SBL2e Dual Serial-to-Ethernet User's Manual

Table of Contents

1. Overview

The NetBurner SBL2e is a dual serial-to-Ethernet device that will enable your serial device to communicate on an Ethernet network using TCP or UDP. In addition to the serial capabilities, the SBL2e also provides a separate network connection command channel for control of digital I/O signals and analog-to-digital readback. A list of features and functions is below:

- 2 Serial UARTs, TTL level, baud rates from 300 to 115,200
- 4 12-bit analog to digital converters, 0-3.3VDC input
- Up to 10 Digital inputs/outputs, also referred to as GPIO
- 10/100 Ethernet
- Serial to Ethernet capability through TCP or UDP
- DHCP and Static IP addressing
- Custom data packetization options
- Web page configuration
- AT command set configuration and operation
- 3.3VDC input power @ 250mA max

2. IPSetup Software Utility

Setup and configuration of your SBL2e requires the IPSetup.exe Windows PC utility available for download at www.netburner.com, in the Support > Public Downloads section: http://www.netburner.com/support/public_downloads.html

3. Initial Configuration

Before you can begin using your SBL2e, three areas of configuration are required to tell your SBL2e what type of serial interface you want to use, the network IP address network port number, and the serial data baud rate.

1. Hardware configuration
2. Network configuration
3. Operational configuration

3.1 *Serial Port Hardware*

The SBL2e has two asynchronous UART type serial ports, referred to as Port 0 and Port 1. The SBL2e provides to TTL level UART serial ports:

Port 0 signals : TX, RX, RTS, CTS
Port 1 signals : TX, RX

If you need RS-232 electrical levels, then you must add the appropriate level shifters to the TTL signals.

The “Debug Serial Port”, or Debug Port for short, is used to display status messages from the application. If your application needs only a single serial port we recommend setting the other serial port to “Debug”.

3.2 *Network Configuration*

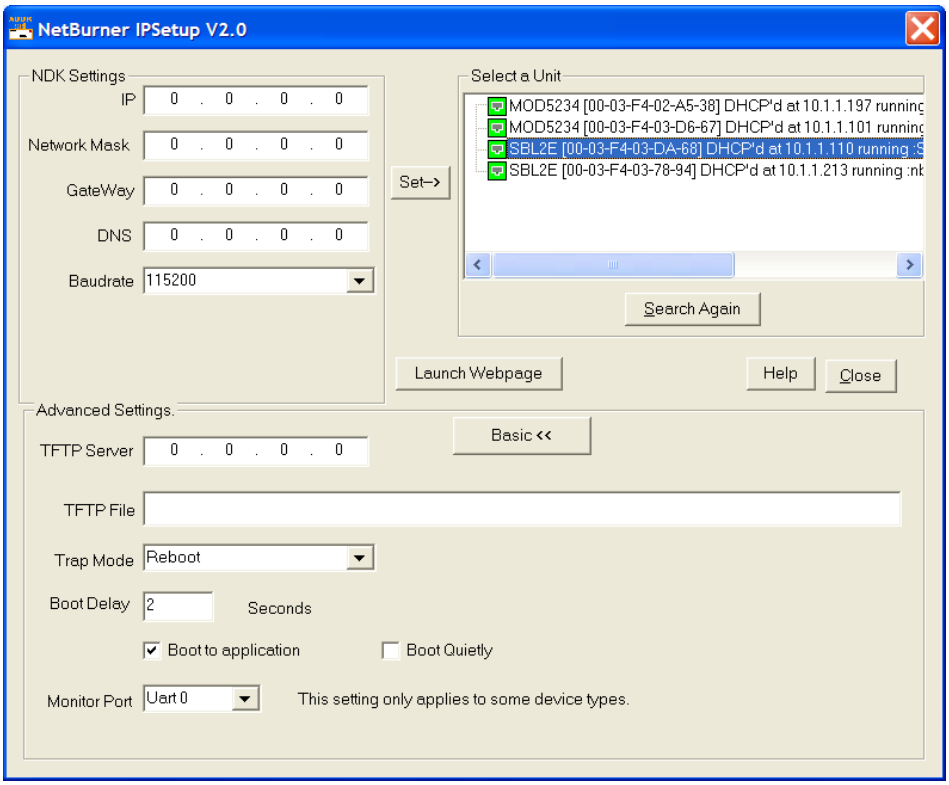
1. Run IPSetup.exe (by double clicking its icon). This program is located on the SBL2e product page at www.netburner.com. To view the Advance Settings, click on the Advanced button (the button name will change to Basic). In this example, I am using Uart 1 as my Monitor port (screen shot below).

2. Locate your SBL2e in the "Select a Unit" pane by matching its MAC address. The MAC address is located on the bottom of your SBL2e. If your SBL2e device does not appear in the list box, verify the power, speed, and link LEDs are illuminated, and click the Search Again button. If you are still unable to see your SBL2e, remove power, correct any cabling errors, reapply power, and click the Search Again button. Note: IP Setup uses a UDP broadcast protocol and will not operate through a router.

   FIREWALLS: IPSetup uses UDP and TCP port number 20034. If your computer’s firewall is blocking this port number you will need to either disable the firewall, or add a rule to the firewall to allow communication on this port number.

3. If your network supports DHCP (factory default): The assigned IP Address will appear in the "Select a Unit" pane. Write down this address. If your network does not support DHCP, configure the IP Address and Network Mask fields as shown in the screen shot below. If you need help selecting values, please read the "Selecting an IP Address" section at the end of this guide. After you have entered all of your values, click the Set button in the center of the IP Setup window to configure your SBL2e with its new parameters. Note: If you do not click the Set button, your values will not be saved. If you have multiple NetBurner devices, make sure you selected your SBL2e in the "Select a Unit" pane (as shown in the screen shot) before you input your information.

A screen-shot of IPSetup is shown below:



3.3 *Operational Configuration*

Once the network parameters are, you can use the web server interface or serial AT commands to modify the settings of your SBL2e. We recommend using the web interface first to familiarize yourself with the operation of the SBL2e. To access the web page, click on the Launch Webpage button in IP Setup, or you can open your web browser, and enter the numeric IP Address in the address field (e.g. http://10.1.1.110).

