



JetBox series Modbus Gateway

User Manual

www.korenixembedded.com

Copyright Notice

Copyright© 2012 Korenix Technology Co., Ltd.

All rights reserved.

Reproduction without permission is prohibited.

Information provided in this manual is intended to be accurate and reliable. However, the original manufacturer assumes no responsibility for its use, or for any infringements upon the rights of third parties that may result from its use.

The material in this document is for product information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, Korenix assumes no liabilities resulting from errors or omissions in this document, or from the use of the information contained herein.

Korenix reserves the right to make changes in the product design without notice to its users.

Acknowledgments

Korenix is a registered trademark of Korenix Technology Co., Ltd.

All other trademarks or registered marks in the manual belong to their respective manufacturers.

1 Overview

The Modbus Gateway is optional value-added software provided by Korenix. The major function of the Modbus Gateway enables serial Modbus RTU (or Modbus ASCII) devices to communicate with Modbus TCP devices.

Modbus is an open serial communications protocol based on master/slave architecture and used to connect a supervisory computer with a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) system. However, Modbus has been extended to operate over Ethernet using the IP protocol suite. Therefore, the Modbus Gateway converts between the Modbus TCP/IP protocol and Modbus ASCII/RTU protocols transparently and let users integrate their control systems easier.

Modbus Gateway upgrades to v2.x

Different from version 1.x, which supports only one TCP to one COM control, we have a great improvement that version 2.x now supports Modbus multi TCP to multi COM control for providing a better solution of devices control and data acquisition. Modbus GW v2.x is compatible with previous version; besides, JetBox with v2.x can communicate with JetBox with v1.x in the same gateway for providing convenience to users.

The difference of settings between v1.x and v2.x is primary in Config profile: two new columns [mapping], [port_parameter] are added in v2.x. If the column [mapping] is left empty, JetBox would support one COM as previous version; otherwise, it would support multi COM when the column is filled correctly. [port_parameter] is to set parameters on each port, and if the column is empty, all parameters will follow default settings in [modbusgw].

2 Applied Korenix Model

The Modbus Gateway v2.x is applied in the following models:

1. JetBox 9500 series: Industrial Network Embedded System
2. JetBox 9400 series: Industrial Embedded VPN Router Computer
3. JetBox 9300 series: Industrial Communication Embedded System
4. JetBox 5300 : Industrial Linux Communication Computer
5. JetBox 3300 series: Industrial Linux Computer

Notice :

1. Please refer to the user manual of JetBox auto-run function to execute the Modbus Gateway automatically.
2. Please refer to official website for model-corresponding SW release.

3 How to use the Modbus Gateway?

3.1 How does it work?

- **Accept Connection**

When you start to execute the Modbus Gateway program, it would automatically run the Modbus TCP server to receive connection from outer Modbus TCP client.

- **Access Data from TCP**

Once the connection is built, the program would start to access the data of TCP socket by orders. If the content is "0"; which means the TCP connection is corrupted, the program will check the packet format. In case the format is not following Modbus TCP protocol, the program would interrupt this connection and access next packet.

- **Data Convert**

When the data is accessed correctly, the PDU data will be captured from Modbus TCP. Then the data would be transformed from Modbus TCP to Modbus RTU/ASCII protocol format according to function code and unit identifier of Modbus RTU/ASCII.

- **Send Request and Wait for Response**

After the data format is transformed, the data will be send to Modbus RTU/ASCII Slave server and wait for response. Besides, once data access from serial port, the time-out mechanism will be activated and waiting-for-response time is set to 5 second. If the response time exceeds the limit, the system will consider the request as failed and process the next data.

- **Write Response back to TCP**

If the Modbus RTU/ASCII Slave server checks there's no error happened about the data, the data will be transformed back to Modbus TCP format. Then the server will determine which TCP master the request is from and write response back.

3.2 Usage

The followings are parameters settings and explanations of this program.

Name :

modbusgw: a mechanism of data transforming between Modbus TCP and Modbus RTU/ASCII.

Synopsis :

```
modbusgw -port <port> [-f <file_name>] [-protocol {rtu | ascii}] [-baud <baud>]
[-parity {none | even | odd}] [-bits {7 | 8}] [-stop {1 | 2}] [-timeout <t>] [-srate <t>]
[-tcp <port>] [-tcpaging <t>]
```

Description :

modbusgw is mechanism of data transforming between Modbus TCP and Modbus RTU/ASCII. We use the mechanism to transform the data between two protocols, and it can help providing unlimited communication and data exchange through different-hardware Modbus devices.

