</td></tr>
<tr><td>FFFF F032 910C</td><td>4</td><td>UI</td><td>Port 0 (Write only): packet containing memory map command</td></tr>
<tr><td>FFFF F032 A100</td><td>--</td><td>--</td><td>Port 1 configuration (same pattern as port 0, above)</td></tr>
<tr><td>FFFF F032 B100</td><td>--</td><td>--</td><td>Port 2, same pattern</td></tr>
<tr><td>FFFF F032 C100</td><td>--</td><td>--</td><td>Port 3, same pattern</td></tr>
<tr><td>FFFF F032 D100</td><td>--</td><td>--</td><td>Port 4, same pattern</td></tr>
<tr><td>FFFF F032 E100</td><td>--</td><td>--</td><td>Port 5, same pattern</td></tr>
<tr><td colspan="4">Last valid address for this area: FFFF F032 EFFF</td></tr>
</table>

Date and Time Configuration—Read/Write

PAC-R
EB
SB
UIO
EIO
G4EB2

Use this area of the memory map to view or change the date and time on SNAP PAC R-series, SNAP PAC EB or SB, SNAP Ultimate, or SNAP-ENET-RTC, which contain a real-time clock. On other devices you cannot set the date and time; this area shows the elapsed time since the device was last turned on.

Starting Address	Length (hex)	Type	Description
FFFF F035 0000	16	S-ZT	Set date and time. Format: YYYY-MM-DD HH:MM:SS.00
Last valid address for this area: FFFF F035 0023			

Status Area Write—Read/Write

PAC-R
PAC-S
EB
SB
UIO
EIO
SIO
LCE
E1
E2
G4EB2

Although this area is read/write, you would normally write to it. Although the area as a whole applies to all memory-mapped devices, some addresses or codes apply to some devices and not others (see features table on page 9).

When you write to this area, you must write all four bytes, or you'll receive an "invalid memory address" error. You can write to this area using either a quadlet or a block write.

Starting Address	Length (hex)	Type	Description
FFFF F038 0000	4	UI	Operation code (Some results returned in F0300248, page 122.) 0x00000**01** – Send Powerup Clear 0x00000**02** – Reset to defaults. CAUTION: See "Details on Operation Codes in Address F0380000" on page 129. 0x00000**03** – Store all configuration data to flash. CAUTION: See details on page 129. 0x00000**04** – Erase all configuration from flash memory and microSD card. CAUTION: See details on page 129. 0x00000**05** – Reset hardware, which is just like cycling power to the device. If point configuration information has not been stored to flash or flash has been cleared (op code 04), points are reset to defaults. After a hardware reset, the device waits for a Powerup Clear before communicating. 0x00000**06** – Clear "Digital Events - Old" configuration (for expanded digital events in firmware 8.1 or higher, use 0x0000000A below). 0x00000**07** – Clear Alarms configuration. 0x00000**08** – Clear PPP configuration. (Not on EB, SB, G4EB2.) 0x00000**09** – Clear Email configuration. (Does not apply to SB.) 0x0000000**A** – Clear "Digital Events - Expanded" configuration (For firmware 8.0 and lower, clears Timers configuration). 0x0000000**B** – Clear PID loops configuration (analog devices only). 0x0000000**C** – Clear data log. 0x0000000**D** – Save configuration and IP address to microSD card. See details on page 129. 0x0000000**E** – Erase configuration and IP address from microSD card. See details on page 129. 0x0000000**F** – Erase firmware from microSD card. See page 129. 0x00000**10** – Erase strategy from microSD card. See page 129. 0x**87654321** – Boot to loader. (UIO, EIO, SIO, and LCE only)

Starting Address	Length (hex)	Type	Description
FFFF F038 0004	4	UI	Set device to send BootP or DHCP request when turned on. 0 (default for all except E1 & E2) = Send BootP or DHCP request only if device's IP address is 0.0.0.0. Current IP address is saved to flash memory and becomes a static IP address. 1 (default for E1 & E2) = Always send BootP or DHCP request when device is turned on. IP address is dynamic (temporary). (Does not apply to SBs.)
FFFF F038 0008	4	B	Set degrees in F or C. 0 (default) = degrees C; non-0 = degrees F
FFFF F038 000C	4	--	Reserved
FFFF F038 0010	4	UI	Set watchdog time in milliseconds (unsigned integer). 0 disables watchdog.
FFFF F038 0014	4	UI	Set TCP/IP minimum Response Timeout (RTO) in milliseconds. (Does not apply to E1s, E2s, or SBs.) For SNAP PAC R-series, SNAP PAC EB, SNAP Ultimate, and SNAP Simple I/O units, the default is a minimum 100 msec between retries. For SNAP Ethernet I/O units, the default is a minimum 3000 msec between retries. For a high-speed application such as motion control, you can write a lower minimum RTO to this address.
FFFF F038 0018	4	UI	Set TCP/IP initial RTO. (Does not apply to E1s, E2s, or SBs.)
FFFF F038 001C	4	UI	Set TCP/IP number of retries. (Does not apply to E1s, E2s, or SBs.)
FFFF F038 0020	4	UI	TCP idle session timeout in milliseconds. 0 = disable. (Does not apply to E1s, E2s, or SBs.)
FFFF F038 0024	4	UI	Wireless configuration: mode (Does not apply to SBs.)
FFFF F038 0028	4	UI	Wireless configuration: ESS identification (Does not apply to SBs.)
FFFF F038 002C	20	--	Reserved
FFFF F038 004C	4	UI	Set maximum digital scan interval for single or 4-channel digital modules, in milliseconds. (This value is ignored in firmware 8.1 and higher; use F038 0294 instead. Not for E1s or E2s.) Default = 1000 (1 sec). Setting the value to -1 shuts down the scanner until power is cycled.
FFFF F038 0050	4	UI	Set maximum analog and high-density digital scan interval (msec). (Not for E1s, E2s, G4EB2s.) Setting the value to -1 shuts down the analog/HDD scanner until power is cycled. As of firmware version 8.1, analog modules with more than 4 points are scanned no faster than every 30 ms, and analog modules with four or fewer points are scanned no faster than every 6 ms, in order to maintain synchronization with the module.
FFFF F038 0054	4	M	Scanner Flags. Used to maximize speed under special circumstances. Binary mask: 0 = Off; 1 = On. Use one or a combination of the following bits. All except the first one require that the I/O unit be restarted. 0x01 = Handle alarms in digital (single/4-ch) scanner rather than analog/HDD scanner 0x02 = Stop analog/HDD/ serial module scanner 0x04 = Stop digital (single/4-ch) scanner 0x08 = Stop PAC Control engine (SNAP PAC R-series and UIO only) (Example: 0100 = Stop digital (4-ch) scanner flag is on; all other scanner flags are off. This means that alarms are handled in the analog/HDD scanner, the analog/HDD scanner is running, the digital (4-ch) scanner is stopped, and the PAC Control engine is running.)

Starting Address	Length (hex)	Type	Description
FFFF F038 0058	4	M	ON mask for scanner flags (see F038 0054). 1 = On; 0 = No change
FFFF F038 005C	4	M	OFF mask for scanner flags (see F038 0054). 1 = On; 0 = No change
FFFF F038 0060	F4	--	Reserved
FFFF F038 0154	40	S-ZT	Host Name (E1 and E2 only). Used only if the unit has a dynamic IP address assigned by the network.
FFFF F038 0194	100	S-ZT	Domain Name (E1 and E2 only).
FFFF F038 0294	4	UI	Sets the digital scan interval. Most reading and writing to 4-ch digital I/O happens asynchronously. However, the following features are updated once per digital scan: digital events - old, digital events -enhanced (formerly Timers), frequency and period measurement, data logging, and alarms if digitally scanned (see F038 0054, Scanner Flags).
FFFF F038 0298	4	F	Value shown when actual value is out of range. Default is –32,768
Last valid address for this area: FFFF F038 0297			

Details on Operation Codes in Address F0380000

NOTE: Many of these op codes affect a microSD card, if the device has one. See the device's user's guide for important information before using the card.

Results for op codes 03, 04, 0D, 0E, 0F, 10, and 11 are returned in F0300248 (see page 122).

0x02—Resets device to defaults (equivalent to an 04 op code followed by an 05). Requires a Powerup Clear after resetting.
CAUTION: If you have firmware 9.0 or newer and a microSD card is present, this code also *erases* any firmware, strategy, IP address, and configuration data from the card. This code does not erase other data files on the microSD card.
If you have older firmware versions, these files on the microSD card are not erased.

0x03—Stores all configuration data to flash memory, so it is restored when the device is turned on.
CAUTION: If you have firmware 9.0 or newer and a microSD card is present, and if the card already has IP address and configuration data stored on it (see op code 0D), code 03 also *overwrites* the configuration data and IP address on the card with the new configuration.
Exception: On a controller using Secure Strategy Distribution (SSD), configuration data is stored to flash but not to the microSD card.
Other data files on the microSD card are not affected.
If you have older firmware versions, nothing is saved to the microSD card.

0x04—Erases all configuration data from flash memory and microSD card. Details:

- Erases all IP address and configuration data from the microSD card, if one is present and has IP and configuration data stored on it (see op code 0D). Firmware and strategy files on the card are not erased. Other data files are not erased.
- Clears error information in the status area (F030000C, etc.: error code, transaction label, source address, error address).
- Clears gains and offsets, latches, min/max data, and counters. Deactivates counters.
- Turns off digital outputs.
- Sets analog outputs to zero scale (0 counts).

0x0D—Requires firmware 9.0 or newer. Applies only to devices with a microSD card slot and a card in place. Stores the controller's IP address and configuration data on the microSD card.
Exception: Data is not stored to a card in a controller using Secure Strategy Distribution (SSD).

0x0E—Requires firmware 9.0 or newer. Applies only to devices with a microSD card slot and a card in place. Erases IP address and configuration data from the microSD card. Does not erase any other files.

0x0F—Requires firmware 9.0 or newer. Applies only to devices with a microSD card slot and a card in place. Erases firmware from the microSD card in the boot directory and in all locations mentioned in the firmware command file. Also erases firmware command and response files from the card. Does not erase IP address, configuration, strategy, or data files from the card.

0x10—Requires firmware 9.0 or newer. Applies only to devices with a microSD card slot and a card in place. Erases strategy from the microSD card at /sdcard0/strategy/. Does not erase IP address, configuration data, firmware, or firmware command files from the card. Does not erase data files.

Modbus Configuration—Read/Write

PAC-R
EB
UIO
EIO
SIO
E1
E2
G4EB2

Use this area of the memory map for changing the format of Modbus floats. After you change format, be sure to store the information to flash using the Status Write area of the device (page 127). The Modbus memory map and information are in form #1678, the *Modbus/TCP Protocol Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F039 0000	4	UI	Modbus float format (unsigned integer) 0x00000000 = Big Endian 0x00000001 = Big Endian, word swapped. Most significant bit of float is in most significant register.
Last valid address for this area: FFFF F039 0003			

Network Security Configuration—Read/Write

PAC-R
PAC-S
EB
SB
UIO
EIO
SIO
LCE
E1
E2
G4EB2

See the *PAC Manager User's Guide* for information on setting up network security for Ethernet-based devices. Also see "FTP User Name/Password Configuration—Read/Write" on page 142.

Starting Address	Length (hex)	Type	Description
FFFF F03A 0004	4	UI	OptoMMP port. Default is 2001.
FFFF F03A 0008	4	UI	Modbus/TCP port. Default is 502.
FFFF F03A 000C	4	UI	(PAC-S, PAC-R, EB, UIO, and EIO only) SNMP port for SNMP agent. Default is 161.
FFFF F03A 0010	4	UI	(PAC-S, PAC-R, EB, UIO, and LCE only; limited use for E1 and E2) FTP port. Default is 21.
FFFF F03A 0020	4	IP	IP Filter Address 0
FFFF F03A 0024	4	IP	IP Filter Mask 0
FFFF F03A 0028	4	IP	IP Filter Address 1
FFFF F03A 002C	4	IP	IP Filter Mask 1
FFFF F03A 0030	4	IP	IP Filter Address 2
FFFF F03A 0034	4	IP	IP Filter Mask 2
FFFF F03A 0038	4	IP	IP Filter Address 3
FFFF F03A 003C	4	IP	IP Filter Mask 3
FFFF F03A 0040	30	IP	Additional IP Filter Addresses and Filter Masks (4–9)
FFFF F03A 0070	4	UI	Stop incoming broadcasts. 0 = off (default)
FFFF F03A 0074	4	UI	PAC Control host task listen port (TCP). Default: 22001. 0 = Do not listen for commands via TCP.
FFFF F03A 0078	4	UI	Enable/disable EtherNet/IP protocol. 0=disabled; 1=enabled (default)
Last valid address for this area: FFFF F03A 0073			

SSI Module Configuration—Read/Write

PAC-R
EB
SB

Use this area to configure SSI (serial synchronous interface) modules on the rack. The table shows both ports on the first module only; additional modules follow the same pattern, starting on even 100 hex boundaries, with each port starting at even 40 hex boundaries.

Before configuring or using these modules, see Opto 22 form 1931, the *SNAP SSI (Serial Synchronous Interface) Module User's Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F03A 1000	4	UI	Module 0, Port 0: Number of bits in SSI frame (4–32)
FFFF F03A 1004	4	UI	Module 0, Port 0: Clock divider
FFFF F03A 1008	4	UI	Module 0, Port 0: Data delay (clock cycles)
FFFF F03A 100C	4	UI	Module 0, Port 0: Most significant data bit offset (0–28)
FFFF F03A 1010	4	UI	Module 0, Port 0: Data bits in the frame (4–32)
FFFF F03A 1014	4	UI	Module 0, Port 0: Error bit offset within the frame. 0–31 = Offset; –1 = No error bit
FFFF F03A 1018	4	UI	Module 0, Port 0: Error bit meaning. 0=high bit indicates error; 1=low bit indicates error
FFFF F03A 101C	4	UI	Module 0, Port 0: Data coding. 0=binary; 1=Gray code
FFFF F03A 1020	4	UI	Module 0, Port 0: Enable scanning. 0=disabled; 1=enabled
FFFF F03A 1024	1C	--	Pad for alignment
FFFF F03A 1040	4	UI	Module 0, Port 1: Number of bits in SSI frame (4–32)
FFFF F03A 1044	4	UI	Module 0, Port 1: Clock divider
FFFF F03A 1048	4	UI	Module 0, Port 1: Data delay (clock cycles)
FFFF F03A 104C	4	UI	Module 0, Port 1: Most significant data bit offset (0–28)
FFFF F03A 1050	4	UI	Module 0, Port 1: Data bits in the frame (4–32)
FFFF F03A 1054	4	UI	Module 0, Port 1: Error bit offset within the frame. 0–31 = Offset; –1 = No error bit
FFFF F03A 1058	4	UI	Module 0, Port 1: Error bit meaning. 0=high bit indicates error; 1=low bit indicates error
FFFF F03A 105C	4	UI	Module 0, Port 1: Data coding. 0=binary; 1=Gray code
FFFF F03A 1060	4	UI	Module 0, Port 1: Enable scanning. 0=disabled; 1=enabled
FFFF F03A 1064	1C	--	Pad for alignment
(Additional modules follow in order on even 100 hex boundaries.)			
FFFF F03A 1F40	40	UI	Module 15, Port 1 Configuration
Last valid address for this area: FFFF F03A 1FFF			

Serial Module Identification—Read Only

PAC-REB UIO EIO SIO

Use this area to find out about SNAP serial communication modules on the rack. The table shows the first two modules only; additional modules follow the same pattern, starting on even 10 hex boundaries.

For important information on using these modules, see Opto 22 form 1191, the *SNAP Serial Communication Module User's Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F03A 7F00	1	UI	Serial module 0: Module hardware type. 0xF0 = SNAP-SCM-232 0xF1 = SNAP-SCM-485 0xF6 = SNAP-SCM-PROFI 0xF8 = SNAP-SCM-MCH16 0xF9 = SNAP-SCM-W2 0xFA = SNAP-SCM-SSI 0xFB = SNAP-SCM-ST2 0xFC = SNAP-SCM-CAN2B
FFFF F03A 7F01	1	UI	Serial module 0: Module hardware subtype 0 = Modules manufactured before June 2003 1 = Modules manufactured June 2003 and after
FFFF F03A 7F02	1	UI	Serial module 0: Module hardware revision (month)
FFFF F03A 7F03	1	UI	Serial module 0: Module hardware revision (day)
FFFF F03A 7F04	2	UI	Serial module 0: Module hardware revision (year)
FFFF F03A 7F06	4	UI	Serial module 0: Module hardware loader revision
FFFF F03A 7F0A	4	UI	Serial module 0: Module hardware firmware revision
FFFF F03A 7F0E	2	--	Pad for alignment
FFFF F03A 7F10	1	UI	Module 1: Module hardware type
FFFF F03A 7F11	1	UI	Module 1: Module hardware subtype
FFFF F03A 7F12	1	UI	Module 1: Module hardware revision (month)
FFFF F03A 7F13	1	UI	Module 1: Module hardware revision (day)
FFFF F03A 7F14	2	UI	Module 1: Module hardware revision (year)
FFFF F03A 7F16	4	UI	Module 1: Module hardware loader revision
FFFF F03A 7F1A	4	UI	Module 1: Module hardware firmware revision
FFFF F03A 7F1E	2	--	Pad for alignment
(Additional modules follow in order on even 10 hex boundaries.)			
Last valid address for this area: FFFF F03A 7FFA			

Serial Module Configuration—Read/Write

PAC-R
EB
UIO
EIO
SIO

Use this area to configure SNAP serial communication modules on the rack. Only two ports—the ports for a serial module in position 0 on the rack—are shown in the table. (SNAP-SCM-PROFI, SNAP-SCM-MCH16, and SNAP-SCM-CAN2B modules have only one port, port A.) Ports for modules in other positions on the rack follow the same pattern and start on even 10 hex boundaries. See the table on page 135 for a list of module and port numbers.

