1. About this manual .................................................................5
   1.1. Copyrights ........................................................................5
   1.2. Trademarks ........................................................................5
   1.3. General .................................................................................5
   1.4. Use of symbols ....................................................................7
   1.5. Document conventions .....................................................7
   1.6. Related documents ...........................................................8
   1.7. Document revisions ..........................................................8
2. Safety information .................................................................9
   2.1. Backup copies .....................................................................9
   2.2. Fatal errors .........................................................................9
3. Communication system configuration ..................................11
   3.1. Modbus line configuration .................................................12
   3.2. Modbus PLC device configuration ......................................20
   3.3. Topics of a PLC device .......................................................25
      3.3.1. Topic parameters .......................................................26
      3.3.2. Address conversion between RTU and Modbus .............29
4. Using the Modbus Master Protocol .....................................31
   4.1. Accessing Modbus data using process objects .................31
   4.2. Configuring the device attribute interface ..........................33
      4.2.1. Attribute interface of a PLC device .............................33
      4.2.2. Example of using device interface commands ............36
      4.2.3. Configuration examples ............................................37
      4.2.4. Modbus TCP/IP protocol configuration examples .......39
5. Appendix. Serial cable wiring diagram ..............................43
6. Index ..................................................................................45
1. About this manual

1.1. Copyrights

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Oy. ABB Oy assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Oy be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Oy be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB Oy, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © 2004 ABB Oy
All rights reserved.

1.2. Trademarks

Registrations and trademarks used in this document include:

Windows: Registered trademark of Microsoft Corporation.

1.3. General

This manual is a programming manual. It describes the principles of interfacing MicroSCADA to an external device using the Modbus protocol.

In order to fully understand the concepts outlined here, the reader should be familiar with the SCIL programming language and general MicroSCADA application techniques. General knowledge about the Modbus protocol and PLC programming is also needed.

Modbus master protocol

The Modbus master protocol is mainly used for a master-slave connection between intelligent devices. This means, in our case, connecting Programmable Logic Controllers (PLCs) to MicroSCADA.

General principles of the Modbus protocol

The Modbus Communications Protocol is an asynchronous, byte packaged protocol used for communications between the master stations and Intelligent Electronic Devices (IEDs) or Remote Terminal Units (RTUs). It provides a transport mechanism for the master’s requests and RTU response messages. It supports one single master station and up to 247 RTUs on a multi-drop line. The Modbus protocol functions as a true serial line or Transmission Control Protocol (TCP) network.
The Modbus protocol has two distinct modes: ASCII Modbus, which uses ASCII-encoded hexadecimal messages and binary Modbus, which uses raw binary messages. The Modbus slave implementation described in this document supports only the binary mode.

All transactions are initiated by transmission of a request from the master station, an RTU may not transmit unsolicited information. Every master station request must be addressed to a specific RTU and some implementations of Modbus do not support the broadcast message request type. A transaction consists of a single master station request, followed by an RTU response or exception frame or a master station timeout if no RTU response is generated.

There are many different kinds of devices, which use the Modbus protocol for communication. Therefore, MicroSCADA Modbus protocol emulation is somewhat restricted. All the features that are available in external devices are not necessarily available in the NET implementation.

The general strategy that is used in this implementation can be summarized as follows:

- PC-NET contains necessary protocol conversion features, which enable it to send and receive Modbus telegrams.
- The SCIL application program in the base system sees the PC-NET Modbus emulator as a PLC type device.
- The PLC type device uses the MicroSCADA process database like a RTU device.
- Cross-reference information of correlation between the RTU addresses and Modbus addresses is stored in the topic configuration data of the PLC device.
- A MicroSCADA application program can refer to an external device data through the MicroSCADA process database, or it can use direct communication attributes to read from or write data to an external device.
- PC-NET is the protocol master. The communication with external devices is done by using the Modbus RTU protocol mode.
- The current version of the Modbus protocol emulator of PC-NET supports functions 1, 2, 3, 4, 5, 6, 16.

**Modbus TCP master protocol**

The Modbus TCP master protocol is used in LAN and WAN networks to connect central stations and outstations to each other. Since the stations use an open TCP/IP interface as a connection to the network, the structure and the characteristics of the network will be invisible to MicroSCADA communication software. The overall performance of Modbus/TCP is higher compared to serial line Modbus due to the faster transmission speed.

The MicroSCADA implementation of Modbus TCP operates as a master. The protocol numbers that are used are equal to the current ones. The SD attribute of the line defines if the line uses serial port, TCP. The SD attribute must be set before the line is taken into use for the first time.

**Examples**

```
#SET NET'SET':SSD'LINE' = "COM5"; line uses serial port 5
#SET NET'SET':SSD'LINE' = "TCP"; line uses TCP connection
```
MicroSCADA is able to keep several connections open to the controlled stations at the same time. Multiple Modbus/TCP lines may be created in the same computer.

The transferred data messages are close to the messages used in serial line based Modbus protocol. Compared with serial line implementation, a seven byte MBAP header is added to the beginning of each message. The Modbus master operates always as a TCP client.

For more information on Modbus TCP, see the following web site: www.modbus.org: Modbus messaging on TCP/IP rev 1.0 8.5.2002.

1.4. Use of symbols

This publication includes information and caution symbols where it is appropriate to point out safety related or other important information. It also points out useful hints for the reader. The corresponding symbols should be interpreted as follows:

Caution icon indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard, which could result in corruption of software or damage to equipment/property.

Information icon alerts the reader to pertinent facts and conditions.

1.5. Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the CTRL key. The Enter and the Shift keys are exceptions, for example press Enter.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key, Enter and so on.
- Press CTRL+C indicates that you must hold down the CTRL key while pressing the C key (to copy a selected object in this case).
- Press ESC E C indicates that you press and release each key in sequence (to copy a selected object in this case).
- The names of push and toggle buttons are boldfaced. For example, click OK.
- The names of menus and menu items are boldfaced. For example, the File menu.
- The following convention is used for menu operations: Menu Name > Menu Item > Cascaded Menu Item. For example: select Edit > Clear > All.
- The Start menu name always refers to the Start menu on the Windows Task Bar.
- System prompts/messages and user responses/input are shown in the Courier font. For example, you may be told to define the RP 570 slave protocol to line 1 of NET 1 and add a master station with number 8 into it, the example string is shown as follows in the procedure:
1.6. **Related documents**

The following MicroSCADA manuals should be available for reference during the use of this manual:

<table>
<thead>
<tr>
<th>Name of the manual</th>
<th>MRS number</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Configuration manual</td>
<td>1MRS751846-MEN</td>
</tr>
<tr>
<td>System Objects manual</td>
<td>1MRS751847-MEN</td>
</tr>
<tr>
<td>Application Objects manual</td>
<td>1MRS751848-MEN</td>
</tr>
</tbody>
</table>

1.7. **Document revisions**

<table>
<thead>
<tr>
<th>Version</th>
<th>Revision number</th>
<th>Date</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>15.03.2002</td>
<td>Document created</td>
</tr>
<tr>
<td>B</td>
<td>8.4.5</td>
<td>16.01.2003</td>
<td>New pictures added in Chapter 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Modbus TCP attribute LD added in Chapter 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Modbus serial line added in Chapter 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Modbus TCP/IP attributes added in Chapter 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TCP additions for LAN / WAN Networ, Example topologies added in Chapter 4</td>
</tr>
<tr>
<td>C</td>
<td>9.0</td>
<td>30.06.2004</td>
<td>Windows NT changed to Windows</td>
</tr>
</tbody>
</table>
2. **Safety information**

This chapter gives information about the prevention of hazards.

