MODBUS 3.1 INTEGRATION KIT FOR OPTOCONTROL

Form 1128-000828 — August, 2000

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
E-mail: support@opto22.com
Web: support.opto22.com

Modbus 3.1 Integration Kit For OptoControl
Form 1128-000828 — August, 2000

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, the 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.

Table of Contents

Modbus Integration Kit 3.1 For OptoControl

Overview

Introduction

The Modbus Integration Kit (Part # OPTOINTMB) allows Opto 22 controllers, using OptoControl, to connect via serial port to any Modbus device and communicate using the Modbus protocol.

The kit contains a set of OptoControl subroutines that transmit message strings as specified in the *Modicon Modbus Protocol Reference Guide* (part number PI-MBUS-300; available from Modicon/AEG Schneider Automation). These subroutines will transmit and receive messages using Modbus standard register, input and coil numbers. The subroutines use a set of extended OptoControl commands to support the Modbus protocol. The desired information is stored or retrieved using OptoControl numeric tables.

This manual assumes the user fully understands OptoControl as well as the Modbus device to be used.

Functions Supported

The following function codes have been implemented in support of Modbus functionality:

Function Code	Description
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
15	Force Multiple Coils
16	Preset Multiple Registers

Getting Started

What is Required

To implement an interface between an Opto 22 controller and a Modbus device, you will need:

- A PC running OptoControl software and the Modbus OptoIntegration Kit.
- An Opto 22 controller with a firmware revision of OptoKernel R2.2a or later.
- A serial cable connecting the Opto 22 controller to the Modbus device.

OptoControl Configuration

The integration kit is a library of OptoControl subroutine files. These subroutines utilize an OptoControl extended library file (ModbusR3.1a.LIB) that is downloaded to the Opto 22 controller before your OptoControl strategy. In addition, there is an OptoControl definition file (ModbusR3.1a.XID) that makes the functions available in the OptoControl Configurator. **All files associated with the Modbus 3.1 Integration Kit are located at C:\Opto22\OptoCtrl\ Modbus3.1**

After completing the installation, follow these steps for configuring your OptoControl strategy.

1. Start OptoControl in Configure Mode and open the strategy that you intend to use with the integration kit.
2. Use the Configure tool on the menu to select Controllers. Then select Download Options, and click the Add button on the Before Run File list. Use the browser to select ModbusR3.1a.LIB from the folder C:\Opto22\OptoCtrl\ Modbus3.1\.
3. Use the Configure tool on the menu to select Subroutine Includes. Click the Add button and use the browser to select each subroutine file (.CSB extension) you wish to include into your strategy from the folder C:\Opto22\OptoCtrl\ Modbus3.1\. When finished, the subroutines should appear in the Strategy Tree under the Subroutines Included folder.
4. For each subroutine you include in the OptoControl Strategy Tree click on the "+" sign next to the submarine icon and doubl- click the folder titled "Ext. Instruction Files Included" (see note 1 below). Click the Add button and use the browser to select ModbusR3.1a.XID from the folder C:\Opto22\OptoCtrl\ Modbus3.1\.

Note 1: *If you are opening a subroutine and you see errors indicating OptoControl cannot find the external instruction file for the subroutine, the references to the .XID file are invalid. Once the .XID files are added to the Ext. Instruction Files Included list (step 4), subsequent openings of the subroutine should be error-free.*

5. The subroutines are now ready to be utilized in your strategy.

Note 2: *It is recommended that you run the example strategies first. Review the logic and the configuration of variables.*

Modbus Coil, Input and Register to OptoControl Table Correlation

Modbus to OptoControl Coil and Input Correlation

Modbus	OptoControl Table	
Coil or Input #	Index	Bit
1	0	0
2	0	1
3	0	2
4	0	3
5	0	4
6	0	5
7	0	6
8	0	7
9	0	8
10	0	9
11	0	10
12	0	11
13	0	12
14	0	13
15	0	14
16	0	15
17	1	0
18	1	1
19	1	2
20	1	3
21	1	4
22	1	5
23	1	6
24	1	7
25	1	8
26	1	9
27	1	10
28	1	11
29	1	12
30	1	13
31	1	14
32	1	15
33	2	0
---	---	---
2,000 *	124	12

Modbus to OptoControl Register Correlation

Modbus Reg #	OptoControl Table Index
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19
---	---
125 *	124

For coils, inputs and 16-bit register operations, bits 16 - 31 are unused in the OptoControl table.

