

Connecting UniOP as a Modbus Slave

This Technical Note describes the UniOP implementation of the Modbus RTU slave protocol. Using this communication driver, multiple UniOP HMI panels can be connected to a Modbus master. A subset of the standard Modbus function codes has been implemented to allow for an easy and efficient information flow between the master and the slaves.

Copyright © 2007 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.

Third-party brands and names are the property of their respective owners.

www.exor-rd.com

Contents

1	Introduction	4
2	Principle of Operation.....	5
3	UniOP Programming	6
3.1	Node Override.....	7

1 Introduction

The Modbus Slave communication driver lets you connect UniOP as a slave in a Modbus RTU network. It is possible to connect multiple UniOP HMI panels to one Modbus master controller. The information exchange will use standard Modbus messages.

This approach will also offer an interesting way to connect UniOP HMI panels to SCADA systems through the universally supported Modbus RTU communication protocol.

2 Principle of Operation

This communication driver will implement a Modbus RTU slave 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 the master and the slave.

The diagram in figure 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 master only on initiative of the master itself.

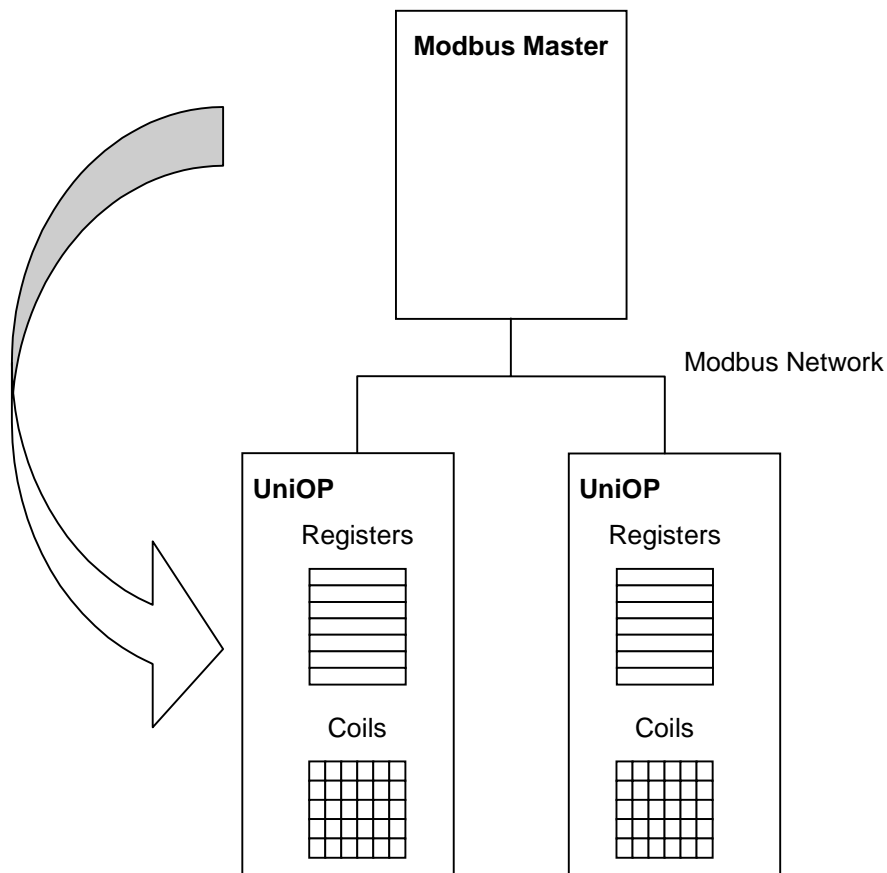


Figure 1 – System Architecture

3 UniOP Programming

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

- 1) Choose the communication driver 'Modbus Slave' from the list in the 'Change Controller Driver...' dialog box.
- 2) UniOP is a slave in the network and will exchange data with a single master controller. Every UniOP in the network must be assigned its own Slave ID. PLC Communication parameters must be set according to the values programmed in the master controller.

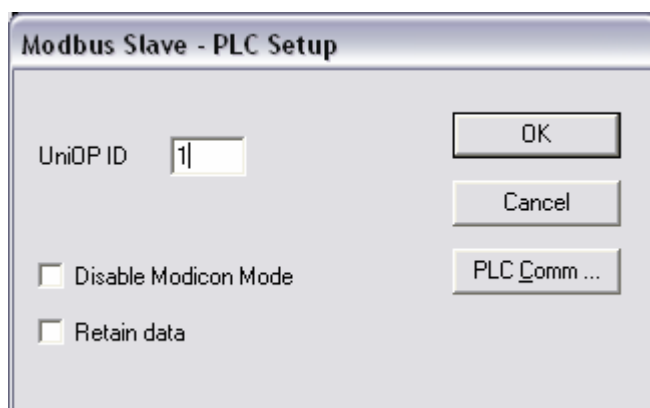


Figure 2 – Controller Setup dialog box

- 3) By default the driver will show address offsets in the 'Modicon Mode' and the starting address offset for registers and coils will be 1. Checking the "Disable Modicon Mode" check box, the driver will shift address offsets on position and the starting address offset will become 0.
- 4) When the "Retain Data" check box is marked the data into the first 500 Holding Registers and into all the Coils of the UniOP internal database will be retained at power cycle in RAM memory with battery back-up.
- 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 link only on initiative of the master controller.

