The 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
© 2014 Moxa Inc. All rights reserved.

Trademarks
The MOXA logo is a registered trademark of Moxa Inc.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

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

Moxa India
Tel: +91-80-4172-9088
Fax: +91-80-4132-1045

Moxa China (Shanghai office)
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-21-5258-5505

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

1. Introduction ............................................................................................................. 1-1
   Overview .................................................................................................................. 1-2
   Software Specifications ........................................................................................... 1-2
   Software Components ......................................................................................... 1-2

2. Software Configuration .......................................................................................... 2-1
   Accessing the DA-820 ............................................................................................ 2-2
      Using a Monitor ..................................................................................................... 2-2
      Using an SSH Client ............................................................................................ 2-2
   Logging in for the First Time .................................................................................. 2-4
   Configuring sudo Access ....................................................................................... 2-5
      Adding a User to the sudo Group ......................................................................... 2-5
      Using the sudo Command .................................................................................... 2-5
   Configuring the System and Hardware Clock .................................................... 2-6
      Setting the Time Manually .................................................................................. 2-6
      Synchronizing with a Network Time Server .................................................... 2-6
      Using a Script to Update the Time Automatically ............................................ 2-7
   Enabling and Disabling Daemons .......................................................................... 2-8
   Setting a Daemon to Run at Startup ...................................................................... 2-9
   Scheduling Tasks Using cron ................................................................................ 2-10
   Configuring samba ............................................................................................... 2-11
   Mounting a USB Storage Device .......................................................................... 2-12
   Checking the Linux Version .................................................................................. 2-12
   Installing and Removing Packages Using APT .................................................. 2-13
      Installing a Package ............................................................................................. 2-13
      Removing a Package ........................................................................................... 2-14
   Setting up a Desktop Environment ....................................................................... 2-14

3. Managing Communications ..................................................................................... 3-1
   Renaming a Network Interface .............................................................................. 3-2
   Configuring Network Settings ............................................................................. 3-2
   Editing an Interface Configuration File ................................................................ 3-2
   Adding IP Addresses Using ifconfig ................................................................. 3-3
   Configuring Telnet and TFTP Servers .................................................................. 3-4
      Enabling the Telnet or TFTP Server ................................................................. 3-4
      Disabling the Telnet or TFTP Server ............................................................... 3-4
   Setting up DNS Client .......................................................................................... 3-4
      /etc/hostname ..................................................................................................... 3-4
      /etc/resolv.conf ................................................................................................. 3-4
      /etc/nsswitch.conf ............................................................................................ 3-5
   Configuring Ethernet Bonding .......................................................................... 3-5
   Configuring the Apache Web Server .................................................................... 3-7
      Default Homepage .............................................................................................. 3-7
      Saving Web Pages to a USB Storage Device .................................................. 3-8
   Configuring IPTABLES ......................................................................................... 3-9
      IPTABLES Hierarchy .......................................................................................... 3-10
      IPTABLES Modules ............................................................................................. 3-11
      Viewing Rules and Deleting Chains ............................................................... 3-12
      Defining Chain Policies ...................................................................................... 3-12
      Adding or Deleting Rules .................................................................................. 3-13
   NAT (Network Address Translation) .................................................................... 3-14
      NAT Example ...................................................................................................... 3-14
   Enabling NAT during System Startup .................................................................. 3-14
   PPP (Point to Point Protocol) .............................................................................. 3-15
      Connecting to a PPP Server Using a Dial-up Connection ................................ 3-15
      Connecting to a PPP Server over a Hard-wired Link ...................................... 3-16
      Checking the Connection .................................................................................. 3-16
      Configuring the Server for Incoming PPP Connections ................................ 3-18
   Configuring PPPoE ................................................................................................. 3-19
   NFS (Network File System) Client ....................................................................... 3-21
   Configuring Ethernet Bridging for Private Networks on Different Subnets ........... 3-22
   Configuring Ethernet Bridging for Private Networks on the Same Subnet .......... 3-25
   Configuring IP Routing ......................................................................................... 3-26

4. Programming Guide ............................................................................................... 4-1
   Getting Product Serial Number ............................................................................ 4-2
   RTC (Real Time Clock) ......................................................................................... 4-2
   UART ..................................................................................................................... 4-2

Error! Bookmark not defined.
5. Optional Modules .................................................................................. 5-1
   Programming Serial Modules ................................................................. 5-2
   Changing the Default Serial Mode ......................................................... 5-3
   Programming the IRIG-B Module ........................................................... 5-3
   Configure the IRIG-B Time Sync Daemon ............................................... 5-5
   Programming the Fiber Module ............................................................... 5-5

6. Managing Disks ...................................................................................... 6-1
   Configuring Software RAID Using mdadm .............................................. 6-2
   Creating a Software RAID Volume ......................................................... 6-2
   Displaying Software RAID Status .......................................................... 6-4
   Replacing a Failed Disk ........................................................................... 6-4

7. Configuring the TPM Module .................................................................. 7-1
   Trusted Platform Module (TPM) and TrouSerS ....................................... 7-2
   Enabling the TPM ................................................................................ 7-3
   Starting TPM Services ........................................................................... 7-3
   Initializing the TPM ............................................................................... 7-3
   Getting the Public Endorsement Key ...................................................... 7-4
   Sealing and Unsealing Data .................................................................. 7-5

8. VMWare ESXi 5.x .................................................................................... 8-1
   About VMWare ESXi Peripheral Control ................................................ 8-2
   Installing the VMWare ESXi Host Driver ................................................. 8-2
   Verifying Package Installation ................................................................ 8-3
   Removing a Package ............................................................................. 8-3
   Installing the VMWare ESXi VMCI Server ............................................. 8-3
   VMWare VMCI Client Example ............................................................. 8-4
   Device Control Layer and API definitions .............................................. 8-4
   VMCI Utility Example Code ................................................................. 8-7
   Compiling the VMCI Utility Example in Linux ....................................... 8-10
   Compiling the VMCI Utility Example in Windows .................................. 8-11

9. System Recovery ................................................................................... 9-1
   Overview ............................................................................................... 9-2
   Setting Up the System Recovery Environment ....................................... 9-2
   Recovering from the Factory Default Image ......................................... 9-3
   Step 1: Preparing the USB Drive ......................................................... 9-3
   Step 2: Setting the BIOS to Boot from the USB Drive .......................... 9-5
   Step 3: Performing a System Recovery ................................................ 9-7
   Step 4: Resetting the BIOS ................................................................. 9-9
   Creating a Custom System Image ......................................................... 9-10

A. Software Components ............................................................................. A-1

Setting the UART Mode ........................................................................... 4-3
Programmable LED Indicators ............................................................... 4-4
Watch Dog Timer .................................................................................... 4-5
The Watchdog Device IOCTL Commands ............................................ 4-6
Examples ............................................................................................... 4-7
TPM ....................................................................................................... 4-7

---

Configuring the TPM Module ................................................................. 5-1
Creating a Custom System Image ............................................................ 5-1
Setting Up the System Recovery Environment ....................................... 5-1
VMCI Utility Example Code ................................................................. 5-1
VMWare VMCI Client Example ............................................................. 5-1
Installing the VMWare ESXi VMCI Server ............................................. 5-1
Installing the VMWare ESXi Host Driver ................................................. 5-1
About VM

---

Step 3: Performing a System Recovery ................................................ 9-7
Step 4: Resetting the BIOS ................................................................. 9-9
Creating a Custom System Image ......................................................... 9-10

---

A. Software Components ............................................................................. A-1
Thank you for purchasing the Moxa DA-820 Series of x86 ready-to-run embedded computers. This manual introduces the software configuration and management of the DA-820-LX computer, which runs the Linux operating system. For hardware installation, connector interfaces, setup, and upgrading the BIOS, refer to the DA-820 Series Embedded Computer User’s Manual.

Linux is an open, scalable operating system that allows you to develop a wide range of innovative, small footprint devices. Software written for desktop PCs can be easily ported to the Linux-based embedded computer with a GNU 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.

The following topics are covered in this chapter:

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

The DA-820 Series embedded computers are based on the Intel 3rd-generation processor and feature two serial ports, four 10/100/1000 Mbps LAN ports, six USB 2.0 hosts, and a CFast socket. The DA-820 Series offers dual VGA outputs, making it exceptionally well suited for industrial applications such as SCADA and factory automation.

The two serial ports on the DA-820 Series make it ideal for connecting a wide range of serial devices, and the four 10/100/1000 Mbps Ethernet ports offer a reliable solution for network redundancy, which taken together promise continuous data communication and management operations. For added convenience, the DA-820 computers come with eight programmable LEDs for status indication. In addition, the CFast socket, and USB and SATA ports provide DA-820 computers with data buffering and storage expansion, which provide the necessary reliability for industrial applications. The IRIG-B time protocol is widely used by electric utilities, industrials, and others to ensure precise time synchronization of power system devices, such as breakers, relays, and meters. The DA-820 Series has an IRIG-B expansion module for your power system to support IRIG-B time synchronization.

Pre-installed with Linux, the DA-820 Series provides programmers with a friendly environment for developing sophisticated, bug-free application software at a lower cost.

Software Specifications

The Linux operating system pre-installed on the DA-820 embedded computers is the Debian Wheezy distribution. The Debian project involves a worldwide group of volunteers who endeavor to produce an operating system distribution 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.

The following figure shows an example software architecture.

**ATTENTION**

Depending on the Linux distribution and build version, not all components are included as shown in the figure.

**ATTENTION**


Software Components

The DA-820-LX computers are pre-installed with the Debian Wheezy 7.2 Linux distribution. For more information on the software components, see Appendix A.
Software Configuration

This chapter describes how to configure and manage a DA-820-LX computer directly from your desktop. There are two ways to connect to the DA-820-LX computer: through a VGA monitor or an SSH console. This chapter describes basic Linux operating system settings.

The following topics are covered in this chapter:

- **Accessing the DA-820**
  - Using a Monitor
  - Using an SSH Client

- **Logging in for the First Time**

- **Configuring sudo Access**
  - Adding a User to the sudo Group
  - Using the sudo Command

- **Configuring the System and Hardware Clock**
  - Setting the Time Manually
  - Synchronizing with a Network Time Server
  - Using a Script to Update the Time Automatically

- **Enabling and Disabling Daemons**
  - Setting a Daemon to Run at Startup

- **Configuring Runlevels Using insserv**

- **Scheduling Tasks Using cron**

- **Mounting a USB Storage Device**

- **Checking the Linux Version**

- **Installing and Removing Packages Using APT**
  - Installing a Package
  - Removing a Package

- **Setting up a Desktop Environment**
Accessing the DA-820

You can access the DA-820 using one of the following methods:

- A monitor connected to the DA-820.
- An SSH client from a computer.

Using a Monitor

1. Connect a monitor to the DA-820 and turn on the monitor and the DA-820.

   **NOTE** The DA-820 may take up to 60 seconds to boot up.

2. A login screen appears. Enter the login information.

   For more information, see the Logging in for the First Time and Account Management sections.

Using an SSH Client

You can access the DA-820 using an SSH client on a computer.

The DA-820-LX comes with four Gigabit ports (LAN1, LAN2, LAN3, and LAN4). The following table shows the default IP address and subnet mast settings on each port.

<table>
<thead>
<tr>
<th>Default IP Address</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN 1  192.168.3.127</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAN 2  192.168.4.127</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAN 3  192.168.5.127</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAN 4  192.168.6.127</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

1. Use a cross-over Ethernet cable to connect your computer to one of the Ethernet ports on the DA-820.
2. Make sure that your computer is on the same subnet as the DA-820.

   For example, if your computer is connected to the LAN1 port on the DA-820, set the IP address of your computer to 192.168.3.126 and the subnet mask to 255.255.255.0. If your computer is connected to the LAN2 port on the DA-820, set the IP address of your computer to 192.168.4.126 and the subnet mask to 255.255.255.0.

3. Start the SSH client on your computer to establish a connection to the DA-820.
4. A login screen appears. Enter the login information.

   For more information, see the Logging in for the First Time and Account Management sections.
Windows Users

This section shows you how to access the DA-820 from a Windows computer.

1. Start an SSH client application.
   You can install PuTTY (a free SSH client) in Windows.

2. Configure the connection information (for example, the host IP address and port number) to connect to the DA-820. The following figure shows the PuTTY configuration screen.

![PuTTY Configuration Screen]

Linux Users

1. To access the DA-820 from a Linux computer, use the ssh command to establish an SSH connection. For example: ssh 192.168.3.127

2. When prompted, enter "yes" to start the connection.

```bash
root@Moxa:~# ssh 192.168.3.127
The authenticity of host '192.168.3.127 (192.168.3.127)' can't be established.
Are you sure you want to continue connection (yes/no)? yes
```
Logging in for the First Time

Complete the following steps to log into the DA-820 for the first time.

1. In the login screen, enter the default account information to log in.
   
   **Login:** moxa
   
   **Password:** moxa

2. When prompted, change the password for the default login account.

   ```
   login as: moxa
   moxa@192.168.27.42's password:
   You are required to change your password immediately (root enforced)
   Linux Moxa 3.14-0.bpo.2-amd64 #1 SMP Debian 3.14.15-2-bpo70+1 x86_64
   ```

   For further information check:

   http://www.moxa.com/

   Mount user file system.

   ```
   moxa@Moxa:~#
   ```

   **ATTENTION**

   For security purposes, it is strongly recommended that you disable the root user and use the sudo command to perform administrative tasks in the DA-920.
Configuring sudo Access

For added security, the `sudo` command provides administrative privileges for trusted users to access the DA-820 without sharing the `root` user password.

Adding a User to the sudo Group

1. Use the `useradd [USER-ACCOUNT-NAME-HERE] sudo` command to add a user to the `sudo` group. For example, `useradd foobar sudo`
2. Use the `group` command to check that the user has been added to the `sudo` group.
3. If required, configure the `sudoers` file.

**NOTE**

For more information about the `sudo` command or the `sudoers` file, refer to the following websites: may

- Linux.com’s introduction to `sudo`
- Debian introduction to the `sudo` command:
  https://wiki.debian.org/sudo
- Ubuntu (a Debian sub-distribution) documentation for the `sudoers` file
  https://help.ubuntu.com/community/Sudoers
- A sample `sudoers` file for an extended network
  http://www.sudo.ws/sudo/sample.sudoers

Using the sudo Command

1. To run commands as the root user, type “sudo” before a command.
   For example, to configure the IP address on Ethernet port 1, enter the following command:
   `sudo ifconfig eth0 192.168.100.100`
2. When prompted, enter the password for your user account.

   ```bash
   moxa@Moxa:~# sudo ifconfig eth0 192.168.100.100
   [sudo] password for moxa:
   moxa@Moxa:~$ sudo ifconfig eth1
   [sudo] password for moxa:
   eth1      Link encap:Ethernet  HWaddr 00:90:e8:00:df:fe
           inet addr:192.168.100.100  Bcast:192.168.100.255  Mask:255.255.255.0
           UP BROADCAST MULTICAST  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
           Interrupt:41 Base address:0xe000
   moxa@Moxa:~$
   ```

To switch to the root user, first disconnect the DA-820 from the network; then, enter the `sudo -i` command.

```bash
moxa@Moxa:~# sudo -i
[sudo] password for moxa:
root@Moxa:~$
```
Configuring the System and Hardware Clock

This section shows you how to configure the system clock and hardware clock (or Real Time Clock) on the DA-820.

Setting the Time Manually

Use the `date` command to display and set the system clock.

```
date MMDDhhmmYYYY
```

Where:
- **MM** = Month
- **DD** = Date
- **hh** = Hour
- **mm** = Minute
- **YYYY** = Year

To display and set the hardware clock, use the `hwclock` command.

The following command sets the hardware clock to the current system clock.

```
hwclock -w
```

The following figure shows an example.

```
MOXA:~# date
Wed Dec 16 03:34:46 CST 2009
MOXA:~# hwclock
Wed 16 Dec 2009 03:35:16 AM CST -0.017600 seconds
MOXA:~# date 121616352009
Wed Dec 16 16:35:00 CST 2009
MOXA:~# hwclock -w
MOXA:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2009
Wed 16 Dec 2009 03:38:13 AM CST -0.016751 seconds
MOXA:~#
```

Synchronizing with a Network Time Server

The DA-820 is pre-installed with an NTP (Network Time Protocol) client. You can use the `ntpd` command to synchronize the system clock with a network time server.

NOTE
Before you set the DA-820 to synchronize with a network time server, make sure that the DA-820 can connect to the Internet.

The following commands synchronize the system clock with the network time server, `time.stdtime.gov.tw` and set the hardware clock to the system clock.

```
root@Moxa:~# ntpdate time.stdtime.gov.tw
root@Moxa:~# hwclock -w
```
For more information on NTP and NTP server, go to http://www.ntp.org.

MOXA:~# date ; hwclock
Wed Sep 10 14:51:19 UTC 2014
Wed 10 Sep 2014 02:51:39 PM UTC -0.938147 seconds
MOXA:~#
MOXA:~# ntpdate time.stdtime.gov.tw
10 Sep 06:54:14 ntpdate[2510]: step time server 220.130.158.52 offset -28682.196417 sec
MOXA:~#
MOXA:~# hwclock -w
MOXA:~# date ; hwclock
Wed Sep 10 06:54:47 UTC 2014
Wed 10 Sep 2014 06:54:48 AM UTC -0.734969 seconds
MOXA:~#

Using a Script to Update the Time Automatically

You can create and run a shell script to update the time automatically on the DA-820.