2.1. **Backup copies**

We suggest that you take backup copies before making any changes, especially the ones that might have side effects. Software and data need to be copied to another place, usually to a CD or a backup tape. A writable CD and DAT tape are commonly used.

Backup copies make it easier to restore application software in case of a disk crash or any other serious failure when stored data is lost. It is therefore recommended that backup copies are taken regularly.

There should be at least two system backup copies and two application copies. A new backup is copied over the oldest backup. This way the latest version is always available, even if the backup procedure fails.

Detailed information on how to take backup copies should be delivered to the customer with the application.

**System backup**

Usually a system backup is taken after the application is made. A backup should be taken again when changes are made to the MicroSCADA system. For example, if the driver configuration or the network set-up is changed.

**Application backup**

An application backup is taken simultaneously with the system backup after the application is made. A backup should be taken again when changes are made to the application. For example, if pictures or databases are edited or new pictures are added.

2.2. **Fatal errors**

A fatal error is an error that causes a break-down or a locked situation in the MicroSCADA program execution.

**Handling**

In case of a fatal error:

1. Write down the possible MicroSCADA error messages.
2. Shut down the MicroSCADA main program. If this cannot be done in the MicroSCADA Control Panel, try to end the task in Windows Task Manager.

   ![Warning](image) Shutting down the base system computers by switching off the power might damage the files.

3. In Windows, the data kept in the main memory at the moment of a fatal error is placed in the drwtsn32.log file. It is placed in a system folder, for example WINNT. Analyze and copy the data in this file.
4. Restart the system.
Report the program break-down together with possible MicroSCADA error messages and the information from the drwtsn32.log file to the MicroSCADA supplier.

**Status codes**

Error messages in SCIL are called status codes. For a list of status codes with short explanations, refer to the Status Codes manual.
3. Communication system configuration

Introduction
Each NET unit contains a set of system objects, which specify line properties, connected devices etc. These objects can be created, modified and deleted by SCIL, and setting the attributes of the objects can change the properties.

Access to the attributes can be one of the following:

• **Read-only:** The attribute can only be read. There are still a few exceptions in which the values can be reset.

• **Write-only:** The attribute can only be written (set).

• **Read, conditional write:** The attribute can be both read and written, but the object must be set out of use (IU = 0) before writing.

• **No limitations:** The attribute can be both read and written without limitations.

The implementation of the Modbus master protocol in MicroSCADA can be divided into two layers: link layer and application layer. Both of these layers have a specific functionality and a set of attributes of their own. The link layer corresponds to a line of a NET unit and the application layer corresponds to a station configured to the line.

The purpose of the communication system configuration is to:

• Create all the system objects needed to establish communication between the master and the slave.

• Adjust the values of the system object attributes to match the physical communication channel and the properties of the slave station.

Setting the attribute values
All line and station attributes have sensible default values but the value of each attribute must be checked against the requirements of the real communication system. The attribute values depend on:

• The physical communication media (e.g. leased telephone line, radio link, and power line carrier). This affects particularly the attributes of the line, e.g. baud rate and parity.

• The network topology used (point-to-point, multidrop). This affects, for example, the link type.

• The size (number of stations) of the system. This affects especially the timeout parameters: the slower the media and bigger the system, the longer timeouts are needed.

Network topologies
The implementation of the Modbus master protocol in MicroSCADA supports direct and serial bus topologies. The direct topology (point-to-point) can be a direct physical cable from point-to-point or a two-node radio, or modem network. The serial bus topology (multi-drop) is commonly made up of many modems with their outputs and inputs tied together, or by using a star-coupler.
The Modbus protocol supports one master on a line. The following figure illustrates the network topologies:

![Network topologies]

**Fig. 3.1 Network topologies**

### 3.1. Modbus line configuration

The line process of a NET unit performs the functions of the link layer. The purpose of the link layer is to send and receive messages with external devices using the Modbus protocol. The link layer provides also frame synchronization and link control.

**Line attributes**

The following attributes can be used to configure the Modbus master lines in MicroSCADA.

- **IU In Use**
  - Indicates whether the line is in use (value 1) or not in use (value 0).
  - **Data type:** Integer
  - **Value:** 0, 1
  - **Index range:** 1...8 (NET line numbering)
  - **Default value:** 0
  - **Access:** No limitations

- **PO Protocol**
  - The data transfer protocol used on the line. The line is defined to the NET by setting this attribute. By setting the attribute to 0 the line definition including all the line attributes is deleted.
MicroSCADA Pro
Modbus Master Protocol
Configuration Manual

Data type: Integer
Value: 0... 35
Value with Modbus master protocol: 25
Index range: 1... 8 (NET line numbering)
Access: Read, conditional write

SD System Device Name
Associates the NET line numbers of PC-NET with the device names of the physical channels of serial ports.
By default, line number 1 is connected to COM1, line 2 to COM2 and so on. By using the SD attribute it is possible to override these default values. This may be necessary if COM ports are used as NET lines or if, for example, a RocketPort card is used.
Data type: Text
Value: See above
Index range: 1...8 (NET line numbering)
Access: Read, conditional write

PS Buffer Pool Size
Specifies the number of message buffers reserved for the line. Each buffer can contain one message. The maximum data content length of a message is 228 bytes.
Data type: Integer
Value: 1... 250
Index range: 1... 8 (NET line numbering)
Default value: 12
Access: Read, conditional write

BR Baud Rate
Transmission rate used on the line.
Data type: Integer
Value: 1...19200
Unit: Bits/s
Index range: 1...8 (NET line numbering)
Default value: 9600
Access: Read, conditional write

PY Parity
Specifies the parity check (if any) used for the characters transferred on the line.
Data type: Integer
Value: 0 = no parity check  
1 = odd parity  
2 = even parity  

