LBNL Power Supply User's Manual

Revision 1.1

1 Revision History

Revision	Date	Comments	Author
1.0	October 28, 2006	First Document Draft	Michael Bankovitch
1.1	December 28, 2006	Second Document Draft	Michael Bankovitch

Trademarks: Windows XP, Windows NT, Windows 2000 are trademarks of Microsoft Inc. HyperTerminal is a trademark of Hilgraeve Inc. Net-Biter is a trademark of Intellicom Innovation AB.

2 Reference Documents

1. *Modbus Protocol Reference Guide*, PI-MBUS-300 Rev. J, June 1996, Modicon Inc., Industrial Automation Systems.
2. *BusyBox Command Descriptions, Rev. 1.00, 2004,* SSV Embedded Systems.
3. LBNL hardware documentation [TBD].

3 Introduction

The LBNL Power Supply has a powerful built-in controller, which allows you to monitor and control the power supply in several ways:

- Using a local terminal connection, via the maintenance console serial port
- Using a serial line connection and Modbus RTU protocol, via the protocol port
- Using a network connection and Modbus TCP protocol, via the Ethernet port
- Using a network connection and Telnet protocol, via the Ethernet port

This manual documents the LBNL Power Supply monitoring and control features. The manual does not contain information on the LBNL Power Supply hardware itself. For details on the hardware features, please refer to the LBNL Power Supply hardware documentation [Ref. 3].

The remaining part of this document is structured as follows:

- *Chapter 4: Connections* presents several methods that can be used to interface the LBNL Power Supply with the host computer.
- *Chapter 5: Modbus Protocol* describes the LBNL Power Supply Modbus registers, and the Modbus function codes that can be used to read and write these registers.
- *Chapter 6: Controller* presents the LBNL Power Supply embedded firmware features.
- *Chapter 7: Client* describes the LBNL Power Supply Linux-based client.
- *Chapter 8: Manual Pages* contains Linux-style manual pages for the various LBNL Power Supply programs.

4 Connections

This chapter presents in detail the LBNL Power Supply external interfaces, and the serial line and network connections that can be used to communicate with it.

The LBNL Power Supply has the following interfaces:

- Maintenance console
- Serial line Modbus RTU interface
- Modbus TCP network interface

The following pages present each of these interfaces in more detail.

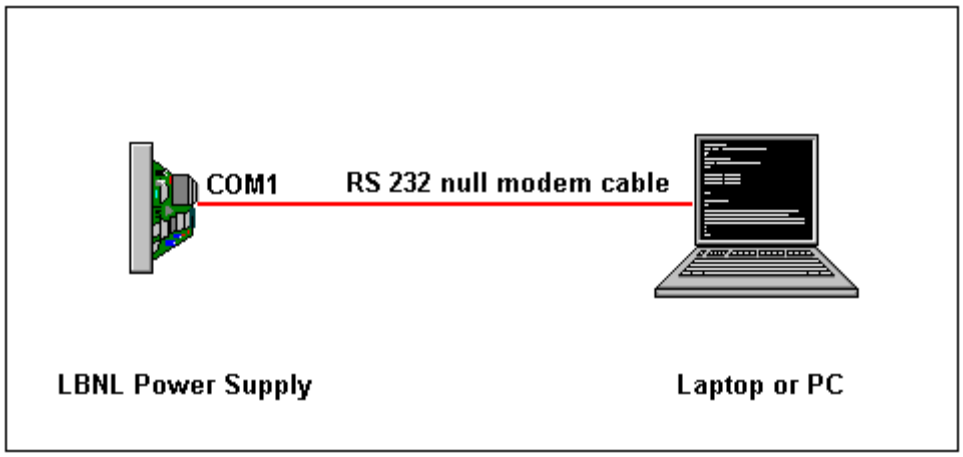
4.1 Maintenance Console

The maintenance console interface provides direct access over a serial line to the LBNL Power Supply controller. Using the maintenance console interface you can:

- Access the LBNL Power Supply controller operating system
- Run a number of low-level diagnostic commands, not available through the Modbus interfaces
- Run the LBNL Power Supply user interface locally, on the controller itself.

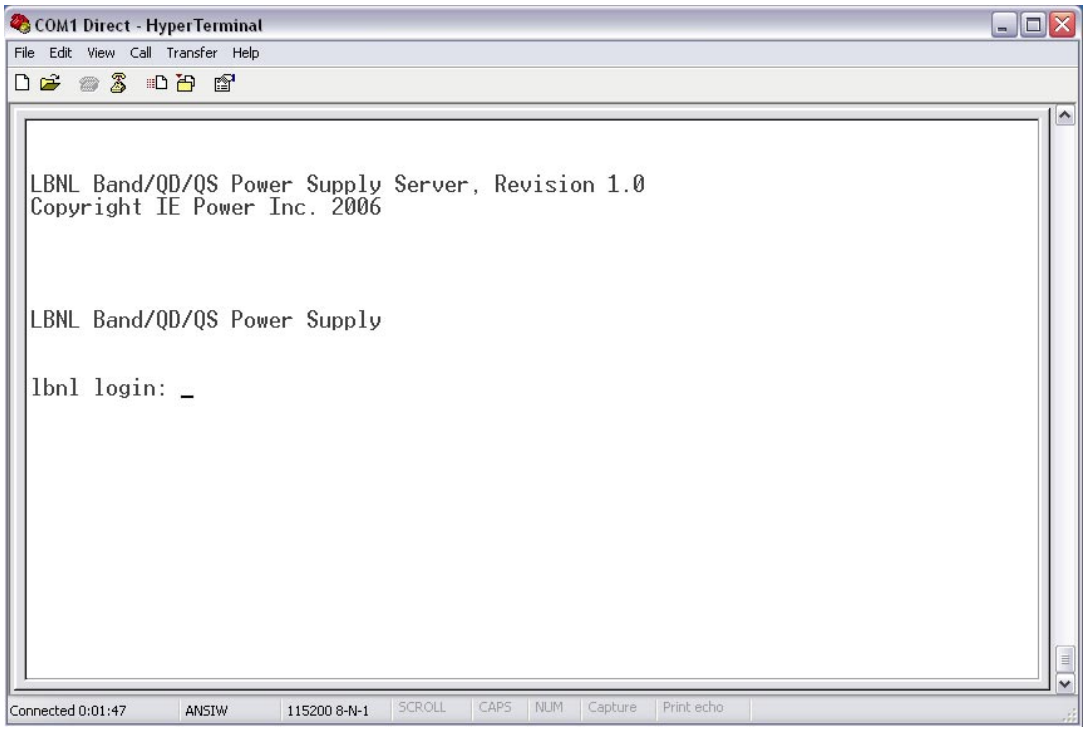
The maintenance console is provided on the LBNL Power Supply serial port COM1. To use this interface, you will need a standard RS232 null modem cable, a PC or a laptop, and some terminal emulation software, such as HyperTerminal.

Connect your PC or a laptop to the COM1 port on the power supply as shown in the following picture:



Set the serial port on your PC or laptop to: 115200 Baud, no parity, 8 bits per character and 1 stop bit

Start your terminal emulation software on the PC or laptop and then power up the power supply. As the LBNL Power Supply controller boots, you will see several pages of diagnostic messages. Once the controller gets ready, you will see the following screen:



At this point, you can login into one of the power supply predefined accounts and use the command set available through the maintenance console. Details on the command set are given later in this manual, in the chapter *Controller*.

4.2 Modbus Interface

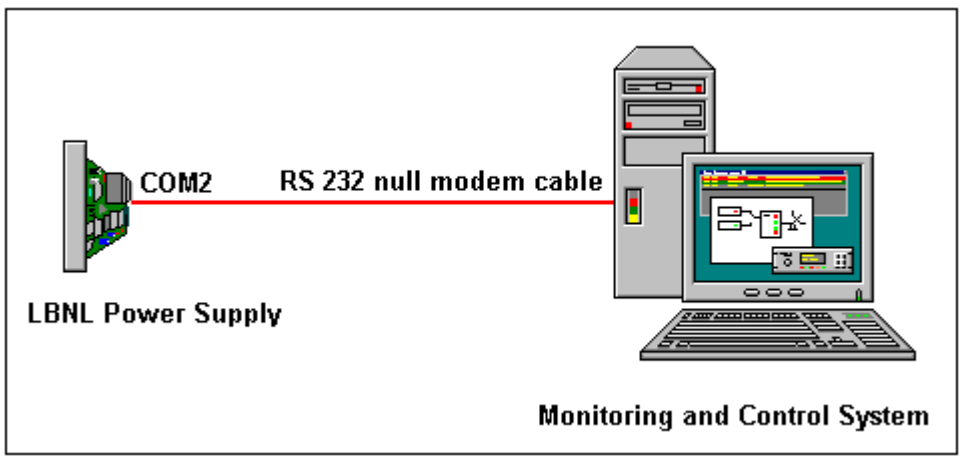
Modbus protocol is a de-facto standard communication protocol, widely used for industry automation, and for process monitoring and controls [Ref. 1]. The LBNL Power Supply supports two versions of the protocol:

- *Modbus RTU* protocol, which is used over serial lines
- Modbus TCP protocol, which allows connections over a TCP/IP local or wide area networks

Both versions of the protocol support the same set of Modbus registers and function codes[1]. Therefore, you can run the same set of Modbus queries using either Modbus RTU or Modbus TCP.

4.2.1 Modbus RTU Interface

If you want to monitor and control your LBNL Power Supply using a serial line, connect COM2 serial port on the power supply to your computer using a RS232 null modem cable, as shown in the following picture.



1. For Modbus terminology and details on the communication protocol itself, please refer to [Ref. 1].

Please note that the COM2 port on the power supply provides only data receive and transmit lines, and signal ground. The port *does not support RS232 control lines*. Therefore, you may need a special cable to loop back RTS/CTS and DTR signals.

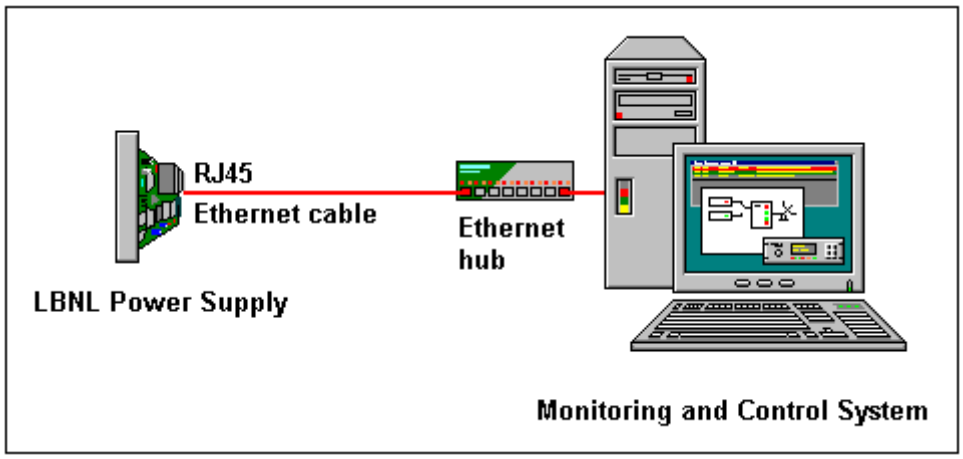
By default, COM2 serial port on the power supply is set to 115200 Baud, no parity, 8 bits per character and one stop bit. You can change these settings by editing the controller configuration file, as described later in this manual, in the chapter *Controller*.

The default Modbus address for the LBNL power supply is 128 (80 hex). You can change this address by editing the controller configuration file.

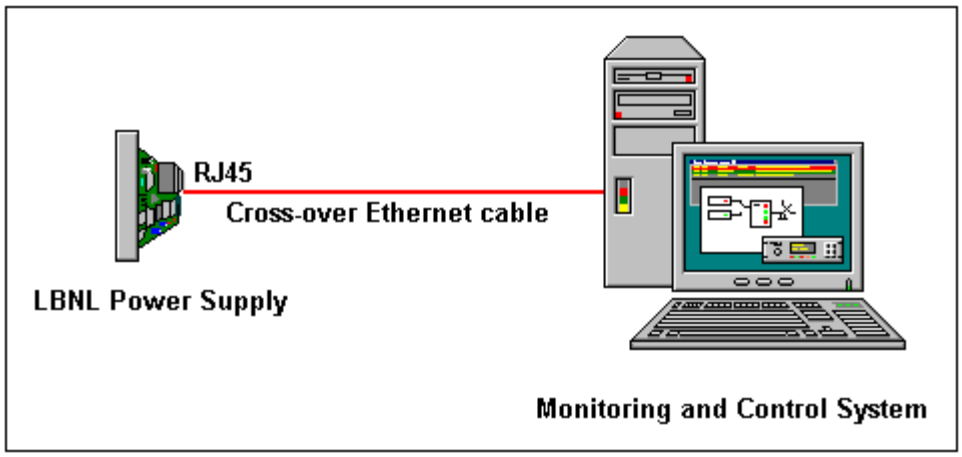
Once you connect your power supply to the computer, you can start using Modbus queries. The LBNL Power Supply Modbus registers and supported function codes are presented in the chapter *Modbus*.