The following figure shows a script example.

```
#!/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.
```

To run the shell script every time the system boots up, complete the following steps:

1. By default, the root file system is mounted with read-only permission. Re-mount the root file system with read-write permission.
   ```
   root@Moxa:~# mount -o remount,rw /
   ```

2. Save the shell script (for example, fixtime) in the /etc/init.d directory and use the chmod command to set the access mode.
   ```
   root@Moxa:~# chmod 755 fixtime
   ```

3. Open the /etc/inittab file and append the following line.
   ```
   root@Moxa:~# echo ‘ntp : 2345 : respawn : /etc/init.d/fixtime’ >> /etc/inittab
   ```

4. Use the umount / command to set the root file directory to read-only mode.
   ```
   root@Moxa:~# umount /
   ```

5. Enter the init q command to re-initialize the kernel.
   ```
   root@Moxa:~# init q
   ```
Enabling and Disabling Daemons

When the DA-820 boots up for the first time, the following daemons are enabled by default:

- **ServiceSyncTime**: IRIG-B Time Sync Daemon
- **Inetd**: Internet Daemons
- **Exim4**: SMTP Server Daemon
- **Sshd**: Secure Shell Server Daemon
- **Httpd**: Apache WWW Server Daemon

To display all running processes, enter the `ps -ef` command.

```
root@Moxa:~# ps -ef
UID    PID  PPID  C   STIME TTY     TIME CMD
root    1    0    0  06:47 ?        00:00:00 init [2]
root    2    0    0  06:47 ?        00:00:00 [kthreadd]
root    3    2    0  06:47 ?        00:00:00 [ksoftirqd/0]
root    5    2    0  06:47 ?        00:00:00 [kworker/0:0H]
root    7    2    0  06:47 ?        00:00:00 [rcu_sched]
root    8    2    0  06:47 ?        00:00:00 [rcu_bh]
root    9    2    0  06:47 ?        00:00:00 [migration/0]
root   10    2    0  06:47 ?        00:00:00 [watchdog/0]
root   11    2    0  06:47 ?        00:00:00 [watchdog/1]
root   12    2    0  06:47 ?        00:00:00 [migration/1]
root   13    2    0  06:47 ?        00:00:00 [ksoftirqd/1]
root   14    2    0  06:47 ?        00:00:00 [kworker/1:0]
root   15    2    0  06:47 ?        00:00:00 [kworker/1:0H]
root   16    2    0  06:47 ?        00:00:00 [kworker/2:0]
root   17    2    0  06:47 ?        00:00:00 [migration/2]
root   18    2    0  06:47 ?        00:00:00 [ksoftirqd/2]
root   19    2    0  06:47 ?        00:00:00 [kworker/2:0]
root   20    2    0  06:47 ?        00:00:00 [kworker/2:0H]
root   21    2    0  06:47 ?        00:00:00 [watchdog/3]
root   22    2    0  06:47 ?        00:00:00 [migration/3]
root   23    2    0  06:47 ?        00:00:00 [ksoftirqd/3]
root   24    2    0  06:47 ?        00:00:00 [kworker/3:0]
root   25    2    0  06:47 ?        00:00:00 [kworker/3:0H]
root   26    2    0  06:47 ?        00:00:00 [khelper]
root   27    2    0  06:47 ?        00:00:00 [kdevtmpfs]
root   28    2    0  06:47 ?        00:00:00 [netns]
root   29    2    0  06:47 ?        00:00:00 [writeback]
root   30    2    0  06:47 ?        00:00:00 [ksmd]
root   31    2    0  06:47 ?        00:00:00 [khugepaged]
root   32    2    0  06:47 ?        00:00:00 [kintegrityd]
root   33    2    0  06:47 ?        00:00:00 [bioset]
root   34    2    0  06:47 ?        00:00:00 [crypto]
root   35    2    0  06:47 ?        00:00:00 [kblistd]
root   36    2    0  06:47 ?        00:00:00 [kworker/0:1]
root   37    2    0  06:47 ?        00:00:00 [khungtaskd]
root   38    2    0  06:47 ?        00:00:00 [kswapd0]
root   39    2    0  06:47 ?        00:00:00 [fsnotify_mark]
root   44    2    0  06:47 ?        00:00:00 [kthrotld]
root   45    2    0  06:47 ?        00:00:00 [ipv6_addrconf]
root   46    2    0  06:47 ?        00:00:00 [kworker/0:2]
root   47    2    0  06:47 ?        00:00:00 [deferwq]
root   57    2    0  06:47 ?        00:00:00 [kworker/3:1]
root  169    2    0  06:47 ?        00:00:00 [ata_sff]
root  181    2    0  06:47 ?        00:00:00 [scsi_eh_0]
root  183    2    0  06:47 ?        00:00:00 [scsi_tmf_0]
root  184    2    0  06:47 ?        00:00:00 [scsi_eh_1]
root  185    2    0  06:47 ?        00:00:00 [scsi_tmf_1]
root  186    2    0  06:47 ?        00:00:00 [scsi_eh_2]
root  187    2    0  06:47 ?        00:00:00 [scsi_tmf_2]
root  188    2    0  06:47 ?        00:00:00 [scsi_eh_3]
root  189    2    0  06:47 ?        00:00:00 [scsi_tmf_3]
root  190    2    0  06:47 ?        00:00:00 [scsi_eh_4]
root  191    2    0  06:47 ?        00:00:00 [scsi_tmf_4]
root  192    2    0  06:47 ?        00:00:00 [scsi_eh_5]
root  193    2    0  06:47 ?        00:00:00 [scsi_tmf_5]
root  196    2    0  06:47 ?        00:00:00 [kworker/u16:4]
root  197    2    0  06:47 ?        00:00:00 [kworker/u16:5]
```
Setting a Daemon to Run at Startup

You can edit the rc.local file to run a daemon at system start-up. Complete the following steps:

1. By default, the root file system is mounted with read-only permission. Re-mount the root file system with read-write permission.

   MOXA:~# mount -o remount,rw /

2. Change to the /etc directory.

   MOXA:~# cd /etc/

3. Type vi rc.local to edit the rc.local file using the vi editor.

   MOXA:/etc/# vi rc.local

4. Add the application daemon that you want to run and save the file.

   The following figure shows an example where the sample program tcp2-release (available on the DA-820 software CD/DVD) is added.

   #!/bin/sh
   # Add you want to run daemon
   /root/tcp2-release

5. Use the umount / command to set the root file directory to read-only mode.

   MOXA:~# umount /
6. After you restart the DA-820, the application daemon appears on the running process list.

```
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 [bdfflush]
 6 root  S [kupdated]
 7 root  S [mtdblockd]
 8 root  S [khubd]
 9 root  S [jffs2_gcd_mtd3]
10 root  D [ixp425_csr]
38 root  1256 S stdef
47 root  1368 S /usr/sbin/inetd
53 root  4464 S /usr/sbin/httpd
65 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
111 root  4044 S /usr/sbin/snmpd -s -l /dev/null
114 root  4480 S /usr/sbin/httpd
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 ttyM0
164 root  1532 S /sbin/getty 115200 ttyM1
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
```

Configuring Runlevels Using insserv

You can use the scripts in the `/etc/init.d/` directory to start or stop a service in Linux. To add or remove the service to or from a run level in `/etc/rcX.d/`, use the `insserv` command.

Complete the following steps to set the runlevel of a service.

1. Create a start-stop script as shown in the following figure. Save the script (for example, as `tcps2`) in the `/etc/rcX.d/` directory.

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          tcps2
# Required-Start:    
# Required-Stop:     
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: tcps2
### END INIT INFO

. /lib/lsb/init-functions
export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"

case "$1" in
  start)
    start-stop-daemon --start --quiet --oknodo --pidfile /var/run/tcps2.pid
    --exec /usr/sbin/tcps2
```

2-10
2. Enter the following `insserv` command to add the service and run the service at startup.

```bash
moxa@Moxa:~# sudo insserv -v -d tcps2
```

3. Use the `ls` command to check the runlevel of the service.

```bash
moxa@Moxa:~# ls -l /etc/rc?.d/*tcps*
```

To remove a service from all runlevels, enter the `insserv -r` command.

```bash
moxa@Moxa:~# insserv -r tcps2
```

Then, use the `ls` command to check the runlevels of the service. The system should display a warning message.

```bash
moxa@Moxa:~# ls -l /etc/rc?.d/*tcps*
ls: cannot access /etc/rc?.d/*tcps*: No such file or directory
moxa@Moxa:~#
```

### Scheduling Tasks Using cron

You can set the DA-820 to run scheduled tasks using `cron`.

The `cron` daemon searches the `/etc/crontab` directory for crontab files. Every minute, the cron daemon checks each command to see if the command should be run. The output of the commands is sent to the owner of the crontab file (or to the user specified in the MAILTO environment variable in the `crontab` file).

You can schedule command executions in a crontab file. The following table shows the format of a crontab file.

<table>
<thead>
<tr>
<th>mm</th>
<th>h</th>
<th>dom</th>
<th>mon</th>
<th>dow</th>
<th>user</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>minute</td>
<td>hour</td>
<td>day of month</td>
<td>month</td>
<td>day of week</td>
<td>user</td>
<td>command</td>
</tr>
<tr>
<td>0-59</td>
<td>0-23</td>
<td>1-31</td>
<td>1-12</td>
<td>0-6 (0 is Sunday)</td>
<td>root</td>
<td>/path/to/your/program</td>
</tr>
</tbody>
</table>

For example, the following crontab line sets the system to run a program at 8:00 every day.

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

The following procedure shows the steps to configure cron to update the system time and hardware time every day at 8:00.

1. Create a script with the following lines and save it in the `/home/` directory (for example, as `fixtime.sh`).

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

2. Change the mode of `fixtime.sh`.

```bash
# chmod 755 fixtime.sh
```

3. Append the following line to the crontab file to run `fixtime.sh` at 8:00 every day.

```bash
* 8 * * * root /home/fixtime.sh
```
Mounting a USB Storage Device

The Debian Linux distribution includes a usbmount package that allows you to automatically mount USB storage devices. usbmount depends on the udev demon that calls the script to mount the USB storage devices automatically at specific mount points. The first USB storage device will be mounted on /media/usb0, and the second USB storage device on /media/usb1, etc.

```
MOXA:~# mount
/dev/hda1 on / type ext2 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,nodev)
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/sdal 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:~#
```

Note that usbmount is only available in text mode and does not support the gnome desktop environment. Alternatively, you can install gnome-volume-manager for better support.

```
MOXA:~# mount -o,remount rw /
MOXA:~# apt-get remove usbmount
MOXA:~# apt-get install gnome-volume-manager
MOXA:~# umount /
```

**ATTENTION**

To prevent data loss, before you disconnect a USB storage device from the DA-820, enter the sync command.

**ATTENTION**

Before you disconnect a USB storage device, exit from the mount directory (for example, /media/usb0 or /media/usb1); otherwise, the automatic un-mount process will not function.

If the automatic un-mount process fails to function, use the umount command to manually unmount the USB storage devices.

Checking the Linux Version

You can use the `uname` (Unix Name) command to view information (for example, name and version) about the Linux distribution installed on the DA-820.

The following figure shows an example.

```
MOXA:~# uname -a
Linux Moxa 3.14-0.bpo.2-amd64 #1 SMP Tue Jun 14 09:42:28 UTC 2011 x86_64 GNU/Linux
MOXA:~#
```
Installing and Removing Packages Using APT

In Debian, you can use APT (Advanced Package Tool) to install and remove packages.

Installing a Package

Complete the following steps to install a package using APT.

1. Before you install a package using APT, you must configure the apt source list /etc/apt/sources.list that is read-only by default.

   Mount the root file system with write permission.
   
   MOXA:~# mount -o remount,rw

2. Edit the /etc/apt/sources.list file in the vi editor.

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

   
   # deb cdrom:[Debian GNU/Linux 7.6.0 _Wheezy_ - Official amd64 NETINST Binary-1 20130615-23:04]/ wheezy main
deb http://ftp.us.debian.org/debian/ wheezy main
deb-sr http://ftp.us.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb-sr http://security.debian.org/ wheezy/updates main
# wheezy-updates, previously known as 'volatile'
deb http://ftp.us.debian.org/debian wheezy-updates main
deb-sr http://ftp.us.debian.org/debian wheezy-updates main

3. Enter the apt-get update command to update the APT source list.

   MOXA:~# apt-get update

4. Enter the apt-get command with the install option to install a package (for example, openswan).

   MOXA:~# apt-get install openswan

5. After the installation is complete, unmount the root directory back to read-only mode.

   MOXA:~# umount /

ATTENTION

The APT cache space /var/cache/apt is located in tmpfs. If you want to install a huge package, link /var/cache/apt to the USB drive or mount it to an NFS space to generate more free space. Use the df -h command to check how much free space is available on tmpfs.

Moxa:~# df -h
Filesystem Size Used Avail Use% Mounted on
rootfs 1.5G 1001M 440M 70% /
udev 10M 748K 9.3M 8% /dev
/dev/sda1 1.5G 1001M 440M 70% /
tmpfs 501M 0 501M 0% /lib/init/rw
tmpfs 501M 0 501M 0% /dev/shm
none 501M 19M 482M 4% /tmp
/dev/sda2 270M 130M 126M 51% /home
Moxa:~#
Removing a Package

You can use one of the following APT commands to remove a package:

- To remove a package without deleting related configuration files.
  
  MOXA:~# apt-get remove openswan
  MOXA:~#

- To remove a package and delete all related configuration files.
  
  MOXA:~# apt-get remove openswan --purge
  MOXA:~#

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

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

Setting up a Desktop Environment

This section shows you how to set up a desktop environment in Debian on the DA-820. By default, the Debian Linux distribution on the DA-820 is not pre-installed with a desktop environment. Debian supports various full-featured graphical environments (such as Gnome and KDE).

To install a desktop environment on the DA-820, use one of the following commands:

- Gnome
  
  moxa@MOXA:~# sudo apt-get install gnome-desktop

- KDE
  
  moxa@MOXA:~# sudo apt-get install kde-standard

- Xfce
  
  moxa@MOXA:~# apt-get install xfce4 xfce4-goodies thunar-archive-plugin

- LXDE
  
  moxa@MOXA:~# sudo apt-get install lxde-core lxde
Managing Communications

The DA-820-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.

The following topics are covered in this chapter:

- Renaming a Network Interface
- Configuring Network Settings
  - Editing an Interface Configuration File
  - Adjusting IP Addresses Using ifconfig
- Configuring Telnet and TFTP Servers
  - Enabling the Telnet or TFTP Server
  - Disabling the Telnet or TFTP Server
- Setting up DNS Client
  - /etc/hostname
  - /etc/resolv.conf
  - /etc/nsswitch.conf
- Configuring Ethernet Bonding
- Configuring the Apache Web Server
  - Default Homepage
  - Disabling the CGI Function
  - Saving Web Pages to a USB Storage Device
- Configuring IPTABLES
  - IPTABLES Hierarchy
  - IPTABLES Modules
  - Viewing Rules and Deleting Chains
  - Defining Chain Policies
  - Adding or Deleting Rules
- NAT (Network Address Translation)
  - NAT Example
  - Enabling NAT during System Startup
- PPP (Point to Point Protocol)
  - Connecting to a PPP Server Using a Dial-up Connection
  - Connecting to a PPP Server over a Hard-wired Link
  - Checking the Connection
  - Configuring the Server for Incoming PPP Connections
- Configuring PPPoE
- NFS (Network File System) Client
- OpenVPN
  - Ethernet Bridging for Private Networks on Different Subnets
  - Ethernet Bridging for Private Networks on the Same Subnet
  - Configuring IP Routing
Renaming a Network Interface

Linux systems use udevd to detect new network interfaces (for example, Ethernet and wireless interfaces). However, because the automatically assigned name for an interface may not match the label, you can specify the naming rules in `/etc/udev/rules.d/01-rename_net_interface.rules`.

The following figure shows an example of the rule content:

```
KERNEL="eth*", KERNELS="0000:00:19.0", NAME="eth3"
KERNEL="eth*", KERNELS="0000:01:00.0", NAME="eth0"
KERNEL="eth*", KERNELS="0000:03:00.0", NAME="eth2"
KERNEL="eth*", KERNELS="0000:02:00.0", NAME="eth1"
```

Configuring Network Settings

There are four 10/100/1000 Ethernet ports on the DA-820-LX computer. The following table shows the default IP addresses and subnet masks on the Ethernet ports.

<table>
<thead>
<tr>
<th>Ethernet Port</th>
<th>Default IP Address</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN1</td>
<td>192.168.3.127</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAN2</td>
<td>192.168.4.127</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAN3</td>
<td>192.168.5.127</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAN4</td>
<td>192.168.6.127</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

You can edit the interface configuration file `interface` to permanently change the network settings.

To change network settings temporarily, use the `ipconfig` command.

Editing an Interface Configuration File

1. Type `cd /etc/network` to change directories.

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

2. Type `vi interfaces` to edit the file in the vi editor. Refer to the following sections for information on how to set an Ethernet port to use a static or dynamic (DHCP) IP address.

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

3. Type the following command to make the network settings take effect.

   ```
   # /etc/init.d/networking restart
   ```

Using Static IP Addresses

The following figure shows a configuration file example where the Ethernet ports are set to use static IP address that you can change.

