

MODBUS/SERIAL INTEGRATION KIT FOR IOPROJECT

Form 1660-080204—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 1660-080204—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
Protocol Supported	2
Modbus/Serial Functions Supported	2
Data Types Supported	2
Installing the Integration Kit	3
Running the Example Strategy	3
Importing the Slave Strategy	3
Adding Master Subroutines	3
Configuration of Subroutines	4
MBMaster Read Coil Status	5
MBMaster Read Input Status	6
MBMaster Read Holding Registers	7
MBMaster Read Input Registers	8
MBMaster Force Single Coil	9
MBMaster Preset Single Register	10
MBMaster Force Multiple Coils	11
MBMaster Preset Multiple Registers	12



Modbus/Serial Integration Kit for ioProject

OPTO 22

Overview

The Modbus/Serial Integration Kit for ioProject (Part # IOP-INT-MBSER) allows Opto 22 controllers, using ioControl, to connect via RS-232 or RS-485 to a Modbus device and communicate using the Modbus RTU or ASCII 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 master
- an example Modbus slave strategy containing the slave chart MB_Slave_Serial that is imported into a strategy to enable an Opto 22 controller to communicate as a Modbus slave

Both the master subroutines and the slave strategy transmit message strings as specified in the *Modbus Application Protocol Specification v1.1a* and *Modbus over Serial Line Specification & Implementation guide v1.0*. 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/Serial, and the Modbus device to be used.

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.

What is Required

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

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

Protocol Supported

The following protocols are supported:

Index	Name
0	RTU protocol
1	ASCII protocol

Modbus/Serial Functions Supported

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

Modbus/Serial Function Code	Name	ioControl Subroutine
01	Read Coils	MBMaster Read Coil Status
02	Read Discrete Inputs	MBMaster Read Input Status
03	Read Holding Registers	MBMaster Read Holding Registers
04	Read Input Registers	MBMaster Read Input Registers
05	Write Single Coil	MBMaster Force Single Coil
06	Write Single Register	MBMaster Preset Single Register
15	Write Multiple Coils	MBMaster Force Multiple Coils
16	Write Multiple Registers	MBMaster Preset Multiple Registers
23	Read/Write Multiple Registers	MBMaster 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 ModbusSerial.zip file to your C: drive. The expanded files will be placed automatically in C:\ModbusSerial.

Running the Example Strategy

The kit includes an example strategy to demonstrate how to use the subroutines in an ioControl strategy. Before including the subroutines into your own strategy, we recommend 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, MBMasterSerial.idb.

Importing the Slave Strategy

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

To copy the Modbus Slave chart to your strategy, you must export the chart MB_Slave_Serial 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.

Adding Master Subroutines

The Modbus master subroutines allow an Opto 22 controller to function as a Modbus master device. Each master subroutine in the integration kit supports one Modbus function code and can function independently of the other subroutines. Therefore, you need only use the subroutines for the Modbus 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:\ModbusSerial\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.

Starting Address is from 1 to 65536, while in Modbus, addresses start at 0. In order to correct for the offset between tables, the subroutines reduce the address by 1 in the packet sent to the Modbus slave.

Modbus Register 1 = Opto 22 Index 1

Modbus Register 4000 = Opto 22 Index 4000

MBMaster Read Coil Status	page 5
MBMaster Read Input Status	page 6
MBMaster Read Holding Registers	page 7
MBMaster Read Input Registers	page 8
MBMaster Force Single Coil	page 9
MBMaster Preset Single Register	page 10
MBMaster Force Multiple Coils	page 11
MBMaster Preset Multiple Registers	page 12

MBMaster Read Coil Status

Name	Description
Slave Address	Integer 32 Variable (1 – 255)
Starting Address	Integer 32 Variable (1 – 65536)
Quantity of Coils	Integer 32 Variable (1 – 2000)
Comm Mode	Integer 32 Variable (0 = RTU protocol, 1 = ASCII protocol)
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 Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster Read Input Status

Name	Description
Slave Address	Integer 32 Variable (1 – 255)
Starting Address	Integer 32 Variable (1 – 65536)
Quantity of Inputs	Integer 32 Variable (1 – 2000)
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
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 Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster 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)
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
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 Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster 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)
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
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 Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster Force 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)
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
Return Status	String No Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster Preset Single Register

Name	Description
Slave Address	Integer 32 Variable (1 – 255)
Register Address	Integer 32 Variable (1 – 65536)
Register Value	Float Variable
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
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
Return Status	String No Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster Force 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)
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
Com Handle	Communication Handle
Wait Time(s)	(Wait time in seconds for slave to respond)
Return Status	String No Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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

MBMaster Preset 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)
Comm Mode	Integer 32 Variable (0 = RTU protocol 1 = ASCII protocol)
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 Port = Not able to open Port Timeout = No response within time limit Too Many Characters = More then 260 characters Address Mismatch = Send and receive address do not match CRC or LRC Mismatch = Checksum error 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