DA-710 Series
Linux User's Manual

First Edition, December 2009

www.moxa.com/product

DA-710 Series
Linux User's Manual

The Moxa software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Trademarks

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas:
Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa China (Shanghai office):
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-10-6872-3958

Moxa Europe:
Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa Asia-Pacific:
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

1

Introduction

Thank you for purchasing the Moxa DA-710 Series of x86 ready-to-run embedded computers. This manual introduces the software configuration and management of the DA-710-LX, which runs the Linux operating system. For hardware installation, connector interfaces, setup, and upgrading the BIOS, please refer to the “DA-710 Series Hardware User’s Manual.”

Linux is an open, scalable operating system that allows you to build a wide range of innovative, small footprint devices. Software written for desktop PCs can be easily ported to the embedded computer with a GNU cross compiler and a minimum of source code modifications. A typical Linux-based device is designed for a specific use, and is often not connected to other computers, or a number of such devices connect to a centralized, front-end host. Examples include enterprise tools such as industrial controllers, communications hubs, point-of-sale terminals, and display devices, which include HMIs, advertisement appliances, and interactive panels.

This chapter covers the following topics:

- ❑ **Overview**
- ❑ **Software Specifications**
- ❑ **Software Components**

Overview

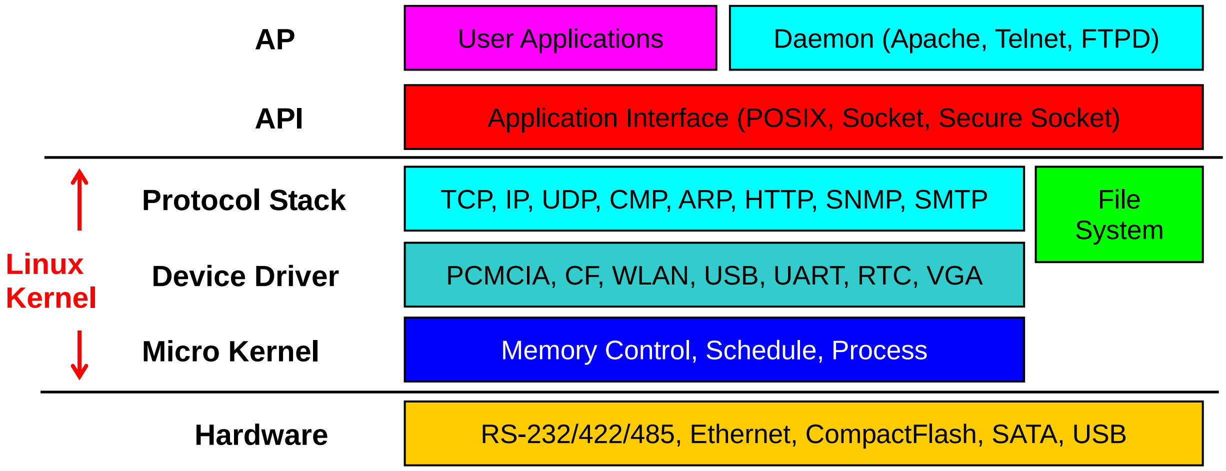
The Moxa DA-710 Series of x86-based rackmount embedded computers are designed for industrial data acquisition applications. Their state-of-the-art two expansion module design gives users a versatile combination of up to 16 RS-232/422/485 serial ports, or up to 4+8 Ethernet ports. This friendly design gives users the advantage of being able to swap out modules quickly and easily. Additional expansion modules will be available soon.

The DA-710 main system is based on the Intel Celeron M processor and GLE960 chipset, which supports standard X86 VGA, USB, PS/2 keyboard/mouse, 4 Gigabit LAN ports, and IDE/SATA disk interface. In addition, the DA-710 supports a CompactFlash Socket and pre-installed embedded ready-to-run operating system. Programmers will find the full-function development examples a great benefit for developing software and building reliable communication applications.

The housing is a standard 4U, 19-inch wide rack-mounted rugged enclosure. This robust, rack-mountable design provides the hardened protection needed for industrial environment applications.

Software Specifications

The Linux operating system pre-installed on the DA-710 embedded computer is the **Debian Lenny 5.0** distribution. The Debian project is a worldwide group of volunteers who endeavor to produce an operating system distribution that composed entirely of free software. The Debian GNU/Linux follows the standard Linux architecture, making it easy to use programs that meet the POSIX standard. Program porting can be done with the GNU Tool Chain provided by Moxa. In addition to Standard POSIX APIs, device drivers for Moxa UART and other special peripherals are also included. An example software architecture is shown below:


AP
User Applications
Daemon (Apache, Telnet, FTPD)
API
Application Interface (POSIX, Socket, Secure Socket)
Linux Kernel
Protocol Stack
TCP, IP, UDP, CMP, ARP, HTTP, SNMP, SMTP
File System
Device Driver
PCMCIA, CF, WLAN, USB, UART, RTC, VGA
Micro Kernel
Memory Control, Schedule, Process
Hardware
RS-232/422/485, Ethernet, CompactFlash, SATA, USB

ATTENTION

Refer to http://www.debian.org/ and http://www.gnu.org/ for information and documentation of the Debian GNU/Linux and free software concept.

ATTENTION

The above software architecture is only an example. Different models or different build revisions of the Linux operating system may include components not shown in the above graphic.

Software Components

The DA-710-LX pre-installed Debian Lenny 5.0 Linux distribution has the following software components:

acpi-support-base	0.109-11	scripts for handling base ACPI events such as the power button
acpid	1.0.8-1lenny1	Utilities for using ACPI power management
adduser	3.110	add and remove users and groups
apache2	2.2.9-10+lenny4	Apache HTTP Server metapackage
apache2-mpm-prefork	2.2.9-10+lenny4	Apache HTTP Server - traditional non-threaded model
apache2-utils	2.2.9-10+lenny4	utility programs for webservers
apache2.2-common	2.2.9-10+lenny4	Apache HTTP Server common files
apt	0.7.20.2+lenny1	Advanced front-end for dpkg
apt-utils	0.7.20.2+lenny1	APT utility programs
aptitude	0.4.11.11-1~lenny1	terminal-based package manager
autoconf	2.61-8	automatic configure script builder
autoconf2.13	2.13-59	automatic configure script builder (obsolete version)
automake	1:1.10.1-3	A tool for generating GNU Standards-compliant Makefiles
automake1.4	1:1.4-p6-13	A tool for generating GNU Standards-compliant Makefiles
autotools-dev	20080123.1	Update infrastructure for config.{guess,sub} files
base-files	5lenny3	Debian base system miscellaneous files
base-passwd	3.5.20	Debian base system master password and group files
bash	3.2-4	The GNU Bourne Again SHell
bash-completion	20080705	programmable completion for the bash shell
bc	1.06.94-3	The GNU bc arbitrary precision calculator language
bind9-host	1:9.5.1.dfsg.P3-1	Version of 'host' bundled with BIND 9.X
binutils	2.18.1~cvs20080103-7	The GNU assembler, linker and binary utilities
bridge-utils	1.4-5	Utilities for configuring the Linux Ethernet bridge
bsdmainutils	6.1.10	collection of more utilities from FreeBSD
bsdutils	1:2.13.1.1-1	Basic utilities from 4.4BSD-Lite
busybox	1:1.10.2-2	Tiny utilities for small and embedded systems
console-common	0.7.80	basic infrastructure for text console configuration
console-data	2:1.07-11	keymaps, fonts, charset maps, fallback tables for console-tool
console-tools	1:0.2.3dbs-65.1	Linux console and font utilities
coreutils	6.10-6	The GNU core utilities
cpio	2.9-13	GNU cpio -- a program to manage archives of files
cpp	4:4.3.2-2	The GNU C preprocessor (cpp)
cpp-4.3	4.3.2-1.1	The GNU C preprocessor
cron	3.0pl1-105	management of regular background processing

debconf	1.5.24	Debian configuration management system
debconf-i18n	1.5.24	full internationalization support for debconf
debian-archive-keyring	2009.01.31	GnuPG archive keys of the Debian archive
debian-faq	4.0.4	The Debian FAQ
debianutils	2.30	Miscellaneous utilities specific to Debian
dhcp3-client	3.1.1-6+lenny2	DHCP client
dhcp3-common	3.1.1-6+lenny2	common files used by all the dhcp3* packages
dictionaries-common	0.98.12	Common utilities for spelling dictionary tools
diff	2.8.1-12	File comparison utilities
dmidecode	2.9-1	Dump Desktop Management Interface data
dnsutils	1:9.5.1.dfsg.P3-1	Clients provided with BIND
dpkg	1.14.25	Debian package management system
e2fslibs	1.41.3-1	ext2 filesystem libraries
e2fsprogs	1.41.3-1	ext2/ext3/ext4 file system utilities
findutils	4.4.0-2	utilities for finding files--find, xargs
ftp	0.17-18	The FTP client
g++	4:4.3.2-2	The GNU C++ compiler
g++-4.3	4.3.2-1.1	The GNU C++ compiler
gcc	4:4.3.2-2	The GNU C compiler
gcc-4.2-base	4.2.4-6	The GNU Compiler Collection (base package)
gcc-4.3	4.3.2-1.1	The GNU C compiler
gcc-4.3-base	4.3.2-1.1	The GNU Compiler Collection (base package)
gdb	6.8-3	The GNU Debugger
gettext-base	0.17-4	GNU Internationalization utilities for the base system
gnupg	1.4.9-3+lenny1	GNU privacy guard - a free PGP replacement
gpgv	1.4.9-3+lenny1	GNU privacy guard - signature verification tool
grep	2.5.3~dfsg-6	GNU grep, egrep and fgrep
groff-base	1.18.1.1-21	GNU troff text-formatting system (base system components)
grub	0.97-47lenny2	GRand Unified Bootloader (Legacy version)
grub-common	1.96+20080724-16	GRand Unified Bootloader, version 2 (common files)
gzip	1.3.12-6	The GNU compression utility
hostname	2.95	utility to set/show the host name or domain name
ifenslave	2	Attach and detach slave interfaces to a bonding device
ifenslave-2.6	1.1.0-10	Attach and detach slave interfaces to a bonding device
ifupdown	0.6.8+nmu1	high level tools to configure network interfaces
initramfs-tools	0.92o	tools for generating an initramfs
initscripts	2.86.ds1-61	Scripts for initializing and shutting down the system
iproute	20080725-2	networking and traffic control tools
iptables	1.4.2-6	administration tools for packet filtering and NAT
iputils-ping	3:20071127-1	Tools to test the reachability of network hosts
klibc-utils	1.5.12-2	small utilities built with klibc for early boot
libacl1	2.2.47-2	Access control list shared library
libapache2-mod-php5	5.2.6.dfsg.1-1+lenny3	server-side, HTML-embedded scripting language (Apache 2 module
libapr1	1.2.12-5+lenny1	The Apache Portable Runtime Library
libaprutil1	1.2.12+dfsg-8+lenny4	The Apache Portable Runtime Utility Library
libattr1	1:2.4.43-2	Extended attribute shared library

libbind9-40	1:9.5.1.dfsg.P3-1	BIND9 Shared Library used by BIND
libblkid1	1.41.3-1	block device id library
libbz2-1.0	1.0.5-1	high-quality block-sorting file compressor library - runtime
libc6	2.7-18	GNU C Library: Shared libraries
libc6-dev	2.7-18	GNU C Library: Development Libraries and Header Files
libc6-i686	2.7-18	GNU C Library: Shared libraries [i686 optimized]
libcap1	1:1.10-14	support for getting/setting POSIX.1e capabilities
libcap2	2.11-2	support for getting/setting POSIX.1e capabilities
libcomerr2	1.41.3-1	common error description library
libconsole	1:0.2.3dbs-65.1	Shared libraries for Linux console and font manipulation
libcwidget3	0.5.12-4	high-level terminal interface library for C++ (runtime files)
libdb4.5	4.5.20-13	Berkeley v4.5 Database Libraries [runtime]
libdb4.6	4.6.21-11	Berkeley v4.6 Database Libraries [runtime]
libdevmapper1.02.1	2:1.02.27-4	The Linux Kernel Device Mapper userspace library
libdns45	1:9.5.1.dfsg.P3-1	DNS Shared Library used by BIND
libedit2	2.11~20080614-1	BSD editline and history libraries
libept0	0.5.22	High-level library for managing Debian package information
libevent1	1.3e-3	An asynchronous event notification library
libexpat1	2.0.1-4	XML parsing C library - runtime library
libgc1c2	1:6.8-1.1	conservative garbage collector for C and C++
libgcc1	1:4.3.2-1.1	GCC support library
libgcrypt11	1.4.1-1	LGPL Crypto library - runtime library
libgdbm3	1.8.3-3	GNU dbm database routines (runtime version)
libgmp3c2	2:4.2.2+dfsg-3	Multiprecision arithmetic library
libgnutls26	2.4.2-6+lenny1	the GNU TLS library - runtime library
libgomp1	4.3.2-1.1	GCC OpenMP (GOMP) support library
libgpg-error0	1.4-2	library for common error values and messages in GnuPG componen
libgpm2	1.20.4-3.1	General Purpose Mouse - shared library
libgssglue1	0.1-2	mechanism-switch gssapi library
libidn11	1.8+20080606-1	GNU libidn library, implementation of IETF IDN specifications
libisc45	1:9.5.1.dfsg.P3-1	ISC Shared Library used by BIND
libisccc40	1:9.5.1.dfsg.P3-1	Command Channel Library used by BIND
libisccfg40	1:9.5.1.dfsg.P3-1	Config File Handling Library used by BIND
libkeyutils1	1.2-9	Linux Key Management Utilities (library)
libklibc	1.5.12-2	minimal libc subset for use with initramfs
libkrb53	1.6.dfsg.4~beta1-5lenny1	MIT Kerberos runtime libraries
libldap-2.4-2	2.4.11-1	OpenLDAP libraries
liblocale-gettext-perl	1.05-4	Using libc functions for internationalization in Perl
liblockfile1	1.08-3	NFS-safe locking library, includes dotlockfile program
liblwres40	1:9.5.1.dfsg.P3-1	Lightweight Resolver Library used by BIND
liblzo2-2	2.03-1	data compression library
libmagic1	4.26-1	File type determination library using "magic" numbers
libmpfr1ldbl	2.3.1.dfsg.1-2	multiple precision floating-point computation

libmysqlclient15off	5.0.51a-24+lenny2	MySQL database client library
libncurses5	5.7+20081213-1	shared libraries for terminal handling
libncursesw5	5.7+20081213-1	shared libraries for terminal handling (wide character support
libnet-lite-ftp-perl	0.54-2	Perl FTP client with support for TLS
libnet-ssleay-perl	1.35-1	Perl module for Secure Sockets Layer (SSL)
libnet-telnet-perl	3.03-3	Script telnetable connections
libnewt0.52	0.52.2-11.3	Not Erik's Windowing Toolkit - text mode windowing with slang
libnfsidmap2	0.20-1	An nfs idmapping library
libpam-modules	1.0.1-5+lenny1	Pluggable Authentication Modules for PAM
libpam-runtime	1.0.1-5+lenny1	Runtime support for the PAM library
libpam0g	1.0.1-5+lenny1	Pluggable Authentication Modules library
libpcap0.8	0.9.8-5	system interface for user-level packet capture
libpci3	1:3.0.0-6	Linux PCI Utilities (shared library)
libpcre3	7.6-2.1	Perl 5 Compatible Regular Expression Library - runtime files
libperl5.10	5.10.0-19lenny2	Shared Perl library
libpkcs11-helper1	1.05-1	library that simplifies the interaction with PKCS#11
libpopt0	1.14-4	lib for parsing cmdline parameters
libpq5	8.3.7-0lenny1	PostgreSQL C client library
libreadline5	5.2-3.1	GNU readline and history libraries, run-time libraries
librpcsecgss3	0.18-1	allows secure rpc communication using the rpcsec_gss protocol
libsasl2-2	2.1.22.dfsg1-23+lenny1	Cyrus SASL - authentication abstraction library
libselinux1	2.0.65-5	SELinux shared libraries
libsensors3	1:2.10.7-1	library to read temperature/voltage/fan sensors
libsepol1	2.0.30-2	Security Enhanced Linux policy library for changing policy bin
libsigc++-2.0-0c2a	2.0.18-2	type-safe Signal Framework for C++ - runtime
libslang2	2.1.3-3	The S-Lang programming library - runtime version
libsnmp-base	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) MIBs and documentati
libsnmp15	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) library
libsqlite3-0	3.5.9-6	SQLite 3 shared library
libss2	1.41.3-1	command-line interface parsing library
libssl0.9.8	0.9.8g-15+lenny1	SSL shared libraries
libstdc++6	4.3.2-1.1	The GNU Standard C++ Library v3
libstdc++6-4.3-dev	4.3.2-1.1	The GNU Standard C++ Library v3 (development files)
libsysfs2	2.1.0-5	interface library to sysfs
libtasn1-3	1.4-1	Manage ASN.1 structures (runtime)
libtext-charwidth-per	0.04-5+b1	get display widths of characters on the terminal
libtext-iconv-perl	1.7-1+b1	converts between character sets in Perl
libtext-wrapi18n-perl	0.06-6	internationalized substitute of Text::Wrap
libusb-0.1-4	2:0.1.12-13	userspace USB programming library
libuuid1	1.41.3-1	universally unique id library

libvolume-id0	0.125-7+lenny1	libvolume_id shared library
libwrap0	7.6.q-16	Wietse Venema's TCP wrappers library
libx11-6	2:1.1.5-2	X11 client-side library
libx11-data	2:1.1.5-2	X11 client-side library
libxapian15	1.0.7-4	Search engine library
libxau6	1:1.0.3-3	X11 authorisation library
libxcb-xlib0	1.1-1.2	X C Binding, Xlib/XCB interface library
libxcb1	1.1-1.2	X C Binding
libxdmcp6	1:1.0.2-3	X11 Display Manager Control Protocol library
libxext6	2:1.0.4-1	X11 miscellaneous extension library
libxml2	2.6.32.dfsg-5	GNOME XML library
libxmuu1	2:1.0.4-1	X11 miscellaneous micro-utility library
linux-image-2.6-686	2.6.26+17+lenny1	Linux 2.6 image on PPro/Celeron/P/PI/P4
linux-image-2.6.26-2-686	2.6.26-17lenny1	Linux 2.6.26 image on PPro/Celeron/P/PI/P4
linux-libc-dev	2.6.26-19	Linux support headers for userspace development
locales	2.7-18	GNU C Library: National Language (locale) data [support]
lockfile-progs	0.1.11-0.1	Programs for locking and unlocking files and mailboxes
login	1:4.1.1-6	system login tools
logrotate	3.7.1-5	Log rotation utility
lrzsz	0.12.21-4.1	Tools for zmodem/xmodem/ymodem file transfer
lsb-base	3.2-20	Linux Standard Base 3.2 init script functionality
lzma	4.43-14	Compression method of 7z format in 7-Zip program
m4	1.4.11-1	a macro processing language
make	3.81-5	The GNU version of the "make" utility.
makedev	2.3.1-88	creates device files in /dev
man-db	2.5.2-4	on-line manual pager
manpages	3.05-1	Manual pages about using a GNU/Linux system
mawk	1.3.3-11.1	a pattern scanning and text processing language
mime-support	3.44-1	MIME files 'mime.types' & 'mailcap', and support programs
minicom	2.3-1	friendly menu driven serial communication program
mktemp	1.5-9	tool for creating temporary files
mlocate	0.21.1-1	quickly find files on the filesystem based on their name
modconf	0.3.9	Device Driver Configuration
module-init-tools	3.4-1	tools for managing Linux kernel modules
mount	2.13.1.1-1	Tools for mounting and manipulating filesystems
mutt	1.5.18-6	text-based mailreader supporting MIME, GPG, PGP and threading
mysql-common	5.0.51a-24+lenny2	MySQL database common files
ncurses-base	5.7+20081213-1	basic terminal type definitions
ncurses-bin	5.7+20081213-1	terminal-related programs and man pages
ncurses-term	5.7+20081213-1	additional terminal type definitions
net-tools	1.60-22	The NET-3 networking toolkit
netbase	4.34	Basic TCP/IP networking system
netcat-traditional	1.10-38	TCP/IP swiss army knife
nfs-common	1:1.1.2-6lenny1	NFS support files common to client and server

ntpdate	1:4.2.4p4+dfsg-8lenny2	client for setting system time from NTP servers
openbsd-inetd	0.20080125-2	The OpenBSD Internet Superserver
openssh-blacklist	0.4.1	list of default blacklisted OpenSSH RSA and DSA keys
openssh-blacklist-extra	0.4.1	list of non-default blacklisted OpenSSH RSA and DSA keys
openssh-client	1:5.1p1-5	secure shell client, an rlogin/rsh/rcp replacement
openssh-server	1:5.1p1-5	secure shell server, an rshd replacement
openssl	0.9.8g-15+lenny5	Secure Socket Layer (SSL) binary and related cryptographic too
openssl-blacklist	0.4.2	list of blacklisted OpenSSL RSA keys
openvpn	2.1~rc11-1	virtual private network daemon
openvpn-blacklist	0.3	list of blacklisted OpenVPN RSA shared keys
passwd	1:4.1.1-6	change and administer password and group data
pciutils	1:3.0.0-6	Linux PCI Utilities
perl	5.10.0-19lenny2	Larry Wall's Practical Extraction and Report Language
perl-base	5.10.0-19lenny2	minimal Perl system
perl-modules	5.10.0-19lenny2	Core Perl modules
php5-common	5.2.6.dfsg.1-1+lenny3	Common files for packages built from the php5 source
portmap	6.0-9	RPC port mapper
ppp	2.4.4rel-10.1	Point-to-Point Protocol (PPP) - daemon
pppconfig	2.3.18	A text menu based utility for configuring ppp
pppoe	3.8-3	PPP over Ethernet driver
pppoeconf	1.18	configures PPPoE/ADSL connections
procps	1:3.2.7-11	/proc file system utilities
proftpd	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon
proftpd-basic	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - binaries
proftpd-mod-ldap	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - LDAP module
proftpd-mod-mysql	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - MySQL module
proftpd-mod-pgsql	1.3.1-17lenny2	versatile, virtual-hosting FTP daemon - PostgreSQL module
python	2.5.2-3	An interactive high-level object-oriented language (default ve
python-minimal	2.5.2-3	A minimal subset of the Python language (default version)
python2.5	2.5.2-15	An interactive high-level object-oriented language (version 2.
python2.5-minimal	2.5.2-15	A minimal subset of the Python language (version 2.5)
readline-common	5.2-3.1	GNU readline and history libraries, common files
rsyslog	3.18.6-4	enhanced multi-threaded syslogd
sed	4.1.5-6	The GNU sed stream editor
snmp	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) applications
snmpd	5.4.1~dfsg-12	SNMP (Simple Network Management Protocol) agents
ssh	1:5.1p1-5	secure shell client and server (metapackage)
ssl-cert	1.0.23	simple debconf wrapper for OpenSSL
sysv-rc	2.86.ds1-61	System-V-like runlevel change mechanism
sysvinit	2.86.ds1-61	System-V-like init utilities
sysvinit-utils	2.86.ds1-61	System-V-like utilities

tar	1.20-1	GNU version of the tar archiving utility
tasksel	2.78	Tool for selecting tasks for installation on Debian systems
tasksel-data	2.78	Official tasks used for installation of Debian systems
tcpd	7.6.q-16	Wietse Venema's TCP wrapper utilities
tcpdump	3.9.8-4	A powerful tool for network monitoring and data acquisition
telnet	0.17-36	The telnet client
telnetd	0.17-36	The telnet server
tftpd	0.17-16	Trivial file transfer protocol server
time	1.7-23	The GNU time program for measuring cpu resource usage
traceroute	2.0.11-2	Traces the route taken by packets over an IPv4/IPv6 network
tzdata	2009g-0lenny1	time zone and daylight-saving time data
ucf	3.0016	Update Configuration File: preserve user changes to config fil
udev	0.125-7+lenny1	/dev/ and hotplug management daemon
update-inetd	4.31	inetd configuration file updater
usbmount	0.0.14.1	automatically mount and unmount USB mass storage devices
usbutils	0.73-10	Linux USB utilities
util-linux	2.13.1.1-1	Miscellaneous system utilities
vim	1:7.1.314-3+lenny2	Vi IMproved - enhanced vi editor
vim-common	1:7.1.314-3+lenny2	Vi IMproved - Common files
vim-runtime	1:7.1.314-3+lenny2	Vi IMproved - Runtime files
vim-tiny	1:7.1.314-3+lenny2	Vi IMproved - enhanced vi editor - compact version
w3m	0.5.2-2+b1	WWW browsable pager with excellent tables/frames support
watchdog	5.4-10	A software watchdog
wget	1.11.4-2	retrieves files from the web
whiptail	0.52.2-11.3	Displays user-friendly dialog boxes from shell scripts
whois	4.7.30	an intelligent whois client
x11-common	1:7.3+19	X Window System (X.Org) infrastructure
xauth	1:1.0.3-2	X authentication utility
zlib1g	1:1.2.3.3.dfsg-12	compression library - runtime

2

Software Configuration

In this chapter, we explain how to operate a DA-710-LX computer directly or from a PC near you. There are three ways to connect to the DA-710-LX computer: through VGA monitor, by using Telnet over the network, or by using an SSH console from a Windows or Linux machine. This chapter describes basic Linux operating system configurations. The advanced network management and configuration will be described in the next chapter “Managing Communications.”

