

Serial Configuration Guide



Intelligent Cyber Secure Platform



Version: 1.17.13-1, Date: Jan 2024



© 2024 iS5 Communications Inc. All rights reserved.

Copyright Notice

© 2024 iS5 Communications Inc. All rights reserved.

No Part of this publication may be reproduced in any form without the prior written consent of iS5 Communications Inc. (iS5).

Trademarks

iS5Com is a registered trademark of iS5. All other trademarks belong to their respective owners.

Regulatory Compliance Statement

Product(s) associated with this publication complies/comply with all applicable regulations. For more details, refer to the Technical Specifications.

Warranty

iS5 warrants that all products are free from defects in material and workmanship for a specified warranty period from the invoice date (5 years for most products). iS5 will repair or replace products found to be defective within this warranty period including shipping costs. This warranty does not cover product modifications or repairs done by persons other than iS5-approved personnel, and this warranty does not apply to products that are misused, abused, improperly installed, or damaged by accident. Refer to the Technical Specifications for the actual warranty period(s) of the product(s) associated with this publication. Warranty certificate available at: <https://is5com.com/warranty>

Disclaimer

Information in this publication is intended to be accurate. iS5 shall not be responsible for its use or infringements on third-parties because of the use of this publication. There may occasionally be unintentional errors on this publication. iS5 reserves the right to revise the contents of this publication without notice.

Contact Information

iS5 Communications Inc. 5895 Ambler Dr., Mississauga, Ontario, L4W 5B7 Tel: 1+ 905-670-0004 Website: <http://www.is5com.com/> Technical Support: E-mail: support@is5com.com Sales Inquiries: [Phoenix Contact Sales Subsidiaries](#)

End User License Agreement (EULA)

TERMS AND CONDITIONS FOR SOFTWARE PROGRAMS AND EMBEDDED SOFTWARE IN PRODUCTS

1) EULA

All products which consist of or include software (including operating software for hardware supplied by Supplier and software in object code format that is embedded in any hardware) and/or any documentation shall be subject to the End User License Agreement (“EULA”) attached hereto as Exhibit A. Buyer shall be deemed to have agreed to be bound by all of the terms, conditions and obligations therein and shall ensure that all subsequent purchasers and licensees of such products shall be further bound by all of the terms, conditions and obligations therein. For software and/or documentation delivered in connection with these Terms and Conditions, that is not produced by Supplier and which is separately licensed by a third party, Buyer’s rights and responsibilities with respect to such software or documentation shall be governed in accordance with such third party’s applicable software license. Buyer shall, on request, enter into one or more separate “click-accept” license agreements or third party license agreements in respect thereto. Supplier shall have no further obligations with respect to such products beyond delivery thereof. Where Buyer is approved by Supplier to resell products, Buyer shall provide a copy of the EULA and applicable third party license agreements to each end user with delivery of such products and prior to installation of any software. Buyer shall notify Supplier promptly of any breach or suspected breach of the EULA or third party license agreements and shall assist Supplier in efforts to preserve Supplier’s or its supplier’s intellectual property rights including pursuing an action against any breaching third parties. For purposes of these terms and conditions: “software” shall mean scripts, programs, macros, computer programs, application programming and other interfaces, tools and other instructions and sets of instructions for hardware to follow, including SQL and other query languages, hypertext markup language (“html”) and other computer mark-up languages; “hardware” shall mean mainframes, personal computers, servers, client/server stations, network equipment, routers, semi-conductor chips, communication lines and other equipment; and “documentation” shall mean documentation supplied by Supplier relating to the development, use, installation, implementation, integration, configuration, operation, modification, maintenance or support of any software.

2) INTELLECTUAL PROPERTY

Buyer shall not alter, obscure, remove, cancel or otherwise interfere with any markings (including without limitation any trademarks, logos, trade names, or labelling applied by Supplier). Buyer acknowledges that Supplier is the sole owner of the trademarks used in association with the products and that Buyer has no right, title or interest whatsoever in such trademarks and any goodwill associated therewith and that all goodwill associated with such trademarks is owned by and shall enure exclusively to and for the benefit of Supplier. Further, Buyer shall not represent in any manner that it has acquired any ownership rights in such trademarks or other intellectual property of Supplier. Supplier will defend any claim against Buyer that any iS5Com branded product supplied under these Terms and Conditions infringes third party patents or copyrights (a “Patent Claim”) and will indemnify Buyer against the final judgment entered by a court of competent jurisdiction or any settlements arising out of a Patent Claim, provided that Buyer: (1) promptly notifies Supplier in writing of the Patent Claim; and (2) cooperates with Supplier in the defence of the Patent Claim, and grants Supplier full and exclusive control of the defence and settlement of the Patent Claim and any subse-

quent appeal. If a Patent Claim is made or appears likely, Buyer agrees to permit Supplier to procure for Buyer the right to continue using the affected product, or to replace or modify the product with one that is at least functionally equivalent. If Supplier determines that none of those alternatives is reasonably available, then Buyer will return the product and Supplier will refund Buyer's remaining net book value of the product calculated according to generally accepted accounting principles. Supplier has no obligation for any Patent Claim related to: (1) compliance with any designs, specifications, or instructions provided by Buyer or a third party on Buyer's behalf; (2) modification of a product by Buyer or a third party; (3) the amount or duration of use which Buyer makes of the product, revenue earned by Buyer from services it provides that use the product, or services offered by Buyer to external or internal Buyers; (4) combination, operation or use of a product with non-Supplier products, software or business processes; or (5) use of any product in any country other than the country or countries specifically authorized by Supplier.

3) EXPORT CONTROLS AND SANCTIONS

- a) In these Term and Conditions, "**Export Controls and Sanctions**" means the export control and sanctions laws of each of Canada, the US and any other applicable country, territory or jurisdiction including the United Nations, European Union and the United Kingdom, and any regulations, orders, guides, rules, policies, notices, determinations or judgements issued thereunder or imposed thereby.
- b) Supplier products, documentation and services provided under these Terms and Conditions may be subject to Canadian, U.S. and other country Export Controls and Sanctions. Buyer shall accept and comply with all applicable Export Control and Sanctions in effect and as amended from time to time pertaining to the export, re-export and transfer of Supplier's products, documentation and services. Buyer also acknowledges and agrees that the export, re-export or transfer of Supplier products, documentation and services contrary to applicable Export Controls and Sanctions may be a criminal offence.
- c) For greater certainty, Buyer agrees that (i) it will not directly or indirectly export, re-export or transfer Supplier products, documentation and services provided under these Terms and Conditions to any individual or entity in violation of any aforementioned Export Controls and Sanctions; (ii) it will not directly or indirectly export, re-export or transfer any such products, documentation and services to any country or region of any country that is prohibited by any applicable Export Controls and Sanctions or for any of the following end-uses, or in any of the following forms unless expressly authorized by any applicable government permit issued under or otherwise expressly permitted by applicable Export Controls and Sanctions:
 - i) For use that is directly or indirectly related to the research, design, handling, storage, operation, detection, identification, maintenance, development, manufacture, production or dissemination of chemical, biological or nuclear weapons, or any missile or other delivery systems for such weapons, space launch vehicles, sounding rockets or unmanned air vehicle systems;
 - ii) Technical information relating to the design, development or implementation of the cryptographic components, modules, interfaces, or architecture of any software; or
 - iii) Source code or pseudo-code, in any form, of any of the cryptographic components, modules, or interfaces of any software.
- d) Buyer confirms that it is not (i) listed as a sanctioned person or entity under any Export Controls and Sanctions list of designated persons, denied persons or specially designated

nationals maintained by the Canadian Department of Foreign Affairs, Trade and Development, the Canadian Department of Public Safety and Emergency Preparedness, the U.S. Office of Foreign Assets Control of the U.S. Department of the Treasury, the U.S. Department of State, the U.S. Department of Commerce, United Nations Security Council, the European Union or any EU member state, HM's Treasury, or any other department or agency of any of the aforementioned countries or territories, or the United Nations or any other country's sanctions-related list; (ii) owned or controlled by such person or entity; or (iii) acting in any capacity on behalf of or for the benefit of such person or entity. Buyer also confirms that this applies equally to any of its affiliates, joint venture partners, subsidiaries and to the best of Buyer's knowledge, any of its agents or representatives.

Exhibit A: End User License Agreement

IMPORTANT – READ CAREFULLY: iS5 Communications Inc. (“**iS5Com**”) licenses the iS5Com Materials (as defined below) subject to the terms and conditions of this end user license agreement (the “**EULA**”). BY SELECTING “ACCEPT” OR OTHERWISE EXPRESSLY AGREEING TO THIS EULA, BY DOWNLOADING, INSTALLING OR USING THE SOFTWARE, OR BY USING THE HARDWARE (AS DEFINED BELOW), ALL OF THE TERMS AND CONDITIONS CONTAINED IN THIS EULA BECOME LEGALLY BINDING ON THE CUSTOMER. This End User License Agreement (the “**EULA**”) supplements the Terms and Conditions or such other terms and conditions between iS5Com or, if applicable, a reseller for iS5Com, and the Customer (as defined below) (in either case, the “**Contract**”).

1) DEFINITIONS

*“**Confidential Information**” means all data and information relating to the business and management of iS5Com, including iS5Com Materials, trade secrets, technology and records to which access is obtained hereunder by the Customer, and any materials provided by iS5Com to the Customer, but does not include any data or information which: (a) is or becomes publicly available through no fault of the Customer; (b) is already in the rightful possession of the Customer prior to its receipt from iS5Com; (c) is already known to the Customer at the time of its disclosure to the Customer by iS5Com and is not the subject of an obligation of confidence of any kind; (d) is independently developed by the Customer; (e) is rightfully obtained by the Customer from a third party; (e) is disclosed with the written consent of iS5Com; or (f) is disclosed pursuant to court order or other legal compulsion.*

- *“**Customer**” means the licensee of the iS5Com Software pursuant to the Contract.*
- *“**iS5Com Documentation**” means Documentation supplied by or on behalf of iS5Com under the Contract relating to the development, use, installation, implementation, integration, configuration, operation, modification, maintenance or support of iS5Com Software, or iS5Com Firmware.*
- *“**iS5Com Firmware**” means iS5Com Software in object code format that is embedded in iS5Com Hardware.*
- *“**iS5Com Hardware**” means Hardware supplied by or on behalf of iS5Com under the Contract.*
- *“**iS5Com Materials**” means, collectively, the iS5Com Software and the iS5Com Documentation.*

- **“i55Com Software”** means Software supplied by or on behalf of i55Com under the Contract. For greater certainty, i55Com Software shall include all operating Software for i55Com Hardware, and i55Com Firmware.
- **“Documentation”** means written instructions and manuals of a technical nature.
- **“EULA”** means this End User License Agreement.
- **“Hardware”** means hardware, mainframes, personal computers, servers, client/server stations, network equipment, routers, semi-conductor chips, communication lines and other equipment.
- **“Intellectual Property Rights”** means any and all proprietary rights provided under: (i) patent law; (ii) copyright law (including moral rights); (iii) trade-mark law; (iv) design patent or industrial design law; (v) semi-conductor chip or mask work law; or (vi) any other statutory provision or common law principle applicable to this EULA, including trade secret law, which may provide a right in either Hardware, Software, Documentation, Confidential Information, ideas, formulae, algorithms, concepts, inventions, processes or know-how generally, or the expression or use of such Hardware, Software, Documentation, Confidential Information, ideas, formulae, algorithms, concepts, inventions, processes or know-how trade secret law; any and all applications, registrations, licenses, sub-licenses, franchises, agreements or any other evidence of a right in any of the foregoing; and all licenses and waivers and benefits of waivers of the intellectual property rights set out herein, all future income and proceeds from the intellectual property rights set out herein, and all rights to damages and profits by reason of the infringement of any of the intellectual property rights set out herein.
- **“Software”** means scripts, programs, macros, computer programs, application programming and other interfaces, tools and other instructions and sets of instructions for hardware to follow, including SQL and other query languages, hypertext markup language (“html”) and other computer mark-up languages.
- **“Third Party License Terms”** means additional terms and conditions that are applicable to Third Party Software.
- **“Third Party Software”** means Software owned by any third party, licensed to i55Com and sublicensed to the Customer.
- **“Update”** means a supplemented or revised version of i55Com Software which rectifies bugs or makes minor changes or additions to the functionality of i55Com Software and is designated by i55Com as a higher release number from, for example, 6.06 to 6.07 or 6.1 to 6.2.

2) LICENSE

– 2.1 License Grant

The i55Com hereby grants to the Customer, subject to any Third Party License Terms, a non-exclusive, non-transferable, non-sublicensable right and licence to use i55Com Materials solely in object code format, solely for the Customer’s own business purposes, solely in accordance with this EULA (including, for greater certainty, subject to Section 6.1 of this EULA) and the applicable i55Com Documentation, and, in the case of i55Com Firmware, solely on i55Com Hardware on which i55Com Firmware was installed, provided that Customer may only install i55Com Software on such number of nodes expressly set out in the Contract.

– 2.2 License Restrictions

Except as otherwise provided in Section 2.1 above, the Customer shall not: (a) copy iS5Com Materials for any purpose, except for the sole purpose of making an archival or back-up copy; (b) modify, translate or adapt the iS5Com Materials, or create derivative works based upon all or part of such iS5Com Materials; (c) assign, transfer, loan, lease, distribute, export, transmit, or sublicense iS5Com Materials to any other party; (d) use iS5Com Materials for service bureau, rent, timeshare or similar purposes; (e) decompile, disassemble, decrypt, extract, or otherwise reverse engineer, as applicable, iS5Com Software or iS5Com Hardware; (f) use iS5Com Materials in a manner that uses or discloses the Confidential Information of iS5Com or a third party without the authorization of such person; (g) permit third parties to use iS5Com Materials in any way that would constitute breach of this EULA; or (h) otherwise use iS5Com Materials except as expressly authorized herein.

– **2.3 Updates and Upgrades**

The license granted hereunder shall apply to the latest version of iS5Com Materials provided to the Customer as of the effective date of this EULA, and shall apply to any Updates and Upgrades subsequently provided to the Customer by iS5Com pursuant to the terms of this EULA. Customer shall only be provided with Updates and/or Upgrades if expressly set out in the Contract.

– **2.4 Versions**

In the event any Update or Upgrade includes an amended version of this EULA, Customer will be required to agree to such amended version in order to use the applicable iS5Com Materials and such amended EULA shall be deemed to amend the previously effective version of the EULA.

– **2.5 Third Party Software**

Customer shall comply with any Third Party License Terms.

3) **OWNERSHIP**

– **3.1 Intellectual Property**

Notwithstanding any other provision of the Contract, iS5Com and the Customer agree that iS5Com is and shall be the owner of all Intellectual Property Rights in iS5Com Materials and all related modifications, enhancements, improvements and upgrades thereto, and that no proprietary interests or title in or to the intellectual property in iS5Com Materials is transferred to the Customer by this EULA. iS5Com reserves all rights not expressly granted to the Customer under Section 2.1.

– **3.2 Firmware**

iS5Com and the Customer agree that any and all iS5Com Firmware in or forming a part of iS5Com Hardware is being licensed and not sold, and that the words “purchase,” “sell” or similar or derivative words are understood and agreed to mean “license,” and that the word “Customer” as used herein are understood and agreed to mean “licensee,” in each case in connection with iS5Com Firmware.

– **3.3 Third Party Software**

Certain of iS5Com Software provided by iS5Com may be Third Party Software owned by one or more third parties and sublicensed to the Customer. Such third parties retain ownership of and title to such Third Party Software, and may directly enforce the Customer’s obligations hereunder in order to protect their respective interests in such Third Party Software.

4) **CONFIDENTIALITY**

– **4.1 Confidentiality**

The Customer acknowledges that i55Com Materials contain Confidential Information of i55Com and that disclosure of such Confidential Information to any third party could cause great loss to i55Com. The Customer agrees to limit access to i55Com Materials to those employees or officers of the Customer who require access to use i55Com Materials as permitted by the Contract and this EULA and shall ensure that such employees or officers keep the Confidential Information confidential and do not use it otherwise than in accordance with the Contract and this EULA. The obligations set out in this Section 4 shall continue notwithstanding the termination of the Contract or this EULA and shall only cease to apply with respect to such part of the Confidential Information as is in, or passes into, the public domain (other than in connection with the Customer's breach of this EULA) or as the Customer can demonstrate was disclosed to it by a third person who did not obtain such information directly or indirectly from i55Com.

– **4.2 Irreparable Harm**

Without limiting any other rights or remedies available to i55Com in law or in equity, the Customer acknowledges and agrees that the breach by Customer of any of the provisions of this EULA would cause serious and irreparable harm to i55Com which could not adequately be compensated for in damages and, in the event of a breach by the Customer of any of such provisions, the Customer hereby consents to an injunction against it restraining it from any further breach of such provisions.

– **4.3 Security**

*Any usernames, passwords and/or license keys ("**Credentials**") provided to you by i55Com shall be maintained by the Customer and its representatives in strict confidence and shall not be communicated to or used by any other persons. THE CUSTOMER SHALL BE RESPONSIBLE FOR ALL USE OF CREDENTIALS, REGARDLESS OF THE IDENTITY OF THE PERSON(S) MAKING SUCH USE, AND WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, IS5COM SHALL HAVE NO RESPONSIBILITY OR LIABILITY IN CONNECTION WITH ANY UNAUTHORIZED USE OF CREDENTIALS.*

5) **LIMITATION OF LIABILITY**

– **5.1 Disclaimer**

EXCEPT FOR THE EXPRESS WARRANTIES MADE BY IS5COM IN THE CONTRACT, (A) IS5COM MAKES NO AND HEREBY EXPRESSLY DISCLAIMS, AND THE PARTIES HERETO HEREBY EXPRESSLY WAIVE AND EXCLUDE TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAWS, AND THE CUSTOMER AGREES NOT TO SEEK OR CLAIM ANY BENEFIT THEREOF, IN EACH CASE, ALL WARRANTIES, CONDITIONS, REPRESENTATIONS OR INDUCEMENTS (AND THERE ARE NO OTHER WARRANTIES, CONDITIONS, REPRESENTATIONS OR INDUCEMENTS, ORAL OR WRITTEN, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, OF ANY KIND WHATSOEVER SET OUT HEREIN) WITH RESPECT TO THE IS5COM MATERIALS, INCLUDING AS TO THEIR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, DESIGN OR CONDITION, COMPLIANCE WITH THE REQUIREMENTS OF ANY APPLICABLE LAWS, CONTRACT OR SPECIFICATION, NON- INFRINGEMENT OF THE RIGHTS OF OTHERS, ABSENCE OF LATENT DEFECTS, OR AS TO THE ABILITY OF THE IS5COM MATERIALS TO MEET CUSTOMER'S REQUIREMENTS OR TO OPERATE OF ERROR

FREE; AND (B) THE IS5COM MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OR CONDITION OF ANY KIND.

– **5.2 Limitation of Liability**

EXCEPT AS EXPRESSLY PROVIDED IN THE CONTRACT, IN NO EVENT SHALL IS5COM BE LIABLE TO THE CUSTOMER OR ANY THIRD PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING UNDER OR IN CONNECTION WITH THIS EULA EVEN IF ADVISED OF THE POSSIBILITY THEREOF. THIS LIMITATION SHALL APPLY IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND OR CLAIM, INCLUDING BREACH OF CONTRACT, NEGLIGENCE, TORT OR ANY OTHER LEGAL THEORY, AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES AND/OR FAILURE OF THE ESSENTIAL PURPOSE OF THIS EULA.

6) **TERM**

– **6.1 Term**

Customer’s right to use i55Com Materials shall terminate at such time as set out in the Contract or upon termination or expiration of the Contract, in each case at which time this EULA shall be deemed to terminate.

– **6.2 Survival**

Each of Sections 1, 2.4, 3, 4, 5, 6.2, and 7 shall survive termination of the EULA.

7) **MISCELLANEOUS**

– **7.1 Miscellaneous**

This EULA is (together with, as applicable, any click-wrap license agreement or Third Party License Terms pertaining to the use of i55Com Materials) the entire agreement between the Customer and i55Com pertaining to the Customer’s right to access and use i55Com Materials, and supersedes all prior or collateral oral or written representations or agreements related thereto. Notwithstanding anything to the contrary contained in the Contract, to the extent of any inconsistency between this EULA and the Contract, or any such applicable click-wrap agreement, this EULA shall take precedence over the Contract and such click-wrap agreement. In the event that one or more of the provisions is found to be illegal or unenforceable, this EULA shall not be rendered inoperative but the remaining provisions shall continue in full force and effect. The parties expressly disclaim the application of the United Nations Convention for the International Sale of Goods. This EULA shall be governed by the laws of the Province of Ontario, Canada, and federal laws of Canada applicable therein. In giving effect to this EULA, neither party will be or be deemed an agent of the other for any purpose and their relationship in law to the other will be that of independent contractors. Any waiver of any terms or conditions of this EULA: (a) will be effective only if in writing and signed by the party granting such waiver, and (b) shall be effective only in the specific instance and for the specific purpose for which it has been given and shall not be deemed or constitute a waiver of any other provisions (whether or not similar) nor shall such waiver constitute a continuing waiver unless otherwise expressly provided. The failure of either party to exercise, and any delay in exercising, any of its rights hereunder, in whole or in part, shall not constitute or be deemed a waiver or forfeiture of such rights, neither in the specific instance nor on a continuing basis. No single or partial exercise of any such right shall preclude any other or further exercise of such right or the exercise of any other right. Customer shall not assign or transfer this EULA or any of its rights or obligations hereunder, in whole or in part, without the prior written consent of

iS5Com. The division of this EULA into sections and the insertion of headings are for convenience of reference only and shall not affect the construction or interpretation of this EULA. References herein to Sections are to sections of this Agreement. Where the word “include”, “includes” or “including” is used in this EULA, it means “include”, “includes” or “including”, in each case, “without limitation”. All remedies provided for iS5Com under this EULA are non-exclusive and are in addition, and without prejudice, to any other rights as may be available to of iS5Com, whether in law or equity. By electing to pursue a remedy, of iS5Com does not waive its right to pursue any other available remedies. The parties acknowledge that they have required this Agreement to be written in English. Les parties aux présentes reconnaissent qu’elles ont exigé que la présente entente soit rédigée en anglais.

– **7.2 Subject to Change**

*Terms and Conditions are subject to change. For the latest information please visit:
<https://is5com.com/terms-and-conditions/>*

Contents

	Serial Configuration Guide	i
	Copyright Notice	ii
	End User License Agreement (EULA)	iii
Chapter: 1	Introduction	1
	Purpose and Scope	1
	CLI Command Modes	1
	User Exec Mode	3
	Privileged Exec Mode	3
	Global Configuration Mode	3
	Interface Configuration Mode	3
	Port Channel Interface Configuration	4
	VLAN Interface Configuration Mode	4
	MRP Interface Configuration Mode	4
	UFD Configuration Mode	5
	DHCP Pool Configuration Mode	5
	Privilege Levels and Command Access	5
	Configuration Terminal Access	9
	CLI Document Convention	10
Chapter: 2	Serial Port Offline Indication	11
	Configuration	11
	Verification	13
Chapter: 3	Modbus Client and Server Roles	14
	Modbus Client and Server Gateway Responsibilities	14
	Modbus Server Role	15
	Modbus Client Role	15
	Modbus Function and Exception Codes	16

Chapter: 4	MODBUS TCP Server Mode Configuration	18
	Listening to the Master19
	Connecting Master to the Server20
	Sending Request from Master to not Configured Slave23
	Configuring Send Exceptions25
	Sending Request from Master to Slave, Which Is Offline27
	Sending Request from Master to Slave30
	Using not Native Modbus Port34
	Connecting Second Master36
	Sending keep-alive to Both Masters37
	Setting max client connections to 239
	Connecting with Third Master41
	Adding Slaves to Second Serial Interface42
	Connecting Second Slave To First Serial Interface43
	Connecting Third Slave To Second Serial Interface44
	Connecting Forth Slave To Second Serial Interface45
	Sending Request from First Master to Slave 1246
	Sending Request from the First Master to the Slave 1248
	Sending Request from the First Master to the Slave 248
	Sending Request from the First Master to the Slave 2148
	Sending Request from the Second Master to the Slave 148
	Sending Request from the Second Master to the Slave 1248
	Sending Request from the Second Master to the Slave 248
	Sending Request from the Second Master to the Slave 2148
	Deleting Forth Slave from the Second Serial Interface48
	Changing Response Time for Second Serial Interface51
	Sending Two Requests from both Masters to Same Slave with no Delay Between 52	
	Configuring RX-to-TX Delay52
	Changing Maximum Pending Messages to 054
	Sending Two Requests from One Master to Same Slave with no Delay Between 55	
	Sending Broadcast Write Request from First Master57
	Sending Broadcast Read Request from the First Master59
	Configuring Turnaround Delay60
	Configuring Hold Time62
	Clearing Serial Configuration and Deleting the Profile64
 Chapter: 5	 MODBUS TCP Client Mode Configuration	 65
	Reconnecting65
	Connecting to Server67
	Disconnecting from Server68
	Connecting to not Native Modbus Port69
	Configuring keep-alive72
	Reconnecting to the Server73

	Connecting to the Master74
	Not Configured Slave Polling (Exception by Client)76
	Adding Slaves to Client79
	Reading Holding Registers from Slave81
	Writing to Holding Registers from Slave84
	Reading Not Supported Number of Holding Registers from Slave (Exception by Slave)88
	Reading Holding Registers with Wrong CRC92
	Reading Holding Registers from Disconnected Slave95
	Disabling forward-exception95
	Enabling forward-exception96
	Configuring DSCP97
	Removing Slaves98
	Writing to Slave Using Slave ID 0	100
	Reading from Slave Using Slave ID 0	101
	Changing response-timeout	102
	Configuring Turnaround Delay	103
	Configuring hold-time Delay	105
	Configuring rx-to-tx Delay	107
	Clearing Serial Configuration and Deleting Profile	108
Chapter: 6	Raw Socket Server Mode Configuration109
	Configuring Packetizing OFF	109
	Configuring Packet Timer	110
	Configuring Packet Character	112
	Configuring Packet Size	113
	Configuring keep-alive	115
	Configuring Maximum Connections	116
	Configuring Turnaround Delay	117
	Configuring Hold Time	119
	Configuring of No Delays	121
	Configuring Rx-to-Tx Delay	122
	Clearing Serial Profile Configuration and Deleting Profile	124
Chapter: 7	Raw Socket IN-OUT Mode Configuration125
	Running Client and Server Modes	125
	Accepting Connection as a Server	126
	Testing Server Mode	127
	Going back to Client and Server Modes	128
	Connecting as Client	128
	Testing Client Mode	129
	Clearing the Serial Profile Configuration and Deleting the Profile	129
Chapter: 8	Raw Socket Client Mode Configuration131
	Connecting	131

	Disconnecting	133
	Sending with Packetizing OFF	133
	Configuring Packet Timer	135
	Configuring Packet Character	136
	Configuring Packet Size	138
	Configuring KEEP ALIVE	139
	Configuring Turnaround Delay	141
	Configuring Hold Time Delay	142
	Configuring No Delays	143
	Configuring Rx-to-Tx Delay	145
	Clearing the Serial Profile Configuration and Deleting the Profile . . .	146
	Enable Mirroring	147
	Serial TCP Mirroring	147
	Verification	149
	Disable Mirroring	150
Chapter: 9	Raw Preemptive Connection Configuration151
	General	152
	Connecting to Permanent Client	152
	Functional Tests for Permanent Client (Same Tests as in Server) . . .	154
	Disabling Timers	155
	Connecting Dynamic Master	155
	Dynamic Master Disconnection after Idle Timeout	156
	Connecting Second Dynamic Master	157
	Configuring Dynamic Packet Timeout	157
	Configuring Dynamic Packet Character	158
	Configuring Packet Size	159
	Sending Data from Permanent Client	161
	Disabling Packetizing	161
	Sending Data with Packetizing OFF	162
	Disconnecting Dynamic Client (back to Permanent Client)	163
	Clearing Serial Configuration and Deleting Profile	164
Chapter: 10	Raw Socket UDP Mode Configuration165
	Configuring UDP	165
	Configuring Maximum Connections	166
	Sending Data	167
	Adding Connections while Profile is Unmapped	167
	Removing Connections while Profile is Unmapped	169
	Adding Connection on the Fly	170
	Sending Data from Registered UDP Connection (all connections, one at a time)	171
	Sending Data from Not Registered UDP Connection	172
	Sending Data from Serial App	172
	Adding Connection on the Fly Not Allowed	173

Removing Connection on the Fly	174
Configuring Packet Timer	175
Configuring Packet Characters Limit	176
Configuring Packet Size	177
Configuring Turnaround Delay	179
Configuring Hold Time Delay	180
Configuring No Delays	181
Configuring RX-TO-TX Delay	183
Clearing the Serial Profile Configuration and Deleting the Profile . . .	184

INTRODUCTION

1. Introduction

Serial communication is used to exchange information between two hosts. The most used serial communication standards are RS-232, RS-422 and RS-485 2 wire and these are supported on a serial module.

Note that if we disable the port-offline feature at one serial port, in this case, the cable connected trap is send out even if the cable is not connected.

This user guide outline the basic and advanced configuration tasks for a serial interface module.

1.1. Purpose and Scope

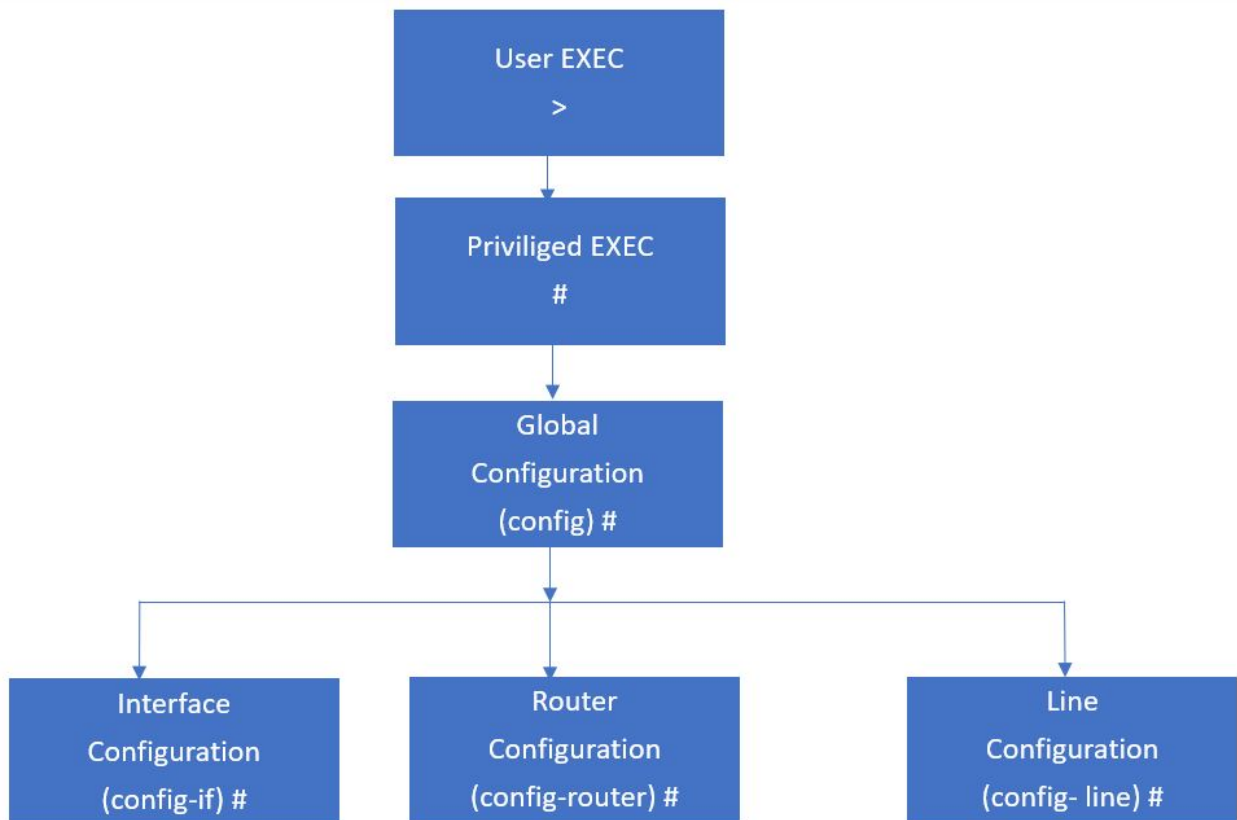
This document describes the serial module configuration. The serial interface has the same system hierarchy and configured in the same manner as any other ISS interface type.

The reader is expected to have a basic knowledge of a serial module as a prerequisite.

1.2. CLI Command Modes

The *CLI* Modes are as follows.

The hierarchical structure of the command modes is as shown on the figure below.

Figure 1: CLI Command Modes

User Exec Mode

Prompt	Access method	Exit Method
Phoenix_Contact>	This is the initial mode to start a session.	logout

Privileged Exec Mode

Prompt	Access method	Exit Method
Phoenix_Contact#	The User EXEC mode command <code>enable</code> is used to enter the Privileged EXEC Mode	To return from the Privileged EXEC mode to User EXEC mode, the command <code>disable</code> is used.

Global Configuration Mode

Prompt	Access method	Exit Method
Phoenix_Contact(config)#	The Privileged EXEC mode command <code>configure terminal</code> is used to enter the Global Configuration Mode.	To return from the Global Configuration Mode to Privileged Mode, the command <code>exit</code> is used.

Interface Configuration Mode

Prompt	Access method	Exit Method
Phoenix_Contact(config-if)#	The Global Configuration mode command <code>interface <interface-type><interface-id></code> is used to enter the Interface Configuration Mode.	To return from the Interface Configuration mode to Global Configuration Mode, the command <code>exit</code> is used. To exit from the Interface Configuration mode to Privileged EXEC Mode, the command <code>end</code> is used.

Port Channel Interface Configuration

Prompt	Access method	Exit Method
Phoenix_Contact(config-if) #	The Global Configuration mode command <code>interface port <port channel-id></code> is used to enter the Port Channel Interface Configuration Mode.	To return from the Port Channel Interface Configuration mode to Global Configuration Mode, the command <code>exit</code> is used. To exit from the Port Channel Interface Configuration mode to Privileged EXEC Mode, the command <code>end</code> is used.

