Embedded Integration Kit User’s Guide

Covering these Device Servers:

CoBox-Micro

CoBox-Mini

CoBox-Mini100

UDS-10B

Revision C 1/02

Part Number 900-226

Copyright and Trademark

WARNING

This product has been designed to comply with the limits for a Class A digital device pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against such interference when operating in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with this guide, may cause harmful interference to radio communications.

Operation of this equipment in a residential area is likely to cause interference in which case the user, at his or her own expense, will be required to take whatever measures may be required to correct the interference.

Changes or modifications to this device not explicitly approved by Lantronix will void the user's authority to operate this device.

Lantronix
15353 Barranca Parkway
Irvine CA 92618, USA
Phone: 949-453-3990

Technical Support
Phone: 800-422-7044 or 949-453-7198
Fax: 949-450-7226
On-line: www.lantronix.com/support

Date	Rev.	Part No.	Comments
12/00	A	900-226	Initial Release
11/01	B	900-226	In Chapter 2, CoBox-Micro, swapped signals of Con4 Pins 3 and 7; cosmetic changes
1/02	C	900-226	Edited Page 2-1 and updated graphic

Chapter 1: Introduction

Chapter 2: CoBox-Micro

Chapter 3: CoBox-Mini

Chapter 4: CoBox-Mini100

Chapter 5: UDS-10B

Chapter 6: Getting Started

Chapter 7: Configuration

1: Introduction

This manual provides the information needed to use the three Lantronix Embedded Integration Kits in conjunction with three of our embedded Universal Device Servers. It also describes (Chapter 5) the UDS-10B Device Server (which does not have an integration kit).

The next four chapters are devoted to embedded Device Servers. The chapters and appendixes that follow them are devoted to those products in relationship to the Embedded Integration Kit.

Embedded Integration Kits

The three Lantronix Embedded Integration Kits (Mini-Kit, Mini100-Kit, Micro-Kit) provide an elegantly simple method of evaluating our CoBox-Mini, CoBox-Mini100, and CoBox-Micro embedded Device Servers. These kits allow software engineers to test our Device Servers with their products prior to the hardware development of those products.

Each kit contains an embedded Device Server, board carrier unit, and all the connectors and data needed to interface our product with your serial device. The Embedded Integration Kits include the following:

- An embedded Device Server: CoBox Micro, CoBox Mini, or CoBox Mini100
- Board carrier unit: with Ethernet (10BaseT RJ45) and serial (DB-9) interfaces
- Power supply
- 10-foot unshielded twisted pair cable
- 10-foot serial cable
- DB-9 to DB-25 converter
- User's Guide (printed and on CD): installation, firmware configurations, application examples, technical tips
- Software CD: latest Redirector software, run-time and HTTP codes, Winsock examples, and PDF version of the User’s Guide.

Embedded Device Servers

The CoBox-Mini, CoBox-Mini100, CoBox-Micro, and UDS-10B embedded Device Servers connect serial devices to Ethernet networks using the IP protocol family (TCP for connection-oriented stream applications and UDP for datagram applications). The main benefit of these smaller, board-level Device Servers is that they can be embedded inside the serial device for which they are providing network connectivity. A few of the different types of serial devices supported are listed below:

- Time/Attendance Clocks and Terminals
- ATM Machines
- CNC Controllers
- Data Collection Devices
- Universal Power Supply (UPS) Management Units
- Telecommunications Equipment
- Data Display Devices
- Security Alarms and Access Control Devices
- Handheld Instruments

Device Servers connect these devices through a TCP data channel or through a Telnet connection to computers or other Device Servers. Datagrams can be sent by UDP.

Network Protocols

Device Servers use IP protocol for network communications. The supported protocols are ARP, UDP, TCP, ICMP, Telnet, TFTP, DHCP, HTTP, SNMP, and BOOTP. For connections to the serial port, TCP, UDP or Telnet protocols are used. Firmware updates can be performed using TFTP.

The Internet Protocol (IP) defines addressing, routing, and data block handling over the network. The Transmission Control Protocol (TCP) assures that no data is lost or duplicated, and that everything sent to the connection arrives correctly at the target.

For typical datagram applications in which devices interact with other devices without maintaining a point-to-point connection, User Datagram Protocol (UDP) is used.

Packing Algorithms

Two selectable packing algorithms define how and when packets are sent to the network. The standard algorithm is optimized for applications in which the Device Server is used in a local environment, allowing for small delays for characters while keeping the packet count low. The alternate packing algorithm optimizes the timing sequence in which a packet will be transmitted. Adjusting parameters in this mode can economize the network data stream.

Ethernet (MAC) Address

The Ethernet address is also referred to as the hardware address or the MAC address. The first three bytes of the Ethernet Address are fixed (e.g., 00-20-4A), identifying the unit as a Lantronix product. The fourth, fifth, and sixth bytes are unique numbers assigned to each Device Server.

Table 1-1: Sample Ethernet Address

`00-20-4A-14-01-18 or 00:20:4A:14:01:18`

Internet Protocol (IP) Address

Every device connected to an IP network must have a unique IP address. This address is used to reference the specific Device Server. See Appendix E for more information on IP Addressing.

Port Numbers

Every TCP connection and every UDP datagram is defined by a destination IP address and a port number. For example, a Telnet application commonly uses port 23. A port number is similar to an extension on a PBX system. See *Port Number* on page 7-8 for more information.

Compatibility

Revision 4.2 is the current firmware driving CoBox-Micro, CoBox-Mini, and UDS-10B products. Revision 4.3 is the current firmware driving the CoBox-Mini100 product.

> **Note:** *Do not load firmware that is lower than Rev. 4.3 onto the CoBox-Mini100. Firmware prior to Rev. 4.3 will render this product inoperable.*

2: CoBox-Micro

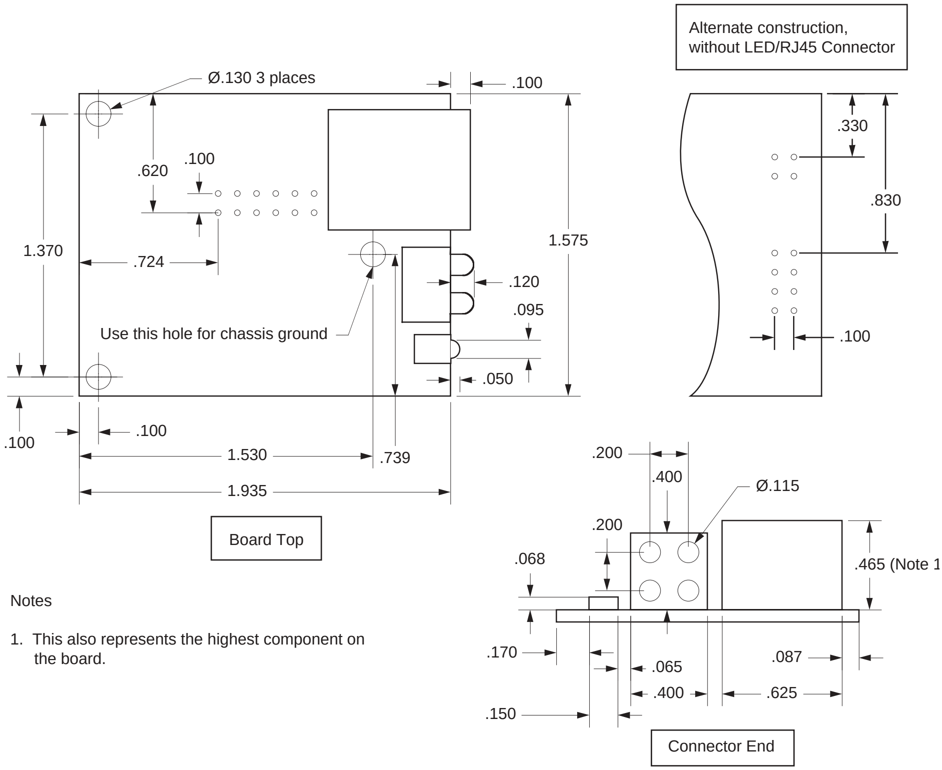
Overview

The CoBox-Micro integrate into products quickly and easily. Serial interfacing is accomplished via a TTL connector, and for Ethernet access, an optional RJ45 (10BASE-T) connector is available. The orientation of its interface pins can be specified to fit your product. It requires 5 volts DC of regulated power with maximum current of 200mA.

The CoBox-Micro's well-developed IP firmware supports protocols such as ARP, UDP, TCP, Telnet, BOOTP, ICMP, SNMP, DHCP, TFTP, and HTTP, as well as other custom protocols. The CoBox-Micro also supports a variety of user-configurable options such as buffer control and packetization, which make it easy to use in most applications.

Board Layout

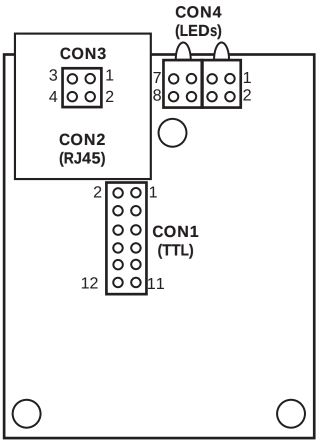
Figure 2-1: CoBox-Micro Board Layout



Connectors

The CoBox-Micro has four connectors: a TTL serial port (CON1), a 10BASE-T RJ45 Connector (CON2), and/or pins instead of the RJ45 connector (CON3) and LEDs (CON4).

Figure 2-2: CoBox-Micro Connector Diagram



At the time of ordering, each connector can be specified as follows:

- Pins (5.46mm) on/off the board and top/bottom
- LEDs on/off the board
- RJ45 on/off the board

Contact Lantronix for information about ordering Device Servers with customized connector configurations.

The Embedded Integration Kit (part number Micro-Kit) includes the CoBox-Micro embedded Device Server. Refer to the following table for a listing of its pinouts.

Table 2-1: CoBox-Micro Connector Pinouts

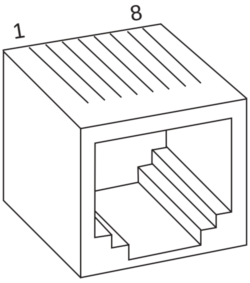
CON1 TTL Serial Port (SIL 2 x 6 Pins)		CON2 10BASE-T (RJ45) Connector		CON3 (SIL 2 x 2 Pins in place of RJ45)		CON4 (SIL 2 x 4 Pins in place of LEDs)	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	+5VDC	1	Tx+	1	Tx+	1	+5VDC
2	GND	2	Tx-	2	Tx-	2	+5VDC
3	RxA (input)	3	Rx+	3	Rx+	3	LED3 (Diagnostics)*
4	TxA (output)	4	None	4	Rx-	4	LED1 (Channel 1)*
5	RTSA (output)	5	None			5	+5VDC
6	DTRA (output)	6	Rx-			6	+5VDC
7	CTSA (input)	7	None			7	LED2 (Channel 2)*
8	DCDA (input)	8	None			8	LED4 (Link)*
9	Reserved						
10	$\overline{\text{RESET}}$ (pull low to reset)						
11	RxB (input)						
12	TxB (output)						

* Current limiting resistor on board is 680 Ohms.
A = Port (Channel) 1 B = Port (Channel) 2

RJ45 Connector

The standard CoBox-Micro ships with an RJ45 10BASE-T Ethernet connector (CON2). At the time of ordering, you can specify whether to include this RJ45 connector. CON3 can be used as an alternative.

Figure 2-3: RJ45 Ethernet Connector (CON2)



1 - Tx+
2 - Tx-
3 - Rx+
6 - Rx-

Status LEDs

The CoBox-Micro has four status LEDs: serial port (Channel) 1 status, serial port (Channel) 2 status, diagnostics, and network link status. See the following table for a complete description of status LED pinout location and function.

Figure 2-4: CoBox-Micro Status LEDs

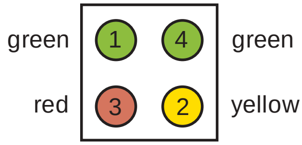


Table 2-2: CoBox-Micro Status LEDs

LED	Description	Location	LED Functions
1	Serial Port (Channel) 1 Status	CON 4, Pin 4	Lights solid green to indicate Channel 1 is *idle*. Blinks green to indicate Channel 1 is connected to the network and *active*.
2	Serial Port (Channel) 2 Status	CON 4, Pin 7	Lights solid yellow to indicate Channel 2 is *idle*. Blinks yellow to indicate Channel 2 is connected to the network and *active*.
3	Diagnostics	CON 4, Pin 3	Blinks or lights solid red in combination with the green (Channel 1) LED to indicate diagnostics and error detection. Red solid, green (Channel 1) blinking: 1x: EPROM checksum error 2x: RAM error 3x: Network controller error 4x: EEPROM checksum error 5x: Duplicated IP address on the network* 6x: Software does not match hardware* Red blinking, green (Channel 1) blinking: 4x: Faulty network connection* 5x: No DHCP response received*
4	Network Link Status	CON 4, Pin 8	Lights solid green to indicate network port is connected to the network.
*non-fatal error			

Product Information Label

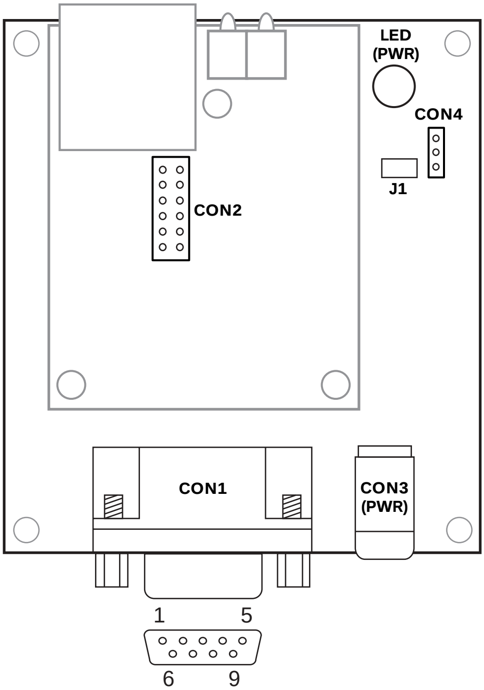
The CoBox-Micro ships with a product information label that can be affixed to the host device. The product label contains information about your specific unit, such as its bar code, serial number, product ID (name), product description, and Ethernet address (also referred to as hardware address or MAC address).

Test Bed

The CoBox-Micro Embedded Integration Kit (part number Micro-Kit) includes a test bed (carrier board) that provides serial connections to the Device Server. The network 10-Base-T RJ45 is provided by the CoBox-Micro Device Server. The test bed contains LEDs, a voltage regulator, a power supply circuit, a reset switch, TTL to RS-232 and RS-232 to TTL conversion hardware, and jumpers for various configuration options.

The test bed allows software engineers to immediately begin developing and testing software applications for the Device Server, rather than delaying the process until the hardware interface for their product is complete.

Figure 2-5: CoBox-Micro Test Bed



Test Bed Connectors

The CoBox-Micro test bed has four connectors: CON1 (Serial Port 1 or Channel 1), CON2 (TTL Interface), CON4 (Serial Port 2 or Channel 2), and CON3, which is a 5VDC power supply connector.