This chapter covers the following topics:

- ❑ **Starting from a VGA Console**
- ❑ **Connecting from a Telnet Console**
- ❑ **Connecting from an SSH Console**
  - ➢ Windows Users
  - ➢ Linux Users
- ❑ **Adjusting the System Time**
  - ➢ Setting the Time Manually
  - ➢ NTP Client
  - ➢ Updating the Time Automatically
- ❑ **Enabling and Disabling Daemons**
- ❑ **Setting the Run-Level**
- ❑ **Cron—Daemon for Executing Scheduled Commands**
- ❑ **Inserting a USB Storage Device into the Computer**
- ❑ **Inserting a CompactFlash Card into the Computer**
- ❑ **Checking the Linux Version**
- ❑ **APT—Installing and Removing Packages**

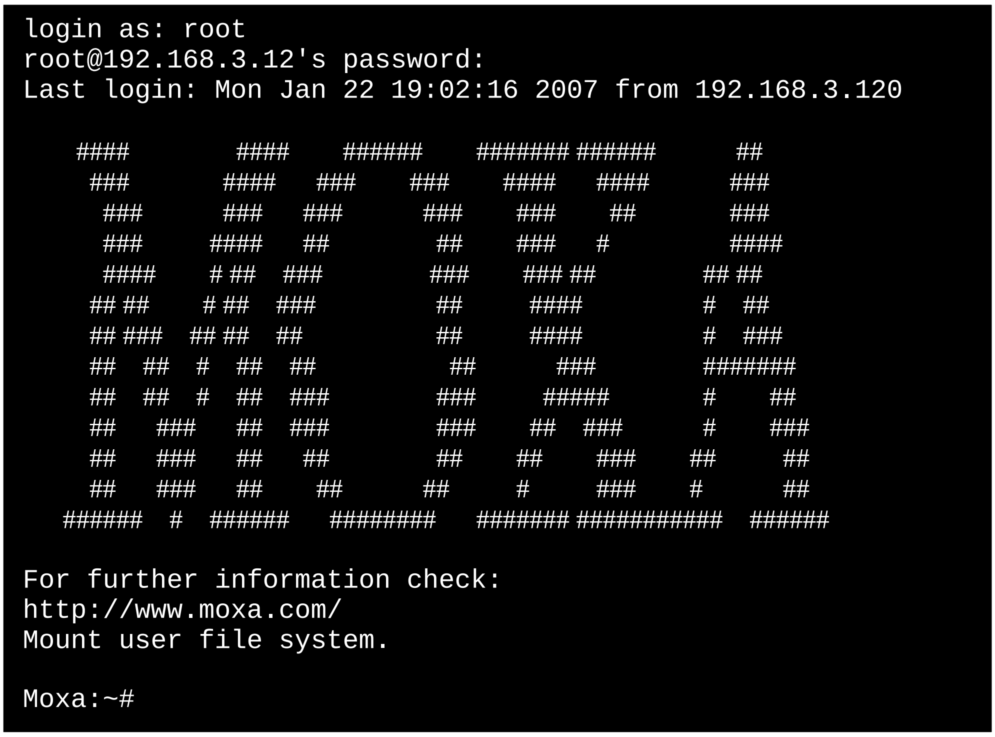
Starting from a VGA Console

Connect the display monitor to the DA-710-LX VGA connector, and then power it up by connecting it to the power adaptor. It takes about 30 to 60 seconds for the system to boot up. Once the system is ready, a login screen will appear on your monitor.

To log in, type the login name and password as requested. The default values are both **root**.

Login: root

Password: root



```
login as: root
root@192.168.3.12's password:
Last login: Mon Jan 22 19:02:16 2007 from 192.168.3.120

        ####                ####        ######        ####### ######            ##
          ###              ####      ###        ###        ####      ####            ###
            ###            ###      ###            ###        ###        ##              ###
            ###          ####      ##                ##        ###      #                  ####
            ####        # ##    ###                ###        ### ##                ## ##
          ## ##        # ##    ###                  ##          ####                  #    ##
          ## ###    ## ##    ##                    ##          ####                  #    ###
          ##    ##    #    ##    ##                    ##            ###                #######
          ##    ##    #    ##    ###                ###          #####              #        ##
          ##      ###      ##    ###                ###        ##    ###            #        ###
          ##      ###      ##      ##                ##        ##        ###        ##          ##
          ##      ###      ##        ##            ##          #          ###        #            ##
      ######    #    ######      ########      ####### ###########    ######

For further information check:
http://www.moxa.com/
Mount user file system.

Moxa:~#
```

Connecting from a Telnet Console

The DA-710-LX computer comes with four basic Gigabit Ethernet ports named LAN1 to LAN4. The default IP addresses and netmasks of the network interfaces are as follows:

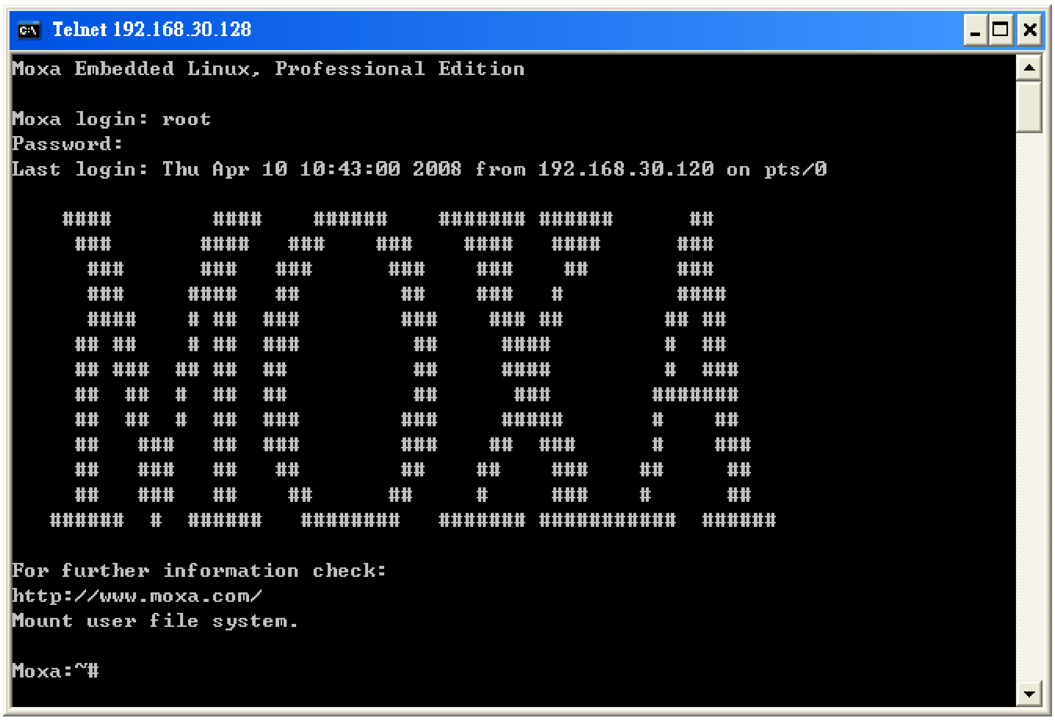
	Default IP Address	Netmask
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0
LAN 3	192.168.5.127	255.255.255.0
LAN 4	192.168.6.127	255.255.255.0

Before using the Telnet client, you should change the IP address of your development workstation so that the network ports are on the same subnet as the IP address for the LAN port that you connect to. For example, if you connect to LAN 1, you could set your PC's IP address to 192.168.3.126, and the netmask to 255.255.255.0. If you connect to LAN 2, you can set your PC's IP address to 192.168.4.126, and the netmask to 255.255.255.0.

Use a cross-over Ethernet cable to connect your development workstation directly to the target computer, or use a straight-through Ethernet cable to connect the computer to a LAN hub or switch. Next, use a Telnet client on your development workstation to connect to the target computer. After a connection has been established, type the login name and password as requested to log on to the computer. The default values are both **root**.

Login: root

Password: root



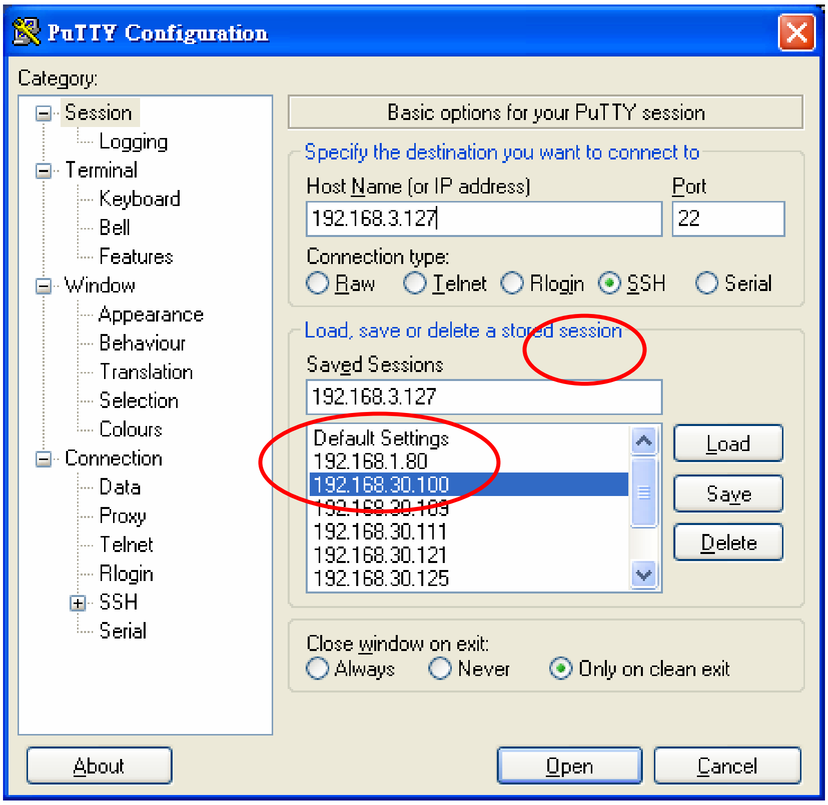
```
Moxa Embedded Linux, Professional Edition

Moxa login: root
Password:
Last login: Thu Apr 10 10:43:00 2008 from 192.168.30.120 on pts/0

    ####        ####    ######    ####### ######      ##
     ###       ####   ###    ###    ####   ####      ###
      ###      ###   ###      ###    ###    ##       ###
      ###     ####   ##        ##    ###   #         ####
      ####    # ##  ###        ###    ### ##        ## ##
     ## ##    # ##  ###         ##     ####         #  ##
     ## ###  ## ##  ##          ##     ####         #  ###
     ##  ##  #  ##  ##          ##      ###        #######
     ##  ##  #  ##  ###        ###     #####       #    ##
     ##   ###   ##  ###        ###    ##  ###      #    ###
     ##   ###   ##   ##        ##    ##    ###    ##     ##
     ##   ###   ##    ##      ##     #     ###    #      ##
   ######  #  ######   ########   ####### ###########  ######

For further information check:
http://www.moxa.com/
Mount user file system.

Moxa:~#
```

Connecting from an SSH Console

The DA-710-LX computer supports an SSH Console to offer users with better security over the network compared to Telnet.

Windows Users

Click on the link http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html to download **PuTTY** (free software) to set up an SSH console for the DA-710-LX in a Windows environment. The following screen shows an example of the configuration that is required.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.3.127

Port: 22

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

Load, save or delete a stored session

Saved Sessions: 192.168.3.127

Default Settings
192.168.1.80
192.168.30.100
192.168.30.109
192.168.30.111
192.168.30.121
192.168.30.125

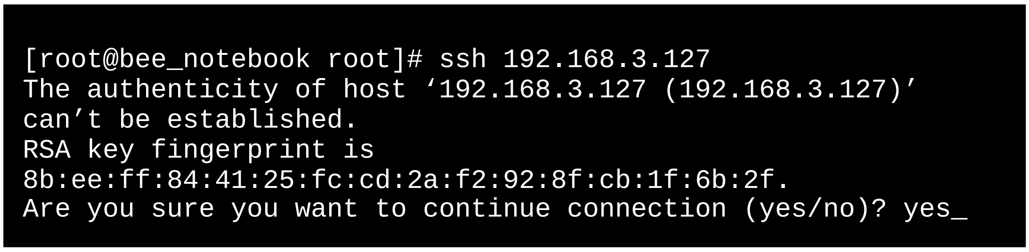
Close window on exit: ○ Always ○ Never ● Only on clean exit

Linux Users

From a Linux machine, use the **ssh** command to access the DA-710-LX’s console utility via SSH.

#ssh 192.168.3.127

Select **yes** to open the connection.



```
[root@bee_notebook root]# ssh 192.168.3.127
The authenticity of host ‘192.168.3.127 (192.168.3.127)’
can’t be established.
RSA key fingerprint is
8b:ee:ff:84:41:25:fc:cd:2a:f2:92:8f:cb:1f:6b:2f.
Are you sure you want to continue connection (yes/no)? yes_
```

Adjusting the System Time

The DA-710-LX has two time settings. One is the system time, and the other is provided by an RTC (Real Time Clock) built into the DA-710- LX’s hardware.

Setting the Time Manually

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

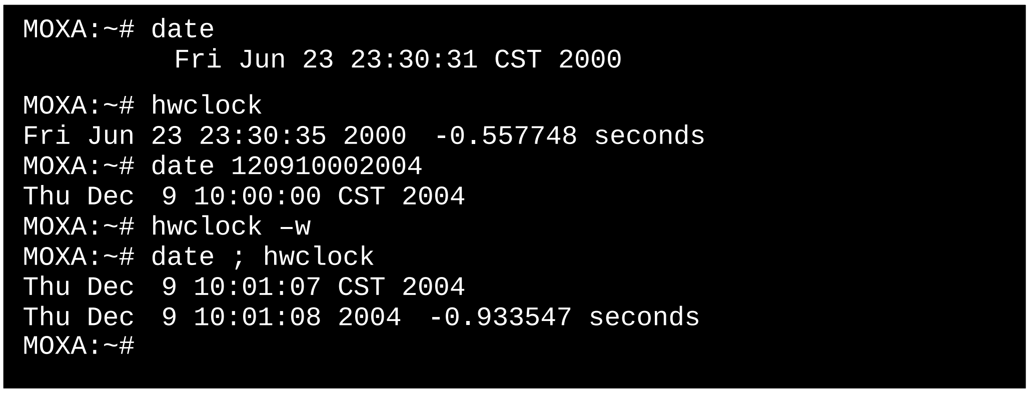
Use the following command to set the system time.

# date MMDDhhmmYYYY

MM:	Month
DD:	Date
hhmm:	Hour and Minute
YYYY:	Year

Use the following command to write the current system time to the RTC.

# hwclock –w



```
MOXA:~# date
         Fri Jun 23 23:30:31 CST 2000
MOXA:~# hwclock
Fri Jun 23 23:30:35 2000  -0.557748 seconds
MOXA:~# date 120910002004
Thu Dec  9 10:00:00 CST 2004
MOXA:~# hwclock –w
MOXA:~# date ; hwclock
Thu Dec  9 10:01:07 CST 2004
Thu Dec  9 10:01:08 2004  -0.933547 seconds
MOXA:~#
```

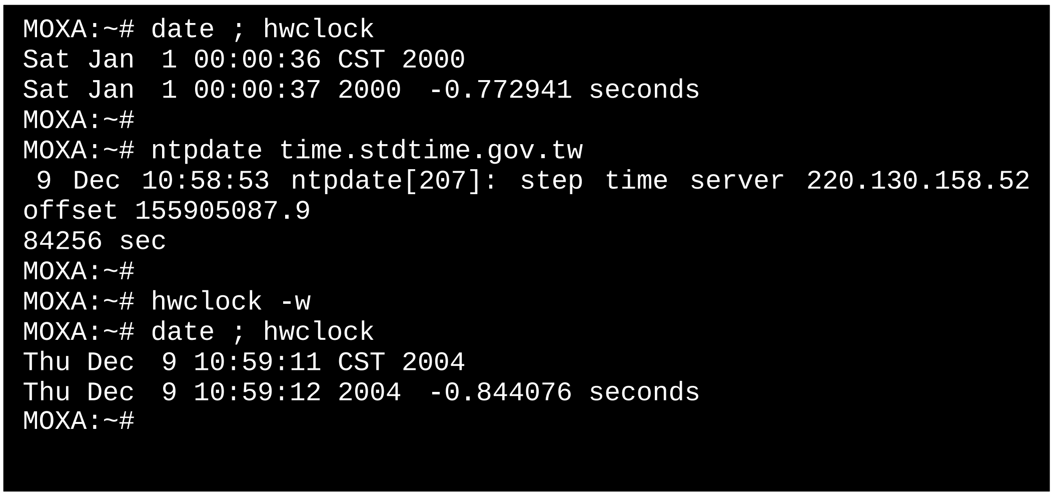
NTP Client

The DA-710-LX 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

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



```
MOXA:~# date ; hwclock
Sat Jan  1 00:00:36 CST 2000
Sat Jan  1 00:00:37 2000  -0.772941 seconds
MOXA:~#
MOXA:~# ntpdate time.stdtime.gov.tw
 9 Dec 10:58:53 ntpdate[207]: step time server 220.130.158.52
offset 155905087.9
84256 sec
MOXA:~#
MOXA:~# hwclock -w
MOXA:~# date ; hwclock
Thu Dec  9 10:59:11 CST 2004
Thu Dec  9 10:59:12 2004  -0.844076 seconds
MOXA:~#
```

ATTENTION

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

Updating the Time Automatically

This section describes how to use a shell script to update the time automatically.

Example shell script for updating the system time periodically

#!/bin/sh
ntpdate time.stdtime.gov.tw
# You can use the time server's ip address or domain
# name directly. If you use domain name, you must
# enable the domain client on the system by updating
# /etc/resolv.conf file.
hwclock –w
sleep 100
# Updates every 100 seconds. The min. time is 100 seconds.
# Change 100 to a larger number to update RTC less often.

Save the shell script using any file name. For example, **fixtime**.