4.2.2 Modbus TCP Interface

If you want to monitor and control your LBNL Power Supply using a TCP ethernet connection, connect the Ethernet port on the power supply to your Ethernet hub, using a standard Ethernet cable, as shown in the following picture.



To connect the power supply directly to the computer, use a cross-over Ethernet cable, as shown below.



To monitor and control your power supply using Modbus TCP, you will have to set two parameters:

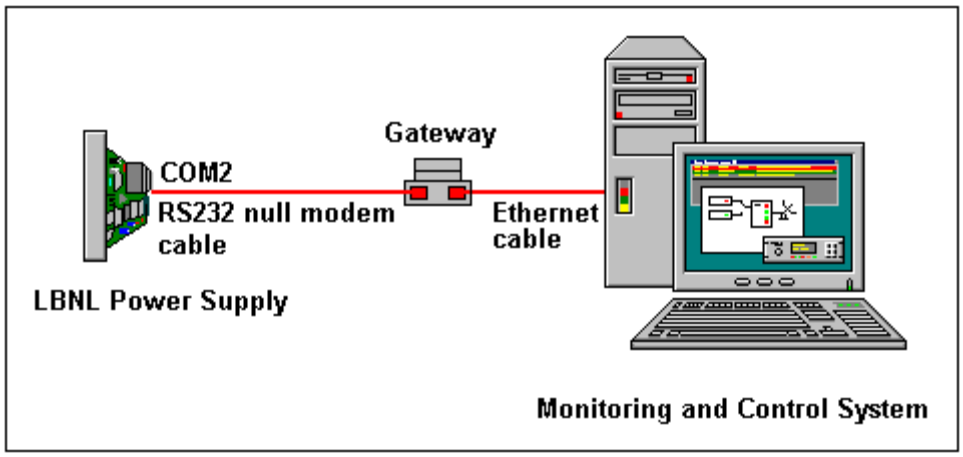
- Set the LBNL Power Supply IP address, using the *ipaddree* command from the LBNL Power Supply maintenance console. Please refer to the chapter *Controller* for details.
- Edit the controller configuration file to select Modbus address. Please refer to the chapter *Controller* for details.

Once you connect your power supply to your TCP/IP network, you can start using Modbus queries. The LBNL Power Supply Modbus registers and supported function codes are presented in the chapter *Modbus*.

4.2.3 Modbus TCP to Modbus RTU Conversion

The LBNL Power Supply controller supports both Modbus RTU and Modbus TCP protocols. To allow you to monitor and control your power supply, we provide with the power supply a Linux-based computer program, referred to as the *client*. You can install and run the LBNL Power Supply client on a PC that runs under Linux operating system.

The LBNL Power Supply client software supports only the Modbus TCP version of the protocol. If you want to use it for the Modbus RTU, you will need a Modbus RTU/TCP gateway, such as NetBiter manufactured by IntelliCom Innovation AB. A typical power supply connection via a Modbus RTU/TCP gateway is depicted in the following picture.



In this scenario, the Modbus RTU/TCP gateway converts the client software queries from the Modbus TCP packets into the Modbus RTU packets and sends them to the power supply. When the power supply replies, the gateway converts the Modbus RTU packets into the Modbus TCP packets and sends them to the client software.

You can always use exactly the same set of Modbus registers and queries, regardless of whether you use Modbus RTU, Modbus TCP, or protocol conversion via a Modbus RTU/TCP gateway.

5 Modbus

To monitor and control the LBNL Power Supply, you can use two versions of the Modbus protocol: Modbus RTU and Modbus TCP. In both cases, the supported Modbus function codes and register set are exactly the same.

The LBNL Power Supply supports the following Modbus function codes:

- *Read Multiple Registers*: function code 3
- *Write Single Register*: function code 6
- *Write Multiple Registers*: function code 16

The list of the LBNL Power Supply Modbus registers is given in the following table[1].

Table 5-1: LBNL Power Supply Modbus Registers

Function Code	Register	Description	Argument/Response
3	4:00001	CSSB and CARG	Bitmap register, used as shown in Table 2
3	4:00002	CAFB and CDFB	Bitmap register, used as shown in Table 3
3, 6, 16	4:00004	Trace Control	ON (0x0001) or OFF (0x0000)
3, 6, 16	4:00005	Table Download	Selects destination table for download, valid range 1 to 10
3, 6, 16	4:00006	Table Activation	Activates / loads table, valid range 1 to 10
3	4:00300	Watchdog	Unsigned 16 bit value, auto-incremented every second
3, 6, 16	4:01901	Power Control	ON (0x0001) or OFF (0x0000)
3, 6, 16	4:01910	Keep Alive	Unsigned 16 bit value

1. This is a preliminary list of registers. New registers may be added as the implementation proceeds.

The *Table Download* register is used to select the destination table to download. The controller supports 10 tables, identified by integer values 1 to 10. Before downloading table contents, you have to select destination table to which you wish to download data.

The *Table Activation* register allows you to select the table you wish to load into the power supply hardware. Setting this register to a value from 1 to 10 selects the table to activate and initiates the process of loading the selected table into the power supply hardware. At the same time, controller configuration file is updated accordingly. This allows the controller to load the same table on any subsequent power up.

The bitmap flags in the LBNL Power Supply Modbus CSSB and CARG registers are defined as follows.

Table 5-2: CSSB and CARG register definition

Bit	Set (1)	Reset (0)	Description
0	Standby	Not ready	Power supply fault state
1	ON	OFF	Power supply on (DC output enabled) or OFF
2	Remote	Local	Remote / local status
3	OK	Fault	Communication
4	Fault	OK	Over current relay tripped
5	Fault	OK	Logic power failure
6	Fault	OK	PS water flow low
7	Fault	OK	Diode bridge over temperature
8	Fault	OK	Output over current
9	Fault	OK	Output over voltage
10	Fault	OK	Ground fault over current
11	Fault	OK	Current regulator fault
12			Not used
13			Not used
14			Not used
15			Not used

The bitmap flags in the LBNL Power Supply Modbus CAFB and CDFB registers are defined as follows.

Table 5-3: CAFB and CDFB register definition

Bit	Set (1)	Reset (0)	Description
0	Fault	OK	AC under voltage
1	Fault	OK	AC over voltage
2	Fault	OK	AC over current
3	Fault	OK	DC under/over voltage
4	Fault	OK	IGBT module fault
5			Not used
6			Not used
7			Not used
8	Fault	OK	Capacitor bank fuse blown
9	Fault	OK	Magnet over temperature or flow low
10	Fault	OK	Main water flow or external fault 1
11	Fault	OK	External fault 2 or 3
12	Fault	OK	Choke / xfmr / cabinet over temperature
13	Fault	OK	Input fuses / smoke
14	Fault	OK	Safety
15	Fault	OK	Main contactor

6 Controller

The LBNL Power Supply is equipped with a powerful controller, responsible for all monitoring and control functions. From the user point of view, the controller provides the following external interfaces:

- Maintenance console
- Network interface, which supports Modbus TCP, Telnet and FTP protocols
- Serial line interface, used for the Modbus RTU protocol

This chapter provides details on each of these interfaces and associated commands and features.

6.1 Maintenance Console

The *Maintenance Console* interface is provided on the controller's serial port COM1. This interface is intended for the maintenance and diagnostic purposes only, and in day-to-day operation, you will not need access to the maintenance console port. Instead, you will access the power supply either using the Modbus protocol, or via Telnet session.

The controller uses the COM1 serial port as it's operating system console. To use maintenance console, you can connect to COM1 either a VT100-compatible dumb terminal, or a PC or laptop, equipped with some terminal emulation software. However, the controller will boot normally even if no terminal is attached to COM1. In fact, during the regular power supply operation there will be no terminal attached to COM1.

While booting, the controller sends to the maintenance console the usual diagnostic messages. When the boot sequence completes, the controller displays the following banner:

```
LBNL Band/QD/QS Power Supply Server, Revision 1.0

Copyright IE Power Inc. 2006
```

This message indicates that the controller is ready to service the Modbus queries. Following this message, the controller displays the login prompt:

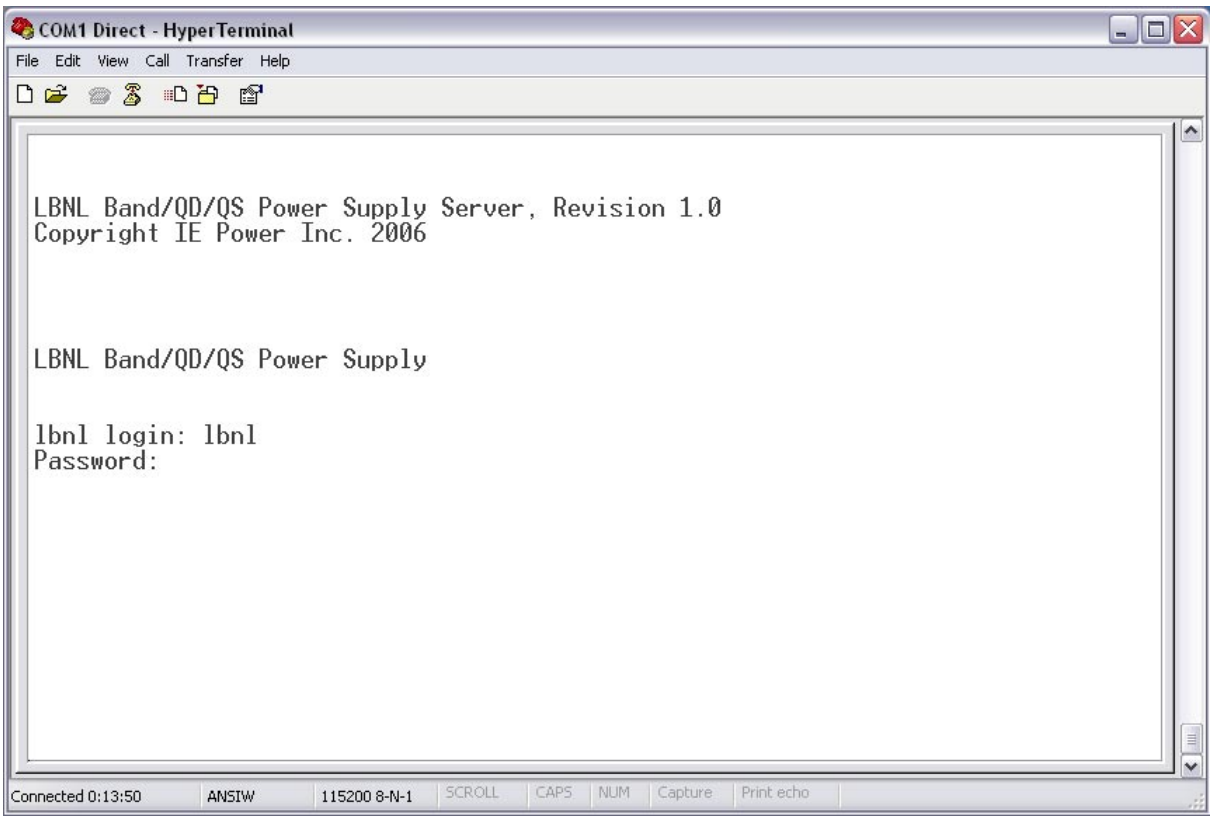
```
LBNL Band/QD/QS Power Supply
lbnl login:
```

At this point, the controller is ready to accept user logins. As shipped, the power supply comes with two predefined login accounts:

- The super-user account *root*
- The local user interface account *lbnl,* with default password *lbnl*

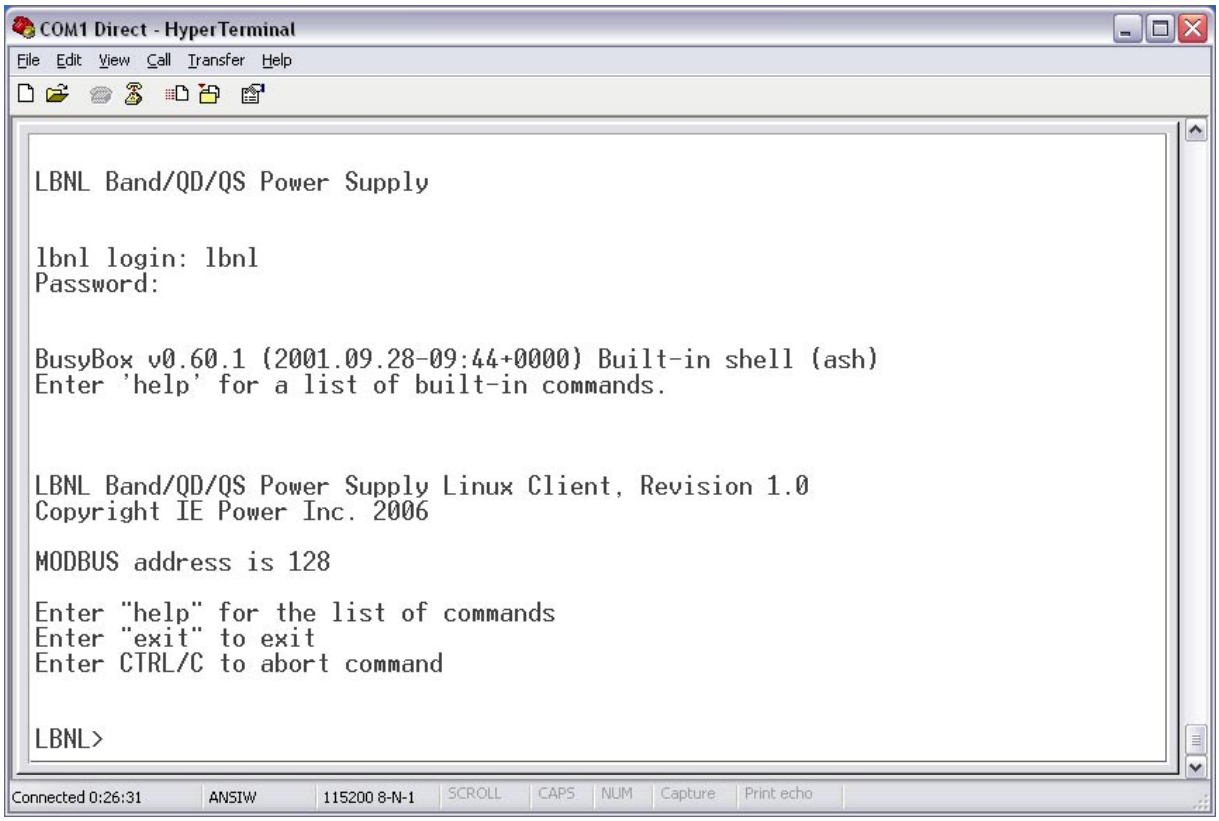
Please contact your supplier to obtain the default login password for the *root* account.

