InTouch® Compact Edition

MODBU Configuration Guide

Invensys Systems, Inc.

Revision A

Last Revision: 1/10/08

Copyright

© 2008 Invensys Systems, Inc. All Rights Reserved.

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

http://www.wonderware.com

For comments or suggestions about the product documentation, send an e-mail message to productdocs@wonderware.com.

Trademarks

All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DT Analyst, Factelligence, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A^2, InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, MaintenanceSuite, MuniSuite, QI Analyst, SCADAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, Wonderware Factelligence, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

Contents

Welcome 5

Documentation Conventions 5

Technical Support 6

Chapter 1 About the MODBU Driver 7

About MODBU Driver 7

Device Specifications 7

Network Specifications 8

Chapter 2 Configuring the MODBU Driver 9

Selecting the PLC Driver 10

Configuring the Connection Type 11

Configuring a Serial Connection 11

Configuring a TCP/IP Connection 12

Configuring a UDP/IP Connection 14

Configuring PLC-specific Settings 15

Configuring Advanced Settings for Serial Connection 17

Configuring Time-out Settings 18

Configuring Protocol Settings 19

Configuring Hand Shake Settings 20

Configuring Buffer Length Settings 21

Disabling and Enabling DTR and IR Settings 22

Configuring Advanced Settings For TCP/IP or UDP/IP Connection 22

Configuring Time-out Settings 23
Configuring Protocol Settings 24
Configuring Buffer Length Settings 25
Associating a PLC Driver with an Access Name 26

Chapter 3 Troubleshooting the MODBU Driver 29

Appendix A MODBU Item References 35

Data Types and Address Ranges 36
Device Registers 38
Other Type Settings 39
Register-Number Item Names 40
TSX Quantum and TSX Momentum 40
TSX Quantum 40
Generic Modbus (4, 5 and 6-digit) 41
Item Names Using Modicon PLC Register Addresses . 42
Absolute Notation Item Names 43
Modulo-10000 Point Item Names 44
Controller Function Codes 44

Index 47

Welcome

This guide shows you how to configure the MODBU driver for InTouch Compact Edition.

This guide assumes you know how to use Microsoft Windows, including navigating menus, moving between applications and moving objects on the screen. If you need help with these tasks, see the Microsoft online help.

You can view this document online or you can print it, in part or entirely, by using the print feature in Adobe Acrobat Reader.

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
Bold	Menus, commands, dialog box names, and dialog box options.
`Monospace`	Code samples and display text.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact Technical Support for help, have the following information ready:

- The type and version of the operating system you are using.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

About the MODBU Driver

This section explains the concepts of the MODBU driver and identifies all hardware and software components required to implement communication between InTouch Compact Edition and compatible PLCs.

About MODBU Driver

You can use the MODBU Driver to enable InTouch Compact Edition applications driver communicate with remote devices using the Modbus protocol.

Device Specifications

To establish communication with the MODBU driver, your target PLC must meet the following specifications:

- Manufacturer: Modicon (or any other device using the Modbus protocol for serial communication)
- Compatible equipment:
  - AEG CPU 984 series
  - AEG CPU compact
  - Siemens S7 S200 (using a special ladder program to emulate the Modbus protocol on the free port)
  - Any device that is compatible with the Modbus protocol
- Programmer software**:** ModSoft

Network Specifications

To communicate with the target device, the network must meet the following specifications:

- Device communication port: Modbus port (AEG 985E) and Free Port (S7-S200)
- Physical protocol: RS232
- Logic protocol**:** Modbus
- Device run-time software**:** None
- Specific PC board**:** None
- Cable wiring scheme**:** See PLC manufacturer documentation.

Chapter 2

Configuring the MODBU Driver

This section explains how to configure the MODBU Driver by completing the following tasks:

- Select the PLC driver
- Configure the connection type
- Configure other PLC-specific settings
- Configure advanced settings
- Associate an InTouch Access Name with the configured driver

You configure the MODBU driver from the **Driver Configuration** dialog box. In a managed InTouch compact application, you access this dialog box from the WindowMaker **Special**, **InTouch Compact Edition Application**, **IO Driver Configuration** menu.

For more information about configuring connection settings, see the documentation of your PLC hardware manufacturer.

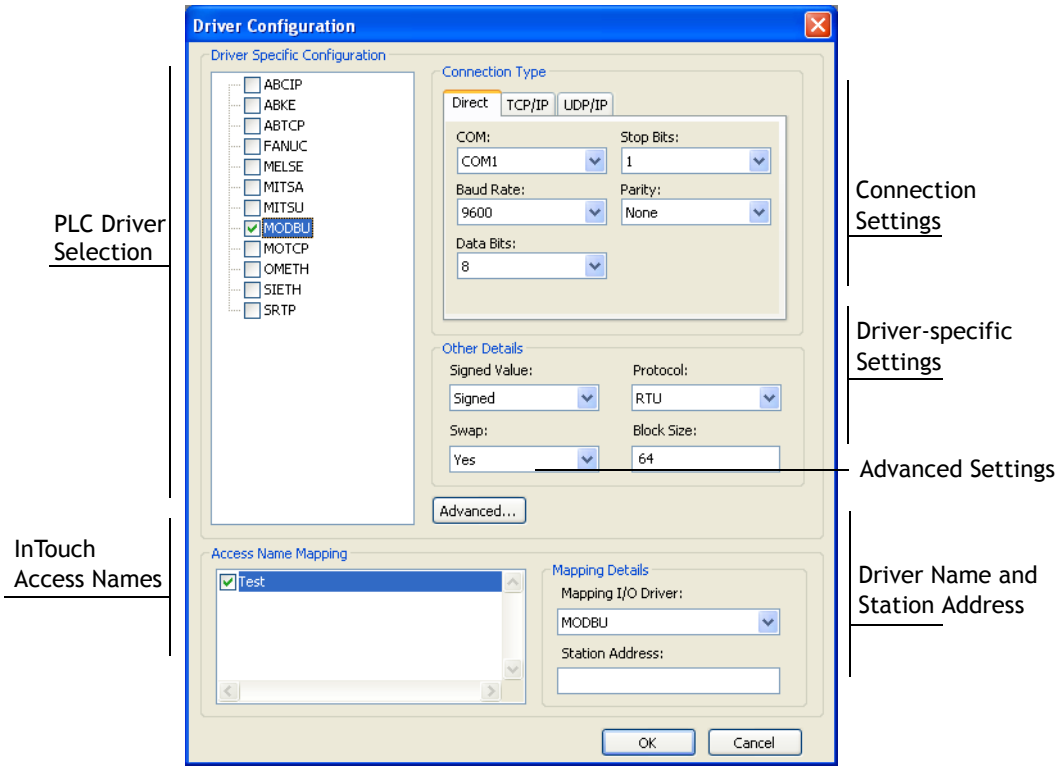
Note For each PLC type, you can configure only one set of driver parameters.