***These tables show the correlation between Modbus data and OptoControl tables. These are examples, and not maximum limits.**

Configuration of Modbus Subroutines

MBMaster Configure Modbus Port:

Configures Modbus serial port for use with Modbus ‘Master ’ subroutines.

PORT CFG STR

Set the serial port configuration by moving a configuration string to a string variable. Enter the variable name in the field next to the prompt PORT CFG STR when calling the subroutine. See OptoControl command “Configure Port” for configuration string options.

TA DELAY (Required for both RTU and ASCII modes)

Create a float variable. Enter the variable name in the field next to the prompt TA DELAY to set the serial port turn around delay value in seconds. (For example 2.0 = 2 seconds = 2,000 msec). This is the amount of time the subroutine will wait for a response from the slave device.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished.

REG DATA TYPE

Sets how the Modbus ‘Master’ subroutines will process input and holding registers. Use the table below to determine the appropriate value to set for your particular data type. Set the register data type by moving an appropriate value to an integer variable. Use this variable to set the REG DATA TYPE when pcalling the subroutine. **Register Data Type 3 (Daniel Option) affects holding registers only.**

The register values supported by this integration kit are:

REG DATA TYPE = 1 *	16-bit unsigned integer	Range = 0 to 65,535
REG DATA TYPE = 2	16-bit signed integer	Range = -32,768 to 32,767
REG DATA TYPE = 3	IEEE float	Range = +/-3.40284 * 10^{-38} to +/-3.402824 * 10^{38}
REG DATA TYPE = 4	32-bit signed integer	Range = -2,147,483,648 to 2,147,483,647

*16-bit unsigned integer is the default for Modicon PLCs.

COMM MODE

Integer variable used to set the Modbus ‘Master’ subroutines to use the ASCII protocol (-1) or the RTU protocol (0).

RTS TX DELAY

Create a float variable. Enter the variable name in the field next to the prompt RTS TX DELAY. This variable sets the delay time in seconds that the subroutine will wait after turning on RTS to transmit the message. For example 0.01 = 10 milliseconds. Set to zero unless using half-duplex, non-buffered, unintelligent radio modems.

MBMaster Read Coil Status: (01)

Copies one or more coil values from a slave device to an OptoControl table. (Function code 01)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Index 0 of the pointer table holds a pointer to the integer table that will be used to store coil information. Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE COIL NUM

Set the slave device starting coil number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE COIL NUM when calling the subroutine. See section about Modbus to OptoControl table correlation.

ADDRESS RANGE

Set the number of coils to be read by moving a value to an integer variable. Enter the variable name in the field next to the prompt ADDRESS RANGE when calling the subroutine.

MASTER COIL NU

Create an integer variable to indicate the starting coil position in the OptoControl table where the coil information is to be stored. Enter the variable name in the field next to the prompt MASTER COIL NU when calling the subroutine. See section about Modbus to OptoControl table correlation.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the "ERROR CODE" parameter.**

- A maximum of 2000 coils can be read at one time with a single command.

MBMaster Read Input Status: (02)

Copies one or more digital input values from a slave device to an OptoControl table. (Function code 02)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Index 0 of the pointer table holds a pointer to the integer table that will be used to store input information. Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE INPUT NU

Set the slave device starting input number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE INPUT N when calling the subroutine. See section about Modbus to OptoControl table correlation.

ADDRESS RANGE

Set the number of inputs to be read by moving a value to an integer variable. Enter the variable name in the field next to the prompt ADDRESS RANGE when calling the subroutine.

MASTER INPUT N

Create an integer variable to indicate the starting input position in the OptoControl table where the input information is to be stored. Enter the variable name in the field next to the prompt MASTER INPUT N when calling the subroutine. See section about Modbus to OptoControl table correlation.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the "ERROR CODE" parameter.**

- A maximum of 2000 inputs can be read at one time with a single command.

MBMaster Read Holding Registers: (03)

Copies one or more holding register values from a slave device to an OptoControl table. (Function code 03)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Create a pointer table variable of length 3. Index 0 of the pointer table holds a pointer to the integer table that will be used to store register information. If the register data type is equal to 3 then index 1of the pointer table holds a pointer to the float table that will be used to store register information. Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE REG NUM

Set the slave device starting register number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE REG NUM when calling the subroutine. See section about Modbus to OptoControl table correlation.