To login, enter the account name, *root* or *lbnl*. The controller will then prompt you to enter password. Once a correct password is submitted, the controller will log you into the requested account.



Normally, you will use the root account *only to configure the controller and to upgrade the controller firmware*. We strongly discourage using the root account for any other purpose. Modifying controller configuration and upgrading the firmware is described later in this chapter.

When you login into the *lbnl* account, the controller will automatically start the LBNL Power Supply user interface program:



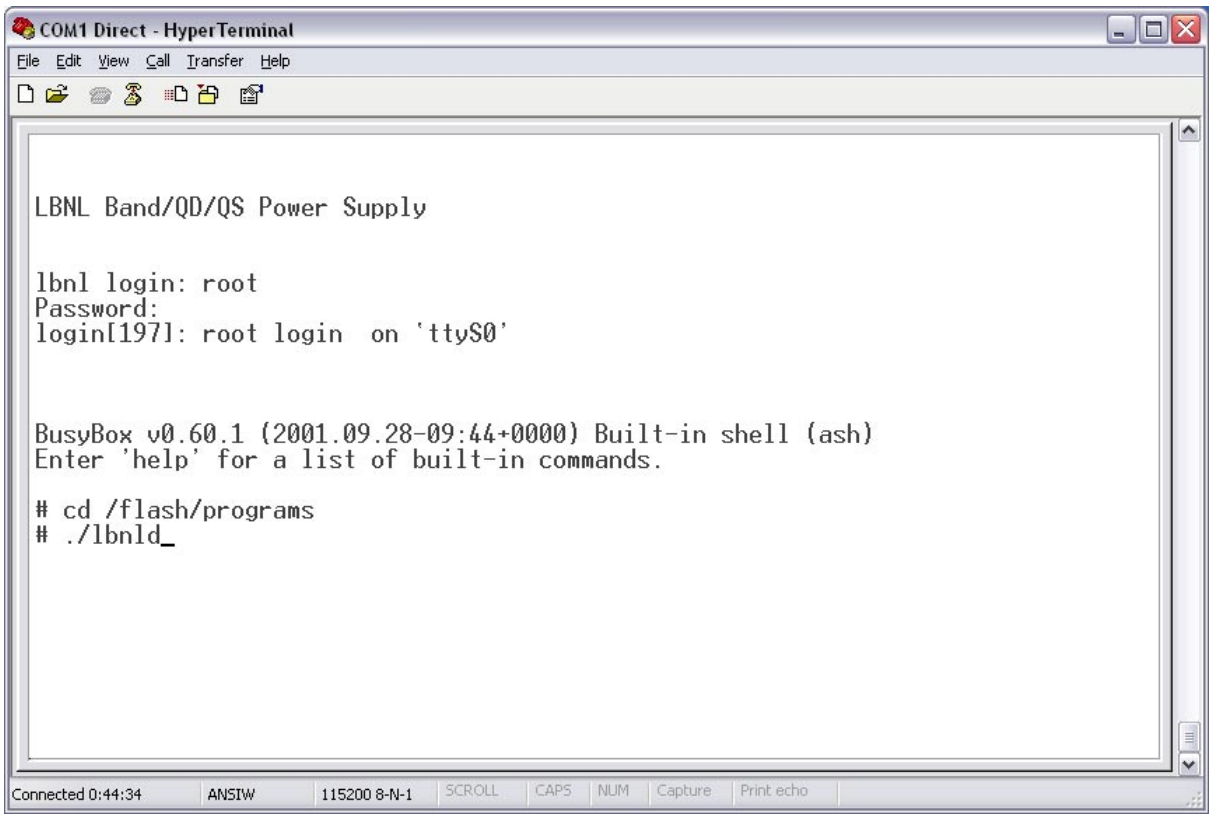
The available commands and their syntax are the same as described in the chapter *Client*, later in this document.

6.2 Startup

The LBNL Power Supply controller boots automatically when you power up the power supply. Initially, the controller performs a number of internal tests and then starts the operating system and the LBNL Power Supply application-specific programs. This process may take between 10 to 15 seconds. After that, the controller is ready to service your Modbus queries and login requests.

Normally, you will never start the controller software manually. A manual startup might be needed only to run some very specific diagnostics, which is usually done by the IE Power Inc. personnel.

To start the controller software manually, you will have to login as *root*, change directory to */flash/programs*, and execute program *lbnld*:



You can start program lbnld in foreground, as shown in the picture above, or in background, by adding '&' at the end of the command line:

```
lbnld &
```

Program *lbnld* supports several *command line options*, which you can use to:

- Enable trace[1] on the maintenance console
- Select a configuration file other than default
- Select the power supply Modbus address

1. The trace feature is described later in this chapter.

- Select the TCP/IP protocol port to be used for table downloads

For example, starting the lbnld program using:

```
lbnld -d
```

enables the trace on the mainenance console.

Details on the *lbnld* program command line options are provided in the chapter *Manual Pages*.

6.3 Shutdown

Normally, you will never shutdown the controller software manually. You will simply power down the entire power supply. A manual shutdown will be required when updating the controller firmware, or before running some application-specific diagnostics.

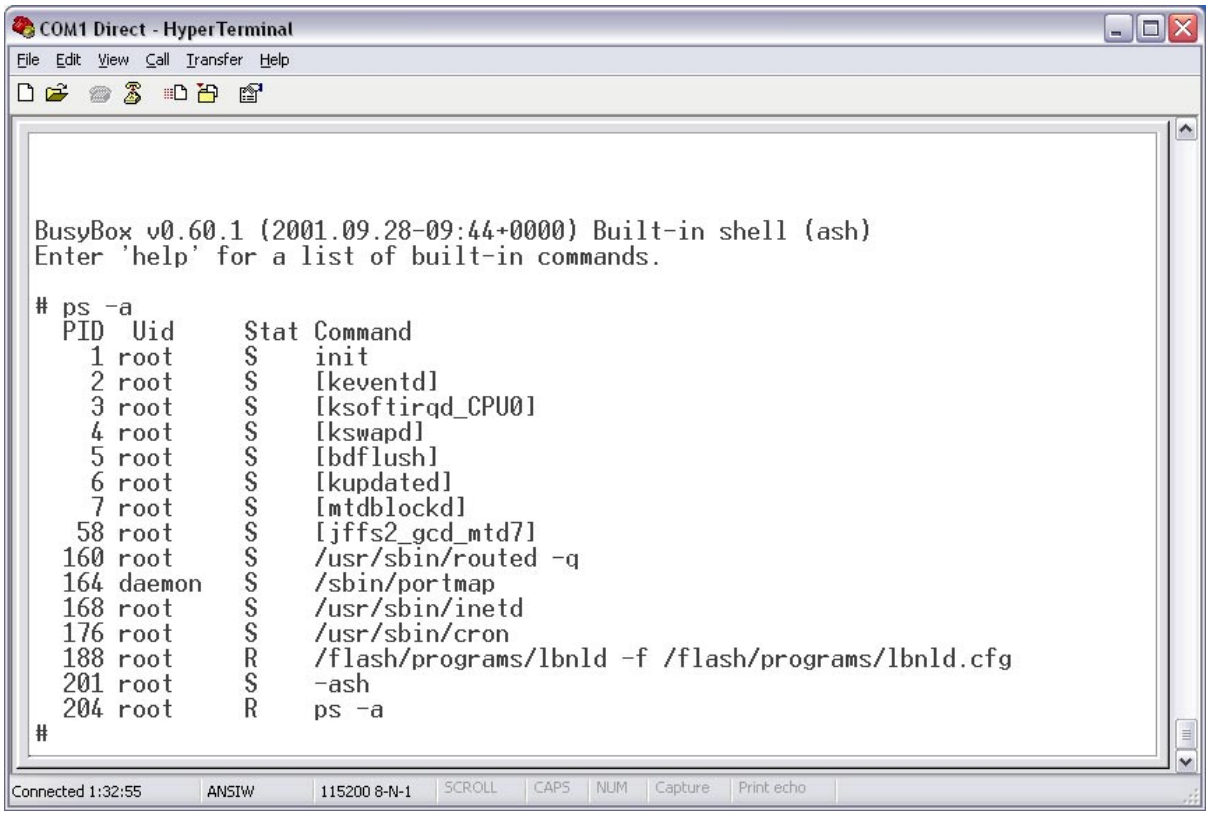
To shutdown the controller software manually, you login as *root*. Then, use the *ps* command to display a list of all currently running programs:

```
ps -a
```

Look for the *lbnld* program in the list, and note the number in the column marked *PID*. In the following sceen snapshot, this number is 188. Then use the *kill* command, with the PID number as an argument, to stop the *lbnld* program:

```
kill 188
```

This will stop the LBNL Power Supply application-specific software. A typical list of programs that run on the controller is depicted in the following picture. If you use the *ps* command again after you stop the *lbnld* program, you will notice that *lbnld* is no longer on the list.



Once you stop the *lbnld* program, you can start it again manually, as described on the previous pages, or simply by reseting the power supply.

6.4 Configuration File

To start and to function properly, the LBNL Power Supply controller needs a number of configuration parameters. For example, the controller needs to initialize the Modbus serial port to certain data rate, parity, character size and number of stop bits. It also needs to know which Modbus address to use.

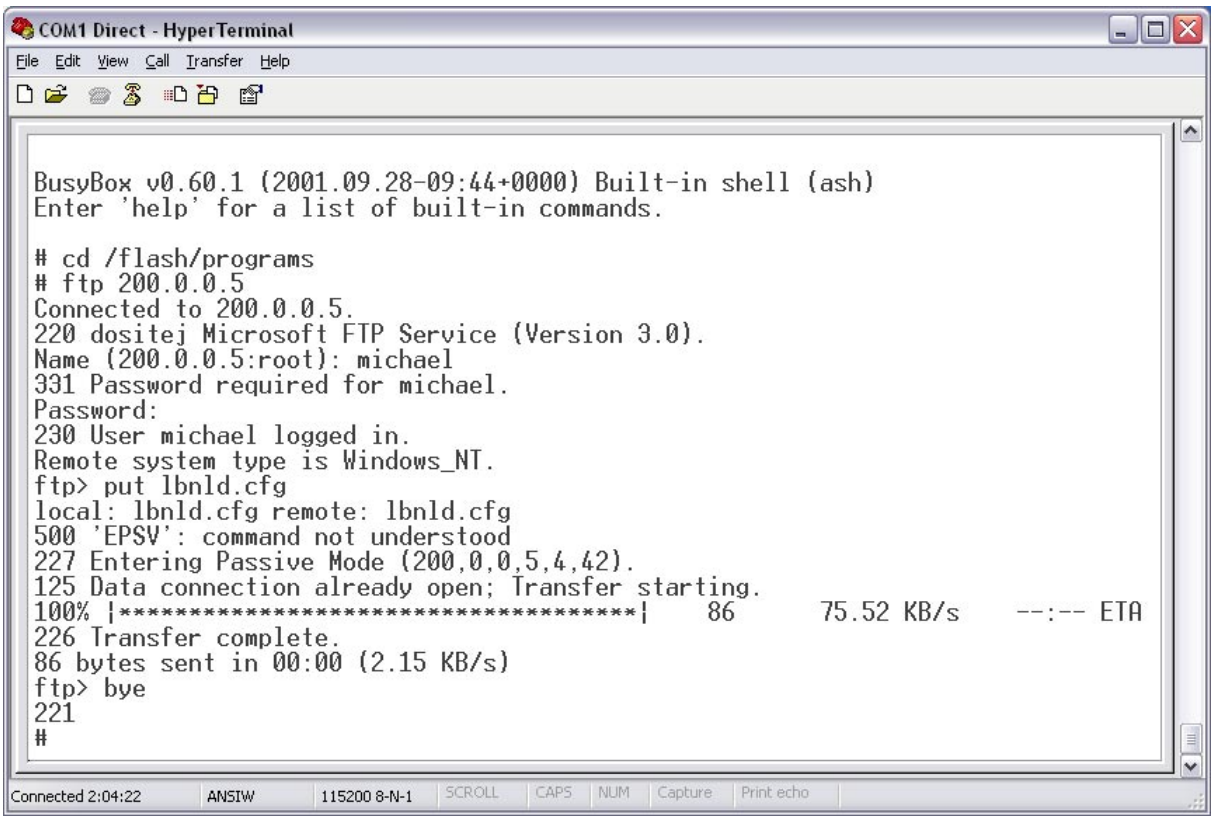
The controller reads its configuration parameters from a special configuration file. By default, this file is */flash/programs/lbnld.cfg*. You can overwrite this default using the *lbnld* program command line options, as shown in the chapter *Manual Pages*.