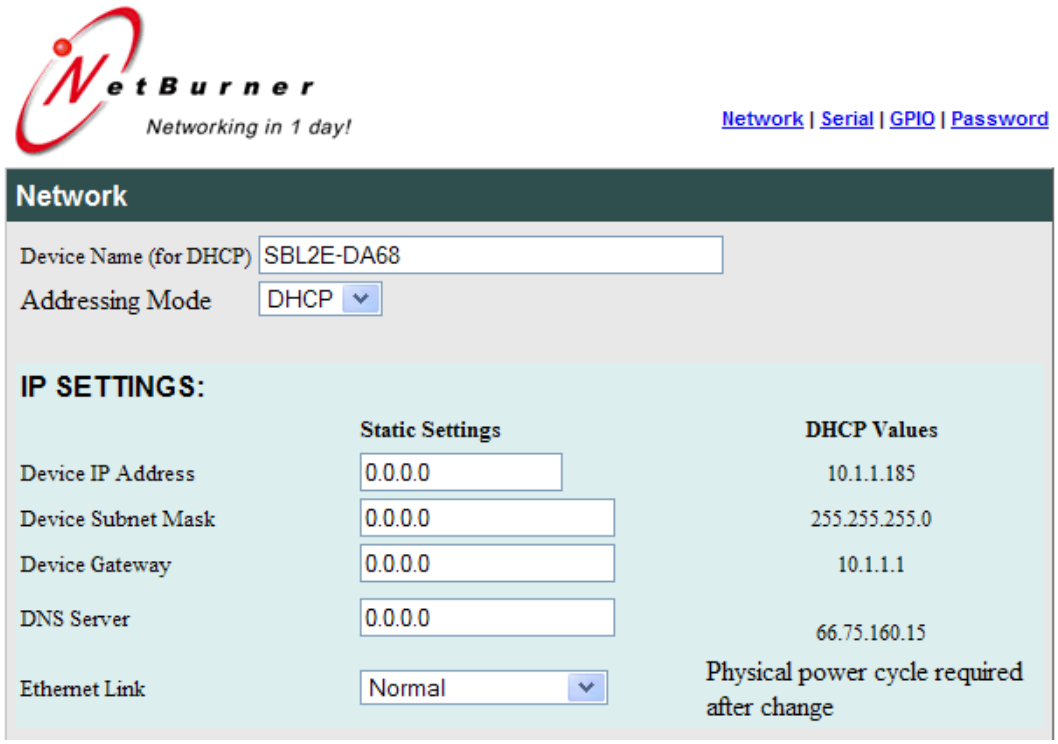
4. Web Page Configuration

Once you have configured the IP address of your device you can connect to the SBL2e web server to configure your serial and network settings. The SBL2e can be configured in one of 3 modes:

1. **TCP Server**: Listen for incoming TCP connections.
2. **TCP Client**: Connect to a target network address when serial data is available, or you can also establish a network connection on SBL2e power-up.
3. **UDP**: Send and receive using UDP packets

4.1 *Network IP Settings Configuration*

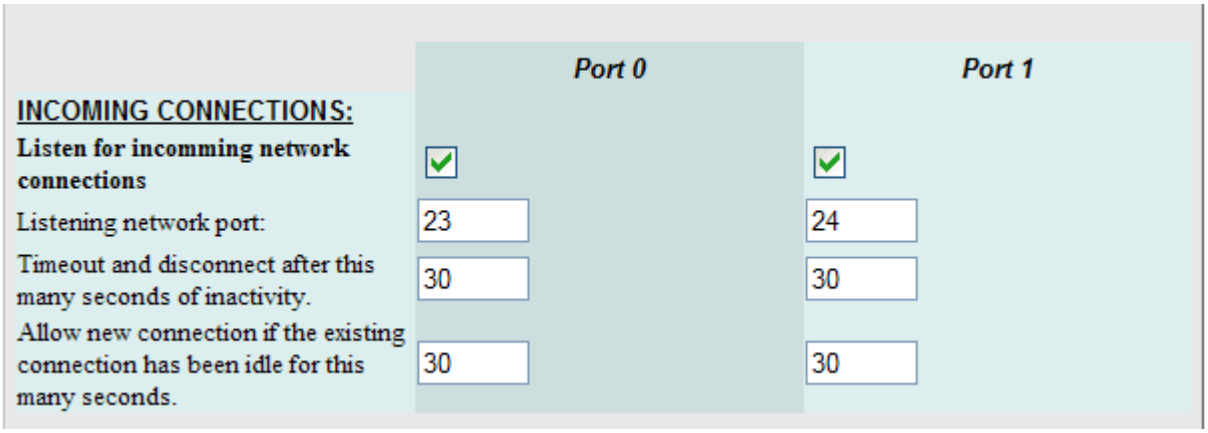
The first section of the Network Configuration page is used to select DHCP or static IP addressing. If you select DHCP, and the you have a DHCP server on your network, the DHCP assigned values will be displayed. To select a static IP address chance the Address Mode to Static, and enter your values in the Static Settings fields.



Device Name	Specifies the device name to send to your DHCP server.
Address Mode	Select between DHCP and Static IP address settings.
Device IP Address Device Subnet Mask Device Gateway Device DNS	If Address Mode is set to DHCP, your DHCP server will provide these values and they will be displayed in the DHCP Assigned Values column. If you wish to specify these values yourself set the Address Mode to "static" and type the values in the appropriate fields. Note that if you do not specify a Gateway or DNS server you will not be able to communicate outside your LAN.
Ethernet Link	Normal = Auto-negotiate. You may also force 10MB or 100MB modes.

