

CoBox Modbus

Universal Thin Servers

Installation Guide

Contents

1 INTRODUCTION

EXTENDED MODBUS SYSTEM EXAMPLE	1-2
Modbus/TCP Master Talking to Modbus/TCP Slave	1-3
Modbus/TCP Master Talking to Modbus/RTU Serial Slave	1-3
Modbus/RTU Serial Master Talking to Modbus/TCP Slave	1-4
Modbus/RTU Serial Master Talking to Modbus/RTU Serial Slave	1-4

2 COBOX MODBUS INTERFACES

CoBox MODBUS SERIAL INTERFACE	2-1
NETWORK INTERFACE	2-1
NETWORK HARDWARE ADDRESS	2-1
CoBox Modbus with Ethernet	2-2
CoBox Modbus with Token Ring	2-2
POWER SUPPLY	2-2
GROUNDING	2-2

3 NETWORK PROTOCOLS

PACKING ALGORITHM	3-1
IP ADDRESS AND MODBUS SLAVE ADDRESS	3-1

4 CONFIGURATION

SERIAL CONFIGURATION	4-1
NETWORK CONFIGURATION	4-2
CoBox's IP ADDRESS	4-2
Default IP Address	4-2
Assigning a New IP Address	4-3
CONFIGURATION PARAMETERS	4-3
BASIC COMMANDS (D/S/Q)	4-4
Default Settings (D)	4-4
Save (S)	4-4
Quit Without Saving (Q)	4-5
NETWORK/IP SETTINGS	4-5
IP Address	4-5
Set Gateway IP Address (Y/N)	4-5
Set Netmask (N for default)	4-5
SERIAL & MODE SETTINGS	4-5
Attached Device (1=Slave, 2=Master)	4-6
Serial Protocol (1=Modbus/RTU, 2=Modbus/ASCII)	4-6
Use Serial Connector (1=DB25, 2=DB9)	4-6
Interface Type (1=RS-232, 2=RS-422, 3=RS-485)	4-6
Enter Serial Parameters (BBBB,D,P,S)	4-6

MODEM CONTROL SETTINGS.....	4-6
DCD Mode (1=Fixed 2=Variable)	4-7
RTS/CTS Mode (1=Fixed 2=Variable)	4-7
Delay after Output of RTS (0-1275 msec, 5ms resolution)	4-7
Wait for CTS to Go Active (N/Y).....	4-7
Delay after CTS Going Active (0-1275 msec, 5ms resolution)	4-7
Delay Dropping RTS after Transmitting (0-1275 msec, 5ms resolution)....	4-7
ADVANCED MODBUS PROTOCOL SETTINGS	4-7
Slave Address (0 for auto, or 1..255 fixed otherwise)	4-8
Allow Modbus Broadcasts (1=Yes 2=No).....	4-8
Character Timeout (10-1275 msec, 5ms resolution)	4-8
Message Timeout (500-60000 msec, 250ms resolution)	4-8
UNIT ID TO IP ADDRESS LOOKUP TABLE	4-9
Select Function A)dd, D)lete, E)xit	4-9
Modbus Address From/To.....	4-9
Slave IP Address.....	4-9

5 MONITOR MODE AND FIRMWARE UPGRADE

MONITOR COMMANDS	5-1
FIRMWARE DOWNLOAD USING SERIAL PORT	5-2
FIRMWARE DISTRIBUTION.....	5-3
FIRMWARE DOWNLOAD USING A NETWORK HOST	5-4

6 CONFIGURING THIRD PARTY SOFTWARE FOR COXBUS MODBUS

MODSOFT v2.6 OR LATER	6-1
WIN-TECH MODSCAN32.....	6-1
IO-SERVER OPC SERVER.....	6-1
INTELLUTION FIX HMI	6-2

7 CONNECTORS' PINOUTS

CHANNEL 1 (RS-232C).....	7-1
CHANNEL 1 (RS-422/485)	7-3
CHANNEL 2 (RS-232C).....	7-4

8 LED STATUS DISPLAY

YELLOW AND GREEN LED.....	8-1
RED LED	8-1

9 TECHNICAL DATA

CPU, MEMORY CONTROLLERS	9-1
SERIAL INTERFACE	9-1
NETWORK INTERFACE.....	9-1
POWER SUPPLY	9-1
POWER CONSUMPTION.....	9-1
LEDs.....	9-1
CASE.....	9-1
DIMENSIONS.....	9-2
WEIGHT	9-2

APPENDIX A TOKEN RING SPECIFIC

APPENDIX B IP ADDRESSES, NETMASK ETC.

APPENDIX C BINARY TO HEX CONVERSION

WARRANTY AND DECLARATIONS

INDEX

1 Introduction

When it comes to planning data communication for open, multi-vendor industrial control systems, Modbus™ is the undisputed first choice of end users and integrators alike. The Modbus/RTU protocol defines how a “master” device polls one or more “slave” devices to read and write data in real-time over RS-232, RS-422, or RS-485 serial data communication. Although it’s not the most powerful protocol available, its rare simplicity allows not only rapid implementation, but remains flexible enough to be applied in virtually all industrial situations. Modbus/TCP is a new extension of Modbus/RTU, which defines how Modbus/RTU messages are encoded within and transported over TCP/IP-based networks. Modbus/TCP is just as simple to implement and flexible to apply as the original Modbus/RTU. You can find the specification for both online at www.modicon.com.

The CoBox Modbus is a new product from Lantronix that allows users to integrate new and existing Modbus/RTU and Modbus/ASCII serial devices to newer TCP/IP network-based devices. Section 1.1 describes a system that integrates 4 Modbus/RTU devices with 4 Modbus/TCP devices.

™ Modbus is a registered trademark of Schneider Automation.

1.1 Extended Modbus System Example

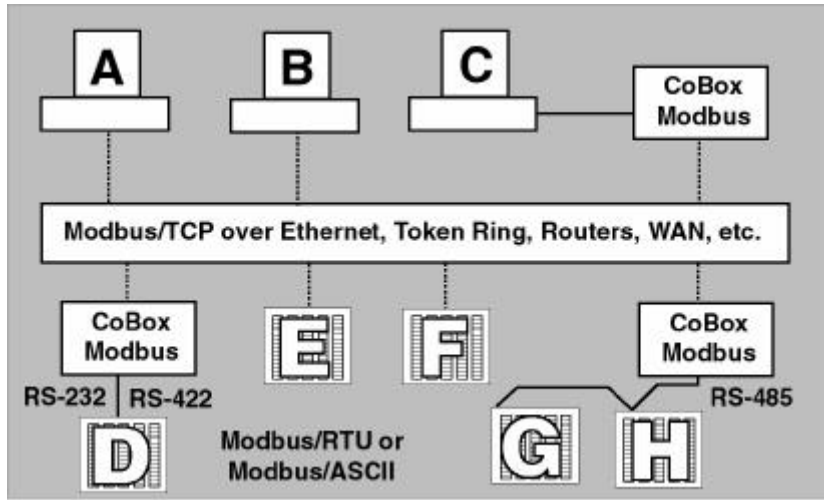


Figure 1-1

In figure 1-1, we can see 4 specific styles of Modbus operations. Modbus/RTU devices are traditionally split into 2 groups.

Modbus slave devices generally are the workhorse devices. Often industrially hardened, they tirelessly perform their tasks 24 hours a day, 365 days a year. Tasks such as flow metering, temperature control, batch loading, or even running entire automated assembly lines. The slave devices are not called “slaves” because they work all the time, they are called slaves because as far as the data communications is concerned, they function as passive servers. Modbus slave devices passively sit and wait for a remote Modbus master device to ask them to report existing data values (Read) or accept new data values (Write).