To change the controller startup parameters, you will have to modify the configuration file. The file itself is a plain text file, therefore you can simply edit its contents using any text editor. The controller itself does not have a built-in text editor. Therefore, you

will have to download the file to your PC, make changes and then put the configuration file back to the controller.

6.4.1 Downloading Configuration File

To download the configuration file from the controller to your PC, you can use the FTP protocol from the controller itself. Login into the controller as *root* and change directory to */flash/programs*. Let us assume that you want to download the file to the computer with IP address 200.0.0.5, using account *michael*. Your typical FTP sesssion will be similar to that captured in the following screen snapshot:



To connect to your PC, type:

```
ftp nnn.nnn.nnn.nnn
```

where *nnn.nnn.nnn.nnn* represents your PC IP address. When prompted, enter your account name, such as *michael* in this example. The FTP will then prompt you to enter the password. Once you login, you can use *cd* command to place the configuration file into an appropriate directory. When transfering text files such as *lbnld.cfg*, you will have to use FTP *ASCII* mode. Finally, use *put* command to transfer your file, and *bye* command to exit FTP:

```
put lbnld.cfg
bye
```

At this point, the configuration file is ready for editing on your PC.

If you don't have an FTP server installed on your PC, you can use the FTP server built into the controller, and an FTP client on the PC. This method is described later in this chapter.

6.4.2 Editing Configuration File

Once you transfer the configuration file onto your PC, you can use any text editor to view or change its contents. If you have Windows operating system on your PC, you can use *Notepad* editor. On a Linux-based PC, you can use *vi*, *gedit* or *emacs* editor.

A typical contents of the controller configuration file will be similar to the following:

```
BAUD 115200
CSIZE 8
PARITY NONE
STOP 1
SERIAL /dev/ttyS1
DEBUG OFF
ADDRESS 128
TABLE 9
DOWNLOAD 4000
```

Each line starts with a *keyword*, followed by an *argument*. The keyword and argument are separated by a single *SPACE* character. Configuration file entries may appear in any order.

Here is the list of all possible configuration file entries:

ADDRESS configuration file entry specifies the Modbus protocol unit address. This parameter is common for both Modbus RTU and Modbus TCP protocols. Valid address range is from 1 to 247. The controller will not service any broadcast Modbus queries.

BAUD configuration file entry specifies the data rate to use for the Modbus RTU interface. Supported data rates are all valid rates from 1200 to 115200 Baud.

CSIZE configuration file entry specifies bits per character to be used for the Modbus RTU interface. This should be always 8.

DEBUG configuration file entry enables trace on the maintenance console upon startup. Valid entries are: ON or OFF.

DOWNLOAD configuration file entry specifies the TCP protocol port to use for the table downloads. If omitted, the controller uses the default value of 4000.

PARITY configuration file entry specifies which parity to use for the Modbus RTU interface. Valid entries are: ODD, EVEN and NONE.

SERIAL configuration file entry specifies which serial port to use for the Modbus RTU interface. For the LBNL Power Supply this should be always */dev/ttyS1*, which represents COM2.

STOP configuration file entry specifies the number of stop bits per to be used for the Modbus RTU interface. This should be always 1.

TABLE configuration file entry specifies which table to load on startup. Valid range is from 1 to 10.

Once you edit changes into the configuration file, you need to upload the file to the controller, and to restart the controller or reset the power supply.

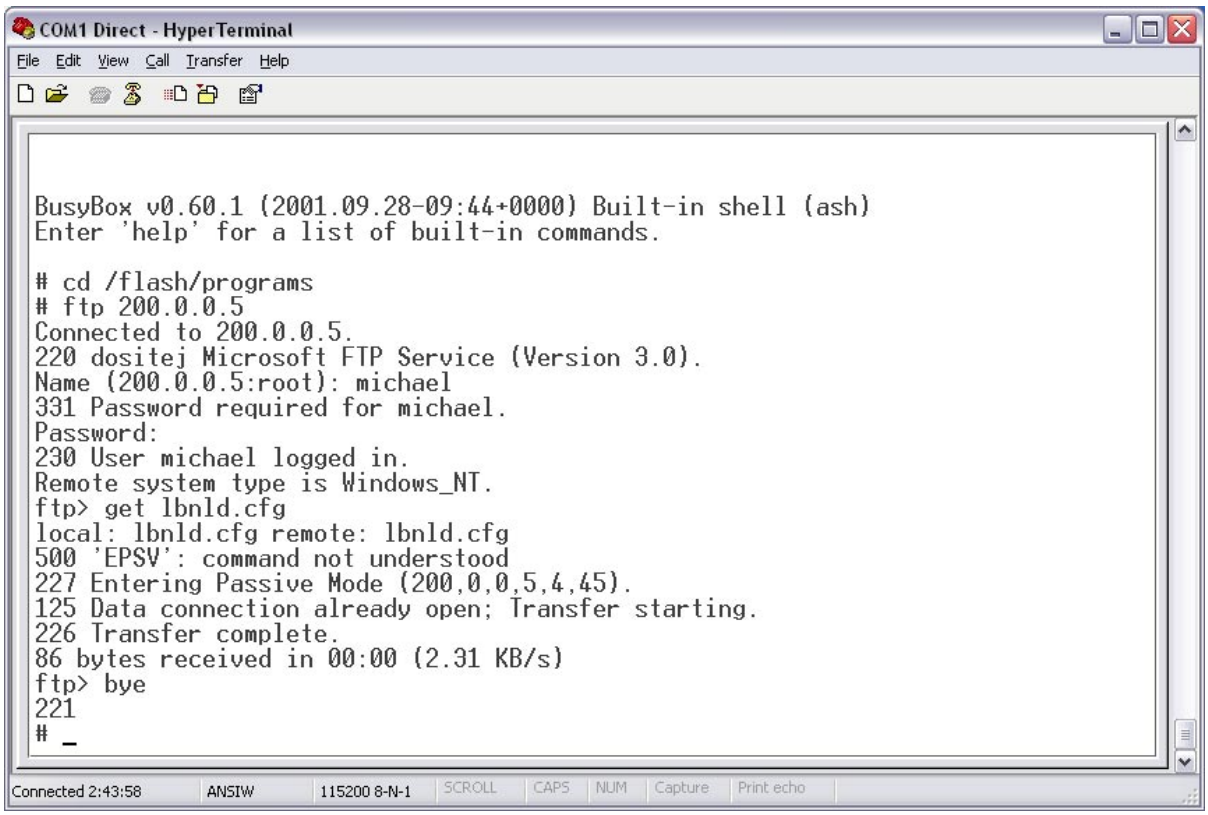
6.4.3 Uploading Configuration File

To upload the configuration file from your PC to the controller, you can use the FTP protocol from the controller itself. Login into the controller as *root* and change directory to */flash/programs*. Let us assume that you want to upload the file from the computer with IP address 200.0.0.5, using account *michael*. Your typical FTP sesssion will be similar to that captured in the following screen snapshot.

To connect to your PC, type:

```
ftp nnn.nnn.nnn.nnn
```

where *nnn.nnn.nnn.nnn* represents your PC IP address. When prompted, enter your account name, such as *michael* in this example. The FTP will then prompt you to enter the password. Once you login, you can use *cd* command to navigate to the directory where you saved the modified configuration file. When transfering text files such as *lbnld.cfg*, you will have to use FTP *ASCII* mode. Finally, use *get* command to transfer your file, and *bye* command to exit FTP:




```
get lbnld.cfg
bye
```

At this point, the modified configuration file is transferred to the controller. To verify its contents, you can use *more* command:

```
more lbnld.cfg
```

The controller will display the present *lbnld.cfg* file contents.

6.4.4 Using FTP Server

In the previous sections, we showed you how to use the LBNL Power Supply controller FTP client to download and upload configuration files. Using the FTP client implies that you have an FTP server running on your PC or laptop. But how to dowload and upload configuration files if you don't have the FTP server on your PC?

The LBNL Power Supply controller itself comes with a built-in FTP server. Therefore, an alternative to the procedures described on the previous pages will be to login into the controller FTP server from your PC or laptop. This means that you will use an FTP client on your PC or laptop, and login into the controller as an FTP server.

There is a wide variety of FTP clients for both Windows and Linux operating systems. You can use an FTP client of your choice. Here we show only how to use FTP client as a console application, which is readily available on both Windows and Linux computers.

From your PC or laptop, you can invoke the FTP client using the power supply IP address:

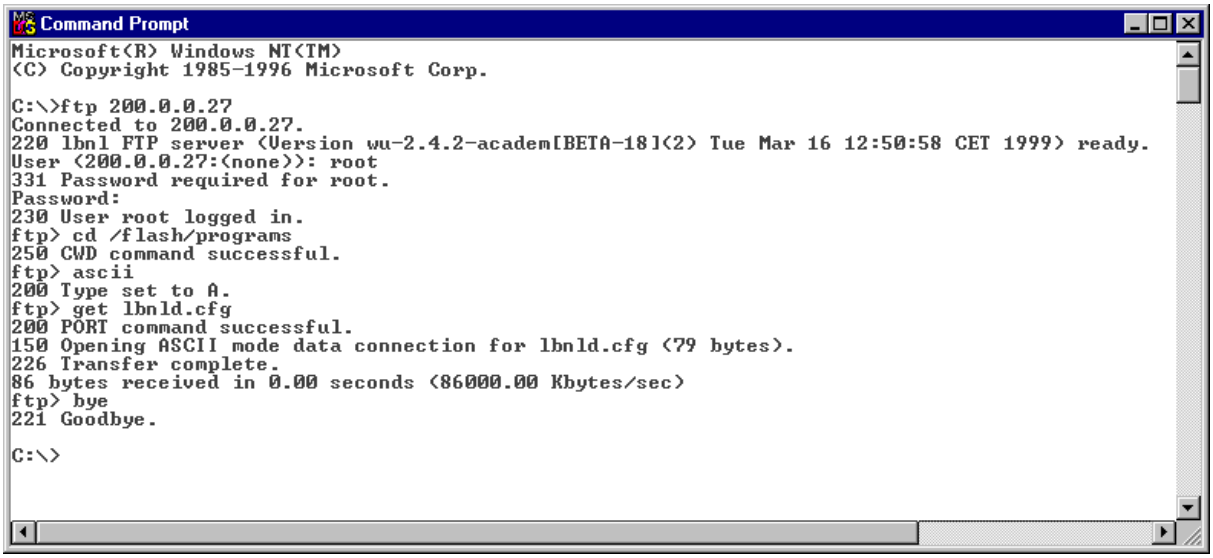
```
ftp nnn.nnn.nnn.nnn
```

or using the power supply name, as in:

```
ftp slot1
```

To connect to the power supply by name, your PC or laptop must be able to resolve the name either using the domain name server, or the *etc/hosts* file on the computer itself.

In either case, the FTP server will prompt you to enter the login account name and password. Login using the *root* account and change directory to */flash/programs*. Make sure that you use ASCII mode for the file transfer. Then, use *get* command to download the file to your computer, *put* command to upload the file to the controller, and *bye* command to close the FTP session. A typical FTP session is depicted in the following screen snapshot:



This example illustrates how to download the configuration file. To upload it, use *put* instead of *get* command.

6.5 Local User Interface

You can run the LBNL Power Supply user interface on the controller itself, by connecting your laptop to the maintenance console serial port COM1, and then logging in into the *lbnl* account.

