



# **SBL2e Dual Serial-to-Ethernet User's Manual**

# Table of Contents

|  |    |
|--|----|
| <i>Table of Contents</i> .....                               | 2  |
| <b>1. Overview</b> .....                                     | 4  |
| <b>2. IPSetup Software Utility</b> .....                     | 4  |
| <b>3. Initial Configuration</b> .....                        | 4  |
| <b>3.1 Hardware Configuration</b> .....                      | 5  |
| <b>3.2 Network Configuration</b> .....                       | 5  |
| <b>3.3 Operational Configuration</b> .....                   | 6  |
| <b>4. Web Page Configuration</b> .....                       | 7  |
| <b>4.1 Network IP Settings Configuration</b> .....           | 7  |
| <b>4.2 Incoming TCP Settings (Server Mode)</b> .....         | 8  |
| <b>4.3 Outgoing TCP Settings (Client Mode)</b> .....         | 9  |
| <b>4.4 GPIO TCP Effects</b> .....                            | 10 |
| <b>4.5 UDP Mode Settings</b> .....                           | 11 |
| <b>4.6 Custom Packetization</b> .....                        | 12 |
| <b>5. Serial Settings</b> .....                              | 13 |
| <b>6. Password Settings</b> .....                            | 14 |
| <b>7. Multi-function Pin Control</b> .....                   | 15 |
| <b>7.1 Multi-function Pin Control</b> .....                  | 16 |
| <b>7.2 Command Syntax</b> .....                              | 16 |
| <b>7.3 Machine Command Mode</b> .....                        | 18 |
| <b>7.4 Machine Command Examples</b> .....                    | 19 |
| 7.4.1 Set Pins 14 and 15 to GPIO Outputs.....                | 19 |
| 7.4.2 Set All GPIO Pins as Inputs and Read Back.....         | 19 |
| 7.4.3 Read A/D Inputs .....                                  | 19 |
| <b>8. AT Commands</b> .....                                  | 20 |
| <b>8.1 Enter and Exit Command Mode</b> .....                 | 20 |
| <b>8.2 Command Syntax</b> .....                              | 21 |
| <b>8.3 System/Network Configuration</b> .....                | 21 |
| <b>8.4 Serial Port Configuration</b> .....                   | 22 |
| 8.4.1 Custom Packetization .....                             | 23 |
| <b>8.5 GPIO Server and Analog-to-digital Read-back</b> ..... | 24 |
| <b>8.6 AT Command Examples</b> .....                         | 25 |
| 8.6.1 Changing the System IP Address.....                    | 25 |
| 8.6.2 Configuring the Serial Server Listen Port.....         | 25 |
| 8.6.3 Configure Outgoing Network Client Connection .....     | 25 |
| 8.6.4 Read Analog-to-Digital Input Channel 1 .....           | 26 |
| 8.6.5 Change Pin to GPIO and Set Output to High .....        | 26 |

**9. LEDs .....27**

**10. RS-232 Null Modem Wiring .....27**

**11. Network IP Address Configuration .....27**

**12. Web Browsers and Proxy Servers .....28**

**13. Testing with a Telnet Connection .....28**

**14. Telnet to Serial Test Procedure.....29**

**Revision History .....30**

## 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 command channel for control of digital I/O signals and analog-to-digital read back. A list of features and functions is below:

- 2 serial UARTs, TTL level, baud rates from 300 to 115,200 bps
- 4 12-bit analog-to-digital converters, 0-3.3V DC 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
- 3.3V DC input power @ 250mA max

## 2. IPSetup Software Utility

Setup and configuration of your SBL2e requires the Windows-based “IPSetup.exe” utility, which is available for download at [www.netburner.com](http://www.netburner.com) in the Support → Public Downloads section:  
[http://www.netburner.com/support/public\\_downloads.html](http://www.netburner.com/support/public_downloads.html)

## 3. Initial Configuration

Before you can start using your SBL2e, three areas of configuration need to be set up: the serial interface to be used, the network IP address of the module, and configuration through the application web page interface.

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

### 3.1 Hardware Configuration

The SBL2e has two asynchronous UART serial ports (TTL level), referred to as Port 0 and Port 1:

Port 0 signals: TX, RX, CTS, RTS

Port 1 signals: TX, RX

If you need RS-232 or RS-485 (Port 0 only) 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 data port, then we recommend setting the other serial port to “DEBUG”.