VLAN Interface Configuration Mode

Prompt	Access method	Exit Method
Phoenix_Contact(config-if) #	The Global Configuration mode command <code>interface vlan <vlan id></code> is used to enter the VLAN Interface Configuration Mode.	To return from the VLAN Interface Configuration mode to Global Configuration Mode, the command <code>exit</code> is used. To exit from the VLAN Interface Configuration mode to Privileged EXEC Mode, the command <code>end</code> is used.

MRP Interface Configuration Mode

Prompt	Access method	Exit Method
Phoenix_Contact(config-mrp)#	The Global Configuration mode command <code>mrp ringid 1s</code> is used to enter the MRP Interface Configuration Mode.	To return from the MRP Interface Configuration mode to Global Configuration Mode, the command <code>exit</code> is used. To exit from the MRP Interface Configuration mode to Privileged EXEC Mode, the command <code>end</code> is used.

UFD Configuration Mode

Prompt	Access method	Exit Method
Phoenix_Contact(config-if) #	The Global Configuration mode command <code>ufd group <group-id (1-65535)></code> is used to enter the UFD Interface Configuration Mode.	To return from the UFD Configuration mode to Global Configuration Mode, the command <code>exit</code> is used. To exit from the UFD Configuration mode to Privileged EXEC Mode, the command <code>end</code> is used.

DHCP Pool Configuration Mode

Prompt	Access method	Exit Method
Phoenix_Contact(dhcp-config) #	The Global Configuration mode command Phoenix_Contact(config)# ip dhcp pool <pool number (1-2147483647)> is used to enter the UFD Interface Configuration Mode.	To return from the DHCP Pool Configuration Mode to Global Configuration Mode, the command <code>exit</code> is used. To exit from the DHCP Pool Configuration Mode to Privileged EXEC Mode, the command <code>end</code> is used.

Privilege Levels and Command Access

The following table will list out the commands available for the different user levels in Privileged and User Exec levels.

Command	First Param	Guest	Tech	Admin	Description
archive	download-sw		x	x	Downloads software image
clear					Clears the specified parameters
	alarm	x	x	x	Alarm related information
	au-message	x	x	x	Address update messages related information
	cfa	x	x	x	CFA module related information
	interfaces	x	x	x	Protocol specific configuration of the interface
	meter-stats	x	x	x	Specific configuration for meter
	poe	x	x	x	PoE related configuration

Command	First Param	Guest	Tech	Admin	Description
	screen	x	x	x	Screen information
	ip		x	x	IP related configuration
	line		x	x	Configures line information
	logs		x	x	Log information
	protocol		x	x	Clears the specified protocol counters
	spanning-tree		x	x	Spanning tree related configuration
	tcp		x	x	TCP related configuration
clock	set		x	x	Sets the system clock value
config-restore					Configures the restore option
	flash		x	x	File in flash to be used for restoration
	norestore		x	x	No configuration restore
	remote		x	x	Remote location configuration
configure	terminal		x	x	Configures the terminal
copy			x	x	Various copy options
debug					Configures trace for the protocol
	ip	x	x	x	IP related configuration
	show	x	x	x	Show mempool status
	sntp	x	x	x	SNTP related configuration
	crypto		x	x	Crypto related information
	cybsec		x	x	Cybsec related information
	dot1x		x	x	PNAC related configuration
	etherchannel		x	x	Etherchannel related information
	firewall		x	x	Firewall related configuration
	garp		x	x	GARP related configuration
	interface		x	x	Configures trace for the interface management
	lACP		x	x	LACP related configuration
	lldp		x	x	LLDP related configuration

Command	First Param	Guest	Tech	Admin	Description
	lms		x	x	LCD notification server
	nat		x	x	Network Address Translation related configuration
	np		x	x	NPAPI configuration
	ptp		x	x	Precision time protocol related configuration
	qos		x	x	QOS related configuration
	security		x	x	Security related configuration
	spanning-tree		x	x	Spanning tree related protocol configuration
	ssh		x	x	SSH related configuration
	tacm		x	x	Transmission and admission control related configuration
	vlan		x	x	VLAN related configuration
display firewall rules				x	Display firewall rules
dot1x	clear	x	x	x	Clear dot1x configuration
	initialize		x	x	State machine and fresh authentication configuration
	re-authenticate		x	x	Re-authentication
dump					Display memory content from the given memory location
	mem		x	x	Dump memory
	que		x	x	Show the queue related information
	sem		x	x	Show the semaphore related information
	task		x	x	Show the task related information
egress bridge			x	x	
end			x	x	Exit to the privileged Exec (#) mode

Command	First Param	Guest	Tech	Admin	Description
erase			x	x	Clears the contents of the startup configuration
exit		x	x	x	Logout
factory reset				x	Reset to factory default configuration
factory reset	users			x	Reset all users on switch
firmware			x	x	Upgrades firmware
generate	tech		x	x	Generate the tech report of various system resources and protocol states for debugging
help		x	x	x	Displays help for commands
ip	igmp snooping clear counters	x	x	x	Clears the IGMP snooping statistics
	clear counters		x	x	Clear operation
	dhcp		x	x	DHCP related configuration
	pim		x	x	PIM related configuration
	ssh		x	x	SSH related information
listuser			x	x	List the user, mode and groups
lock			x	x	Lock the console
logout		x	x	x	Logout
memtrace			x	x	Configures memtrace
no ip					IP related information
	dhcp		x	x	DHCP related configuration
	ssh		x	x	SSH related information
no debug					Configures trace for the module
	ip	x	x	x	Stops debugging on IGMP or PIM
	sntp	x	x	x	Stops debugging on SNTP related configurations
	additional options...		x	x	Stops debugging for other options
ping					

Command	First Param	Guest	Tech	Admin	Description
	A.B.C.D	x	x	x	Ping host
	ip dns host name	x	x	x	Ping host
	ip A.B.C.D	x	x	x	Ping host
	vrf	x	x	x	Ping vrf instance
readarpfromHardware ip	A.B.C.D		x	x	Reads the arp for the given IP
readregister			x	x	Reads the value of the register from the hardware
release dhcp			x	x	Performs release operation
reload			x	x	Restarts the switch
renew dhcp			x	x	Performs renew operation
run script			x	x	Runs CLI commands
shell				x	Shell to Linux prompt
show		x	x	x	Shows configuration or information
sleep		x	x	x	Puts the command prompt to sleep
ssl				x	Configures secure sockets layer related parameters
snmpwalk mib					Allows the user to view Management Information Base related configuration.
	name	x	x	x	
	oid	x	x	x	
traceroute					Traces route to the destination IP
	A.B.C.D		x	x	
write			x	x	Writes the running-config to a flash file
writeregister			x	x	writes in the specified register

Configuration Terminal Access

The Guest user level does not have access to the configuration terminal.

The Administration level has access to all commands in the configuration terminal.

The Technical level has access to all commands in the configuration terminal with the following exceptions listed below.

- bridge-mode
- enableuser
- mst
- password
- traffic

1.3. CLI Document Convention

To provide a consistent user experience, this *CLI* document convention adheres to the Industry Standard *CLI* syntax.

In addition, the font and format are updated to show *DITA* / Structured Framemaker 2019 layout.

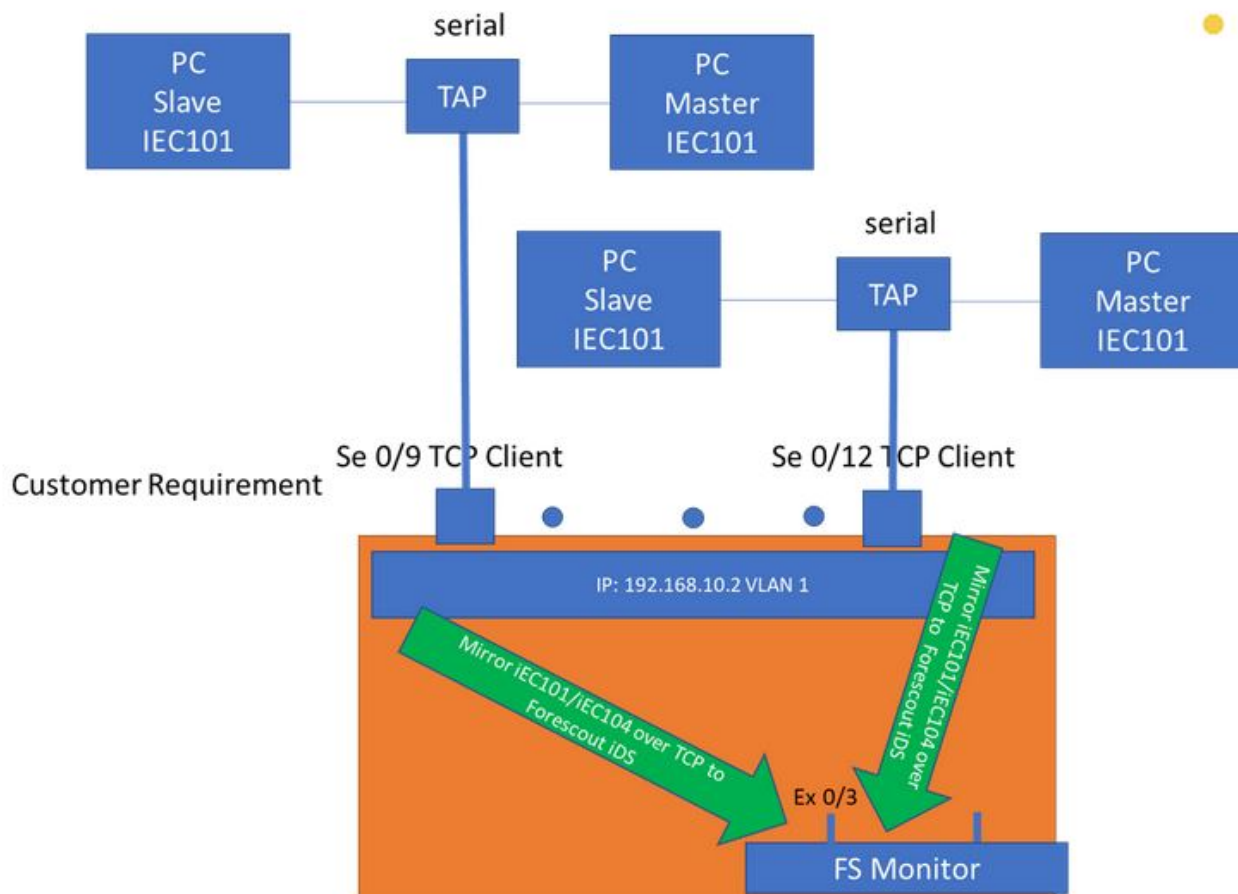
Convention	Usage	DESCRIPTION
<i>Italics</i>	User inputs for <i>CLI</i> command	<code>configure terminal</code>
Font as shown	Syntax of the <i>CLI</i> command	<code>configure terminal</code>
<>	Parameter inside the brackets <> indicate the Input fields of syntax	<code><integer (100-1000)></code>
[]	Parameter inside [] indicate optional fields of syntax	<code>show split-horizon [all]</code>
{ }	Grouping parameters in the syntax	<code>ip address <ip-address> [secondary {node0 node1}]</code>
	Separating grouped parameters in the syntax	<code>set http authentication-scheme {default basic digest}</code>
Font & format as shown	Example & CLI command outputs	<pre>Phoenix_Contact# show split-horizon interface 1 Ingress Port VlanId StorageType Egress List ===== Gi0/1 - Volatile Gi0/2,Gi0/3,Gi0/6</pre>
Note	Notes	NOTE: All commands are case-sensitive

2. Serial Port Offline Indication

This section describes the use of the serial port offline indication feature use.

CONTEXT:

The Serial port offline indication feature is used to monitor whether or not a serial tap is connected to a serial interface with this feature enabled. The topology is as follows.



2.1. Configuration

1. **CLI:** Enable the serial port offline indication feature.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface serial 0/9
```

- Enable the serial port offline indication feature.

```
Phoenix_Contact(config-serial-if)# serial-port-offline enable
```

2. **WebUI:** Enable the serial port offline indication feature.

FOR EXAMPLE: Type the following:

- The feature shall also be configured from WebUI from the Serial Interface Configuration page as shown below.

The feature can be enabled by choosing the **Enabled/Disabled** option from the **Serial Line Monitoring** column.

- When disabled, the **Serial Cable State** would display **Not Applicable (NA)**.
- When enabled, the state shall be **Connected/Disconnected** based on presence of Serial tap.

3. **Alarms & SNMP Traps**

FOR EXAMPLE: When the feature is enabled, Alarm is set to indicate whenever there is no serial tap connected. The alarm would be cleared when a serial tap is connected. The Alarm ID shall indicate the interface where the tap is absent/present by adding the interface index to the base Alarm ID: SNMP traps are triggered as well, when disconnecting/connecting the serial tap with serial interface index and status of the cable presence. The SNMP trap would be raised with the trap object '**serialCableStateTrap**'. The trap message also binds the values of the object '**ifIndex**' and '**ifMainSerialIfCableConnected**'.

4. **LCD & Port Map**

FOR EXAMPLE: The value of the '**ifMainSerialIfCableConnected**' is reflected in the Port Map status in WebUI and LCD. The serial interface status shall always be displayed as green if the feature is not enabled.

If the feature is enabled, then status shall be:

- green if the tap is connected, and
- red if the tap is disconnected.



2.2. Verification

1. The configuration can be verified using the show command as shown below.

FOR EXAMPLE: Type the following:

```

TESTPROMPT(config-serial-if)# end
TESTPROMPT#
TESTPROMPT#
TESTPROMPT# sh int se 0/9
Interface name      : serial 0/9
Admin status       : Up
Baudrate            : 9600
Data bits           : 8
Parity              : None
Stop bits           : 1
Flow control        : None
Interface type      : rs232
Termination resistor : Disabled
Force HD            : Disabled
Turn around delay   : 0 secs
Hold time           : 0 secs
Post Tx delay       : 0 secs
Rx to Tx delay      : 0 secs
Serial Monitor status : Enabled
Serial Cable status : Disconnected

TESTPROMPT#
TESTPROMPT#
TESTPROMPT# sh int se 0/10
Interface name      : serial 0/10
Admin status       : Up
Baudrate            : 9600
Data bits           : 8
Parity              : None
Stop bits           : 1
Flow control        : None
Interface type      : rs232
Termination resistor : Disabled
Force HD            : Disabled
Turn around delay   : 0 secs
Hold time           : 0 secs
Post Tx delay       : 0 secs
Rx to Tx delay      : 0 secs
Serial Monitor status : Enabled
Serial Cable status : Connected

TESTPROMPT# █

```

2. The functionality shall be verified by checking the Alarms history under the switch alarms.

FOR EXAMPLE: Type the following:

```

3023 SWITCH Sep/5/15:05:02 CLR Gi0/23 Interface link state UP Critical
3212 SWITCH Sep/5/15:13:58 SET Ser0/10 Serial Line status DOWN Critical
3212 SWITCH Sep/5/15:14:00 CLR Ser0/10 Serial Line status UP Critical
3017 SWITCH Sep/5/21:23:57 SET Gi0/17 Interface link state DOWN Critical
3017 SWITCH Sep/5/21:24:00 CLR Gi0/17 Interface link state UP Critical

TESTPROMPT#
TESTPROMPT#
TESTPROMPT#

```

NOTE: When the feature is disabled after already enabled, then any existing alarms in the alarm history and Syslog history shall be cleared. There is no way to distinguish between an actual cleared alarm and the alarm cleared due to disabling of the feature. This is a limitation of the current design. However, the scenario of enabling and disabling is not foreseen often in the field.

3. Modbus Client and Server Roles

This section describes the network topology of the RAPTOR as a Modbus Client or Server.

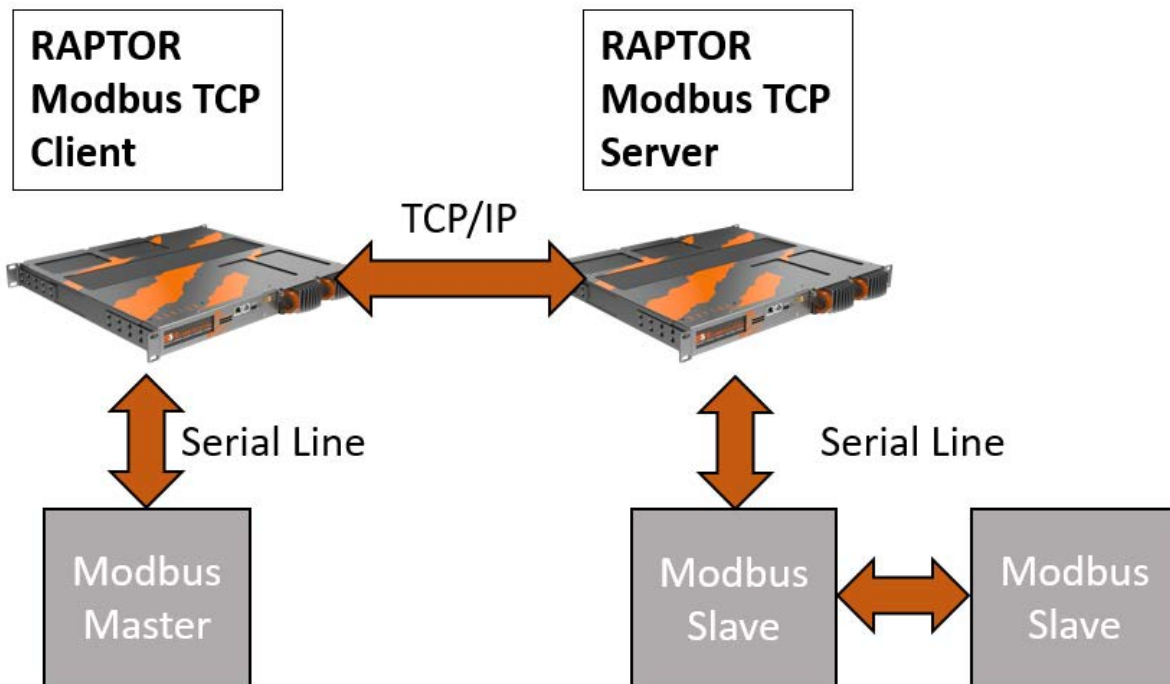
3.1. Modbus Client and Server Gateway Responsibilities

RAPTOR Client and server gateways do the following:

- Translate TCP messages to serial messages and vice versa.
- Act like a pipe encapsulating serial modbus messages in TCP packets.
- Do not check the validity of function codes and data inside messages. This is done in master and slave applications.
- Create exceptions and send them to the proper destination.

The RAPTOR should be used as a client or a server when the Master and Slave are not native TCP speakers.

Figure 1: RAPTOR as a Modbus Client or Server

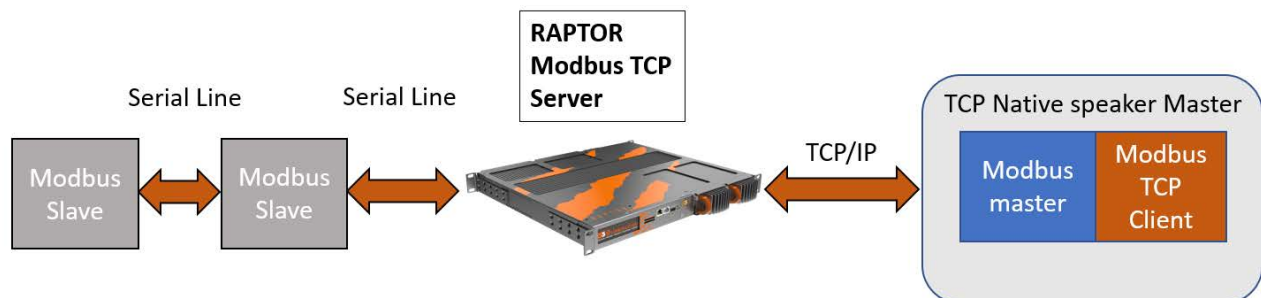


3.2. Modbus Server Role

The RAPTOR when used as a modbus server can do the following:

- Accept up to 64 connections from clients
- Work with multiple serial interfaces simultaneously
- Support up to 247 slaves on all serial interfaces. Please note that a much lower number of slaves is recommended for reasons surrounding serial link contention.
- Receive TCP modbus requests from TCP modbus clients
- Validates received messages for length, function code and slave ID
- Returns proper exception in the case of validation errors and if return exceptions are enabled
- If there are no active transactions on the interface:
 - Converts the TCP modbus message to a serial modbus message
 - Sends serial messages to the proper serial interface
 - Keeps the original message for response validation
 - Starts the response timer
- If there is an active transaction on the interface, stores the request in a queue that can hold up to 16 requests.
- When the active transaction is over (by receiving a response or no response in the configured time), initiates new transaction by pulling the next available request from the requests queue.
- Validates received response (length, CRC, function code, slave ID) against the stored original request.
- Converts received responses to TCP messages.
- Sends the TCP messages to the proper master.

Figure 2: Modbus TCP Server Topology



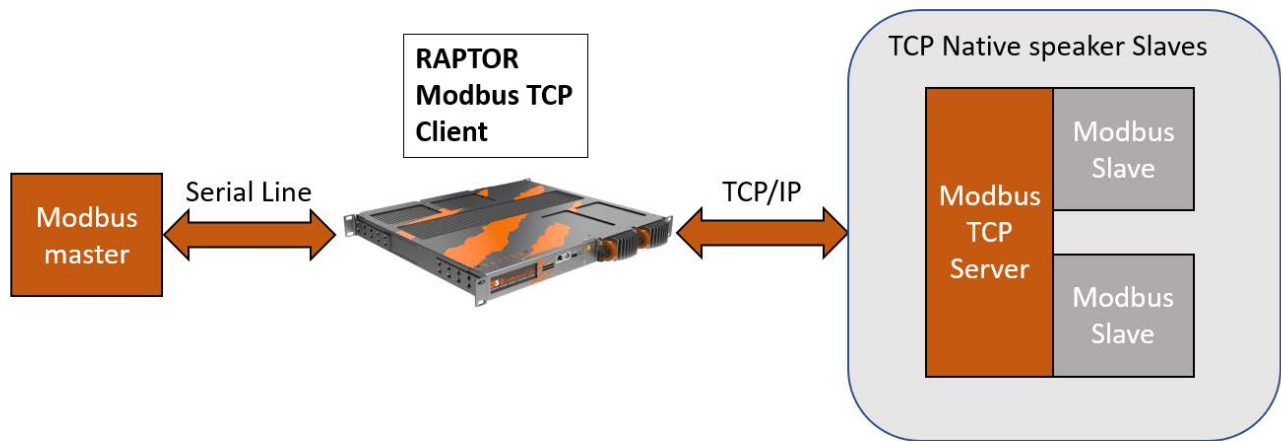
3.3. Modbus Client Role

The RAPTOR when used as a modbus client can do the following:

- Receives serial commands from the serial master
- Validates received messages (length, CRC, function code, slave ID)

- Returns the proper exception in the case of validation error and if return exceptions are enabled.
- Converts the serial modbus message to a TCP modbus message.
- Sends the TCP message to the server while keeping the original request for the response validation.
- Receives a response, and validates it using the original request.
- Converts received TCP messages to serial messages.
- Sends converted serial messages to the serial master.
- Starts the response timer.
- In case of no response from the server, sends proper exception to the master.

Figure 3: Modbus TCP Client Topology



3.4. Modbus Function and Exception Codes

A response function code is valid if:

- The response function code matches the request function code.
- The response function code is an exception created using the request function code.

The exception function code is equivalent to the request function code with its most significant bit set to logic 1. Example: the 0x83 code is the exception for the function code 0x03.

Table 1: Exception codes generated by a Slave (Sheet 1 of 2)

Name	Description
ILLEGAL_FUNCTION	The function code is not an allowable action for the slave.
ILLEGAL_DATA_ADDRESS	The data address is not an allowable address for the slave.

Table 1: Exception codes generated by a Slave (Continued) (Sheet 2 of 2)

Name	Description
ILLEGAL_DATA_VALUE	The data field is not an allowable value for the slave.
ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so.
SLAVE_DEVICE_BUSY	The slave is engaged in processing a long-duration program command.
MEMORY_PARITY_ERROR	The extended file area failed to pass a consistency check.

Table 2: Exception codes generated by a Gateway

Name	Description
ILLEGAL_FUNCTION	The function code is not an allowable action for the slave. RAPTOR: returned if function code does not match the broadcast command.
SERVER_DEVICE_FAILURE	An unrecoverable error occurred while the slave was attempting to perform the requested actions. RAPTOR: not returned
SERVER_DEVICE_BUSY	The server is busy with other commands and has no place to store the request. RAPTOR: returned if hold time is over, no place to store the request
GATEWAY_PATH_UNAVAILABLE	The gateway is mis-configured or overloaded. RAPTOR: returned if slave is not in the slaves list. No connection to a server.
GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND	No response was obtained from the target device. RAPTOR: no response was obtained from the target device.

4. MODBUS TCP Server Mode Configuration

This section lists the *CLI* configuration steps for TCP Server Configuration for MODBUS connection.

CONTEXT:

The MODBUS is a request-response protocol. A client (can request the MODBUS server to perform an action and the server will respond with that action.

The client:

- Connects to the MODBUS server IP address on port 502 (default)
- Sends MODBUS TCP request packets to the server

The MODBUS server:

- Receives the MODBUS TCP request packets from the server
- Takes action as per the request

1. Configure VLAN 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure VLAN 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact(config-if)# end
```

```
Phoenix_Contact# show ip interface
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-serial-if)# baud-rate 115200
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/10
-   For example, configure the baud rate.
Phoenix_Contact(config-serial-if)# baud-rate 57600
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

4.1. Listening to the Master

1. Start a server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# local server port modbus
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# add slave-id
1,12,13,14,15,16,17,18,19,20 interface serial 0/9
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 0
KeepAlive interval (sec) : 240
```

```

Max pending messages      : 16
Current pending messages: 0
Send exceptions           : YES
RXed from slave msg      : 0
TXed to slave msg        : 0
RXed from master msg     : 0
TXed to master msg       : 0
Exceptions created        : 0
    
```

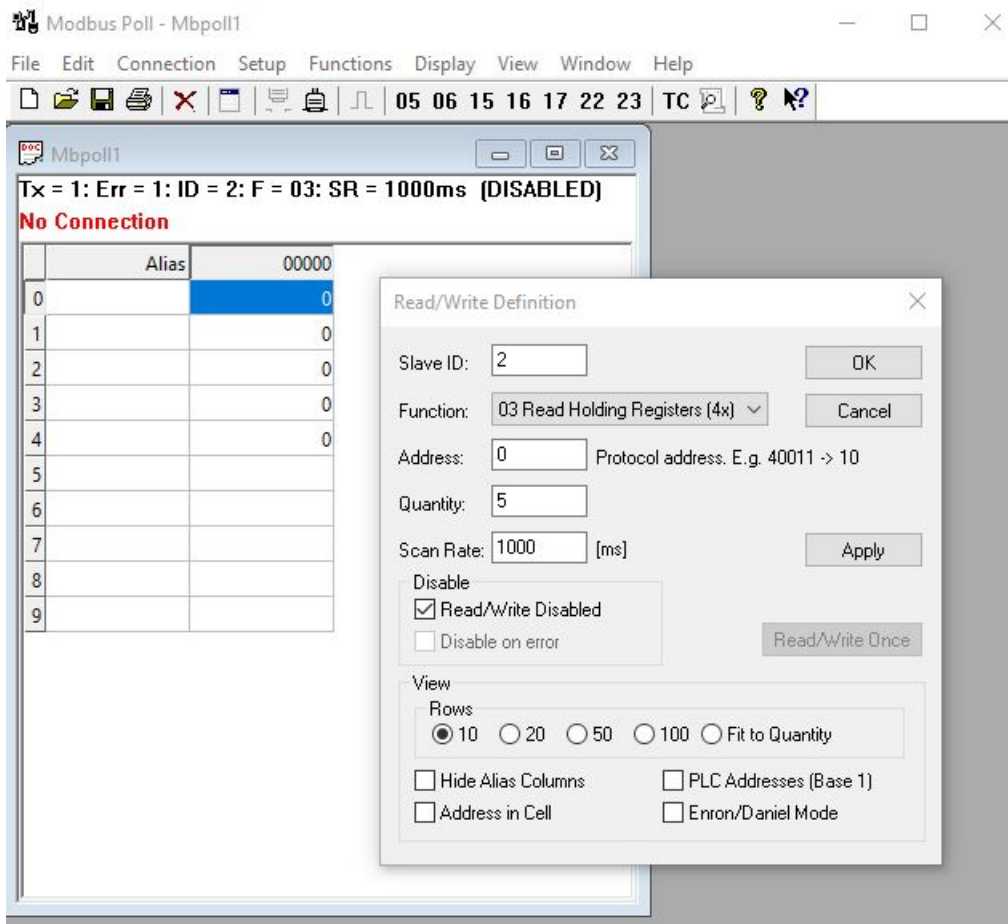
3. Verify default parameters.

4.2. Connecting Master to the Server

1. Open Master / Client Application. Go to the **Read/Write Definition** screen and check **Read/Write Disabled**. Click **OK**.

FOR EXAMPLE: Type the following:

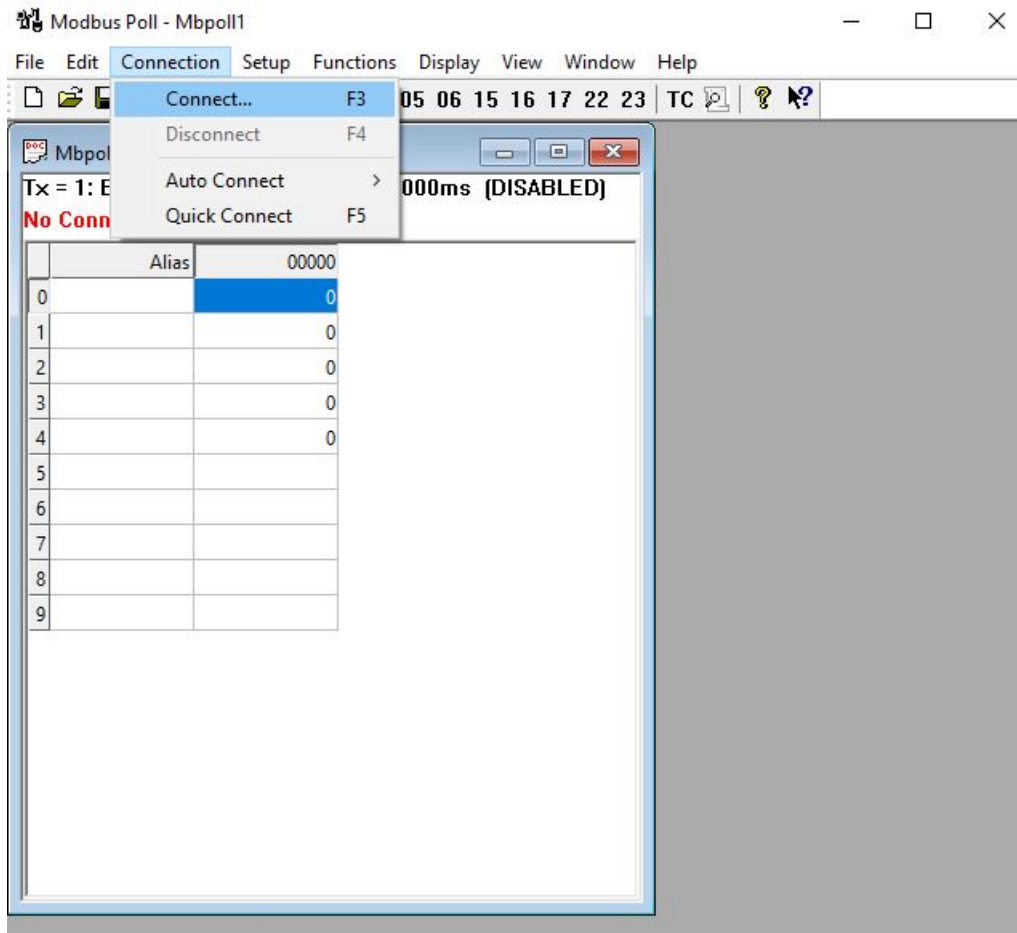
Figure 1: Connecting Master to the Server



2. Go to the **Connection**.

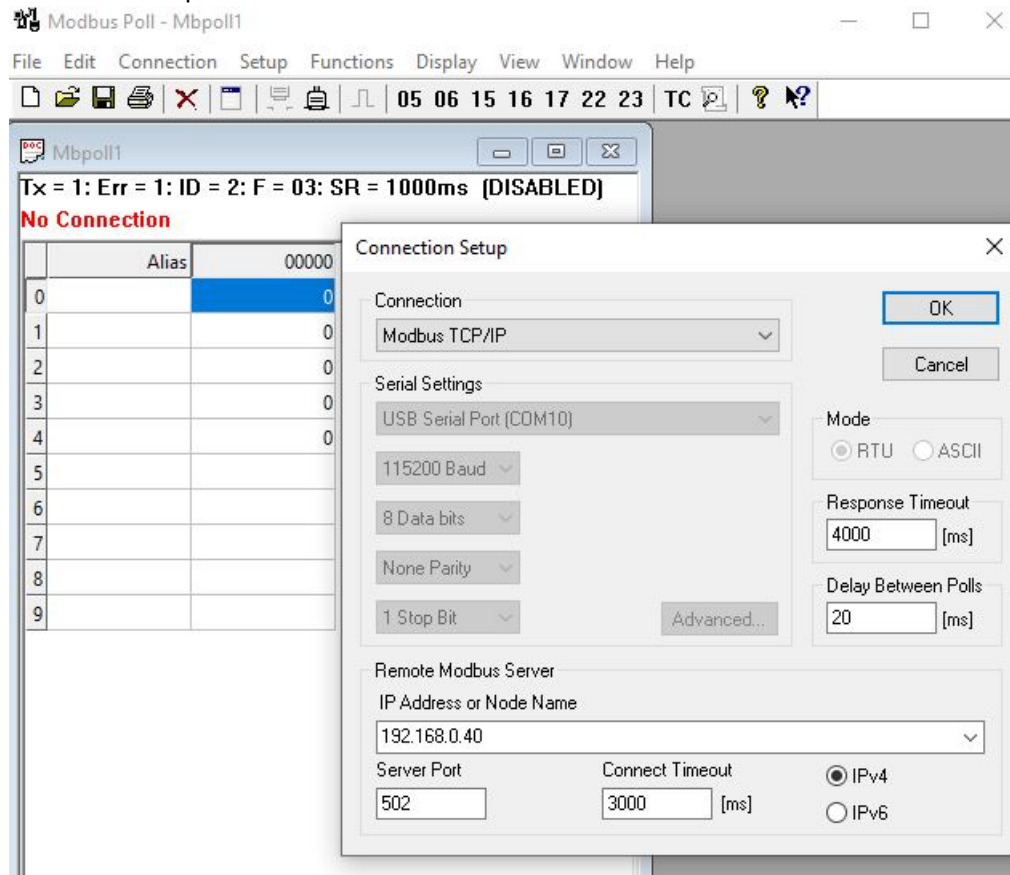
FOR EXAMPLE: Type the following:

Figure 2: Connection



3. In **Connection Setup** dialog box, go to **Connection** field and set connection type and Server's IP and Port. Click **OK**.