Index range: 1...8 (NET line numbering)  
Default value: 2  
Access: Read, conditional write  

**RD Receiver Data Bit Count**  
Specifies the number of data bits in each received character.  
Data type: Integer  
Value: 5, 6, 7 or 8  
Unit: Data bits  
Index range: 1...8 (NET line numbering)  
Default value: 8  
Access: Read, conditional write  

**SB Stop Bits**  
Specifies the number of stop bits attached to each transmitted character.  
Data type: Integer  
Value: 1 or 2  
Unit: Stop bits  
Index range: 1...8 (NET line numbering)  
Default value: 1  
Access: Read, conditional write  

**TD Transmitter Data Bit Count**  
Specifies the number of data bits in each transmitted character.  
Data type: Integer  
Value: 5, 6, 7 or 8  
Unit: Data bits  
Index range: 1...8 (NET line numbering)  
Default value: 8  
Access: Read, conditional write  

**PD Polling Delay**  
Delay between polling messages. The master sends the request with an interval defined by this attribute.  
Data type: Integer  
Value: 0...65535  
Unit: Milliseconds
| Index range: | 1...8 (NET line numbering) |
| Default value: | 40 |
| Access: | Read, conditional write |

**DE  CTS Delay**

Time delay (in milliseconds) between the activation of the RTS signal (Request to Send) and the start of a new transmission. With the value 0, the transmission starts immediately when the CTS signal is detected to be signalled.

- **Data type:** Integer
- **Value:** 0...65535
- **Unit:** Milliseconds
- **Index range:** 1...8 (NET line numbering)
- **Default value:** 0
- **Access:** Read, conditional write

**HT  Header Timeout**

Specifies the maximum waiting time in milliseconds within which the first byte of a link layer response should have been received after the transmission of a message. If no response has been received within this time, new attempts are performed the number of times specified by the Enquiry Limit. If no response is still obtained, the station is be suspended.

- **Data type:** Integer
- **Value:** 0...65535
- **Unit:** Milliseconds
- **Index range:** 1...8 (NET line numbering)
- **Default value:** 700
- **Access:** Read, conditional write

**TI  Response Timeout**

The time in milliseconds that the DNP link waits for the end of the link layer response.

- **Data type:** Integer
- **Value:** 0...255
- **Unit:** Seconds
- **Index range:** 1...8 (NET line numbering)
- **Default value:** 2
- **Access:** No limitations

**RI  Receive Interrupt Enable Delay**

Defines the delay in milliseconds after which the receiver of a NET line is enabled when a message has been issued.
Data type: Integer
Value: 0...255
  0 = receiver enabled all the time
  1…9 = receiver enabled right after transmission
  10… = receiver enabled as stated by the value
Unit: Milliseconds
Index range: 1...8 (NET line numbering)
Default value: 5
Access: Read, conditional write

**EN**

**Enquiry Limit**

Specifies the maximum number of times that a message is retransmitted after a timeout.

Data type: Integer
Value: 1...255
Index range: 1...8 (NET line numbering)
Default value: 1
Access: Read, conditional write

**SG**

**Modem Signal**

An attribute for direct supervision and control of the state of the modem signal. This attribute applies to all protocols. It is used for diagnostics and testing.

Data type: Integer
Value: 0,1
  0 = Passive signal
  1 = Active signal
Index range: 100 * line nr + signal no. Signal no. 5 = CTS, 8 = DCD, 20 = DTR
Access: No limitations

Writing this attribute may cause erroneous operation of the protocol.

**MI**

**Message Identification**

Object address of system messages.

Data type: Integer
Value: 1...32760
Index range: 1...8 (NET line numbering)
Default value: 6000 + (100 * NET number) + line number
Access: Read, conditional write
**MS Message Application**

The number of the application that is the receiver of the system messages generated by the line.

- **Data type:** Integer
- **Value:** 1... 32
- **Default value:** 1
- **Index range:** 1... 8 (NET line numbering)
- **Access:** Read, conditional write

**PM Protocol Mode**

The type of the used Modbus protocol.

- **Data type:** Integer
- **Value:** 0, 1
  - 0: RTU
  - 1: ASCII
- **Index range:** 1...8 (NET line numbering)
- **Default value:** 0
- **Access:** Read, conditional write

**DC Diagnostic Counters**

The line protocols gather statistical information about the events on the lines by incrementing a number of diagnostic counters. All the major events and error situations of the communication have their own counters.

When accessing diagnostic counters, the attribute is indexed according to the formula:

\[
100 \times \text{(line number)} + \text{(diagnostic counter number)}
\]

Modbus master protocol supports the following counters:

1. Transmitted telegrams
2. Failed transmissions
4. Transmitted commands
5. Transmitted replies
11. Received messages
12. Parity errors
13. Overrun errors
14. Check sum errors
15. Framing errors
16. Buffer overflow errors
Data type: Integer
Value: 0...30000
Index range: See above
Access: Read-only, the values can be reset

**Example configuration of Modbus Serial line**

The following example shows the Modbus serial line configuration:

![Modbus serial line configuration](image)

*Fig. 3.1.-1 Example of Modbus serial line configuration*

**Modbus TCP master attribute**

The following attribute can be used for configuring the Modbus master/TCP lines in MicroSCADA. The LD attribute is meaningful only if there are multiple IP addresses in the computer.
**LD Local Address**

The IP address, which is used locally. The setting of this attribute is necessary when the computer has multiple IP addresses. The addresses, which Modbus master line must use, are defined. The setting of this attribute must be done before the line is taken into use for the first time.

Value: String containing a valid IP address, max 29 characters
Access: Read/write

This attribute accepts the IP address in the following form:

```
#SET NET1:SLD1="62.236.144.120"
```

![View Attributes](image)

**DC Diagnostic Counters**

The line protocols gather statistical information about the events on the lines by incrementing a number of diagnostic counters. All the major events and error situations of the communication have their own counters.

When accessing diagnostic counters, the attribute is indexed according to the formula:

```
100 * (line number) + (diagnostic counter number)
```
Modbus master protocol supports the following counters:

1. Transmitted telegrams
2. Failed transmissions
4. Transmitted commands
5. Transmitted replies
11. Received messages
12. Parity errors
13. Overrun errors
14. Check sum errors
15. Framing errors
16. Buffer overflow errors
20. TCP connect
21. TCP accept
22. TCP close

Data type: Integer
Value: 0...30000
Index range: See above
Access: Read-only, the values can be reset

3.2. Modbus PLC device configuration

The PLC station device is the heart of the MicroSCADA Modbus protocol converter. It converts communication messages from MicroSCADA’s internal protocol to the Modbus protocol and vice versa. The PLC device stores necessary information of protocol and address conversion in topic data. The PLC device also stores data which is scanned from an external device to an internal PC-NET database. The purpose of this storing is to minimise the amount of messages between the base system and PC-NET (PC-NET only sends changed data to the base system). Changed data is sent to the base system process database as RTU process data.