For important information on using these modules, see Opto 22 form 1191, the *SNAP Serial Communication Module User's Guide*. For the SNAP-SCM-MCH16, see form #1673, the *SNAP PAC Motion Control Subsystem User's Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F03A 8000	4	UI	Port A on serial module in position 0 on the rack: IP port number for access to serial port. Default is 22500.
FFFF F03A 8004	4	UI	Port A, position 0: Baud rate
FFFF F03A 8008	1	UI	Port A, position 0: Parity. None = 0x4E; even = 0x45; odd = 0x4F SNAP-SCM-PROFI and SNAP-SCM-MCH16 = even (0x45)
FFFF F03A 8009	1	UI	Port A, position 0: Data Bits SNAP-SCM-PROFI = 7; SNAP-SCM-MCH16 = 8 only.
FFFF F03A 800A	1	UI	Port A, position 0: Stop Bits. SNAP-SCM-PROFI = 1
FFFF F03A 800B	1	UI	Port A, position 0: Hardware flow control. 1 = Yes, 0 = No
FFFF F03A 800C	1	UI	Port A, position 0: Send test message on powerup 1 = Yes; 0 = No
FFFF F03A 800D	3	--	Pad for alignment
FFFF F03A 8010	4	UI	Port B on serial module in position 0 on the rack: IP port number for access to serial port. Default is 22501.
FFFF F03A 8014	4	UI	Port B, position 0: Baud rate
FFFF F03A 8018	1	UI	Port B, position 0: Parity
FFFF F03A 8019	1	UI	Port B, position 0: Data Bits
FFFF F03A 801A	1	UI	Port B, position 0: Stop Bits
FFFF F03A 801B	1	UI	Port B, position 0: Hardware flow control. 1 = Yes, 0 = No
FFFF F03A 801C	1	UI	Port B, position 0: Send test message on powerup
FFFF F03A 801D	3	--	Pad for alignment
(Additional ports follow in order on even 10 hex boundaries. See port table, below.)			
FFFF F03A 8200	4	UI	Port A, position 0: End-of-message (EOM) characters. The device can check any one of up to four characters as the EOM indicator. [Example: 0x0D0A0000 looks for a 13 (hex 0D) or 10 (hex 0A)] EOM checking occurs only when using the serial module port with serial events. Max. message size = 250 bytes (Messages that exceed 249 bytes are received as 249-byte chunks followed by a smaller chunk containing all characters up to the EOM character.)
FFFF F03A 8204	C	--	Pad for alignment

Starting Address	Length (hex)	Type	Description
FFFF F03A 8210	4	UI	Port B, position 0: End-of-message characters.
(Additional ports follow in order on even 10 hex boundaries.)			
FFFF F03A 8400	1	UI	Position 0: Module mode of operation (0 = 2-wire; 1 = 4-wire) SNAP-SCM-PROFI =2-wire (0).
FFFF F03A 8401	1	UI	Position 1: Module mode of operation (0 = 2-wire; 1 = 4-wire) SNAP-SCM-PROFI =2-wire (0).
(Additional modules follow in order.)			
Last valid address for this area: FFFF F03A 840F			

PAC-REB UIO EIO SIO

Serial Module and Port Numbers

For quick reference, the following table shows serial modules and ports, their default IP port numbers, and their starting memory map addresses, beginning with FFFF F03A 8000. Remember that SNAP-SCM-PROFI, SNAP-SCM-MCH16, and SNAP-SCM-CAN2B modules have only one port, Port A.

Module	Port	Default IP Port	Memory Map Address
0	A	22500	8000
0	B	22501	8010
1	A	22502	8020
1	B	22503	8030
2	A	22504	8040
2	B	22505	8050
3	A	22506	8060
3	B	22507	8070
4	A	22508	8080
4	B	22509	8090
5	A	22510	80A0
5	B	22511	80B0
6	A	22512	80C0
6	B	22513	80D0
7	A	22514	80E0
7	B	22515	80F0

Module	Port	Default IP Port	Memory Map Address
8	A	22516	8100
8	B	22517	8110
9	A	22518	8120
9	B	22519	8130
10	A	22520	8140
10	B	22521	8150
11	A	22522	8160
11	B	22523	8170
12	A	22524	8180
12	B	22525	8190
13	A	22526	81A0
13	B	22527	81B0
14	A	22528	81C0
14	B	22529	81D0
15	A	22530	81E0
15	B	22531	81F0

Wiegand Serial Module Configuration—Read/Write

PAC-R
EB
UIO
EIO

(Does not apply to SNAP Simple I/O) Use this area to configure SNAP Wiegand serial communication modules on the rack. Only a few modules and ports are shown in the table. Other ports and modules on the rack follow the same pattern. See the table on page 137 for a list of module and port numbers.

For important information on using these modules, see Opto 22 form 1191, *SNAP Serial Communication Module User's Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F03A 8500	4	UI	Wiegand module 0: Loader version number
FFFF F03A 8504	4	UI	Wiegand module 0: Firmware (kernel) version number
FFFF F03A 8508	4	UI	Wiegand module 1: Loader version number
FFFF F03A 850C	4	UI	Wiegand module 1: Firmware (kernel) version number
(Modules 2–15 follow in order.)			
FFFF F03A 8600	4	UI	Port A on Wiegand module in position 0 on the rack: IP port number for access to serial port. Default is 22500.
FFFF F03A 8604	4	UI	Port A, position 0: Data format (O for Opto format, 26, 30, 32, 34, 35, 36, 37, 40, or C for custom format)
FFFF F03A 8608	4	UI	Port A, position 0: Total Length (hex) of data in the transmission
FFFF F03A 860C	4	UI	Port A, position 0: First bit of the site code
FFFF F03A 8610	4	UI	Port A, position 0: Length of the site code, in bits
FFFF F03A 8614	4	UI	Port A, position 0: First bit of the badge code (should be the next bit after the site code)
FFFF F03A 8618	4	UI	Port A, position 0: Length of the badge code, in bits
FFFF F03A 861C	24	--	Pad for alignment
FFFF F03A 8640	4	UI	Port B on Wiegand module in position 0 on the rack: IP port number for access to serial port. Default is 22501.
FFFF F03A 8644	4	UI	Port B, position 0: Data format (O for Opto format, 26, 30, 32, 34, 35, 36, 37, 40, or C for custom format)
FFFF F03A 8648	4	UI	Port B, position 0: Total length of data in the transmission
FFFF F03A 864C	4	UI	Port B, position 0: First bit of the site code
FFFF F03A 8650	4	UI	Port B, position 0: Length of the site code, in bits
FFFF F03A 8654	4	UI	Port B, position 0: First bit of the badge code (should be the next bit after the site code)
FFFF F03A 8658	4	UI	Port B, position 0: Length of the badge code, in bits
FFFF F03A 865C	24	--	Pad for alignment
(Additional ports follow in order on even 40 hex boundaries. See table on the next page.)			
Last valid address for this area: FFFF F03A 8DC3			

PAC-REB UIO EIO

Wiegand Module and Port Numbers

For quick reference, the following table shows Wiegand modules and ports, their default IP port numbers, and their starting memory map addresses, beginning with FFFF F03A 8600.

Module	Port	Default IP Port	Memory Map Address
0	A	22500	8600
0	B	22501	8640
1	A	22502	8680
1	B	22503	86C0
2	A	22504	8700
2	B	22505	8740
3	A	22506	8780
3	B	22507	87C0
4	A	22508	8800
4	B	22509	8840
5	A	22510	8880
5	B	22511	88C0
6	A	22512	8900
6	B	22513	8940
7	A	22514	8980
7	B	22515	89C0

Module	Port	Default IP Port	Memory Map Address
8	A	22516	8A00
8	B	22517	8A40
9	A	22518	8A80
9	B	22519	8AC0
10	A	22520	8B00
10	B	22521	8B40
11	A	22522	8B80
11	B	22523	8BC0
12	A	22524	8C00
12	B	22525	8C40
13	A	22526	8C80
13	B	22527	8CC0
14	A	22528	8D00
14	B	22529	8D40
15	A	22530	8D80
15	B	22531	8DC0

SNAP-SCM-CAN2B Serial Module Configuration—Read/Write

PAC-R EB

Use this area to configure SNAP-SCM-CAN2B serial communication modules on the rack. Each module has only one port. Only two modules are shown in the table; others on the rack follow the same pattern. See the table on page 137 for a list of module and port numbers.

If the Data Masks and Filters are all set to 0, then all CAN packets will be received. If you want the SNAP-SCM-CAN2B module to provide filtering, then configure the Data Masks and Filters. Always start with the highest priority mask and filter, Data Mask 0 and Filter 0. See additional information in "CAN Packet Table" on page 139.

For important information on using these modules, see Opto 22 form 1191, *SNAP Serial Communication Module User's Guide*, and form 1537, the *SNAP-SCM-CAN2B Module Data Sheet*.

Starting Address	Length (hex)	Type	Description
FFFF F03A 9000	4	UI	Module 0: TCP/UDP Port Number.
FFFF F03A 9004	4	UI	Module 0: BAUD Rate. Supports 1000000, 500000, 250000, 125000, 100000, 50000, 20000, 10000 bps
FFFF F03A 9008	4	UI	Module 0: Data Mask 0 for Filters 0 and 1. This mask determines which bits in the CAN ID are examined. If any mask bit is set to a zero, that bit will automatically be accepted, regardless of the filter bit. See truth table below.
FFFF F03A 900C	4	UI	Module 0: Filter 0. Highest priority filter. See truth table below.
FFFF F03A 9010	4	UI	Module 0: Filter 1. If filter value is 0, unused, then Filter 0 will be used instead. See truth table below.
FFFF F03A 9014	4	UI	Module 0: Data Mask 1 for Filters 2, 3, 4, and 5. This mask determines which bits in the CAN ID are examined. If any mask bit is set to a zero, that bit will automatically be accepted, regardless of the filter bit. See truth table below. If mask is 0, unused, then Data Mask 0 will be used instead.
FFFF F03A 9018	4	UI	Module 0: Filter 2. If filter is 0, unused, then Filter 1 will be used instead. See truth table below.
FFFF F03A 901C	4	UI	Module 0: Filter 3. If filter is 0, unused, then Filter 2 will be used instead. See truth table below.
FFFF F03A 9020	4	UI	Module 0: Filter 4. If filter is 0, unused, then Filter 3 will be used instead. See truth table below.
FFFF F03A 9024	4	UI	Module 0: Filter 5. If filter is 0, unused, then Filter 4 will be used instead. Lowest priority filter. See truth table below.
FFFF F03A 9028	4	UI	Module 0: Error Code. Where the SNAP-SCM-CAN2B reports any errors it encounters. See Error Code table below. Related to LED 4 above.
FFFF F03A 902C	4	UI	Module 1: TCP/UDP Port Number.
FFFF F03A 9030	4	UI	Module 1: BAUD Rate. Supports 1000000, 500000, 250000, 125000, 100000, 50000, 20000, 10000 bps
FFFF F03A 9034	4	UI	Module 1: Data Mask 0 for Filters 0 and 1. This mask determines which bits in the CAN ID are examined. If any mask bit is set to a zero, that bit will automatically be accepted, regardless of the filter bit. See truth table below.

Starting Address	Length (hex)	Type	Description
FFFF F03A 9038	4	UI	Module 1: Filter 0. Highest priority filter. See truth table below.
FFFF F03A 903C	4	UI	Module 1: Filter 1. If filter value is 0, unused, then Filter 0 will be used instead. See truth table below.
FFFF F03A 9040	4	UI	Module 1: Data Mask 1 for Filters 2, 3, 4, and 5. This mask determines which bits in the CAN ID are examined. If any mask bit is set to a zero, that bit will automatically be accepted, regardless of the filter bit. See truth table below. If mask is 0, unused, then Data Mask 0 will be used instead.
FFFF F03A 9044	4	UI	Module 1: Filter 2. If filter is 0, unused, then Filter 1 will be used instead. See truth table below.
FFFF F03A 9048	4	UI	Module 1: Filter 3. If filter is 0, unused, then Filter 2 will be used instead. See truth table below.
FFFF F03A 904C	4	UI	Module 1: Filter 4. If filter is 0, unused, then Filter 3 will be used instead. See truth table below.
FFFF F03A 9050	4	UI	Module 1: Filter 5. If filter is 0, unused, then Filter 4 will be used instead. Lowest priority filter. See truth table below.
FFFF F03A 9054	4	UI	Module 1: Error Code. Where the SNAP-SCM-CAN2B reports any errors it encounters. See Error Code table below. Related to LED 4 above.
(Additional modules follow in order on even 2C hex boundaries. See table on page 140.)			
Last valid address for this area: FFFF F03A 92BF			

CAN Packet Table

The table below shows how CAN packets are accepted or rejected. It applies to memory map addresses for Data Masks and Filters. X = Don't care; so, for example, if you want to receive CAN ID 0x00**FF05**00 and only care about the part in bold, then set your Mask to 0x00FFFF00 and your Filter to 0x00FF0500. .

Data Mask Bit	Filter Bit	CAN ID Bit	Accept/Reject
0	X	X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

CAN Error Codes

The SNAP-SCM-CAN2B keeps track of how many errors have occurred. Possible errors are CRC Error, Acknowledge Error, Form Error, Bit Error, and Stuff Error. If one of these errors is detected, LED 4 is lit and the error counter is incremented.

To clear the error condition in the module, read the Error Code memory map address (0xF03A9028 for module 0). The table below defines the error codes returned in this address

Error Code	Description
0	Error—Active State. The SNAP-SCM-CAN2B has received less than 96 errors.
-1	Error—Active State. The SNAP-SCM-CAN2B has received 96 or more errors but less than 128 errors.
-2	Receiver Overflow. A CAN packet was dropped. This error occurs if the SNAP-SCM-CAN2B can't keep up with the traffic on the CAN bus.
-3	Error—Passive State. The SNAP-SCM-CAN2B has received 128 or more errors but less than 255 errors.
-4	Bus—Off State. The SNAP-SCM-CAN2B has received 255 or more errors. CAN packets can neither be received or transmitted. Since the SNAP-SCM-CAN2B only receives CAN packets, this state should never occur.

PAC-REB

SNAP-SCM-CAN2B Module and Port Numbers

For quick reference, the following table shows SNAP-SCM-CAN2B modules and ports, their default IP port numbers, and their starting memory map addresses, beginning with FFFF F03A 9000. Each module has only one port, Port A.

Module	Port	Default IP Port	Memory Map Address
0	A	22500	0xF03A9000
1	A	22502	0xF03A902C
2	A	22504	0xF03A9058
3	A	22506	0xF03A9084
4	A	22508	0xF03A90B0
5	A	22510	0xF03A90DC
6	A	22512	0xF03A9108
7	A	22514	0xF03A9134

Module	Port	Default IP Port	Memory Map Address
8	A	22516	0xF03A9160
9	A	22518	0xF03A918C
10	A	22520	0xF03A91B8
11	A	22522	0xF03A91E4
12	A	22524	0xF03A9210
13	A	22526	0xF03A923C
14	A	22528	0xF03A9268
15	A	22530	0xF03A9294

SNMP Configuration—Read/Write

PAC-S
PAC-R
EB
UIO
EIO
G4EB2

(Does not apply to SNAP Simple I/O) Use this area to configure SNMP if you are sending messages to an enterprise management system, such as Computer Associates' Unicenter TNG® or Hewlett Packard's Open View®, using the Simple Network Management Protocol. For more information on using SNMP, see page 49.

Starting Address	Length (hex)	Type	Description
FFFF F03C 0000	20	S-ZT	sysName—the name assigned to the SNAP device as a managed node within the SNMP management system
FFFF F03C 0020	20	S-ZT	sysLocation—the physical location of the device
FFFF F03C 0040	20	S-ZT	sysContact—the ID of the contact person for the device
FFFF F03C 0060	4	UI	Enable authentication trap. Yes = 1; No = 0. Default is 0.
FFFF F03C 0064	4	UI	Enable cold start trap. Yes = 1; No = 0. Default is 0.
FFFF F03C 0068	14	S-ZT	Community 1: Name of community (string)
FFFF F03C 007C	4	UI	Community 1: Set read access privileges. Yes = 1; No = 0. Default is 0.
FFFF F03C 0080	4	UI	Community 1: Set write access privileges. Yes = 1; No = 0. Default is 0.
FFFF F03C 0084	4	UI	Community 1: Set trap access privileges. Yes = 1; No = 0. Default is 0.
FFFF F03C 0088	14	S-ZT	Community 2: Name of community (string)
FFFF F03C 009C	4	UI	Community 2: Set read access privileges. Yes = 1; No = 0. Default is 0.
FFFF F03C 00A0	4	UI	Community 2: Set write access privileges. Yes = 1; No = 0. Default is 0.
FFFF F03C 00A4	4	UI	Community 2: Set trap access privileges. Yes = 1; No = 0. Default is 0.
(Communities 3–8 follow in order.)			
FFFF F03C 0168	14	S-ZT	Host 1: Name of community host belongs to (string)
FFFF F03C 017C	4	IP	Host 1: IP address
FFFF F03C 0180	14	S-ZT	Host 2: Name of community host belongs to (string)
FFFF F03C 0194	4	IP	Host 2: IP address
FFFF F03C 0198	14	S-ZT	Host 3: Name of community host belongs to (string)
FFFF F03C 01AC	4	IP	Host 3: IP address
(Hosts 4–16 follow in order.)			
FFFF F03C 0308	4	UI	SNMP trap destination port. Default is 162.
FFFF F03C 030C	4	UI	SNMP trap version: 0 = agent sends SNMPv1 traps; 1 = agent sends SNMPv2 notifications. Default = 0. Affects all traps/notifications; takes effect immediately.
Last valid address for this area: FFFF F03C 030B			

FTP User Name/Password Configuration—Read/Write

PAC-R
PAC-S
EB
G4EB2

Use this area to configure the user name and password for FTP use. For more information on using FTP and the controller or brain's file system, see the *PAC Manager User's Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F03D 0000	40	S-ZT	User name for FTP server access
FFFF F03D 0040	40	S-ZT	Password for FTP server access
Last valid address for this area: FFFF F03D 0079			

PPP Configuration—Read/Write

PAC-R
PAC-S
UIO
EIO
LCE

Use this area to configure PPP if you are communicating with the controller or brain via a modem connection. For more information on setting up PPP, see the *PAC Manager User's Guide*.