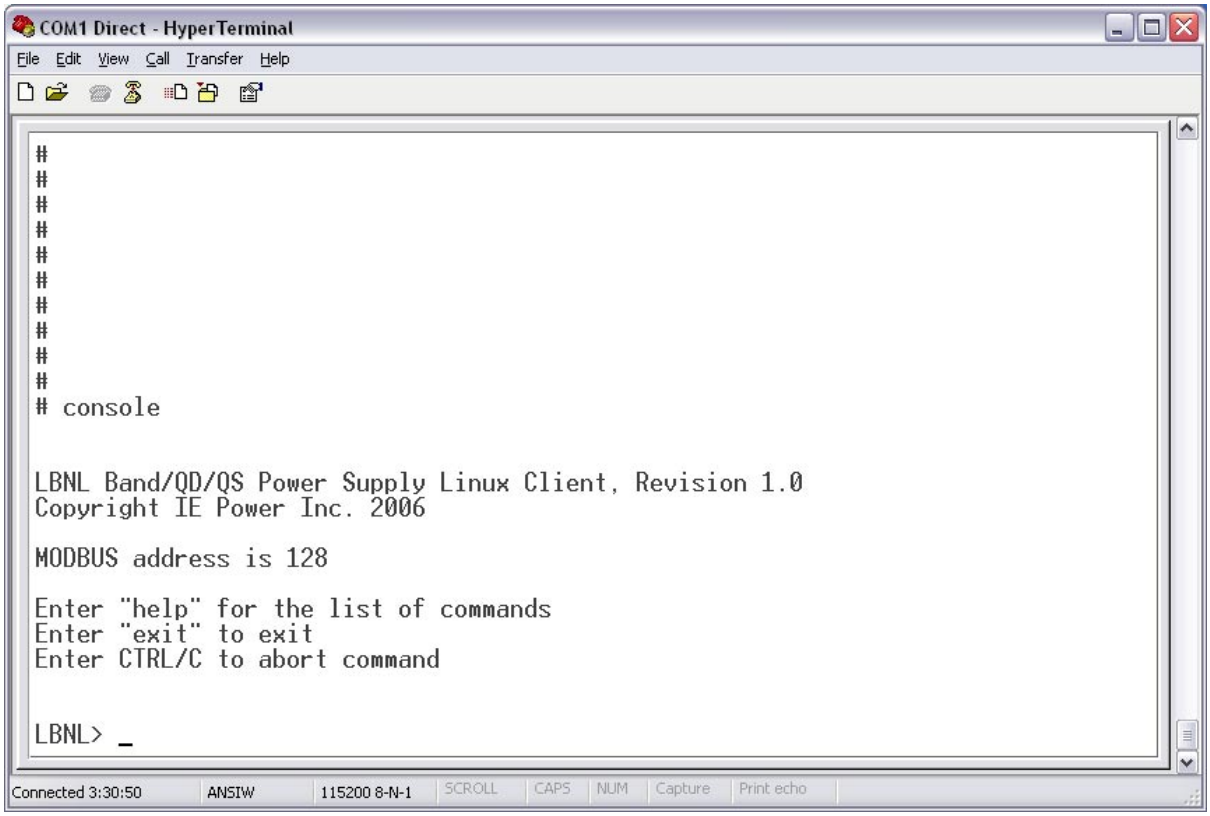
When you login as *lbnl*, the controller will automatically start user interface on the maintenance console. When you exit user interface, the controller will automatically log you out. This feature ensures that users without *root* priviledges cannot access the rest of the controller software. The *root* account should be accessible only to the system administrators.

If you login as *root*, you can explicitly start the user interface program by changing directory to */flash/programs* and then running the *console* script:

```
cd /flash/programs

console
```

Whether you login as *lbnl*, or start console program from *root* account, the controller will display messages similar to those in the screen snapshot below.



The LBNL Power Supply local user interface is the same as the user interface program that we provide for the Linux-based PCs. The Linux-based user interfaceis presented in the chapter *Client*. Please refer to that chapter for details on the command set.

6.6 Telnet Interface

If you have access to a Linux-based PC, you will simply install the Linux-based client software that we provide with the power supply. You will then use the client software to monitor and control the power supply via Modbus TCP protocol. However, if you have a Windows-based PC or laptop, you can still connect to the LBNL Power Supply over a TCP/IP network using a Telnet session.

If you have TCP/IP connectivity to the power supply, you can connect to it using the following command on your PC or laptop:

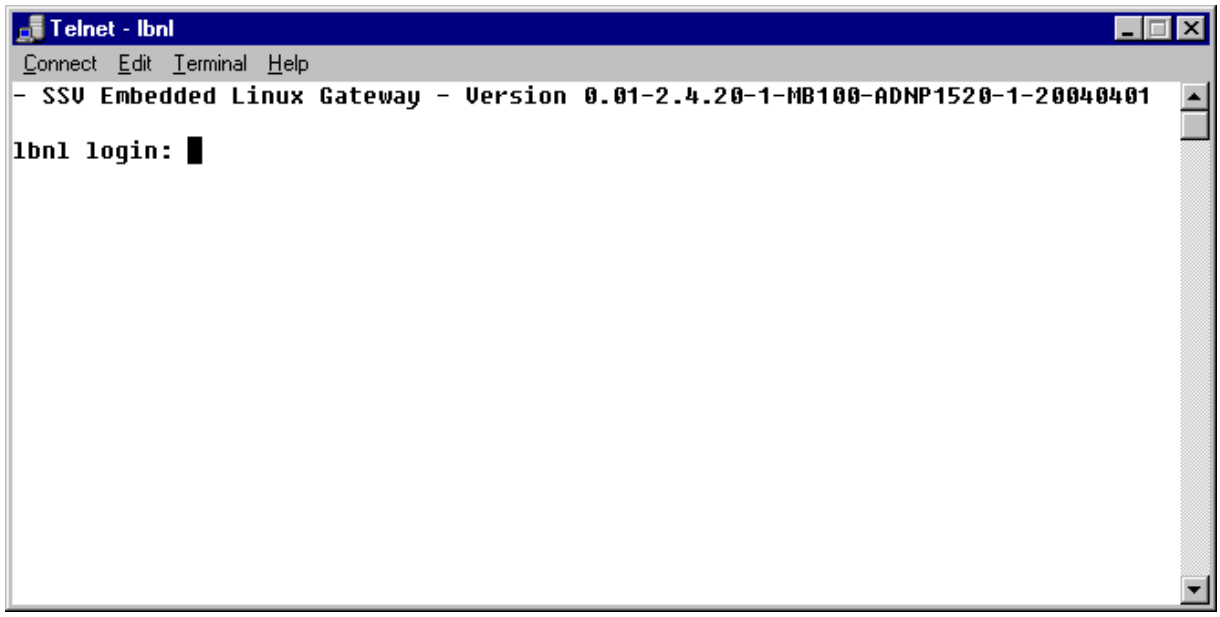
```
telnet nnn.nnn.nnn.nnn
```

where *nnn.nnn.nnn.nnn* represents the power supply IP address.

You can also assign a name to the power supply. This can be done using a domain name server on your IP network, or by editing the *etc/hosts* file on your computer or laptop. For example, if you add an entry for the power supply named *slot1* to the *etc/hosts* file, you can connect to the power supply by typing:

```
telnet slot1
```

When you start a telnet session with the power supply, the controller will display the telnet login screen similar to the following one:



At this point, you can login into the controller using either *lbnl* or *root* account.

Note that you may need to setup the telnet program on your computer to properly handle *DELETE* and *BACKSPACE* keys.

6.7 Modbus

The LBNL Power Supply controller provides two Modbus protocol interfaces:

- Modbus RTU protocol interface on serial port COM2
- Modbus TCP protocol interface on Ethernet port

The controller supports the same Modbus function codes and register set on both interfaces. You can use either of these interfaces to monitor and control the power supply, and you can use both of these interfaces simulateneously.

When using the Modbus TCP protocol, the controller supports up to 16 simultaneous TCP connections. It is up to the client application to open and close a TCP connection for each Modbus query, or to keep the TCP connection open during the entire session with a given power supply.

When you use the Modbus TCP protocol, the LBNL Power Supply becomes a multi-user system. This means that more than one Modbus TCP client can connect to the power supply at the same time. The LBNL Power Supply controller executes Modbus TCP queries sequentially, as *transactions*. If it receives multiple queries from different clients, the controller will queue them in order of arrival and execute them one by one. The controller will *not* service the next query before the previous query is processed.

In a multi-user environment, *it is your responsibility to ensure that operators will not send conflicting commands to the same power supply*. This applies to any Modbus TCP device, and not just to LBNL Power Supply.

With the LBNL Power Supply we provide a simple text-based Modbus TCP client. This client software is documented in the next chapter. You can also build your own client software, using details on function codes and registers provided in the chapter *Modbus*, earlier in this manual.

6.8 Table Download

For its operation, the LBNL power supply requires a lookup table stored into its dual-access RAM memory. The controller can store up to 10 tables into its flash memory, and then select one of these tables and load it into the power supply RAM.

Loading a table into the power supply RAM is a two step process. First, you have to download the table into the controller flash memory. Next, you have to activate it.

The controller stores tables in a dedicated directory named */flash/tables*. The tables are named *table01* to *table10*.

Before downloading table data, you have to select the target table. This is done by setting a Modbus register associated with the table download to a value from 1 to 10. For example, if you set download register to 3, the controller will save binary data that you download into the table */flash/tables/table03*.

To receive binary data, the controller uses a dedicated TCP/IP connection over its Ethernet port. The overall sequence of downloading a table is as follows:

- Establish a TCP connection to the controller TCP protocol port 4000[2]
- Set the controller download Modbus register to a value from 1 to 10
- Download binary data
- Disconnect from the controller TCP protocol port 4000

You can download tables using the LBNL power supply client software *select* and *download* commands. These commands are described later in this manual.

6.9 Table Activation

As described above, you can save up to 10 tables into the controller flash memory. After that, at any point in time you can select any of these tables and load it into the power supply RAM memory.

To activate a given table, simply set the controller table activation Modbus register to a value from 1 to 10. The controller will validate the selected table and load it into the power supply RAM. At the same time, the controller will update its configuration file. On any subsequent power up or reset, the controller will get the current table number from the configuration file and load it.

6.10Diagnostics

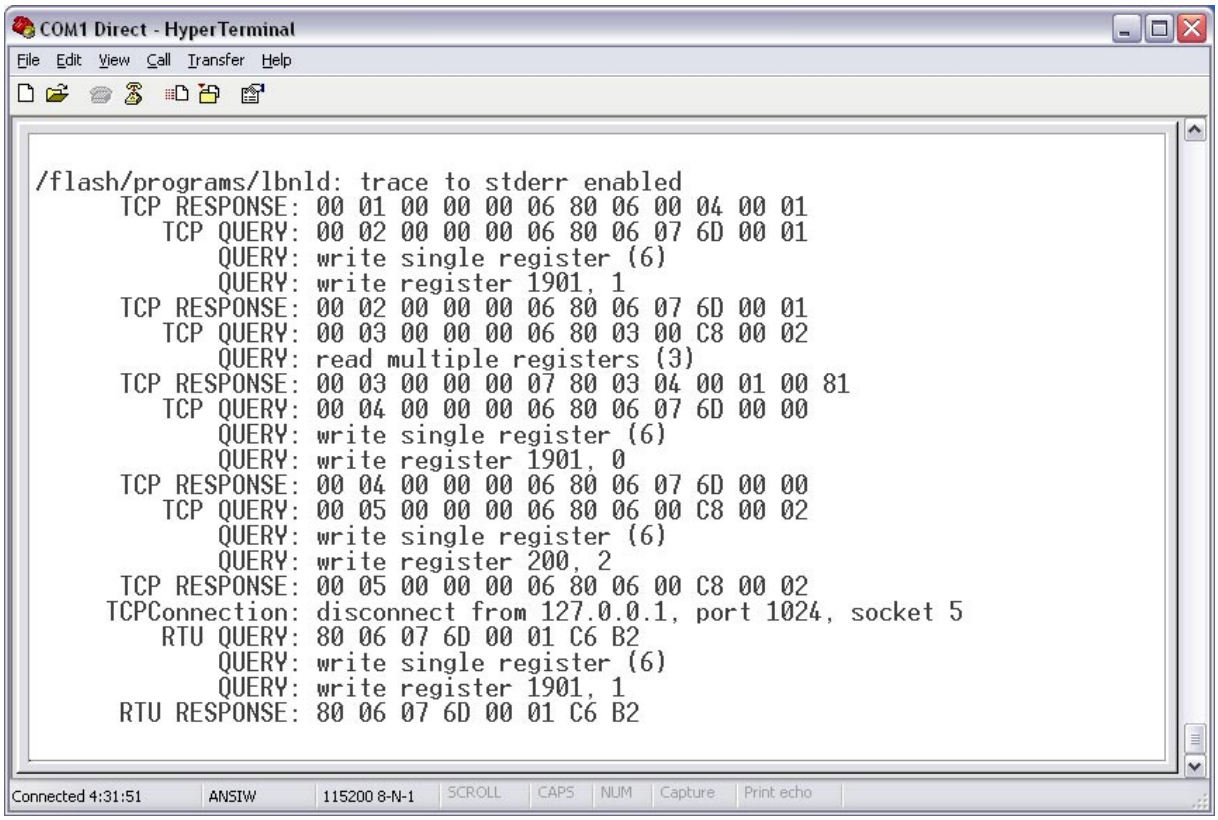
The LBNL Power Supply controller comes with a set of diagnostic utilities and features. There are two types of diagnostics:

2. You can change the default TCP protocol port number using the controller command line options, or by modifying the controller configuration file.