**IU** **In Use**

Indicates whether the line is in use (value 1) or not in use (value 0).

Data type: Integer
Value: 0 or 1
Default value: 0
Access: No limitations

**LI** **Line Number**

The number of the NET line the station is connected to.

Data type: Integer
Value: 1...8
Default value: 1
Access: Read, conditional write

Setting this attribute is not needed when the station is created by using the DV attribute.

SA Station Address
The station address of the Modbus master station (ID of the slave).
Data type: Integer
Value: 0...255
Default value: 1
Access: Read, conditional write

AL Allocation
Allocates the station to an application. When the AL attribute has the value 1, the station is reserved by the application specified by the AS attribute. All the spontaneous messages from the station are sent to this application.
Data type: Integer
Value: 0 or 1
Access: No limitations

AS Allocating Application
Specifies the allocating application of the station (see the AL attribute). The allocating application gets all the spontaneous process data from the station. This application is also the only one that is allowed to set the device communication attributes.
Data type: Integer
Value: 0...32
0 = no application
Access: Read-only

MI Message Identification
Object address of the system messages.
Data type: Integer
Value: 1...32760
Default value: 28000 + station number
Access: Read, conditional write
MS  Message Application
The number of the application that is the receiver of the system messages generated by the line.
Data type: Integer
Value: 1...32
Default value: 1
Access: Read, conditional write

DC  Diagnostic Counters
The values of the diagnostic counters which the NET unit keeps for the station. The counters have the following meaning:
1. Suspension information (0 = OK, 1 = suspended)
2. Suspension counter
3. Transmitted data messages
4. Transmitted command messages
5. Transmitted confirmation messages
6. Received data messages
7. Received command messages
8. Received confirmation messages
9. Received unknown messages
Data type: Integer
Value: 1...65535
Index range: 1...20
Access: Read-only

An example configuration of a PLC device
The 8.4.3 (or newer) System Configuration Tool may be used for configuration since it supports the Modbus master protocol. The following figure shows an example configuration of a PLC device.
Fig. 3.2.-1 Example configuration of PLC device

The PLC device object is created by using the Base System Configuration Picture, as described below.

1. Create a PLC station type:

CREATE_OBJECT -> STATION_TYPES ->

- **STY NR:** 28
- **TYPE NAME:** PLC
- **DATABASE TYPE:** RTU
2. Create a PLC Station object:

CREATE_OBJECT -> STATIONS  ->

| STA NR:   | 1 |
| TRANS.TYPE: | EXTERNAL |
| NODE NUMBER: | 1 |
| TRANSL. OBJ. NUM: | 1 |
| STATION TYPE: | PLC |

**Modbus TCP/IP attributes**

**IA**  
**Internet Address**

The IP address or the host name of the remote host. The connection is established with a device in this address by using port number 502. The line must be taken into use at least once before writing to this attribute.

Value: Any string, max 29 characters
Access: Read/write

This attribute accepts the IP address in the following form:

```
#SET STA1:SIA="62.236.144.120"
```

or as an alias name

```
#SET STA1:SIA="GRACE"
```

When an alias name is used, it must be defined in the TCP host file

```
%windir\system32\drivers\etc\hosts.
```

**CT**  
**Connecting Timeout**

The maximum time of the TCP connect operation. The value of this attribute depends on the speed of LAN, remote station and the possible routers between MicroSCADA and the substation. The value should be lower than the HT attribute of the line, but high enough to enable reliable reconnecting of the substation. In a multidrop configuration, a value that is too high may cause communication disturbances, if some of the stations are not available.

Value: 0.60000
Unit: Milliseconds
Access: Read/Write
Default: 500 ms

**ET**  
**rEconnecting Timeout**

The interval of reconnecting attempt while communication is not established.
3.3. Topics of a PLC device

The scanning of external device data is controlled by the PLC device topics. Since the PLC device can contain several different topics (max. 100), it is possible to divide the memory and IO (input/output) into separate external device areas. These areas can contain different types of data and they can have individual scanning intervals.
3.3.1. **Topic parameters**

The topic parameters are stored in the PLC station topic parameter table. The memory needed by the topic is dynamically reserved, based on the address and format parameters. The parameter table is filled in by writing a vector to the TP attribute, which is also possible to read.

Syntax of a topic configuration command:

```
STA'n':STP'index'= (Allocation, FirstObjectAddress, LastObjectAddress, Type, BaseAddress, Format, Interval, delta)
```

The 8.4.3 (or newer) System Configuration Tool may be used for configuration, since it supports the Modbus master protocol. The following figure shows an example of the topic configuration under the PLC station type.

![Topic Configuration Editor](image)

**Fig. 3.3.1.-1 Topic configuration with System Configuration Tool**

The topic configuration vector contains the parameters below:

**Allocation**

This item specifies whether the topic is in use or not. The memory needed for the topic is reserved, when the topic is taken into use.

**FirstObjectAddress**

This parameter specifies the First MicroSCADA Process Object Address used with this topic. The object address and object type parameters specify together the actual process object address (OA), where the first item in the topic is stored. See below:

\[
OA = 4096 \times \text{Object_Type} + \text{FirstObjectAddress}
\]

This calculation is also done automatically in the process object tool. The user has to add the same address parameters as with the topic. When the actual object address is read, the calculated object address is shown instead of the parameter value.
One object address contains a 16 bits data field, for example 16 binary inputs in a row or one holding register value depending on the topic type. Address space for every object type is $4096 \times 16$ bits = 65536 bits. One topic can create several Modbus protocol requests.

**LastObjectAddress**

This parameter is the object address of the last topic item. The number of items reserved by the topic is calculated as shown below:

Number of items = LastObjectAddress - FirstObjectAddress

**Type**

This parameter specifies the data type of the process objects. The following table shows the possible data types of a PLC device.

<table>
<thead>
<tr>
<th>Type</th>
<th>Type of process object</th>
<th>Possible data formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>Object command</td>
<td>IO_BIT, M_BIT</td>
</tr>
<tr>
<td>3:</td>
<td>Digital set-point</td>
<td>INT, WORD</td>
</tr>
<tr>
<td>4:</td>
<td>Analog set-point</td>
<td>CHAR, INT, WORD, LONG, FLOAT</td>
</tr>
<tr>
<td>6:</td>
<td>Analog value</td>
<td>CHAR, INT, WORD, LONG, FLOAT</td>
</tr>
<tr>
<td>7:</td>
<td>Indication (single or double)</td>
<td>IO_BIT, M_BIT</td>
</tr>
<tr>
<td>8:</td>
<td>Pulse counter</td>
<td>LONG</td>
</tr>
<tr>
<td>9:</td>
<td>Digital value</td>
<td>INT, WORD</td>
</tr>
</tbody>
</table>

With the process object type “indication”, one object address (OA) contains 16 bits.