Starting Address	Length (hex)	Type	Description
FFFF F03E 0000	4	IP	Local IP address for PPP interface. Default is 192.168.0.1
FFFF F03E 0004	4	IP	Remote IP address. This IP address is assigned to devices that connect to the brain or controller and request an IP address. *This address must be on the same subnet as the Local IP address.* Default is 192.168.0.2
FFFF F03E 0008	4	UI	Serial port speed. Default is 19,200 bps.
FFFF F03E 000C	4	UI	Serial port parity. Default is none. None = 0x10; even = 0x4; odd = 0x0
FFFF F03E 0010	4	UI	Serial port stop bits. Possible values:1 or 2. Default is 1.
FFFF F03E 0014	4	UI	Serial port data bits. Possible values: 6–8. Default is 8.
FFFF F03E 0018	40	S-ZT	Modem initialization string (null terminated). This string is sent to the modem once when the brain or controller boots. The modem must be configured to **ignore DTR** and **use no flow control**. The modem is reset (ATZ) between each call. We suggest that settings be saved to a profile in the modem's NVRAM and the modem configured to load that profile every time it is reset. The modem initialization string can be used to perform this task whenever the brain or controller powers up. See the device's user's guide for more information. Default is: AT&D0^M~~~~
FFFF F03E 0058	4	IP	Subnet mask for Local IP address. Default is 0.0.0.0—Required only if you are using classless IP addressing. If not, leave it at 0 and the brain or controller will calculate the appropriate subnet from the Local IP address.
FFFF F03E 005C	4	UI	Maximum number of times PPP will retry to authenticate a link. Default is 3.
FFFF F03E 0060	4	UI	Serial port flow control. 0 = none; 1 = hardware (RTS/CTS)

Starting Address	Length (hex)	Type	Description
FFFF F03E 0064	40	S-ZT	Modem hangup string. This string is sent by the brain or controller to the modem when it wants the modem to hang up. (Note: A ^t in this string forces an LCP terminate request packet in a PPP frame to be sent, which can be useful to force packet-switched data devices with a local PPP interface to hang up and enter command mode.) Default is: ~~~+++~~~ATH0^M~~~
FFFF F03E 00A4	4	UI	(PAC-S and PAC-R only) Send commands to the PPP service: 0 = None 1 = Start PPP service. If PPP is Disabled, rereads all PPP configuration settings and starts the service. Check PPP status in F03EB800. 2 = Stop PPP service. Applies only if PPP is not disabled. Read F03EB800 to monitor shutdown progress. 3 = Start outgoing link. Applies only if PPP is Idle or Listening (see F03EB800). Starts an outgoing PPP connection. 4 = Stop link. Applies only if PPP status is Outgoing Connected or Incoming Connected (see F03EB800). Closes the PPP link.
FFFF F03E 00A8	4	UI	Echo request period (in seconds). Default = 10 s.
FFFF F03E 00AC	4	UI	Echo request retries. Default = 3.
FFFF F03E 00B0	4	UI	Connection establishment timeout (in seconds). Default = 60 s.
Last valid address for this section: FFFF F03E 00B3			
FFFF F03E 6000	4	UI	Enable PPP for incoming calls. Default is 0. Enable = 1, Disable = 0
FFFF F03E 6004	4	UI	Set default gateway to PPP interface. Default is 0. Yes = 1, No = 0. If enabled, all IP packets addressed to destinations that are not on the same subnet as the Ethernet interface will be sent out the PPP interface. Enable this only when the remote device can route packets to their ultimate destination. The setting is in effect only when an incoming PPP session is active.
FFFF F03E 600C	40	S-ZT	Modem listen string (null terminated). Default is: ATS0=1^M~” Prior to waiting for a call, the brain or controller resets the modem and sends this string to it. The string is sent every time the brain or controller prepares to listen for a call.
FFFF F03E 604C	4	UI	Inactivity timeout. Number of seconds an incoming link can remain idle before the link is closed. Default is 30.
FFFF F03E 6050	10	S-ZT	(Applies to firmware R5.0 and older*) Login (null terminated). Login name for authentication on incoming PPP sessions.
FFFF F03E 6060	10	S-ZT	(Applies to firmware R5.0 and older*) Password (null terminated). Password for authentication on incoming PPP sessions.
FFFF F03E 6070	40	S-ZT	(Applies to firmware R5.1 and newer*) Login (null terminated). Login name for authentication on incoming PPP sessions.
FFFF F03E 60B0	40	S-ZT	(Applies to firmware R5.1 and newer*) Password (null terminated). Password for authentication on incoming PPP sessions.
Last valid address for this section: FFFF F03E 60EF			
FFFF F03E B000	4	UI	Enable PPP for outgoing calls (dial on demand). Enable = 1; disable = 0. Default is 0.

Starting Address	Length (hex)	Type	Description
FFFF F03E B004	4	UI	Specify local IP address. Yes = 1; No = 0. Default is 0. For outgoing calls, the brain or controller can either specify the IP address it wants to use (entered in FFFF F03E 0000) or ask the remote device to assign an address.
FFFF F03E B008	4	UI	Set default gateway to PPP interface. Yes = 1; No = 0. Default is 0. If enabled, all IP packets addressed to destinations that are not on the same subnet as the Ethernet interface will be sent out the PPP interface. Enable this only when the remote device can route packets to their ultimate destination. The setting is in effect only when an outgoing PPP session is active.
FFFF F03E B00C	4	--	Reserved
FFFF F03E B010	10	S-ZT	(Applies to firmware R5.0 and older*) Login (null terminated). Login name for authentication on outgoing PPP sessions.
FFFF F03E B020	10	S-ZT	(Applies to firmware R5.0 and older*) Password (null terminated). Password for authentication on outgoing PPP sessions.
FFFF F03E B030	4		PPP inactivity timeout in seconds. Default is 30. If the brain or controller sends no traps for this number of seconds after the PPP session is negotiated, the modem will hang up. A zero in this field disables the timer.
FFFF F03E B034	40	S-ZT	Phone number to dial for outgoing PPP connections (null terminated).
FFFF F03E B074	4	UI	Maximum connection time in seconds. Default is 0 (disabled). Maximum amount of time an outgoing PPP connection can stay connected after successful negotiation.
FFFF F03E B078	4	UI	Maximum number of times the brain or controller will try redialing if the first attempt fails. Default is 0.
FFFF F03E B07C	4	UI	Retry interval in seconds. Default is 0. Number of seconds the brain or controller will wait before trying to redial after the first attempt fails.
FFFF F03E B080	4	UI	Disable time in seconds. Default is 0. If the maximum connect time or maximum number of retries has been reached, the outgoing PPP dialer waits this long before doing anything.
FFFF F03E B084	4	UI	Link always connected for outgoing PPP. If always connected, brain or controller dials out only on powerup and anytime the PPP link goes down. Default is 0 (not always connected). 0 is used for most applications that use circuit-switched data service to communicate. 1 is used for most applications that use packet-switched data service.
FFFF F03E B088	40	S-ZT	(Applies to firmware R5.1 and newer*) Login (null terminated). Login name for authentication on outgoing PPP sessions.
FFFF F03E B0C8	40	S-ZT	(Applies to firmware R5.1 and newer*) Password (null terminated). Password for authentication on outgoing PPP sessions.
Last valid address for this area: FFFF F03E B107			

* Changed login and password addresses: As of firmware version 5.1, login and password were lengthened to 0x40 bytes. Applications using the old memory map addresses will still work, as values will be read from the old addresses and stored to the new addresses. To take advantage of the longer login and password, however, you need to change your application to use the new addresses.

PPP Status—Read Only

PAC-R
PAC-S
UIO
EIO
LCE

Use this area to check status on Point-to-Point Protocol (PPP) communication using a modem.

Starting Address	Length (hex)	Type	Description
FFFF F03E B800	4	UI	Connection status (differs for SNAP devices: EIO: 0 - Idle 1 - Connecting 2 - Listening 3 - Connected 4 - Disconnecting 5 - Disconnected PAC-S, LCE, and UIO: 0 - Idle 1 - Outgoing Connecting 2 - Outgoing Connected 3 - Outgoing Disconnecting 4 - Listening 5 - Incoming Connecting 6 - Incoming Connected 7 - Incoming Disconnecting 8 - Shutting Down 9 - Disabled
FFFF F03E B804	4	UI	Connection type: 0 - Outgoing; 1 - Incoming; 2 - None
FFFF F03E B808	4	UI	Number of retries (times the outgoing dialer has tried to dial out). When this number reaches the number configured in F03E B078 as the maximum number of retries, the disable timer starts running and the outgoing dialer stops dialing until the disable timer expires
FFFF F03E B80C	4	UI	Number of milliseconds left on the idle timer before it expires and closes the connection. 0 = disabled. Activity on the PPP link resets the dle timer to 1000 times the value in F03E B030 for outgoing connections or F03E 604C for incoming connections.This timer limits the duration of a PPP link with no activity.
FFFF F03E B810	4	UI	Number of milliseconds left on the retry timer before the outgoing dialer tries to dial out again. The retry timer starts running when an outgoing dial attempt fails. Its initial value is the value configured in F03E B07C.
FFFF F03E B814	4	UI	Number of milliseconds left on the disable timer. While the disable timer is active, the outgoing dialer is prevented from dialing out. Initial value is the value configured in F03E B080.
FFFF F03E B818	4	UI	Number of milliseconds left on the outgoing connect timer. This timer limits the length of an outgoing PPP connection regardless of activity. When an outgoing PPP link is established, this timer is set to the value configured in F03E B074. When the timer expires, the outgoing PPP link is terminated and the disable timer is started.
FFFF F03E B81C	4	UI	Number of buffered outgoing frames waiting for a PPP link to be established.
FFFF F03E B820	4	UI	UART TX buffer overruns. Number of bytes of transmit data that were dropped because the modem was not ready to receive it. The brain or controller waits for up to one second for the modem to be ready. After that, the brain or controller drops one byte of transmit data and flags the modem internally as offline. When the device has another byte of transmit data to send, it checks once to see if the modem is ready to receive it; if not, it drops the byte. If the modem is ready, the device sends the data and flags the modem internally as online.

Starting Address	Length (hex)	Type	Description
FFFF F03E B824	4	IP	Current local IP address (SNAP Ultimate and SNAP-LCE only). Current IP address assigned to the PPP interface by a remote peer; valid only while the PPP link is connected. Applies only If outgoing PPP is configured not to specifiy its local IP address.
Last valid address for this area: FFFF F03F FFFF			

Streaming Configuration—Read/Write

PAC-R EB UIO EIO SIO G4EB2

Use this area to configure data streaming from the I/O unit to a host. For more information, see page 51. This area is stored to flash.

Starting Address	Length (hex)	Type	Description
FFFF F03F FFC4	4	B	Enable or disable I/O mirroring (Boolean) 0 = disable; non-0 = enable
FFFF F03F FFC8	4	UI	Beginning address of the data you want to stream. Do not use if you are streaming the entire streaming section shown on page 168. Use only to stream a portion of the streaming section or some other portion of the memory map.
FFFF F03F FFCC	4	UI	Size of data to stream. Do not use if you are streaming the entire streaming section shown on page 168. Use only to stream a portion of the streaming section or some other portion of the memory map. Maximum data size: 1480 bytes.
FFFF F03F FFD0	4	B	Streaming On/Off (Boolean). 0 = off; non-0 = on.
FFFF F03F FFD4	4	UI	Streaming Interval, in milliseconds (unsigned integer)
FFFF F03F FFD8	4	UI	IP port number to stream to
FFFF F03F FFDC	4	--	Reserved
FFFF F03F FFE0	4	IP	Stream target IP address #1
FFFF F03F FFE4	4	IP	Stream target IP address #2
FFFF F03F FFE8	4	IP	Stream target IP address #3
FFFF F03F FFEC	4	IP	Stream target IP address #4
FFFF F03F FFF0	4	IP	Stream target IP address #5
FFFF F03F FFF4	4	IP	Stream target IP address #6
FFFF F03F FFF8	4	IP	Stream target IP address #7
FFFF F03F FFFC	4	IP	Stream target IP address #8
Last valid address for this area: FFFF F03F FFFF			

Digital Bank Read—Read Only

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Applies to most digital modules. Do not use for SNAP high-density digital modules; see "SNAP High-Density Digital—Read Only" on page 179.

For help in interpreting data, see "Mask Data" on page 58. For general information on using counters, see page 31.

Starting Address	Length (hex)	Type	Description
FFFF F040 0000	8	M	State of digital points (mask)
FFFF F040 0008	8	M	State of on-latches (mask)
FFFF F040 0010	8	M	State of off-latches (mask)
FFFF F040 0018	8	M	Active counters (mask)
FFFF F040 0020	4	M	Reserved for completion of pulse measurement or one-time frequency or period measurement on digital points 32–63
FFFF F040 0024	4	M	Completion of pulse measurement or one-time frequency or period measurement on digital points 0–31. 1 = pulse or measurement complete 0 = incomplete or not applicable
FFFF F040 0100	100	UI	Counter data (unsigned integer)
Last valid address for this area: FFFF F040 01FF			

Digital Bank Write—Read/Write

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

NOTE: To clear counters and latches, use the "Digital Read and Clear" area on page 166.

Applies to most digital modules. Do not use for SNAP high-density digital modules; see "SNAP High-Density Digital Write—Read/Write" on page 181.

Although this area is read/write, you would normally write to it. For help in formatting data, see "Mask Data" on page 58. For general information on using counters, see page 31.

Starting Address	Length (hex)	Type	Description
FFFF F050 0000	8	M	Turn on (mask)
FFFF F050 0008	8	M	Turn off (mask)
FFFF F050 0010	8	M	Activate counters (mask)
FFFF F050 0018	8	M	Deactivate counters (mask)
Last valid address for this area: FFFF F050 001F			

Analog Bank Read—Read Only

PAC-R
EB
SB
UIO
EIO
SIO
E2

For help in interpreting data, see "IEEE Float Data" on page 60.

Starting Address	Length (hex)	Type	Description
FFFF F060 0000	100	F	Analog data (float—Engineering Units)
FFFF F060 0100	100	F	Analog data (float—counts)
FFFF F060 0200	100	F	Analog Min Value (float—Engineering Units)
FFFF F060 0300	100	F	Analog Max Value (float—Engineering Units)
Last valid address for this area: FFFF F060 03FF			

Analog Bank Write—Read/Write

PAC-R
EB
SB
UIO
EIO
SIO
E2

Although this area is read/write, you would normally write to it. For help in formatting data, see "IEEE Float Data" on page 60.

Starting Address	Length (hex)	Type	Description
FFFF F070 0000	100	F	Analog output (float—Engineering Units)
FFFF F070 0100	100	F	Analog output (float—counts)
Last valid address for this area: FFFF F070 01FF			

Digital Point Read—Read Only

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

NOTE: To clear counters and latches, use the "Digital Read and Clear" area on page 166.

Applies to most digital modules. Do not use for SNAP high-density digital modules; see "SNAP High-Density Digital—Read Only" on page 179.

See "Digital Point Data (4-Channel Modules)" on page 60 for help in interpreting data. For information on using counters, see page 31. Only the first three points are shown in the table. Each successive point starts on an even 40 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F080 0000	4	B	Module 0, Point 0 (0,0): Point state. This address is read/write. Read the point state at this address or write to it: 0 = turn point off; non-0 = turn point on
FFFF F080 0004	4	B	0,0: On-latch state. 0 = off; non-0 = on
FFFF F080 0008	4	B	0,0: Off-latch state. 0 = off; non-0 = on
FFFF F080 000C	4	B	0,0: Counter active? 0 = off; non-0 = on
FFFF F080 0010	4	UI	0,0: Feature value (counter, pulse measurement, frequency or period measurement). Units and resolution are in increments of 100 µsec. Max value is 4,294,967,295. When it reaches max, the value rolls over to zero and continues.
FFFF F080 0040	4	B	0,1: Point state. This address is read/write. Read the point state at this address or write to it: 0 = turn point off; non-0 = turn point on
FFFF F080 0044	4	B	0,1: On-latch state. 0 = off; non-0 = on
FFFF F080 0048	4	B	0,1: Off-latch state. 0 = off; non-0 = on
FFFF F080 004C	4	B	0,1: Counter active? 0 = off; non-0 = on
FFFF F080 0050	4	UI	0,1: Feature value (counter, pulse measurement, frequency or period measurement). Max value is 4,294,967,295. When it reaches max, the value rolls over to zero and continues
FFFF F080 0080	4	B	0, 2: Point state. This address is read/write. Read the point state at this address or write to it: 1 = turn point on; 0 = turn point off.
FFFF F080 0084	4	B	0,2: On-latch state. 0 = off; non-0 = on
FFFF F080 0088	4	B	0,2: Off-latch state. 0 = off; non-0 = on
FFFF F080 008C	4	B	0,2: Counter active? 0 = off; non-0 = on
FFFF F080 0090	4	UI	0,2: Feature value (counter, pulse measurement, frequency or period measurement). Units and resolution are in increments of 100 µsec. Max value is 4,294,967,295. When it reaches max, the value rolls over to zero and continues
(Additional points follow in order on even 40 hex boundaries.)			
FFFF F080 0FC0	4	B	15,3: Point state. This address is read/write. Read the point state at this address or write to it: 0 = turn point off; non-0 = turn point on
Last valid address for this area: FFFF F080 0FD3			

Digital Point Write—Read/Write

PAC-R
EB
SB
UIO
EIO
SIO
E1
G4EB2

Applies to most digital modules. Do not use for SNAP high-density digital modules; see "SNAP High-Density Digital Write—Read/Write" on page 181.

Although this area is read/write, you would normally write to it. See "Digital Point Data (4-Channel Modules)" on page 60 for help in formatting data. For information on using counters, see page 31.

NOTE: To turn points on and off, you can also write to the point state address in the Digital Point Read area (page 149).

Only the first three points are shown in the table. Each successive point starts on an even 40 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F090 0000	4	B	Module 0, Point 0 (0,0): Turn on (Boolean) 0 = no effect; non-0 = turn on point
FFFF F090 0004	4	B	0,0: Turn off (Boolean) 0 = no effect; non-0 = turn off point
FFFF F090 0008	4	B	0,0: Activate counter (Boolean) 0 = no effect; non-0 = activate
FFFF F090 000C	4	B	0,0: Deactivate counter (Boolean) 0 = no effect; non-0 = deactivate
FFFF F090 0040	4	B	0,1: Turn on (Boolean) 0 = no effect; non-0 = turn on point
FFFF F090 0044	4	B	0,1: Turn off (Boolean) 0 = no effect; non-0 = turn off point
FFFF F090 0048	4	B	0,1: Activate counter (Boolean) 0 = no effect; non-0 = activate
FFFF F090 004C	4	B	0,1: Deactivate counter (Boolean) 0 = no effect; non-0 = deactivate
FFFF F090 0080	4	B	0,2: Turn on (Boolean) 0 = no effect; non-0 = turn on point
FFFF F090 0084	4	B	0,2: Turn off (Boolean) 0 = no effect; non-0 = turn off point
FFFF F090 0088	4	B	0,2: Activate counter (Boolean) 0 = no effect; non-0 = activate
FFFF F090 008C	4	B	0,2: Deactivate counter (Boolean) 0 = no effect; non-0 = deactivate
(Additional points follow in order on even 40 hex boundaries.)			
FFFF F090 0FC0	4	B	15,3: Turn on (Boolean) 0 = no effect; non-0 = turn on point
Last valid address for this area: FFFF F090 0FCF			

(Old) Analog Point Read—Read Only

UIO
EIO
SIO
E2

Use this section only for E2 I/O units and for other I/O units with firmware versions 7.1 or lower. For units with firmware version 8.0 or higher, see page 116.

See "IEEE Float Data" on page 60 for help in interpreting data.