Options :

-f <file_name>

Name of specified profile. **Default : ./modbusgw.cfg**

Once '**f**' is typed, the default profile will be brought in automatically.

```
# cat modbusgw.cfg
[modbusgw]
port=ttyS1
protocol=rtu
baud=115200
parity=none
bits=8
stop=1
timeout=5
srate=100
tcp=502
tcpaging=420
[mapping]
port1_start=1
port1_end=50
port2_start=51
port2_end=103
[port_parameter]
port1_protocol=rtu
port1_baud=115200
port1_parity=none
port1_bits=8
port1_stop=1
port2_protocol=rtu
port2_baud=9600
port2_parity=none
port2_bits=8
port2_stop=1
port4_protocol=ascii
port4_baud=38400
port4_parity=none
port4_bits=7
port4_stop=1
```

[modbusgw]

Default column. (the column is left unchanged to be compatible with previous version)

[mapping]

Set the range of slave ID. (If the column is left empty, Modbus support one COM as previous version.)

Ex: Set port1_start=1, port1_end=50. In this example, if the slave ID of data is located between 1 to 50, the data will be sent to port 1.

[port_parameter]

Set parameters on each port.

NOTE: Except shared values such as tcp 、srate 、timeout 、tcpaging. These shared values are setted in [modbusgw]. If user does not set this column, all parameters will follow default settings in [modbusgw]

-port <port>

Device name for serial port.

Ex : the default values of ttyS1 、ttyS2 are set to ttyS1.

-protocol {rtu | ascii}

Set Modbus protocol. **Default : RTU**

-baud <baud>

Set data transfer rate.

User can set transfer rate listed in the following table:

{50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800} **Default: 9600**

-parity {none | even | odd}

Set parity check parameters to avoid errors during data transferring. **Default: none**

-bits {7 | 8}

Set the length of each data. **Default: 8 bits**

-stop {1 | 2}

Set the length of stop bit. **Default: 1**

-timeout <t> (second)

Set the wait-for-respond-time of data transferring from Modbus TCP to Modbus RTU/ASCII through Modbus Gateway. Range from 0~600 seconds. **Default: 5 seconds**

NOTE: There's also response-time-out setting in HMI or simulator. Please check the settings in both side are consistent or the value set in simulator must be larger than program to avoid errors happened during transferring.

-srate <t> (millisecond)

Set the scan rate from 0~10000ms. **Default: 200ms**

NOTE: Please adjust the parameter properly to leave delay time for data transferring. It takes time to respond for Modbus RTU/ASCII server when there're multi connections at one time, so time-out would happen if TCP server keeps sending data to RTU/ASCII.

-tcp <port>

Set the port numbers of Modbus TCP server from 1~65535. **Default: 502**

-tcpaging <t>

The system will automatically interrupt the connection to avoid occupying channel if the TCP connection is failed or idling abnormally. Range from 1~7200 seconds. **Default: 420s**

Example :

```
modbusgw -port ttyS1 -protocol rtu -baud 115200 -parity even -bits 8 -stop 1  
-timeout 5 -srate 200 -tcp 502 -tcpaging 100  
or  
modbusgw -f ./modbusgw.cfg
```

Notice:

1. All parameter settings shall be written in lower case alphabet. (Ex: 'a')
2. Slave ID should be unique and can't be used repeatedly.
3. Parameters must be set and applied correctly when multi-port function is activated. Command mode is only suitable for single port connection.

3.3 Specification

LAN	
Ethernet	10/100 Mbps, RJ45
Serial Interface	

Interface	RS-232/422/485, software selectable
No. of Port	4
Serial Communication Parameters	
Parity	None · Odd · Even
Data bits	7 · 8
Stop bits	1 · 2
Baud Rate Speed	50 · 75 · 110 · 134 · 150 · 300 · 600 · 1200 · 1800 · 2400 · 4800 · 9600 · 19200 · 38400 · 57600 · 115200 · 230400 · 460800
Software Features	
Operation Modes	TCP Slave · RTU Master · ASCII Master
Multi TCP Master to Multi RTU/ASCII Slave	Up to 32 Connections