```
# 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
```
Using Dynamic IP Addresses with DHCP

The following figure shows an example where the `eth0` interface is set to request for a dynamic IP address from a DHCP server.

```plaintext
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

After you change the configuration file, enter the following command to make the changes take effect immediately.

```
# /etc/init.d/networking restart

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

Adjusting IP Addresses Using ifconfig

You can temporarily change the IP address of an Ethernet port using the `ipconfig` command. IP address settings made using the `ipconfig` command is not saved on the system ROM.

The following figure shows an example where the IP address of the `eth0` port is change to 192.168.1.1.

```
Moxa:~# ifconfig eth0 192.168.1.1
Moxa:~#
```

Configuring Telnet and TFTP Servers

In addition to supporting Telnet client/server and TFTP client/server, the DA-820-LX also supports SSH and sftp client/server.

To enable or disable the Telnet or ftp server, you must edit the file `/etc/inetd.conf`.

1. Mount the root file system with write permission.
   ```
   Moxa:~# mount -o remount,rw /
   ```

2. Type `cd /etc` to change the directory.
   ```
   Moxa:~# cd /etc
   ```

3. Type `vi inetd.conf` to edit the configuration file in the vi editor.
   ```
   Moxa:/etc# vi inetd.conf
   ```
Enabling the Telnet or TFTP Server

The following example shows the content of the file `/etc/inetd.conf`. By default, the Telnet and TFTP servers are enabled.

```
telnet        stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
 tftp           dgram   udp     wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd
```

Disabling the Telnet or TFTP Server

To disable the Telnet or TFTP server, add an “#” prefix to an entry in the configuration file.

The following figure shows an example where the TFTP server is disabled.

```
telnet        stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
 #tftp           dgram   udp     wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd
```

After you finish editing the interface configuration file, unmount the root directory back to read-only mode and restart the `inetd` service.

```
MOXA:~# umount /
MOXA:~# service openbsd-inetd restart
```

Setting up DNS Client

To configure the DNS client on the DA-820, edit the following configuration files:

- `/etc/hostname`
- `/etc/resolv.conf`
- `/etc/nsswitch.conf`

```
/etc/hostname
```

1. Edit the `/etc/hostname` file in vi editor.
   ```
   moxa@MOXA:~# sudo vi /etc/hostname
   MOXA
   ```
2. Re-configure the hostname.
   ```
   moxa@MOXA:~# sudo /etc/init.d/hostname.sh start
   ```
3. Check the new hostname.
   ```
   moxa@MOXA:~# hostname
   ```

```
/etc/resolv.conf
```

The `resolv.conf` file contains the information that allows the system to resolve names into IP addresses. For example, before you can set the system time to synchronize with the `time.stdtime.gov.tw` NTP server, you must add a DNS server IP address in the `resolv.conf` file.

Check with your network administrator to obtain the DNS server IP address information.

You must add an entry in the `resolv.conf` file in the format: `nameserver [IP address]`
The following figure shows an example of the `resolv.conf` file content.

```
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`

The `nsswitch.conf` file defines the reading sequence of the `/etc/hosts` or `/etc/resolv.conf` file to resolve the IP address.

In the following example, the sequence setting for `hosts` sets the system to check the `/etc/hosts` file first and then use the DNS service to resolve IP addresses.

```
#!/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
```

**Configuring Ethernet Bonding**

You can use the bonding driver in Linux to aggregate multiple network interfaces into a single logical "bonded" interface. To use the bonding feature, first load the bonding driver with mode settings. Then, use the `ifenslave` command to add two or more Ethernet interfaces to the `bond0` interface.

The following figure shows a sample script (`/etc/init.d/bonding.sh`) that aggregates interfaces `eth1` and `eth2` into interface `bond0`.

```
#!/bin/bash

### BEGIN INIT INFO
# Provides: bonding
# Short-Description: Start the bonding service, bond eth1 and eth2.
# Required-Start: $all
# Required-Stop: $all
# Should-Start: $all
```

3-5
# Should-Stop:
# Default-Start:  2 3 4 5
# Default-Stop:  0 1 6
### END INIT INFO

NAME=bonding
PATH=/bin:/usr/bin:/sbin:/usr/sbin
case "$1" in
  start)
    # to set ethX interfaces as slave the bond0 must have an ip
    if [ "$2" == "" ]; then
      $0
      exit 1
    fi
    echo "Starting bonding service: $NAME."
    modprobe bonding mode=1 miimon=100  # load bonding module
    ifdown eth2  # putting down eth2
    ifdown eth1  # putting down eth1
    ifconfig bond0 hw ether 00:90:E8:00:00:00  # change mac address
    ifconfig bond0 $2 netmask 255.255.255.0 up  # set ip address
    ifenslave bond0 eth2  # set eth2 in slave for bond0
    ifenslave bond0 eth1  # set eth1 in slave for bond0
  ;;
  stop)
    echo "Stopping bonding service: $NAME"
    ifenslave -d bond0 eth2  # release eth2 from bond0
    ifenslave -d bond0 eth1  # release eth1 from bond0
    ifconfig bond0 down  # putting down bond0
    modprobe -r bonding  # unload bonding module
    ifup eth2
    ifup eth1
  ;;
  restart)
    $0 stop
    $0 start $2
  ;;
  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|restart} [ip address]"
    exit 1
  ;;
esac
exit 0

Use the `insserv` command to set the runlevel of the `bonding.sh` script.

moxa@MOXA:~# sudo insserv -v -d bonding.sh

Use the `insserv -r` command to remove the script from all runlevels.

moxa@MOXA:~# sudo insserv -r bonding.sh
Configuring the Apache Web Server

Default Homepage

The configuration file for the Apache web server is `/etc/apache2/sites-enabled/000-default` and the default homepage file is `/var/www/apache2-default/index.html`.

You can save the files for your homepage in the following directory:

`/var/www`

Save your CGI pages in the following directory:

`/var/www`

Before you modify the homepage, complete the following steps to test whether the Apache web server on the DA-820 is running:

1. Open a web browser (for example, Microsoft Internet Explorer or Mozilla Firefox). Then, enter the LAN IP address of the DA-820 in the address bar (for example, http://192.168.3.127)
2. Test the default CGI page. Enter the address in the format:

   http://[IP address]/cgi-bin/w3mmail.cgi

   For example, http://192.168.3.127/cgi-bin/w3mmail.cgi.

Disabling the CGI Function

By default, the CGI function is enabled. If you want to disable this function, edit the `/etc/apache2/sites-enabled/000-default` file.

1. Type `vi /etc/apache2/sites-enabled/000-default` to open the configuration file in vi.
2. Type `#` to comment out the following lines:

   ```
   #ScriptAlias /cgi-bin/ /var/www/cgi-bin/
   #<Directory "/var/www/cgi-bin/">
   #   AllowOverride None
   #   Options ExecCGI -MultiViews +SymLinksIFOwnerMatch
   #     #Order allow,deny
   #     # Order deny,allow
   #     # Allow from all
   # </Directory>
   ```

   The following figure shows an example.

   ![](image)

3. Re-start the Apache web server.

   `moxa@MOXA:~# sudo service apache2 restart`
ATTENTION
If you develop your own CGI applications, make sure that the CGI file is executable.

Saving Web Pages to a USB Storage Device

If you save web pages on a USB storage device, you can configure the Apache web server to open web pages saved on the USB storage device. You can obtain samples files shown in this section from the Moxa website.

1. Prepare the web pages and then save the pages on a USB storage device.
   You can download the web page test suite from the web site at http://www.w3.org/MarkUp/Test/HTML401.zip.

2. Uncompress the zip file to your computer and transfer the files (for example, using FTP) to the /media/usb0 directory on the DA-820.

3. Mount the root file system with write permission.

   MOXA:~# mount -o remount,rw /

4. Type vi /etc/apache2/sites-available/default and vi /etc/apache2/sites-available/default-ssl to edit the configuration files.

5. Change the DocumentRoot directory to the USB storage directory (for example, /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>

   "etc/apache2/sites-available/default"

   <VirtualHost *:443>
   ...
   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>

   "etc/apache2/sites-available/default-ssl"

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

   cd /etc/init.d
   ./apache2 restart
6. Start a web browser on your computer and type the LAN IP address of the DA-820-LX in the address bar.

7. Unmount the root directory back to read-only mode.

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

8. Re-start the Apache web server.

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

### Configuring IPTABLES

You can use the IPTABLES package to configure packet filtering, network address translation (NAT), firewall, and packet mangling in Linux.

The DA-820-LX supports three types of IPTABLES: Filter tables, NAT tables, and Mangle tables. Each table contains 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**.

**Filter Table**

The Filter table contains the following chains:

- INPUT chain
- OUTPUT chain
- FORWARD chain

**NAT Table**

The NAT table contains the following chains:

- PREROUTING chain—Translates the destination IP address (DNAT).
- POSTROUTING chain—Works after the routing process and before the Ethernet device process to translate the source IP address (SNAT).
- OUTPUT chain—Creates local packets.

**Sub-tables**

The Sub table contains the following chains:

- Source NAT (SNAT)—Changes the first source IP address of the packet.
- Destination NAT (DNAT)—Changes the first destination IP address of the packet.
- 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**

The Mangle table contains the following 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
     \                     /  \\
      \                   /   \\
       \                 /    \
        \               /     \
         \             /      \
          \           /        \
           \        /          \
            \      /            \
             \    /              \
              \  /                \
               \ /                  \
                \                     \
     NAT Table POSTROUTING Chain
     \                     /  \\
      \                   /   \\
       \                 /    \
        \               /     \
         \             /      \
          \           /        \
           \        /          \
            \      /            \
             \    /              \
              \  /                \
               \ /                  \
                \                     \
     Outgoing Packets
```
IPTABLES Modules

The DA-820-LX supports the following sub-modules. Make sure that you use the module that matches your application.

<table>
<thead>
<tr>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>arptable_filter.ko</td>
</tr>
<tr>
<td>arp_tables.ko</td>
</tr>
<tr>
<td>arpt_mangle.ko</td>
</tr>
<tr>
<td>ip_conntrack_ama.nda.ko</td>
</tr>
<tr>
<td>ip_conntrack_ftp.ko</td>
</tr>
<tr>
<td>ip_conntrack_h323.ko</td>
</tr>
<tr>
<td>ip_conntrack_irc.ko</td>
</tr>
<tr>
<td>ip_conntrack_ko</td>
</tr>
<tr>
<td>ip_conntrack_netbios.ns.ko</td>
</tr>
<tr>
<td>ip_conntrack_netlink.ko</td>
</tr>
<tr>
<td>ip_conntrack_pptp.ko</td>
</tr>
<tr>
<td>ip_conntrack_proto_sctp.ko</td>
</tr>
<tr>
<td>ip_conntrack_sip.ko</td>
</tr>
<tr>
<td>ip_conntrack_tftp.ko</td>
</tr>
<tr>
<td>ip_nat_amanda.ko</td>
</tr>
<tr>
<td>ip_nat_ftp.ko</td>
</tr>
<tr>
<td>ip_nat_h323.ko</td>
</tr>
<tr>
<td>ip_nat_irc.ko</td>
</tr>
<tr>
<td>ip_nat_ko</td>
</tr>
<tr>
<td>ip_nat_pptp.ko</td>
</tr>
<tr>
<td>ip_nat_sip.ko</td>
</tr>
<tr>
<td>ip_nat_snmp_basic.ko</td>
</tr>
<tr>
<td>ip_nat_tftp.ko</td>
</tr>
<tr>
<td>ip_queue.ko</td>
</tr>
<tr>
<td>iptable_filter.ko</td>
</tr>
<tr>
<td>iptable_mangle.ko</td>
</tr>
<tr>
<td>iptable_nat.ko</td>
</tr>
<tr>
<td>iptable_raw.ko</td>
</tr>
<tr>
<td>ip_tables.ko</td>
</tr>
<tr>
<td>ipt_addrtype.ko</td>
</tr>
<tr>
<td>ipt_ah.ko</td>
</tr>
<tr>
<td>ipt_CLUSTERIP.ko</td>
</tr>
<tr>
<td>ipt_dscp.ko</td>
</tr>
<tr>
<td>ipt_DSCP.ko</td>
</tr>
<tr>
<td>ipt_ecn.ko</td>
</tr>
<tr>
<td>ipt(ECN).ko</td>
</tr>
<tr>
<td>ipt_hashlimit.ko</td>
</tr>
<tr>
<td>ipt_iprange.ko</td>
</tr>
<tr>
<td>ipt_LOG.ko</td>
</tr>
<tr>
<td>ipt MASQUERADE.ko</td>
</tr>
<tr>
<td>ipt_NETMAP.ko</td>
</tr>
<tr>
<td>ipt_owner.ko</td>
</tr>
<tr>
<td>ipt_recent.ko</td>
</tr>
<tr>
<td>ipt_REDIRECT.ko</td>
</tr>
<tr>
<td>ipt_REJECT.ko</td>
</tr>
<tr>
<td>ipt SAME.ko</td>
</tr>
<tr>
<td>ipt_TCPMSS.ko</td>
</tr>
<tr>
<td>ipt_tos.ko</td>
</tr>
<tr>
<td>ipt_TOS.ko</td>
</tr>
<tr>
<td>ipt_TTL.ko</td>
</tr>
<tr>
<td>ipt_ULOG.ko</td>
</tr>
</tbody>
</table>

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 the `lsmod` command to check whether the `ip_tables` module has already been loaded in the DA-820-LX. Use the `modprobe` command to insert and enable the module.

ATTENTION

IPTABLES rules are used for packet filtering and NAT. Make sure that you set the IPTABLES rules correctly. An improper rule may prevent a remote host from connecting to the DA-820 on the LAN or through PPP. It is recommended that you use the VGA console to configure IPTABLES.

For more information on IPTABLES, go to the following web sites:

- http://www.linuxguruz.com/iptables/
The following sections show you how to configure and manage IPtables rules.

### Viewing Rules and Deleting Chains

This section describes the `iptables` commands to view rules and delete chains.

**Usage**

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

- `-t tables`: Table to manipulate (default: 'filter'). For example, nat or filter.
- `-L [chain]`: 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.

**Example**

```
# 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
```

### Defining Chain Policies

This section describes the commands you use to define chain policies.

**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-820-LX.
- `OUTPUT`: For locally-generated packets.
- `FORWARD`: For packets routed out through the DA-820-LX.
- `PREROUTING`: To alter packets as soon as they come in.
- `POSTROUTING`: To alter packets as they are about to be sent out.

**Example**

```
# 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.
Adding or Deleting Rules

This section describes the commands you use to add or delete rules.

Usage


-A: Append one or more rules to the end of the selected chain.
-1: 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-820-LX's port 137, 138, 139

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

Example 7: Log TCP packets that visit DA-820-LX's port 25.

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

Example 8: Load the ipt_mac module and drop all packets from MAC address 01:02:03:04:05:06.

```
# modprobe ipt_mac
```

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

ATTENTION

For Example 8, you must first use the command `# modprobe ipt_mac` to load the ipt_mac module.
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-820-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.

NOTE

NAT Example

The following figure shows a network example where the IP address of all packets leaving LAN1 are changed to 192.168.3.127 (using the ipt_MASQUERADE module).

Enabling NAT during System Startup

You can create a shell script to enable NAT during startup. The following shows an example script.

```
#!/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:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
export PATH
export PATH
export PATH
export PATH
# Step 3. Enable IP masquerade.
#ehco 1 > /proc/sys/net/ipv4/ip_forward
#modprobe ipt_MASQUERADE
#iptables 

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 or PPP access is almost identical to connecting directly to a network through the DA-820-LX Ethernet port. Since PPP is a peer-to-peer system, the DA-820-LX can also use PPP to link two networks (or a local network to the Internet) to create a Wide Area Network (WAN).

Connecting to a PPP Server Using a Dial-up Connection

You can use the following command to connect to a PPP server through a 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 ""' user username password password /dev/ttyS0 115200 crtscts modem
```

If the PPP server does not prompt for the username and password, enter the command 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/ttyS0 115200 crtscts modem
```

The following describes the parameters and options for the pppd command.

```
connect 'chat etc...'
```

This option sets the system to contact the PPP server.

The `chat` program is used to dial into 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).

**login: username word: password**  
Log in with username and password.

Note: 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.

**crtscs**  
Use hardware flow control between the computer and modem.

You must use this option if the baudrate is set at 115200.

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

**defaultroute**  
After the PPP link is established, set it as the default route (especially if the PPP link connects to the Internet).

**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 you have to enter a username and password, use the following command (note that noipdefault is optional):

```bash
pppd connect 'chat -v " " " ' noipdefault /dev/ttyS0 19200 crtscts
```

If you do not have to enter a username and password, use the following command (note that noipdefault is optional, and the username and password are both "root"):

```bash
pppd connect 'chat -v " " " ' user root password root noipdefault /dev/ttyS0 19200 crtscts
```

### Checking the Connection

After you have set up a PPP connection, you can use various methods to test the connection.

First, use the ipconfig command.

```bash
# /sbin/ifconfig
```

Depending on your distribution, the command might be located in a different directory. After executing the command, you should be able to see all of the network interfaces that are up.
The following figure shows an output example.

**ppp0** should be one of the network interfaces. **inet addr** indicates the IP address of the computer, and **P-t-P** indicates the IP address of the server.

```
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
```

Next, test using the ping command.

