

MODBUS/TCP INTEGRATION KIT FOR IOPROJECT

Form 1644-080204—February 2008

OPTO 22

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

**Modbus/TCP Integration Kit for ioProject
Form 1644-080204—February 2008**

Copyright © 2008 Opto 22.

All rights reserved.

Printed in the United States of America.

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 later 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.

Cyrano, Opto 22 FactoryFloor, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, *mistic*, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDataLink, OptoDisplay, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, PAC Control, PAC Display, PAC Manager, PAC Project, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC System, SNAP Simple I/O, SNAP Ultimate I/O, and SNAP Wireless LAN I/O are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C++, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Unicenter is a registered trademark of Computer Associates International, Inc. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric. Wiegand is a registered trademark of Sensor Engineering Corporation. Nokia, Nokia M2M Platform, Nokia M2M Gateway Software, and Nokia 31 GSM Connectivity Terminal are trademarks or registered trademarks of Nokia Corporation. Sony is a trademark of Sony Corporation. Ericsson is a trademark of Telefonaktiebolaget LM Ericsson.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Table of Contents

Overview.....	1
What is Required.....	1
Modbus/TCP Functions Supported.....	2
Data Types Supported.....	2
Installing the Integration Kit	3
Changing the Modbus Slave TCP Port	3
Running the Example Strategy.....	3
Importing the Slave Strategy	4
Using Communication Handles	4
Adding Master Subroutines	4
Configuration of Subroutines	5
MB 01 Read Coils.....	5
MB 02 Read Discrete Inputs	6
MB 03 Read Holding Registers	7
MB 04 Read Input Registers	8
MB 05 Write Single Coil	9
MB 06 Write Single Register	10
MB 15 Write Multiple Coils	11
MB 16 Write Multiple Registers.....	12



Modbus/TCP Integration Kit for ioProject

Overview

The Modbus/TCP Integration Kit for ioProject (Part # IOP-INT-MBTCP) allows Opto 22 controllers, using ioControl, to connect via Ethernet to any Ethernet/TCP-Modbus device and communicate using the Modbus/TCP protocol.

The Integration Kit contains:

- a set of ioControl *master* subroutines that are added to a strategy to enable an Opto 22 controller to communicate as a Modbus/TCP master
- an example Modbus/TCP *slave* strategy containing the slave chart MBSlaveTCP that is imported into a strategy to enable an Opto 22 controller to communicate as a Modbus/TCP slave

Both the master subroutines and the slave strategy transmit message strings as specified in the *Modbus Application Protocol Specification v1.1a* and *Modbus Messaging on TCP/IP Implementation Guide v1.0a*. Both guides are available on the web at <http://Modbus-IDA.org>.

The master subroutines and slave strategy transmit and receive messages using Modbus standard register, input and coil numbers. The desired information is stored or retrieved using ioControl numeric tables.

This manual assumes the user fully understands how to use ioControl, Modbus/TCP, and the Modbus device to be used.

What is Required

Before including the subroutines in your strategy, you will need:

- a PC running ioControl software and the Modbus/TCP Integration Kit for ioProject .
- ioProject Basic 7.0 or later

Modbus/TCP Functions Supported

The following Modbus/TCP function codes are supported by ioControl subroutines:

Modbus/TCP Function Code	Name	PAC Control Subroutine
01	Read Coils	MB_01_Read_Coils
02	Read Discrete Inputs	MB_02_Read_Discrete_Inputs
03	Read Holding Registers	MB_03_Read_Holding_Registers
04	Read Input Registers	MB_04_Read_Input_Registers
05	Write Single Coil	MB_05_Write_Single_Coil
06	Write Single Register	MB_06_Write_Single_Register
15	Write Multiple Coils	MB_15_Write_Multiple_Coils
16	Write Multiple Registers	MB_16_Write_Multiple_Registers
23	Read Write Multiple Registers	MB_23_Read_Write_Holding_Registers

Data Types Supported

The following data types are supported:

Index	Name
0	16-bit unsigned (Modbus standard and default)
1	16-bit signed
2	Floating point (Uses two registers)
3	Floating point (Swapped. Uses two registers.)
4	32-bit signed (Uses 2 registers)
5	32-bit signed (Swapped. Uses 2 registers.)