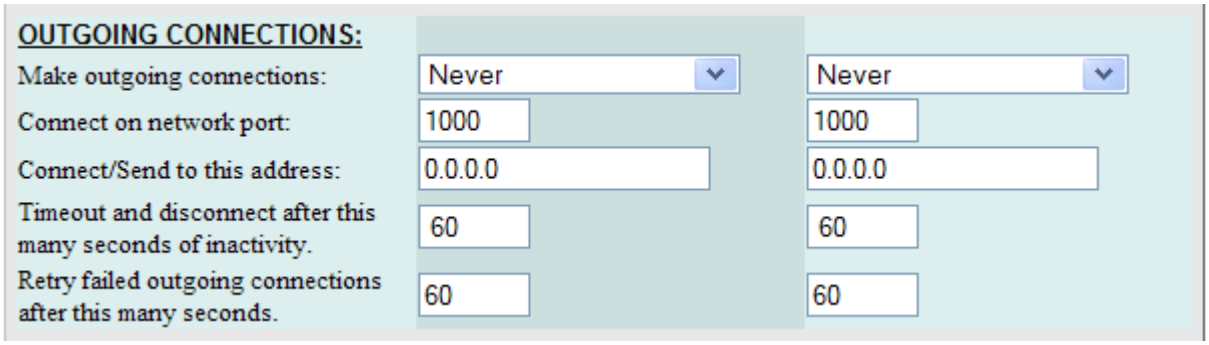
4.2 *Incoming Connections (Server Mode)*

This section configures the SBL2e in a device server mode in which it listens for incoming TCP connections (or UDP packets if enabled) for each serial port.



Listen for incoming network connections	Select checkbox to enable the port to listen for incoming TCP connection requests or UDP packets. Checking this box will override the TCP Client mode.
Listening network port	Port number to listen on. The default port is 23 (telnet). The listen port numbers for Port0 and Port1 must be different.
Timeout and disconnect after this many seconds of inactivity.	Terminate TCP connection if no incoming network data or outgoing serial data has occurred. This is useful because there is no way to detect if a client has crashed or abnormally terminated unless unacknowledged data exists and times out. A value of 0 disables this feature. This field has no effect in UDP mode.
Allow new connection if the existing connection has been idle for this many seconds.	Similar to the disconnect timeout, but does not disconnect a connection until a new connection is requested. A value of 0 disables this feature. This field has no effect in UDP mode.

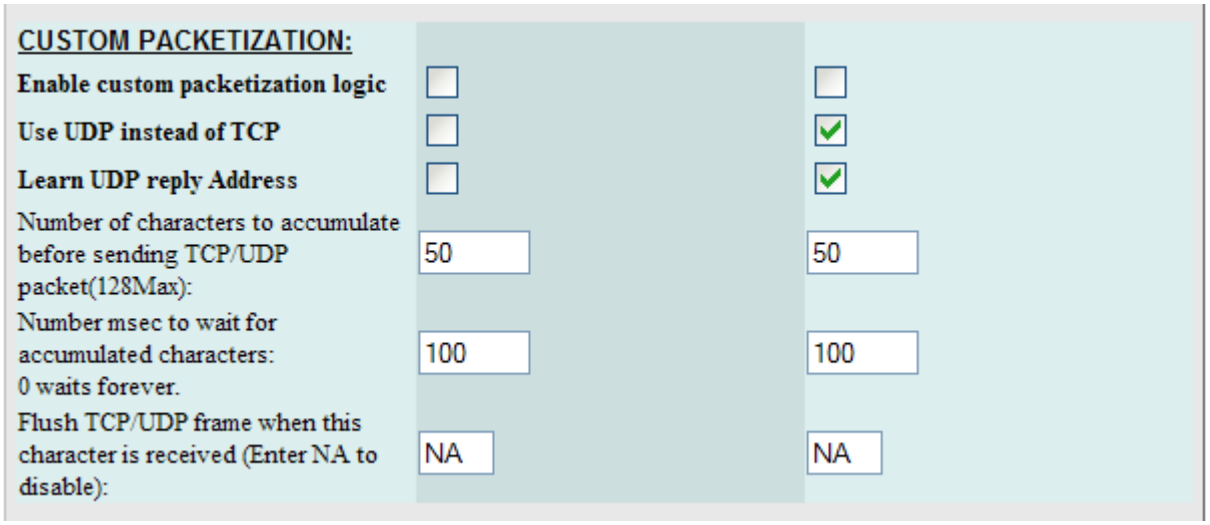
4.3 *Outgoing Connections (Client Mode)*



Make outgoing connections	Selects between connect on power-up for a permanent TCP connection, or make a TCP connection when serial data available. This feature has no effect on UDP.
Connect on network port	Specifies destination TCP or UDP port number
Connect to this address	Specifies destination IP address for TCP or UDP. When using UDP you must specify an address and port number in this section, or enable the Learn UDP Reply Address feature in the Custom Packetization section.
Timeout and disconnect after this many seconds of inactivity.	Terminate TCP connection if no incoming network data or outgoing serial data has occurred. This is useful because there is no way to detect if a client has crashed or abnormally terminated unless unacknowledged data exists and times out. A value of 0 disables this feature. This feature has no effect on UDP.
Retry failed outgoing connections after this many seconds.	Number of seconds to wait before retrying an outgoing connection. This feature has no effect on UDP.