Selecting the PLC Driver

Before you configure the PLC driver, you need to select it in the **Driver Configuration** dialog box. You can select more than one PLC driver to configure. All selected PLC drivers can be associated with InTouch Access Names.

To select the PLC driver

1. In WindowMaker, click **Special** from the menu bar, point to **InTouch compact Edition Application** and then click **I/O Driver Configuration**. The **Driver Configuration** dialog box appears.



2. From the list in the left pane, select the check box next to the PLC driver you want to configure.

   The selected PLC driver appears in the **Mapping I/O Driver** list in the **Mapping Details** area of the **Driver Configuration** dialog box. You can associate it with an InTouch Access Name. For more information, refer to Associating a PLC Driver with an Access Name on page 26.

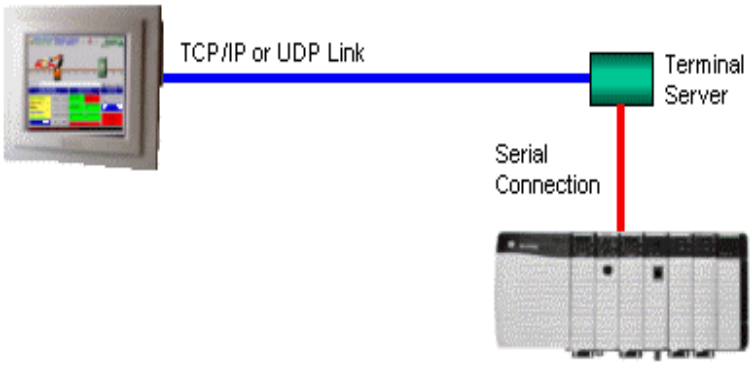
You can now configure the connection settings, advanced settings, and other PLC-specific settings.

Configuring the Connection Type

You need to configure the connection type that the MODBU driver uses to communicate with a PLC.

The serial driver allows the use of TCP/IP or UDP/IP encapsulation. The encapsulation mode provides communication with serial devices connected to terminal servers on Ethernet or wireless networks. A terminal server converts TCP/IP or UDP/IP messages on Ethernet or wireless networks to serial data. After the message is converted to a serial form, you can connect standard devices that support serial communications to the terminal server.

The following diagram provides an example of this solution.



Note See the manufacturer documentation to configure gateway for TCP/IP and UDP/IP mode.

You can configure the following connection types:

- Direct serial
- TCP/IP
- UDP/IP

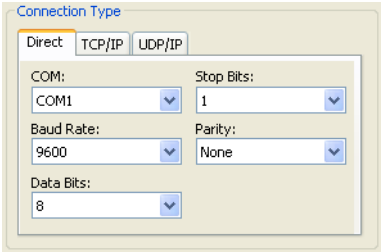
Configuring a Serial Connection

If the panel is directly connected to the PLC, you need to configure the following settings:

- COM port
- Stop bits
- Baud rate
- Parity
- Data bits

To configure a serial connection

1. In the **Driver Configuration** dialog box, select the driver you want to configure.
2. In the **Connection Type** area, click the **Direct** tab.



3. Assign values to the following settings:
   - In the **COM** list, select the serial communication port.
   - In the **Stop Bits** list, select the number of stop bits.
   - In the **Baud Rate** list, select the communication speed.
   - In the **Parity** list, select the parity.
   - In the **Data Bits** list, select the number of bits per character or code word.
4. Click **OK**.

Configuring a TCP/IP Connection

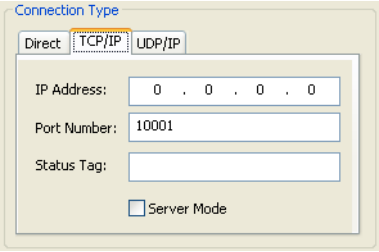
If the panel uses a TCP/IP connection to communicate with the PLC, you need to configure the following TCP/IP settings:

- IP Address
- Port Number
- Status Tag
- Server Mode

Note When **Server mode** is enabled, the driver listens to the TCP/IP port specified instead of trying to establish a connection (the driver is a TCP/IP server). The remote PLC/Device is responsible for connecting to the Wonderware Compact Panel. After TCP/IP connection has been made, all the data is sent through it. When **Server Mode** is enabled, the IP Address field is not used because the driver does not try to connect with the PLC

To configure a TCP/IP connection

1. In the **Driver Configuration** dialog box, select the driver you want to configure.
2. In the **Connection Type** area, click the **TCP/IP** tab.



3. Configure the following settings:
   - In the **IP Address** box, type the IP address of the terminal server that supports serial connection.
   - In the **TCP Port Number** box, type the port number assigned to the terminal server that supports serial connection.
   - In the **Status Tag** box, type the name of an InTouch discrete tag. The tag value is set to True if the TCP/IP connection is established, otherwise False.
   - Select **Server Mode**, if the remote PLC is responsible for connecting to the Wonderware Compact Panel.