Modbus master devices generally are higher-level computers, devices in which data and software are very important. The most common example of Modbus master devices is the “Human-Machine-Interface” (HMI) computers which allow human operators to monitor, adjust, and maintain the operations of the field devices. Modbus master devices are clients which actively go out and “Read” from and/or “Write” to remote Modbus slave devices to monitor or adjust slave behavior.

1.1.1 Modbus/TCP Master Talking to Modbus/TCP Slave.

Devices A, B, E, and F are all new Modbus/TCP devices, which are improved over Modbus/RTU (see more about Modbus/RTU limitations below). All 4 devices can function concurrently as both Modbus master and Modbus slave. Both computers A and B can treat controller E as a slave, polling data in real-time. Yet controller E can also act as a master and poll data from controller F, which can in turn also act as a master to write alarm data directly up to computers A and B to alert the operators to the alarm condition. Traditional Modbus/RTU requires slave devices even with life threatening alarm conditions to sit patiently and wait for a remote master to poll the specific data that caused the alarm condition.

While this new functionality of Modbus/TCP may not impress computer experts, it is really revolutionary for such a simple and flexible protocol as Modbus to offer such functionality. Therefore, Modbus/TCP offers exciting new design options for industrial users, which the Lantronix CoBox Modbus extends to traditional Modbus/RTU serial devices.

1.1.2 Modbus/TCP Master Talking to Modbus/RTU Serial Slave.

Devices D, G, and H are traditional Modbus/RTU slave devices. Device D uses a point-to-point electrical interface like RS-232. This allows only a single Modbus/RTU master to talk to device D. However, the CoBox Modbus makes device D appear on the Modbus/TCP network as a full Modbus/TCP slave device. All Modbus/TCP enabled devices A, B, E, and F can actively share access to slave device D. A limitation in traditional Modbus/RTU implementation expects devices to be dedicated as either master or slave devices, so device D can only act as a Modbus slave.

Devices G and H are different from device D. They share a single RS-485 “multi-drop” line which strictly limits them to act as slaves to a single Modbus/RTU master. However, a little of the new Modbus/TCP and CoBox Modbus magic still applies—all Modbus/TCP enabled devices A, B, E, and F can actively share access to both slave devices G and H. CoBox Modbus manages and coordinates the shared access. In fact, the CoBox Modbus allows up to 8 concurrent Modbus masters to share access to the slaves.

1.1.3 Modbus/RTU Serial Master Talking to Modbus/TCP Slave.

Device C is a traditional Modbus/RTU master device. Yet the CoBox Modbus makes device C appear to the TCP/IP network as a Modbus/TCP master—plus all of the Modbus/TCP slaves on the TCP/IP network (A, B, D, E, F, G, & H) appear as traditional Modbus/RTU slave devices. The only limitation is the traditional Modbus/RTU assumption that device C is dedicated as a master only. Therefore Modbus/TCP master devices A, B, E, and F cannot treat device C as a Modbus/TCP slave.

1.1.4 Modbus/RTU Serial Master Talking to Modbus/RTU Serial Slave.

Finally, master device C can poll traditional Modbus/RTU slave devices D, G, and H as if they were directly multi-dropped on an attached RS-485 line. The CoBox Modbus transparently bridges traditional Modbus/RTU devices across any TCP/IP network. This means users can start implementing for Modbus/TCP long before all of their required products exist with Modbus/TCP and network interfaces.

2 CoBox Modbus Interfaces

Depending on the model, the CoBox Modbus supports different electrical device interfaces with different network topologies.

2.1 CoBox Modbus Serial Interface

Channel 1 (CH1) is a DB25 female connector, software selectable for RS-232 (with full modem control RTS/CTS handshaking), RS-422, or RS-485 both 2 and 4-wire support. Channel 2 (CH2) is a DB9 male connector, supporting only RS-232 (with full modem control RTS/CTS handshaking). Both CH1 and CH2 support baud rates 300, 1200, 2400, 4800, 9600, 19,200, 38,400 and 115k.

NOTE: *CoBox Modbus is a one-port device, meaning that only one port (channel) can be used at a time (software selectable).*

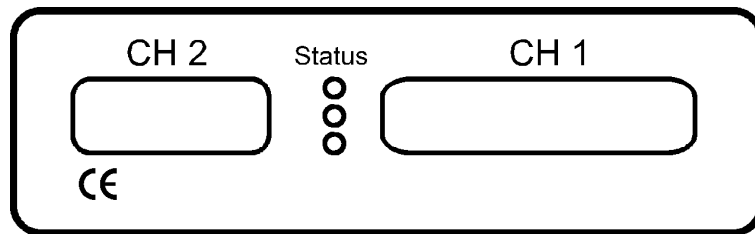


Figure 2-1

2.2 Network Interface

Ethernet models support 10Mbit, Token Ring models support 4 and 16 Mbit switch or jumper selectable.

2.3 Network Hardware Address

The first 3 bytes are fixed, and read 00-20-4A. The fourth, fifth, and sixth bytes represents the type of the product and the serial number in Hex notation.

2.3.1 CoBox Modbus with Ethernet

10BaseT (RJ-45 connector) and AUI (DB15 connector) for external transceiver connections.

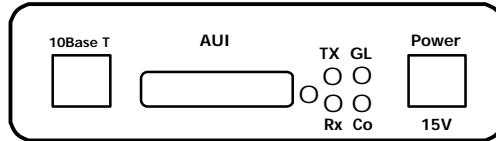


Figure 2-2

2.3.2 CoBox Modbus with Token Ring

STP (DB9 connector) and UTP (RJ45 connector) with ring speed of 4/16Mbit switch selectable located on the unit.

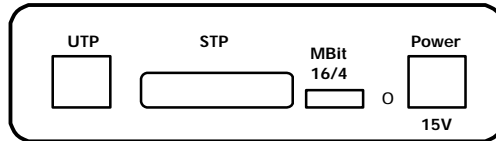


Figure 2-3

2.4 Power Supply

Depending on the model, the CoBox Modbus has different power supply needs. In Europe (230VAC) and USA (110VAC), the CoBox Modbus external models are shipped with an AC power adapter. Power consumption varies with connected transceivers, phantom current, etc. Typically, the Ethernet version consumes around 250mA with the 10BaseT interface activated, while the Token Ring version needs approximately 400mA at 15V (including the phantom drive).

2.5 Grounding

The CoBox Modbus Ethernet UTP interface, as required by the Ethernet standards, is 100% galvanically isolated.

3 Network Protocols

The CoBox Modbus uses TCP/IP protocols for network communications. The supported protocols are: ARP, UDP, TCP, ICMP, Telnet, TFTP, DHCP, and SNMP. Per the Modbus/TCP specification, the Modbus data messages are transported by TCP for error-free data transport end-to-end. Firmware updates can be done with the TFTP protocol. The IP protocol defines addressing, routing, and data block handling over the network.

3.1 Packing Algorithm

Traditional Modbus/RTU requires a “character time-out” to signal the end of a Modbus/RTU packet. This stretches out the overall response cycle. Fortunately, the CoBox Modbus uses an intelligent length-predictive algorithm to detect the end of standard Modbus messages. This allows better performance plus the CoBox Modbus falls back to using a user definable “character time-out” to manage non-standard or user-defined Modbus functions.

3.2 IP Address and Modbus Slave Address

Every device connected to the TCP/IP network including the CoBox Modbus must have a unique IP address. When multiple Modbus devices share a single IP, then Modbus/TCP includes an additional address called the Unit ID. See Appendix B for a complete description of IP Addressing.

When the CoBox Modbus is receiving Modbus/TCP messages from remote masters, the Unit ID is converted to use in the Modbus/RTU message as the slave address.

When the CoBox Modbus is receiving Modbus/RTU messages from local serial masters, a user-defined lookup table is used to match the 8-bit Modbus slave address to a remote IP address. The Modbus slave address received is used as the Unit ID.

