

Modbus Test Utility for PAC Display

Introduction

The Modbus Test Utility uses Modbus Function 3 (Read Holding Registers) to test communication between an Opto 22 SNAP PAC controller and a Modbus device and to help you determine the data type used by your Modbus device. The utility supports Modbus®/TCP and Modbus serial using the controller serial port or the SNAP SCM serial module. It also supports Modbus serial RTU mode or ASCII mode.

The Modbus Test Utility is a PAC Display project, which you set up and run in PAC Display Runtime to communicate with your Modbus device.

This document explains how to use the test utility. It assumes you already understand how to use PAC Display, Modbus, and the Modbus device to be used.

The test utility does the following things:

- Reads two registers at the starting register you enter. Data types 2, 3, 4, and 5 use two registers to transport the data.
- Shows the outgoing and incoming Modbus packet in Hex. It also shows the description of any received error.
- Shows the received data as the following data types:

Type	Description
Raw data	Decimal
Data type 0	16 bit unsigned, decimal
Data type 1	16 bit signed, decimal
Data type 2	Floating point, decimal
Data type 3	Floating point swapped, decimal
Data type 4	32 bit signed, decimal
Data type 5	32 bit signed swapped, decimal

What is Required

You will need a PC with PAC Project 8.1a (Basic or Pro).

Installation

1. Extract the zip file, ModbusTestUtilityv81a.zip, to your hard drive (C:).
2. Open PAC Control, browse to C:\ModbusTestUtility\Strategy, and then open the strategy file, ModbusTestUtility.idb.
3. Select Configure > Control Engines to configure a control engine for the strategy. If you need help with this, see the *PAC Control User's Guide*, form 1700.
4. Download and run the strategy.
5. Open PAC Display, browse to C:\ModbusTestUtility\Display, and then open the project file, ModbusTest.UUI.
6. Select Configure > Control Engines to open the Control Engines dialog box.
7. Click Add, select the strategy file, and then click Open to open the Control Engines dialog box.
8. Click the Browse button, and then select the control engine you added to the strategy file.

NOTE: If you unzipped the files to somewhere other than the C: root directory, you will need to browse to the strategy file.

9. Click OK, and then click OK again.
10. Select Tools > AutoCorrect Tags.
The Select Window dialog box opens with all the windows selected.
11. Click OK.
When the process is completed, an AutoCorrect Tags message appears.
12. Click OK.
13. Select Tools > Regenerate IO Scanner Tag Names.
14. Select File > Save Project and Load Runtime.

Using the Modbus Utility

When the Modbus Utility window first opens in Runtime, a prompt is provided at the top of the window to guide you in using the utility. Use the prompts along the along with the instructions below.



1. In the initial Runtime window, select either Modbus TCP or Modbus Serial to open the appropriate window.

For Modbus TCP:

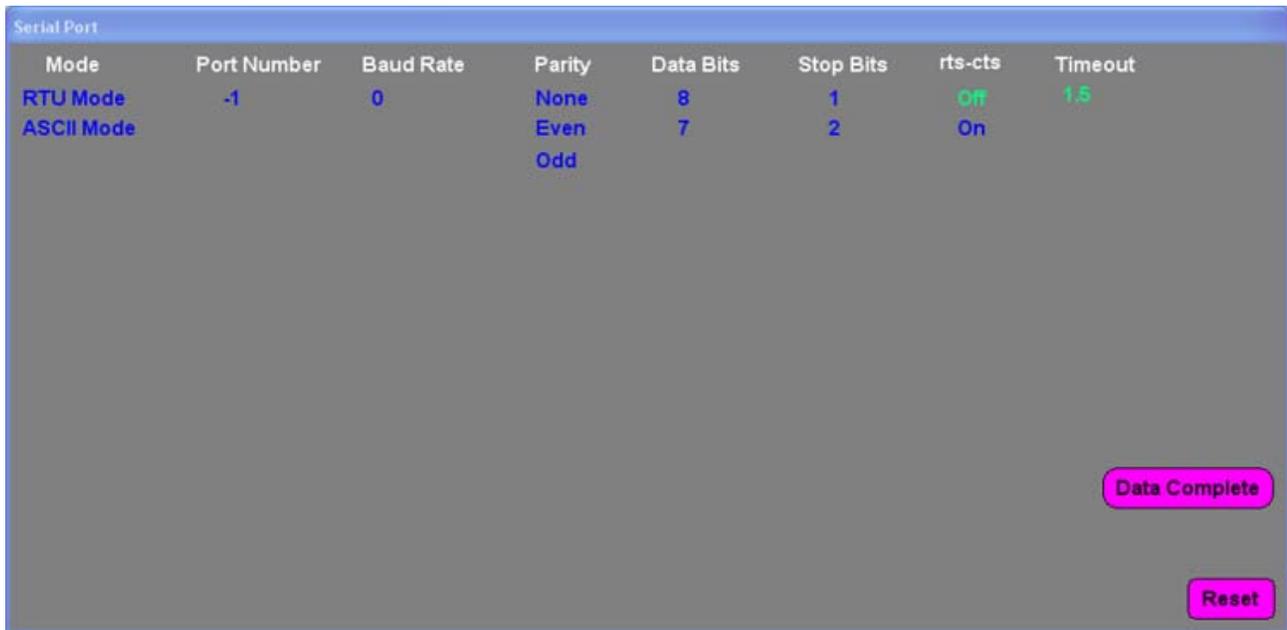
If you select Modbus TCP, the following window opens:



- a. Enter the slave IP address with your mouse on the on-screen keypad, and then click Enter.
For example: 10.175.52.110
This opens the Enter Port Number window.
- b. Enter the port number in the same fashion. Port 502 is the standard port.
- c. If the settings are correct, click Yes.
The strategy opens the port to the device.
- d. Proceed to [step 2](#) on [page 6](#).

