ICC

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

Modbus RTU Firewall Router

Driver Manual

April 15, 2020

TABLE OF CONTENTS

1 Modbus RTU Firewall Router

1.1 Overview

This driver supports selective routing of Modbus RTU packets between the device’s serial ports, using a set of rules to accept or reject certain packets. Some notes of interest are:

- Deep packet inspection allows matching by device addresses, function codes, register/coil/discrete input numbers, and values.
- All Modbus function codes are supported, including standard function codes (1 - 8, 11 - 12, 15 - 17, 20 - 24, 43) and custom or vendor-specific function codes.
- All Modbus device addresses are supported, including broadcasts.
- Configurable default policy. Select between an “Accept All Policy” and add “Reject Rules” or a “Reject All Policy” and add “Accept Rules”.
- Automatically blocks invalid Modbus packets that do not conform to the Modbus standard.
- Supports generating exception responses for invalid and rejected request packets.
- Diagnostics are available for each rule, allowing insight into which rules are evaluated for each received packet, why a rule did not match, and which rule, if any, matched and caused the received packet to be accepted or rejected.
- Independent baud rate, parity, and transmit delay settings for each port allow routing of Modbus packets between otherwise incompatible networks.
- Separate instances of the driver run on each serial port and use the device’s internal database to exchange the packet length and packet data.
- When the driver starts on the network port, it automatically stops any protocol driver which may have been running on the host port.
- While the driver is running, all other device functionality is disabled, excluding USB communications.

1.2 Definitions

The term “registers” is used to collectively refer to holding registers and input registers.

The term “discretes” is used to collectively refer to coils and discrete inputs.

1.3 Firewall Router Settings

1.3.1 Master Network Settings

These settings apply to the network port or Port A of the device. The Modbus master should be connected to this port.

Baud Rate

Selects the baud rate of the network.

Parity

Selects the parity and number of stop bits.

Transmit Delay

Defines the time in milliseconds that the driver will wait before transmitting each packet. This is a useful feature for certain master devices or infrastructure components (such as radio modems) that may require a given amount of time to place themselves into a “receiving mode” where they are capable of listening for slave responses. If no delay is required, setting this field to 0 instructs the driver to transmit packets as soon as possible.

1.3.2 Slave Network Settings

These settings apply to the host port or Port B of the device. The Modbus slave(s) should be connected to this port.

Baud Rate

Selects the baud rate of the network.

Parity

Selects the parity and number of stop bits.

Transmit Delay

Defines the time in milliseconds that the driver will wait before transmitting each packet. This is a useful feature for certain master devices or infrastructure components (such as radio modems) that may require a given amount of time to place themselves into a “receiving mode” where they are capable of listening for slave responses. If no delay is required, setting this field to 0 instructs the driver to transmit packets as soon as possible.

1.3.3 Rejected Request Packet Options

Generate Exception Response

When enabled, the driver will respond to invalid and rejected request packets with a Modbus exception response. The exception response generated by the driver for rejected packets depends on the selected policy. When the “Accept All Policy” is used, the driver will respond with an exception code corresponding to the rejected reason from the matched “Reject Rule”. When the “Reject All Policy” is used, the driver will respond with exception code *0x0A - Gateway Path Unavailable* when a packet does not match any defined “Accept Rules”.

Additionally, because the baud rate and transmit delay may be configured independently for each port, it is possible that a new request packet can be received on one port before the other port has finished transmitting the last packet. If this occurs, the driver will respond with exception code *0x06 - Slave Device Busy*.

1.3.4 Master Network to Slave Network Data

These settings define the locations in the device's database that store the packet length and packet data for each accepted packet received from the master network and transmitted to the slave network.

Note that these settings are fixed, and are therefore provided for reference only.

Packet Length Database Address

The database location that stores the 16-bit length of each accepted packet received on the master network port. A non-zero value in this location triggers the driver running on the slave network port to transmit the packet and reset this value to 0, indicating the packet has been transmitted.

Packet Data Database Address

The starting address of the database location that stores the data for each accepted packet received on the master network. The driver will allocate 256 bytes for the packet, which is the maximum size of a Modbus RTU packet.

1.3.5 Slave Network to Master Network Data

These settings define the locations in the device's database that store the packet length and packet data for each accepted packet received from the slave network and transmitted to the master network.

Note that these settings are fixed, and are therefore provided for reference only.

Packet Length Database Address

The database location that stores the 16-bit length of each accepted packet received on the slave network port. A non-zero value in this location triggers the driver running on the master network port to transmit the packet and reset this value to 0, indicating the packet has been transmitted.

Packet Data Database Address

The starting address of the database location that stores the data for each accepted packet received on the slave network. The driver will allocate 256 bytes for the packet, which is the maximum size of a Modbus RTU packet.

1.4 Policies and Rules

To configure the firewall router, first a policy must be selected that defines the default action taken if a received packet does not match any defined rules. Then, rules are added, each of which may define a set of criteria to check a received packet against. The driver supports two, mutually exclusive, methods of evaluating received packets. An "Accept All Policy" may be used in conjunction with "Reject Rules" or a "Reject All Policy" may be used in conjunction with

"Accept Rules". Note that this configuration applies only to the driver running on the master network port. The driver on the slave network port always uses the "Accept All Policy" with no "Reject Rules" defined.

Rules are evaluated in the order in which they are defined. If the received packet matches a rule, the packet is immediately accepted or rejected (depending on the type of rule), and no further rules are evaluated. If a rule has multiple criteria selected, the received packet must meet all of the selected criteria in the rule. Therefore, if a received packet does not contain fields corresponding to one or more criteria in a rule, the rule will not match and the driver will move on to the next defined rule. Note that this does not apply to the device address and function code criteria, as all Modbus packets must include these fields.