FOR EXAMPLE: Type the following:

Figure 3: Connection Setup

- Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role           : TCP Server
Serial interface : 9
Slaves ID      : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 1
Client 1 IP : 192.168.0.100
```

```

Client 1 port          : 59418
Client 1 byte counter  : 0
KeepAlive interval (sec): 240
Max pending messages   : 16
Current pending messages: 0
Send exceptions        : YES
RXed from slave msg    : 0
TXed to slave msg     : 0
RXed from master msg   : 0
TXed to master msg    : 0
Exceptions created     : 0

```

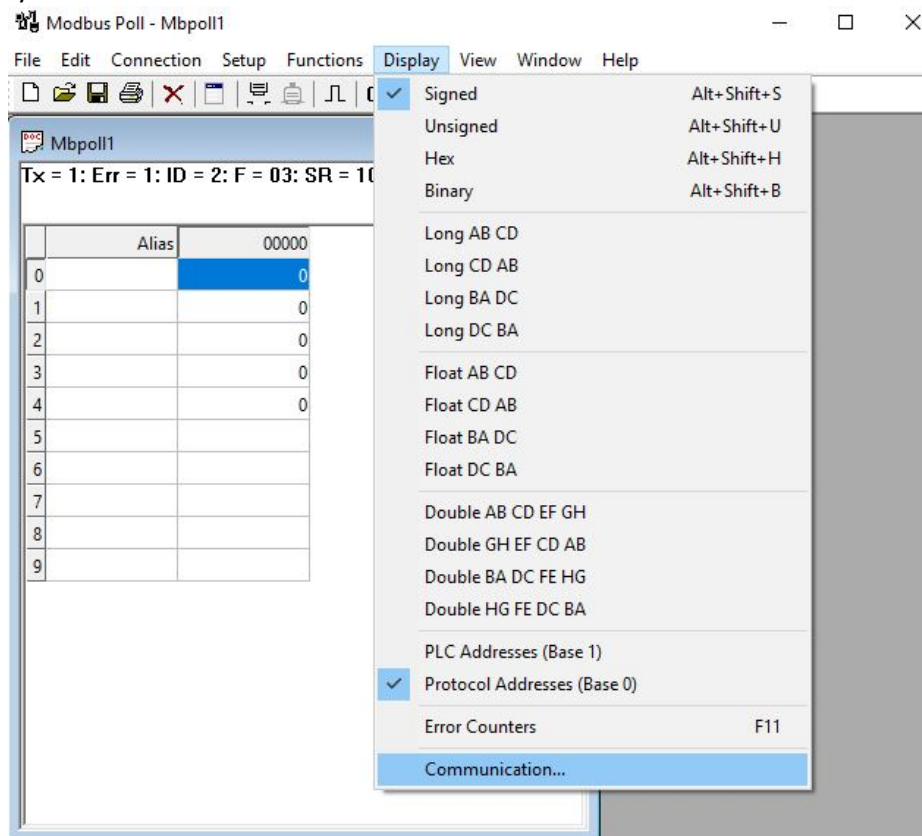
5. Verify that response time in the Master is bigger then in the Server.

4.3. Sending Request from Master to not Configured Slave

1. In the Master, click **Display** menu and click **Communication Traffic**. Click **OK**.

FOR EXAMPLE: Type the following:

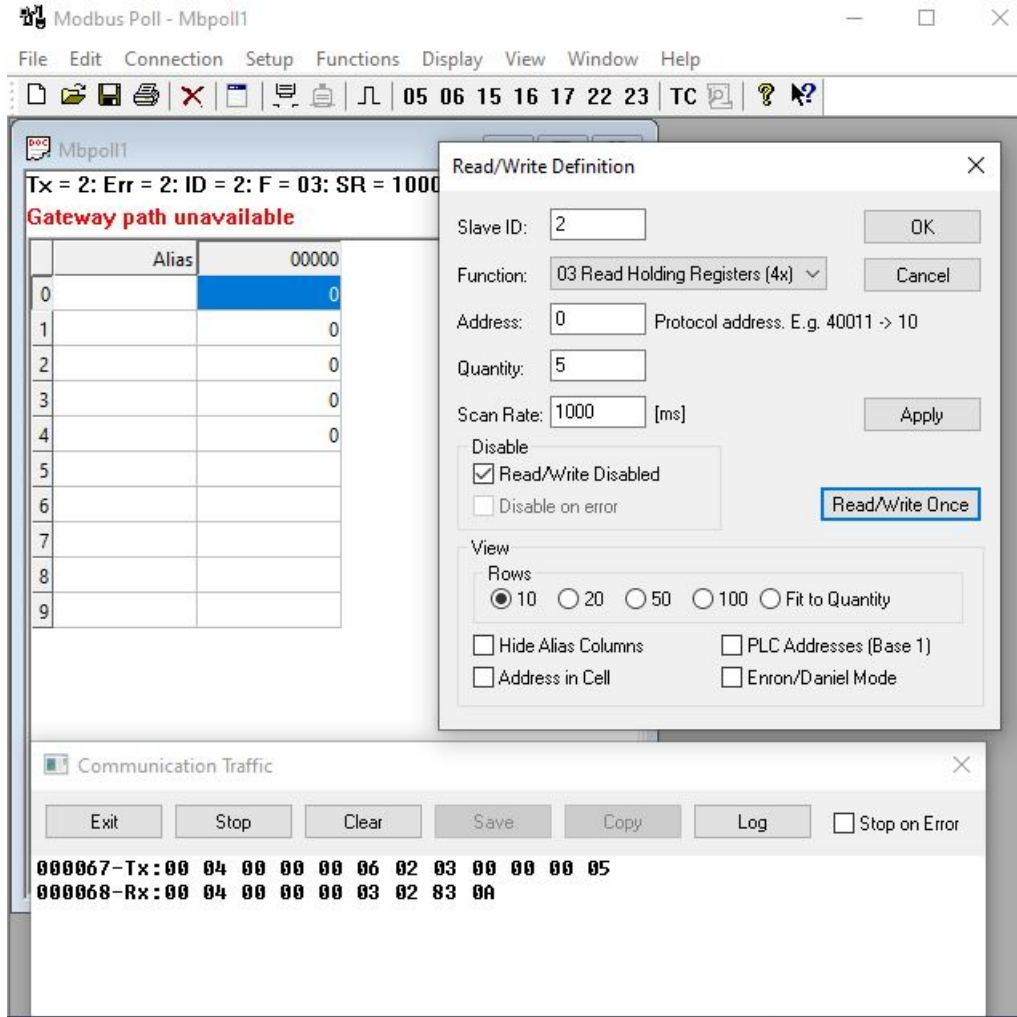
Figure 4: Display Menu



- Click **03 Read Holding Registers** from Slave ID 1. Use Slave ID that is not configured.

FOR EXAMPLE: Type the following:

Figure 5: 03 Read Holding Registers



- Verify receiving exception.
- Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the track of sent exceptions.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
```

```

Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec)  : 0
Local Server Port     : 502
Max clients allowed   : 64
Clients number        : 1
Client 1 IP           : 192.168.0.100
Client 1 port         : 59418
Client 1 byte counter : 12
KeepAlive interval (sec) : 240
Max pending messages  : 16
Current pending messages : 0
Send exceptions       : YES
RXed from slave msg   : 0
TXed to slave msg    : 1
RXed from master msg  : 1
TXed to master msg   : 1
Exceptions created    : 1

```

4.4. Configuring Send Exceptions

1. Disable exceptions to be send to the Master.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# transmit-exception
disable
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```



```

Protocol : Modbus
Role      : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec): 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 1
KeepAlive interval (sec): 240
Max pending messages : 16
Current pending messages: 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 0
RXed from master msg : 0
TXed to master msg : 0
Exceptions created : 1

```

3. Verify the exception from the server is not sent to the Master, but was created.
4. Verify Master announce "Response Time Out" exception.
5. Enable exception to be send to the Master.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# transmit-exception
enable
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

6. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the track of sent exception.

FOR EXAMPLE: Type the following:

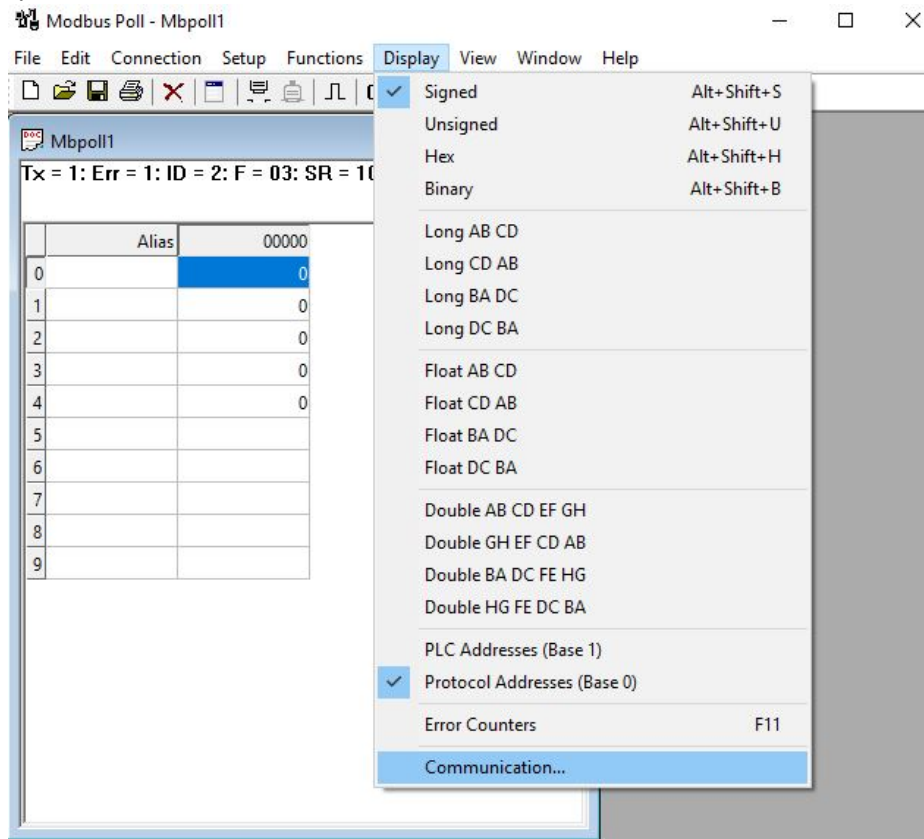
```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 1
KeepAlive interval (sec): 240
Max pending messages : 16
Current pending messages: 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 0
RXed from master msg : 0
TXed to master msg : 0
Exceptions created : 1
```

4.5. Sending Request from Master to Slave, Which Is Offline

1. In the Master, click **Display** menu and click **Communication Traffic**. Click **OK**.

FOR EXAMPLE: Type the following:

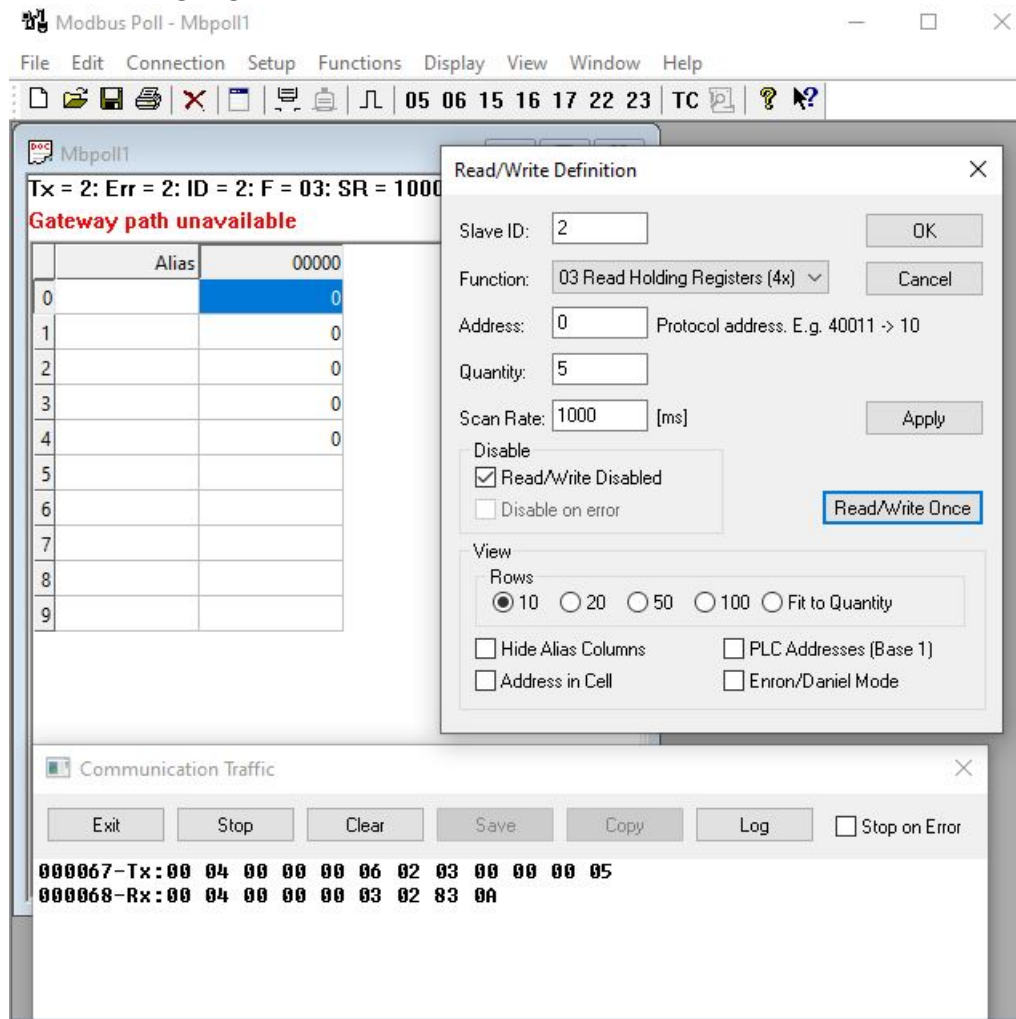
Figure 6: Display Menu



2. Click **03 Read Holding Registers** from Slave ID 1.

FOR EXAMPLE: Perform the following:

Figure 7: 03 Read Holding Registers



3. Verify receiving exception.
4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the track of sent exceptions.

FOR EXAMPLE: Type the following:

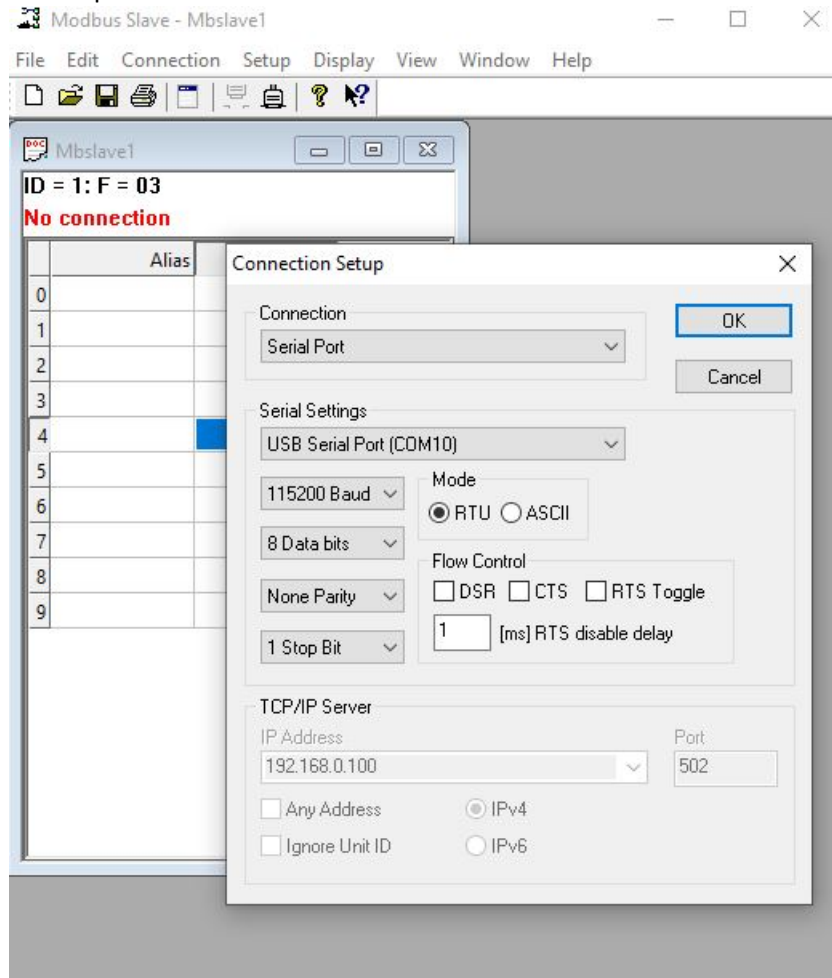
```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role           : TCP Server
Serial interface : 9
Slaves ID      : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec)   : 0
Local Server Port      : 502
Max clients allowed    : 64
Clients number         : 1
Client 1 IP            : 192.168.0.100
Client 1 port          : 59418
Client 1 byte counter  : 12
KeepAlive interval (sec): 240
Max pending messages   : 16
Current pending messages: 0
Send exceptions        : YES
RXed from slave msg    : 0
TXed to slave msg     : 1
RXed from master msg   : 1
TXed to master msg     : 1
Exceptions created     : 1
```

4.6. Sending Request from Master to Slave

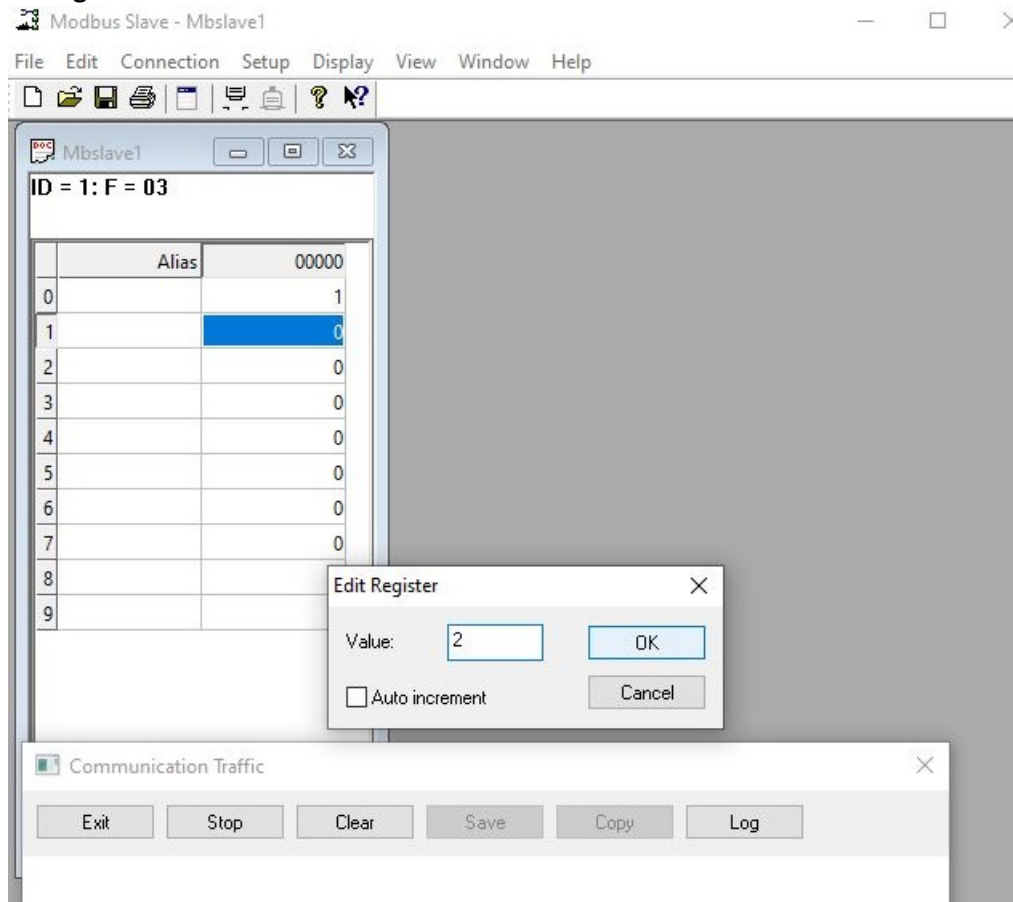
1. In the Master, open **Connection Setup** screen, apply serial line related configuration, and connect by clicking **Communication**. Click **OK**.

FOR EXAMPLE: Type the following:

Figure 8: Connection Setup

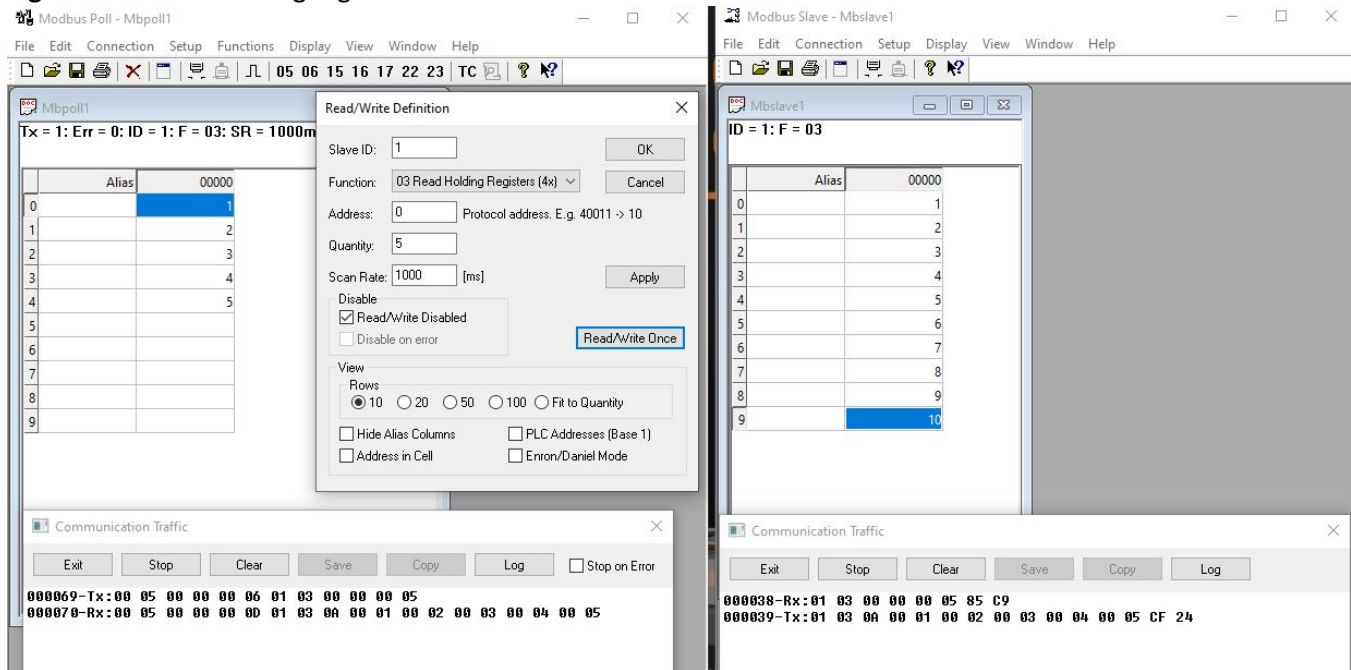
- Apply wanted values to the Slave's hold registers by double clicking on single register.
FOR EXAMPLE: Type the following:

Figure 9: Edit Registers



3. In the Master, click **Display** menu and click **Communication Traffic**. Click **OK**.
4. Read the holding registers from the slave.
FOR EXAMPLE: Perform the following.

Figure 10: Read holding registers from the slave



- Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the track of sent exceptions.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus

Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 1
Client 1 IP : 192.168.0.100
Client 1 port : 61121
Client 1 byte counter : 48
KeepAlive interval (sec) : 240
Max pending messages : 16
Current pending messages : 0
```



```

Send exceptions          : YES
RXed from slave msg     : 2
TXed to slave msg      : 2
RXed from master msg    : 2
TXed to master msg     : 2
Exceptions created      : 1

```

4.7. Using not Native Modbus Port

1. Start a server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# local server port modbus
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role          : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID      : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Local Server Port : 15010
```

```
Max clients allowed : 64
```

```
Clients number : 0
```

```
KeepAlive interval (sec): 240
```

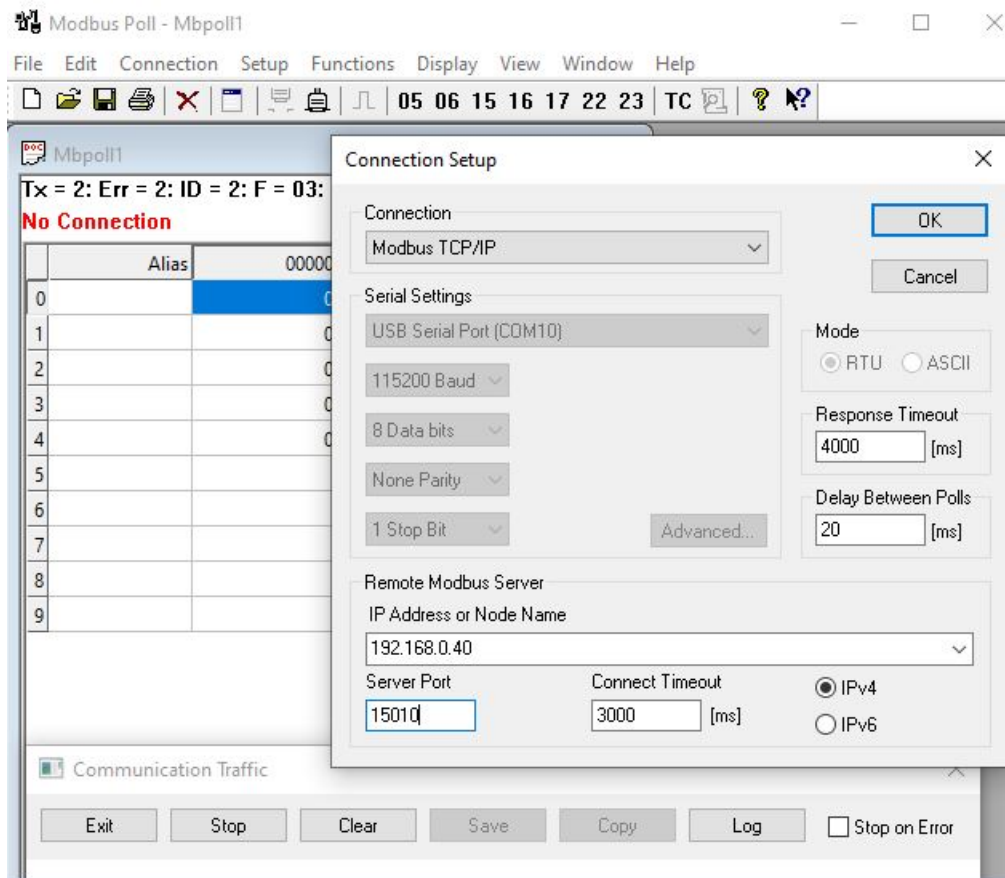
```

Max pending messages      : 16
Current pending messages: 0
Send exceptions           : YES
RXed from slave msg      : 0
TXed to slave msg        : 0
RXed from master msg     : 0
TXed to master msg       : 0
Exceptions created        : 0
    
```

3. Disconnect the Master, open **Connection**, change the server's port number, and reconnect with master using new server's port.

FOR EXAMPLE: Perform the following.

Figure 11: Connection Setup



4.8. Connecting Second Master

1. Perform the same procedure as in section “Connecting Master to the Server”.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify it. The client number should be 2 and the IPs and ports of both clients should be shown.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 55470
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 55481
Client 2 byte counter : 564
KeepAlive interval (sec) : 240
Max pending messages : 16
Current pending messages : 0
Send exceptions : YES
RXed from slave msg : 47
TXed to slave msg : 47
RXed from master msg : 47
TXed to master msg : 47
Exceptions created : 0
```

4.9. Sending keep-alive to Both Masters

1. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that keepalive time is 240 seconds (default value). The keep alive messages should be sent every 240 seconds to both clients.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Local Server Port : 502
```

```
Max clients allowed : 64
```

```
Clients number : 2
```

```
Client 1 IP : 192.168.0.100
```

```
Client 1 port : 55470
```

```
Client 1 byte counter : 0
```

```
Client 2 IP : 192.168.0.100
```

```
Client 2 port : 55481
```

```
Client 2 byte counter : 564
```

```
KeepAlive interval (sec): 240
```

```
Max pending messages : 16
```

```
Current pending messages: 0
```

```
Send exceptions : YES
```

```
RXed from slave msg : 47
```

```
TXed to slave msg : 47
```

```
RXed from master msg : 47
```

```
TXed to master msg : 47
```

```
Exceptions created : 0
```

```
... 86099.179359 192.168.0.40 192.168.0.100 TCP 60 [TCP Keep-Alive] 502 → 55470 [ACK] Seq=0 Ack=1 Win=4380 Len=0
... 86099.179459 192.168.0.100 192.168.0.40 TCP 54 [TCP Keep-Alive ACK] 55470 → 502 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
... 86172.011858 192.168.0.40 192.168.0.100 TCP 60 [TCP Keep-Alive] 502 → 55481 [ACK] Seq=1363 Ack=565 Win=3816 Len=0
... 86172.011916 192.168.0.100 192.168.0.40 TCP 54 [TCP Keep-Alive ACK] 55481 → 502 [ACK] Seq=565 Ack=1364 Win=2100992 Len=0
... 86339.820863 192.168.0.40 192.168.0.100 TCP 60 [TCP Keep-Alive] 502 → 55470 [ACK] Seq=0 Ack=1 Win=4380 Len=0
... 86339.820978 192.168.0.100 192.168.0.40 TCP 54 [TCP Keep-Alive ACK] 55470 → 502 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
```

86339 - 86099 = 240

2. Change keep-alive time to 60 seconds (minimum).

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# keep-alive timeout 60
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

3. Reconnect Masters.

4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that keepalive time is 240 seconds (default value). The keep alive messages should be sent every 240 seconds to both clients.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Local Server Port : 502
```

```
Max clients allowed : 64
```

```
Clients number : 2
```

```
Client 1 IP : 192.168.0.100
```

```
Client 1 port : 55470
```

```
Client 1 byte counter : 0
```

```
Client 2 IP : 192.168.0.100
```

```
Client 2 port : 55481
```

```
Client 2 byte counter : 564
```

```
KeepAlive interval (sec): 60
```

```
Max pending messages : 16
```

```

Current pending messages: 0
Send exceptions           : YES
RXed from slave msg      : 47
TXed to slave msg        : 47
RXed from master msg     : 47
TXed to master msg       : 47
Exceptions created        : 0

```

4.10. Setting max client connections to 2

1. Change maximum client connections to 2.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# max client connections 2
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the maximum clients number.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Local Server Port : 15010
```

```
Max clients allowed : 2
```

```
Clients number : 0
```

```

Client 1 IP           : 192.168.0.100
Client 1 port         : 55470
Client 1 byte counter : 0
Client 2 IP           : 192.168.0.100
Client 2 port         : 55481
Client 2 byte counter : 564
KeepAlive interval (sec): 60
Max pending messages : 16
Current pending messages: 0
Send exceptions       : YES
RXed from slave msg  : 0
TXed to slave msg    : 0
RXed from master msg : 0
TXed to master msg   : 0
Exceptions created    : 0

```

3. Connect both Masters to the Server. No problems should be observed.
4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the number of connected clients.