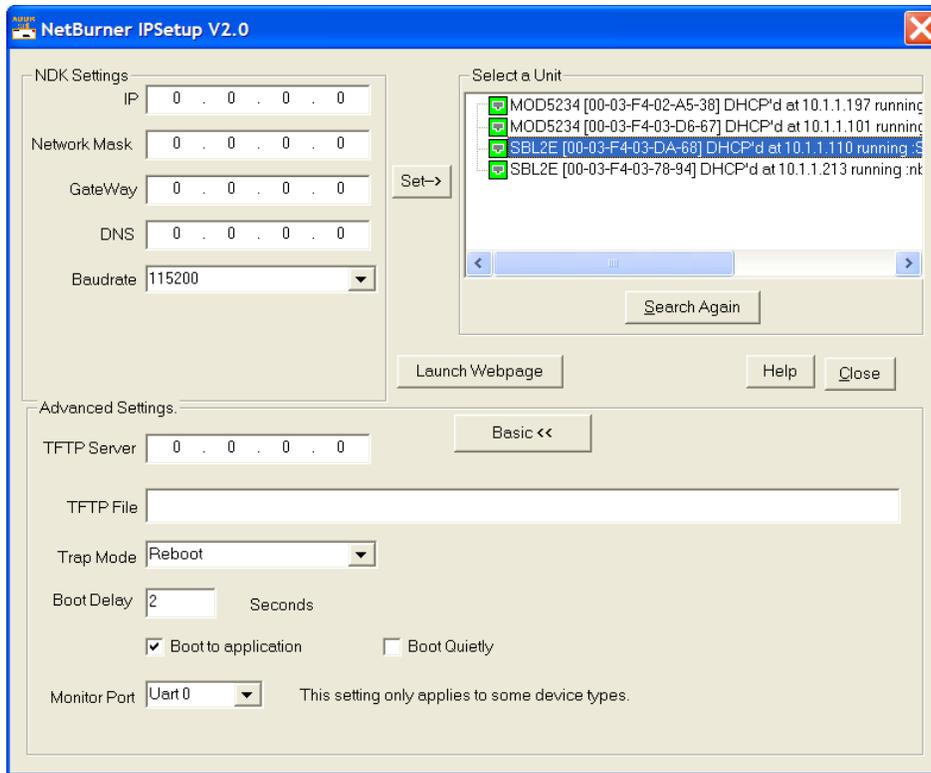
### 3.2 Network Configuration

1. Run the “IPSetup.exe” configuration utility by double-clicking its icon. This program is located on the SBL2e product page at [www.netburner.com](http://www.netburner.com). To view the advanced settings, click on the “Advanced...” button (the button name will then change to “Basic <<”). In this example, UART 0 is used as the monitor port (screenshot below).
2. Locate your SBL2e in the "Select a Unit" list box by matching its MAC address. The MAC address is located on the no-component side of your SBL2e module. If your SBL2e does not appear in the list box, then verify that the power and Ethernet speed/link LEDs are illuminated, and then click the “Search Again” button. If you are still unable to see your SBL2e, disconnect power and correct any cabling errors, reconnect power, and then click the “Search Again” button. Note that “IPSetup.exe” uses a UDP broadcast protocol, and therefore will not operate through a router to the Internet.

**Firewalls: “IPSetup.exe” uses UDP and TCP port number 20034. If your computer’s firewall is blocking this port number, then you will need to either disable the firewall or add a rule to the firewall to allow communication through this port number.**

3. If your network supports DHCP (factory default): the assigned IP address will appear in the "Select a Unit" list box – write down this address. If your network does not support DHCP, then configure the IP and network mask address fields as shown in the screen shot below. If you need help selecting values, then please read the “Network IP Address Configuration” section located near the end of this guide. After you have entered all of your values, click the “Set→” button in the center of the “IPSetup” window to configure your SBL2e with its new parameters. Note that if you do not click the “Set→” button, your values will not be saved to the target platform selected. If you have multiple NetBurner devices, make sure you select your SBL2e in the "Select a Unit" list box (as shown in the screenshot) before you enter your information.

A screenshot of “IPSetup” is shown below with the expanded advanced settings:



### 3.3 Operational Configuration

Once the network parameters are set, you can use the web server interface or serial AT commands to modify the settings of your SBL2e. We recommend using the web server interface first to get familiarized with the operation of the SBL2e. Click on the “Launch Webpage” button in “IPSetup” to access the web page, or you can open a web browser and enter the numeric IP address of the module 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 page to configure your serial and network settings. The SBL2e can be configured in one of three modes:

1. **TCP server:** Listen for incoming TCP connections through a specified port.
2. **TCP client:** Connect to a target network address and port number when serial data is available, or you can also establish a network connection on 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 you have a DHCP server on the network, the DHCP-assigned values will be displayed under the “DHCP Values” column. To select a static IP address, change the “Addressing mode” to “Static”, and enter your values in the fields under the “Static Settings” column.



