

Connecting UniOP to Modbus/TCP

This Technical Note contains the information needed to connect UniOP to control devices using the Modbus/TCP protocol using an Ethernet connection.

Contents

1	Introduction	3
1.1	Communication Media	3
2	Setting up the Device for Communication.....	3
3	Setting up UniOP	3
3.1	Controller Setup	3
3.2	Panel Setup.....	5
4	Modbus/TCP Data Types.....	6
5	Converting Projects.....	6
5.1	Converting Serial Modbus Projects to Modbus/TCP projects.....	6
5.2	Converting Modbus/TCP projects to Serial Modbus projects	7
	Appendix A. Communication Error Codes	8
	Appendix B. Implementation Details	9
	Appendix C. Requirements and Compatibility	10

Tn165

Ver. 1.00

© 2004 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 protocol is included in the Designer file UPLC160.DLL. UniOP operator panels can be connected to an Ethernet TCP/IP network using the Modbus/TCP communication protocol.

The optional SCM11 communication module is required to communicate over the 10 Mbit Ethernet using the TCP/IP protocol. The ETAD01 adapter is required to connect to the Ethernet using standard UTP patch cable.

1.1 Communication Media

The supported Ethernet physical layer is 10BASE-T. The SCM11 module requires the ETAD01 adapter to convert the 9-pin connector of the AUX port to the standard Ethernet 10BASE-T RJ45 connector. The adapter is delivered with the communication module.

2 Setting up the Device for Communication

Various Modbus/TCP-capable devices can be connected to the UniOP operator panel. To set-up your Modbus/TCP device please refer to the documentation you received with the device.

UniOP identifies Modbus/TCP devices using their IP addresses. You should take note of these addresses as you assign them because you will need them later in the set-up phase of the user interface application.

Different physical media, gateways, routers and hubs can be used in the communication network. Also, other devices can independently make simultaneous use of the network. However, it is important to ensure that the traffic generated by these devices does not degrade the communication speed (round-trip time) to an unacceptable level. Too slow communication between the UniOP and the Modbus/TCP device may result in low display update rate.

The UniOP implementation of the protocol operates as a Modbus/TCP client only. The UniOP Modbus/TCP protocol uses the standard port number 502 as the destination port. UniOP Modbus/TCP protocol supports the standard commonly referred as "Ethernet II".

3 Setting up UniOP

To create a UniOP application for Modbus over Ethernet, select the driver "Modbus/TCP" from the list of available communication drivers in the Change Controller Driver... dialog box.

3.1 Controller Setup

Figure 1 below shows the "Controller Setup..." dialog box for the Modbus/TCP protocol.

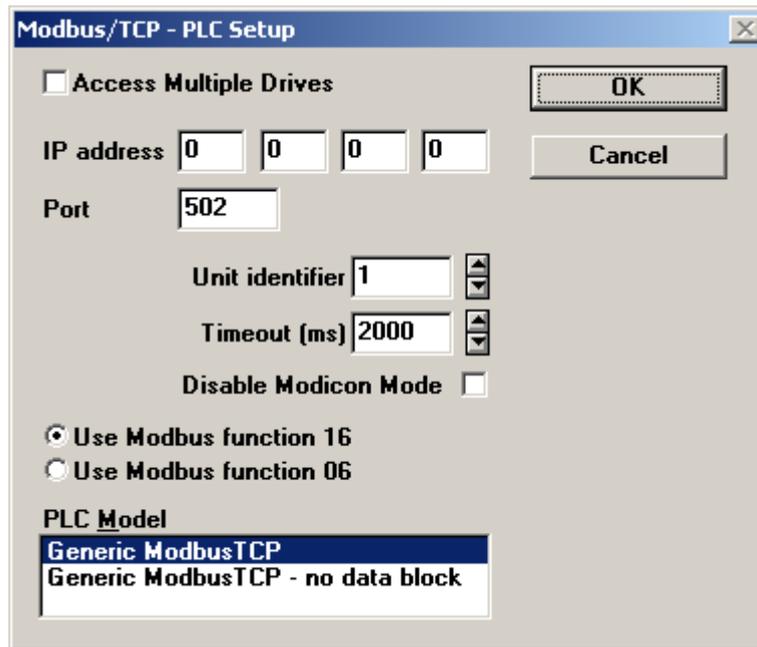


Figure 1 - Controller Setup dialog box

The IP address identifies the Modbus/TCP device on the network. The Unit Identifier is rarely used and in most cases can be left zero. The value of the Unit Identifier is simply copied into the Unit Identifier field of the Modbus/TCP communication frame. Usually it is used when communicating over Ethernet-to-Serial bridges and is then representing the Slave ID.