FOR EXAMPLE: Type the following:

```

Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role           : TCP Server
Serial interface : 9
Slaves ID      : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 15010
Max clients allowed : 2
Clients number : 2
Client 1 IP           : 192.168.0.100
Client 1 port         : 57904
Client 1 byte counter : 0
Client 2 IP           : 192.168.0.100
Client 2 port         : 57908
Client 2 byte counter : 0

```

```
KeepAlive interval (sec): 60
Max pending messages    : 16
Current pending messages: 0
Send exceptions         : YES
RXed from slave msg     : 0
TXed to slave msg      : 0
RXed from master msg   : 0
TXed to master msg     : 0
Exceptions created     : 0
```

4.11. Connecting with Third Master

1. Try to connect with the third Master. There is no visible error in the third Master application that can be seen, but the connection should be refused.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the number of connected clients. It should be 2 as the maximum clients number.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role           : TCP Server
Serial interface : 9
Slaves ID      : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 15010
Max clients allowed : 2
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 55470
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 55481
Client 2 byte counter : 564
KeepAlive interval (sec): 60
Max pending messages : 16
```



```

Current pending messages: 0
Send exceptions           : YES
RXed from slave msg      : 0
TXed to slave msg        : 0
RXed from master msg     : 0
TXed to master msg       : 0
Exceptions created       : 0

```

4.12. Adding Slaves to Second Serial Interface

1. Add a second serial interface to the profile by adding slaves to it.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# add slave-id
2,21,22,23,24,25 interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the number of connected clients. It should be 2 as the maximum clients number.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Serial interface : 10
```

```
Slaves ID : 2 21 22 23 24 25
```

```

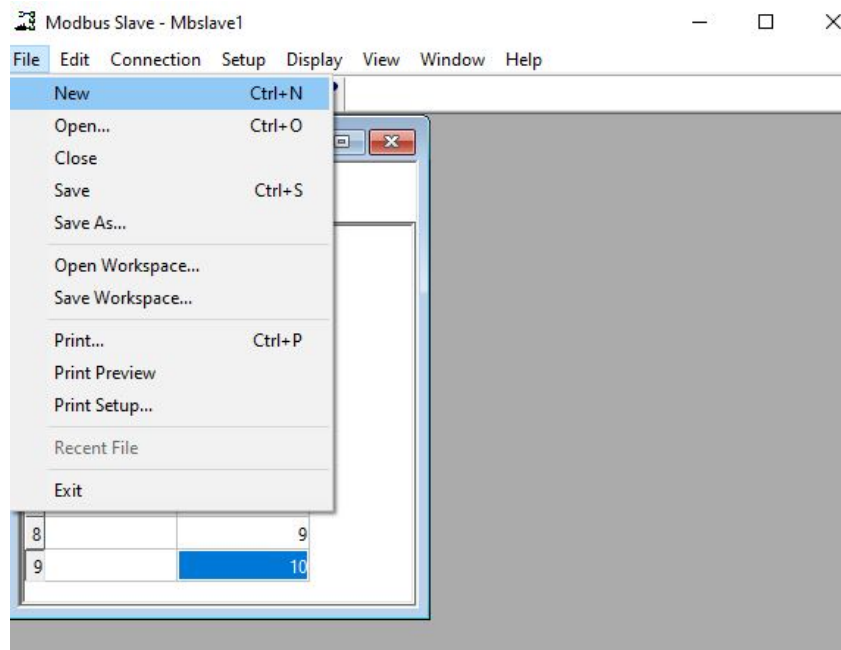
Serial byte counter      : 0
Response time (msec)    : 2000
Turnaround delay(msec) : 0
Hold time(msec)         : 0
Rx-to-Tx delay(msec)   : 0
Local Server Port       : 502
Max clients allowed     : 2
KeepAlive interval (sec): 60
Max pending messages    : 16
Current pending messages: 0
Send exceptions         : YES
RXed from slave msg     : 0
TXed to slave msg       : 0
RXed from master msg    : 0
TXed to master msg     : 0
Exceptions created      : 0

```

4.13. Connecting Second Slave To First Serial Interface

1. Connect Slave Id 12 to the first interface (0/9). Go to **New** menu.
FOR EXAMPLE: Perform the following.

Figure 12: New Menu

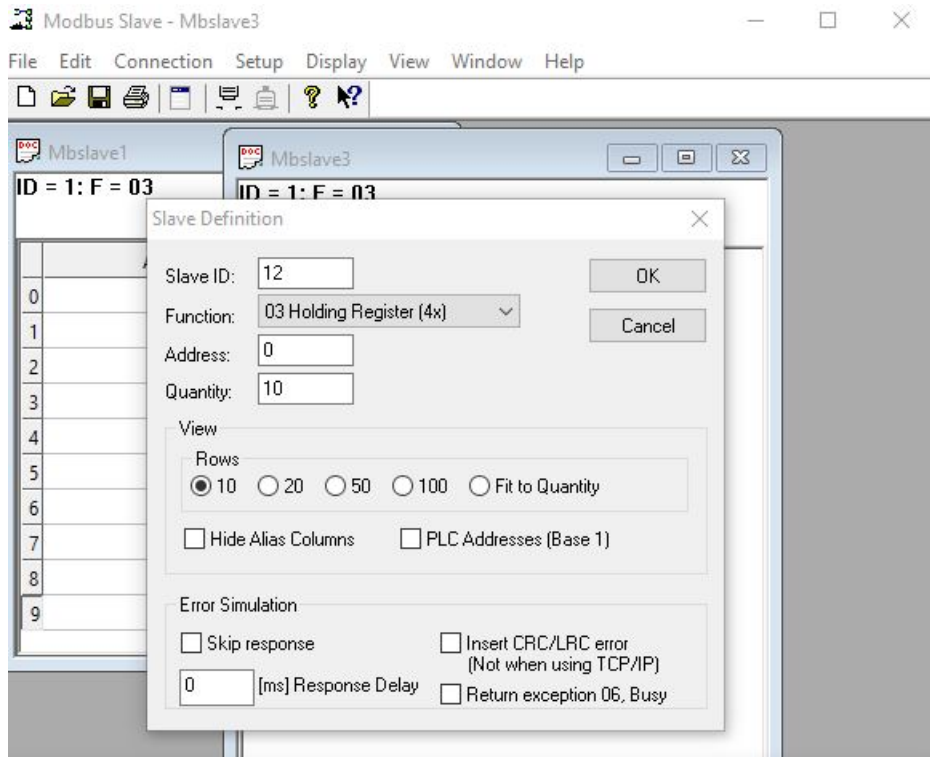


2. Click **new Slave** and click **Setup->Slave Definition** or press F8.

3. Set Slave ID to 12.

FOR EXAMPLE: Perform the following

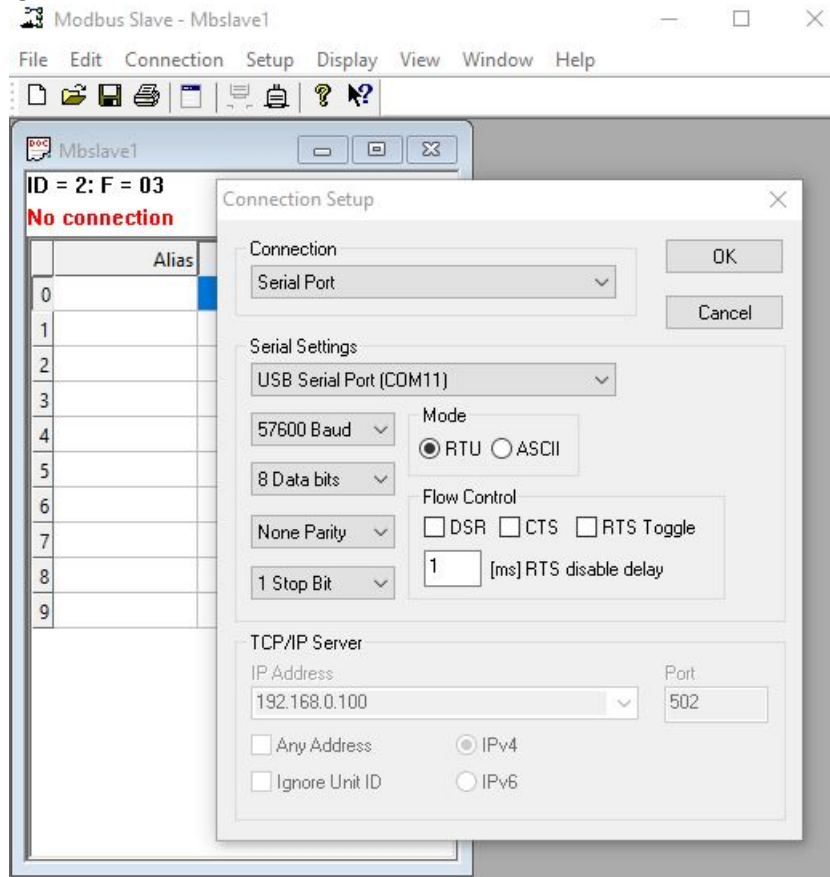
Figure 13: Slave Definition

4. Click **OK**.

4.14. Connecting Third Slave To Second Serial Interface

1. Open new Modbus Slave Application.
2. Click **Setup->Slave Definition** or press F8.
3. Set Slave ID to 2. Click **OK**.
4. Click **Connection->Connect** or press F3.
5. Set parameters for serial interface: connection type: Serial Port; new serial interface number; and new baud rate: 57600.

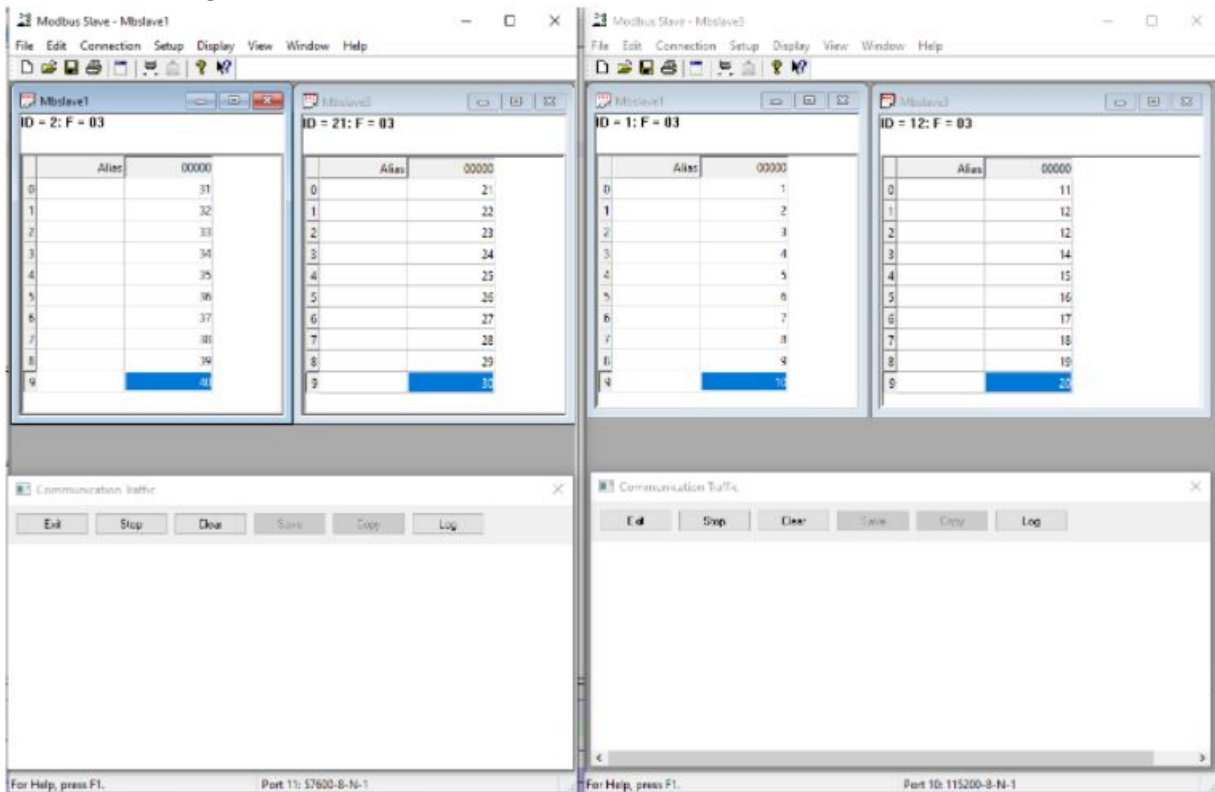
FOR EXAMPLE: See below.

Figure 14: Connecting Third Slave To Second Serial Interface

6. Click **OK**. A new serial connection is established.

4.15. Connecting Forth Slave To Second Serial Interface

1. Open new Modbus Slave Application.
2. Click **File->New** or press Ctrl+N.
3. Click **new Slave->Tab Setup->Slave Definition** or press F8.
4. Set Slave ID to 21. Click **OK**.
5. Click **Display->Communication**.
6. Set any values to slaves 2 and 21.
FOR EXAMPLE: See below.

Figure 15: Connecting Forth Slave To Second Serial Interface

4.16. Sending Request from First Master to Slave 12

1. In the **Read/Write definition (F8)**, select Slave ID 1.
2. Click **Read/Write once**.
3. Verify the Holding registers in the Master updated with the Slave's ID 1 information.
4. Verify that Communication Traffic windows show the transaction in the Master and in the Slave application.
5. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that counters are incremented.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
```

```
Response time (msec) : 2000
Turnaround delay(msec): 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 21 22 23 24 25
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec): 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 2
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 60441
Client 1 byte counter : 12
Client 2 IP : 192.168.0.100
Client 2 port : 60448
Client 2 byte counter : 0
KeepAlive interval (sec): 60
Max pending messages : 16
Current pending messages: 0
Send exceptions : YES
RXed from slave msg : 1
TXed to slave msg : 1
RXed from master msg : 1
TXed to master msg : 1
Exceptions created : 0
```

4.17. Sending Request from the First Master to the Slave 12

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Slave ID 12.

4.18. Sending Request from the First Master to the Slave 2

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Slave ID 2.

4.19. Sending Request from the First Master to the Slave 21

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Slave ID 21.

4.20. Sending Request from the Second Master to the Slave 1

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Second Master Application and Slave ID 1.

4.21. Sending Request from the Second Master to the Slave 12

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Second Master Application and Slave ID 12.

4.22. Sending Request from the Second Master to the Slave 2

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Second Master Application and Slave ID 2.

4.23. Sending Request from the Second Master to the Slave 21

1. Repeat steps from “Sending Request from the First Master to the Slave 1”, but use Second Master Application and Slave ID 21.

4.24. Deleting Forth Slave from the Second Serial Interface

1. Delete Slave ID 21 from the profile.
FOR EXAMPLE: Type the following:
 - Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
-   Configure profile PROF_10_MOD_CLIENT.
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# remove slave-id 21
interface serial 0/10
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that slave was removed.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 2
Clients number : 0
KeepAlive interval (sec) : 60
Max pending messages : 16
Current pending messages : 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 0
```



```
RXed from master msg      : 0
TXed to master msg        : 0
Exceptions created         : 0
```

3. Reconnect the Masters and send request from one of the Masters to Slave ID 21.
4. Verify that Master received exception "Gateway path unavailable".
5. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that exception was created.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role           : TCP Server
Serial interface : 9
Slaves ID      : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID      : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 2
Clients number : 1
Client 1 IP : 192.168.0.100
Client 1 port : 62798
Client 1 byte counter : 12
KeepAlive interval (sec) : 60
Max pending messages : 16
Current pending messages : 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 0
RXed from master msg : 1
```

```
TXed to master msg      : 1
Exceptions created      : 1
```

4.25. Changing Response Time for Second Serial Interface

1. Change the response time for the interface 0/10.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# response-timeout 50
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact((modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that slave was removed.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Serial interface : 10
```

```
Slaves ID : 2 22 23 24 25
```

```
Serial byte counter : 0
```

```
Response time (msec) : 50
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec) : 0
```

```
Rx-to-Tx delay(msec) : 0
```

```
Local Server Port      : 502
Max clients allowed   : 2
Clients number        : 0
KeepAlive interval (sec): 60
Max pending messages  : 16
Current pending messages: 0
Send exceptions       : YES
RXed from slave msg   : 0
TXed to slave msg     : 0
RXed from master msg  : 0
TXed to master msg    : 0
Exceptions created    : 0
```

4.26. Sending Two Requests from both Masters to Same Slave with no Delay Between

1. Read Holding registers from the same slave with no delay between readings.
2. Verify that the reading is done, Masters' holding registers were updated and messages are seen in the communication screens of the masters and the slave.

4.27. Configuring RX-to-TX Delay

1. Set rx-to-tx delay to 1000 for interface 0/10.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

2. Connect with the Masters and Slaves.

3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that rx-to-tx delay is 1000 msec for interface 0/10.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 50
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 1000
Local Server Port : 502
Max clients allowed : 2
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 56436
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 56437
Client 2 byte counter : 0
KeepAlive interval (sec) : 60
```

```

Max pending messages      : 16
Current pending messages: 0
Send exceptions           : YES
RXed from slave msg      : 0
TXed to slave msg        : 0
RXed from master msg     : 0
TXed to master msg       : 0
Exceptions created        : 0

```

4. Read holding registers twice with no delay between reads.
5. Verify that the first response in the Master's **Communication Traffic** screen appears immediately, and the second response only after 1 second approximately.

4.28. Changing Maximum Pending Messages to 0

1. Change the number of maximum pending messages to 0.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# max pending messages 0
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the number of the maximum pending messages was changed.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```
Role : TCP Server
```

```
Serial interface : 9
```

```
Slaves ID : 1 12 13 14 15 16 17 18 19 20
```

```
Serial byte counter : 0
```

```
Response time (msec) : 2000
```

```
Turnaround delay(msec) : 0
```

```
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0
Serial interface     : 10
Slaves ID            : 2 22 23 24 25
Serial byte counter  : 0
Response time (msec) : 50
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 1000
Local Server Port    : 502
Max clients allowed  : 2
Clients number       : 2
Client 1 IP          : 192.168.0.100
Client 1 port        : 56436
Client 1 byte counter : 0
Client 2 IP          : 192.168.0.100
Client 2 port        : 56437
Client 2 byte counter : 0
KeepAlive interval (sec) : 60
Max pending messages : 0
Current pending messages : 0
Send exceptions      : YES
RXed from slave msg  : 0
TXed to slave msg    : 0
RXed from master msg : 0
TXed to master msg   : 0
Exceptions created   : 0
```

4.29. Sending Two Requests from One Master to Same Slave with no Delay Between

1. Read holding registers twice with no delay between reads.
2. Verify that the first response in the Master's **Communication Traffic** screen appears immediately, and the second request responds with exception "Slave device busy".
3. Change the maximum pending messages to 10.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# max pending messages 10
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the number of the maximum pending messages was changed to 10.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 50
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 1000
Local Server Port : 502
Max clients allowed : 2
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 56436
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 56437
Client 2 byte counter : 0
KeepAlive interval (sec) : 60
Max pending messages : 10
```

```
Current pending messages: 0
Send exceptions           : YES
RXed from slave msg      : 0
TXed to slave msg        : 0
RXed from master msg     : 0
TXed to master msg       : 0
Exceptions created       : 0
```

5. Set rx-to-tx delay to 0 for interface 0/10.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

4.30. Sending Broadcast Write Request from First Master

1. Write to the Slave ID 0 by changing function number to 16 in the **Read/Write Definition** screen and clicking **Read/Write Once**.
2. Verify that all slaves were written with data.
3. Verify that “Timeout Exception” was reported on master.

4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the messages that were sent and received.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 50
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 2
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 57886
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 57889
Client 2 byte counter : 33
KeepAlive interval (sec) : 240
Max pending messages : 10
Current pending messages : 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 2
RXed from master msg : 1
TXed to master msg : 0
Exceptions created : 0
```

4.31. Sending Broadcast Read Request from the First Master

1. Read from the Slave ID 0 by changing function number to 3 in the **Read/Write Definition** screen and clicking **Read/Write Once**.
2. Verify that “Illegal function” was reported at the master.
3. Verify that “Timeout Exception” was reported at the master.
4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the messages that were sent and received.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
Profile : PROF_MOD_SERVER
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 50
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 57886
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 57889
Client 2 byte counter : 45
KeepAlive interval (sec) : 240
Max pending messages : 10
Current pending messages : 0
```

```

Send exceptions           : YES
RXed from slave msg      : 0
TXed to slave msg       : 2
RXed from master msg    : 2
TXed to master msg      : 1
Exceptions created       : 1

```

4.32. Configuring Turnaround Delay

1. Change turnaround delay to be 1000 msec for interface 0/10.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# turnaround delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

2. Reconnect with Masters and Slaves.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that turnaround delay is 1000 msec for interface 0/10.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec): 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 50
Turnaround delay(msec): 1000
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 1000
Local Server Port : 502
Max clients allowed : 64
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 56123
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 56124
Client 2 byte counter : 0
KeepAlive interval (sec): 60
Max pending messages : 10
Current pending messages: 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 0
RXed from master msg : 0
TXed to master msg : 0
Exceptions created : 0
```

4. Read holding registers twice with no delay between reads.
5. Verify that response arrives to the master immediately by looking in the Master's **Communication Traffic** screen. In Master and Slave, the response appears right after the request.

6. Read the holding registers twice. The second reading should be done immediately after the first time.
7. Verify that the second response in the Slave's **Communication Traffic** screen appears immediately after request, but in the Master's **Communication Traffic** screen, the response appears approximately after 1 second.

4.33. Configuring Hold Time

1. Unmap the profile. Set the hold time to 100 msec. Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# hold-time 100
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_MOD_SERVER
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER)# role server
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# connection-map
```

```
Phoenix_Contact(modbus-PROF_MOD_SERVER-server)# end
```

2. Connect Masters and Slaves.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the hold time value is 100.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface all
```

```
Profile : PROF_MOD_SERVER
```

```
Protocol : Modbus
```

```

Role : TCP Server
Serial interface : 9
Slaves ID : 1 12 13 14 15 16 17 18 19 20
Serial byte counter : 0
Response time (msec) : 2000
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Serial interface : 10
Slaves ID : 2 22 23 24 25
Serial byte counter : 0
Response time (msec) : 50
Turnaround delay(msec) : 1000
Hold time(msec) : 100
Rx-to-Tx delay(msec) : 0
Local Server Port : 502
Max clients allowed : 64
Clients number : 2
Client 1 IP : 192.168.0.100
Client 1 port : 56655
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 56656
Client 2 byte counter : 0
KeepAlive interval (sec) : 60
Max pending messages : 10
Current pending messages : 0
Send exceptions : YES
RXed from slave msg : 0
TXed to slave msg : 0
RXed from master msg : 0
TXed to master msg : 0
Exceptions created : 0

```

4. Read holding registers from the slave.
5. Verify that response arrives to the master immediately by looking in the **Communication Traffic** screen. In Master and Slave, the response appears right after the request.
6. Read holding registers twice. The second reading should be done immediately after the first time.
7. Verify that the second response in the Slave's **Communication Traffic** screen appears immediately after request, but in the Master's **Communication Traffic** screen, the response appears approximately after 100 milliseconds. It can be visually distinguished from 1 sec.

4.34. Clearing Serial Configuration and Deleting the Profile

1. Unmap the profile. Set the rx-to-tx delay to 100 msec. Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# no serial profile PROF_MOD_SERVER
```

```
Phoenix_Contact(config)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

5. MODBUS TCP Client Mode Configuration

This section lists the *CLI* configuration steps for *TCP* Client Configuration for MODBUS connection.

CONTEXT:

NOTE:

Server MODBUS Application disconnects sometimes. Ensure that messages are sent to the server when it is connected.

1. Configure *VLAN* 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure *VLAN* 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact (config-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-serial-if)# baud-rate 115200
```

```
Phoenix_Contact (config-serial-if)# no shutdown
```

```
Phoenix_Contact (config-serial-if)# end
```

5.1. Reconnecting

1. Map the client without preparing the server for connection.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile `PROF_10_MOD_CLIENT`.

```
Phoenix_Contact(config)# serial connection-type modbus profile  
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```



```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# local server port
modbus
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# remote ipv4 address
192.168.0.100 port modbus
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID :
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : NO
Reconnect Timeout (sec) : 15
KeepAlive interval (sec) : 240
Forward exceptions : 1
Response time (msec) : 1000
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 0
Turnaround delay (msec) : 0
Hold time (msec) : 0
Rx-to-Tx delay : 0
```

3. Verify default parameters.
4. Use WireShark to verify the connection attempts done within the configured reconnect timer interval. Actual attempts in software are done every "Reconnect timer (sec)" seconds, and can be seen when "debug serial critical" is configured; Wireshark shows only those attempts which are

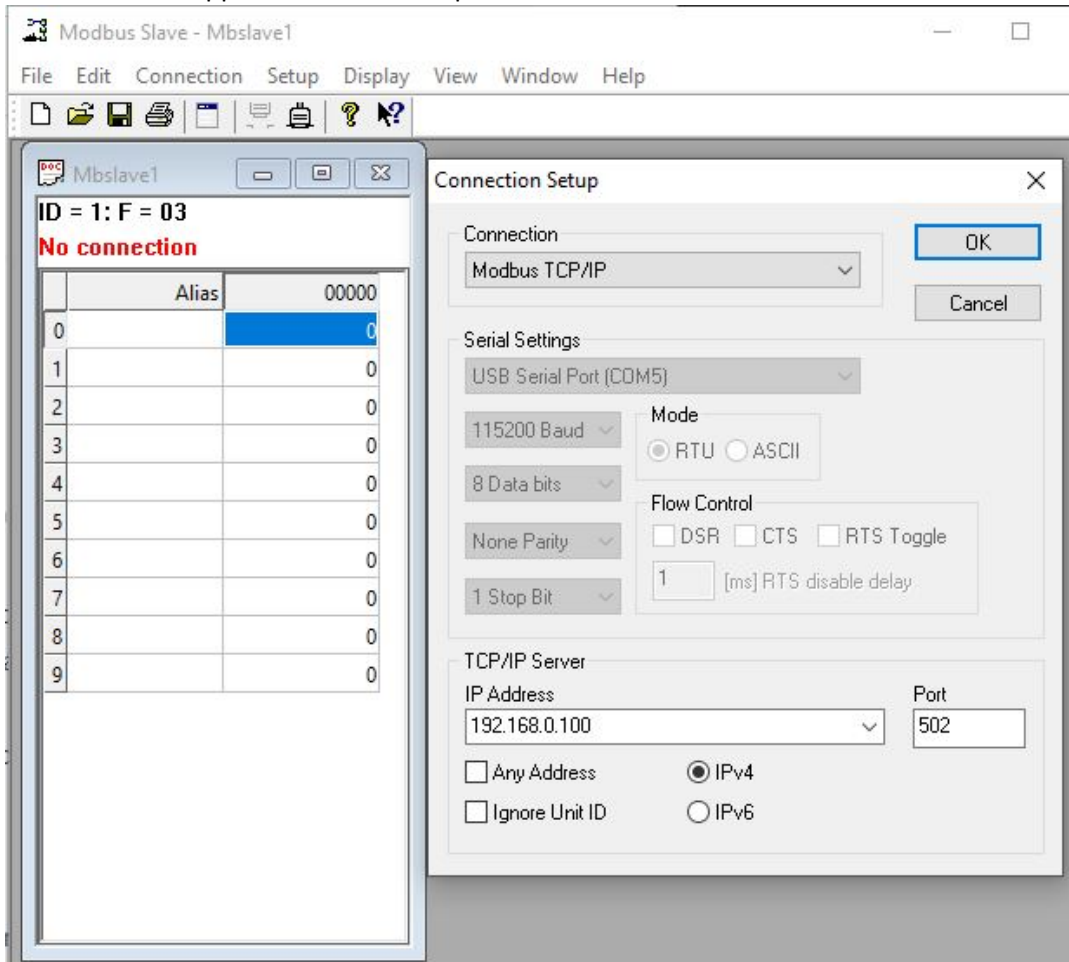
executed only when TCP stack is ready to execute them, so if the reconnect timeout is lengthy, the Wireshark attempts are executed with the “reconnect timer” attempts.

5.2. Connecting to Server

1. Start Server (Modbus Slave App) to accept connection from the client. The default Slave ID will be 1.

FOR EXAMPLE: Perform the following:

Figure 1: Modbus Slave App - Connection Setup



2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role           : TCP Client
Serial interface : 10
```

```
Slaves ID           :  
Remote Server IP    : 192.168.0.100  
Remote Server Port  : 502  
Local Client Port   : 502  
Serial byte counter : 0  
Server byte counter : 0  
Connection to server : YES  
Reconnect Timeout (sec) : 15  
KeepAlive interval (sec): 240  
Forward exceptions  : 1  
Response time (msec) : 1000  
RXed from master msg : 0  
TXed to master messages : 0  
RXed from slave msg : 0  
TXed to slave msg : 0  
Exceptions created  : 0  
DSCP                : 0  
Turnaround delay(msec) : 0  
Hold time(msec)     : 0  
Rx-to-Tx delay      : 0
```

5.3. Disconnecting from Server

1. Close the server.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10  
Profile : PROF_10_MOD_CLIENT  
Protocol : Modbus  
Role           : TCP Client  
Serial interface : 10  
Slaves ID      :  
Remote Server IP : 192.168.0.100  
Remote Server Port : 502  
Local Client Port : 502  
Serial byte counter : 0  
Server byte counter : 0  
Connection to server : NO
```

```
Reconnect Timeout (sec) : 15
KeepAlive interval (sec): 240
Forward exceptions      : 1
Response time (msec)   : 1000
RXed from master msg   : 0
TXed to master messages : 0
RXed from slave msg    : 0
TXed to slave msg      : 0
Exceptions created     : 0
DSCP                   : 0
Turnaround delay (msec) : 0
Hold time (msec)       : 0
Rx-to-Tx delay         : 0
```

3. Verify that Packetizing is disabled - turnaround, hold, and rx-to-tx timers are 0, serial counters are 0.
4. Use WireShark to verify connection attempts done within the configured reconnect timer interval.

5.4. Connecting to not Native Modbus Port

1. Disconnect the server.
2. In the **Connection** tab, change the listen server port number 15010.
3. Connect the server.
4. Change back remote and local port number by.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# remote ipv4 address
192.168.0.100 port 15010
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# local server port
15010
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

5. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP : 192.168.0.100
Remote Server Port : 15010
Local Client Port : 15010
Serial byte counter : 0
Server byte counter : 0
Connection to server : YES
Reconnect Timeout (sec) : 15
KeepAlive interval (sec): 60
Forward exceptions : 1
Response time (msec) : 1000
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 0
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay : 0
```

6. Change back remote and local port number.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# remote ipv4 address
192.168.0.100 port change back remote and local port number by
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# local server port
change back remote and local port number by
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

7. Disconnect the server.
8. In the **Connection** tab, change the listen server port number 502.
9. Connect the server.
10. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : YES
Reconnect Timeout (sec) : 15
KeepAlive interval (sec): 60
Forward exceptions : 1
Response time (msec) : 50
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 29
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay : 0
```

5.5. Configuring keep-alive

1. Use Wireshark to verify that keep-alive messages are sent according to keep-alive interval.
2. Unmap the profile, change the keepalive interval to different one. Map the profile. Connect with previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# keep-alive timeout 60
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

3. Use Wireshark to verify that keep-alive messages are sent according to keep-alive interval.
4. Run “show serial interface serial 0/10” and verify the keep-alive value.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

```
Protocol : Modbus
```

```
Role : TCP Client
```

```
Serial interface : 10
```

```
Slaves ID : 1 2 4 5 6 7 8 9
```

```
Remote Server IP : 192.168.0.100
```

```
Remote Server Port : 502
```

```
Local Client Port : 502
```

```
Serial byte counter : 0
```

```
Server byte counter : 0
```

```
Connection to server : YES
```

```
Reconnect Timeout (sec) : 4
```

```
KeepAlive interval (sec): 60
```

```
Forward exceptions : 1
```

```
Response time (msec) : 2000
```

```
RXed from master msg : 0
```

```
TXed to master messages : 0
```

```
RXed from slave msg : 0
```

```
TXed to slave msg : 0
```

```

Exceptions created      : 0
DSCP                   : 0
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay         : 0

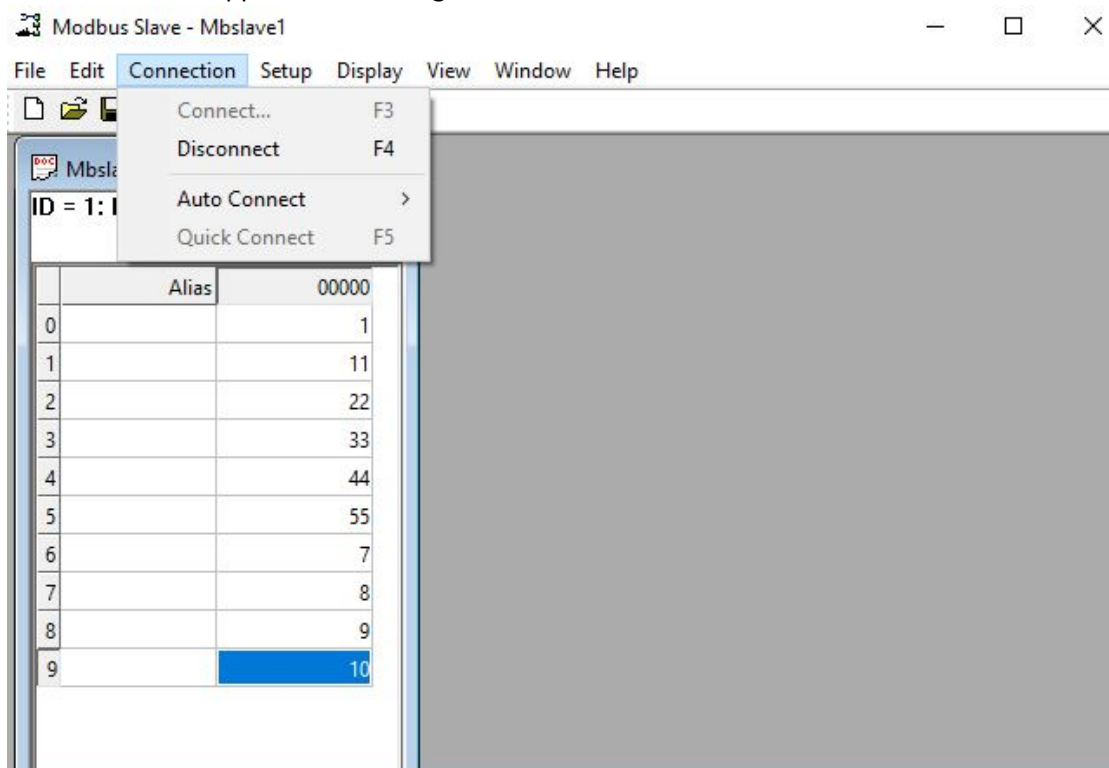
```

5.6. Reconnecting to the Server

1. Close connection in the slave (the server) - click **Disconnect**.

FOR EXAMPLE: Type the following:

Figure 2: Modbus Slave App - Disconnecting



2. Run "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (not connected).

FOR EXAMPLE: Type the following:

```

Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role           : TCP Client
Serial interface : 10
Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP : 192.168.0.100

```

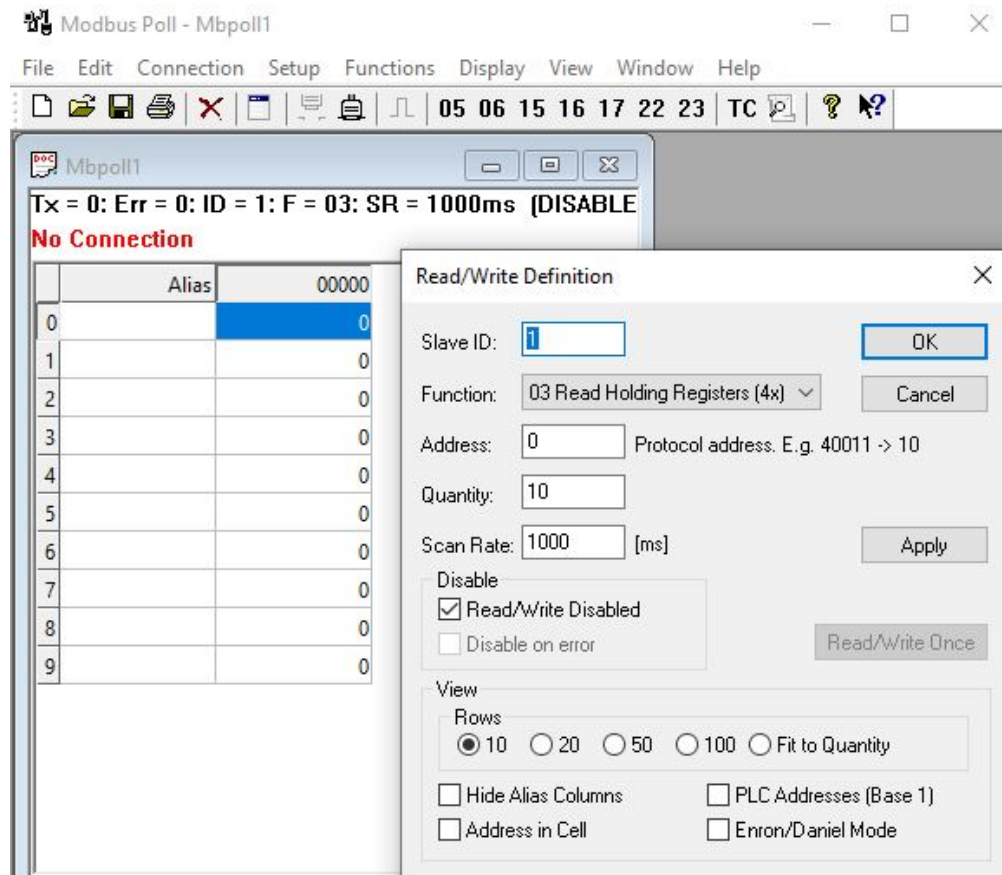


```
Remote Server Port      : 15010
Local Client Port      : 15010
Serial byte counter    : 0
Server byte counter    : 0
Connection to server   : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec) : 60
Forward exceptions     : 1
Response time (msec)   : 2000
RXed from master msg   : 0
TXed to master messages : 0
RXed from slave msg    : 0
TXed to slave msg     : 0
Exceptions created     : 0
DSCP                   : 0
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay        : 0
```

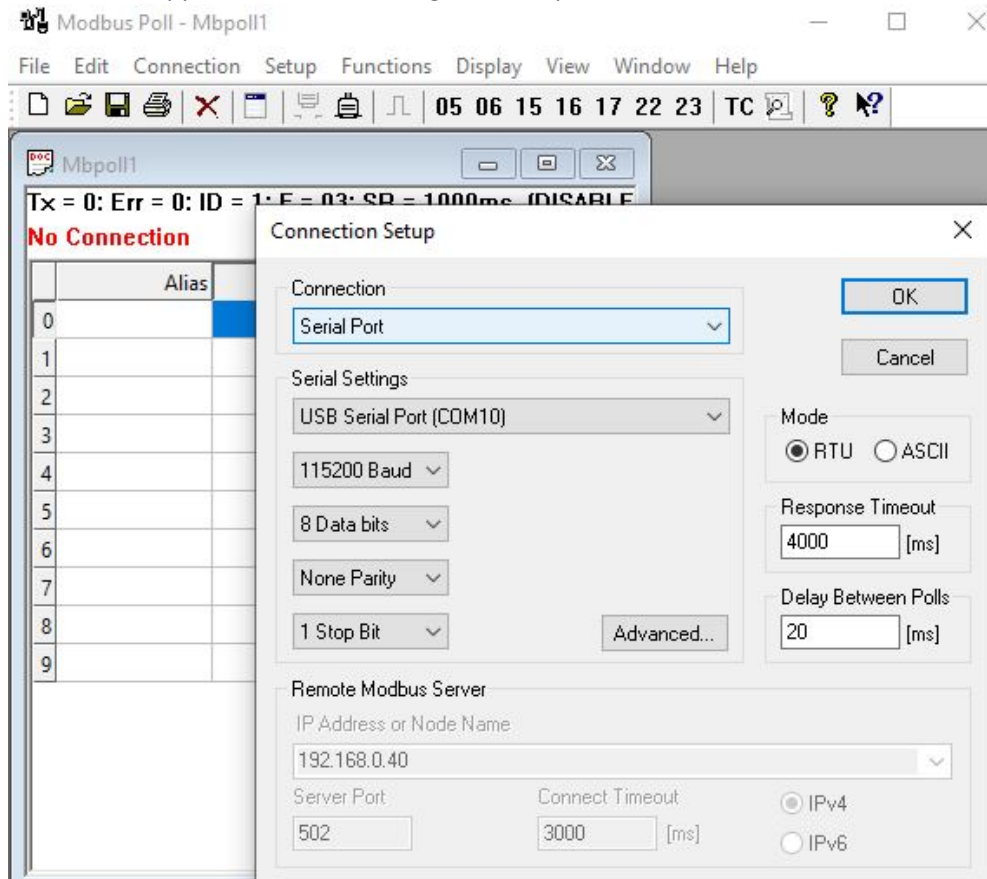
3. Connect the server in the same way as was disconnected.
4. Run "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (connected).