**Base Address**

This Modbus address is the first item address of the topic in the PLC memory. With binary indications the address space is 1 - 65536. With 16 bits registers it is 1 - 4096.

**Format**

This parameter specifies how data is stored in external devices. Possible formats are shown in the following table.

<table>
<thead>
<tr>
<th>Format</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO_BIT</td>
<td>1</td>
<td>Bit in PLC’s input or output.</td>
</tr>
<tr>
<td>M_BIT</td>
<td>2</td>
<td>Memory bit in PLC’s working memory.</td>
</tr>
<tr>
<td>CHAR</td>
<td>3</td>
<td>Unsigned 8 type object in PLC’s registers. A register can allocate two CHAR.</td>
</tr>
</tbody>
</table>
The most significant word - the least significant word (and vice versa)

Interval

This is the frequency with which topic data is read from an external device. The interval units are milliseconds. If the interval is 0, the topic is not polled.

Delta

If the topic type is an analog value (type=4), the delta value is used to minimise the amount of updating messages from the NET to the base system. The new analog value is sent to the base system, when the change or the sum (integral) of changes is bigger than the delta value.

An example configuration of data topics

The following example shows how to configure data topics of a PLC device.

```
#SET STA1:SIU=0
#SET NET1:SIU4=0

; INIDICATION BLOCK
; OA 28673..28673, IO_BIT, SINGLE INDICATION, BASE ADDRESS 1
#SET STA1:STP(1)=( 1,1,1,7,000,1,1000,0)

;DIGITAL VALUE BLOCK
```
3.3.2. Address conversion between RTU and Modbus

Addressing system of Modbus protocol

The Modbus protocol and Modicon PLCs divide addressable memory into four different areas as shown below.

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xxxx</td>
<td>Discrete outputs and discrete coils</td>
</tr>
<tr>
<td>1xxxx</td>
<td>Discrete inputs</td>
</tr>
<tr>
<td>3xxxx</td>
<td>Input registers</td>
</tr>
<tr>
<td>4xxxx</td>
<td>Holding registers</td>
</tr>
</tbody>
</table>

The external communication unit refers to these address areas by using different message functions as shown below.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Read coil status (0xxxx)</td>
</tr>
<tr>
<td>02</td>
<td>Read input status (1xxxx)</td>
</tr>
<tr>
<td>03</td>
<td>Read holding registers (4xxxx)</td>
</tr>
<tr>
<td>04</td>
<td>Read input registers (3xxxx)</td>
</tr>
</tbody>
</table>
Addressing systems of RTU process objects in MicroSCADA

The object address (OA) of the RTU type process objects is a packed 16 bit value, which contains the actual address and also the type of data as shown below:

\[ OA = 4096 \times \text{Object Type} + \text{Object Address} \]

Object Type is one of the object types described in the table below.

### Table 3.3.2-1 Object type codes and their process object types

<table>
<thead>
<tr>
<th>Object Type Code</th>
<th>Process object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>Object command</td>
</tr>
<tr>
<td>3:</td>
<td>Digital set-point</td>
</tr>
<tr>
<td>4:</td>
<td>Analog set-point</td>
</tr>
<tr>
<td>6:</td>
<td>Analog value</td>
</tr>
<tr>
<td>7:</td>
<td>Indication (single or double)</td>
</tr>
<tr>
<td>8:</td>
<td>Pulse counter</td>
</tr>
<tr>
<td>9:</td>
<td>Digital value</td>
</tr>
</tbody>
</table>

Addressing Modbus objects from MicroSCADA

This section describes the addressing of Modbus objects from MicroSCADA. The following table shows the relationship between the Modbus message functions and topic parameters.

### Table 3.3.2-2 Modbus message functions and topic parameters

<table>
<thead>
<tr>
<th>Function</th>
<th>Topic type</th>
<th>Topic format</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>7</td>
<td>M_BIT</td>
</tr>
<tr>
<td>02</td>
<td>7</td>
<td>IO_BIT</td>
</tr>
<tr>
<td>03</td>
<td>6, 8, 9</td>
<td>CHAR, WORD, INT, LONG, FLOAT</td>
</tr>
<tr>
<td>04</td>
<td>6, 9</td>
<td>IN_WORD, IN_CHAR, IN_INT, IN_LONG, IN_FLOAT, IN_MSB_LONG, IN_MSB_FLOAT</td>
</tr>
<tr>
<td>05</td>
<td>1</td>
<td>IO_BIT or M BIT</td>
</tr>
<tr>
<td>06</td>
<td>3, 4</td>
<td>CHAR, WORD, LONG, FLOAT</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>M BIT (only when writing to the DI attribute)</td>
</tr>
<tr>
<td>16</td>
<td>3, 4</td>
<td>CHAR, WORD, LONG, FLOAT (when vector written)</td>
</tr>
</tbody>
</table>

Topic type = Process object type
4. Using the Modbus Master Protocol

This chapter describes how to transfer data between MicroSCADA and an external device by using the Modbus protocol serial line or TCP network.

Requirements
The following software is required:

• MicroSCADA Software revision 8.4.1 or later version
• Operating system - Windows

You also need general knowledge about the Modbus protocols and PLC programming.

Install the software as described in their respective manuals. The installation of MicroSCADA software is described in the MicroSCADA installation manual.

4.1. Accessing Modbus data using process objects

The most straightforward way to read data from a PLC device is to use tools in the MicroSCADA package. The first step is to create topics to PC-NET, for example in the System Configuration Tool and then request data from the device. Creating topics is explained in Section 3.3.1.

Below is an example of a topic for indication data:

![Topic Configuration Editor](image)

*Fig. 4.1.-1  Example of topic configuration*
The requested data can be used by the MicroSCADA database, if the RTU type of the process objects are created. The Object Navigator Tool can be used to create these objects. The next example shows a process object suitable for the topic above.

