Tech-note

Connecting UniOP Using Generic Modbus RTU

This Technical Note contains the information needed to connect UniOP to control devices using the Modbus RTU standard protocol over a serial communication link.

Contents

1 Introduction

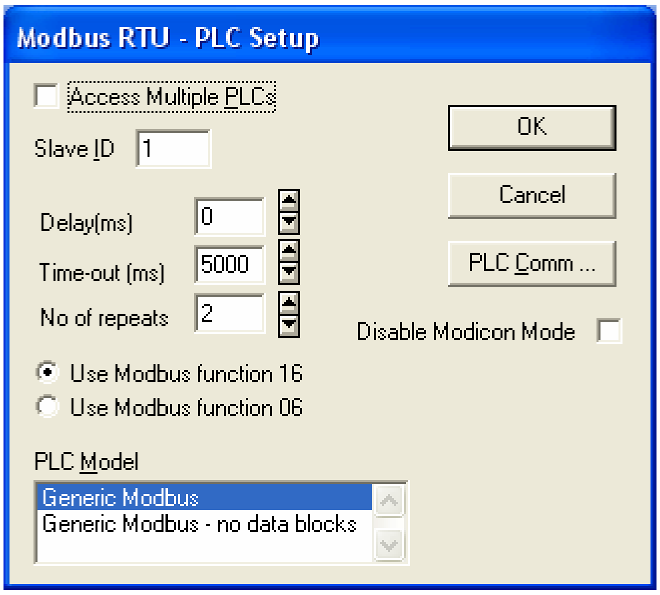
The Modbus RTU generic driver is included in the Designer 6 file D32Uplc166.DLL.
UniOP operator panels can be connected to a Modbus network as the network master using this generic driver.

2 Setting up UniOP for Communication

To create a UniOP application for a generic Modbus application, select the driver “Modbus RTU” from the list of available communication drivers in the Change Controller Driver… dialog box.

2.1 Controller Setup

Figure 1 below shows the "Controller Setup..." dialog box for the Modbus RTU protocol.
The Slave ID address is the address of the Modbus device on the network.



Access Multiple PLCs [ ]
Slave ID: 1
Delay(ms): 0
Time-out (ms): 5000
No of repeats: 2
Disable Modicon Mode [ ]
[✓ Use Modbus function 16] Use Modbus function 06
PLC Model: [✓ Generic Modbus] Generic Modbus - no data blocks

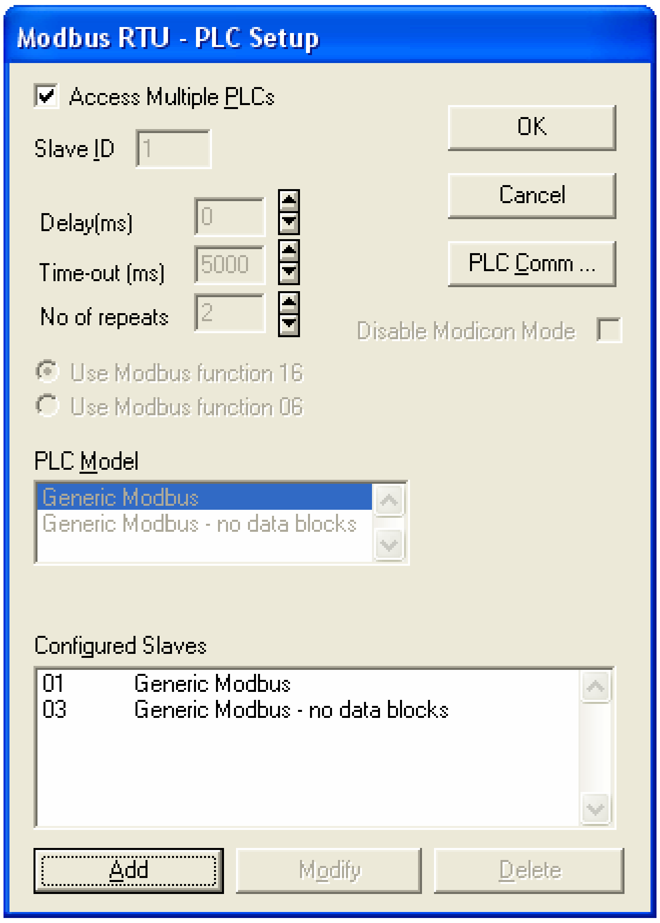
Figure 1 - Controller Setup dialog box

Disable Modicon Mode This checkbox defines if the addressing mode should follow the original Modicon standard (first address is 1) or a more generic approach where the address for the variables starts from index zero.
If Modicon Mode is enabled, the valid range for each data type is: 1…65536.
If Modicon Mode is disabled, the valid 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.