The “Disable Modicon Mode” checkbox allows specifying whenever the addressing mode should follow the original Modicon standard, or a more general approach where the address for the variables starts from index zero.

If Modicon Mode is enabled, the legal range for each data type is: 1...65536.

If Modicon Mode is disabled, the legal range for each data type is: 0...65535.

Please note that the permissible addressing range does not mean that the controller connected to the panel is actually containing that amount of variables.

One of two PLC models can be chosen. Selecting the “Generic ModbusTCP” PLC model will indicate to UniOP that it is allowed to access multiple data items at once resulting in greater display refresh ratio. Selecting the “Generic ModbusTCP - no data blocks” will force the UniOP to access single data item only per communication session. This will, of course, slow down the display refresh ratio, but can be useful when communicating to devices that have only basic Modbus functionality implemented. Check appendix B for more details.

The “**Time-out**” parameter defines the time inserted by the protocol between two retries of the same message. It is expressed in milliseconds.

The “**Port**” parameter allows to change the default port number used by the Modbus TCP driver; it could be useful whenever the communication passes through Routers or Internet gateways where the default port number is already in use.

Two radio buttons are available to specify what Modbus Function will be used for write operations to the Holding Registers data type. The user can select between the function 06 (preset single register) and function 16 (preset multiple registers).

If the Modbus function 06 is selected, the protocol will always use function 06 to write to the controller, even to write multiple consecutive registers.

If the Modbus function 16 is selected, the protocol will always use function 16 to write to the controller, even for a single register write request and the “Byte Count” parameter of the query is set to two. Using function 16 may result in higher communication performance
By default the Modbus Function 16 is used and should be disabled only for those controller that do not support it.

The protocol allows the connection to multiple Modbus/TCP devices connected to one UniOP panel. In that case, the “Access Multiple Drives” option has to be used. The dialog box will change and will appear as shown below in Figure 2.

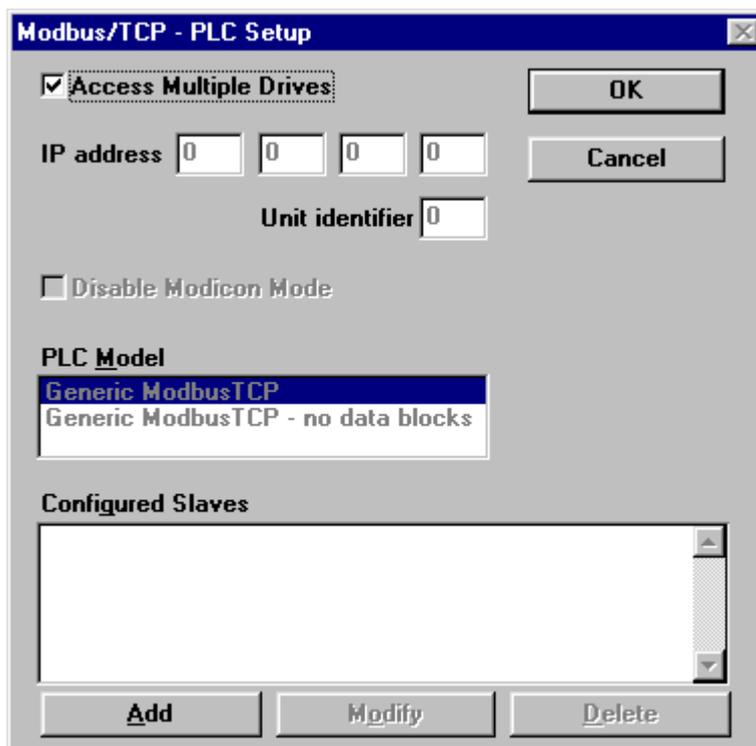


Figure 2 - Controller Setup Dialog Box for multiple controllers

By Pressing down the “Add” button you may add new Modbus/TCP devices to the network and enter their IP address. UniOP can communicate with several Modbus/TCP devices.

3.2 Panel Setup

The operator panel must have an IP address to connect to Ethernet.
The IP address of the panel must be entered in Panel Setup dialog box in the tab “External Devices”.
Enter the IP address in the “Ethernet board” field as shown in Figure 3 below.