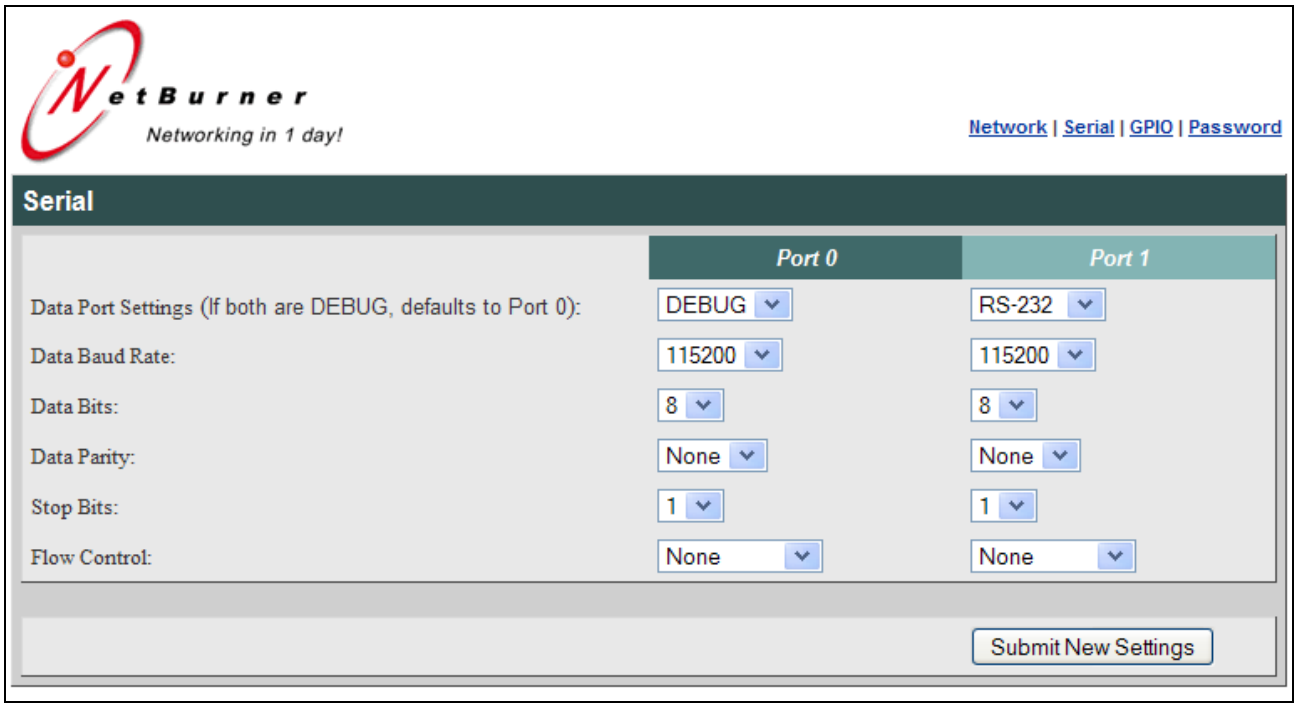
4.4 *Custom Packetization*

Custom packetization can apply to TCP and UDP communication. Unlike TCP, UDP is a connectionless protocol. The SBL2e provides two methods to determine the destination network IP address: you can specify a static IP address, or you can use the source IP address of the last received UDP packet by selecting the "Learn UDP Reply Address" checkbox.



Use custom packetization logic	Enables/disables custom packetization settings
Use UDP instead of TCP	Check to enable UDP
Learn UDP reply Address	Send outbound serial to the IP address from the last received UDP packet. Useful for clients that may have changing IP addresses. If not using this feature, you must specify the destination IP address and port number in the Outgoing Connections section.
Number of characters to accumulate before sending TCP/UDP packet	Maximum number of characters to accumulate from the serial port before sending them out the network port. This setting will be overridden if the accumulation delay time setting is used and the delay time expires.
Number msec to wait for accumulated characters (0 waits forever)	Maximum wait time in milliseconds for received serial characters before sending any available characters out the network port.
Flush TCP frame when this character is received (Enter NA to disable)	Send all accumulated serial data upon receipt of this character from the serial port. Enter the decimal value of the character. For example, a line feed is 10.

5. Serial Settings



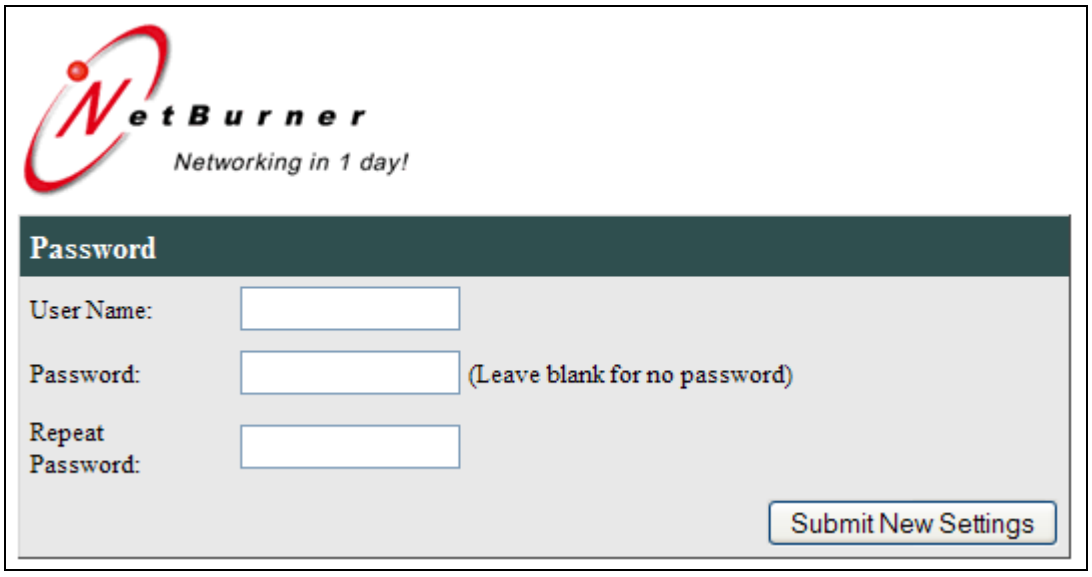
The SBL2e provides two TTL level UARTs, Port 0 and Port 1. The RS-232 setting will enable the TTL level signals for transmit, receive, RTS and CTS, but you will need to add external level shifters if you need the RS-232 electrical signals instead of 0-3.3VDC TTL

RS-232 Select for use as TTL or if you have a RS-232 level shifter.
DEBUG Specifies whether or not to use the serial port as stdin, stdout and stderr for serial status messages or debugging.

Data Port Settings (If both are DEBUG, defaults to Port 0)	Select serial mode RS-232 or DEBUG.
Data Baud Rate	Set serial baud rate
Data bits	Serial data bits
Data parity	Serial parity
Stop bits	Number of stop bits
Flow control	Set to None for no flow control. If using RS-232 or TTL valid selections are None, Xon/Xoff software flow control, or RTS/CTS hardware flow control.

