korenix

JetBox 5630

Linux User Manual

www.korenix.com

0.0.4

Copyright Notice

Information provided in this manual is intended to be accurate and reliable. However, the original manufacturer assumes no responsibility for its use, or for any infringements upon the rights of third parties that may result from its use.
The material in this document is for product information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, Korenix assumes no liabilities resulting from errors or omissions in this document, or from the use of the information contained herein.
Korenix reserves the right to make changes in the product design without notice to its users.

Acknowledgments

Korenix is a registered trademark of Korenix Technology Co., Ltd.
All other trademarks or registered marks in the manual belong to their respective manufacturers.

Table of Content

Chapter 1 Overview

One of the advantages of adopting Korenix JetBox series industrial computers is ready-to-use. All these years, Korenix is devoted to enhance usability and functions of embedded computers in industrial domain. Korenix operating system provides device drivers, protocol stacks, system utilities, supporting services and daemons to make system integration simple. Besides, Korenix also provides application development toolkits for users to build their own applications easily.

The stylish compact JetBox 5630Gf-w series are industrial layer 3 VPN routers with Linux computing capability. It is a gateway to connect different network groups such as Ethernet and serial control in a complex networking architecture and manage peripherals at the front-end site. With Gigabit Ethernet, fiber connection and ability of network redundancy, JetBox 5630Gf-w series can be applied in crossroads or highway for flow control and traffic monitoring of remote transportation control. Besides, JetBox 5630Gf-w series are designed with features of compact, reliable and robust to adopt in various industrial vertical markets with hazardous environment such as transportation, surveillance and environmental monitoring.

Applied Models

JetBox 5630 series

Note: SW features might be different according to different products.

Chapter 2 Getting Start

2-1 System login

Users can enter the JetBox Linux environment via the user name: root and no password is required.

login : root

password : (none)

2-1-1 Serial Console

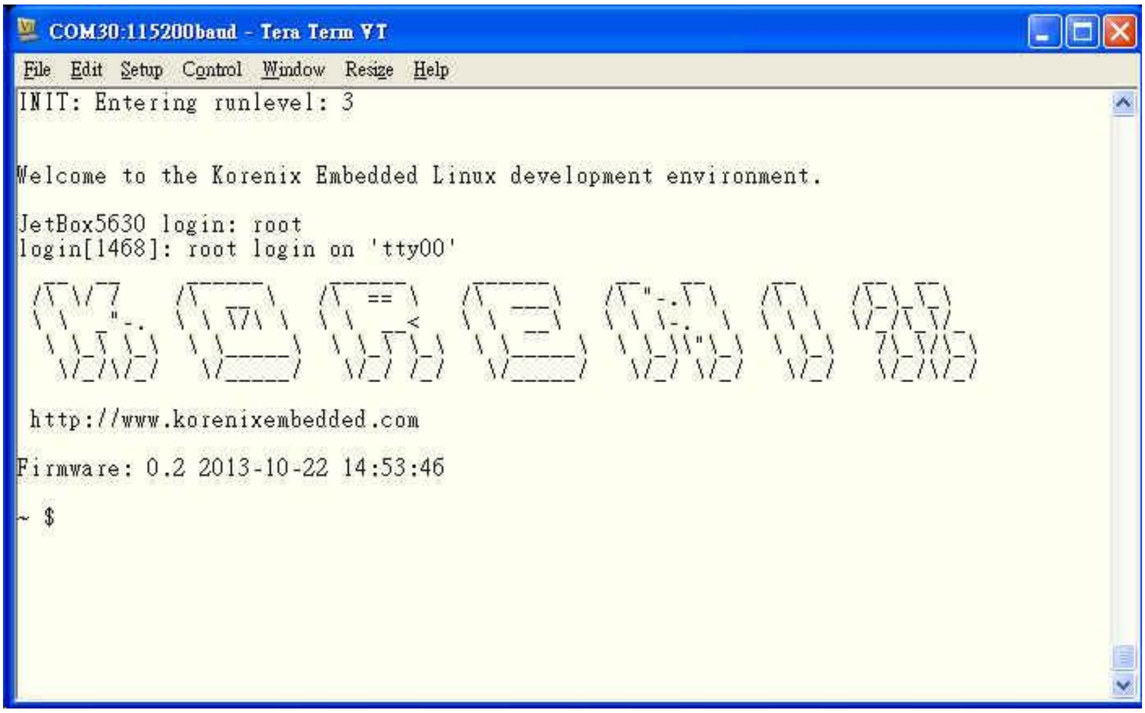
The serial console port gives users a convenient way of connecting to JetBox console utility. This method is particularly useful when using JetBox for the first time. The signal is transmitted over a direct serial connection, so you do not need to know either of JetBox's IP address in order to connect to the serial console utility.

Use the serial console port settings shown below.

Baud rate	115200bps
Parity	None
Data bits	8
Stop bits	1
Flow Control	None

Serial console port setting

Once the connection is established, the following windows will open.


COM30:115200baud - Tera Term VT
File Edit Setup Control Window Resize Help

```
INIT: Entering runlevel: 3

Welcome to the Korenix Embedded Linux development environment.

JetBox5630 login: root
login[1468]: root login on 'tty00'
 __  __     ______     ______     ______     __   __     __     __  __
/\ \/ /    /\  __ \   /\  == \   /\  ___\   /\ "-.\ \   /\ \   /\_\_\_\
\ \  _"-.  \ \ \/\ \  \ \  __<   \ \  __\   \ \ \-.  \  \ \ \  \/_/\_\/_
 \ \_\ \_\  \ \_____\  \ \_\ \_\  \ \_____\  \ \_\\"\_\  \ \_\   /\_\/\_\
  \/_/\/_/   \/_____/   \/_/ /_/   \/_____/   \/_/ \/_/   \/_/   \/_/\/_/

 http://www.korenixembedded.com

Firmware: 0.2 2013-10-22 14:53:46

~ $
```

Serial console screen

To log in, type the Login name and password as requested. The default values as following.

Login: **root**

Password: none

2-1-2 Telnet Console

If you know IP addresses and netmasks, then you can use Telnet to connect to JetBox.

The default IP address and Netmask for each port is given below:

	Default IP address	Netmask
LAN	192.168.10.1	255.255.255.0
WAN	DHCP	

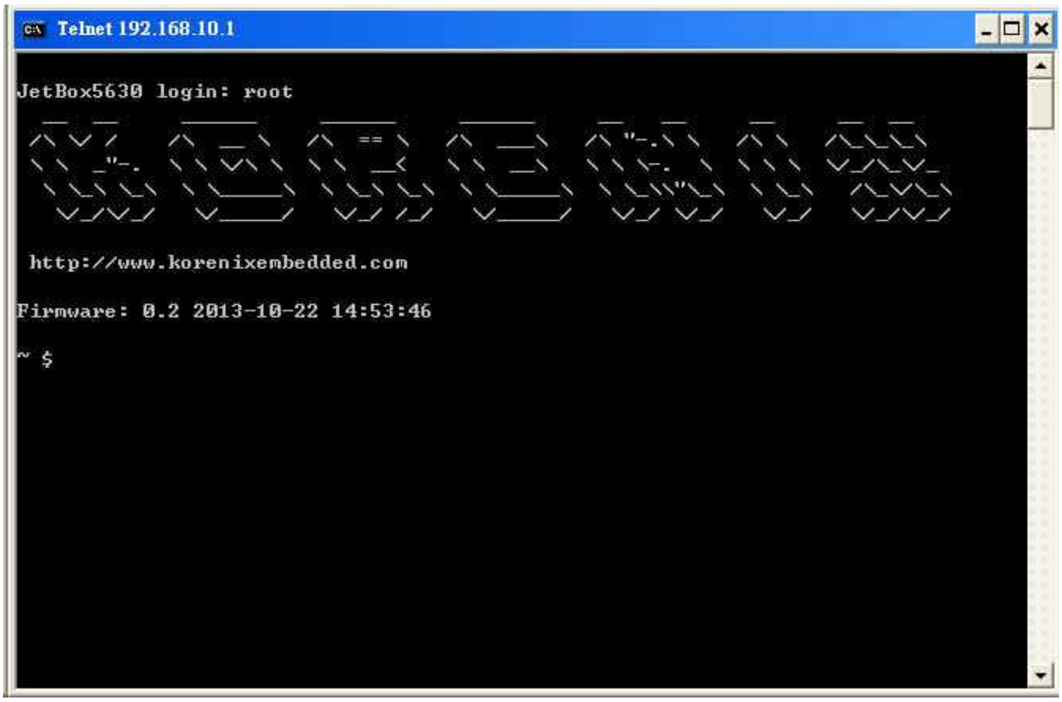
Default IP address and Netmask

Use a cross-over Ethernet Cable to connect directly from your PC to JetBox 8100. You should first modify your PC's IP address and netmask so that your PC is on the same subnet as JetBox.

To connect to a hub or switch connected to your local LAN, use a straight-through Ethernet cable. The default IP address and Netmask are shown above. To login, type the Login name and password as requested. The default values as following:

Login: **root**

Password: none



```
JetBox5630 login: root

 http://www.korenixembedded.com

Firmware: 0.2 2013-10-22 14:53:46

~ $
```

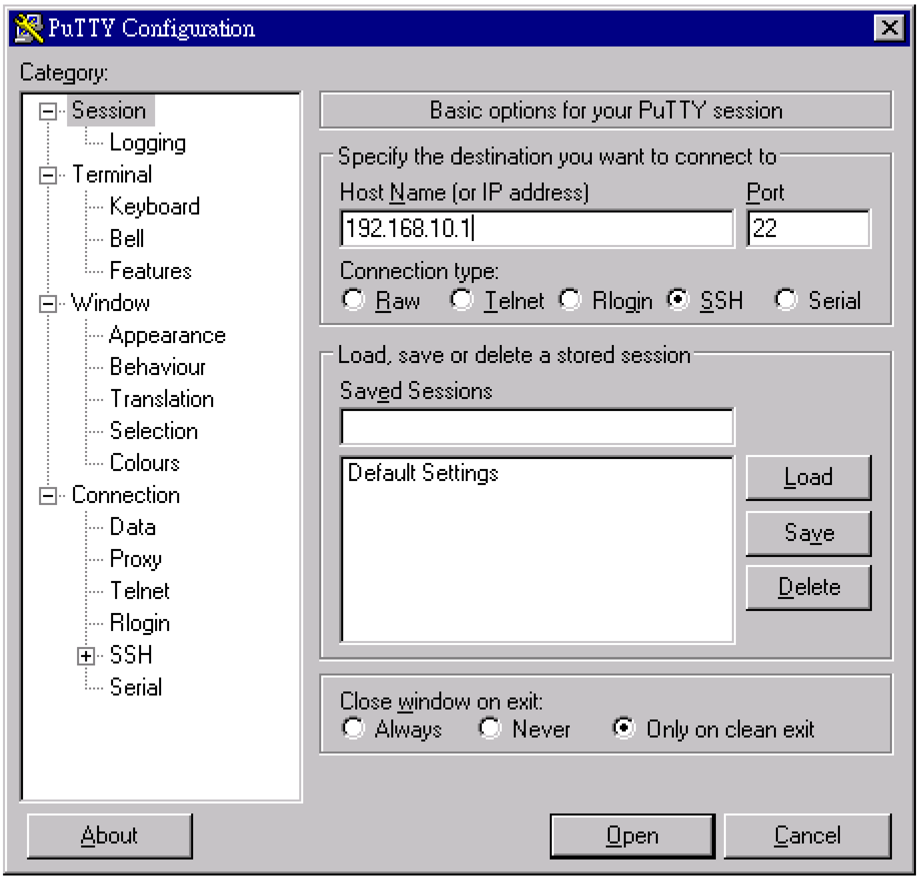
Telnet console screen

You can proceed with the configuration of JetBox's network settings when you reach the bash command shell. Configuration instructions are given in the next section.

2-1-3 SSH Console

We also supports an SSH Console to offer users with better security options.

Click on the link putty to download PuTTy(freeware) and set up an SSH console for JetBox in a Windows environment. The following figure shows an example of the configuration that is required.



Host Name (or IP address): 192.168.10.1

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Windows PuTTy setting

2-2 Configure Ethernet Interface

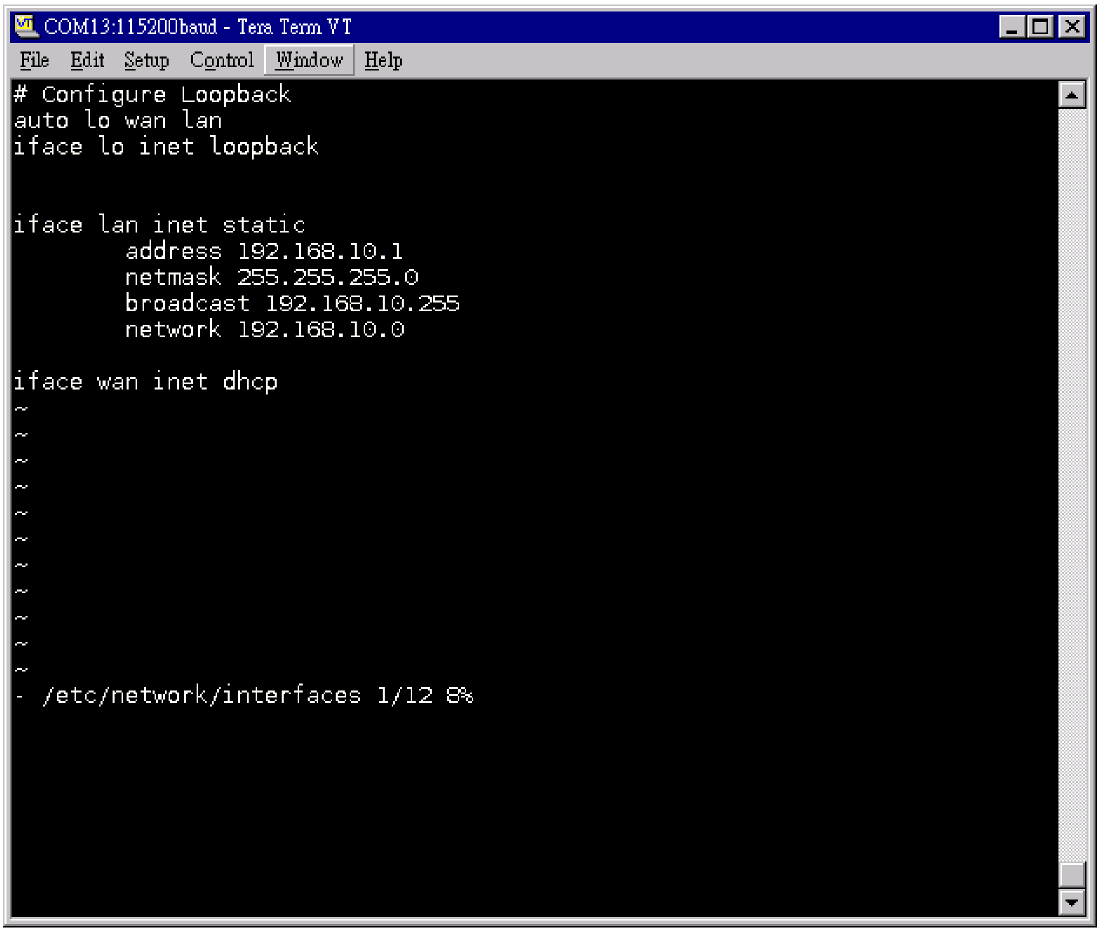
JetBox’s network setting can be modified with the serial console, or over the network.

2-2-1 Modifying Network Settings with the Serial Console

In this section, we use the serial console to modify JetBox’s network settings

- **Change Network Configuration**

  Follow the instructions given in a previous section to access JetBox’s Console Utility via the serial Console port, and then type 'vi /etc/network/interfaces' to edit network configuration file with vi editor.


COM13:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
# Configure Loopback
auto lo wan lan
iface lo inet loopback


iface lan inet static
        address 192.168.10.1
        netmask 255.255.255.0
        broadcast 192.168.10.255
        network 192.168.10.0

iface wan inet dhcp
~
~
~
~
~
~
~
~
~
~
~
- /etc/network/interfaces 1/12 8%
```

Edit Network configuration file

- ## Static and Dynamic IP address

Static IP address:

As shown in below, 4 fields must be modified: **address**, **netmask**, **broadcast** and **network**. The default IP addresses are 192.168.10.1.

```
iface lan inet static
        address 192.168.10.1
        netmask 255.255.255.0
        broadcast 192.168.10.255
        network 192.168.10.0
```

Dynamic IP addresses:

By default, the Jetbox is configured for “static” IP addresses on LAN port and dhcp on WAN interface. To configure LAN port to request an IP address dynamically, remove the original settings and add the following line.

```
iface lan inet dhcp
```

Default setting for LAN port	Default setting for WAN port
iface lan inet static address 192.168.10.1 netmask 255.255.255.0 broadcast 192.168.10.255 network 192.168.10.0	iface wan inet dhcp

Default Gateway:

When static IP Address setting is used, add a default gateway is to set another keyword “gateway”. For example:

```
iface wan inet static
        address 192.168.1.2
        netmask 255.255.255.0
        broadcast 192.168.1.255
        network 192.168.1.0
        gateway 192.168.1.1    # This will set default gateway to
                               #192.168.1.1 on WAN port
```

2-2-2 Static and Dynamic IP address

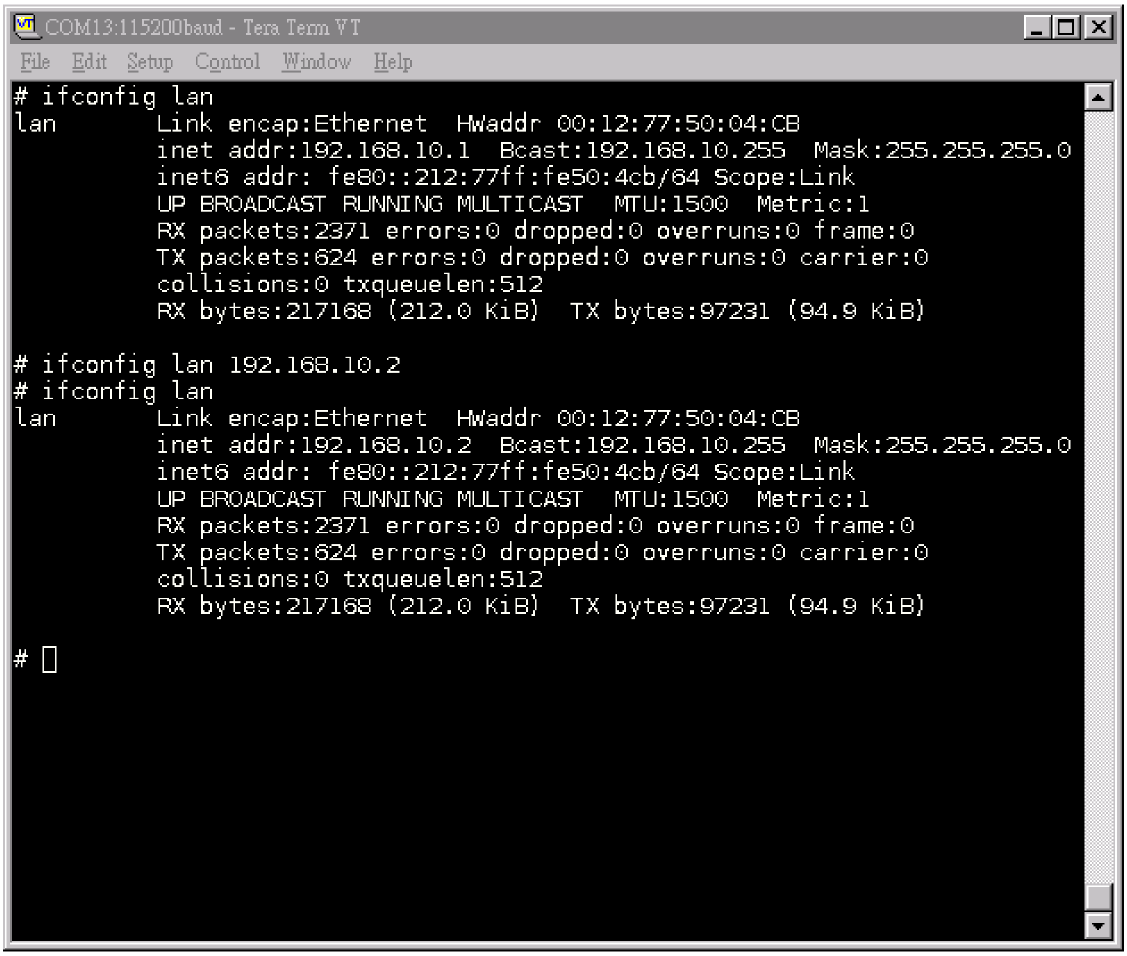
After the /etc/network/interfaces file have been modified, issue the following command to apply the network settings immediately:

```
/etc/init.d/network restart
```

2-2-3 Modifying Network Settings over the Network

Same the previous section, IP settings can be modified over the network, too. There is another way to change the IP address without modifying the file /etc/network/interfaces, but the new settings will not be saved to the flash disk.

For example, type the command **#ifconfig lan 192.168.10.2** to change the IP address of LAN interface to 192.168.10.2.


COM13:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
# ifconfig lan
lan       Link encap:Ethernet  HWaddr 00:12:77:50:04:CB
          inet addr:192.168.10.1  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::212:77ff:fe50:4cb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2371 errors:0 dropped:0 overruns:0 frame:0
          TX packets:624 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:217168 (212.0 KiB)  TX bytes:97231 (94.9 KiB)

# ifconfig lan 192.168.10.2
# ifconfig lan
lan       Link encap:Ethernet  HWaddr 00:12:77:50:04:CB
          inet addr:192.168.10.2  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::212:77ff:fe50:4cb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2371 errors:0 dropped:0 overruns:0 frame:0
          TX packets:624 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:217168 (212.0 KiB)  TX bytes:97231 (94.9 KiB)

# 
```

Network Setting over the Network

2-3 Test Program Developing – Hello.c

In this section, we use the standard “Hello” programming example to illustrate how to develop a program for the JetBox5630. In general, program development involves the following seven steps.