4 Configuration

The CoBox Modbus can be configured by various remote or local methods. The simplest method is to use a terminal emulation program to locally access the first serial port (CH1), or network-savvy users can use a Telnet connection to configure the unit remotely over the network.

The CoBox Modbus configuration is stored in nonvolatile memory and is retained without AC power. The configuration can be changed any time. The CoBox Modbus performs a reset after the configuration has been changed and stored.

NOTE: *If you enter the configuration mode while the unit is operating, even if nothing is changed, this would disrupt your network operation.*

4.1 Serial Configuration

An ASCII terminal or PC with a terminal emulation application is connected to the first serial port (CH1) of the CoBox Modbus by a standard modem cable. The terminal (or PC) should be configured for 9600 Baud, no parity, 8 data bits, and 1 stop bit. A common terminal emulation program known to work is the HyperTerminal accessory included with Windows 95/98/NT.

To enter configuration mode, the power on the CoBox Modbus must be cycled (powered off and back on). After power-up, the self-test begins. About a half second later the red LED starts blinking. Now three lowercase “x” characters must be sent to the CoBox Modbus. These characters must all be sent within approximately 1 second to start configuration mode.

NOTE: *The easiest way to enter configuration mode is to hold down the “x” key at the terminal (emulation) and then powering the CoBox Modbus on. This ensures that the x characters arrive in time.*

4.2 Network Configuration

To configure over the network, a Telnet connection to port 9999 must be established. If you know the assigned IP address you can establish a Telnet connection to port 9999. Under Windows 95/98/NT, open an MS-DOS command window and type the command “telnet x.x.x.x 9999”, where x.x.x.x is an IP address already configured in the CoBox Modbus and 9999 is the desired TCP/IP port. Make sure you put a space between x.x.x.x and 9999.

4.3 CoBox’s IP Address

4.3.1 Default IP Address

The CoBox Modbus is shipped with a default IP address of 0.0.0.0, which automatically enables the DHCP within the CoBox Modbus.

With DHCP-enabled CoBox, If there is a DHCP server to respond to CoBox’s request when it’s booting up, the CoBox will then get an IP address, a gateway address, and a subnet mask from the DHCP server. These addresses will not be shown in the CoBox’s configuration screens (you will still see 0.0.0.0), however if you enter the “monitor mode” (see Section 5.1) and from 0> prompt type NC (upper case) you will be able to see the IP configuration of the CoBox Modbus.

NOTE: *Using the ARP command (see below) you can always override the IP address given to the CoBox Modbus by your DHCP server.*

With DHCP enabled CoBox, If there is no DHCP server on the network, then the CoBox’s request will eventually time out and the unit will boot up with no IP address. As soon as a static IP address is assigned to the CoBox, the DHCP support will be disabled within the product. To re-enable DHCP support, the IP address should be set back to 0.0.0.0.

4.3.2 Assigning a New IP Address

If the IP Address of the CoBox Modbus is unknown or undefined, the following steps outline how to assign a temporary IP address over the network.

- a) Set a static ARP with the desired IP address using the hardware address of the CoBox Modbus, which is printed on the product label. The following example shows the use of ARP in Win95/98/NT, (from the DOS prompt) and UNIX environments when the hardware address of the CoBox Modbus is 00-20-4A-01-64-0B.

NOTE: *In order for the ARP command to work in Windows, the ARP table on the PC must have at least 1 IP address defined other than its own. Type "ARP -A" at the DOS command prompt to verify that there is at least 1 entry in the ARP table. If there is no other entry beside the local machine, ping another IP machine on your network to build the ARP table. This has to be a host other than the machine on which you are working. Once there is at least 1 entry in the ARP table, use the following commands to ARP an IP address to the CoBox Modbus.*

arp -s 191.12.3.77 00-20-4A-01-64-0B

The command example for most Unix systems is:

arp -s 191.12.3.77 00:20:4A:01:64:0B

- b) Open a Telnet connection to port number 1. This connection will fail, but the CoBox Modbus will change its IP address to the one designated in the arp command line.

telnet 191.12.3.77 1

- c) Open a Telnet connection to port 9999 and set all required parameters.

telnet 191.12.3.77 9999

NOTE: *The temporary IP address by ARP is reverted after every power reset of the CoBox Modbus. Be sure to log into CoBox Modbus and store the parameters to make the changes permanent.*

4.4 Configuration Parameters

When you enter configuration mode you should see the following message.

***Lantronix, Inc. CoBox Modbus
Serial Number 150-087 Software Version V01.1 (990709)**

Press Enter to go into Setup Mode, wait to close

Press <ENTER> to change parameters, otherwise after a short delay the CoBox will automatically restart. You'll be happy to know that other than the IP address and serial port settings (baud rate, etc.) the CoBox Modbus is preconfigured to be suitable with most applications. For all settings you are shown the default values, if you just press the < ENTER > key, the existing values will stay unchanged.

```
Lantronix, Inc. CoBox Modbus
Serial Number 204-041 Software Version V01.1 (990709)

Press Enter to go into Setup Mode, wait to close

CoBox MB Setup, Lantronix, Inc.
  Resetting to factory defaults
1) Network/IP Settings:
  Ethernet Interface TPI
  IP Address ..... 194.039.078.254
  Default Gateway ..... --- not set ---
  Netmask ..... --- not set ---
2) Serial & Mode Settings:
  Protocol ..... Modbus/RTU,Slave(s) attached
  Serial Interface ..... 9600,8,E,1,RS232,DB25
3) Modem Control Settings:
  DCD Output ..... Fixed High/Active
  CTS Output ..... Fixed High/Active
4) Advanced Modbus Protocol settings:
  Slave Addr/Unit ID Source .. Modbus/TCP header
  Modbus Serial Broadcasts ... Disabled
  Character, Message Timeout.. 00050 ms,05000 ms

D)efault settings, S)ave, Q)uit without save
Select Command or parameter set (1..4) to change:
```

Figure 4-1

4.5 Basic Commands (D/S/Q)

The main CoBox Modbus configuration menu is shown in figure 4-1. The CoBox Modbus offers 3 basic options.

4.5.1 Default Settings (D)

Entering “D” resets all parameters to the factory default as shown above. Only the IP address is not changed.

4.5.2 Save (S)

Entering “S” saves the currently displayed parameter settings into non-volatile memory and exits configuration mode. It will trigger a reset.

4.5.3 Quit Without Saving (Q)

Entering “Q” throws away any parameter changes you have made and exits configuration mode. It will trigger a reset.

4.6 Network/IP Settings

To change the basic network parameters, press “1”. The following values can be set/changed.

4.6.1 IP Address

The IP address must be set to a unique value on your network. If you are not familiar with IP addressing on your network, please consult your system administrator. Please refer to Appendix B for more details about IP addresses.

If the CoBox Modbus is set to an address already in use, it will display an error code with the LEDs and it will not operate properly. If you understand and plan to use DHCP, set the IP to 0.0.0.0 to activate DHCP (see section 4.3).

4.6.2 Set Gateway IP Address (Y/N)

Most users could choose “N” for this case. You only need to choose “Y” if your CoBox Modbus must communicate to remote TCP/IP networks through a router or gateway. If you select “Y”, you must also enter the IP address of the default gateway within your local network. If in doubt, consult your network administrator.

4.6.3 Set Netmask (N for default)

Most users could select “N”, which causes the CoBox Modbus to automatically use the standard netmask appropriate for the IP address you’ve entered. Users who want a non-standard netmask need to enter the number of host bits in the IP address. Appendix B covers the calculation of non-standard netmasks in detail.

4.7 Serial & Mode Settings

To change the basic serial parameters, press “2”. The following values can be set/changed.