Table 2-3: CoBox-Micro Test Bed Connector Pinouts

CON1 Serial Port (Channel) 1[a]		CON2 TTL Interface		CON4 Serial Port (Channel) 2[b]	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	DTRA (output)	1	+5 VDC	1	TxB (output)
2	TxA (output)	2	GND	2	RxB (input)
3	RxA (input)	3	TxA (output)	3	GND
4	DCDA (input)	4	RxA (input)		
5	GND	5	CTSA (input)		
6	None	6	DCDA (input)		
7	CTSA (input)	7	RTSA (output)		
8	RTSA (output)	8	DTRA (output)		
9	None	9	None		
		10	None		
		11	TxB (output)		
		12	RxB (input)		

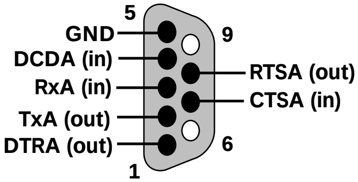
a. "CON1 Serial Port (Channel) 1" is also designated as "A".
b. "CON4 Serial Port (Channel) 2" is also designated as "B".

Note: *CON3 is a PCB-mounted, center-positive 5VDC power supply connector.*

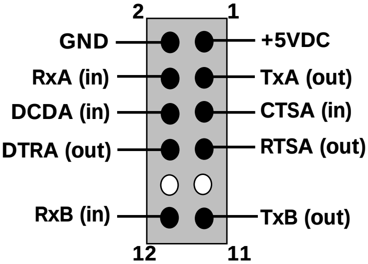
When J1 is closed, the input voltage required is 5 VDC (+ - 5%). When open, input voltage is 6-9 VDC (nominal 6VDC). The 3-pin header next to J1 is the second serial port.

Figure 2-6: Pinout Configurations

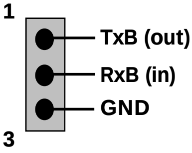
CON1 Serial Port



CON2 TTL Interface



CON4 Serial Connector



3: CoBox-Mini

Overview

The CoBox-Mini can be designed as an attachment to a PCB board. The serial interface is accomplished via TTL connectors which include transmit, receive, and full handshaking. UTP or AUI network interfaces can be utilized. The CoBox-Mini supports network speeds of 10Mbps.

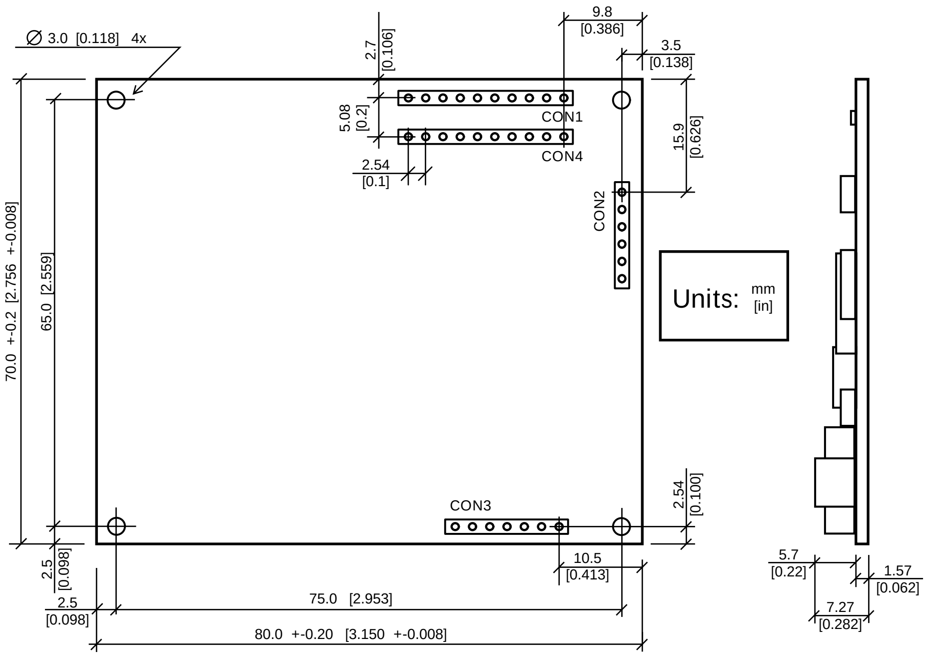
Both the length and orientation of the CoBox-Mini's interface pins can be specified to fit your product's architectural requirements. The CoBox-Mini requires 5 volts DC of regulated power with maximum current of 200mA for normal operation.

The CoBox-Mini's well-developed IP firmware supports protocols such as ARP, UDP, TCP, BOOTP, Telnet, ICMP, SNMP, DHCP, TFTP, and HTTP, as well as other custom protocols. The CoBox-Mini also supports a variety of user-configurable options such as buffer control and packetization, which make it easy to use in most applications.

Note: *This manual documents the present version (Rev 2) of the CoBox-Mini. This latest version of the CoBox-Mini is distinguished from the earlier version by its dual TTL serial ports (CON1 and CON4). The earlier version had one TTL serial port.*

Board Layout

Figure 3-1: CoBox-Mini Board Layout



Connectors

The CoBox-Mini has four connectors:

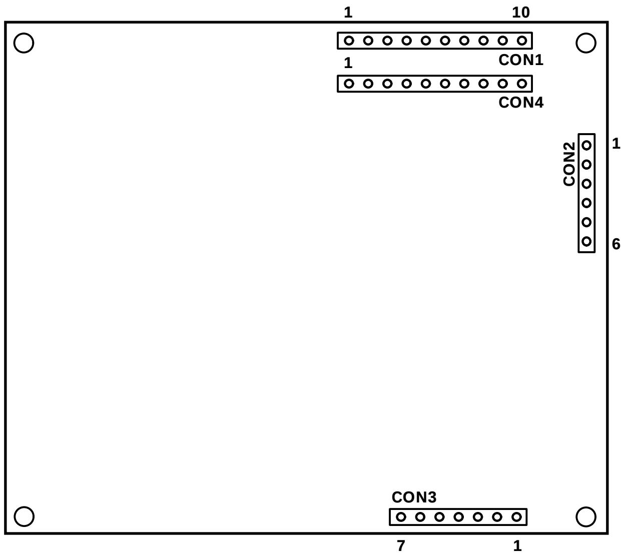
1. TTL serial port (CON1)
2. AUI connector (CON2)
3. UTP connector (CON3)
4. TTL serial port 2 and full handshaking signals (CON4)

At the time of ordering, each connector can be specified with the following pin configurations:

- Not present
- Top mounted, length of 5.46 mm or 6.76 mm
- Bottom mounted, length of 5.46 mm or 6.76 mm

Contact Lantronix or visit our Web site (www.lantronix.com) for information about ordering Device Servers with various connector configurations.

Figure 3-2: CoBox-Mini Connector Layout



The Embedded Integration Kit includes the CoBox-Mini embedded device server. Refer to the following table for a listing of its pinouts.

Table 3-1: CoBox-Mini Connector Pinouts

CON1 TTL Serial Port		CON4 TTL Serial Port with Full Handshaking		CON2 AUI Connector		CON3 10BASE-T Connector		
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**RJ45**
1	+5VDC	1	DTRA (output)	1	CD+	1	Tx+	1
2	GND	2	GND	2	CD-	2	Tx-	2
3	TxA (output)	3	RxB (input)	3	Rx+	3	Rx+	3
4	RxA (input)	4	TxB (output)	4	Rx-	4	Rx-	6
5	$\overline{\text{RESET}}$ (pull low to reset)	5	CTSA (input)	5	Tx+	5	None	
6	DCDA (input)	6	CTSB (input)	6	Tx-	6	LED 4 (Link)	
7	LED1 (Channel 1)	7	RTSA (output)			7	+5VDC (LED)	
8	LED2 (Channel 2)	8	DTRB (output)					
9	LED3 (Diagnostic)	9	DCDB (input)					
10	+5VDC (for LEDs)	10	RTSB (output)					
A = Port (Channel) 1		B = Port (Channel) 2						

Status LEDs

The CoBox-Mini has four status LEDs: serial port (Channel) 1 status, serial port (Channel) 2 status, diagnostics, and network link status. See the following table for a complete description of LED functions and pinout locations.

Table 3-2: CoBox-Mini Status LEDs

LED	Description	Location	Integration Kit LED Functions
1	Serial Port (Channel) 1 Status	CON1, Pin 7	Steady green to indicate Serial Port (Channel) 1 is *idle*. Blinking green to indicate Channel 1 is connected to the network and *active*.
2	Serial Port (Channel) 2 Status	CON1, Pin 8	Steady yellow to indicate Channel 2 is *idle*. Blinking yellow to indicate Channel 2 is connected to the network and *active*.
3	Diagnostic	CON1, Pin 9	Steady or blinking red in combination with the green (Channel 1) LED to indicate diagnostics and error detection. Red solid, green (Channel 1) blinking: 1x: EPROM checksum error 2x: RAM error 3x: Network controller error 4x: EEPROM checksum error 5x: Duplicated IP address on the network* 6x: Software does not match hardware* Red blinking, green (Channel 1) blinking: 4x: Faulty network connection* 5x: No DHCP response received*
4	Network Link Status	CON3, Pin 6	Lights solid green to indicate the network port is connected to the network.
	+5VDC	CON1, Pin 10	Power for Channel 1, Diagnostics, and Channel 2 LEDs
	+5VDC	CON3, Pin 7	Power for Link LED
*non-fatal error			

Product Information Label

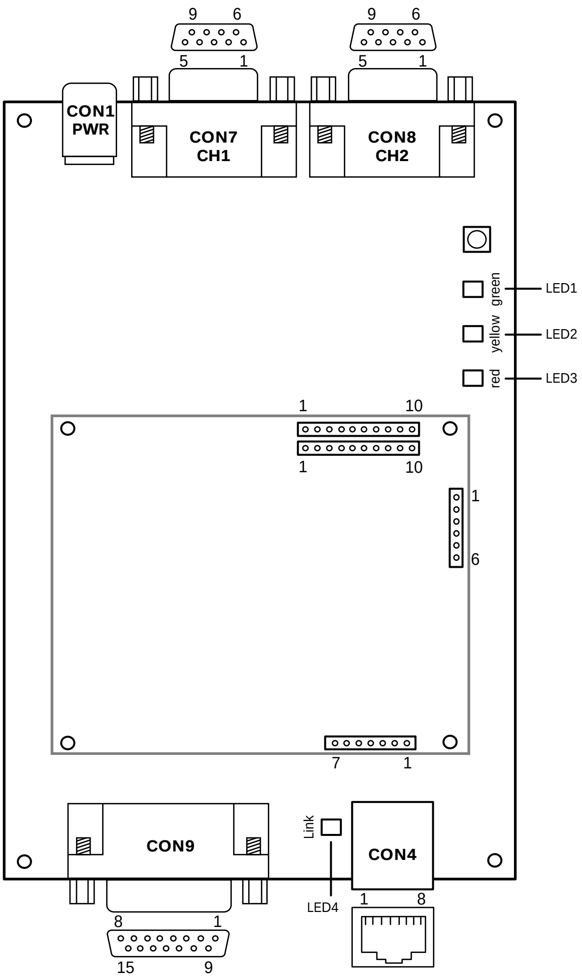
The CoBox-Mini ships with a product information label that can be removed from the unit and affixed to the host device. The product label contains information about your specific unit, such as its bar code, serial number, product ID (name), product description, and Ethernet address (also referred to as hardware address or MAC address).

Test Bed

The CoBox-Mini Embedded Integration Kit (part number Mini-Kit) includes a test bed (carrier board) that provides serial and network connections to the Device Server. The test bed contains diagnostic LEDs, a voltage regulator, a power supply circuit, a reset switch, TTL to RS-232 and RS-232 to TTL conversion hardware.

The test bed allows software engineers to immediately begin developing and testing software applications for the Device Server, rather than delaying the process until the hardware interface for their product is complete.

Figure 3-3: CoBox-Mini Test Bed



Test Bed Connectors

The CoBox-Mini test bed has five external connectors:

1 CON4 (a 10Base-T RJ45 Connector)

2 CON7 (Serial Port 1or Channel 1)

3 CON8 (Serial Port 2 or Channel 2)

4 CON9 (an AUI Connector)

5 CON1 (a 9-30 VDC power supply connector) (also accepts 9 -25 VAC)

Table 3-3: CoBox-Mini Test Bed Connector Pinouts

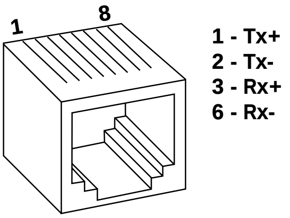
CON4 10Base-T (RJ45)		CON7 Serial Port (Channel) 1[a]		CON8 Serial Port (Channel) 2[b]		CON9 AUI Connector	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	Tx+	1	DCDA (input)	1	DCDB (input)	1,4,6,11,14	GND
2	Tx-	2	RxA (input)	2	RxB (input)	2	COL+
3	Rx+	3	TxA (output)	3	TxB (output)	3	Tx+
4	Decoupling	4	DTRA (output)	4	DTRB (output)	5	Rx+
5	Decoupling	5	GND	5	GND	7	None
6	Rx-	6	None	6	None	8	None
7	Decoupling	7	RTSA (output)	7	RTSB (output)	9	COL-
8	Decoupling	8	CTSA (input)	8	CTSB (input)	10	Tx-
		9	None	9	None	12	Rx-
						13	+12VDC[c]
						15	None

a. "Serial Port (Channel) 1" is also designated as "A".
b. "Serial Port (Channel) 2" is also designated as "B".
c. +12VDC is achievable if the voltage is higher than 16VDC or 13 VAC.

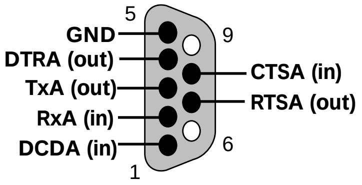
Note: *CON1 is a PCB-mounted, center-positive 9-30VDC power supply connector.*

Figure 3-4: Pinout Configurations

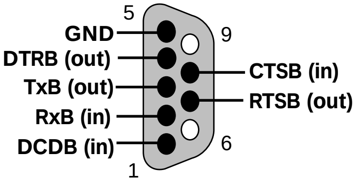
CON4 RJ45 Connector



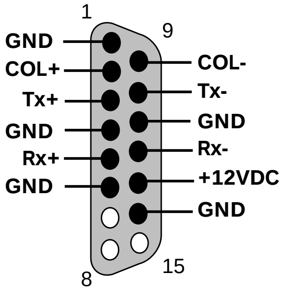
CON7 Serial Connector



CON8 Serial Connector



CON9 AUI Connector



4: CoBox-Mini100

Overview

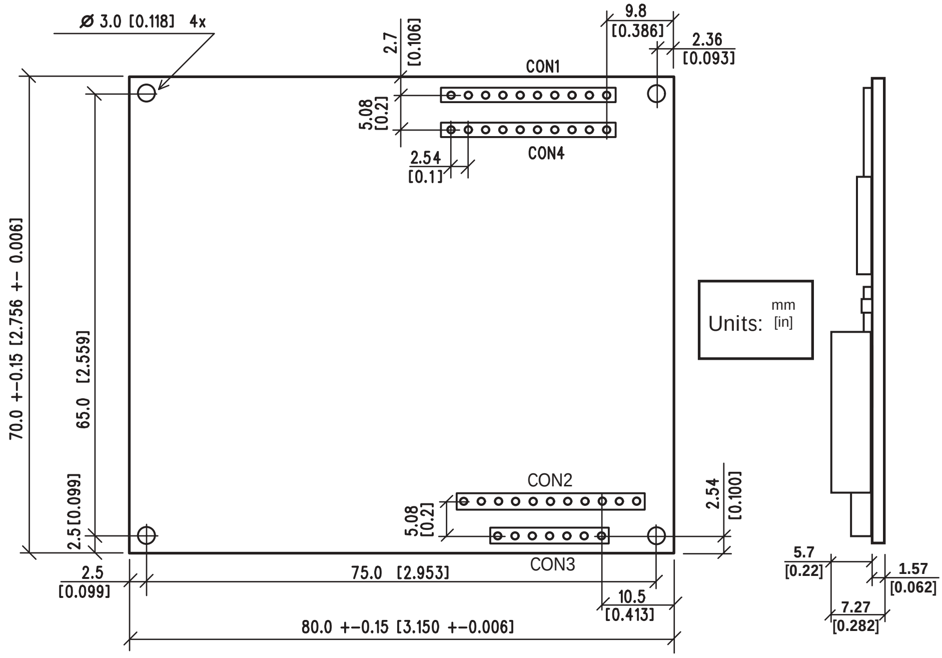
The CoBox-Mini100 can be designed as an attachment to a PCB board design. Serial interface is accomplished via TTL connector which includes transmit, receive, and hardware handshaking. The CoBox-Mini100 supports network speeds of 10Mbps or 100Mbps.