Step 1:

Connect the JetBox5630 to a Linux PC.

Step 2:

Install SDK on the Linux PC.

Step 3:

Set the cross compiler and PATH environment variables.

Step 4:

Code and compile the program.

Step 5:

Download the program to the JetBox5630 via FTP.

Step 6:

Debug the program

- If bugs are found, return to Step 4.
- If no bugs are found, continue with Step 7.

Step 7:

Back up the user directory (distribute the program to additional JetBox5630 units if needed).

2-3-1 Installing the SDK (Linux)

The Linux Operating System must be pre-installed in the PC before installing the JetBox5630 SDK. Ubuntu core or compatible versions are recommended. The SDK requires approximately 750 MB of hard disk space on your PC. The JetBox5630 SDK can be downloaded from Korenix web site. To install the SDK, it is simply a matter of extracting a tarball at the proper place:

```
mkdir -p /korenix
tar jxvf jetbox5630-sdk-0.1.tgz -C /korenix_sdk
```

NOTE

- To install the Toolchain, you must grant root permission.
- Toolchains used to not be relocatable! You must install them in the location they were built for.

Install toolchain is simple, just extract to the your directory with the command.
Next, set up the PATH environment variable, **go to the /korenix_sdk/jetbox5630**. And type follow command to set up environment variable

```
root@:/korenix_sdk/jetbox5630#./setup_5630.sh
5630_SDK setup completed!
Please use command ". linux-devkit/environment-setup" to source environment
```

Source the environment variable

```
root@:/korenix_sdk/jetbox5630#. linux-devkit/environment-setup
```

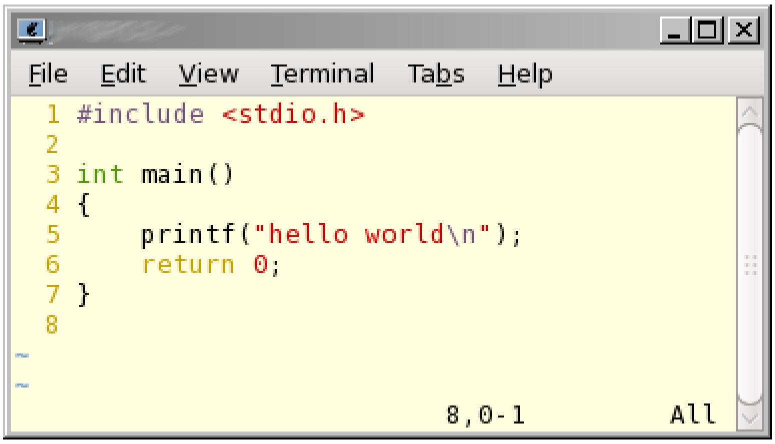
Now you can build the program and run it on JetBox5630.

2-3-2 Compiling Hello.c

If you have been compiling a program on X86, then you will find the only difference is the GCC command is start with **arm-arago-linux-gnueabi-**.

That is because we want to differentiate with the stand GCC compiler, and the prefix also tell you – it is for arm, little-endian platform program.

Below is a simple hello.c program:



```
#include <stdio.h>

int main()
{
    printf("hello world\n");
    return 0;
}

```

To compile the helo.c, use our Toolchain to compile the hello.c:

```
arm-arago-linux-gnueabi-gcc hello.c -o helloworld
```

The output executable file is the *helloworld*.

2-3-3 Uploading “helloworld” to JetBox5630 and Running the Program

Use the following command to upload helloworld to the JetBox5630 via FTP.

(Please refer to Chapter 3-3 to enable ftp server)

1. From the PC, type:

# ftp xxx.xxx.xxx.xxx

2. Use *bin* command to set the transfer mode to Binary mode, and the put command to initiate the file transfer:

ftp> binary

ftp> put helloworld

```
[root@server ~]# ftp 192.168.10.1
Connected to 192.168.10.1 (192.168.10.1).
220 ProFTPD 1.3.1 Server (ProFTPD TEST Installation) [::ffff:192.168.10.1]
Name (192.168.10.1:root): root
331 Password required for root
Password:
230 User root logged in
```

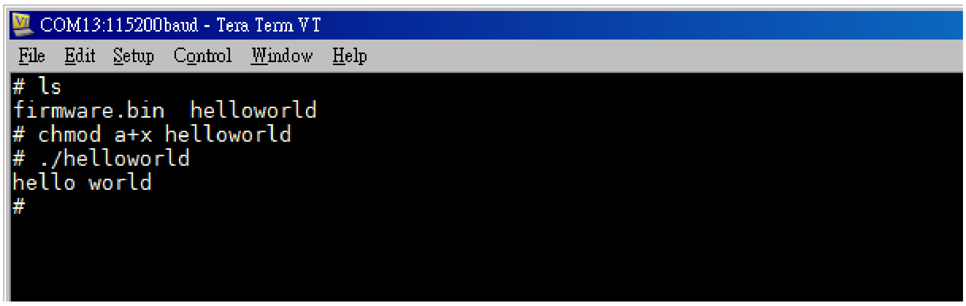
```
Remote system type is UNIX.
ftp> cd /home/
250 CWD command successful
ftp> binary
200 Type set to I
ftp> put helloworld
local: helloworld remote: helloworld
227 Entering Passive Mode (192,168,1,176,19,6).
150 Opening BINARY mode data connection for helloworld
226 Transfer complete
4455 bytes sent in 4.1e-05 secs (108658.54 Kbytes/sec)
ftp> bye
221 Goodbye.
```

3. From the JetBox5630 console, type:

# chmod +x helloworld

# ./helloworld

The word hello world will be printed on the screen.


COM13:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
# ls
firmware.bin  helloworld
# chmod a+x helloworld
# ./helloworld
hello world
#
```

Chapter 3 System Feature

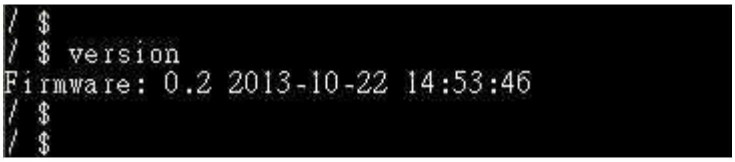
This chapter includes information about version control, deployment, updates, and peripherals. The information in this chapter will be particularly useful when you need to run the same application on several JetBox units.

3-1 System Version

To determine the hardware capability of your JetBox, and what kind of software functions are supported, check the version numbers of your JetBox's hardware, kernel, and user file system. Contact Korenix to determine the hardware version. You will need the Production S/N (Serial number), which is located on the JetBox5630's back label.

To check the **firmware** version, type:

```
# version
```



```
/ $
/ $ version
Firmware: 0.2 2013-10-22 14:53:46
/ $
/ $
```

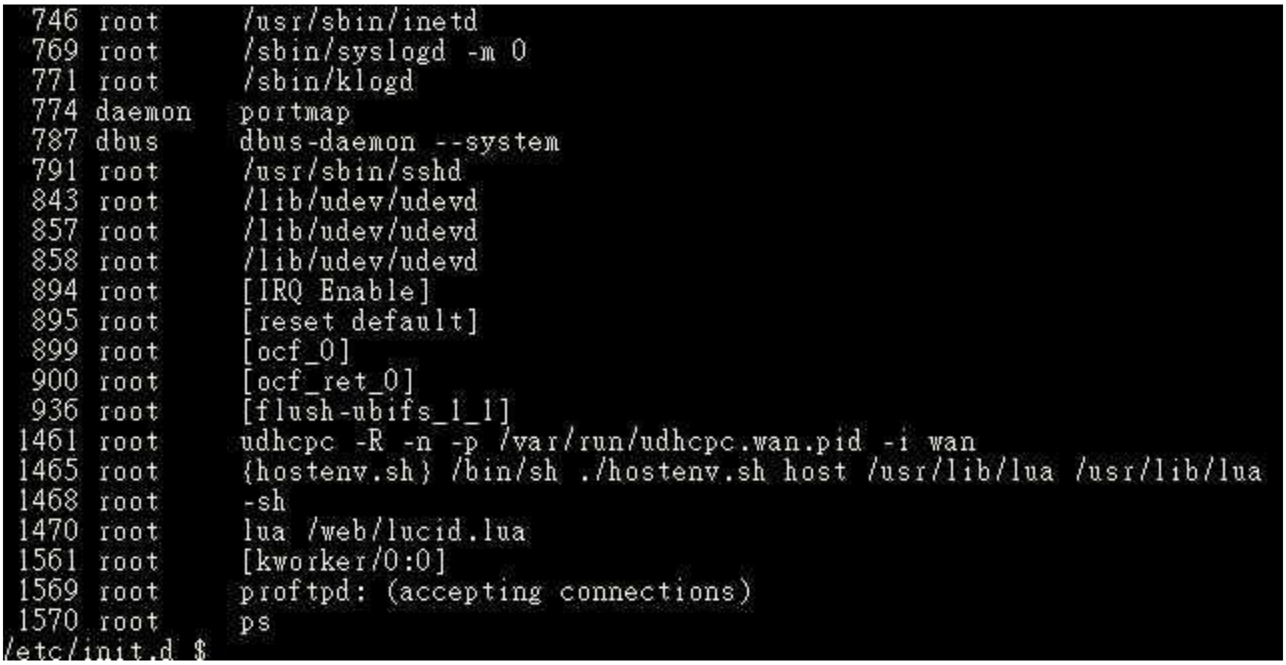
Figure 3-1 Firmware version

3-2 Enable/Disable Daemons

The following daemons are enabled when the JetBox 5630 boots up for the first time.

Service name	Description
inetd	internet daemons
telnetd	telnet daemon
sshd	secure shell daemon
proftpd	ftp daemon

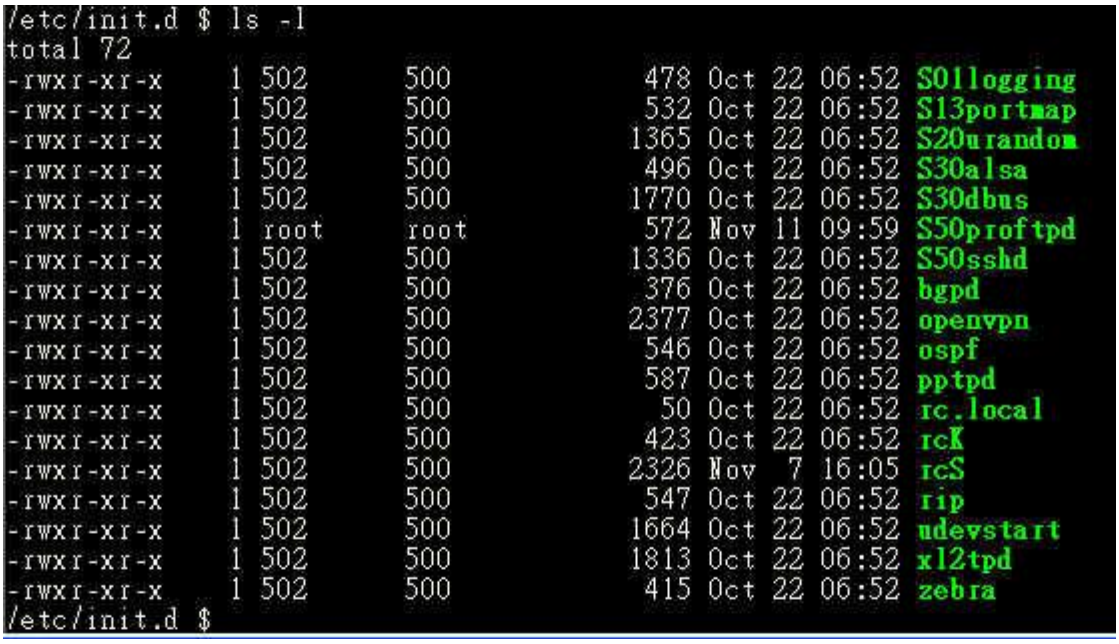
Type the command "**ps**" to list all processes currently running.



```
 746 root      /usr/sbin/inetd
 769 root      /sbin/syslogd -m 0
 771 root      /sbin/klogd
 774 daemon    portmap
 787 dbus      dbus-daemon --system
 791 root      /usr/sbin/sshd
 843 root      /lib/udev/udevd
 857 root      /lib/udev/udevd
 858 root      /lib/udev/udevd
 894 root      [IRQ Enable]
 895 root      [reset default]
 899 root      [ocf_0]
 900 root      [ocf_ret_0]
 936 root      [flush-ubifs_1_1]
1461 root      udhcpc -R -n -p /var/run/udhcpc.wan.pid -i wan
1465 root      {hostenv.sh} /bin/sh ./hostenv.sh host /usr/lib/lua /usr/lib/lua
1468 root      -sh
1470 root      lua /web/lucid.lua
1561 root      [kworker/0:0]
1569 root      proftpd: (accepting connections)
1570 root      ps
/etc/init.d $
```

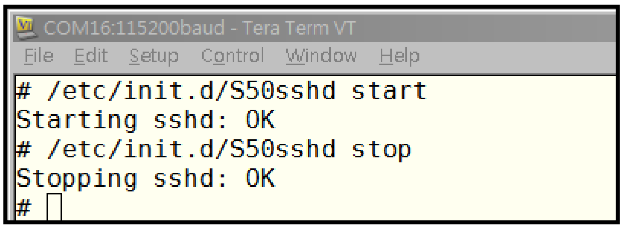
daemons status

The /etc/init.d directory is the repository for all available init scripts.



```
/etc/init.d $ ls -l
total 72
-rwxr-xr-x    1 502      500           478 Oct 22 06:52 S01logging
-rwxr-xr-x    1 502      500           532 Oct 22 06:52 S13portmap
-rwxr-xr-x    1 502      500          1365 Oct 22 06:52 S20urandom
-rwxr-xr-x    1 502      500           496 Oct 22 06:52 S30alsa
-rwxr-xr-x    1 502      500          1770 Oct 22 06:52 S30dbus
-rwxr-xr-x    1 root     root          572 Nov 11 09:59 S50proftpd
-rwxr-xr-x    1 502      500          1336 Oct 22 06:52 S50sshd
-rwxr-xr-x    1 502      500           376 Oct 22 06:52 bgpd
-rwxr-xr-x    1 502      500          2377 Oct 22 06:52 openvpn
-rwxr-xr-x    1 502      500           546 Oct 22 06:52 ospf
-rwxr-xr-x    1 502      500           587 Oct 22 06:52 pptpd
-rwxr-xr-x    1 502      500            50 Oct 22 06:52 rc.local
-rwxr-xr-x    1 502      500           423 Oct 22 06:52 rcK
-rwxr-xr-x    1 502      500          2326 Nov  7 16:05 rcS
-rwxr-xr-x    1 502      500           547 Oct 22 06:52 rip
-rwxr-xr-x    1 502      500          1664 Oct 22 06:52 udevstart
-rwxr-xr-x    1 502      500          1813 Oct 22 06:52 xl2tpd
-rwxr-xr-x    1 502      500           415 Oct 22 06:52 zebra
/etc/init.d $
```

Here is an example of starting and stopping the ssh daemon:



COM16:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
# /etc/init.d/S50sshd start
Starting sshd: OK
# /etc/init.d/S50sshd stop
Stopping sshd: OK
#
```

You can start the service by adding the first argument start, and stop the service by adding the first argument stop

3-3 Setting System Time

The JetBox5630 has two time settings. One is the system time, and the other is the RTC (Real Time Clock) time kept by the JetBox5630's hardware.

Use the **#date** command to query the current system time or set a new system time.

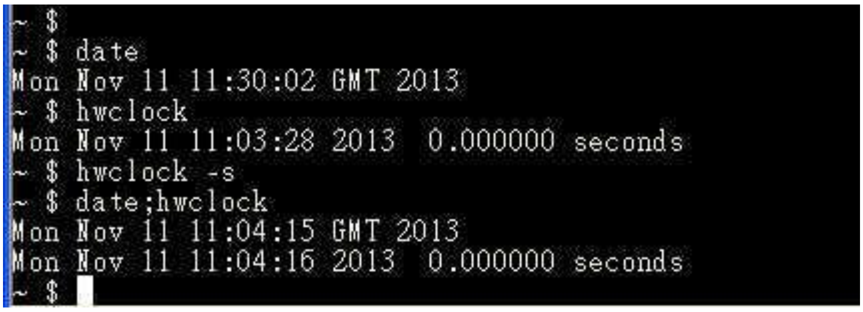
```
#date MMDDhhmmYYYY
MM = Month
DD = Date
hhmm = hour and minute
YYYY = Year
```

Use **#hwclock** to query the current RTC time

Use the following command to set system time from hardware clock:

```
#hwclock -s
```

The following figure illustrates how to update the system time and set the RTC time.



```
~ $
~ $ date
Mon Nov 11 11:30:02 GMT 2013
~ $ hwclock
Mon Nov 11 11:03:28 2013  0.000000 seconds
~ $ hwclock -s
~ $ date;hwclock
Mon Nov 11 11:04:15 GMT 2013
Mon Nov 11 11:04:16 2013  0.000000 seconds
~ $
```

Setting the Time Manually

3-4 Adjust System Time

If you only wish to synchronize your clock when the device boots up, you can use **ntpdate**. This may be appropriate for some devices which are frequently rebooted and only require infrequent synchronization.

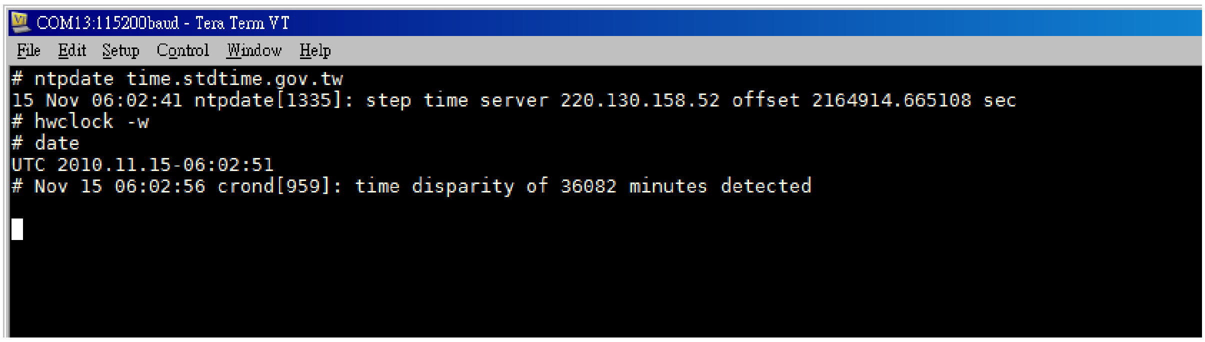
Using ntpdate at boot time is also a good idea for devices that run ntpd. The ntpd program changes the clock gradually, whereas ntpdate sets the clock, no matter how great the difference between a device's current clock setting and the correct time.

3-4-1 NTP Client

The JetBox has a built-in NTP (Network Time Protocol) client that is used to initialize a time request to a remote NTP server.

Use **#ntpdate** to update the system time.

```
#ntpdate time.stdtime.gov.tw

#hwclock -w
```


COM13:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
# ntpdate time.stdtime.gov.tw
15 Nov 06:02:41 ntpdate[1335]: step time server 220.130.158.52 offset 2164914.665108 sec
# hwclock -w
# date
UTC 2010.11.15-06:02:51
# Nov 15 06:02:56 crond[959]: time disparity of 36082 minutes detected
```

NTP client request

Visit http://www.ntp.org for more information about NTP and NTP server addresses.

NOTE

Before using the NTP client utility, check your IP and DNS settings to make sure that an Internet connection is available.

3-5 Cron Daemon (Schedule jobs)

Cron is a daemon to execute scheduled commands. Cron wakes up every minute, examining the /etc/crontab, checking each command to see if it should be run in the current minute.

Crontab syntax :

A crontab file has five fields for specifying day, date and time followed by the command to be run at that interval.

```
*     *     *   *    *       command to be executed
-     -     -   -    -
|     |     |   |    |
|     |     |   |    +----- day of week (0 - 6) (Sunday=0)
|     |     |   +------- month (1 - 12)
|     |     +--------- day of month (1 - 31)
|     +----------- hour (0 - 23)
+------------- min (0 - 59)
```

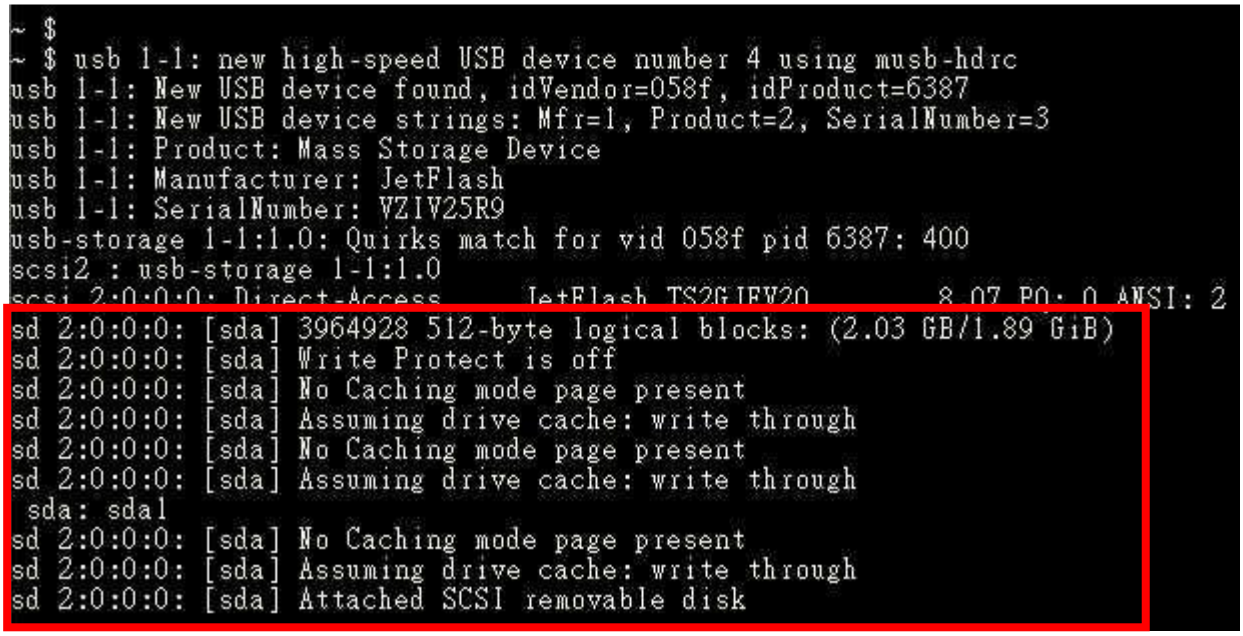
Crontab example :

A line in crontab file like below removes the tmp files from /tmp each day at 6:30 PM.