How to run the shell script automatically when the kernel boots up

Because the root file system is mounted in Read-only mode, we need to re-mount it using writable permission.

```
# mount -o remount,rw /dev/hda1 /
```

Copy the example shell script **fixtime** to directory **/etc/init.d**, and then use **chmod 755 fixtime** to change the shell script mode.

```
# chmod 755 fixtime
```

Next, use **vi** editor to edit the file **/etc/inittab**.

```
# vi /etc/inittab
```

Add the following line to the bottom of the file:

```
ntp : 2345 : respawn : /etc/init.d/fixtime
```

After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
# umount /
```

Use the command **#init q** to re-initialize the kernel.

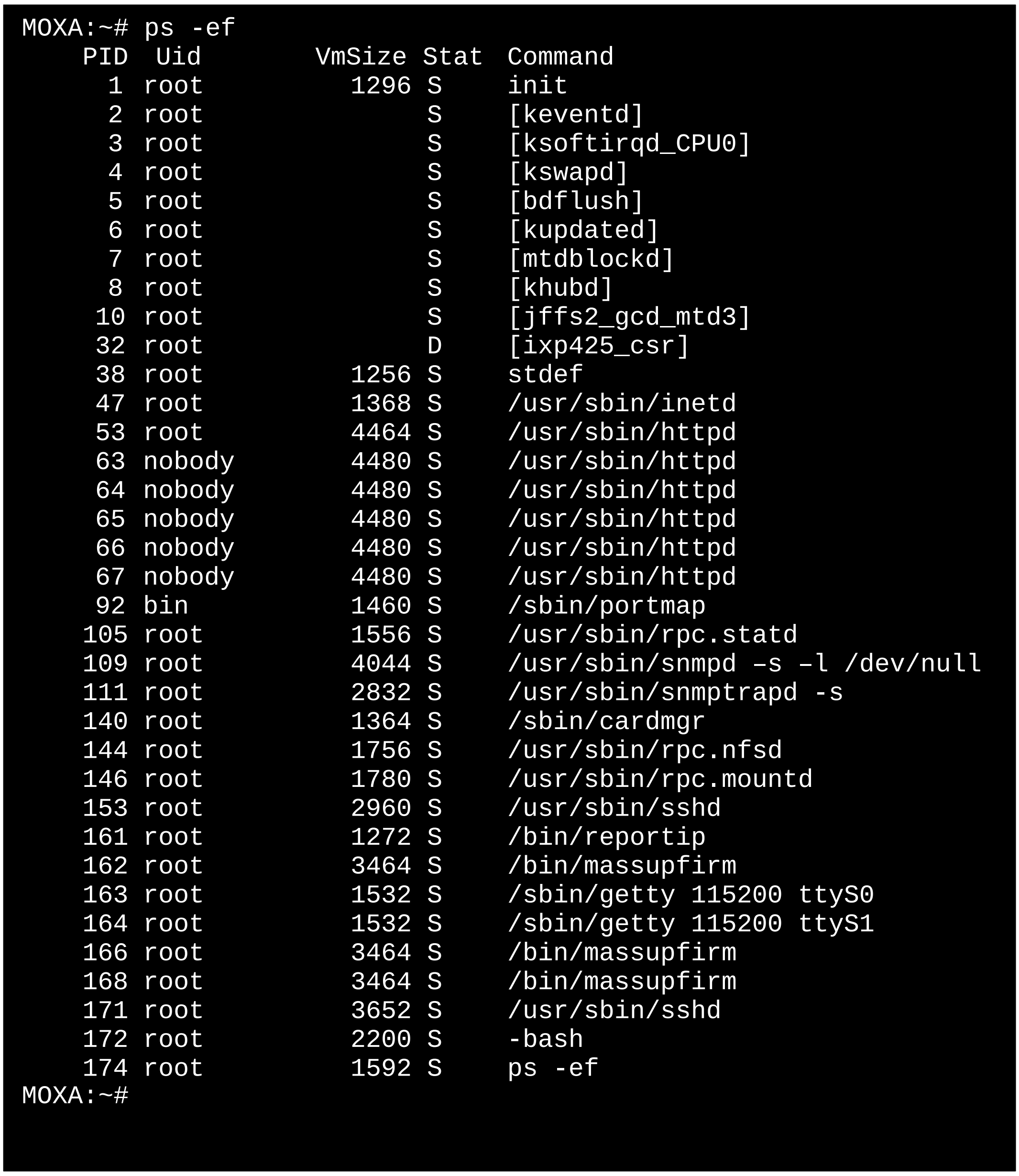
```
# init q
```

Enabling and Disabling Daemons

The following daemons are enabled when the DA-710-LX boots up for the first time.

• **snmpd**	SNMP Agent Daemon
• **telnetd**	Telnet Server/Client Daemon
• **inetd**	Internet Daemons
• **ftpd**	FTP Server/Client Daemon
• **sshd**	Secure Shell Server Daemon
• **httpd**	Apache WWW Server Daemon

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



```
MOXA:~# ps -ef
    PID  Uid       VmSize Stat  Command
      1 root          1296 S    init
      2 root               S    [keventd]
      3 root               S    [ksoftirqd_CPU0]
      4 root               S    [kswapd]
      5 root               S    [bdflush]
      6 root               S    [kupdated]
      7 root               S    [mtdblockd]
      8 root               S    [khubd]
     10 root               S    [jffs2_gcd_mtd3]
     32 root               D    [ixp425_csr]
     38 root          1256 S    stdef
     47 root          1368 S    /usr/sbin/inetd
     53 root          4464 S    /usr/sbin/httpd
     63 nobody        4480 S    /usr/sbin/httpd
     64 nobody        4480 S    /usr/sbin/httpd
     65 nobody        4480 S    /usr/sbin/httpd
     66 nobody        4480 S    /usr/sbin/httpd
     67 nobody        4480 S    /usr/sbin/httpd
     92 bin           1460 S    /sbin/portmap
    105 root          1556 S    /usr/sbin/rpc.statd
    109 root          4044 S    /usr/sbin/snmpd –s –l /dev/null
    111 root          2832 S    /usr/sbin/snmptrapd -s
    140 root          1364 S    /sbin/cardmgr
    144 root          1756 S    /usr/sbin/rpc.nfsd
    146 root          1780 S    /usr/sbin/rpc.mountd
    153 root          2960 S    /usr/sbin/sshd
    161 root          1272 S    /bin/reportip
    162 root          3464 S    /bin/massupfirm
    163 root          1532 S    /sbin/getty 115200 ttyS0
    164 root          1532 S    /sbin/getty 115200 ttyS1
    166 root          3464 S    /bin/massupfirm
    168 root          3464 S    /bin/massupfirm
    171 root          3652 S    /usr/sbin/sshd
    172 root          2200 S    -bash
    174 root          1592 S    ps -ef
MOXA:~#
```

To run a private daemon, you can edit the file **rc.local** as follows:

1. Because the root file system is mounted in Read-only mode, you need to re-mount it with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Type **cd /etc/** to change directories.

```
MOXA:~# cd /etc/
```

3. Type **vi rc.local** to edit the configuration file with vi editor.

```
MOXA:/etc/# vi rc.local
```

4. Next, add the application daemon that you want to run. We use the example program **tcps2-release** which you can find in the CD to illustrate, and configure it to run in the background.

```
# !/bin/sh
# Add you want to run daemon
/root/tcps2-release &~
```

5. After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

6. You should be able to find the enabled daemon after you reboot the system.

```
MOXA:~# ps -ef
    PID  Uid        VmSize Stat  Command
      1 root          1296 S     init
      2 root               S     [keventd]
      3 root               S     [ksoftirqd_CPU0]
      4 root               S     [kswapd]
      5 root               S     [bdflush]
      6 root               S     [kupdated]
      7 root               S     [mtdblockd]
      8 root               S     [khubd]
     10 root               S     [jffs2_gcd_mtd3]
     32 root               D     [ixp425_csr]
     38 root          1256 S     stdef
     47 root          1368 S     /usr/sbin/inetd
     53 root          4464 S     /usr/sbin/httpd
     63 nobody        4480 S     /usr/sbin/httpd
     64 nobody        4480 S     /usr/sbin/httpd
     65 nobody        4480 S     /usr/sbin/httpd
     66 nobody        4480 S     /usr/sbin/httpd
     67 nobody        4480 S     /usr/sbin/httpd
     92 bin           1460 S     /sbin/portmap
     97 root          1264 S     /root/tcps2-release
    105 root          1556 S     /usr/sbin/rpc.statd
    109 root          4044 S     /usr/sbin/snmpd –s –l
    /dev/null
    111 root          2832 S     /usr/sbin/snmptrapd -s
    140 root          1364 S     /sbin/cardmgr
    144 root          1756 S     /usr/sbin/rpc.nfsd
    146 root          1780 S     /usr/sbin/rpc.mountd
    153 root          2960 S     /usr/sbin/sshd
    161 root          1272 S     /bin/reportip
    162 root          3464 S     /bin/massupfirm
    163 root          1532 S     /sbin/getty 115200 ttyS0
    164 root          1532 S     /sbin/getty 115200 ttyS1
    166 root          3464 S     /bin/massupfirm
    168 root          3464 S     /bin/massupfirm
    171 root          3652 S     /usr/sbin/sshd
    172 root          2200 S     -bash
    174 root          1592 S     ps -ef
MOXA:~#
```

Setting the Run-Level

To set the Linux run-level and execution priority of a program, use the following command (because the root file system is mounted in Read-only mode, we need to re-mount it with write permission).

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

Edit a shell script to execute **/root/tcps2-release** and save to **tcps2** as an example.

#cd /etc/rc2.d

#ln –s /etc/root/tcps2 S60tcps2

or

#ln –s /etc/root/tcps2 k30tcps2

```
MOXA:~# cd /etc/rc2.d
MOXA:/etc/rc2.d#
MOXA:/etc/rc2.d# ls
S19nfs-common       S25nfs-user-server  S99showreadyled
S20snmpd            S55ssh
S24pcmcia           S99rmnologin
MOXA:/etc/rc2.d#
MOXA:/etc/rc2.d# ln –s /root/tcps2-release S60tcps2
MOXA:/etc/rc2.d# ls
S19nfs-common       S25nfs-user-server  S99rmnologin
S20snmpd            S55ssh              S99showreadyled
S24pcmcia           S60tcps2
MOXA:/etc/rc2.d#
```

The command **SxxRUNFILE** has the following meaning:

S:	**Start the run file while Linux boots up.**
xx:	**A number between 00-99. The smaller number has a higher priority.**
RUNFILE:	**The script file name**

The command **KxxRUNFILE** has the following meaning:

K:	**Start the run file while Linux shuts down or halts.**
xx:	**A number between 00-99. The smaller number has a higher priority.**
RUNFILE:	**The script file name**

To remove the daemon, remove the run file from /etc/rc2.d by using the following command:

```
MOXA:~# rm –f /etc/rc2.d/S60tcps2
```

After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

Cron—Daemon for Executing Scheduled Commands

The Cron daemon will search **/etc/crontab** for crontab files, which are named after accounts in **/etc/passwd**.

Cron wakes up every minute and checks each command to see if it should be run in that minute. When executing commands, output is mailed to the owner of the **crontab** (or to the user named in the MAILTO environment variable in the **crontab**, if such a user exists).

Modify the file **/etc/crontab** to set up your scheduled applications. **Crontab** files have the following format:

mm	**h**	**dom**	**mon**	**dow**	**user**	**command**
minute	hour	date	month	week	user	command
0-59	0-23	1-31	1-12	0-6 (0 is Sunday)		

For example, if you want to launch a program at 8:00 every day.

```
#minute hour  date month  week  user  command
*        8    *     *      *    root  /path/to/your/program
```

The following example demonstrates how to use **Cron** to update the system time and RTC time every day at 8:00.

1. Write a shell script named fixtime.sh and save it to **/home/**.

   #!/bin/sh
   ntpdate time.stdtime.gov.tw
   hwclock –w
   exit 0

2. Change mode of **fixtime.sh**

   # chmod 755 fixtime.sh

3. Modify /etc/crontab file to run fixtime.sh at 8:00 every day.

   Add the following line to the end of crontab:

   *** 8 * * *   root /home/fixtime.sh**

Inserting a USB Storage Device into the Computer

Since mounting USB storage devices manually can be difficult, a Debian package named **usbmount** to mount the USB drivers automatically. **usbmount** relies on **udev** to mount USB storage devices automatically at certain mount points. The USB storage devices will be mounted on **/media/usb0, /media/usb1**, etc.

```
MOXA:~# mount
/dev/hda1 on / type ext2 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
procbususb on /proc/bus/usb type usbfs (rw)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts
(rw,noexec,nosuid,gid=5,mode=620)
/dev/hdb2 on /home type ext2 (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sda1 on /media/usb0 type vfat
(rw,noexec,nodev,sync,noatime,gid=25,dmask=0007,fmask=0117)
/dev/sdb1 on /media/usb1 type vfat
(rw,noexec,nodev,sync,noatime,gid=25,dmask=0007,fmask=0117)
MOXA:~#
```

ATTENTION

Remember to type the command **# sync** before you disconnect the USB storage device. If you do not issue the command, you may lose data.

ATTENTION

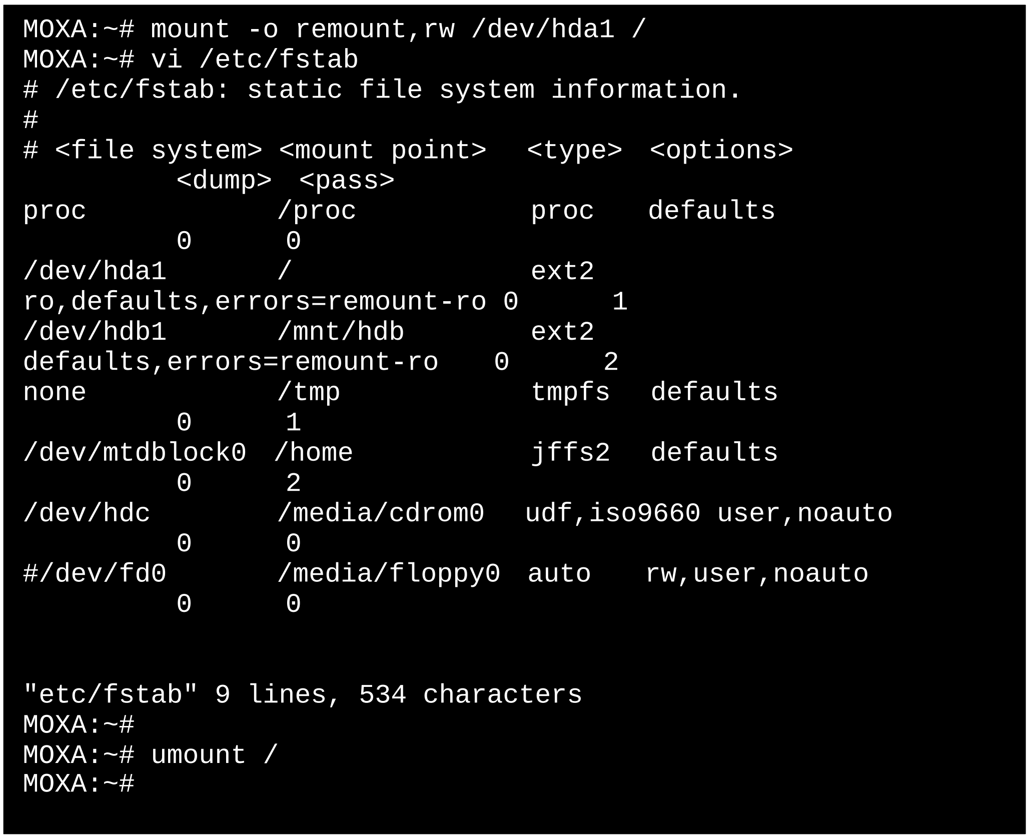
Remember to exit the **/media/usb0 or /media/usb1** directory when you disconnect the USB storage device. If you stay in **/media/usb0** or **/media/usb1**, the automatic un-mount process will fail. If that happens, type **# umount /media/usb0** to un-mount the USB device manually.

Inserting a CompactFlash Card into the Computer

The CompactFlash card is treated as a local disk drive in the DA-710-LX computer. It is identified as a block device at **/dev/hdb**. You can add one line to **/etc/fstab** to force the CompactFlash card to mount automatically at boot time.

ATTENTION

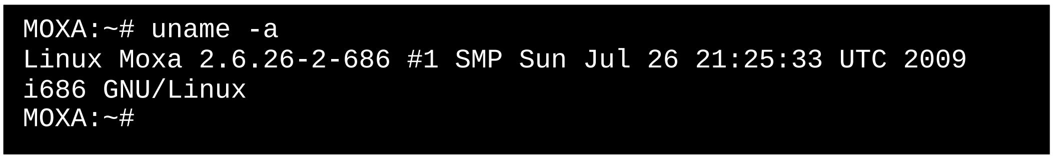
The DA-710 Series Embedded Computer does not support the CompactFlash hot swap function. You must remove the power source first before inserting or removing the CompactFlash card. If you do not shut down the power source, you could damage your CompactFlash card.



```
MOXA:~# mount -o remount,rw /dev/hda1 /
MOXA:~# vi /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>
         <dump>  <pass>
proc            /proc           proc    defaults
         0      0
/dev/hda1       /               ext2
ro,defaults,errors=remount-ro 0      1
/dev/hdb1       /mnt/hdb        ext2
defaults,errors=remount-ro    0      2
none            /tmp            tmpfs   defaults
         0      1
/dev/mtdblock0  /home           jffs2   defaults
         0      2
/dev/hdc        /media/cdrom0   udf,iso9660 user,noauto
         0      0
#/dev/fd0       /media/floppy0  auto    rw,user,noauto
         0      0


"etc/fstab" 9 lines, 534 characters
MOXA:~#
MOXA:~# umount /
MOXA:~#
```

Checking the Linux Version

The program **uname**, which stands for "Unix Name" and is part of the Unix operating system, prints the name, version, and other details about the operating system running on the computer. Use the **-a** option to generate a response similar to the one shown below:



```
MOXA:~# uname -a
Linux Moxa 2.6.26-2-686 #1 SMP Sun Jul 26 21:25:33 UTC 2009
i686 GNU/Linux
MOXA:~#
```

APT—Installing and Removing Packages

APT is the Debian tool used to install and remove packages. Before installing a package, you need to configure the apt source file, **/etc/apt/sources.list**, which is located in the read-only partition.

1. Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Next, configure the **/etc/apt/sources.list** using **vi** editor.

```
MOXA:~# vi /etc/apt/sources.list

#
# deb cdrom:[Debian GNU/Linux 5.0.2a _Lenny_ - Official i386
NETINST Binary-1 20
090817-16:43]/ lenny main

#deb cdrom:[Debian GNU/Linux 5.0.2a _Lenny_ - Official i386
NETINST Binary-1 200
90817-16:43]/ lenny main

deb http://ftp.us.debian.org/debian/ lenny main
deb-src http://ftp.us.debian.org/debian/ lenny main

deb http://security.debian.org/ lenny/updates main contrib
deb-src http://security.debian.org/ lenny/updates main contrib

deb http://volatile.debian.org/debian-volatile lenny/volatile
main
deb-src http://volatile.debian.org/debian-volatile
lenny/volatile main
```

3. Update the source list after you configure it.

```
MOXA:~# apt-get update
MOXA:~#
```

4. Once you indicate which package you want to install (**openswan**, for example), type:

```
MOXA:~# apt-get install openswan
MOXA:~#
```

5. Use one of the following commands to remove a package:
   (a) For a simple package removal:

```
MOXA:~# apt-get remove openswan
MOXA:~#
```

   (b) For a complete package removal:

```
MOXA:~# apt-get remove openswan --purge
MOXA:~#
```

6. If the installation is complete, remember to umount the root directory back to read-only mode.

```
MOXA:~# umount /
MOXA:~#
```

ATTENTION

The APT cache space **/etc/cache/apt** is located in **tmpfs**. If you need to install a huge package, link **/etc/cache/apt** to USB mass storage or mount it to an NFS space to generate more free space. Use **df –h** to check how much free space is available on **tmpfs**.

```
MOXA:~# df -h
Filesystem           Size  Used Avail Use% Mounted on
rootfs              772M  397M  335M  55% /
udev                 10M   68K   10M   1% /dev
/dev/hda1            772M  397M  335M  55% /
tmpfs               502M     0  502M   0% /lib/init/rw
tmpfs               502M     0  502M   0% /dev/shm
none                502M   19M  483M   4% /tmp
/dev/hda2            133M   73M   53M  59% /home
MOXA:~#
```

ATTENTION

You can free up the cache space with the command **# apt-get clean**

```
MOXA:~# apt-get clean
MOXA:~#
```

3

Managing Communications

The DA-710-LX ready-to-run embedded computer is a network-centric platform designed to serve as a front-end for data acquisition and industrial control applications. This chapter describes how to configure the various communication functions supported by the Linux operating system.

This chapter covers the following topics:

- ❑ **Changing the Network Settings**
  - ➢ Changing the “interfaces” Configuration File
  - ➢ Adjusting IP Addresses with “ifconfig”
  - ➢ Configuring Multiple LAN Ports for Expansion Module
  - ➢ Configuring Multiple Serial Ports for Expansion Modules
- ❑ **Serial Port Operation Mode**
- ❑ **Telnet/FTP Server**
- ❑ **DNS Client**
- ❑ **Apache Web Server**
  - ➢ Default Homepage
  - ➢ Disabling the CGI Function
- ❑ **IPTABLES**
  - ➢ IPTABLES Hierarchy
  - ➢ IPTABLES Modules
  - ➢ Observe and Erase Chain Rules
  - ➢ Define Policy for Chain Rules
  - ➢ Append or Delete Rules
- ❑ **NAT (Network Address Translation)**
  - ➢ NAT Example
  - ➢ Enabling NAT at Bootup
- ❑ **PPP (Point to Point Protocol)**
  - ➢ Connecting to a PPP Server over a Simple Dial-up Connection
  - ➢ Connecting to a PPP Server over a Hard-wired Link
  - ➢ Checking the Connection
  - ➢ Setting up a Machine for Incoming PPP Connections
- ❑ **PPPoE**
- ❑ **NFS (Network File System) Client**
- ❑ **SNMP (Simple Network Management Protocol)**
- ❑ **OpenVPN**
  - ➢ Ethernet Bridging for Private Networks on Different Subnets
  - ➢ Ethernet Bridging for Private Networks on the Same Subnet
  - ➢ Routed IP

Changing the Network Settings

The DA-710-LX computer has four basic Gigabit Ethernet ports named LAN1 to LAN4. The LAN Port Expansion Module supports an additional four 10/100 Mbps Ethernet ports named LAN5 to LAN8. The default IP addresses and netmasks of the network interfaces are as follows:

	Default IP Address	Netmask
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0
LAN 3	192.168.5.127	255.255.255.0
LAN 4	192.168.6.127	255.255.255.0

These network settings can be modified by changing the **interfaces** configuration file, or they can be adjusted temporarily with the **ifconfig** command.

Changing the "interfaces" Configuration File

1. Type **cd /etc/network** to change directory.

```
MOXA:~# cd /etc/network
```

2. Type **vi interfaces** to edit the network configuration file with **vi** editor. You can configure the DA-710-LX's Ethernet ports for static or dynamic (DHCP) IP addresses.

```
MOXA:/etc/network# vi interfaces
```

Static IP Address

As shown in the example shown below, the default static IP addresses can be modified.

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
       address 192.168.3.127
       netmask 255.255.255.0
       broadcast 192.168.3.255

auto eth1
iface eth1 inet static
       address 192.168.4.127
       netmask 255.255.255.0
       broadcast 192.168.4.255

auto eth2
iface eth2 inet static
       address 192.168.5.127
       netmask 255.255.255.0
       broadcast 192.168.5.255
```

Dynamic IP Address using DHCP

To configure one or both LAN ports to request an IP address dynamically, replace **static** with **dhcp** and then delete the rest of the lines.