4.7.1 Attached Device (1=Slave, 2=Master)

As mentioned in the introduction, Modbus/RTU devices are defined as either slave or master devices. Enter “1” if the attached device is a slave (such as controller or PLC) or “2” if the attached device is a master (such as a computer running graphical human-machine-interface (HMI) software).

4.7.2 Serial Protocol (1=Modbus/RTU, 2=Modbus/ASCII)

Serial Modbus comes in two flavors. Modbus/RTU uses 8-bit data bytes to send binary information. However, some devices cannot handle 8-bit data bytes, so Modbus/ASCII is used, which is a slower protocol where each 8-bit data byte is converted to 2, 8-bit ASCII characters. Since CoBox Modbus converts both to/from Modbus/TCP fully, you can mix any combination of RTU and ASCII devices on a Modbus/TCP network.

4.7.3 Use Serial Connector (1=DB25 2=DB9)

The CoBox Modbus hardware includes 2 physical serial ports. It allows you to choose either CH1 (25-pin DCE port) or CH2 (9-pin DTE port).

4.7.4 Interface Type (1=RS-232, 2=RS-422, 3=RS-485)

This applies only to channel 1 (CH1) which allows the CoBox to deal with the software-related details of using RS-232, RS-422, and RS-485.

4.7.5 Enter Serial Parameters (B,D,P,S)

Enter the baud rate (300/ 600/ 1200/ 2400/ 4800/ 9600/ 19,200/ 38,400, or 115,000), data bits (7/8), parity (N/O/E), and stop bits (1/2) in the classic “DOS Mode Command” style. Examples are: 9600,8,E,1 or 1200,7,O,2.

All CoBox models support baud rates of 300 to 38400, while the dual-port external models also support 115000 baud. This setting must match exactly the setting on the attached Modbus device.

NOTE: *Regardless of this setting, CoBox Modbus configuration is always done on CH1 with RS-232 and a setting of 9600,8,N,1.*

4.8 Modem Control Settings

When using RS-232, CoBox Modbus has a number of user-definable “Modem Control” parameters to manage RTS/CTS handshaking for half-duplex radio modems.

4.8.1 DCD Mode (1=Fixed 2=Variable)

Answer “1” and the CoBox Modbus DCD output is fixed high. Answer “2” and it is only high if there is at least 1 active TCP connection. For a CoBox with attached slave, this means at least 1 Modbus/TCP master is interested in our data. For a CoBox with attached master, this means at least 1 slave Modbus/TCP device accepted a connection.

4.8.2 RTS/CTS Mode (1=Fixed 2=Variable)

Answer “1” and the CoBox RTS/CTS output is fixed high. Answer “2” and it acts as modem handshaking.

4.8.3 Delay after Output of RTS (0-1275 msec, 5ms resolution)

Only asked if RTS/CTS mode is variable. After the CoBox asserts the RTS/CTS signal, it delays from 0 to 1275 msec before continuing. Normally this is set to 0. Only set a value here if your device, modem or cable is non-standard.

4.8.4 Wait for CTS to Go Active (N/Y)

Only asked if RTS/CTS mode is variable. Answering “N” causes the CoBox Modbus to ignore the RTS/CTS response from the modem. Answering “Y” causes the CoBox Modbus to wait for the RTS/CTS response from the modem. Don’t answer “Y” unless you know that the cable is wired properly to support this signal.

4.8.5 Delay after CTS Going Active (0-1275 msec, 5ms resolution)

Only asked if RTS/CTS mode is variable and set to wait for CTS to go active. After the CoBox sees the modem assert an RTS/CTS response input, it delays from 0 to 1275 msec before transmitting. If the CoBox waits without seeing a valid response from the modem, it will return the Modbus exception response 0x0B (hex) to the Modbus/TCP requesting master.

4.8.6 Delay Dropping RTS After Transmitting (0-1275 msec, 5ms resolution)

Only asked if RTS/CTS mode is variable. After the CoBox completes transmission, it delays from 0 to 1275 msec before dropping the RTS/CTS output.

4.9 Advanced Modbus Protocol Settings

Changing these parameters takes a bit of thought and planning.

4.9.1 Slave Address (0 for auto, or 1..255 fixed otherwise)

Modbus/TCP includes a “Unit ID” field, which is used to address multiple Modbus slaves at a single IP address. Unfortunately, some first generation software drivers assumed a single slave at each IP and always set the Unit ID field to 0. This causes the CoBox Modbus problems because it requires the Unit ID for the Modbus/RTU “Slave Address”. To support these older applications, the CoBox allows you to force a fixed address for Modbus/RTU and Modbus/ASCII, but note that this restricts you to a single serial slave device per CoBox Modbus.

Setting this value to 0 causes the CoBox to use the Modbus/TCP Unit ID as received. Setting it to any other address causes the CoBox to always use the set value as a fixed address.

4.9.2 Allow Modbus Broadcasts (1=Yes 2=No)

This actually relates to the previous issue. The default is 2/No, in which case CoBox Modbus always assumes a Modbus/TCP “Unit ID” of 0 really means Modbus slave address 1. Answering “No” here is like setting a fixed address of 1 (parameter above), except the fixed address is only used if the Modbus/TCP “Unit ID” is 0.

NOTE: *In the current software version for CoBox Modbus, a true Modbus broadcast is only supported when a serial slave device is attached. A Modbus broadcast from a serial master device is discarded regardless of this parameter setting.*

4.9.3 Character Timeout (10-1275 msec, 5ms resolution)

This sets the timeout between characters received. Official Modbus/RTU defines a 3.5 character time-out, but complex devices have various interrupts which can cause 5 to 10 character “pauses” during transmission. A safe value for general use with Modbus/RTU is 50msec and this is fixed to 1000msec for Modbus/ASCII.

NOTE: *Setting this value lower than 50msec will not improve performance and may even make it worse. CoBox Modbus uses an intelligent length-predicting algorithm to detect end-of-message in Modbus/RTU. This “character timeout” is only used with user-defined or non-standard Modbus functions.*

4.9.4 Message Timeout (500-60000 msec, 250ms resolution)

This sets the timeout for a response from a connected slave.

4.10 Unit ID to IP Address Lookup Table

These parameters only apply when a serial Modbus master device is attached. Since traditional Modbus uses 8-bit slave addresses and a TCP/IP network requires 32-bit IP addresses, CoBox Modbus uses this table to map an 8-bit address into an IP/Unit ID combination. The 8-bit address is used to both select the desired IP and as the Unit ID sent. The table holds 8 entries, and any Modbus slave address not found in the table returns an exception response to the master. Below is an example of adding an entry.

```
A)dd, D)elete, E)xit - select function: A
Modbus addr from (191) 220
Modbus addr to (191) 255
Slave IP address (192) .(168) .(000) .(000) 32

1): 001-009: 192.168.000.030
2): 010-190: 192.168.000.031
3): 220-255: 192.168.000.032
```

4.10.1 Select Function A)dd, D)elete, E)xit

You can either add or delete entries in the IP address table. They are automatically sorted into increasing order. Enter “E” when you are satisfied with the table to return to the main menu.

4.10.2 Modbus Address From/To

This is the minimum/maximum Modbus slave address (inclusive) to forward to this IP address.

4.10.3 Slave IP Address

This is the IP address of the remote Modbus/TCP slave.

5 Monitor Mode and Firmware Upgrade

To enter monitor mode, the same principal as setting the parameters is used (see Section 4.3) Instead of entering 3 “x” keys, key in “xx1”. Within 1 second of power-up, the CoBox Modbus will respond with a special prompt. To start the monitor mode without network functions (no network connections), enter “xx2”. To enter the monitor mode, in addition to “xx1” and “xx2” you can also type “yyy” and log in.