[Network](#) | [Serial](#) | [GPIO](#) | [Password](#)

| Network               |   |  |
|-----------------------|---|--|
| Device name for DHCP: | <input type="text" value="SBL2E-DA68"/> |  |
| Addressing mode:      | <input type="text" value="DHCP"/>       |  |
| IP Settings           | Static Settings                         | DHCP Values                                |
| Device IP address:    | <input type="text" value="0.0.0.0"/>    | 10.1.1.191                                 |
| Device subnet mask:   | <input type="text" value="0.0.0.0"/>    | 255.255.255.0                              |
| Device gateway:       | <input type="text" value="0.0.0.0"/>    | 10.1.1.1                                   |
| DNS server:           | <input type="text" value="0.0.0.0"/>    | 66.75.160.15                               |
| Ethernet link:        | <input type="text" value="Normal"/>     | Physical power cycle required after change |

|   |  |
|---|--|
| Device name for DHCP  | Specify the device name to send to the DHCP server if available.   |
| Addressing mode   | Select between DHCP and static IP address modes.   |
| Device IP address<br>Device subnet mask<br>Device gateway<br>DNS server | If the addressing mode is set to DHCP, then the DHCP server (if available) will provide these values, which will be displayed in the “DHCP Values” column. If you wish to specify these values yourself, then set the addressing mode to “Static” and type the values in the fields below the “Static Settings” column. Note that if a gateway or DNS server is not specified, then the device will not be able to communicate outside the local area network (LAN). |
| Ethernet link   | “Normal” is auto-negotiation. You may also force 10 or 100 Base-T half-duplex modes.   |

## 4.2 Incoming TCP Settings (Server Mode)

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

| <b>Incoming TCP Settings</b>   | PORT 0                              | PORT 1                              |
|--|-------------------------------------|-------------------------------------|
| Listen for incoming network connections:   | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Listening network port:  | <input type="text" value="23"/>     | <input type="text" value="24"/>     |
| Timeout and disconnect after this many seconds of inactivity:                        | <input type="text" value="60"/>     | <input type="text" value="60"/>     |
| Allow new connection if the existing connection has been idle for this many seconds: | <input type="text" value="30"/>     | <input type="text" value="30"/>     |

|   |  |
|---|--|
| Listen for incoming network connections   | Check the box to enable the listening port to wait for incoming TCP connection requests or UDP packets. Note that checking the box will override the client mode.  |
| Listening network port  | Listen for incoming connections on the given port. The default port is 23 (also the default Telnet port) on Port 0, and 24 on Port 1. The listening port numbers for Ports 0 and 1 must be different.  |
| Timeout and disconnect after this many seconds of inactivity                        | Terminate the connection if no incoming network data or outgoing serial data has passed for the number of given seconds since the last data pass. This is useful since 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 | This is similar to the timeout-and-disconnect, but does not disconnect until a new connection is requested. A value of ‘0’ disables this feature. This field has no effect in UDP mode.  |

### 4.3 Outgoing TCP Settings (Client Mode)

**Outgoing TCP Settings**

|   |                    |                    |
|---|--------------------|--------------------|
| Make outgoing connections:                                    | Never ▾            | Never ▾            |
| Connect on network port:                                      | 1000               | 1001               |
| Connect/send to this address:                                 | (Enter IP address) | (Enter IP address) |
| Timeout and disconnect after this many seconds of inactivity: | 60                 | 60                 |
| Retry failed outgoing connections after this many seconds:    | 10                 | 10                 |

|  |  |
|--|--|
| Make outgoing connections                                    | Select between connecting on power-up for a permanent connection, or make a connection only when serial data becomes available. This feature has no effect on UDP.   |
| Connect on network port                                      | Specify the destination TCP or UDP port number.  |
| Connect/send to this address                                 | Specify the destination IP address for TCP or UDP. When using UDP, an address and port number must be specified in this section, or check the box to enable “Learn UDP reply address” under the “UDP Mode Settings” section.   |
| Timeout and disconnect after this many seconds of inactivity | Terminate the connection if no incoming network data or outgoing serial data has passed for the number of given seconds since the last data pass. This is useful since 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. |
| Retry failed outgoing connections after this many seconds    | Wait a given number of seconds before retrying an outgoing connection if it fails. This feature has no effect in UDP mode.   |

#### 4.4 GPIO TCP Effects

This section allows for the configuration of the GPIO pins to be in a particular state when a connection is made, or to initiate a connection based on the state of a GPIO pin. The pin being used must be set to GPIO in the GPIO web configuration page before these settings can be used.

