Tech-note

Connecting UniOP as Modbus/TCP Server

This Technical Note describes the UniOP implementation of the Modbus/TCP Server protocol. Using this communication driver, multiple UniOP HMI panels acting as Servers can be connected to a Modbus/TCP network. A subset of the standard Modbus function codes has been implemented to allow for an easy and efficient information flow between the Clients on the network and the UniOP Servers.

***Important**: this Technical Note applies to Modbus/TCP communication driver identified by the name 'Modbus/TCP SRV' and included in the Designer files D32Uplc203.DLL. This communication driver requires an operator panel of hardware type –0045 or newer with firmware release 5.40 or higher and equipped with a communication module type SCM11with firmware release 2.01G or SCM11-C with firmware release 4.01G.*

Contents

1 Introduction 3

2 Principle of Operation.....4

3 UniOP Programming 5

Tn210
Ver. 1.00
© 2005 Sitek S.p.A. – Verona, Italy
Subject to change without notice
The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.
www.exor-rd.com

1 Introduction

The Modbus/TCP SRV communication driver lets you connect UniOP as a Server in a Modbus/TCP network. It is possible for Modbus/TCP Clients to connect then to multiple UniOP HMI panels acting as Servers. The information exchange will use standard Modbus messages over TCP/IP.
This approach will also offer an interesting way to connect UniOP HMI panels to SCADA systems through the universally supported Modbus/TCP communication protocol.

2 Principle of Operation

This communication driver will implement a Modbus/TCP Server unit in UniOP. A subset of the complete range of Modbus Function Codes will be supported. The available Function Codes will allow the transfer of data between Clients on the TCP network and the Server.
The diagram in Figure 1 shows the system architecture. Note that UniOP is actually simulating the communication interface of a PLC: it has two data types (Coils and Registers) that are respectively Boolean and 16 bit integers.
UniOP will always access data in its internal memory. Data can be transferred to and from the Modbus Client only on the initiative of the Client itself.

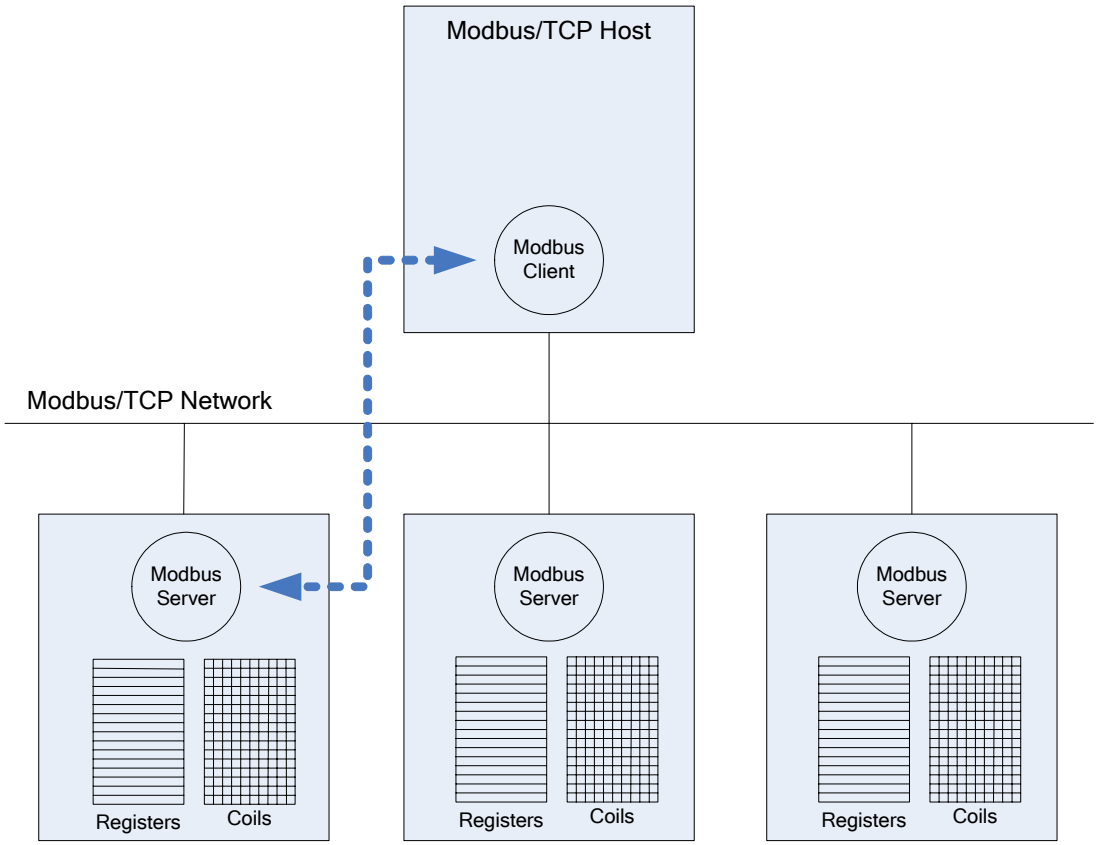

Figure 1 – System Architecture

3 UniOP Programming

To configure UniOP for operation as a Server in a Modbus TCP network, follow the steps listed in this chapter.