```
30  18  *  *  *         rm /tmp/*
```

3-6 Connect Peripherals

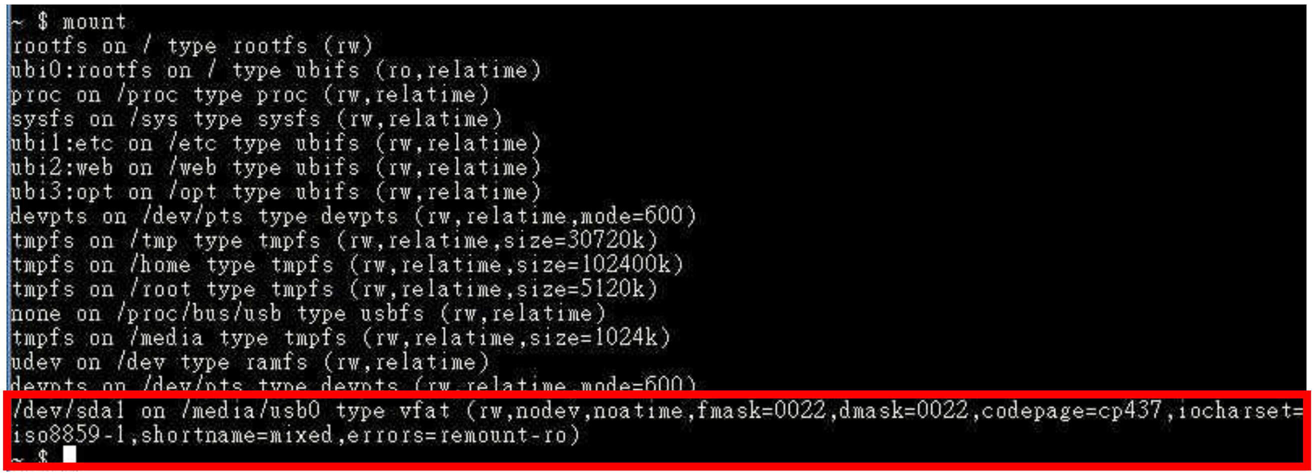
While plug-in a USB mass storage or a SD card, use #**dmesg** command can help showing USB-storage device status.



```
~ $
~ $ usb 1-1: new high-speed USB device number 4 using musb-hdrc
usb 1-1: New USB device found, idVendor=058f, idProduct=6387
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1: Product: Mass Storage Device
usb 1-1: Manufacturer: JetFlash
usb 1-1: SerialNumber: VZIV25R9
usb-storage 1-1:1.0: Quirks match for vid 058f pid 6387: 400
scsi2 : usb-storage 1-1:1.0
scsi 2:0:0:0: Direct-Access     JetFlash TS2GJFV20        8.07 PQ: 0 ANSI: 2
sd 2:0:0:0: [sda] 3964928 512-byte logical blocks: (2.03 GB/1.89 GiB)
sd 2:0:0:0: [sda] Write Protect is off
sd 2:0:0:0: [sda] No Caching mode page present
sd 2:0:0:0: [sda] Assuming drive cache: write through
sd 2:0:0:0: [sda] No Caching mode page present
sd 2:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 2:0:0:0: [sda] No Caching mode page present
sd 2:0:0:0: [sda] Assuming drive cache: write through
sd 2:0:0:0: [sda] Attached SCSI removable disk
```

usb-storage device scan status

JetBox5630 support auto-mount external storage, just use mount command to check the mount point.



```
~ $ mount
rootfs on / type rootfs (rw)
ubi0:rootfs on / type ubifs (ro,relatime)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
ubi1:etc on /etc type ubifs (rw,relatime)
ubi2:web on /web type ubifs (rw,relatime)
ubi3:opt on /opt type ubifs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600)
tmpfs on /tmp type tmpfs (rw,relatime,size=30720k)
tmpfs on /home type tmpfs (rw,relatime,size=102400k)
tmpfs on /root type tmpfs (rw,relatime,size=5120k)
none on /proc/bus/usb type usbfs (rw,relatime)
tmpfs on /media type tmpfs (rw,relatime,size=1024k)
udev on /dev type ramfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600)
/dev/sda1 on /media/usb0 type vfat (rw,nodev,noatime,fmask=0022,dmask=0022,codepage=cp437,iocharset=
iso8859-1,shortname=mixed,errors=remount-ro)
~ $
```

As the picture shows, the usb-storage has been mounted on /media/usb0. You can access your data in the /media/usb0 folder.

To un-mount the usb-storage, execute #umount <mount path>. For example, issue #umount /media/usb0 can un-mount the previous mounted directory.

NOTE

To be able to unmount a device, you have to close all the open files in it. Type **sync** can help commits all pending writes, which can then be removed in a safe way.

Chapter 4 Network Feature

In this chapter, we explain how to configure JetBox various communication functions.

4-1 Telnet

Service name	telnetd
Description	A Telnet server
Config files	/etc/inetd.conf
Start file	/etc/init.d/rcS
Start command	
Stop command	
Support command	
Default	up

Enabling the Telnet server

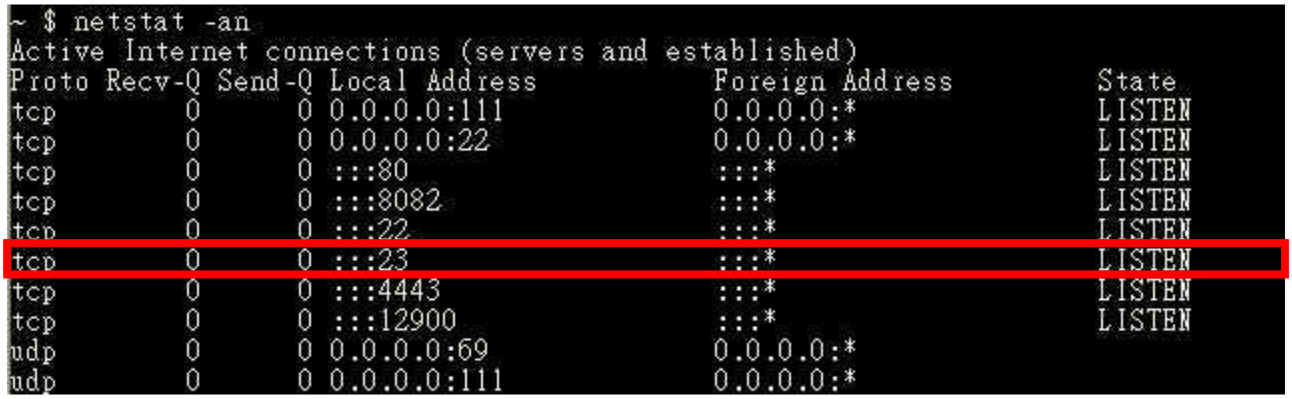
The following example shows the default content of the file /etc/inetd.conf. The default is to enable the Telnet server:

```
telnet stream tcp6 nowait root /usr/sbin/telnetd telnetd -i
```

Disabling the Telnet server

Disable the daemon by typing '#' in front of the first character of the row to comment out the line.

```
# telnet stream tcp6 nowait root /usr/sbin/telnetd telnetd -i
```



```
~ $ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 :::80                   :::*                    LISTEN
tcp        0      0 :::8082                 :::*                    LISTEN
tcp        0      0 :::22                   :::*                    LISTEN
tcp        0      0 :::23                   :::*                    LISTEN
tcp        0      0 :::4443                 :::*                    LISTEN
tcp        0      0 :::12900                :::*                    LISTEN
udp        0      0 0.0.0.0:69              0.0.0.0:*
udp        0      0 0.0.0.0:111             0.0.0.0:*
```

netstat

NOTE

We also support telnet with IPv6 address.

Example：telnet fe80::212:77ff:fe50:1ba8%eth0

4-2 SSHD

Service name	sshd
Description	A ssh server
Config files	/etc/sshd_config /etc/ssh_config /etc/ssh_host_dsa_key /etc/ssh_host_dsa_key.pub /etc/ssh_host_ecdsa_key /etc/ssh_host_ecdsa_key.pub /etc/ssh_host_key /etc/ssh_host_key.pub /etc/ssh_host_rsa_key /etc/ssh_host_rsa_key.pub
Start file	/etc/init.d/S50sshd
Start command	/etc/init.d/S50sshd start
Stop command	/etc/init.d/S50sshd stop
Support command	
Default	up

Re-generate sshd host keys

The JetBox5630 comes with a set of default sshd host keys. To re-generate it, remove them and restart the ssh daemon.

```
# rm -f /etc/ssh_host_dsa_key /etc/ssh_host_dsa_key.pub /etc/ssh_host_ecdsa_key
/etc/ssh_host_ecdsa_key.pub /etc/ssh_host_key /etc/ssh_host_key.pub
/etc/ssh_host_rsa_key /etc/ssh_host_rsa_key.pub
# /etc/init.d/S50sshd restart
```

NOTE

We also support ssh login with IPv6 address.

Example：ssh fe80::212:77ff:fe50:1ba8%eth0

4-3 FTP

Service name	proftpd
Description	A Highly configurable FTP server
Config files	/etc/proftpd.conf
Start file	/etc/init.d/proftpd
Start command	/etc/init.d/proftpd start
Stop command	/etc/init.d/proftpd stop
Support command	
Default	down

Enabling root login

Edit the /etc/proftpd.conf and add the following line, then restart the FTP server.

```
RootLogin on
```

4-4 DNS

To set up DNS client, you need to edit two configuration files:
/etc/resolv.conf,
/etc/hosts (optional)

/etc/hosts - The static table lookup for host names

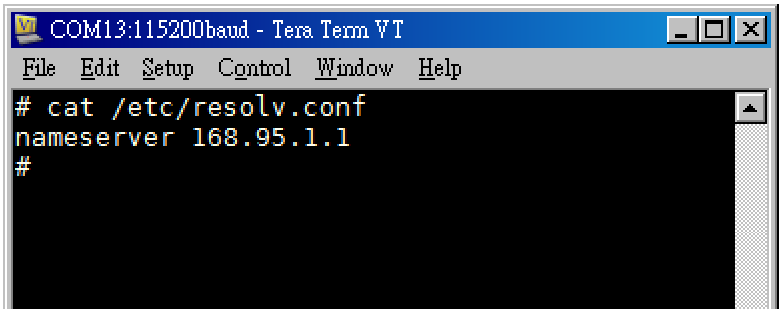
This is the first file that the Linux system reads to resolve the host name and IP address.

/etc/resolv.conf – DNS resolver configuration file

This is the most important file that you need to edit when using DNS for the other programs. For example, before you use #ntpdate time.nist.goc to update the system time, you will need to add the DNS server address to the file. Ask your network administrator which DNS server address you should use.

The DNS server's IP address is specified with the "nameserver" command. For example, add the following line to /etc/resolv.conf if the DNS server's IP address is 168.95.1.1:

```
nameserver 168.95.1.1
```


COM13:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
# cat /etc/resolv.conf
nameserver 168.95.1.1
#
```

nameserver

4-5 IPTABLES

program name	iptables
Description	Administration tool for IPv4 packet filtering and NAT
Usage: iptables -[AD] chain rule-specification [options] iptables -[RI] chain rulenum rule-specification [options] iptables -D chain rulenum [options] iptables -[LFZ] [chain] [options] iptables -[NX] chain iptables -E old-chain-name new-chain-name iptables -P chain target [options] iptables -h (print this help information)	

iptables is a user space application program on JetBox5630 that allows to configure the tables provided by the Linux kernel firewall (implemented as different Netfilter modules) and the chains and rules it stores. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables as a special for Ethernet frames. A firewall using iptables is said to be a stateful firewall.

iptables splits the packet handling into three different tables, each of which contain a number of chains. The firewalling rules, which we create, are included within a particular chain. The three tables are:

1. **filter**: used for packet filtering
2. **nat**: used to provide packet modification capabilities; NAT/PAT and IP masquerading
3. **mangle**: used for setting packet options and marking packets for further filtering or routing

The **filter** table is the default table for any rule. It is where the bulk of the work in an iptables firewall occurs. This table contains three chains:

1. INPUT: used for traffic which is entering our system and belongs to an IP address which is on our local machine
2. OUTPUT: used for traffic which originated on the local system, otherwise known as the firewall
3. FORWARD: used for traffic which is being routed between two network interfaces on our firewall

There are three main targets for a rule within the filter table.

1. ACCEPT: allows the packet to be passed through the firewall without any noticeable interaction
2. DROP: simply drops the packet as if it has never been in the system
3. REJECT: drops the packet then sends a ICMP reply back to the client telling it why the connection failed

Example:

Add rules

The basic syntax of an iptables command is:

```
iptables -A INPUT -s 192.168.20.0/24 -j ACCEPT
```

This would add a rule into the INPUT chain, which matches any packet with a source address in the 192.168.20.0 subnet. If a packet matches this criteria, then it would use the ACCEPT target, which simply allows the packet on through.

Remove rules

To delete the first rule in the chain, we would do:

```
iptables -D INPUT 1
```

List rules

To list the rules we have on our system use:

```
iptables -L
```

Flush rules

To flush (drop) all the rules we can use:

```
iptables -F
```

A more complete tutorial can be found at:

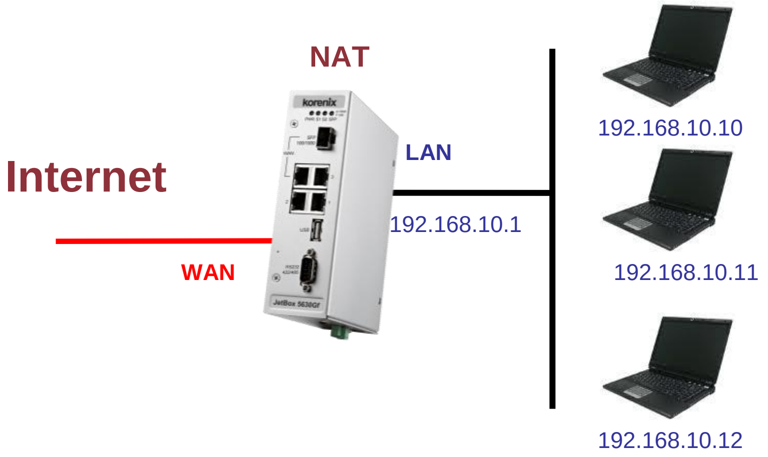
http://www.linode.com/wiki/index.php/Netfilter_IPTables_Mini_Howto

4-6 NAT

Network address translation (NAT) is the process of modifying network address information in datagram (IP) packet headers while in transit across a traffic routing device for the purpose of remapping one IP address space into another.

A basic NAT scenario:

The 2 interfaces concerned will be WAN and LAN.
WAN: This will be the interface connected to the Internet.
LAN: This interface will be connected to the private network.



Assuming that you have already configured your system to be able to connect to Internet, run the following command to enable NAT.

```
iptables -t nat -A POSTROUTING -o wan -j MASQUERADE
iptables -A FORWARD -i wan -o lan -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i lan -o wan -j ACCEPT
```

NOTE
If WAN interface use PPPOE to connect to ISP, use **ppp0** interface instead of WAN port.

To save current setting of iptables to a file, use the following command:

```
iptables-save > /etc/iptables-rules
```

To restore the previous saved file, use the following command:

```
iptables-restore /etc/iptables-rules
```

4-7 Dail-up Service

Service name	pppd
Description	Point-to-Point Protocol Daemon
Config files	/etc/options /etc/pap-secrets /etc/chap-secrets
Start file	
Start command	
Stop command	
Support command	**pppd** **chat** **pppdump**
Default	down

PPP (Point to Point Protocol) is used to run IP (Internet Protocol) and other network protocols over a serial link. PPP can be used for direct serial connections (using a null-modem cable) over a Telnet link, and links established using a modem over a telephone line. Usually pppd is called by other daemon, like PPPoE, PPTP, and Wvdial.

4-8 PPPoE

Service name	pppoe
Description	Point-to-Point Protocol over ethernet
Config files	/etc/ppp/pppoe.conf
Start file	
Start command	
Stop command	
Support command	**pppoe-connect** **pppoe-setup** **pppoe-start** **pppoe-stop** **pppoe-status**
Default	down

Point-to-Point Protocol over Ethernet is a network protocol for encapsulating Poing-to-Point Protocol (PPP) frames inside Ethernet frames. It is used mainly with ADSL services where individual users connect to the ADSL transceiver over Ethernet.

You can just inpput pppoe-setup to configure the PPPoE. First you should obtain PPPOE username and password from your Internet provider. Second, filled in these parameters to pppoe-setup dialog:

- **Ethernet Interface: wan**
- **User name: <from your ISP>**
- **Activate-on-demand: No**
- **Primary DNS: 168.95.1.1**
- **Firewalling: NONE**

```
# pppoe-setup
Welcome to the Roaring Penguin PPPoE client setup.  First, I will run
some checks on your system to make sure the PPPoE client is installed
properly...

Looks good!  Now, please enter some information:

USER NAME
>>> Enter your PPPoE user name (default bxxxnxnx@sympatico.ca): 7360011@hinet.net

INTERFACE
>>> Enter the Ethernet interface connected to the DSL modem
(default eth0): wan

Do you want the link to come up on demand, or stay up continuously?
>>> Enter the demand value (default no): no

DNS

Please enter the IP address of your ISP's primary DNS server.
>>> Enter the secondary DNS server address here:

PASSWORD

>>> Please enter your PPPoE password:
```

Then use the pppoe-start command to start dial to connect network.

4-9 NFS

program name	mount
Description	A NFS client
Usage: mount -t nfs -o nolock NFS_Server_Address:/directory /mount/point	
Example: A NFS server export it's /root directory with IP address 192.168.1.10, want to mount to JetBox's /mnt directory: mount -t nfs -o nolock 192.168.1.10:/root /mnt	

NOTE

Read the following links for more information about setting up a NFS server:

- http://nfs.sourceforge.net/nfs-howto/
- http://nfs.sourceforge.net/nfs-howto/ar01s04.html

4-10 Samba

Service name	smbd
Description	Server to provide SMB/CIFS services to clients
Config files	/etc/samba/smb.conf
Start command	/etc/init.d/samba start
Stop command	/etc/init.d/samba stop
Default	down

smbd is the server daemon that provides file sharing and printing services to Windows clients. The server provides filespace and printer services to clients using the SMB (or CIFS) protocol. This is compatible with the LanManager protocol and can service LanManager clients. These include MSCLIENT 3.0 for DOS, Windows for Workgroups, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, DAVE for Macintosh, and smbfs for Linux..

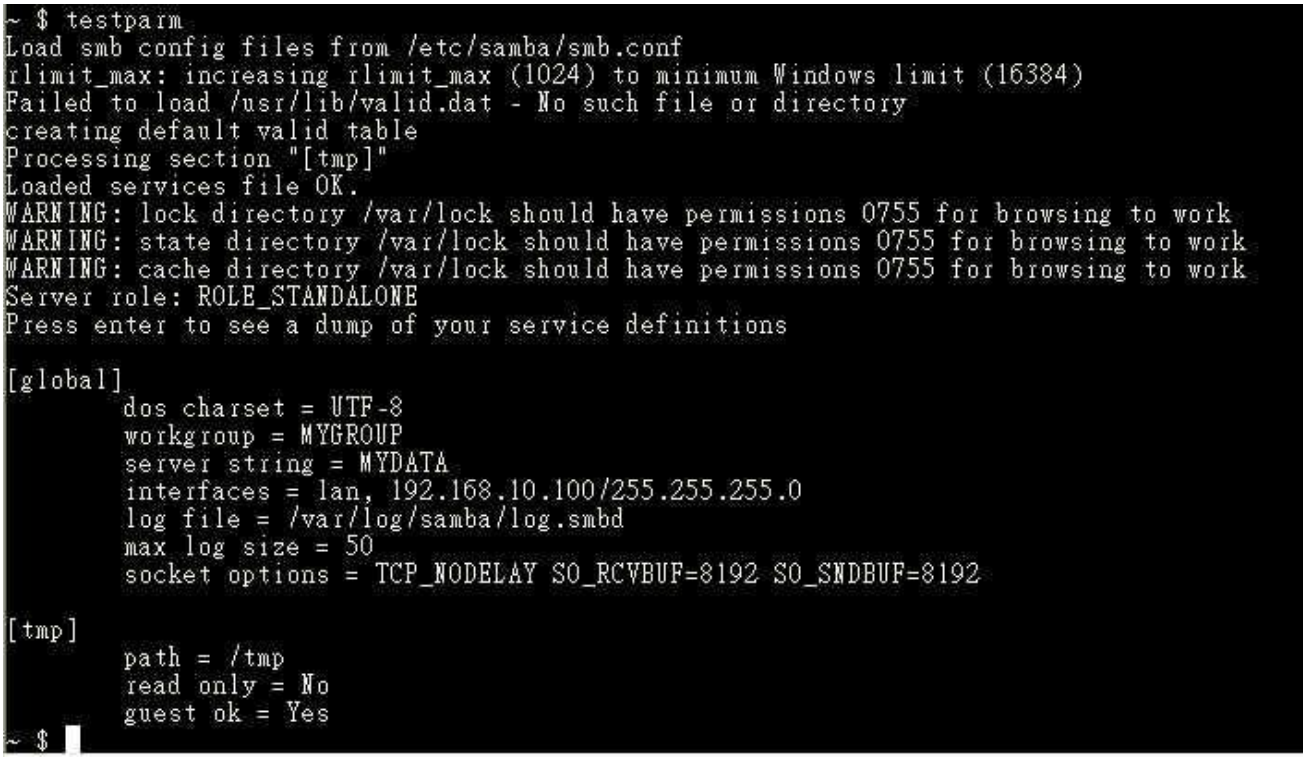
Example:

Edit Config File ：/etc/samba/smb.conf

Samba configuration on a Linux (or other UNIX machine) is controlled by a single file, /etc/smb.conf. This file determines which system resources you want to share with the outside world and what restrictions you wish to place on them

```
#======================= Global Settings =====================================
[global]
dos charset = UTF-8
workgroup = MYGROUP
server string = MYDATA
max log size = 50
security = user
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
log file = /var/log/samba/log.smbd
interfaces = lan 192.168.10.100/255.255.255.0
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
#============================ Share Definitions ==============================
[tmp]
path = /tmp
public = yes
read only = no
writable = yes
```

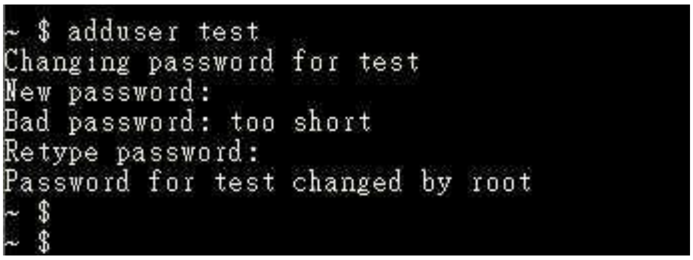
Testparm — check an smb.conf configuration file for internal correctness



```
~ $ testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Failed to load /usr/lib/valid.dat - No such file or directory
creating default valid table
Processing section "[tmp]"
Loaded services file OK.
WARNING: lock directory /var/lock should have permissions 0755 for browsing to work
WARNING: state directory /var/lock should have permissions 0755 for browsing to work
WARNING: cache directory /var/lock should have permissions 0755 for browsing to work
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
        dos charset = UTF-8
        workgroup = MYGROUP
        server string = MYDATA
        interfaces = lan, 192.168.10.100/255.255.255.0
        log file = /var/log/samba/log.smbd
        max log size = 50
        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

[tmp]
        path = /tmp
        read only = No
        guest ok = Yes
~ $
```

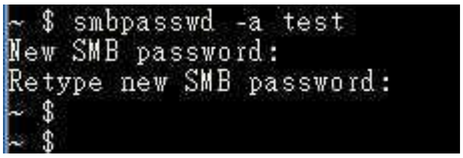
Smbpasswd：Add user to access sharing folder

First, create a user in the local. Use **adduser** command



```
~ $ adduser test
Changing password for test
New password:
Bad password: too short
Retype password:
Password for test changed by root
~ $
~ $
```

Create samba user password for **test**



```
~ $ smbpasswd -a test
New SMB password:
Retype new SMB password:
~ $
~ $
```

Now, you can access the sharing folder in Windows. Go to Start → Run and type command

```
\\192.168.10.100
```

Input samba user name and password and you can see the sharing folder.