To enter the monitor mode using a Telnet connection, after the Telnet session is established, you will see the following message.

Lantronix UTS Snr 222-195 V3.5

Press Enter to go into Setup Mode

At this point, type M (upper case). If you see 0> prompt, it means that you have entered the monitor mode successfully.

5.1 Monitor Commands

The following commands are available in the monitor mode. Many commands have an IP address as an optional parameter (x.x.x.x). If it is given, the command is applied to another CoBox Modbus with that IP address. If no IP address is given, the command is executed locally.

All commands must be given in capital letters; only blanks (spaces) are accepted between parameters.

DL	Firmware download to the CoBox Modbus
SF x.x.x.x	Send firmware to CoBox Modbus with IP x.x.x.x
VS x.x.x.x	Query software header record (16-byte)
GC x.x.x.x	Get configuration as HEX records
SC x.x.x.x	Set configuration from HEX records
PI x.x.x.x	Check with Ping if x.x.x.x is alive and reachable
AT	Show the CoBox Modbus’s ARP table entries
TT	Shows all the incoming and outgoing TCP connections (used only with “monitor mode” from Telnet)

- NC** Shows the IP configuration of the CoBox Modbus.
- RS** Resets the power on the CoBox Modbus.
- SI x.x.x.x:n.n.n.n** With this command, you can remotely assign an IP address to another CoBox Modbus, where x.x.x.x is the new IP address and n.n.n.n is the remote CoBox Modbus serial number written twice. For example:
- SI194.39.78.234:146.138.146.138
- IP address = 194.39.78.234
- Remote CoBox Modbus serial # (146-138) = 146.138.146.138

NOTE: *Since this is obtained by sending broadcast packets, this IP assignment cannot be done over the routers.*

- QU** Quit - exit diagnostics mode

Command result codes:

- 0** OK, no error
- 1** No answer from remote device
- 2** Cannot reach remote device or does not answer
- 8** Wrong parameter(s)
- 9** Invalid command

5.2 Firmware Download Using Serial Port

Downloading is done in monitor mode. Once the CoBox Modbus is in monitor mode, by using the “DL” command, the CoBox Modbus will wait for the firmware image in Intel Hex format. This must only be sent through the serial interface. When the end record is received, the CoBox Modbus checks the integrity of the firmware image and then programs the new firmware in the flash ROM. Do not switch off the power supply at this time. A loss of power while reprogramming will result in a corrupt program image and a nonfunctional CoBox Modbus.

The following figure displays a typical screen while programming the new firmware. After a complete reprogramming, the CoBox Modbus restarts.

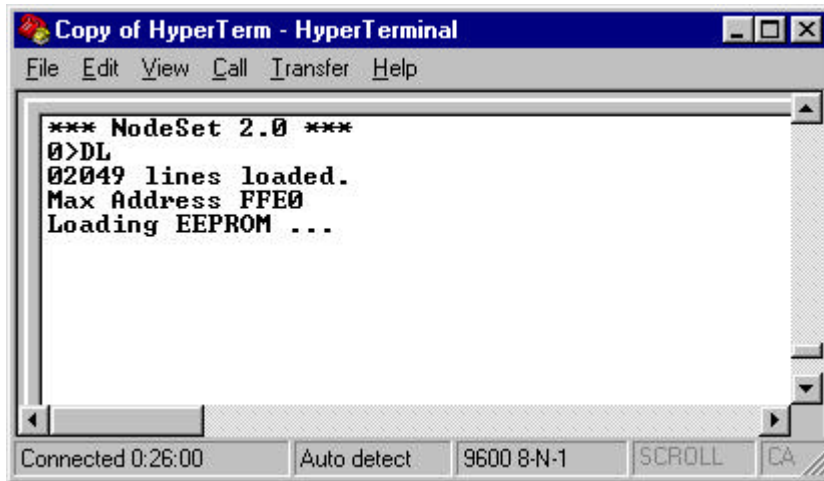


Figure 5-1

5.3 Firmware Distribution

To distribute the firmware of one CoBox Modbus to others, the "SF" command is used. After entering monitor mode on the CoBox Modbus, simply send the firmware with the "SF" command to the other CoBoxes.

5.4 Firmware Download Using a Network Host

To download new firmware from a computer to a CoBox Modbus, it is necessary to have a TFTP client sending a binary file. The parameters to send the binary firmware are shown in figure 5-2.

NOTE: *The file to be downloaded must be the .ROM (binary) image and not the .HEX (ASCII) version.*

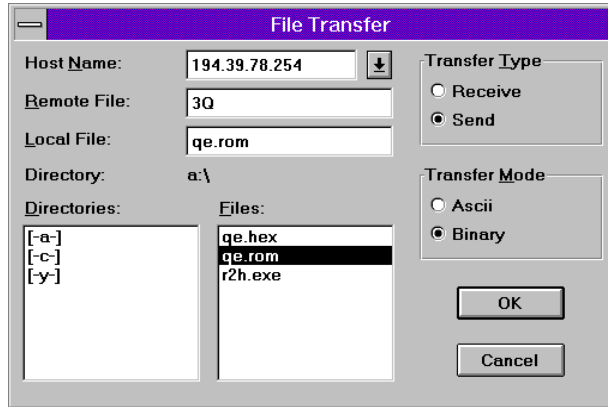


Figure 5-2

6 Configuring Third Party Software for CoBox Modbus

6.1 Modsoft v2.6 or later

Modsoft is the traditional tool used to program and configure Modicon equipment. To support Modbus/TCP you require Modsoft version 2.6 or later. Set the Communications Parameters as follows:

- **Protocol** : **TCP/IP**
- **Dest_Port** : **502**
- **Dest_Index** : **Modbus/RTU slave address**
- **TCP/IP Address** : **as required**

NOTE: *Modsoft's "Dest_Index" is the Modbus/TCP "Unit ID" parameter, therefore you do not have to use the "Fixed Slave Address" setting in the CoBox Modbus and RS-485 multi-drop is supported.*

6.2 Win-Tech ModScan32

Wintech (at www.win-tech.com) puts out a number of useful tools for general purpose Modbus support. Of greatest interest is ModScan32, which allows you to select and poll coils and registers from various Modbus/RTU, Modbus/ASCII, and Modbus/TCP slave devices. This is a good tool for troubleshooting.

NOTE: *Wintech properly supports the Modbus/TCP "Unit ID" parameter, therefore you do not have to use the "Fixed Slave Address" setting in the CoBox Modbus and RS-485 multi-drop is supported.*

6.3 IO-Server OPC Server

IOServer (at www.ioserver.com) makes a general purpose OPC server which properly supports Modbus/TCP. This OPC server can be accessed by Intellution, Genesis, Wonderware, and most other commercial HMI products.

NOTE: *IOServer properly supports the Modbus/TCP "Unit ID" parameter, therefore you do not have to use the "Fixed Slave Address" setting in the CoBox Modbus and RS-485 multi-drop is supported.*

6.4 Intellution FIX HMI

The CoBox MB allows Windows workstations with FIX to access Modbus/RTU devices by TCP/IP over Ethernet. To configure, just enter the CoBox Modbus's IP address.

NOTE: *At the time of this writing, Intellution does not properly support the Modbus/TCP "Unit ID" parameter, therefore you may have to use the "Fixed Slave Address" setting in the CoBox Modbus and only a single slave device is supported.*

7 Connectors' Pinouts

7.1 Channel 1 (RS-232C)

This connector is available on the CoBox Modbus External Models. Configuration is DCE (modem-like), 25-pin female SubD—you can use the standard cable as recommended by your device manufacturer to connect to a modem.