Both the length and orientation of the CoBox-Mini100's interface pins can be specified to fit your product's architectural requirements. The CoBox-Mini100 requires 5 volts DC of regulated power with maximum current of 250mA for normal operation.

The CoBox-Mini100's well-developed IP firmware supports protocols such as ARP, UDP, TCP, BOOTP, Telnet, ICMP, SNMP, DHCP, TFTP, and HTTP, as well as other custom protocols. The CoBox-Mini100 also supports a variety of user-configurable options such as buffer control and packetization, which make it easy to use in most any application.

Board Layout

Figure 4-1: CoBox-Mini100 Board Layout



Connectors

The CoBox-Mini100 has three connectors:

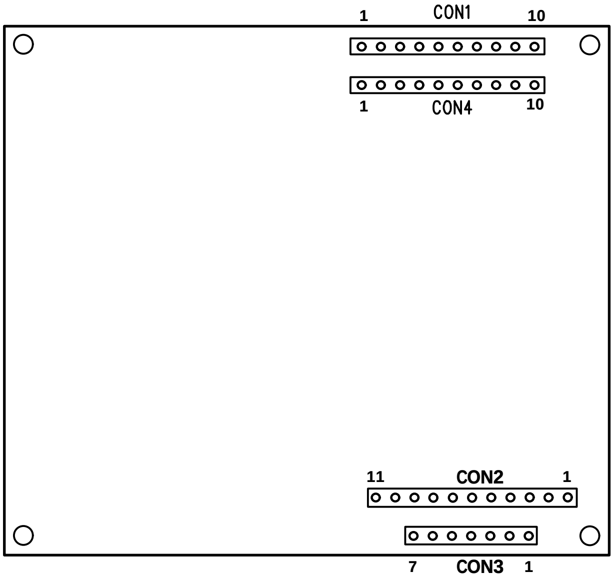
1. TTL serial ports (CON1 and CON4)
2. 10/100BASE-T connector (CON3)
3. 10/100BASE-TX connector (CON2)

At the time of ordering, each connector can be specified as follows:

- Not present
- Top mounted, length of 5.46 mm or 6.76 mm
- Bottom mounted, length of 5.46 mm or 6.76 mm

Contact Lantronix or visit our Web site (www.lantronix.com) for information about ordering Device Servers with various connector configurations.

Figure 4-2: CoBox-Mini100 Connector Layout



The Embedded Integration Kit includes the CoBox-Mini100 embedded device server. Refer to the following table for a listing of its pinouts.

Table 4-1: CoBox-Mini100 Connector Pinouts

CON1 TTL Serial Port		**CON4** TTL Serial Port		***CON3** 10/100BASE-T Connector		
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**RJ45**
1	+5VDC	1	DTRA (output)	1	Tx+	1
2	GND	2	GND	2	Tx-	2
3	TxA (output)	3	RxB (input)	3	Rx+	3
4	RxA (input)	4	TxB (output)	4	Rx-	6
5	$\overline{\text{RESET}}$ (pull low to reset)	5	CTSA (input)	5	Decoupling (Shield)	
6	DCDA (input)	6	CTSB (input)	6	LED4 (Link)**	
7	LED1 (Channel 1)	7	RTSA (output)	7	+5VDC (for LED4)	
8	LED2 (Channel 2)	8	DTRB (output)			
9	LED3 (Diagnostics)	9	DCDB (input)			
10	+5VDC (for LEDs 1-3)	10	RTSB (output)			

***CON2** 10/100BASE-TX Connector		
Pin	**Signal**	**RJ45**
1	Tx+	1
2	Tx-	2
3	Rx+	3
4	Decoupling	4
5	Decoupling	5
6	Rx-	6
7	Decoupling	7
8	Decoupling	8
9	Decoupling (Shield)	
10	LED5 (speed 100MBit)**	
11	+5VDC (for LED5)	

***For Ethernet connection, either CON2 or CON3 should be used. The use of CON2 is recommended for harsh and noisy environments.**

****LED4 indicates 10M network speed, and LED5 indicates 100M network speed. Both LED4 and LED5 can be used when either CON2 or CON3 is used.**

Status LEDs

The CoBox-Mini100 has five status LEDs: serial port (Channel) 1 status, serial port (Channel) 2 status, diagnostics, and two network link LEDs. See the following table for a complete description of LED functions and pinout locations.

Table 4-2: CoBox-Mini100 Status LEDs

LED	Description	Location	Integration Kit Functions
1	Serial Port (Channel) 1 Status	CON1, Pin 7	Lights solid green to indicate Serial Port (Channel) 1 is *idle*. Blinks green to indicate Channel 1 is connected to the network and *active*.
2	Serial Port (Channel) 2 Status	CON1, Pin 8	Lights solid yellow to indicate Channel 2 is *idle*. Blinks yellow to indicate Channel 2 is connected to the network and *active*.
3	Diagnostic	CON1, Pin 9	Blinks or lights solid red in combination with the green (Channel 1) LED to indicate diagnostics and error detection. Red solid, green (Channel 1) blinking: 1x: EPROM checksum error 2x: RAM error 3x: Network controller error 4x: EEPROM checksum error 5x: Duplicated IP address on the network 6x: Software does not match hardware Red blinking, green (Channel 1) blinking: 4x: Faulty network connection* 5x: No DHCP response received*
4	10M Network Link Status	CON3, Pin 6	Lights solid green to indicate the network port is connected to the network with 10MBit speed.
5	100M Network Link Status	CON2, Pin2	Lights solid green to indicate the network port is connected to the network with 100MBit speed.
***non-fatal error**			

Product Information Label

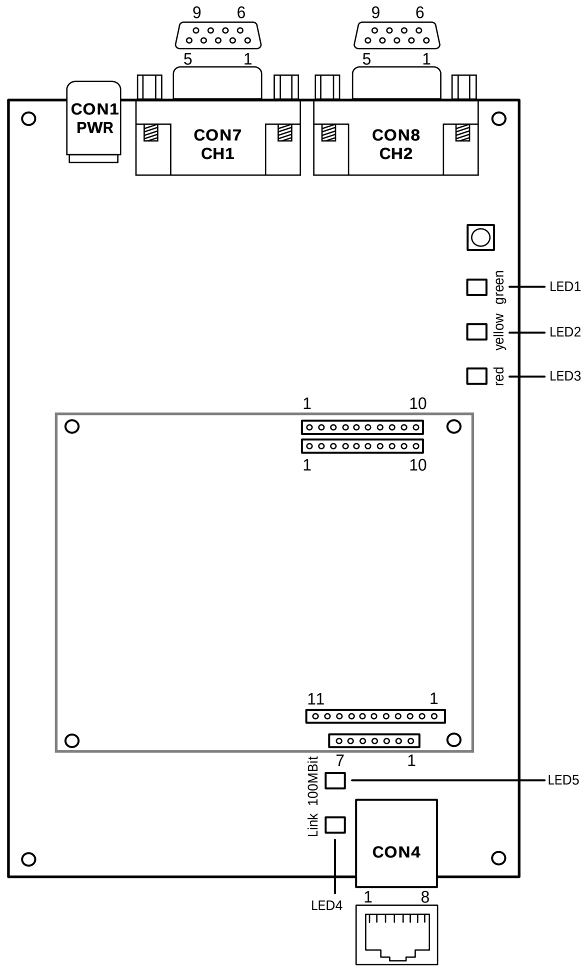
The CoBox-Mini100 ships with a product information label that can be affixed to the host device. The product label contains information about your specific unit, such as its bar code, serial number, product ID (name), product description, and Ethernet address (also referred to as hardware address or MAC address).

Test Bed

The CoBox-Mini100 Embedded Integration Kit (part number Mini100-Kit) includes a test bed (carrier board) that provides serial and network connections to the Device Server. The test bed contains diagnostic LEDs, a voltage regulator, a power supply circuit, a reset switch, TTL to RS-232 and RS-232 to TTL hardware conversion.

The test bed allows software engineers to immediately begin developing and testing software applications for the Device Server, rather than delaying the process until the hardware interface for their product is complete.

Figure 4-3: CoBox-Mini100 Test Bed



Test Bed Connectors

The CoBox-Mini100 test bed has five external connectors: CON4 (a 10/100Base-TX RJ45 connector), CON7 (serial port 1 or Channel 1), CON8 (serial port 2 or Channel 2), CON1 (a 9-30 VDC power supply connector).

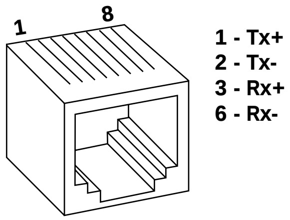
Table 4-3: CoBox-Mini100 Test Bed Connector Pinouts

CON4 10/100Base-TX (RJ45)		CON7 Serial Port (Channel) 1[a]		CON8 Serial Port (Channel) 2[b]	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	Tx+	1	DCDA (input)	1	DCDB (input)
2	Tx-	2	RxA (input)	2	RxB (input)
3	Rx+	3	TxA (output)	3	TxB (output)
4	Decoupling	4	DTRA (output)	4	DTRB (output)
5	Decoupling	5	GND	5	GND
6	Rx-	6	None	6	None
7	Decoupling	7	RTSA (output)	7	RTSB (output)
8	Decoupling	8	CTSA (input)	8	CTSB (input)
		9	None	9	None

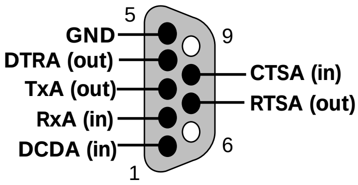
a. “CON7 Serial Port (Channel) 1” is also designated as “A”.
b. “CON8 Serial Port (Channel) 2” is also designated as “B”.

Figure 4-4: Pinout Configurations

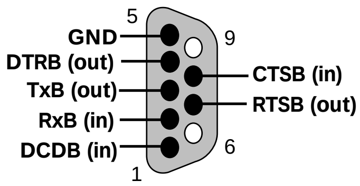
CON4 RJ45 Connector



CON7 Serial Connector



CON8 Serial Connector



5: UDS-10B

Overview

The UDS-10B (part number UDS-10B) is designed to be easily and quickly integrated into products. Serial interface is accomplished via a female DB25 (DCE) connector which provides the necessary signals for both RS232 and RS422/RS485 interfaces.

In addition, an RJ45 (10BASE-T) connector is available for Ethernet access. It will support network speeds of 10Mbps.

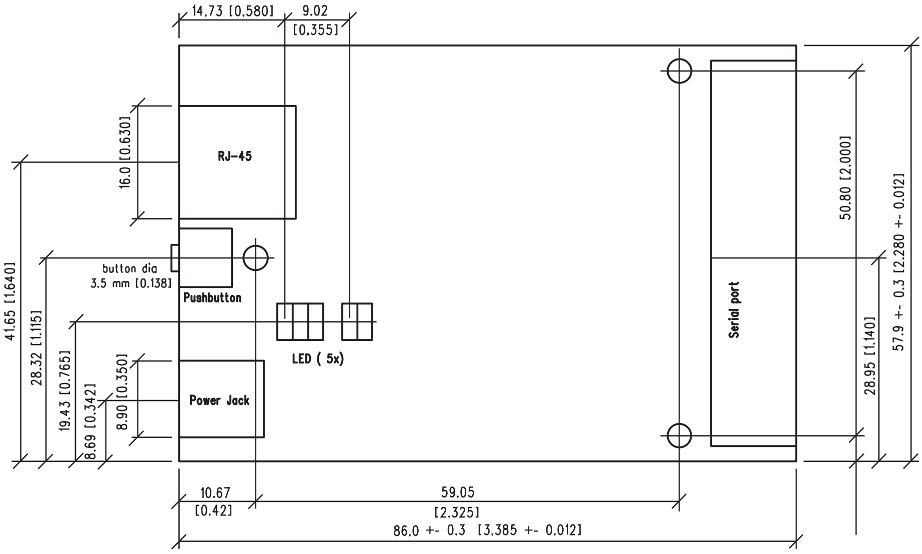
The UDS-10B board comes with various power supply options:

- a power jack for 9-30 VDC or 9-24 VAC with maximum 140mA.
- power via the DB25 inputs at either the regulated +5 VDC with maximum 200mA or the unregulated 9-30 VDC.

The UDS-10B's well-developed IP firmware supports protocols such as ARP, UDP, TCP, Telnet, ICMP, SNMP, BOOTP, DHCP, TFTP, HTTP, and custom protocols.

Board Layout

Figure 5-1: UDS-10B Board Layout

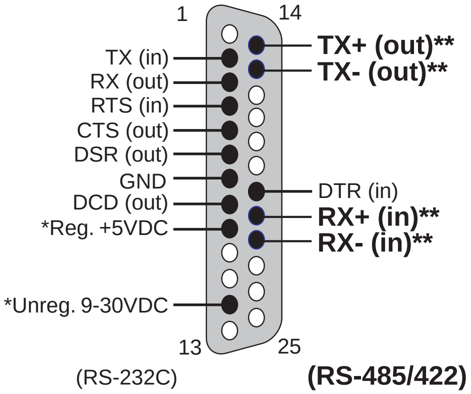


Units: mm [in]

Serial Interface

The UDS-10B has a female DB25 DCE serial port that supports RS-232 and RS-485/422 serial standards (selectable) up to 115.2 Kbps.

Figure 5-2: UDS-10B Serial Interface



*The Device Server can alternately be powered up via the serial port using one of these pins.

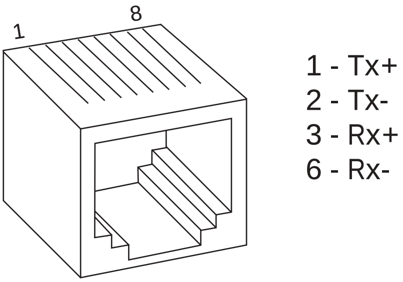
**The minus sign (-) is sometimes represented as A (e.g., TXA). The plus sign (+) is sometimes represented as B (e.g., TXB).