ADDRESS RANGE

Set the number of registers to be read by moving a value to an integer variable. Enter the variable name in the field next to the prompt ADDRESS RANGE when calling the subroutine.

MASTER REG NU

Create an integer variable to indicate the starting register position in the OptoControl table where the register information is to be stored. Enter the variable name in the field next to the prompt MASTER REG NU when calling the subroutine. See section about Modbus to OptoControl table correlation.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the "ERROR CODE" parameter.**

- A maximum of 125 16-bit registers can be read at one time with a single command.
- A maximum of 62 32-bit registers can be read at one time with a single command.

MBMaster Read Input Registers: (04)

Copies one or more input register values from a slave device to an OptoControl table. (Function code 04)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Create a pointer table variable of length 3. Index 0 of the pointer table holds a pointer to the integer table that will be used to store register information. Index 1 of the pointer table is unused. Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE REG NUM

Set the slave device starting register number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE REG NUM when calling the subroutine. See section about Modbus to OptoControl table correlation.

ADDRESS RANGE

Set the number of registers to be read by moving a value to an integer variable. Enter the variable name in the field next to the prompt ADDRESS RANGE when calling the subroutine.

MASTER REG NU

Create an integer variable to indicate the starting register position in the OptoControl table where the register information is to be stored. Enter the variable name in the field next to the prompt MASTER REG NU when calling the subroutine. See section about Modbus to OptoControl table correlation.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the “ERROR CODE” parameter.**

- A maximum of 125 16-bit registers can be read at one time with a single command.

MBMaster Force Single Coil: (05)

Forces a coil in a slave device to either ON or OFF. (Function code 05)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE COIL NUM

Set the slave device coil number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE COIL NUM when calling the subroutine. See section about Modbus to OptoControl table correlation.

FORCE DATA

Set the value to be forced (Decimal 65280 = ON, Decimal 0 = OFF) by moving a value to an integer variable. Enter the variable name in the field next to the prompt COIL VALUE when calling the subroutine.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the "ERROR CODE" parameter**

MBMaster Preset Single Register: (06)

Sends a register value to a slave from an OptoControl variable. (Function code 06)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

REG NUMBER

Set the slave device register number by moving a value to an integer variable. Enter the variable name in the field next to the prompt REG NUMBER when calling the subroutine. See section about Modbus to OptoControl table correlation.

PRESET DATA

Set the value to be written by moving a value to an integer variable. Enter the variable name in the field next to the prompt PRESET DATA when calling the subroutine. (This command cannot be used to write float data. Use function code 16)

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the “ERROR CODE” parameter.**

MBMaster Force Multiple Coils: (15)

Forces one or more coils in a slave device to either ON or OFF based on bits in an OptoControl table. (Function code 15)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Index 0 of the pointer table holds a pointer to the integer table that will be used to store coil information. Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE COIL NUM

Set the slave device starting coil number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE COIL NUM when calling the subroutine. See section about Modbus to OptoControl table correlation.

ADDRESS RANGE

Set the number of coils to be written by moving a value to an integer variable. Enter the variable name in the field next to the prompt ADDRESS RANGE when calling the subroutine.

MASTER COIL NU

Create an integer variable to indicate the starting coil position in the OptoControl table where the coil information, to be written to the slave, is stored. Enter the variable name in the field next to the prompt MASTER COIL NU when calling the subroutine. See section about Modbus to OptoControl table correlation.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the “ERROR CODE” parameter.**

- A maximum of 800 coils can be written at one time with a single command.

MBMaster Preset Multiple Registers: (16)

Sends one or more register values to a slave device from an OptoControl table. (Function code 16)

SLAVE ADDRESS

Set the destination slave device address by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

POINTER TABLE

Create a pointer table variable of length 3. Index 0 of the pointer table holds a pointer to the integer table that will be used to store the register information (to be written to the slave). If the register data type is equal to 3 then index 1of the pointer table holds a pointer to the float table that will be used to store register information. Index 2 holds a pointer to the string table which will hold the transmit and receive strings (for diagnostic purposes). Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE REG NUM

Set the slave device starting register number by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE REG NUM when calling the subroutine. See section about Modbus to OptoControl table correlation.

ADDRESS RANGE

Set the number of registers to be written by moving a value to an integer variable. Enter the variable name in the field next to the prompt ADDRESS RANGE when calling the subroutine.