Pin	Direction	Function
1		None
2	to CoBox Modbus	TxD Transmitted data
3	from CoBox Modbus	RxD Received data
4	to CoBox Modbus	RTS Request to send
5	from CoBox Modbus	CTS Clear to send
6	from CoBox Modbus	DSR Data set ready
7		Ground
8	from CoBox Modbus	DCD Data carrier detect
20	to CoBox Modbus	DTR Data Terminal Ready

Figure 7-1

NOTE: Please make sure that other pins are not connected, as there are other signals on the connector. Improper wiring of these pins might damage the CoBox Modbus.

NOTE: DSR output (pin #6) is fixed high/ON, and DCD (pin #8) is user configurable. Vary the cable design to use DCD as DSR if your software benefits.

<p>CH1 to a Computer with 9-pin DTE: Most desktop computers now come with a 9-pin D-Sub connector - sometimes called an "AT" style serial connector. You can use any standard "Modem Cable" suitable from a 9-pin computer to a 25-pin modem.</p>	<table border="1"> <thead> <tr> <th colspan="2">PC/DTE COM Port 9-pin M</th> <th colspan="2">CoBox/DCE CH1 25-pin F</th> </tr> </thead> <tbody> <tr> <td>DCD</td> <td>1</td> <td></td> <td>8</td> </tr> <tr> <td>RXD</td> <td>2</td> <td></td> <td>3</td> </tr> <tr> <td>TXD</td> <td>3</td> <td></td> <td>2</td> </tr> <tr> <td>DTR</td> <td>4</td> <td></td> <td>20</td> </tr> <tr> <td>GND</td> <td>5</td> <td></td> <td>7</td> </tr> <tr> <td>DSR</td> <td>6</td> <td></td> <td>6</td> </tr> <tr> <td>RTS</td> <td>7</td> <td></td> <td>4</td> </tr> <tr> <td>CTS</td> <td>8</td> <td></td> <td>5</td> </tr> </tbody> </table>	PC/DTE COM Port 9-pin M		CoBox/DCE CH1 25-pin F		DCD	1		8	RXD	2		3	TXD	3		2	DTR	4		20	GND	5		7	DSR	6		6	RTS	7		4	CTS	8		5
PC/DTE COM Port 9-pin M		CoBox/DCE CH1 25-pin F																																			
DCD	1		8																																		
RXD	2		3																																		
TXD	3		2																																		
DTR	4		20																																		
GND	5		7																																		
DSR	6		6																																		
RTS	7		4																																		
CTS	8		5																																		
<p>CH1 to a Computer with 25-pin DTE: Older computers and many field devices come with a 25-pin D-Sub connector – sometimes called an "XT" style serial connector. You can use any standard "Modem Cable" suitable from a 25-pin computer to a 25-pin modem.</p>	<table border="1"> <thead> <tr> <th colspan="2">PC/DTE COM Port 25-pin M</th> <th colspan="2">CoBox/DCE CH1 25-pin F</th> </tr> </thead> <tbody> <tr> <td>TXD</td> <td>2</td> <td></td> <td>2</td> </tr> <tr> <td>RXD</td> <td>3</td> <td></td> <td>3</td> </tr> <tr> <td>RTS</td> <td>4</td> <td></td> <td>4</td> </tr> <tr> <td>CTS</td> <td>5</td> <td></td> <td>5</td> </tr> <tr> <td>DSR</td> <td>6</td> <td></td> <td>6</td> </tr> <tr> <td>GND</td> <td>7</td> <td></td> <td>7</td> </tr> <tr> <td>DCD</td> <td>8</td> <td></td> <td>8</td> </tr> <tr> <td>DTR</td> <td>20</td> <td></td> <td>20</td> </tr> </tbody> </table>	PC/DTE COM Port 25-pin M		CoBox/DCE CH1 25-pin F		TXD	2		2	RXD	3		3	RTS	4		4	CTS	5		5	DSR	6		6	GND	7		7	DCD	8		8	DTR	20		20
PC/DTE COM Port 25-pin M		CoBox/DCE CH1 25-pin F																																			
TXD	2		2																																		
RXD	3		3																																		
RTS	4		4																																		
CTS	5		5																																		
DSR	6		6																																		
GND	7		7																																		
DCD	8		8																																		
DTR	20		20																																		

7.2 Channel 1 (RS-422/485)

The same CH1 connector supports RS-485 for the external CoBox Modbus Models.

Pin	Function
1	None
7	Ground
14	DOUT +
15	DOUT -
21	DIN+
22	DIN -

Figure 7-2

NOTE: For RS-485 2-wire functionality pins 14 and 21, as well as 15 and 22, must be connected together.

7.3 Channel 2 (RS-232C)

This connector is available on the CoBox Modbus External Models. Configuration is PC-like (9-pin male DB). CH2 is a 9-pin DTE port just like you find on most computers. You can use the standard cable as recommended by your device manufacturer to connect to a computer.

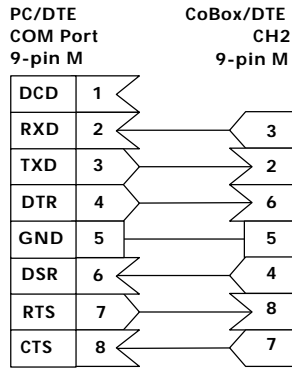
Pin	Direction	Function
1	to CoBox Modbus	DCD (fixed high)
2	to CoBox Modbus	RxD Received data
3	From CoBox Modbus	TxD Transmitted data
4	From CoBox Modbus	DTR (acts like DCD)
5		Ground
7	From CoBox Modbus	RTS
8	to CoBox Modbus	CTS

Figure 7-3

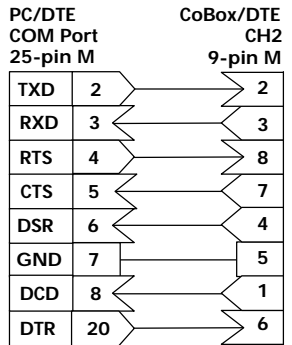
NOTE: DCD output (pin #1) is fixed high/ON, and DTR (pin #4) is user configurable. Vary the cable design to use DCD and DTR if your software benefits.

CH2 to a Computer with 9-pin DTE:

Most desktop computers now come with a 9-pin D-Sub connector – sometimes called an “AT” style serial connector. You can use a “Null-Modem” or “crossed cable”.



CH2 to a Computer w/ 25-pin DTE: Older computers and many field devices come with a 25-pin D-Sub connector – sometimes called an “XT” style serial connector. Again, a Null-Modem (or crossed) cable is required.



8 LED Status Display

Depending on the model, 3 different LEDs display the status of the CoBox Modbus. Most models also have some network diagnostic LEDs such as Good Link (GL), Receive (RX), Transmit (TX), or Collision (CL). They are labeled on the product.

8.1 Yellow and Green LED

The upper green LED displays the Modbus/TCP activity.

The lower yellow LED displays the Modbus/RTU activity.

The red LED is off during normal operation.

8.2 Red LED

If the red LED is solid on or blinking, the green LED patterns will indicate the type of error.

Red LED solid on, green LED blinking:

- 1x: EPROM-checksum error
- 2x: RAM-error
- 3x Network controller error (Token Ring)
- 4x: E²PROM checksum error
- 5x: IP address already used on the network

Red LED blinking, green LED blinking:

- 4x: The network connection is faulty. This LED pattern should only appear after power up. Even though the CoBox Modbus is going into operation mode, the problem will potentially persist.
- 5x: No DHCP response was received

9 Technical Data

Some values are for the standard external models only.