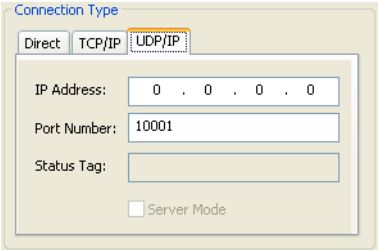
Configuring a UDP/IP Connection

If the panel uses a UCP/IP connection to communicate with the PLC, you need to configure the following settings:

- IP address
- Port number

To configure a UDP/IP connection

1. In the **Driver Configuration** dialog box, select the driver you want to configure.
2. In the **Connection Type** area, click the **UDP/IP** tab.



3. Configure the following settings:
   - In the **IP Address** box, type the IP address of the terminal server that supports serial connection.
   - In the **UDP Port Number** box, type the port number assigned to the terminal server that supports serial connection.
4. Click **OK**.

Configuring PLC-specific Settings

This section shows how to configure settings that are specific to the MODBU Driver.

The following table describes communication settings and their values.

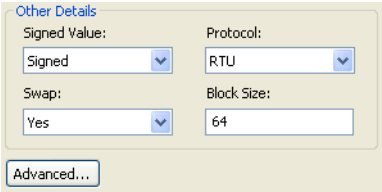
Setting	Default Value	Valid Values	Description	
Signed / Unsigned	Unsigned	Unsigned (legacy)	You should not use these options because they are just to keep compatibility with older versions.	This field just works using 3X, 4X, DW, DWS, DW3, DW3S or ID register types.
		Signed (legacy)		
		Unsigned	Does not allow negative values. INTEGER - 0 to 65535. DWORD - 0 to 4294967295.	
		Signed	Allow negative values INTEGER - -32768 to 32767 DWORD - -2147483648 to 2147483647	
Protocol (ASCII or RTU)	RTU	ASCII	Each eight-bit word is sent as two four-bit ASCII characters, allowing for a time interval between characters without causing errors.	
		RTU	Each eight-bit word is sent as two four-bit hexadecimals, allowing for greater density and faster throughput. In most cases, we recommend using this protocol.	

Setting	Default Value	Valid Values	Description
Swap (Yes or No)	Yes	No swap or swap	This option enables or disables the Swap Word order in the reading and writing commands. • **No Swap** - disable swap word order • **Swap** - enable swap word order
			This parameter will work with the FP, FPS, FP3, FP3S, DW, DWS, DW3, and DW3S register types.
		Write item or write group	This option chooses the function 0x06 or 0x10 to write. **Write Item** • Write on tag change (Standard Driver Sheet) and Write (Main Driver Sheet) use the Modbus function 0x06. • Write Trigger (Standard Driver Sheet) uses the Modbus function 0x10. **Write Group** • Any writing in the driver uses the Modbus function 0x10.
			• Modbus function 0x06 (Preset Single Register) • Modbus function 0x10 (Preset Multiple Registers) For more information about the functions consult the protocol guide
Block Size (ERO-xxx)	64	0 to 512	• The nominal size (in words) of each block of data to be transmitted, as determined by the processing capacity of the device. • A block size of 1 configured will send 1 item
		ERO-xxx	Address used to set equipment to local or remote (used for ERO equipment only).

Note For more information about configuring MODBU Driver-specific settings, see the documentation of your PLC hardware manufacturer.

To configure MODBU-specific settings

1. In the **Driver Configuration** dialog box, select the driver you want to configure.



2. In the **Other Details** area, assign values to the following settings:
   - **Signed Value**
   - **Protocol**
   - **Swap**
   - **Block Size**
3. Click **OK**.

Configuring Advanced Settings for Serial Connection

This section explains how to configure advanced settings for a MODBU driver using serial a connection. You can configure the following settings:

- Time-out settings
- Protocol settings
- Hand Shake settings
- Buffer Length settings
- Setting DTR and IR

  Depending on the PLC driver you select, one or more of these settings might be disabled.

Note For more information about configuring advanced settings, see the documentation of your PLC hardware manufacturer.

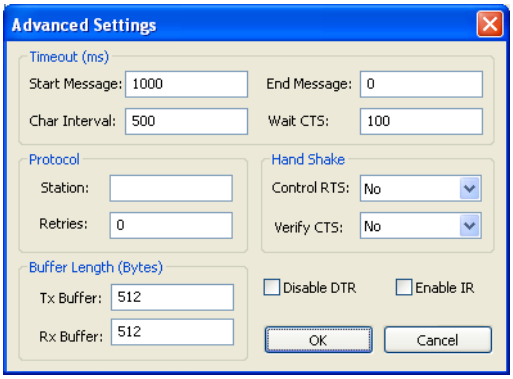
Configuring Time-out Settings

You can configure the following time-out settings:

- Start Message
- End Message
- Interval between characters
- Clear to Send wait

To configure time-out settings

1. In the **Driver Specific Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced**. The **Advanced Settings** dialog box appears.



3. In the **Timeout** area, assign values to the following settings:
   - **Start Message** - Time-out period in milliseconds for the message to start.
   - **End Message** - Time-out period in milliseconds for the message to end.
   - **Char Interval** - Time-out period in milliseconds between each character.
   - **Wait CTS** - Time-out period in milliseconds for the Clear to Send wait.
4. Click **OK**.

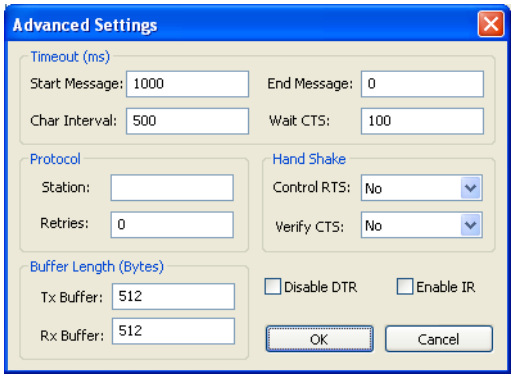
Configuring Protocol Settings

You can configure the following protocol settings:

- Station slave address
- Number of retries

To configure Protocol settings

1. In the **Driver Specifc Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced**. The **Advanced Settings** dialog box appears.