Note: *For RS-485 2-wire functionality, pins 14 and 21 and pins 15 and 22 must be connected to each other.*

Network Interface

The UDS-10B has a 9-30VDC power plug, a reset switch, and an RJ45 (10Base-T) Ethernet port that supports 10 Mbps.

Figure 5-3: UDS-10B Network Interface



LEDs

The UDS-10B has five LEDs: Link, Network Transmit/Receive, Collision, Diagnostics, and Status.

Figure 5-4: UDS-10B LEDs

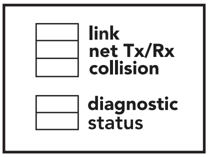


Table 5-1: UDS-10B LEDs

LED	Function
Link	Lights solid green to indicate network port is connected to the network.
Net Tx/Rx	Blinks yellow to indicate network packets are transmitting and receiving.
Collision	Blinks red to indicate network collisions.
Diagnostic	Blinks or glows red in combination with the green Status LED to indicate diagnostics and error detection. (This code should only appear after power up. Even though the Device Server is going into operation mode, the problem will potentially persist.) Red (diagnostic) solid, green (Status LED) blinking: 1x: EPROM checksum error 2x: RAM error 3x: Network controller error 4x: EEPROM checksum error 5x: Duplicated IP address on the network* 6x: Software does not match hardware* Red (diagnostic) blinking, green (Status LED) blinking: 4x: Faulty network connection* 5x: No DHCP response received*
Status	Lights solid green to indicate serial port is *idle*. Blinks green to indicate serial port is *active*.
*non-fatal error	

Product Information Label

A product information label is located on of the UDS-10B, and contains the following information about your specific unit:

- Bar Code
- Serial Number
- Product ID (name)
- Product Description
- Ethernet Address (also referred to as Hardware Address or MAC Address)

6: Getting Started

This chapter covers the required steps to get the Device Server on-line and working. There are two basic methods used to log into the Device Server and setup the IP address:

- Network Port Login: Make a Telnet connection to the network port (9999).
- Serial Port Login: Connect a terminal or a PC running a terminal emulation program to the Device Server's first serial port (CH 1).

It is important to consider the following points before logging into and configuring the Device Server:

- The Device Server's IP address must be configured before a network connection is available.
- Only one person at a time may be logged into the network port. This eliminates the possibility of several people simultaneously attempting to configure the Device Server.
- Network port logins can be disabled. The system manager will not be able to access the unit. This port can also be password protected.
- Only one terminal at a time can be connected to the serial port. (In RS-485 mode, the Device Server is capable of multidrop connections.)

Default IP Address

The Device Server ships with a default IP address set to 0.0.0.0, which automatically enables DHCP within the Device Server.

Provided a DHCP server exists on the network, it will supply the Device Server with an IP address, gateway address, and subnet mask when the Device Server boots up. (If no DHCP server exists, the Device Server will respond with a diagnostic error: the red Diagnostic LED blinks continuously and the green Status LED blinks five times.)

This IP address will **not** appear in the Device Server's configuration screens; however, if you enter Monitor Mode from the serial port with network connection enabled (see Monitor Mode on page B-1), and issue the **NC** (Network Connection) command, you will see the Device Server's IP configuration.

AutoIP

AutoIP allows a Device Server to obtain an address in a network that does not have a DHCP server. (Windows 98 and 2000 also support AutoIP.)

AutoIP assigns a random valid address to the Device Server in the range of 169.254.x.1 to 169.254.x.1(x can be between 0 and 255). This range of IP addresses is not to be used over the Internet. If a Device Server has not been configured manually and cannot find a DHCP server, it automatically chooses an address from the reserved range. The Device Server then uses the Address Resolution Protocol (ARP) to send out a request asking whether any node is using that same address. If another node is using the same address, the Device Server assigns another IP address, reboots, and repeats the sequence.

Note: *AutoIP-enabled Device Servers are constantly looking for DHCP servers. If a DHCP server becomes available on the network, the AutoIP-enabled Device Server switches to the DHCP server-provided address, and the unit reboots. If the DHCP server exists but denies the Device Server an IP address, the Device Server does not attach to the network, but waits and retries.*

AutoIP allows a small network of AutoIP-enabled devices to be set up without any need for a DHCP server or static IP addresses.

AutoIP can be disabled by setting the IP address to 0.0.1.0. The 1 in the third octet is the disabling factor.

Setting the IP Address

The Device Server's IP address must be configured before a network connection is available. If the IP address was not set automatically via DHCP, set it now using a network or serial port login and the setup (configuration) menu.

DHCP Naming

The DHCP name of the Device Server can be changed. The default name of the Device Server is Cxxxxxx, where xxxxxx is the last 6 digits of the Mac address. This option can be changed to LTXdd, where 0.0.0.dd is the IP address assigned (dd should be a number between 1 and 99). For example, if the IP address is set to 0.0.0.5, the resulting DHCP name is LTX05.

A DHCP name of the customer's own choosing (up to 8 characters) can also be designed. This option can be set in the server configuration menu.

Figure 6-1: Server Configuration Menu

```
Change DHCP device name (LTRX) ? (N) Y
Enter new DHCP device name : LTRXYES
```

Network Port Login

The ARP method is available under UNIX and Windows-based systems. The Device Server will set its address from the first directed TCP/IP packet it receives.

1 On a **UNIX** host, create an entry in the host's ARP table using the intended IP address and the hardware address of the Device Server, which is found on the product label.

Figure 6-2: ARP on UNIX

```
arp -s 191.12.3.77 00:20:4a:xx:xx:xx
```

In order for the ARP command to work on **Windows**, the ARP table on the PC must have at least one IP address defined other than its own. If the ARP table is empty, the command will return an error message. Type `ARP -A` at the DOS command prompt to verify that there is at least one entry in the ARP table.

If the local machine is the only entry, ping another IP address on your network to build a new entry in the ARP table; the IP address must be a host other than the machine on which you are working. Once there is at least one additional entry in the ARP table, use the following command to ARP an IP address to the Device Server:

Figure 6-3: ARP on Windows

```
arp -s 191.12.3.77 00-20-4a-xx-xx-xx
```

2 Now open a Telnet connection to port 1. The connection will fail quickly (3 seconds), but the Device Server will temporarily change its IP address to the one designated in this step.

Figure 6-4: Telnet to Port 1

```
telnet 191.12.3.77 1
```

3 Finally, open a Telnet connection to port 9999 and set all required parameters.

Figure 6-5: Telnet to Port 9999

```
telnet 191.12.3.77 9999
```

Note: *This IP address is temporary and will revert to the default value when the Device Server's power is reset, unless you log into the Device Server and store the changes permanently. Refer to Chapter 7 for instructions on permanently configuring the IP address.*

Serial Port Login

1 Connect a console terminal or PC running a terminal emulation program to the Device Server's first serial port (CH 1). The configuration serial port settings are 9600 baud, 8 bits, no parity, 1 stop bit.

2 To enter Setup (configuration) Mode, cycle the Device Server's power (power off and back on). After power-up the self-test begins and the red Diagnostic LED starts blinking. You have one second to enter three lowercase "x" characters.

Note: *The easiest way to enter Setup Mode is to hold down the "x" key at the terminal while powering up the Device Server.*

3 Select **0** (Server Configuration) and follow the prompts until you get to IP address.

4 Enter the new IP address.

5 Select **9** to save the configuration and exit Setup Mode.

The Device Server performs a power reset.

Using Another Device Server

To access another device server remotely, enter the SI command in Moniro Mode. Refer to Appendix B which describes how to get to Monitor Mode. Refer to table B-1 which lists all Monitor Mode commands.

7: Configuration

Certain parameters must be configured before the Device Server can function on a network. The Device Server can be locally or remotely configured using the following procedures:

- Use a Telnet connection to configure the unit over the network.
- Use a terminal or terminal emulation program to access the first serial port (CH 1) locally.
- Use a standard Web browser to access the Device Server's internal Web pages and configure the unit over the network. This is the easiest and preferred method.
- Use UDP datagrams to configure the unit over the network.
- Use a Hex file to configure the unit over the network.

The Device Server's configuration is stored in nonvolatile memory (NVRam) and is retained without power. The configuration can be changed at any time. The Device Server performs a reset after the configuration has been changed and stored.

Network Configuration Using a Web Browser

If your Device Server already has an IP address (see Chapter 6, *Getting Started*), you can log into it using a standard Web browser with Java enabled.

1 Type the Device Server's IP address into the Web browser's URL (Address/ Location) field.

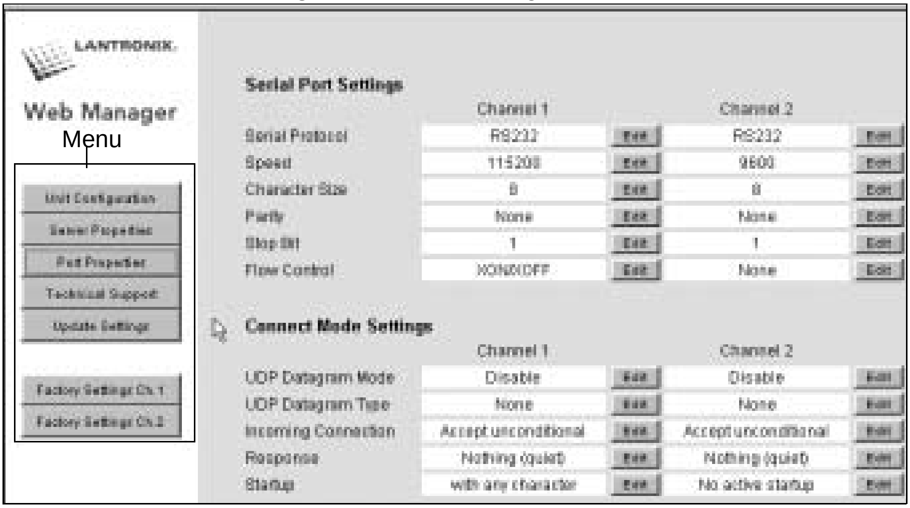
Figure 7-1: Web Browser Login

Once you have connected to the Device Server, you will see the Lantronix Web Manager interface.

2 Select **Connect** to login and gain access to the configuration menu.

3 Use the menu to navigate to subpages where you can configure server settings.

Figure 7-2: Web Manager Interface



Network Configuration Using a Telnet Connection

To configure the Device Server over the network, establish a Telnet connection to port **9999**. From the Windows Start menu, click **Run** and type the following command, where x.x.x.x is the IP address and 9999 is the Device Server's fixed network configuration port number.

Figure 7-3: Network Login Using Telnet

```
telnet x.x.x.x 9999
```

Note: *Be sure to include a space between the IP address and 9999.*

Serial Configuration

For local configuration, a terminal or a PC running a terminal emulation program can be connected to the Device Server's serial port. The terminal (or emulation) should be configured for 9600 baud, 8-bit, no parity, 1 stop bit, and no flow control.

To enter Setup (configuration) Mode, cycle the Device Server's power (power off and back on). After power-up, the self-test begins and the Diagnostic and Status LEDs start blinking. You must enter three lowercase "x" characters (**xxx**) within one second after powering up in order to start the configuration mode.

Note: *The easiest way to enter Setup Mode is to hold down the "x" key on your keyboard while powering up the Device Server.*

Network Configuration Using UDP

The Device Server can also be configured or queried over the network using UDP datagrams. For more information, see *Network Configuration Using UDP* on page E-1.

Network Configuration Using a Hex File

When configuring a number of Device Servers identically, it is useful to create a template setup record. For more information, see *Network Configuration Using a Hex File* on page E-3.

Configuration Parameters

After entering Setup Mode (confirm by pressing **Enter**), you can configure the parameters by entering one of the numbers on the Change Setup Menu, or you can confirm default values by pressing **Enter**. Be sure to store the new configurations when you are finished. The Device Server will then perform a power reset.

Figure 7-4: Setup (Configuration) Mode Screen

```
Serial Number 5411011  MAC address 00:20:4A:54:10:03
Software version V4.30 (001206)
Press Enter to go into Setup Mode


*** basic parameters
Hardware: Ethernet Autodetect
IP addr 192.168.1.10, gateway 192.168.001.001,netmask
255.255.255.000

***************** Security *****************
Telnet setup is        enabled
TFTP download is       enabled
Port 77FEh is          enabled
Web setup is           enabled
Enhanced password is disabled
Encryption is          disabled

***************** Channel 1 *****************
Baudrate 9600, I/F Mode 4C, Flow 00
Port 10001
Remote IP Adr: --- none ---, Port 00000
Connect Mode : C0  Disconn Mode: 00
Flush   Mode : 00




Change Setup  : 0 Server configuration
                1 Channel 1 configuration
                6 Security
                7 Factory defaults
                8 Exit without save
                9 Save and exit                Your choice ?
```

Server Configuration

Select **0** to configure the Device Server's basic parameters.

IP Address

The IP address must be set to a unique value in your network. See Appendix E for more information about IP Addressing.

Note: *The Device Server cannot connect to the network if the assigned IP address is already in use by another device.*

Gateway Address

The gateway address, or router, allows communication to other LAN segments. The gateway address should be the IP address of the router connected to the same LAN segment as the Device Server.

Note: *The gateway address must be within the local network.*

Subnet Mask

A netmask defines the number of bits taken from the IP address that are assigned for the host section.

Note: *Class A: 24 bits; Class B: 16 bits; Class C: 8 bits.*

The Device Server prompts for the number of host bits to be entered, then calculates the netmask, which is displayed in standard decimal-dot notation when the saved parameters are displayed (for example, 255.255.255.0).

Telnet Configuration Password

Setting the Telnet configuration password prevents unauthorized access of the setup menu via a Telnet connection to port 9999. The password is limited to 4 characters. An enhanced password setting of 16 characters is available under Security Settings.

Note: *No password is required to access the setup menu via the serial connection.*

DHCP Naming

The DHCP name of the Device Server can be changed. The default name of the Device Server is Cxxxxxx, where xxxxxx is the last 6 digits of the Mac address. This option can be changed to LTXdd, where 0.0.0.dd is the IP address assigned (dd should be a number between 1 and 99). For example, by setting the IP address to 0.0.0.5, the resulting DHCP name is LTX05.

A DHCP name of the customer's own choosing (up to 8 characters) can also be designated. This option can be set in the server configuration menu.

Figure 7-5: Server Configuration Menu

```
Change DHCP device name (LTRX) ? (N) Y
Enter new DHCP device name : LTRXYES
```

Serial Channel (Port) Configuration

Select **1** to configure the Device Server's channel-specific parameters.

Baud Rate

The Device Server and attached serial device, such as a modem, must agree on a speed or baud rate to use for the serial connection. Valid baud rates are 300, 600, 1200, 2400, 4800, 9600 (default), 19200, 38400, 57600 (only on AMD-based hardware platforms), and 115200 bits per second.