9.1 CPU, Memory Controllers

V.40 CPU, 10MHz clock
Z85C30 SCC Serial Communications Controller
National Semiconductor DP839xx Ethernet Controller or TMS340 chipset for Token Ring
Version
128kByte RAM, 128 or 256kByte Flash PROM
256 Byte EPROM for parameter storage

9.2 Serial Interface

25-pin D-shell female connector (DCE pinout)
Speed software selectable 300 to 115k baud
Software selectable RS-232C or RS-422/485

Second channel RS-232C, DB9 male (DTE pinout) interface
Speed software selectable 300 to 115k baud

9.3 Network Interface

15-pin AUI interface (External Ethernet only) for transceiver connection to:
Thick Wire Ethernet
Thin Ethernet
Fiber Optics

Integrated 10-BaseT port (RJ-45 connector)

9.4 Power Supply

Power Plug (supplied), 15-18 Volt AC, max. 800mA

9.5 Power Consumption

Max. 9 Watts

9.6 LEDs

3 LEDs for channel status display
4 LEDs for network interface status (Ethernet)

9.7 Case

Aluminum case, with removable mounting brackets

9.8 Dimensions

180 x 155 x 40 mm (7.1 x 6.1 x 1.6 in)

9.9 Weight

Approx. 500g without AC power adapter

A Token Ring Specific

A.1 Additional Information for Token Ring Versions

The following information describes differences for Token Ring versions.

A.1.1 Token Ring Insertion Process Needs Time

When the CoBox Modbus attempts to enter the ring, extensive tests of the ring physical interface are done before opening the loop. These tests take a few seconds. After opening the ring, the CoBox Modbus waits for a token from a ring master. If there is no other device in the ring, it may take up to 30 seconds until the ring insertion process is completed and the red LED stops blinking.

A.2 Network Interface

The Token Ring version of CoBox Modbus has a standard D-Shell 9-pin female connector (STP) to be connected to a standard Token Ring drop cable and an RJ-45 connector (UTP) to be connected to a UTP Token Ring medium. The used connector will be automatically detected. The ring interface speed is configured by a jumper, to prevent accidental configuration. The jumper can be set through a hole in the back of the case. The factory setting of this jumper is 16Mbit.

A.3 Token Ring Simulator

The following schematic defines a simple Token Ring simulator that can be used to test the CoBox Modbus network interface without a MAU.

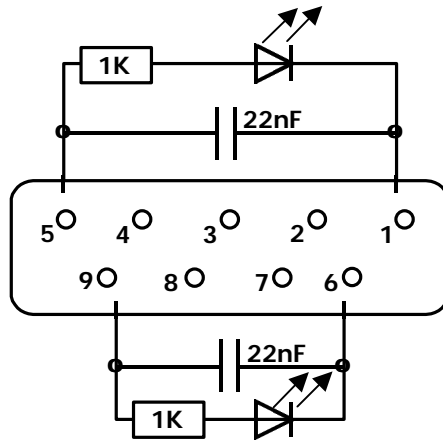


Figure A-1

B IP Addresses, Netmask etc.

B.1 IP Addressing

An IP address is a 32-bit value, divided into 4 octets of 8 bits each. The standard representation is 4 decimal numbers (in the range of 0..255), divided by dots.

Example: 192.2.1.123

This is called decimal-dot notation.

The IP address is divided in 2 parts: network and host. To support different needs, 3 “network classes” have been defined. Depending on the network class, the last 1, 2, or 3 bytes define the host, while the remaining part defines the network. In the following, “x” stands for the host part of the IP address:

B.2 Class A Network

IP address 1.x.x.x to 127.x.x.x

Only 127 different networks of this class exist. These have a very large number of potential connected devices (up to 16,777,216)

Example: 10.0.0.1, (network 10, host 0.0.1)

B.3 Class B Network

IP address 128.0.x.x to 191.255.xxx.xxx

These networks are used for large company networks. Every network can consist of up to 65,534 devices.

Example: 172.1.3.2 (network 172.1, host 3.2)

B.4 Class C Network

IP address 192.0.0.xxx to 223.255.255.xxx

Example: 192.7.1.9 (network 192.7.1, host 9)

These network addresses are most common and are often used in small companies. These networks can consist of a maximum number of 254 hosts.

Example: 192.7.1.9 (network 192.7.1, host 9)

The remaining addresses 224.x.x.x - 239.x.x.x are defined as “class D” and are used as a multicast addresses.

The addresses 240.x.x.x - 254.x.x.x are defined as “class E” and are reserved addresses.

B.5 Network Address

The host address with all host bits set to “0” is used to address the network as a whole (for example in routing entries).

B.6 Broadcast Address

The address with the host part bits set to “1” is the broadcast address, meaning “for every station”.

Network and Broadcast addresses must not be used as a host address (e.g. 192.168.0.0 identifies the entire network, 192.168.0.255 identifies the broadcast address).

B.7 IP Netmask

The netmask is used to divide the IP address differently from the standard defined by the classes A, B, and C. A netmask defines how many bits from the IP address are to be taken as the network section and how many bits are to be taken as the host section. When the number of host bits is entered, the CoBox Modbus calculates the netmask. The netmask is displayed in standard decimal-dot notation.

	Network Bits	Host Bits	Netmask
Class A	8	24	255.0.0.0
Class B	16	16	255.255.0.0
Class C	24	8	255.255.255.0

Figure B-1

B.7.1

Netmask	Host bits
255.255.255.252	2
255.255.255.248	3
255.255.255.240	4
255.255.255.224	5
255.255.255.192	6
255.255.255.128	7
255.255.255.0	8
255.255.254.0	9
255.255.252.0	10
255.255.248.0	11
.	.
.	.
255.128.0.0	23
255.0.0.0	24

Figure B-2

B.7.2 Private IP Networks and the Internet

If your network is not connected to the Internet and there are no plans to make such a connection you may use any IP address you wish.

If you are not connected to the Internet and have plans to connect, or you are connected to the Internet and want to operate your CoBox Modbus on an Intranet you should use one of the sub-networks below. These network numbers have been reserved for such networks. If you have any questions about IP assignment consult your network administrator.

- Class A 10.x.x.x**
- Class B 172.16.x.x**
- Class C 192.168.0.x**

B.7.3 Network RFCs

For more information regarding IP addressing see the following documents. These can be located on the World Wide Web using one of the directories or indices.

RFC 950	Internet Standard Subnetting Procedure
RFC 1700	Assigned Numbers
RFC 1117	Internet Numbers
RFC 1597	Address Allocation for Private Internets

C Binary to Hex Conversion

Hexadecimal digits have values from 0..15, represented as 0...9, A (for 10), B (for 11) ... F (for 15). The following table can serve as a conversion chart for binary-to-decimal-to-hex.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Figure C-1

To convert a binary value to a hexadecimal representation, the upper and lower 4 bits are treated separately, resulting in a 2-digit hexadecimal number.

Warranty Statement

Lantronix warrants for a period of FIVE YEARS from the date of shipment that each CoBox server supplied shall be free from defects in material and workmanship. During this period, if the customer experiences difficulties with a product and is unable to resolve the problem by phone with Lantronix Technical Support, a Return Material Authorization (RMA) will be issued. Following receipt of a RMA number, the customer is responsible for returning the product to Lantronix, freight prepaid. Lantronix, upon verification of warranty will, at its option, repair or replace the product in question, and return it to the customer freight prepaid. No services are handled at the customer's site under this warranty.

Lantronix warrants software for a period of sixty (60) days from the date of shipment that each software package supplied shall be free from defects and shall operate according to Lantronix specifications. Any software revisions required hereunder cover supply of distribution media only and do not cover, or include, any installation. The customer is responsible for return of media to Lantronix and Lantronix for freight associated with replacement media being returned to the customer.

Lantronix shall have no obligation to make repairs or to cause replacement required through normal wear and tear of necessitated in whole or in part by catastrophe, fault or negligence of the user, improper or unauthorized use of the Product, or use of the Product in such a manner for which it was not designed, or by causes external to the Product, such as, but not limited to, power or failure of air conditioning.