Data types 2-5 use two registers to transfer data. In the Holding or Input registers, place data in the odd indexes.

The even registers are used internally.

Installing the Integration Kit

To install the integration kit on your computer, unzip the ModbusTCP.zip file to your C: drive. The expanded files will be placed automatically in C:\ModbusTCP.

Changing the Modbus Slave TCP Port

An Opto 22 SNAP PAC controller has a built-in Modbus/TCP slave capability, which only provides access to the I/O portions of the memory map (see Note below). The slave toolkit, on the other hand, provides access to both the strategy, portions of the memory map, and the I/O. Therefore, in order for the slave toolkit to work, you must disable the built-in Modbus slave functionality by changing the Modbus slave TCP port to a value of 0 (zero) as follows.

NOTE: I/O access only applies to rack-mounted controllers, such as the SNAP-PAC-R1 or R2

1. Open ioManager.
2. In the ioManager main window, click the Inspect button.
3. In the Device Name field, type the IP address for the SNAP PAC controller (or choose it from the drop-down list).
4. Click Communications and choose Network Security.
5. Under PORTS, click the Value field for Modbus.
6. Change the value to 0 (zero). Click Apply.
7. Click Status Write.
8. Under Operation Commands, choose Send configuration to flash. Click Send Command.
9. Under Operation Commands again, choose Restart Device from powerup. Click Send Command.

For more details on using ioManager, see the *ioManager User's Guide*, form 1440.

Running the Example Strategy

The kit includes an example strategy to demonstrate how to use the subroutines in an ioControl strategy. Before importing the subroutines into your own strategy, it is recommended that you first run the example strategy to see how to use the subroutines in an ioControl strategy, especially the strategy logic and the configuration of variables.

To run the example master strategy, start ioControl, and then open the strategy file, MBMasterTCP.idb.

Importing the Slave Strategy

The Modbus/TCP slave strategy allows an Opto 22 controller to function as a Modbus/TCP slave device. Unlike the subroutines used in master strategies, which are run as needed, the MBSlaveTCP chart is started in the Powerup chart and must run all the time. After the chart is started it continuously monitors port 502 for Modbus traffic.

To copy the Modbus Slave chart to your strategy, you must export the chart MBSlaveTCP as an ioControl chart export file (.cxf file) and then import it into your strategy. For more information, see Chapter 8 of the [ioControl User's Guide](#).

Start the Modbus Slave chart in the Powerup chart of your strategy.

Using Communication Handles

Be sure to use a separate TCP communication handle for each chart that uses the Modbus/TCP subroutines.

In ioControl, if two charts were to run simultaneously while sharing an open communication handle, each chart would be able to read and write data from the communication handle as if the other running chart didn't exist. Because these reads and writes are not synchronized between the charts, it is possible for one chart to read the other chart's data.

Adding Master Subroutines

The Modbus master subroutines allow an Opto 22 controller to function as a Modbus/TCP master device. Each master subroutine in the integration kit supports one Modbus/TCP function code and can function independently of the other subroutines. Therefore, you need only use the subroutines for the Modbus/TCP functions that you require. For more information about subroutines, see the [ioControl User's Guide](#).

When you decide which subroutines you need, include them in your strategy as follows:

1. Start ioControl in Configure Mode and open the strategy that you intend to use with the integration kit.
2. Select Configure→Subroutines Included to open the Subroutine Files dialog.
3. Click the Add button and use the browser to select each subroutine file (.ISB extension) you wish to include in your strategy from the folder C:\MbusTCP\Subs.
4. Click OK.

The subroutines appear in the Subroutines Included folder and are ready to be used in your strategy.