- Built-in *trace*, which dumps useful information to the controller maintenance console
- A set of low level command line utilities, used to read and write the controller registers and memory locations

6.10.1 Trace

The *trace* feature is very useful when integrating and testing the LBNL Power Supply controller against a third party Modbus client software. When enabled, the trace displays on the controller maintenance console received and sent Modbus packets, actions taken to service Modbus queries, and so on. A typical trace output might look like the following screen snapshot:



In the example above you can see both Modbus TCP and Modbus RTU packets being serviced by the controller. You can see how the controller parses the Modbus queries and processes them, and how it manages TCP connections.

To look at the trace output, you will need to connect a PC or a laptop to the controller maintenance console serial port COM1. Note that the controller operating system processes new line characters only if you login. Therefore, to have the trace output formated like in the example above, you would have to login as root.

If you use a terminal emulation program such as HyperTerminal, you can also capture trace into a file, for further static analysis. Simply use the HyperTerminal *Capture Text* feature.

There are several ways to start the controller trace:

- You can start the controller software with the trace option on. To do so, edit the configuration file and include statement *DEBUG ON* into it.
- Start the controller software with debug command line option as in *lbnld -d*
- Enable the trace remotely from the client, or locally, using the local user interface. In either case, use the *TRACE ON* command to enable trace and *TRACE OFF* to disable it.

6.10.2 Command Line Utilities

The LBNL Power Supply controller comes with a number of low level commands that can be used from the command line prompt to examine controller's input/output registers and memory locations, and to change their contents. These commands are intended for use by qualified personnel only, as changing registers and memory locations can seriously affect controller operation.

In this chapter we only briefly describe the available commands. Details on how to use them are given in the chapter *Manual Pages*. The available diagnostic commands are:

inb	command allow you to read the content of a 8 bit register.
outb	command allow you to write the content of a 8 bit register.
inw	command allow you to read the content of a 16 bit register.
outw	command allow you to write the content of a 16 bit register.
rdmem	command allow you to read the content of a memory location.
wrmem	command allow you to write the content of a memory.

All of these utilities are located in the */flash/programs* directory. To use them, you must login using the *root* account.

6.11Firmware Upgrades

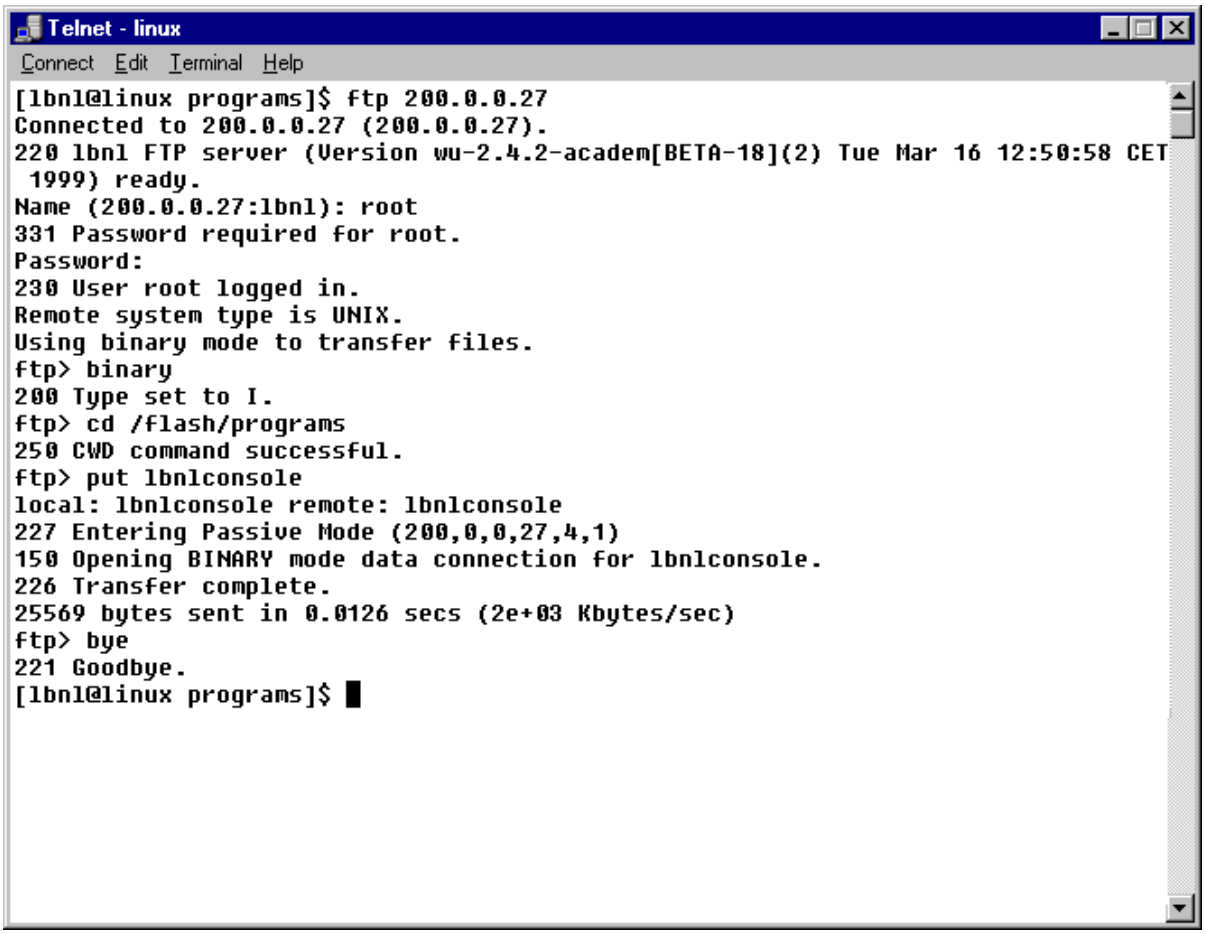
From time to time you may need to install firmware upgrades on the LBNL Power Supply controller. In fact, upgrading the controller firmware means replacing one or both of the following programs:

```
/flash/programs/lbnld
/flash/programs/lbnlconsole
```

To replace any of these programs, you need to follow these steps:

- List all programs presently running on the controller, as described in section *Shutdown*, earlier in this chapter.
- Stop the program that you need to replace, as described in the same section.
- From your PC or laptop, login into the controller FTP server, as described in section *Using FTP Server.*
- By typing FTP command *binary*, make sure that you use binary mode for file transfer.
- Using FTP command *cd*, change remote directory to */flash/programs*.
- Transfer the file using the FTP command *put*.
- Close the FTP session using the FTP command *bye*.
- Reset the power supply.

In the following screen snapshot we illustrate how to replace the user interface program *lbnlconsole*:


Telnet - linux
Connect Edit Terminal Help
[lbnl@linux programs]$ ftp 200.0.0.27
Connected to 200.0.0.27 (200.0.0.27).
220 lbnl FTP server (Version wu-2.4.2-academ[BETA-18](2) Tue Mar 16 12:50:58 CET
1999) ready.
Name (200.0.0.27:lbnl): root
331 Password required for root.
Password:
230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> binary
200 Type set to I.
ftp> cd /flash/programs
250 CWD command successful.
ftp> put lbnlconsole
local: lbnlconsole remote: lbnlconsole
227 Entering Passive Mode (200,0,0,27,4,1)
150 Opening BINARY mode data connection for lbnlconsole.
226 Transfer complete.
25569 bytes sent in 0.0126 secs (2e+03 Kbytes/sec)
ftp> bye
221 Goodbye.
[lbnl@linux programs]$

7 Client

The LBNL Power Supply comes with an additional computer program referred to as the *Modbus TCP Client*, or simply the *client*. This program provides the user interface to all Modbus function codes and registers supported by the LBNL Power Supply controller. In addition, the program also supports an application-specific command used to download tables to the power supply.

The name of the file that contains the client program is *lbnlconsole*. You can execute the *lbnlconsole* program on any Linux-based PC. To install it, simply copy the program into an appropriate directory on your computer, and make sure that it has proper permissions. Once you save the file onto your PC, you can adjust its permissions using the following command:

```
chmod 755 lbnlconsole
```

The same *lbnlconsole* program is also pre-loaded onto the LBNL Power Supply controller itself. Therefore, you can run the client program locally, on the controller, or remotely, on a Linux-based PC. In both cases, the command set is exactly the same.

The client supports one TCP connection at a time. Therefore, you need to connect to a given power supply, issue a number of Modbus queries, disconnect, connect to the next power supply, and so on.

To connect to any given power supply, you have to set its IP address, and its Modbus address.

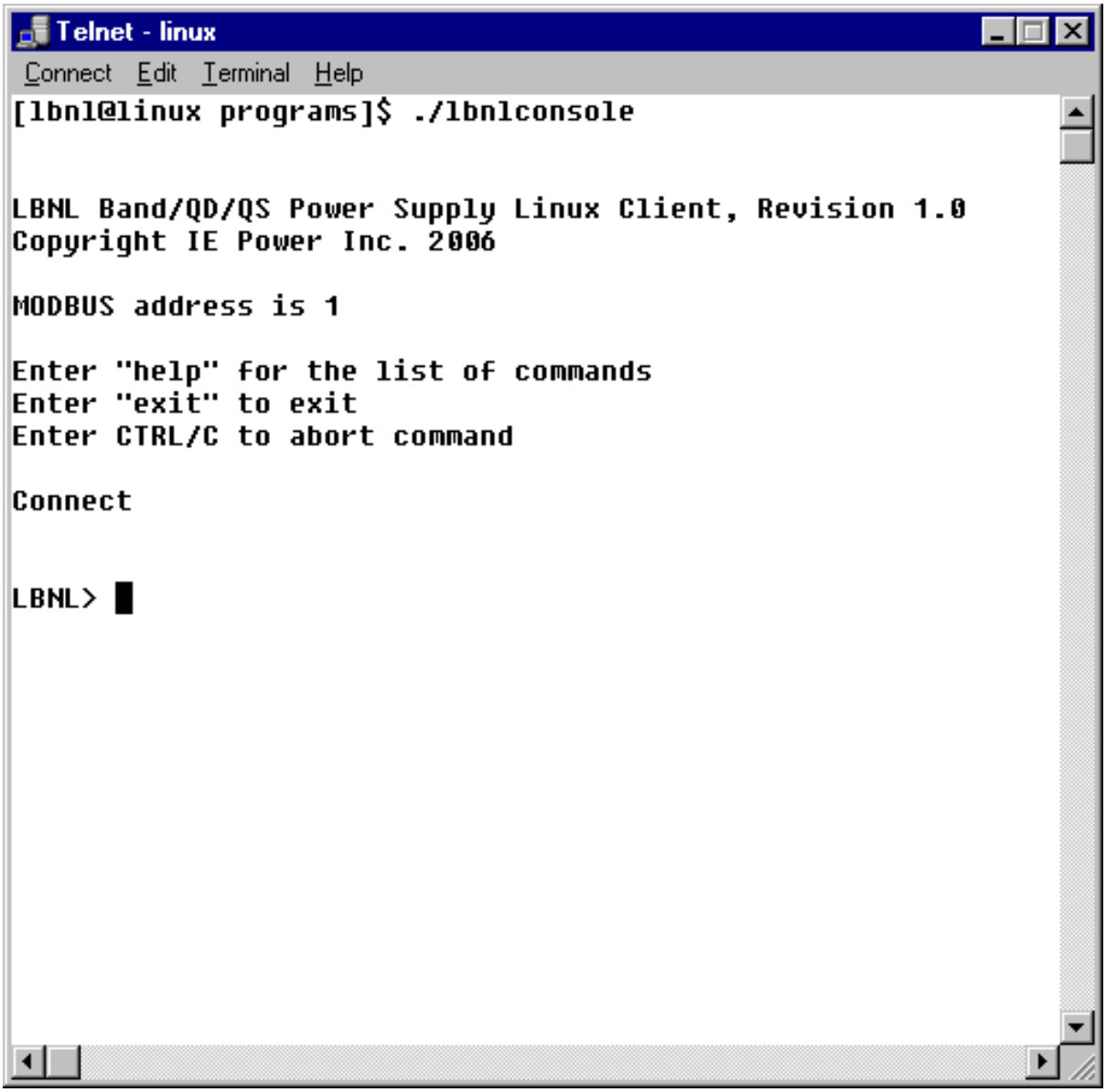
You can also connect to a given power supply indirectly, using a Modbus gateway, such as NetBiter. In such case, you will select the gateway IP address and the power supply Modbus address. When the client sends a Modbus TCP packet to the gateway, the gateway converts it into a Modbus RTU packet. The power supply sends its response to the gateway as Modbus RTU packet, and the gateway converts it into the Modbus TCP packet and sends it to the client.

The set of Modbus commands that you can use through the gateway is exactly the same as when you use a direct TCP connection to the power supply itself. However, you can download tables only through a direct TCP connection to the power supply itself.

The simplest way to start the client is to type:

```
lbnlconsole
```

The program will display its banner, followed by a few hints on how to use it:



When you start the program, the Modbus address is set to 1. You will have to change it, using the *address* command. To display a list of valid commands, enter *help*. To exit the program, use *exit* or *quit*. If you make any mistakes while entering parameters, you can abort the command by typing CTRL/C.

At this point, the program is not connected to any power supply. Hence, the last line in the above example is a reminder that you need to *connect* to the desired power supply.