**GPIO TCP Effects**

Set the following GPIO pin on connect (the pin must be set as GPIO out):

NA ▾

NA ▾

Make a connection when the following GPIO pin is true (the pin must be set as GPIO in):

NA ▾

NA ▾

|  |   |
|--|---|
| Set the following GPIO pin on connect (the pin must be set as GPIO out)                | Drive the GPIO pin to a desired state when a connection has been established. When setting this parameter, the state must be opposite of the GPIO out function selected in the GPIO web page. For example, if pin 3 going to a high state on connect is desired, then pin 3 must be set to "GPIO Out 0" in the GPIO web page, and "Pin 3 Hi" for this setting (Hi = 1, Low = 0). No effect takes place if "NA" is selected. |
| Make a connection when the following GPIO pin is true (the pin must be set as GPIO in) | Establish a connection once the GPIO pin is in a desired state. The pin must be set to "GPIO In" in order to use this setting. For example, if "Pin 7 Low" is selected, then a connection will be made once pin 7 is detected to go into a low state. No effect takes place if "NA" is selected.  |

## 4.5 UDP Mode Settings

Unlike TCP, UDP is a connectionless protocol. The SBL2e provides two methods to determine the destination network IP address: you can either specify an IP address manually, or use the source IP address of the last received UDP packet by checking the “Learn UDP reply address” box.

**UDP Mode Settings**

Use UDP instead of TCP:

Learn UDP reply address:

|                         |  |
|-------------------------|--|
| Use UDP instead of TCP  | Check the box to enable UDP communication mode.  |
| Learn UDP reply address | Check the box to send received serial data to the source IP address that sent the last received UDP packet. This is useful for clients that may have changing IP addresses. If not using this feature, then the destination IP address and port number must be specified in the “Outgoing TCP Settings” section. |

## 4.6 Custom Packetization

Custom packetization can apply to TCP and UDP communication.

**Custom Packetization**

Enable custom packetization logic:

Number of characters to accumulate before sending TCP/UDP packet (128 max):

Number of milliseconds to wait for accumulated characters ('0' waits forever):

Flush TCP/UDP frame when this character is received (enter "NA" to disable):

|   |   |
|---|---|
| Use custom packetization logic  | Enable (check) or disable (uncheck) custom packetization settings.  |
| Number of characters to accumulate before sending TCP/UDP packet              | This is the number of incoming characters to accumulate from the serial port before sending them out through the network port. This setting is overridden if the accumulation wait time setting (next option below) expires.            |
| Number of milliseconds to wait for accumulated characters ('0' waits forever) | This is the maximum time in milliseconds to wait for received serial characters before sending the accumulated data, if any, out the network port.  |
| Flush TCP/UDP frame when this character is received (Enter "NA" to disable)   | Send all accumulated serial data upon receipt of the given character from the serial port. Enter the decimal value of the character to be recognized. For example, the decimal representation for the "LF" (line feed) character is 10. |

## 5. Serial Settings



| Serial  | PORT 0                              | PORT 1                              |
|---|-------------------------------------|-------------------------------------|
| Data port settings (DEBUG defaults to port 0 if both are set to DEBUG): | DEBUG                               | RS-232                              |
| Data baud rate:   | 115200                              | 115200                              |
| Custom baud rate:   | 0                                   | 0                                   |
| Data bits:  | 8                                   | 8                                   |
| Stop bits:  | 1                                   | 1                                   |
| Data parity:  | None                                | None                                |
| Flow control:   | None                                | None                                |
| Allow AT commands:  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| AT attention command char:  | 43 (+)                              | 43 (+)                              |

The SBL2e provides two TTL level UARTs: Port 0 and 1. The “RS-232”, “RS-485 Half Duplex” (Port 0 only), and “RS-485 Full Duplex” (Port 0 only) data port settings will enable the TTL level signals. You will need to add external level shifters if RS-232/RS-485 electrical signals are needed instead of 0-3.3V DC TTL.