1) Choose the communication driver 'Modbus/TCP SRV' from the list in the 'Select Controller' dialog box. The 'Controller Setup' button can then be pressed to display the 'Modbus/TCP SRV – Controller Setup' dialog box as shown in Figure 2.

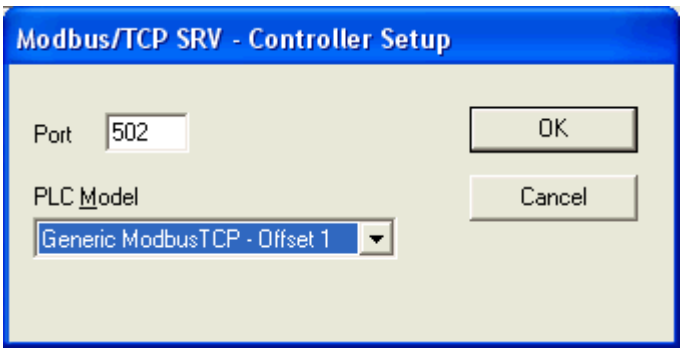

Figure 2 – Controller Setup dialog box

2) By default, the Modbus TCP protocol uses port 502 for communication with the nodes in the network. Unless your network uses a different port you should leave the Port setting to the default value of 502 otherwise set it to the value expected by your Modbus TCP network.

3) The Controller Setup dialog box allows you to select between two different flavours of Modbus/TCP. You can select the flavour most appropriate for your needs. The 'Generic Modbus/TCP – Offset 0' model type implements a Holding Register range of between 40000 and 41098 and an Output Coils range of between 0 and 9998. The 'Generic Modbus/TCP – Offset 1' model type, on the other hand, implements a Holding Register range of between 40001 and 41099 and an Output Coils range of between 1 and 9999. Please note that the address range used in the Modbus frames will always be respectively between 0 and 1098 for the Holding Registers and between 0 an 9998 for Coils.

4) UniOP is a Server in a TCP network and will exchange data with one or more Clients in the network. Every UniOP in the network must be assigned its own IP address. This can be set in the 'Panel Setup' dialog box. This dialog box is displayed by selecting the 'Project' menu item and by then selecting the 'Panel Setup' option. Selecting the 'External Devices' tab allows you to enter the IP address of the UniOP panel as shown below

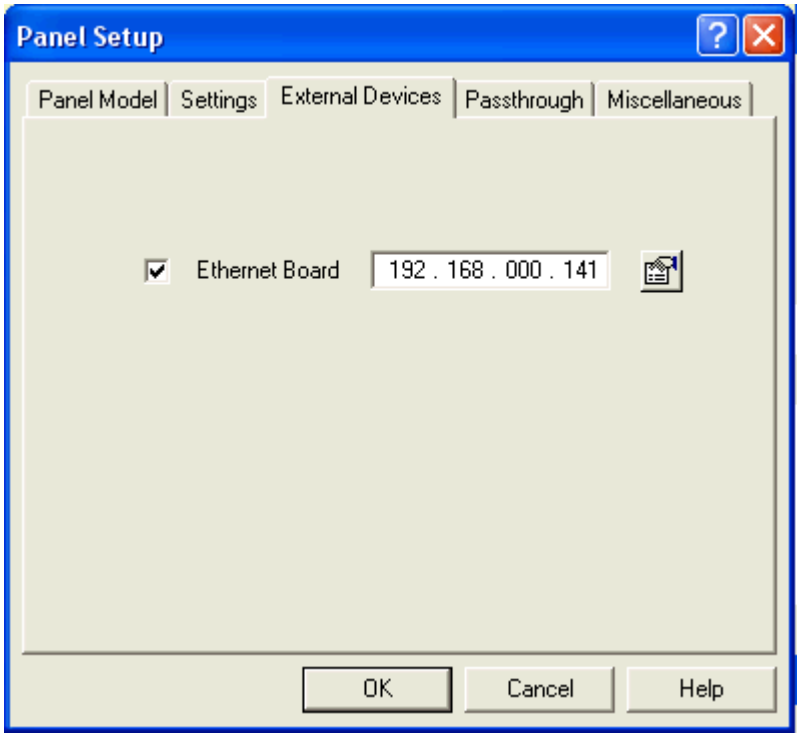

Figure 3 – Panel Setup dialog box

5) The RDA (Reserved Data Area), mailbox and alarms can be freely positioned in the UniOP Register area.