5.7. Connecting to the Master

1. Open Master Application.
2. Go to **Read/Write Definition** tab and select **Read/Write disable** and click **OK**.
FOR EXAMPLE: Type the following:

Figure 3: Modbus Slave App - Read / Write Definition

3. Go to **Connection** tab and connect to the client running on the device through a serial port.
FOR EXAMPLE: Type the following:

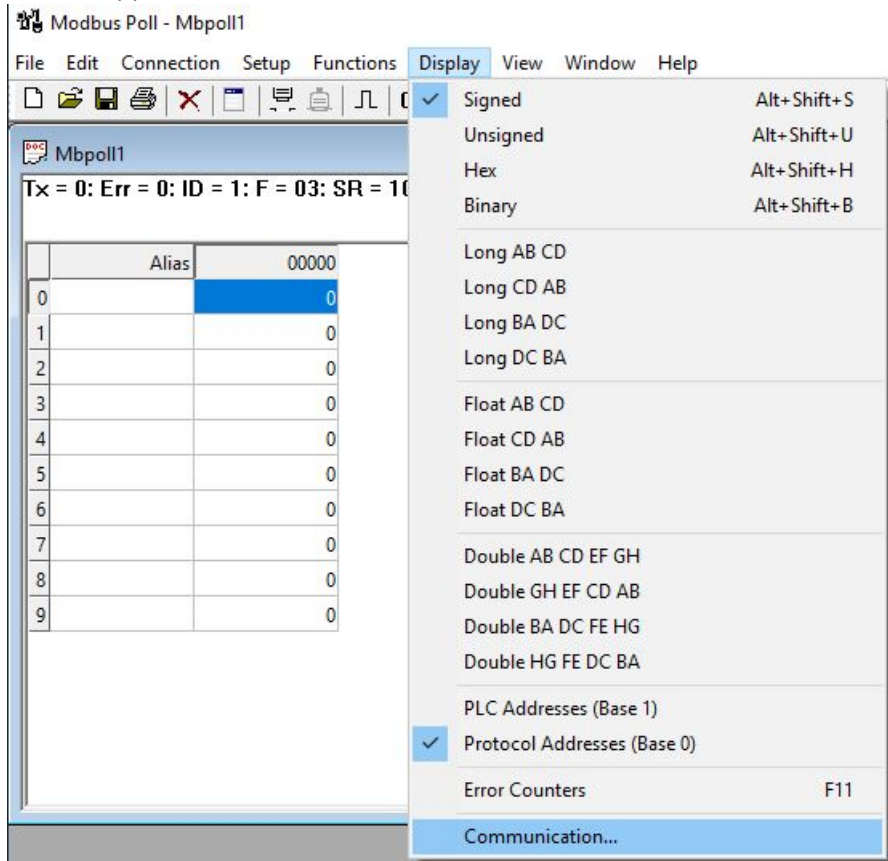
Figure 4: Modbus Slave App - Connection through a serial port

5.8. Not Configured Slave Polling (Exception by Client)

1. Go to **Display** and click **Communication**.

FOR EXAMPLE: Type the following:

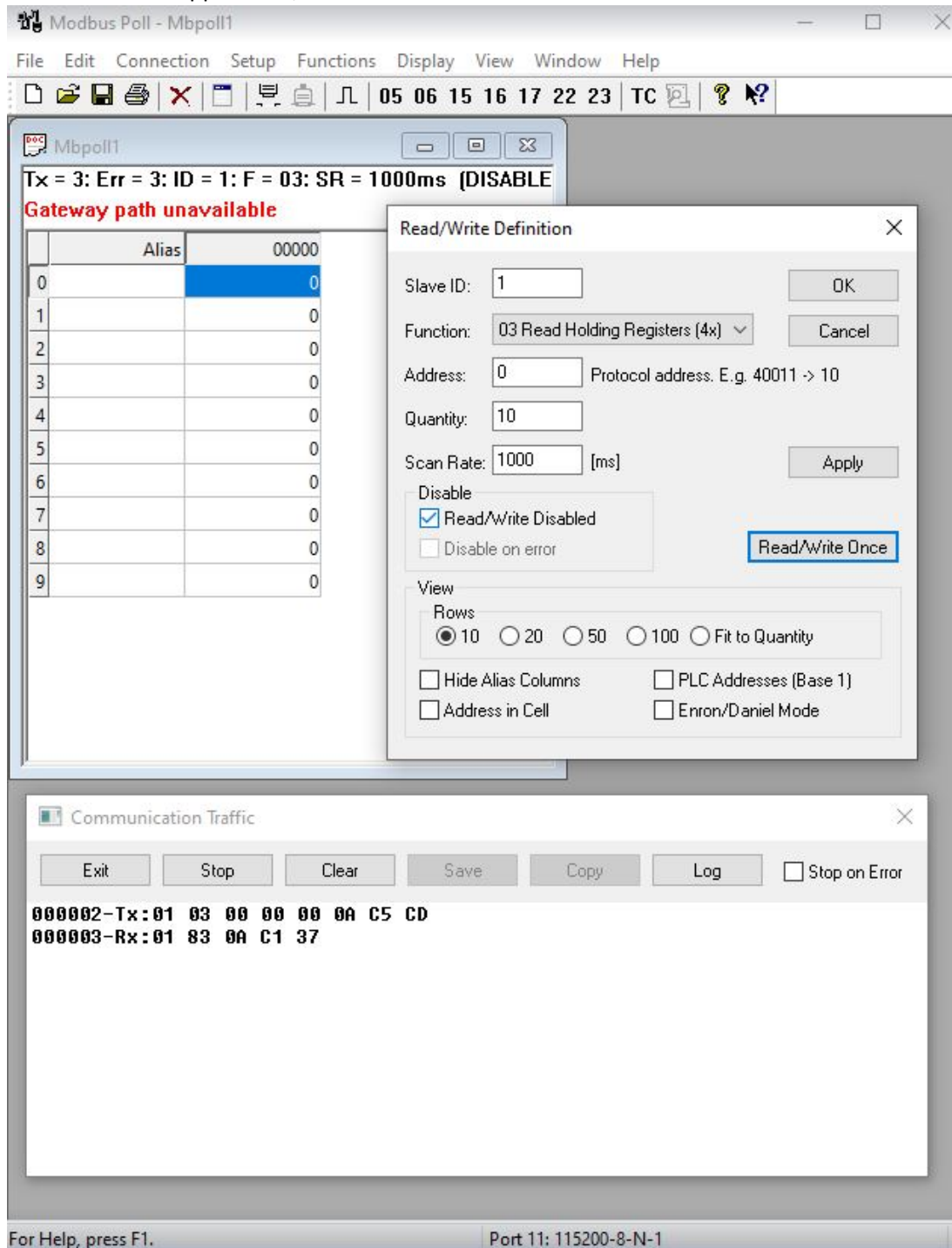
Figure 5: Modbus Slave App - Communication



2. In the Slave Application, in the same way as in Master, go to **Display** and click **Communication**.
3. Go to **Read/Write definition** and click **Read/Write Once** with the default parameters.

FOR EXAMPLE: Type the following:

Figure 6: Modbus Slave App - Read / Write Definition



- Verify that master receives exception "Gateway path unavailable" because the Slave 1 is not configured at the Client.
- Run "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that slaves were added.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID :
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : YES
Reconnect Timeout (sec) : 15
KeepAlive interval (sec): 240
Forward exceptions : 1
Response time (msec) : 1000
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 1
DSCP : 0
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay : 0
```

5.9. Adding Slaves to Client

1. Unmap the profile. Add multiple slaves (not more than 10 in a single command). Map the profile back.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# add slave-id
1,2,3,4,5,6,7,8,9,247
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that slaves were added.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 3 4 5 6 7 8 9 247
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : NO
Reconnect Timeout (sec) : 15
KeepAlive interval (sec): 240
Forward exceptions : 1
Response time (msec) : 1000
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 0
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay : 0
```

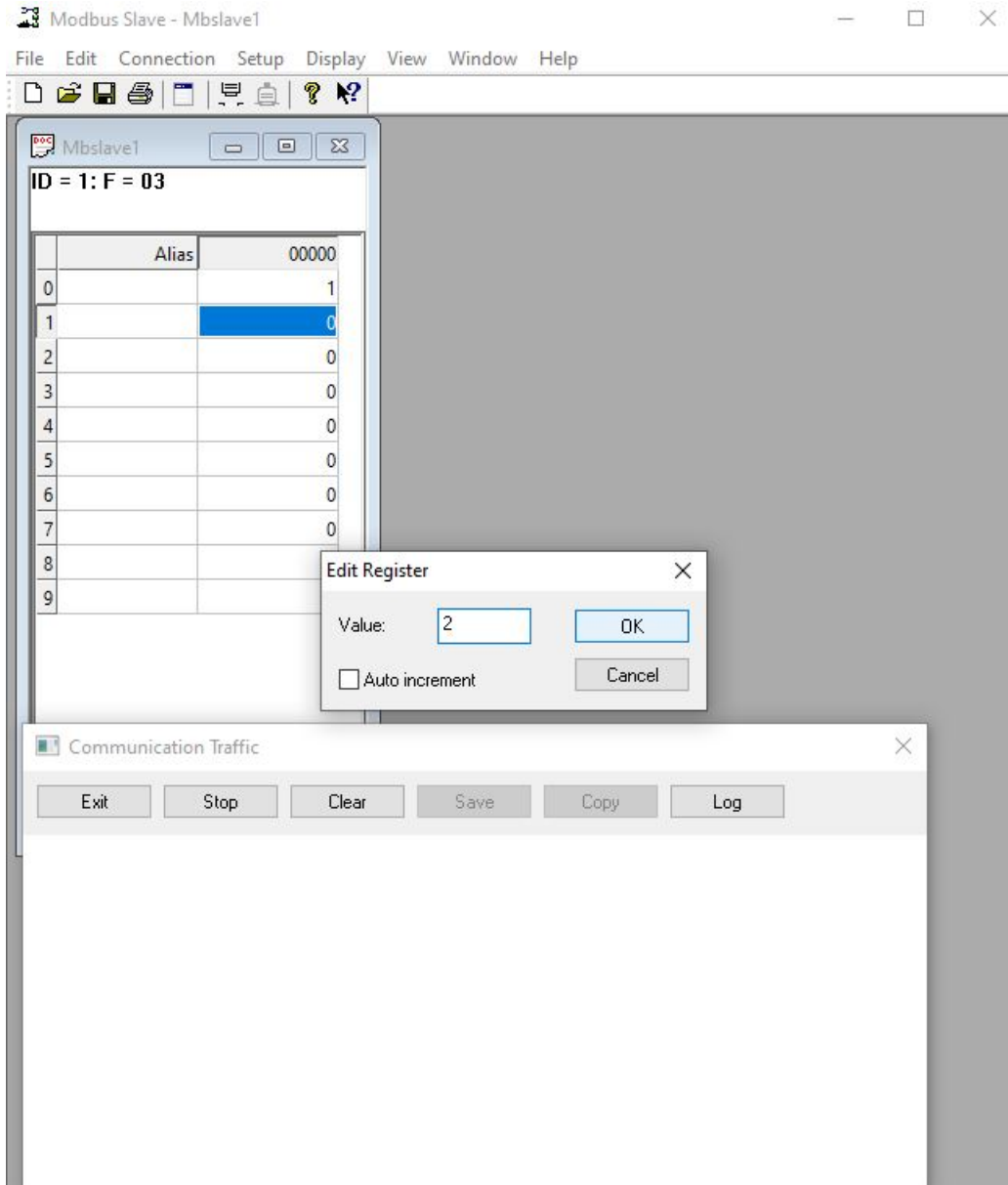
3. Verify default parameters.
4. Use WireShark to verify the connection attempts done within the configured reconnect timer interval. Actual attempts in software are done every "Reconnect timer (sec)" seconds, and can be seen when "debug serial critical" is configured; Wireshark shows only those attempts which are executed only when TCP stack is ready to execute them, so if the reconnect timeout is lengthy, the Wireshark attempts are executed with the "reconnect timer" attempts.

5.10. Reading Holding Registers from Slave

1. Put any values in 10 slave's holding registers to see them read and updated in the Master Application after the reading. Double-click the field and update the value.

FOR EXAMPLE: Perform the following.

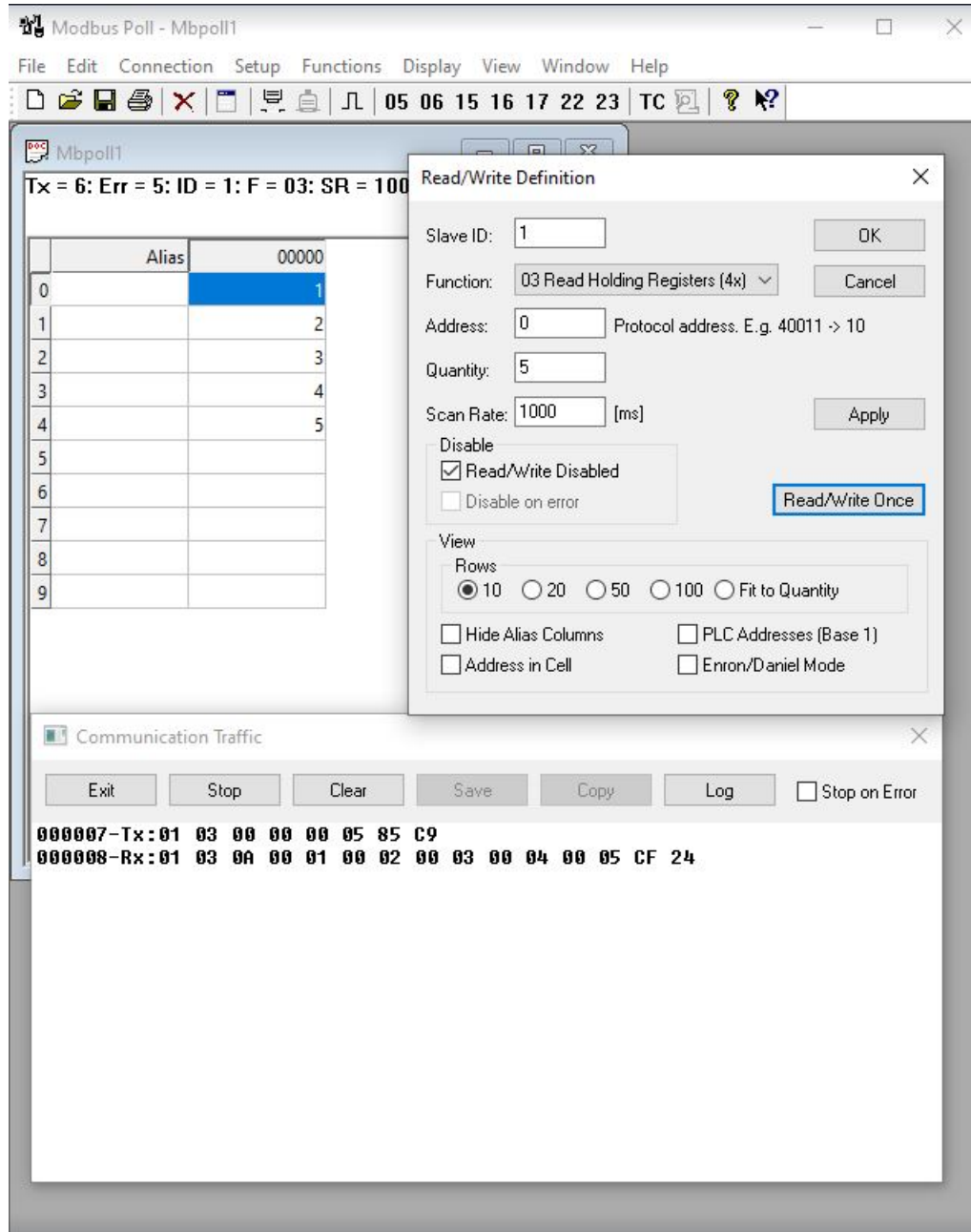
Figure 7: Modbus Slave - Edit Register



2. Send read command from the Master to Slave 1. Quantity of 5 read registers.

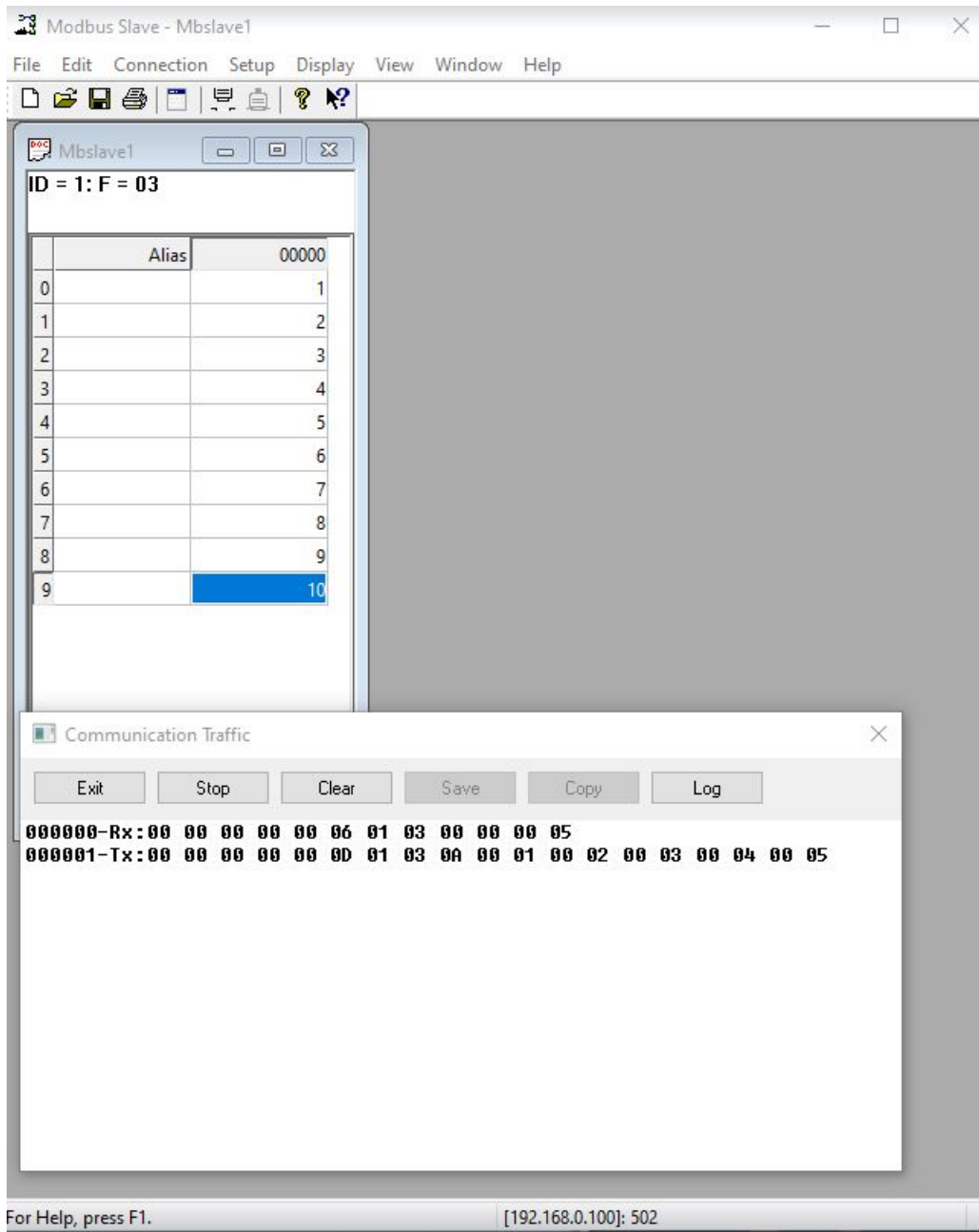
FOR EXAMPLE: Perform the following.

Figure 8: Modbus Slave - 5 Read



- Verify by looking into the **Communication Traffic** screen, that the request is sent, received in the Slave, and answered; if the answer is accepted in the Master's **Communication Traffic** screen, and the data has been updated in the Master Holding Registers to be equal to the five first registers of Slave's register.

FOR EXAMPLE: Perform the following.

Figure 9: Modbus Slave - Communication Traffic

- Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify counters were incremented with 1.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
```

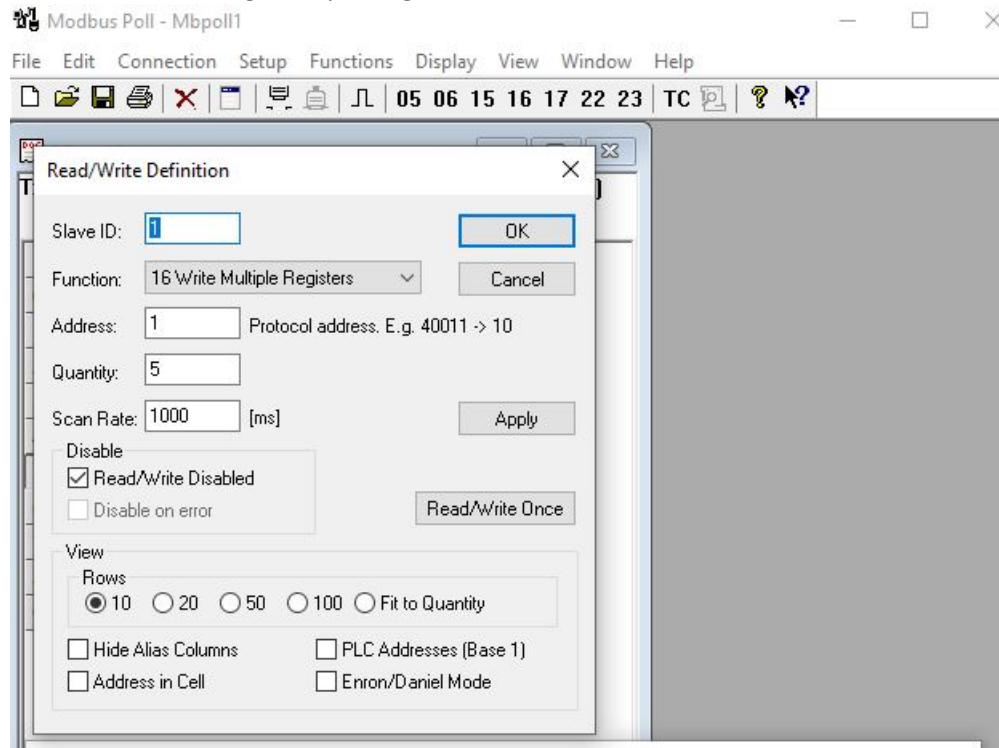
```
Slaves ID : 1 2 3 4 5 6 7 8 9 247
Remote Server IP      : 192.168.0.100
Remote Server Port   : 502
Local Client Port    : 502
Serial byte counter   : 0
Server byte counter   : 0
Connection to server  : YES
Reconnect Timeout (sec) : 4
KeepAlive interval (sec): 240
Forward exceptions    : 1
Response time (msec) : 2000
RXed from master msg  : 1
TXed to master messages : 1
RXed from slave msg   : 1
TXed to slave msg    : 1
Exceptions created    : 0
DSCP                  : 0
Turnaround delay (msec) : 0
Hold time (msec)     : 0
Rx-to-Tx delay       : 0
```

5.11. Writing to Holding Registers from Slave

1. In the **Read/Write Definition** dialog box, go to **Function** field and select **16 Write Multiple Registers**. Then, in the **Address** field, type **1**, and in **Quantity**, type **5**. Click **OK**.

FOR EXAMPLE: Perform the following.

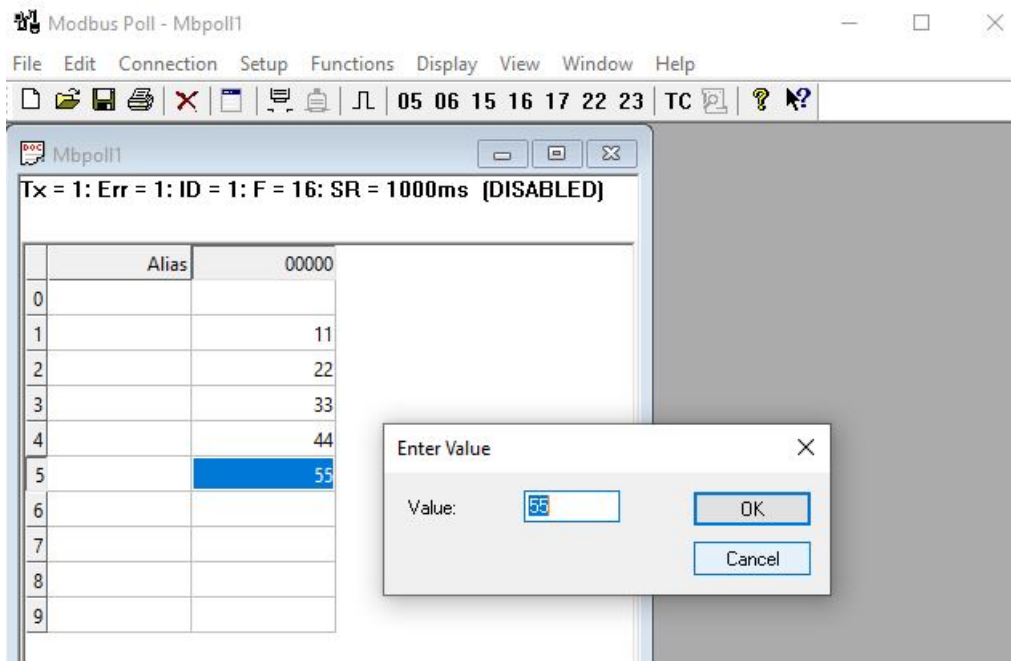
Figure 10: Modbus Slave - Writing Multiple Registers



2. Enter any values in the 10 master’s holding registers to see them written and updated in the Slave Application after the write. To update the value, double click the field and click **OK**.

FOR EXAMPLE: Perform the following.

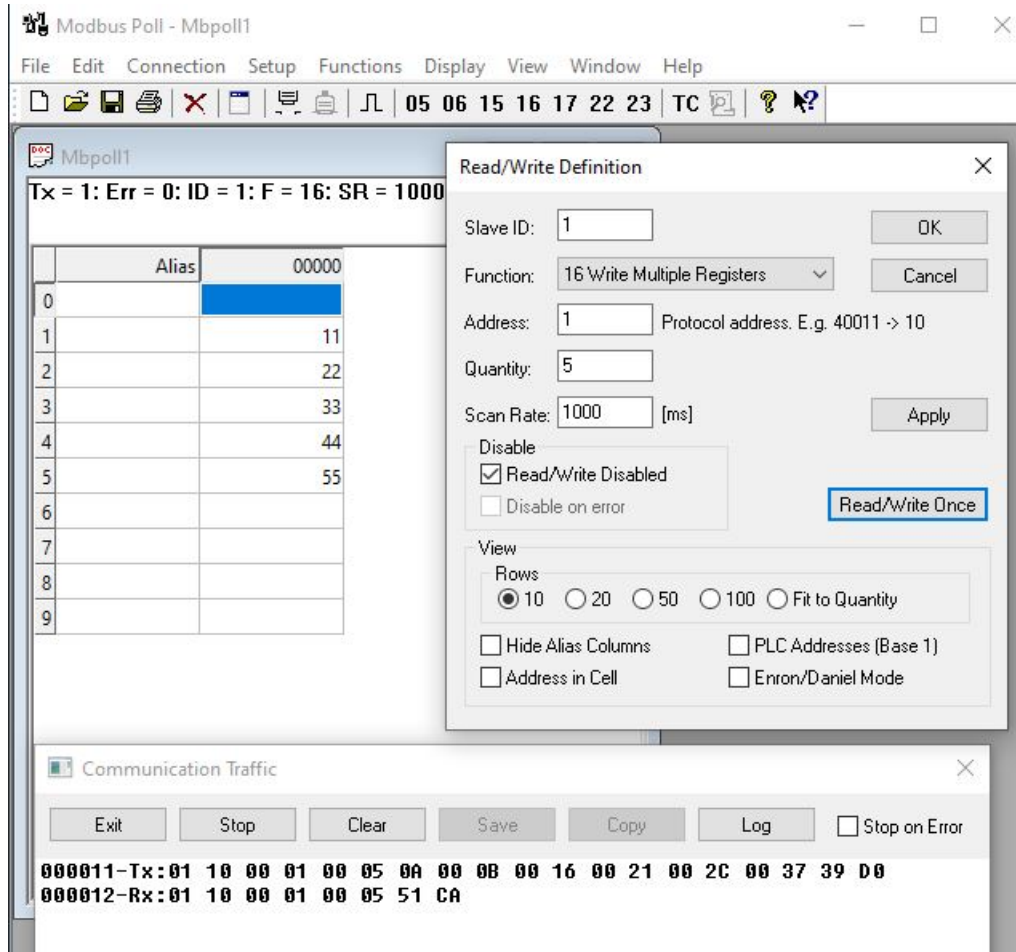
Figure 11: Modbus Slave - Enter Value



- Send write command from the Master to Slave 1. Write 5 registers from 1 to 6.

FOR EXAMPLE: Perform the following.

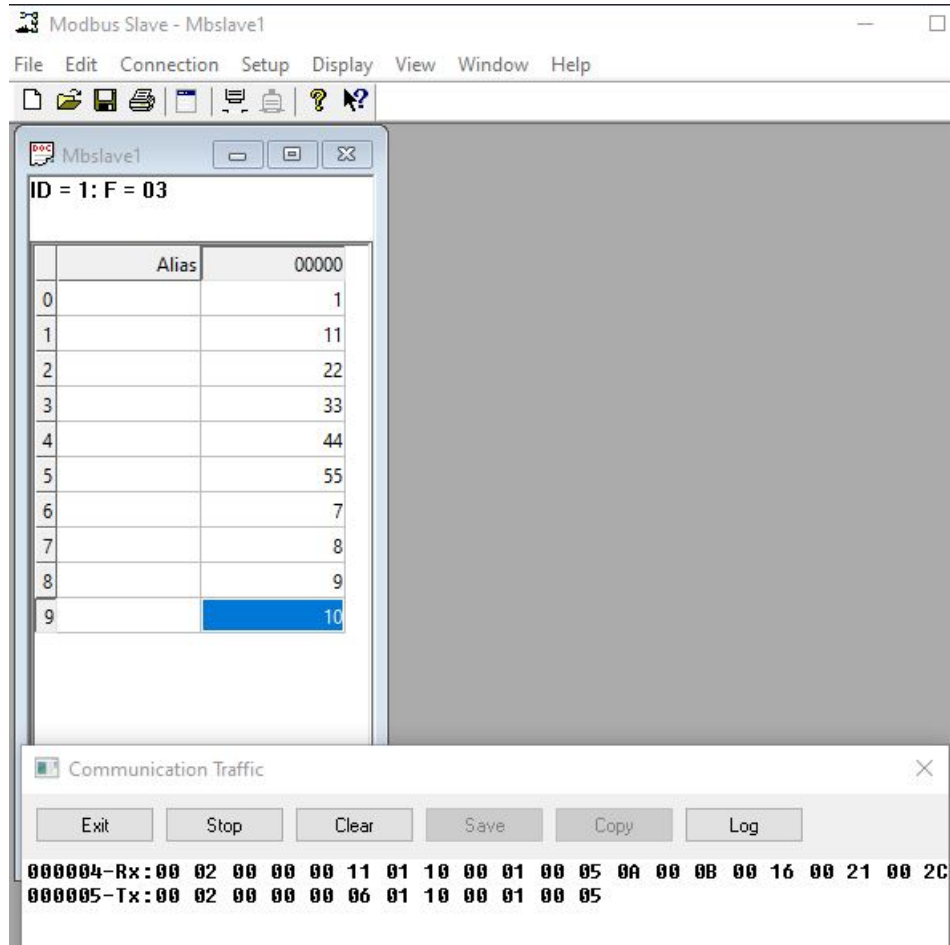
Figure 12: Modbus Slave - Enter Value



- Verify by looking into the **Communication Traffic** screen, that the request is sent, received in the Slave, and acknowledged; Slave's registers 1 to 6 are written with received data and the answer is accepted in the Master's **Communication Traffic** screen. No exception should be observed.

FOR EXAMPLE: Perform the following.

Figure 13: Modbus Slave - Enter Value



- Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify counters were incremented with 1 and now have value 2.

FOR EXAMPLE: Type the following:

```

Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 3 4 5 6 7 8 9 247
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : YES
Reconnect Timeout (sec) : 4

```

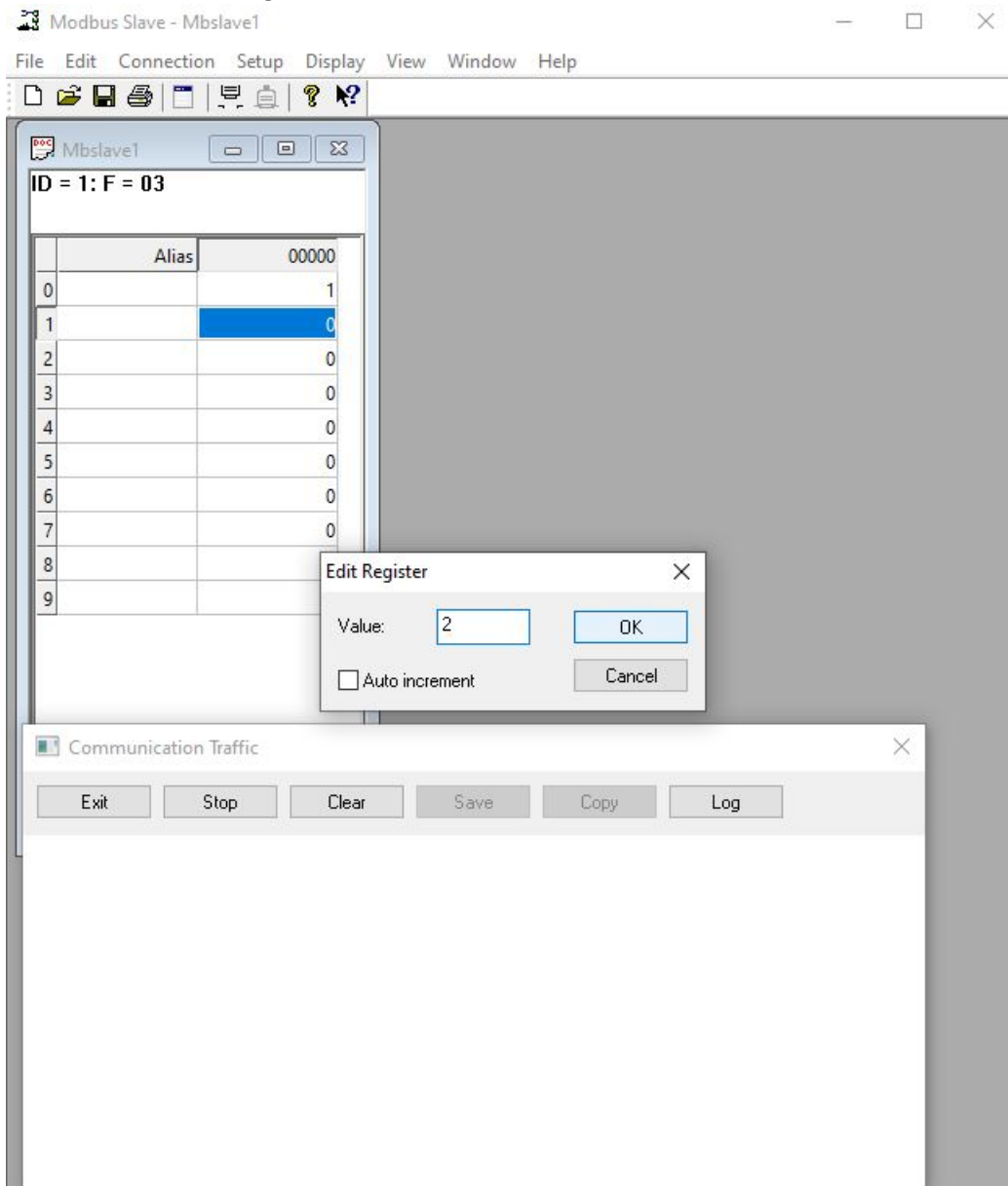
```

KeepAlive interval (sec): 240
Forward exceptions      : 1
Response time (msec)   : 2000
RXed from master msg   : 2
TXed to master messages : 2
RXed from slave msg    : 2
TXed to slave msg      : 2
Exceptions created     : 0
DSCP                   : 0
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay         : 0
    
```

5.12. Reading Not Supported Number of Holding Registers from Slave (Exception by Slave)

1. Put any values in 10 slave's holding registers to see them read and updated in the Master Application after the reading. Double-click the field and update the value. Or leave them as is, since they should not be sent to the master)

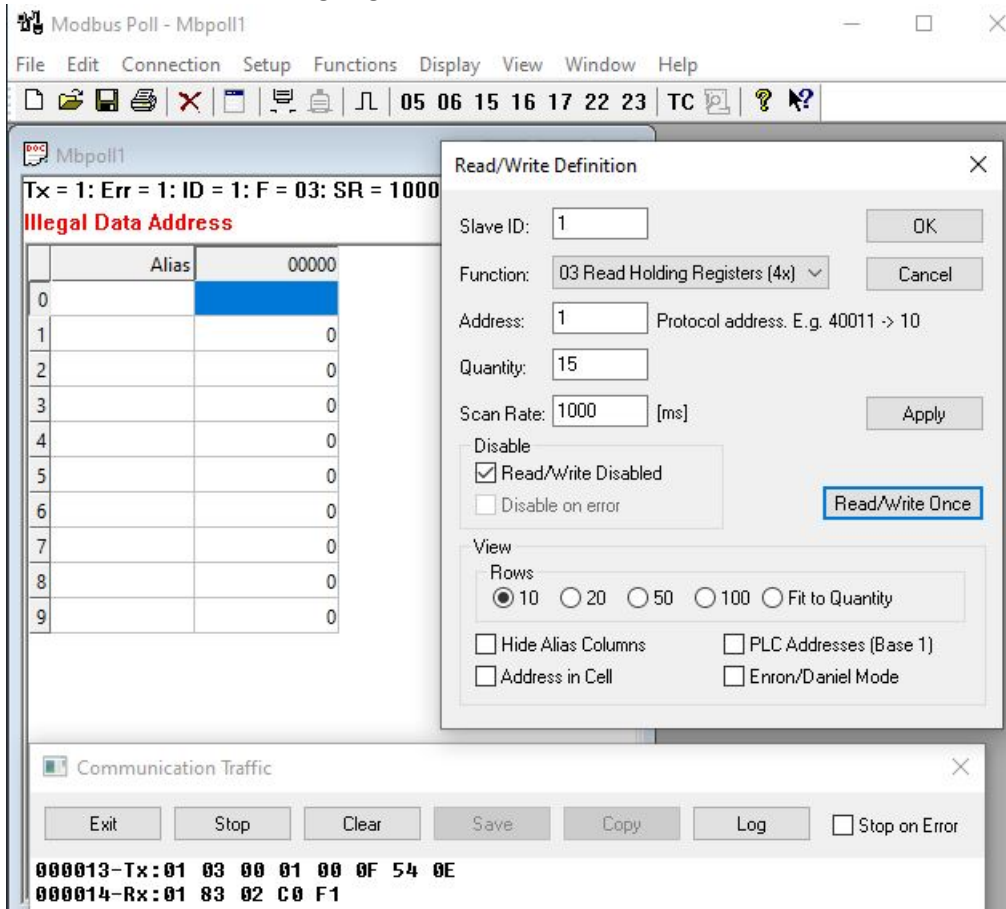
FOR EXAMPLE: Perform the following.

Figure 14: Modbus Slave - Edit Register

2. In the **Read/Write Definition** dialog box, go to **Function** field and select **03 Read Holding Registers**. Then, in **Address** field, type **1**, and in **Quantity**, type **15**. Click **OK**.

FOR EXAMPLE: Perform the following.

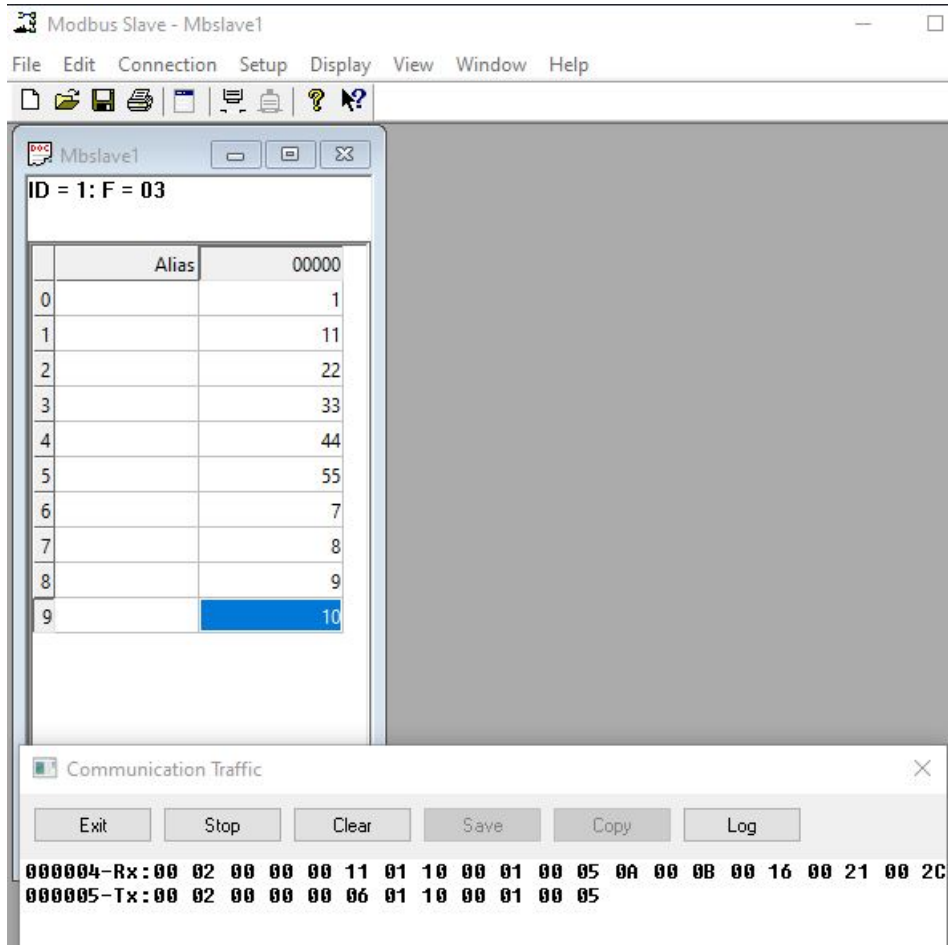
Figure 15: Modbus Slave - Read Holding Registers



- Verify by looking into the **Communication Traffic** screen, that the request is sent, received in the Slave, and answered with exception; the answer is accepted in the Master's **Communication Traffic** screen has been shown in red color in the Master Application.

FOR EXAMPLE: Perform the following.

Figure 16: Modbus Slave - Enter Value



- Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify counters were incremented with 1 (now at 3).

FOR EXAMPLE: Type the following:

```

Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 3 4 5 6 7 8 9 247
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : YES
Reconnect Timeout (sec) : 4

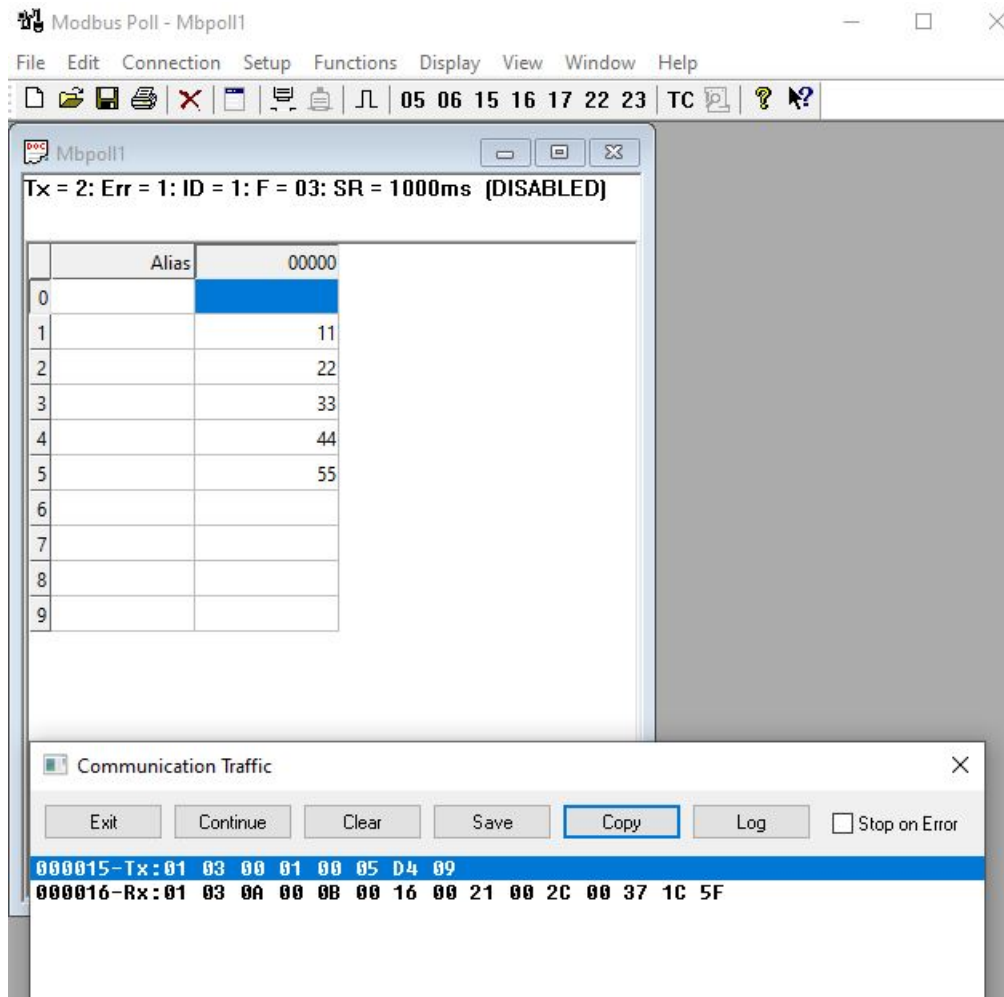
```

```
KeepAlive interval (sec): 240
Forward exceptions      : 1
Response time (msec)   : 2000
RXed from master msg   : 3
TXed to master messages : 3
RXed from slave msg    : 3
TXed to slave msg      : 3
Exceptions created     : 1
DSCP                   : 0
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay         : 0
```

5.13. Reading Holding Registers with Wrong CRC

1. Perform a normal reading and verify that it worked properly.
2. In the **Communication Traffic** screen of the Master, click **Copy**.

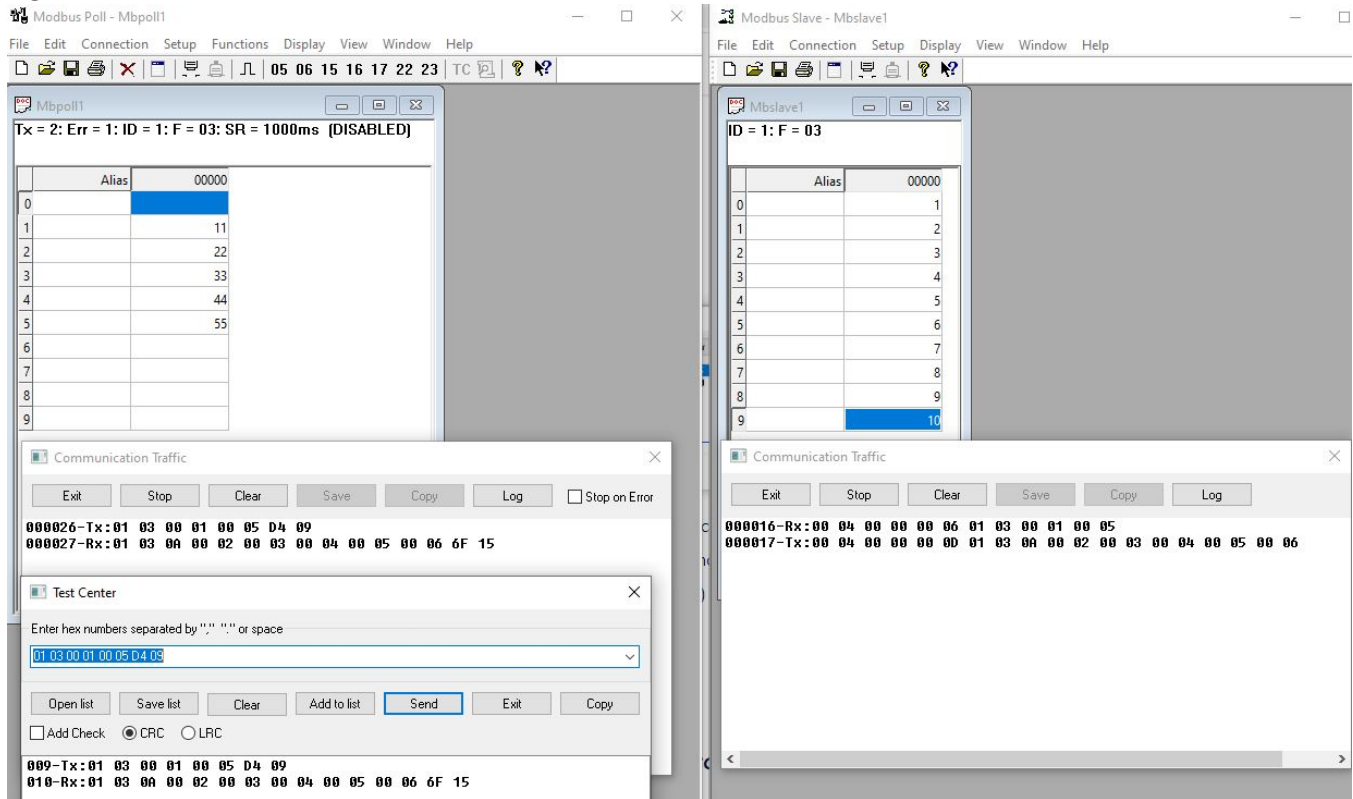
FOR EXAMPLE: Perform the following.

Figure 17: Modbus Slave - Communication Traffic- Master

3. After that, in the **Communication Traffic** screen of the Master, click **Continue**.
4. Open the **Test Center** screen from the **Functions** tab. Copy the message (only the message part) to the **Test Center** screen. Click **Send**.

FOR EXAMPLE: Perform the following.

Figure 18: Modbus Slave - Communication Traffic - Master and Slave



5. In the **Communication Traffic** screen of the Master, verify that message sent from master and received answer are the same as in “Reading Holding Registers from the Slave” chapter. Same about Slave **Communication Traffic** screen.
6. Change any number inside the message sent from **Test Center** screen.
7. Use the command “debug serial critical” to enable debugging.
8. Send the message again.
9. Verify detection of CRC mismatch in the debug messages.
 FOR EXAMPLE: The following appears:
 [SER_IP_DBG] : SerialIpModbusValidateSerialMsg(), CRC mismatch, received: 0xd408, calculated: 0xd409
 [SER_IP_DBG] : SerialIpModbusTcpClientSerialFdProc(): request serial message at serial 10 is invalid, ignoring
10. Use the command “debug serial none” to disable debugging.
11. Verify that there is no response received on the master side (no response should be sent upon CRC error).
12. Close the **Test Center** screen.

5.14. Reading Holding Registers from Disconnected Slave

1. Go to **Connection** and disconnect a Slave.
2. Execute read command. Use “Reading Holding Registers from the Slave” chapter as example.
3. Verify the master received “Gateway path unavailable” exception like in NOT CONFIGURED SLAVE POLLING chapter.
4. Connect back the slave.
5. Execute read command again.
6. Verify master read the slave’s registers properly.

5.15. Disabling forward-exception

1. Disable forward-exception.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# forward-exception
disable
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that forwarding is disabled.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

```
Protocol : Modbus
```

```
Role : TCP Client
```

```
Serial interface : 10
```

```
Slaves ID : 1 2 3 4 5 6 7 8 9 247
```

```
Remote Server IP : 192.168.0.100
```

```
Remote Server Port : 502
```

```
Local Client Port : 502
```

```

Serial byte counter      : 0
Server byte counter     : 0
Connection to server    : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec): 60
Forward exceptions      : NO
Response time (msec)    : 2000
RXed from master msg    : 0
TXed to master messages : 0
RXed from slave msg     : 0
TXed to slave msg      : 0
Exceptions created      : 0
DSCP                    : 0
Turnaround delay (msec) : 0
Hold time (msec)        : 0
Rx-to-Tx delay          : 0

```

3. Perform all steps from chapter “Reading Holding Registers from Disconnected Slave”.
4. Verify the client is not responding to the master when the slave is disconnected.
5. Verify the “Timeout error” in the master application.

5.16. Enabling forward-exception

1. Enable forward-exception.

FOR EXAMPLE: Perform the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# forward-exception
enable
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that forwarding is enabled.

5.17. Configuring DSCP

1. Change Differentiated Services Field Codepoints (DSCP) value.

FOR EXAMPLE: Perform the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# DSCP 28
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the DSCP value.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

```
Protocol : Modbus
```

```
Role : TCP Client
```

```
Serial interface : 10
```

```
Slaves ID : 1 2 3 4 5 6 7 8 9 247
```

```
Remote Server IP : 192.168.0.100
```

```
Remote Server Port : 502
```

```
Local Client Port : 502
```

```
Serial byte counter : 0
```

```
Server byte counter : 0
```

```
Connection to server : NO
```

```
Reconnect Timeout (sec) : 4
```

```
KeepAlive interval (sec): 60
```

```
Forward exceptions : NO
```

```
Response time (msec) : 2000
```

```
RXed from master msg : 0
```

```
TXed to master messages : 0
```

```
RXed from slave msg : 0
```

```
TXed to slave msg : 0
```

```
Exceptions created : 0
```

```
DSCP : 28
```

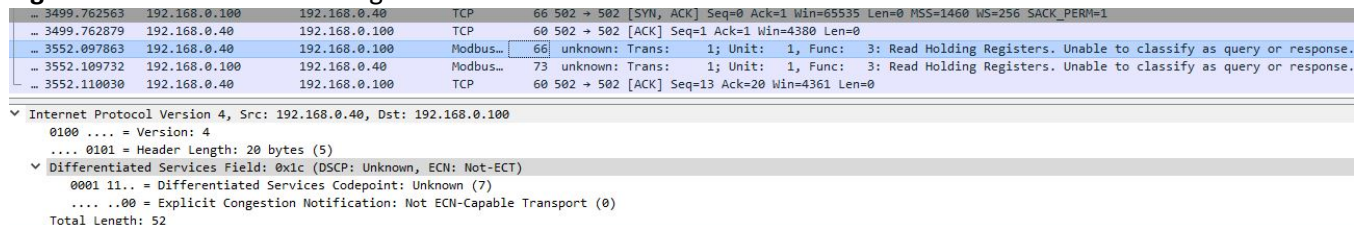


```
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay : 0
```

3. Open Wireshark.
4. Send any message to the Slave.
5. Use Wireshark to verify that the DSCP level is 28.

FOR EXAMPLE:

Figure 19: Wireshark Showing DSCP of 28



No.	Time	Source	Destination	Protocol	Length	Info
...	3499.762563	192.168.0.100	192.168.0.40	TCP	66	502 → 502 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
...	3499.762879	192.168.0.40	192.168.0.100	TCP	60	502 → 502 [ACK] Seq=1 Ack=1 Win=4380 Len=0
...	3552.097863	192.168.0.40	192.168.0.100	Modbus...	66	unknown: Trans: 1; Unit: 1, Func: 3: Read Holding Registers. Unable to classify as query or response.
...	3552.109732	192.168.0.100	192.168.0.40	Modbus...	73	unknown: Trans: 1; Unit: 1, Func: 3: Read Holding Registers. Unable to classify as query or response.
...	3552.110030	192.168.0.40	192.168.0.100	TCP	60	502 → 502 [ACK] Seq=13 Ack=20 Win=4361 Len=0

▾ Internet Protocol Version 4, Src: 192.168.0.40, Dst: 192.168.0.100
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 ▾ Differentiated Services Field: 0x1c (DSCP: Unknown, ECN: Not-ECT)
 0001 11.. = Differentiated Services Codepoint: Unknown (7)
00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
 Total Length: 52

5.18. Removing Slaves

1. Remove the slaves.

FOR EXAMPLE: Perform the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# remove slave-id
1,3,247
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the that the specified slaves were removed.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

```
Protocol : Modbus
```

```
Role : TCP Client
```

```
Serial interface : 10
```

```

Slaves ID : 2 4 5 6 7 8 9
Remote Server IP      : 192.168.0.100
Remote Server Port   : 502
Local Client Port    : 502
Serial byte counter  : 0
Server byte counter  : 0
Connection to server : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec): 60
Forward exceptions   : NO
Response time (msec) : 2000
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg  : 0
TXed to slave msg   : 0
Exceptions created   : 0
DSCP                 : 28
Turnaround delay(msec) : 0
Hold time(msec)     : 0
Rx-to-Tx delay      : 0

```

3. Execute chapter “Not Configured Slave Polling (Exception by Client)” and verify receiving exception.
4. Add a slave ID of 1.

FOR EXAMPLE: Perform the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# add slave-id 1
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

5. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that a slave was added.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

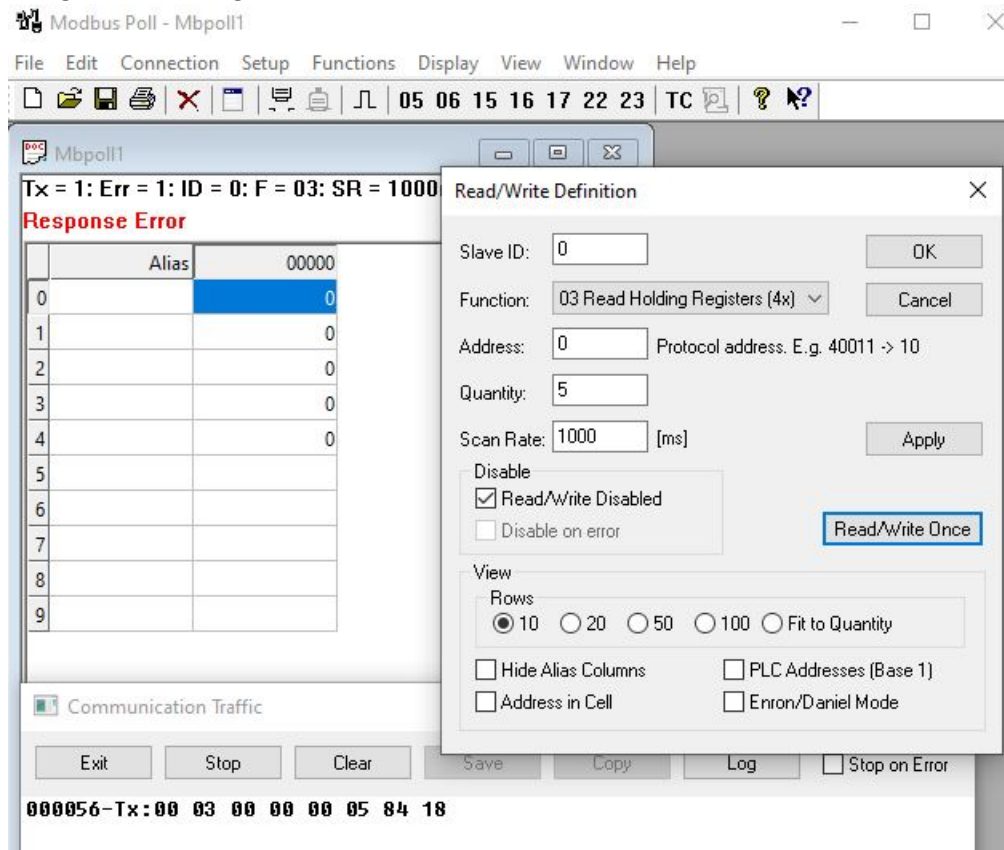
```
Protocol : Modbus
Role      : TCP Client
Serial interface : 10
Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec): 60
Forward exceptions : NO
Response time (msec) : 2000
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 28
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay : 0
```

5.19. Writing to Slave Using Slave ID 0

1. Configure the master for writing and set wanted values in its registers.
2. Select Slave ID 0 in the **Read/Write Definition** screen.
3. Write to the client.
4. Verify that there is no response to the request, but the data was written.

FOR EXAMPLE: Perform the following.

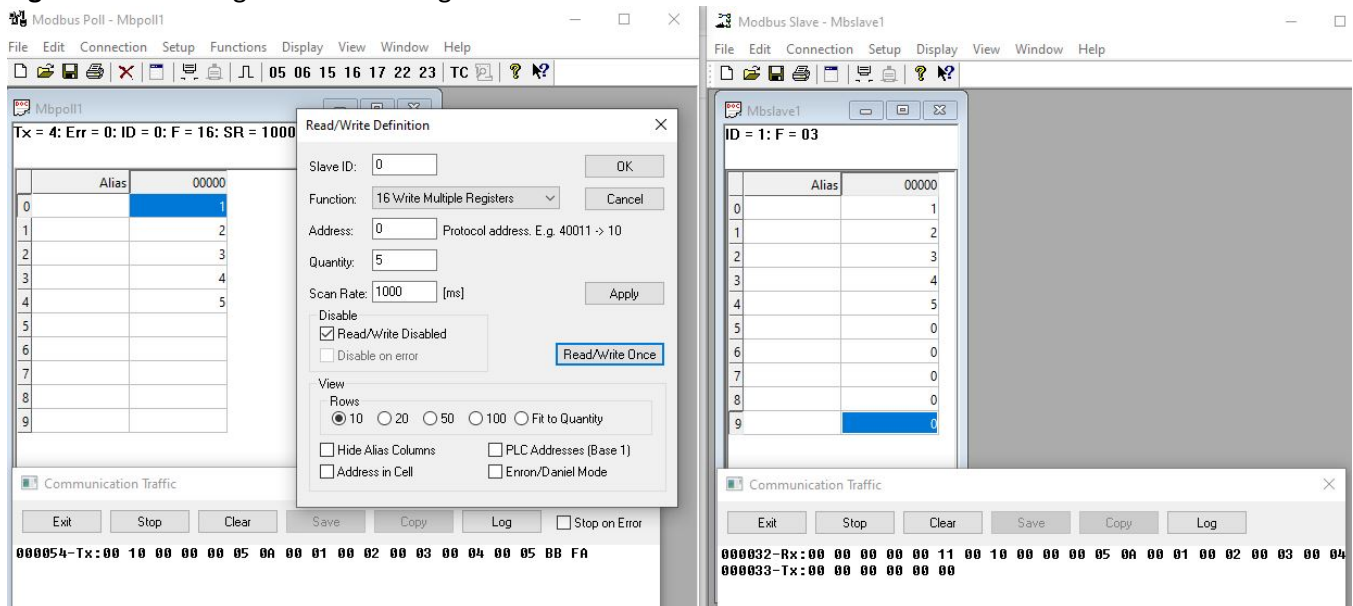
Figure 20: Writing to Slave Using Slave ID 0



5.20. Reading from Slave Using Slave ID 0

1. Configure the master for reading and set wanted values in the registers of the master and slave.
2. Select Slave ID 0 in the **Read/Write Definition** screen.
3. Read from the client.
4. Verify that the master is receiving exception.

FOR EXAMPLE: Perform the following.

Figure 21: Reading from Slave Using Slave ID 0

5.21. Changing response-timeout

1. Change the response time.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# response-timeout 50
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that response-timeout was changed.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

```
Protocol : Modbus
```

```
Role : TCP Client
```

```
Serial interface : 10
```

```

Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP      : 192.168.0.100
Remote Server Port    : 502
Local Client Port     : 502
Serial byte counter   : 0
Server byte counter   : 0
Connection to server  : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec): 60
Forward exceptions    : NO
Response time (msec)  : 50
RXed from master msg  : 0
TXed to master messages : 0
RXed from slave msg   : 0
TXed to slave msg     : 0
Exceptions created    : 0
DSCP                  : 0
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay        : 0

```

3. In the **Read/Write Definition**. screen, select Slave ID of 1.
4. Read the Holding Registers from the Slave.
5. If the response from the slave arrives after 50 msec, the master should get the “device failed to respond” exception, even if the slave responded with a proper data.

5.22. Configuring Turnaround Delay

1. Unmap the profile. Set the turnaround delay to 1000 msec. Map the profile. Connect to the server (slave).

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

```
Phoenix_Contact # configure terminal
Phoenix_Contact(config)# interface serial 0/10
Phoenix_Contact(config-serial-if)# turnaround delay 1000
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact # configure terminal
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that turnaround delay is 1000 msec.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec) : 60
Forward exceptions : NO
Response time (msec) : 50
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 0
Turnaround delay(msec) : 1000
```

```
Hold time(msec)           : 0
Rx-to-Tx delay           : 0
```

3. Read the Holding Registers from the Slave.
4. Verify that response arrives to the master immediately by looking into the **Communication Traffic** screen. In Master and Slave the response appear right after the request.
5. Read holding registers twice. The second reading should be done immediately after the first time.
6. Verify that the second response in the Slave's **Communication Traffic** screen appears immediately after request; however, in the Master's **Communication Traffic** screen, the response appears approximately after 1 second.

5.23. Configuring hold-time Delay

1. Unmap the profile. Set the hold-time delay to 100 msec. Map the profile. Connect to the server (slave).

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# hold-time 100
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```


2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that hold-time delay is 100 msec.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile : PROF_10_MOD_CLIENT
Protocol : Modbus
Role : TCP Client
Serial interface : 10
Slaves ID : 1 2 4 5 6 7 8 9
Remote Server IP : 192.168.0.100
Remote Server Port : 502
Local Client Port : 502
Serial byte counter : 0
Server byte counter : 0
Connection to server : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec): 60
Forward exceptions : NO
Response time (msec) : 50
RXed from master msg : 0
TXed to master messages : 0
RXed from slave msg : 0
TXed to slave msg : 0
Exceptions created : 0
DSCP : 0
Turnaround delay(msec) : 1000
Hold time(msec) : 100
Rx-to-Tx delay : 0
```

3. Read the Holding Registers from the Slave.
4. Verify that response arrives to the master immediately by looking into the **Communication Traffic** screen. In Master and Slave the response appear right after the request.
5. Read holding registers twice. The second read should be done immediately after the first time.
6. Verify that the second response in the Slave's **Communication Traffic** screen appears immediately after request; however, in the Master's **Communication Traffic** screen, the response appears approximately after 100 msec. Actually, it is $\text{time_of_Rxed_response} - \text{time_of_Rxed_request} + 100 \text{ msec}$). It can be visually distinguished from 1 sec.