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
```

After modifying the boot settings of the LAN interface, issue the following command to activate the LAN settings immediately.

# /etc/init.d/networking restart

```
MOXA:~# /etc/init.d/networking restart
```

Adjusting IP Addresses with "ifconfig"

IP settings can be adjusted during run-time, but the new settings will not be saved to the flash ROM without modifying the file **/etc/network/interfaces**. For example, type the command **# ifconfig eth0 192.168.1.1** to change the IP address of LAN1 to 192.168.1.1.

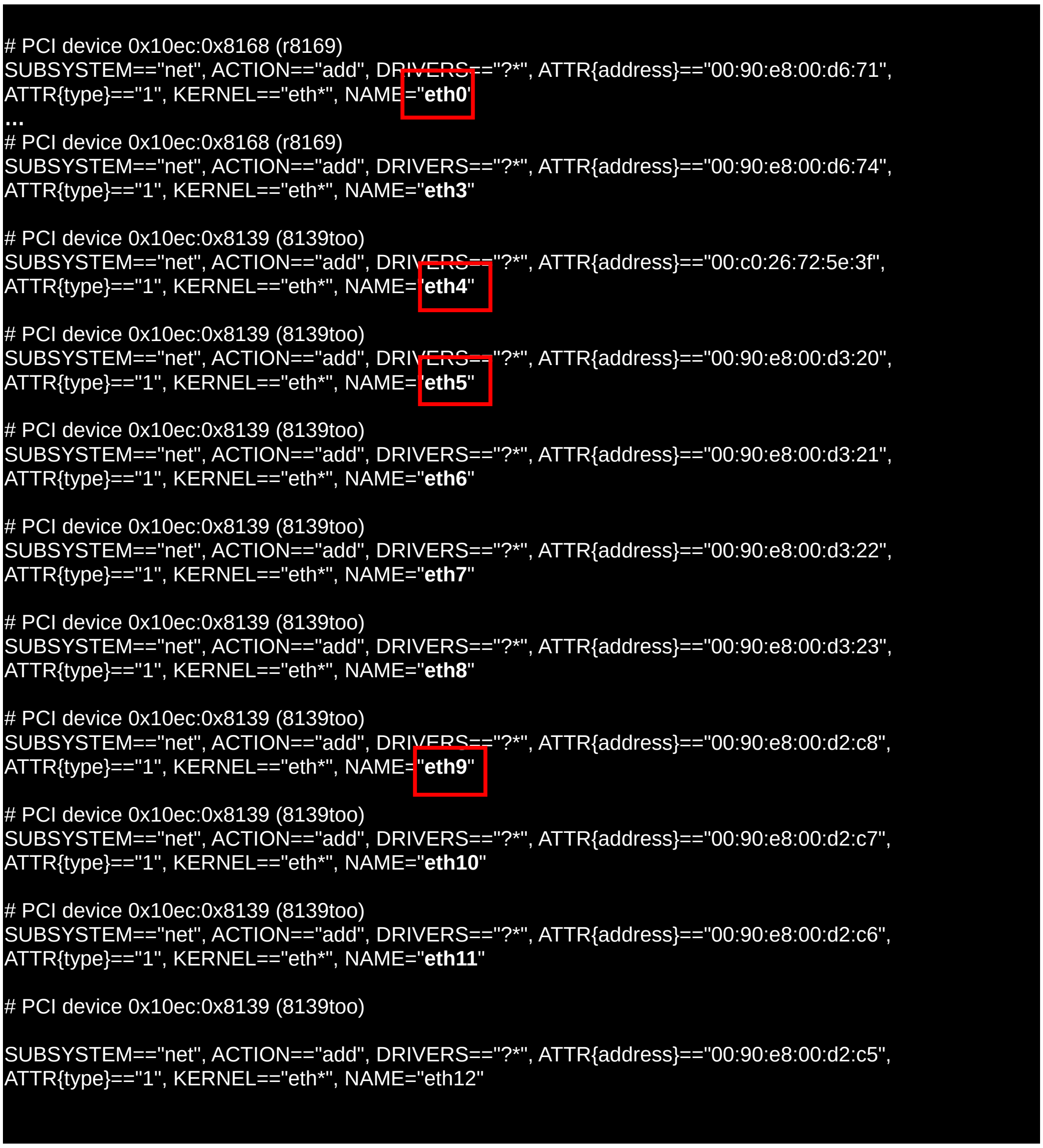
```
MOXA:~# ifconfig eth0 192.168.1.1
MOXA:~#
```

Configuring Multiple LAN Ports for Expansion Modules

This section explains how to configure the LAN interface when you have inserted one or more LAN or switch modules in your DA-710's expansion PCI slots.

In the following example, we will insert two DA-LN04-RJ LAN modules and one DA-SW08-RJ switch module. Follow the steps below.

1. Make sure the DA-710 computer is powered off.
2. Insert the three modules on the rear panel of the DA-710.
3. Turn on the DA-710 computer.
4. When the system has rebooted, udev daemon will detect these modules and generate a file: **/etc/udev/rules.d/70-persistent-net.rules**.
5. Use an editor such as vi to configure **/etc/udev/rules.d/70-persistent-net.rules**.



```
# PCI device 0x10ec:0x8168 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d6:71",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
...
# PCI device 0x10ec:0x8168 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d6:74",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth3"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:c0:26:72:5e:3f",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth4"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d3:20",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth5"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d3:21",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth6"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d3:22",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth7"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d3:23",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth8"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d2:c8",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth9"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d2:c7",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth10"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d2:c6",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth11"

# PCI device 0x10ec:0x8139 (8139too)

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:90:e8:00:d2:c5",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth12"
```

You may see the configuration for each port. See the following descriptions for details.

- **LAN port eth0 ~ eth3:** the default network configuration for default LAN1 to LAN 4.
- **Switch port eth4:** the network configuration for the switch port.
- **LAN port eth5 ~ eth8:** the network configuration for the LAN ports on the first expansion module.
- **LAN port eth9 ~ eth12:** the network configuration for the LAN ports on the second expansion module.

6. If you want to assign the port number on your own, simply change the unique number in **ethn**, value; **n** refers to the port number you want to change. For example, you may change eth5 to eth8.
7. When you finish editing, save the file and reboot your computer so that the new configuration can take effect.
8. For example, if you would like to let eth5 take effect, you can either issue "ifup eth5" or add configurations into /etc/network/interfaces:

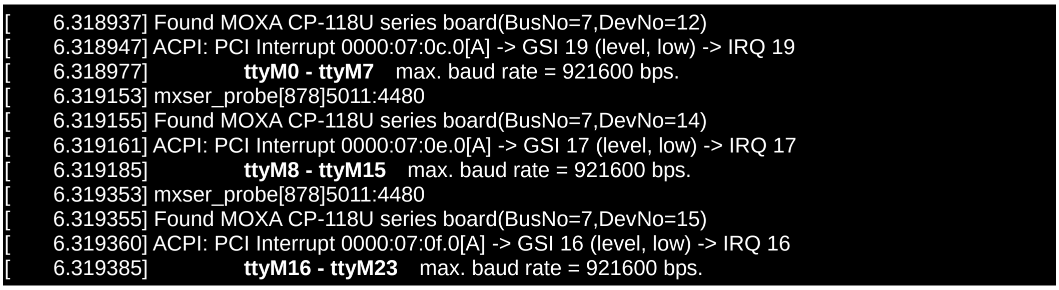
```
auto eth5

iface eth5 inet static

    address 192.168.8.127

    netmask 255.255.255.0

    broadcast 192.168.8.255
```

Configuring Multiple Serial Ports for Expansion Modules

This section explains how to configure the serial port interface when you have inserted one or more serial expansion modules on your DA-710's expansion PCI slots.

In the following case, we will insert three DA-SP08-DB expansion modules. Follow the steps below.

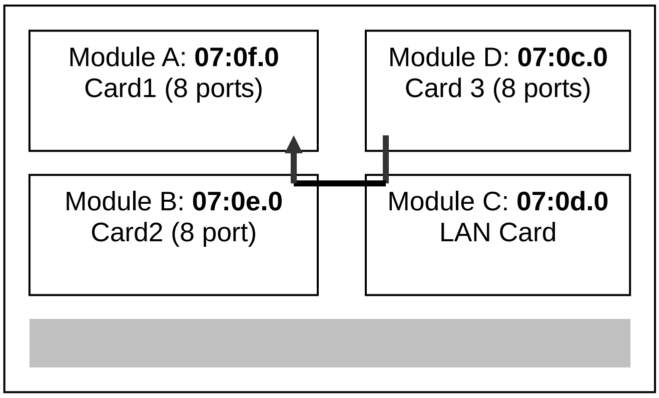
1. Make sure the DA-710 computer is turned off.
2. Insert the three modules on the rear panel of the DA-710.
3. Turn on the DA-710 computer.
4. When the system has rebooted, udev daemon will detect these modules and generate many files under **/dev/ttyMn**. (n will range from 0 to 23, 24 ports in total for 3 serial expansion modules)
5. Use the **dmesg** command to view more information for the serial port configuration. See the following figure for details.



```
[    6.318937] Found MOXA CP-118U series board(BusNo=7,DevNo=12)
[    6.318947] ACPI: PCI Interrupt 0000:07:0c.0[A] -> GSI 19 (level, low) -> IRQ 19
[    6.318977]          ttyM0 - ttyM7    max. baud rate = 921600 bps.
[    6.319153] mxser_probe[878]5011:4480
[    6.319155] Found MOXA CP-118U series board(BusNo=7,DevNo=14)
[    6.319161] ACPI: PCI Interrupt 0000:07:0e.0[A] -> GSI 17 (level, low) -> IRQ 17
[    6.319185]          ttyM8 - ttyM15    max. baud rate = 921600 bps.
[    6.319353] mxser_probe[878]5011:4480
[    6.319355] Found MOXA CP-118U series board(BusNo=7,DevNo=15)
[    6.319360] ACPI: PCI Interrupt 0000:07:0f.0[A] -> GSI 16 (level, low) -> IRQ 16
[    6.319385]          ttyM16 - ttyM23    max. baud rate = 921600 bps.
```

However, please note that the order of the PCI device addresses will be decided by the sequence of Module D, Module C, Module B and Module A. See the following for details.

- **07:0c.0 (Card3) : ttyM0~ttyM7**
- **07:0d.0 LAN CARD**
- **07:0e.0 (Card2): ttyM8~ttyM15**
- **07:0f .0 (Card1): ttyM16 ~ttyM23**



6. You may also use the **lspci** command to view more information on each PCI slot.

```
root@Moxa:~#lspci
07:0c.0 Serial controller: Moxa Technologies Co Ltd CP118U (8-port RS-232/422/485 Smart Universal PCI)
07:0d.0 PCI bridge: Tundra Semiconductor Corp. Device 8100 (rev 01)
07:0e.0 Serial controller: Moxa Technologies Co Ltd CP118U (8-port RS-232/422/485 Smart Universal PCI)
07:0f.0 Serial controller: Moxa Technologies Co Ltd CP118U (8-port RS-232/422/485 Smart Universal PCI)
```

7. To change the order of device nodes and module names, you can write a rule in **/etc/udev/rules.d/96-moxa.rules**

```
#CARD 0
KERNEL=="ttyM16", SYMLINK="ttyN0"
KERNEL=="ttyM17", SYMLINK="ttyN1"
KERNEL=="ttyM18", SYMLINK="ttyN2"
KERNEL=="ttyM19", SYMLINK="ttyN3"
KERNEL=="ttyM20", SYMLINK="ttyN4"
KERNEL=="ttyM21", SYMLINK="ttyN5"
KERNEL=="ttyM22", SYMLINK="ttyN6"
KERNEL=="ttyM23", SYMLINK="ttyN7"

#CARD 1
KERNEL=="ttyM8", SYMLINK="ttyN8"
KERNEL=="ttyM9" , SYMLINK="ttyN9"
KERNEL=="ttyM10", SYMLINK="ttyN10"
KERNEL=="ttyM11", SYMLINK="ttyN11"
KERNEL=="ttyM12", SYMLINK="ttyN12"
KERNEL=="ttyM13", SYMLINK="ttyN13"
KERNEL=="ttyM14", SYMLINK="ttyN14"
KERNEL=="ttyM15", SYMLINK="ttyN15"

#CARD 2
KERNEL=="ttyM0", SYMLINK="ttyN16"
KERNEL=="ttyM1", SYMLINK="ttyN17"
KERNEL=="ttyM2", SYMLINK="ttyN18"
KERNEL=="ttyM3", SYMLINK="ttyN19"
```

```
KERNEL=="ttyM4", SYMLINK="ttyN20"
KERNEL=="ttyM5", SYMLINK="ttyN21"
KERNEL=="ttyM6", SYMLINK="ttyN22"
```

Serial Port Operation Mode

The serial port expansion module has 8 serial ports named COM1 to COM8. The ports support RS-232, RS-422, 4-wire RS-485, and 4-wire RS-485 operation modes with baudrate settings up to 921600 bps.

By default, the serial interface is set to RS-232. You can use the **setinterface** command to change the serial port operation mode, as indicated below:

setinterface device-node [interface-no]

device-node: /dev/ttyMn; n = 0,1,2,...
interface-no: [see following table]:

interface-no	**Operation Mode**
None	Display current setting
0	RS-232
1	2-wire RS-485
2	RS-422
3	4-wire RS-485

For example, use the following commands to set **/dev/ttyM0** to RS-422:

```
MOXA:~# setinterface /dev/ttyM0 2
MOXA:~# setinterface /dev/ttyM0
Now setting is RS422 interface.
MOXA:~#
```

Telnet/FTP Server

In addition to supporting Telnet client/server and FTP client/server, the DA-710-LX also supports SSH and sftp client/server. To enable or disable the Telnet/ftp server, you need to edit the file **/etc/inetd.conf**.

1. Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Type **# cd /etc** to change the directory.

```
MOXA:~# cd /etc
```

3. Type **# vi inetd.conf** to edit the configuration file.

```
MOXA:/etc# vi inetd.conf
```

Enabling the Telnet/FTP Server

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

```
discard dgram udp wait root /bin/discard
discard stream tcp nowait root /bin/discard
telnet stream tcp nowait root /bin/telnetd
ftp stream tcp nowait root /bin/ftpd -l
```

Disabling the Telnet/FTP Server

Disable the daemon by typing "**#**" in front of the first character of the row to comment out the line. For example, to disable the **FTP** server, use the following commands:

```
discard dgram udp wait root /bin/discard
discard stream tcp nowait root /bin/discard
telnet stream tcp nowait root /bin/telnetd
#ftp stream tcp nowait root /bin/ftpd -l
```

After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

DNS Client

The DA-710-LX supports DNS client (but not DNS server). To set up DNS client, you need to edit three configuration files: **/etc/hostname, /etc/resolv.conf,** and **/etc/nsswitch.conf**.

/etc/hostname

1. Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Edit **/etc/hostname**:

```
MOXA:~# vi /etc/hostname
MOXA
```

3. After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

4. Re-configure the hostname.

```
MOXA:~# /etc/init.d/hostname.sh start
```

5. Check the new hostname.

```
MOXA:~# hostname
```

/etc/resolv.conf

This is the most important file that you need to edit when using DNS. For example, before you using **# ntpdate time.stdtime.gov.tw** 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 (assuming the DNS server's IP address is 168.95.1.1):

nameserver 168.95.1.1

```
MOXA:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16
nameserver 168.95.1.1
nameserver 140.115.1.31
nameserver 140.115.236.10
MOXA:/etc#
```

/etc/nsswitch.conf

This file defines the sequence of files, **/etc/hosts** or **/etc/resolv.conf**, to be read to resolve the IP address.

The **hosts** line in **/etc/nsswitch.conf** means use **/etc/host** first and DNS service to resolve the address.

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch
functionality.
# If you have the `glibc-doc-reference' and `info' packages
installed, try:
# `info libc "Name Service Switch"' for information about this
file.

passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Apache Web Server

Default Homepage

The Apache web server’s main configuration file is **/etc/apache2/sites-available/default**, with the default homepage located at **/var/www/apache2-default/index.html**.

Save your own homepage to the following directory:

/var/www/apache2-default

Save your CGI page to the following directory:

/var/www/apache2-default/cgi-bin/

Before you modify the homepage, use a browser (such as Microsoft Internet Explore or Mozilla Firefox) from your PC to test if the Apache web server is working. Type the LAN1 IP address in the browser’s address box to open the homepage. For example, if the default IP address 192.168.3.127 is still active, type:

http://192.168.3.127/

To test the default CGI page, type:

http://192.168.3.127/cgi-bin/w3mmail.cgi

Disabling the CGI Function

The CGI function is enabled by default. If you want to disable the function, modify the file **/etc/apache2/sites-available/default**.

1. Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

2. Type **# vi /etc/apache2/sites-available/default** to edit the configuration file.

```
MOXA:/etc# vi /etc/apache2/sites-available/default
#ScriptAlias /cgi-bin/ /var/www/apache2-default/cgi-bin/
#<Directory "/var/www/apache2 default/cgi-bin/">
#   AllowOverride None
#   Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
#   #Order allow,deny
#   Order deny,allow
#   Allow from all
#</Directory>
```

3. After you finish writing or modifying the code, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

4. Re-start the apache server.

```
MOXA:~# /etc/init.d/apache2 restart
```

ATTENTION

When you develop your own CGI application, make sure your CGI file is executable.

Saving Web Pages to a USB Storage Device

Some applications may have web pages that take up a lot of memory space. This section describes how to save web pages to the USB mass storage device, and then configure the Apache web server's DocumentRoot to open these pages. The files used in this example can be downloaded from Moxa's website.

1. Prepare the web pages and then save the pages to the USB storage device. Click on the following link to download the web page test suite: **http://www.w3.org/MarkUp/Test/HTML401.zip**.
2. Uncompress the zip file to your desktop PC, and then use FTP to transfer it to the DA-710-LX's **/media/usb0** directory.
3. Mount the root file system with write permission.

```
MOXA:~# mount -o remount,rw /dev/hda1 /
```

4. Type **# vi /etc/apache2/sites-available/default** to edit the configuration file.

```
MOXA:/etc# vi /etc/apache2/sites-available/default
```

5. Change the DocumentRoot directory to the USB storage directory **/media/usb0/www**.

```
...
<VirtualHost *:80>
...
...
       DocumentRoot /media/usb0/www
       <Directory />
              Options FollowSymLinks
              AllowOverride None
       </Directory>
...
...
       ScriptAlias /cgi-bin/ /media/usb0/www/cgi-bin/
       <Directory "/media/usb0/www/cgi-bin/">
              AllowOverride None
              Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
              Order allow,deny
              Allow from all
       </Directory>
...
</VirtualHost>
...
<VirtualHost *:443>
...
...
       DocumentRoot /media/usb0/www
       <Directory />
              Options FollowSymLinks
              AllowOverride None
       </Directory>
...
...
       ScriptAlias /cgi-bin/ /media/usb0/www/cgi-bin/
       <Directory "/media/usb0/wwwz/cgi-bin/">
              AllowOverride None
              Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
              Order allow,deny
              Allow from all
       </Directory>
...
</VirtualHost>
```

6. Use the following commands to restart the Apache web server:

```
#cd /etc/init.d
#./apache2 restart
```

7. Open your browser and connect to the DA-710-LX by typing the current LAN1 IP address in the browser's address box.
8. After finishing modification or writing, remember to execute "umount /" to change the root directory back to Read-only mode.

```
MOXA:~# umount /
```

9. Re-start the apache server.

```
MOXA:~# /etc/init.d/apache2 restart
```

ATTENTION

Visit the Apache website at http://httpd.apache.org/docs/ for more information about setting up Apache servers.

IPTABLES

IPTABLES is an administrative tool for setting up, maintaining, and inspecting the Linux kernel's IP packet filter rule tables. Several different tables are defined, with each table containing built-in chains and user-defined chains.

Each chain is a list of rules that apply to a certain type of packet. Each rule specifies what to do with a matching packet. A rule (such as a jump to a user-defined chain in the same table) is called a **target**.

The DA-710-LX supports three types of IPTABLES: Filter tables, NAT tables, and Mangle tables.

Filter Table—includes three chains:

INPUT chain
OUTPUT chain
FORWARD chain

NAT Table—includes three chains:

PREROUTING chain—transfers the destination IP address (DNAT).

POSTROUTING chain—works after the routing process and before the Ethernet device process to transfer the source IP address (SNAT).

OUTPUT chain—produces local packets.

Sub-tables

Source NAT (SNAT)—changes the first source packet IP address.

Destination NAT (DNAT)—changes the first destination packet IP address.

MASQUERADE—a special form for SNAT. If one host can connect to the Internet, then the other computers that connect to this host can connect to the Internet when the computer does not have an actual IP address.

REDIRECT—a special form of DNAT that re-sends packets to a local host independent of the destination IP address.

Mangle Table—includes two chains

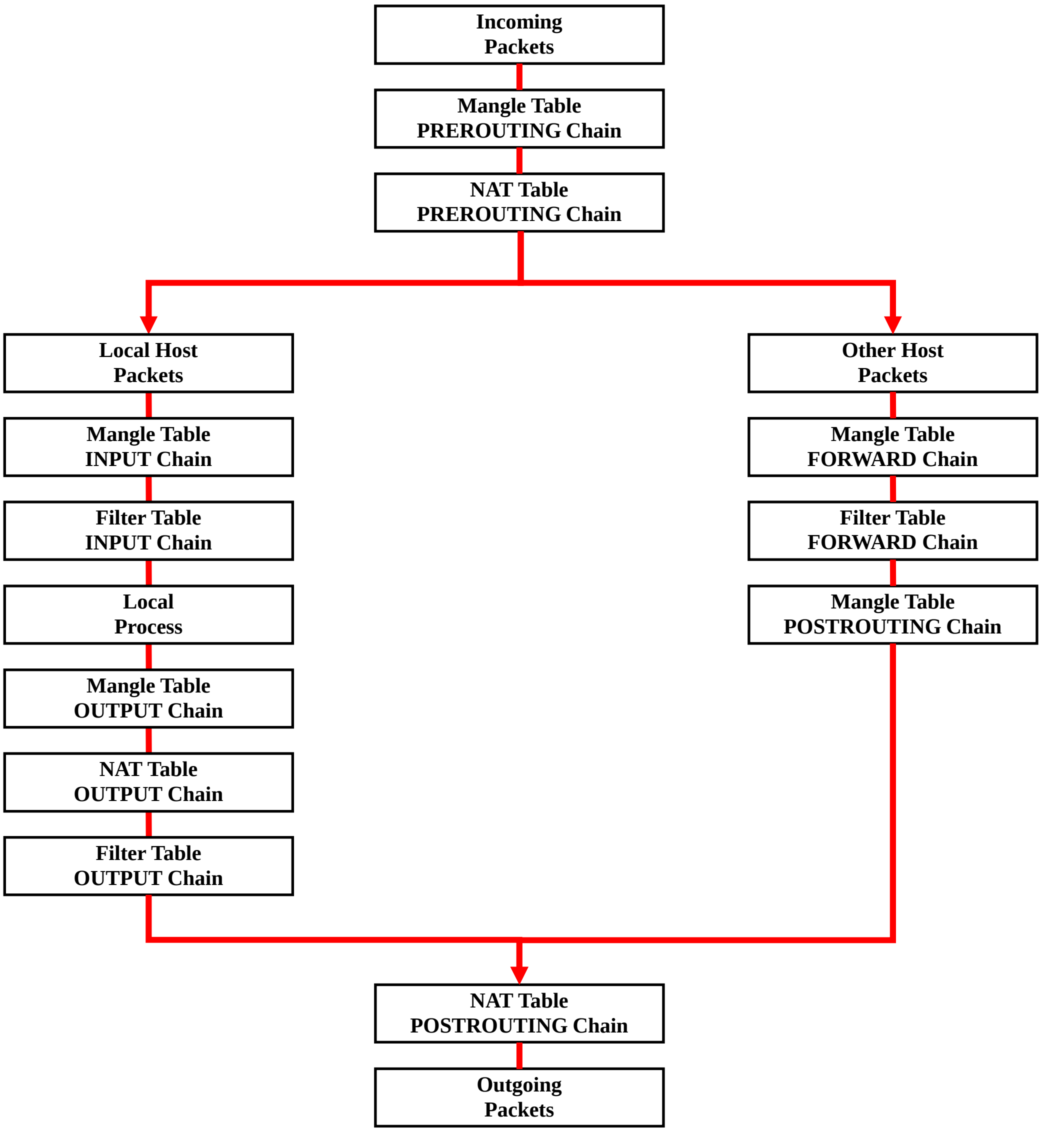
PREROUTING chain—pre-processes packets before the routing process.

OUTPUT chain—processes packets after the routing process.

Mangle tables can have one of three extensions—TTL, MARK, TOS.

IPTABLES Hierarchy

The following figure shows the IPTABLES hierarchy.


Incoming Packets
Mangle Table PREROUTING Chain
NAT Table PREROUTING Chain
Local Host Packets
Mangle Table INPUT Chain
Filter Table INPUT Chain
Local Process
Mangle Table OUTPUT Chain
NAT Table OUTPUT Chain
Filter Table OUTPUT Chain
Other Host Packets
Mangle Table FORWARD Chain
Filter Table FORWARD Chain
Mangle Table POSTROUTING Chain
NAT Table POSTROUTING Chain
Outgoing Packets

IPTABLES Modules

The DA-710-LX supports the following sub-modules. Be sure to use the module that matches your application.

arptable_filter.ko	arp_tables.ko	arpt_mangle.ko	ip_conntrack_amanda.ko
ip_conntrack_ftp.ko	ip_conntrack_h323.ko	ip_conntrack_irc.ko	ip_conntrack.ko
ip_conntrack_netbios_ns.ko	ip_conntrack_netlink.ko	ip_conntrack_pptp.ko	ip_conntrack_proto_sctp.ko
ip_conntrack_sip.ko	ip_conntrack_tftp.ko	ip_nat_amanda.ko	ip_nat_ftp.ko
ip_nat_h323.ko	ip_nat_irc.ko	ip_nat.ko	ip_nat_pptp.ko
ip_nat_sip.ko	ip_nat_snmp_basic.ko	ip_nat_tftp.ko	ip_queue.ko
iptable_filter.ko	iptable_mangle.ko	iptable_nat.ko	iptable_raw.ko
ip_tables.ko	ipt_addrtype.ko	ipt_ah.ko	ipt_CLUSTERIP.ko
ipt_dscp.ko	ipt_DSCP.ko	ipt_ecn.ko	ipt_ECN.ko
ipt_hashlimit.ko	ipt_iprange.ko	ipt_LOG.ko	ipt_MASQUERADE.ko
ipt_NETMAP.ko	ipt_owner.ko	ipt_recent.ko	ipt_REDIRECT.ko
ipt_REJECT.ko	ipt_SAME.ko	ipt_TCPMSS.ko	ipt_tos.ko
ipt_TOS.ko	ipt_ttl.ko	ipt_TTL.ko	ipt_ULOG.ko

The basic syntax to enable and load an IPTABLES module is as follows:

```
# lsmod
# modprobe ip_tables
# modprobe iptable_filter
#modprobe iptable_mangle
#modprobe iptable_nat
```

Use **lsmod** to check if the **ip_tables** module has already been loaded in the DA-710-LX. Use **modprobe** to insert and enable the module.

Use **iptables, iptables-restore, iptables-save** to maintain the database.

ATTENTION

IPTABLES plays the role of packet filtering or NAT. Be careful when setting up the IPTABLES rules. If the rules are not correct, remote hosts that connect via a LAN or PPP may be denied. We recommend using the VGA console to set up the IPTABLES. Click on the following links for more information about IPTABLES.

http://www.linuxguruz.com/iptables/
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

Since the IPTABLES command is very complex, to illustrate the IPTABLES syntax we have divided our discussion of the various rules into three categories: Observe and erase chain rules, Define policy rules, and Append or delete rules.

Observe and Erase Chain Rules

Usage:

# iptables [-t tables] [-L] [-n]

-t tables: Table to manipulate (default: 'filter'); example: nat or filter.
-L [chain]: List List all rules in selected chains. If no chain is selected, all chains are listed.
-n: Numeric output of addresses and ports.

# iptables [-t tables] [-FXZ]

-F: Flush the selected chain (all the chains in the table if none is listed).
-X: Delete the specified user-defined chain.
-Z: Set the packet and byte counters in all chains to zero.

Examples:

# iptables -L -n

In this example, since we do not use the -t parameter, the system uses the default "filter" table. Three chains are included: INPUT, OUTPUT, and FORWARD. INPUT chains are accepted automatically, and all connections are accepted without being filtered.

```
# iptables –F
# iptables –X
# iptables -Z
```

Define Policy for Chain Rules

Usage:

# iptables [-t tables] [-P] [INPUT, OUTPUT, FORWARD, PREROUTING, OUTPUT, POSTROUTING] [ACCEPT, DROP]

-P: Set the policy for the chain to the given target.
INPUT: For packets coming into the DA-710-I-LX.
OUTPUT: For locally-generated packets.
FORWARD: For packets routed out through the DA-710-I-LX.
PREROUTING: To alter packets as soon as they come in.
POSTROUTING: To alter packets as they are about to be sent out.

Examples:

```
#iptables –P INPUT DROP
#iptables –P OUTPUT ACCEPT
#iptables –P FORWARD ACCEPT
#iptables –t nat –P PREROUTING ACCEPT
#iptables –t nat –P OUTPUT ACCEPT
#iptables -t nat –P POSTROUTING ACCEPT
```

In this example, the policy accepts outgoing packets and denies incoming packets.

Append or Delete Rules

Usage:

```
# iptables [-t table] [-AI] [INPUT, OUTPUT, FORWARD] [-io interface] [-p tcp, udp, icmp, all] [-s IP/network] [--sport ports] [-d IP/network] [--dport ports] –j [ACCEPT. DROP]
```

-A: Append one or more rules to the end of the selected chain.
-I: Insert one or more rules in the selected chain as the given rule number.
-i: Name of an interface via which a packet is going to be received.
-o: Name of an interface via which a packet is going to be sent.
-p: The protocol of the rule or of the packet to check.
-s: Source address (network name, host name, network IP address, or plain IP address).
--sport: Source port number.
-d: Destination address.
--dport: Destination port number.
-j: Jump target. Specifies the target of the rules; i.e., how to handle matched packets.

For example, ACCEPT the packet, DROP the packet, or LOG the packet.

Examples:

Example 1: Accept all packets from the lo interface.

```
# iptables –A INPUT –i lo –j ACCEPT
```

Example 2: Accept TCP packets from 192.168.0.1.

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.0.1 –j ACCEPT
```

Example 3: Accept TCP packets from Class C network 192.168.1.0/24.

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.1.0/24 –j ACCEPT
```

Example 4: Drop TCP packets from 192.168.1.25.

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.1.25 –j DROP
```

Example 5: Drop TCP packets addressed for port 21.

```
# iptables –A INPUT –i eth0 –p tcp --dport 21 –j DROP
```

Example 6: Accept TCP packets from 192.168.0.24 to DA-710-I-LX's port 137, 138, 139

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.0.24 --dport 137:139 –j ACCEPT
```

Example 7: Log TCP packets that visit DA-710-I-LX's port 25.

```
# iptables –A INPUT –i eth0 –p tcp --dport 25 –j LOG
```

Example 8: Drop all packets from MAC address 01:02:03:04:05:06.

```
# iptables –A INPUT –i eth0 –p all –m mac --mac-source 01:02:03:04:05:06 –j DROP
```

ATTENTION

In Example 8, remember to issue the command **# modprobe ipt_mac** first to load the module **ipt_mac**.

NAT (Network Address Translation)

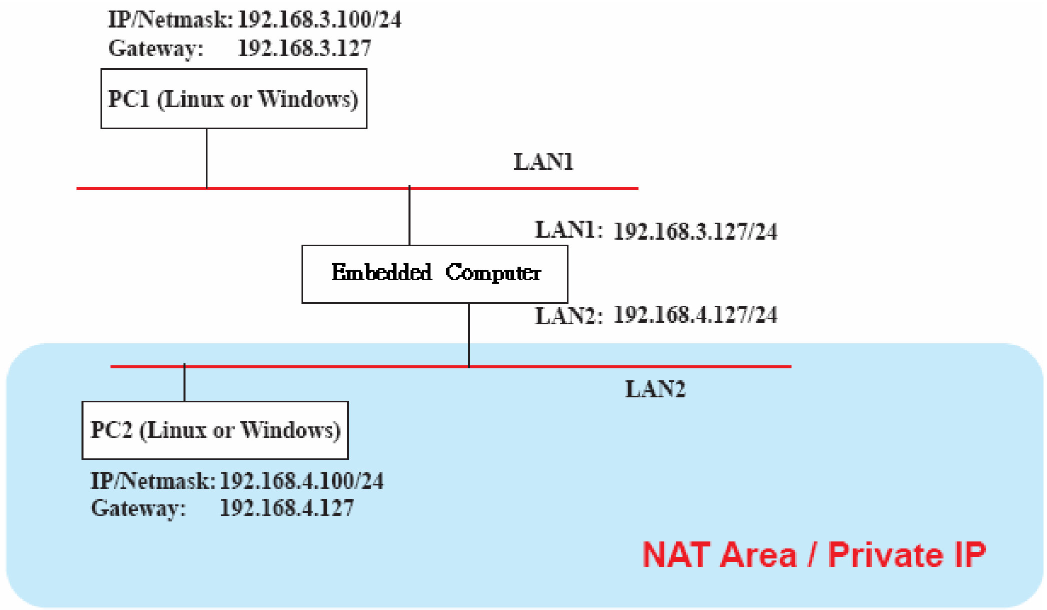
The NAT (Network Address Translation) protocol translates IP addresses used on one network into IP addresses used on a connecting network. One network is designated the inside network and the other is the outside network. Typically, the DA-710-LX connects several devices on a network and maps local inside network addresses to one or more global outside IP addresses, and un-maps the global IP addresses on incoming packets back into local IP addresses.

ATTENTION

Click on the following links for more information about NAT.
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

NAT Example

The IP address of all packets leaving LAN1 are changed to **192.168.3.127** (you will need to load the module **ipt_MASQUERADE**):



```
#ehco 1 > /proc/sys/net/ipv4/ip_forward
#modprobe ipt_MASQUERADE
#iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE
```

Enabling NAT at Bootup

In most real world situations, you will want to use a simple shell script to enable NAT when the DA-710-LX boots up. The following script is an example.

```
#!/bin/bash
# If you put this shell script in the /home/nat.sh
# Remember to chmod 744 /home/nat.sh
# Edit the rc.local file to make this shell startup automatically.
# vi /etc/rc.local
# Add a line in the end of rc.local /home/nat.sh

EXIF= “eth0”    #This is an external interface for setting up a valid IP address.
EXNET= “192.168.4.0/24”    #This is an internal network address.

# Step 1. Insert modules.

# Here 2> /dev/null means the standard error messages will be dump to null device.

modprobe ip_tables    2> /dev/null
modprobe ip_nat_ftp    2> /dev/null
modprobe ip_nat_irc    2> /dev/null
modprobe ip_conntrack    2> /dev/null
modprobe ip_conntrack_ftp    2> /dev/null
modprobe ip_conntrack_irc    2> /dev/null

# Step 2. Define variables, enable routing and erase default rules.

PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
export PATH
echo “1” > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -F
/sbin/iptables -X
/sbin/iptables -Z
/sbin/iptables -F -t nat
/sbin/iptables -X -t nat
/sbin/iptables -Z -t nat
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT

# Step 3. Enable IP masquerade.
```

PPP (Point to Point Protocol)

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.

Modem/PPP access is almost identical to connecting directly to a network through the DA-710-LX’s Ethernet port. Since PPP is a peer-to-peer system, the DA-710-LX can also use PPP to link two networks (or a local network to the Internet) to create a Wide Area Network (WAN).

ATTENTION

Click on the following links for more information about PPP.

http://tldp.org/HOWTO/PPP-HOWTO/index.html
http://axion.physics.ubc.ca/ppp-linux.html

Connecting to a PPP Server over a Simple Dial-up Connection

The following command is used to connect to a PPP server by modem. Use this command for old ppp servers that prompt for a login name (replace “username” with the correct name) and password (replace “password” with the correct password). Note that “debug crtscts” and “defaultroute 192.1.1.17” are optional.

```
#pppd connect ‘chat -v ““ ATDT5551212 CONNECT ““’ ogin: username word: password’
/dev/ttyM0 115200 debug crtscts modem defaultroute 192.1.1.17
```

If the PPP server does not prompt for the username and password, the command should be entered as follows. Replace “username” with the correct username and replace “password” with the correct password.

```
#pppd connect ‘chat -v ““ ATDT5551212 CONNECT ““’ user username password password
/dev/ttyM0 115200 crtscts modem
```

The pppd options are described below:

connect ‘chat etc...’ This option gives the command to contact the PPP server. The **chat** program is used to dial a remote computer. The entire command is enclosed in single quotes because pppd expects a one-word argument for the **connect** option. The options for **chat** are given below:

-v verbose mode; log what we do to syslog

“ “ Double quotes—don’t wait for a prompt, but instead do ... (note that you must include a space after the second quotation mark)

ATDT5551212 Dial the modem, and then ...

CONNECT Wait for an answer.

“ “ Send a return (null text followed by the usual return)

ogin: username word: password
Log in with username and password.

Refer to the chat man page, chat.8, for more information about the **chat** utility.

/dev/ Specify the callout serial port.

115200 The baud rate.

debug Log status in syslog.

crtscts Use hardware flow control between computer and modem (at 115200 this is a must).

modem Indicates that this is a modem device; pppd will hang up the phone before and after making the call.

defaultroute Once the PPP link is established, make it the default route; if you have a PPP link to the Internet, this is probably what you want.

192.1.1.17 This is a degenerate case of a general option of the form x.x.x.x:y.y.y.y. Here x.x.x.x is the local IP address and y.y.y.y is the IP address of the remote end of the PPP connection. If this option is not specified, or if just one side is specified, then x.x.x.x defaults to the IP address associated with the local machine’s hostname (located in /**etc/hosts**), and y.y.y.y is determined by the remote machine.

Connecting to a PPP Server over a Hard-wired Link

If a username and password are not required, use the following command (note that **noipdefault** is optional):

#pppd connect ‘chat –v“ “ “ “ ’ noipdefault /dev/ttyM0 19200 crtscts

If a username and password is required, use the following command (note that **noipdefault** is optional, and root is both the username and password):

#pppd connect ‘chat –v“ “ “ “ ’ user root password root noipdefault /dev/ttyM0 19200 crtscts

Checking the Connection

Once you have set up a PPP connection, there are some steps you can take to test the connection. First, type:

#/sbin/ifconfig

Depending on your distribution, the command might be located elsewhere. After executing the command, you should be able to see all of the network interfaces that are UP.

ppp0 should be one of them, and you should recognize the first IP address as your own and the **P-t-P address** (point-to-point address, the address of your server). The output is similar to the following:

```
lo      Link encap Local Loopback
        inet addr 127.0.0.1   Bcast 127.255.255.255 Mask
255.0.0.0
        UP LOOPBACK RUNNING   MTU 2000   Metric 1
        RX packets 0 errors 0 dropped 0 overrun 0

ppp0    Link encap Point-to-Point Protocol
        inet addr 192.76.32.3   P-t-P 129.67.1.165 Mask
255.255.255.0
        UP POINTOPOINT RUNNING   MTU 1500   Metric 1
        RX packets 33 errors 0 dropped 0 overrun 0
        TX packets 42 errors 0 dropped 0 overrun 0
```

Now, type:

#ping z.z.z.z

where z.z.z.z is the address of your name server. The output is similar to the following:

```
MOXA:~# ping 129.67.1.165
PING 129.67.1.165 (129.67.1.165): 56 data bytes
64 bytes from 129.67.1.165: icmp_seq=0 ttl=225 time=268 ms
64 bytes from 129.67.1.165: icmp_seq=1 ttl=225 time=247 ms
64 bytes from 129.67.1.165: icmp_seq=2 ttl=225 time=266 ms
^C
--- 129.67.1.165 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 247/260/268 ms
MOXA:~#
```

Try typing:

#netstat -nr

This should show three routes similar to the following:

```
Kernel routing table
Destination   Gateway   Genmask         Flags   Metric  Ref Use
iface
129.67.1.165  0.0.0.0   255.255.255.255 UH      0       0   6
ppp0
127.0.0.0     0.0.0.0   255.0.0.0       U       0       0   0 lo
0.0.0.0       129.67.1.165  0.0.0.0     UG      0       0   6298
ppp0
```

If your output looks similar but does not have the "destination 0.0.0.0" line (which refers to the default route used for connections), you may have run pppd without the **defaultroute** option. At this point, you can try using Telnet, ftp, or finger, bearing in mind that you will have to use numeric IP addresses unless you have configured **/etc/resolv.conf** correctly.

Setting up a Machine for Incoming PPP Connections

Method 1: pppd dial-in with pppd commands

This first example applies to using a modem, and requiring authorization with a username and password.

#pppd /dev/ttyM0 115200 crtscts modem 192.168.16.1:192.168.16.2 login auth

You should also add the following line to the file **/etc/ppp/pap-secrets**:

***   *   ""   ***

The first star (*) lets everyone login. The second star (*) lets every host connect. The pair of double quotation marks ("") indicates that the file **/etc/passwd** can be used to check the password. The last star (*) is to let any IP connect.

The following example does not check the username and password:

# pppd/dev/ttyM0 115200 crtscts modem 192.168.16.1:192.168.16.2

Method 2: pppd dial-in with pppd script

Configure a dial-in script **/etc/ppp/peer/dialin**

```
# You usually need this if there is no PAP authentication
noauth
#auth
#login

# The chat script (be sure to edit that file, too!)
init "/usr/sbin/chat -v -f /etc/ppp/ppp-ttyM0.chat"

# Set up routing to go through this PPP link
defaultroute

# Default modem (you better replace this with /dev/ttySx!)
/dev/ttyM0

# Speed
115200

# Keep modem up even if connection fails
persist
crtscts
modem
192.168.16.1:192.168.16.2
debug
-detach
```

Configure the chat script **/etc/ppp/ppp-ttyM0.chat**

```
SAY     'Auto Answer ON\n'
''      ATS0=1
```

Start the **pppd** dial-in service.

```
# pppd call dialin
```

ATTENTION

If you hope to have auto dial-in service, you can respawn the dial-in service in **/etc/inittab**.

```
MOXA:~# mount –o remount,rw /dev/hda1 /
MOXA:~# echo “p0:2345:respawn:pppd call dialin” >>
/etc/inittab
MOXA:~# umount /
```

PPPoE

The following procedure is for setting up PPPoE:

1. Connect the DA-710-LX's LAN port to an ADSL modem with a cross-over cable, HUB, or switch.
2. Log in to the DA-710-LX as the root user.
3. Edit the file **/etc/ppp/chap-secrets** and add the following:
   "username@hinet.net" * "password" *

```
# Secrets for authentication using CHAP
# client        server  secret                  IP addresses

# PPPOE example, if you want to use it, you need to unmark it
and modify it
"username@hinet.net"   *      "password"      *
```

username@hinet.net is the username obtained from the ISP to log in to the ISP account. **password** is the corresponding password for the account.

4. Edit the file **/etc/ppp/pap-secrets** and add the following:
   "username@hinet.net" * "password" *

```
# ATTENTION: The definitions here can allow users to login
without a
# password if you don't use the login option of pppd! The
mgetty Debian
# package already provides this option; make sure you don't
change that.

# INBOUND connections

# Every regular user can use PPP and has to use passwords
from /etc/passwd
*      hostname       ""      *
"username@hinet.net"    *      "password"      *

# UserIDs that cannot use PPP at all. Check your /etc/passwd
and add any
# other accounts that should not be able to use pppd!
guest   hostname       "*"     -
master  hostname       "*"     -
root    hostname       "*"     -
support hostname       "*"     -
stats   hostname       "*"     -

# OUTBOUND connections
```

username@hinet.net is the username obtained from the ISP to log in to the ISP account. **password** is the corresponding password for the account.

5. Edit the file **/etc/ppp/options** and add the following line:
   plugin rp-pppoe

```
# received.  Note: it is not advisable to use this option
with the persist
# option without the demand option.  If the active-filter
option is given,
# data packets which are rejected by the specified activity
filter also
# count as the link being idle.
#idle <n>

# Specifies how many seconds to wait before re-initiating the
link after
# it terminates.  This option only has any effect if the
persist or demand
# option is used.  The holdoff period is not applied if the
link was
# terminated because it was idle.
#holdoff <n>

# Wait for up n milliseconds after the connect script
finishes for a valid
# PPP packet from the peer.  At the end of this time, or when
a valid PPP
# packet is received from the peer, pppd will commence
negotiation by
# sending its first LCP packet.  The default value is 1000 (1
second).
# This wait period only applies if the connect or pty option
is used.
#connect-delay <n>

# Load the pppoe plugin
plugin rp-pppoe.so

# ---<End of File>---
```

6. If you use LAN1 to connect to the ADSL modem, add file **/etc/ppp/options.eth0**. If you use LAN2 to connect to the ADSL modem, then add **/etc/ppp/options.eth1**, etc.

```
name username@hinet.net
mtu 1492
mru 1492
defaultroute
noipdefault
~
~
"/etc/ppp/options.eth0" 5 lines, 67 characters
```

Type your username (the one you set in the **/etc/ppp/pap-secrets and /etc/ppp/chap-secrets** files) after the **name** option. You may add other options as desired.

7. Set up DNS.

If you are using DNS servers supplied by your ISP, edit the file **/etc/resolv.conf** by adding the following lines of code:

nameserver ip_addr_of_first_dns_server
nameserver ip_addr_of_second_dns_server

For example:

nameserver 168.95.1.1
nameserver 139.175.10.20

```
MOXA:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16
nameserver 168.95.1.1
nameserver 139.175.10.20
nameserver 140.115.1.31
nameserver 140.115.236.10
MOXA:/etc#
```

8. Use the following command to create a **pppoe** connection:
   #pppd eth0

The ADSL modem is connected to the **LAN1** port, which is named **eth0**. If the ADSL modem is connected to **LAN2**, use **eth1**, etc.

9. Type **#ifconfig ppp0** to check if the connection is OK. If the connection is OK, you should see the IP address of ppp0. Use **#ping** to test the IP address.