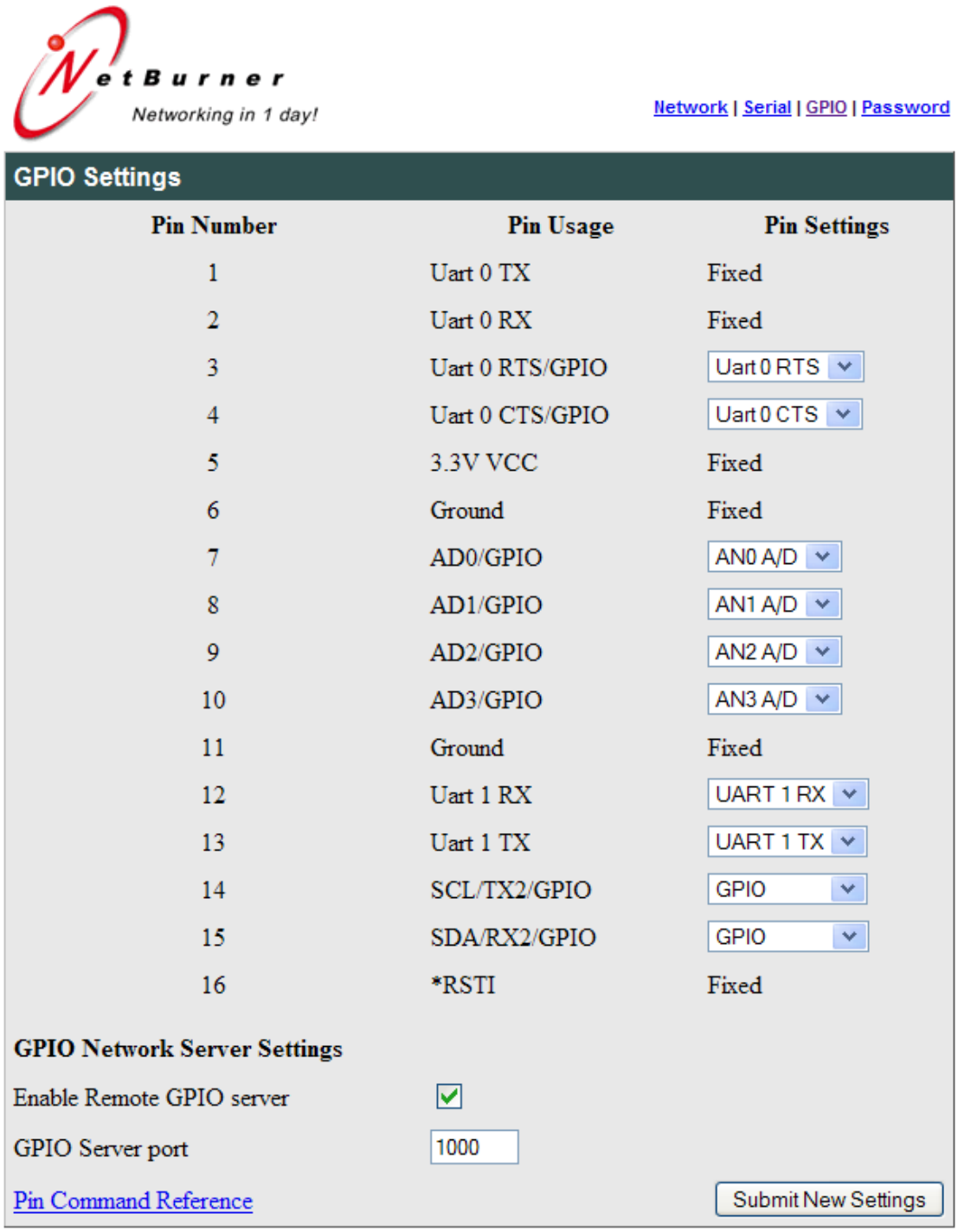
6. Password Settings

Use the Password Settings screen to enable, disable or change an existing password. Only one password is allowed. To clear a password leave the fields blank and click on submit.



7. Multi-Function Pin Control

The 16-pin interface connector of the SBL2e has 10 signals with multiple functions, such as general purpose digital I/O, analog to digital conversion, and serial hardware handshaking. The GPIO Settings configuration screen enables you to select the functions of the multi-function pins.



7.1 *Multi-Function Pin Control*

In addition to Serial-to-Ethernet capabilities, the SBL2e provides external control and readback functionality for General Purpose Input/Outputs and Analog to Digital inputs.

- The function of each multi-function pin is set in the GPIO configuration web page (GPIO is the most common function).
- The "GPIO Network Server" on the SBL2e listens on the TCP port number specified in the "GPIO Server Port" field. For a quick test, you can use the telnet utility: "telnet <ip address> 1000" for the default port of 1000.
- Some pins are fixed, but are shown for reference.
- The pin configuration can only be done through the HTML web page or serial AT commands.
- Access to the pin functions can only be done through a TCP connection to the GPIO Network Server (not the serial port).

7.2 *Command Syntax*

Once you have established a TCP connection to the GPIO Network Server, the following commands can be used, where "xx" is a value from 03 to 15. All commands and queries must be terminated by a line feed character: 0x0A. You may also terminate with a carriage return and line feed; the carrage return will be ignored by the parser.

General Purpose I/O Functions:

Pxx = 0	Set pin output low.
Pxx = 1	Set pin out put high.
Pxx = I	Set pin as input.
Pxx?	Read input value. Note that this command will automatically program the pin as an input. The value of an output pin cannot be read. The return value is 0 or 1 in the format: "0, OK\r\n" or "1, OK\r\n".

Examples:
"P14=0\n" returns "0, OK\r\n"
"P15?\n" returns "1, OK\r\n" if the pin is logic high

Analog to Digital Functions:

Pxx?	Read input value. The value read is a number from 0-4095 representing the number of counts. The counts represent a ratiometric value to the 3.3V reference voltage. The maximum voltage input is 3.3V.

Example:
"P07?\n" , returns "1234, OK\r\n" if the number of counts was 1,234.

Network Functions:

X	If the TCP client application, such as telnet, sends an 'X', the connection with the GPIO Network Server will be terminated.

Example:
"X\n" , returns "0, Closing\r\n"

Character Echo:
Echos characters back to TCP client. Default is echo off.

