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 MMI 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.

Contents

1. Introduction 1
2. Principle of Operation 2
3. UniOP Programming 3
Appendix A. Communication Error Codes 5
Appendix B. Implementation Details 6
Appendix C. Technical Data and Connection Information 7

1. Introduction

The Modbus Slave communication driver lets you connect UniOP as a slave in a Modbus network. It will be possible to connect multiple UniOP MMI panels to a Modbus master controller and exchange information using standard Modbus messages.
This approach will also offer an interesting way to connect UniOP MMI panels to SCADA systems through the universally supported Modbus 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 will transferred to and from the Modbus master only on initiative of the master itself.

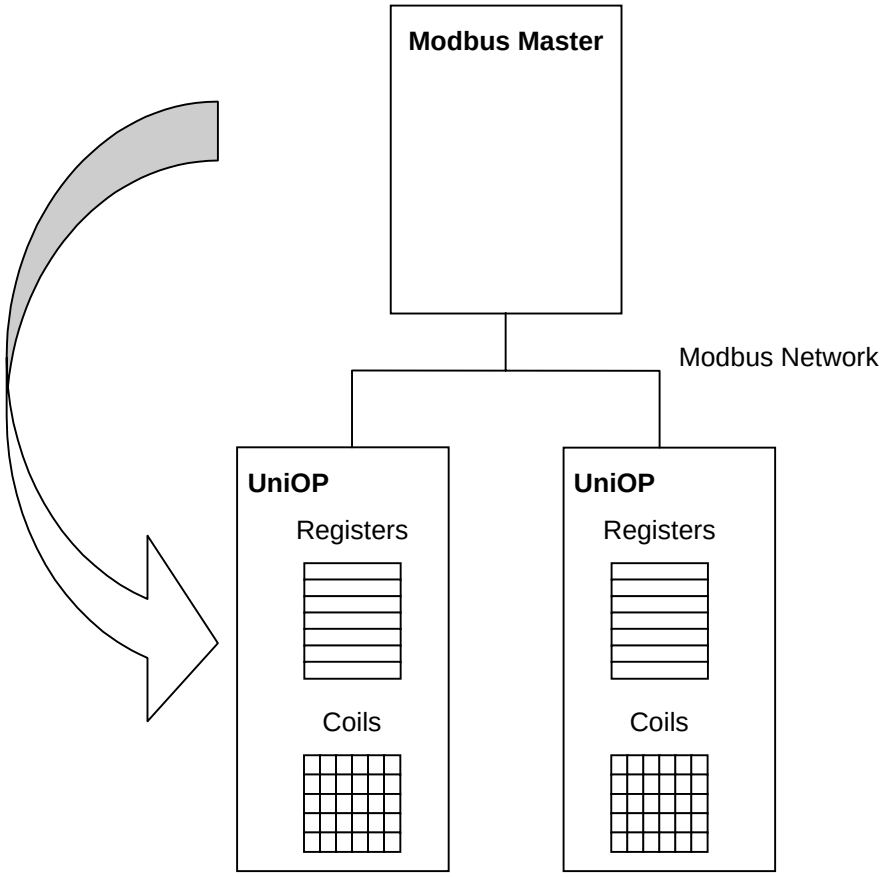

Figure 1 – System Architecture

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.

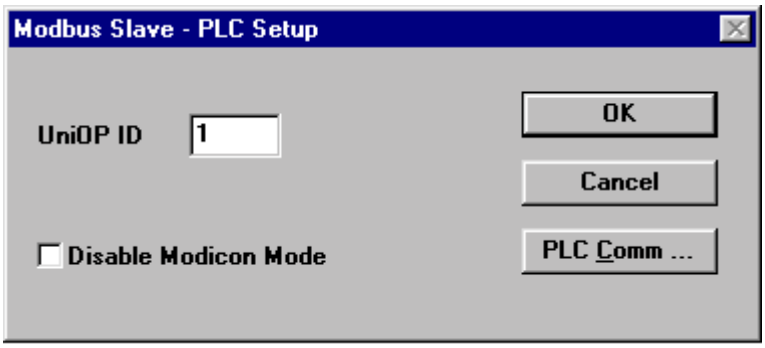

Figure 2 – Controller Setup dialog box

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 down address offsets and the starting address offset will become 0.