Only the first three points are shown in the table. Each successive point starts on an even 40 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0A0 0000	4	F	Module 0, Point 0 (0,0): Analog input/output data (float—Engineering Units)
FFFF F0A0 0004	4	F	0,0: Analog input/output data (float—counts)
FFFF F0A0 0008	4	F	0,0: Analog Min Value (float—Engineering Units)
FFFF F0A0 000C	4	F	0,0: Analog Max Value (float—Engineering Units)
FFFF F0A0 0010	4	--	Reserved
FFFF F0A0 0040	4	F	0,1: Analog input/output data (float—Engineering Units)
FFFF F0A0 0044	4	F	0,1: Analog input/output data (float—counts)
FFFF F0A0 0048	4	F	0,1: Analog Min Value (float—Engineering Units)
FFFF F0A0 004C	4	F	0,1: Analog Max Value (float—Engineering Units)
FFFF F0A0 0080	4	F	0,2: Analog input/output data (float—Engineering Units)
FFFF F0A0 0084	4	F	0,2: Analog input/output data (float—counts)
FFFF F0A0 0088	4	F	0,2: Analog Min Value (float—Engineering Units)
FFFF F0A0 008C	4	F	0,2: Analog Max Value (float—Engineering Units)
(Additional points follow in order on even 40 hex boundaries.)			
FFFF F0A0 0FC0	4	F	15,3: Analog input/output data (float—Engineering Units)
Last valid address for this area: FFFF F0A0 0FCF			

(Old) Analog Point Write—Read/Write

UIO
EIO
SIO
E2

Use this section only for E2 I/O units and for other I/O units with firmware versions 7.1 or lower. For units with firmware version 8.0 or higher, see page 117.

Although this area is read/write, you would normally write to it. See "IEEE Float Data" on page 60 for help in formatting data.

Only the first three points are shown in the table. Each successive point starts on an even 40 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0B0 0000	4	F	Module 0, Point 0 (0,0): Output analog data (float—Engineering Units)
FFFF F0B0 0004	4	F	0,0: Output analog data (float—counts)
FFFF F0B0 0008	4	F	0,0: TPO resolution
FFFF F0B0 000C	4	F	0,0: TPO period (float—units of time in seconds. Valid range: 0.25 to 64.0 seconds, in 0.25 steps.) (Not for E2s.)
FFFF F0B0 0010	4	--	0,0: Reserved
FFFF F0B0 0014	4	UI	0,0: Load cell fast settle level—For SNAP-AILC modules only. (Not for E2s.) Use with load cell filter weight (next address) to get filtered readings faster. Valid values: 0–32,767. 0 = disabled;
FFFF F0B0 0018	4	UI	0,0: Load cell filter weight—For SNAP-AILC modules only. (Not for E2s.) Valid values: 0–255; default = 128. A larger value increases filtering. Use with F0B00014 to get the filtered reading faster. Note that 0, 1, or 255 value disables fast settle level (F0B00014). The second channel on the module is the filtered reading of the first channel.
FFFF F0B0 0040	4	F	0,1: Output analog data (float—Engineering Units)
FFFF F0B0 0044	4	F	0,1: Output analog data (float—counts)
FFFF F0B0 004C	4	F	0,1: TPO period (float—units of time in seconds. Valid range: 0.25 to 64.0 seconds, in 0.25 steps.) (Not for E2s.)
FFFF F0B0 0080	4	F	0,2: Output analog data (float—Engineering Units)
FFFF F0B0 0084	4	F	0,2: Output analog data (float—counts)
FFFF F0B0 008C	4	F	0,2: TPO period (float—units of time in seconds. Valid range: 0.25 to 64.0 seconds, in 0.25 steps.) (Not for E2s.)
(Additional points follow in order on even 40 hex boundaries.)			
FFFF F0B0 0FC0	4	F	15,3: Output analog data (float—Engineering Units)
Last valid address for this area: FFFF F0B0 0FCF			

(Old) Analog and Digital Point Configuration Information—Read/Write

E1
E2
UIO
EIO
SIO

Use this section only for E1 and E2 I/O units and for other I/O units with firmware versions 7.1 or lower. For units with firmware version 8.0 or higher, see page 114.

See page 22 for configuration information. This area does not apply to SNAP high-density digital modules, which do not require configuration.

Only the first two points are shown in the table. Each successive point starts on an even 40 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0C0 0000	4	UI	Module 0, Point 0 (0,0): Read only. Module Type (unsigned integer). For SNAP I/O units, module type is the value reported by an analog, serial, or high-density digital module on a SNAP rack. For analog modules on E2 units, the equivalent SNAP module (derived from point configuration) is returned. Zero is returned for digital modules, no module, or points that don't exist (for example, the upper two points on a two-channel analog module). See table on page 156 for values.
FFFF F0C0 0004	4	UI	0,0: Point Type (unsigned integer). • Use 0x0000 0100 for digital inputs. • Use 0x0000 0180 for digital outputs. • For analog types, see tables starting on page 23.
FFFF F0C0 0008	4	UI	0,0: Point Feature (unsigned integer) Digital feature values: • 0x0000 0001 for counter input (configures and starts the counter) • 0x0000 0002 for on-time totalizer input • 0x0000 0003 for period measurement (continuous) • 0x0000 0004 for simple quadrature counter input • 0x0000 0005 for frequency measurement (continuous) • 0x0000 0009 for on-pulse duration measurement (one-time) • 0x0000 000A for off-pulse duration measurement (one-time) • 0x0000 000B for period measurement (one-time) • 0x0000 000C for frequency measurement (one-time) • 0x0000 0012 for off-time totalizer input • 0x0000 0041 quadrature counter input with index. For quadrature counter information, see page 32. Analog feature values: none at present To disable point features, use 0x0000 0000
FFFF F0C0 000C	4	F	0,0: Point offsets (float)
FFFF F0C0 0010	4	F	0,0: Point gains (float)
FFFF F0C0 0014	4	F	0,0: Point high scaling factors (float)
FFFF F0C0 0018	4	F	0,0: Point low scaling factors (float)
FFFF F0C0 001C	4	--	0,0: Reserved
FFFF F0C0 0020	4	F	0,0: Average filter weight (float) (Does not apply to E2.)
FFFF F0C0 0024	4	F	0,0: Watchdog value. EU float for analog and digital. 0 = off, non-0 = on.
FFFF F0C0 0028	4	B	0,0: Enable watchdog (Boolean) 0 = disable; non-0 = enable

Starting Address	Length (hex)	Type	Description
FFFF F0C0 002C	4	--	Reserved
FFFF F0C0 0030	10	S-ZT	Point name
FFFF F0C0 0040	4	UI	0,1: Read only. Module Type (unsigned integer). For SNAP I/O units, module type is the value reported by an analog or special-purpose module on a SNAP rack. For analog modules on E2 units, the equivalent SNAP module (derived from point configuration) is returned. Zero is returned for digital modules, no module, or points that don't exist (for example, the upper two points on a two-channel analog module). See table on page 156 for values.
FFFF F0C0 0044	4	UI	0,1: Point Type (unsigned integer). • Use 0x0000 0100 for digital inputs. • Use 0x0000 0180 for digital outputs. • For analog types, see table on page 23.
FFFF F0C0 0048	4	UI	0,1: Point Feature (unsigned integer) Digital feature values: • 0x0000 0001 for counter input (configures and starts the counter) • 0x0000 0002 for on-time totalizer input • 0x0000 0003 for period measurement (continuous) • 0x0000 0004 for simple quadrature counter input • 0x0000 0005 for frequency measurement (continuous) • 0x0000 0009 for on-pulse duration measurement (one-time) • 0x0000 000A for off-pulse duration measurement (one-time) • 0x0000 000B for period measurement (one-time) • 0x0000 000C for frequency measurement (one-time) • 0x0000 0012 for off-time totalizer input • 0x0000 0041 quadrature counter input with index. For quadrature counter information, see page 32. Analog feature values: none at present To disable point features, use 0x0000 0000
FFFF F0C0 004C	4	F	0,1: Point offsets (float)
FFFF F0C0 0050	4	F	0,1: Point gains (float)
FFFF F0C0 0054	4	F	0,1: Point high scaling factors (float)
FFFF F0C0 0058	4	F	0,1: Point low scaling factors (float)
FFFF F0C0 005C	4	--	0,1: Reserved
FFFF F0C0 0060	4	F	0,1: Average filter weight (float)
FFFF F0C0 0064	4	F	0,1: Watchdog value. EU float for analog and digital. 0 = off, non-0 = on.
FFFF F0C0 0068	4	B	0,1: Enable watchdog (Boolean) 0 = disable; non-0 = enable
FFFF F0C0 006C	4	--	Reserved
FFFF F0C0 0070	10	S-ZT	Point name
(Additional points follow in order on even 40 hex boundaries.)			
FFFF F0C0 0FC0	4	UI	15,15: Read only. Module type.
FFFF F0C0 1000	4	F	0,0: Analog output lower clamp (float).
FFFF F0C0 1004	4	F	0,0: Analog output upper clamp (float).

Starting Address	Length (hex)	Type	Description
FFFF F0C0 1008	4	F	0,1: Analog output lower clamp (float).
FFFF F0C0 100C	4	F	0,1: Analog output upper clamp (float).
(Additional points follow in order)			
FFFF F0C0 11FC	4	F	15,3: Analog output lower clamp (float).
Last valid address for this area: FFFF F0C0 11FF			

PAC-R
EB
SB
UIO
EIO
SIO
E1
E2
G4EB2

SNAP I/O Module Types

I/O module types are reported by SNAP I/O units but not by E1s, E2s, or G4EB2s.

However, in the module type memory map address (page 114, address FFFF F010 0000 for module 0, point 0), E1, E2, and G4EB2 I/O units show the equivalent SNAP module, derived from the configured point type.

The following table lists values for SNAP I/O module types.

This hex value	Indicates this module type
00	4-ch digital or empty*
04	SNAP-AICTD
09	SNAP-AITM-2
0A	SNAP-AIPM
0B	SNAP-AILC
0C	SNAP-AILC-2
0E	SNAP-AIRTD-10
0F	SNAP-AIRTD-1K
10	SNAP-AIRTD
12	SNAP-AIV
20	SNAP-AITM-i
21	SNAP-AITM2-i
22	SNAP-AIMA-i
23	SNAP-AIV-i
24	SNAP-AIV2-i
25	SNAP-pH/ORP
26	SNAP-AIMA-iSRC SNAP-AIMA-iSRC-FM
27	SNAP-AIMA2-i
28	SNAP-AIARMS-i SNAP-AIARMS-i-FM
29	SNAP-AIVRMS-i SNAP-AIVRMS-i-FM
32	SNAP-AITM-4i
40	SNAP-AIMA-4

This hex value	Indicates this module type
41	SNAP-AIV-4
42	SNAP-AICTD-4
43	SNAP-AIR40K-4
44	SNAP-AIMV-4
45	SNAP-AIMV2-4
48	SNAP-AIPM-3V
49	SNAP-AIPM-3
4A	SNAP-AIMA-8
4B	SNAP-AIV-8
4C	SNAP-AICTD-8
4D	SNAP-AIMA-32 SNAP-AIMA-32-FM
4E	SNAP-AIV-32 SNAP-AIV-32-FM
4F	SNAP-AITM-8 SNAP-AITM-8-FM
64	SNAP-AIMA
66	SNAP-AITM
69	SNAP-AIRATE
70	SNAP-AIVRMS
71	SNAP-AIARMS
A3	SNAP-AOA23
A5	SNAP-AOV25
A7	SNAP-AOV27
A8	SNAP-AOA28

This hex value	Indicates this module type
A9	SNAP-AOD29
B3	SNAP-AOA23-iSRC SNAP-AOA23-iSRC-FM
D0	SNAP-PID-V
E0	SNAP-IDC-32 SNAP-IDC-32-FM
E1	SNAP-ODC-32-SRC SNAP-ODC-32-SRC-FM
E2	SNAP-ODC-32-SNK SNAP-ODC-32-SNK-FM
E3	SNAP-IAC-A-16
E4	SNAP-IAC-16
E5	SNAP-IDC-16
E6	SNAP-IDC-32N
E7	SNAP-IAC-K-16
E8	SNAP-IDC-HT-16
EA	SNAP-IDC-32DN
EB	SNAP-IDC-32D
F0	SNAP-SCM-232
F1	SNAP-SCM-485 SNAP-SCM-485-422
F6	SNAP-SCM-PROFI
F8	SNAP-SCM-MCH16
F9	SNAP-SCM-W2
FA	SNAP-SCM-SSI
FB	SNAP-SCM-ST2
FC	SNAP-SCM-CAN2B

* Digital modules with more than four points are individually listed in this table.

(Old) Digital Events and Reactions—Read/Write

PAC-R
EB
UIO
EIO
G4EB2

Use this section only for I/O units with firmware versions 8.0 or lower. For units with firmware version 8.1 or higher, see page 158.

(Does not apply to SNAP Simple I/O. Does not apply to SNAP high-density digital modules.) See page 44 for information on configuring digital events and reactions.

IMPORTANT: To reduce scanning time, the I/O unit stops scanning digital events when it reaches an unused event. Make sure you use event numbers in order, starting with the lowest.

Only the first two digital event/reactions are shown in the table. Other event/reactions follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0D0 0000	8	M	Digital event 0: Points on? (mask)
FFFF F0D0 0008	8	M	Digital event 0: Points off? (mask)
FFFF F0D0 0010	8	M	Digital reaction 0: Turn points on (mask)
FFFF F0D0 0018	8	M	Digital reaction 0: Turn points off (mask)
FFFF F0D0 0020	8	M	Digital event 0: Scratch Pad bits on? (mask)
FFFF F0D0 0028	8	M	Digital event 0: Scratch Pad bits off? (mask)
FFFF F0D0 0030	8	M	Digital reaction 0: Turn Scratch Pad bits on (mask)
FFFF F0D0 0038	8	M	Digital reaction 0: Turn Scratch Pad bits off (mask)
FFFF F0D0 0040	8	M	Digital event 1: Points on? (mask)
FFFF F0D0 0048	8	M	Digital event 1: Points off? (mask)
FFFF F0D0 0050	8	M	Digital reaction 1: Turn points on (mask)
FFFF F0D0 0058	8	M	Digital reaction 1: Turn points off (mask)
FFFF F0D0 0060	8	M	Digital event 1: Scratch Pad bits on? (mask)
FFFF F0D0 0068	8	M	Digital event 1: Scratch Pad bits off? (mask)
FFFF F0D0 0060	8	M	Digital reaction 1: Turn Scratch Pad bits on (mask)
FFFF F0D0 0068	8	M	Digital reaction 1: Turn Scratch Pad bits off (mask)
(Other digital event/reactions follow in order on even 40 hex boundaries.)			
Last valid address for this area: FFFF F0D0 1FFF			

Digital Events - Expanded (formerly Timers)—Read/Write

PAC-R EB SB G4EB2

For I/O units with firmware versions 8.1 and higher, use this section for all digital event configuration. See addresses below.

For I/O units with firmware versions 8.0 and lower, this section provides Timer configuration only, with limited options. See addresses on page 161.

For event capabilities of devices and firmware versions, see the charts starting on page 38.

Addresses for Firmware 8.1 and Higher

IMPORTANT: To reduce scanning time, the I/O unit stops scanning digital events when it reaches an unused event. Make sure you use event numbers in order, starting with the lowest.

Only the first two digital events are shown in the table. Others follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0D4 0000	8	M	Event 0: Trigger #1 ON mask that triggers the event (details of event trigger are determined by the mask in F0D4 0044)
FFFF F0D4 0008	8	M	Event 0: Trigger #1 OFF mask that triggers the event
FFFF F0D4 0010	8	M	Event 0: Trigger #2 ON mask that triggers the event
FFFF F0D4 0018	8	M	Event 0: Trigger #2 OFF mask that triggers the event
FFFF F0D4 0020	8	M	Event 0: Trigger #1 ON mask to be set as a reaction to the event
FFFF F0D4 0028	8	M	Event 0: Trigger #1 OFF mask to be set as a reaction to the event
FFFF F0D4 0030	8	M	Event 0: Trigger #2 ON mask to be set as a reaction to the event
FFFF F0D4 0038	8	M	Event 0: Trigger #2 OFF mask to be set as a reaction to the event
FFFF F0D4 0040	4	UI	Event 0: Time between event and reaction, in milliseconds