![Process Object Configuration](image)

**Fig. 4.1.-2 Example of process object**

The used topic requests data in the Modbus address space from bit address 1 to 32 (2*16). One MicroSCADA block object address contains 16 bits. The process object in the example uses the first bit in the second block and the Modbus address is 17. For the rest of the binary data in the same topic, one process object must be created for every bit. With double binary, two consecutive bits are used and the process object type is the double binary RTU object.
Analog format data handling is similar, except that no bit handling is used. The used process object refers to the defined Modbus register address. For instance, the topic for analog input is defined in the object addresses 100-101 and the Modbus base address is 10. In that case, the process object with the address 100 shows data from the Modbus register address 10.

4.2. Configuring the device attribute interface

Another way to read data from a PLC device is to use the NET attribute interface. A PLC station contains own database based on the created topics. This database can be accessed via the attributes described below.

4.2.1. Attribute interface of a PLC device

The attribute interface of a PLC device is described in the following section.

Communication attributes

The base system communicates with the PLC by using communication attributes. The object of a communication attribute is the NET topic. When the base system writes (or reads) the communication attribute, the station process changes the read or written data to the PLC message. The PLC message is created by using the type of topic and attribute parameters. The written data is sent forward from the NET without storing the data in the internal database of the NET. The value which comes from the PLC by reading communication attributes is not stored in the NET database. The index for the communication attribute must fit some of the topics with the right topic type. The index must be between IndexFirst and IndexLast.

The communication attributes are the following:

**BV  Bit Value**

This attribute refers to a vector of bits in the PLC memory or IO. When this attribute is written, a command function 15hex (Force multiple coils) is issued to Modbus line. When this attribute is read, a single indication or a vector of single indications is returned. The BV attribute can be used only with topics in which the type is BIT. The indices for BV attribute are calculated as follows:

index = Object_Address * 16 + bit number (when reading from the BV attribute)

index = Object_Address (when writing to the BV attribute)

Access: R/W

Index range: 0..65535

Value range: 0..1

Example of writing

#SET STA1:SBV(1..3)=(1,1,1)  
; issues a 'Force multiple coils' command to addresses 1..3  
(requires object command topic)

Example of reading

@RET=STA1:SBV(16..31)  
; returns a vector of single indications from address 1, bits 0..15  
(requires indication topic)
MICROSCADA PRO
MODBUS MASTER PROTOCOL
CONFIGURATION MANUAL

**SI**  
**Single Indication**
This attribute refers to a single bit in the PLC memory or IO. Writing to this attribute changes the status of one bit in the PLC. The SI attribute is possible to use only with topics whose type is bit. The index for the SI attribute is calculated as follows:

index = Object_Address * 16 + bit number (when reading from the SI attribute)
index = Object_Address (when writing to the SI attribute)

Access: R/W
Index range: 0...65535
Value range: 0...1

**DI**  
**Double Indication**
This attribute refers also to the PLC bit memory or bit IO. Double indication has (as single indication) two logical states (0 and 1), but in the double indication both directions have their own bits (0 -> 01, 1->10). It is possible to use the DI attribute only with topics whose type is bit. The index for the DI attribute is calculated as follows:

index = Object_Address * 16 + bit number, when reading from the DI attribute
index = Object_Address, when writing to the DI attribute

The use of double indication needs support from the PLC application program, because the PLC program language does not directly support double indication data type.

Access: R/W
Index range: 0...65535
Value range: 0...3

**AV**  
**Analog Value**
This attribute refers to the register data in the PLC memory. The register is always a 16-bit word in the PLC memory, but the PLC program can use registers in 1, 2 or 4 byte format. The PLC program can also use successive registers in different formats. For technical reasons, this is not allowed for those register areas which are transferred to MicroSCADA. The whole register area, which is referred by one topic must have equal format (type parameter) as topic.

index = Object_Address

Access: R/W
Index range: 0..65535
Value range: 0...0xFFFFFFFF

**DV**  
**Digital Value**
This attribute refers to the register data of the PLC memory. The register is used as a 16-bit word.

index = Object_Address
GD  General Request of Data
This attribute is not exactly a communication attribute. It is a request to send all the internal PLC NET database data to MicroSCADA. It does not perform any communication between the NET and a remote PLC. When data is sent, the PLC station sends the system’s status message to MicroSCADA. When reading, the value of GD is 1 until the updating is ready.

Access: R/W
Index range: 0...65535
Value range: 0...65535

AD  Additional Data
This attribute should be read when the response vector to the previous reading of the AV, DV, DI or SI attributes is shorter than expected.

The response to the AD attribute can be a vector of items of the same type as the previous response to the reading of the AV, DV, SI or DI attributes.

The response to the AD reading is formed, if the answer from the device does not fit into one internal message. No new request is made to the device when AD is read.

If a new AV, DV, DI or SI request is made or AD is read, the stored AD response is cleared.

Error 13832 PLCC_NO_ADDITIONAL_DATA_AVAILABLE is returned to SCIL, if no data is stored.

Indexing: No
Access: Read only
Value range 0..0xFFFFFFFF, range depends on the preceding request

Example:
Attribute read STA5:SDV(3001..3125) returns a vector of 114 words. Because this is less than expected, the SCIL application should branch to make an additional read of the AD attribute (STA5:SAD, no indexing). This returns a vector of 11 words. These two vectors may be concatenated by the SCIL application if needed.

The application should not make a new read of DV, AV, SI or DI before the AD attribute is read, otherwise the data for AD is lost. This can be a problem in a rare occasion of multiple SCIL instances accessing the same STA object at the same time.
4.2.2. **Example of using device interface commands**

The MicroSCADA application program can write data to an external device through the process database (MicroSCADA generates a process message automatically, when the output type of an RTU process object is updated). Another method to send commands from MicroSCADA is to use communication attributes as described in the following section.

**Writing object commands**

Object commands (e.g. switching device open/close commands) are sent as control relay output block messages. This message is a multi-purpose command. This section gives an example on how to write object commands.

**Topic configuration**

\[ \#SET \text{STA1:STP}(5)=(1,1,30,1,0,1,0,0) \]

**Process object configuration**

Name: PLC\_1\_OC  
Station: 1  
Index: 2  
OA: 2  
Type: Object command

The following commands set the same binary output (OUT\_BIT1) in the PLC.

\[ \#SET \text{PLC}\_1\_OC:P2 = 1 \]
\[ \#SET \text{STA1:SSI}(2) = 1 ;-> \text{index = object\_address} \]

**Writing analog setpoints**

**Process object configuration**

Name: PLC\_1\_AS  
Station: 1  
Index: 1  
OA: 1  
Type: Analog setpoint

If 16-bit values are used:

**Topic Configuration**

\[ \#SET \text{STA1:STP}(7)=(1,1,3,4,313,4,0,0) \]

The following commands set the register value 314 to 1234 in the PLC:

\[ \#SET \text{PLC}\_1\_AS:P1 = 1234 \]
\[ \#SET \text{STA1:SAV}(1) = 1234 \]

The following command updates a vector in the PLC (reg 314 = 1234, reg 315 = 1, reg 316=2)

\[ \#SET \text{STA1:SAV}(1..3) = (1234,1,2) \]

If 32-bit values are used:

**Configuration**

\[ \#SET \text{STA1:STP}(7)=(1,1,3,4,313,6,0,0) \]
The following commands sets the register value 314 to 1 and the register value 315 to 57920 in the PLC:

```
#SET PLC_1_AS:P1 = 123456
#SET STA1:SAV(1) =123456
```

If the topic format is 7 (MSB_LONG), the values in the registers are used in opposite order.