4-11 SNMP

Service name	snmpd
Description	SNMP Daemon
Config files	/etc/snmp/snmpd.conf
Start file	
Start command	
Stop command	
Support command	snmpget snmpset
Default	down

Net-SNMP is a suite of applications used to implement SNMP v1, SNMP v2c and SNMP v3 using both IPv4 and IPv6. It supports RFC 1213 MIB-II.

For more information, read the following links about NET-SNMP:

http://www.net-snmp.org/wiki/index.php/Tutorials

4-12 OpenVPN

Service name	openvpn
Description	A full-featured SSL VPN
Config files	/etc/openvpn/ /etc/openvpn/easy-rsa/
Start file	/etc/init.d/openvpn
Start command	/etc/init.d/openvpn start
Stop command	/etc/init.d/openvpn stop
Default	down

OpenVPN is a full-featured SSL VPN which implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or username/password credentials, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface.

For more information, download the step-by-step how to from Korenix website:
http://www.korenixembedded.com/support/faqs/vpn

4-13 IPSec

Service name	ipsec
Description	A full-featured IPSec VPN
Config files	/etc/ipsec.conf
Start file	/etc/init.d/ipsec
Start command	/etc/init.d/ipsec start
Stop command	/etc/init.d/ipsec stop
Default	down

Usage:

```
ipsec setup [ --showonly ] {--start|--stop|--restart}
ipsec setup –status

ipsec auto [--showonly] [--asynchronous] --up connectionname
ipsec auto [--showonly] --{add|delete|replace|down} connectionname
ipsec auto [--showonly] --{route|unroute} connectionname
ipsec auto [--showonly] --{ready|status|rereadsecrets|rereadgroups}
ipsec auto [--showonly] --{rereadcacerts|rereadaacerts|rereadocspcerts}
ipsec auto [--showonly] --{rereadacerts|rereadcrls|rereadall}
ipsec auto [--showonly] [--utc] --{listpubkeys|listcerts}
ipsec auto [--showonly] [--utc] --{listcacerts|listaacerts|listocspcerts}
ipsec auto [--showonly] [--utc] --{listacerts|listgroups}
ipsec auto [--showonly] [--utc] --{listcrls|listocsp|listall}
ipsec auto [--showonly] --purgeocsp
```

OpenSwan is an implementation of IPsec Protocol for Linux. You can create a VPN using ipsec command. ipsec invokes any of several utilities involved in controlling the IPsec encryption/authentication system, running the specified command with the specified arguments as if it had been invoked directly.

For more information, download the step-by-step how to from Korenix website:
http://www.korenixembedded.com/support/faqs/vpn
Or read OpenSwan Wiki:
http://wiki.openswan.org/

4-14 PPTP Client

program name	pptp
Description	A Point-to-Point potocol client
Config file	/etc/ppp/options.pptp /etc/ppp/chap-secrets

Usage:

```
  pptp <hostname> [<pptp options>] [[--] <pppd options>]


Or using pppd's pty option:
  pppd pty "pptp <hostname> --nolaunchpppd <pptp options>"


Available pptp options:
  --phone <number>       Pass <number> to remote host as phone number
  --nolaunchpppd         Do not launch pppd, for use as a pppd pty
  --quirks <quirk>       Work around a buggy PPTP implementation
                         Currently recognised values are BEZEQ_ISRAEL only
  --debug                Run in foreground (for debugging with gdb)
  --sync                 Enable Synchronous HDLC (pppd must use it too)
```

--timeout <secs>	Time to wait for reordered packets (0.01 to 10 secs)
--nobuffer	Disable packet buffering and reordering completely
--idle-wait	Time to wait before sending echo request
--max-echo-wait	Time to wait before giving up on lack of reply
--logstring <name>	Use <name> instead of 'anon' in syslog messages
--localbind <addr>	Bind to specified IP address instead of wildcard
--loglevel <level>	Sets the debugging level (0=low, 1=default, 2=high)

PPTP establishes the client side of a Virtual Private Network (VPN) using the Point-to-Point Tunneling Protocol (PPTP). Use this program to connect to an employer's PPTP based VPN, or to certain cable and ADSL service providers.

Example:

The PPTP Server has the following information:

- The IP address of the server ($SERVER)
- The authentication domain name ($DOMAIN)
- The username you are to use ($USERNAME)
- The password you are to use ($PASSWORD)

In the steps below, substitute these values manually..

Edit the config file /etc/ppp/options.pptp.client, which sets options common to all tunnels:

```
lock
name $USERNAME
password $PASSWORD
require-mppe
```

Add authentication information to /etc/ppp/chap-secrets file:

```
# Secrets for authentication
$USERNAME * $PASSWORD *
```

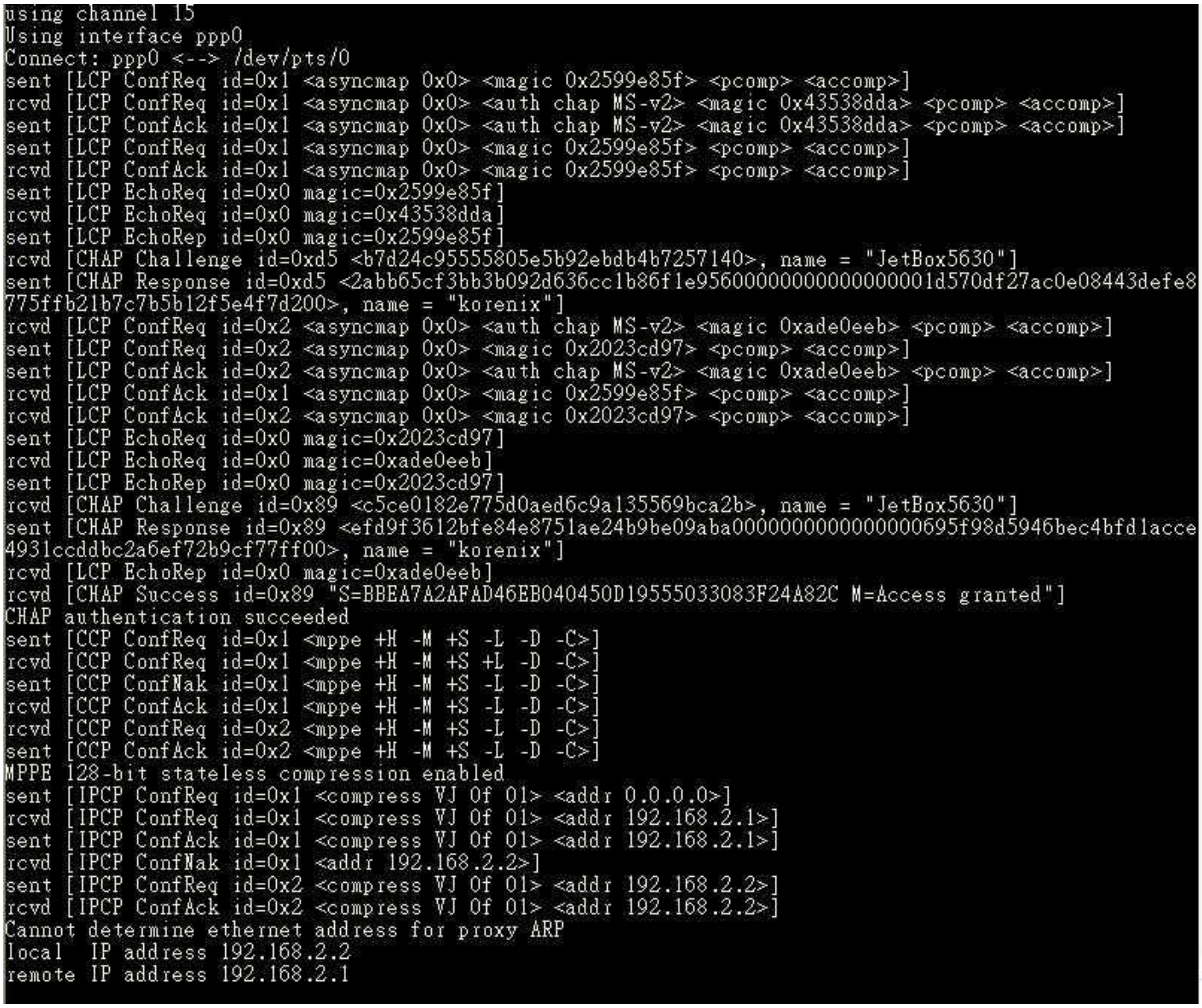
Create a /etc/ppp/peers/$TUNNEL_NAME file.

```
# Secrets for authentication
debug
pty "pptp $SERVER --nolaunchpppd"
file /etc/ppp/options.pptp.client
```

Start the tunnel using the pppd command:

```
pppd call $TUNNEL_NAME logfd 2 nodetach
```

The following picture is the success output with tunnel name “test” and server ip address “192.168.20.100”.



```
using channel 15
Using interface ppp0
Connect: ppp0 <--> /dev/pts/0
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x2599e85f> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <auth chap MS-v2> <magic 0x43538dda> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <auth chap MS-v2> <magic 0x43538dda> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x2599e85f> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x2599e85f> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0x2599e85f]
rcvd [LCP EchoReq id=0x0 magic=0x43538dda]
sent [LCP EchoRep id=0x0 magic=0x2599e85f]
rcvd [CHAP Challenge id=0xd5 <b7d24c95555805e5b92ebdb4b7257140>, name = "JetBox5630"]
sent [CHAP Response id=0xd5 <2abb65cf3bb3b092d636cc1b86f1e95600000000000000001d570df27ac0e08443defe8
775ffb21b7c7b5b12f5e4f7d200>, name = "korenix"]
rcvd [LCP ConfReq id=0x2 <asyncmap 0x0> <auth chap MS-v2> <magic 0xade0eeb> <pcomp> <accomp>]
sent [LCP ConfReq id=0x2 <asyncmap 0x0> <magic 0x2023cd97> <pcomp> <accomp>]
sent [LCP ConfAck id=0x2 <asyncmap 0x0> <auth chap MS-v2> <magic 0xade0eeb> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x2599e85f> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x2 <asyncmap 0x0> <magic 0x2023cd97> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0x2023cd97]
rcvd [LCP EchoReq id=0x0 magic=0xade0eeb]
sent [LCP EchoRep id=0x0 magic=0x2023cd97]
rcvd [CHAP Challenge id=0x89 <c5ce0182e775d0aed6c9a135569bca2b>, name = "JetBox5630"]
sent [CHAP Response id=0x89 <efd9f3612bfe84e8751ae24b9be09aba0000000000000000695f98d5946bec4bfd1acce
4931ccddbc2a6ef72b9cf77ff00>, name = "korenix"]
rcvd [LCP EchoRep id=0x0 magic=0xade0eeb]
rcvd [CHAP Success id=0x89 "S=BBEA7A2AFAD46EB040450D19555033083F24A82C M=Access granted"]
CHAP authentication succeeded
sent [CCP ConfReq id=0x1 <mppe +H -M +S -L -D -C>]
rcvd [CCP ConfReq id=0x1 <mppe +H -M +S +L -D -C>]
sent [CCP ConfNak id=0x1 <mppe +H -M +S -L -D -C>]
rcvd [CCP ConfAck id=0x1 <mppe +H -M +S -L -D -C>]
rcvd [CCP ConfReq id=0x2 <mppe +H -M +S -L -D -C>]
sent [CCP ConfAck id=0x2 <mppe +H -M +S -L -D -C>]
MPPE 128-bit stateless compression enabled
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0>]
rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 192.168.2.1>]
sent [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 192.168.2.1>]
rcvd [IPCP ConfNak id=0x1 <addr 192.168.2.2>]
sent [IPCP ConfReq id=0x2 <compress VJ 0f 01> <addr 192.168.2.2>]
rcvd [IPCP ConfAck id=0x2 <compress VJ 0f 01> <addr 192.168.2.2>]
Cannot determine ethernet address for proxy ARP
local  IP address 192.168.2.2
remote IP address 192.168.2.1
```

NOTE

If you are using a PPTP Server that does not require an authentication domain name, omit the slashes as well as the domain name.

If you need to setup a PPTP server on Windows XP as a test target, please see the following link: http://www.onecomputerguy.com/networking/xp_vpn_server.htm

4-15 PPTP Server

Service name	pptpd
Description	A Point-to-Point potocol server
Config files	/etc/pptpd.conf /etc/ppp/chap-secrets /etc/ppp/options.pptpd
Start file	/etc/init.d/pptpd
Start command	/etc/init.d/pptpd start
Stop command	/etc/init.d/pptpd stop
Default	down

pptpd is the Poptop PPTP daemon, which manages tunneled PPP connections encapsulated in GRE using the PPTP VPN protocol. It may contain features like IP address management and TCP wrappers if compiled in.

Example:

Edit the configuration file: **/etc/pptpd.conf** for IP ranges and option file, for example:

```
option /etc/ppp/options.pptpd
localip 192.168.0.1
remoteip 192.168.0.234-238,192.168.0.245
```

Add user/password lists to **/etc/ppp/chap-secrets**:

```
username * password *
```

Edit the /etc/ppp/options.pptpd

```
auth
require-mppe
require-mschap-v2
```

Start the tunnel using the command:

```
/etc/init.d/pptpd start
```

4-16 L2TP Server

Service name	xl2tpd
Description	L2TP (Layer 2 Tunneling Protocol)
Config files	/etc/xl2tpd/xl2tpd.conf /etc/ppp/chap-secrets /etc/ppp/options.xl2tpd
Start file	/etc/init.d/xl2tpd
Start command	/etc/init.d/xl2tpd start
Stop command	/etc/init.d/ xl2tpd stop
Default	down

L2TP (Layer 2 Tunneling Protocol) is a tunneling protocol used for VPNs. It uses the UDP port 1701 to communicate. It doesn't have any encryption, but we can encrypt the L2TP packets by using it with IPSec. For theoretical information on L2TP you can visit its Wiki

Example:

Edit the configuration file: **/etc/xl2tpd/xl2tpd.conf** for **LNS** section, for example:

```
[lns default]
ip range = 192.168.10.2 - 192.168.10.100
local ip = 192.168.10.1
require chap = yes
require authentication = yes
pppoptfile = /etc/ppp/options.l2tp_server.xl2tpd
length bit = no
ppp debug = yes
```

Option file : **/etc/ppp/options.l2tp_server.xl2tpd**

```
debug
lock
auth
```

Add user/password lists to **/etc/ppp/chap-secrets**:

```
username * password *
```

Start the L2TP Server using the follow command:

```
/etc/init.d/xl2tpd start
```

For more information, please refer to http://linux.die.net/man/5/xl2tpd.conf

4-17 L2TP Client

Service name	xl2tpd
Description	L2TP (Layer 2 Tunneling Protocol)
Config files	/etc/xl2tpd/xl2tpd.conf /etc/ppp/chap-secrets /etc/ppp/options.xl2tpd
Start file	/etc/init.d/xl2tpd
Start command	/etc/init.d/xl2tpd start
Stop command	/etc/init.d/ xl2tpd stop
Default	down

Example:

Edit the configuration file: **/etc/xl2tpd/xl2tpd.conf** for **LAC** section, for example:

```
[lac l2tp_client]
name = korenix
lns = 192.168.10.2
pppoptfile = /etc/ppp/options.xl2tp.l2tp_client
ppp debug = yes
```

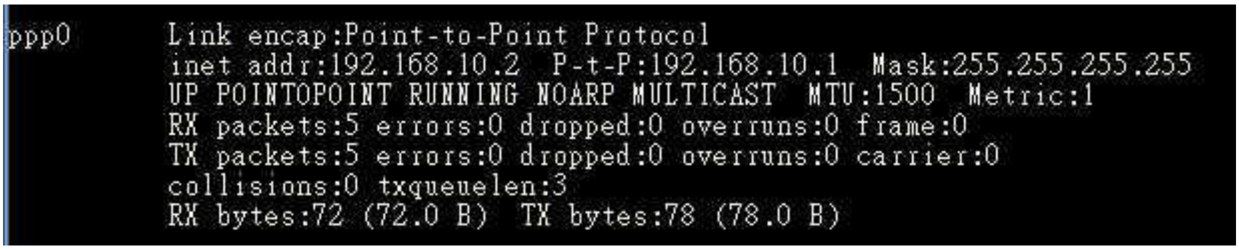
Option file : **/etc/ppp/options.xl2tp.l2tp_client**

```
debug
lock
name korenix
password korenix
```

Start the L2TP Client using the below command:

```
/etc/init.d/xl2tpd start
echo 'c l2tp_client' > /var/run/xl2tpd/l2tp-control
```

You will see the tunnel interface when it creates successfully.