3. In the **Protocol** area, assign values to the following settings:
   - **Station** - Specify the slave network address.
   - **Retries** - Specify how many times the driver attempts to execute the same communication command before reporting a communication error.
4. Click **OK**.

Configuring Hand Shake Settings

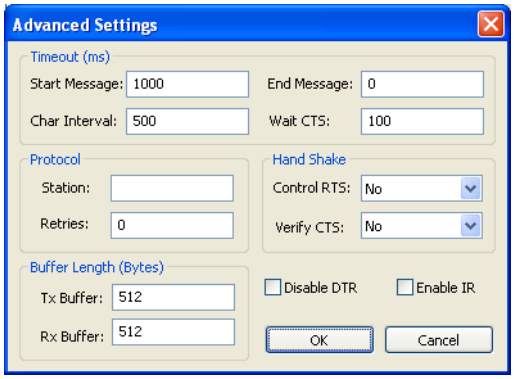
You can configure the following Hand Shake settings:

- Request to Send control
- Clear to Send verification

Hand Shake settings determine how the PLC driver communicates with the PLC over a communication channel after the physical establishment of the channel and before the normal information transfer.

To configure Hand Shake settings

1. In the **Driver Specific Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced**. The **Advanced Settings** dialog box appears.



3. In the **Hand Shake** area, assign values to the following settings:
   - **Control RTS** - Specify whether to use the Request to Send control.The following options are available:

     No: Do not set the RTS (Request to Send) Hand Shake signal. Important: If you are using Windows 95/98 or Windows CE with the correct RS232/RS485 adapter (i.e. without RTS control), then you must select this option.

     Yes: Set the RTS (Request to Send) Hand Shake signal before communication. Important: If you are using Windows NT and the Cutler-Hammer RS232/RS485 adapter, then you must select this option.

     Yes+echo: Set the RTS (Request to Send) Hand Shake signal before communication, and echo the signal received from the target device.

     The default value is No.
   - **Verify CTS** - Specify whether to use the Clear to Send type of verification.

The default value is No.

4 Click **OK**.

Configuring Buffer Length Settings

You can configure the following Buffer Length settings:

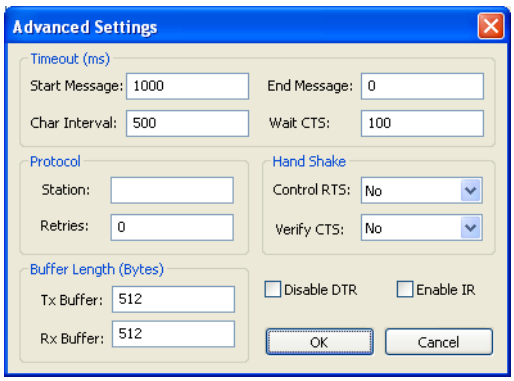
- Transmission Buffer (Tx) Length
- Reception Buffer (Rx) Length

The Buffer Length settings determine the number of bytes allocated from memory as a temporary buffer. The driver stores data in the temporary buffer while communicating with the PLC.

To configure Buffer Length settings

1 In the **Driver Specific Configuration** dialog box, select the driver you want to configure.

2 Click **Advanced**. The **Advanced Settings** dialog box appears.



3 In the **Buffer Length (Bytes)** area, assign values to the following settings:

- **Tx Buffer** - Transmission buffer size in bytes.
- **Rx Buffer** - Reception buffer size in bytes.

4 Click **OK**.

Disabling and Enabling DTR and IR Settings

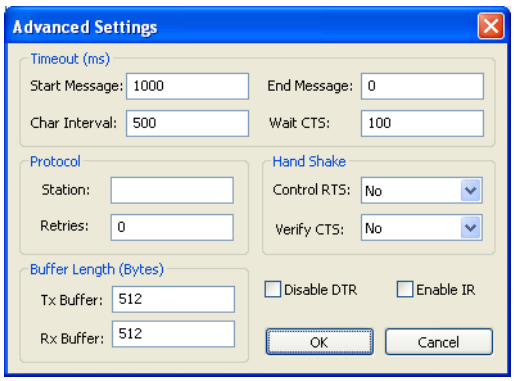
You can enable or disable the DTR and IR settings.

Setting DTR (Data Terminal Ready) means that the panel sends an additional signal to the PLC to indicate that the panel is ready to receive data.

Setting IR means that an infrared interface is used instead of a standard serial port to communicate with the device.

To enable or disable DTR and IR Settings

1. In the **Driver Specific Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced**. The **Advanced Settings** dialog box appears.



3. Assign values to the following settings:
   - Select or clear the **Disable DTR** check box.
   - Select or clear the **Enable IR** check box.
4. Click **OK**.

Configuring Advanced Settings For TCP/IP or UDP/IP Connection

This section explains how to configure advanced settings for a MODBU driver using TCP/IP or a UDP/IP connection. You can configure the following:

- Time-out settings
- Protocol settings
- Buffer Length settings

  Depending on the PLC driver you select, one or more of these settings might be disabled.

Note For more information about configuring advanced settings, see the documentation of your PLC hardware manufacturer.