Interface Mode

The Interface (I/F) Mode is a bit-coded byte entered in hexadecimal notation. Use the following table to select Interface Mode settings:

Table 7-1: Interface Mode Options

Option	Bit 7	6	5	4	3	2	1	0
RS-232C							0	0
RS-422/485							0	1
RS-485 2-wire							1	1
7 Bit					1	0		
8 Bit					1	1		
No Parity			0	0				
Even Parity			1	1				
Odd Parity			0	1				
1 Stop bit	0	1						
2 Stop bits	1	1						

Note: *Embedded units require external serial drivers to support RS-232 or RS-485.*

The following table demonstrates how to build some common Interface Mode settings:

Table 7-2: Common Interface Mode Settings

Option	Binary	Hex
RS-232C, 8-bit, No Parity, 1 stop bit	0100 1100	4C
RS-232C, 7-bit, Even Parity, 1 stop bit	0111 1000	78
RS-485 2-Wire, 8-bit, No Parity, 1 stop bit	0100 1111	4F
RS-422, 8-bit, Odd Parity, 2 stop bits	1101 1101	DD

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Flow Control

Flow control sets the local handshake method for stopping serial input/output. Generally, flow control is not required if the connection is used to pass a blocked protocol with block sizes less than 1k (ACK/NAK) and/or speeds up to 19200. Use the following table to select

Table 7-3: Flow Control Options

Option	**Hex**
No flow control	00
XON/XOFF flow control	01
Hardware handshake with RTS/CTS lines	02
XON/XOFF pass characters to host	05

Port Number

This setting represents the source port number in TCP connections, and is the number used to identify the channel for remotely initiating connections. Range: 1024-65535.

Note: *Port numbers 14000-14009 are reserved exclusively for use with the Comm Port Redirector application (see Comm Port Redirector on page 8-1).*

Port number 9999 is reserved exclusively for Telnet setup.

Port number 77FEh is reserved exclusively for configuration.

If the UDP mode is selected, the port number functions as the UDP source port number for outgoing datagrams. Datagrams sent to the Device Server with this port number are received to this channel.

Connect Mode

Connect Mode defines how the Device Server makes a connection, and how it reacts to incoming connections over the network. Use the following table to select Connect Mode options:

Table 7-4: Connect Mode Options

Option **Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Incoming (network) Connection								
Never accept incoming	0	0	0					
Accept incoming with DTR*	0	1	0					
Accept unconditional	1	1	0					
Response								
Nothing (quiet)				0				
Character response (C=conn, D=disconn, N=unreachable)				1				
Startup (on serial connection)								
No active startup					0	0	0	0
With any character					0	0	0	1
With active DTR*					0	0	1	0
With CR (0x0D) only					0	0	1	1
Manual connection					0	1	0	0
Autostart					0	1	0	1
Datagram Type								
Directed UDP					1	1	0	0
Modem Mode								
With Echo				1	0	1	1	0
Without Echo				0	0	1	1	0

***On DTE devices, use DCD (input) instead of DTR. DTR is an output signal on a DTE serial port. The DTE devices are the CoBox Micro, CoBox Mini, and CoBox Mini100.**

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Manual Connection

If manual connection startup is configured (**C** + address/port), only the portion not provided in the command string is used. In manual mode, the last byte of the address must be provided.

For example, if the Device Server's configured remote IP address is 129.1.2.3 and the TCP port number is 1234:

Table 7-5: Manual Connection Address Example

Command String	Result
C121.2.4.5/1	Complete override; connection is started with host 121.2.4.5, port 1
C5	Connect to 129.1.2.5, port 1234
C28.10/12	Connect to 129.1.28.10, port 12

Autostart (Automatic Connection)

For the serial port, automatic TCP connection to a network node can be configured by setting the remote IP address and the TCP port number parameters. If automatic connection is selected, all parameters must be provided.

Datagram Type

When selecting this option, you will be prompted for the Datagram type. Enter **01** for directed UDP.

Modem (Emulation) Mode

In Modem Mode, the Device Server presents a modem interface to the attached serial device. It accepts AT-style modem commands, and handles the modem signals correctly.

Normally there is a modem connected to a local PC and a modem connected to a remote machine. A user must dial from the local PC to the remote machine, accumulating phone charges for each connection. Modem Mode allows you to replace modems with Device Servers, and to use an Ethernet connection instead of a phone call, without having to change communications applications and make potentially expensive phone calls.

Note: *If the Device Server is in Modem Mode and the serial port is idle, the Device Server can still accept network TCP connections to the serial port if Connect Mode is set to **C6** (no echo) or **D6** (echo).*

Modem Mode is selected by setting the connect mode to **06** (no echo) or **16** (echo).

In Modem Mode, echo refers to the echo of all of the characters entered in command mode; it does *not* mean to echo data that is transferred.Quiet mode (no echo) refers to the modem not sending an answer to the commands received.

To enter command mode:

There *must* be 1-second guardtime (no datatraffic) before sending +++.

There *must not* be a break longer that 1 second between +s.

There *must* be another 1 second guardtime after the last + is sent.

The Device Server acknowledges with an OK to indicate that it is in command mode.

Enter ATH and press **Enter**. It is echoed if echo is enabled. ATH is acknowledged by another OK.

Table 7-6: Modem Mode Commands

Command	Function
ATDTx.x.x.x,pppp or ATDTx.x.x.x/pppp	Makes a connection to an IP address (x.x.x.x) and a remote port number (pppp).
ATDTx.x.x.x	Makes a connection to an IP address (x.x.x.x) and the remote port number defined within the Device Server.
ATD0.0.0.0	Forces the Device Server into monitor mode if a remote IP address and port number are defined within the Device Server.
ATD	Forces the Device Server into monitor mode if a remote IP address and port number **are not** defined within the Device Server.
ATDx.x.x.x	Makes a connection to an IP address (x.x.x.x) and the remote port number defined within the Device Server.
ATH	Hangs up the connection (Entered as **+++ ATH**).

Note: *All other AT commands with Modem Mode set to* ***16*** *or* ***D6*** *acknowledge with an OK, but no action is taken.*

Remote IP Address

Selecting Autostart (automatic connection) in Connect Mode makes a connection to this IP address on the network. This is the destination IP address used with an outgoing connection.

Remote Port

The remote TCP port number must be set to use automatic connections. This parameter defines the port number on the target host to which a connection is attempted.

Note: *To connect an ASCII terminal to a host using the Device Server for login purposes, use the remote port number **23** (Internet standard port number for Telnet services).*

Disconnect Mode

Note: *In Disconnect Mode, DTR drop either drops the connection or is ignored. Use the following table to select Disconnect Mode Options:*

Table 7-7: Disconnect Mode Options

Option Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Disconnect with DTR drop[a]	1							
Ignore DTR[a]	0							
Telnet mode and terminal type setup[b]		1						
Channel (port) password[c]				1				
Hard disconnect[d]					0			
Disable hard disconnect					1			
State LED off with connection[e]								1
Disconnect with EOT (^D)[f]			1					

a. **DTR is an output signal on a DTE serial port. The DTE devices are the CoBox Micro, CoBox Mini, and CoBox Mini100. On DTE devices, use DCD (input) instead of DTR.**

b. **The CoBox will send the "Terminal Type" upon an outgoing connection.**

c. **A password is required for a connection to the serial port from the network.**

d. **The TCP connection will close even if the remote site does not acknowledge the disconnection.**

e. **When there is a network connection to or from the serial port, the state LED will turn off instead of blink.**

f. **When Ctrl D or Hex 04 are detected, the connection is dropped. Both Telnet mode and Disconnect with EOT must be enabled for Disconnect with EOT to function properly. Ctrl D will only be detected going from the serial port to the network.**

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Flush Mode (Buffer Flushing)

Using this parameter, you can control line handling and network buffers with connection startup and disconnect. You can also select between two different packing algorithms. Use the following table to select Flush Mode options:

Table 7-8: Flush Mode Options

Function Bit	7	6	5	4	3	2	1	0
Input Buffer (Serial to Network)								
Clear active connection (from serial)				1				
Clear passive connection (from network)			1					
Clear at disconnect		1						
Output Buffer (Network to Serial)								
Clear active connection (from network)								1
Clear passive connection (from serial)							1	
Clear at disconnect						1		
Alternate Packing Algorithm								
Enable	1							

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Pack Control

Alternate packing algorithm settings are enabled in Flush Mode. Use the following table to select Pack Control options:

Table 7-9: Pack Control Options

Option **Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Idle Time								
Force transmit: 12ms							0	0
Force transmit: 52ms							0	1
Force transmit: 250ms							1	0
Force transmit: 5sec							1	1
Trailing Characters								
None					0	0		
One					0	1		
Two					1	0		
Send Characters								
Semdchar define 2-Byte Sequence				1				
Send Immediately After Sendchars			1					

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Idle Time

Idle time to “Force transmit” defines how long the Device Server will wait between characters before sending accumulated characters, regardless of the recognition of send characters.

Trailing Characters

In some applications, CRC, Checksum, or other trailing characters follow the end-of-sequence character; this option helps to adapt frame transmission to the frame boundary.

Send Characters

If send characters are enabled, the Device Server interprets the sendchars as a 2-byte sequence; if not set, they are interpreted independently.

If “Send Immediately After Characters” is **not** set, any characters already in the serial buffer are included in the transmission after a “transmit” condition is found. If set, the Device Server sends immediately after recognizing the transmit condition (Sendchar or time-out).

Note: *A transmission might occur if status information needs to be exchanged or an acknowledgment needs to be sent.*

Inactivity (Disconnect) Time-out

Use this parameter to set an inactivity time-out. The connection is dropped if there is no activity on the serial line before the set time expires. Enter time in the following format: mm:ss, where “m” is the number of minutes and “s” is the number of seconds. To disable the inactivity time-out, enter **00:00**.

Send Characters

You can enter up to two characters in hexadecimal representation in the parameters “sendchar.” If a character received on the serial line matches one of these characters, it is sent immediately, along with any awaiting characters, to the TCP connection. This minimizes the response time for specific protocol characters on the serial line (for example, ETX, EOT, etc.). Setting the first sendchar to **00** disables the recognition of the characters.

Alternatively, the two characters can be interpreted as a sequence (see Pack Control on page 7-14).

Telnet Terminal Type

This parameter appears only if the terminal type option is enabled in Disconnect Mode (see Disconnect Mode on page 7-12). If set, you can use the terminal name for the Telnet terminal type. Enter only one name.

If the terminal type option is enabled, the Device Server also reacts to the EOR (end of record) and binary options, which can be used for applications like terminal emulation to IBM hosts.

Note: *The telnet terminal type password and the channel (port) password cannot be active at the same time.*

Channel (Port) Password

This parameter appears only if the channel (port) password option is enabled in Disconnect Mode (Disconnect Mode on page 7-12). If set, you can set a password for the serial port. The maximum length of the password is 16.

Note: *The Telnet terminal type password and the channel (port) password cannot be active at the same time.*

Telnet Configuration Password

Setting the Telnet configuration password prevents unauthorized access of the setup menu via a Telnet connection to port 9999. The password is limited to 4 characters.

Note: *No password is required to access the setup menu via a serial connection.*

Security Settings

Note: *We recommend that you set security over the dedicated network or over the serial setup. If you set parameters over the network (telnet 9999), someone else could capture these settings.*

Select **6** to choose security settings. A normal security setting consists of four characters. Extended security consists of 16 characters.

Disable Telnet Setup

This setting defaults to the **N** (No) option. The **Y** (Yes) option disables access to this Configuration Menu by Telnet (port 9999). It only allows access locally via the serial port of the Device Server.

Disable TFTP Firmware Upgrade

This setting defaults to the **N** (No) option. The **Y** (Yes) option disables the use of TFTP to perform network firmware upgrades. With this option, firmware upgrades can be performed only by using a *.hex file over the serial port of the Device Server.

Disable Port 77FE (Hex)

Port 77FE is a setting that allows OEMs and others to configure the Device Server remotely. You may wish to disable this capability for security purposes. (For more information about remote configuration, see the Lantronix Embedded Integration Kit user guide on the Lantronix web site (http://www.lantronix.com).

The default setting is the **N** (No) option, which enables remote configuration. You can configure the Device Server only by using Telnet or serial configuration. The **Y** (Yes) option disables remote configuration.

Note: *The Yes option disables many of the GUI tools for configuring the Device Server, including the embedded Web Page Configuration tool.*

Disable Web Setup

This setting defaults to the **N** (option). The **Y** (Yes) option disables the use of the Web Page Configuration tool that is built into the Device Server.

Enable Encryption

This setting defaults to the **N** (No) option, which completely disables the 128-bit Twofish encryption algorithm. The **Y** (Yes) option enables 128-bit Twofish encryption and a 32-hexadecimal (0-9, A-F) digit encryption string to be set. Only another Device Server configured with an identical encryption string will be able to correctly decode data sent from this Device Server. For more information on the Twofish encryption algorithm, see www.counterpane.com.

Note: *If you select **Y**, you must complete the settings. To escape, complete the settings and exit without saving.*

Enable Enhanced Password

This setting defaults to the **N** (option), which allows you to set a 4-character password that protects the Configuration Menu. The **Y** (Yes) option allows you to set a 16-character password.

Factory Default Settings

Select **7** to reset the Device Server's serial port to the factory default settings. The server configurations remain unchanged.

Exit Configuration Mode

Select **8** to exit the configuration mode without saving any changes, or select **9** to exit and save all changes. All values are stored in nonvolatile memory, and the Device Server resets.

8: Using the Device Server

Comm Port Redirector

The Lantronix Comm Port Redirector application allows PCs to share modems and other serial devices connected to a Device Server using Windows-based applications. The Comm Port Redirector intercepts communications to specified COM ports and sends them over an IP network connection to the Device Server's serial port. This enables the PC to use the Device Server's serial port as if it were one of the PC's COM ports. Using their existing communications software, users can dial out to a remote host through a modem connected to the Device Server.

Redirector Setup

To setup the Comm Port Redirector software:

1. Install the Redirector software. The software and installation instructions are included on the distribution CD.
2. In the Redirector's configuration screen, select **Port Setup** and add as many COM ports as you need (for example, one for each Device Server).
3. Under each port, select **Add IP** and enter the IP address (**Host**) of the Device Server that you want to assign to that port, and then enter the **TCPPort** number (3000 to 3009).

Note: *Remember the TCPPort number. You will need it to configure the Device Server.*

4. Save the configurations and reboot your PC.

Device Server Configuration

The following procedure should be repeated for each Device Server defined in the Redirector setup, above.

1. Enter the Device Server's Setup (configuration) Mode (see Chapter 7).
2. Set the **Port Number** to 11000 higher than the TCPPort number selected in the Redirector setup above (for example, if the TCPPort number was 3005, set the Device Server's Port Number to 14005).
3. Save the configurations and exit Setup Mode.

Note: *When using the Redirector, the Device Server does not change its serial port configuration to match the PC application's serial settings. Ensure that the Device Server serial port's configuration matches the configuration of your serial device.*

9: Serial Timing

Hardware

The signal condition and rise and fall times on the serial input and output lines for the CoBox-Micro, CoBox-Mini, and CoBox-Mini100 conform to CMOS input/output standards. For protection, a 220 Ohm resistor is used in series.

For the CoBox UDS-10B serial timing, refer to the RS232, RS422/485 standards that are applicable (see MAX 232 chip set requirements).

