

Modbus Integration Kit for PAC Control

Introduction

The Modbus Integration Kit for PAC Control (part number PAC-INT-MB) allows Opto 22 hardware controllers using PAC Control to communicate using the Modbus Serial RTU, Modbus Serial ASCII, or Modbus/TCP protocol. It also allows a SoftPAC software-based controller to communicate using Modbus/TCP.

This kit replaces the Modbus/TCP Integration Kit for PAC Control (part # PAC-INT-MBTCP) and the Modbus Serial Integration Kit for PAC Control (part # PAC-INT-MBSER).

The Integration Kit is a zip file containing two zip files. Choose the one you need:

- Master to use the SNAP PAC controller as a Modbus Master
- Slave to use the PAC as a Modbus Slave

Each subkit contains a sample PAC Control strategy and a series of subroutines you can add to your own PAC Control strategy to enable Modbus communication. Complete documentation is included within the strategy and the subroutines for convenience.

NOTE: If you are using an older version of this integration kit that does not contain documentation within the strategy and subroutines and you do not want to change kits, see form 2009 for documentation of the older kit.

System Requirements

- A PC running PAC Control (Basic or Pro) 9.3 or newer and the Modbus Integration Kit
- SNAP PAC controller with firmware version 9.3 or newer, or SoftPAC software-based controller version 9.3 or newer

The Integration Kit and the Modbus Protocol

Master and slave subroutines transmit message strings as specified in the *Modbus Application Protocol Specification v1.1a*, the *Modbus Messaging on TCP/IP Implementation Guide v1.0a*, and the *Modbus Over Serial Line Specification & Implementation Guide v1.0*. The guides are available on the web at www.modbus.org.

The master and slave subroutines transmit and receive messages using Modbus standard holding and input registers, and input and coil numbers. You store or retrieve the desired information using PAC Control numeric tables.

Master Kit. The master integration kit includes a subroutine for each Modbus command. You read and write information to a slave device using the appropriate subroutine. In general, the subroutine parameters consist of a communication handle, a parameter table that specifies the type of communication, and input or output variables for the data to read or write.

Slave Kit. The slave integration kit includes one subroutine that loops indefinitely while it listens for incoming Modbus requests and sends appropriate responses according to its configuration. The slave subroutine parameters consist of a communication handle, a

parameter table that specifies the type of communication, and a pointer table that points to tables that store Coil, Input, Input Register, and Holding Register data.

Prerequisites

Before you use the Modbus Integration Kit, you should fully understand how to use PAC Control, the Modbus protocol, and the Modbus device you'll be using. Even for those who are experienced with the Modbus protocol, we highly recommend reading the *Modicon Modbus Protocol Reference Guide*, which is available at this link:

http://www.modbus.org/docs/PI_MBUS_300.pdf

We especially recommend the following two sections of the Modbus guide:

- *Chapter 2: Data and Control Functions* of the Modicon guide explains the key subtleties of register, coil, and input numbering/naming as opposed to register, coil, and input addressing. This helps eliminate a common point of confusion, even for those who are experienced with the Modbus protocol.
- *Appendix A: Exception Responses* of the Modicon guide discusses the possible exception codes a Modbus device can reply with and what the codes mean. This helps when diagnosing communication problems.

Data Types Supported in the Input and Holding Registers

The following data types are supported for input and holding registers:

- 16-bit unsigned (Modbus standard and default)
- 16-bit signed
- Floating point (Uses two registers)
- 32-bit signed (Uses 2 registers)

NOTE: Most Modbus devices store 32-bit data values in two consecutive 16-bit registers. However, Opto 22 SNAP PAC controllers store 32-bit data values in individual table elements because tables support full 32-bit data values.

When accessing 32-bit data in most Modbus protocol devices, the data is stored in two consecutive 16-bit registers. Data that is 16 bits is sometimes referred to as a *word*, just as 8-bit data is referred to as a *byte*.

Modbus protocol messages treat each 16-bit register, or word, as 2 bytes with the high-order byte coming before the low-order byte. When device manufacturers started supporting 32-bit data in Modbus messages, there was no standard regarding the order of 16-bit registers (words) in the message for 32-bit data. So some Modbus devices put the bytes of the high-order register (word) first, and the low-order register (word) second. Other devices do the opposite.

In order to provide flexibility when communicating with both types of devices, the Modbus Integration Kit includes a Word Order configuration parameter, which is passed to each subroutine. If the data you see is not correct, it may be because the word order is backwards

compared to your Modbus device. If so, you can change the word order by changing the value of element 3 in the configuration parameter table:

- To have the low-order word in the low-numbered register, use a word order parameter value of 0 (ntParameters[3] = 0).
- To have the high-order word in the low-numbered register, use a word order parameter value of 1 (ntParameters[3] = 1).

For Help

Complete documentation for the integration kits is within the sample strategies and subroutines. If you cannot find the information you need there or in the Modbus protocol documents, contact Opto 22 Product Support. Product Support is free.

Phone:	800-TEK-OPTO (800-835-6786) 951-695-3080 (Hours are Monday through Friday, 7 a.m. to 5 p.m. Pacific Time)	<i>NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.</i>
Fax:	951-695-3017	
Email:	support@opto22.com	
Opto 22 website:	www.opto22.com	