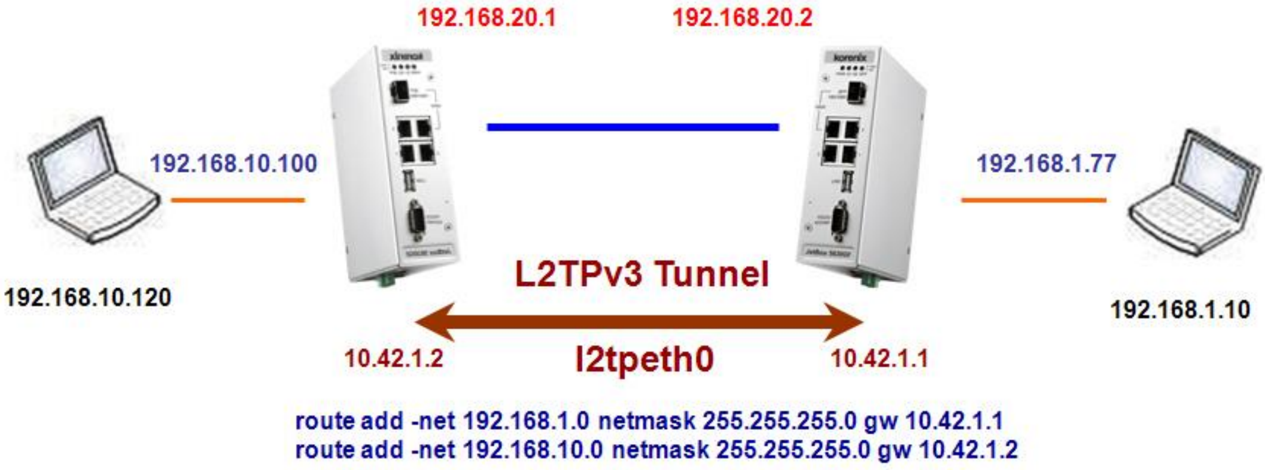
```
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.10.2  P-t-P:192.168.10.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:72 (72.0 B)  TX bytes:78 (78.0 B)
```

Stop the L2TP Client using the below command:

```
/etc/init.d/xl2tpd start
echo 'd l2tp_client' > /var/run/xl2tpd/l2tp-control
```

4-18 L2TPv3

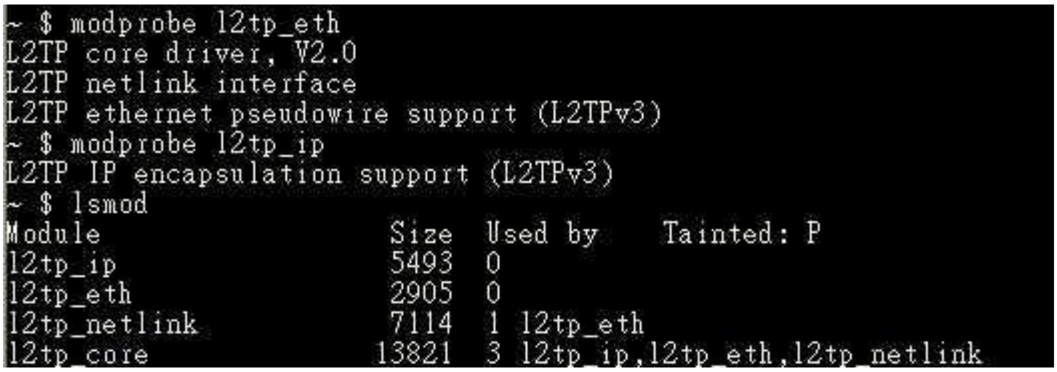
Service name	
Description	L2TP was only ever designed to carry PPP traffic. The new revision of the L2TP protcol (known as L2TPv3) changes the protocol so that it can carry data frame formats other than PPP. Each L2TPv3 session carries one data frame type which is agreed by both peers when the session is established and is effectively a virtual physical wire of that data link type. It is often referred to as a "pseudowire" for that reason. Many L2TP pseudowire types are already defined: PPP, ethernet, VLAN, HDLC, Frame Relay and various ATM flavours.
Default	down


192.168.20.1
192.168.20.2
192.168.10.100
192.168.1.77
192.168.10.120
192.168.1.10
L2TPv3 Tunnel
10.42.1.2
l2tpeth0
10.42.1.1
route add -net 192.168.1.0 netmask 255.255.255.0 gw 10.42.1.1
route add -net 192.168.10.0 netmask 255.255.255.0 gw 10.42.1.2

Example:

Before to create L2TPv3 tunnel, you first load l2tp_eth and l2tp_ip driver

```
~$ modprobe l2tp_eth
~$ modprobe l2tp_ip
```



```
~ $ modprobe l2tp_eth
L2TP core driver, V2.0
L2TP netlink interface
L2TP ethernet pseudowire support (L2TPv3)
~ $ modprobe l2tp_ip
L2TP IP encapsulation support (L2TPv3)
~ $ lsmod
Module                  Size  Used by    Tainted: P
l2tp_ip                 5493  0
l2tp_eth                2905  0
l2tp_netlink            7114  1 l2tp_eth
l2tp_core              13821  3 l2tp_ip,l2tp_eth,l2tp_netlink
```

Using **/sbin/ip** command to create L2TPv3 tunnel

Site A : 10.42.1.1

```
~$ /sbin/ip l2tp add tunnel tunnel_id 3000 peer_tunnel_id 4000 encap udp local
192.168.20.1 remote 192.168.20.2 udp_sport 5000 udp_dport 6000
```

```
~$ /sbin/ip l2tp add session tunnel_id 3000 session_id 1000 peer_session_id 2000
```

```
~$ /sbin/ip link set l2tpeth0 up
```

```
~$ /sbin/ip addr add 10.42.1.1 peer 10.42.1.2 dev l2tpeth0
```

Add route rule

```
~$ route add -net 192.168.10.0 netmask 255.255.255.0 gw 10.42.1.2
```

Site B : 10.42.1.2

```
~$ /sbin/ip l2tp add tunnel tunnel_id 4000 peer_tunnel_id 3000 encap udp local
192.168.20.2 remote 192.168.20.1 udp_sport 6000 udp_dport 5000
```

```
~$ /sbin/ip l2tp add session tunnel_id 4000 session_id 2000 peer_session_id 1000
```

```
~$ /sbin/ip link set l2tpeth0 up
```

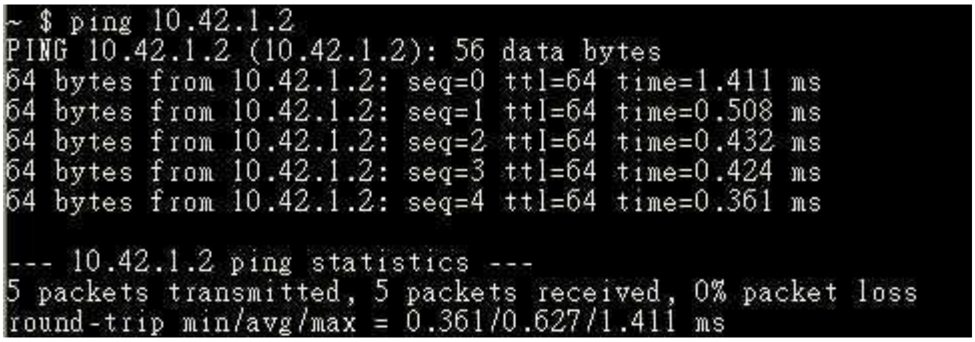
```
~$ /sbin/ip addr add 10.42.1.2 peer 10.42.1.1 dev l2tpeth0
```

Add route rule

```
~$ route add -net 192.168.1.0 netmask 255.255.255.0 gw 10.42.1.1
```

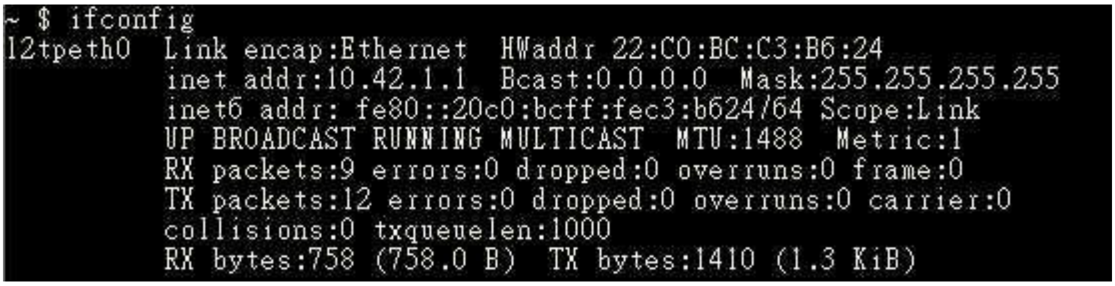
Now the link should be usable. Add static routes as needed to have data sent over the new link.

In Site A, ping 10.42.1.2



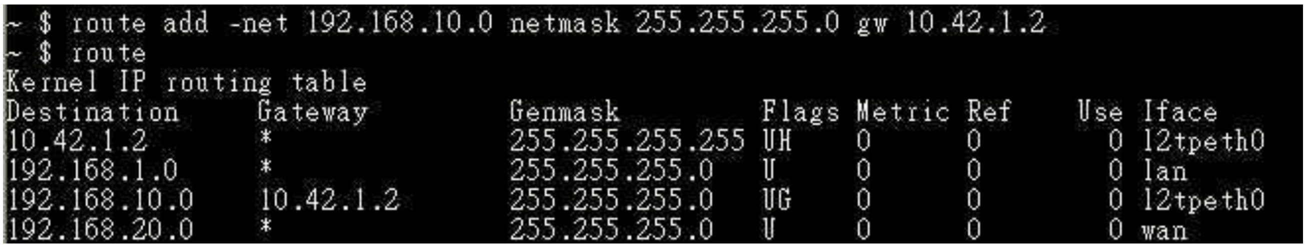
```
~ $ ping 10.42.1.2
PING 10.42.1.2 (10.42.1.2): 56 data bytes
64 bytes from 10.42.1.2: seq=0 ttl=64 time=1.411 ms
64 bytes from 10.42.1.2: seq=1 ttl=64 time=0.508 ms
64 bytes from 10.42.1.2: seq=2 ttl=64 time=0.432 ms
64 bytes from 10.42.1.2: seq=3 ttl=64 time=0.424 ms
64 bytes from 10.42.1.2: seq=4 ttl=64 time=0.361 ms

--- 10.42.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.361/0.627/1.411 ms
```

L2TPv3 tunnel interface



```
~ $ ifconfig
l2tpeth0  Link encap:Ethernet  HWaddr 22:C0:BC:C3:B6:24
          inet addr:10.42.1.1  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: fe80::20c0:bcff:fec3:b624/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1488  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:758 (758.0 B)  TX bytes:1410 (1.3 KiB)
```

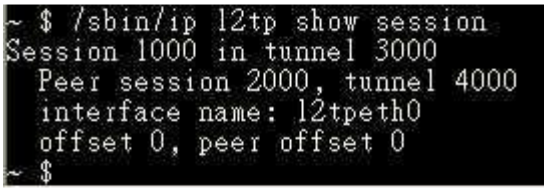
Routing Table



```
~ $ route add -net 192.168.10.0 netmask 255.255.255.0 gw 10.42.1.2
~ $ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.42.1.2       *               255.255.255.255 UH    0      0        0 l2tpeth0
192.168.1.0     *               255.255.255.0   U     0      0        0 lan
192.168.10.0    10.42.1.2       255.255.255.0   UG    0      0        0 l2tpeth0
192.168.20.0    *               255.255.255.0   U     0      0        0 wan
```

Show Tunnel Information

```
~$ /sbin/ip l2tp show session
```



```
~ $ /sbin/ip l2tp show session
Session 1000 in tunnel 3000
  Peer session 2000, tunnel 4000
  interface name: l2tpeth0
  offset 0, peer offset 0
~ $
```

Delete Tunnel

```
~$ /sbin/ip l2tp del tunnel tunnel_id 3000
~$ /sbin/ip l2tp del session tunnel_id 3000 session_id 1000
```

4-19 Routing -- Zebra Daemon

Service name	zebra
Description	Zebra is an advanced routing software package that provides TCP/IP based routing protocols.
Config files	/etc/zebra.conf
Start file	/etc/init.d/zebra
Start command	/etc/init.d/zebra start
Stop command	/etc/init.d/zebra stop
Default	down

There are four routing daemons in use, and there is one manager daemon.

- ospfd, ripd, bgpd
- Zebra

Configuration options:

- Each of the daemons has its own config file. For example, zebra's default config file name is /etc/zebra.conf.
- The daemon name plus .conf is the default config file name. You can specify other config file using the -f options when starting the daemon.
- Check the log files for proper operation. For example, you can type less -f /var/log/zebra.log to check the zebra log. Keep in mind that you have to add the log-file location to the respective daemon configuration file.

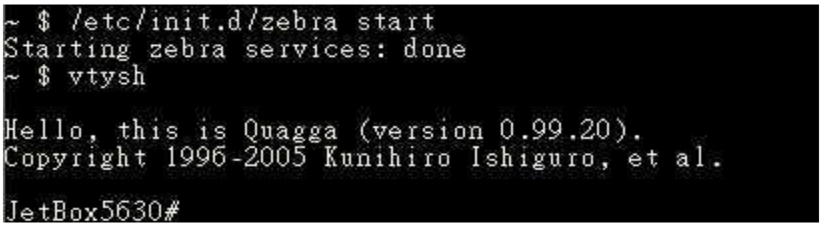
The administrator has two options to modify runtime configurations via the command-line interface (CLI):

1. (**Preferred**) Use the integrated Zebra shell **vtysh** by typing vtysh. vtysh expects its configuration to reside in /etc/vtysh.conf.
2. Telnet localhost <port> ,e.g. port 2601 connects to the ospfd. Zebra uses ports from 2600 to 2607 for daemon connections.

Service	zebra	ripd	ospfd	bgpd
Port	2601	2602	2604	2605

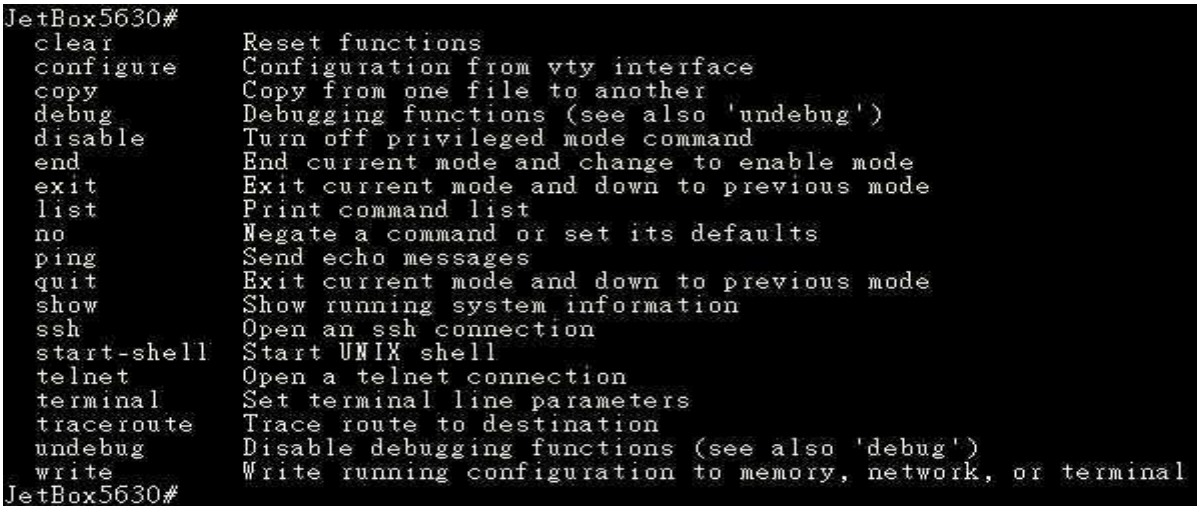
Example:

1. Start zebra: /etc/init.d/zebra start
2. Type vtysh to enter Zebra shell



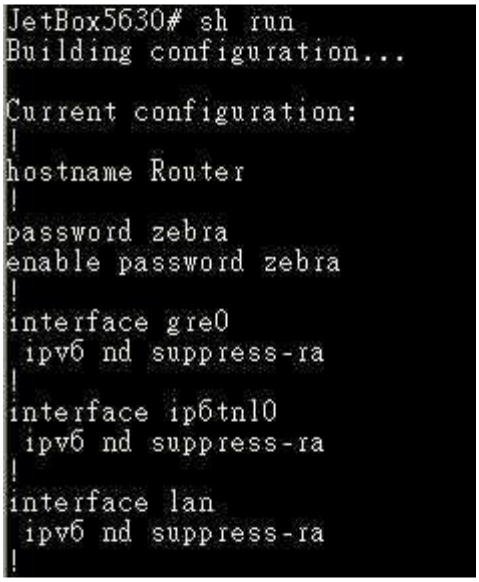
```
~ $ /etc/init.d/zebra start
Starting zebra services: done
~ $ vtysh

Hello, this is Quagga (version 0.99.20).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

JetBox5630#
```

3. Query for supported command with '?'



```
JetBox5630#
  clear         Reset functions
  configure     Configuration from vty interface
  copy          Copy from one file to another
  debug         Debugging functions (see also 'undebug')
  disable       Turn off privileged mode command
  end           End current mode and change to enable mode
  exit          Exit current mode and down to previous mode
  list          Print command list
  no            Negate a command or set its defaults
  ping          Send echo messages
  quit          Exit current mode and down to previous mode
  show          Show running system information
  ssh           Open an ssh connection
  start-shell   Start UNIX shell
  telnet        Open a telnet connection
  terminal      Set terminal line parameters
  traceroute    Trace route to destination
  undebug       Disable debugging functions (see also 'debug')
  write         Write running configuration to memory, network, or terminal
JetBox5630#
```

4. Display the current running config: # sh run



```
JetBox5630# sh run
Building configuration...

Current configuration:
!
hostname Router
!
password zebra
enable password zebra
!
interface gre0
 ipv6 nd suppress-ra
!
interface ip6tnl0
 ipv6 nd suppress-ra
!
interface lan
 ipv6 nd suppress-ra
!
```

For all Zebra suppored command, please refer:
http://www.zebra.org/zebra/Command-Index.html#Command%20Index

4-20 Ser2net

program name	ser2net
Description	ser2net is a Linux program which will connect a network to the serial port. It could be like a bridge between the ethernet cable and the serial cable.
config file	/etc/ser2net.conf
Usage: -c <config file> - use a config file besides /etc/ser2net.conf -C <config line> - Handle a single configuration line. This may be specified multiple times for multiple lines. This is just like a line in the config file. This disables the default config file, you must specify a -c after the last -C to have it read a config file, too. -p <controller port> - Start a controller session on the given TCP port -P <file> - set location of pid file -n - Don't detach from the controlling terminal -d - Don't detach and send debug I/O to standard output -u - Disable UUCP locking -b - Do CISCO IOS baud-rate negotiation, instead of RFC2217 -v - print the program's version and exit	

Example:

- **Setup a TCP server with following operation parameter:**

  Serial port : 1
  TCP port : 62001
  Baud rate : 9600
  Data bits : 8
  Parity : none
  Stop bit : 1
  Hardware flow control : none
  State : raw state
  timeout : never timeout
  modem mode : none

Edit /etc/ser2net.conf add the following line:

```
62001:raw:0:/dev/ttyS1:9600 NONE 1STOPBIT 8DATABITS LOCAL -RTSCTS
```

Then run the ser2net program:

```
ser2net &
```

For more information, please see section 7-5 .

4-21 WiFi Configuration

In JetBox5630, we have built-in AWUS036NEH wireless driver. You can easily install and use it to connect Ethernet.

4-21-1 Introduction

AWUS036NEH IEEE 802.11b/g/n Wireless USB adapter provides users to launch IEEE 802.11b/g/n wireless network at 150 Mbps in the 2.4GHz band, which is also compatible with IEEE 802.11b/g wireless devices at 54 Mbps. You can configure AWUS036NEH with ad-hoc mode to connect to other 2.4GHz wireless computers, or with Infrastructure mode to connect to a wireless AP or router for accessing to Internet. AWUS036NEH includes a convenient Utility for scanning available networks and saving preferred networks that users usually connected with. Security encryption can also be configured by this utility. AWUS036NEH includes a convenient Utility for scanning available networks and saving preferred networks that users usually connected with. Security encryption can also be configured by this utility.

	description
interface	ra0
Driver Name	rt5370sta
Driver file	rt5370sta.ko
Config files	/etc/Wireless/RT2870STA/RT2870STA.dat
Default	Load driver on boot up

4-21-2 Configure with iwpriv

Usage 1: iwpriv ra0 set [parameters]=[val]

Note: Execute one iwpriv/set command simultaneously.

1. Config STA link with AP which is OPEN/NONE(Authentication/Encryption)

```
# iwpriv ra0 set NetworkType=Infra
# iwpriv ra0 set AuthMode=OPEN
# iwpriv ra0 set EncrypType=NONE
# iwpriv ra0 set SSID="AP's SSID"
```

2. Config STA to link with AP and OPEN/WEP(Authentication/Encryption)

```
Default Key ID = 1
# iwpriv ra0 set NetworkType=Infra
# iwpriv ra0 set AuthMode=OPEN
# iwpriv ra0 set EncrypType=WEP
# iwpriv ra0 set Key1="AP's wep key"
# iwpriv ra0 set DefaultKeyID=1
# iwpriv ra0 set SSID="AP's SSID"
```

3. Config STA to link with AP which is SHARED/WEP(Authentication/Encryption)

```
# iwpriv ra0 set NetworkType=Infra
# iwpriv ra0 set AuthMode=SHARED
# iwpriv ra0 set EncrypType=WEP
# iwpriv ra0 set Key1="AP's wep key"
# iwpriv ra0 set DefaultKeyID=1
# iwpriv ra0 set SSID="AP's SSID"
```

4. Config STA to create/link as Adhoc mode, which is OPEN/NONE(Authentication/Encryption)

```
# iwpriv ra0 set NetworkType=Adhoc
# iwpriv ra0 set AuthMode=OPEN
# iwpriv ra0 set EncrypType=NONE
# iwpriv ra0 set SSID="Adhoc's SSID"
```

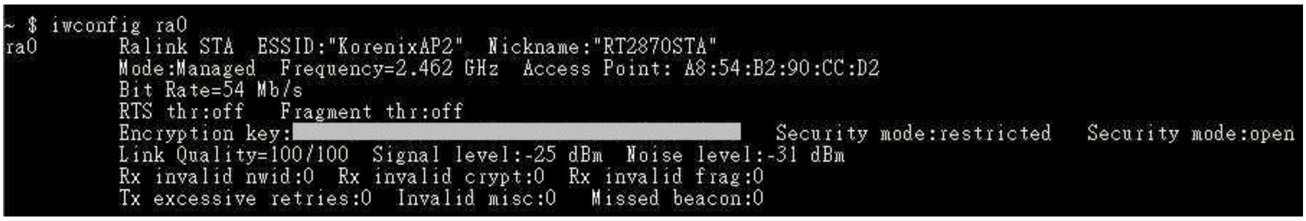
5. Turn off the wireless interface

```
# ifconfig ra0 down
```

The necessary driver should be automatically loaded. If necessary, it can be manually loaded via:

```
# insmod /lib/modules/3.2.0/kernel/drivers/net/rt5370sta.ko
```

6. Check the wireless interface with “iwconfig ra0”



```
~ $ iwconfig ra0
ra0       Ralink STA  ESSID:"KorenixAP2"  Nickname:"RT2870STA"
          Mode:Managed  Frequency=2.462 GHz  Access Point: A8:54:B2:90:CC:D2
          Bit Rate=54 Mb/s
          RTS thr:off   Fragment thr:off
          Encryption key:[redacted]   Security mode:restricted   Security mode:open
          Link Quality=100/100  Signal level:-25 dBm  Noise level:-31 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

4-21-3 Configure with wpa_supplicant

wpa_supplicant is the IEEE 802.1X/WPA component that is used in the client stations. It implements key negotiation with a WPA Authenticator and it controls the roaming and IEEE 802.11 authentication/association of the wireless driver.

In JetBox5630, if you want to configure wireless with WPA encryption. We recommend youuse wpa_supplicant to set up.

Example :

We want to connect to a AP with WPA and TKIP encryption. First we need to edit config file /etc/wpa_supplicant_ra0.conf

```
ctrl_interface=/var/run/wpa_supplicant
network={
        ssid="KorenixAP2"
        key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
        pairwise=TKIP
        group=CCMP TKIP WEP104 WEP40
        psk="1234567890"
}
```

Using wpa_supplicant command to connect.