E	Enable echo
e	Disable echo

Example:
"E\n" , returns "0, Echo on\r\n"

Firmware Version :

V	Query firmware version

Example:
"V\n" , returns "0, Firmware Version: SBL2e 1.3 8/5/2009\r\n"

Command and Query Return Strings:
Each command and query will return an integer value followed by a ',' and optional text to provide details on the response.

- Any successful command will return: "0, OK\r\n".
- Any command error will return a negative number, followed by an explanation of the error. For example, sending the command "abcd\n" returns: "-3, Syntax Error\r\n".
- Any readback/query response will be the value of the readback followed by the status. For example, sending "P07?\n" could return "2034, OK\n". The 2034 is the decimal number of counts.
- Any readback/query that generates an error will return a negative number instead of the readback value.

Error Codes:

-1	Pin not in GPIO mode
-2	Invalid pin number
-3	Command syntax error

Note: At this time the I2C functions are not supported.

7.3 *Machine Command Mode*

The "Machine Command Mode" is so named because the data sent and received is in hex or binary, as opposed to the AT Command Mode which is more of a human readable text syntax and format. All non-programmable pins such as power, ground and /RSTI, will read back as 0 and programming commands will have no effect on them.

Syntax:

- All commands are terminated with a linefeed, which is a hexadecimal value of 0x10, also represented by a '\n' character in C. A carriage return value of 0x13 (or '\r') is ignored, so commands may be terminated with '\n' or "\r\n".
- Command parameters are in hexadecimal format, with the most significant bit (MSB) first.

Return Values

- Commands to set parameters will return a status value 0 on success, or 1 for a syntax error.
- Commands that return values will be in format: "<status>,<result>", where status is 0 (success) or 1 (syntax error) and the return value is one or more hexadecimal values, depending on the command. Hexadecimal values have the MSB first.
- Return values are terminated by "\r\n".

Command Summary:

MGxxxx	GPIO Enable: Set pin configuration, primary function (0) or GPIO mode (1).
MG	GPIO Enable Query: Return primary/GPIO pin configuration.
MRxxxx	GPIO Direction: Set the GPIO pin direction, input (0) or output (1).
MR	GPIO Direction Query: Return GPIO pin direction.
MOxxxx	GPIO Output Set: Set individual GPIO output pins, low (0) or high (1).
MO	GPIO Output Query: Return programmed state of all GPIO output pins.
MH	Set all GPIO output pins high (1)
ML	Set all GPIO output pins low (0)
MP	Save current settings as the power-on state
MQ	Query all A/D and GPIO values
MT	Terminate the GPIO command TCP connection
M?	Return all GPIO input and output pin values, low (0) or high (1).

7.4 *Machine Command Examples*

7.4.1 Set Pins 14 and 15 to GPIO Outputs

```
MG6000      // Set pin configuration to GPIO mode for both pins
0           // Return value of 0 for success
MR6000      // Set GPIO pin direction to output for both pins.
0           // Return value of 0 for success
MO6000      // Set GPIO output pin value to 1 for both pins
0           // Return value of 0 for success
MO          // Query all output pin values
0,0x6000    // Success, driving outputs high on pins 14, 15
MO4000      // Set pin 15 low, pin 14 high
0           // Success
MO          // Query all output pin values
0,0x4000    // Pin 15 low, pin 14 high
M?          // Query all input and output pins
0,0x5BCC    // Pin 15 low, pin 14 high, remaining pins report input value.
```

7.4.2 Set All GPIO Pins as Inputs and Readback

```
MG7BCC      // Set pin configuration to GPIO mode for all GPIO capable pins
0           // Return value of 0 for success
MR0000      // Set GPIO pin direction to input for all pins.
0           // Return value of 0 for success
M?          // Read all GPIO input pin values
0,0x3BCC    // Return value of 0 for success (note pins are floating inputs on dev. board)
```

7.4.3 Read A/D Inputs

```
MG03C0      // Set pin configuration to A/D primary function for all 4 A/D inputs
0           // Return value of 0 for success
MQ          // Read A/D inputs

// 0=success, A/D values of pins 7,8,9,10, all GPIO inputs as a 16-bit word
0, 0x4F38, 0x4910, 0x42A0, 0x4FE0, 0x3BCC
```

8. AT Commands

As an alternative to the web page configuration, you can configure the SBL2e over a serial port using an AT command format. The configuration sequence is:

1. Enter command mode
2. Send configuration change commands
3. Save changes and exit

The AT command operation is different for configuration commands versus GPIO pin programming commands. Please refer to "GPIO Server and Analog to Digital Readback" later in this section for additional details.

8.1 *Enter and Exit Command Mode*

AT configuration commands can only be processed when the SBL2e is in "AT Command Mode".

To enter AT command mode:

1. Pause for 1 second (send no data).
2. Send "+++".
3. Wait for 1 second.
4. The device will respond with "\r\nOK".

To exit command mode:

1. Wait 30 seconds for timeout, any changes will be lost.
2. Send "AT&X" or "ATO" to exit and discard any changes.
3. Send "AT&P" to exit and save new settings.
4. Send "AT&F" to reset all settings to factory default (does not save).

Returns:

1, Returning to active connection
2, Returning to listening connection
3, Returning to active UDP mode
0, Returning to idle

Exiting command mode with any changes terminate all existing TCP connections.

8.2 *Command Syntax*

The format of an AT command is:

```
AT#<command>=<parameter>,<parameter>, ….. <cr>
```

The format of an AT query is:

```
AT#<command>?<cr>
```

Syntax rules:

- All white space outside quotations is ignored
- All commands and queries are terminated by a carriage return <cr> (decimal value 13), if a line feed <lf> follows a <cr>, it will be ignored.