```
# ping z.z.z.z
```

where z.z.z.z is the address of a server. The output should be 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:~#
```

Then, use the **netstat** command to display the routing table.

```
# netstat -nr
```

You should see 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    10
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 to test the connection. Note that you will have to use numeric IP addresses unless you have configured **/etc/resolv.conf** correctly.
Configuring the Server for Incoming PPP Connections

**Method 1: pppd dial-in with pppd commands**

Use this method if you are using a modem and require login authentication.

```
# pppd /dev/ttyS0 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 (*) allows everyone to log in. The second star (*) allows every host to connect. The pair of double quotation marks (""") indicates that the file `/etc/passwd` can be used to check the password. The last star (*) allows any remote IP addresses to connect.

The following example does not check the username and password:

```
# pppd/dev/ttyS0 115200 crtscts modem 192.168.16.1:192.168.16.2
```

**Method 2: pppd dial-in with pppd script**

Create a dial-in script `/etc/ppp/peer/dialin` as shown in the following figure.

```
# 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-ttyS0.chat"

# Set up routing to go through this PPP link
defaultroute

# Default modem (you better replace this with /dev/ttySx!)
/dev/ttyS0

# 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-ttyS0.chat` as shown in the following figure.

```
SAY 'Auto Answer ON\n'
\nATS0=1
```

Start the `pppd` dial-in service.

```
# pppd call dialin
```

**NOTE** If you want to set up auto dial-in service, use the `respawn` command to execute the dial-in service in `/etc/inittab`. The following figure shows an example.

```
MOXA:~# mount -o remount,rw /dev/sda1 /
MOXA:~# echo "p0:2345:respawn:pppd call dialin" >> /etc/inittab
MOXA:~# umount /
```
Configuring PPPoE

Complete the following steps to configure PPPoE:

1. Connect the DA-820-LX’s LAN port to an ADSL modem directly using a cross-over cable, or through a HUB or switch.
2. Log in to the DA-820-LX as the root user.
3. Open the `/etc/ppp/chap-secrets` file in `vi` and add the following line:
   
   ```
   "username@hinet.net" * "password" *
   ```

   In the following example, `username@hinet.net` is the username obtained from the ISP and `password` is the corresponding password for the account.

   ```
   # 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" *
   ```

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

   # ATTENTION: The definitions here can allow users to log in 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
6. If you use LAN1 to connect to the ADSL modem, add the file `/etc/ppp/options.eth0`. Similarly, if you use LAN2, add `/etc/ppp/options.eth1`.

   ```
   name username@hinet.net
   mtu 1492
   mru 1492
   defaultroute
   noipdefault
   ```

   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 needed.

7. Set up DNS.

   If you are using DNS servers supplied by your ISP, add the nameserver information in the file `/etc/resolv.conf` in the following format:

   ```
   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
   ```

   Use the following command to create a `pppoe` connection:

   ```bash
   pppd eth0
   ```

8. The ADSL modem is connected to the **LAN1** port, use `eth0`. If the ADSL modem is connected to **LAN2**, use `eth1`. And similarly for other ports.

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.

10. If you want to disconnect the connection, use the `kill` command to kill the `pppd` process.
NFS (Network File System) Client

You can use Network File System (NFS) to mount a disk partition on a remote machine (as if it were on a local hard drive). This allows fast, seamless sharing of files across a network. NFS enables you to develop applications for the DA-820-LX without worrying about the amount of disk space that will be available. The DA-820-LX only supports NFS client protocol.

NOTE

For more information on NFS, visit the following web sites:
http://www.ietf.org/rfc/rfc1213.txt
http://www.faqs.org/rfcs/rfc1317.html

Complete the following steps to mount a remote NFS server:

1. Scan the NFS Server’s shared directory:
   
   ```
   # showmount -e HOST
   ```
   
   showmount: Shows the mount information of an NFS Server
   -e: Shows 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
   ```
   (where 192.168.3.100 is the IP address of the NFS server.)

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, set the configuration files in the `/etc/openvpn` directory, where script files and key files are stored. After an OpenVPN connection is established, the system performs all network data operations in the `/etc/openvpn` directory.
Ethernet Bridging for Private Networks on Different Subnets

This section describes the steps to configure Ethernet bridging for the example network shown in the following figure.

In this example, Host A represents the machine that belongs to OpenVPN A, and Host B belongs to OpenVPN B. The two remote subnets are configured for a different range of IP addresses. To connect these two networks over the Internet, the external interfaces of the OpenVPN machines should be configured to use a static IP address, or be connected to another device (such as a firewall or DSL modem).

1. Generate a preset shared key.
   
   ```
   # openvpn --genkey --secret secrouter.key
   ```

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

   **NOTE**
   A preshared key is stored in the directory /etc/openvpn/secrouter.key. You can use this key for testing purposes. We recommend that you create a new key dedicated for network deployment.

3. On the OpenVPN A machine, change 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
   ```

4. Configure the routing table in `/etc/openvpn/tap0-br.sh script`.
   
   ```
   #!/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
   ```

3-22
5. Configure the bridge interface in `/etc/openvpn/bridge`.

```bash
#!/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"
...
```

6. Restart the bridge script file to make the bridge interface configuration take effect.

```bash
#/etc/openvpn/bridge restart
```

7. On the OpenVPN B machine, change the remote address in the configuration file `/etc/openvpn/tap0-br.conf`.

```bash
# 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
```

8. Configure the routing table in `/etc/openvpn/tap0-br.sh script` file.

```bash
#!/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
```

9. Configure the bridge interface in `/etc/openvpn/bridge`.

```bash
#!/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="192.168.8.175"
...
10. Restart the bridge script file to make the bridge interface configuration take effect.
   
   /etc/openvpn/bridge restart

   **NOTE**
   You can select the cipher and authentication algorithm to use for the VPN connection. To show the list of cipher and authentication algorithms, use the following commands:
   
   openvpn --show-ciphers
   openvpn --show-auths

11. Start the OpenVPN client on the OpenVPN A and OpenVPN B machines.
   
   openvpn --config /etc/openvpn/tap0-br.conf

   If the message **Peer Connection Initiated with 192.168.8.173:5000** appears, the connection between OpenVPN machines has been established successfully on UDP port 5000.

   **NOTE**
   You can use the following command to 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, create the following symbolic links:
   
   ```
   ln -sf /etc/init.d/openvpn /etc/rc0.d/K80openvpn
   ln -sf /etc/init.d/openvpn /etc/rc6.d/K80openvpn
   ```

12. On each OpenVPN machine, check the routing table by typing the command `route`.

   For this example, both interface **eth1** and device **tap0** 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. When the OpenVPN program detects traffic on the virtual device, the traffic is sent to the peer.

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Genmask</th>
<th>Flags</th>
<th>Metric</th>
<th>Ref</th>
<th>Use</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.5.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth2</td>
</tr>
<tr>
<td>192.168.4.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>br0</td>
</tr>
<tr>
<td>192.168.3.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
<tr>
<td>192.168.30.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth3</td>
</tr>
<tr>
<td>192.168.8.0</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>br0</td>
</tr>
</tbody>
</table>

13. To create an indirect connection to Host B from Host A, add the following routing entry:

   # 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, add the following routing entry:

   # route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0

   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.

14. To terminate the OpenVPN programs, type the command:

   # killall -TERM openvpn
Ethernet Bridging for Private Networks on the Same Subnet

This section shows you how to configure Ethernet bridging for the private networks on the same subnet as shown in the following figure.

The configuration procedure is the same as the for the previous example, except that you must type "#" to comment out the line starting with "up" in the files `/etc/openvpn/tap0-br.conf` (OpenVPN A) and `/etc/openvpn/tap0-br.conf` (OpenVPN B).

The following shows an example.

```plaintext
# 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
```
Configuring IP Routing

This section shows you how to configure IP routing for the network example shown in the following figure.

1. On the OpenVPN A machine, change 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
   ```

   **NOTE** In the `tun.conf` file, the first argument of `ifconfig` is the local internal interface and the second argument is the internal interface at the remote peer.

2. Configure the routing table in script file `/etc/openvpn/tun.sh`.

   ```
   #!/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
   ```

   **NOTE** In the `tun.sh` file, `$5` is the argument that the OpenVPN program passes to the script file. Its value is the second argument of `ifconfig` in the `tun.conf` file.
3. On the OpenVPN B machine, change the remote address in configuration file `/etc/openvpn/tun.conf`.

```plaintext
# 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
```

4. Configure the routing table in script file `/etc/openvpn/tun.sh`.

```plaintext
#!/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
```

5. Check the routing table after you run the OpenVPN programs. Type the command `route`.

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Genmsk</th>
<th>Flags</th>
<th>Metric</th>
<th>Ref</th>
<th>Use</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.4.174</td>
<td>*</td>
<td>255.255.255.255</td>
<td>UH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>tun0</td>
</tr>
<tr>
<td>192.168.4.0</td>
<td>192.168.4.174</td>
<td>255.255.255.0</td>
<td>UG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>tun0</td>
</tr>
<tr>
<td>192.168.2.0</td>
<td>*</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth1</td>
</tr>
<tr>
<td>192.168.8.0</td>
<td>*</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
</tbody>
</table>
The following topics are covered in this chapter:

- Getting Product Serial Number
- RTC (Real Time Clock)
- UART
  - Setting the UART Mode
- Programmable LED Indicators
- Watch Dog Timer
  - The Watchdog Device IOCTL Commands
  - Examples
- TPM
Getting Product Serial Number

You can use the `dmidecode` command to display the product information (for example, device manufacturer and device serial number).

```
moxa@Moxa:~$ sudo dmidecode -s "baseboard-manufacturer"
Moxa
moxa@Moxa:~$ sudo dmidecode -s "baseboard-serial-number"
TACCA1000000
```

The following table shows the list of command options to display other product information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bios-vendor</td>
<td></td>
</tr>
<tr>
<td>bios-version</td>
<td></td>
</tr>
<tr>
<td>bios-release-date</td>
<td></td>
</tr>
<tr>
<td>system-manufacturer</td>
<td></td>
</tr>
<tr>
<td>system-product-name</td>
<td></td>
</tr>
<tr>
<td>system-version</td>
<td></td>
</tr>
<tr>
<td>system-serial-number</td>
<td></td>
</tr>
<tr>
<td>system-uid</td>
<td></td>
</tr>
<tr>
<td>baseboard-manufacturer</td>
<td></td>
</tr>
<tr>
<td>baseboard-product-name</td>
<td></td>
</tr>
<tr>
<td>baseboard-version</td>
<td></td>
</tr>
<tr>
<td>baseboard-serial-number</td>
<td></td>
</tr>
<tr>
<td>baseboard-asset-tag</td>
<td></td>
</tr>
<tr>
<td>chassis-manufacturer</td>
<td></td>
</tr>
<tr>
<td>chassis-type</td>
<td></td>
</tr>
<tr>
<td>chassis-version</td>
<td></td>
</tr>
<tr>
<td>chassis-serial-number</td>
<td></td>
</tr>
<tr>
<td>chassis-asset-tag</td>
<td></td>
</tr>
<tr>
<td>processor-family</td>
<td></td>
</tr>
<tr>
<td>processor-manufacturer</td>
<td></td>
</tr>
<tr>
<td>processor-version</td>
<td></td>
</tr>
<tr>
<td>processor-frequency</td>
<td></td>
</tr>
</tbody>
</table>

RTC (Real Time Clock)

The device node is located in `/dev/rtc`. The DA-820-LX supports standard Linux simple RTC control. You must include `<linux/rtc.h>`.

- **Function: RTC_RD_TIME**
  ```
  int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
  Description: read time information from the RTC. It will return the value on argument 3.
  ```
- **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 embedded serial ports on DA-820-LX use the general 8250 device driver. The tty device nodes are `/dev/ttyS0` and `/dev/ttyS1`. The embedded serial ports support standard Linux termios control with RS-232/422/485 serial ports.
Setting the UART Mode

You can use the `setuartmode` command to configure the UART operating mode.

Get/set the serial port mode utility

Usage: `setuartmode -n [the nth port] [-g|-s] [-h]

- h: Show this information.
- n [the nth port]: Indicate the n-th serial port.
  1 for the first serial port (default)
  2 for the second serial port.
- g: Get the n-th serial port interface type.
- s: Set the n-th serial port interface.
  0 for RS-232.
  1 for RS-485-2W.
  2 for RS-422/RS-485-4W.

The following table shows some `setuartmode` command examples.

<table>
<thead>
<tr>
<th>Command Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setuartmode -n 1 -s 0</td>
<td>Sets the first serial port to operate in RS-232 mode.</td>
</tr>
<tr>
<td>setuartmode -n 1 -s 1</td>
<td>Sets the first serial port to operate in RS-485-2W mode.</td>
</tr>
<tr>
<td>setuartmode -n 1 -s 2</td>
<td>Sets the first serial port to operate in RS-422/485-4W mode.</td>
</tr>
<tr>
<td>setuartmode -n 1 -g</td>
<td>Displays the operating mode of the first serial port.</td>
</tr>
<tr>
<td>setuartmode -n 2 -s 0</td>
<td>Sets the second serial port to operate in RS-232 mode.</td>
</tr>
<tr>
<td>setuartmode -n 2 -s 1</td>
<td>Sets the second serial port to operate in RS-485-2W mode.</td>
</tr>
<tr>
<td>setuartmode -n 2 -s 2</td>
<td>Sets the second serial port to operate in RS-422/485 mode.</td>
</tr>
<tr>
<td>setuartmode -n 2 -g</td>
<td>Displays the operating mode of the second serial port.</td>
</tr>
</tbody>
</table>

You can install a Moxa multi-port serial board (for example, CP-102UL, CP-114UL, or CP-118U) on the DA-820 computer. The tty device nodes are `/dev/ttyMUE0`, `/dev/ttyMUE1`, etc. You can use the `muestty` utility to configure the UART to function in RS-232/422/485 mode.

Usage: `muestty <operation> device`

Device: The MUE series device node

Operation: -h Help
- g Get interface and terminator type
- i intf Set interface type with options below
- t value Set termination resistor with options below

intf RS232 RS-232 mode
RS422 RS-422 mode
RS4852W RS-485 2-wire mode
RS4854W RS-485 4-wire mode
value NONTERM Non termination resistor
120TERM 120ohm termination resistor
The following table shows some `muestty` command examples.

<table>
<thead>
<tr>
<th>Command Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>muestty -i RS422 /dev/ttyMUE2</code></td>
<td>Sets the MUE interface.</td>
</tr>
<tr>
<td><code>muestty -t 120TERM /dev/ttyMUE2</code></td>
<td>Sets the MUE termination resistor.</td>
</tr>
</tbody>
</table>

By default, the serial interface on the DA-820 is set to RS-232 mode. You can use the `setinterface` command to change the serial port operation mode.

**setinterface** device-node [interface-no]

device-node: `/dev/ttyMn; n = 0,1,2,...`

interface-no: [see following table]:

<table>
<thead>
<tr>
<th>Interface Number</th>
<th>Operation Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Display current setting</td>
</tr>
<tr>
<td>0</td>
<td>RS-232</td>
</tr>
<tr>
<td>1</td>
<td>2-wire RS-485</td>
</tr>
<tr>
<td>2</td>
<td>RS-422</td>
</tr>
<tr>
<td>3</td>
<td>4-wire RS-485</td>
</tr>
</tbody>
</table>

For example, use the following commands to set `/dev/ttyM0` to operate in RS-422 mode.

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

**Programmable LED Indicators**

There are four programmable LED indicators on the front panel of the DA-820. The programmable LED device file is located in `/dev/pled`. Each LED can be accessed via the `/dev/pled` device node.

The following table shows some examples to control the programmable LED indicators.

<table>
<thead>
<tr>
<th>Command Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>echo 10000000 &gt; /dev/pled</code></td>
<td>Turns on the first LED and turns off all other LEDs.</td>
</tr>
<tr>
<td><code>echo 00000000 &gt; /dev/pled</code></td>
<td>Turns off all LEDs.</td>
</tr>
<tr>
<td><code>echo 01000000 &gt; /dev/pled</code></td>
<td>Turns on the second LED and turns off all other LEDs.</td>
</tr>
<tr>
<td><code>echo 11010000 &gt; /dev/pled</code></td>
<td>Turns on the first and last LEDs, and turns off the second and third LEDs.</td>
</tr>
</tbody>
</table>
Watch Dog Timer

You can enable the Watch Dog Timer (WDT) that acts like a watchdog function. The WDT sets the system to reboot if an application does not acknowledge within the time specified (between 1 millisecond to 255 seconds).

Debian project supports a watchdog daemon to check the health of your system. If programs are no longer executed, it sets the system to perform a hard reset. The standard watchdog driver and package have been installed in the DA-820.

You can use the `insserv` command to execute the watchdog function during system startup.

```
moxa@Moxa:~$ sudo insserv -v -d watchdog
[sudo] password for moxa:
insserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: enable service ../init.d/watchdog -> /etc/init.d/..../init.d/K01watchdog
ininsserv: creating .depend.boot
ininsserv: creating .depend.start
ininsserv: creating .depend.stop
moxa@Moxa:~$
```

Use the `ls l` command to check the runlevel of the watchdog function.

```
moxa@Moxa:~$ ls -l /etc/rc?.d/*watchdog*
```

```
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc0.d/K01watchdog -> ../init.d/watchdog
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc1.d/K01watchdog -> ../init.d/watchdog
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc2.d/S23watchdog -> ../init.d/watchdog
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc3.d/S23watchdog -> ../init.d/watchdog
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc4.d/S23watchdog -> ../init.d/watchdog
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc5.d/S23watchdog -> ../init.d/watchdog
-rwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc6.d/K01watchdog -> ../init.d/watchdog
moxa@Moxa:~$
```

The watchdog configure file is located in `/etc/watchdog.conf`. By default, the watchdog daemon is set to check the watchdog device every 60 seconds. You can configure real-time settings for the watchdog daemon to lock itself into memory and can never be swapped out. The real-time setting prevents any delay in acknowledgement under heavy system load.