|                           |   |
|---------------------------|---|
| Data port settings        | Select the serial communication mode. Port 0 supports RS-232, debug, RS-485 half and full duplex; Port 1 supports only RS-232 and debug. If both serial ports are set to debug, then debug defaults to Port 0 only.   |
| Data baud rate            | Set the serial baud rate in bits per second.  |
| Custom baud rate          | Use for non-standard baud rates by typing in the baud rate value as a decimal number (in bits per second). “Custom” must be selected in the “Data baud rate” field in order for this custom baud rate to take effect. |
| Data bits                 | Number of data bits (5, 6, 7, or 8).  |
| Stop bits                 | Number of stop bits (1 or 2).   |
| Data parity               | Set the serial parity (odd, even, or none).   |
| Flow control              | Set to “None” for no flow control. Flow control is valid only for RS-232 mode, where both serial ports support XON/XOFF software flow control, but only Port 0 supports RTS/CTS hardware flow control.                |
| Allow AT commands         | Enable/disable the use of serial AT commands.   |
| AT attention command char | Specify the decimal representation of the character to use to enter AT command mode. The default is the industry standard ‘+’ character (its decimal value is 43).  |

## 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 then click on the “Submit New Settings” button.



[Network](#) | [Serial](#) | [GPIO](#) | **Password**

| Password         |   |
|------------------|---|
| User name:       | <input type="text"/>  |
| Password:        | <input type="text"/> <small>Leave blank for no password</small> |
| Repeat password: | <input type="text"/>  |

## 7. Multi-function Pin Control

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



**GPIO**

| Pin Number | Pin Usage              | Power-up Pin Settings | Current Setting |
|------------|------------------------|-----------------------|-----------------|
| 1          | UART 0 TX              | Fixed                 | Fixed           |
| 2          | UART 0 RX              | Fixed                 | Fixed           |
| 3          | UART 0 RTS / GPIO      | Uart 0 RTS ▾          | Uart 0 RTS      |
| 4          | UART 0 CTS / GPIO      | Uart 0 CTS ▾          | Uart 0 CTS      |
| 5          | 3.3 Vcc                | Fixed                 | Fixed           |
| 6          | Ground                 | Fixed                 | Fixed           |
| 7          | AD0 / GPIO             | AD0 A/D ▾             | AD0             |
| 8          | AD1 / GPIO             | AD1 A/D ▾             | AD1             |
| 9          | AD2 / GPIO             | AD2 A/D ▾             | AD2             |
| 10         | AD3 / GPIO             | AD3 A/D ▾             | AD3             |
| 11         | Ground                 | Fixed                 | Fixed           |
| 12         | UART 1 RX / GPIO       | UART 1 RX ▾           | Uart 1 Rx       |
| 13         | UART 1 TX / GPIO       | UART 1 TX ▾           | Uart 1 Tx       |
| 14         | SCL / UART 2 TX / GPIO | GPIO In ▾             | GPIO In=1       |
| 15         | SDA / UART 2 RX / GPIO | GPIO In ▾             | GPIO In=0       |
| 16         | *RSTl                  | Fixed                 | Fixed           |

Enable high current drive (low = 2 mA; high = 10 mA):

**GPIO Network Server Settings**

Enable remote GPIO server:

GPIO server port:

[Pin Command Reference](#)

## 7.1 Multi-function Pin Control

In addition to serial-to-Ethernet capabilities, the SBL2e provides external control and read-back 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 of the pins are fixed, but are shown for reference.
- The pin configurations can only be done through the GPIO configuration web page or through 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 a TCP connection is established 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 ["LF" = 10 (decimal) = 0xA (hexadecimal)]. You may also terminate with a carriage return ["CR" = 13 (decimal) = 0xD (hexadecimal)] and a line feed; the carriage return will be ignored by the parser.

### General-purpose I/O functions:

|       |   |
|-------|---|
| Pxx=0 | Set pin output low.   |
| Pxx=1 | Set pin output high.  |
| Pxx=I | Set pin as input.   |
| Pxx?  | Read the pin 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". |