```
ppp0    Link encap Point-to-Point Protocol
        inet addr 192.76.32.3   P-t-P 129.67.1.165 Mask
255.255.255.0
        UP POINTOPOINT RUNNING   MTU 1500   Metric 1
        RX packets 33 errors 0 dropped 0 overrun 0
        TX packets 42 errors 0 dropped 0 overrun 0
```

10. If you want to disconnect it, use the kill command to kill the **pppd** process.

NFS (Network File System) Client

The Network File System (NFS) is used to mount a disk partition on a remote machine (as if it were on a local hard drive), allowing fast, seamless sharing of files across a network. NFS allows users to develop applications for the DA-710-LX without worrying about the amount of disk space that will be available. The DA-710-LX supports only NFS client protocol.

ATTENTION

Click on the following links for more information about NFS.

http://www.ietf.org/rfc/rfc1213.txt
http://www.faqs.org/rfcs/rfc1317.html

The following procedures illustrate how to mount a remote NFS Server.

1. Scan the NFS Server's shared directory.

```
#showmount   –e   HOST
```

showmount:	Show the mount information of an NFS Server
-e:	Show the NFS Server's export list.
HOST:	IP address or DNS address

2. Establish a mount point on the NFS Client site.

```
#mkdir   –p   /home/nfs/public
```

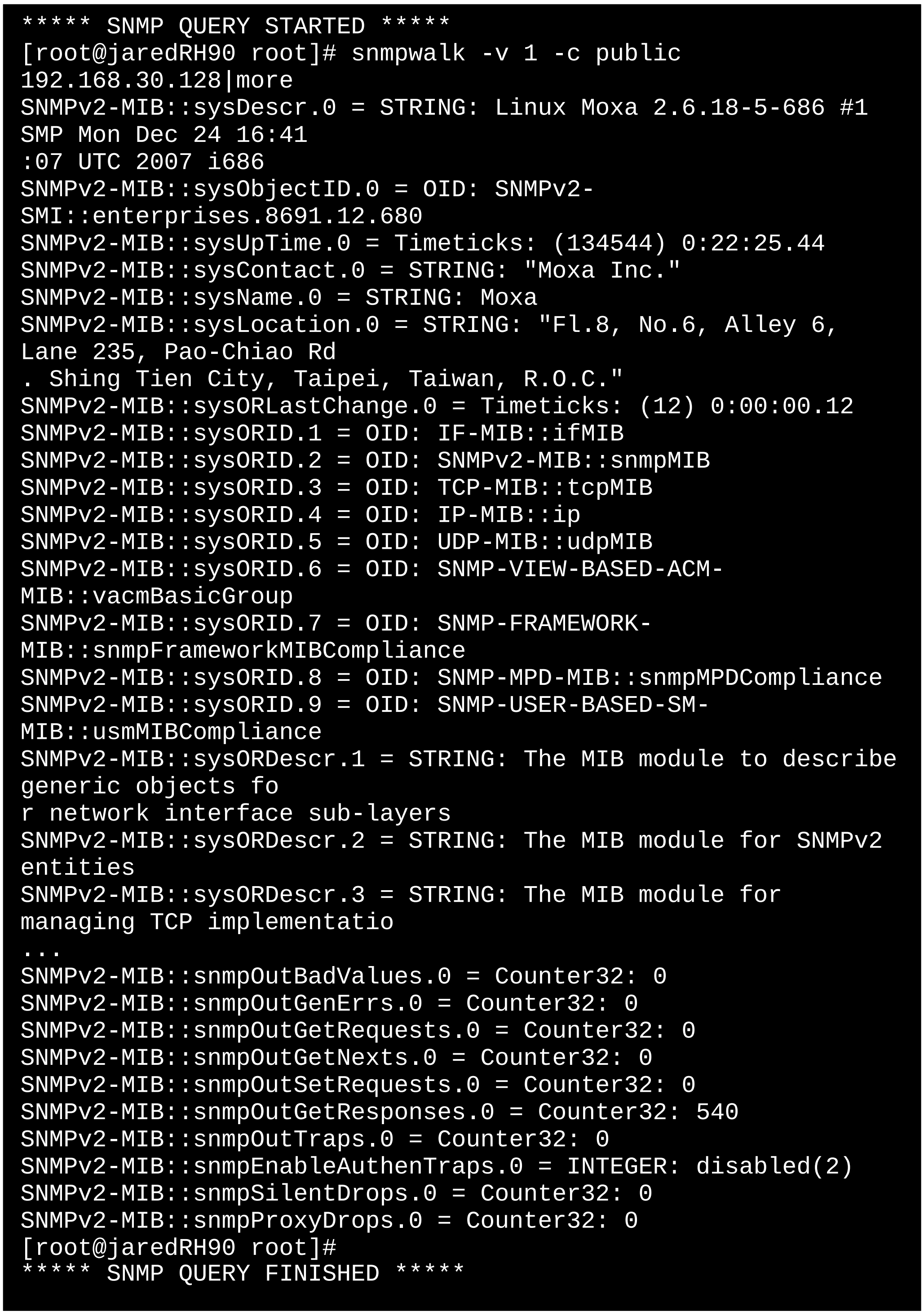
3. Mount the remote directory to a local directory.

```
#mount -t nfs -o nolock 192.168.3.100:/home/public /home/nfs/public
```

This is where 192.168.3.100 is the example IP address of the NFS server.

SNMP (Simple Network Management Protocol)

The DA-710-LX comes with the SNMP V1 (Simple Network Management Protocol) agent software pre-installed. It supports RFC1317 **RS-232 like group** and **RFC 1213 MIB-II**. The following shows example shows an SNMP agent responding to a query from the SNMP browser on the host site:



```
***** SNMP QUERY STARTED *****
[root@jaredRH90 root]# snmpwalk -v 1 -c public
192.168.30.128|more
SNMPv2-MIB::sysDescr.0 = STRING: Linux Moxa 2.6.18-5-686 #1
SMP Mon Dec 24 16:41
:07 UTC 2007 i686
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-
SMI::enterprises.8691.12.680
SNMPv2-MIB::sysUpTime.0 = Timeticks: (134544) 0:22:25.44
SNMPv2-MIB::sysContact.0 = STRING: "Moxa Inc."
SNMPv2-MIB::sysName.0 = STRING: Moxa
SNMPv2-MIB::sysLocation.0 = STRING: "Fl.8, No.6, Alley 6,
Lane 235, Pao-Chiao Rd
. Shing Tien City, Taipei, Taiwan, R.O.C."
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (12) 0:00:00.12
SNMPv2-MIB::sysORID.1 = OID: IF-MIB::ifMIB
SNMPv2-MIB::sysORID.2 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.3 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.4 = OID: IP-MIB::ip
SNMPv2-MIB::sysORID.5 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.6 = OID: SNMP-VIEW-BASED-ACM-
MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.7 = OID: SNMP-FRAMEWORK-
MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.8 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.9 = OID: SNMP-USER-BASED-SM-
MIB::usmMIBCompliance
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module to describe
generic objects fo
r network interface sub-layers
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for SNMPv2
entities
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for
managing TCP implementatio
...
SNMPv2-MIB::snmpOutBadValues.0 = Counter32: 0
SNMPv2-MIB::snmpOutGenErrs.0 = Counter32: 0
SNMPv2-MIB::snmpOutGetRequests.0 = Counter32: 0
SNMPv2-MIB::snmpOutGetNexts.0 = Counter32: 0
SNMPv2-MIB::snmpOutSetRequests.0 = Counter32: 0
SNMPv2-MIB::snmpOutGetResponses.0 = Counter32: 540
SNMPv2-MIB::snmpOutTraps.0 = Counter32: 0
SNMPv2-MIB::snmpEnableAuthenTraps.0 = INTEGER: disabled(2)
SNMPv2-MIB::snmpSilentDrops.0 = Counter32: 0
SNMPv2-MIB::snmpProxyDrops.0 = Counter32: 0
[root@jaredRH90 root]#
***** SNMP QUERY FINISHED *****
```

ATTENTION

Click on the following links for more information about **RFC1317 RS-232** like group and **RFC 1213 MIB-II**.

http://www.faqs.org/rfcs/rfc1317.html
http://www.ietf.org/rfc/rfc1213.txt

OpenVPN

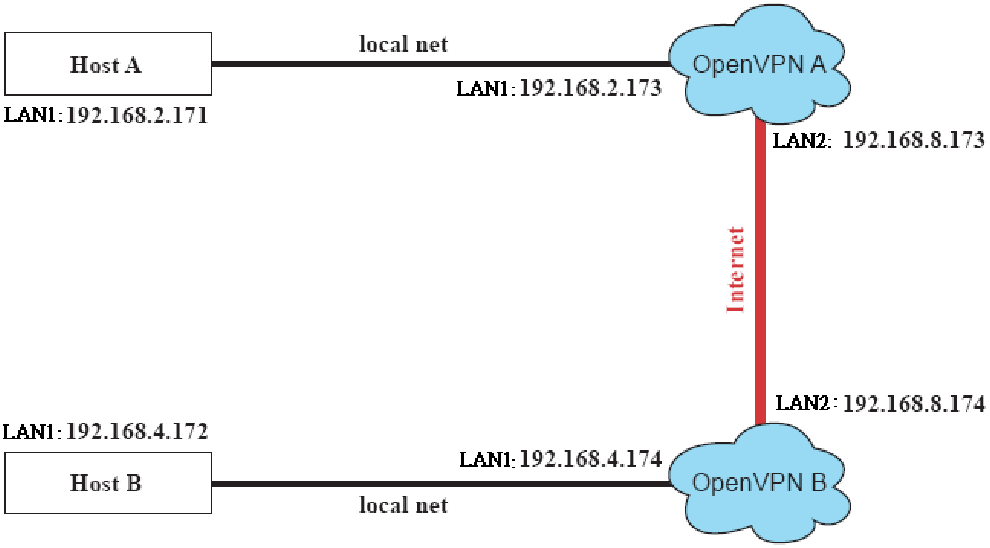
OpenVPN provides two types of tunnels for users to implement VPNS: **Routed IP Tunnels** and **Bridged Ethernet Tunnels**.

An Ethernet bridge is used to connect different Ethernet networks together. The Ethernets are bundled into one bigger, "logical" Ethernet. Each Ethernet corresponds to one physical interface (or port) that is connected to the bridge.

On each OpenVPN machine, you should carry out configurations in the **/etc/openvpn** directory, where script files and key files reside. Once established, all operations will be performed in that directory.

Ethernet Bridging for Private Networks on Different Subnets

1. Set up four machines, as shown in the following diagram.



Host A represents the machine that belongs to OpenVPN A, and Host B represents the machine that belongs to OpenVPN B. The two remote subnets are configured for a different range of IP addresses. When this configuration is moved to a public network, the external interfaces of the OpenVPN machines should be configured for static IPs, or connected to another device (such as a firewall or DSL box) first.

2. Generate a preset shared key by typing the command:
   # openvpn --genkey --secret secrouter.key

3. Copy the file that is generated to the OpenVPN machine:
   # scp /etc/openvpn/secrouter.key 192.168.8.174:/etc/openvpn

ATTENTION

A preshared key is located at **/etc/openvpn/secrouter.key**. You can use it for testing purposes. We suggest creating a new key for non-testing purpose.

4. On machine OpenVPN A, modify the remote address in the configuration file **/etc/openvpn/tap0-br.conf**.

```
# point to the peer
remote 192.168.8.174
dev tap0
port 1194
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
up /etc/openvpn/tap0-br.sh
#comp-lzo
```

5. Next, modify the routing table in the **/etc/openvpn/tap0-br.sh script** file.

```
#--------------------------Start---------------------------
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.4.0 netmask 255.255.255.0 dev br0
#----------------------------end---------------------------
```

And then configure the bridge interface in **/etc/openvpn/bridge**.

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"
eth_ip="192.168.8.173"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.174"
...
```

Start the bridge script file to configure the bridge interface.

# /etc/openvpn/bridge restart

6. On machine OpenVPN B, modify the remote address in configuration file **/etc/openvpn/tap0-br.conf**.

```
# point to the peer
remote 192.168.8.173
dev tap0
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
up /etc/openvpn/tap0-br.sh
#comp-lzo
```

7. Next modify the routing table in **/etc/openvpn/tap0-br.sh script** file.

```
#----------------------------------Start---------------------
--------
#!/bin/sh
# value after “-net” is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 dev br0
#---------------------------------- end ---------------------
--------
```

And then configure the bridge interface in **/etc/openvpn/bridge**.

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"
eth_ip="192.168.8.174"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.173"
...
```

Start the bridge script file to configure the bridge interface.

# /etc/openvpn/bridge restart

ATTENTION

Select cipher and authentication algorithms by specifying **cipher** and **auth**. To see which algorithms are available, type:

# openvpn --show-ciphers
# openvpn --show-auths

8. Start both OpenVPN peers on machine OpenVPN A and OpenVPN B.
   # openvpn --config /etc/openvpn/tap0-br.conf&

If you see the line **Peer Connection Initiated with 192.168.8.173:5000**on each machine, the connection between OpenVPN machines has been established successfully on UDP port 5000.

ATTENTION

You can create link symbols to start the OpenVPN service at boot time:
# ln -sf /etc/init.d/openvpn /etc/rc2.d/S16openvpn

To stop the service, you should create these links:
# ln -sf /etc/init.d/openvpn /etc/rc0.d/K80openvpn
# ln -sf /etc/init.d/openvpn /etc/rc6.d/K80openvpn

9. On each OpenVPN machine, check the routing table by typing the command **# route**

```
Destination     Gateway Genmsk          Flags   Metric  Ref
    Use Iface
192.168.5.0     0.0.0.0 255.255.255.0   U       0       0    0
    eth2
192.168.4.0     0.0.0.0 255.255.255.0   U       0       0    0
    br0
192.168.3.0     0.0.0.0 255.255.255.0   U       0       0    0
    eth0
192.168.30.0    0.0.0.0 255.255.255.0   U       0       0    0
    eth3
192.168.8.0     0.0.0.0 255.255.255.0   U       0       0    0
    br0
```

Interface **eth1** and device **tap0** both connect to the bridging interface, and the virtual device **tun** sits on top of **tap0**. This ensures that all traffic coming to this bridge from internal networks connected to interface eth1 write to the TAP/TUN device that the OpenVPN program monitors. Once the OpenVPN program detects traffic on the virtual device, it sends the traffic to its peer.

10. To create an indirect connection to Host B from Host A, you need to add the following routing item:

    # route add –net 192.168.4.0 netmask 255.255.255.0 dev eth0

    To create an indirect connection to Host A from Host B, you need to add the following routing item:

    # route add –net 192.168.2.0 netmask 255.255.255.0 dev eth0

    Now ping Host B from Host A by typing:

    # ping 192.168.4.174

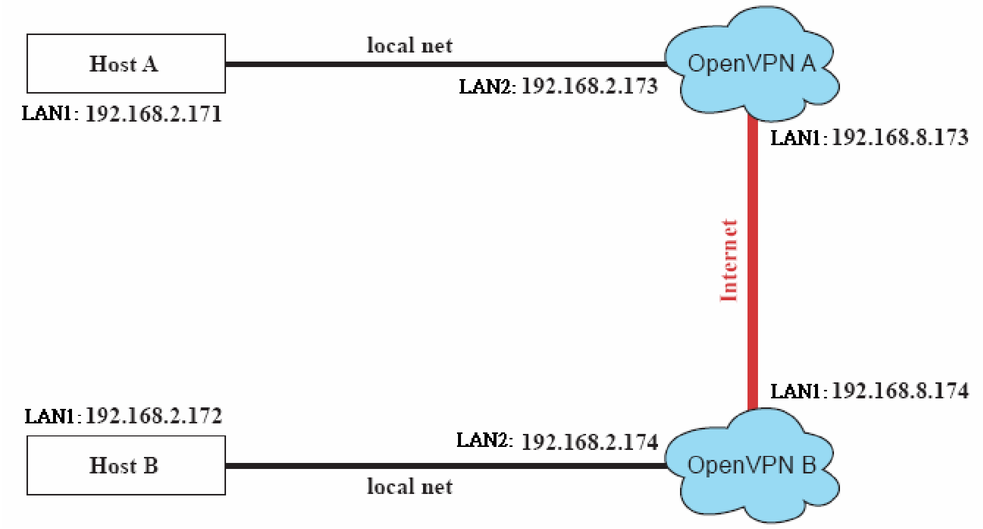
    A successful ping indicates that you have created a VPN system that only allows authorized users from one internal network to access users at the remote site. For this system, all data is transmitted by UDP packets on port 5000 between OpenVPN peers.

11. To shut down OpenVPN programs, type the command:

    # killall -TERM openvpn

Ethernet Bridging for Private Networks on the Same Subnet

1. Set up four machines, as shown in the following diagram.

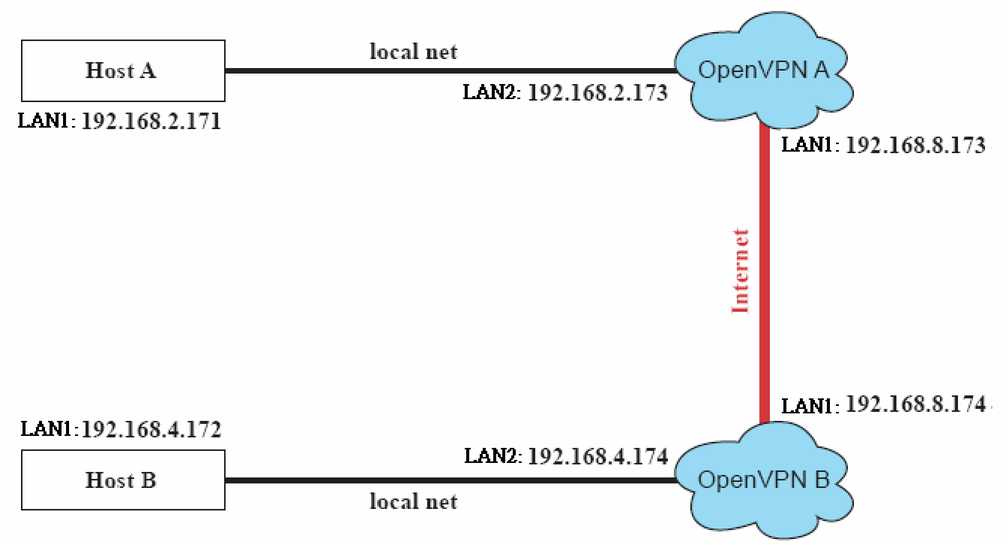


2. The configuration procedure is almost the same as for the previous example. The only difference is that you will need to comment out the parameter **up** in **/etc/openvpn/tap0-br.conf** of OpenVPN A and **/etc/openvpn/tap0-br.conf** of OpenVPN B.

```
# point to the peer
remote 192.168.8.174
dev tap0
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
#up /etc/openvpn/tap0-br.sh
#comp-lzo
```

Routed IP

1. Set up four machines, as shown in the following diagram.



2. On machine OpenVPN A, modify the remote address in configuration file **/etc/openvpn/tun.conf**.

```
# point to the peer
remote 192.168.8.174
dev tun
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
ifconfig 192.168.2.173 192.168.4.174
up /etc/openvpn/tun.sh
```

3. Next, modify the routing table in script file **/etc/openvpn/tun.sh**.

```
#---------------------------Start----------------------------
#!/bin/sh
# value after “-net” is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 gw $5
#----------------------------end-----------------------------
```

4. On machine OpenVPN B, modify the remote address in configuration file **/etc/openvpn/tun.conf**.

```
# point to the peer
remote 192.168.8.173
dev tun
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
ifconfig 192.168.4.174 192.168.2.173
up /etc/openvpn/tun.sh
```

And then modify the routing table in script file **/etc/openvpn/tun.sh**.

```
#-------------------------Start----------------------------
#!/bin/sh
# value after “-net” is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 gw $5
#--------------------------end-----------------------------
```

The first argument of parameter **ifconfig** is the local internal interface and the second argument is the internal interface at the remote peer.

$5 is the argument that the OpenVPN program passes to the script file. Its value is the second argument of **ifconfig** in the configuration file.

5. Check the routing table after you run the OpenVPN programs, by typing the command **# route**.

```
Destination     Gateway         Genmsk          Flags   Metric
    Ref Use Iface
192.168.4.174   *               255.255.255.255 UH      0
    0   0   tun0
192.168.4.0     192.168.4.174   255.255.255.0   UG      0
    0   0   tun0
192.168.2.0     *               255.255.255.0   U       0
    0   0   eth1
192.168.8.0     *               255.255.255.0   U       0
    0   0   eth0
```

4

Programmer Guide

This chapter covers the following topics:

❑ **Device API**
❑ **RTC (Real Time Clock)**
❑ **UART**
❑ **Digital I/O**
❑ **Programmable LEDs**

Device API

The DA-710 supports control devices with the **ioctl** system API. The interface is shown as below.

```
int ioctl(int d, int request,…);
```

Input:

<d> open device node return file handle

<request> argument in or out

Refer to desktop Linux's man page for detailed documentation:

```
#man ioctl
```

RTC (Real Time Clock)

The device node is located at **/dev/rtc**. The DA-710 supports standard Linux 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.

UART

The DA-710 supports standard Linux termios control. The Moxa UART Device only supports RS-232 and the normal tty device node is **/dev/ttyS0** and **/det/ttyS1**.

However, when installed with Moxa's DA Series Expansion Modules, it can support RS-232, RS-422, 2-wire RS-485, and 4-wire RS485, depending on the expansion module models. Our system will generate tty device node as **/dev/ttyM0 … /dev/ttyMn**. For example, if you have installed the 8-port DA-SP08-I-DB serial module, the device node will be **/dev/ttyM0 to /dev/ttyM7**.

To configure the serial ports, follow these steps.

1. You must include **"moxadevice.h**", which you can find in the folder \example\moxalib in CD.

```
#define RS232_MODE 0

#define RS485_2WIRE_MODE 1

#define RS422_MODE 2

#define RS485_4WIRE_MODE 3
```

2. Function: MOXA_SET_OP_MODE

```
int ioctl(fd, MOXA_SET_OP_MODE, &mode)
```

Description Set the interface mode. Argument 3 mode will pass to the UART device driver and change it.

3. Function: MOXA_GET_OP_MODE

```
int ioctl(fd, MOXA_GET_OP_MODE, &mode)
```

Description Get the interface mode. Argument 3 mode will return the interface mode.

There are two Moxa private ioctl control definitions for setting up special baudrates.