For further information, visit Advanced Micro Devices (AMD) on the web and refer to their documentation. Read the Am186ES/Am188ES data sheet and User’s Manual.

10: Integration Guidelines

The chapter describes the test setup and resultant emission profiles of the CoBox-Mini, CoBox-Mini100, and CoBox-Micro. General guidelines are also provided to help you reach the necessary standards for your applications.

This chapter assumes that the user is familiar with basic technical terms and abbreviations and understands Lantronix product features. For product information and additional documentation please visit the Lantronix home page www.Lantronix.com.

Test Environment

None of the units under test (UUT) have a driver IC on board to provide RS232 or RS485 levels at their pin headers. None of the UUTs have their own power supply circuitry. Therefore, all UUTs were hooked up to different evaluation boards, which provided a voltage regulator and the appropriate driver circuitry and connectors for a serial RS232 interface. For your own test requirements test boards can be obtained from Lantronix, please contact OEM sales.

First Setup

A preliminary scan was made in one of two types of sealed chambers. One type of sealed chamber was prepared to reduce emission echo. The other type was prepared to absorb reflection. Either type of chamber is acceptable.

The antenna used was a broadband type (EMCO biconilog 3142) connected to an Hewlett Packard spectrum analyzer HP8566B, in combination with an RF pre-selector HP 85685A and an Quasi peak adapter HP 85650A.

UUT cable connections and its antenna distance was determined according to the requirements of the applicable standards. The UUT was connected to an Ethernet hub using a common CAT5 twisted pair 10Base-T cable. Power was provided to the test board by a regular wall cube. Output for the CoBox-Mini and CoBox-Mini100 test beds were 15 VAC. Its output for the CoBox-Micro test bed was 9 VAC.

Second Setup

The result of the first scan was passed to a second outdoor test area in order to quantify the emission value. This second test area was setup similarly to the one described in the section above.

The distance of the antenna to the UUT was 10m according to the EN 55022 (CISPR22) requirement. The emission value for standards expecting different antenna distances (FCC) was automatically recalculated during the measurement. The antenna height, polarization and the orientation of the UUT to the antenna was varied for each pre-scanned frequency to find the maximum emission value (antenna height 1m-4m, polarization vertical and horizontal; UUT on a rotating desk 0° to 360°).

EMI test results

The radiated emission listing shows that the CoBox-Micro UUT passed FCC class-A limits as well as CISPR22 class-A. Highest emission outputs are at 80Mhz and 120Mhz, leading to a margin of 5dB at 120Mhz to the more stringent CISPR22 class-A limit.

Table 10-1: CoBox-Micro Results

Frequency MHz	Field Strength EN55022 Level at 10m dBµV/m	Limit Class A dBµV/m	DELTA dB	Remarks
40.015	21.53	40.00	-18.47	
60.003	27.75	40.00	-12.25	
80.008	30.26	40.00	-9.74	
100.015	22.97	40.00	-17.03	
120.008	34.95	40.00	-5.05	
140.015	21.16	40.00	-18.84	
160.008	25.01	40.00	-14.99	
180.014	22.54	40.00	-17.46	
200.01	24.95	40.00	-15.05	
208.539	20.85	40.00	-19.15	
220.015	26.96	40.00	-13.04	
228.541	27.20	40.00	-12.80	
240.01	35.46	47.00	-11.54	
260.015	31.85	47.00	-15.15	
265.71	30.59	47.00	-16.41	
268.52	30.65	47.00	-16.35	
280.015	27.18	47.00	-19.82	

Table 10-1: CoBox-Micro Results, cont.

Frequency MHz	Field Strength EN55022 Level at 10m dBμV/m	Limit Class A dBμV/m	DELTA dB	Remarks
285.79	27.37	47.00	-19.63	
297.107	32.75	47.00	-14.25	
320.02	25.45	47.00	-21.55	
340.026	27.49	47.00	-19.51	
360.025	31.54	47.00	-15.46	
380.025	34.01	47.00	-12.99	
400.024	30.62	47.00	-16.38	
420.03	30.64	47.00	-16.36	
440.02	27.20	47.00	-19.80	
480.029	27.28	47.00	-19.72	
520.035	27.57	47.00	-19.43	

The radiated emission listing shows that the CoBox-Mini UUT passed FCC class-B limits as well as CISPR22 class-B. Highest emission outputs were at 60 Mhz and 80 Mhz, leading to a margin of 3dB at 60 Mhz to the class B limit.

Table 10-2: CoBox-Mini Results

Frequency MHz	Field Strength Level at 3 m dBμV/m	Limit FCC Class B dBμV/m	DELTA dB	Remarks
35.30	29.96	40.00	-10.04	
37.66	29.73	40.00	-10.27	
40.01	34.24	40.00	-5.76	
60.01	36.95	40.00	-3.05	
80.02	35.31	40.00	-4.69	
100.02	34.82	43.50	-8.68	
120.01	30.40	43.50	-13.10	

Table 10-2: CoBox-Mini Results, cont.

Frequency MHz	Field Strength Level at 3 m dBµV/m	Limit FCC Class B dBµV/m	DELTA dB	Remarks
140.02	35.11	43.50	-8.39	
220.03	33.01	46.00	-12.99	
260.03	35.55	46.00	-10.45	
261.16	32.28	46.00	-13.72	
271.72	34.03	46.00	-11.97	

The radiated emission listing shows, that the unit passed FCC class-B limits as well as CISPR22 class-B. Barracuda is used as alias for the mini 10/100 during the development phase. Highest emission outputs are at 40Mhz and 50Mhz leading to a margin of 5dB at 40Mhz to the CISPR22 class B limit.

Table 10-3: CoBox-Mini100

Frequency MHz	Field Strength Level at 3 m dBµV/m	Limit FCC Class B dBµV/m	DELTA dB	Remarks
40.00	34.99	40.00	-5.01	
50.00	34.25	40.00	-5.75	
100.01	35.72	43.50	-7.78	
110.83	30.07	43.50	-13.43	
125.01	30.99	43.50	-12.51	
150.00	37.34	43.50	-6.16	
175.00	29.55	43.50	-13.95	
200.01	36.10	43.50	-7.40	
250.01	33.30	46.00	-12.70	
325.01	33.32	46.00	-12.68	
375.02	35.68	46.00	-10.32	

Table 10-3: CoBox-Mini100, cont.

Frequency MHz	Field Strength Level at 3 m dBµV/m	Limit FCC Class B dBµV/m	DELTA dB	Remarks
450.01	37.69	46.00	-8.31	

General Guidelines

This section covers the following topics:

- power supply
- network connector
- virtual ground
- serial signals
- emission improvements
- CoBox-Mini
- CoBox-Mini100
- CoBox-Micro

Power Supply

The CoBox-Mini, CoBox-Mini100, and CoBox-Micro run at 5 VDC nominal, ±5%. The current consumption varies for the different products and depends upon their operating conditions. Refer to the current requirements listed in the product specification in order to design an appropriate power supply.

To maintain the necessary voltage, provide ground to the appropriate connector header with a low inductance and low DC resistance path. The best solution is a solid ground plane.

Place a de-coupling capacitor pair as close as possible to the connector headers of the board’s power supply. We recommend a ceramic (X7R material or equivalent, value 0,022uF to 0,1uF) and a low DC resistance (electrolytic or tantalum value 10uF to 100uF) capacitor.

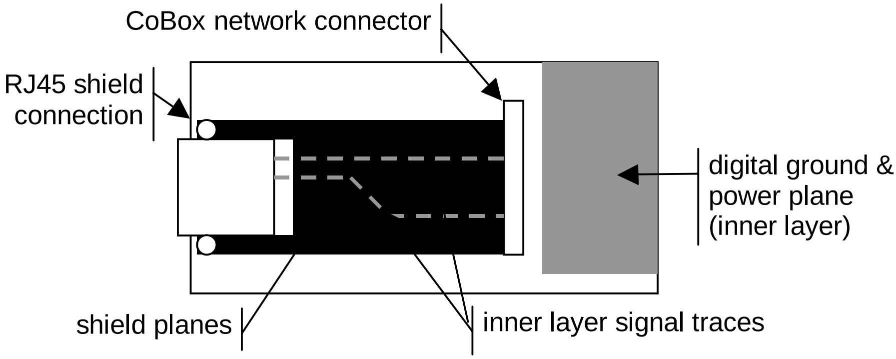
Network Connector

If you want to add an RJ45 connector, we recommend that it be at least partially shielded in case it will be used in a noisy environment. (Please refer to the product specific section.)

Take care regarding the trace length and routing for the two differential pairs, TX and RX. Neither of them may cross or run in parallel with any digital signal. Neither of them may cross or run in parallel with any digital signal nor run through a digital ground or power plane. The trace length inside of the unit running from the CoBox to the RJ45 should be as short as possible. The trace length may have an impact on signal quality (i.e. link length), especially if internal ambient noise is a factor.

If trace length cannot be shortened or the internal noise frequencies are hitting the carrier frequency, or the multiples of these (depending on the product and operating mode either 10Mhz or 100Mhz and up to the 11^{th} overtone), we suggest a different strategy. Use a multi-layer board and a separated shield layer on the solder and assembly sides of the board which are routed in the inner layers. (Refer to the following figure.) These shield layers can either be connected to the RJ45s shield or to a virtual ground signal provided by the CoBox. (See also next section and product specific section.)

Figure 10-1: Multi-layer board strategy



Virtual Ground

The CoBox units provide a virtual ground at a (plated) mounting hole. It is a ground imitation. It uses the tap of two capacitors (ceramic 0.1uF) conducted symmetrically to ground and VCC. In absence of a solid ground (earth) this can be used for shielding or balancing metal parts of the case.

Serial Signals

CoBox TTL level serial input and output signals are protected by 220 Ohm resistors. These resistors provide a simple output shortage protection for infinitive duration (by limiting the current). They also reduce conducted interferences at higher frequencies to the base board

Additional Emission Improvements

Depending on the used voltage regulator and base board design, conducted interferences can sometimes be emitted by the power supply cord. If the voltage level there is low, common mode chokes are the appropriate barrier to avoid these frequencies being emitted via the power cord as an antenna.

Common mode chokes help pass the conducted emission requirements of the EN55022 for frequencies below 30 Mhz. Metal cases or partial metal shielding inside the unit can also help to reduce emission levels so that even more stringent standards can be passed.

CoBox-Mini Ethernet Integration

The AUI port connector doesn’t provide any DC decoupling transformer circuitry. The customer must provide the necessary circuitry on the base board. Trace length from the header to the transformer should be kept as short as possible. For additional AUI guidelines please see also the transformer manufactures application notes.

CoBox-Mini100

Take care when laying out the trace for the RJ45. We recommend the use of shields described in the "Network Connector" section above. On the optional 11 pin header for the network connector, additional termination for the unused wires of the cable is provided to improve the signal quality on longer lines.

CoBox-Micro

If the CoBox Micro is used with the on-board RJ45 connector, we suggest that you provide ground level to the plated mounting hole near the RJ45. That shielded cable will be tight to the appropriate level, however the virtual ground is also provided there.

A: Contact Information

If you are experiencing an error that is not listed in Appendix B, or if you are unable to fix the error, there are a number of other troubleshooting options:

- Look on the Lantronix Web site for technical FAQs and documentation updates.
- For information pertaining to your system's configuration, refer to your system's documentation or technical support. For example, for specific questions about the Microsoft Windows Operating System, refer to the Microsoft Knowledge Base Web site at www.support.microsoft.com/directory.
- Contact your dealer or Lantronix Technical Support at 800-422-7044 (US) or 949-453-3990. Technical Support is also available via Internet E-mail at **support@lantronix.com**.

Problem Report Procedure

When you report a problem, please provide the following information:

- Your name
- Your company name, address, and phone number
- Product model (CoBox-Mini, CoBox-Mini100, CoBox-Micro, UDS-10B)
- Serial number (7-digit number under the bar code label)
- Software version
- Network configuration
- Description of the problem
- Status of the unit when the problem occurred (please try to include information on user and network activity at the time of the problem)

Full Contact Information

Corporate Offices

15353 Barranca Parkway
Irvine, CA 92618, USA
Phone: (949) 453-3990
Fax: (949) 453-3995
World Wide Web: www.lantronix.com

Sales Offices

The Americas
15353 Barranca Parkway
Irvine, CA 92618, USA
Phone: (949) 450-7227
Fax: (949) 450-7231
E-mail: sales@lantronix.com

Europe, Middle East, and Africa
Minervum 1707
4817 ZK Breda
The Netherlands
Phone: +31 (0) 76 565 8176
Fax: +31 (0) 76 565 8179
E-mail: eu_sales@lantronix.com

France
Phone: +33 (0) 139 30 41 74
Fax: +33 (0) 139 30 41 73
E-mail: eu_sales@lantronix.com

Germany
Phone: +49 (0) 7720 3016 20
Fax: +49 (0) 7720 3016 88
E-mail: eu_sales@lantronix.com

Pacific Rim
46 East Coast Road
East Gate #10-01
Singapore 428766
Phone: +65 447 4222
Fax: +65 344 0614
E-mail: asiapacsales@lantronix.com

Spain and Portugal
Phone: +34 (0) 91760 0846
Fax: +34 (0) 91760 0846
E-mail: eu_sales@lantronix.com

United Kingdom and Ireland
Phone: +44 (0) 118 945 1555
Fax: +44 (0) 118 945 1663
E-mail: eu_sales@lantronix.com

International Sales
Phone: (949) 450-7227
Fax: (949) 450-7231
E-mail: intsales@lantronix.com

Technical Support

Technical Support
Phone: (800) 422-7044 or (949) 453-7198
Fax: (949) 450-7226
FAQs, Documentation, and Firmware Downloads: www.lantronix.com/support
E-mail: support@lantronix.com

Europe, Middle East, and Africa Technical Support
Phone: +49 (0) 7720 3016 57
Fax: +49(0) 7720 3016 88
E-mail: eu_techsupp@lantronix.com

B: Troubleshooting

Monitor Mode

Monitor Mode is a command-line interface used for diagnostic purposes (see *Table B-1: Monitor Mode Commands* on page B-2). There are two ways to enter Monitor Mode: locally via the serial port or remotely via the network.

Entering Monitor Mode Via the Serial Port

To enter Monitor Mode locally:

1 Follow the same principles used in setting the serial configuration parameters (see *Serial Configuration* on page 7-2).

Instead of typing three “x” keys, however, type **zzz** (or **xxl**) to enter Monitor Mode with network connections.

Type **yyy** (or **yyy**) to enter Monitor Mode **without** network connections.

2 A **0**> prompt indicates that you have successfully entered Monitor Mode.

Entering Monitor Mode Via the Network Port

To enter Monitor Mode using a Telnet connection:

1 First establish a Telnet session to the configuration port (9999). The following message appears:

Figure B-1: Entering Monitor Mode Via the Network

```
Serial Number 1400280  MAC address 00:20:4A:14:01:18
Software Version 4.3 (xxxxxx)
Press Enter to go into Setup Mode
```

2 Type **M** (upper case).

3 A **0**> prompt indicates that you have successfully entered Monitor Mode.

Monitor Mode Commands

The following commands are available in Monitor Mode. Many commands have an IP address as an optional parameter (xxx.xxx.xxx.xxx). If the IP address is given, the command is applied to another Device Server with that IP address. If no IP address is given, the command is executed locally.