<table>
<tr><th>Starting Address</th><th>Length (hex)</th><th>Type</th><th>Description</th></tr>
<tr><td>FFFF F0D4 0044</td><td>4</td><td>M</td><td>Event 0: Event detail mask, specifying triggers and reactions. For examples, see “Event Detail Mask” on page 45.
<table>
<tr><th>Bit</th><th>Purpose</th></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>Trigger 1—Latches = 1; digital point state = 0</td></tr>
<tr><td>2</td><td>If bit 1 is set (latches)—On-latches = 1; off-latches = 0</td></tr>
<tr><td>3</td><td>Reaction 1—Clear on- and off-latches = 1; other reaction = 0</td></tr>
<tr><td>4</td><td>Trigger 1—Use HDD module specified in F0D4 0050 = 1; use 4-ch digital module = 0</td></tr>
<tr><td>5</td><td>Reaction 1—Use HDD module specified in F0D4 0054 = 1; use 4-ch digital module = 0</td></tr>
<tr><td>6</td><td>Trigger 1—Use Scratch Pad Integer 64 specified in F0D4 0058 = 1; use Scratch Pad Bit = 0</td></tr>
<tr><td>7</td><td>Reaction 1—Use Scratch Pad Integer 64 specified in F0D4 005C = 1; use Scratch Pad Bit = 0</td></tr>
<tr><td>8</td><td>Reaction—Send reaction only once = 1; send reaction continuously, as long as the event is still occurring = 0</td></tr>
<tr><td>9–14</td><td>Reserved</td></tr>
<tr><td>15</td><td>Trigger 1—Scratch Pad bits = 1; other trigger = 0</td></tr>
<tr><td>16</td><td>Trigger 2—Digital point state or latch = 1; other trigger = 0</td></tr>
<tr><td>17</td><td>Trigger 2—If bit 16 is set, digital point latch = 1; digital point state = 0</td></tr>
<tr><td>18</td><td>Trigger 2—If bits 16 and 17 are set, on-latch = 1; off-latch = 0</td></tr>
<tr><td>19</td><td>Reaction 2—Digital point state = 1; other reaction = 0</td></tr>
<tr><td>20</td><td>Reaction 1—Scratch Pad bits = 1; other reaction = 0</td></tr>
<tr><td>21</td><td>Reaction 2—Clear digital point latches = 1; other reaction = 0</td></tr>
<tr><td>22–31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>FFFF F0D4 0048</td><td>4</td><td>UI</td><td>Event 0: Time remaining until the reaction, in milliseconds</td></tr>
<tr><td>FFFF F0D4 004C</td><td>4</td><td>UI</td><td>Event 0: Event state:
0 = event is enabled but not occurring now
1 = event is occurring
2 = event has occurred and reaction has not been sent (in delay period)
3 = event has occurred and reaction has been sent (applies only if bit 8 in F0D4 0044 is set to send the reaction just once)
4 = event is disabled</td></tr>
<tr><td>FFFF F0D4 0050</td><td>4</td><td>UI</td><td>Event 0: HDD module number to use for digital event trigger #1</td></tr>
<tr><td>FFFF F0D4 0054</td><td>4</td><td>UI</td><td>Event 0: HDD module number to use for digital reaction #1</td></tr>
<tr><td>FFFF F0D4 0058</td><td>4</td><td>UI</td><td>Event 0: Scratch Pad Integer 64 index to use for digital event trigger #2</td></tr>
<tr><td>FFFF F0D4 005C</td><td>4</td><td>UI</td><td>Event 0: Scratch Pad Integer 64 index to use for digital event reaction #2</td></tr>
<tr><td>FFFF F0D4 0060</td><td>20</td><td>--</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Starting Address</th><th>Length (hex)</th><th>Type</th><th>Description</th></tr>
<tr><td>FFFF F0D4 0080</td><td>8</td><td>M</td><td>Event 1: Trigger #1 ON mask that triggers the event (details of event trigger are determined by the mask in F0D4 0044)</td></tr>
<tr><td>FFFF F0D4 0088</td><td>8</td><td>M</td><td>Event 1: Trigger #1 OFF mask that triggers the event</td></tr>
<tr><td>FFFF F0D4 0090</td><td>8</td><td>M</td><td>Event 1: Trigger #2 ON mask that triggers the event</td></tr>
<tr><td>FFFF F0D4 0098</td><td>8</td><td>M</td><td>Event 1: Trigger #2 OFF mask that triggers the event</td></tr>
<tr><td>FFFF F0D4 00A0</td><td>8</td><td>M</td><td>Event 1: Trigger #1 ON mask to be set as a reaction to the event</td></tr>
<tr><td>FFFF F0D4 00A8</td><td>8</td><td>M</td><td>Event 1: Trigger #1 OFF mask to be set as a reaction to the event</td></tr>
<tr><td>FFFF F0D4 00B0</td><td>8</td><td>M</td><td>Event 1: Trigger #2 ON mask to be set as a reaction to the event</td></tr>
<tr><td>FFFF F0D4 00B8</td><td>8</td><td>M</td><td>Event 1: Trigger #2 OFF mask to be set as a reaction to the event</td></tr>
<tr><td>FFFF F0D4 00C0</td><td>4</td><td>UI</td><td>Event 1: Time between event and reaction, in milliseconds</td></tr>
<tr><td>FFFF F0D4 00C4</td><td>4</td><td>M</td><td>Event 1: Event detail mask, specifying triggers and reactions. For examples, see “Event Detail Mask” on page 45.
<table>
<tr><th>Bit</th><th>Purpose</th></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>Trigger 1—Latches = 1; digital point state = 0</td></tr>
<tr><td>2</td><td>If bit 1 is set (latches)—On-latches = 1; off-latches = 0</td></tr>
<tr><td>3</td><td>Reaction 1—Clear on- and off-latches = 1; other reaction = 0</td></tr>
<tr><td>4</td><td>Trigger 1—Use HDD module specified in F0D4 0050 = 1; use 4-ch digital module = 0</td></tr>
<tr><td>5</td><td>Reaction 1—Use HDD module specified in F0D4 0054 = 1; use 4-ch digital module = 0</td></tr>
<tr><td>6</td><td>Trigger 1—Use Scratch Pad Integer 64 specified in F0D4 0058 = 1; use Scratch Pad Bit = 0</td></tr>
<tr><td>7</td><td>Reaction 1—Use Scratch Pad Integer 64 specified in F0D4 005C = 1; use Scratch Pad Bit = 0</td></tr>
<tr><td>8</td><td>Reaction—Send reaction only once = 1; send reaction continuously, as long as the event is still occurring = 0</td></tr>
<tr><td>9–14</td><td>Reserved</td></tr>
<tr><td>15</td><td>Trigger 1—Scratch Pad bits = 1; other trigger = 0</td></tr>
<tr><td>16</td><td>Trigger 2—Digital point state or latch = 1; other trigger = 0</td></tr>
<tr><td>17</td><td>Trigger 2—If bit 16 is set, digital point latch = 1; digital point state = 0</td></tr>
<tr><td>18</td><td>Trigger 2—If bits 16 and 17 are set, on-latch = 1; off-latch = 0</td></tr>
<tr><td>19</td><td>Reaction 2—Digital point state = 1; other reaction = 0</td></tr>
<tr><td>20</td><td>Reaction 1—Scratch Pad bits = 1; other reaction = 0</td></tr>
<tr><td>21</td><td>Reaction 2—Clear digital point latches = 1; other reaction = 0</td></tr>
<tr><td>22–31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>FFFF F0D4 00C8</td><td>4</td><td>UI</td><td>Event 1: Time remaining until the reaction, in milliseconds</td></tr>
</table>

Starting Address	Length (hex)	Type	Description
FFFF F0D4 00CC	4	UI	Event 1: Event state: 0 = event is enabled but not occurring now 1 = event is occurring 2 = event has occurred and reaction has not been sent (in delay period) 3 = event has occurred and reaction has been sent (applies only if bit 8 in F0D4 00C4 is set to send the reaction just once) 4 = event is disabled
FFFF F0D4 00D0	4	UI	Event 1: HDD module number to use for digital event trigger #1
FFFF F0D4 0054	4	UI	Event 1: HDD module number to use for digital reaction #1
FFFF F0D4 0058	4	UI	Event 1: Scratch Pad Integer 64 index to use for digital event trigger #2
FFFF F0D4 005C	4	UI	Event 1: Scratch Pad Integer 64 index to use for digital event reaction #2
FFFF F0D4 0060	20	--	Reserved
(Other digital events follow in order on even 80 hex boundaries.)			
Last valid address for this area: FFFF F0D4 FFFF			

Addresses for Firmware 8.0 and Lower

PAC-REB UIO EIO

For I/O units with firmware versions 8.0 and lower, this section provides only Timer (delay) configuration for digital events. See addresses below.

For I/O units with firmware versions 8.1 and higher, use this section for all digital event configuration. See addresses on page 158.

For event capabilities of devices and firmware versions, see the charts starting on page 38.

IMPORTANT: To reduce scanning time, the I/O unit stops scanning digital events when it reaches an unused event. Make sure you use event numbers in order, starting with the lowest.

Only the first two timer events are shown in the table. Others follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0D4 0000	8	M	Timer 0: Digital ON mask that starts the timer
FFFF F0D4 0008	8	M	Timer 0: Digital OFF mask that starts the timer
FFFF F0D4 0010	8	M	Timer 0: Scratch Pad ON mask that starts the timer
FFFF F0D4 0018	8	M	Timer 0: Scratch Pad OFF mask that starts the timer
FFFF F0D4 0020	8	M	Timer 0: Digital ON mask to be set when the timer expires
FFFF F0D4 0028	8	M	Timer 0: Digital OFF mask to be set when the timer expires
FFFF F0D4 0030	8	M	Timer 0: Scratch Pad ON mask to be set when the timer expires
FFFF F0D4 0038	8	M	Timer 0: Scratch Pad OFF mask to be set when the timer expires
FFFF F0D4 0040	4	UI	Timer 0: Length of the timer delay, in milliseconds
FFFF F0D4 0044	4	--	Reserved

Starting Address	Length (hex)	Type	Description
FFFF F0D4 0048	4	UI	Timer 0: Time remaining until the timer expires, in milliseconds
FFFF F0D4 004C	4	UI	Timer 0: Timer state. 0 = Timer not set; 1 = Timer set; 2 = Timer is ticking
FFFF F0D4 0050	30	--	Pad for alignment
FFFF F0D4 0080	8	M	Timer 1: Digital ON mask that starts the timer
FFFF F0D4 0088	8	M	Timer 1: Digital OFF mask that starts the timer
FFFF F0D4 0090	8	M	Timer 1: Scratch Pad ON mask that starts the timer
FFFF F0D4 0098	8	M	Timer 1: Scratch Pad OFF mask that starts the timer
FFFF F0D4 00A0	8	M	Timer 1: Digital ON mask to be set when the timer expires
FFFF F0D4 00A8	8	M	Timer 1: Digital OFF mask to be set when the timer expires
FFFF F0D4 00B0	8	M	Timer 1: Scratch Pad ON mask to be set when the timer expires
FFFF F0D4 00B8	8	M	Timer 1: Scratch Pad OFF mask to be set when the timer expires
FFFF F0D4 00C0	8	UI	Timer 1: Length of the timer delay, in milliseconds
FFFF F0D4 00C4	4	--	Reserved
FFFF F0D4 00C8	4	UI	Timer 1: Time remaining until the timer expires, in milliseconds
FFFF F0D4 00CC	8	UI	Timer 1: Timer state. 0 = Timer not set; 1 = Timer set; 2 = Timer is ticking
FFFF F0D4 00D0	30	--	Pad for alignment
(Other timers follow in order on even 80 hex boundaries.)			
Last valid address for this area: FFFF F0D4 FFFF			

Scratch Pad—Read/Write

PAC-R
PAC-S
EB
SB
UIO
EIO
LCE
G4EB2

For EIO units, only Scratch Pad bits apply. For other devices, the Scratch Pad area is used to share data among devices on the network. Since each Scratch Pad read or write operation is atomic, reads and writes will not interfere with each other.

Scratch Pad 64-Bit Integers apply only to devices with firmware version 8.1 and higher. For more information on using the Scratch Pad area, see page 37.

NOTE: Scratch Pad strings can be null terminated or can be binary strings with embedded nulls. Length is automatically calculated, but can be written to force a specific length. If the length and the string are written together, the length written is used.

Starting Address	Length (hex)	Type	Description
FFFF F0D8 0000	8	M	Current state of Scratch Pad bits (mask). 1 = bit on; 0 = bit off.
FFFF F0D8 0400	8	M	Set Scratch Pad On mask. 1 = set bit on; 0 = no change.
FFFF F0D8 0408	8	M	Set Scratch Pad Off mask 1 = set bit off; 0 = no change.
FFFF F0D8 1000	4	I	Scratch Pad 32-bit Integer element 0.
FFFF F0D8 1004	4	I	Scratch Pad 32-bit Integer element 1.
FFFF F0D8 1008	4	I	Scratch Pad 32-bit Integer element 2.
(Additional 32-bit Integer elements follow in order on even 4 hex boundaries. Total at this address: 1024. Additional 9216 starting at address FFFF F0DA 0000)			
FFFF F0D8 1FFC	4	I	Scratch Pad 32-bit Integer element 1023.
FFFF F0D8 2000	4	F	Scratch Pad float element 0.
FFFF F0D8 2004	4	F	Scratch Pad float element 1.
FFFF F0D8 2008	4	F	Scratch Pad float element 2.
(Additional float elements follow in order on even 4 hex boundaries. Total at this address: 1024. Additional 9216 starting at address FFFF F0DC 0000)			
FFFF F0D8 2FFC	4	F	Scratch Pad float element 1023.
FFFF F0D8 3000	2	UI	Scratch Pad string element 0: Length (integer). If length is not written here, it is automatically calculated by counting to the first null.
FFFF F0D8 3002	80	S-PL	Scratch Pad string element 0: String
FFFF F0D8 3082	2	UI	Scratch Pad string element 1: Length (integer). If length is not written here, it is automatically calculated by counting to the first null.
FFFF F0D8 3084	80	S-PL	Scratch Pad string element 1: String
FFFF F0D8 3104	2	UI	Scratch Pad string element 2: Length (integer). If length is not written here, it is automatically calculated by counting to the first null.
FFFF F0D8 3106	80	S-PL	Scratch Pad string element 2: String
(Additional string elements follow in order on even 82 hex boundaries. Total: 64.)			
FFFF F0D8 4FFE	2	UI	Scratch Pad string element 63: Length (integer). If length is not written here, it is automatically calculated by counting to the first null.

Starting Address	Length (hex)	Type	Description
FFFF F0D8 5000	80	S-PL	Scratch Pad string element 63: String
Last valid address in this section: FFFF F0D8 507F			
FFFF F0DA 0000	4	I	Scratch Pad 32-bit Integer element 1024
FFFF F0DA 0004	4	I	Scratch Pad 32-bit Integer element 1025
FFFF F0DA 0008	4	I	Scratch Pad 32-bit Integer element 1026
(Additional 32-bit Integer elements follow in order on even 4 hex boundaries. Total at this address: 9216. Elements 0–1023 start at address FFFF F0D8 1000)			
FFFF F0DA 8FFC	4	I	Scratch Pad 32-bit Integer element 10,239
Last valid address in this section: FFFF F0DA 1FFFF			
FFFF F0DC 0000	4	F	Scratch Pad float element 1024
FFFF F0DC 0004	4	F	Scratch Pad float element 1025
FFFF F0DC 0008	4	F	Scratch Pad float element 1026
(Additional float elements follow in order on even 4 hex boundaries. Total at this address: 1024. Elements 0–1023 start at address FFFF F0D8 2000)			
FFFF F0DC 8FFC	4	F	Scratch Pad float element 10,239
Last valid address in this section: FFFF F0DC 8FFF			
FFFF F0DE 0000	8	M	Scratch Pad 64-bit Integer element 0
FFFF F0DE 0008	8	M	Scratch Pad 64-bit Integer element 1
FFFF F0DE 0010	8	M	Scratch Pad 64-bit Integer element 2
(Additional 64-bit Integer elements follow in order on even 8 hex boundaries. Total at this address: 1024)			
FFFF F0DE 1FF8	8	M	Scratch Pad 64-bit Integer element 1023
Last valid address in this section: FFFF F0DE 1FFF			
Last valid address for this area: FFFF F0DC 81FF			

(Old) Analog Point Calculation and Set—Read Only

UIO
EIO
SIO
E2

Use this section only for E2 I/O units and for other I/O units with firmware versions 7.1 or lower. For units with firmware version 8.0 or higher, see page 115.

When you read data in this area, the data is returned and set.

CAUTION: Reading any portion of a quadlet (for example, the first byte of a four-byte quadlet) sets the offset or gain but returns only the one-byte value requested, which is incomplete.

Offset assumes that zero-scale counts are on the input. Gain assumes that full-scale counts are on the input. Offset and gain are typically used for calibration, and offset should be done before gain.

Only the first four points are shown. Each successive point starts on an even 4 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0E0 0000	4	F	Module 0, Point 0 (0,0): Offset in Engineering Units (float)
FFFF F0E0 0004	4	F	0,1: Offset in Engineering Units (float)
FFFF F0E0 0008	4	F	0,2: Offset in Engineering Units (float)
FFFF F0E0 000C	4	F	Point 3: Offset in Engineering Units (float)
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0E0 00FC	4	F	15,15 Offset in Engineering Units (float)
FFFF F0E0 0100	4	F	0,0: Gain (float)
FFFF F0E0 0104	4	F	0,1: Gain (float)
FFFF F0E0 0108	4	F	0,2: Gain (float)
FFFF F0E0 010C	4	F	Point 3: Gain (float)
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0E0 01FC	4	F	15,15 Gain (float)
Last valid address for this area: FFFF F0E0 01FF			

(Old) Digital Read and Clear—Read Only

UIO
EIO
SIO
E1

Use this section only for E1 I/O units and for other I/O units with firmware versions 7.1 or lower. For units with firmware version 8.0 or higher, see page 118. Do not use this section for SNAP high-density digital points; see "SNAP High-Density Digital Read and Clear—Read/Write" on page 180.

When you read data from this area, the data is returned and then cleared.

Only the first two points are shown. Each successive point starts on an even 4 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0F0 0000	4	UI	Module 0, Point 0 (0,0): Counts (unsigned integer)
FFFF F0F0 0004	4	UI	0,1: Counts (unsigned integer)
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0F0 00FC	4	UI	15,3: Counts (unsigned integer)
FFFF F0F0 0100	4	B	0,0: On-Latch (Boolean) 0 = off; non-0 = on
FFFF F0F0 0104	4	B	0,1: On-Latch (Boolean) 0 = off; non-0 = on
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0F0 01FC	4	B	15,3: On-Latch (Boolean) 0 = off; non-0 = on
FFFF F0F0 0200	4	B	0,0: Off-Latch (Boolean) 0 = off; non-0 = on
FFFF F0F0 0204	4	B	0,1: Off-Latch (Boolean) 0 = off; non-0 = on
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0F0 02FC	4	B	15,3: Off-Latch (Boolean) 0 = off; non-0 = on
Last valid address for this area: FFFF F0F0 02FF			

(Old) Analog Read and Clear/Restart—Read Only

UIO
EIO
SIO
E2

Use this section only for E2 I/O units and for other I/O units with firmware versions 7.1 or lower. For units with firmware version 8.0 or higher, see page 115.

When you read data from this area, the data is returned and then cleared or reset.

CAUTION: If you read or write less than a quadlet in this area of the memory map, the returned data will be useless and the information will be erased (cleared).

Only the first two points are shown. Each successive point starts on an even 4 hex boundary and follows the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F0F8 0000	4	F	Module 0, Point 0 (0,0): Min Data (float)
FFFF F0F8 0004	4	F	0,1: Min Data (float)
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0F8 00FC	4	F	15,3: Min Data (float)
FFFF F0F8 0100	4	F	0,0: Max Data (float)
FFFF F0F8 0104	4	F	0,1: Max Data (float)
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F0F8 01FC	4	F	15,3: Max Data (float)
Last valid address for this area: FFFF F0F8 01FF			

Streaming—Read Only

PAC-R
EB
UIO
EIO
SIO
G4EB2

This area can be used in two ways:

- If you want this information in a compact form, you can read this area all at once instead of reading several other sections. (Like the two packed areas, "Analog EU or Digital Counter Packed Data—Read" on page 169 and "Digital Packed Data—Read/Write" on page 170, this area efficiently reads commonly needed data using few transactions.)
- You can stream data from the I/O unit to PCs or other host devices. Streaming sends data at stated intervals automatically and does not require a response. For more information, see "Streaming Data" on page 51.

Starting Address	Length (hex)	Type	Description
FFFF F100 0000	100	F	Analog Engineering Units data for 64 points (float)
FFFF F100 0100	100	UI	Digital point feature data (unsigned integer). Feature data for each point depends upon the point configuration.
FFFF F100 0200	8	M	State of 4-channel digital points (mask). Does not include high-density digital points.
FFFF F100 0208	8	M	State of digital on-latches (mask). Does not include high-density digital points.
FFFF F100 0210	8	M	State of digital off-latches (mask). Does not include high-density digital points.
FFFF F100 0218	8	M	Active counters (mask)
Last valid address for this area: FFFF F100 021F			

Analog EU or Digital Counter Packed Data—Read

PAC-R
EB
SB
G4EB2

This area works with I/O units that have firmware version 8.0 and higher. This area has space for 512 points, the maximum currently available on one I/O unit. Addresses contain either analog Engineering Units (EU) or digital counters, depending on the module installed at that location. In some ways this area is similar to the Digital Packed Data area (see next section) and the Streaming area (see page 168); all of these sections are efficient ways to read needed data with the fewest number of transactions.

See "Formatting and Interpreting Data" on page 58 for help in understanding the data you read.