```
MOXA_SET_SPECIAL_BAUD_RATE
MOXA_GET_SPECIAL_BAUD_RATE
```

If you use this ioctl to set a special baudrate, the termios cflag will be B4000000, in which case the B4000000 define will be different. If the baudrate you get from termios (or from calling tcgetattr()) is B4000000, you must call ioctl with MOXA_GET_SPECIAL_BAUD_RATE to get the actual baudrate.

Example to set the baudrate

```
#include “moxadevice.h”
#include <termios.h>
struct termios term;
int fd, speed;
fd = open(“/dev/ttyM0”, O_RDWR);
tcgetattr(fd, &term);
term.c_cflag &= ~(CBAUD | CBAUDEX);
term.c_cflag |= B4000000;
tcsetattr(fd, TCSANOW, &term);
speed = 500000;
ioctl(fd, MOXA_SET_SPECIAL_BAUD_RATE, &speed);
```

Example to get the baudrate

```
#include “moxadevice.h”
#include <termios.h>
struct termios term;
int fd, speed;
fd = open(“/dev/ttyM0”, O_RDWR);
tcgetattr(fd, &term);
if ( (term.c_cflag & (CBAUD|CBAUDEX)) != B4000000 ) {
// follow the standard termios baud rate define
} else {
ioctl(fd, MOXA_GET_SPECIAL_BAUD_RATE, &speed);
}
```

Baudrate inaccuracy

Divisor = 921600/Target Baud Rate. (Only Integer part)

ENUM = 8 * (921600/Target - Divisor) (Round up or down)

Inaccuracy =((Target Baud Rate – 921600/(Divisor + (ENUM/8))) / Target Baud Rate)* 100%

E.g.,

To calculate 500000 bps

Divisor = 1, ENUM = 7,

Inaccuracy = 1.7%

* To work reliably, you should set inaccuracy under 2%.

Special Note

1. If the target baudrate is not a special baudrate (e.g. 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600), the termios cflag will be set to the same flag.
2. If you use **stty** to get the serial information, you will get speed equal to 0 for the special baudrate.

Digital I/O

Digital Output channels can be set to high or low. The channels are controlled by the function call **set_dout_state( )**. Use the digital input channels to detect the state change of the digital input signal. The DI channels can also be used to detect whether or not the state of a digital signal changes during a fixed period of time. This can be done by the function call, **set_din_event( )**.

Return error code definitions:

```
#define DIO_ERROR_PORT -1 // no such port

#define DIO_ERROR_MODE -2 // no such mode or state

#define DIO_ERROR_CONTROL -3 // open or ioctl fail

#define DIO_ERROR_DURATION -4 // The value of duration is not 0
or not in the range, 40 <= duration <= 3600000 milliseconds (1
hour)

#define DIO_ERROR_DURATION_20MS -5 // The value of duration must
be a multiple of 20 ms

#define DIO_OK 0
```

DIN and DOUT definitions:

```
#define DIO_HIGH 1

#define DIO_LOW 0
```

Moxa functions for DI/DO

Function	**int set_dout_state(int doport, int state)**
Description	Set the DOUT port to high or low state.
Input	<doport> The DOUT port you want to set. Port starts from 0 to 3 <state> Set high or low state; DIO_HIGH (1) for high, DIO_LOW (0) for low.
Output	none
Return	refer to the error code

Function	**int get_din_state(int diport, int *state)**
Description	Get the DIN port state
Input	<diport> The DIN port to get the state of. Port numbering is from 0 to 3 <state> Save the current state
Output	<state> DIO_HIGH (1) for high, DIO_LOW (0) for low
Return	Refer to the error code

Function	**int get_dout_state(int doport, int *state)**
Description	Get the DOUT port state
Input	<doport> The DOUT port to get the state of. <state> Save the current state.
Output	<state> DIO_HIGH (1) for high, DIO_LOW (0) for low
Return	Refer to the error code

Function	**int set_din_event(int diport, void (*func)(int diport), int mode, long int duration)**
Description	Set the DIN event when the state is changed from high to low or from low to high.
Input	<diport> The port that will be used to detect the DIN event. Port numbering is from 0 to 3. This value depends on your device. <(*func) (int diport)> Not NULL: Returns the call back function. When the event occurs, the call back function will be invoked. NULL: Clear this event <mode> DIN_EVENT_HIGH_TO_LOW (1): From high to low DIN_EVENT_LOW_TO_HIGH (0): From low to high DIN_EVENT_CLEAR (-1): Clear this event <duration> 0: Detect the din event DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH without duration Not 0: Detect the din event DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH with duration. Note: The value of "duration" must be a multiple of 20 milliseconds. The range of "duration" is 0, or 40 <= duration <= 3600000 milliseconds. The error of the measurement is 24 ms. For example, if the DIN duration is 200 ms, this event will be generated when the DIN pin stays in the same state for a time between 176 ms and 200 ms.
Output	None
Return	Refer to the error code

Function	**int get_din_event(int diport, int *mode, long int *duration)**
Description	To retrieve the DIN event configuration, including mode (DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH), and the value of "duration."
Input	<diport> Which DIN port you want to retrieve <mode> Save the set event. <duration> The duration the DIN port is kept in high or low state. - return to the current duration value of diport
Output	<mode> DIN_EVENT_HIGH_TO_LOW (1): From high to low DIN_EVENT_LOW_TO_HIGH(0): From low to high DIN_EVENT_CLEAR(-1): Clear this event <duration> The value of duration should be 0 or 40 <= duration <= 3600000 milliseconds.
Return	Refer to the error code

Special Note

1. You have to build the moxalib in advance for DI/DO. The moxalib is included in the folder **\example\moxalib** in CD.
2. Make sure to link the library **libmoxalib** for DI/DO programming, and include the header file **moxadevice.h**. Only one program at a time can use the DI/DO library.
3. Due to hardware limitation, you need to modify MIN_DURATION as 60 for DA-710.

Examples

DIO Program Source Code File Example

File Name: tdio.c

Description: This program connects DO1 to DI1, changes the digital output state to high or low by manual input, then detects and counts the state changed events from DI1.

```
#include   <stdio.h>
#include   <stdlib.h>
#ifdef NO_MOXADEVICE_HEADER
   #include   "moxadevice.h"
#else
   #include   <moxadevice.h>
#endif
#include   <fcntl.h>

/* Due to hardware limitation, MIN_DURATION should be 60 for DA710
*/
```

```
#define MIN_DURATION 40

static char *DataString[2]={"Low  ", "High "};
static void hightolowevent(int diport)
{
   printf("\nDIN port %d high to low.\n", diport);
}

static void lowtohighevent(int diport)
{
   printf("\nDIN port %d low to high.\n", diport);
}

int main(int argc, char * argv[])
{
   int    i, j, state, retval;
   unsigned long duration;

   while( 1 ) {
      printf("\nSelect a number of menu, other key to exit.   \n\
   1.set high to low event       \n\
   2.get now data.          \n\
   3.set low to high event       \n\
   4.clear event            \n\
   5.set high data.         \n\
   6.set low data.          \n\
   7. quit              \n\
   8. show event and duration            \n\
Choose : ");
   retval =0;
      scanf("%d", &i);
      if ( i == 1 ) { // set high to low event
         printf("Please keyin the DIN number : ");
         scanf("%d", &i);
         printf("Please input the DIN duration, this minimun
value must be over %d : ",MIN_DURATION);
```

```
            scanf("%lu", &duration);
            retval=set_din_event(i, hightolowevent,
DIN_EVENT_HIGH_TO_LOW, duration);
        } else if ( i == 2 ) {  // get now data
            printf("DIN data  : ");
            for ( j=0; j<MAX_DIN_PORT; j++ ) {
                get_din_state(j, &state);
                printf("%s", DataString[state]);
            }
            printf("\n");
            printf("DOUT data : ");
            for ( j=0; j<MAX_DOUT_PORT; j++ ) {
                get_dout_state(j, &state);
                printf("%s", DataString[state]);
            }
            printf("\n");
        } else if ( i == 3 ) {  // set low to high event
            printf("Please keyin the DIN number : ");
            scanf("%d", &i);
            printf("Please input the DIN duration, this minimun
value must be over %d : ",MIN_DURATION);
            scanf("%lu", &duration);
            retval = set_din_event(i, lowtohighevent,
DIN_EVENT_LOW_TO_HIGH, duration);
        } else if ( i == 4 ) {  // clear event
            printf("Please keyin the DIN number : ");
            scanf("%d", &i);
            retval=set_din_event(i, NULL, DIN_EVENT_CLEAR, 0);
        } else if ( i == 5 ) {  // set high data
            printf("Please keyin the DOUT number : ");
            scanf("%d", &i);
            retval=set_dout_state(i, 1);
        } else if ( i == 6 ) {  // set low data
            printf("Please keyin the DOUT number : ");
            scanf("%d", &i);
            retval=set_dout_state(i, 0);
        } else if ( i == 7 ) {  // quit
```

```
    break;
 } else if ( i == 8 ) {  // show event and duration
    printf("Event:\n");
    for ( j=0; j<MAX_DOUT_PORT; j++ ) {
       retval=get_din_event(j, &i, &duration);
       switch ( i ) {
       case DIN_EVENT_HIGH_TO_LOW :
          printf("(htl,%lu)", duration);
          break;
       case DIN_EVENT_LOW_TO_HIGH :
          printf("(lth,%lu)", duration);
          break;
       case DIN_EVENT_CLEAR :
          printf("(clr,%lu)", duration);
          break;
       default :
          printf("err " );
          break;
       }
    }
    printf("\n");
 } else {
    printf("Select error, please select again !\n");
 }
switch(retval)  {
       case DIO_ERROR_PORT:
          printf("DIO error port\n");
          break;
       case DIO_ERROR_MODE:
          printf("DIO error mode\n");
          break;
       case DIO_ERROR_CONTROL:
          printf("DIO error control\n");
          break;
       case DIO_ERROR_DURATION:
          printf("DIO error duratoin\n");
```

```
             case DIO_ERROR_DURATION_20MS:
                 printf("DIO error! The duratoin is not a multiple
of 20 ms\n");
                 break;
       }
   }

   return 0;
}
```

DIO Program Make File Example

```
include ../compile.mk
CC=$(PREFIX)gcc
STRIP=$(PREFIX)strip
AR=$(PREFIX)ar
LNAME=moxalib
all:   release
release: $(MOXALIB_OBJ)
       $(AR) rcs lib$(LNAME).a $(MOXALIB_OBJ)
%.o:%.c
       $(CC) -c $<
install:       lib$(LNAME).a
       cp -a lib$(LNAME).a $(MOXALIB_INSTALL_DIR)
       cp -a moxadevice.h /usr/local/arm-linux/include
       cp -a moxadevice.h /usr/local/arm-linux/arm-linux/include
clean:
       /bin/rm -f *.o *.a
```

Programmable LEDs

The DA-710 offers 4 programmable LED indicators for specific use. To configure these LED indicators, see the following steps:

1. Check if **moxa_pled.ko** module has been loaded.

```
MOXA: ~# lsmod | grep moxa_pled
moxa_pled                    2464     0
```

If **moxa_pled.ko** module has not been loaded, do the following command.

```
MOXA: ~# modprobe moxa_pled
```

2. See the following explanation.

*module name: moxa_pled.ko

*Usage:

echo bit_{pled1} bit_{pled2} bit_{pled3} bit_{pled4} > /dev/pled

bit value:

    1: enable

    0:disable

3. See the following example.

```
MOXA: ~# echo 1001 > /dev/pled
```

By doing so, you will enable the first and the fourth LED indicators.

5

System Recovery

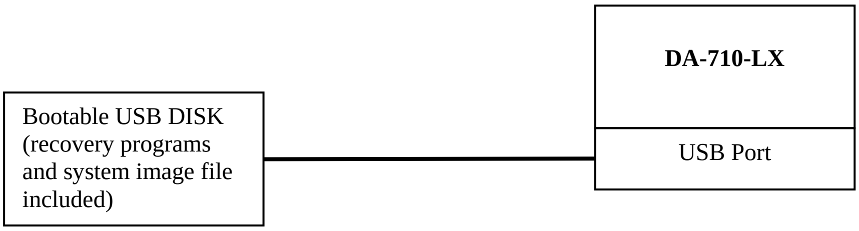
The DA-710-LX is installed with the Embedded Linux operating system, which is located in the Flash DOM (CompactFlash card) shipped with the DA-710-LX computer. Although it rarely happens, you may find on occasion that operating system files and/or the disk file system are damaged. This chapter describes how to recover the Linux operating system.

This chapter covers the following topics:

❑ **Recovery Environment**
❑ **Recovery Procedure**

Recovery Environment

The recovery environment includes the DA-710-LX embedded computer and a bootable USB disk with the recovery programs and system image file.


DA-710-LX
USB Port
Bootable USB DISK (recovery programs and system image file included)

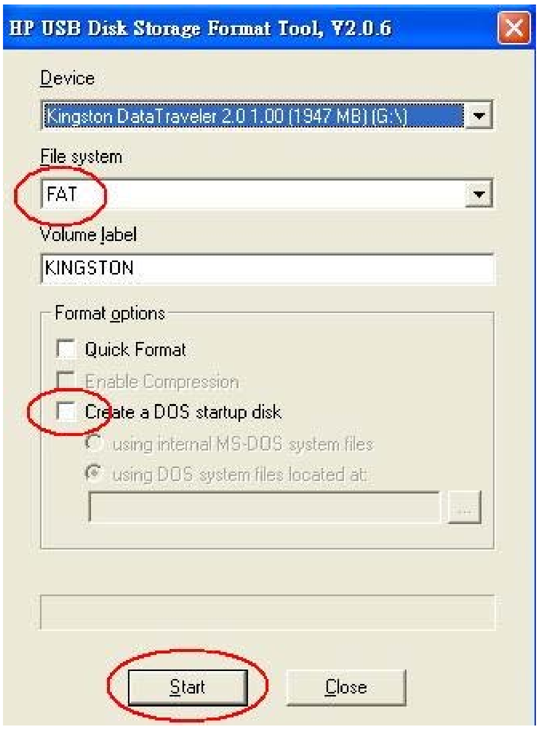
Recovery Procedure

Step 1: Format an Empty USB Disk.

a. Prepare a USB disk that has at least a 256 MB capacity.

b. Format your USB disk with the **HP USB Disk Format Tool**. Open the utility and select the device and FAT file system. You need empty disk only. DO NOT check the option **Create a DOS startup disk**.

c. Click **Start**.


HP USB Disk Storage Format Tool, V2.0.6

Device: Kingston DataTraveler 2.0 1.00 (1947 MB) (G:\)

File system: FAT

Volume label: KINGSTON

Format options

□ Quick Format

□ Enable Compression

□ Create a DOS startup disk

○ using internal MS-DOS system files

● using DOS system files located at:

ATTENTION

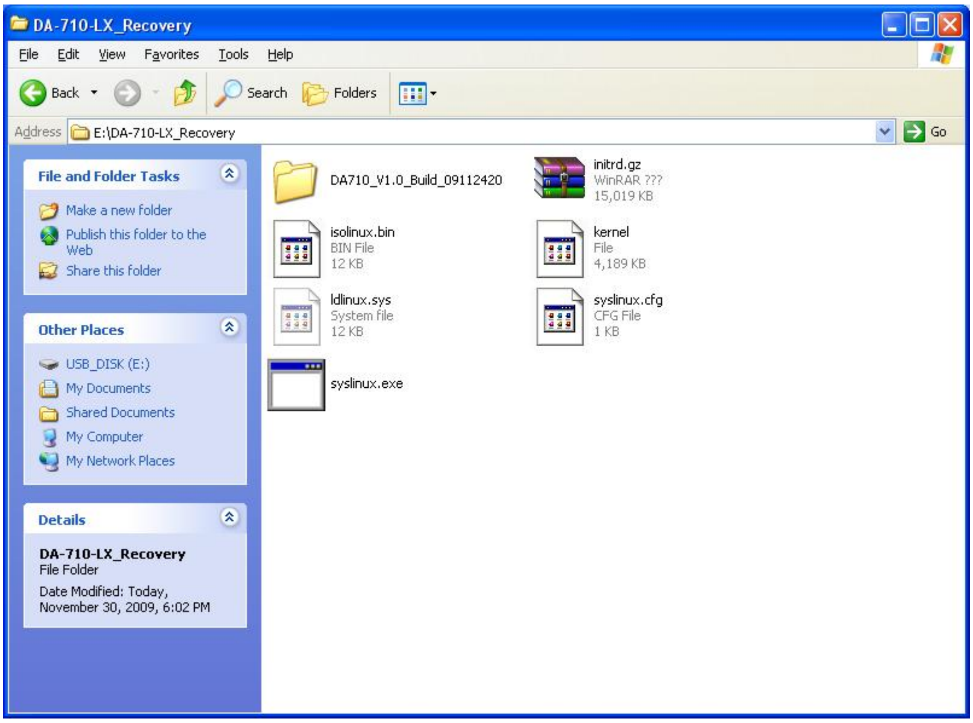
The HP USB Disk Storage Format Tool can be downloaded from many web sites. Do a search on **HP USB Disk Storage Format Tool** from any search engine to locate the tool.

Step 2: Create a Linux Bootable USB Disk.

a. You can find the **firmware** directory in the Recovery CD shipped with the DA-710-LX computer.

b. Configure Windows Explorer to show hidden files (including protected operating system files).

c. Copy all files in the **firmware** directory to the root directory of your USB disk.



DA-710-LX_Recovery

Address E:\DA-710-LX_Recovery

Name	Type	Size
DA710_V1.0_Build_09112420		
initrd.gz	WinRAR ???	15,019 KB
isolinux.bin	BIN File	12 KB
kernel	File	4,189 KB
ldlinux.sys	System file	12 KB
syslinux.cfg	CFG File	1 KB
syslinux.exe		

Details

DA-710-LX_Recovery
File Folder
Date Modified: Today, November 30, 2009, 6:02 PM

d. Open a DOS prompt and type **M:\syslinux.exe M:** to create a bootable Linux disk. In this example, M: is the USB Disk drive number.

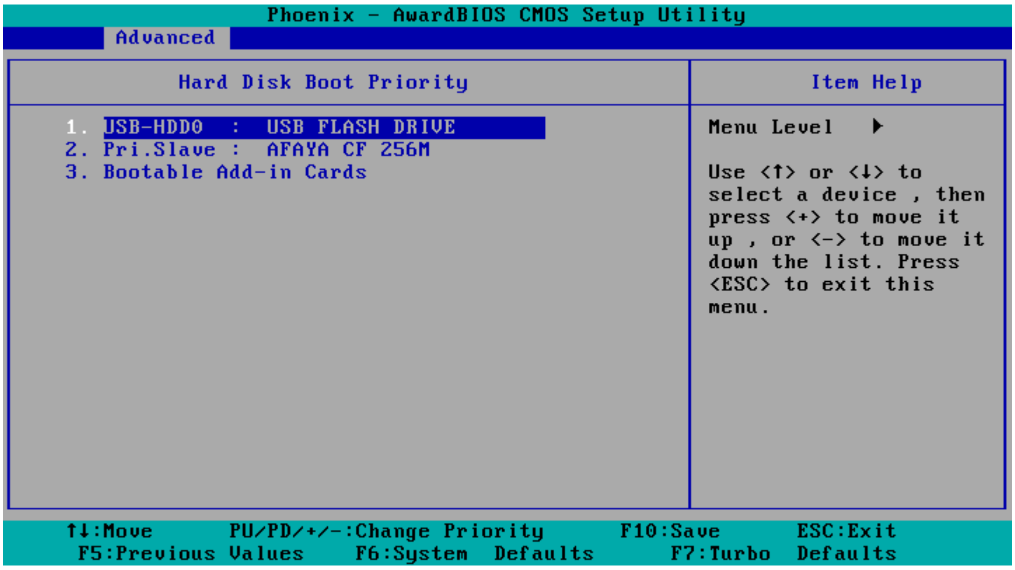
Step 3: Set up the BIOS to Boot from a USB Disk.

a. Insert the USB disk.

b. Power on and press **DEL** to enter the bios setup menu.

c. Select **Advanced → Hard Disk Boot Priority** and then press **Enter**.

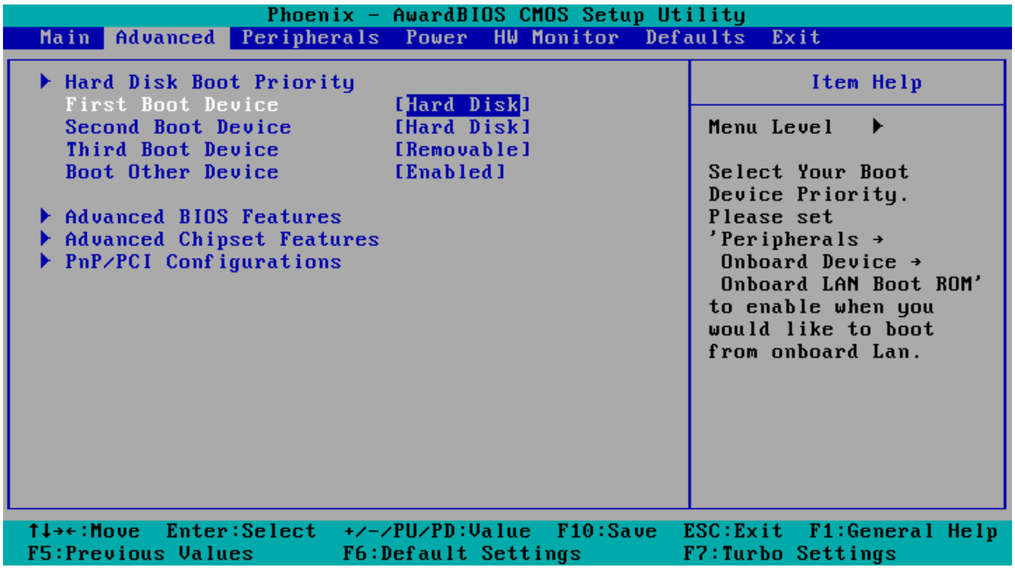
d. From the setup menu, use "↑" or "↓" to select the USB device



```
                   Phoenix - AwardBIOS CMOS Setup Utility
          Advanced

          Hard Disk Boot Priority                      Item Help

    1. USB-HDD0  :  USB FLASH DRIVE            Menu Level   ▶
    2. Pri.Slave :  AFAYA CF 256M
    3. Bootable Add-in Cards                   Use <↑> or <↓> to
                                               select a device , then
                                               press <+> to move it
                                               up , or <-> to move it
                                               down the list. Press
                                               <ESC> to exit this
                                               menu.