If float values are used, the configuration is:

```
#SET STA1:STP(7)=(1,1,3,4,313,8,0,0)
```

The following command sets the value 3.21 to registers 314, 315

```
#SET STA1:SAV(1)=3.21
```

**Writing digital setpoints**

Process object configuration:

Name: PLC_1_DS Station: 1
Index: 1 OA: 1
Type: Digital setpoint

**Topic configuration**

```
#SET STA1:STP(6)=(1,1,3,3,253,5,0,0)
```

The following commands set the register value 254 to 1234 in the PLC:

```
#SET PLC_1_DS:P1 = 1234
#SET STA1:SDV(1) =1234
```

Updating a vector (reg 254 = 1234, reg 255 = 1, reg 256=2)

```
#SET STA1:SDV(1..3) = (1234,1,2)
```

**Configuration examples**

As previously described, the NET continuously scans the PLC memory. Only data changes are sent to MicroSCADA. The PLC station sends data to the MicroSCADA process database.

The process database object address is calculated as:

\[ OA = ((PLC\_address - Topic\_Base\_Address) + StartOA) + 4096\times Topic\_Type \]

It is also possible to read data from the PLC by using communication attributes as described in the next section.

**Reading indications**

Process object configuration:

Name: PLC_1_SI Station: 1
Index: 1 OA: 1
Type: Indication OB: 0
Topic configuration

Topic configuration

#SET STA1:STP(1)=( 1,1,1,7,0,1,1000,0)

Scanning

The NET reads the state of 16 input bits starting from BIT0 with the interval of 1 second. The process value PLC_1_SI:P1 is updated, if the state of BIT0 of PLC is changed.

The direct reading of the INPUT BIT0 state is possible with the following SCIL command:

@BIT = STA1:SSI(16) ;-> index = 16*object_address + bit_address

and reading vector (16 values):

@BIT_V = STA1:SSI(16..31)

Reading analog values

Process object configuration

Name: PLC_1_AV Station: 1
Index: 1 OA: 1
Type: Analog value

Topic configuration

#SET STA1:STP(3)=(1,1,20,6,199,4,10000,10)

Scanning

The NET unit reads 20 registers from the PLC starting from register 200 with the interval of 10 seconds. The process value PLC_1_AV:P1 will be updated, if the register 200 value is changed.

The direct reading of the register 200 value is possible with the following SCIL command:

@VALUE = STA1:SAV(1)

and reading a vector:

@VALUE = STA1:SAV(1..20)

Reading digital values

Process object configuration

Name: PLC_1_DV Station: 1
Index: 1 OA: 1
Type: Digital value

Configuration

#SET STA1:STP(2)=(1,1,2,9,399,4,10000,000)
Scanning

The NET reads two registers from the PLC memory starting from register 400 with the interval of 1 second.

The process value PLC_1_DV:P1 is updated, if the register 400 is changed. The following SCIL command reads the value of register 400 from the PLC:

@VALUE = STA1:SDV(1)

and reading a vector:

@VALUE = STA1:SAV(1..2)

Reading digital values from input registers

Process object configuration

Name: PLC_1_IR Station: 1
Index: 1 OA: 1
Type: Digital value

Configuration

#SET STA1:STP(2)=(1,1,2,9,399,10,1000,0)

Scanning

The NET reads two input registers from the PLC memory starting from register 400 with the interval of 1 second.

The process value PLC_1_DV:P1 is updated, if the register 400 is changed. The following SCIL command reads the value of input register 400 from the PLC:

@VALUE = STA1:SDV(1)

4.2.4. Modbus TCP/IP protocol configuration examples

A setup of one station:

@NET=2

@STA = 1
@LINE = 1

#SET NET'NET':SPO'LINE'=25
#SET NET'NET':SDV(28)=('STA','LINE')
#SET NET'NET':SSD'LINE'="TCP"
#SET NET'NET':SEN'LINE'=3
#SET NET'NET':SLD'LINE'="62.236.144.119"
#SET NET'NET':SMS'LINE'='NET'
#SET NET'NET':SPM'LINE'=0
#SET NET'NET':SIU'LINE'=1
#SET NET'NET':SIU'LINE'=0

#SET STA'STA':SAL=1
#SET STA'STA':SSA=1
#SET STA'STA':SIA="GRACE"
#SET STA'STA':SET=3
#SET STA'STA':SIU=1
A setup of three stations:

@NET=2
@LINE = 1

#SET NET'NET':SPO'LINE'=25
#SET NET'NET':SDV(28)=(1,'LINE')
#SET NET'NET':SDV(28)=(2,'LINE')
#SET NET'NET':SDV(28)=(3,'LINE')
#SET NET'NET':SSD'LINE'="TCP"
#SET NET'NET':SEN'LINE'=3
#SET NET'NET':SMS'LINE'='NET'
#SET NET'NET':SPM'LINE'=0
#SET NET'NET':SIU'LINE'=1
#SET NET'NET':SIU'LINE'=0

@STA = 1

#SET STA'STA':SAL=1
#SET STA'STA':SSA='STA'
#SET STA'STA':SIA="GINA"
#SET STA'STA':SIU=1

@STA = 2

#SET STA'STA':SAL=1
#SET STA'STA':SSA='STA'
#SET STA'STA':SIA="LIZ"
#SET STA'STA':SIU=1
@STA = 3

#SET STA'STA':SAL=1
#SET STA'STA':SSA='STA'
#SET STA'STA':SIA="GRACE"
#SET STA'STA':SIU=1

#SET NET'NET':SIU'LINE'=1

TCP additions for LAN/WAN network

MicroSCADA Modbus TCP uses the default IP address provided by the operating system.

The created Modbus TCP line reserves a port number

2501+linenumber

for its internal use.