```
# wpa_supplicant -B -ira0 -c /etc/wpa_supplicant.conf
```

For more information, please refer to

http://linux.die.net/man/8/wpa_supplicant

http://www.freebsd.org/cgi/man.cgi?query=wpa_supplicant.conf&sektion=5

4-22 wvdial

Connecting to the internet via wvdial

wvdial is a Point-to-Point Protocol dialer: it dials a modem and starts pppd in order to connect to the Internet. When wvdial starts, it first loads its configuration from /etc/wvdial.conf.

wvdial.conf

The configuration file /etc/wvdial.conf is the equivalent to the Windows "ini" file format, with sections named in square brackets and a number of variable = value pairs within each section. Here is a sample configuration file.

```
# /etc/wvdial.conf - wvdial configuration file

[Dialer Defaults]
Phone = *99#
Stupid Mode = 1
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP","internet"
Modem Type = Analog Modem
Baud = 115200
New PPPD = yes
Modem = /dev/ttyUSB3
ISDN = 0
Dial Command = ATDT
Username = username
Password = password
```

This example uses Init3 to setup **APN** as *internet*. Changes with your apn and don't remove double quote. For example, in Taiwan, we use internet as APN.

When wvdial is in **Stupid Mode**, it does not attempt to interpret any prompts from the terminal server. It starts pppd immediately after the modem connects. Apparently there are ISP's that actually give you a login prompt, but work only if you start PPP, rather than logging in.

Phone: customize to your country or provider for internet connection.
i.e.: in Taiwan, we use ***99#**

Username, **Password**: change with your username and password if needed, and set **Stupid Mode** to 0. Some providers don’t use username and password; you can just leave them blank.

Modem: The location of the device that **wvdial** should use as your modem.
i.e.: for Sierra MC8092, it uses /dev/ttyUSB3

Connecting to the internet for the first time

First, Enter wvdial at the root prompt to connect:

```
# wvdial
```

wvdial initializes the modem and connects to the ISP's server.

```
# wvdial
--> WvDial: Internet dialer version 1.61
--> Cannot get information for serial port.
--> Initializing modem.
--> Sending: ATZ
OK
--> Sending: ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
OK
--> Sending: AT+CGDCONT=1,"IP","internet"
OK
--> Modem initialized.
--> Sending: ATDT*99#
--> Waiting for carrier.
CONNECT 7200000
--> Carrier detected.  Starting PPP immediately.
```

wvdial starts the pppd daemon.

```
--> Starting pppd at Tue Jun  8 23:47:02 2010
--> Pid of pppd: 1235
--> Using interface ppp0
--> local  IP address 116.59.241.151
--> remote IP address 10.64.64.64
--> primary  DNS address 213.229.248.161
--> secondary DNS address 193.189.160.11
```

Testing the connection

Use Ping to test the connection by querying the ISP's nameservers with a domain name eg. www.google.com. Open a terminal or terminal window and enter the ping command. You should see replies like these.

```
# ping www.google.com
PING www.google.com (64.233.181.104): 56 data bytes
64 bytes from 64.233.181.104: seq=0 ttl=45 time=359.476 ms
64 bytes from 64.233.181.104: seq=1 ttl=45 time=356.268 ms
```

```
64 bytes from 64.233.181.104: seq=2 ttl=45 time=346.154 ms

64 bytes from 64.233.181.104: seq=3 ttl=45 time=336.108 ms


--- www.google.com ping statistics ---

4 packets transmitted, 4 packets received, 0% packet loss

round-trip min/avg/max = 336.108/349.501/359.476 ms
```

Hanging UP

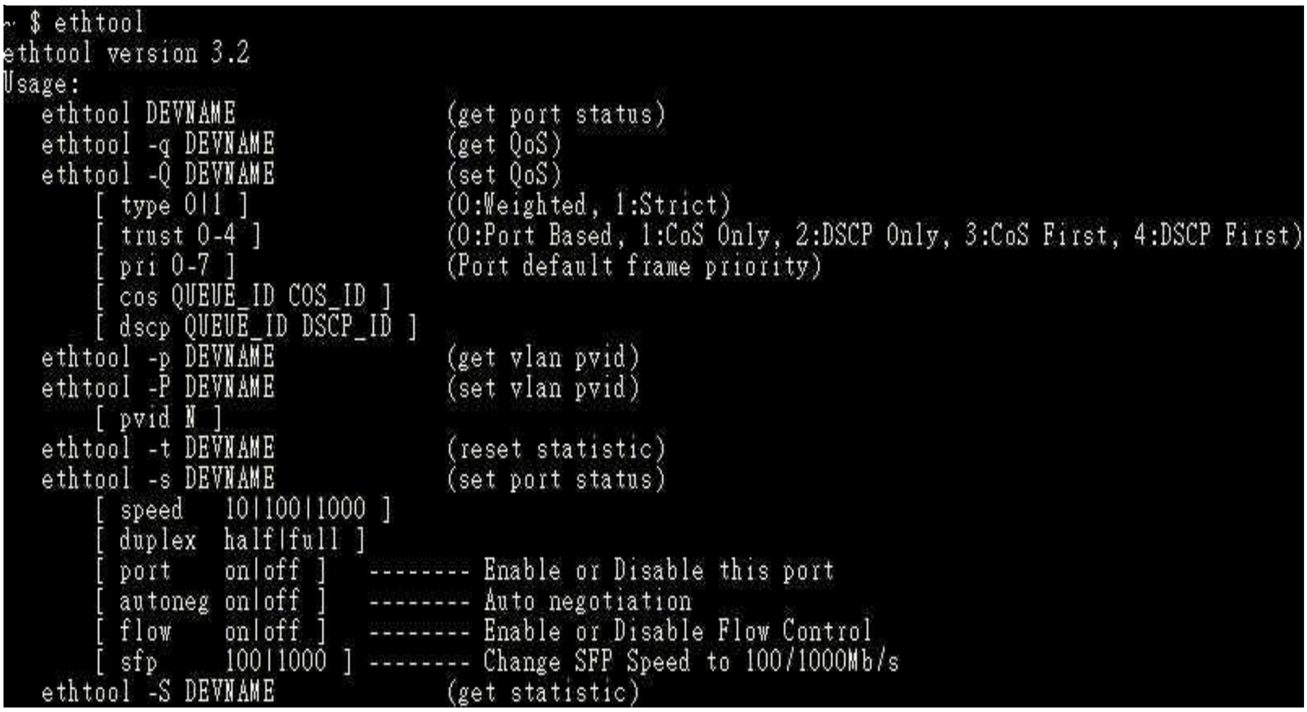
Hang up with ctrl-C (hold down the Control key and press the C key).

```
Caught signal 2:  Attempting to exit gracefully...

--> Terminating on signal 15

--> Connect time 108.3 minutes.

--> Disconnecting at Wed Jun  9 01:35:23 2010
```

Chapter 5 Korenix Feature

5-1 Ethtool

LAN switch port configuration tool



```
~ $ ethtool
ethtool version 3.2
Usage:
   ethtool DEVNAME                (get port status)
   ethtool -q DEVNAME             (get QoS)
   ethtool -Q DEVNAME             (set QoS)
       [ type 0|1 ]               (0:Weighted, 1:Strict)
       [ trust 0-4 ]              (0:Port Based, 1:CoS Only, 2:DSCP Only, 3:CoS First, 4:DSCP First)
       [ pri 0-7 ]                (Port default frame priority)
       [ cos QUEUE_ID COS_ID ]
       [ dscp QUEUE_ID DSCP_ID ]
   ethtool -p DEVNAME             (get vlan pvid)
   ethtool -P DEVNAME             (set vlan pvid)
       [ pvid N ]
   ethtool -t DEVNAME             (reset statistic)
   ethtool -s DEVNAME             (set port status)
       [ speed   10|100|1000 ]
       [ duplex  half|full ]
       [ port    on|off ]   -------- Enable or Disable this port
       [ autoneg on|off ]   -------- Auto negotiation
       [ flow    on|off ]   -------- Enable or Disable Flow Control
       [ sfp     100|1000 ] -------- Change SFP Speed to 100/1000Mb/s
   ethtool -S DEVNAME             (get statistic)
```

Example :

Get port status

```
# ethtool lan:2
Port Status:
       Medium: Copper
       Speed: 100Mb/s
       Duplex: Full
       Flow Control: off
Port Setting:
       Port: on
       Auto-negotiation: on
       Link detected: Up
```

Set port's flow control on

```
# ethtool -s lan:2 flow on
Port Status:
       Medium: Copper
       Speed: 100Mb/s
       Duplex: Full
       Flow Control: on
Port Setting:
       Port: on
       Auto-negotiation: on
       Link detected: Up
```

Set port's auto-negotiation off

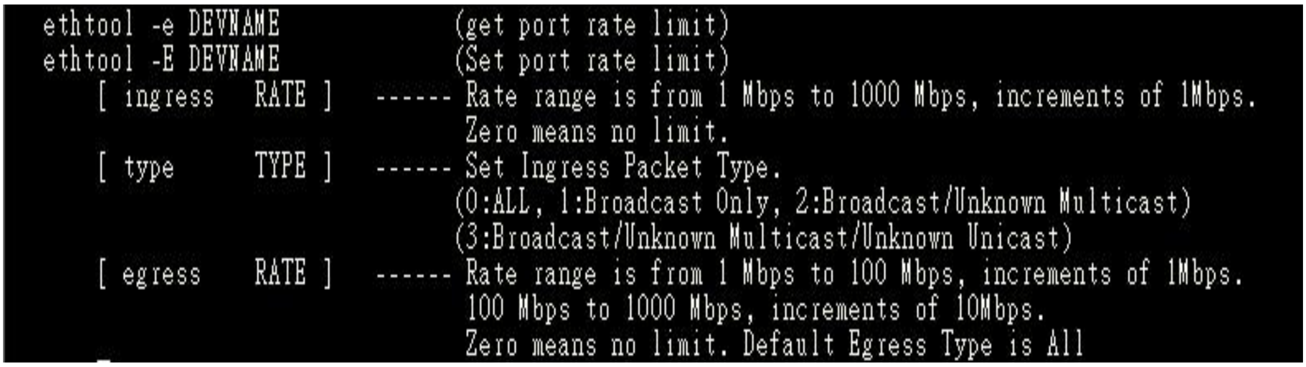
```
# ethtool -s lan:2 autoneg off
Port Status:
       Medium: Copper
       Speed: 100Mb/s
       Duplex: Full
       Flow Control: off
Port Setting:
       Port: on
       Auto-negotiation: off
       Link detected: Up
```

Set port's pvid to 2

```
# ethtool -P lan:2 pvid 2
```

5-2 Rate Limit Control

Rate limiting is used to control the rate of traffic that is sent or received on a network interface. For ingress rate limiting, traffic that is less than or equal to the specified rate is received, whereas traffic that exceeds the rate is dropped. For egress rate limiting, traffic that is less than or equal to the specified rate is sent, whereas traffic that exceeds the rate is dropped
You can program separate transmit (Egress Rule) and receive (Ingress Rule) rate limits at each port by ethtool.



```
ethtool -e DEVNAME                (get port rate limit)
ethtool -E DEVNAME                (Set port rate limit)
    [ ingress   RATE ]   ------ Rate range is from 1 Mbps to 1000 Mbps, increments of 1Mbps.
                                 Zero means no limit.
    [ type      TYPE ]   ------ Set Ingress Packet Type.
                                (0:ALL, 1:Broadcast Only, 2:Broadcast/Unknown Multicast)
                                (3:Broadcast/Unknown Multicast/Unknown Unicast)
    [ egress    RATE ]   ------ Rate range is from 1 Mbps to 100 Mbps, increments of 1Mbps.
                                 100 Mbps to 1000 Mbps, increments of 10Mbps.
                                 Zero means no limit. Default Egress Type is All
```

Example :

Set port 1 ingress rate is 20Mbps and ingress type is Broadcast/Unknown Multicast.

```
# ethtool -E lan:1 ingress 20 type 2
# ethtool -e lan:1
Rate Limit Status:
        Ingress Rate: 20 Mbps
        Ingress Type: Broadcast/Unknown Multicast
        Egress Rate: 0 Mbps
```

Packet type : The packet types of the Ingress Rule listed here include Broadcast Only、Broadcast/ Unknown Multicast、Broadcast/Unknown Multicast/Unknown Unicast or All. The packet types of the Egress Rule (outgoing) only support all packet types.
Ingress Rate : Valid values are from 1Mbps-1000Mbps. The step of the rate is 1 Mbps. Default value of Ingress Rule is “**10**” Mbps.
Egress Rate : Valid values are from 1Mbps-100Mbps. The step of the rate is 1 Mbps. And 100Mbps-1000Mbps. The step of the rate is 10 Mbps. Default value of Egress Rule is **0** Mbps. 0 stands for disabling the rate control for the port.

5-3 SFP and Copper Combo

In JetBox5630, wan port is a RJ-45/SFP combo port. It support 10/100/1000 Base-TX and 100 Base-FX /1000 Base-SX. When you use SFP interface, you can change SFP speed to 100/1000 by using ethtool.

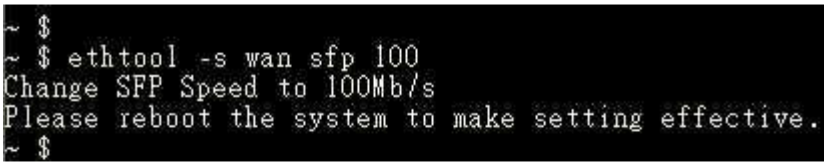
For example :

Default SFP Speed is 1000

```
# ethtool wan
Port Status:
       Medium: SFP
       Speed: 1000Mb/s
       Duplex: Full
       Flow Control: off
Port Setting:
       Port: on
       Auto-negotiation: on
       Link detected: Up
```

If you want to change SFP speed to 100, use ethtool to change speed.

```
# ethtool -s wan sfp 100
```



```
~ $
~ $ ethtool -s wan sfp 100
Change SFP Speed to 100Mb/s
Please reboot the system to make setting effective.
~ $
```

NOTE

1. To change SFP speed you need to reboot the system to make it effective.
2. Please make sure the spec of SFP matching with the SFP speed setting, or exception conditions would happen.

5-4 Vconfig

program name	vconfig
Description	Create and remove VLAN devices on LAN port

Usage:

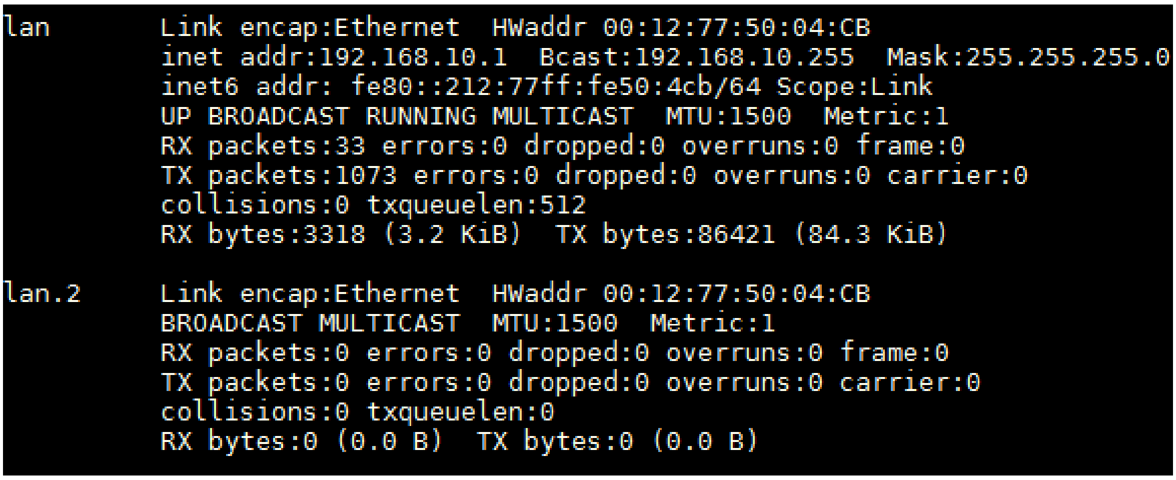
Options:

	add	[interface-name] [vlan_id] tag [port_id] untag [port_id]
		[example : vconfig add lan 2 tag 1 untag 2,3]
	rem	[vlan-name]
	show	[Display Vlan Table]

Example:

- **Add a vlan interface**

# vconfig add lan 2 untag 2,3



```
lan       Link encap:Ethernet  HWaddr 00:12:77:50:04:CB
          inet addr:192.168.10.1  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::212:77ff:fe50:4cb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:33 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1073 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:512
          RX bytes:3318 (3.2 KiB)  TX bytes:86421 (84.3 KiB)

lan.2     Link encap:Ethernet  HWaddr 00:12:77:50:04:CB
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- **Delete a vlan interface**

# vconfig rem lan.2

- **Show all vlan interface**

```
# vconfig show
//------------------------------------------//
VLAN ID       : 2
Tag Port      :
Un-Tag Port   : 2,3
```

NOTE

When you add a VLAN interface, you can only see it by typing “ifconfig -a”. It will display with lan.2”.

To enable it, issue “ifconfig lan.2 up

5-5 LED Set

program name	let_set
Description	Control LED status

Usage:

```
led_set -g LED_NUM - Set LED to Green
led_set -y LED_NUM - Set LED to Yellow
led_set -d LED_NUM - Set LED to Off
```

Example:

- **Set LED 1 to green**

```
# led_set –g 1
```

- **Set LED 1 to yellow**

```
# led_set –y 1
```

- **Turn off LED 1**

```
# led_set -d 1
```

5-6 Serial Control

JetBox5630 have on serial port that support RS232/RS422/RS485 4Wire.

The device name is **/dev/ttyO2**.

program name	serialctl
Description	Get/Set Serial Port Configuration

Usage:

```
serialctl -g     - Get Serial Port Mode
serialctl -m 1 - RS232 Mode
serialctl -m 2 - RS422 Mode
serialctl -m 3 - RS485 4 Wire Mode
```

Example:

- **Set serial port mode to RS232**

```
# serialctl -m 1
```

- **Get serial port status**

```
# serialctl -g
Serial mode : RS232
```

- **Using microcom to test (Remember to plug-in RS232 lookback)**

```
# microcom –D/dev/ttyO2
test…test..
```

5-7 Multiple Super Ring

Korenix is proud to announce that it has launched its patented Rapid Super Ring (RSR) network redundancy technology in its JetBox 5630 / 5633 series industrial embedded Routing Computers for ensuring reliability, scalability and high performance of industrial network infrastructures.

The RSR provides less than 5mllisecond recovery time and ZERO ms restoration time, allowing users to perform reliable data transmission and computing without link loss, topology change or data failure.

With the new RSR feature, IPC providers can easily setup the industrial network with automatic Ring Master selection, efficiently control the ring status with minimum bandwidth consumption as well as detect and fast react to the failures through received notifications and alarms. The RSR is backward compatible with Super ring technology and therefore can be used in a large network along with other redundant rings providing a complete reliable networking solution.

Korenix JetBox 5630 / 5633 series are embedded Linux Ready VPN systems designed with a rich interface, including LAN / WAN, USB, Serial and console ports to deliver maximum flexibility to IPC providers. The devices support complete Layer 3 routing capabilities for efficiently managing extended network groups in industrial environments. Featuring VPN and the latest DMVPN functionalities, they can be perfectly used by IPC providers for establishing dynamic, long-distance and secure overlay networks.

MSR compatible models

The MSR is compatible with other Korenix models listed as follows:

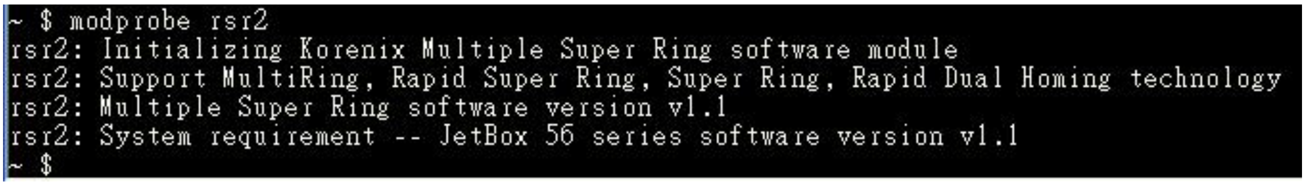
JN5628G, JN5010G, JN5008G-P, JN4510, JN4010
JN4706, JN4706f, JN3706, JN3706f
JN4506-RJ, JN4506-M12
(Firmware v2.0 or above)

Kindly visit Korenix website or contact Korenix sales for latest MSR compatible model list.