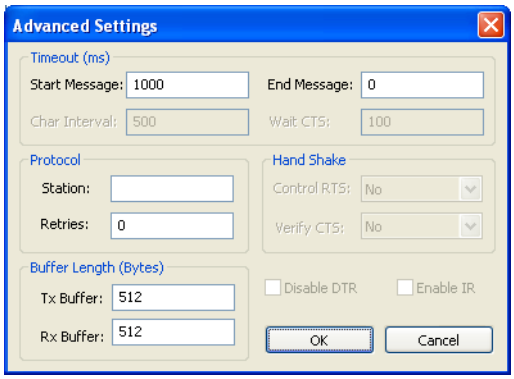
Configuring Time-out Settings

You can configure the following time-out settings:

- Start Message
- End Message

To configure time-out settings

1. In the **Driver Specific Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced**. The **Advanced Settings** dialog box appears.



3. In the **Timeout** area, assign values to the following settings:
   - **Start Message** - Time-out period in milliseconds for the message to start.
   - **End Message** - Time-out period in milliseconds for the message to end.
4. Click **OK**.

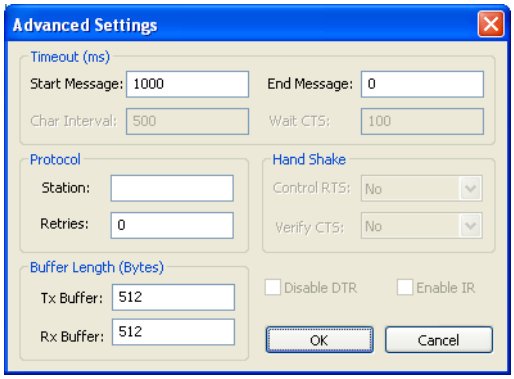
Configuring Protocol Settings

You can configure the following protocol settings:

- Station slave address
- Number of retries

To configure Protocol settings

1. In the **Driver Specific Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced.** The **Advanced Settings** dialog box appears.



3. In the **Protocol** area, assign values to the following settings:
   - **Station** - Specify the slave network address.
   - **Retries** - Specify how many times the driver attempts to execute the same communication command before reporting a communication error.
4. Click **OK**.

Configuring Buffer Length Settings

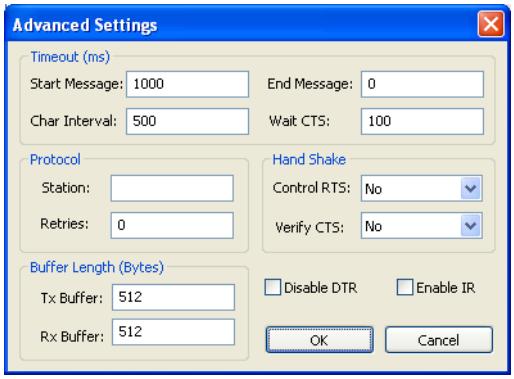
You can configure the following Buffer Length settings:

- Transmission Buffer (Tx) Length
- Reception Buffer (Rx) Length

The Buffer Length settings determine the number of bytes allocated from memory as a temporary buffer. The driver stores data in the temporary buffer while communicating with the PLC.

To configure Buffer Length settings

1. In the **Driver Specific Configuration** dialog box, select the driver you want to configure.
2. Click **Advanced**. The **Advanced Settings** dialog box appears.



3. In the **Buffer Length (Bytes)** area, assign values to the following settings:
   - **Tx Buffer** - Transmission buffer size in bytes.
   - **Rx Buffer** - Reception buffer size in bytes.
4. Click **OK**.

Associating a PLC Driver with an Access Name

After configuring the PLC driver, you need to associate it with an existing InTouch Access Name. For more information about creating an InTouch Access Name, see the *InTouch HMI Data Management Guide*.

When you associate an PLC driver with an Access Name, you must also specify a Station Address.

The Station Address is used by the MODBU driver to specify slave network addresses of PLCs. The Station Address in the Access Name should comply with the following syntax:

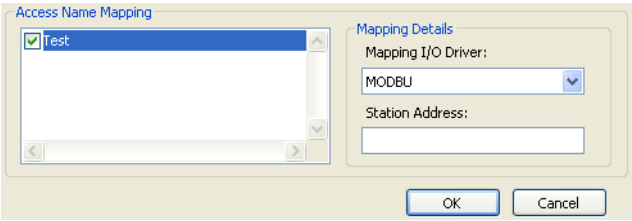
[*PLC Model*]: <*Slave ID*>

PLC Model	MBTCP Object (see MBTCP Documentation)
TSX Quantum	TSX Quantum
TSX Momentum	TSX Momentum
TSX Premium	TSX Premium
Generic 4	Modbus PLC with Register Size=4
Generic 5	Modbus PLC with Register Size=5
Generic 6	Modbus PLC with Register Size=6
G4	Same as Generic 4
G5	Same as Generic 5
G6	Same as Generic 6

Note For more information about configuring the station address, see the documentation of your PLC hardware manufacturer.

To associate a PLC driver with an Access Name

1. Open the **Driver Configuration** dialog box.
2. In the **Access Name Mapping** area, locate the InTouch Access Name you want to associate.



Note If no Access Names appear in the list, you need to define an Access Name from the InTouch HMI.

3. Select the Access Name you want to associate.
4. In the **Mapping I/O Driver** list, select the configured driver.
5. In the **Station Address** box, type the Station Address.

   When you move the mouse pointer over the **Station Address** box, a tooltip shows the required syntax for a station address.
6. Click **OK**.

Chapter 3

Troubleshooting the MODBU Driver

You can detect communication errors between an InTouch Compact Edition application and the PLC by using the following item names:

Item name	Purpose
SysReadStatus	Receives error code for a failed read operation from the PLC
SysWriteStatus	Receives error code for a failed write operation to the PLC
SysReadCompleted	Indicates if the read operation from the PLC completes (TRUE) or not (FALSE)
SysWriteCompleted	Indicates if the write operation to the PLC completes (TRUE) or not (FALSE)

Note You can associate only one InTouch tag with each item name.

To troubleshoot communication problems

1. In WindowMaker, configure a set of I/O tags to associate with the error item names:
   - An I/O discrete tag to point at the item name SysReadCompleted.
   - An I/O discrete tag to point at the item name SysWriteCompleted.