Configuration of Subroutines

The following tables list the parameters for each function code and describe the type of data for each parameter:

MB 01 Read Coils	page 5
MB 02 Read Discrete Inputs	page 6
MB 03 Read Holding Registers	page 7
MB 04 Read Input Registers	page 8
MB 05 Write Single Coil	page 9
MB 06 Write Single Register	page 10
MB 15 Write Multiple Coils	page 11
MB 16 Write Multiple Registers	page 12

MB 01 Read Coils

Name	Description
Slave Address	Integer 32 Variable (1–255)
Starting Address	Integer 32 Variable (1–65536)
Quantity of Coils	Integer 32 Variable (1–2000)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	Wait time in seconds for slave to respond)
MB Coils 0X	Integer 32 Table (The subroutine will support coils 1–65535)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 02 Read Discrete Inputs

Name	Description
Slave Address	Integer 32 Variable (1–255)
Starting Address	Integer 32 Variable (1–65536)
Quantity of Inputs	Integer 32 Variable (1–2000)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
MB Inputs 1X	Integer 32 Table (The subroutine will support Inputs 1–65535)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 03 Read Holding Registers

Name	Description
Slave Address	Integer 32 Variable (1–255)
Starting Address	Integer 32 Variable (1–65536)
Qty of H Registers	Integer 32 Variable (1–125)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
MB H Register 4X	Float Table (The subroutine will support registers 1–65535)
Data Type	Integer 32 Variable 0 = 16-bit unsigned (Modbus standard and default) 1 = 16-bit signed 2 = Floating Pt. 3 = Floating Pt. (swapped) 4 = 32-bit signed 5 = 32-bit signed (swapped)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 04 Read Input Registers

Name	Description
Slave Address	Integer 32 Variable (1–255)
Starting Address	Integer 32 Variable (1–65536)
Qty of I Registers	Integer 32 Variable (1–125)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
MB I Register 3X	Float Table (The subroutine will support registers 1–65535)
Data Type	Integer 32 Variable 0 = 16-bit unsigned (Modbus standard and default) 1 = 16-bit signed 2 = Floating Pt. 3 = Floating Pt. (swapped) 4 = 32-bit signed 5 = 32-bit signed (swapped)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 05 Write Single Coil

Name	Description
Slave Address	Integer 32 Variable (1–255)
Output Address	Integer 32 Variable (1–65536)
Output State	Integer 32 Variable (0 = OFF 1 = ON)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 06 Write Single Register

Name	Description
Slave Address	Integer 32 Variable (1–255)
Register Address	Integer 32 Variable (1–65536)
Register Value	Float Variable
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
Data Type	Integer 32 Variable 0 = 16-bit unsigned (Modbus standard and default) 1 = 16-bit signed 2 = Floating Pt. 3 = Floating Pt. (swapped) 4 = 32-bit signed 5 = 32-bit signed (swapped)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 15 Write Multiple Coils

Name	Description
Slave Address	Integer 32 Variable (1–255)
Starting Address	Integer 32 Variable (1–65536)
Quantity of Coils	Integer 32 Variable (1–1968)
Output Value	Integer 32 Table (The subroutine will support coils 1–65535)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error

MB 16 Write Multiple Registers

Name	Description
Slave Address	Integer 32 Variable (1–255)
Starting Address	Integer 32 Variable (1–65536)
Qty of Registers	Integer 32 Variable (1–120)
Register Value	Float Table (The subroutine will support registers 1–65535)
Identifier	Integer 32 Variable (Used for transaction pairing)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
Data Type	Integer 32 Variable 0 = 16-bit unsigned (Modbus standard and default) 1 = 16-bit signed 2 = Floating Pt. 3 = Floating Pt. (swapped) 4 = 32-bit signed 5 = 32-bit signed (swapped)
Return Status	String No Session = Not able to open session Timeout = No response within time limit Too Many Characters = More then 260 characters Identifier Mismatch = Send and receive identifier do not match Function and Exception code = Error from PDU OK = Success
Put Status In	Integer 32 Variable 0 = Success -67 = Out of memory -69 = Null object error