5.24. Configuring rx-to-tx Delay

1. Unmap the profile. Set the turnaround delay and hold-time to 0, and the rx-to-tx delay to 1000 msec. Map the profile. Connect to the server (slave).

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# no connection-map
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# hold-time 100
```

```
Phoenix_Contact(config-serial-if)# turnaround delay 0
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type modbus profile
PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT)# role client
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# connection-map
interface serial 0/10
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that rx-to-tx delay is 1000 msec.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/10
```

```
Profile : PROF_10_MOD_CLIENT
```

```
Protocol : Modbus
```

```
Role : TCP Client
```

```
Serial interface : 10
```

```
Slaves ID : 1 2 4 5 6 7 8 9
```

```
Remote Server IP : 192.168.0.100
```

```

Remote Server Port      : 502
Local Client Port      : 502
Serial byte counter     : 0
Server byte counter    : 0
Connection to server   : NO
Reconnect Timeout (sec) : 4
KeepAlive interval (sec) : 60
Forward exceptions     : NO
Response time (msec)   : 50
RXed from master msg   : 0
TXed to master messages : 0
RXed from slave msg    : 0
TXed to slave msg     : 0
Exceptions created     : 0
DSCP                   : 0
Turnaround delay (msec) : 0
Hold time (msec)       : 0
Rx-to-Tx delay        : 1000

```

3. Read the Holding Registers from the Slave.
4. Verify that the response in the Slave's **Communication Traffic** screen appears approximately after 1 second.

5.25. Clearing Serial Configuration and Deleting Profile

1. Delete the profile. Set Rx-to-Tx delay to 0.

FOR EXAMPLE: Type the following:

– Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

– Configure profile PROF_10_MOD_CLIENT.

```
Phoenix_Contact(config)# no serial profile PROF_10_MOD_CLIENT
```

```
Phoenix_Contact(modbus-PROF_10_MOD_CLIENT-client)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/10
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

2. Run “show serial profile interface serial 0/10” to verify that profile is not mapped.

6. Raw Socket Server Mode Configuration

This section lists the *CLI* configuration steps for Server Configuration for Raw Socket connection.

1. Configure VLAN 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure VLAN 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact (config-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-serial-if)# baud-rate 115200
```

```
Phoenix_Contact (config-serial-if)# no shutdown
```

```
Phoenix_Contact (config-serial-if)# end
```

6.1. Configuring Packetizing OFF

1. Send data from one of the clients. Verify that serial interface app receives it.
2. Check the counters if they correspond to the number of bytes sent by the client. Check the counters using "show serial profile ..." command
3. Verify that packetizing is disabled.
4. Send some data from the serial interface app.
5. Verify if all data is received by all clients with no visible delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_IN
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 0
Max connections : 64
Packetizing : OFF
TCP buffering : enabled
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 58698
Client 1 RX byte cnt : 5
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
Client 2 IP : 192.168.0.100
Client 2 port : 58705
Client 2 RX byte cnt : 7
Client 2 TX byte cnt : 0
Client 2 Packets retry : 0
Client 2 Bytes retry : 0
```

6.2. Configuring Packet Timer

1. Unmap the profile, enable packetizing, and set packet timeout to 1000 msec (1 sec). Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_IN.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
```

```
Phoenix_Contact(raw-PROF_9_IN)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_IN)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_IN-TCP)# direction in
```

- Configure the packet timeout and establish connection.

```
Phoenix>Contact(raw-PROF_9_IN-TCP-in)# packetizing enable
Phoenix>Contact(raw-PROF_9_IN-TCP-in)# packet timeout 1000
Phoenix>Contact(raw-PROF_9_IN-TCP-in)# exit
Phoenix>Contact(raw-PROF_9_IN-TCP)# exit
Phoenix>Contact(raw-PROF_9_IN)# connection-map interface serial 0/9
Phoenix>Contact(raw-PROF_9_IN)# end
```

2. Send some data from the serial interface app.
3. Verify all data received by all clients with a delay of 1 sec (this can be easily compared to the no delay state of the previous example with packetizing disabled).
4. Verify the values of packet parameters.

FOR EXAMPLE: Type the following:

```
Phoenix>Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 0
Max connections : 64
Packetizing : ON
TCP buffering : disabled
Packet size : 1400
Packet timer(msec) : 1000
Packet char(hex) : disabled
Turnaround delay(msec) :
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 50282
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
```

6.3. Configuring Packet Character

1. Unmap the profile and set packet character to "ENTER" (13 dec). Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_IN.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
```

NOTE: the command shown below is only needed if there is an existing mapping of the profile to an interface.

```
Phoenix_Contact(raw-PROF_9_I)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_I)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_I-TCP)# direction in
```

- Configure the packet character and establish connection.

```
Phoenix_Contact(raw-PROF_9_I-TCP-in)# packet timeout 0
```

NOTE: To set the packet character, packetizing has to be enabled and packet timeout has to be set as 0.

```
Phoenix_Contact(raw-PROF_9_I-TCP-in)# packet char 13
```

```
Phoenix_Contact(raw-PROF_9_I-TCP-in)# exit
```

```
Phoenix_Contact(raw-PROF_9_I-TCP)# exit
```

```
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_I)# end
```

2. Send some data from the serial interface app using sometimes an "ENTER" character.
3. Verify that data arrives to all clients only upon using "ENTER" character, not after any timeout. (since the packet timeout is configured to 0).
4. Verify the values of packet parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_IN
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Mode : Server
```

```
Direction : In
```

```
Transport : TCP
```

```
Local Server Port : 15010
```

```
Number of clients : 2
```

```
Serial RX byte counter : 0
```

```

Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 0
Max connections : 64
Packetizing : ON
TCP buffering : disabled
Packet size : 1400
Packet timer (msec) : 0
Packet char(hex) : d
Turnaround delay(msec) :
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 50282
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0

```

6.4. Configuring Packet Size

1. Unmap the profile and set pack size to 16 bytes. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_IN.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
```

```
Phoenix_Contact(raw-PROF_9_I)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_I)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_I-TCP)# direction in
```

- Configure the packet size and establish connection.

```
Phoenix_Contact(raw-PROF_9_I-TCP-in)# packet size 16
```

```
Phoenix_Contact(raw-PROF_9_I-TCP-in)# exit
```

```
Phoenix_Contact(raw-PROF_9_I-TCP)# exit
```

```
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_I)# end
```

2. Send 17 bytes of data from the serial interface app without using "ENTER" character.
3. Verify that 16 bytes of data arrive to all clients.

4. Type "ENTER" character.
5. Verify that remaining data arrives to all clients.
6. Verify the values of packet parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 0
Max connections : 64
Packetizing : ON
TCP buffering : disabled
Packet size : 16
Packet timer(msec) : 0
Packet char(hex) : d
Turnaround delay(msec) :
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 50282
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
```

7. Unmap the profile and set packet size to 1400. Map the profile. Connect with previously connected clients
8. Verify packet size by running "show serial interface serial 0/9" command.

6.5. Configuring keep-alive

1. Use Wireshark to verify that keepalive messages are sent according to the established keepalive interval.
2. Unmap the profile and configure the keep-alive feature as 100 sec. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# no connection-map
Phoenix_Contact(raw-PROF_9_I)# transport protocol tcp
Phoenix_Contact(raw-PROF_9_I-TCP)# direction in
Phoenix_Contact(raw-PROF_9_I-TCP-in)# keep-alive timeout 100
Phoenix_Contact(raw-PROF_9_I-TCP-in)# exit
Phoenix_Contact(raw-PROF_9_I-TCP)# exit
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_I)# end
```

3. Use Wireshark to verify that keepalive messages are sent according to the new keep-alive interval.
4. Verify the values of keep-alive value.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 100
Dynamic master enabled : 0
Max connections : 64
Packetizing : ON
TCP buffering : disabled
```

```

Packet size : 16
Packet timer(msec) : 0
Packet char(hex) : d
Turnaround delay(msec) :
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 50282
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0

```

6.6. Configuring Maximum Connections

1. Unmap the profile and change the max connections number to be equal to the number of clients. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# local server port 15012
Phoenix_Contact(raw-PROF_9_I)# no connection-map
Phoenix_Contact(raw-PROF_9_I)# transport protocol tcp
Phoenix_Contact(raw-PROF_9_I-TCP)# direction in
Phoenix_Contact(raw-PROF_9_I-TCP-in)# packet char off
Phoenix_Contact(raw-PROF_9_I-TCP-in)# packet timeout 10
Phoenix_Contact(raw-PROF_9_I-TCP-in)# max connections 2
Phoenix_Contact(raw-PROF_9_I-TCP-in)# exit
Phoenix_Contact(raw-PROF_9_I-TCP)# exit
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_I)# end

```

2. Try to add an additional client. Confirm that it fails because the number of clients reached its maximum.
3. Verify the maximum clients number.

FOR EXAMPLE: Type the following:

```

Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9

```

```
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15012
Number of clients : 2
Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 100
Dynamic master enabled : 0
Max connections : 2
Packetizing : ON
TCP buffering : disabled
Packet size : 16
Packet timeout(msec) : 10
Packet char(hex) : d
Turnaround delay(msec) :
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 51320
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
Client 2 IP : 192.168.0.100
Client 2 port : 51321
Client 2 RX byte cnt : 0
Client 2 TX byte cnt : 0
Client 2 Packets retry : 0
Client 2 Bytes retry : 0
```

6.7. Configuring Turnaround Delay

1. Unmap the profile and change the Turnaround delay to be maximum value of 1000 ms. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# no connection-map
Phoenix_Contact(raw-PROF_9_I)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# turnaround delay 100
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_I)# end
```

2. Send 2 messages from a client with no delay between them.
3. Verify that the first message is received on the serial app immediately, and the second one after turnaround delay.
4. Verify the turnaround delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15012
Number of clients : 2
Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 100
Dynamic master enabled : 0
Max connections : 2
Packetizing : ON
TCP buffering : disabled
```

```

Packet size : 16
Packet timer(msec) : 10
Packet char(hex) : d
Turnaround delay(msec) :1000
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 51320
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
Client 2 IP : 192.168.0.100
Client 2 port : 51321
Client 2 RX byte cnt : 0
Client 2 TX byte cnt : 0
Client 2 Packets retry : 0
Client 2 Bytes retry : 0

```

6.8. Configuring Hold Time

1. Unmap the profile and change the Hold time to be 100 msec. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# no connection-map
Phoenix_Contact(raw-PROF_9_I)# end

```

```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# hold-time 100
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end

```

```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN

```

```
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_I)# end
```

2. Send 2 messages from a client with no delay between them.
3. Verify that the first message is received on the serial app immediately, and the second one after the hold time value, even that the turnaround delay still 1000.
4. Verify the value of the turnaround delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15012
Number of clients : 2
Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 100
Dynamic master enabled : 0
Max connections : 2
Packetizing : ON
TCP buffering : disabled
Packet size : 16
Packet timeout(msec) : 10
Packet char(hex) : d
Turnaround delay(msec) :1000
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 51320
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
Client 2 IP : 192.168.0.100
Client 2 port : 51321
Client 2 RX byte cnt : 0
```

```
Client 2 TX byte cnt : 0
Client 2 Packets retry : 0
Client 2 Bytes retry : 0
```

6.9. Configuring of No Delays

1. Unmap the profile and change the Hold time and the Turnaround delay to be 0. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# no connection-map
Phoenix_Contact(raw-PROF_9_I)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# turnaround delay 0
Phoenix_Contact(config-serial-if)# hold-time 0
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_I)# end
```

2. Verify the turnaround delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode Server
Direction : In
Transport : TCP
Local Server Port : 15012
Number of clients : 2
```



```

Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 100
Dynamic master enabled : 0
Max connections : 2
Packetizing : ON
TCP buffering : disabled
Packet size 16
Packet timeout(msec) : 10
Packet char(hex) : d
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Client 1 IP : 192.168.0.100
Client 1 port : 51320
Client 1 RX byte cnt : 0
Client 1 TX byte cnt : 0
Client 1 Packets retry : 0
Client 1 Bytes retry : 0
Client 2 IP : 192.168.0.100
Client 2 port : 51321
Client 2 RX byte cnt : 0
Client 2 TX byte cnt : 0
Client 2 Packets retry : 0
Client 2 Bytes retry : 0

```

3. Send 1 message from serial interface and 1 message from a client with minimum delay between them.
4. Verify that the messages arrived immediately.

6.10. Configuring Rx-to-Tx Delay

1. Unmap the profile and change the Rx-to-Tx to be maximum value of 1000 ms. Map the profile. Connect with previously connected clients.

FOR EXAMPLE: Type the following:

– Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
```

```
Phoenix_Contact(raw-PROF_9_I)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_I)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# rx-to-tx delay 1000
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_I)# end
```

2. Verify the values of the hold time, turnaround delay, and rx-to-tx delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : In
Transport : TCP
Local Server Port : 15012
Number of clients : 2
Is suspended by Admin : NO
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 100
Dynamic master enabled : 0
Max connections : 2
Packetizing : ON
TCP buffering : disabled
Packet size : 16
Packet timeout(msec) : 10
Packet char(hex) : d
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 1000
```

```
Client 1 IP : 192.168.0.100
Client 1 port : 58698
Client 1 byte counter : 0
Client 2 IP : 192.168.0.100
Client 2 port : 58705
Client 2 byte counter : 0
```

3. Send a message from the serial interface and a message from a client with a minimum delay between them.
4. Verify that the message from the serial interface to the client is received immediately, but client to serial message comes with the configured Rx-to-Tx delay.

6.11. Clearing Serial Profile Configuration and Deleting Profile

1. Delete the profile and set Rx-to-Tx delay to 0.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# no serial profile PROF_9_IN
Phoenix_Contact(raw-PROF_9_I)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

2. Run “show serial profile interface serial 0/9” to verify that profile is not mapped.

7. Raw Socket IN-OUT Mode Configuration

This section lists the *CLI* configuration steps for IN-OUT Mode Configuration for Raw Socket connection.

1. Configure VLAN 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure VLAN 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact (config-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-serial-if)# baud-rate 115200
```

```
Phoenix_Contact (config-serial-if)# no shutdown
```

```
Phoenix_Contact (config-serial-if)# end
```

7.1. Running Client and Server Modes

1. Map the profile without preparing the server for connection or connecting a client.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_IN_OUT profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN_OUT
```

```
Phoenix_Contact(raw-PROF_9_IN_OUT)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP)# direction in-out
```

```
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP-InOut)# local client port 15011
```

```
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP-InOut)# local server port 15010
```

```
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP-InOut)# remote ipv4 address  
192.168.0.100 port 15011
```

```
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP-InOut)# EXIT
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP)# EXIT
Phoenix_Contact(raw-PROF_9_IN_OUT)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_IN_OUT)# END
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status of the IN_OUT profile (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN_OUT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Client/Server
Direction : InOut
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15011
Local Server Port     : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
Local Client Port     : 15011
KeepAlive interval (sec): 240
Reconnect timer(sec)  : 4
```

3. Verify with the WireShark connection attempts done within the configured reconnect timer interval; actual attempts in software are done every "Reconnect timer(sec)" seconds, can be seen when "debug serial critical" is configured, but Wireshark shows only those attempts which are executed only when TCP stack is ready to execute them. So if the reconnect timeout is big, the Wire-shark attempts are executed with the "reconnect timer" attempts.

7.2. Accepting Connection as a Server

1. Connect with a client application to the server using a configured server IP and port.
2. Open serial application to the configured serial interface number using the configured serial parameters.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status and if the mode became a server.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_IN_OUT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Server
Direction : InOut
Transport : TCP
Local Server Port      : 15010
Number of clients      : 1
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec): 240
Dynamic master enabled : 0
Max connections        : 64
Packetizing            : OFF
TCP buffering           : disabled
Turnaround delay (msec) : 0
Hold time (msec)       : 0
Rx-to-Tx delay (msec)  : 1000
Client 1 IP             : 192.168.0.100
Client 1 port           : 5299
lClient 1 byte counter : 0
```

4. Verify with the WireShark that connection attempts are stopped.

7.3. Testing Server Mode

CONTEXT:

Perform configurations and tests from the Server Configuration - refer to the section *Raw Socket Server Mode Configuration*. All tests should give the same results.

NOTE: REPLACE the profile name and direction with following:

```
Phoenix_Contact # configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN_OUT
Phoenix_Contact(raw-PROF_9_IN_OUT)# transport protocol tcp
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP)# direction in-out
```

NOTE: No need to configure VLAN1 and serial interface baud rate again.

Don't delete the profile at the end of server testing.

7.4. Going back to Client and Server Modes

1. Close the client application.
2. Use Wireshark to verify that Client mode reconnect attempts are back.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status of the INOUT profile (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN_OUT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Client/Server
Direction : InOut
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15011
Local Server Port     : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
Local Client Port     : 15011
KeepAlive interval (sec): 240
Reconnect timer(sec)  : 4
```

7.5. Connecting as Client

1. Start Server to accept connection from the client.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status in Client Mode (it should be connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_IN_OUT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Client
Direction : Out
```

```

Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port   : 15011
Serial RX byte counter : 0
Serial TX byte counter : 0
Network byte counter  : 0
Connection to server  : YES
Local Client Port     : 15011
KeepAlive interval (sec): 240
Reconnect timer(sec)  : 4
Packetizing           : OFF
TCP buffering         : disabled
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec)  : 1000

```

7.6. Testing Client Mode

CONTEXT:

Perform configurations and tests from the Client Configuration - refer to the section *Raw Socket Client Mode Configuration*. All tests should give the same results.

NOTE: REPLACE the profile name and direction with following:

```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_IN_OUT
Phoenix_Contact(raw-PROF_9_IN_OUT)# transport protocol tcp
Phoenix_Contact(raw-PROF_9_IN_OUT-TCP)# direction in-out

```

NOTE: No need to configure VLAN1 and serial interface baud rate again.

Don't delete the profile at the end of server testing

7.7. Clearing the Serial Profile Configuration and Deleting the Profile

1. Delete the profile and set the Rx-to-Tx delay to 0.

FOR EXAMPLE: Type the following:

– Enter the Global Configuration Mode.

```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# no serial profile PROF_9_IN_OUT
Phoenix_Contact(raw-PROF_9_IN_OUT)# end

```

```

Phoenix_Contact # configure terminal

```



```
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

2. Run “show serial profile interface serial 0/9” to verify that profile is not mapped.

8. Raw Socket Client Mode Configuration

This section lists the *CLI* configuration steps for Client Mode Configuration for Raw Socket connection.

1. Configure VLAN 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure VLAN 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact (config-serial-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-serial-if)# baud-rate 115200
```

```
Phoenix_Contact (config-serial-if)# no shutdown
```

```
Phoenix_Contact (config-serial-if)# end
```

8.1. Connecting

1. Map the client without preparing the server for connection.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# direction OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# local client port 15010
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# remote ipv4 address  
192.168.0.100 port 15010
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP) # EXIT
Phoenix_Contact(raw-PROF_9_OUT) # connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_OUT) # END
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : NO
Local Client Port     : 15010
KeepAlive interval (sec) : 240
Reconnect timer(sec)  : 4
Packetizing           : OFF
TCP buffering         : disabled
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec)  : 0
```

3. Confirm that packetizing is disabled, all serial timers (turnaround, hold, and rx-to-tx) are 0, and counters are 0.
4. Verify with the WireShark connection attempts done within the configured reconnect timer interval; actual attempts in software are done every "Reconnect timer(sec)" seconds, can be seen when "debug serial critical" is configured, but Wireshark shows only those attempts which are executed only when TCP stack is ready to execute them. So if the reconnect timeout is big, the Wire-shark attempts are executed with the "reconnect timer" attempts.

8.2. Disconnecting

1. Close the server.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : NO
Local Client Port     : 15010
KeepAlive interval (sec) : 240
Reconnect timer(sec)  : 4
Packetizing           : OFF
TCP buffering         : disabled
Turnaround delay (msec) : 0
Hold time (msec)      : 0
Rx-to-Tx delay (msec) : 0
```

8.3. Sending with Packetizing OFF

1. Send any data from the server. Verify that the serial interface app receives it.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to check counters. Counters should have the same bytes number in field "TCP RX byte counter" and "Serial TX byte counter" that was sent by the client.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```

Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 4
TCP RX byte counter   : 4
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : NO
Local Client Port     : 15010
KeepAlive interval (sec) : 240
Reconnect timer(sec)  : 4
Packetizing           : OFF
TCP buffering         : disabled
Turnaround delay (msec) : 0
Hold time (msec)      : 0
Rx-to-Tx delay (msec) : 0

```

3. Send any data from the serial interface app.
4. Verify all data received on the server with no visible delay.
5. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to check counters. Counters should have the same bytes number in field "TCP RX byte counter" and "Serial TX byte counter" that was sent by the client.

FOR EXAMPLE: Type the following:

```

Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 8
Serial TX byte counter : 4
TCP RX byte counter   : 4

```

```

TCP TX byte counter      : 0
TCP Packets retry       : 0
TCP Bytes retry         : 0
Connection to server    : YES
Local Client Port       : 15010
KeepAlive interval (sec): 240
Reconnect timer(sec)    : 4
Packetizing             : OFF
TCP buffering           : disabled
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay(msec)   : 0

```

8.4. Configuring Packet Timer

1. Unmap the profile, enable packetizing, and set packet timer to 1000 (1 sec). Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(config)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# direction OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# packetizing enable
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# packet timeout 1000
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_OUT)# END
```

2. Send any data from the serial interface app.
3. Verify that all data received on the server with a delay of 1 sec (easily can be compared to the no delay state of packetizing disabled).
4. Run “show serial interface serial 0/9” and verify the value of pack parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_OUT
```

```

Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 18
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : YES
Local Client Port     : 15010
KeepAlive interval (sec): 240
Reconnect timer(sec)  : 4
Packetizing           : ON
TCP buffering         : disabled
Pack size             : 1400
Pack timer(msec)     : 1000
Pack char             : disabled
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0

```

8.5. Configuring Packet Character

1. Unmap the profile, set pack character to "ENTER" (13). Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact # configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(config)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# direction OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# packet timeout 0
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# packet character 13
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# EXIT
Phoenix_Contact(raw-PROF_9_OUT-TCP)# EXIT
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_OUT)# END
```

2. Send some data from the serial interface app with using "ENTER" character sometimes.
3. Verify that data arrives to the server only upon using "ENTER" character.
4. Run "show serial interface serial 0/9" and verify the value of pack parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 18
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : NO
Local Client Port     : 15010
KeepAlive interval (sec) : 240
Reconnect timer(sec)  : 4
Packetizing           : ON
TCP buffering         : disabled
Pack size              : 1400
Pack timer(msec)      : 0
Pack char(hex) : d
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec)  : 0
```


8.6. Configuring Packet Size

1. Unmap the profile and set pack size to 16. Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact # configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(config)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# direction OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# packet size 16
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_OUT)# END
```

2. Send any 17 bytes of data from the serial interface app without using "ENTER" character.
3. Verify that 16 bytes of data arrive to the server.
4. Type "ENTER" character.
5. Verify that the remaining data arrives to the server.
6. Run "show serial interface serial 0/9" and verify the value of pack parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_OUT
```

```
Serial interface : 9
```

```
Mode : Client
```

```
Direction : Out
```

```
Transport : TCP
```

```
Remote Server IP      : 192.168.0.100
```

```
Remote Server Port    : 15010
```

```
Serial RX byte counter : 18
```

```
Serial TX byte counter : 0
```

```
TCP RX byte counter   : 0
```

```
TCP TX byte counter   : 0
```

```
TCP Packets retry     : 0
```

```
TCP Bytes retry       : 0
```

```
Connection to server  : NO
```

```

Local Client Port      : 15010
KeepAlive interval (sec): 240
Reconnect timer(sec)  : 4
Packetizing           : ON
TCP buffering         : disabled
Pack size             : 16
Pack timer(msec)      : 0
Pack char (hex)      : d
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec) : 0

```

7. Unmap the profile, set pack size to 1400. Map the profile. Connect with previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(config)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# transport protocol tcp
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# direction OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# packet size 1400
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT-TCP)# EXIT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_OUT)# END
```

8.7. Configuring KEEP ALIVE

1. Use Wireshark to verify that keepalive messages are sent according to keepalive interval.
2. Unmap the profile, change the keepalive interval to different one. Map the profile. Connect with previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(config)# no connection-map
Phoenix_Contact(raw-PROF_9_OUT)# transport protocol tcp
Phoenix_Contact(raw-PROF_9_OUT-TCP)# direction OUT
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# keep alive timeout 60
Phoenix_Contact(raw-PROF_9_OUT-TCP-OUT)# EXIT
Phoenix_Contact(raw-PROF_9_OUT-TCP)# EXIT
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_OUT)# END
```

3. Use Wireshark to verify that keepalive messages are sent according to keepalive interval.
4. Run “show serial interface serial 0/9” and verify the keep alive value.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 18
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : YES
Local Client Port     : 15010
KeepAlive interval (sec) : 60
Reconnect timer(sec)  : 4
Packetizing           : ON
TCP buffering         : disabled
Pack size             : 16
Pack timer(msec)      : 0
Pack char (hex)      : d
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec) : 0
```

8.8. Configuring Turnaround Delay

1. Unmap the profile and change the turnaround delay to be maximum. Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# turnaround delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

2. Send 2 messages from the server with no delay between them.
3. Verify that the first message received on the serial app immediately, and the second one after turnaround delay.
4. Run “show serial interface serial 0/9” and verify the turnaround delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_OUT
```

```
Serial interface : 9
```

```
Mode : Client
```

```
Direction : Out
```

```
Transport : TCP
```

```
Remote Server IP : 192.168.0.100
```

```
Remote Server Port : 15010
```

```
Serial RX byte counter : 0
```

```
Serial TX byte counter : 0
```

```

TCP RX byte counter      : 0
TCP TX byte counter      : 0
TCP Packets retry        : 0
TCP Bytes retry          : 0
Connection to server     : YES
Local Client Port        : 15010
KeepAlive interval (sec) : 60
Reconnect timer(sec)    : 4
Packetizing              : ON
TCP buffering            : disabled
Pack size                 : 16
Pack timer(msec)         : 0
Pack char (hex) : d
Turnaround delay(msec)  : 1000
Hold time(msec)          : 0
Rx-to-Tx delay(msec)    : 0

```

8.9. Configuring Hold Time Delay

1. Unmap the profile and change the Hold time to be 100 msec. Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# hold-time 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_9_OUT)# end
```

2. Send 2 messages from the server with no delay between them.
3. Verify that the first message received on the serial app immediately, and the second one after turnaround delay.
4. Run “show serial interface serial 0/9” and verify the turnaround delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : YES
Local Client Port     : 15010
KeepAlive interval (sec) : 60
Reconnect timer(sec)  : 4
Packetizing           : ON
TCP buffering         : disabled
Pack size             : 16
Pack timer(msec)      : 0
Pack char (hex)      : d
Turnaround delay(msec) : 1000
Hold time(msec)       : 100
Rx-to-Tx delay(msec)  : 0
```

8.10. Configuring No Delays

1. Unmap the profile and change the hold time and the turnaround delay to be 0. Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# hold-time 0
```

```
Phoenix_Contact(config-serial-if)# turnaround delay 0
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

2. Run “show serial interface serial 0/9” and verify that there are no hold time and turnaround delays.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_9_OUT
```

```
Serial interface : 9
```

```
Mode : Client
```

```
Direction : Out
```

```
Transport : TCP
```

```
Remote Server IP : 192.168.0.100
```

```
Remote Server Port : 15010
```

```
Serial RX byte counter : 0
```

```
Serial TX byte counter : 0
```

```
TCP RX byte counter : 0
```

```
TCP TX byte counter : 0
```

```
TCP Packets retry : 0
```

```
TCP Bytes retry : 0
```

```
Connection to server : YES
```

```
Local Client Port : 15010
```

```
KeepAlive interval (sec): 60
```

```

Reconnect timer(sec)      : 4
Packetizing               : ON
TCP buffering             : disabled
Pack size                 : 16
Pack timer(msec)         : 0
Pack char (hex)          : d
Turnaround delay(msec)   : 0
Hold time(msec)          : 0
Rx-to-Tx delay(msec)     : 0

```

3. Send 1 message from the serial and 1 message form the server with minimal delay between them.
4. Confirm that the messages arrived immediately.

8.11. Configuring Rx-to-Tx Delay

1. Unmap the profile and change the Rx-to-Tx to its maximum. Map the profile. Connect with a previously connected server.

FOR EXAMPLE: Type the following:

– Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

– Configure profile PROF_9_OUT.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# no connection-map
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_9_OUT
```

```
Phoenix_Contact(raw-PROF_9_OUT)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_9_OUT)# end
```

2. Run “show serial interface serial 0/9” and verify the hold time, turnaround, and rx-to-tx delays.

FOR EXAMPLE: Type the following:


```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_OUT
Serial interface : 9
Mode : Client
Direction : Out
Transport : TCP
Remote Server IP      : 192.168.0.100
Remote Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
TCP RX byte counter   : 0
TCP TX byte counter   : 0
TCP Packets retry     : 0
TCP Bytes retry       : 0
Connection to server  : YES
Local Client Port     : 15010
KeepAlive interval (sec) : 60
Reconnect timer(sec)  : 4
Packetizing           : ON
TCP buffering         : disabled
Pack size             : 16
Pack timer(msec)      : 0
Pack char (hex)      : d
Turnaround delay(msec) : 0
Hold time(msec)       : 0
Rx-to-Tx delay(msec) : 1000
```

3. Send 1 message from the serial and 1 message form the server with minimal delay between them.
4. Confirm that the message from serial to the server is received immediately, but server to serial after Rx-to-Tx delay.

8.12. Clearing the Serial Profile Configuration and Deleting the Profile

1. Delete the profile and set Rx-to-Tx delay to 0.

FOR EXAMPLE: Type the following:

– Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# no serial profile PROF_9_OUT
Phoenix_Contact(raw-PROF_9_I)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

2. Run “show serial profile interface serial 0/9” to verify that profile is not mapped.

8.13. Enable Mirroring

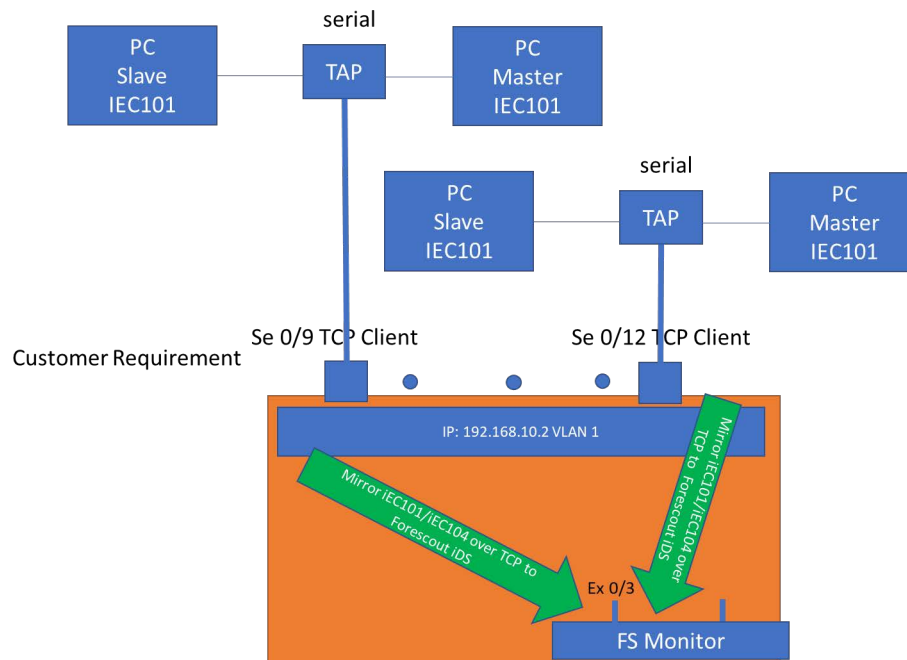
To enable the Serial TCP mirroring feature on raw-socket profile acting as TCP client, use the **enable mirroring interface** command in Direction (Out) Mode (Raw). Enabling this feature would simulate TCP stream with the received serial data as payload for the TCP traffic.