8.3 *System/Network Configuration*

Example: `AT#SYSIP=10.1.1.100<cr>`

```
#SYS
     IP=<ip address>
     MK=<mask ip>
     GW=<gateway ip>
     DN=<dns ip>
     DH=1            // To enable DHCP, set IP to 0.0.0.0.
     NB=<device name>

     UN=<string>
     PW=<string>

     GS=<enable/disable gpio server>     // 0 = disable, 1 = enable
     GP=<gpio server listen port>

#CUR //Get current active values
     IP?
     MK?
     GW?
     DN?
     ST?   // Return port status for current port
     S0?   // Return port status for port 0
     S1?   // Return port status for port 1
```

Return Values:

- Connected to IP xx.xx.xx.xx
- Listening on port xx
- UDP mode with learned send-to IP Address : xx.xx.xx.xx
- UDP mode send to Address : xx.xx.xx.xx
- Idle

8.4 *Serial Port Configuration*

Example: `AT#SER0LN=0<cr>`

```
#SERn                  // n = serial port number, 0, 1
      NP=< T|U|L >     // network protocol T = tcp, U = udp, L = udp learn
      SM=<R|D|H|F >    // R=RS232, D=debug, H=Half duplex RS485, F=Full duplex RS485
      LN=< 0|1 >       // 1 = listen for connection 0= don't listen
      BR=<baud rate >
      DB=< data bits > // 7, 8
      PR=< N|O|E >     //  none, odd, even
      ST=< stop bits > // 1, 2
      FL=< N|S|H >     // none, software or hardware

      SP=<listen network port number>
      SD=<disconnect timeout in seconds>
      SO=<override timeout in seconds>

      CM=<N|P|R>       // outgoing connection mode: never (listen mode only),
                           power-up or upon received serial data
      CI=<dest ip name/addr>
      CP=<destination port number>
      CD=<disconnect timeout in seconds>
      CR=<retry timeout in seconds>
```

8.4.1 Custom Packetization

The custom packetization settings provide control over when serial data is packetized and sent to the destination network host. You can configure the SBL2e to send serial data after a certain number of characters have been received, a certain amount of time has expired, or upon receipt of a specific character.

Example: `AT#SER0LNPE=1<cr>  // enable packetization options`

`PE=<0 | 1>`	`0 = disable packetization, 1 = enable`
`PN=<1-128>`	`Number of serial characters to accumulate before sending a packet.`
`PT=<0-32768>`	`Number of milliseconds since last character to wait before sending a packet. A value of 0 waits forever.`
`PC=<hex char>`	`Sends all accumulated serial data upon receipt of the specified ASCII character. The character value must be entered as a hexadecimal number.`

Return Values:

A '?'at the end of the command will return the setting. For example, `AT#PE?<cr>` will return a value of 0 or 1.

8.5 *GPIO Server and Analog to Digital Readback*

In addition to the system and serial configuration, you can also use AT commands to send commands and receive status information from the GPIO Server. For example, you can set pins GPIO pins high or low, or read the analog-to-digital inputs.

A very important point to remember when accessing the GPIO and A/D pins is that the operations will occur immediately upon issuing the AT command. The typical sequence of events in this case are:

1. Enter command mode
2. Send configuration change commands
3. Exit command mode with "AT&X" or "ATO" to avoid losing any existing TCP serial connections.

Example: `AT#P07?<cr> or AT#P07M=GPIO<cr>`

```
#Pnn                    // Where n or nn is the pin number
#Pn
      ?                 // Read a GPIO or A/D pin, ERR if pin is not a GPIO or A/D.
     =<0|1>             // Write GPIP pin high or low, ERR if not GPIO
     M=<mode>           // Modes: GPIO, RX, TX, RTS, CTS, AD
     M?                 // Return the current mode of the pin
#PD?                    // Return the system pin current drive level: 0 = 2mA, 1 = 10mA
#PD=<1|0>               // Set all pin current drive levels: 0 = 2mA, 1 = 10mA
```

8.6 *AT Command Examples*

The following examples display the full transcript of serial communication including commands sent and replies from the SBL2e. Commands are marked in red, responses are marked in blue.

8.6.1 Changing the System IP Address

This example first queries the current IP address, and then changes the current IP to 10.1.1.79. Once the IP address change request is made, the settings are saved, and the device reboots.

```
+++
OK>AT#CURIP?10.1.1.99
OK>AT#SYSIP=10.1.1.79
OK>AT&PIP Address Changed Rebooting

Waiting 2sec(s) to start 'A' to abort
```

8.6.2 Configuring the Serial Server Listen Port

This example first queries the current TCP Serial Server listening port number, and then changes the current port number to 30. Once the port number change request is made, the settings are saved. No system restart was required.

```
+++
OK>AT#SER0SP?23
OK>AT#SER0SP=30
OK>AT&P2,Returning to listening connection
```

8.6.3 Configure Outgoing Network Client Connection

This example changes the client connection's IP and port. Once the changes are made, the settings are saved. No system restart was required.

```
+++
OK>AT#SER0CI=10.1.1.78
OK>AT#SER0CP=30
OK>AT&P2,Returning to listening connection
```

8.6.4 Read Analog to Digital Input Channel 1

This example reads the A2D pin 8. The value is immediately returned.

```
+++
OK>AT#P08?2248
```

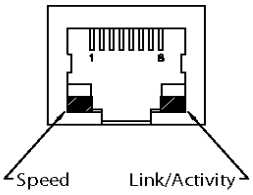
8.6.5 Change Pin to GPIO and Set Output to High

This example checks the current pin mode, and finds it in a2d channel 1. The mode is then updated to GPIO, and set to high.

```
+++
OK>AT#P08M?AD1
OK>AT#P08M=GPIO
OK>AT#P08M?GPIO In=1
OK>AT#P08=1
OK>
```

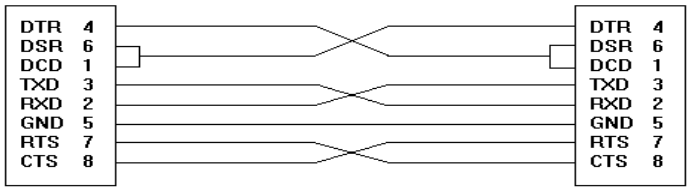
9. LEDs

Power LED: Illuminated while power is applied.
LED1 on RJ-45: Ethernet speed – 10 (off) or 100 (on)
LED2 on RJ-45: Link and data activity



10. RS-232 NULL Modem Wiring

The following table and diagram shows how to create a null modem cable/adapter for RS-232 connections.



