

**SIMPLWINDOWS
NAME:**

Modbus Discrete Output Poll and Control

CATEGORY:

HVAC

VERSION:

1.0

SUMMARY:

Polls and controls a single Modbus discrete output point

GENERAL NOTES:

Modbus is a generic communications protocol. It allows a group of addressable "points" to be accessed by the Crestron system. There are digital points, which have two states - on and off. There are also analog points, which allow 16 bit numbers to be accessed. Some points are read only, while others are read/write.

Modbus supports two different command formats - RTU and ASCII. This module uses the RTU format. You must be sure that the Modbus device is also set for RTU mode.

Modbus does not specify a communications medium or format, so for information on baud rate, parity, etc., as well as cabling information, you must consult the documentation that comes with the Modbus system. The Modbus device that was tested at Crestron was an Automated Logic Portal, and all connection information below is for that device only.

Each device on a Modbus system is uniquely addressed using it's slave address, and it's point number. This module has a single parameter field for the slave address (SLAVE-ADDRESS-HEX). This parameter needs to be entered as a 2 digit hex number with no suffix. So for slave 1, enter 01. This module has two parameter fields which specify the point number. By using two bytes (high byte and low byte) it allows a larger addressing range. These parameters also need to be entered as 2 digit hex numbers with no suffix. The point numbers are also offset by one. So if you want to access point number 2, you would enter 00 for the POINT-NUMBER-HIGH-HEX and enter 01 for the POINT-NUMBER-LOW-HEX parameters.

The Modbus system does not do real time updating. Therefore we must poll each point for the current setting. This module has a POLL input for this purpose. The polling sequence should be constructed by using an Oscillator with a high time of .1s and a low time of .9s driving the clock input of a Ring counter. Each output of the ring counter should drive separate POLL inputs on the Crestron Modbus modules. So if there are ten Modbus points that are being monitored, there would be ten Crestron Modbus modules in the program, and the Ring Counter would have ten outputs, so that each point could be polled in turn. See the demo program on the Crestron FTP site for an example of this implementation.

This module must be used in conjunction with the Modbus Processor module. The serial output of this module will be sent to the Modbus processor module for formatting. Also, the serial input to this module must come from the Modbus Processor module.

This module will provide the status and control of a single discrete output point.

This module uses Modbus function code 01, which is designated for reading the contents of Discrete Outputs. It also uses function code 05 for setting discrete outputs.

CRESTRON HARDWARE REQUIRED:	CNMSX, CNXCOM, ST-COM
SETUP OF CRESTRON HARDWARE:	The port should be set as follows: Baud Rate - 9600 Parity - None Data Bits - 8 Stop Bits - 1
VENDOR FIRMWARE:	None
VENDOR SETUP:	The Automated logic system must be in RTU mode. Use the Diagnostic Port connector for communications.
CABLE DIAGRAM:	CNSP-124

CONTROL:

ON	D	Pulse to turn the point on
OFF	D	Pulse to turn the point off
POLL	D	Pulse this input to poll the desired point
FROM-MODBUS- PROCESSOR\$	S	Serial signal routed from the Crestron Modbus Processor module
SLAVE-ADDRESS- HEX	P	Two digit hex representation of the slave address. For slave 1, enter 01
POINT-NUMBER- HIGH-HEX	P	Two digit hex representation of the most significant byte of the 16 bit point address
POINT-NUMBER- LOW-HEX	P	Two digit hex representation of the least significant byte of the 16 bit point address

FEEDBACK:

ON-FB	D	True feedback indicating when the point is in the on state
TO-MODBUS- PROCESOR\$	S	Serial signal routed to the Crestron Modbus Processor module

OPS USED FOR TESTING:	5.12.01x
COMPILER USED FOR TESTING:	SimplWindows Version 1.51.08
SAMPLE PROGRAM:	Modbus Demo Program
REVISION HISTORY:	None