MASTER REG NU

Create an integer variable to indicate the starting register position in the OptoControl table where the register information (to be written) is to be stored. Enter the variable name in the field next to the prompt MASTER REG NU when calling the subroutine. See section about Modbus to OptoControl table correlation.

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished. **The exception codes returned by Modbus slave devices are returned in the "ERROR CODE" parameter.**

- A maximum of 100 16-bit registers can be written at one time with a single command.
- A maximum of 50 32-bit registers can be written at one time with a single command.

Modbus Slave

The Modbus Slave subroutine allows the Opto 22 controller to acts as a Modbus "slave" device. It runs continuously in the chart that started it. It will only stop or return an error if it has a problem configuring or acquiring the serial port. After it is started it continuously monitors the associated serial port for Modbus traffic.

PORT CFG STR

Set the serial port configuration by moving a configuration string to a string variable. Enter the variable name in the field next to the prompt PORT CFG STR when calling the subroutine. See OptoControl command "Configure Port" for configuration string options.

REG DATA TYPE

Sets how the Modbus Slave subroutine will process input and holding registers. Use the table below to determine the appropriate value to set for your particular data type. Set the register data type to use by moving an appropriate value to an integer variable. Use this variable to set the REG DATA TYPE when calling the subroutine. **Register Data Type 3 (Daniel Option) affects holding registers only.**

The register values supported by this integration kit are:

REG DATA TYPE = 1 *	16-bit unsigned integer	Range = 0 to 65,535
REG DATA TYPE = 2	16-bit signed integer	Range = -32,768 to 32,767
REG DATA TYPE = 3	IEEE float	Range = +/-3.40284 * 10^{-38} to +/-3.402824 * 10^{38}
REG DATA TYPE = 4	32-bit signed integer	Range = -2,147,483,648 to 2,147,483,647

* 16-bit unsigned integer is the default for Modicon PLCs.

POINTER TABLE

Create a pointer table variable of length 6. The pointer table should be initialized as follows:

Index 0	Pointer to the integer table that will be used to store coil information
Index 1	Pointer to the integer table that will be used to store input status information
Index 2	Pointer to the string table which will hold the first 127 characters of the transmit and receive strings. The string table should be configured as follows: Length = 2, Width = 127. (These strings are preserved for diagnostic purposes.)
Index 3	Pointer to the integer table that will be used to store input register information
Index 4	Pointer to the integer table that will be used to store holding register information
Index 5	Pointer to the float table that will be used to store holding register information

All elements of the pointer table must be initialized to ensure trouble-free operation. If you are not using some specific tables, configure the length parameter for that table to a value of 1 (one) so that the memory requirements are minimized.

Enter the pointer table name in the field next to the prompt POINTER TABLE when calling the subroutine.

SLAVE ADDRESS

Set the slave device address (the slave should respond to) by moving a value to an integer variable. Enter the variable name in the field next to the prompt SLAVE ADDRESS when calling the subroutine.

COMM MODE

Integer variable used to set the Modbus 'Master' subroutines to use the ASCII protocol (-1) or the RTU protocol (0).

ERROR CODE

Create an integer variable. Enter the variable name in the field next to the prompt ERROR CODE. This variable holds the error code when the subroutine is finished.

RTS TX DELAY

Enter the delay time (in seconds) in the field next to the prompt RTS TX DELAY. For example 0.01 = 10 milliseconds. This is the amount of time the subroutine will wait after turning on RTS to transmit the message. Set to zero unless using half-duplex, non-buffered, unintelligent radio modems.

Error Codes

Errors Reported Via Error Code Parameter

Error Code	Description
0	Good response
-1	Timeout Error
-2	Wrong number of characters (odd length)
-3	Invalid character (not ASCII HEX)
-4	Message too short
-5	Message too long (buffer overflow)
-6	LRC or CRC bad
-7	Destination address global for read function
-8	Too many or too few elements specified in address range
-9	Could not acquire port
-40	Could not acquire port
-50	Improper port configuration string syntax

The exception codes sent from Modbus slave devices are also returned in the "ERROR CODE" parameter

Exception Codes For the Modbus "Slave" Subroutine

These codes are returned via serial port to the "Master" polling device.

Exception Code	Description
1	Illegal Function Received
2	Illegal Data Address
3	Illegal Data Value