Note: *All commands must be given in capital letters, with blank spaces between the parameters.*

Table B-1: Monitor Mode Commands

Command	Command Name	Function
DL	Download	Download firmware to the Device Server via the serial port in hex format
SF x.x.x.x	Send Firmware	Send firmware to Device Server with IP address x.x.x.x
VS x.x.x.x	Version	Query software header record (16 bytes) of Device Server with IP address x.x.x.x
GC x.x.x.x	Get Configuration	Get configuration of Device Server with IP address x.x.x.x as hex records (120 bytes)
SC x.x.x.x	Send Configuration	Set configuration of Device Server with IP address x.x.x.x from hex records
PI x.x.x.x	Ping	Ping Device Server with IP address x.x.x.x to check device status
AT	ARP Table	Show the Device Server's ARP table entries
TT	TCP Connection Table	Shows all incoming and outgoing TCP connections
NC	Network Connection	Shows the Device Server's IP configuration
RS	Reset	Resets the Device Server's power
SI xxx.xxx.xxx.xxx:yyy.yyy.yyy.yyy	Send/Set IP Address	Remotely assign an IP address to a Device Server, where xxx.xxx.xxx.xxx is the IP address, and yyyy.yyy.yyyy.yyy is the two-part identification number at the bottom of the label, converted to decimal, and written twice.
QU	Quit	Exit diagnostics mode
G0, G1, ...,GE, GF	Get configuration from memory page	Gets a memory page of configuration information from the device.
S0, S1,...,SE, SF	Set configuration to memory page	Sets a memory page of configuration information on the device.

Note: *Entering any of the commands listed above will generate one of the following command response codes:*

Table B-2: Command Response Codes

Response	**Meaning**
0>	OK; no error
1>	No answer from remote device
2>	Cannot reach remote device or no answer
8>	Wrong parameter(s)
9>	Invalid command

C: Updating Firmware

Obtaining Firmware

Current firmware files are available on the distribution CD. Firmware updates and release notes for Device Servers can be downloaded directly from Lantronix in one of the following ways: via the Lantronix World Wide Web site (www.lantronix.com), or using anonymous FTP through the Internet (ftp://ftp.lantronix.com/pub).

Reloading Firmware

There are three ways to update the Device Server's internal operational code (CBX*.ROM or CBX*.HEX): via TFTP, via another device server, or via the serial port.

You can also update the Device Server's internal Web interface (CBXW*.COB) via TFTP.

Reloading Firmware Via TFTP

To download new firmware from a computer:

1. Use a TFTP client to send a binary file to the Device Server (CBX*.ROM to upgrade the Device Server's internal operational code and CBXW*.COB to upgrade its internal Web interface).

   Note: *TFTP requires the **.ROM** (binary) version of the Device Server's internal operational code.*

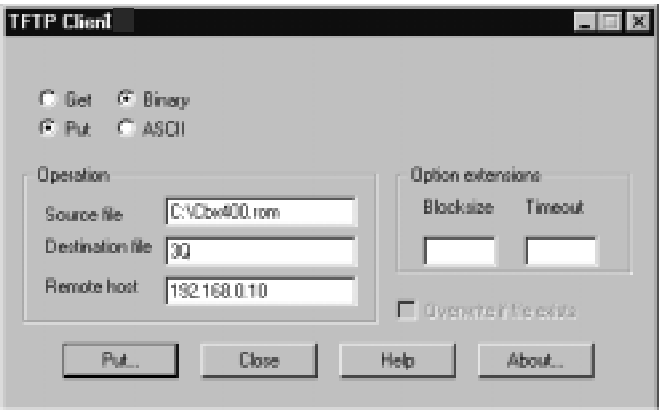
   A Include the full path in the **Source File**.

   B The **Destination File** is **3Q** (upper case) for the internal operational code and **WEB6** for the internal Web interface.

   C **Remote Host** is the IP address of the unit being upgraded.

   D Select **Put** to transfer the file to the Device Server.

Figure C-1: TFTP Dialog Box



2 Wait for the Device Server to perform a power reset after the firmware has been loaded.

Reloading Firmware Via Another Device Server

To upgrade firmware on another Device Server over the network:

1 Enter the host Device Server's Monitor Mode (see *Monitor Mode* on page B-1).

2 Send the firmware to the receiving Device Server using the **SF** command, where x.x.x.x is the receiving Device Server's IP address.

Figure C-2: Sending Firmware to Another Device Server

```
SF x.x.x.x
```

3 The Device Server performs a power reset after the firmware has been loaded.

Note: *You can only update the Device Server's internal Web interface using TFTP.*

Reloading Firmware Via the Serial Port

To download firmware from a computer via the Device Server's serial port:

1 Enter Monitor Mode (see *Monitor Mode* on page B-1).

2 To download the firmware to the Device Server using the **DL** command.

A Select **Send Text File** and select the CBX*.HEX file to be downloaded.

Note: *The downloaded file must be the **.HEX** (ASCII) version.*

3 After the final record is received, the Device Server checks the integrity of the firmware image before programming the new firmware into the flash ROM.

Note: ***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 Device Server. This process may take up to 10 minutes when using the serial method.*

4 The following message displays when the firmware upgrade is complete.

Figure C-3: Firmware Upgrade Screen Display

```
*** NodeSet 2.0 ***
0>DL
02049 lines loaded.
Max Address FFE0
Loading EEPROM ...
```

5 The Device Server performs a power reset after the firmware has been loaded.

Note: *You can only update the Device Server's internal Web interface using TFTP.*

D: Data

Measurements

The CoBox-Mini and Mini100 are dimensionally identical. The CoBox-Micro and UDS-10B are of different sizes.

Table D-1: Dimensions

Micro	H x W x D: 0.682 x 1.574 x 1.929 in (17.27 x 40.0 x 49.0 mm) Weight: 0.8 lbs (0.35 Kg)
Mini and Mini100	H x W x D: 0.282 x 3.150 x 2.756 in (7.27 x 8.00 x 70.0 mm) Weight: 0.09 lbs (0.04 Kg)
UDS-10B	H x W x D: 3.385 x 2.280 x 0.55 in (86.0 x 57.9 x 14 mm) Weight: 0.8 lbs (0.35 Kg)

Specifications

The following table list technical information common to all four of these Device Servers.

Table D-2: CoBox: Micro, Mini, Mini100 and UDS-10B Specifications

Protocols Supported	ARP, UDP, TCP, Telnet, ICMP, SNMP, DHCP, BOOTP, TFTP, and HTTP
Network Interface	10Base-T or AUI (only with CoBox-Mini, transformer and resistors/ capacitors required for the AUI connection)
Serial Interface	2 TTL serial interfaces (Async) for CoBox -Micro, Mini, Mini100 1 RS232/485 DB-25 port
Data Rates	300 bps to 115.2 Kbps
Serial Line Formats	Characters: 7 or 8 data bits Stop bits: 1 or 2 Parity: odd, even, none
Modem Control	DTR, DCD, CTS, RTS
Flow Control	XON/XOFF (software) CTS/RTS (hardware) None
Management	Internal web server SNMP (read only) Serial login Telnet login
System Software	downloadable from a TCP/IP host (TFTP) or over serial port
LEDs	good link (green) network Transmit/Receive Data (yellow) collision (red) diagnostics (red) status (green, channel 1)
Compatibility	Ethernet: Version 2.0/IEEE 802.3
Power Requirements	5V (±5%) regulated @ 200 - 250mA
Buffer Size	Serial: 2K Network: 2K
Memory	Flash: 512K RAM: 128K NVRAM: 2K
Environmental	Operating Temperature: 5to 50C (41 to 122F) Storage Temperature: -40 to 66C (-40 to 151F)

E: Supplemental Information

Network Configuration Using UDP

The Device Server can also be configured or queried over the network using UDP datagrams. The Device Server has a UDP listener set for port 30718 (77FE Hex). Responses from the Device Server are returned to the source port of the UDP packet.

The first three bytes of the UDP data block should be set to zero. The fourth byte selects the function as described in the following table:

Table E-1: UDP Configuration

Byte	Command	Parameters	Notes
03	Node Reset	2 bytes, software type	These 2 bytes are used to prevent accidental reset of the Device Server. (Value for standard CoBox firmware: 33 51 [Hex], **3Q**)
F6	Query for Firmware Version	None	The Device Server responds with the **F7** block below.
F7	Firmware Information	First 16 bytes of the firmware image, and 4 bytes device information and serial number.	The first 16 bytes of the firmware image contain the software type (offset 4,5) and checksum (offset 14,15). The last two bytes of the device information contain the serial number.
F8	Query for Setup Record	None	The Device Server responds with the **F9** block below.
F9	Configuration Readback	120 byte setup record (see *Setup Records* on page E-7)	n/a
FA	Set Configuration	120 byte setup record (see *Setup Records* on page E-7)	The IP address (byte 0-3) will not be overridden using **FA**. See **FD** for this functionality.
FB	Configuration Change Acknowledge	None	This block is sent back to the host requesting a configuration change (**FB**). After sending out this block, the Device Server resets and uses the new configuration sent with the **FA** command.
FC	Set IP Address	First 8 bytes must be set to the string **IP-SETUP** (Hex 49 50 2D 53 45 54 55 50). Next 2 bytes have to be set to **00**. Next 2 bytes must contain the serial number. Next 4 bytes have to be the new IP address.	This block can be sent as a broadcast, because the serial number is unique. It provides one method to set the IP address of the Device Server if is on the local network and the serial number is known. Remember, broadcasts are only 'heard' on the subnet on which they are generated. No reply is sent by the Device Server, which restarts using the new IP address after the block is received. Example (all in Hex): 49 50 2D 63 45 54 55 50 00 00 2A 12 81 00 01 02 IP address of the node with serial number 42-18 set to 129.0.1.2
FD	Set Configuration and IP Address	Same as **FA**, but changes IP address as well (bytes 0-3).	n/a

Network Configuration Using a Hex File

When configuring a number of Device Servers identically, it is useful to create a template setup record. The setup record can then be sent to the “target” Device Servers from a “master” Device Server via “cut and paste” or UDP (see *Network Configuration Using UDP* on page E-1).

Device Servers use a 120-byte setup record in Intel Hex format. This format facilitates the transfer of binary data using ASCII characters. See *Setup Records* on page E-7and *The Intel Hex Format* on page E-5 for information about setup records and converting them to Intel Hex format.

Figure E-1: Sample Setup Record in Intel Hex Format

```
:20000010AC10C81D0000100000000000AC10010B4C0200001127000000000000C000000011
:200020100000000000000000000000000000000000000000000000000000000000000000B0
:200040104C0200001227000000000000C00000000000000000000000000000000000000049
:180060100000000000000000000000000000000000000000000000000078
:00000001FF
```

Acquiring a Valid Setup Record

There are a number of ways to acquire a valid setup record:

- Copy the setup record of a properly configured Device Server via Monitor Mode (easiest method).
- Request the setup record of a properly configured Device Server via another Device Server on the network.
- Build the setup record in software.
- From a host PC, request the setup record of a properly configured Device Server via UDP.

To copy the setup record of a properly configured Device Server:

1. Configure a “master” Device Server with the desired parameters.
2. Enter Monitor Mode on the master Device Server (see *Monitor Mode* on page B-1).
3. At the prompt, enter **GC** followed by a carriage return.

   The Device Server will respond with its setup record in Intel Hex format.
4. Copy the setup record into a text file and save it for future use.

To request the setup record of a properly configured Device Server via another Device Server on the network:

1. Configure a “master” Device Server with the desired parameters and place it on the network.

2. Place another Device Server (the “target”) on the network. (Use the master and SC “Send Configure” command to set the target unit.

3. Enter Monitor Mode (with network support enabled) on the target Device Server (see *Monitor Mode* on page B-1).

4. At the prompt, enter **GC x.x.x.x** followed by a carriage return, where x.x.x.x is the IP address of the master Device Server.

   The master Device Server responds by sending its setup record to the target Device Server, and the target Device Server reboots with the new configuration.

To build the setup record in software:

1. Create a 120-byte setup record.

2. Convert it to an Intel Hex record (see *The Intel Hex Format* on page E-5).

3. Copy the setup record into a text file and save it for future use.

To request the setup record of a properly configured Device Server via UDP:

1. Configure a Device Server with the desired parameters and place it on the network.

2. From a host PC, send the **F8** datagram to the Device Server (see *Network Configuration Using UDP* on page E-1). The Device Server responds with the **F9** datagram, which includes its setup record.

3. Send a previously saved setup record from a host PC via UDP.

Sending a Setup Record

There are also a number of ways to *send* a setup record to a Device Server:

- Send a previously saved setup record via Monitor Mode (easiest method).

- Send the setup record of a properly configured Device Server to another Device Server on the network.

- Send a previously saved setup record from a host PC via UDP.

To send a setup record via Monitor Mode:

1. Configure a “master” Device Server with the desired parameters and place it on the network.
2. Place another Device Server (the “target”) on the network.
3. Enter Monitor Mode (with network support enabled) on the master Device Server (see *Monitor Mode* on page B-1)
4. At the prompt, enter **SC**, the IP address of the target, and a carriage return.
5. Send the setup record to the target Device Server.

Note: *For example, using Hyperterminal, copy the setup record and select “Paste to Host” to send it to the Device Server. The Device Server reboots with the new configuration.*

To send a previously saved setup record to a Device Server via UDP, from a host PC, send the **FA** (or **FD**) datagram to the “target” Device Server (see *Network Configuration Using UDP* on page E-1).

Note: *The Device Server responds with the **FB** datagram. Refer to the table.*

The Intel Hex Format

With this format, 8-bit binary data can be sent and received as ASCII text. The transmission is blocked in records, and every record has its own checksum.

The record begins with a colon (**:**) and consists of a block length (2-character Hex), a 16-bit address (4-character Hex), and a block type (2-character Hex). It is built by adding all binary 8-bit values and taking the complement, so adding all byte values (including length, address, and type) should yield zero.

Example:

00000001FF
End record, type 01, length 00, address 00 00, checksum FF.

01002000805F
Data record consisting of one byte (value 80 Hex) for address 0020 (32 decimal).

For communication with the node, the following block types are defined:

Table E-2: Block Types

Option	Hex
Data block program memory (firmware)	00
End record	01
Data block configuration memory	10

To get and set the node configuration, 120 bytes should be exchanged at once in 32-Byte records. The IP address in the record (bytes 0 to 3) will be ignored (unless the UDP **FD** command is being used).

Calculating the Checksum

As mentioned in Table E-2: Block Types above, the last two characters of an Intel Hex setup record represent a checksum of the data in the line. Since the checksum is a two-digit hexadecimal value, it can represent a value from 0 to 255.

The checksum is calculated by summing the value of the data on the line and taking the two's complement of the sum.

> **Note:** *Do not include the leading colon or the checksum byte in the sum.*

Example:

0300300002337A1E

Record length: 03 (3 bytes of data)
Address: 0030 (the 3 bytes will be stored at 0030, 0031, and 0032)
Record Type: 00 (normal data)
Data: 02, 33, 7A
Checksum: 1E

03 + 00 + 30 + 00+ 02 + 33 + 7A = E2

The two's complement of E2 is 1E. See *Calculating the Two's Complement* below.