The *LBNL* prompt indicates that the program is ready for your next command.

You can enter commands in either capital or lower case letters, but you cannot mix capital or lower case letters.

7.1 Command Line Options

The client supports several *command line options*. The formal calling syntax for the *lbnlconsole* program is provided in the chapter *Manual Pages*. Here we briefly illustrate what you can do using these options.

When you start the client, you can specify which power supply you want to connect to:

```
lbnlconsole -h 200.0.0.27
```

or:

```
lbnlconsole -h slot1
```

When you invoke it as shown above, the client will immediately connect to the specified power supply.

By default, the Modbus TCP protocol uses protocol port 502. If, for any reason, you wish to use any other protocol port, you can overwrite this default using:

```
lbnlconsole -p nnn
```

where *nnn* stands for the protocol port you wish to use. When you invoke it as shown above, the client will use protocol port *nnn* instead of 502.

Finally, you can specify which Modbus address to use, by starting the client as follows:

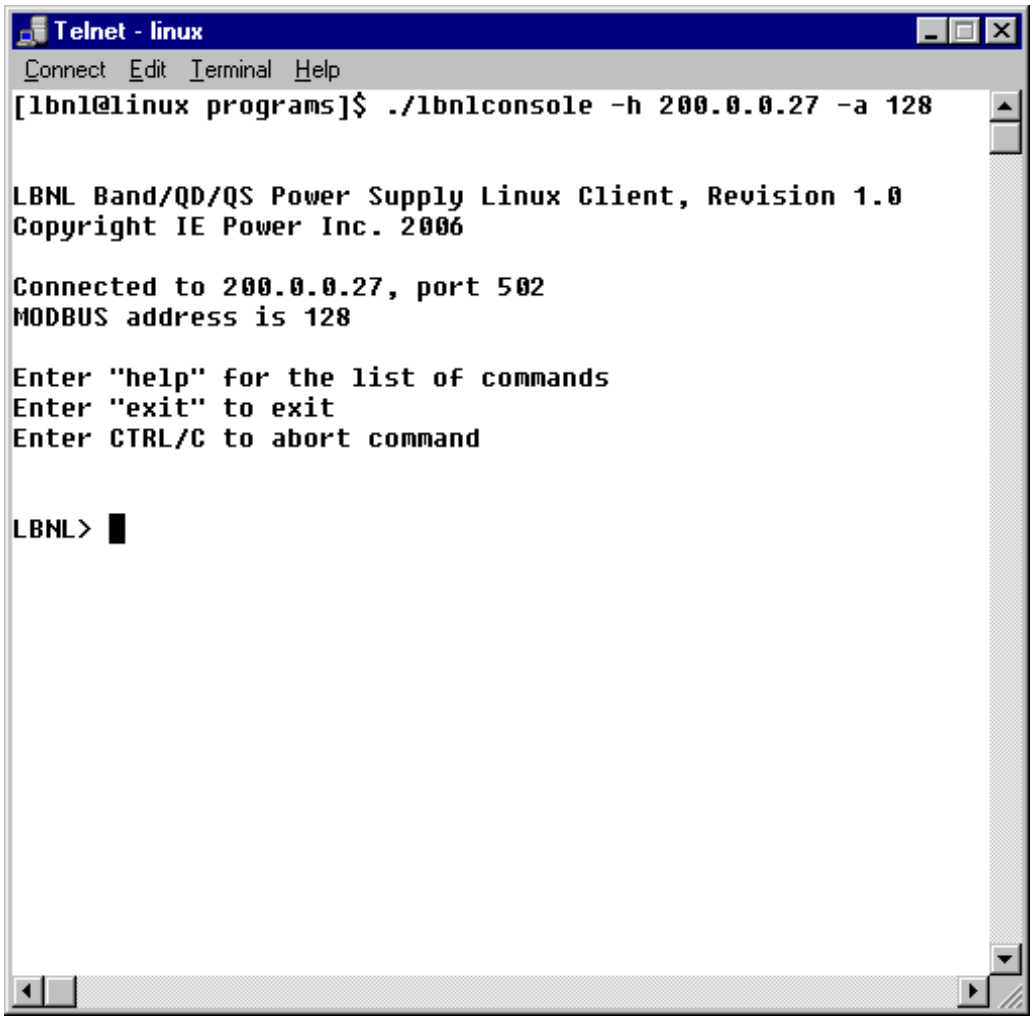
```
lbnlconsole -a nnn
```

where *nnn* stands for the Modbus address, between 1 and 247. In this case, the client will set the Modbus address to *nnn*.

For example, if you invoke the client as follows:

```
lbnlconsole -h 200.0.0.27 -a 128
```

the client will connect to the power supply with IP address 200.0.0.27, using Modbus address 128. In such case, the initial screen will be slightly different than before:



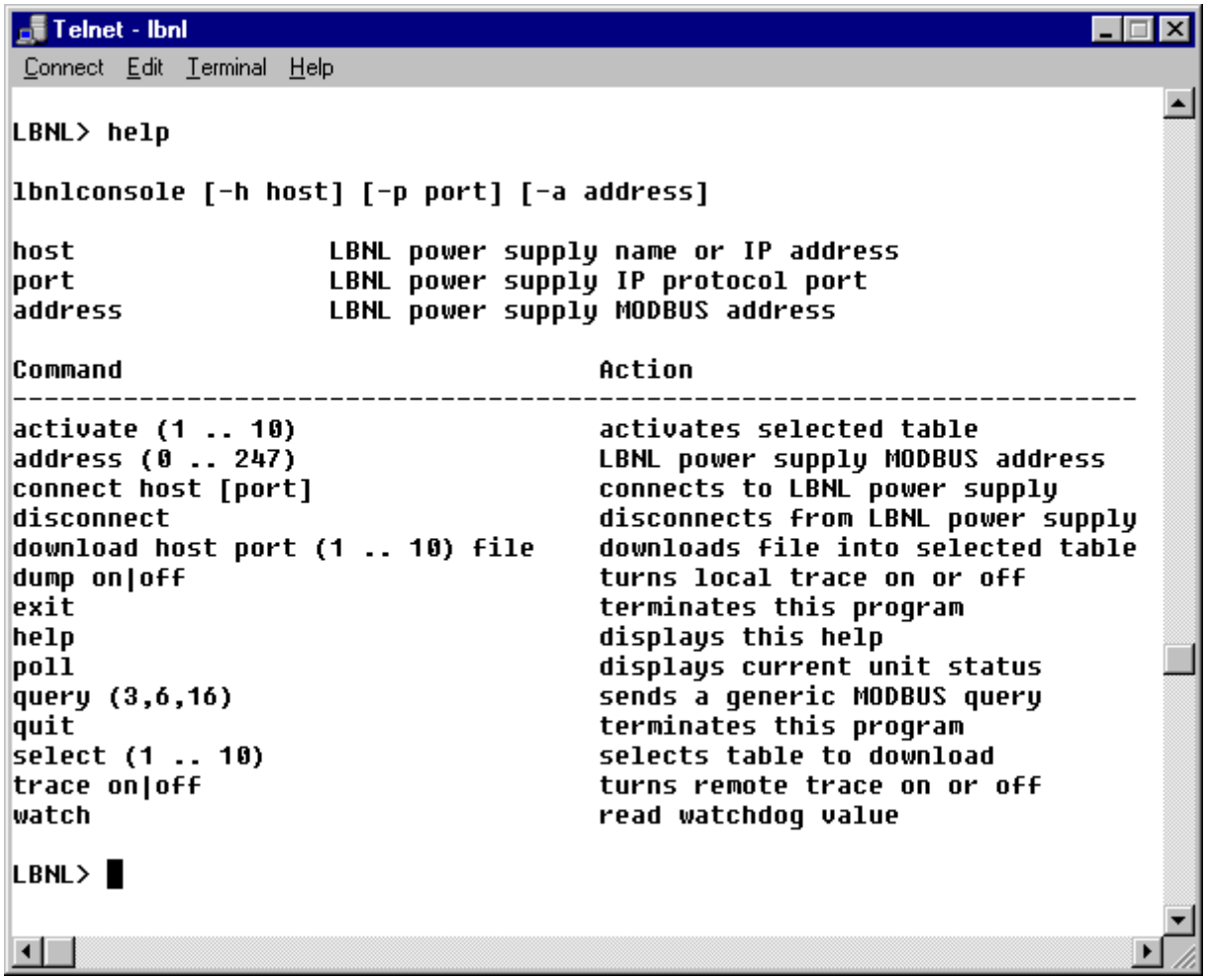
Note that the client now informs you that it is already connected to the protocol port 502 on 200.0.0.27, and the Modbus address is set to 128.

7.2 Commands

You can view the entire list of available commands by typing *help*:

```
LBNL> help
```

The help screen is depicted in the following screen snapshot:



You can enter any command in either capital or lower case letters, but you cannot mix capital or lower case letters. You can enter each command in two ways:

- You can enter the command name followed by the required parameters on the same line. This is referred to as the command *short form*.
- You can enter only the command name and press ENTER. The command will then prompt you to enter the required parameters. This is referred to as the command *interactive form*.

For example, at the LBNL prompt you can enter *POWER ON*. The client will translate your entry into an appropriate Modbus query and execute it immediately. Alternatively, you could enter just keyword *power*:

```
LBNL> power
Enter on | off:
```

Note how the program prompts you to enter the required parameter, and offers you valid parameter values *on* or *off*. Some commands, such as *query*, require more than one parameter.

If you enter an invalid command name, the client will reply as follows:

```
LBNL> hello
Invalid command
LBNL>
```

Similarly, if you enter an invalid parameter, the client will reply as follows:

```
LBNL> power
Enter on | off: none
Invalid parameter
LBNL>
```

To terminate the client, use commands *exit* or *quit:*

```
LBNL> exit
```

or:

```
LBNL> quit
```

The following pages contain details on all client commands.

7.2.1 Activate Command

The *activate* command is used to select one of the tables saved in the controller flash memory and load it into the power supply RAM memory. The short form of this command is:

```
LBNL> activate nn
```

where *nn* stands for a table number from 1 to 10. Command interactive form is:

```
LBNL> activate
Enter table index (1 .. 10): nn
```

When you enter a correct table number, the client responds with:

```
Table nn activated
```

If you enter an invalid table number, the client responds with:

```
Invalid table index
```

7.2.2 Address Command

The *address* command is used to set the Modbus address. The selected address remains in effect until the subsequent *address* command. The short form for this command is:

```
LBNL> address nnn
```

where *nnn* stands for a number from 1 to 247. Command interactive form is:

```
LBNL> address
Enter (1 .. 247): nnn
```

When you enter a correct Modbus address, the client responds with:

```
MODBUS address set to nnn
```

7.2.3 Connect Command

The *connect* command is used to connect to a given power supply, or to the Modbus gateway. The TCP connection remains in effect until the subsequent *disconnect* command. The short form for this command is:

```
LBNL> connect host [port]
```

where:

host	stands for the IP address in nnn.nnn.nnn.nnn notation, or a known host name.
[port]	represents the TCP protocol port number. The square brackets [] indicate that this parameter is optional. If omitted, the client uses the Modbus TCP default protocol port 502.

Command interactive form is:

```
LBNL> connect
Enter power supply name or address: host
Enter protocol port (MODBUS default 502): [port]
```

When you enter a valid IP address or name and protocol port, the client responds with:

```
Connected to host, port nnn
```

where *nnn* stands for the actual port number. If you attempt to connect to a power supply, while the client is already connected to an another device, the client will respond with a message simmilar to thsi one:

```
Already connected to 200.0.0.27, port 502
```

7.2.4 Disconnect Command

The *disconnect* command is used to disconnect from a power supply or Modbus gateway. The short form for this command is:

```
LBNL> disconnect
```

This command does not require any parameters, therefore it does not have the interactive form. The client responds to *disconnect* command with:

```
Disconnected from host, port nnn
```

where:

host	stands for the IP address in nnn.nnn.nnn.nnn notation, or a known host name.
nnn	represents the TCP protocol port number.

If you use the *disconnect* command when the client is not connected to any device, the client responds with:

```
Not connected
```

7.2.5 Download Command

The *download* command is used to download a table into the controller flash memory. You can download up to 10 tables. The tables are identified by their *table number*, or index. Before using *download* command, select the destination table using the *select* command. The short form for this command is:

```
LBNL> download host nnn file
```

where:

host	stands for the IP address in *nnn.nnn.nnn.nnn* notation, or a known host name.
nnn	represents the TCP protocol port number. By default, the controller accepts download connections on port 4000.
file	represents a fully qualified path to the binary file that contains table data. This is the file on your PC that you wish to download into the selected table.