There are no understandings, agreements, representations or warranties, express or implied, including warranties of merchantability or fitness for a particular purpose, other than those specifically set out above or by any existing contract between the parties. Any such contract states the entire obligation of Lantronix. The contents of this document shall not become part of or modify any prior or existing agreement, commitment or relationship.

The information, recommendation, description and safety notations in this or other documents supplied by Lantronix are based on general industry experience and judgment with respect to such hardware and software. THIS INFORMATION SHOULD NOT BE CONSIDERED TO BE ALL INCLUSIVE OR COVERING ALL CONTINGENCIES.

NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE OR MERCHANTABILITY, OR WARRANTIES ARISING FROM COURSE OF DEALING OR USAGE OF TRADE, ARE MADE REGARDING THE INFORMATION, RECOMMENDATIONS, DESCRIPTIONS AND SAFETY NOTATIONS CONTAINED HEREBY AND IN HARDWARE AND SOFTWARE SPECIFICATION DOCUMENTATION, OR INSTRUCTIONS SUPPLIED BY LANTRONIX. In no event will Lantronix be responsible to the user in contract, i tort (including negligence), strict liability or otherwise for any special, indirect, incidental or consequential damage or loss of equipment, plant or power system, cost of capital, loss of profits or revenues, cost of replacement power, additional expenses in the use of existing software, hardware, equipment or facilities, or claims against the user by its employees or customers resulting from the use of the information, recommendations, descriptions and safety notations supplied by Lantronix. Lantronix liability is limited (at its election) to (1) refund of buyer's purchase price for such affected products (without interest); (2) repair of such products, or (3) replacement of such products, provided however, that the buyer follows the procedures set forth herein.

Warranty claims must be received by Lantronix within the applicable warranty period. A replaced product, or part thereof, shall become the property of Lantronix and shall be returned to Lantronix at the Purchaser's expense. ALL RETURN MATERIAL MUST BE ACCOMPANIED BY A RETURN MATERIAL AUTHORIZATION NUMBER ASSIGNED BY LANTRONIX.

Declaration of Conformity

according to ISO/IEC Guide 22 and BS 7514

Manufacturer's Name: Lantronix
Manufacturer's Address: 15353 Barranca Parkway
Irvine, CA 92618
USA

The manufacturer hereby declares that the product:

Product Name: CoBox Modbus Universal Thin Servers
Product Number(s): CoBox-E1M-01/-02/-03
CoBox-TRM-01/-02/-03

Conforms to the following standards or other normative documents:

Electromagnetic Emissions: EN55022, 1994 Class "B"
Electromagnetic Immunity: EN6100-3-2
EN6100-3-3
EN50082-2, 1995
Product Safety: EN60950, 1988 +A1, A2

Supplementary Information:

These products have been verified as being compliant within the class B limits of the FCC Radio Frequency Devices Rules (FCC Part 15, Subpart B), revised as of October 1993.

"The product complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC."

28 January, 1999
Irvine, CA 92618



A

ARP, 4-2
command, 4-2

B

Basic Commands, 4-4

C

CH1 to Computer w/ 25-pin DTE, 7-2
CH1 to Computer w/ 9-pin DTE, 7-2
CH2 to Computer w/ 25-pin DTE, 7-4
CH2 to Computer w/ 9-pin DTE, 7-3
Character Timeout, 4-8
Clear to send, 7-1
CoBox Modbus
Ethernet, 2-2
Token Ring, 2-2
CoBox Modbus Interfaces
Hardware address, 2-1
Network, 2-1
Serial, 2-1
Configuration, 4-1
Network, 4-2
Parameters. see Parameters
Serial, 4-1

D

Data carrier detect, 7-1
Data set ready, 7-1
Data Terminal Ready, 7-1
DCD Mode, 4-7
Default Settings, 4-4
Delay After CTS Going Active, 4-7
Delay After Output of RTS, 4-7
Delay Dropping RTS After Transmitting, 4-7
DHCP, 3-1, 4-2, 4-5, 8-1

F

Firmware Upgrade, 5-1

G

Grounding, 2-2

H

Hardware Address
CoBox Modbus with Ethernet, 2-2
CoBox Modbus with Token Ring, 2-2
Network, 2-1
HMI, 1-2

HyperTerminal accessory, 4-1

I

Intellution FIX HMI, 6-2
IOServer OPC Server, 6-1
IP Address, 3-1, B-4
default, 4-2
IP Addressing, 3-1

M

Message Timeout, 4-8
Modbus
Master Devices, 1-2
Slave Devices, 1-2
Modbus Broadcasts, 4-8
Modbus Slave Address, 3-1
Modsoft v2.6 or later, 6-1
Modem Control Settings, 4-6
Monitor Mode
Monitor Commands, 5-1

N

Netmask, 4-5, B-1, B-2, B-3
Network/IP Settings, 4-4
Network Protocols, 3-1
ARP, 3-1
DHCP, 3-1
ICMP, 3-1
IP address, 3-1
Packing algorithm, 3-1
Slave address, 3-1
SNMP, 3-1

TCP, 3-1
TFTP, 3-1
UDP, 3-1

P

Packing Algorithm, 3-1
Parameters, 4-3
Advanced Modbus Protocol Settings, 4-7
Modbus Broadcasts, 4-8
Attached Device, 4-6
Basic Commands, 4-4
Character Timeout, 4-8
DCD Mode, 4-7
Default Settings, 4-4
Delay After CTS Going Active, 4-7
Delay After Output of RTS, 4-7
Delay Dropping RTS After Transmitting, 4-7
IP address, 4-4

- Message Timeout, 4-8
- Modbus Address From/To, 4-9
- Modem Control Settings, 4-6
- Network/IP Settings, 4-4
- Quit Without Saving, 4-5
- RTS/CTS Mode, 4-7
- Save (S), 4-4
- Select Function, 4-9
- Serial & Mode Settings, 4-5
- Serial Interface Type, 4-6
- Serial Parameters - Baud Rate etc, 4-6
- Serial Protocol, 4-6
- Set Gateway IP address, 4-5
- Set Netmask, 4-5
- Slave Address, 4-8
- Slave IP Address, 4-9
- Unit ID to IP Address Table, 4-9
- Use Serial Connector, 4-6
- Wait for CTS to Go Active, 4-7

Power Supply, 2-2

R

- Received data, 7-1
- Request to send, 7-1
- RJ-45 connector, 9-1
- RS-232, 1-1, 1-3, 2-1, 4-6, 7-1, 7-3, 7-4
- RS-485, 1-1, 1-3, 1-4, 2-1, 4-6, 6-1, 7-3, 7-4
- RS-422, 1-1
 - RTS/CTS Mode, 4-7
- RTU
 - Serial Master, 1-3
 - Serial Slave, 1-3

S

- Serial connector
 - AT style, 6-2, 6-3
 - XT style, 6-2, 6-4
- Serial Data Communication
 - RS-232, 1-1
 - RS-422, 1-1
 - RS-485, 1-1
- Serial & Mode Settings, 4-5
- Serial Interface Type, 4-6
- Serial Master, 1-3
- Serial Slave, 1-3
- Serial Parameters - Baud Rate etc, 4-6
- Serial Protocol, 4-6
- STP, 2-2

T

- TCP
 - Master, 1-3
 - Slave, 1-3
- TCP/IP, 1-4, 3-1, 4-2, 4-5, 4-9, 6-1, 6-2
- Technical Data, 8-1
- Telnet, 4-2
- Token Ring, 2-2, 8-1, 9-1, A-1, A-2
 - simulator, A-2
- Transmitted data, 7-1

U

- Unit ID, 3-1
- Unix
 - command example, 4-3
- UTP, 2-2

W

- Hyper Terminal accessory, 4-1
- Win-Tech ModScan32, 6-1