11. Network IP Address Configuration

If you are part of an existing network, and are not using DHCP, you need to specify a Static IP Address and Network Mask. IP Addresses are required to route packets from place to place on an Intranet/Internet. If you are on your own LAN there are some standard private address ranges that are commonly used:

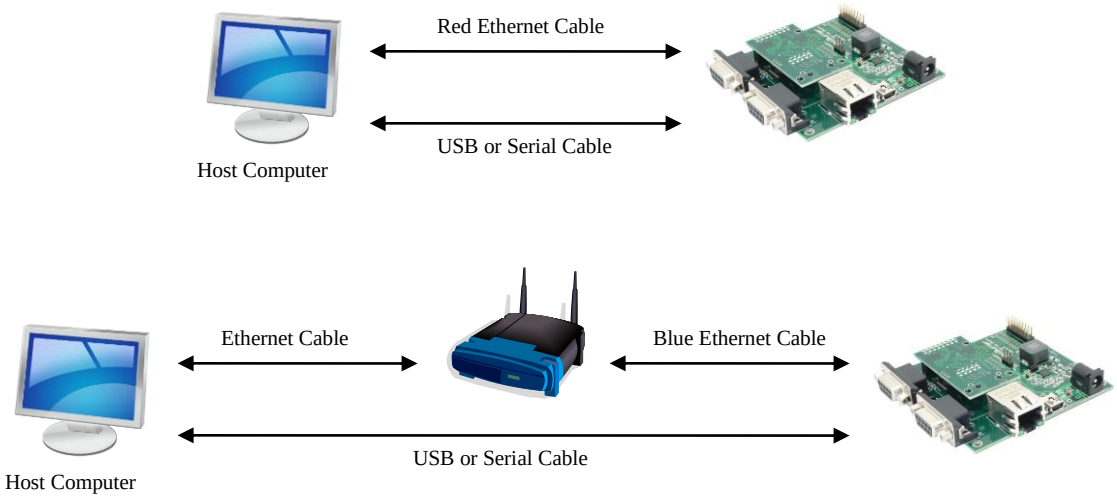
Class A: 10.0.0.0 to 10.255.255.255
Class B: 172.16.0.0 to 172.31.255.255
Class C: 192.168.0.0 to 192.168.255.255

12. Web Browsers and Proxy Servers

If you are working on a corporate LAN that uses a proxy server for Internet web browsing, you will need to exclude the IP Address of your SBL2e in your web browser's proxy server settings/preferences. Otherwise, an attempt to connect to a web page on the LAN will fail because the proxy server will attempt to route the request outside the LAN. For most web browsers, this can be accomplished in the advanced settings for the proxy server configuration. Set the Network Mask for your host computer's network adapter and your SBL2e to 255.255.255.0.

13. Testing with a Telnet Connection

A quick way to test the functionality of your Serial-to-Ethernet connection is with the Telnet program and an Serial Terminal Program, such as the NetBurner MTTTY program included on the CD-ROM. To run this test, configure your system as one of the two examples shown below. The Red Ethernet cable is a cross-wired cable that can be used for a direct connection without a network hub/switch. The Blue Ethernet cable is a standard straight-through Ethernet cable that should be used if you have a network hub/switch.



The objective of this example is to use a single host computer running telnet and a serial terminal program to send data in either direction. When you type text in the telnet window, it should appear in the serial terminal window and visa versa.

In the following example, an IP Address of 10.1.1.79 is be used for the SBL2e. Replace this IP address with the actual IP Address of your SBL2e.

14. Telnet to Serial Test Procedure

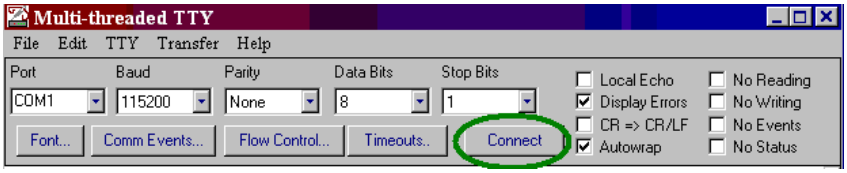
1. Connect your hardware in one of the above configurations.

2. Open a command prompt window on your host computer.

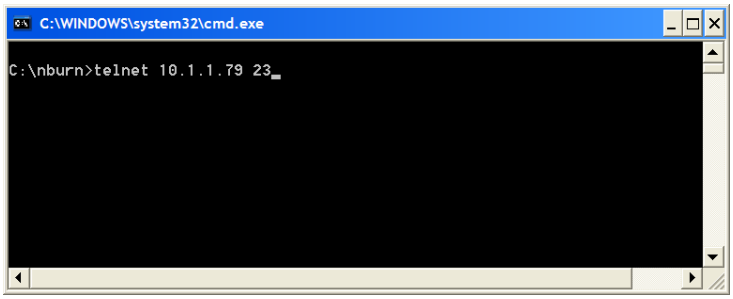
3. Verify everything is connected correctly by executing the command “ping 10.1.1.79” and pressing the Enter key on your keyboard (remember to substitute your actual IP Address)

4. Run either HyperTerminal or MTTTY. Set the baud rate to the value you assigned to the SBL2e during configuration (default is 115,200). To use MTTTY:

    - Run MTTTY.exe, you should see the screen below. The Port setting is the PC serial communication port of your host computer.
    - Click the MTTTY **Connect** button.



5. In your command prompt window, run Telnet by typing: “telnet 10.1.1.79 23” after the prompt, and press the Enter key. The “23” is the network port number of the SBL2e that is listening for incoming connections. This example syntax uses the default value of 23 for SBL2e serial port 0, which is connected to the USB serial interface and to the SBL2e serial port 0. If you have modified the listening network port number in the SBL2e web interface, replace this port number with the listening port number that you assigned.



6. At this point, anything you type in the Telnet window should appear in the serial terminal window and vice versa.

Revision History

Revision	Date	Description
1.0		Internal draft
1.1	8/10/2009	Initial release
1.2	12/17/2009	Added AT command set
1.3	2/22/2010	Clarified description of packetization settings to indicate the settings only apply to the outgoing serial to Ethernet direction.
1.4	5/5/2010	Added machine commands