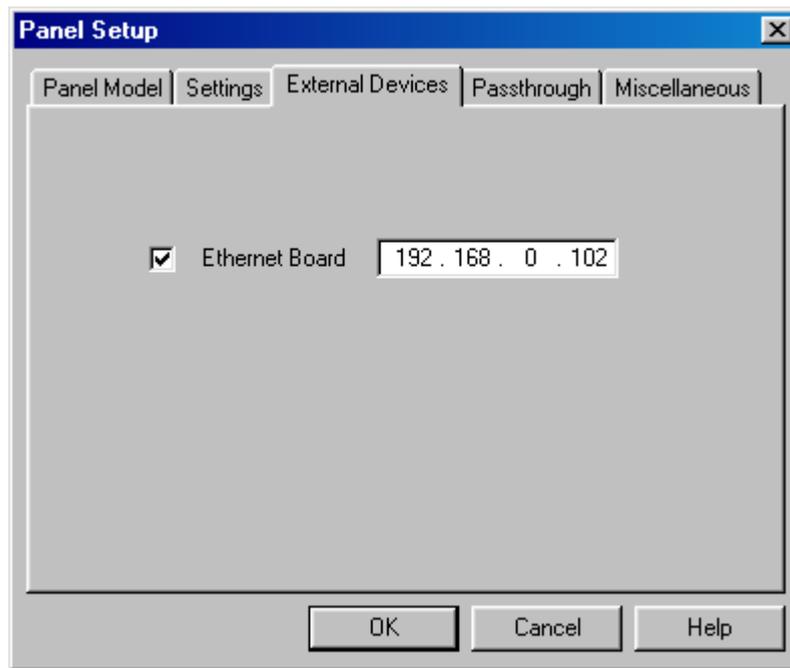


Figure 3

4 Modbus/TCP Data Types

The following standard Modbus data types can be accessed using the UniOP implementation of the Modbus/TCP protocol:

Data type	Access mode
Input bits	read only
Output coils	read/write
Holding registers	read/write
Input registers	read only

Up to 65535 variables of each data type can be addressed.

5 Converting Projects

The Modbus/TCP is compatible with several other Modbus protocols and projects can be easily converted among them. However, as the Modbus/TCP is an Ethernet-based protocol, some special steps have to be followed when converting from or to a serial Modbus driver.

5.1 Converting Serial Modbus Projects to Modbus/TCP projects

1. Modify Slave ID's in the project configured for serial Modbus and make sure that they all follow the increasing sequence 1,2,3...n, where n is the number of slaves used in the project.

2. Convert the project to Modbus/TCP.
3. Open the Controller Setup dialog box and create the PLC network including n nodes with unique IP addresses. The driver will link the first Modbus/TCP node to all PLC references previously linked to the PLC node with Slave ID number 1 and so on.

5.2 Converting Modbus/TCP projects to Serial Modbus projects

1. Convert the project to the serial Modbus driver.
2. Open the Controller Setup dialog box and create the PLC network with as many slave nodes as there were before the conversion (n). Set Slave ID numbers in the increasing sequence 1,2,3...n.
3. The first IP address in the list is converted to Slave ID 1, the second IP address in the list is converted to Slave ID 2, and so on.
4. Return to Controller Setup and adjust Slave ID's according to the actual configuration of the serial Modbus network.

Appendix A. Communication Error Codes

Current communication status is displayed in the System Menu of the UniOP.

A message and a numeric error code describe the error status.

The message reports the current communication status. The number shows the code of the current communication error or, if the communication is correct, the code of the last error encountered. When the error code 0 is shown, it means there have been no communication errors since this system start-up.

Code	Description	Notes
00	No error	There are no communication errors and there have been no errors since start-up.
05	Time out (receiving)	No response received from the controller within the timeout period (2 sec).
06	Response error	Received frame is shorter than expected
07	General communication error	The panel hardware and firmware does not support TCP/IP. The Modbus/TCP Controller returned some exception. See the Modbus/TCP specification for description of exception.
09	Time out (sending)	The protocol could not send data for a long time (6 sec).

Appendix B. Implementation Details

This Modbus/TCP implementation supports only a subset of the standard Modbus/TCP Function Codes.

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	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register*	Presets a value in a Register
16	Preset Multiple Registers*	Presets value in multiple Registers

* When the “Generic ModbusTCP” PLC model is selected in the Controller Setup (see chapter 3.1), function code 16 (preset multiple registers) will always be used for writing Holding Registers, even if there is no need to write more than a single Holding Register. Instead, when the “Generic ModbusTCP – no data blocks” PLC model is selected, function 06 (preset single register) will be always used for writing to Holding Registers data type.

Appendix C. Requirements and Compatibility

This communication driver is included in the Designer DLL file UPLC160.DLL. The initial release level is 3.00 for the communication driver and 5.00 for the DLL (both version numbers can be seen in the Change Controller Driver dialog box of the Designer software).

A communication module of type SCM11 is required.

The UniOP panel must have hardware type -0045 or -0050; firmware version number 4.42 or higher is required to support operation with the SCM11 module.