Example:

```
"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 the pin input value. The value read is a number from 0 to 4095 representing the number of counts. The counts represent a ratiometric value to the 3.3-Volt reference voltage. The maximum voltage input is 3.3 Volts. |
|------|--|

Example:

"P07?\n" Returns "1234, OK\r\n" if the number of counts read was 1234.

### Network functions:

|   |   |
|---|---|
| X | If a TCP client application, such as Telnet, sends an 'X', then the connection with the GPIO network server will be terminated. |
|---|---|

Example:

"X\n" Returns "0, Closing\r\n"

### Character echo:

This command echoes characters back to the TCP client. Echo is disabled by default.

|   |               |
|---|---------------|
| 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 v1.58 Nov 17 2010\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 read-back/query response will be the value of the read-back followed by the status. For example, sending "P07?\n" could return "2034, OK\r\n". The value 2034 is the number of counts in decimal representation.
- Any read-back/query response that generates an error will return a negative number instead of the read-back value.

Error Codes:

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

### 7.3 Machine Command Mode

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

#### Syntax

- All commands are terminated with the line feed character (0x10 in hexadecimal representation; also represented as '\n' in the C language). The carriage return character (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 of 0 on success, or 1 for a syntax error.
- Commands that return values will be in the format: "<status>,<result>", where status is 0 (success) or 1 (syntax error) and the result is one or more hexadecimal values, depending on the command. Hexadecimal values display the most significant bit first.
- Return values are terminated by "\r\n".

#### Command summary

|         |   |
|---------|---|
| MGxxxxx | GPIO enable – Set the pin configuration: primary function (0) or GPIO mode (1). |
| MG      | GPIO enable query – Return the pin configuration values.                        |
| MRxxxxx | GPIO direction – Set the GPIO pin direction: input (0) or output (1).           |
| MR      | GPIO direction query – Return the GPIO pin direction values.                    |
| MOxxxxx | GPIO output set – Set individual GPIO output pins: low (0) or high (1).         |
| MO      | GPIO output query – Return the 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 initial 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 both pins to function as GPIO
0 // Return value of 0 for success
MR6000 // Set the direction of both GPIO pins to be outputs
0 // Return value of 0 for success
MO6000 // Set the output state of both GPIO pins to be high (1)
0 // Return value of 0 for success
MO // Query all output pin values
0,0x6000 // Success, driving outputs high on pins 14 and 15
MO4000 // Set pin 15 low (0) and pin 14 high (1)
0 // Return value of 0 for success
MO // Query all output pin values
0,0x4000 // Success, driving output low on pin 15 and high on pin 14
M? // Query all input and output pins
0,0x5BCC // Success, pin 15 is low, pin 14 is high, and the remaining
// pins report input values

```

### 7.4.2 Set All GPIO Pins as Inputs and Read Back

```

MG7BCC // Set all multi-function pins to GPIO
0 // Return value of 0 for success
MR0000 // Set the direction for all GPIO pins to be inputs
0 // Return value of 0 for success
M? // Read all GPIO input pin values
0,0x3BCC // Return value of 0 for success (note that pins are floating
// inputs on development board)

```

### 7.4.3 Read A/D Inputs

```

MGFC3F // Set the 4 A/D pins to their primary function (A/D inputs)
0 // Return value of 0 for success
MQ // Read the A/D inputs

// 0 = success; first 4 16-bit hex values are A/D readings for pins 7, 8,
// 9, and 10; last 16-bit hex value is the reading for all GPIO inputs
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 the 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 programming GPIO pins with machine commands. Please refer to the “GPIO Server and Analog-to-digital Read-back” sub-section for additional details.

### 8.1 *Enter and Exit Command Mode*

The AT command configurations 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 AT command mode:

1. Wait 30 seconds for time-out; 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.

Returns:

- 1 – Returning to active connection.
- 2 – Returning to listening connection.
- 3 – Returning to active UDP mode.
- 0 – Returning to idle.

Exiting AT command mode with any changes terminates all existing TCP connections.

## 8.2 Command Syntax

The following is the format for an AT command:

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

The following is the format for an AT command query:

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

Syntax rules:

- All white space outside quotations are ignored.
- All commands and queries are terminated by a carriage return <CR> (13 in decimal representation). 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 listening port>
```

```
#CUR
```

```
// Get current active values
IP? // Return the device IP address
MK? // Return the mask address
GW? // Return the gateway address
DN? // Return the DNS address
ST? // Return port status for current port
S0? // Return port status for port 0
S1? // Return port status for port 1
MA? // Return MAC address of the unit
```

Return values:

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

## 8.4 Serial Port Configuration

Example: AT#SER0LN=0<cr>

```
#SERn                // n = serial port number: 0, 1
NP=<T|U|L>          // Net protocol: T = TCP, U = UDP, L = UDP learn
SM=<R|D|H|F>        // R = RS232, D = DEBUG, H = RS485 HD, F = RS485 FD
LN=<0|1>             // 1 = listen for connection, 0 = don't listen
BR=<baud rate>      // Baud rate value in bits per second
DB=<data bits>       // 7, 8
PR=<N|O|E>          // Parity: none, odd, even
ST=<stop bits>       // 1, 2
FL=<N|S|H>          // Flow control: none, software, or hardware

SP=<listening network port number>
SD=<disconnect time-out in seconds>
SO=<time-out in seconds before allowing new connection>

CM=<N|P|R>          // Outgoing connection mode: never (listen mode
                    // only), power-up, or upon received serial data

CI=<destination IP name/address>
CP=<destination port number>
CD=<disconnect time-out in seconds>
CR=<retry time-out in seconds after failed connection>

// Make an outgoing TCP connection when the specified GPIO input
// pin number signal is active. Allowable parameters:
// Active Low:  -3, -4, -7, -8, -9, -10, -12, -13, -14, -15
// Active High: +3, +4, +7, +8, +9, +10, +12, +13, +14, +15
// Example: SER1GC=+7  Make outgoing connection when pin 7 input
//                    goes high
GC=<GPIO pin number and polarity>

// GPIO pin reflects the status of a TCP connection,
// active = connected, inactive = not connected.
// Allowable parameters:
// Active Low:  -3, -4, -7, -8, -9, -10, -12, -13, -14, -15
// Active High: +3, +4, +7, +8, +9, +10, +12, +13, +14, +15
// Example: SER1GS=+3  Make pin 3 go high for active TCP connection
GS=<GPIO pin number and polarity>
```