You can configure the `/etc/watchdog.conf` file to enable watchdog and specify related settings.

```
... watchdog-device = /dev/watchdog
... interval = 60
... realtime = yes
... priority = -10
...
```

To remove watchdog from runlevels, use the following command.

```
moxa@Moxa:~# sudo insserv -r watchdog
```

The following command checks whether watchdog is in the runlevels.

```
moxa@Moxa:~# ls -l /etc/rc?.d/*watchdog*
ls: cannot access /etc/rc?.d/*watchdog*: No such file or directory
moxa@Moxa:~#
```
# The Watchdog Device IOCTL Commands

<table>
<thead>
<tr>
<th>IOCTL</th>
<th>Description</th>
<th>Input</th>
<th>Output</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDIOC_GETSUPPORT</td>
<td>This returns the support of the card itself</td>
<td>None</td>
<td>(struct watchdog_info *) arg</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
<tr>
<td>WDIOC_GETSTATUS</td>
<td>This returns the status of the card</td>
<td>None</td>
<td>(int *)arg</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
<tr>
<td>WDIOC_GETBOOTSTATUS</td>
<td>This returns the status of the card that was reported at bootup.</td>
<td>None</td>
<td>(int *)arg</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
<tr>
<td>WDIOC_SETOPTIONS</td>
<td>This lets you set the options of the card. You can either enable or disable the card this way.</td>
<td>None</td>
<td>(int *)arg</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
<tr>
<td>WDIOC_KEEPALIVE</td>
<td>This pings the card to tell it not to reset your computer.</td>
<td>None</td>
<td>None</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
<tr>
<td>WDIOC_SETTIMEOUT</td>
<td>Set the watchdog timeout</td>
<td>arg: 1 ~ 255 seconds</td>
<td>None</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
<tr>
<td>WDIOC_GETTIMEOUT</td>
<td>Get the current watchdog timeout.</td>
<td>None</td>
<td>arg: 1 ~ 255 seconds</td>
<td>On success, return 0. Otherwise, return &lt; 0 value.</td>
</tr>
</tbody>
</table>
Examples

The example file `watchdog-simple.c` acknowledges the watchdog every 10 seconds.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(void)
{
    int fd = open("/dev/watchdog", O_WRONLY);
    int ret = 0;
    if (fd == -1) {
        perror("watchdog");
        exit(EXIT_FAILURE);
    }
    while (1) {
        ret = write(fd, "\0", 1);
        if (ret != 1) {
            ret = -1;
            break;
        }
        sleep(10);
    }
    close(fd);
    return ret;
}
```

TPM

Through the TCG Software Stack (TSS) API, TPM services provides the following features:

1. RSA key pair generation
2. RSA encryption and decryption using PKCS v1.5 and OAEP padding
3. RSA signature and verification
4. Extend data into the TPM’s PCRs and log these events
5. Seal data to arbitrary PCRs
6. Random Number Generation
7. RSA key storage

For more information, see the chapter on Configuring the TMP Module or refer to the TrouSurS web site at http://trousers.sourceforge.net/
The DA-820 comes with PCI/PCI express expansion slots that allow users to install various expansion modules. This chapter describes how to program these modules with different communication interfaces.

The following topics are covered in this chapter:

- **Programming Serial Modules**
  - Configuring Serial Port Mode
  - Changing the Default Serial Mode
- **Programming the IRIG-B Module**
  - Configure the IRIG-B Time Sync Daemon
- **Programming the Fiber Module**
Programming Serial Modules

You can install serial expansion modules (for example, CP-102, CP-104, CP-112 or CP-118U) in the DA-820. The drivers for the expansion modules are already installed in the official DA-820-LX firmware.

After the system is started, use the `lsmod` command to check the `mxser` module.

```
moxa@Moxa:~# lsmod|more
Module      Size  Used by
mxser       50615 0
```

If the `mxser` module is loaded, the `/dev/ttyM0~/dev/ttyMn` device files are created in the system. The `/dev/ttyM0` device file is used to control serial port 0, `/dev/ttyM1` device file controls serial port 1, and similarly for the `/dev/ttyM2` device file, etc.

```
root@Moxa:~# ls -al /dev/ttyM*
crw-rw----T 1 root dialout 30,  0 Aug 14 13:37 /dev/ttyM0
```

Configuring Serial Port Mode

You can use the `setinterface` command to display a list of parameters and command options for serial port configuration.

```
Moxa:~# setinterface
Usage: setinterface device-node [interface-no]
device-node = /dev/ttyMn; n = 0,1,2,...
interface-no  - following:
none - to view now setting
0 - set to RS232 interface
1 - set to RS485-2WIRES interface
2 - set to RS422 interface
3 - set to RS485-4WIRES interface
Moxa:~#
```

The following lists the serial modes:

0 - Set to RS232 interface
1 - Set to RS485-2WIRES interface
2 - Set to RS422 interface
3 - Set to RS485-4WIRES interface
The following example checks the current interface setting. In this example, serial port 1 is set as an RS-485-2Wire interface. (M0 refers to port 1, M1 refers to port2, and so on)

```
Moxa: ~# setinterface /dev/ttyM0
Now setting is RS485-2WIRES interface.
```

The following example sets serial port 1 to RS-422 mode.

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

## Changing the Default Serial Mode

To change the default mode for serial interfaces, edit `/etc/udev/rules.d/96-moxa.rules`. Change the command line `RUN+="/bin/setinterface /dev/ttyM%n 0"`.

```
Moxa:~# vi /etc/udev/rules.d/96-moxa.rules
...
# Example to set the device, DA-SP08-I-DB, 0x1393:0x1180 default as 485-2W mode interface
KERNEL=="ttyM0", RUN="/bin/setinterface /dev/ttyM%n 1"
```

To set the default serial mode to RS-232, enter the following command.

```
RUN="/bin/setinterface /dev/ttyM%n 0"
```

To set the default serial mode to RS-485 2-wire, enter the following command.

```
RUN="/bin/setinterface /dev/ttyM%n 1"
```

To set the default serial mode to RS-422, enter the following command.

```
RUN="/bin/setinterface /dev/ttyM%n 2"
```

To set the default serial mode to RS-485 4-wire, use the following command.

```
RUN="/bin/setinterface /dev/ttyM%n 3"
```

Restart your computer.

```
Moxa:~# reboot
```

When the computer has been restarted, check if the setting has been loaded as the default value.

```
Moxa:~# setinterface /dev/ttyM0
Now setting is RS485-2WIRES interface.
Moxa:~#
```

### NOTE

Since the CP-102, CP-104, CP-112 and CP-118U expansion module use the same driver, mxser.ko, the `/dev/ttyM*` device scanning order is pre-defined in the driver. This means the device scanning order is fixed. You have to check the order in which the expansion modules are installed in the system and verify with the scan order by `/dev/ttyM*`.

## Programming the IRIG-B Module

You can install an IRIG-B module in the DA-820. The IRIG-B module uses the `moxa_irigb` driver. When the system boots up with the IRIG-B module installed, you can use the `lsmod` command to check if the IRIG-B module is loaded.

```
root@Moxa:~# lsmod|grep irig
moxa_irigb 12683 1
```
The following example shows that the moxa_irigb driver is loaded at startup.

```bash
root@Moxa:~# grep moxa_irigb /etc/modules
moxa_irigb
```

The following example shows that the IRIG-B time sync daemon is running with default settings.

```bash
root@Moxa:~# ps aux|grep ServiceSyncTime
root  3078  0.0  0.1  16136  1140 ?        S    10:43   0:00
/usr/sbin/ServiceSyncTime -t 1 -o 1 -i 10
```

The following figure shows the help information on using the ServiceSyncTime command.

```bash
root@Moxa:~# ServiceSyncTime -h
IRIG-B time sync daemon.
-t [interface type]
  0 - TTL
  1 - DIFF
  default value is 1
-o [port in output mode] - Enable the IRIG1, IRIG-B module 1, in output mode
  2 - Enable IRIG-B module 1 in output mode
  default value is 2
-I - inverse the input or output signal
-w [PPS width] - Set the wide of pulse per second in ms
  The PPS width should be 0 ~ 1000.
  default value is 0
-d - Disable time sync
  Default this daemon enables the IRIG-B time sync from source port to system time.
-s [Time Source] - The sync source from IRIG-B decoded module 1.
  2 - IRIG-B decoded module 1
  default value is 2
-i [Time sync interval] - The time interval in seconds to sync the IRIG-B time into system time.
  1 ~ 86400 Time sync interval. Default is 10 second.
```

The following example enables the IRIG-B time sync daemon to sync time from PORT2 (IRIG-B module 1) in TTL interface type every 10 seconds, and sets the PORT2 (IRIG-B module 1) output signal. The input and output signals are not inverted.

```bash
root@Moxa:~# ServiceSyncTime -t 0 -s 2 -o 2 -i 10
```

The following example enables the IRIG-B time sync daemon to sync time from PORT2 (IRIG-B module 1) in DIFF interface type every 10 seconds, and sets the PORT2 (IRIG-B module 1) output signal. The input and output signals are not inverted.

```bash
root@Moxa:~# ServiceSyncTime -t 1 -s 2 -o 2 -i 10
```

The following example enables the IRIG-B time sync daemon to sync time from PORT2 (IRIG-B module 1) in DIFF interface type every 10 seconds, and sets the PORT2 (IRIG-B module 1) output signal. Inverse the output signal if the cable connection is crossed.

```bash
root@Moxa:~# ServiceSyncTime -t 1 -s 2 -i 10 -I 1
```

The following example disables the IRIG-B time sync daemon.

```bash
root@Moxa:~# ServiceSyncTime -d
```
Configure the IRIG-B Time Sync Daemon

The IRIG-B time sync daemon is managed by the /etc/init.d/mx_irigb.sh script. The default configuration, MX_IRIGB_OPTS, is set in /etc/init.d/mx_irigb.sh.

```
root@Moxa:~# sudo vi /etc/init.d/mx_irigb.sh
...
MX_IRIGB_OPTS="-t 1 -o 2 -i 10"
...
```

After you edit the script file, restart the daemon.

```
root@Moxa:~# sudo service mx_irigb.sh restart
```

Programming the Fiber Module

You can install a fiber module in the DA-820. The fiber module uses the e1000e.ko driver that comes with the official DA-820-LX firmware. You can use the `ifconfig` command to configure this Ethernet interface.

```
Moxa:~# cat /proc/net/dev
eth0
__
eth4
eth5
Moxa:~# ifconfig eth6 192.168.9.127 up
__
Moxa:~# ifconfig eth4
eth4   Link encap:Ethernet  HWaddr 00:90:e8:00:e0:07
      UP BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
      Interrupt:45 Base address:0x8000
__
```
Managing Disks

The DA-820-LX computers come with a software-RAID management platform designed to serve as a front-end for data acquisition and industrial control applications. This chapter describes how to configure the volume supported by the Linux operating system.

The following topics are covered in this chapter:

- **Configuring Software RAID Using mdadm**
  - Creating a Software RAID Volume
  - Displaying Software RAID Status
  - Replacing a Failed Disk
Configuring Software RAID Using mdadm

In Linux, you can use the mdadm utility to manage software RAID devices. The RAID volume is built in Linux, not in BIOS. mdadm, which replaces the previous utility mdctl, enables you to administer and manager multiple devices (the “md” in the name).

Creating a Software RAID Volume

The DA-820-LX computers come with two SATA disk slots. You can manage linear, RAID0, or RAID1 volumes on these two SATA disks.

1. If a RAID device is created previously, stop it and create your own RAID devices.

   root@Moxa:~# cat /proc/mdstat
   Personalities: [raid0]
   md0: active raid0 sdc[0] sdb[1]
       195371008 blocks super 1.2 512k chunks
   root@Moxa:~# mdadm –stop /dev/md0
   
   root@Moxa:~# cat /proc/mdstat
   Personalities : [raid0]
   Unused devices: <none>

2. Unmount the disks if they are mounted.

   root@Moxa:~# umount /media/disk1p1
   root@Moxa:~# umount /media/disk2p1
   root@Moxa:~# /etc/init.d/mxhtspd.sh stop

3. Create the partitions on the disk.

   root@Moxa:~# fdisk /dev/sdb1
   root@Moxa:~# fdisk /dev/sdc1
   root@Moxa:~# fdisk /dev/sdd1
   root@Moxa:~# fdisk /dev/sde1

4. Create the RAID volume.

   The following figure shows the mdadm options that you can use for RAID volume creation. You can choose to create a linear mode, striping mode, or mirror mode in RAID volume.

   -C: create
   -v: verbose
   -l: RAID level, options are: linear, raid0, 0, stripe, raid1, 1, mirror, raid4, 4, raid5, 5, raid6, 6, raid10, 10, multipath, mp, faulty. Obviously some of these are synonymous.
   -n: the number of disks

5. Create a software RAID volume.

   The following example creates a linear mode software RAID.

   root@Moxa:~# mdadm –Cv -llinear -n2 /dev/md0 /dev/sd{b,c}1

   The following example creates a striping mode software RAID 0.

   root@Moxa:~# mdadm –Cv -l0 -n2 /dev/md0 /dev/sd{b,c}1

   The following example creates a mirror mode software RAID 1.

   root@Moxa:~# mdadm –Cv -l1 -n2 /dev/md0 /dev/sd{b,c}1

   The following example creates a mirror mode software RAID 5.

   root@Moxa:~# mdadm –Cv -l5 -n3 /dev/md0 /dev/sd{b,c,d}1
The following example creates a mirror mode software RAID 10.

```bash
root@Moxa:~# mdadm --create /dev/md0 /dev/sdb1[bcde]1
```

6. Check the RAID device information in /proc/mdstat.

The following example shows the status of RAID 10 and linear RAID.

```bash
root@Moxa:~# cat /proc/mdstat
Personalities: [raid0] [linear]
md0: active raid0 sdb1[1] sdc1[0]
    19539504 blocks super 1.2 OK rounding
Unused devices: <none>
```

The following example shows the status of RAID 1.

```bash
root@Moxa:~# cat /proc/mdstat
Personalities: [linear] [raid10] [raid5] [raid4] [raid1]
    488252736 blocks super 1.2 [4/4] [UUUU]
    resync=PENDING
Unused devices: <none>
```

The following example shows the status of RAID 5.

```bash
root@Moxa:~# cat /proc/mdstat
Personalities: [linear] [raid10] [raid5] [raid4] [raid1]
md0: active (auto-read-only) raid5 sde1[4](S) sdd1[2] sdc1[1] sdb1[0]
    1464757248 blocks super 5 level 5, 512k chunk, algorithm 2 [4/3] [UUU_]
Unused devices: <none>
```

The following example shows the status of RAID 10.

```bash
root@Moxa:~# cat /proc/mdstat
Personalities: [linear] [raid10]
    976504832 blocks super 512K chunks 2 near-copies [4/4] [UUU_]
    resync=PENDING
Unused devices: <none>
```

7. Format the RAID volume.

```bash
root@Moxa:~# mkfs.ext4 /dev/md0
```

8. Mount the RAID volume.

```bash
root@Moxa:~# mount /dev/md0 /mnt/raid
```

9. Start the RAID volume automatically for the next system startup.

If you want to start the array automatically, edit the /etc/mdadm/mdadm.conf file.

```bash
root@Moxa:~# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

The following figure shows an example.

```bash
DEVICE /dev/sdb1 /dev/sdc1
CREATE owner=root group=disk mode=0660 auto=yes
HOMEHOST <system>
MAILADDR your_email@xxx.com
ARRAY /dev/md0 metadata=1.2 name=Moxa:0 UUID=45ae9db6f30741ec:b22eff98:2dadbl2d
```

10. Edit the /etc/fstab file and add the following line to mount the RAID volume.

```bash
/dev/md0 /mnt/raid ext4 defaults 0 2
```
11. Unmount the root file system and reboot.
   
   The array should be started and mounted at /mnt/raid.

   ```
   root@Moxa:~# umount /
   ```

### Displaying Software RAID Status

You can view the software RAID status by displaying `/proc/mdstat`.

The following example shows the status while the array is running.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear]
md0 : active linear sdb1[1] sdc1[0]
      23436724 blocks super 1.2 0k rounding
```

The following example shows the status while the array is not running.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear]
unused devices: <none>
```

### Replacing a Failed Disk

If the array is running in mirror mode and one of the disks fails, you should replace the failed disk with a new one.

1. Check the status of the RAID array. In the following example, sdb1[0](F) indicates that the sdb disk is failed.

   ```
   md1 : active raid1 sdb1[1] sdc1[0] (F)
       17920384 blocks [2/1] [ _U ]
   ```

   You can simulate a disk failure by entering the following command.