1.4.1 Rule Settings

Description

This 32-character (max) field is strictly for user reference: it is not used at any time by the driver.

1.4.1.1 Device Address Range

These settings allow matching the received packet's device address field against a defined address range.

Include in Rule Criteria

Check this option to include device address range matching in this rule's criteria.

Address Low

Enter the lowest address in the range of device addresses to match. Allowable values are in the range 0 - 247.

Address High

Enter the highest address in the range of device addresses to match. Allowable values are in the range 0 - 247.

1.4.1.2 Function Code Range

These settings allow matching the received packet's function code field against a defined range of function codes.

Include in Rule Criteria

Check this option to include function code range matching in this rule's criteria.

Function Code Low

Select the lowest function code in the range of function codes to match. Available function codes include 1 - 8, 11 - 12, 15 - 17, 20 - 24, and 43.

Function Code Low Value

This field is enabled only when the "Function Code Low" selection is set to "Other". Enter a custom function code value (0 - 255) for the lowest function code in the range of function codes to match.

Function Code High

Select the highest function code in the range of function codes to match. Available function codes include 1 - 8, 11 - 12, 15 - 17, 20 - 24, and 43.

Function Code High Value

This field is enabled only when the "Function Code High" selection is set to "Other". Enter a custom function code value (0 - 255) for the highest function code in the range of function codes to match.

1.4.1.3 Register/Discrete Range

These settings allow matching the registers or discretes targeted by the received packet against a defined range of register/discrete numbers.

Include in Rule Criteria

Check this option to include register/discrete number range matching in this rule's criteria.

Register/Discrete Low

Enter the lowest register/discrete number in the range of registers/discretes to match. Allowable values are in the range 1 - 65536.

Register/Discrete High

Enter the highest register/discrete number in the range of registers/discretes to match. Allowable values are in the range 1 - 65536.

1.4.1.4 Value Range

These settings allow matching the values of registers or discretes contained in the received packet.

Include in Rule Criteria

Check this option to include value range matching in this rule's criteria.

Value Low

Enter the lowest value in the range of register/discrete values to match. Allowable values are in the range 0 - 65535.

Value High

Enter the highest value in the range of register/discrete values to match. Allowable values are in the range 0 - 65535.

1.5 Diagnostics Object

Each rule can optionally include a diagnostics object for debugging and diagnostics.

Diagnostics Database Address

Enter the database address at which to store the diagnostics information.

1.5.1 Definition of Diagnostics Counters

TX Counter

Increments when the rule is evaluated.

RX Counter

Increments when the rule matches the received packet and causes the packet to be accepted (for “Accept Rules”) or rejected (for “Reject Rules”). The “Current Status” is set to “No Error”.

RX Error Counter

Increments when the rule does not match the received packet. The “Current Status” and “Last Error” are updated to show the reason the rule did not match the packet.

1.6 Timeout Considerations

When using a firewall device between Modbus RTU devices, it is possible that certain requests from the master may time out, although they succeed when the devices are directly connected together. This can happen if the master device’s timeout time is set to a low value and the request or response packet is large.

As with any firewall device that performs deep packet inspection, the driver uses a “store and forward” approach, whereby, the following steps occur.

1. The packet is received in its entirety by one of the device’s serial ports.
2. The packet is validated and each field is inspected and compared against the defined rules.
3. If the packet is accepted, it is then transmitted by the device’s other serial port.

Because of this, every request and response packet takes twice as long to traverse the network, since it must be encoded onto the network twice - first by the originating device and again by the firewall device. Due to the size of certain Modbus RTU packets, such as reading or writing a large number of consecutive registers, and the baud rate used, the time it takes for a packet to be encoded onto the RS-485 bus may be significant.

For example the response packet for reading 100 registers is 205 bytes in length. At a baud rate of 9600, it takes a minimum of 213ms for the packet to be transmitted onto the network. Therefore, when traversing through a firewall device, it would take at least double this amount of time, or 426ms. If the master device’s timeout time was set to a value of 300ms, this would cause the master’s request to timeout when the firewall device is used but communicate successfully without the firewall device.

1.7 “Generate Exception Response” Precautions

The Modbus protocol has no indication within a packet of whether the packet is a request or a response. Additionally, the Modbus RTU Firewall Router driver is stateless, meaning it evaluates each received packet without retaining any knowledge of previously received packets. Therefore, there are certain conditions where it is impossible for the driver to differentiate between a request and a response packet. If the “Generate Exception Response” option is enabled, the driver may send an exception response after rejecting a response packet. Table 1 below lists the specific Modbus packets that are impacted. If one or more of the following conditions can exists on your network, do not enable the “Generate Exception Response” option or create rules to ensure these request and response packets are not rejected.

Table 1: Impacted Modbus Packets

Function Code(s)	Condition
05 (0x05) Write Single Coil 06 (0x06) Write Single Register 21 (0x15) Write File Record 22 (0x16) Mask Write Register	Requests and responses are identical.
08 (0x08) Diagnostics	Requests and responses are identical for some sub-functions.
01 (0x01) Read Coils 02 (0x02) Read Discrete Inputs	Requests targeting discretes 768 - 1023 may be identical to responses reading 17 to 24 discretes.
23 (0x17) Read/Write Multiple Registers	Responses with a value of [PACKET_LENGTH - 13] in the 4th register data low byte are identical to requests.
43 (0x2B) Encapsulated Interface Transport	Requests and responses are not differentiated for MEI Types other than *14 - Read Device Identification*.

ICC

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

230 Horizon Drive, Suite 100
Verona, WI USA 53593
Tel: [608] 831-1255   Fax: [608] 831-2045

http://www.iccdesigns.com

Printed in U.S.A