## 8.4.1 Custom Packetization

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

Example: `AT#SER0PE=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 to wait since last character before sending a packet. A value of '0' waits forever.

`PC=<hex char>` Sends all accumulated serial data upon receipt of a 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 current setting. For example, "`AT#PE?<cr>`" will return a value of '0' or '1'.

## 8.5 GPIO Server and Analog-to-digital Read-back

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

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 is:

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, error if pin not GPIO or A/D
=<0|1>       // Set GPIO pin high or low, error if pin not GPIO
M=<mode>     // Modes: GPIO, RX, TX, RTS, CTS, and AD
M?          // Read the current mode of the pin

#PD?        // Read the system pin's current drive level:
           // 0 = 2 mA, 1 = 10 mA

#PD=<1|0>    // Set the current drive level for all pins:
           // 0 = 2 mA, 1 = 10 mA
```

## 8.6 AT Command Examples

The following examples display the full transcript of serial communication of commands sent and replies received to and from the SBL2e. Commands are marked in red, and 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 is made, the setting is saved. No system restart is required.

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

### 8.6.3 Configure Outgoing Network Client Connection

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

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

#### 8.6.4 Read Analog-to-Digital Input Channel 1

This example reads the AD1 pin (pin 8). The value read is immediately returned.

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

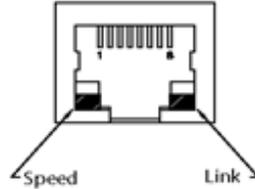
#### 8.6.5 Change Pin to GPIO and Set Output to High

This example checks the current mode of pin 8, which indicates that it is in the primary mode function, AD1. The mode is updated to GPIO, and then set to a high state.

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

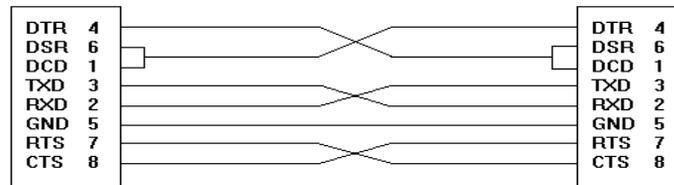
## 9. LEDs

- LED1 on RJ-45: Ethernet speed – 10 (off) or 100 (on)
- LED2 on RJ-45: Link



## 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 a network that does not have a DHCP server present, then you will need to specify a static IP and network mask address. IP addresses are required to route packets from one point to another on an Intranet or Internet. If you are on your own local area network (LAN), then there are some standard private address ranges that you can use:

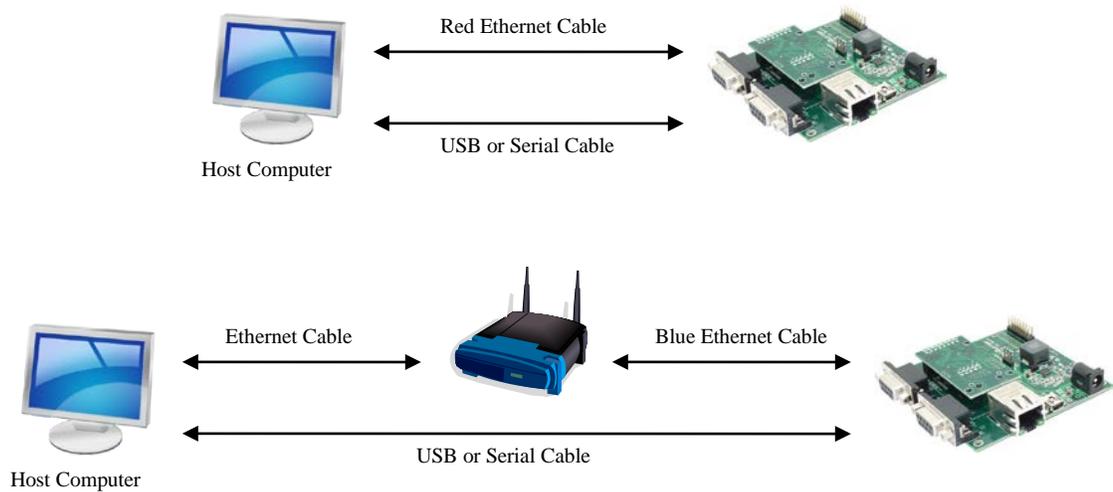
- Class A: 10.0.0.0 to 10.255.255.255
- Class B: 172.16.0.0 to 172.31.255.255
- Class C: 192.168.0.0 to 192.168.255.255