    ↑↓:Move      PU/PD/+/-:Change Priority      F10:Save      ESC:Exit
     F5:Previous Values      F6:System  Defaults      F7:Turbo  Defaults
```

e. Press "+" to move the selection up to the first priority, and press **Esc** to exit the setup menu.

f. Make sure the first boot device is **Hard Disk**. If not, press **Enter** to change it.



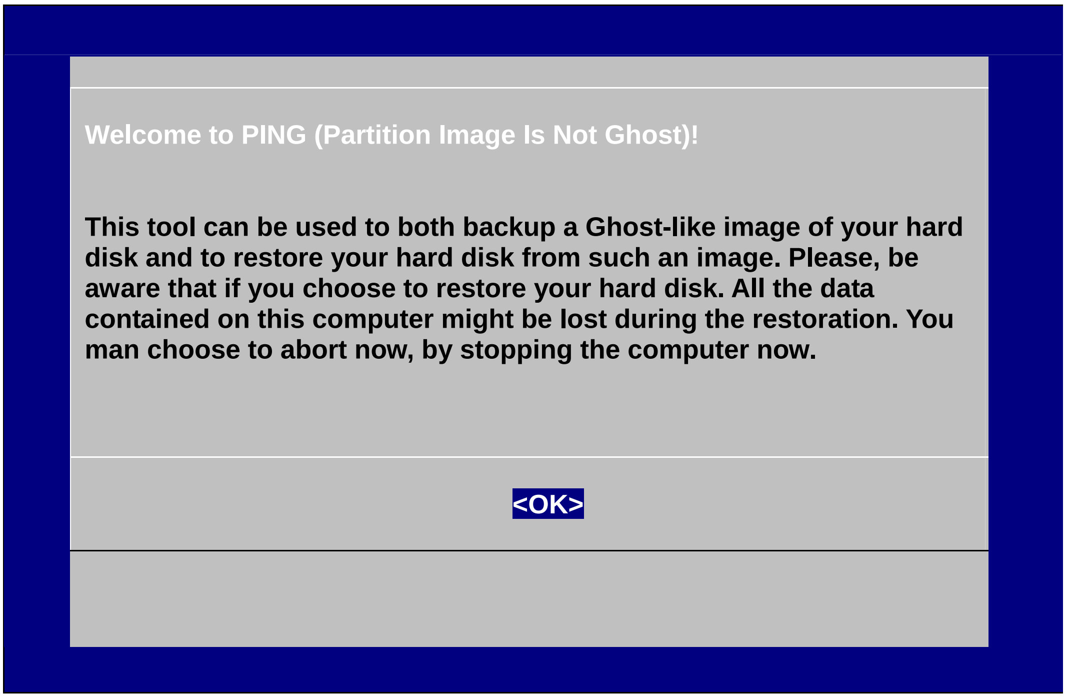
```
                   Phoenix - AwardBIOS CMOS Setup Utility
   Main  Advanced  Peripherals  Power  HW Monitor  Defaults  Exit

   ▶ Hard Disk Boot Priority                           Item Help
     First Boot Device           [Hard Disk]
     Second Boot Device          [Hard Disk]    Menu Level   ▶
     Third Boot Device           [Removable]
     Boot Other Device           [Enabled]      Select Your Boot
                                                Device Priority.
   ▶ Advanced BIOS Features                     Please set
   ▶ Advanced Chipset Features                  'Peripherals →
   ▶ PnP/PCI Configurations                      Onboard Device →
                                                 Onboard LAN Boot ROM'
                                                to enable when you
                                                would like to boot
                                                from onboard Lan.

  ↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
  F5:Previous Values        F6:Default Settings        F7:Turbo Settings
```

g. Select **Exit → Save & Exit Setup** and then press **Enter**.

h. Choose **Y** to save to the CMOS and then exit.

Step 4: Recover the Linux system from a USB Disk.

a. If the BIOS setup is correct, it will boot from the USB disk. Follow the steps below to set up recovery parameters.

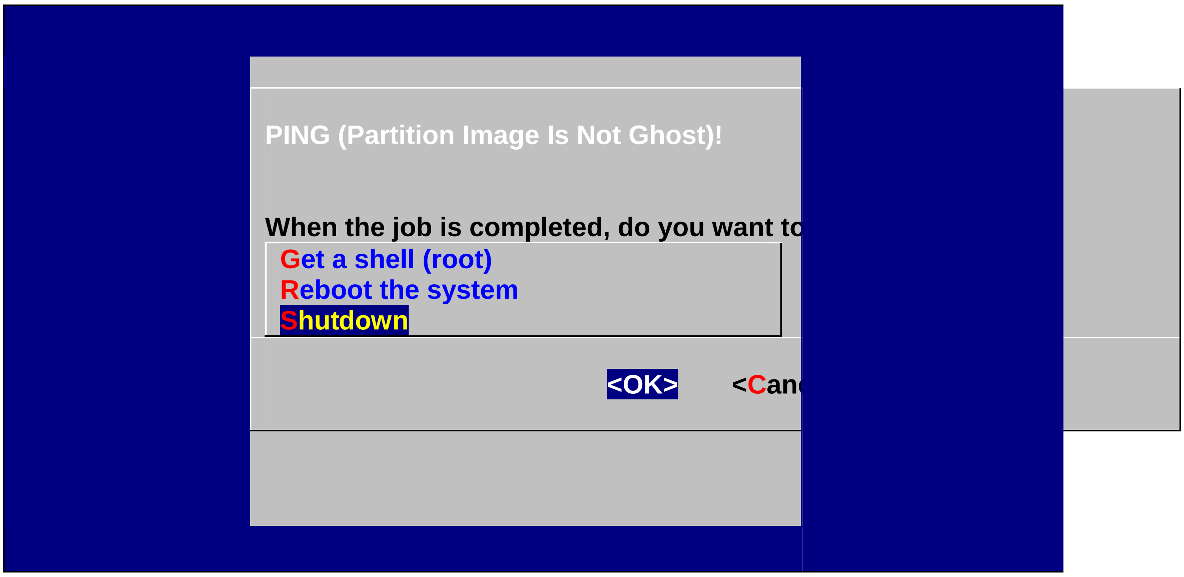


Welcome to PING (Partition Image Is Not Ghost)!

This tool can be used to both backup a Ghost-like image of your hard disk and to restore your hard disk from such an image. Please, be aware that if you choose to restore your hard disk. All the data contained on this computer might be lost during the restoration. You man choose to abort now, by stopping the computer now.

b. Choose **OK** to go to the next step.

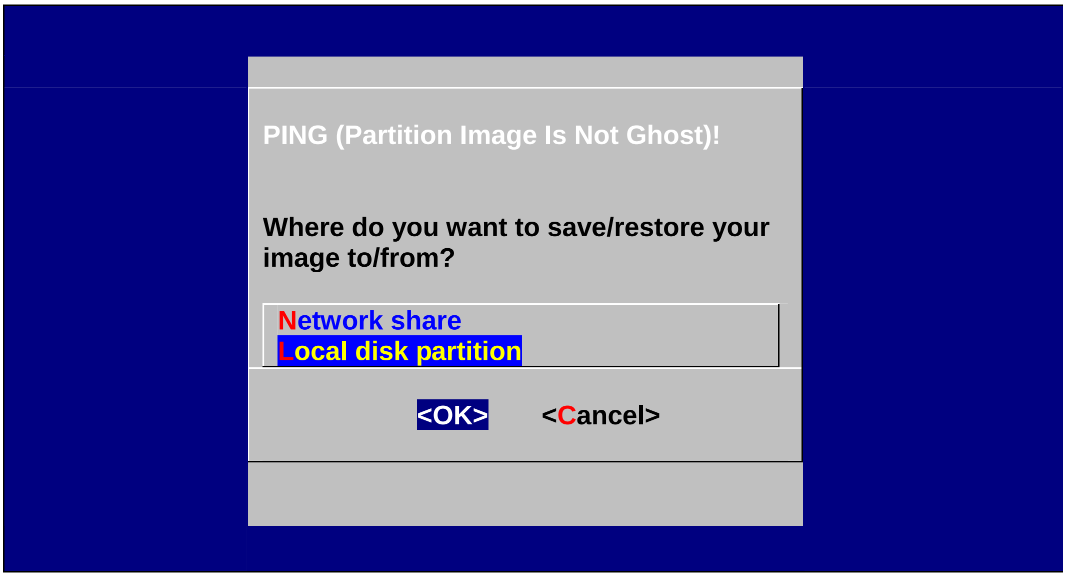
c. Choose shut down the DA-710-LX when the restoration is finished.



PING (Partition Image Is Not Ghost)!

When the job is completed, do you want to

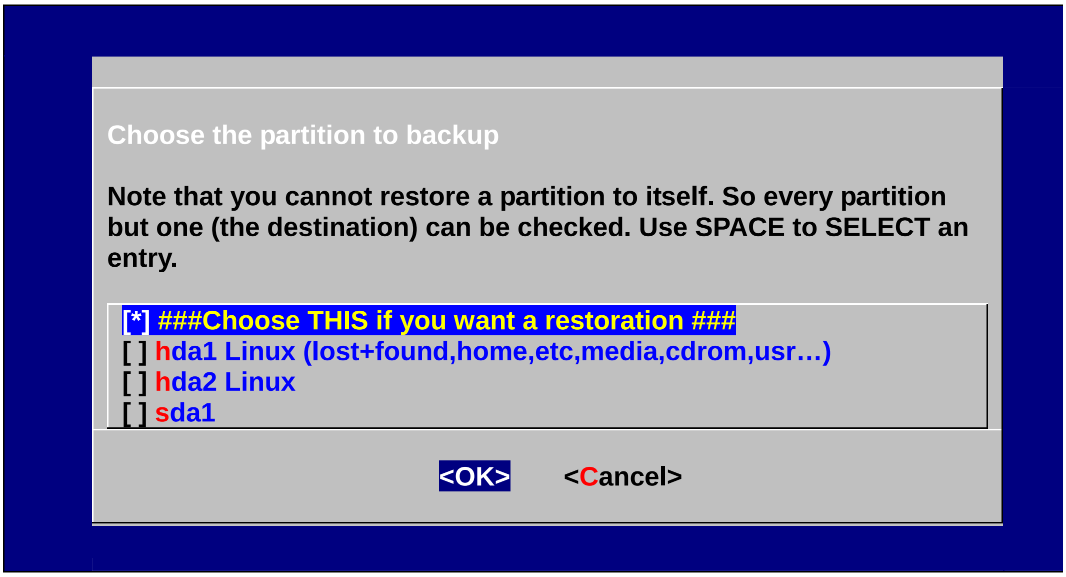
d. Choose restore image from **Local disk partition**.



PING (Partition Image Is Not Ghost)!

Where do you want to save/restore your image to/from?

e. Choose **### Choose THIS if you want a restoration ###**

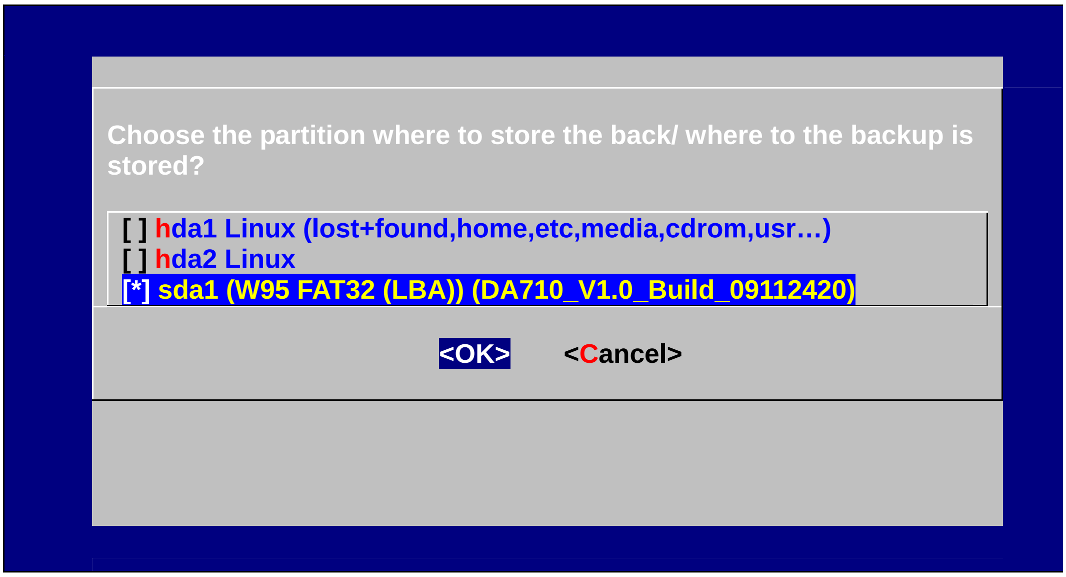


Choose the partition to backup

Note that you cannot restore a partition to itself. So every partition but one (the destination) can be checked. Use SPACE to SELECT an entry.

[*] ###Choose THIS if you want a restoration ###
[ ] hda1 Linux (lost+found,home,etc,media,cdrom,usr…)
[ ] hda2 Linux
[ ] sda1

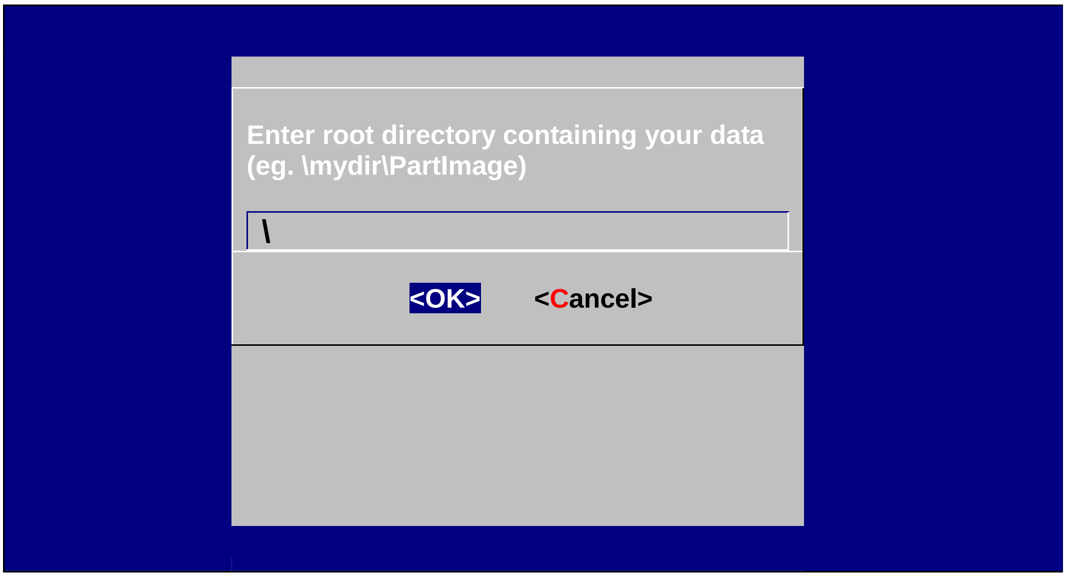
f. Choose the restoration source device **sda1**.



Choose the partition where to store the back/ where to the backup is stored?

```
[ ] hda1 Linux (lost+found,home,etc,media,cdrom,usr…)
[ ] hda2 Linux
[*] sda1 (W95 FAT32 (LBA)) (DA710_V1.0_Build_09112420)
```

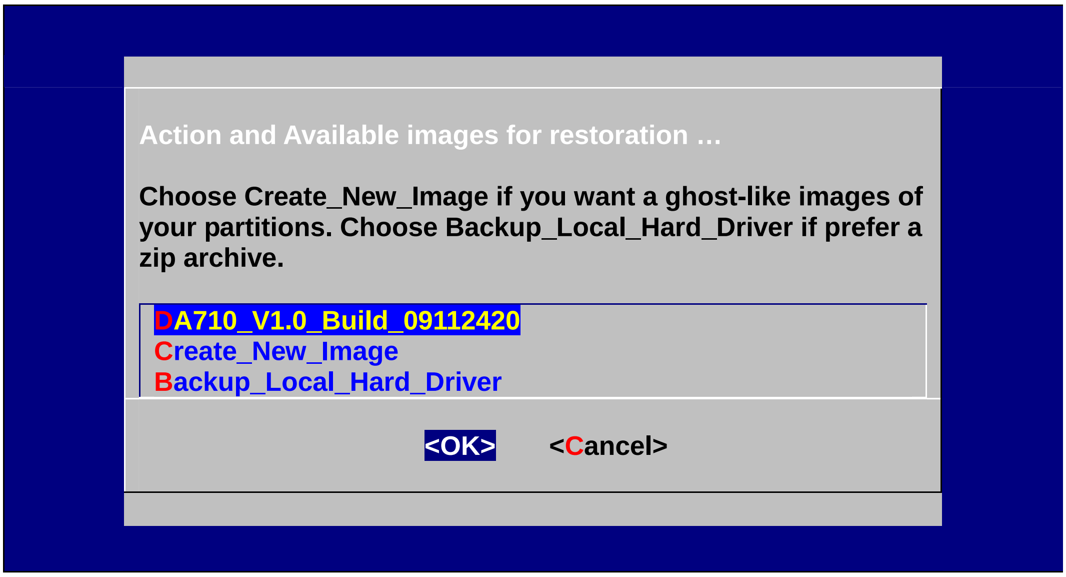
g. Enter "\" to choose the root directory of the restoration image.



Enter root directory containing your data (eg. \mydir\PartImage)

```
\
```

h. Choose **DA710_V1.0_Build_08031316** for the restoration image.

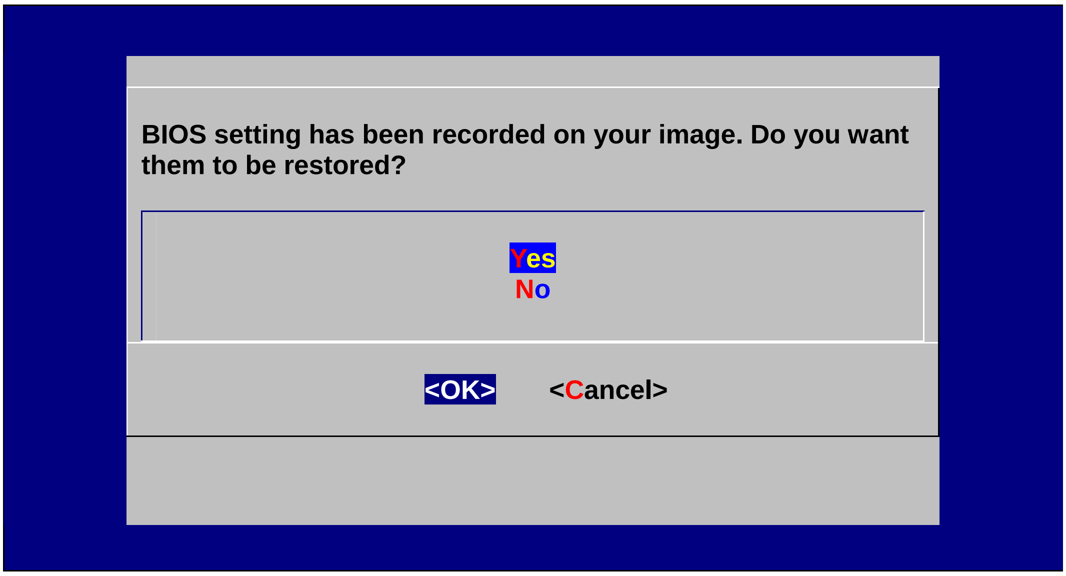


Action and Available images for restoration …

Choose Create_New_Image if you want a ghost-like images of your partitions. Choose Backup_Local_Hard_Driver if prefer a zip archive.

DA710_V1.0_Build_09112420
Create_New_Image
Backup_Local_Hard_Driver

i. Choose **Yes** to start the restoration. After the restoration is finished, the system will halt and you will need to reboot to restart the restored system.



BIOS setting has been recorded on your image. Do you want them to be restored?

Yes
No

When operation is finished, turn off the computer and remove the USB disk.

ATTENTION

DO NOT turn off the power during system recovery, as the system may crash.

Step 5: Set up the BIOS back to boot from DOM or CompactFlash Disk.

a. Power on and press **DEL** to enter the bios setup menu.

b. Select **Advanced → Hard Disk Boot Priority** and then press **Enter**.

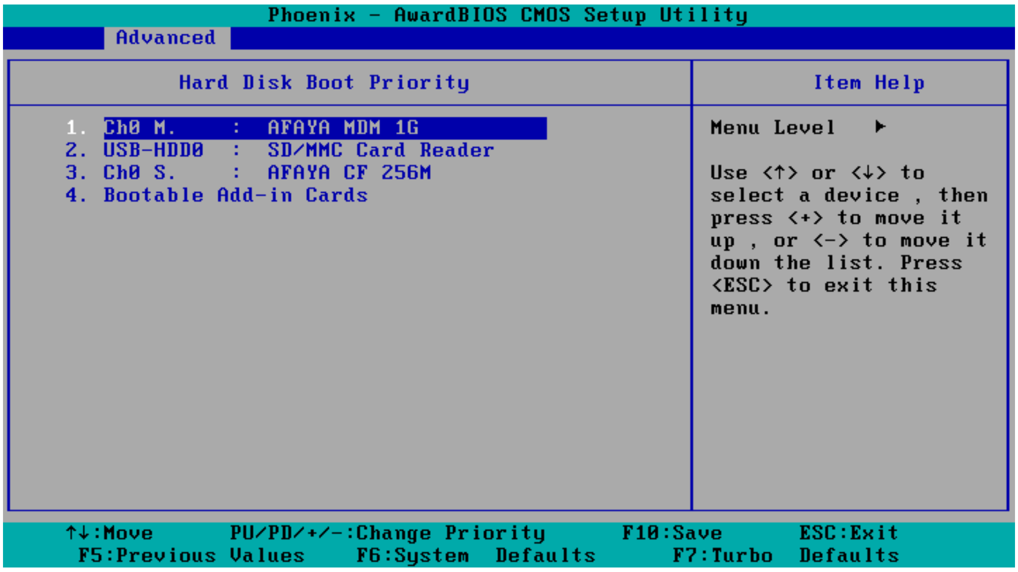
c. From the setup menu, use **"↑"** or **"↓"** to select the DOM or CompactFlash device.

d. Press **"+"** to move the selection up to the first priority, and press **Esc** to exit the setup menu.

e. Select **Exit → Save & Exit Setup** and then press **Enter**.

f. Choose **Y** to save to the CMOS and then exit.

g. Wait a few minutes for the system to boot. When the recovery process is finished, you will again be able to see the Linux desktop.



```
                 Phoenix - AwardBIOS CMOS Setup Utility
        Advanced

              Hard Disk Boot Priority                      Item Help

     1. Ch0 M.      :   AFAYA MDM 1G              Menu Level   ►
     2. USB-HDD0    :   SD/MMC Card Reader
     3. Ch0 S.      :   AFAYA CF 256M             Use <↑> or <↓> to
     4. Bootable Add-in Cards                     select a device , then
                                                  press <+> to move it
                                                  up , or <-> to move it
                                                  down the list. Press
                                                  <ESC> to exit this
                                                  menu.

     ↑↓:Move      PU/PD/+/-:Change Priority      F10:Save      ESC:Exit
      F5:Previous Values      F6:System  Defaults       F7:Turbo  Defaults
```