- An I/O integer tag to point at the item name SysReadStatus.
- An I/O integer tag to point at the item name SysWriteStatus.

2. Configure animation display links for the I/O tags, such as:
   - Fill color animations for the I/O tags associated with the SysReadCompleted and SysWriteCompleted items.
   - Analog value display links for the I/O tags associated with the SysReadStatus and SysWriteStatus items.
3. At run time, if a read or write error occurs between the InTouch Compact Edition and the PLC, the fill color animation indicates a problem. The value next to the fill color animation shows the error code.

Find the MODBU error codes in the following table. The table also includes typical problems associated with an error code and possible solutions.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
1	Invalid serial Port	The serial port specified is invalid	Specify a valid serial port
2	Invalid / unsupported item name	Item name specified is either invalid or unsupported	Check your item names for unsupported or invalid items
3	Invalid data values	Specified item name does not exist on the device so protocol received invalid data	Verify that specified item name exists on the device
4	Equipment failed	Equipment failed or out of order	Check equipment state
5	Ack	Ack action error during communication	Check the I/O driver configuration
6	Equipment in use	Command invalid when equipment is in use	Check the I/O driver configuration. Verify the documentation on the equipment in use for details on what would return this response

Error Code	Description	Possible Causes	Procedure to Solve
7	Negative Ack	Ack action error during communication	Check the I/O driver configuration
8	Memory parity error	Invalid communication parameter	Check the I/O driver configuration
9	Out of memory	There is not enough memory for the I/O Server	Check the amount of RAM memory available for your computer
10	Invalid / unsupported item name	Item name specified is either invalid or unsupported	Check your item names for unsupported or invalid items
11	Invalid / unsupported item name	Item name specified is either invalid or unsupported	Check your item names for unsupported or invalid items
12	Invalid lock size	Internal error when generating the protocol block sizes	Contact Technical Support and send your published application
13	Invalid CRC	Invalid CRC in response message	• Check the cable wiring • Check the station number • Check the I/O driver configuration
14	Time out waiting CTS	CTS signal was not received, could not send request	• Check the cable • Check if the communication line is busy
15	Time out waiting to start a message	Response was not received from the device	• Check the cable connection • Make sure that the I/O Server settings match the PLC configuration
16	Time out waiting for a message to finish	Started receiving a response but the communication was suspended	• Check the cable connection • Make sure that the I/O Server settings match the PLC configuration
17	Time out between rx char	Response was not received from the device	• Check the cable connection • Make sure that the I/O Server settings match the PLC configuration

Error Code	Description	Possible Causes	Procedure to Solve
18	Invalid BCD value	Tried reading an invalid BCD value	Verify that the PLC value is valid
19	Invalid BCD value	Tried writing a negative BCD value	Only positive BCD values can be entered
20	Invalid word swap	Tried swap word with invalid type	Put 'No Swap' in swap parameters
32	Time out waiting for a tx message to finish	Started receiving a response but the communication was suspended	• Check the cable connection • Make sure that the I/O Server settings match the PLC configuration
33	Invalid driver configuration file	Internal error	Contact Technical Support and send your published application
34	Invalid/ unsupported item name	Item name specified is either invalid or unsupported	Check your item names for unsupported or invalid items
35	Driver API not initialized	Internal error	Contact Technical Support and send your published application
36	Invalid/ unsupported item name	Item name specified is either invalid or unsupported	Check your item names for unsupported or invalid items
37	Invalid/ unsupported item name	Item name specified is either invalid or unsupported	Check your item names for unsupported or invalid items
38	Invalid station	Station specified for this driver is invalid	Check the I/O driver configuration and specify a valid station address
39	Invalid block size	Internal error when generating the protocol block sizes	Contact Technical Support and send your published application

Error Code	Description	Possible Causes	Procedure to Solve
-15	Time out start message	• Disconnected cables • PLC is turned off, in stop mode, or in error mode • Wrong station number • Wrong RTS/CTS control settings	• Check cable wiring • Check the PLC state – it must be RUN • Check the station number • Check the I/O driver configuration
-17	Time out between rx char	• PLC in stop mode or in error mode • Wrong station number • Wrong parity • Wrong RTS/CTS configuration settings	• Check cable wiring • Check the PLC state – it must be RUN • Check the station number • Check the I/O driver Configuration

Appendix

MODBU Item References

This section provides MODBU I/O driver reference information to configure the MODBU driver.

If an “OUT OF RANGE” item is used in the InTouch Compact Edition application, then the quality for that item is shown as 0 or BAD. It is also possible that the quality of one or more valid items goes to 0 or BAD due to an “OUT OF RANGE” item being used.

Invalid values in items (with/without suffixes) can return unpredictable results. You should ensure that valid values are written for each data type. For example, 0 to 65535 is the valid value range for 'word' type of data.

The MODBU driver supports float values with 4 bytes with up to 7 digits. Therefore, the valid range for float values is 3.4E +/- 38 (7 digits). Values outside this range will result in an invalid value written to the tag. Furthermore, values with more than 7 digits of precision will be rounded at the seventh digit.

The MODBU driver supports valid BCD values from 0 to 9999 and valid LBCD values from 0 to 99999999. Invalid BCD values (e.g.: 1010 in binary base) will result in an invalid value written to the tag.

Note The MODBU driver does not support DBCS, only ANSI characters.

Data Types and Address Ranges

The following table lists all the data types and address ranges that are valid for the MODBU driver.