   ```
   root@Moxa:~# mdadm --manage /dev/md0 --fail /dev/sdb1
   mdadm: set /dev/sdb1 faulty in /dev/md0
   root@Moxa:~# sync
   ```

2. Remove a failed disk from the RAID array.

   ```
   root@Moxa:~# mdadm --r /dev/md0 /dev/sdb1
   mdadm: hot removed /dev/sdb1 from /dev/md0
   ```

3. Replace the failed disk drive and add the drive volume into the RAID array.

   ```
   root@Moxa:~# mdadm --a /dev/md0 /dev/sdb1
   ```

4. Display mdstat to check the RAID array status. The following figure shows that the RAID array has automatically recovered.

   ```
   root@Moxa:~# cat /proc/mdstat
   Personalities : [raid1]
   md0 : active raid1 sdb1[0] sdc1[1]
       7806522 blocks super 1.2 [2/1] [ _U ]
       [===>........................] recovery = 10.6% (831488/7806522) finish=0.9min
   speed=118784K/sec
   unused devices: <n>
   ```
DA-820-LX Series integrate a Trusted Platform Module (TPM) that provides added protection in the system.

The following topics are covered in this chapter:

- Trusted Platform Module (TPM) and TrouSerS
- Enabling the TPM
- Starting TPM Services
- Initializing the TPM
- Getting the Public Endorsement Key
- Sealing and Unsealing Data
Configuring the TPM Module

Trusted Platform Module (TPM) and TrouSerS

TPM is a microcontroller that can securely store information such as passwords, certificates, or encryption keys which are used to authenticate the platform. TMP provides hardware-based data protection because the private key used to protect the data is never exposed in the clear outside of the TPM's own internal memory area.

A TPM can also be used to store platform measurements to help ensure a trusted platform. Data can also be protected by these measurements as well as requiring the platform to be in the same configuration to access the data as when the data was first protected.

The TPM specification was written by a computer industry consortium called the Trusted Computing Group (TCG). TrouSerS implements the TCG Software Stack (TSS) that contains the tcsd daemon and the TPM tool for you to access to and communicate with the TPM. These packages are pre-installed on the DA-820-LX.

The following table lists the supported TPM tool commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpm_changeownerauth</td>
<td>Change the authorization data associated with the owner or SRK.</td>
</tr>
<tr>
<td>tpm_clear</td>
<td>Return the TPM to the default state (unowned, disabled, inactive).</td>
</tr>
<tr>
<td>tpm_createek</td>
<td>Create an Endorsement Key pair in the TPM.</td>
</tr>
<tr>
<td>tpm_getpubek</td>
<td>Display the public portion of the Endorsement Key in the TPM.</td>
</tr>
<tr>
<td>tpm_resedalock</td>
<td>Reset the dictionary attack lock for the user (requires owner authentication).</td>
</tr>
<tr>
<td>tpm_restrictpubek</td>
<td>Restrict the ability to display the public portion of the Endorsement Key to the owner.</td>
</tr>
<tr>
<td>tpm_revokeek</td>
<td>Revoke the Endorsement Key pair of the TPM.</td>
</tr>
<tr>
<td>tpm_sealddata</td>
<td>Seal input data to the system TPM.</td>
</tr>
<tr>
<td>tpm_selftest</td>
<td>Request the TPM to perform selftest and report.</td>
</tr>
<tr>
<td>tpm_setactive</td>
<td>Change the TPM active state.</td>
</tr>
<tr>
<td>tpm_setclearable</td>
<td>Disable the TPM clear operation.</td>
</tr>
<tr>
<td>tpm_setenable</td>
<td>Change the TPM enable state.</td>
</tr>
<tr>
<td>tpm_setoperatorauth</td>
<td>Set the operator authorization value in the TPM.</td>
</tr>
<tr>
<td>tpm_setownable</td>
<td>Change if the TPM allows tpm_takeownership operation.</td>
</tr>
<tr>
<td>tpm_setpresence</td>
<td>Change the TPM physical presence states or settings.</td>
</tr>
<tr>
<td>tpm_takeownership</td>
<td>Set up an owner on the TPM.</td>
</tr>
<tr>
<td>tpm_version</td>
<td>Display the TPM version and manufacturer information.</td>
</tr>
</tbody>
</table>

The following table lists the PKCS#11 data management commands of the TPM tool.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpmtoken_import</td>
<td>Import an X.590 certificate and/or an RSA key pair into the user’s TPM PKCS#11 data store.</td>
</tr>
<tr>
<td>tpmtoken_init</td>
<td>Initialize the user’s TPM PKCS#11 data store.</td>
</tr>
<tr>
<td>tpmtoken_objects</td>
<td>Display the objects in the user’s TPM PKCS#11 data store.</td>
</tr>
<tr>
<td>tpmtoken_protect</td>
<td>Encrypt or decrypt data using a symmetric key stored in the user’s TPM PKCS#11 data store.</td>
</tr>
<tr>
<td>tpmtoken_setpasswd</td>
<td>Change the passwords associated with the user’s TPM PKCS#11 data store.</td>
</tr>
</tbody>
</table>

NOTE
Access following links for more information about TPM and TrouSerS TPM specification:
http://www.trustedcomputinggroup.org/resources/tpm_main_specification
http://trousers.sourceforge.net/
http://ibmswtpm.sourceforge.net/tpm_tss.html
Enabling the TPM

To start using the TPM on the DA-820, you must first enable the TPM function from boot loader and start the related services in the system.

1. Turn on the DA-820 and press <F2> to enter the BIOS configuration screen.
2. Select Security > TPM Operation > Enable and Activate.
4. Restart the DA-820 to make the changes take effect.

Starting TPM Services

After you enable the TPM, you can start the trousers and opencryptoki services.

The following figure shows the command example. For information on using the insserv and /etc/init.d/command, see the Enabling and Disabling Daemons section.

```bash
moxa@Moxa:~$ sudo insserv -d trousers
moxa@Moxa:~$ sudo insserv -d opencryptoki
moxa@Moxa:~$ sudo /etc/init.d/trousers start
moxa@Moxa:~$ sudo /etc/init.d/opencryptoki start
```

The following command example shows the TMP version.

```bash
moxa@Moxa:~$ sudo tpm_version
TPM 1.2 Version Info:
  Chip Version: 1.2.3.17
  Spec Level: 2
  Errata Revision: 2
  TPM Vendor ID: IFX
  Vendor Specific data: 03110008 00
  TPM Version: 01010000
  Manufacturer Info: 49465800
```

**NOTE** Before you enter a TMP command, make sure that the TMP daemon is running; otherwise, the following error message appears. 
```
Tspi_Context_Connect failed: 0x00003011 - layer=tsp, code=0011 (17), Communication failure
```

Initializing the TPM

The first step to using the TPM is to configure ownership settings.

Enter the `tpm_takeownership` command and enter the owner password and SRK password.

```bash
moxa@Moxa:~$ sudo tpm_takeownership
Enter owner password:
Confirm password:
Enter SRK password:
Confirm password:
```

Enter owner password and SRK password twice as it requested. Notice that owner and SRK passwords which are very important and must not be lost.
NOTE
You can configure the ownership settings once. You must remember the passwords you configure and store them in a secure location.

If the following error message appears, the endorsement key is not set. Enter the sudo tpm_createek command to create an endorsement key.

Tspi_TPM_GetPubEndorsementKey failed: 0x00000023 - layer=tpm, code=0023 (35), No EKlease make sure .....
Sealing and Unsealing Data

You can use the `tpm_sealdata` and `tpm_unsealdata` commands to seal and unseal sensitive data.

The following table describes the command options.

<table>
<thead>
<tr>
<th>Parameter Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-i, --infile FILE</code></td>
<td>Filename containing the key to seal/unseal. The default is STDIN.</td>
</tr>
<tr>
<td><code>-o, --outfile FILE</code></td>
<td>Filename to write sealed/unsealed key to. The default is STDOUT.</td>
</tr>
<tr>
<td><code>-p, --pcr NUMBER</code></td>
<td>PCR to seal data to. The default is none. This option can be specified multiple times to choose more than one PCR.</td>
</tr>
</tbody>
</table>

The `tpm_sealdata` command retrieves random data from the TPM. To do this, the `tpmGetRandom` function invokes the `Tspi_TPM_GetRandom()` method of the TPM class. Then, the `tpm_sealdata` command sets the SRK policy using the Policy and Context classes. The next functions build an RSA key object that will be created by the TPM. Then, an RSA key is created and loaded. The subsequent functions build an encrypted data object that will hold the encrypted version of the symmetric key. The final functions encrypt the given data and seal it to the symmetric key. It is possible to invoke this command with several command line parameters.

The following figure shows the `tpm_sealdata` command example.

```
moxa@Moxa:~$ tpm_sealdata -i secret -o secret.enc -p 12 -p 14
Enter SRK password: [redacted]
```

The following figure shows the `tpm_unsealdata` command example.

```
moxa@Moxa:~$ tpm_unsealdata -i secret.enc -o plain
```
The DA-820-LX computers come with 64-bit CPU and hardware virtualization support that allow you to use the DA-820 as a VMWare ESXi virtualization server.

This chapter describes how to manage the peripherals on the VMWare ESXi host from a VMWare ESXi guest system.

The following topics are covered in this chapter:

- About VMWare ESXi Peripheral Control
- Installing the VMWare ESXi Host Driver
  - Verifying Package Installation
  - Removing a Package
- Installing the VMWare ESXi VMCI Server
- VMWare VMCI Client Example
- Device Control Layer and API definitions
- VMCI Utility Example Code
- Compiling the VMCI Utility Example in Linux
- Compiling the VMCI Utility Example in Windows
About VMWare ESXi Peripheral Control

VMWare ESXi is a free bare-metal hypervisor that virtualizes servers to enable you to consolidate your applications on a system. User can install a Windows or Linux guest operating system running on VMWare ESXi.

VMWare can access most PCI cards via PCI pass-through but not for some I/O port devices. For these I/O port devices, you can use VMCI communication to control the physical device on VMWare ESXi 5.x.

The following figure shows an overview of the communication between the VMWare guest system and VMWare ESXi 5.x. In this example, a VMCI server and mx_vmci_server running on VMWare ESXi 5.x are used to communicate with the VMCI client (mx_exsi_pled, mx_exsi_usb_power and mx_exsi_set_uart8250_mode).

The client and server use VMWare VMCI socket for communication. The VMCI client open source codes contain the VMCI client utility, mx_exsi_pled.c, mx_exsi_usb_power.c, and mx_exsi_set_uart8250.c. You can reference this code to develop your own application.

<table>
<thead>
<tr>
<th>VMWare Guest OS (Windows/Debian Wheezy amd64)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pled control</td>
</tr>
</tbody>
</table>

VMCI communication

mx_vmci_server

<table>
<thead>
<tr>
<th>VMWare ESXi 5.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA-820PLED driver</td>
</tr>
</tbody>
</table>

Moxa Embedded Computer

Installing the VMWare ESXi Host Driver

To control the non-PCI based peripherals, you must install the VMWare ESXi host drivers using the esxcli utility.

1. Set the system in maintenance mode.
   ```bash
   # esxcli system maintenanceMode set -e true -t 0
   ```
2. Change the host acceptance level to CommunitySupported.
   ```bash
   # esxcli software acceptance set --level=CommunitySupported
   ```
3. Install the programmable LED, USB power control, and UART mode driver packages.
4. Install the programmable LED package.
   ```bash
   # esxcli software vib install -v /tmp/DA820PLED.vib
   ```
5. Install the USB power control package.
   ```bash
   # esxcli software vib install -v /tmp/DA820USB.vib
   ```
6. Install the UART mode control package driver.
   ```bash
   # esxcli software vib install -v /tmp/DA820UARTM.vib
   ```
7. Restart VMWare ESXi.
Verifying Package Installation

After you install all the packages, you can verify that the packages have been installed properly.

The following command example verifies the DA820PLED package installation.

```
# esxcli software vib list|grep DA820PLED
```

The following command example verifies the DA820USB package installation

```
# esxcli software vib list|grep DA820USB
```

The following command example verifies the DA820UARTM package installation

```
# esxcli software vib list|grep DA820UARTM
```

Removing a Package

You can use the `esxcli` command to remove a package from VMWare ESXi.

The following command example uninstalls the DA820PLED package.

```
# esxcli software vib remove -n DA820PLED
```

NOTE  To make the changes take effect, you must restart VMWare ESXi after you uninstall a package.

Installing the VMWare ESXi VMCI Server

On VMWare ESXi, DA820VMCIS.vib is the VMCI server. You can use the `esxcli` command with the `-f` option to force install the `DA820VMCIS.vib` package. The following figure shows a command example.

```
# esxcli software vib install -v /tmp/DA820VMCIS.vib -f
```

Installation Result

```
Message: Operation finished successfully.
Reboot Required: false
VIBs Installed: Moxa_bootbank_DA820VMCI_1.0
VIBs Removed:
VIBs Skipped:
```

Then, restart VMWare ESXi to make the changes take effect. After the system starts up, check that `mx_vmci_server` is running for the VMCI client utility to connect and control programmable LED, USB power on the front and rear panels, or UART mode.

```
# ps |grep mx_vmci_server
34564 34564 mx_vmci_server /usr/lib/vmware/misc/bin/mx_vmci_server
```
VMWare VMCI Client Example

The following figure shows sample code in the MCI_Client file that controls the programmable LED, USB power on the front and rear panels, and UART mode.

mx_exsi_pled.c: The VMCI client utility for controlling the programmable LED.
mx_exsi_usb_power.c: The VMCI client utility for controlling the front/rear end USB power.
mx_exsi_set_uart8250_mode.c: The VMCI client utility for setting the UART RS-232/422/485 mode.
device_control.h: Declare the device control API, such as device_open(), device_close(), ...
device_control.c: Implement the device control API, such as device_open(), device_close(), ...
front_end_usb_power_ctrl.h: Declare the front end USB power control API.
front_end_usb_power_ctrl.c: Implement the front end USB power control API.
rear_end_usb_power_ctrl.h: Declare the rear end USB power control API.
rear_end_usb_power_ctrl.c: Implement the rear end USB power control API.
pled_ctrl.h: Declare the programmable LED control API.
pled_ctrl.c: Implement the programmable LED control API.

Device Control Layer and API definitions

This section includes the definition for the device_contrl function to define a virtual layer for controlling the programming LED, USB-powered device on the front and rear panels.

int device_open (char* type)
/*
 * @brief open a device by given information
 * @param type the device name (pled, front_end_usb_power,
 * rear_end_usb_power, uart8250_mode,...)
 * @return
 * > 0: file descriptor;
 * <= 0 : Error
 */
/* pled */
int fd_pled = device_open("pled");
/* USB power */
int fd_usb_power_front = device_open("front_end_usb_power");
int fd_usb_power_rear = device_open("rear_end_usb_power");
/* uart8250_mode */
int fd_uart = device_open("uart8250_mode");
void device_close (int file_descriptor)
/*
 * @brief close a device
 * @param The file descriptor as an unique identifier return from device_open
 * @return void
 */
/* pled */
device_close(fd_pled);
/* USB power */
device_close(fd_front_end_usb_power);
device_close(fd_rear_end_usb_power);
/* uart8250_mode */
device_close(fd_uart);
int device_list (int file_descriptor);

/*
 * @brief list out the available devices in the given address
 * @param The file descriptor as an unique identifier return from device_open
 * @return number of device,
 * > 0 : The available number of devices. The device index starts from 1.
 * <= 0 : Error occurred.
 */

/* pled
 * example of the devices may be:
 * file_descriptor = fd_pled
 * the return num should be 8 of DA-820-LX
 */
int num = device_list(fd_pled);

/* front_end_usb_power
 * example of the devices may be:
 * file_descriptor = fd_front_end_usb_power
 * the return num should be 1
 */
int num = device_list(fd_front_end_usb_power);

/* rear_end_usb_power
 * example of the devices may be:
 * file_descriptor = fd_rear_end_usb_power
 * the return num should be 1
 */
int num = device_list(fd_rear_end_usb_power);

/* uart8250_mode
 * example of the devices may be:
 * file_descriptor = fd_uart
 * the return num should be 2
 */
int num = device_list(fd_uart);

int device_get (int file_descriptor, int index, int &value);

/*
 * @brief get some data of the specified device
 * @param The file descriptor as an unique identifier return from device_open
 * @param the index of the corresponding device.
 * @param the return from device_get()
 * @return
 *    >= 0: Success;
 *    < 0 : Error
 */

/* pled */
int value;

/* The first LED index start from 1 */
int res = device_get(fd_pled, 1, &value);

/* USB power */
int on_off; /* on: 1; off: 0 */
int res = device_get(fd_front_end_usb_power, 1, &on_off); /* Front */
int res = device_get(fd_rear_end_usb_power, 1, &on_off); /* rear */

/* UART mode */
/* 1 for the first serial port; 2 for the second serial port */
/* mode: 0 - RS232; 1 - RS485-2WIRES; 2 - RS422/RS485-4WIRES */
int res = device_get(fd_pled, 1, &mode);
int device_set (int file_descriptor, int index, int value);
/*
 * @brief set some data of the specified device
 * @param The file descriptor as an unique identifier return from device_open
 * @param the index of the corresponding device.
 * @param the value to be set to the device
 * @return
 *     >= 0: Success;
 *     < 0 : Error
 */
int on_off=1; /* on: 1; off: 0 */
int res = device_set(fd_pled, 1, on_off);
/* USB power front end */
int on_off=1; /* on: 1; off: 0 */
int res = device_set(fd_front_end_usb_power, 1, on_off);
int on_off=0; /* on: 1; off: 0 */
int res = device_set(fd_rear_end_usb_power, 1, on_off);
/* UART mode */
/* 1 for the first serial port; 2 for the second serial port */
/* mode: 0 - RS232; 1 - RS485-2WIRES; 2 - RS422/RS485-4WIRES */
int result = device_set(fd, 1, mode);
int device_read (char* type, char* data, size_t size);
/*
 * @brief read data from the specified device
 * @param The file descriptor as an unique identifier return from device_open
 * @param data read from the device.
 * @param size the size of the data read from the device.
 * @return
 *   0   : End of file
 *   >   : The number of data read from the device
 *   <0  : Error
 */
unsigned char data;
int ret = device_read(fd_pled, &data, sizeof(data));
/* Front end and rear end USB power control */
unsigned char data;
int ret = device_read(fd_front_end_usb_power, &data, sizeof(data));
/* Front end USB power */
int ret = device_read(fd_rear_end_usb_power, &data, sizeof(data));
/* UART mode */
/* device_read() for UART mode is not supported now */
int device_write (char* type, char* data, size_t size);
/*
 * @brief write data to the specified device
 * @param The file descriptor as an unique identifier return from device_open
 * @param data the data written to the device.
 * @param size the size of the data read from the device.
 * @return
 *   0   : End of file
 *   >0  : The number of data written to the device
 *   <0  : Error
 */
VMCI Utility Example Code

The `vmci_example.tar.gz` package contains the `mx_esxi_pled.c`, `mx_esxi_set_uart8250_mode.c`, and `mx_esxi_usb_power.c` source files.