In TCP/IP mode (connection-oriented), the connection is established to port

502

do the slave device. The slave device must accept connections from this port. If not explicitly specified, only one connection to each Modbus TCP server is established at the same time. The second example below describes the gateway configuration. The IP address of the slave is configured with IA-attribute of the station object.
Example topologies:

**Fig. 4.2.-1** MicroSCADA as Modbus TCP master in multidrop environment

<table>
<thead>
<tr>
<th>Modbus TCP Line</th>
<th>STA1</th>
<th>STA2</th>
<th>STA3</th>
<th>STA4</th>
<th>IP 62.236.145.130</th>
</tr>
</thead>
<tbody>
<tr>
<td>; Line SD = &quot;TCP&quot;; connection-oriented mode</td>
<td>; connection made to port 502 of each IP-address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Station 1 IA = 62.236.145.119</td>
<td>Station 2 IA = 62.236.145.120</td>
<td>Station 3 IA = 62.236.145.121</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MicroSCADA**

**LAN/WAN**

**Station 1 (IED, RTU...)**
- IP-address 62.236.145.119
- Listens to port 502

**Station 2 (IED, RTU...)**
- IP-address 62.236.145.120
- Listens to port 502

**Station 3 (IED, RTU...)**
- IP-address 62.236.145.121
- Listens port 502

**Fig. 4.2.-2** MicroSCADA as Modbus TCP master in multidrop environment with gateway

**RS-485**

**Station 1 (IED, RTU...)**
- SA = 1

**Station 2 (IED, RTU...)**
- SA = 2

**Station 3 (IED, RTU...)**
- SA = 3
Appendix. Serial cable wiring diagram

When connecting the Modbus master to a MicroSCADA slave using a direct serial cable, the wiring illustrated by the following figure can be used:

![Serial cable wiring diagram](image.png)

**Fig. 5.-1 Serial cable wiring diagram**
6. Index

A
Accessing Modbus data ................................................................. 31
AD ................................................................. 22, 35
Additonal Data ................................................................. 35
Address conversion ................................................................. 20, 29
AL ................................................................. 21
Allocating Application ......................................................... 21
Allocation ................................................................. 21, 26
Analog format ................................................................. 33
Analog values ................................................................. 28, 38
Application Backup ......................................................... 9
AS ................................................................. 21
AV ................................................................. 34

B
Backup copies ................................................................. 9
Base Address ................................................................. 27
Basic line attributes ................................................................. 12
Baud Rate ................................................................. 13
Binary output ................................................................. 36
Bit handling ................................................................. 33
Bit Value ................................................................. 33
Block object address ................................................................. 32
BR ................................................................. 13
Buffer Pool Size ................................................................. 13
BV ................................................................. 33

C
Communication attributes ......................................................... 33
connecting Timeout ................................................................. 24
CT ................................................................. 24
CTS ................................................................. 15

D
Data transmission attributes ......................................................... 13
DC ................................................................. 17, 19, 22
DE ................................................................. 15
Delta ................................................................. 28
Device attribute interface ......................................................... 33
DI ................................................................. 30, 34
Diagnostic Counters ......................................................... 17, 19, 22
Digital Value ................................................................. 34
Double Indication ......................................................... 34
DV ................................................................. 21, 34
<table>
<thead>
<tr>
<th>Letter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>EN Enquiry Limit</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>ET</td>
<td>24</td>
</tr>
<tr>
<td>F</td>
<td>FirstObjectAddress</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Float values</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Format</td>
<td>27</td>
</tr>
<tr>
<td>G</td>
<td>GD General Request of Data</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>General strategy</td>
<td>6</td>
</tr>
<tr>
<td>H</td>
<td>Header Timeout</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>HT</td>
<td>15</td>
</tr>
<tr>
<td>I</td>
<td>IA In Use</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Indication data</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Intelligent Electronic Devices (IEDs)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Internet Address</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Interval</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>IU</td>
<td>12, 20</td>
</tr>
<tr>
<td>L</td>
<td>LastObjectAddress</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>LD</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>LI</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Line attributes</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Line Number</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Local Address</td>
<td>19</td>
</tr>
<tr>
<td>M</td>
<td>Message Application</td>
<td>17, 22</td>
</tr>
<tr>
<td></td>
<td>Message Identification</td>
<td>16, 21</td>
</tr>
<tr>
<td></td>
<td>MI</td>
<td>16, 21</td>
</tr>
<tr>
<td></td>
<td>MicroSCADA database</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Modbus TCP master attribute</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Modbus TCP master protocol</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Modbus TCP/IP attributes</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Modem Signal</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Modicon</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>MS</td>
<td>17, 22</td>
</tr>
<tr>
<td></td>
<td>Multi-drop network topology</td>
<td>11</td>
</tr>
</tbody>
</table>
MicroSCADA Pro
Modbus Master Protocol
Configuration Manual

N
NET attribute interface .............................................................................33

O
OA ......................................................................................................26, 30
Object commands .....................................................................................36
Object Navigator Tool ..............................................................................32

P
Parity .........................................................................................................13
PC-NET ................................................................................................6, 31
PD .............................................................................................................14
PLC .......................................................................................................5, 34
PLC device ...............................................................................................31
PLC station ...............................................................................................33
PM ............................................................................................................17
PO .............................................................................................................12
Point-to-point network topology ..............................................................11
Polling Delay ............................................................................................ 14
Process object ........................................................................................... 32
Protocol .....................................................................................................12
Protocol master ...........................................................................................6
Protocol Mode .......................................................................................... 17
PS ..............................................................................................................13
PY ..............................................................................................................13

R
RD .............................................................................................................14
Read data ..................................................................................................31
Receive Interrupt Enable Delay ................................................................15
Receiver Data Bit Count ...........................................................................14
rEconnecting Timeout ..............................................................................24
Register address .......................................................................................23
Remote Terminal Units (RTUs) .................................................................5
Request data ..............................................................................................31
Response Timeout ....................................................................................15
RI ..............................................................................................................15
RTS signal ................................................................................................15
RTU ..........................................................................................................32
RTU addresses ..........................................................................................6

S
SA ...............................................................................................................21
SB .............................................................................................................14
SD .............................................................................................................13
SG .............................................................................................................16
SI .............................................................................................................34
Single
Bit ....................................................................................................... 34
Indication ......................................................................................... 34
Station Address ............................................................................... 21
Status codes .................................................................................... 10
Stop Bits .......................................................................................... 14
System
  Backup ............................................................................................ 9
  Messages ....................................................................................... 16
System Configuration Tool ............................................................... 26, 31
System Device Name ....................................................................... 13

T
TD ................................................................................................. 14
TI ................................................................................................. 15
Topic
  Configuration .............................................................................. 26
  Parameters .................................................................................. 26
TP .................................................................................................. 26
Transmitter Data Bit Count ............................................................... 14
Type .............................................................................................. 27

W
Wiring ............................................................................................ 43
Writing object commands ................................................................. 36