Data Types	Sample Syntax	Valid Range of Initial Addresses	Comments
0X	0X:0	Varies according to equipment	Coil status: Reads and writes events using Modbus instructions 01, 05, and 15
1X	1X:0	Varies according to equipment	Input status: Reads events using Modbus instruction 02
STA	STA:0	Varies according to equipment	Exception Status: Reads events using Modbus instruction 07
3X	3X:0	Varies according to equipment	Input register: Reads events using Modbus instruction 04
4X	4X:0	Varies according to equipment	Holding register: Reads and writes events using Modbus instructions 03, 06 and 16
FP	FP:0	Varies according to equipment	Floating-point value (Holding Register): Reads and writes floating-point values using two consecutive Holding Registers
FPS	FPS:0	Varies according to equipment	Floating-point value (Holding Register): Reads and writes floating-point values using two consecutive Holding Registers with Byte Swap
FP3	FP3:0	Varies according to equipment	Floating-point value (Input Register): Reads floating-point values using two consecutive Input Registers
FP3S	FP3S:0	Varies according to equipment	Floating-point value (Input Register): Reads floating-point values using two consecutive Input Registers with Byte Swap
DW	DW:0	Varies according to equipment	Dword value (Holding Register): Reads and writes Dword values using two consecutive Holding Registers

Data Types	Sample Syntax	Valid Range of Initial Addresses	Comments
DWS	DWS:0	Varies according to equipment	Dword value (Holding Register): Reads and writes Dword values using two consecutive Holding Registers with Byte Swap
DW3	DW3:0	Varies according to equipment	Dword value (Input Register): Reads Dword values using two consecutive Input Registers
DW3S	DW3S:0	Varies according to equipment	Dword value (Input Register): Reads Dword values using two consecutive Input Registers with Byte Swap
BCD3	BCD3:0	Varies according to equipment	BCD value (Input Register): Reads events using Modbus instruction 04
BCD	BCD:0	Varies according to equipment	BCD value (Holding Register): Reads and writes events using Modbus instructions 03, 06 and 16
BCDDW	BCDDW:0	Varies according to equipment	BCD 32-bit integer value (Holding Register): Reads and writes 32-bit integer values using two consecutive Holding Registers
BCDDWS	BCDDWS:0	Varies according to equipment	BCD 32-bit integer value (Holding Register): Reads and writes 32-bit integer values using two consecutive Holding Registers with Byte Swap
BCDDW3	BCDDW3:0	Varies according to equipment	BCD 32-bit integer value (Input Register): Reads 32-bit integer values using two consecutive Input Registers
BCDDW3S	BCDDW3S:0	Varies according to equipment	BCD 32-bit integer value (Input Register): Reads 32-bit integer values using two consecutive Input Registers with Byte Swap
ID	ID:0	Varies according to equipment	Report Slave ID using Modbus instruction 17

Data Types	Sample Syntax	Valid Range of Initial Addresses	Comments
ST	ST:0	Varies according to equipment	String value (Holding Register): Reads and writes String values using consecutive Holding Registers
STS	STS:0	Varies according to equipment	String value (Holding Register): Reads and writes String values using consecutive Holding Registers with Byte Swap

Device Registers

For examples of how device registers are specified using Header and Address, see the following table:

Device Register	Header	Address
00001	0x:1	0
00010	0x:0	10
01020	0x:1000	20
10001	1x:1	0
10010	1x:0	10
11020	1x:1000	20
30001	3x:1	0
30010	3x:0	10
31020	3x:1000	20
30000 and 30001	FP3:0	0
31001 and 31000	FP3S:0	1000
40001	4x:1	0
40010	4x:0	10
41020	4x:1000	20
40010 (bit 0)	4x:0	10.0
41010 (bit 7)	4x:1000	10.7
40001 and 40002	FP:1	0
40013 and 40014	FP:0	13
41021 and 41022	FP:1000	21

Device Register	Header	Address
40001 and 40002	DW:1	0
40013 and 40014	DW:0	13

Other Type Settings

Parameter	In Touch Compact Edition Configuration
Slave address	Slave Address in the Station field
Swap 16-bit registers	This functionality can be expressed by using the SWAP option to 1 in the CE Driver configuration. Notice that this option cannot be set differently for different devices as the Wonderware I/O Server can. Also, you cannot have a different setting for reals and for longs
Support multiple coil write	Option not supported by CEView for the first release
Swap string bytes	Option not supported by CEView for the first release
Support multiple register write	Option not supported by CEView for the first release
Bit order format	MOTCP supports only the B1 B2 … B16 order format
Register size	The user can specify the register size in the station field by using the models Generic 4, Generic 5, and Generic 6
String variable style	Only C Style is supported (0 in the string will indicate end of string)
Register type	Only Binary is supported
Block I/O size	CEView divides the requests into blocks with at most 64 words (128 bytes)
Maximum address ranges	CEView I/O Server does not impose limits. The PLC will return error if an invalid address is requested

Register-Number Item Names

TSX Quantum and TSX Momentum

Register Type	TSX Quantum/ TSX Momentum	Modicon Micro	Compact 984-265	Compact 984-145	Tag Type	Access	Supported by InTouch Compact Edition
Output Coils	1-65536	1-9999	1-65536	1-9999	Discrete	Read/write	Yes
Contacts	100001-165536	10001-19999	100001-165536	10001-19999	Discrete	Read-only	Yes
Input	300001-365536	30001-39999	300001-365536	30001-39999	Analog	Read-only	Yes
Holding	400001-465536	40001-49999	400001-465536	40001-49999	Analog	Read/write	Yes
Extended	6x0000-6x9999			6x000-6x999	Analog	Read/write	No

TSX Quantum

Register Type	TSX Premium	Tag Type	Access	Supported by InTouch Compact Edition
Output Coils	1-65536	Discrete	Read/write. Mapped to %M PLC object; that is, Output Coils and Contacts Registers are the same	Yes
Contacts	100001-165536	Discrete	Read-only. Mapped to %M PLC object; that is, Output Coils and Contacts Registers are the same	Yes