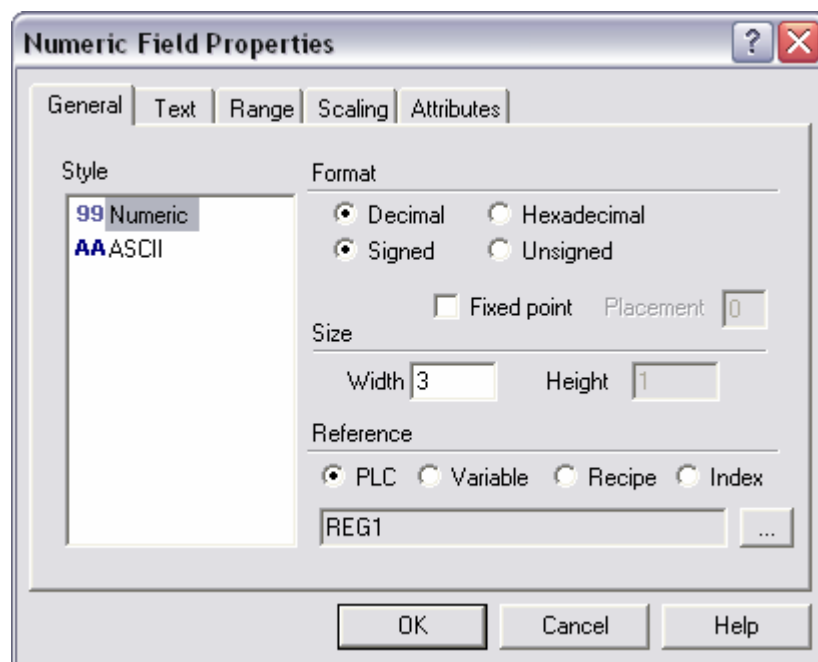


Figure 3 – Data Field Definition dialog box

3.1 Node Override

The UniOP ID address defined in “Controller Setup” can be changed at run-time using the special Data Type called “Node Override”.

Note: Using the Node Override feature requires Modbus Slave driver version 5.07 – 4.03 or above and firmware version V5.00 or above

Figure 4 shows the Modbus Slave Define Field dialog box when the special “Node Number Override” data type has been selected.

Node Number Override is an internal UniOP variable that allows changing the settings of the UniOP ID parameter specified at programming time in Controller Setup.

Node Number Override can be placed in any page of the project and the UniOP ID can be changed with a simple data entry operation on this variable.

Once the user changes the UniOP ID at run-time it will persist after the power cycle.

Changes to Node Number Override take effect immediately.

To restore the original project settings at run-time, the Node Number Override variable must be set to “0”.

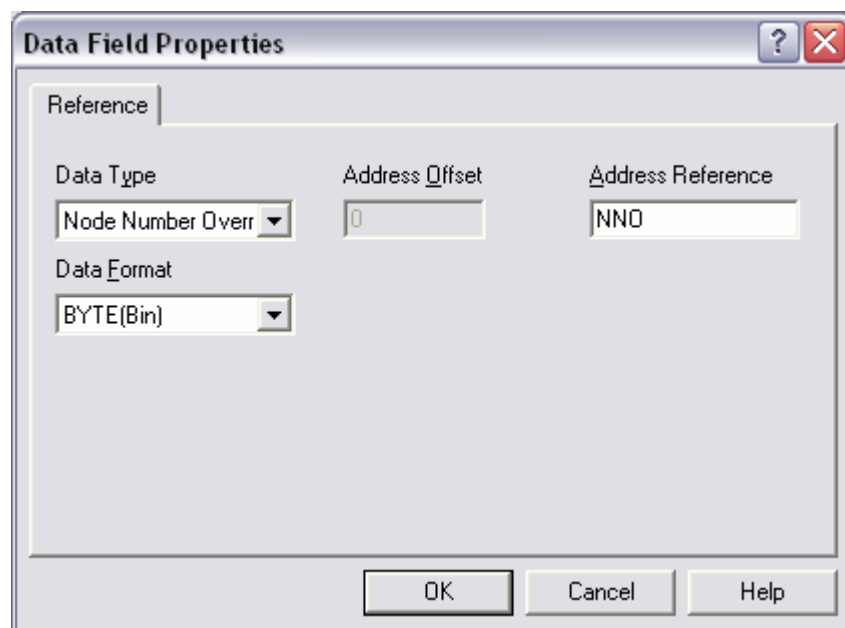


Figure 4 – Node Override

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.
01	Request not accepted	Memory range error
07	General communication error	Should never happen, internal software error.

Appendix B. Implementation Details

This Modbus RTU slave implementation supports only a subset of the standard Modbus Function Codes. Only the Function Codes necessary for the data exchange between UniOP and the Modbus master 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
03	Read Holding 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
08	Loopback Diagnostic Test	Only sub function 00 (Return Query Data) is supported
15	Force Multiple Coils	Forces multiple UniOP Coils to either ON or OFF
16	Preset Multiple Registers	Presets value in multiple UniOP Registers
17	Report Slave ID	Returns some diagnostic information of the controller present at the slave address

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

UniOP will return the Exception Code 03 (Illegal Data Value) if a sub function other than 00 is specified for Function 08.

The amount of memory available in UniOP is as follows:

Data Type	Type	Range
Coils	Bit	0 – 2047
Registers	Word	0 – 2047

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

Appendix C. Technical Data and Connection Information

Connect with baud rates values between 9600 and 38400 bauds.

UniOP supports connection with RS-232, RS-422 (4 wires) and RS-485.