Delay This parameter defines a fixed time delay in the communication between the end of the last received frame an the starting of a new request; when set to 0, the new request will be issued as soon as the internal system is able to

	reschedule it.
No of repeats	This parameter defines the number of times a certain message will be sent to the controller before reporting the communication error status. A value of 1 for the parameter “No of repeats” means that the panel will eventually report the communication error status if the response to the first request packet is not correct.
Time-out	This parameter defines the time inserted by the protocol between two retries of the same message. It is expressed in milliseconds.
Use Modbus function 06 **Use Modbus function 16**	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 to dot support it.
Generic Modbus – no data blocks	This model forces the creation of Designer projects where the panel will make only communications requests including only one register. For each Function Code which supports multiple items access, the “No. of Points” parameter of the Modbus request frame is always fixed to one. This option will result in lower communication performance and should be used only when connecting to controllers that only can exchange one register at a time

The protocol allows the connection to multiple Modbus slave devices connected to one UniOP panel. In that case, the “**Access Multiple PLCs**” option must be used.
When Access Multiple PLCs is checked, the dialog box will change and will appear as shown in Figure 2 below.


Modbus RTU - PLC Setup

[✓] Access Multiple PLCs

Slave ID: 1

Delay(ms): 0

Time-out (ms): 5000

No of repeats: 2

Disable Modicon Mode [ ]

(●) Use Modbus function 16

( ) Use Modbus function 06

PLC Model:
Generic Modbus
Generic Modbus - no data blocks

Configured Slaves:

```
01        Generic Modbus
03        Generic Modbus - no data blocks
```

Figure 2 - Controller Setup Dialog Box for multiple controllers

Press the “Add” button to add new Modbus devices to the network and enter their address.

A typical situation in which UniOP is connected to several slave devices and a project page contains variables coming from all the partners, may become critical from the point of view of the data refresh time, when one or more of the slaves is missing or not responding.
A proper combination of the parameters “Time-out” and “No of repeats” allows to fine tune the application. The Modbus master, in fact, can be configured to skip very quickly over the nodes that are not responding.

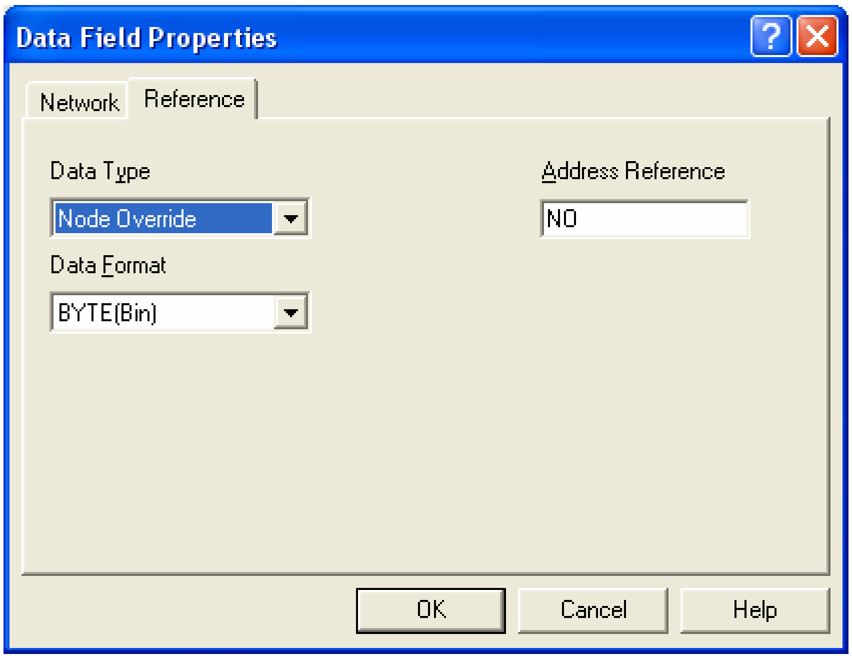
Note: *The value “0” for Slave ID indicates that all the write-only commands to that device are intended as broadcast. It is not possible to place in a screen data fields using this node number. This reference can only be used in write-only operations. Broadcast commands are supported from driver version 5.25 – 4.11.*

2.2 Node Override

The Slave ID parameter you have entered in “Controller Setup” can be changed at run-time using the special Data Type called “Node Override”.

Note: *Node Override requires the Modbus RTU driver version 5.07 – 4.04 or above and firmware version V4.50S or above*

Figure 3 shows the Modbus RTU Define Field dialog box when the special “Node Override” data type is selected.
Node Override is an internal UniOP variable that makes possible changing the settings for the Slave ID parameter specified at programming time in Controller Setup.
Node Override can be placed in a page and the Slave ID can be changed with a simple data entry operation over this variable.