Register Type	TSX Premium	Tag Type	Access	Supported by InTouch Compact Edition
Input	300001-365536	Analog	Read-only. Mapped to %MW PLC object; that is, Input Registers and Holding Registers are the same	Yes
Holding	400001-465536	Analog	Read/write. Mapped to %MW PLC object; that is, Input Registers and Holding Registers are the same. Poking a value to a particular bit in the Holding register in this TSX Premium PLC is not supported	Yes

Generic Modbus (4, 5 and 6-digit)

Register Type	Generic Modbus (6-Digit)	Generic Modbus (5-Digit)	Generic Modbus (4-Digit)	Tag Type	Access	Supported by InTouch Compact Edition
Output Coils	1-65536	1-9999	1-999	Discrete	Read/Write	Yes
Contacts	100001-165536	10001-19999	1001-1999	Discrete	Read-Only	Yes
Input	300001-365536	30001-39999	3001-3999	Analog	Read-Only	Yes
Holding	400001-465536	40001-49999	4001-4999	Analog	Read/Write	Yes
Extended	6x0000-6x9999	6x000-6x999		Analog	Read/Write	No

Item Names Using Modicon PLC Register Addresses

Item Name	Description	Supported by InTouch Compact Edition
400001	When no spaces and no letters follow the register number, the register contents are treated as a 16-bit unsigned quantity	Yes
400001 S	When a space and the letter "S" follow the register number, the register contents are treated as a 16-bit signed quantity	Yes
400001 I	When a space and the letter "I" follow the register number, the register contents are treated as a 32-bit signed quantity. This takes up two consecutive registers	Yes
400001 L	When a space and the letter "L" follow the register number, the register contents are treated as a 32-bit signed quantity	Yes
400001 F	When a space and the letter "F" follow the register number, the register contents are treated as a floating-point quantity taking up two consecutive registers	Yes
400001 U	When a space and the letter "U" follow the register number, the register contents are treated as a 32-bit unsigned quantity	No
400001-400003 M	• When a space and the letter "M" follow the register number or register number pair separated by a dash, the register contents are treated as ASCII data • Each register contains up to two (2) ASCII characters • This example represents six (6) ASCII characters	Yes
300001:10	• When a colon and a number from 1 (one) to 16 follow the register number, the register contents are treated as discrete data • This example represents bit 10 of the input register 300001	Yes

Absolute Notation Item Names

Item Name	Description	Supported by InTouch Compact Edition
nnnnn **DO**	• Discrete Output • Refers to the same data Modbus calls "coils" • Valid range is 0 (zero) **DO** through 65535 **DO**	Yes
nnnnn **DI**	• Discrete Input • Refers to the same data called "contacts" by Modbus • Valid range is 0 (zero) **DI** through 65535 **DI**	Yes
nnnnn **IR**	• Input Register • Refers to the same data called "input register" • Valid range is 0 (zero) **IR** through 65535 **IR**	Yes
nnnnn **HR**	• Holding Register • Refers to the same data Modbus calls "holding register" • Valid range is 0 (zero) **HR** through 65535 **HR**	Yes
nnnnn **PV**	• Process Variable • Refers to holding register, but treated as floating points and assumes two (2) registers per floating-point number • Valid range is 0 (zero) **PV** through 65535 **PV**	No

Modulo-10000 Point Item Names

Item Name	Description	Supported by InTouch Compact Edition
400001-400002	• Can represent numbers between 0 and 99,999,999 • Register 400001 = <9999> • Register 400002 = <9999>	No
400005-400007	• Can represent numbers between 0 and 2,147,483,646 • Register 400005 = <21> • Register 400006 = <4748> • Register 400007 = <3646>	No

Controller Function Codes

Code	Name	Description	Supported by InTouch Compact Edition
01 (0x01)	Read Coils	Reads the ON/OFF status of discrete outputs (0X references, coils) in the slave	Y
02 (0x02)	Read Discrete Inputs	Reads the ON/OFF status of discrete inputs (1XXXXX references) in the slave	Y
03 (0x03)	Read Holding Registers	Reads the binary contents of holding registers (4XXXXX references) in the slave	Y
04 (0x04)	Read Input Registers	Reads the binary contents of input registers (3XXXXX references) in the slave	Y
05 (0x05)	Write Single Coil	Forces a single coil (0X reference) to either ON or OFF	Y

Code	Name	Description	Supported by InTouch Compact Edition
06 (0x06)	Write Single Register	Presets a value into a single holding register (4XXXXX reference)	Y
15 (0x0F)	Write Multiple Coils	Forces each coil (0XXXXX reference) in a sequence of coils to either ON or OFF	N
16 (0x10)	Write Multiple Registers	Presets values into a sequence of holding registers (4XXXXX references)	Y - only when writing 32-bit values
20 (0x14)	Read General Reference	Returns the contents of registers in the Extended Memory file (6XXXXX) references	N
21 (0x15)	Write General Reference	Writes the contents of registers in the Extended Memory file (6XXXXX) references	N
22 (0x16)	Mask Write Holding Register	Modifies the contents of a specified (4XXXXX reference) holding register using a combination of an AND mask, an OR mask, and the register's current contents. The function can be used to set or clear individual bits in the register	N

Index

A

Access Name 26
advanced settings
  description 9
  serial connection 17
  TCP/IP or UDP/IP connection 22

B

Buffer Length settings 17

C

connection type
  description 11
  serial connection 11
  TCP/IP connection 12
  UDP/IP connection 14

D

device specifications 7
documentation conventions 5
driver characteristics 8

H

Hand Shake settings 17

I

item references
  control RTS settings 36
  data types and address ranges 36
  description 35
  device registers 38
  generic Modbus 41
  Modicon PLC register addresses 42
  other type settings 39
  TSX Momentum 40
  TSX Quantum 40

M

MODBU driver
  description 7
  network specifications 8

P

protocol settings 17

S

setting DTR and IR 17

T

Technical Support 6
time-out settings 17
troubleshooting
  communication problems 29

configure animation display 30
Troubleshooting the MODBU driver 29
TSX Quantum 40