Only the first six points on the first module are shown in the table. Successive points and modules follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F100 1000	4	F/UI	Module 0, Point 0 (0,0): If analog point: value in EU; if digital point, digital counter value (counter must be configured as point feature).
FFFF F100 1004	4	F/UI	0,1: Analog point value or digital counter value
FFFF F100 1008	4	F/UI	0,2: Analog point value or digital counter value
FFFF F100 100C	4	F/UI	0,3: Analog point value or digital counter value
FFFF F100 1010	4	F/UI	0,4: Analog point value or digital counter value
FFFF F100 1014	4	F/UI	0,5: Analog point value or digital counter value
(Additional points follow in order on 4 hex boundaries.) (Additional modules follow in order on 80 hex boundaries.)			
Last valid address for this area: FFFF F100 17FF			

Digital Packed Data—Read/Write

PAC-R EB SB G4EB2

This area works with I/O units that have firmware version 8.0 and higher. This area has space for 512 points, the maximum currently available on one I/O unit.

Like the Analog EU or Digital Counter Packed Data area (see previous section) and the Streaming area (see page 168); this section is an efficient way to read or write needed data with the fewest number of transactions.

Data Format is as follows.

If the module has 32 points, the mask contains the state of all 32 points, as illustrated below:

At address:	F1001800								→	F1001803							
These bit numbers:	7	6	5	4	3	2	1	0	→	7	6	5	4	3	2	1	0
Show data for these points:	31	30	29	28	27	26	25	24	→	7	6	5	4	3	2	1	0

If the module has 4 points, the mask contains the following data:

At address:	F1001800	F1001801	F1001802								F1001803							
These bit numbers:	[ignore]	[ignore]	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Show this data:	[zeros—ignore]	[zeros—ignore]	Counter?	Off-latch	On-latch	State	Counter?	Off-latch	On-latch	State	Counter?	Off-latch	On-latch	State	Counter?	Off-latch	On-latch	State
On this point:	--	--	3	3	3	3	2	2	2	2	1	1	1	1	0	0	0	0

Only the first four modules are shown in the table. Successive modules follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F100 1800	4	M	Module 0. If high-density module, shows states of all points. If 4-channel module, shows state, whether on-latch and off-latch are set, and whether point is a configured as a counter.
FFFF F100 1804	4	M	Module 1 (same)
FFFF F100 1808	4	M	Module 2 (same)
FFFF F100 180C	4	M	Module 3 (same)
Last valid address for this area: FFFF F100 183F			

Alarm Event Settings—Read/Write

PAC-R
EB
SB
UIO
EIO

(Does not apply to SNAP Simple I/O) Use this area to configure deviation, high limit, and low limit alarm events and reactions. See page 48 for information on configuring alarms and their reactions.

Only the first two alarms are shown in the table. Successive alarms follow the same pattern and start on even 80 hex boundaries.

Starting Address	Length (hex)	Type	Description
FFFF F110 0000	4	B	Alarm 0: Deviation alarm: in alarm state? (Boolean)
FFFF F110 0004	4	B	Alarm 0: Enable deviation alarm (Boolean)
FFFF F110 0008	4	F	Alarm 0: Deviation alarm: previous deviation value (float)
FFFF F110 000C	4	F	Alarm 0: Deviation alarm, deviation amount (float)
FFFF F110 0010	8	M	Alarm 0: Deviation alarm reaction: set Scratch Pad bits on (mask)
FFFF F110 0018	8	M	Alarm 0: Deviation alarm reaction: set Scratch Pad bits off (mask)
FFFF F110 0020	4	B	Alarm 0: High alarm: in alarm state? (Boolean)
FFFF F110 0024	4	B	Alarm 0: Enable high alarm (Boolean)
FFFF F110 0028	4	F	Alarm 0: High alarm setpoint
FFFF F110 002C	4	F	Alarm 0: High alarm deadband
FFFF F110 0030	8	M	Alarm 0: High alarm reaction: set Scratch Pad bits on (mask)
FFFF F110 0038	8	M	Alarm 0: High alarm reaction: set Scratch Pad bits off (mask)
FFFF F110 0040	4	B	Alarm 0: Low alarm: in alarm state? (Boolean)
FFFF F110 0044	4	B	Alarm 0: Enable low alarm (Boolean)
FFFF F110 0048	4	F	Alarm 0: Low alarm setpoint
FFFF F110 004C	4	F	Alarm 0: Low alarm deadband
FFFF F110 0050	8	M	Alarm 0: Low alarm reaction: set Scratch Pad bits on (mask)
FFFF F110 0058	8	M	Alarm 0: Low alarm reaction: set Scratch Pad bits off (mask)
FFFF F110 0060	4	UI	Alarm 0: Address of value to check. Use to set multiple alarms on one point. See the *PAC Manager User's Guide* for information.
FFFF F110 0064	4	UI	Alarm 0: Is value a float? Yes = 1; no = 0. Yes is default.
FFFF F110 0068	18	--	Pad for alignment
FFFF F110 0080	4	B	Alarm 1: Deviation alarm: in alarm state? (Boolean)
FFFF F110 0084	4	B	Alarm 1: Enable deviation alarm (Boolean)
FFFF F110 0088	4	F	Alarm 1: Deviation alarm: previous deviation value (float)
FFFF F110 008C	4	F	Alarm 1: Deviation alarm, deviation amount (float)
FFFF F110 0090	8	M	Alarm 1: Deviation alarm reaction: set Scratch Pad bits on (mask)
FFFF F110 0098	8	M	Alarm 1: Deviation alarm reaction: set Scratch Pad bits off (mask)
FFFF F110 00A0	4	B	Alarm 1: High alarm: in alarm state? (Boolean)

Starting Address	Length (hex)	Type	Description
FFFF F110 00A4	4	B	Alarm 1: Enable high alarm (Boolean)
FFFF F110 00A8	4	F	Alarm 1: High alarm setpoint
FFFF F110 00AC	4	F	Alarm 1: High alarm deadband
FFFF F110 00B0	8	M	Alarm 1: High alarm reaction: set Scratch Pad bits on (mask)
FFFF F110 00B8	8	M	Alarm 1: High alarm reaction: set Scratch Pad bits off (mask)
FFFF F110 00C0	4	B	Alarm 1: Low alarm: in alarm state? (Boolean)
FFFF F110 00C4	4	B	Alarm 1: Enable low alarm (Boolean)
FFFF F110 00C8	4	F	Alarm 1: Low alarm setpoint
FFFF F110 00CC	4	F	Alarm 1: Low alarm deadband
FFFF F110 00D0	8	M	Alarm 1: Low alarm reaction: set Scratch Pad bits on (mask)
FFFF F110 00D8	8	M	Alarm 1: Low alarm reaction: set Scratch Pad bits off (mask)
FFFF F110 00E0	4	UI	Alarm 1: Address of value to check. Use to set multiple alarms on one point. See the *PAC Manager User's Guide* for information.
FFFF F110 00E4	4	UI	Alarm 1: Is value a float? Yes = 1; no = 0. Yes is default.
FFFF F110 00E8	18	--	Pad for alignment
(Additional alarms follow in order on even 80 hex boundaries.)			
FFFF F110 1F80	80	--	Alarm 63
Last valid address for this area: FFFF F110 1FFF			

Event Message Configuration—Read/Write

PAC-S PAC-R EB UIO EIO G4EB2

See page 49 in this guide and the *PAC Manager User's Guide* for information on configuring messages as reactions to events or alarms. Only the first two messages are shown. Other messages follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F120 0000	4	UI	Message 0: Current message state. Messages become active when Scratch Pad conditions are met. To acknowledge receipt of an active message, write a 2 to this address. The message will not be sent again until the conditions are met again. 0 = Inactive, 1 = Active, 2 = Acknowledged
FFFF F120 0004	8	M	Message 0: Scratch Pad bits on? (mask)
FFFF F120 000C	8	M	Message 0: Scratch Pad bits off? (mask)
FFFF F120 0014	4	B	Message 0: Enable streaming (Boolean). Be sure to fill in streaming configuration section. (See page 146.)
FFFF F120 0018	4	UI	Message 0: Stream period in seconds 0 = send once, 604800 = maximum

Starting Address	Length (hex)	Type	Description
FFFF F120 001C	4	B	Message 0: Enable email (Boolean). Be sure to fill in email configuration section. (See page 175.)
FFFF F120 0020	4	UI	Message 0: Email period in seconds 0 = send once; 604800 = maximum
FFFF F120 0024	4	B	Message 0: Enable SNMP trap (Boolean). 0 = disable; non-0 = enable
FFFF F120 0028	4	UI	Message 0: SNMP trap period in seconds 0 = send once; 604800 = maximum
FFFF F120 002C	4	UI	Message 0: SNMP trap type
FFFF F120 0030	4	UI	Message 0: Priority. 0 = high; 1 = low. High is default. Applies only if you are using SNMP with a modem connection, and the I/O unit is dialing out.
FFFF F120 0034	4	--	Reserved
FFFF F120 0038	4	B	Message 0: Enable serial module message (Boolean). 0 = disable; non-0 = enable
FFFF F120 003C	4	M	Message 0: Serial ports to receive message (mask) Bits 0–31 correspond to ports 0–31.
FFFF F120 0040	80	S-ZT	Message 0: Message text. Limited to 127 characters. Used with email, serial, and (optional) SNMP messages. For memory map copying, source memory map address or data goes here. Plugins can be used in message text; see page 50 for information.
FFFF F120 00C0	4	UI	Message 1: Current message state 0 = Inactive, 1 = Active, 2 = Acknowledged
FFFF F120 00C4	8	M	Message 1: Scratch Pad bits on? (mask)
FFFF F120 00CC	8	M	Message 1: Scratch Pad bits off? (mask)
FFFF F120 00D4	4	B	Message 1: Enable streaming (Boolean). Be sure to fill in streaming configuration section. (See page 146.)
FFFF F120 00D8	4	UI	Message 1: Stream period in seconds 0 = send once, 604800 = maximum
FFFF F120 00DC	4	B	Message 1: Enable email (Boolean) Be sure to fill in email configuration section. (See page 175.)
FFFF F120 00E0	4	UI	Message 1: Email period in seconds 0 = send once; 604800 = maximum
FFFF F120 00E4	4	B	Message 1: Enable SNMP trap (Boolean). 0 = disable; non-0 = enable
FFFF F120 00E8	4	UI	Message 1: SNMP trap period in seconds 0 = send once; 604800 = maximum
FFFF F120 00EC	4	UI	Message 1: SNMP trap type
FFFF F120 00F0	4	UI	Message 1: Priority. 0 = high; 1 = low. High is default. Applies only if you are using SNMP with a modem connection, and the I/O unit is dialing out.
FFFF F120 00F4	4	--	Reserved

Starting Address	Length (hex)	Type	Description
FFFF F120 00F8	4	B	Message 1: Enable serial module message (Boolean). 0 = disable; non-0 = enable
FFFF F120 00FC	4	M	Message 1: Serial ports to receive message (mask) Bits 0–31 correspond to ports 0–31.
FFFF F120 0100	80	S-ZT	Message 1: Message text. Limited to 127 characters. Used with email, serial, and (optional) SNMP messages. For memory map copying, source memory map address or data goes here. Plugins can be used in message text; see page 50 for information.
(Additional messages follow in order.)			
FFFF F120 5F40	4	UI	Message 127: Current message state
FFFF F120 6000	1FFF	--	Reserved
FFFF F120 8000	4	UI	Message 0, memory map copying: Memory map address to copy data to (destination address), on the same or a different I/O unit.
FFFF F120 8004	4	IP	Message 0, memory map copying: IP address of destination I/O unit. Use 0.0.0.0 if copying to an address on the same I/O unit.
FFFF F120 8008	4	UI	Message 0, memory map copying: IP port for destination I/O unit. (Ignored if same I/O unit.)
FFFF F120 800C	4	UI	Message 0, memory map copying: How often to copy data, in milliseconds. (0 = send once)
FFFF F120 8100	4	UI	Message 1, memory map copying: Memory map address to copy data to (destination address), on the same or a different I/O unit.
FFFF F120 8104	4	IP	Message 1, memory map copying: IP address of destination I/O unit. Use 0.0.0.0 if copying to an address on the same I/O unit.
FFFF F120 8108	4	UI	Message 1, memory map copying: IP port for destination I/O unit. (Ignored if same I/O unit.)
FFFF F120 810C	4	UI	Message 1, memory map copying: How often to copy data, in milliseconds. (0 = send once)
(Additional messages follow in order.)			
FFFF F120 87F0	4	UI	Message 127, memory map copying: Memory map address to copy data to (destination address), on the same or a different I/O unit.
FFFF F120 9000	100	S-ZT	Message 0: Text of most recent message (read only)
FFFF F120 9100	4	UI	Message 0: Length of most recent message (read only)
FFFF F120 9104	100	S-ZT	Message 1: Text of most recent message (read only)
FFFF F120 9204	4	UI	Message 1: Length of most recent message (read only)
(Additional messages follow in order.)			
Last valid address for this area: FFFF F121 11FF			

Email Configuration—Read/Write

PAC-S
PAC-R
EB
UIO
EIO
G4EB2

Use this section to set up email addresses to which the I/O unit will send event messages or logged data. Also use "Event Message Configuration—Read/Write" on page 172 to set up the messages or "Data Logging Configuration—Read/Write" on page 187 to set up data logging. See Chapter 2 for more information on configuring events and messages and logging data.

Starting Address	Length (hex)	Type	Description
FFFF F130 0000	4		IP address of the SMTP mail server the I/O unit will use to send email.
FFFF F130 0004	4	UI	SMTP port number for sending messages (usually 25).
FFFF F130 0008	32	S-ZT	TO: Email address of the person to receive the message.
FFFF F130 003A	32	S-ZT	FROM: Email address of the I/O unit. Use an address that will identify the I/O unit to the recipient of the message.
FFFF F130 006C	32	S-ZT	RE: Subject of the email. Plugins are allowed. See page 50 for more information.
FFFF F130 00A0	4	UI	Timeout: the length of time in milliseconds the I/O unit should wait for a response from the email server. Default is 30,000. On PAC-R and EB, if no connection is made, the device stores up to 129 messages. When communication is restored, all stored messages are sent.

Serial Event Configuration—Read/Write

PAC-R
EB
UIO
EIO

(Does not apply to SB brains or SNAP Simple I/O) Use this section to set up events for serial communication modules attached to the SNAP Ethernet-based I/O unit. For more information on serial communication modules, see Opto 22 form #1191, the *SNAP Serial Communication Module User's Guide*. For more information on serial events and reactions, see page 48.

Before you configure serial events, configure the serial modules. (See page 134.) In the following table, only the first two events are shown. Other events follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F154 0000	4	M	Serial Event 0: Serial port mask. Bits 0-31 correspond to ports 0-31. On bits represent serial ports to monitor for the event string. Event occurs if *any* of the ports receives the event string. NOTE: A null entry here disables this entry and any higher numbered entries.
FFFF F154 0004	4	--	Reserved
FFFF F154 0008	4	UI	Serial Event 0: SNMP trap type (if sending an SNMP trap as a reaction to the serial event)
FFFF F154 000C	4	UI	Serial Event 0: SNMP trap period—how often, in seconds, to send the trap as a reaction to the serial event
FFFF F154 0010	28	S-ZT	Serial Event 0: Pattern string for the event. Wildcards (* and ?) are allowed.

Starting Address	Length (hex)	Type	Description
FFFF F154 0038	28	S-ZT	Serial Event 0: Reaction string to be sent with the SNMP trap. Plug-ins allowed ($!_str_ or $!_port_ and others). See page 50 for more information.
FFFF F154 0060	8	M	Serial Event 0: Scratch Pad bits to turn on
FFFF F154 0068	8	M	Serial Event 0: Scratch Pad bits to turn off
FFFF F154 0070	4	UI	Serial Event 0: Enable email message (0 = Disable, 1 = Enable)
FFFF F154 0074	4	M	Serial Event 1: Serial port mask. Bits 0-31 correspond to ports 0-31. On bits represent serial ports to monitor for the event string. Event occurs if *any* of the ports receives the event string. NOTE: A null entry here disables this entry and any higher numbered entries.
FFFF F154 0074	4	--	Reserved
FFFF F154 007C	4	UI	Serial Event 1: SNMP trap type (if sending an SNMP trap as a reaction to the serial event)
FFFF F154 0080	4	UI	Serial Event 1: SNMP trap period—how often, in seconds, to send the trap as a reaction to the serial event
FFFF F154 0084	28	S-ZT	Serial Event 1: Pattern string for the event. Wildcards (* and ?) are allowed.
FFFF F154 00AC	28	S-ZT	Serial Event 1: Reaction string to be sent with the SNMP trap. Plug-ins allowed ($!_str_ or $!_port_ and others). See page 50 for more information.
FFFF F154 00D4	8	M	Serial Event 1: Scratch Pad bits to turn on
FFFF F154 00DC	8	M	Serial Event 1: Scratch Pad bits to turn off
FFFF F154 00E4	4	UI	Serial Event 1: Enable email message (0 = Disable, 1 = Enable)
(Additional events follow in order.)			
FFFF F154 0E80	4	UI	Serial Event 0: SNMP trap priority. High = 0; low = 1. High is default. Applies only if you are using SNMP with a modem connection, and the I/O unit is dialing out.
FFFF F154 0E84	4	UI	Serial Event 1: SNMP trap priority (0 = High, 1 = Low)
FFFF F154 0E88	4	UI	Serial Event 2: SNMP trap priority (0 = High, 1 = Low)
(Trap priority addresses for additional events follow in order.)			
FFFF F154 0F00	4	UI	Serial Event 0: Disable SNMP trap (0 = No, 1 = Yes)
FFFF F154 0F04	4	UI	Serial Event 1: Disable SNMP trap (0 = No, 1 = Yes)
FFFF F154 0F08	4	UI	Serial Event 2: Disable SNMP trap (0 = No, 1 = Yes)
(Disable trap addresses for additional events follow in order.)			
Last valid address for this area: FFFF F154 0EFC			

Wiegand Serial Event Configuration—Read/Write

PAC-REB UIO EIO

Use this section to set up events for Wiegand serial communication modules attached to the SNAP Ethernet-based I/O unit. For more information on Wiegand modules, see Opto 22 form #1191, the *SNAP Serial Communication Module User's Guide*. For more information on serial events and reactions, see page 48.