## 12. Web Browsers and Proxy Servers

If you are working on a corporate LAN that uses a proxy server for Internet web browsing, then you will need to exclude the IP address of the SBL2e in your web browser's proxy server settings or 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 application made available in the Windows operating system and a serial terminal program, such as the NetBurner MTTY terminal application 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 the need of a network hub or switch. The blue Ethernet cable is a standard straight-through Ethernet cable that should be used if you have a network hub or switch.

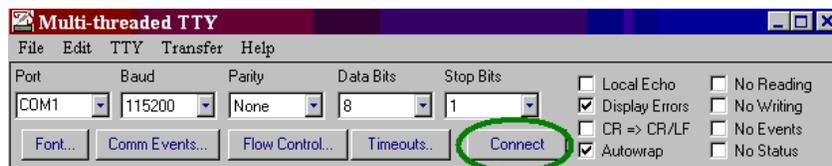


The objective of this example is to use a single host computer running Telnet and a serial terminal program to send data in either direction. When you type text in the Telnet window, it should appear in the serial terminal window and vice versa.

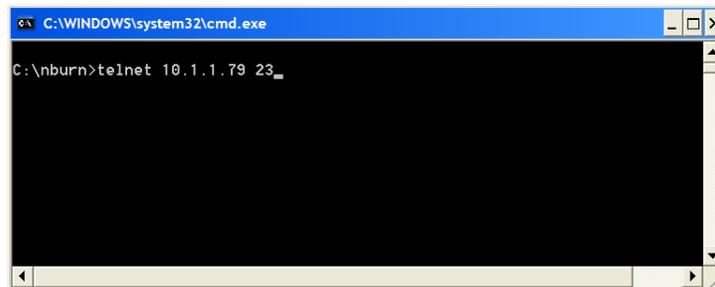
In the following example, an IP address of "10.1.1.79" is assigned to the SBL2e. Replace this IP address with the actual IP Address of your SBL2e.

## 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 then press the <Enter> key on your keyboard (remember to substitute your actual IP address).
4. Run either HyperTerminal or MTTY. Set the baud rate to the value you assigned to the SBL2e during configuration (default is 115,200 bps). To use MTTY:
  - Run “MTTTY.exe” – you should see the screen below. The port setting is the PC serial communication port of your host computer.
  - Click the MTTY “Connect” button.



5. In the command prompt window, run Telnet by typing: “telnet 10.1.1.79 23” after the prompt, and press the <Enter> key. The value “23” is the network port number of the SBL2e that is listening for incoming connections. This example uses the default value of 23 for SBL2e serial port 0, which is connected to the USB serial interface and to SBL2e serial port 0. If you have modified the listening network port number in the SBL2e network web page, 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      | N/A        | Internal draft.   |
| 1.1      | 2009.08.10 | Initial release.  |
| 1.2      | 2009.12.17 | Added AT command set.   |
| 1.3      | 2010.02.22 | Clarified description of packetization settings to indicate the settings only apply in the serial-to-Ethernet (not Ethernet-to-serial) direction.   |
| 1.4      | 2010.05.05 | Added machine commands.   |
| 1.5      | 2010.11.19 | Updated dual serial application web page screenshots; separated UDP-specific options from “Custom Packetization” section and created a new “UDP Mode Settings” section; added mention of RS-485 half/full duplex mode settings for serial web page; added mention of custom baud rates to serial options table; made other trivial updates to the document. |
| 1.6      | 2011.01.11 | Added new "GPIO TCP Effects" subsection in the "Web Page Configuration" section.  |
| 1.7      | 2011.03.18 | Added SERnGC and SERnGS AT commands.  |
| 1.8      | 2012.03.27 | Removed “LN” typo in AT command example usage for custom packetization in section 8.4.1; added information on usage of the new “MA” AT command in conjunction with the “#CUR” command to get the module’s MAC address in section 8.3.   |