In this example, `mx_esxi_pled.c` uses the following functions to control the programmable LED:

- `device_open()`
- `device_get()`
- `device_set()`
- `device_write()`
- `device_read()`

The following shows the example code.

```c
#include <sys/types.h>
/*
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifndef WIN32
#include <sys/un.h>
#include <unistd.h>
#endif
#include <errno.h>
#include "vmci_sockets.h"
#include "device_control.h"
#include "device_control.h" /* Define the packet format */

void usage(char *name) {
    printf("Get/set the programmable LED utility\n");
    printf("Usage: %s -l -n [-r|-w] [-g|-s] [-h]\n", name);
    printf("Show the mx_esxi_pled information if no argument apply.\n");
    printf("-h: Show this information.\n");
    printf("-l: List the number of LEDs.\n");
    printf("-n: Indicate the n-the LED.\n");
    printf("-r: Read the LED bitmap.\n");
    printf("-w: Write the bitmap to the LED.\n");
    printf("-g: Get the LED port value.\n");
    printf("-s: Set the value to the LED port.\n");
}
#endif WIN32
```
extern int getopt(int nargc, char *const *nargv, const char *ostr);
#endif
extern int optind, opterr, optopt;
extern char *optarg;
/* entry point */
int main(int argc, char *argv[])
{
    int fd, result;
    int num;
    int value = 0;
    int nth = 0;
    int bWrite = 0;
    int bRead = 0;
    int bSet = 0;
    int bGet = 0;
    int bList = 0;
    char led_bitmap[]="00000000";
    char optstring[] = "hn:s:gw:rl";
    char c;
    if ( argc == 1 ) {
        usage(argv[0]);
        return 0;
    }
    while ((c = getopt(argc, argv, optstring)) != -1)
    {
        switch (c) {
            case 'h':
                usage(argv[0]);
                return 0;
            case 'n':
                nth=atoi(optarg);
                if ( nth <=0 || nth >8 ) {
                    printf(" nth:%d is not in 1 and 8\n", nth);
                    return 0;
                }
                break;
            case 's':
                bSet = 1;
                value=atoi(optarg);
                break;
            case 'g':
                bGet = 1;
                break;
            case 'w':
                bWrite = 1;
                strcpy(led_bitmap, optarg);
                break;
            case 'r':
                bRead = 1;
                break;
            case 'l':
                bList = 1;
                break;
            case '?':
                printf("Invalid option\n");
        }
    }
usage(argv[0]);
return 0;
default:
    usage(argv[0]);
    return 0;
}
fd = device_open("pled");
if (fd < 0) {
    printf("device_open() fail\n");
goto main_close;
}
if ((nth >= 1 && nth <= 8) && bGet == 0 && bSet == 0) {
    printf("The -n option should be used with -g or -s\n");
    printf("EX: To set the first led on by, `mx_exsi_pled -n 1 -s 1`\n");
goto main_close;
}
else if (bList == 1) {
    num = device_list(fd);
    if (num < 0) {
        printf("device_list() fail\n");
goto main_close;
    }
    printf("pled number:%d\n", num);
}
else if (bGet == 1) {
    result = device_get(fd, nth, &value);
    printf("Get pled[%d] value %d\n", nth, value);
    if (result < 0) {
        printf("device_list() fail\n");
goto main_close;
    }
}
else if (bSet == 1) {
    if (value == 0) {
        printf("Turn off the LED, %d\n", nth);
        result = device_set(fd, nth, 0);
    } else {
        printf("Turn on the LED, %d\n", nth);
        result = device_set(fd, nth, 1);
    }
    if (result < 0) {
        printf("device_set() fail\n");
goto main_close;
    }
}
else if (bRead == 1) {
    result = device_read(fd, led_bitmap, sizeof(led_bitmap));
    if (result < 0) {
        printf("device_read() fail, result:%d\n", result);
goto main_close;
    }
    printf(" The led_bitmap is %s\n", led_bitmap);
}
else if ( bWrite == 1 ) {
    /* The size of led_bitmap should include the \0 or \n */
    result = device_write(fd, led_bitmap, strlen(led_bitmap)+1);
    if ( result < 0 ) {
        printf("device_write() fail\n");
        goto main_close;
    }
    printf("Write the LED bitmap:%s\n", led_bitmap);
}

main_close:
    device_close(fd);
    return 0 ;
}

Compiling the VMCI Utility Example in Linux

To compile the VMCI client utility in Linux, enter the commands shown in the following figure to uncompress the vmci_example.tar.gz file and compile the VMCI client.

```
root@Debian:~# tar xzvf vmci_example.tar.gz
root@Debian:~# cd vmci_example
root@Debian:~# make
```

The system creates the mx_esxi_pled, mx_esxi_set_uart8250_mode, and mx_esxi_usb_power files in the folder src/vmci_client/.

To control the programmable LED on VMWare ESXi 5.x, use the `mx_esxi_pled` command.

```
root@Debian:~# root@Debian7ESXi:/tmp# ./mx_esxi_pled
Get/set the programmable LED utility
Usage: ./mx_esxi_pled -l -n [-r|-w] [-g|-s] [-h]
    Show the mx_esxi_pled information if no argument apply.
    -h: Show this information.
    -l: List the number of LEDs.
    -n: Indicate the n-the LED.
    -r: Read the LED bitmap.
    -w: Write the bitmap to the LED.
    -g: Get the LED port value.
    -s: Set the value to the LED port.
```

The following table describes some `mx_esxi_pled` command examples.

<table>
<thead>
<tr>
<th>Command Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>./mx_esxi_pled -l</td>
<td>Displays the number of LEDs.</td>
</tr>
<tr>
<td>./mx_esxi_pled -w &quot;11110000&quot;</td>
<td>Turns on LEDs 1, 2, 3, and 4. Turns off LEDs 5, 6, 7, and 8.</td>
</tr>
<tr>
<td>./mx_esxi_pled -w &quot;10101010&quot;</td>
<td>Turns on LEDs 1, 3, 5, and 7. Turns off LEDs 2, 4, 6, and 8.</td>
</tr>
<tr>
<td>./mx_esxi_pled -n 1 -s 1</td>
<td>Turns on LED 1.</td>
</tr>
<tr>
<td>./mx_esxi_pled -n 8 -s 1</td>
<td>Turns on LED 8.</td>
</tr>
<tr>
<td>./mx_esxi_pled -n 8 -s 0</td>
<td>Turns off LED 8.</td>
</tr>
</tbody>
</table>
Compiling the VMCI Utility Example in Windows

This section shows you how to compile the VMCO client in Windows.

NOTE Before you start, make sure that you have Visual Studio 2010 installed in Windows.

1. Unzip the `vmci_example.tar.gz` file.
2. Double-click `mx_esxi_control.sln` to open the project.

Compile the following files to generate the executable files:

- `mx_esxi_pled`
- `mx_esxi_serial_interface_mode`
- `mx_esxi_usb_power`

3. Copy the following files to your virtual machine:
   - `devctrl.dll`
   - `mx_esxi_pled.exe`
   - `mx_esxi_serial_interface.exe`
   - `mx_esxi_usb_power.exe`
4. Install the Visual Studio 2010 redistribute package(x86), if it is not already installed.

You can run the `mx_esxi_pled` command to control the programmable LEDs, serial interface, and USB power on VMWare ESXi 5.x.

The following figures show the command examples.
C:\Program Files\vmware\vmware_eXtended interface

Get/set the serial port mode utility

Usage: mx_esxi_serial_interface.exe -l -a [-g|-s] [-h]

-l: Show the mx_esxi_serial_interface information if no argument apply.
-a: Show this information.
-g: List the number of serial ports.
-s: Indicate the n-th serial port.
1 for the first serial port (default).
2 for the second serial port.
-g: Select the n-th serial port interface type.
-s: Select the n-th serial port interface.
0 for RS-232 (default).
1 for RS-485-2W.
2 for RS-422/485-4W.

Example:
To set the first serial port as RS-232 mode.
root@Moxai:~# mx_esxi_serial_interface -n 1 -s 0
To set the first serial port as RS-485-2W mode.
root@Moxai:~# mx_esxi_serial_interface -n 1 -s 1
To set the first serial port as RS-422/485-4W mode.
root@Moxai:~# mx_esxi_serial_interface -n 1 -s 2
To set the first serial port mode.
root@Moxai:~# mx_esxi_serial_interface -n 1 -g
To set the second serial port as RS-232 mode.
root@Moxai:~# mx_esxi_serial_interface -n 2 -s 0
To set the second serial port as RS-485-2W mode.
root@Moxai:~# mx_esxi_serial_interface -n 2 -s 1
To set the second serial port as RS-422/485-4W mode.
root@Moxai:~# mx_esxi_serial_interface -n 2 -s 2
To get the second serial port mode.
root@Moxai:~# mx_esxi_serial_interface -n 2 -g
The DA-820-LX ready-to-run embedded computers are an embedded Linux platform. This chapter describes the recovery process.

The following topics are covered in this chapter:

- **Overview**
- **Setting Up the System Recovery Environment**
- **Recovering from the Factory Default Image**
  - Step 1: Preparing the USB Drive
  - Step 2: Setting the BIOS to Boot from the USB Drive
  - Step 3: Performing a System Recovery
  - Step 4: Resetting the BIOS
- **Creating a Custom System Image**
Overview

This section describes the recovery process in the event of a system failure.

You can recover the system using one of the following:

- Factory default image
- User-created image

Setting Up the System Recovery Environment

To set up the system recovery environment, you need the following:

- A DA-820 computer
- A USB drive with minimum 4 GB of storage space
- A copy of the recovery suite

The following is an overview of the steps to setting up the system recovery environment. The subsequent sections include detailed information. If you have already created an image on a USB drive, skip to Step 3.

1. From the DA-820 software CD/DVD, copy the bootable recovery environment (an ISO image) to the USB drive.
2. Choose a recovery image type to create.
3. If you choose to create a bare-bones image and use this image for system recovery, any applications or scripts you install later will be lost if a recovery is required.
4. Configure the BIOS to have the system boot from the USB port first during startup.
5. When the system next restarts, it will boot into the Clonezilla recovery environment on the USB drive.
6. Create a copy of a fully configured system on the USB drive. This is the alternative to the stock OS recovery method in Step 2.
7. Perform a system recovery. You can also perform this step to test the setup.
8. Reset the BIOS back to its original settings.
Recovering from the Factory Default Image

Step 1: Preparing the USB Drive

1. In the \recovery\DA-820-LX_Recovery\clonezilla directory on the DA-820 software CD/DVD, start the Clonezilla imaging program (within the current OS) by running tuxboot-windows-23.exe.
2. In the Tuxboot screen, select Pre-Downloaded and select ISO from the drop-down list.
3. Click the ellipsis (…) button to select the Clonezilla ISO image on the software CD/DVD.
4. In the \Recovery\DA-820-LX_Recovery\clonezilla directory on the software DVD, select the Clonezilla recovery environment ISO image.
5. In the Tuxboot screen, configure the following fields:
   - **Type:** Select **USB Drive** from the drop-down list.
   - **Drive:** Select a letter for the drive on which the USB drive is mounted.

6. Click **OK**. The system copies the Clonezilla recovery environment and the boot loader to the USB drive.

7. Click **Exit** to close the application.

**NOTE** You must delete the EFI directory on the USB drive.

8. From the `/media/cd0/recovery/os_image` folder on the software CD/DVD, copy the operating system image to the `/media/usb0/home/partimag` folder on the USB drive.

   The following figure shows a command example to copy the image file.

   ```bash
   moxa@Moxa:~# cp -a /media/cd0/recovery/os_image /media/usb0/home/partimag/
   ```

   You have created a USB recovery drive that enables you to perform a system recovery on the DA-820 using the factory default image.

   To create your own system recovery image, see the Creating a Custom System Image.
Step 2: Setting the BIOS to Boot from the USB Drive

This section shows you how to configure the BIOS to set the system to boot from a USB drive.

1. Restart the DA-820 and press F2 during the POST process until you hear a long beep.
2. In the BIOS configuration screen, click SCU to enter the BIOS setup menu.
3. Use the left or right arrow key to navigate to the Boot tab; then, press [Enter].
4. In the Boot screen, use the up or down arrow key to select Legacy and press [Enter].
5. Use the up or down arrow key to select **Boot Type Order** and press [Enter].

6. Use the up or down arrow key to select **USB** and use the plus or minus signs (+ -) to move the option to the first boot priority position.
Step 3: Performing a System Recovery

Connect the USB drive to any of the DA-820’s USB ports and then reboot the computer. The system will boot from the USB into the Clonezilla boot loader.

1. Connect the USB drive to a USB port on the DA-820 and restart the DA-820.
   The system boots from the USB drive.

2. Select **Clonezilla Live Restore Disk** to boot into the system restoration environment.

   ![Clonezilla Live Restore Disk](image)

   Wait for the boot process to complete.

   ![Clonezilla Boot Loader](image)

   **NOTE**

   To cancel the system recovery process and exit from the Clonezilla boot loader, press any letter or press Ctrl+C.

3. A message appears warning you that you are about to overwrite the hard drive and erase all data on the partition listed (for example, sda1). Enter **Y** (case insensitive).

   ![Clonezilla Warning Message](image)
The Clonezilla boot loader copies the system image from the USB drive to the primary system drive. This erases all existing data on the primary system drive.

Wait for the process to complete; depending on the system, this might take up to 10 minutes.

4. Complete the restoration process by selecting (0) **Poweroff** to shut down the computer.

**WARNING**

If the Power Switch remains inserted on the front panel of the computer and is at the ON position, the system performs a soft reboot. To prevent a soft reboot, either use the switch to turn off the computer immediately following the shutdown or may remove the power switch from the front panel and then use the console to shut down the computer by pressing 0.

5. After the computer is turned off, remove the USB drive and store it in a safe location.
Step 4: Resetting the BIOS

After you have performed a system recovery, reset the BIOS to enable the system to boot from the disk drive. This prevents the system from booting from an unauthorized USB drives and provides added system security during startup.

**ATTENTION**
The DA-820 does not differentiate a bootable and non-bootable USB drive. If you set the DA-820 to boot from a USB drive, the startup process will terminate if a non-bootable USB drive is connect to a USB port on the DA-820.

1. Restart the DA-820 and press **F2** during the POST process until you hear a long beep.
2. In the BIOS configuration screen, click **SCU** to enter the BIOS setup menu.
3. Use the left or right arrow key to navigate to the **Boot** tab; then, press [Enter].
4. In the **Boot** screen, use the up or down arrow key to select **Legacy** and press [Enter].
5. Use the up or down arrow keys to select **Boot Type Order** and press [Enter].

6. Use the up or down arrow key to select **Hard Disk Drive** and use the plus or minus signs (+ -) to move the option to the first boot priority position.

7. Press **F10** and press [Enter] to save the changes and exit from the BIOS configuration interface. The system automatically reboots from the hard disk drive.

### Creating a Custom System Image

This section shows you how to create a system recovery image from a system with customized applications and scripts.

The procedure below describes a configuration for restoring a complete system that has been customized with user applications and scripts. Here, you will save to the USB drive a copy of the entire system as it is currently configured to be used as a full system recovery image should the system crash. During this process, **all files on your USB that are mounted under F:|home\partimag| will be overwritten.**

1. gijfghj
2. Configure the BIOS to set the system to boot from a USB drive. For more information, see **Step 2: Setting the BIOS to Boot from the USB Drive.**
3. Restart the DA-820.
4. In the System Save & Recovery Utility screen, select clonezilla live save disk and press [Enter].

![CloneZilla Live Save Disk]

The DA-820 boots into the image creation environment. Wait for the boot process to complete.