Before you configure Wiegand serial events, configure the Wiegand modules. (See page 136.) In the following table, only the first two events are shown. Other events follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F156 0000	4	M	Wiegand Event 0: Wiegand port mask. Bits 0-31 correspond to ports 0-31. On bits represent serial ports to monitor for the event string. Event occurs if *any* of the ports receives the event string. NOTE: A null entry here disables this entry and any higher numbered entries.
FFFF F156 0004	4	--	Reserved
FFFF F156 0008	4	UI	Wiegand Event 0: SNMP trap type (if sending an SNMP trap as a reaction to the serial event)
FFFF F156 000C	4	UI	Wiegand Event 0: SNMP trap period—how often, in seconds, to send the trap as a reaction to the serial event
FFFF F156 0010	28	S-ZT	Wiegand Event 0: Pattern string for the event. Wildcards (* and ?) are allowed.
FFFF F156 0038	28	S-ZT	Wiegand Event 0: Reaction string to be sent with the SNMP trap. Plug-ins allowed ($!_str_ or $!_port_ and others). See page 50 for more information.
FFFF F156 0060	8	M	Wiegand Event 0: Scratch Pad bits to turn on
FFFF F156 0068	8	M	Wiegand Event 0: Scratch Pad bits to turn off
FFFF F156 0070	4	UI	Wiegand Event 0: Enable email message (0 = Disable, 1 = Enable)
FFFF F156 0074	4	M	Wiegand Event 1: Wiegand port mask. Bits 0–31 correspond to ports 0–31. On bits represent ports to monitor for the event string. Event occurs if *any* of the ports receives the event string. NOTE: A null entry here disables this entry and any higher numbered entries.
FFFF F156 0078	4	--	Reserved
FFFF F156 007C	4	UI	Wiegand Event 1: SNMP trap period—how often, in seconds, to send the trap as a reaction to the event
FFFF F156 0080	4	UI	Wiegand Event 1: SNMP trap type (if sending an SNMP trap as a reaction to the event)
FFFF F156 0084	28	S-ZT	Wiegand Event 1: Pattern string for the event. Wildcards (* and ?) are allowed.
FFFF F156 00AC	28	S-ZT	Wiegand Event 1: Reaction string to be sent with the SNMP trap. Plug-ins allowed ($!_str_ or $!_port_ and others). See page 50 for more information.
FFFF F156 00D4	8	M	Wiegand Event 1: Scratch Pad bits to turn on
FFFF F156 00DC	8	M	Wiegand Event 1: Scratch Pad bits to turn off
FFFF F156 00E4	4	UI	Wiegand Event 1: Enable email message (0 = Disable, 1 = Enable)

Starting Address	Length (hex)	Type	Description
(Additional events follow in order.)			
FFFF F156 0E80	4	UI	Wiegand Event 0: SNMP trap priority. High = 0; low = 1. High is default. Applies only if you are using SNMP with a modem connection, and the I/O unit is dialing out. See the device's user guide for details.
FFFF F156 0E84	4	UI	Wiegand Event 1: SNMP trap priority (0 = High, 1 = Low)
FFFF F156 0E88	4	UI	Wiegand Event 2: SNMP trap priority (0 = High, 1 = Low)
(Trap priority addresses for additional events follow in order.)			
FFFF F156 0F00	4	UI	Wiegand Event 0: Disable SNMP trap (0 = No, 1 = Yes)
FFFF F156 0F04	4	UI	Wiegand Event 1: Disable SNMP trap (0 = No, 1 = Yes)
FFFF F156 0F08	4	UI	Wiegand Event 2: Disable SNMP trap (0 = No, 1 = Yes)
(Disable SNMP traps for additional events follow in order.)			
Last valid address for this area: FFFF F156 0F7F			

SNAP High-Density Digital—Read Only

PAC-REB SB UIO EIO SIO

Use this section to read the state of points on SNAP high-density digital input and output modules on the I/O unit; also use it to read latches and counters for high-density input points.

IMPORTANT: For state and latches, data is shown in 64-bit masks. The lower 32 bits correspond to the 32 channels on the module, with bit 0 representing point 0. The upper 32 bits are not currently used. For example, for Module 0 point state, addresses F1808000 through F1808003 will be zero-filled; address F1808004 will contain data for points 31 through 24, and so on.

Starting Address	Length (hex)	Type	Description
FFFF F180 8000	8	M	Module 0: Point state. 0 = Off; 1 = On
FFFF F180 8008	8	M	Module 0: On-latch state. 0 = Off; 1 = On
FFFF F180 8010	8	M	Module 0: Off-latch state. 0 = Off; 1 = On
FFFF F180 8018	28	--	Module 0: Reserved
FFFF F180 8040	8	M	Module 1: Point state. 0 = Off; 1 = On
FFFF F180 8048	8	M	Module 1: On-latch state. 0 = Off; 1 = On
FFFF F180 8050	8	M	Module 1: Off-latch state. 0 = Off; 1 = On
FFFF F180 8058	28	--	Module 1: Reserved
(Additional modules follow in order on even 40 hex boundaries.)			
FFFF F180 83C0	8	M	Module 15: Point state. 0 = Off; 1 = On
FFFF F180 83C8	8	M	Module 15: On-latch state. 0 = Off; 1 = On
FFFF F180 83D0	8	M	Module 15: Off-latch state. 0 = Off; 1 = On
FFFF F180 83D8	28	--	Module 15: Reserved
FFFF F180 9000	4	UI	Module 0, Point 0 (0,0): Counter value
FFFF F180 9004	4	UI	0,1: Counter value
FFFF F180 9008	4	UI	0,2: Counter value
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F180 9100	4	UI	1,0: Counter value
FFFF F180 9104	4	UI	1,1: Counter value
FFFF F180 9108	4	UI	1,2; Counter value
(Additional points and modules follow in order.)			
FFFF F180 9F00	4	UI	15,0: Counter value
Last valid address for this area: FFFF F180 9FFE			

SNAP High-Density Digital Read and Clear—Read/Write

PAC-R
EB
SB
UIO
EIO
SIO

When you read data from this area, the data for all SNAP high-density digital points is returned and then cleared. To clear individual latches, read latches first if needed (see"SNAP High-Density Digital—Read Only" on page 179) and then write a mask to this section with bits set corresponding to the latches to clear.

IMPORTANT: Latch data is shown in 64-bit masks. The lower 32 bits correspond to the 32 channels on the module, with bit 0 representing point 0. The upper 32 bits are not currently used. For example, for Module 0 on-latch state, addresses F180A000 through F180A003 will be zero-filled; address F180A004 will contain data for points 31 through 24, and so on.

Starting Address	Length (hex)	Type	Description
FFFF F180 A000	8	M	Module 0: On latch state. 0 = Off; 1 = On
FFFF F180 A008	8	M	Module 0: Off latch state. 0 = Off; 1 = On
FFFF F180 A010	10	--	Module 0: Reserved
FFFF F180 A020	8	M	Module 1: On latch state. 0 = Off; 1 = On
FFFF F180 A028	8	M	Module 1: Off latch state. 0 = Off; 1 = On
FFFF F180 A030	10	--	Module 1: Reserved
(Additional modules follow in order on even 20 hex boundaries.)			
FFFF F180 A1E0	8	M	Module 15: On latch state. 0 = Off; 1 = On
FFFF F180 A1E8	8	M	Module 15: Off latch state. 0 = Off; 1 = On
FFFF F180 A1F0	10	--	Module 15: Reserved
FFFF F180 B000	4	UI	Module 0, Point 0 (0,0): Returns counter value and sets counter to zero.
FFFF F180 B004	4	UI	0,1: Returns counter value and sets counter to zero.
FFFF F180 B008	4	UI	0,2: Returns counter value and sets counter to zero.
(Additional points follow in order on even 4 hex boundaries.)			
FFFF F180 B100	4	UI	1,0: Returns counter value and sets counter to zero.
FFFF F180 B104	4	UI	1,1: Returns counter value and sets counter to zero.
FFFF F180 B108	4	UI	1,2: Returns counter value and sets counter to zero.
(Additional points and modules follow in order.)			
FFFF F180 BF00	4	UI	15,0: Returns counter value and sets counter to zero.
Last valid address for this area: FFFF F180 BFFE			

SNAP High-Density Digital Write—Read/Write

PAC-REB SB UIO EIO SIO

Use this section to write to SNAP high-density digital output modules. Data is sent in 64-bit masks. The lower 32 bits correspond to the 32 channels on the module, with bit 0 representing point 0. The upper 32 bits are not currently used. For example, the On mask for Module 0 would appear as shown in the following table:

This address	Contains this data
F180C000	Zero-filled (not currently used)
F180C001	
F180C002	
F180C003	
F180C004	Points 31–24
F180C005	Points 23–16
F180C006	Points 15–8
F180C007	Points 7–0

NOTE: If the same bit is set in both the On mask and the Off mask, the point will be turned off.

Starting Address	Length (hex)	Type	Description
FFFF F180 C000	8	M	Module 0: On mask (bitmask representing points to turn on.) 1 = Turn on; 0 = Ignore
FFFF F180 C008	8	M	Module 0: Off mask (bitmask representing points to turn off.) 1 = Turn off; 0 = Ignore
FFFF F180 C010	30	--	Module 0: Reserved
FFFF F180 C040	8	M	Module 1: On mask (bitmask representing points to turn on.) 1 = Turn on; 0 = Ignore
FFFF F180 C048	8	M	Module 1:Off mask (bitmask representing points to turn off.) 1 = Turn off; 0 = Ignore
FFFF F180 C050	30	--	Module 1: Reserved
(Additional modules follow in order on even 40 hex boundaries.)			
FFFF F180 C3C0	8	M	Module 15: On mask (bitmask representing points to turn on.) 1 = Turn on; 0 = Ignore
FFFF F180 C3C8	8	M	Module 15:Off mask (bitmask representing points to turn off.) 1 = Turn off; 0 = Ignore
FFFF F180 C3D0	30	--	Module 15: Reserved
Last valid address for this area: FFFF F180 C3FE			

PID Configuration and Status—Read/Write

PAC-R
EB
SB
UIO
EIO

Use this section to configure proportional/integral/ derivative (PID) loops to run on the I/O unit and to check PID status. (For configuring PID *modules*, see page 188.) For additional information on using PID loops, see page 55. Number of PID loops available depends on the I/O unit: there are 16 on SNAP Ethernet, 32 on SNAP Ultimate, and 96 on SNAP PAC R-series and SNAP PAC EB I/O units.

The first table shows addresses for current algorithm values and status flags. The second table includes scanned values and configuration addresses. In both tables, only the first two PIDs are shown. Others follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F200 0000	4	F	PID 0: Current value for Error (Input – Setpoint)
FFFF F200 0004	4	F	PID 0: Current value for P (gain) contribution
FFFF F200 0008	4	F	PID 0: Current value for I (integral) contribution
FFFF F200 000C	4	F	PID 0: Current value for D (derivative) contribution
FFFF F200 0010	4	F	PID 0: Current value for Integral
FFFF F200 0014	14	--	Reserved for internal use
FFFF F200 0028	4	UI	PID 0: Scan counter
FFFF F200 002C	4	UI	PID 0: Status flags: 1 = Input is below Input low scale; 2 = Input is above Input high scale; 4 = Output has been forced to a predetermined level because input is out of range.
FFFF F200 0030	4	UI	PID 0: Status bits ON. Used to control individual Status flag bits. Write a 1 to turn the bit on.
FFFF F200 0034	4	UI	PID 0: Status bits OFF. Used to control individual Status flag bits. Write a 1 to clear the bit.
FFFF F200 0038	4	F	PID 0: Current input value when input is provided by a host (application or supervisory system). When F210 0044 is set to zero, meaning that the input value is coming from a host, the host writes the Input value here. Applies only to Velocity C algorithm and firmware 8.5e or higher.
FFFF F200 003C	14	--	Reserved
FFFF F200 0050	4	F	PID 1: Current value for Error (Input – Setpoint)
FFFF F200 0054	4	F	PID 1: Current value for P (gain) contribution
FFFF F200 0058	4	F	PID 1: Current value for I (integral) contribution
FFFF F200 005C	4	F	PID 1: Current value for D (derivative) contribution
FFFF F200 0060	4	F	PID 1: Current value for Integral
FFFF F200 0064	14	--	Reserved for internal use
FFFF F200 0078	4	UI	PID 1: Scan counter
FFFF F200 007C	4	UI	PID 1: Status flags: 1 = PV low; 2 = PV high; 4 = output forced
FFFF F200 0080	4	UI	PID 1: Status bits ON. Used to control individual Status flag bits. Write a 1 to turn the bit on.

Starting Address	Length (hex)	Type	Description
FFFF F200 0084	4	UI	PID 1: Status bits OFF. Used to control individual Status flag bits. Write a 1 to clear the bit.
FFFF F200 0088	4	F	PID 1: Current input value when input is provided by a host (application or supervisory system). When F210 0044 is set to zero, meaning that the input value is coming from a host, the host writes the Input value here. Applies only to Velocity C algorithm and firmware 8.5e or higher.
FFFF F200 008C	14	--	Reserved
(Additional PIDs follow in order on even 50 hex boundaries. See additional PID addresses in the table below.)			
Last valid address for this area: FFFF F200 2EDF			

This second table shows scanned values and configuration addresses. Only the first two PIDs are shown. Others follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F210 0000	4	F	PID 0: Input or process variable (PV) value. If the value at memory map address F2100044 is not zero, the value at that address is copied into this location prior to each PID calculation.
FFFF F210 0004	4	F	PID 0: Setpoint value. If the value at memory map address F2100048 is not zero, the value at that address is copied into this location prior to each PID calculation.
FFFF F210 0008	4	F	PID 0: Current value of feed forward
FFFF F210 000C	4	F	PID 0: Current value of Output
FFFF F210 0010	4	F	PID 0: Tuning: P (Gain) value. Should be positive for cooling, negative for heating.
FFFF F210 0014	4	F	PID 0: Tuning: I (Integral) value, in units of one per second. Increase value to increase contribution.
FFFF F210 0018	4	F	PID 0: Tuning: D (Derivative) value, in seconds. Increase value to increase contribution.
FFFF F210 001C	4	F	PID 0: Tuning: Feed forward gain
FFFF F210 0020	4	F	PID 0: Configuration: Maximum output change allowed. Zero disables this feature.
FFFF F210 0024	4	F	PID 0: Configuration: Minimum output change allowed. Zero disables this feature.
FFFF F210 0028	4	F	PID 0: Scaling: Input low range
FFFF F210 002C	4	F	PID 0: Scaling: Input high range
FFFF F210 0030	4	F	PID 0: Scaling: Output lower clamp
FFFF F210 0034	4	F	PID 0: Scaling: Output upper clamp
FFFF F210 0038	4	F	PID 0: Configuration: PID scan time in seconds. Minimum value is 0.001 (1 millisecond).
FFFF F210 003C	4	F	PID 0: Configuration: If Input (PV) is below low range, output is set to this value.

Starting Address	Length (hex)	Type	Description
FFFF F210 0040	4	F	PID 0: Configuration: If Input (PV) is above high range, output is set to this value.
FFFF F210 0044	4	UI	PID 0: Input memory map address for PID input or cascading PIDs. Specifies the memory map address from which the input value is copied to F2100000 prior to each PID calculation. If this address is 0, no input value is copied, and the input value must be written directly to F2100038 by an application. Use 0 when supervisory system supplies or processes this value.
FFFF F210 0048	4	UI	PID 0: For cascading PIDs. Specifies the memory map address from which the setpoint value is copied to F2100004 prior to each PID calculation. If this address is 0, no setpoint value is copied, and the setpoint value must be written directly to F2100004 by an application. Use 0 when supervisory system supplies or processes this value.
FFFF F210 004C	4	UI	PID 0: For cascading PIDs. Specifies the memory map address to which the output value is written after each PID calculation. If this address is 0, the output value is not written. Use 0 when supervisory system supplies or processes this value.
FFFF F210 0050	4	UI	PID 0: Configuration: Algorithm choice. 0 = PID disabled 1 = Velocity algorithm (Type B; deprecated in favor of Type C) 2 = ISA algorithm 3 = Parallel algorithm 4 = Interacting algorithm 5 = Velocity algorithm (Type C; firmware 8.5e and higher)
FFFF F210 0054	4	UI	PID 0: Configuration: Manual mode. 1 = Yes; 0 = No
FFFF F210 0058	4	UI	PID 0: Configuration flags: 1 = Enable square root of Input. 2 = Force output when Input is out of range. Set range in addresses F210 0028 and F210 002C. 4 = Switch to manual mode when input goes out of range.
FFFF F210 005C	4	UI	PID 0: Configuration bits ON. Used to control individual Configuration flag bits. Write a 1 to turn the bit on.
FFFF F210 0060	4	UI	PID 0: Configuration bits OFF. Used to control individual Configuration flag bits. Write a 1 to clear the bit.
FFFF F210 0064	4	F	PID 0: Current value at the memory map address from which the input value is copied.
FFFF F210 0068	4	F	PID 0: Current value at the memory map address from which the setpoint value is copied.
FFFF F210 006C	14	--	Reserved
FFFF F210 0080	4	F	PID 1: Input or process variable (PV) value. If the value at memory map address F21000C4 is not zero, the value at that address is copied into this location prior to each PID calculation.
FFFF F210 0084	4	F	PID 1: Setpoint value. If the value at memory map address F21000C8 is not zero, the value at that address is copied into this location prior to each PID calculation.
FFFF F210 0088	4	F	PID 1: Current value of feed forward
FFFF F210 008C	4	F	PID 1: Current value of Output

Starting Address	Length (hex)	Type	Description
FFFF F210 0090	4	F	PID 1: Tuning: P (Gain) value. Should be positive for cooling, negative for heating.
FFFF F210 0094	4	F	PID 1: Tuning: I (Integral) value, in units of one per second. Increase value to increase contribution.
FFFF F210 0098	4	F	PID 1: Tuning: D (Derivative) value, in seconds. Increase value to increase contribution.
FFFF F210 009C	4	F	PID 1: Tuning: Feed forward gain
FFFF F210 00A0	4	F	PID 1: Configuration: Maximum output change allowed. Zero disables this feature.
FFFF F210 00A4	4	F	PID 1: Configuration: Minimum output change allowed. Zero disables this feature.
FFFF F210 00A8	4	F	PID 1: Scaling: Input low range
FFFF F210 00AC	4	F	PID 1: Scaling: Input high range
FFFF F210 00B0	4	F	PID 1: Scaling: Output lower clamp
FFFF F210 00B4	4	F	PID 1: Scaling: Output upper clamp
FFFF F210 00B8	4	F	PID 1: Configuration: PID scan time in seconds. Minimum value is 0.001 (1 millisecond).
FFFF F210 00BC	4	F	PID 1: Configuration: If Input (PV) is below low range, output is set to this value.
FFFF F210 00C0	4	F	PID 1: Configuration: If Input (PV) is above high range, output is set to this value.
FFFF F210 00C4	4	UI	PID 1: Input memory map address for PID input or cascading PIDs. Specifies the memory map address from which the input value is copied to F2100000 prior to each PID calculation. If this address is 0, no input value is copied, and the input value must be written directly to F2100038 by an application. Use 0 when supervisory system supplies or processes this value.
FFFF F210 00C8	4	UI	PID 1: For cascading PIDs. Specifies the memory map address from which the setpoint value is copied to F2100084 prior to each PID calculation. If this address is 0, no setpoint value is copied, and the setpoint value must be written directly to F2100084 by an application. Use 0 when supervisory system supplies or processes this value.
FFFF F210 00CC	4	UI	PID 1: For cascading PIDs. Specifies the memory map address to which the output value is written after each PID calculation. If this address is 0, the output value is not written. Use 0 when supervisory system supplies or processes this value.
FFFF F210 00D0	4	UI	PID 1: Configuration: Algorithm choice. 0 = PID disabled 1 = Velocity algorithm (Type B; deprecated in favor of Type C) 2 = ISA algorithm 3 = Parallel algorithm 4 = Interacting algorithm 5 = Velocity algorithm (Type C; firmware 8.5e and higher)
FFFF F210 00D4	4	UI	PID 1: Configuration: Manual mode. 1 = Yes; 0 = No

Starting Address	Length (hex)	Type	Description
FFFF F210 00D8	4	UI	PID 1: Configuration flags: 1 = Enable square root of Input. 2 = Force output when Input is out of range. Set range at addresses F210 00A8 and F210 00AC. 4 = Switch to manual mode when input goes out of range.
FFFF F210 00DC	4	UI	PID 1: Configuration bits ON. Used to control individual Configuration flag bits. Write a 1 to turn the bit on.
FFFF F210 00E0	4	UI	PID 1: Configuration bits OFF. Used to control individual Configuration flag bits. Write a 1 to clear the bit.
FFFF F210 00E4	4	F	PID 1: Current value at the memory map address from which the input value is copied.
FFFF F210 00E8	4	F	PID 1: Current value at the memory map address from which the set-point value is copied.
FFFF F210 00EC	14	--	Reserved
(Additional PIDs follow in order on even 80 hex boundaries.)			
Last valid address for this area: FFFF F210 47FF			

Data Logging Configuration—Read/Write

PAC-REB SB UIO EIO G4EB2

Use this section to configure up to 64 memory map addresses you wish to log data from (log points). The values from all log points go into the same data log file, which you can access using the data log section of the memory map (page 188). Data from this composite file can be emailed; to do so, also use page 175 to set up email. See page 54 for more information on data logging.