5-7-1 Setting from console

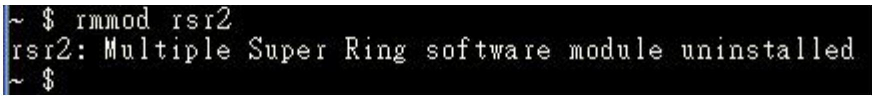
1. Multiple Super Ring module Enable/Disable

# **modprobe rsr2** - Enable Multiple Super Ring module



```
~ $ modprobe rsr2
rsr2: Initializing Korenix Multiple Super Ring software module
rsr2: Support MultiRing, Rapid Super Ring, Super Ring, Rapid Dual Homing technology
rsr2: Multiple Super Ring software version v1.1
rsr2: System requirement -- JetBox 56 series software version v1.1
~ $
```

# **rmmod rsr2** - Disable Multiple Super Ring module



```
~ $ rmmod rsr2
rsr2: Multiple Super Ring software module uninstalled
~ $
```

2. Create/Delete a ring

# **rsr2ctl create RINGID** - Create a ring with a ring ID (RINGID: 0-31)

# **rsr2ctl delete RINGID** - Delete a ring with a ring ID (RINGID: 0-31)

3. Start/Stop a ring

# **rsr2ctl ring RINGID start** - Start a ring

# **rsr2ctl ring RINGID stop** - Stop a ring

4. Change ring name

# **rsr2ctl ring RINGID name NAME** - The default ring name is “Ring RINGID”

5. Change ring priority

# **rsr2ctl ring RINGID priority PRIORITY -** Default priority is 128

# **rsr2ctl ring RINGID priority default** - Change ring priority to default priority (128)

6. Change ring port and ring port cost

# **rsr2ctl ring RINGID port PORTID1 PORTID2** - Change ring port to PORTID1 PORTID2 (PORTID: 1-3), the default PORTID1 PORTID2 are 1 and 2

# **rsr2ctl ring RINGID cost PORTCOST1 PORTCOST2** - Change ring port cost to PORTCOST1 PORTCOST2 (PORTCOST: 0-255), the default PORTCOST is 128

# **rsr2ctl ring RINGID cost default default** - Change ring port cost default port cost (128)

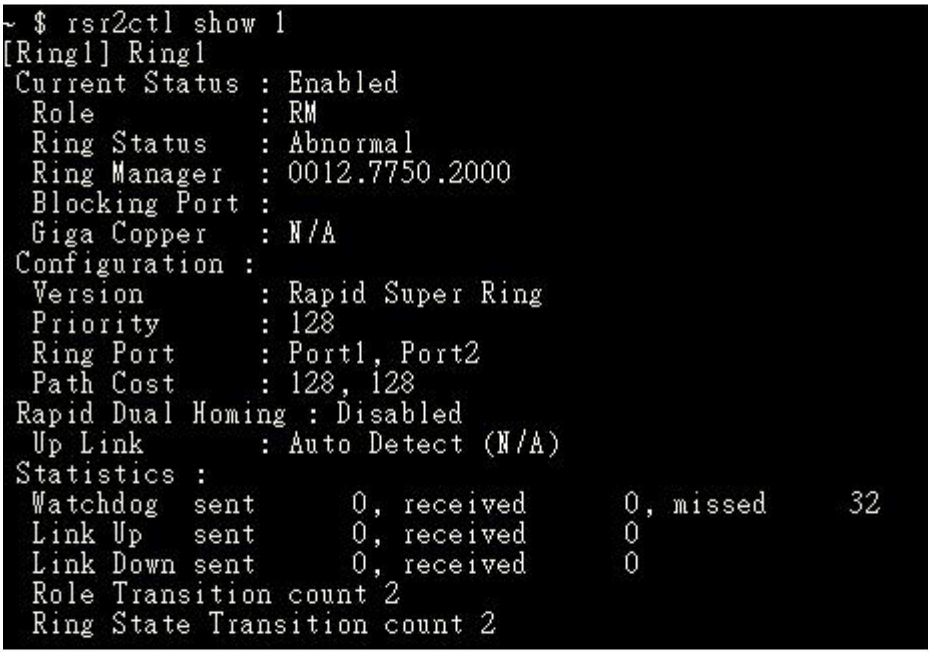
7. Enable/Disable Rapid Dual-Homing feature

# **rsr2ctl ring RINGID rdh enable** - Enable Rapid Dual-Homing feature

# **rsr2ctl ring RINGID rdh disable** - Disable Rapid Dual-Homing feature

8. Show a ring or all rings information

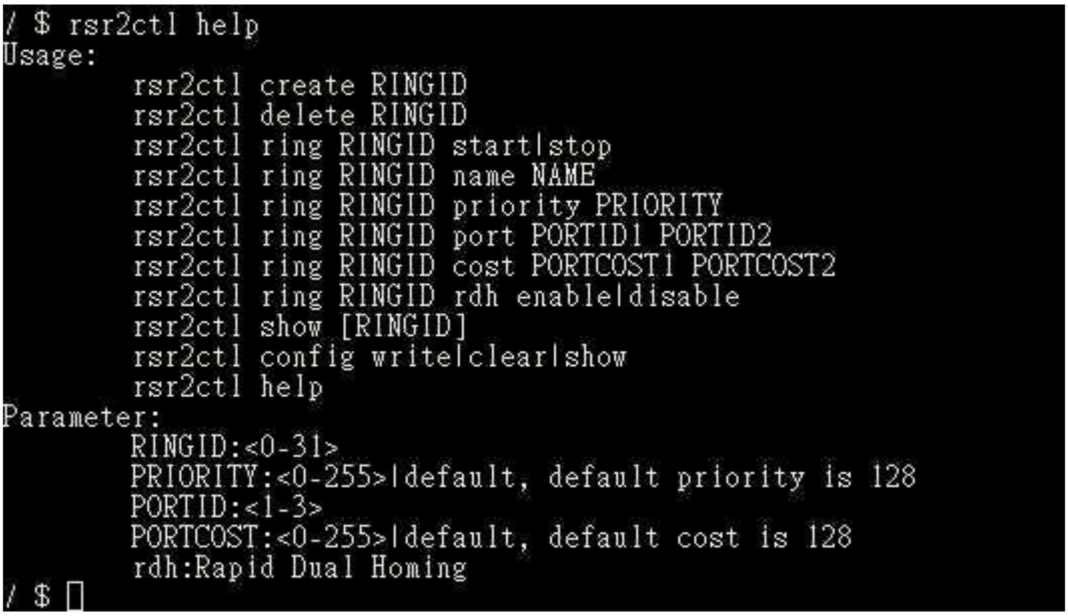
# **rsr2ctl show RINGID** - Show ring information



```
~ $ rsr2ctl show 1
[Ring1] Ring1
 Current Status : Enabled
  Role          : RM
  Ring Status   : Abnormal
  Ring Manager  : 0012.7750.2000
  Blocking Port :
  Giga Copper   : N/A
 Configuration :
  Version       : Rapid Super Ring
  Priority      : 128
  Ring Port     : Port1, Port2
  Path Cost     : 128, 128
 Rapid Dual Homing : Disabled
  Up Link       : Auto Detect (N/A)
 Statistics :
  Watchdog  sent       0, received       0, missed     32
  Link Up   sent       0, received       0
  Link Down sent       0, received       0
  Role Transition count 2
  Ring State Transition count 2
```

9. Show rsr2ctl help

# **rsr2ctl help** - Show rsr2ctl command help



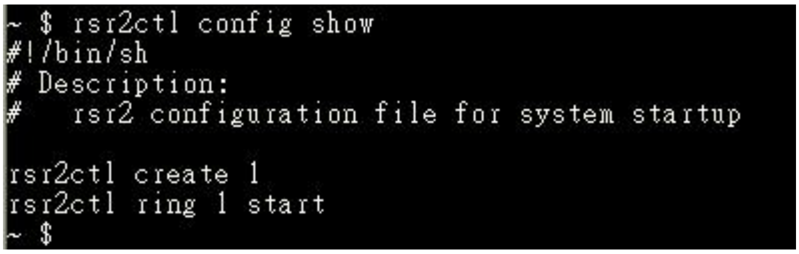
```
/ $ rsr2ctl help
Usage:
        rsr2ctl create RINGID
        rsr2ctl delete RINGID
        rsr2ctl ring RINGID start|stop
        rsr2ctl ring RINGID name NAME
        rsr2ctl ring RINGID priority PRIORITY
        rsr2ctl ring RINGID port PORTID1 PORTID2
        rsr2ctl ring RINGID cost PORTCOST1 PORTCOST2
        rsr2ctl ring RINGID rdh enable|disable
        rsr2ctl show [RINGID]
        rsr2ctl config write|clear|show
        rsr2ctl help
Parameter:
        RINGID:<0-31>
        PRIORITY:<0-255>|default, default priority is 128
        PORTID:<1-3>
        PORTCOST:<0-255>|default, default cost is 128
        rdh:Rapid Dual Homing
/ $ 
```

10. Ring configuration maintain

# rsr2ctl config write - Write ring running configuration to device flash as boot-up configuration. Boot-up configuration file is /etc/rsr2.conf

# **rsr2ctl config clear** - Clear ring boot-up configuration and save to device flash

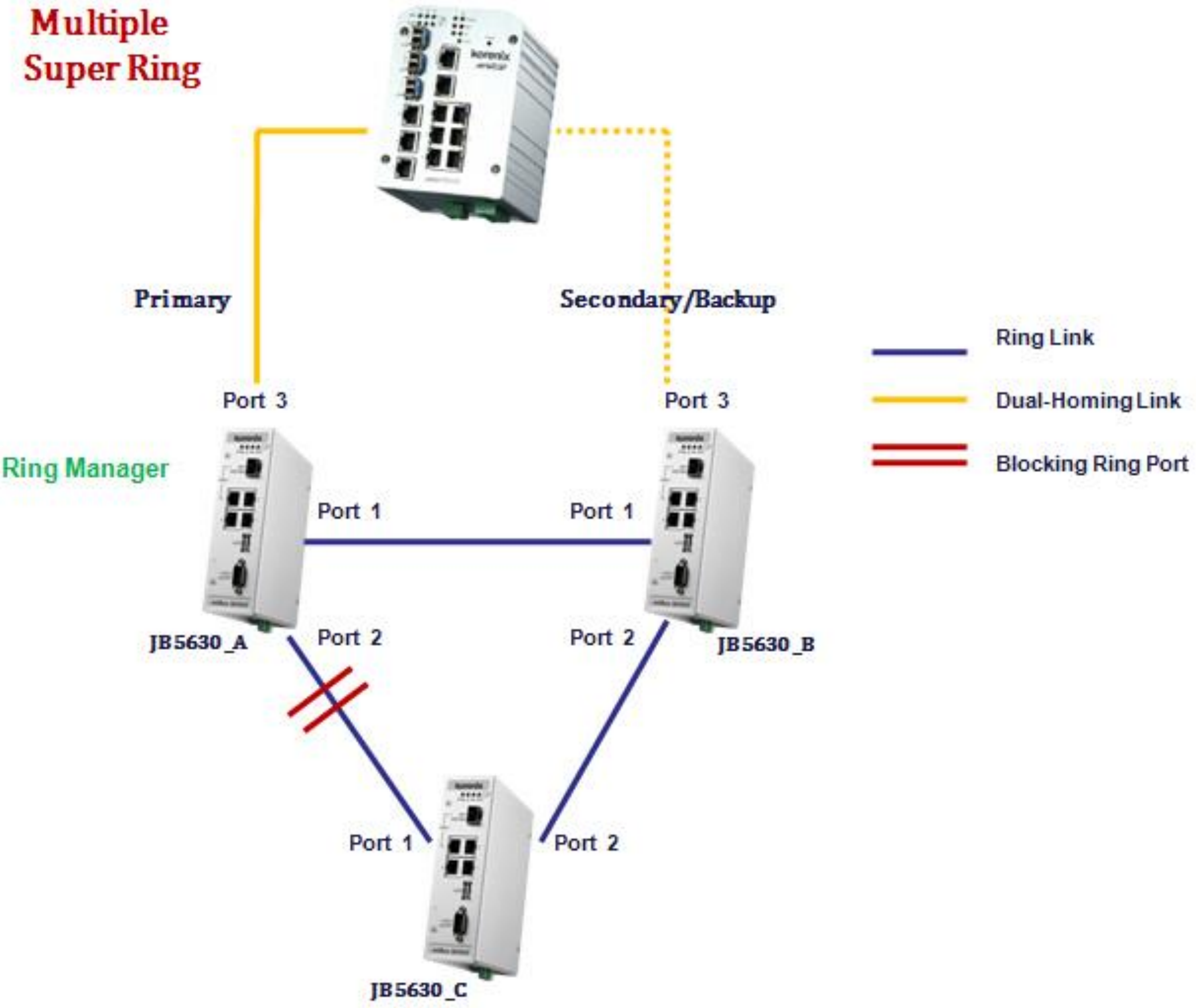
# **rsr2ctl config show** - Show the boot-up configuration in device flash



```
~ $ rsr2ctl config show
#!/bin/sh
# Description:
#   rsr2 configuration file for system startup

rsr2ctl create 1
rsr2ctl ring 1 start
~ $
```

5-7-2 Example

Use 3 JetBox5630 to create a ring and dual-homing to a JetNet network Ring and Dual-Homing Topology:



JB5630_A commands:

```
# modprobe rsr2
# rsr2ctl create 1
# rsr2ctl ring 1 priority 255
# rsr2ctl ring 1 rdh enable
# rsr2ctl ring 1 start
```

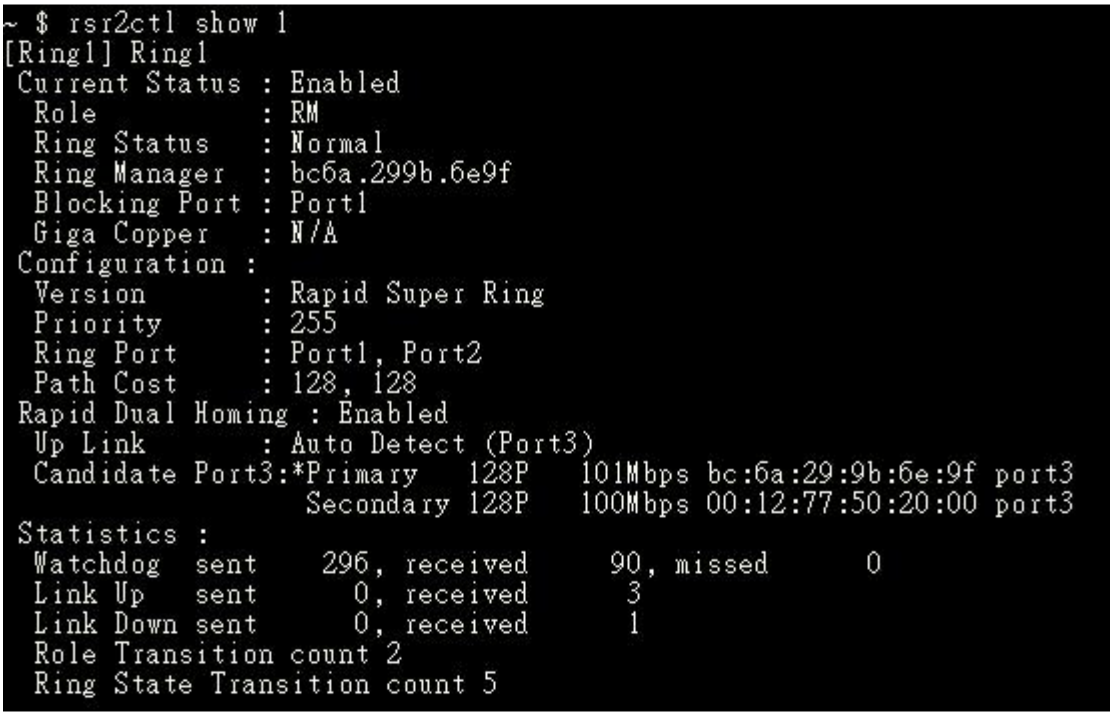
JB5630_B commands:

```
# rsr2ctl create 1
# rsr2ctl ring 1 rdh enable
# rsr2ctl ring 1 start
```

JB5630_C commands:

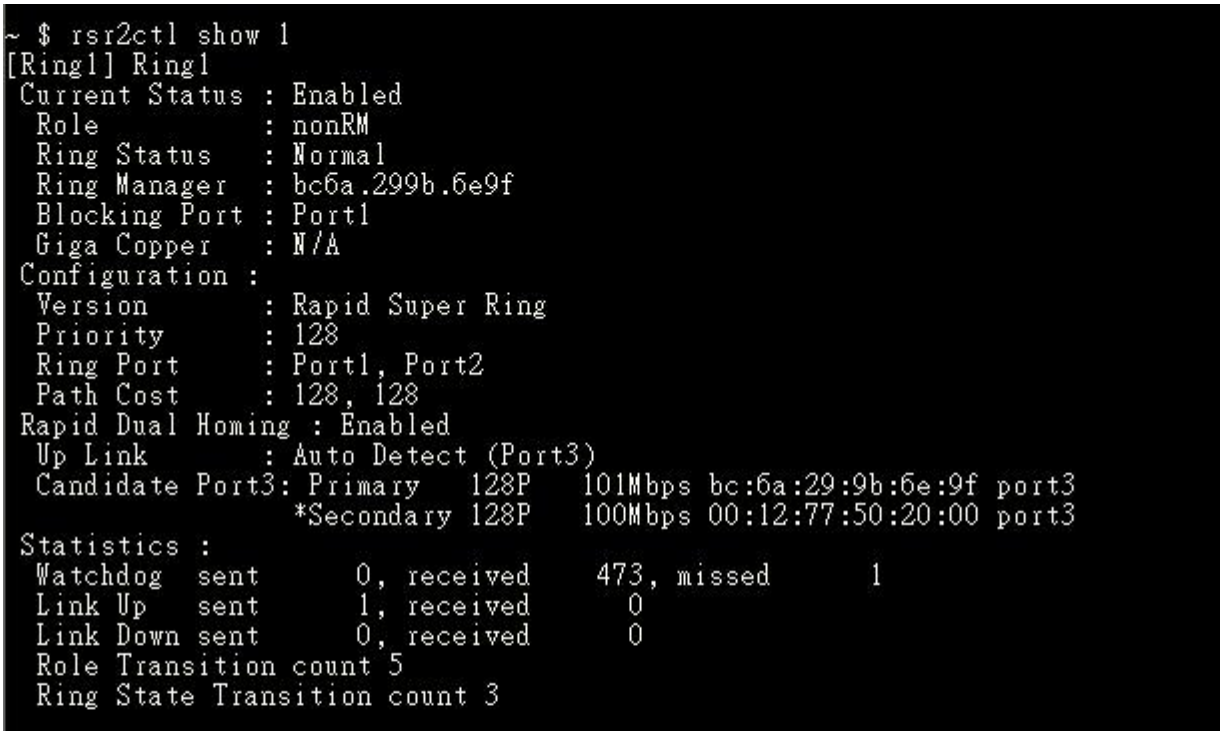
```
# rsr2ctl create 1
# rsr2ctl ring 1 start
```

JB5630_A ring information:



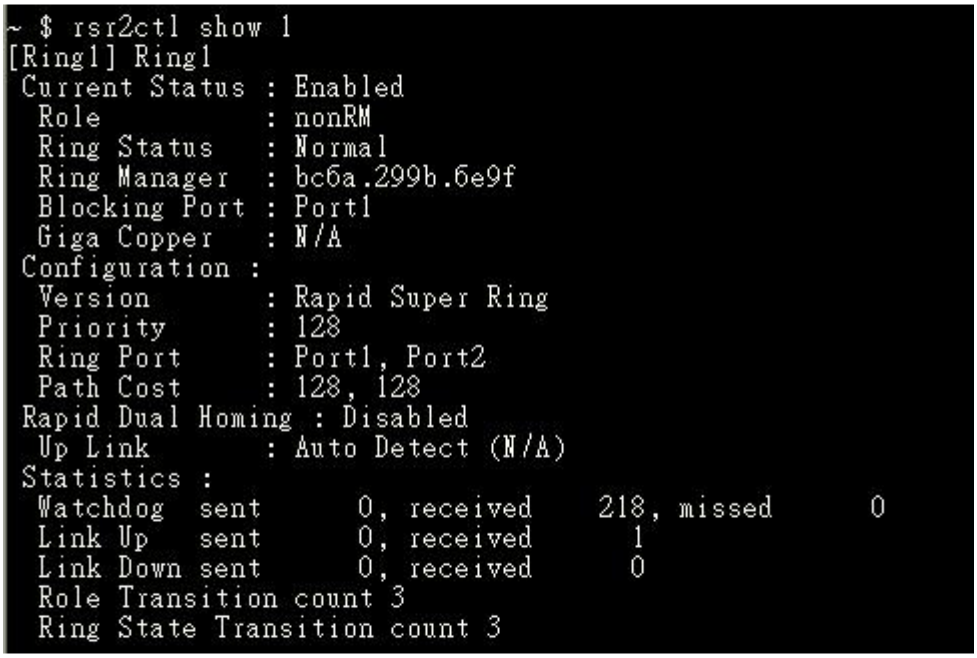
```
~ $ rsr2ctl show 1
[Ring1] Ring1
 Current Status : Enabled
  Role          : RM
  Ring Status   : Normal
  Ring Manager  : bc6a.299b.6e9f
  Blocking Port : Port1
  Giga Copper   : N/A
 Configuration :
  Version       : Rapid Super Ring
  Priority      : 255
  Ring Port     : Port1, Port2
  Path Cost     : 128, 128
 Rapid Dual Homing : Enabled
  Up Link       : Auto Detect (Port3)
  Candidate Port3:*Primary   128P   101Mbps bc:6a:29:9b:6e:9f port3
                   Secondary 128P   100Mbps 00:12:77:50:20:00 port3
 Statistics :
  Watchdog  sent    296, received     90, missed      0
  Link Up   sent      0, received      3
  Link Down sent      0, received      1
  Role Transition count 2
  Ring State Transition count 5
```

JB5630_B ring information:



```
~ $ rsr2ctl show 1
[Ring1] Ring1
 Current Status : Enabled
  Role          : nonRM
  Ring Status   : Normal
  Ring Manager  : bc6a.299b.6e9f
  Blocking Port : Port1
  Giga Copper   : N/A
 Configuration :
  Version       : Rapid Super Ring
  Priority      : 128
  Ring Port     : Port1, Port2
  Path Cost     : 128, 128
 Rapid Dual Homing : Enabled
  Up Link       : Auto Detect (Port3)
  Candidate Port3: Primary   128P   101Mbps bc:6a:29:9b:6e:9f port3
                  *Secondary 128P   100Mbps 00:12:77:50:20:00 port3
 Statistics :
  Watchdog  sent      0, received    473, missed      1
  Link Up   sent      1, received      0
  Link Down sent      0, received      0
  Role Transition count 5
  Ring State Transition count 3
```

JB5630_C ring information:



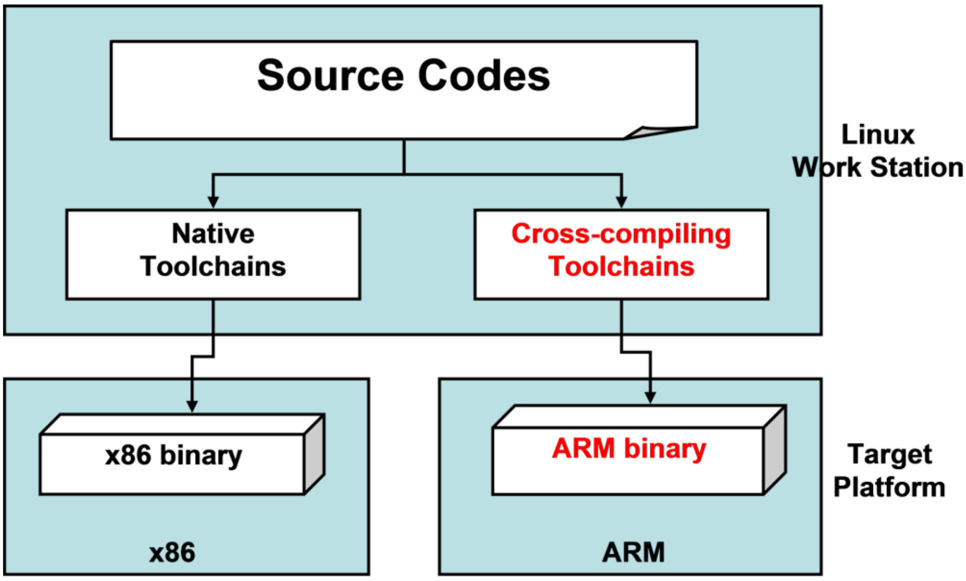
```
~ $ rsr2ctl show 1
[Ring1] Ring1
 Current Status : Enabled
  Role          : nonRM
  Ring Status   : Normal
  Ring Manager  : bc6a.299b.6e9f
  Blocking Port : Port1
  Giga Copper   : N/A
 Configuration :
  Version       : Rapid Super Ring
  Priority      : 128
  Ring Port     : Port1, Port2
  Path Cost     : 128, 128
 Rapid Dual Homing : Disabled
  Up Link       : Auto Detect (N/A)
 Statistics :
  Watchdog  sent      0, received    218, missed      0
  Link Up   sent      0, received      1
  Link Down sent      0, received      0
  Role Transition count 3
  Ring State Transition count 3
```

Chapter 6 Programmer's Guide

6-1 Toolchain Introduction

To ensure that an application will be able to run correctly when installed on Jetbox, you must ensure that it is compiled and linked to the same libraries that will be present on the Jetbox.