Serial TCP Mirroring

Serial TCP mirroring feature is intended to monitor the serial data from the Serial Tap as TCP traffic.

CONTEXT:

The destination port simulates an exchange between a TCP client and TCP server and forwards (mirrors) it on a separate destination Ethernet port.

Figure 1: Serial TCP Mirroring

1. To enable the Serial TCP mirroring feature on raw-socket profile acting as TCP client 1, perform the following:

FOR EXAMPLE: Example 1

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_X2
Phoenix_Contact(raw-PROF_X2)# transport protocol tcp
Phoenix_Contact(raw-PROF_X2-TCP)# direction out
Phoenix_Contact(raw-PROF_X2-TCP-Out)# enable mirroring interface
gigabitethernet 0/10
Phoenix_Contact(raw-PROF_X2-TCP-Out)# remote ipv4 address 192.168.10.16
port 15030
Phoenix_Contact(raw-PROF_X2-TCP-Out)# local client port 15036
Phoenix_Contact(raw-PROF_X2-TCP-Out)# exit
Phoenix_Contact(raw-PROF_X2-TCP)# exit
Phoenix_Contact(raw-PROF_X2)# connection-map interface serial 0/17
Phoenix_Contact(raw-PROF_X2)# end
```

2. Example 2

FOR EXAMPLE: Perform the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_X2
```

```
Phoenix_Contact(raw-PROF_X2)# transport protocol tcp
Phoenix_Contact(raw-PROF_X2-TCP)# direction out
Phoenix_Contact(raw-PROF_X2-TCP-Out)# enable mirroring interface
gigabitethernet 0/10 dest-mac
00:01:02:03:04:05 source-ip 192.168.11.12
Phoenix_Contact(raw-PROF_X2-TCP-Out)# exit
Phoenix_Contact(raw-PROF_X2-TCP-Out)# remote ipv4 address 192.168.10.16
port 15030
Phoenix_Contact(raw-PROF_X2-TCP-Out)# local client port 15036
Phoenix_Contact(raw-PROF_X2-TCP)# exit
Phoenix_Contact(raw-PROF_X2)# connection-map interface serial 0/17
Phoenix_Contact(raw-PROF_X2)# end
```

FOR EXAMPLE:

Verification

Once the feature is enabled on any given serial profile, the serial data received on the corresponding serial port shall be encapsulated as a TCP packet with the specific L2, L3 and L4 headers as specified in the configuration. This encapsulated traffic would be sent on the configured destination port. The Wire-shark capture of the traffic received from the destination port should show a valid TCP transaction.

CONTEXT:

```
Phoenix_Contact# show serial profile interface serial 0/10
Profile                : PROF_X1_OUT
Status                 : Active
Serial interface       : 18
Protocol               : RAW Socket
Mode                   : Client
Direction              : Out
Transport              : TCP
Remote Server IP       : 192.168.10.15
Remote Server Port     : 15030
Serial RX byte counter : 100015
Serial TX byte counter : 0
TCP RX byte counter    : 0
TCP Bytes retry        : 0
Connection to server   : TCP Mirroring enabled
Mirror Destination Port : 10
Mirror Source IP        : 192.168.111.112
Mirror Destination Mac  : 00:01:02:03:04:05
Local Client Port      : 15035
KeepAlive interval (sec) : 240
```

```
Reconnect timer(sec)      : 120
Packetizing               : OFF
TCP buffering             : disabled
Turnaround delay(msec)   : 0
Hold time(msec)          : 0
Rx-to-Tx delay(msec)     : 0
```

8.14. Disable Mirroring

To disable the Serial TCP mirroring feature on raw-socket profile acting as TCP client, use the **disable mirroring interface** command in Direction (Out) Mode (Raw). Disabling the feature would mean that the serial profile would act as traditional TCP client attempting to connect to the specified remote TCP server.

CONTEXT:

To disable the Serial TCP mirroring feature on raw-socket profile acting as TCP client, perform the following:

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_X2
Phoenix_Contact(raw-PROF_X2)# transport protocol tcp
Phoenix_Contact(raw-PROF_X2-TCP)# direction out
Phoenix_Contact(raw-PROF_X2-TCP-Out)# disable mirroring
Phoenix_Contact(raw-PROF_X2-TCP-Out)# exit
Phoenix_Contact(raw-PROF_X2-TCP)# exit
Phoenix_Contact(raw-PROF_X2)# connection-map interface serial 0/17
Phoenix_Contact(raw-PROF_X2)# end

Phoenix_Contact(config)# serial connection-type raw profile PROF_X1_OUT
Phoenix_Contact(raw-PROF_X1_OUT)# transport protocol tcp
Phoenix_Contact(raw-PROF_X1_OUT-TCP)# disable mirroring
Phoenix_Contact(raw-PROF_X1_OUT-TCP)#
```

9. Raw Preemptive Connection Configuration

This section lists the *CLI* configuration steps for Raw Preemptive connection.

1. Configure VLAN 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure VLAN 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact (config-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-serial-if)# baud-rate 115200
```

```
Phoenix_Contact (config-serial-if)# no shutdown
```

```
Phoenix_Contact (config-serial-if)# end
```

- Configure profile PROF_9_PREEMPT.

```
Phoenix_Contact(config)# serial connection-type preemptive-raw profile  
PROF_9_PREEMPT
```

```
Phoenix_Contact(preemptive-PROF_9_PREEMPT)# local server port 15010
```

```
Phoenix_Contact(preemptive-PROF_9_PREEMPT)# permanent-client ipv4 address  
192.168.0.10
```

```
Phoenix_Contact(preemptive-PROF_9_PREEMPT)# connection-map interface  
serial 0/9
```

9.1. General

1. Open serial application to the configured serial interface number using configured serial parameters.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status. The current number of clients is 0.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_PREEMPT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 0
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : OFF
TCP buffering : disabled
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: 0
Dynamic master idle TO : 10
```

3. Verify that turnaround delay, hold time and rx-to-tx delay are all 0; Dynamic master is not connected and the maximum client number (connections) is 2.

9.2. Connecting to Permanent Client

1. Connect with a client application using permanent IP address to the server using configured server IP and port.
2. Perform the following configuration.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type preempt profile preempt1
Phoenix_Contact(preemptive-preempt1)# local server port 15070
Phoenix_Contact(preemptive-preempt1)# permanent-client ipv4 address
192.168.20.66
Phoenix_Contact(preemptive-preempt1)dynamic idle time 60
Phoenix_Contact(preemptive-preempt1)# connection-map interface serial
0/11
```

3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the permanent client is connected and dynamic master is not.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/11
Profile : preempt1
Serial interface : 11
Protocol : RAW Socket
Serial interface : 11
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15070
Number of clients : 0
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : OFF
TCP buffering : disabled
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: NO
Dynamic master idle TO : 10
Permanent Client IP : 192.168.20.66
Permanent Client port : 51651
Permanent Client RX cnt : 0
Permanent Client TX cnt : 0
Permanent Packets retry : 0
Permanent Bytes retry : 0
```


4. Connect with the permanent client IP address of an additional client.
5. Confirm that the connection is refused.
6. Run again one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the only one permanent client is connected, current clients number is 1, and a dynamic master is not connected.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : preempt1
Serial interface : 11
Protocol : RAW Socket
Serial interface : 11
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15070
Number of clients : 0
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : OFF
TCP buffering : disabled
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: NO
Dynamic master idle TO : 10
Permanent Client IP : 192.168.20.66
Permanent Client port : 51651
Permanent Client RX cnt : 0
Permanent Client TX cnt : 0
Permanent Packets retry : 0
Permanent Bytes retry : 0
```

9.3. Functional Tests for Permanent Client (Same Tests as in Server)

Functional tests include all tests, except the deletion of the profile. Perform all tests from the server test plan - refer to section "Raw Socket Server Mode Configuration". All results should be in place except of number of clients (this mode has only one client). Do not delete this profile.


```
Permanent Client port    : 63262
Permanent Client RX cnt  : 0
Permanent Client TX cnt  : 0
Permanent Packets retry  : 0
Permanent Bytes retry    : 0
Dynamic Master IP        : 192.168.0.101
Dynamic Master Port      : 63293
Dynamic Master RX cnt    : 0
Dynamic Master TX cnt    : 0
Dynamic Packets retry    : 0
Dynamic Bytes retry      : 0
```

3. Verify that Dynamic master is connected, the number of clients is 2, and the number of current clients is 2 (permanent and dynamic master).

9.6. Dynamic Master Disconnection after Idle Timeout

1. Wait 10 seconds (the Dynamic master idle timeout (TO) is 10) without sending any data from the dynamic master.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the dynamic master is disconnected due to idle timeout.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_PREEMPT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : OFF
TCP buffering : disabled
Turnaround delay(msec) : 0
```

```

Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: NO
Dynamic master idle TO : 10
Permanent Client IP      : 192.168.0.100
Permanent Client port    : 51651
Permanent Client RX cnt  : 0
Permanent Client TX cnt  : 0
Permanent Packets retry  : 0
Permanent Bytes retry    : 0

```

3. Verify that Dynamic master is connected, the number of clients is 2 and the number of current clients is 2 (permanent and dynamic master).

9.7. Connecting Second Dynamic Master

1. Try to connect with an additional client which IP address differs from the permanent client's IP address (a dynamic master).
2. Verify that second dynamic master is not connected
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status.
4. Verify that Dynamic master is connected, the number of max clients is 2, and the number of current clients is 2 (permanent and dynamic).

9.8. Configuring Dynamic Packet Timeout

1. Unmap the profile and set dynamic master inactive TO to 1000 (1 sec). Map the profile. Connect with previously connected clients.
 - Configure profile PROF_9_PREEMPT.


```

Phoenix_Contact(config)# serial connection-type preemptive-raw profile
PROF_9_PREEMPT
Phoenix_Contact(preemptive-p2)# no connection-map
Phoenix_Contact(preemptive-p2)# dynamic packet timeout 1000
Phoenix_Contact(preemptive-p2)# connection-map interface serial 0/9
Phoenix_Contact(preemptive-p2)# end

```
2. Send data from the serial interface app.
3. Verify that all data is received by the master client only with a delay of 1 sec (it easily can be compared to the no delay state with packetizing disabled).

4. Send "ENTER" character from serial interface app.
5. Verify the "ENTER" character arrives at the master client approximately after 1 sec.

9.9. Configuring Dynamic Packet Character

1. Unmap the profile, set dynamic packet character to 98 ('b'). Map the profile. Connect with previously connected clients.

– Configure profile PROF_9_PREEMPT.

```
Phoenix_Contact(config)# serial connection-type preemptive-raw profile
PROF_9_PREEMPT
Phoenix_Contact(preemptive-p2)# no connection-map
Phoenix_Contact(preemptive-p2)# packetizing enable
Phoenix_Contact(preemptive-p2)# dynamic packet timeout 0
Phoenix_Contact(preemptive-p2)# dynamic packet char 98
Phoenix_Contact(preemptive-p2)# connection-map interface serial 0/9
Phoenix_Contact(preemptive-p2)# end
```

2. Send data from the serial interface app using "b" character sometimes.
3. Verify that data arrives at the master client only upon using the "b" character.
4. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the value of the dynamic packet character.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_PREEMPT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : ON
TCP buffering : disabled
```

```

Pack size                : 1400
Pack timer(msec)        : 0
Pack char(hex)           : 62
Turnaround delay(msec)  : 0
Hold time(msec)         : 0
Rx-to-Tx delay(msec)    : 0
Dynamic master connected: YES
Dynamic master idle TO  : 10
Dynamic pack timer(msec): 10
Dynamic pack char        : disabled
Permanent Client IP     : 192.168.0.100
Permanent Client port   : 63262
Permanent Client RX cnt : 0
Permanent Client TX cnt : 0
Permanent Packets retry : 0
Permanent Bytes retry   : 0
Dynamic Master IP       : 192.168.0.101
Dynamic Master Port     : 63293
Dynamic Master RX cnt   : 0
Dynamic Master TX cnt   : 0
Dynamic Packets retry   : 0
Dynamic Bytes retry     : 0

```

9.10. Configuring Packet Size

1. Unmap the profile and set packet size to 16. Map the profile. Connect with previously connected clients.
 - Configure profile PROF_9_PREEMPT.


```

Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type preemptive-raw profile
PROF_9_PREEMPT
Phoenix_Contact(preemptive-p2)# no connection-map
Phoenix_Contact(preemptive-p2)# packet size 16
Phoenix_Contact(preemptive-p2)# connection-map interface serial 0/9
Phoenix_Contact(preemptive-p2)# end

```
2. Send some 17 bytes of data from the serial interface app without using "b" character.
3. Verify that 16 bytes of data arrives at the master client.
4. Type a "b" character.
5. Verify that remaining data arrives at the master client.

6. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the packet size.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_PREEMPT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : ON
TCP buffering : disabled
Pack size : 16
Pack timer(msec) : 0
Pack char(hex) : 62
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: YES
Dynamic master idle TO : 10
Dynamic pack timer(msec): 10
Dynamic pack char : disabled
Permanent Client IP : 192.168.0.100
Permanent Client port : 63262
Permanent Client RX cnt : 0
Permanent Client TX cnt : 0
Permanent Packets retry : 0
Permanent Bytes retry : 0
Dynamic Master IP : 192.168.0.101
Dynamic Master Port : 63293
Dynamic Master RX cnt : 0
Dynamic Master TX cnt : 0
```

```
Dynamic Packets retry    : 0
Dynamic Bytes retry     : 0
```

7. Unmap the profile and set packet size to 1400 (back to the default value). Connect with previously connected clients.

- Configure profile PROF_9_PREEMPT.

```
Phoenix_Contact(config)# serial connection-type preemptive-raw profile
PROF_9_PREEMPT
Phoenix_Contact(preemptive-p2)# no connection-map
Phoenix_Contact(preemptive-p2)# packet size 1400
Phoenix_Contact(preemptive-p2)# connection-map interface serial 0/9
Phoenix_Contact(preemptive-p2)# end
```

9.11. Sending Data from Permanent Client

1. Send data from the permanent client.
2. Confirm that the data is ignored and not sent to the serial interface.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the Serial TX byte counter did not change.

9.12. Disabling Packetizing

1. Unmap the profile and disable packetizing. Map the profile. Connect with previously connected clients.

- Configure profile PROF_9_PREEMPT.

```
Phoenix_Contact(config)# serial connection-type preemptive-raw profile
PROF_9_PREEMPT
Phoenix_Contact(preemptive-p2)# no connection-map
Phoenix_Contact(preemptive-p2)# packetizing disable
Phoenix_Contact(preemptive-p2)# connection-map interface serial 0/9
Phoenix_Contact(preemptive-p2)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the packetizing status.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_PREEMPT
Serial interface : 9
Protocol : RAW Socket
```



```
Serial interface : 9
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 2
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : OFF
TCP buffering : disabled
Pack size : 1400
Pack timer(msec) : 0
Pack char(hex) : 62
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: YES
Dynamic master idle TO : 10
Dynamic pack timer(msec): 10
Dynamic pack char      : disabled
Permanent Client IP    : 192.168.0.100
Permanent Client port  : 63262
Permanent Client RX cnt : 0
Permanent Client TX cnt : 0
Permanent Packets retry : 0
Permanent Bytes retry  : 0
Dynamic Master IP      : 192.168.0.101
Dynamic Master Port    : 63293
Dynamic Master RX cnt  : 0
Dynamic Master TX cnt  : 0
Dynamic Packets retry  : 0
Dynamic Bytes retry    : 0
```

9.13. Sending Data with Packetizing OFF

1. Send some data from the serial interface app.
2. Verify that all data is received by the master client with no delay (no packetizing).

9.14. Disconnecting Dynamic Client (back to Permanent Client)

1. Disconnect the dynamic master client.
2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status; now, the number of clients is 1.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_9_PREEMPT
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Mode : Preemptive (Server)
Direction : In
Transport : TCP
Local Server Port : 15010
Number of clients : 1
Serial RX byte counter : 0
Serial TX byte counter : 0
KeepAlive interval (sec) : 240
Dynamic master enabled : 1
Max connections : 2
Packetizing : OFF
TCP buffering : disabled
Turnaround delay(msec) : 0
Hold time(msec) : 0
Rx-to-Tx delay(msec) : 0
Dynamic master connected: NO
Dynamic master idle TO : 60
Permanent Client IP      : 192.168.0.100
Permanent Client port    : 51651
Permanent Client RX cnt  : 0
Permanent Client TX cnt  : 0
Permanent Packets retry  : 0
Permanent Bytes retry   : 0
```

3. Confirm that turnaround delay, hold time, and Rx-to-Tx delay are all 0; Dynamic master is not connected, and the maximum clients number is 2.

9.15. Clearing Serial Configuration and Deleting Profile

1. Delete the profile.

- Configure profile PROF_9_PREEMPT.

```
Phoenix_Contact # configure terminal
```

```
Phoenix_Contact(config)# no serial connection PROF_9_PREEMPT
```

```
Phoenix_Contact(preemptive-p2)# end
```

2. Run "show serial profile interface serial 0/9" to verify that that profile is not mapped.

10. Raw Socket UDP Mode Configuration

This section lists the *CLI* configuration steps for *UDP* Configuration for Raw Socket connection. Use *UDP* to speed up transmissions by enabling the transfer of data before an agreement is provided by the receiving party or to broadcast to any machine(s) listening to the server.

1. Configure VLAN 1.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode.

```
Phoenix_Contact(config)# interface vlan 1
```

- Configure VLAN 1.

```
Phoenix_Contact(config-if)# no shutdown
```

```
Phoenix_Contact(config-if)# ip address 192.168.0.40 255.255.255.0
```

```
Phoenix_Contact (config-if)# end
```

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Enter the Interface Configuration Mode for 0/9.

```
Phoenix_Contact(config)# interface serial 0/9
```

- For example, configure the baud rate.

```
Phoenix_Contact(config-if)# baud-rate 115200
```

```
Phoenix_Contact (config-if)# no shutdown
```

```
Phoenix_Contact (config-if)# end
```

10.1. Configuring UDP

1. Map the profile and configure *UDP* (User Datagram Protocol) and local port.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# local server port 15010
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# EXIT
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# END
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status of the UDP profile (it should be not connected).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 64
Remote connections   : 0
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
Pack size            : 1400
Pack timer(msec)     : 10
Pack char (hex)      : disabled
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0
```

10.2. Configuring Maximum Connections

1. Unmap the profile and configure maximum connections as 2. Map the profile back.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
Phoenix_Contact(raw-PROF_1_UDP-UDP)# max udp connections 2
Phoenix_Contact(raw-PROF_1_UDP-UDP)# EXIT
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_1_UDP)# END
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status of the UDP profile (the number of maximum connections should be 2).

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 2
Remote connections  : 0
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Server Port    : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
Pack size            : 1400
Pack timer(msec)     : 10
Pack char (hex)      : disabled
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0
```

10.3. Sending Data

1. Try to send data from *UDP* app.
2. Verify that no data passed to the serial app.

10.4. Adding Connections while Profile is Unmapped

1. Unmap the profile and add 2 connections.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# add udp-host 192.168.0.100 port
50537
Phoenix_Contact(raw-PROF_1_UDP-UDP)# add udp-host 192.168.0.100 port
58725
Phoenix_Contact(raw-PROF_1_UDP-UDP)# EXIT
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_1_UDP)# END
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the current connections are 2 and the proper remote connections IPs and ports.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 2
Remote connections   : 2
Remote Ip 1          : 192.168.0.100
Remote Port 1        : 50537
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 0
Remote Ip 2          : 192.168.0.100
Remote Port 2        : 58725
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 0
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Port           : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
Pack size            : 1400
Pack timer(msec)     : 10
Pack char (hex): disabled
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0
```

10.5. Removing Connections while Profile is Unmapped

1. Remove two connections.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# remove udp-host 192.168.0.100 port 50537
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# remove udp-host 192.168.0.100 port 58725
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# EXIT
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# END
```

2. Verify that connection is removed.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the remote connection was removed, max connections are 2, current connection is 1, and local ports are correct.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Transport : UDP
```

```
Max connections      : 2
```

```
Remote connections   : 0
```

```
Remote Packets Dropped : 0
```

```
Remote Bytes Dropped : 0
```

```
Local Port           : 15010
```

```
Serial RX byte counter : 7
```

```
Serial TX byte counter : 5
```

```
Pack size            : 1400
```

```
Pack timer(msec)     : 10
```

```
Pack char (hex)      : disabled
```

```
Turnaround delay(msec) : 0
```



```
Hold time(msec)           : 0
Rx-to-Tx delay(msec)     : 0
```

10.6. Adding Connection on the Fly

1. Add two connections.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# add udp-host 192.168.0.100 port
50537
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# add udp-host 192.168.0.100 port
58725
```

```
Phoenix_Contact(raw-PROF_1_UDP-UDP)# EXIT
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# END
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the current connections are 2 and the remote connections IPs and ports are proper.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Transport : UDP
```

```
Max connections           : 2
```

```
Remote connections        : 2
```

```
Remote Ip 1               : 192.168.0.100
```

```
Remote Port 1             : 50537
```

```
Remote Udp 1 RX cnt       : 0
```

```
Remote Udp 1 TX cnt       : 0
```

```
Remote Ip 2               : 192.168.0.100
```

```
Remote Port 2             : 58725
```

```
Remote Udp 1 RX cnt       : 0
```

```
Remote Udp 1 TX cnt       : 0
```

```
Remote Packets Dropped    : 0
```

```
Remote Bytes Dropped      : 0
Local Port                 : 15010
Serial RX byte counter    : 0
Serial TX byte counter    : 0
Pack size                  : 1400
Pack timer(msec)          : 10
Pack char (hex)           : disabled
Turnaround delay(msec)    : 0
Hold time(msec)           : 0
Rx-to-Tx delay(msec)     : 0
```

10.7. Sending Data from Registered UDP Connection (all connections, one at a time)

1. Attempt to send data from *UDP* app.
2. Verify that data passed to the serial app.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that serial TX counter is incremented.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 2
Remote connections   : 2
Remote Ip 1          : 192.168.0.100
Remote Port 1        : 50537
Remote Udp 1 RX cnt  : 5
Remote Udp 1 TX cnt  : 0
Remote Ip 2          : 192.168.0.100
Remote Port 2        : 58725
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 0
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Port           : 15010
```

```
Serial RX byte counter : 0
Serial TX byte counter : 5
Pack size               : 1400
Pack timer(msec)       : 10
Pack char (hex)        : disabled
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay(msec)   : 0
```

10.8. Sending Data from Not Registered UDP Connection

1. Attempt to send data from *UDP* app.
2. Verify no data passed to the serial app.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that serial TX counter is not incremented.

10.9. Sending Data from Serial App

1. Attempt to send data from *UDP* app.
2. Verify that data passed to the serial app.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that serial TX counter is incremented.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 2
Remote connections   : 2
Remote Ip  1         : 192.168.0.100
Remote Port 1       : 50537
Remote Udp 1 RX cnt : 5
Remote Udp 1 TX cnt : 7
Remote Ip  2         : 192.168.0.100
Remote Port 2       : 58725
```

```
Remote Udp 1 RX cnt    : 0
Remote Udp 1 TX cnt    : 7
Remote Packets Dropped : 0
Remote Bytes Dropped   : 0
Local Port              : 15010
Serial RX byte counter  : 7
Serial TX byte counter  : 5
Pack size               : 1400
Pack timer(msec)       : 10
Pack char (hex)        : disabled
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay(msec)   : 0
```

10.10. Adding Connection on the Fly Not Allowed

1. Add 1 connection.
2. Verify that connection is not added due to max connections number reached.
3. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify the connection status, and that the max connections are 2, current connections are 2, and ports are correct.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 2
Remote connections   : 2
Remote Ip  1         : 192.168.0.100
Remote Port 1        : 50537
Remote Udp 1 RX cnt  : 5
Remote Udp 1 TX cnt  : 7
Remote Ip  2         : 192.168.0.100
Remote Port 2        : 58725
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 7
Remote Packets Dropped : 0
```

```

Remote Bytes Dropped      : 0
Local Port                : 15010
Serial RX byte counter    : 7
Serial TX byte counter    : 5
Pack size                 : 1400
Pack timer(msec)         : 10
Pack char (hex)          : disabled
Turnaround delay(msec)   : 0
Hold time(msec)          : 0
Rx-to-Tx delay(msec)    : 0

```

10.11. Removing Connection on the Fly

1. Remove 1 connection.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP)# remove udp-host 192.168.0.100 port 50537
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Run one of the commands "show serial profile all" or "show serial profile name YOUR_PROFILE_NAME" or "show serial profile interface serial 0/YOUR_INT_NUM" to verify that the current connections are 2 and the remote connections IPs and ports are correct.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Transport : UDP
```

```
Max connections      : 2
```

```
Remote connections   : 1
```

```
Remote Ip 1          : 192.168.0.100
```

```
Remote Port 1        : 58725
```

```
Remote Udp 1 RX cnt  : 5
```

```
Remote Udp 1 TX cnt  : 7
```

```
Remote Packets Dropped : 0
```

```

Remote Bytes Dropped      : 0
Local Port                 : 15010
Serial RX byte counter    : 7
Serial TX byte counter    : 5
Pack size                  : 1400
Pack timer(msec)          : 10
Pack char (hex)           : disabled
Turnaround delay(msec)   : 0
Hold time(msec)           : 0
Rx-to-Tx delay(msec)     : 0

```

10.12. Configuring Packet Timer

1. Unmap the profile and set the packet timer to 1000 (1 sec). Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP)# packet timeout 1000
```

```
Phoenix_Contact(raw-PROF_1_UDP)# exit
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Send data from the serial interface app.
3. Verify all data received on the *UDP* app with a delay of 1 sec (it easily can be compared to the no delay state of packetizing disabled).
4. Run “show serial interface serial 0/9” and verify the values of packet parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Transport : UDP
```

```
Max connections      : 64
```

```
Remote connections   : 1
```

```
Remote Ip 1          : 192.168.0.100
```

```
Remote Port 1        : 50961
```

```

Remote Udp 1 RX cnt      : 0
Remote Udp 1 TX cnt      : 8
Remote Packets Dropped   : 0
Remote Bytes Dropped     : 0
Local Port                : 15010
Serial RX byte counter    : 8
Serial TX byte counter    : 0
Pack size                 : 1400
Pack timer(msec)         : 1000
Pack char (hex)          : disabled
Turnaround delay(msec)   : 0
Hold time(msec)          : 0
Rx-to-Tx delay(msec)    : 0

```

10.13. Configuring Packet Characters Limit

1. Unmap the profile and set pack char to "ENTER" (13). Map the profile. Connect with previously connected server.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile
PROF_1_UDPPROF_1_UDPPROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP)# packet timeout 1000
```

```
Phoenix_Contact(raw-PROF_1_UDP)# packet char 13
```

```
Phoenix_Contact(raw-PROF_1_UDP)# exit
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Send data from the serial interface app with using "ENTER" character sometimes.
3. Verify that data arrives to the UDP app only upon using "ENTER" character.
4. Run "show serial interface serial 0/9" and verify the values of packet parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```

Serial interface : 9
Transport : UDP
Max connections      : 64
Remote connections   : 1
Remote Ip 1          : 192.168.0.100
Remote Port 1        : 50961
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 8
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Port           : 15010
Serial RX byte counter : 8
Serial TX byte counter : 0
Pack size            : 1400
Pack timer(msec)     : 0
Pack char (hex)      : d
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0

```

10.14. Configuring Packet Size

1. Unmap the profile and set packet size to 16. Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
```

```
Phoenix_Contact(raw-PROF_1_UDP)# packet size 16
```

```
Phoenix_Contact(raw-PROF_1_UDP)# exit
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Send 17 bytes of data from the serial interface app without using "ENTER" character.
3. Verify that 16 bytes of data arrives at the *UDP* app.
4. Type "ENTER" character.
5. Verify that remaining data arrives at the *UDP* app.

6. Run “show serial interface serial 0/9” and verify the values of packet parameters.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 64
Remote connections   : 1
Remote Ip 1          : 192.168.0.100
Remote Port 1        : 50961
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 8
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Port           : 15010
Serial RX byte counter : 8
Serial TX byte counter : 0
Pack size            : 16
Pack timer(msec)     : 0
Pack char (hex)      : d
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0
```

7. Unmap the profile and set packet size to 1400 (default). Map the profile.

FOR EXAMPLE: Type the following:

– Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

– Configure profile PROF_1_UDP profile.

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
Phoenix_Contact(raw-PROF_1_UDP)# transport protocol udp
Phoenix_Contact(raw-PROF_1_UDP)# packet size 1400
Phoenix_Contact(raw-PROF_1_UDP)# exit
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_1_UDP)# end
```

10.15. Configuring Turnaround Delay

1. Unmap the profile and change the turnaround delay to be maximum. Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# turnaround delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Send 2 messages from the UDP app with no delay between them.
3. Verify that the first message is received on the serial app immediately, and the second one after the turnaround delay.
4. Run “show serial interface serial 0/9” and verify the turnaround delay .

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Transport : UDP
```

```
Max connections      : 64
```

```
Remote connections   : 1
```

```
Remote Ip 1          : 192.168.0.100
```

```
Remote Port 1        : 50961
```

```
Remote Udp 1 RX cnt  : 12
```

```

Remote Udp 1 TX cnt      : 0
Remote Packets Dropped  : 0
Remote Bytes Dropped    : 0
Local Port               : 15010
Serial RX byte counter   : 0
Serial TX byte counter   : 12
Pack size                : 1400
Pack timer(msec)        : 0
Pack char (hex)         : d
Turnaround delay(msec)  : 1000
Hold time(msec)         : 0
Rx-to-Tx delay(msec)    : 0

```

10.16. Configuring Hold Time Delay

1. Unmap the profile and change the hold time to be 100. Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# hold-time 100
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Send 2 messages from the UDP app with no delay between them.
3. Verify that the first message is received on the serial app immediately, and the second after the turnaround delay.

4. Run “show serial interface serial 0/9” and verify the hold time and turnaround delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 64
Remote connections   : 1
Remote Ip 1          : 192.168.0.100
Remote Port 1        : 50961
Remote Udp 1 RX cnt  : 12
Remote Udp 1 TX cnt  : 0
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Port           : 15010
Serial RX byte counter : 0
Serial TX byte counter : 12
Pack size            : 1400
Pack timer(msec)     : 0
Pack char (hex)      : d
Turnaround delay(msec) : 1000
Hold time(msec)      : 100
Rx-to-Tx delay(msec) : 0
```

10.17. Configuring No Delays

1. Unmap the profile and change the change the hold time and the turnaround delay to be 0. Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# hold-time 0
Phoenix_Contact(config-serial-if)# turnaround delay 0
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Run “show serial interface serial 0/9” and verify the hold time, turnaround delay, and rx-to-tx delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
Profile : PROF_1_UDP
Serial interface : 9
Protocol : RAW Socket
Serial interface : 9
Transport : UDP
Max connections      : 64
Remote connections   : 1
Remote Ip 1          : 192.168.0.100
Remote Port 1        : 50961
Remote Udp 1 RX cnt  : 0
Remote Udp 1 TX cnt  : 0
Remote Packets Dropped : 0
Remote Bytes Dropped : 0
Local Port           : 15010
Serial RX byte counter : 0
Serial TX byte counter : 0
Pack size            : 1400
Pack timer(msec)     : 0
Pack char (hex)      : d
Turnaround delay(msec) : 0
Hold time(msec)      : 0
Rx-to-Tx delay(msec) : 0
```

3. Send 1 message from the serial and 1 message from the UDP app with minimal delay between them.
4. Verify that the messages arrived immediately.

10.18. Configuring RX-TO-TX Delay

1. Unmap the profile and change the RX-to-TX to be the maximum (1000). Map the profile.

FOR EXAMPLE: Type the following:

- Enter the Global Configuration Mode.

```
Phoenix_Contact# configure terminal
```

- Configure profile PROF_1_UDP profile.

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# no connection-map
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# interface serial 0/9
```

```
Phoenix_Contact(config-serial-if)# shutdown
```

```
Phoenix_Contact(config-serial-if)# rx-to-tx delay 1000
```

```
Phoenix_Contact(config-serial-if)# no shutdown
```

```
Phoenix_Contact(config-serial-if)# end
```

```
Phoenix_Contact# configure terminal
```

```
Phoenix_Contact(config)# serial connection-type raw profile PROF_1_UDP
```

```
Phoenix_Contact(raw-PROF_1_UDP)# connection-map interface serial 0/9
```

```
Phoenix_Contact(raw-PROF_1_UDP)# end
```

2. Run “show serial interface serial 0/9” and verify the hold time, turnaround delay, and rx-to-tx delay.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# show serial profile interface serial 0/9
```

```
Profile : PROF_1_UDP
```

```
Serial interface : 9
```

```
Protocol : RAW Socket
```

```
Serial interface : 9
```

```
Transport : UDP
```

```
Max connections      : 64
```

```
Remote connections   : 1
```

```
Remote Ip 1          : 192.168.0.100
```

```
Remote Port 1        : 50961
```

```
Remote Udp 1 RX cnt    : 12
Remote Udp 1 TX cnt    : 0
Remote Packets Dropped : 0
Remote Bytes Dropped   : 0
Local Port              : 15010
Serial RX byte counter : 0
Serial TX byte counter : 12
Pack size               : 1400
Pack timer(msec)       : 0
Pack char (hex)        : d
Turnaround delay(msec) : 0
Hold time(msec)        : 0
Rx-to-Tx delay(msec)   : 1000
```

3. Send 1 message from the serial and 1 message from the UDP app with minimal delay between them.
4. Verify that message from serial to the UDP app is received immediately, but from the UDP app to serial after the Rx-to-Tx delay.

10.19. Clearing the Serial Profile Configuration and Deleting the Profile

1. Delete the profile and set Rx-to-Tx delay to 0.

FOR EXAMPLE: Type the following:

```
Phoenix_Contact# configure terminal
Phoenix_Contact(config)# no serial profile PROF_1_UDP
Phoenix_Contact(raw-PROF_9_I)# end
```

```
Phoenix_Contact# configure terminal
Phoenix_Contact(configure)# interface serial 0/9
Phoenix_Contact(config-serial-if)# shutdown
Phoenix_Contact(config-serial-if)# rx-to-tx delay 0
Phoenix_Contact(config-serial-if)# no shutdown
Phoenix_Contact(config-serial-if)# end
```

2. Run “show serial profile interface serial 0/9” to verify that the profile is not mapped.