6) Dynamic data fields (variables) can be defined in the conventional way using the Data Field Definition dialog box (see figure below). Note that referencing a coil or register will always result in accessing the UniOP internal memory. Data can be transferred over the Modbus TCP link only on the initiative of the Modbus Client.

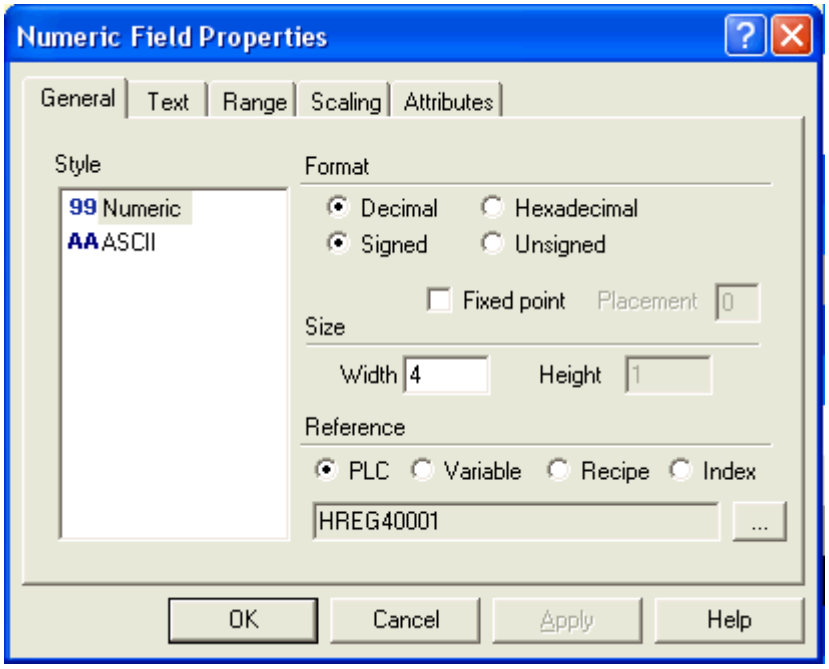

Figure 4 – Data Field Definition dialog box

Appendix A. Communication Error Codes

UniOP is a Server station in the Modbus TCP network. The current implementation of the protocol will not report any communication error code in the System Menu page.

Appendix B. Requirements

The following requirements apply to this communication driver:

Designer and Firmware	
Designer File	D32Uplc203.dll
Panel firmware version	5.40 or higher
Communication Modules	
SCM11	2.01G or higher
SCM11-C	4.01G or higher

Appendix C. Implementation Details

This Modbus TCP Server implementation supports only a subset of the standard Modbus Function Codes. Only the Function Codes necessary for the data exchange between UniOP and a Modbus TCP Client have been implemented.
The supported Function Codes are listed in the table below.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the UniOP Coil area
02	Read Input Status	Reads multiple bits in the UniOP Coil area
03	Read Holding Registers	Read multiple UniOP Registers
03	Read Input Registers	Read multiple UniOP Registers
05	Force Single Coil	Forces a single UniOP Coil to either ON or OFF
06	Preset Single Register	Presets a value in a UniOP Register
15	Force Multiple Coils	Forces multiple UniOP Coils to either ON or OFF
16	Preset Multiple Registers	Presets value in multiple UniOP Registers

UniOP will return the Exception Code 01 (Illegal Function) if the Function Code received in the query is not supported.

The amount of memory available in UniOP is as follows:

Data Type	Type	Range
Coils	Bit	0 – 9998
Registers	Word	0 – 1098

UniOP will return the Exception Code 02 (Illegal Data Address) if the Data Address received in the query exceeds the predefined data ranges.

Please note that for the ‘Generic Modbus/TCP – Offset 1’ and ‘Generic Modbus/TCP – Offset 0’ model types the ‘Read Coil Status’ and ‘Read Input Status’ function codes both access the same ‘Coil’ memory area in the UniOP memory. Also, the ‘Read Holding Registers’ and ‘Read Input Registers’ function codes both access the same ‘Register’ area in the UniOP memory.

Please note that a single UniOP Modbus TCP Server can only support a maximum of 3 concurrent connections. This means that up to 3 Clients could be connected to a UniOP Modbus TCP Server at the same time. However, a 4th Client, should it attempt to connect while the other 3 Clients are connected, will be unable to do so.