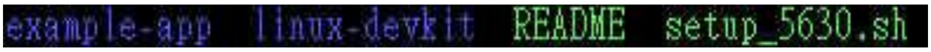
The cross-compiling toolchain that comes with Jetbox5630 contains a suite of Korenix compilers and other tools, as well as the libraries and headers that are necessary to compile applications for Jetbox5630. The build environment must be running Linux and install with the Jetbox5630 Toolchain. We have confirmed that the following Linux distributions can be used to install the tool chain: Ubuntu 10.4, Centos 6.3



The Toolchain will need about 750 MB of hard disk space on your Linux PC. The Jetbox5630 toolchain is included in the Jetbox5630 SDK, which can download from http://www.korenixembedded.com.

The SDK can be extract at any directory, for example, your HOME directory.
You can extract the SDK with following command:

```
tar zxvf jetbox5630-sdk-<version>.tgz
```



```
example-app  linux-devkit  README  setup_5630.sh
```

The README file will teach you how to install the Toolchain, and application examples are in the **example-app** directory.

6-1-1 Compiling Applications and Libraries

To compile a simple C application, just use the arm-arago-linux-gnueabi-gcc compiler instead of the regular one:

```
arm-arago-linux-gnueabi-gcc source-code.c -o output
```

6-1-2 Tools Available in the Host Environment

The following cross compiler tools are provided:

arm-arago-linux-gnueabi-ar	Manage archives (static libraries)
arm-arago-linux-gnueabi-as	Assembler
arm-arago-linux-gnueabi-c++filt	Demangle C++ and Java symbols
arm-arago-linux-gnueabi-cpp	C preprocessor
arm-arago-linux-gnueabi-g++	C++ compiler
arm-arago-linux-gnueabi-gcc	C compiler
arm-arago-linux-gnueabi-gccbug	Shell script which is used to simplify the creation of bug reports
arm-arago-linux-gnueabi-gcov	coverage testing tool
arm-arago-linux-gnueabi-gdb	The GNU Debugger
arm-arago-linux-gnueabi-gdbtui	The GNU Debugger Text User Interface
arm-arago-linux-gnueabi-gprof	Display call graph profile data
arm-arago-linux-gnueabi-ld	Linker
arm-arago-linux-gnueabi-nm	Lists symbols from object files
arm-arago-linux-gnueabi-objcopy	Copies and translates object files
arm-arago-linux-gnueabi-objdump	Displays information about object files
arm-arago-linux-gnueabi-ranlib	Generates indexes to archives (static libraries)
arm-arago-linux-gnueabi-readelf	Displays information about ELF files
arm-arago-linux-gnueabi-size	Lists object file section sizes
arm-arago-linux-gnueabi-strings	Prints strings of printable characters from files (usually object files)
arm-arago-linux-gnueabi-strip	Removes symbols and sections from object files (usually debugging information)

6-2 Device API

ioctl

Name

ioctl - control device

Library

Standard C Library (libc, -lc)

Synopsis

#include <sys/ioctl.h>

int **ioctl**(*int d, unsigned long request, ...*);

Description

The **ioctl**() system call manipulates the underlying device parameters of special files. In particular, many operating characteristics of character special files (e.g. terminals) may be controlled with **ioctl**() requests. The argument *d* must be an open file descriptor.

The third argument to **ioctl**() is traditionally named *char *argp*. Most uses of **ioctl**(), however, require the third argument to be a *caddr_t* or an *int*.

An **ioctl**() *request* has encoded in it whether the argument is an ‘‘in’’ argument or ‘‘out’’ argument, and the size of the argument *argp* in bytes. Macros and defines used in specifying an ioctl *request* are located in the file <*sys/ioctl.h*>.

Return Values

If an error has occurred, a value of -1 is returned and *errno* is set to indicate the error.

Please use the desktop Linux’s man page for detailed documentation:

#man ioctl

6-3 RTC

The device node is located at /dev/rtc0. Jetbox supports Linux standard simple RTC control. You must include <linux/rtc.h>

.

1. Function: RTC_RD_TIME
   int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
   Description: read time information from RTC. It will return the value on argument 3.

2. Function: RTC_SET_TIME
   int ioctl(fd, RTC_SET_TIME, struct rtc_time *time);
   Description: set RTC time. Argument 3 will be passed to RTC.

6-4 Watch Dog Timer

A Watchdog Timer (WDT) is a hardware circuit that can reset the computer system in case of a software fault. You probably knew that already.

The Watchdog Driver has one basic role: to talk to the card and send signals to it so it doesn't reset your computer ... at least during normal operation.

The ioctl API:

- Pinging the watchdog using an ioctl:
  WDIOC_KEEPALIVE:,
  This ioctl does exactly the same thing as a write to the watchdog device, so the main loop in the program could be:

```
while (1) {
        ioctl(fd, WDIOC_KEEPALIVE, 0);
        sleep(10);
}
```

  The argument to the ioctl is ignored.

- Setting and getting the timeout:
  To modify the watchdog timeout on the fly with the SETTIMEOUT ioctl, driver has the WDIOF_SETTIMEOUT flag set in their option field. The argument is an integer representing the timeout in seconds. The driver returns the real timeout used in the same variable, and this timeout might differ from the requested one due to limitation of the hardware.

```
int timeout = 45;
ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
printf("The timeout was set to %d seconds\n", timeout);
```

Starting with the Linux 2.4.18 kernel, it is possible to query the current timeout using the GETTIMEOUT ioctl.

```
ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
printf("The timeout was is %d seconds\n", timeout);
```

6-5 GPIO

This section provides the usage information of GPIO Linux driver usage, both in user and kernel space.

Driver Usage :

Kernel Level

- Allocate memory to GPIO line, can be achieved by doing gpio_request()

```
err = gpio_request(30, "sample_name");
```

- Depending on the requirement set GPIO as input or output pin then set gpio value as high or low. Setting the GPIO pin 30 as input

```
gpio_direction_input(30);
```

- Make pin 30 as output and set the value as high.

```
gpio_direction_output(30, 1);
```

- Exporting that particular pin (30) to sysfs entry then use this API

```
gpio_export(30, true);
```

- Get value from GPIO pin

```
gpio_get_value(30);
```

User Space – sysfs control

- Enable GPIO sysfs support in kernel configuration and build the kernel

```
Device Drivers  ---> GPIO Support  ---> /sys/class/gpio/... (sysfs interface)
```

- Sysfs entries : Export the particular GPIO pin for user control. GPIO30 is taken as example.

```
$ echo 30 > /sys/class/gpio/export
```

- Change the GPIO pin direction to in/out

```
$ echo "out" > /sys/class/gpio/gpio30/direction
```

or

```
$ echo "in" > /sys/class/gpio/gpio30/direction
```

- Change the value

```
$ echo 1 > /sys/class/gpio/gpio30/value
```

or

```
$ echo 0 > /sys/class/gpio/gpio30/value
```

- Unexport the GPIO pin

```
$ echo 30 > /sys/class/gpio/unexport
```

NOTE

GPIO's which are used already in the drivers can not be control from sysfs, unless until driver export that particular pin.

- Run these commands for knowing what are the GPIO's already requested in the drivers.

```
$ mount -t debugfs debugfs /sys/kernel/debug
$ cat /sys/kernel/debug/gpio
```

Chapter 7 Appendix

7-1 Firmware Upgrade

Firmware upgrade can be done by the “firmware_up” command.

```
# firmware_up


Usage: -f [firmware file]
       -t [firmware file] [tftp server]
       -w [http or ftp url]
```

For example, put the new firmware in a USB storage and plug-in to JetBox5630. After it automatic mounted, you can upgrade with the following command:

```
firmware-up -f /media/usb0/Jetbox5630-20140101.bin
```

Where /media/usb0 is the usb mounted directory, and Jetbox5630-20140101.bin is the firmware name.

NOTE
During firmware upgrading, please do not power off device.
When upgrade complete, the system will reboot automatically.

7-2 USB Driver for GSM modems

Some USB 3G modems use the driver USB_SERIAL_COPTION in linux kernel. However, not all user specified USB idProduct and idVendor are include in the driver. For USB idProduct and idVendor not included in linux kernel v2.6.20, JetOS has other way to add it.

```
modprobe c_option vendor=idVender product=idProduct
```

7-3 Software Specification

Item	Protocol	Notes	JetBox 5630
Boot Loader			U-boot
Kernel			3.2.0
	ARP		x
	PPP		x
	CHAP		x
	IPv4		x
	IPv6		x
	PAP		x
	ICMP		x
	TCP		x
	UDP		x
	NFS		x
File System			
JFFS2			x
NFS			x
Ext2			x
Ext3			x
VFAT			x
FAT			x
Base SW package			
Shell		OS shell command	GNU ash
Busybox		Linux normal command utility	1.19.4
telnetd	Telnet	telnet server daemon	x
inetd		TCP server manager program	x
udhcp	DHCP	DHCP client/server	x
syslogd			x
e2fsprogs		Ext2/Ext3 file system utilities	1.42
i2c-tools	I2C	I2C tools for Linux	3.1.0
mtd		MTD/JFFS2 utilities	1.1.0
microcom		Serial port terminal	1.02
pciutils	PCI	PCI utilities	3.1.9
setserial		RS-232 serial port setting tool	2.17

Item	Protocol	Notes	JetBox 5630
usbmount	USB	automatically mounts USB mass storage devices	0.0.22
usbutils	USB	USB utilities	0.04
Network related SW package			
bridge-utils		Ethernet bridge utility	1.5
ethtool		Ethernet configure tool	3.2
iptables		NAT setting tool	1.4.12.2
net-snmp	SNMP v1/v2c/v3	SNMP support package	5.7.1
ntp	NTP	NTP utility	4.2.8
openssh	SSH1.0/2.0	SSH support package	5.9p1
openssl	SSL	SSL support package	1.0.0g
openvpn	OpenVPN	VPN tool	2.2.2
openswan	IPsec	Ipsec for Linux	2.6.37
pppd	PPP	PPP protocol for Linux	2.4.5
rp-pppoe	PPPoE	PPPOE support package	3.1.0
pptp-linux	PPTP	PPTP protocol for Linux	1.7.2
proftpd	FTP	FTP daemon	1.3.3g
samba		SMB (Windows network) support package	3.5.12
bind	DNS	DNS server	9.6
xl2tp	L2TP	L2TP protocol for Linux	1.2.7
mrouted	DVMRP	DVMRP multicast routing protocol	3.9.4
quagga	OSPFv1.0/2.0, RIPv1.0/2.0/ng, BGP4, ISIS	unicast routing protocol	0.99.20
wireless-tools	802.11	Tools of WLAN card	29
Linux tool chain			
Gcc		C/C++ PC Cross Compiler	4.5.3
glib			2.0

7-4 Busybox command

busybox(V1.19.4): Linux command collection

File Manager	
cp	copy file
ls	list file
ln	make symbolic link file
mount	mount and check file system
rm	delete file
chmod	change file owner & group & user
chown	change file owner
chgrp	change file group
sync	Sync file system, let system file buffer be saved to hardware
mv	move file
pwd	display now file directly
df	list now file system space
mkdir	make new directory
rmdir	delete directory

Editor	
vi	text editor
cat	dump file context
zcat	compress or expand files
grep	search string on file
cut	get string on file
find	find file where are there
more	dump file by one page
test	test file exist or not
sleep	sleep(seconds)
echo	Echo string
awk	Pattern scanning and processing language.
diff	compare two files or directories
sed	perform text transformations on a file or input from a pipeline.
xargs	execute a specified command on every item from standard input.

Archival Utilities	
bzip2/bunzip2	Compress/Uncompress bzip FILE
cpio	Extract or list files from a cpio archive
gzip/gunzip	Compress/Uncompress FILE with maximum compression.
tar	Create, extract, or list files from a tar file
unzip	Extract files from ZIP archives

System logging	
syslogd	Utility used to record logs of all the significant events
klogd	Utility which intercepts and logs all messages from the Linux kernel and sends to the 'syslogd'
logger	Utility to send arbitrary text messages to the system log

Network	
ping	ping to test network
arp	Manipulate the system ARP cache
arping	Ping host by ARP packets
ftpget	Retrieve a remote file via FTP
ftpput	Store a remote file via FTP
nslookup	Tool to query Internet name servers
pscan	Simple network port scanner
traceroute	Utility to trace the route of IP packets
wget	Utility for non-interactive download of files from HTTP, HTTPS, and FTP servers.
udhcpc	DHCP client
route	routing table manager
netstat	display network status
Ifconfig	set ip address and configure network interfaces
traceroute	trace route
tftp	Trivial File Transfer Protocol client
telnet	Telnet client

Others	
dmesg	dump kernel log message
stty	stty is used to change and print terminal line settings
zcat	dump .gz file context
mknod	make device node
free	display system memory usage
date	print or set the system date and time
env	run a program in a modified environment
clear	clear the terminal screen
reboot	reboot / power off/on the server
halt	halt the server
du	estimate file space usage
hostname	show system’s host name
kill/killall	Send specified signal to the specified process or process group

For complete command usage and explanation, please refer to following website: http://www.busybox.net/downloads/BusyBox.html

7-5 Ser2net Manual

Name

ser2net - Serial to network proxy

Synopsis

ser2net [-c configfile] [-C configline] [-p controlport] [-n] [-d] [-b] [-v] [-P pidfile]

Description

The **ser2net** daemon allows telnet and tcp sessions to be established with a unit's serial ports.

The program comes up normally as a daemon, opens the TCP ports specified in the configuration file, and waits for connections. Once a connection occurs, the program attempts to set up the connection and open the serial port. If another user is already using the connection or serial port, the connection is refused with an error message.

Options

-c config-file

Set the configuration file to one other than the default of **/etc/ser2net.conf**

-C config-line

Handle a single configuration line. This may be specified multiple times for multiple lines. This is just like a line in the config file. This disables the default config file, you must specify a -c after the last -C to have it read a config file, too.

-n

Stops the daemon from forking and detaching from the controlling terminal. This is useful for running from init.

-d

Like -n, but also sends the system logs to standard output. This is most useful for debugging purposes.

-P pidfile

If specified, put the process id (pid) of ser2net in the pidfile, replacing whatever was in that file previously. A pidfile is not created by default, you must specify this to create one. Note also that this filename must be specific with the full path, as ser2net will change directory to "/" when it becomes a daemon. when it

-u

If UUCP locking is enabled, this will disable the use of UUCP locks.

-b

Cisco IOS uses a different mechanism for specifying the baud rates than the mechanism described in RFC2217. This option sets the IOS version of setting the baud rates. The default is RFC2217's.

-v

Prints the version of the program and exits.

-p controlport

Enables the control port and sets the TCP port to listen to for the control port. A port number may be of the form [host,]port, such as 127.0.0.1,2000 or localhost,2000. If this is specified, it will only bind to the IP address specified for the port. Otherwise, it will bind to all the addresses on the machine.

If the port number is zero, that means that standard in/out will be used for the only input/output, and only one port should be specified in the config. This way, it can be used from inetd.

Control Port

The control port provides a simple interface for controlling the ports and viewing their status. To accomplish this, it has the following commands:

showport [<TCP port>]

Show information about a port. If no port is given, all ports are displayed.

showshortport [<TCP port>]

Show information about a port, each port on one line. If no port is given, all ports are displayed. This can produce very wide output.

help

Display a short list and summary of commands.

exit

Disconnect from the control port.

version

Display the version of this program.

monitor <type> <tcp port>

Display all the input for a given port on the calling control port. Only one direction may be monitored at a time. The type field may be *tcp* or *term* and specifies whether to monitor data from the TCP port or from the serial port Note that data monitoring is best effort, if the controller port cannot keep up the data will be silently dropped. A controller may only monitor one thing and a port may only be monitored by one controller.

monitor stop

Stop the current monitor.

disconnect <tcp port>

Disconnect the tcp connection on the port.

setporttimeout <tcp port> <timeout>

Set the amount of time in seconds before the port connection will be shut down if no activity has been seen on the port.

setportconfig <tcp port> <config>

Set the port configuration as in the device configuration in the **/etc/ser2net.conf** file. If conflicting options are specified, the last option will be the one used. Note that these will not change until the port is disconnected and connected again.

Options *300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200* set the various baud rates. *EVEN, ODD, NONE* set the parity. *1STOPBIT, 2STOPBITS* set the number of stop bits. *7DATABITS,8DATABITS* set the number of data bits. *[-]XONXOFF* turns on (off) XON/XOFF support. *[-]RTSCTS* turns on (- off) hardware flow control. *[-]LOCAL* ignores (- checks) the modem control lines (DCD, DTR, etc.)

setportcontrol <tcp port> <controls>

Modify dynamic port controls. These do not stay between connections. Controls are: *DTRHI, DTRLO* Turns on and off the DTR line. *RTSHI, RTSLO* Turns on and off the RTS line.

setportenable <tcp port> <enable state> Sets the port operation state. Valid states are: *off* to shut the TCP port down, *raw* to enable the TCP port transfer all I/O as-is, *rawlp* to enable the TCP port input and device output without termios setting, and *telnet* to enable the TCP port is up run the telnet negotiation protocol on the port.

Configuration

Configuration is accomplished through the file **/etc/ser2net.conf**. A file with another name or path may be specified using the *-c* option, or individual config lines may be specified with the *-C* option. This file consists of one or more entries with the following format:

<TCP port>:<state>:<timeout>:<device>:<options>

or

BANNER:<banner name>:<banner text>

FIELDS

TCP port

Name or number of the TCP/IP port to accept connections from for this device. A port number may be of the form [host,]port, such as 127.0.0.1,2000 or localhost,2000. If this is specified, it will only bind to the IP address specified for the port. Otherwise, it will bind to all the ports on the machine.

state Either **raw** or **rawlp** or **telnet** or **off.** *off* disables the port from accepting connections. It can be turned on later from the control port. *raw* enables the port and transfers all data as-is between the port and the long. *rawlp* enables the port and transfers all input data to device, device is open without any termios setting. It allow to use /dev/lpX devices and printers connected to them. *telnet* enables the port and runs the telnet protocol on the port to set up telnet parameters. This is most useful for using telnet.

timeout

The time (in seconds) before the port will be disconnected if there is no activity on it. A zero value disables this funciton.

device The name of the device to connect to. This must be in the form of **/dev/<device>**.

device configuration options

Sets operational parameters for the serial port. Values may be separated by spaces or commas. Options *300,1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200* set the various baud rates. *EVEN, ODD, NONE* set the parity. *1STOPBIT, 2STOPBITS* set the number of stop bits. *7DATABITS, 8DATABITS* set the number of data bits.*[-]XONXOFF* turns on (- off) XON/XOFF support. *[-]RTSCTS* turns on (- off) hardware flow control. *[-]LOCAL*ignores (- checks) the modem control lines (DCD, DTR, etc.) *[-]HANGUP_WHEN_DONE* lowers (- does not lower) the modem control lines (DCD, DTR, etc.) when the connection closes. *NOBREAK* Disables automatic clearing of the break setting of the port. *rem_ctl* allows remote control of the serial port parameters via RFC 2217. See the README for more info. *<banner name>* displays the given banner when a user connects to the port.

banner name

A name for the banner; this may be used in the options of a port.

banner text

The text to display as the banner. This may contain normal "C" escape strings, and it may also contain, \d for the device name, \p for the TCP port number, and \s for the serial port parameters (eg 9600N81) of the given connection.

Blank lines and lines starting with '#' are ignored.

Security

ser2net uses the tcp wrappers interface to implement host-based security. See hosts_access(5) for a description of the file setup. Two daemons are used by ser2net, "ser2net" is for the data ports and "ser2net-control" is for the control ports.

Signals

SIGHUP

If ser2net receives a SIGHUP, it will reread it configuration file and make the appropriate changes. If an inuse port is changed or deleted, the actual change will not occur until the port is disconnected.

Error

Almost all error output goes to syslog, not standard output.

Files

/etc/ser2net.conf

See Also

telnet(1) , hosts_access(5)

Known Problems

None.

Author

Corey Minyard <minyard@acm.org>

7-6 Customer Service

Korenix Technology Co., Ltd.
Business service: sales@korenixembedded.com, sales@korenix.com
Customer service: koreCARE@korenix.com
Web Site: http://www.korenixembedded.com, http://www.korenix.com