Calculating the Two's Complement

The two's complement of a number is the value that must be added to the number to reach a Hexadecimal value of 100 (256 in decimal). In the example above, E2 + 1E = 100.

You can also calculate the two's complement by subtracting the sum from 100. Using the example above again, 100 - E2 = 1E. It may help to use a scientific calculator.

Setup Records

A setup record consists of 120 bytes. They are transmitted at once from and to the node. Unused bytes should be initialized as **00**. Table E-3: Setup Record Construction defines the structure of a setup record:

Table E-3: Setup Record Construction

Byte(s)	**Function**
00-03	IP address of the unit (x.x.x.x)
04	Reserved (**0**)
05	Flag BYTE Bit 7: Reserved (**0**) Bit 6: Set **1** for AUI, **0** for 10BASE-T (CoBox-Micro only) Bits 5-0: Reserved (**0**)
06	Number of host bits for subnetting; if **0**, matching standard netmask for Class A, B, C is used.
07	Reserved (**0**)
08-11	Telnet configuration password (**0** if not used)
12-15	Gateway IP address (**0,0,0,0** if not used)
16-63	48-byte Channel 1 parameters; parameter setup Channel 1 (see Table E-4: Channel Parameters)
64-111	48-byte Channel 2 parameters; parameter setup Channel 2 (see Table E-4: Channel Parameters))
112-119	Reserved (**0**)

Channel Parameters

Use the following table to select setup record parameters for Channels 1 and 2:

Table E-4: Channel Parameters

Byte(s) (Channel 1)	Byte(s) (Channel 2)	Function
16	64	Interface Mode (see Table E-5: Interface Mode Options)
17	65	Line Speed Bits 7-5: Reserved Bits 4-0: Baud Rate (see Table E-7: Baud Rate Settings)
18	66	Flow Control (see Table E-8: Flow Control Options)
19	67	Reserved
20-21	68-69	Own TCP port low-byte, high-byte (Intel)
22-23	70-71	Remote TCP port low byte, high-byte (Intel)
24-27	72-75	Remote IP address (low/high low/high)
28	76	Connect Mode (see Table E-9: Connect Mode Options)
29	77	Disconnect Mode (see Table E-10: Disconnect Mode Options)
30	78	Disconnect w/ inactivity time-out, minutes (**00** if unused)
31	79	Disconnect w/ inactivity time-out, seconds (**00** if unused)
32-33	80-81	Characters to trigger send immediately (sendchar)
34	82	Flush mode (see Table E-11: Flush Mode Options)
35	83	Pack Control (see Table E-12: Pack Control Options)
36-47	84-95	Reserved (**0**)
48-63	96-111	a) Terminal name for Telnet terminal type option (15 characters max), **0**-terminated. If set and Bit 6 in Disconnect Mode is set, Telnet connection will be assumed. b) Password for Passworded Socket Connection (Bit 4 in Disconnect Mode Set).

Interface Mode

The Interface (I/F) Mode is a bit-coded byte entered in hexadecimal notation. Use the following table to select Interface Mode settings:

Table E-5: Interface Mode Options

Option	Bit 7	6	5	4	3	2	1	0
RS-232C							0	0
RS-422/485							0	1
RS-485 2-wire							1	1
7 Bit					1	0		
8 Bit					1	1		
No Parity			0	0				
Even Parity			1	1				
Odd Parity			0	1				
1 Stop bit	0	1						
2 Stop bits	1	1						

Note: *Embedded units require external serial drivers to support RS-232 or RS-485.*

The following table demonstrates how to build some common Interface Mode settings:

Table E-6: Common Interface Mode Settings

Option	Binary	Hex
RS-232C, 8-bit, No Parity, 1 stop bit	0100 1100	4C
RS-232C, 7-bit, Even Parity, 1 stop bit	0111 1000	78
RS-485 2-Wire, 8-bit, No Parity, 1 stop bit	0100 1111	4F
RS-422, 8-bit, Odd Parity, 2 stop bits	1101 1101	DD

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Baud Rate

The Device Server and attached serial device must agree on a speed or baud rate to use for the serial connection. Use the following table to select Baud Rate settings:

Table E-7: Baud Rate Settings

Speed (bps)	Hex
38400	00
19200	01
9600	02
4800	03
2400	04
1200	05
600	06
300	07
115200	08
57600	09

Flow Control

Flow control sets the local handshaking method for stopping serial input/output. Generally, flow control is not required if the connection is used to pass a blocked protocol with block sizes less than 1k (ACK/NAK) and/or speeds of 19200 or less. Use the following table to select Flow Control options:

Table E-8: Flow Control Options

Option	Hex
No flow control	00
XON/XOFF flow control	01
Hardware handshake with RTS/CTS lines	02
XON/XOFF pass characters to host	05

Connect Mode

Connect Mode defines how the Device Server makes a connection, and how it reacts to incoming connections over the network. Use the following table to select Connect Mode options:

Table E-9: Connect Mode Options

Option Bit	7	6	5	4	3	2	1	0
Incoming (network) Connection								
Never accept incoming	0	0	0					
Accept incoming with DTR*	0	1	0					
Accept unconditional	1	1	0					
Response								
Nothing (quiet)				0				
Character response (C=conn, D=disconn, N=unreachable)				1				
Startup (on serial connection)								
No active startup					0	0	0	0
With any character					0	0	0	1
With active DTR*					0	0	1	0
With CR (0x0D) only					0	0	1	1
Manual connection					0	1	0	0
Autostart					0	1	0	1
Datagram Type								
Directed UDP					1	1	0	0
Modem Mode								
With Echo				1	0	1	1	0
Without Echo				0	0	1	1	0

***DTR is an output signal on a DTE serial port. The DTE devices are the CoBox Micro, CoBox Mini, and CoBox Mini100. On DTE devices, use DCD (input) instead of DTR.**

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Disconnect Mode

In Disconnect Mode, DTR drop either drops the connection or is ignored. Use the following table to select Disconnect Mode Options:

Table E-10: Disconnect Mode Options

Option Bit	7	6	5	4	3	2	1	0
Disconnect with DTR drop	1							
Ignore DTR	0							
Telnet mode and terminal type setup		1						
Channel (port) password				1				

***DTR is an output signal on a DTE serial port. The DTE devices are the CoBox Micro, CoBox Mini, and CoBox Mini100. On DTE devices, use DCD (input) instead of DTR.**

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Flush Mode (Buffer Flushing)

Using this parameter, you can control line handling and network buffers with connection startup and disconnect. You can also select between two different packing algorithms. Use the following table to select Flush Mode options:

Table E-11: Flush Mode Options

Function Bit	7	6	5	4	3	2	1	0
Input Buffer (Serial to Network)								
Clear active connection (from serial)				1				
Clear passive connection (from network)			1					
Clear at disconnect		1						
Output Buffer (Network to Serial)								
Clear active connection (from network)								1
Clear passive connection (from serial)							1	
Clear at disconnect						1		
Alternate Packing Algorithm								
Enable	1							

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

Pack Control

Alternate packing algorithm settings are enabled in Flush Mode. Use the following table to select Pack Control options:

Table E-12: Pack Control Options

Option **Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Idle Time								
Force transmit: 12ms							0	0
Force transmit: 52ms							0	1
Force transmit: 250ms							1	0
Force transmit: 5sec							1	1
Trailing Characters								
None					0	0		
One					0	1		
Two					1	0		
Send Characters								
Sendchars Define 2-Byte Sequence				1				
Send Immediately After Sendchars			1					

Note: *See Binary to Hexadecimal Conversion on page E-18 for more information on converting binary values to hexadecimal format.*

IP Addresses

Each TCP/IP node on a network host has a unique IP address. This address provides the information needed to forward packets on the local network and across multiple networks if necessary.

IP addresses are specified as **x.x.x.x**, where each **x** is a number from 1 to 254; for example, 192.0.1.99. The Device Server must be assigned a unique IP address to use TCP/IP network functionality.

IP addresses contain three pieces of information: the **network**, the **subnet**, and the **host**.

Network Portion

The network portion of the IP address is determined by the network type: Class A, B, or C.

Table E-13: Network Portion of IP Address

Network Class	Network Portion of Address
Class A	First byte (2nd, 3rd, and 4th bytes are the host)
Class B	First 2 bytes (3rd and 4th bytes are the host)
Class C	First 3 bytes (4th byte is the host)

In most network examples, the host portion of the address is set to zero.

Table E-14: Available IP Addresses

Class	Reserved	Available
A	0.0.0.0 127.0.0.0	1.0.0.0 to 126.0.0.0
B	128.0.0.0 191.255.0.0	128.1.0.0 to 191.254.0.0
C	192.0.0.0 223.255.255.0	192.0.1.0 to 223.255.254.0
D, E	224.0.0.0 to 255.255.255.254 255.255.255.255	None (Check this)

Consider the IP address **36.1.3.4**. This address is a Class A address; therefore, the network portion of the address is 36.0.0.0 and the host portion is 1.3.4.

Subnet Portion

The subnet portion of the IP address represents which **sub-network** the address is from. Sub-networks are formed when an IP network is broken down into smaller networks using a **subnet mask**.

A router is required between all networks and all sub-networks. Generally, hosts can send packets directly only to hosts on their own sub-network. All packets destined for other subnets are sent to a router on the local network.

Host Portion

The host portion of the IP address is a unique number assigned to identify the host.

Network Address

A host address with all host bits set to **0** addresses the network as a whole (for example, in routing entries).

Figure E-2: Sample Network Address

```
192.168.0.0
```

Broadcast Address

A host address with all host bits set to **1** is the broadcast address, meaning for “for every station.”

Figure E-3: Sample Broadcast Address

```
192.168.0.255
```

Network and broadcast addresses must not be used as a host address; for example, 192.168.0.0 identifies the entire network, and 192.168.0.255 identifies the broadcast address.

IP Subnet Mask

An IP subnet mask divides IP address differently than the standards defined by the classes A, B, and C. An IP subnet mask defines the number of bits to be taken from the IP address as the network or host sections. The Device Server prompts for the number of host bits to be entered and then calculates the netmask, which is displayed in standard decimal-dot notation (for example, 255.255.255.0) when saved parameters are displayed.

Table E-15: Standard IP Network Netmasks

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

Table E-16: Netmask Examples

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

Private IP Networks and the Internet

If your network is not and will not be connected to the Internet, you may use any IP address. If your network is connected or will be connected to the Internet, or if you intend to operate the Device Server on an intranet, you should use one of the reserved sub-networks. Consult your network administrator with questions about IP address assignment.

Network RFCs

For more information about IP addresses, refer to the following documents, which can be located on the World Wide Web using one of the following directories or indices:

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

Binary to Hexadecimal Conversion

Many of the Device Server's configuration procedures require you to assemble a series of options (represented as bits) into a complete command (represented as a byte). The resulting binary value must be converted to a hexadecimal representation.

Hexadecimal digits have values ranging from 0 to F, which are represented as 0-9, A (for 10), B (for 11), etc. To convert a binary value (for example, 0010 0011) to a hexadecimal representation, the upper and lower four bits are treated separately, resulting in a two-digit hexadecimal number (in this case, 4C).

Use the following table to convert values from binary to hexadecimal.

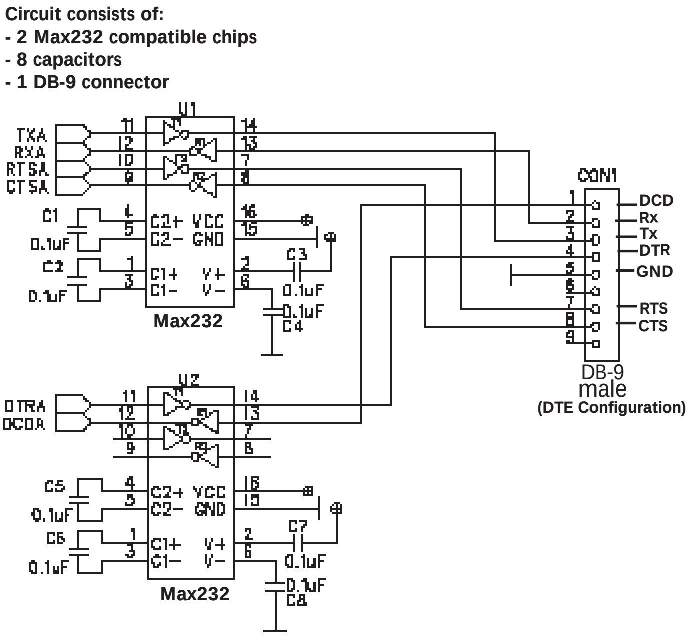
Table E-17: Binary to Hexadecimal Conversion Table

Decimal	Binary	Hex
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

F: Typical TTL-RS232 Circuits

The following figures demonstrate typical TTL-to-RS232 conversion circuits.

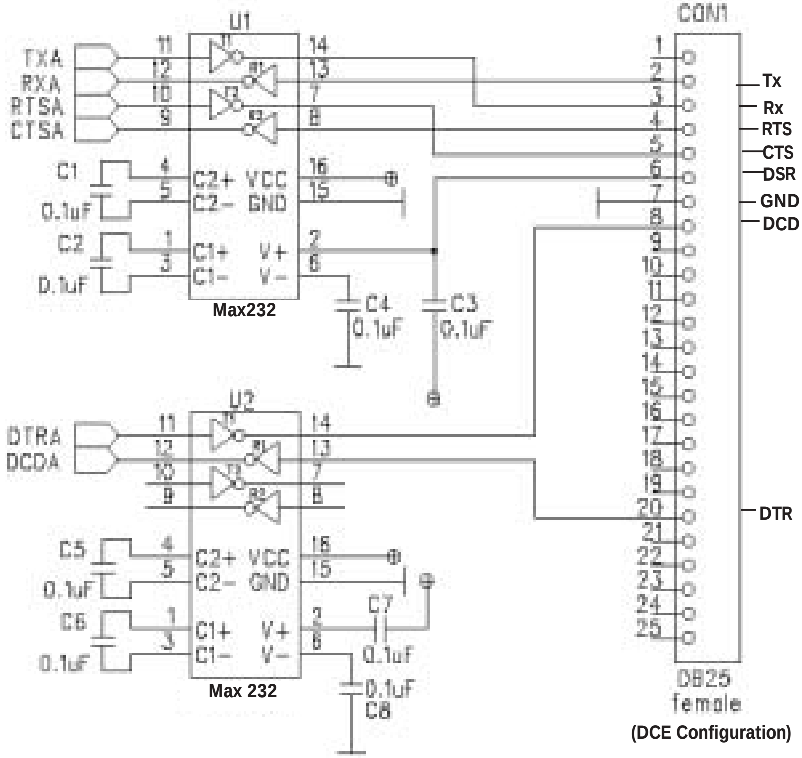
Figure F-1: TTL-RS232 (DB-9) Circuit



Note: *If you need just the DTR and DCD or RTS and CTS signals, you can use only one Max232 chip.*

Figure F-2: TTL-RS232 (DB-25) Circuit

Circuit consists of:
- 2 Max232 compatible chips
- 8 capacitors
- 1 DB-25 connector



Note: *If you need just the DTR and DCD or RTS and CTS signals, you can use only one Max232 chip.*

Warranty Statement

Lantronix warrants for a period of ONE years from the date of shipment that each Lantronix Device 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, in 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**.