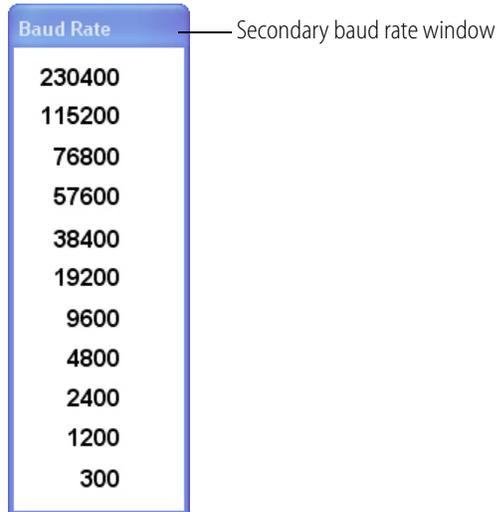
For Modbus Serial:

If you select Modbus Serial in the initial window, the following window appears:



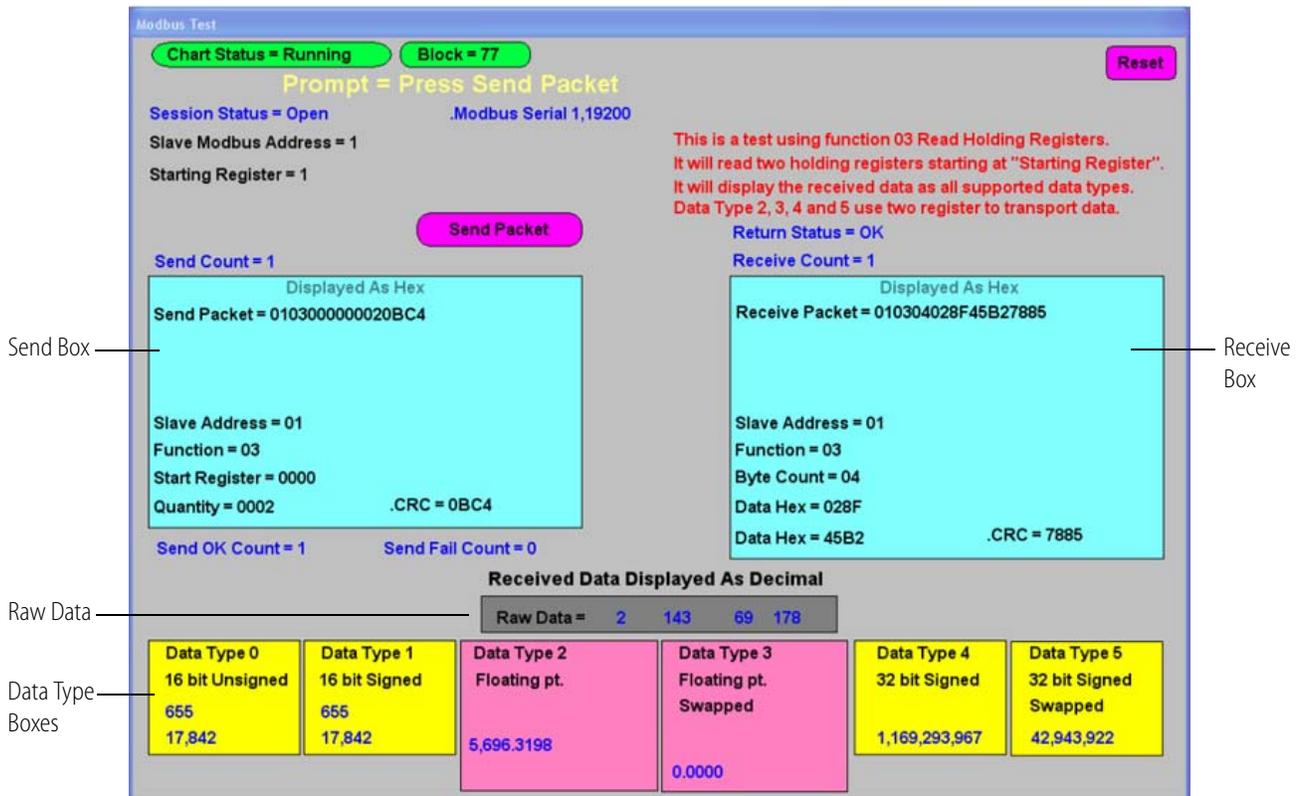
- a. Select the data that you want to change.

- b. For Port Number and Baud, a secondary window opens; make your selection, and the window disappears. For all the rest, simply click your selection in the main window.



2. Click Open Test Window.

After communicating with your Modbus device, a test window similar to the following example opens:



3. Click Send Packet to test communications and obtain packet data from the device.

The test window shows the following things:

- A **Send Box** contains the send packet data, the raw Modbus data in Hex
 - A **Receive Box** contains the received packet data, the raw Modbus data in Hex
 - A **Raw Data Box** shows the data portion in decimal
 - Five **Data Type Boxes** at the bottom of the test window show the received packet converted to all five data types
4. To determine the correct data type, find the Data Type Box that contains the value you would expect from the Modbus device.

In the following example for slave address 1 and starting register 1, the value expected from the Modbus device is 5,700.0000, which is very close to the value returned for Data Type 2.

Data type	
Data Type 0 16 bit Unsigned 655 17,842	Data Type 1 16 bit Signed 655 17,842
Data Type 2 Floating pt. 5,696.3198	Data Type 3 Floating pt. Swapped 0.0000
Data Type 4 32 bit Signed 1,169,293,967	Data Type 5 32 bit Signed Swapped 42,943,922

Expected value