Only the first three log points are shown. Other log points follow the same pattern.

Starting Address	Length (hex)	Type	Description
FFFF F300 0000	8	M	Log Point 0: Scratch Pad bits on? (mask)
FFFF F300 0008	8	M	Log Point 0: Scratch Pad bits off? (mask)
FFFF F300 0010	4	UI	Log Point 0: Memory map address to log data from
FFFF F300 0014	4	UI	Log Point 0: Data format of value to be logged
FFFF F300 0018	4	UI	Log Point 0: How often to log the value, in milliseconds (If you change this interval, you must save to flash and restart the I/O unit.)
FFFF F300 001C	8	M	Log Point 1: Scratch Pad bits on? (mask)
FFFF F300 0024	8	M	Log Point 1: Scratch Pad bits off? (mask)
FFFF F300 002C	4	UI	Log Point 1: Memory map address to log data from
FFFF F300 0030	4	UI	Log Point 1: Data format of value to be logged
FFFF F300 0034	4	UI	Log Point 1: How often to log the value, in milliseconds (If you change this interval, you must save to flash and restart the I/O unit.)
FFFF F300 0038	8	M	Log Point 2: Scratch Pad bits on? (mask)
FFFF F300 0040	8	M	Log Point 2: Scratch Pad bits off? (mask)
FFFF F300 0048	4	UI	Log Point 2: Memory map address to log data from
FFFF F300 004C	4	UI	Log Point 2: Data format of value to be logged
FFFF F300 0050	4	UI	Log Point 2: How often to log the value, in milliseconds (If you change this interval, you must save to flash and restart the I/O unit.)
(Additional log points follow in order.)			
FFFF F300 0700	4	B	Enable email for sending data log file. Applies to ALL log points. Also configure email (page 175).
FFFF F300 0704	4	UI	Number of data log entries in each email message. Applies to ALL log points.
Last valid address for this area: FFFF F300 0707			

Data Log—Read/Write

PAC-R
EB
SB
UIO
EIO
G4EB2

(Does not apply to SNAP Simple I/O) Use this section to access data logged from memory map addresses set up in "Data Logging Configuration—Read/Write" on page 187. The data from all addresses goes into one data log file. The data log is a circular buffer; it holds 300 samples of 14 bytes (hex) per entry, and the newest data item replaces the oldest one. See "Logging Data" on page 54 for more information on data logging, and see page 54 for how to interpret information from these data log addresses.

Although this area is read/write, you would normally read it only. You could write to it to clear it, however.

Starting Address	Length (hex)	Type	Description
FFFF F302 0000	14	*	First entry in the data log
FFFF F302 0014	14	*	Second entry in the data log
Additional entries follow in order.			
FFFF F302 175C	14	*	Last entry in the data log
Last valid address in this area: FFFF F302 176F			

* Binary data. See page 54.

PID Module Configuration—Read/Write

PAC-R
UIO
EIO

Use this area to configure PID modules on the rack. (**NOTE:** PID modules are at their end of life and are *not recommended* for new development. Instead, use PID loops that run on the I/O unit. To configure PID loops, see page 182.)

The table shows addresses for a PID module in rack position zero. Modules in other positions on the rack follow the same pattern and start on even 100 hex boundaries.

CAUTION: Values for many of these memory map addresses require you to calculate them in advance. Some values are read-only and cannot be changed. For explanations and important information on using these addresses, see Opto 22 form 1263, the SNAP PID Module User's Guide.

Starting Address	Length (hex)	Type	Description
FFFF F400 0000	4	I	PID module in position 0 on the rack: Control word
FFFF F400 0004	4	I	Position 0: State of status flags (read only)
FFFF F400 0008	4	I	Position 0: Scantime base (scantime resolution)
FFFF F400 000C	4	I	Position 0: Scantime multiplier
FFFF F400 0010	4	I	Position 0: TPO period multiplier
FFFF F400 0014	4	I	Position 0: Output
FFFF F400 0018	4	I	Position 0: Tune PID: proportional (gain)
FFFF F400 001C	4	I	Position 0: Tune PID: integral ratio

Starting Address	Length (hex)	Type	Description
FFFF F400 0020	4	I	Position 0: Tune PID: derivative ratio
FFFF F400 0024	4	I	Position 0: Setpoint
FFFF F400 0028	4	I	Position 0: Process variable
FFFF F400 002C	4	I	Position 0: Filter exponential
FFFF F400 0030	4	I	Position 0: Setpoint low limit
FFFF F400 0034	4	I	Position 0: Setpoint high limit
FFFF F400 0038	4	I	Position 0: Process low limit
FFFF F400 003C	4	I	Position 0: Process high limit
FFFF F400 0040	4	I	Position 0: Output low limit
FFFF F400 0044	4	I	Position 0: Output high limit
FFFF F400 0048	4	I	Position 0: Output slew rate
FFFF F400 004C	4	I	Position 0: Output limit deadband
FFFF F400 0050	4	I	Position 0: Current PID (read only)
FFFF F400 0054	4	I	Position 0: Last PID (read only)
FFFF F400 0058	4	I	Position 0: Oldest PID (read only)
FFFF F400 005C	4	I	Position 0: Current PID error (read only)
FFFF F400 0060	4	I	Position 0: Last PID error (read only)
FFFF F400 0064	4	I	Position 0: Output change (read only)
FFFF F400 0068	4	I	Position 0: Output value (read only)
FFFF F400 006C	4	I	Position 0: Scantime countdown (read only)
FFFF F400 0100	4	I	PID module in position 1 on the rack: Control word
(Additional modules follow in order on even 100 hex boundaries.)			
FFFF F400 0F00	4	I	PID module in position 15 on the rack: Control word
Last valid address for this area: FFFF F400 0F6F			

Control Engine—Read/Write

PAC-S PAC-R

Use this section to view and change features in R-series and S-series PACs. For more information on these features, see the *PAC Control User's Guide*, form 1700.

Starting Address	Length (hex)	Type	Description
FFFF F408 0000	4	UI	Control engine operation. 0 = control engine disabled; 1 = control engine enabled. On a PAC R controller, disabling the control engine means all processor resources are available for use by the brain.
FFFF F408 0004	4	UI	Control engine feature. 0 = No feature; 1 = background downloading; 2 = secure strategy downloads; 3 = redundant controller (Do not set the redundant controller feature here; use the PAC Redundancy Manager software in PAC Project Professional 9.0 or newer to set it.)
Last valid address in this area: FFFF F408 0007			

Serial Brain Communication—Read/Write

SB

Use this section with SNAP PAC SB brains.

Starting Address	Length (hex)	Type	Description
FFFF F700 2000	4	UI	Turnaround time delay, port 0. Number of ms the brain should wait before sending the response back to the host.
FFFF F700 2100	4	B	Communications debug flag. 1 = communication error blink codes are enabled; 0 = communication error blink codes are disabled. These blink codes help in troubleshooting. Do not leave them enabled for normal operation, as they slow down processing speed. See the brain user's guide for more information.
Last valid address in this area: FFFF F700 2103			

microSD Card—Read/Write

PAC-S PAC-R

Use this section with microSD cards in R-series and S-series PACs. Also see address F030 0230 in the "Status Area Read—Read Only" section (page 119). See the PAC's user guide for additional information on using microSD cards.

Starting Address	Length (hex)	Type	Description
FFFF F700 2200	4	B	Enable/disable use of microSD card to update firmware and strategies. Non-zero = enabled; 0 = disabled. Default is enabled. Requires firmware R8.4a or newer.
FFFF F700 2204	4	UI	Number of bytes free on the microSD card. Requires firmware R8.5a or newer.
Last valid address in this area: FFFF F700 2207			

WLAN Status—Read Only

PAC-S
PAC-R
EB

Wired+Wireless models only.

Use this section to read status information for SNAP PAC controllers and brains communicating over a wireless LAN. Also see "WLAN Enable—Read/Write" on page 194 and "WLAN Configuration—Read/Write" on page 192.

Starting Address	Length (hex)	Type	Description
FFFF F700 3000	2	--	Pad for alignment
FFFF F700 3002	6	UI	MAC address for WLAN interface.
FFFF F700 3008	4	UI	WLAN state: 0 = disconnected, 1 = inactive, 2 = scanning, 3 = associating, 4 = associated, 5 = WPA 4-way handshake, 6 = WPA group handshake, 7 = completed
FFFF F700 300C	2	--	Pad for alignment
FFFF F700 300E	6	UI	ID for current BSS (basic service set), if WLAN state = completed
FFFF F700 3014	20	S-ZT	SSID (service set identifier) of current BSS, if WLAN state = completed
FFFF F700 3034	4	UI	Channel frequency (MHz)
FFFF F700 3038	4	UI	Rate at which data is received (kbps)
FFFF F700 303C	4	UI	Rate at which data is transmitted (kbps)
FFFF F700 3040	4	UI	Signal to noise ratio (dB)
FFFF F700 3044	4	I	Signal level (dBm)
FFFF F700 3048	4	I	Noise level (dBm)
FFFF F700 304C	4	UI	Number of missed beacon frames
FFFF F700 3050	4	UI	Number of connection events
FFFF F700 3054	4	UI	Number of disconnection events
FFFF F700 3058	4	UI	Number of low signal strength events
FFFF F700 305C	4	UI	Total packets received
FFFF F700 3060	4	UI	Total packets transmitted
FFFF F700 3064	4	UI	Total bytes received
FFFF F700 3068	4	UI	Total bytes transmitted
FFFF F700 306c	4	UI	Bad packets received
FFFF F700 3070	4	UI	Packet transmit problems
FFFF F700 3074	4	UI	Incoming packets dropped due to resource constraints
FFFF F700 3078	4	UI	Outgoing packets dropped due to resource constraints
FFFF F700 307c	4	UI	Received packets that were too long or too short
FFFF F700 3080	4	UI	Received packets with wrong network ID/SSID
FFFF F700 3084	4	UI	Unable to code/decode (WEP)

Starting Address	Length (hex)	Type	Description
FFFF F700 3088	4	UI	Can't perform MAC reassembly
FFFF F700 308C	4	UI	Maximum MAC retries reached
Last valid address in this area: FFFF F700 308F			

WLAN Configuration—Read/Write

PAC-S PAC-R EB

Wired+Wireless models only. Use this section to configure wireless LAN communication parameters. You must store to flash memory and cycle power after any configuration change.

NOTE: As with all Ethernet devices, only one default gateway is in use at a time. This default gateway is always the gateway configured for the highest priority active interface (active means it has a link and is capable of communication). Default gateways are prioritized from highest to lowest priority through the following interfaces:

1. *PPP*
2. *Ethernet 0*
3. *Ethernet 1*
4. *WLAN*

Also see "WLAN Enable—Read/Write" on page 194 and "WLAN Status—Read Only" on page 191.

Starting Address	Length (hex)	Type	Description
FFFF F700 4000	4	UI	Enable/disable wireless network configuration. 0=disabled, 1=enabled. Default is disabled; must be enabled for the WLAN to work. In addition, address F800 0000 must also be enabled to use the wireless network.
FFFF F700 4004	4	IP	IP address for the wireless interface
FFFF F700 4008	4	IP	Subnet mask for the wireless interface
FFFF F700 400C	4	IP	Default Gateway IP address for the wireless interface
FFFF F700 4010	4	IP	Secondary Gateway IP address for the wireless interface
FFFF F700 4014	4	IP	Primary DNS Server IP address for the wireless interface
FFFF F700 4018	4	IP	Secondary DNS Server IP address for the wireless interface
FFFF F700 401C	24	S-ZT	Service Set Identifier (SSID, or network name). Use a literal string enclosed in double quotes, max. 32 ASCII characters inside the quotes. Or use an ASCII hex string (each ASCII character represented by two hex digits) with no quotes. Examples: Literal string in double quotes: "Hello" Equivalent ASCII hex string, no quotes: 48656C6C6F NULL=Any SSID.
FFFF F700 4040	4	--	Reserved
FFFF F700 4044	4	UI	IEEE 802.11 operation mode: 0=Infrastructure mode, 1=Ad-hoc (IBSS) mode.

Starting Address	Length (hex)	Type	Description
FFFF F700 4048	4	UI	Security protocol: 1=WPA/TKIP encryption 2=WPA2/CCMP encryption
FFFF F700 404C	4	UI	Authenticated key management protocol: 2=WPA-PSK 4=Plain text (unencrypted) or static WEP 16=Ad-hoc TKIP/CCMP encryption
FFFF F700 4050	4	UI	IEEE 802.11 authentication algorithm: 1=Open System Authentication, required by WPA/WPA2 2=Shared Key Authentication (less secure, WEP only)
FFFF F700 4054	4	UI	Unicast cipher for WPA: 1=Group keys (ad-hoc), 2=64-bit WEP key, 4=128-bit WEP key, 8=TKIP encryption, 16=AES encryption
FFFF F700 4058	4	UI	Broadcast/multicast cipher for WPA: 1=Group keys (ad-hoc), 2=64-bit WEP key, 4=128-bit WEP key, 8=TKIP encryption, 16=AES encryption
FFFF F700 405C	44	S-ZT	WPA 256-bit pre-shared key: Use 64 hex digits (32 bytes, no quotes; may represent an ASCII string or binary data). Or use ASCII passphrase (8 to 63 characters within double quotes). Set to 0xFF if unused.
FFFF F700 40A0	24	S-ZT	Static WEP key0: Use 10 hex or 26 hex digits (5 or 13 bytes, no quotes). Or use ASCII passphrase (5 or 13 characters within double quotes).
FFFF F700 40C4	24	S-ZT	Static WEP key1: Use 10 hex or 26 hex digits (5 or 13 bytes, no quotes). Or use ASCII passphrase (5 or 13 characters within double quotes).
FFFF F700 40E8	24	S-ZT	Static WEP key2: Use 10 hex or 26 hex digits (5 or 13 bytes, no quotes). Or use ASCII passphrase (5 or 13 characters within double quotes).
FFFF F700 410C	24	S-ZT	Static WEP key3: Use 10 hex or 26 hex digits (5 or 13 bytes, no quotes). Or use ASCII passphrase (5 or 13 characters within double quotes).
FFFF F700 4130	4	UI	Default WEP key: 0=WEP key0, 1=WEP key1, 2=WEP key2, 3=WEP key3.
FFFF F700 4134	4	UI	Channel for ad-hoc mode. Example: 2412 = 802.11b/g channel 1. See list of channels.
FFFF F700 413C	2C4	--	Pad for alignment
Addresses F7004400 through F700553B are reserved.			
Last valid address in this area: FFFF F700 553B			

WLAN Enable—Read/Write

PAC-S
PAC-R
EB

Wired+Wireless models only.

Use this section to enable wireless LAN communications. Also see "WLAN Configuration—Read/Write" on page 192 and "WLAN Status—Read Only" on page 191.

Starting Address	Length (hex)	Type	Description
FFFF F800 0000	4	B	Wireless LAN communication. 0=Disable, 1=Enable. Default is disabled.
FFFF F800 0004	4	UI	WLAN logging to /sdcard0/ath0.log (requires microSD card). 0=Disabled; 1=Log errors and warnings; 2=Log debug information
FFFF F800 0008	4	UI	Inactivity timeout for received Ethernet packets on WLAN interface (in seconds). 0=disabled. Provides recovery for an invalid association. If this timer expires, unit will disassociate, re-scan, and associate.
Last valid address in this area: FFFF F800 000B			

IP Settings—Read/Write

PAC-S
PAC-R
EB
G4EB2

If you are not using PAC Manager to change IP addressing for the Ethernet network interface(s) on the device, you can use this area of the memory map to do so. **All changes made in this area require two steps** (PAC Manager does these steps automatically):

1. Send the new address (8 bytes consisting of the address [4 bytes] plus the 1s complement of it [4 bytes]). The 1s complement requirement is designed to avoid accidental changes.
2. Send a reset hardware command (see Status Area Write address F0380000, operation code 0x00000005) to cycle power and make the changes take effect.

Starting Address	Length (hex)	Type	Description
FFFF FFFF F008	8	IP	IP address for interface 0 (primary IP address)
FFFF FFFF F010	8	IP	Subnet mask for interface 0
FFFF FFFF F018	8	IP	Primary default gateway address for interface 0
FFFF FFFF F020	8	IP	Primary DNS server address for interface 0
FFFF FFFF F028	40	--	Reserved
FFFF FFFF F050	8	IP	Secondary IP address (PAC-S & PAC-R only)
FFFF FFFF F058	8	IP	Secondary IP subnet mask (PAC-S & PAC-R only)
FFFF FFFF F060	8	IP	(Read only) Secondary MAC address (PAC-S & PAC-R only)
FFFF FFFF F068	8	IP	Secondary IP default gateway address (PAC-S & PAC-R only)
FFFF FFFF F070	8	IP	Secondary IP DNS server address (PAC-S & PAC-R only)
Last valid address in this area: FFFF FFFF F077			

Index

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W