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

5) 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.

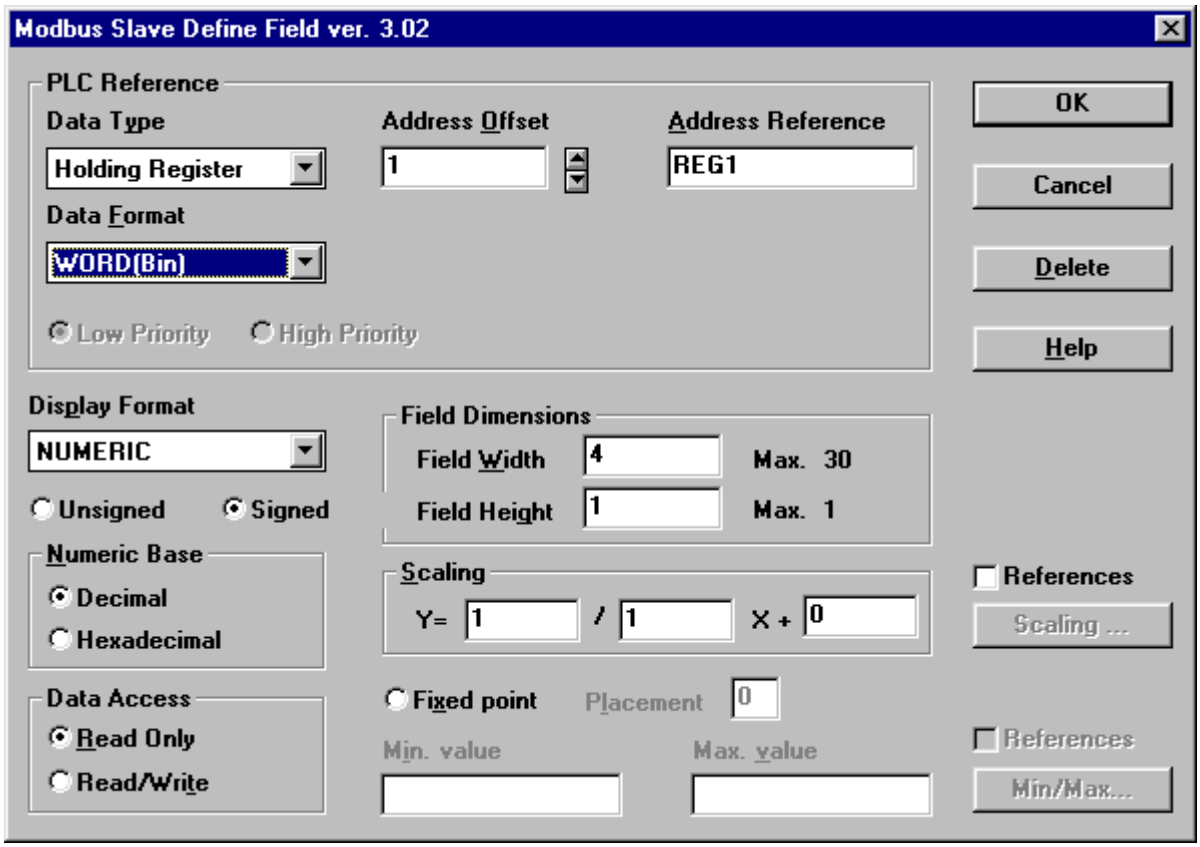

Figure 3 – Data Field Definition dialog box

Appendix A. Communication Error Codes

UniOP is a slave in the communication link. The current implementation will not report any communication error code in the System Menu page.

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 subfunction 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 subfunction 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 of 9600 or 19200 bauds.

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