Command interactive form is:

```
LBNL> download
Enter host name: lbnl
Enter protocol port: 4000
Enter file name: my_table
```

When you enter the correct parameters, the client responds with:

```
Connected to lbnl. port 4000
file my_table downloaded
Disconnected from lbnl, port 4000
```

7.2.6 Dump Command

The *dump* command is used to enable or disable dumping Modbus packets on the screen. This feature is very useful during the system integration and when tracing configuration problems in the field. The short form for this command is:

```
LBNL> dump on | off
```

where *on* | *off* represents valid parameter values *on* or *off*. Command interactive form is:

```
LBNL> dump
Enter on | off:
```

When you enable the dump, the client responds with:

```
                : trace to stderr enabled
                :
   SEND: 00 01 00 00 00 06 80 06 07 6D 00 01
RECEIVE: 00 01 00 00 00 06 80 06 07 6D 00 01
                :
                :
```

When you disable the dump, the client responds with:

```
                : trace disabled
```

7.2.7 Exit Command

The *exit* command is used to terminate the client program execution and to return to the operating system prompt. The short form for this command is:

```
LBNL> exit
```

This command does not require any parameters, therefore it does not have the interactive form. If connected to a Modbus device, before exit the client implicitly executes a *disconnect* command first, and responds with:

```
Disconnected from host, port nnn
```

where:

host	stands for the IP address in nnn.nnn.nnn.nnn notation, or a known host name.
nnn	represents the TCP protocol port number.

7.2.8 Help Command

The *help* command displays a list of client program command line options, followed by a list of valid commands and required parameters. The short form for this command is:

```
LBNL> help
```

This command does not require any parameters, therefore it does not have the interactive form. The client responds with:

```
lbnlconsole [-h host] [-p port] [-a address]

host                LBNL power supply name or IP address
port                LBNL power supply IP protocol port
address             LBNL power supply MODBUS address

Command                                 Action
-------------------------------------------------------------------------
activate (1 .. 10)                      activates selected table
address (0 .. 247)                      LBNL power supply MODBUS address
connect host [port]                     connects to LBNL power supply
disconnect                              disconnects from LBNL power supply
download host port (1 .. 10) file       downloads file into selected table
dump on|off                             turns local trace on or off
exit                                    terminates this program
help                                    displays this help
poll                                    displays current unit status
query (3,6,16)                          sends a generic MODBUS query
quit                                    terminates this program
select (1 .. 10)                        selects table to download
trace on|off                            turns remote trace on or off
watch                                   read watchdog value
```

7.2.9 Poll Command

The *poll* command queries the power supply status registers, and displays a list of status messages, based on the reported power supply status. The short form for this command is:

```
LBNL> poll
```

This command does not require any parameters, therefore it does not have the interactive form. The client responds with a list of status messages similar to this one:

```
CSSB and CARG status:

Power supply on standby
Power supply on
Power supply in remote
Communication fault detected
Over current relay normal
Logic power failure
Power supply water flow low
Diode bridge temperature normal
Output over current
Output voltage normal
Ground fault over current
Current regulator normal

Press ENTER to continue...
```

Since all messages cannot fit into a standard 25 row screen, the command displays the CSSB and CARG status first, and then prompts you to press ENTER. After you read the first page and press ENTER, the client displays the CAFB and CDFB status, similar to the following one:

```
AC voltage normal
AC over voltage
AC over current
```

```
DC voltage normal
IGBT module normal
Capacitor bank fuse blown
Magnet over temperature or flow low
Main water flow and external normal
No external faults 2 or 3
Choke/xmfr and cabinet temperature normal
Input fuses blown or smoke detected
Safety normal
Main contactor normal
```

7.2.10 Query Command

The *query* command is a generic command that allows you to read or write Modbus registers not only in LBNL Power Supply, but in any other Modbus-compatible piece of equipment as well.

The query command supports the following function codes:

- Read multiple registers, function code 3
- Write single register, function code 6
- Write multiple registers, function code 16

The short form for this command is:

```
LBNL> query n
```

where *n* stands for the function code 3, 6, or 16. If you don't specify the function code, the command will prompt you as follows:

```
LBNL> query
Function codes:
   read multiple registers:   3
   write single register:     6
   write multiple registers: 16

Enter function code (3 | 6 | 16):
```

Once you enter a valid function code, the command will prompt you to enter the remaining parameters, depending on the function code.

To read multiple registers, you must specify the address of the first register[1], and the number of registers that you want to read:

1. Note that Modbus protocol always operates on consecutive, adjacent register locations.

```
Enter register start address: nnnn
Enter number of registers: m
```

where *nnnn* stands for a decimal number representing the first register address, and *m* is the number of consecutive registers that you want to read.

For example, if you wanted to read two registers starting from location 200, the command and response would look like this:

```
LBNL> query 3
Enter register start address: 200
Enter number of registers: 2
Register 200 value 67
Register 201 value 129
```

You can always use function code 3 to read a single register as well. Just enter 1 for the number of registers.

To write a single register, use function code 6. The command will prompt you to enter register address and the value to write into the register:

```
LBNL> query 6
Enter register address: nnnn
Enter register value: mmmm
```

where *nnnn* stands for a decimal number representing the register address, and *mmmm* is the integer value to write into the register.

For example, if you wanted to write value 127 into a register located at address 200, the command and response would look like this:

```
LBNL> query 6
Enter register address: 200
Enter register value: 127
Register 200 is set to 127
```

Finally, you can write values into a number of consecutive registers using a single query. To do so, use function code 16. The command will first prompt you to enter the address of the first register. Next, it will prompt you to enter the number of register you want to write. Based on that number, the command will then prompt you to enter the value for each register.

For example, writing values into 10 registers starting from address 100, would look on the screen as follows:

```
LBNL> query 16
Enter register start address: 100
Enter number of registers: 10
Enter register 100 value: 345
Enter register 101 value: 238
Enter register 102 value: 1024
Enter register 103 value: 65532
Enter register 104 value: 128
Enter register 105 value: 725
Enter register 106 value: 0
Enter register 107 value: 1
Enter register 108 value: 62
Enter register 109 value: 8653
Register values set
```

7.2.11 Quit Command

The *quit* command is used to terminate the client program execution and to return to the operating system prompt. The short form for this command is:

```
LBNL> quit
```

This command does not require any parameters, therefore it does not have the interactive form. If connected to a Modbus device, before exit the client implicitly executes a *disconnect* command first, and responds with:

```
Disconnected from host, port nnn
```

where:

host	stands for the IP address in nnn.nnn.nnn.nnn notation, or a known host name.
nnn	represents the TCP protocol port number.

7.2.12 Select Conmmand

The *select* command is used to select the target table before downloading binary data. The controller can save up to 10 tables into its flash memory. These tables are identified by their table number, or index, from 1 to 10. To activate downloaded tables, use *activate* command. The short form for this command is:

```
LBNL > select nn
```

where *nn* stands for the table number, or index, from 1 to 10. Command interactive form is:

```
LBNL > select
Enter table index (1 .. 10):
```

When you enter a valid table number, the client responds with:

```
Table nn selected for download
```

If you enter an invalid value, the client responds with:

```
Invalid table index
```

7.2.13 Trace Command

The *trace* command is used to enable or disable the remote trace on the LBNL Power Supply maintenance console. For more details on the power supply trace utility, please refer to the section *Trace* in the chapter *Controller*. This feature is very useful during the system integration and when tracing configuration problems in the field.

The short form for the *trace* command is:

```
LBNL> trace on | off
```

where *on* | *off* represents valid parameter values *on* or *off*. Command interactive form is:

```
LBNL> trace
Enter on | off:
```

When you enable the trace, the client responds with:

```
Remote trace is on
```

When you disable the trace, the client responds with:

```
Remote trace is off
```

7.2.14 Watch Command

The *watch* command reads the current value of the watchdog timer in the LBNL Power Supply. The power supply increments its watchdog timer every second, and wraps around after 65535 seconds. On power up, the watchdog timer is reset to zero. Reading different values when this register is polled indicates that the LBNL Power Supply controller works normally.

The short form for the *watch* command is:

```
LBNL> watch
```

This command does not require any parameters, therefore it does not have the interactive form. The power supply returns its watchdog timer as follows:

```
Watchdog value nnnnn
```

where:

nnnnn represents the current watchdog register value.

7.3 Modbus Exception Codes

If you send to the LBNL Power Supply an invalid function code, address or value, the power supply will respond with an *Modbus Exception Code*. In general, Modbus exception codes are defined as summarized in the following table:

Table 7-1: Modbus Exception Codes

Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the slave was attempting to perform the requested action.
05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed.
06	SLAVE DEVICE BUSY	The slave is engaged in processing a long–duration program command. The master should retransmit the message later when the slave is free.

When a Modbus device returns an error, the client presents the error on the screen using the following format:

```
ERROR n: MESSAGE
```

where:

n	is the error exception code.
MESSAGE	is the text message associated with the exception code.

For example, if you attempt to read a non-existent Modbus register, the client will display the following error message:

```
ERROR 2: INVALID REGISTER ADDRESS
```

Some Modbus devices may use additional error codes, not listed in the above table. When it encounters an unknown error code, the client responds with:

```
ERROR n: UNKNOWN ERROR
```

8 Manual Pages

This chapter contains Linux-style manual pages for the programs installed on the LBNL Power Supply controller, and for the programs intended to run on Linux-based PCs.

The information on each program is presented using a standard set of headings:

NAME	contains program name and a single-line description
SYNOPSIS	specifies the exact program calling syntax
DESCRIPTION	contains expanded descritpion of the program
OPTIONS	lists all applicable command line options

8.1 console

NAME

console - local user interface on the LBNL Power Supply

SYNOPSIS

console

DESCRIPTION

This script file resides in the LBNL Power Supply directory */flash/programs*. It is used to start the *lbnlconsole* program with the command line options appropriate for the power supply local user interface.

The script connects to localhosts and specifies the Modbus address. By default, this script uses Modbus address 128. If you change the address in the *lbnld.cfg* configuration file, you must modify the *console* script accordingly.

By default, the *console* script contains the following line:

```
/flash/programs/lbnlconsole -hlocalhost -a128
```

OPTIONS

none

8.2 inb

NAME

inb - read 8-bit byte from an IO register

SYNOPSIS

inb hex_port

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. It is used to read a single byte from the specified IO register *hex_port*. Register address must be entered as a hexa-decimal number from 0 to ffff.

OPTIONS

none

8.3 inw

NAME

inw - read 16-bit word from an IO register

SYNOPSIS

inw hex_port

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. It is used to read a single 16-bit word from the specified IO register *hex_port*. Register address must be entered as a hexadecimal number from 0 to ffff.

OPTIONS

none

8.4 lbnld

NAME

lbnld - LBNL Power Supply server

SYNOPSIS

lbnld [-a address] [-d] [-f file] [-p port][&]

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. This is the power supply server, responsible for all monitoring and control functions. The program normally starts at LBNL Power Supply boot time, as a background daemon.

To run *lbnld* in foreground, login into the LBNL Power Supply maintenance console as *root* and change directory to */flash/programs*.

OPTIONS

-a address

specifies the LBNL Power Supply Modbus address. To use this command line option you must remove the *ADDRESS* entry from the *lbnld.cfg* configuration file. Otherwise, the configuration file will overwrite the command line option.

-d

enables the LBNL Power Supply trace to the maintenance console.

-f file

selects a configuration file other than default *lbnld.cfg*.

-p port

selects the TCP/IP protocol port used to download tables. If omitted, the program uses the default value of 4000.

8.5 lbnlconsole

NAME

lbnlconsole - LBNL Power Supply user interface

SYNOPSIS

lbnlconsole [-h host] [-p protocol_port] [-a modbus_address]

DESCRIPTION

This program is the LBNL Power Supply user interface. The program can run from any Linux-based PC, or on the LBNL Power Supply controller itself.

The command set and other features are described in the LBNL Power Supply controller manual, chapter *Client*.

OPTIONS

-a modbus_address

selects the Modbus address to use on startup. Valid values are integers from 1 to 247. Once *lbnlconsole* starts, you can always change the address using the *address* command.

-h host

specifies the IP address or host name to connect to on startup. Once *lbnlconsole* starts, you can always change the connection using the *disconnect* and *connect* commands.

-p protocol_port

specifies the TCP protocol port to use to connect to a Modbus device. If omitted, *lbnlconsole* uses the Modbus default protocol port 502. Once *lbnlconsole* starts, you can always change the protocol port using the *disconnect* and *connect* commands.

8.6 outb

NAME

outb - write 8-bit value into an IO register

SYNOPSIS

outb hex_port hex_value

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. It is used to write a single byte into the specified IO register *hex_port*. Register address must be entered as a hexadecimal number from 0 to ffff. Register value must be entered as a hexadecimal number from 0 to ff.

OPTIONS

none

8.7 outw

NAME

outw - write 16-bit value into an IO register

SYNOPSIS

outw hex_port hex_value

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. It is used to write a 16-bit word into the specified IO register *hex_port*. Register address and value must be entered as hexadecimal numbers from 0 to ffff.

OPTIONS

none

8.8 rdmem

NAME

rdmem - read specified memory location

SYNOPSIS

rdmem hex_address

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. It is used to read a 16-bit word from the specified memory location *hex_address*. Memory address must be entered as a hexadecimal number.

OPTIONS

none

8.9 wrmem

NAME

wrmem - write value into the specified memory location

SYNOPSIS

wrmem hex_address hex_value

DESCRIPTION

This program resides in the LBNL Power Supply directory */flash/programs*. It is used to write a 16-bit word into the specified memory location *hex_address*. Memory address and value must be entered as hexadecimal numbers.

OPTIONS

none