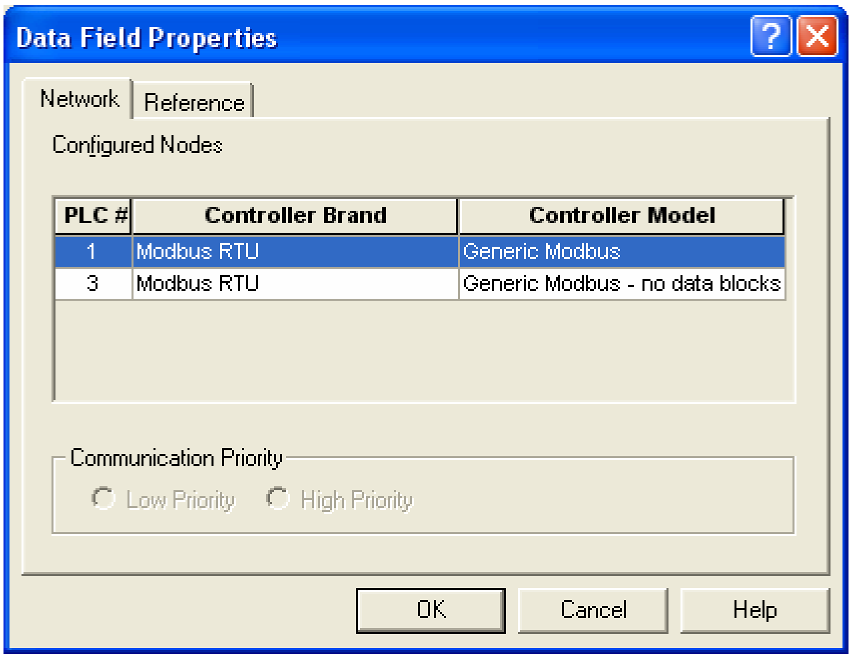
Data Type: Node Override
Data Format: BYTE(Bin)
Address Reference: N0

Figure 3 – Node Override

Changes to “Node Override” will require a UniOP power cycle to be effective with driver versions older than Drv 4.12 Dll 5.25.
From Drv 4.12 Dll 5.25 the changes will be effective runtime, without the need of a power cycle.

When the “Controller Setup” is configured for “Access to Multiple Controller” there is one “Node Override” variable for each of the nodes included in the configuration. In the case of Multiple Controllers the dialog changes as shown in Figure 4.
The PLC Slave ID list box will define which Slave ID will be affected.

Note: *When “Access Multiple Controllers” is enabled, UniOP can store* ***up to 8*** *different Node Override variables; that is, if Controller Setup includes more than 8 slaves, only 8 of them can be re-mapped using the Node Override option.*


Data Field Properties

Network | Reference

Configured Nodes

PLC #	Controller Brand	Controller Model
1	Modbus RTU	Generic Modbus
3	Modbus RTU	Generic Modbus - no data blocks

Communication Priority

Low Priority High Priority

Figure 4 – Node Override for Multiple Controller access

To restore the original project settings, the Node Override variable must be set to “0”.

Node Override parameters are stored in Battery Backed RAM, protected by CRC and duplicate copy. In case of memory corruption, parameters are restored to backed-up values if possible.

2.3 The Node Control

The “Node Control” data type allows access to special memory registers held internally in UniOP panel. These memory registers act as “switches” that make it possible to enable/disable access to particular controllers in the controller network at run-time.

The Node Control data type is available only if the ‘Access Multiple Controllers’ is configured in the Controller Setup dialog box.

When the access to particular controller configured in the network is disabled, the UniOP will not produce any request to that controller. Thanks to that, if the controller is temporary disconnected, the UniOP will not report the communication error and considerable amount of time will be saved, leading to faster display refresh rate.

The Node Control registers are represented as bits. They can be changed at run time; setting the Node Control register to ‘1’ will enable access the actual controller; setting it to 0 will disable access to the actual controller.

In the Define Field dialog box, the “PLC Slave ID” combo box is used to reference each controller configured in the network.

The configuration of the Node Control registers is stored in RAM with battery back up.

3 Modbus RTU Data Types

The following standard Modbus data types can be accessed using the UniOP implementation of the Modbus RTU 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.

Note that the address values actually used in the Modbus packet frame are: “Reference Address – 1”. When the ‘Disable Modicon Mode’ option is checked in Controller Setup, the address used in the Modbus frames on the serial line are the same as the reference entered in the Define Field dialog box.

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.
04	**Negative ACK**	NAK returned by the controller
05	**Time out (receiving)**	No response received from the controller within the timeout period.
06	**Response error**	Error code field (BCC) received from controller is wrong.
07	**General communication error**	General un-known communication error

Appendix B. Implementation Details

This Modbus RTU implementation supports only a subset of the standard Modbus Function Codes. The supported Function Codes are listed in the table below.

Code	Function	Description
01	Read Coil Status	Reads multiple bits
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

Communication speed with controllers is supported up to 57600 baud.

Floating point data format is compliant to the IEEE standard.

Note: *Baud rate greater than 38400 are only supported starting for firmware version V4.22. Speed is anyway still limited to 9600 for UniOP models without PC/Printer port.*