5. A message appears warning you that the system is about to overwrite the `/home/partimag` folder on the USB drive and erase all data in the folder. Enter `Y` (case insensitive).

```
Setting the TERM as linux

clonezilla live save disk /home/partimag

Shutting down the Logical Volume Manager
No volume group found
No volume group found
Finished Shutting down the Logical Volume Manager
Selected device [sda] found
The selected device: sda

Activating the partition info in /proc... done!
Selected device [sda] found!
The selected device: sda
Searching for data partition(s)...
Excluding swap partition(s)...
Excluded partitions including extended or swap: sda1
Collecting info... done!
Searching for swap partition(s)...
Excluding swap partition(s)...
Excluded partitions including extended or swap: sda1
Collecting info... done!
The data partition to be saved: sda1
The swap partition to be saved: activated the partition info in /proc... done!
Selected device [sda1] found!
The selected devices: sda1
Setting /dev/sda1 info...

The following step is to save the hard disk/partition(s) on this machine as an image:

 mountpoints

VirtualBox

 sda (210MB WXX_HARDISK_ ata-00000000-0605A64B-00-000000-00000000-00000000)

 sda1 (210MB /dev/sda1)

Are you sure you want to continue? [y/N] y
```

**WARNING**

Regardless of system image type (factory default or custom image), the system creates a recovery image using the same filename. This means that you cannot save more than one system image on a USB drive.

The system copies all data on the hard drive to the USB drive. This might take up to half an hour, depending on the amount of data to copy. Wait until the process is complete.
ATTENTION

Do NOT remove the USB drive from the DA-820 during the image creation process.

6. Turn off the DA-820. Type 0 and press [Enter].

7. After the DA-820 is turned off, remove the USB drive and store it in a safe location.

8. Reset the BIOS to set the system to boot from the hard disk drive. You may test the image by performing a system recovery.
## Software Components

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acpi</td>
<td>1.6-1</td>
<td>amd64 displays information on ACPI devices</td>
</tr>
<tr>
<td>acpi-support-base</td>
<td>0.140-5+deb7u2</td>
<td>Contains scripts for handling base ACPI events (for example, the power button)</td>
</tr>
<tr>
<td>acpid</td>
<td>1:2.0.16-1+deb7u1</td>
<td>amd64 Advanced Configuration and Power Interface event daemon</td>
</tr>
<tr>
<td>adduser</td>
<td>3.113+nmu3</td>
<td>To add and remove users and groups</td>
</tr>
<tr>
<td>alsa-base</td>
<td>1.0.25+3~deb7u1</td>
<td>ALSA driver configuration files</td>
</tr>
<tr>
<td>alsa-utils</td>
<td>1.0.25-4</td>
<td>amd64 Utilities for configuring and using ALSA</td>
</tr>
<tr>
<td>apache2</td>
<td>2.2.22-13+deb7u3</td>
<td>amd64 Apache HTTP Server metapackage</td>
</tr>
<tr>
<td>apache2-mpm-prefork</td>
<td>2.2.22-13+deb7u3</td>
<td>amd64 Apache HTTP Server - traditional non-threaded model</td>
</tr>
<tr>
<td>apache2-utils</td>
<td>2.2.22-13+deb7u3</td>
<td>amd64 utility programs for webservers</td>
</tr>
<tr>
<td>apache2.2-bin</td>
<td>2.2.22-13+deb7u3</td>
<td>amd64 Apache HTTP Server common binary files</td>
</tr>
<tr>
<td>apache2.2-common</td>
<td>2.2.22-13+deb7u3</td>
<td>amd64 Apache HTTP Server common files</td>
</tr>
<tr>
<td>apt</td>
<td>0.9.7.9+deb7u2</td>
<td>amd64 commandline package manager</td>
</tr>
<tr>
<td>apt-utils</td>
<td>0.9.7.9+deb7u2</td>
<td>amd64 package management related utility programs</td>
</tr>
<tr>
<td>aptitude</td>
<td>0.6.8.2-1</td>
<td>amd64 terminal-based package manager</td>
</tr>
<tr>
<td>aptitude-common</td>
<td>0.6.8.2-1</td>
<td>All architecture indepedent files for the aptitude package manager</td>
</tr>
<tr>
<td>base-files</td>
<td>7.1wheezy6</td>
<td>amd64 Debian base system miscellaneous files</td>
</tr>
<tr>
<td>base-passwd</td>
<td>3.5.26</td>
<td>amd64 Debian base system master password and group files</td>
</tr>
<tr>
<td>bash</td>
<td>4.2+dfsg-0.1</td>
<td>amd64 GNU Bourne Again SHEll</td>
</tr>
<tr>
<td>bash-completion</td>
<td>1:2.0-1</td>
<td>All programmable completion for the bash shell</td>
</tr>
<tr>
<td>binutils</td>
<td>2.22-8</td>
<td>amd64 GNU assembler, linker and binary utilities</td>
</tr>
<tr>
<td>bridge-utils</td>
<td>1.5-6</td>
<td>amd64 utilities for configuring the Linux Ethernet bridge</td>
</tr>
<tr>
<td>bsdmainutils</td>
<td>9.0.3</td>
<td>amd64 collection of more utilities from FreeBSD</td>
</tr>
<tr>
<td>bsdutils</td>
<td>1:2.20.1-5.3</td>
<td>amd64 Basic utilities from 4.4BSD-Lite</td>
</tr>
<tr>
<td>build-essential</td>
<td>11.5</td>
<td>amd64 informational list of build-essential packages</td>
</tr>
<tr>
<td>busybox</td>
<td>1:1.20.0-7</td>
<td>amd64 tiny utilities for small and embedded systems</td>
</tr>
<tr>
<td>bzip2</td>
<td>1.0.6-4</td>
<td>amd64 high-quality block-sorting file compressor – utilities</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>console-setup</td>
<td>1.88</td>
<td>All console font and keymap setup program</td>
</tr>
<tr>
<td>console-setup-linux</td>
<td>1.88</td>
<td>All Linux specific part of console-setup</td>
</tr>
<tr>
<td>coreutils</td>
<td>8.13-3.5</td>
<td>amd64 GNU core utilities</td>
</tr>
<tr>
<td>cpio</td>
<td>2.11+dfsg-0.1</td>
<td>amd64 GNU cpio (a program to manage archives of files)</td>
</tr>
<tr>
<td>cpp</td>
<td>4:4.7.2-1</td>
<td>amd64 GNU C preprocessor (cpp)</td>
</tr>
<tr>
<td>cpp-4.6</td>
<td>4.6.3-14</td>
<td>amd64 GNU C preprocessor</td>
</tr>
<tr>
<td>cpp-4.7</td>
<td>4.7.2-5</td>
<td>amd64 GNU C preprocessor</td>
</tr>
<tr>
<td>cron</td>
<td>3.0p1-124</td>
<td>amd64 process scheduling daemon</td>
</tr>
<tr>
<td>da820-irigb</td>
<td>1.0</td>
<td>amd64 Moxa IRIG-B time sync daemon</td>
</tr>
<tr>
<td>dash</td>
<td>0.5.7-3</td>
<td>amd64 POSIX-compliant shell</td>
</tr>
<tr>
<td>debconf</td>
<td>1.5.49</td>
<td>All Debian configuration management system</td>
</tr>
<tr>
<td>debconf-i18n</td>
<td>1.5.49</td>
<td>All full internationalization support for debconf</td>
</tr>
<tr>
<td>debian-archive-keyring</td>
<td>2012.4</td>
<td>All GnuPG archive keys of the Debian archive</td>
</tr>
<tr>
<td>debianutils</td>
<td>4.3.2</td>
<td>amd64 Miscellaneous utilities specific to Debian</td>
</tr>
<tr>
<td>dialog</td>
<td>1.1-20120215-2</td>
<td>amd64 Displays user-friendly dialog boxes from shell scripts</td>
</tr>
<tr>
<td>diffutils</td>
<td>1:3.2-6</td>
<td>amd64 File comparison utilities</td>
</tr>
<tr>
<td>discover</td>
<td>2.1.2-5.2</td>
<td>amd64 hardware identification system</td>
</tr>
<tr>
<td>discover-data</td>
<td>2.2010.10.18</td>
<td>All data lists for Discover hardware detection system</td>
</tr>
<tr>
<td>dmidecode</td>
<td>2.11-9</td>
<td>amd64 SMBIOS/DMI table decoder</td>
</tr>
<tr>
<td>dmsetup</td>
<td>2:1.02.74-8</td>
<td>amd64 Linux Kernel Device Mapper userspace library</td>
</tr>
<tr>
<td>dpkg</td>
<td>1.16.15</td>
<td>amd64 Debian Kernel Device Mapper</td>
</tr>
<tr>
<td>dpkg-dev</td>
<td>1.16.15</td>
<td>All Debian package development tools</td>
</tr>
<tr>
<td>e2fslibs:amd64</td>
<td>1.42.5-1.1</td>
<td>amd64 ext2/ext3/ext4 file system libraries</td>
</tr>
<tr>
<td>e2fsprogs</td>
<td>1.42.5-1.1</td>
<td>amd64 ext2/ext3/ext4 file system utilities</td>
</tr>
<tr>
<td>ethtool</td>
<td>1:3.4.2-1</td>
<td>amd64 display or change Ethernet device settings</td>
</tr>
<tr>
<td>exim4-base</td>
<td>4.80-7</td>
<td>amd64 support files for all Exim MTA (v4) packages</td>
</tr>
<tr>
<td>exim4-config</td>
<td>4.80-7</td>
<td>All configuration for the Exim MTA (v4)</td>
</tr>
<tr>
<td>exim4-daemon-light</td>
<td>4.80-7</td>
<td>amd64 lightweight Exim MTA (v4) daemon</td>
</tr>
<tr>
<td>fakeroot</td>
<td>1.18.4-2</td>
<td>amd64 tool for simulating superuser privileges</td>
</tr>
<tr>
<td>findutils</td>
<td>4.4.2-4</td>
<td>amd64 utilities for finding files--find, xargs</td>
</tr>
<tr>
<td>firmware-linux-free</td>
<td>3.2</td>
<td>All Binary firmware for various drivers in the Linux kernel</td>
</tr>
<tr>
<td>g++</td>
<td>4:4.7.2-1</td>
<td>amd64 GNU C++ compiler</td>
</tr>
<tr>
<td>g++-4.7</td>
<td>4.7.2-5</td>
<td>amd64 GNU C++ compiler</td>
</tr>
<tr>
<td>gcc</td>
<td>4:4.7.2-1</td>
<td>amd64 GNU C compiler</td>
</tr>
<tr>
<td>gcc-4.6</td>
<td>4.6.3-14</td>
<td>amd64 GNU C compiler</td>
</tr>
<tr>
<td>gcc-4.6-base:amd64</td>
<td>4.6.3-14</td>
<td>amd64 GCC, the GNU Compiler Collection (base package)</td>
</tr>
<tr>
<td>gcc-4.7</td>
<td>4.7.2-5</td>
<td>amd64 GNU C compiler</td>
</tr>
<tr>
<td>gcc-4.7-base:amd64</td>
<td>4.7.2-5</td>
<td>amd64 GCC, the GNU Compiler Collection (base package)</td>
</tr>
<tr>
<td>gettext-base</td>
<td>0.18.1.1-9</td>
<td>amd64 GNU Internationalization utilities for the base system</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gnupg</td>
<td>1.4.12-7+deb7u4</td>
<td>amd64 GNU privacy guard - a free PGP replacement</td>
</tr>
<tr>
<td>gpgv</td>
<td>1.4.12-7+deb7u4</td>
<td>amd64 GNU privacy guard - signature verification tool</td>
</tr>
<tr>
<td>grep</td>
<td>2.12-2</td>
<td>amd64 GNU grep, egrep and fgrep</td>
</tr>
<tr>
<td>groff-base</td>
<td>1.21-9</td>
<td>amd64 GNU troff text-formatting system (base system components)</td>
</tr>
<tr>
<td>grub-common</td>
<td>1.99-27+deb7u2</td>
<td>amd64 GRand Unified Bootloader (common files)</td>
</tr>
<tr>
<td>grub-pc</td>
<td>1.99-27+deb7u2</td>
<td>amd64 GRand Unified Bootloader, version 2 (PC/BIOS version)</td>
</tr>
<tr>
<td>grub-pc-bin</td>
<td>1.99-27+deb7u2</td>
<td>amd64 GRand Unified Bootloader, version 2 (PC/BIOS binaries)</td>
</tr>
<tr>
<td>grub2-common</td>
<td>1.99-27+deb7u2</td>
<td>amd64 GRand Unified Bootloader (common files for version 2)</td>
</tr>
<tr>
<td>gzip</td>
<td>1.5-1.1</td>
<td>amd64 GNU compression utilities</td>
</tr>
<tr>
<td>hdparm</td>
<td>9.39-1+b1</td>
<td>amd64 tune hard disk parameters for high performance</td>
</tr>
<tr>
<td>heirloom-mailx</td>
<td>12.5-2</td>
<td>amd64 feature-rich BSD mail(1)</td>
</tr>
<tr>
<td>hostname</td>
<td>3.11</td>
<td>amd64 utility to set/show the host name or domain name</td>
</tr>
<tr>
<td>ifupdown</td>
<td>0.7.8</td>
<td>amd64 high level tools to configure network interfaces</td>
</tr>
<tr>
<td>initramfs-tools</td>
<td>0.115~bpo70+1</td>
<td>All generic modular initramfs generator</td>
</tr>
<tr>
<td>initscripts</td>
<td>2.88dsf-41+deb7u1</td>
<td>amd64 scripts for initializing and shutting down the system</td>
</tr>
<tr>
<td>insserv</td>
<td>1.14.0-5</td>
<td>amd64 boot sequence organizer using LSB init.d script dependency information</td>
</tr>
<tr>
<td>install-info</td>
<td>4.13a.dfsg.1-10</td>
<td>amd64 Manage installed documentation in info format</td>
</tr>
<tr>
<td>iproute</td>
<td>20120521-3+b3</td>
<td>amd64 networking and traffic control tools</td>
</tr>
<tr>
<td>iptables</td>
<td>1.4.14-3.1</td>
<td>amd64 administration tools for packet filtering and NAT</td>
</tr>
<tr>
<td>iputils-ping</td>
<td>3:20101006-1+b1</td>
<td>amd64 Tools to test the reachability of network hosts</td>
</tr>
<tr>
<td>isc-dhcp-client</td>
<td>4.2.2.dfsg.1.5+deb70u6</td>
<td>amd64 ISC DHCP client</td>
</tr>
<tr>
<td>isc-dhcp-common</td>
<td>4.2.2.dfsg.1.5+deb70u6</td>
<td>amd64 common files used by all the isc-dhcp* packages</td>
</tr>
<tr>
<td>kbd</td>
<td>1.15.3-9</td>
<td>amd64 Linux console font and keytable utilities</td>
</tr>
<tr>
<td>keyboard-configuration</td>
<td>1.88</td>
<td>All system-wide keyboard preferences</td>
</tr>
<tr>
<td>klibc-utils</td>
<td>2.0.1-3.1</td>
<td>amd64 small utilities built with klibc for early boot</td>
</tr>
<tr>
<td>kmod</td>
<td>9-3</td>
<td>amd64 tools for managing Linux kernel modules</td>
</tr>
<tr>
<td>laptop-detect</td>
<td>0.13.7</td>
<td>amd64 attempt to detect a laptop</td>
</tr>
<tr>
<td>libacl1:amd64</td>
<td>2.2.51-8</td>
<td>amd64 Access control list shared library</td>
</tr>
<tr>
<td>libalgorithm-diff-perl</td>
<td>1.19.02-2</td>
<td>All modules to find differences between files</td>
</tr>
<tr>
<td>libalgorithm-diff-xs-perl</td>
<td>0.04-2+b1</td>
<td>amd64 module to find differences between files (XS accelerated)</td>
</tr>
<tr>
<td>libalgorithm-merge-perl</td>
<td>0.08-2</td>
<td>All Perl modules for three-way merge of textual data</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>libapache2-mod-php5</td>
<td>5.4.4-14+deb7u12</td>
<td>amd64 server-side, HTML-embedded scripting language (Apache 2 module)</td>
</tr>
<tr>
<td>libapr1</td>
<td>1.4.6-3+deb7u1</td>
<td>amd64 Apache Portable Runtime Library</td>
</tr>
<tr>
<td>libaprutil1</td>
<td>1.4.1-3</td>
<td>amd64 Apache Portable Runtime Utility Library</td>
</tr>
<tr>
<td>libaprutil1-dbdsq1ite3</td>
<td>1.4.1-3</td>
<td>amd64 Apache Portable Runtime Utility Library - SQLite3 Driver</td>
</tr>
<tr>
<td>libaprutil1-ldap</td>
<td>1.4.1-3</td>
<td>amd64 Apache Portable Runtime Utility Library - LDAP Driver</td>
</tr>
<tr>
<td>libapt-inst1.5:amd64</td>
<td>0.9.7.9+deb7u2</td>
<td>amd64 deb package format runtime library</td>
</tr>
<tr>
<td>libapt-pkg4.12:amd64</td>
<td>0.9.7.9+deb7u2</td>
<td>amd64 package management runtime library</td>
</tr>
<tr>
<td>libasound2:amd64</td>
<td>1.0.25-4</td>
<td>amd64 shared library for ALSA applications</td>
</tr>
<tr>
<td>libasprintf0c2:amd64</td>
<td>0.18.1.1-9</td>
<td>amd64 GNU library to use fprintf and friends in C++</td>
</tr>
<tr>
<td>libattr1:amd64</td>
<td>1:2.4.46-8</td>
<td>amd64 Extended attribute shared library</td>
</tr>
<tr>
<td>libblkid1:amd64</td>
<td>2.20.1-5.3</td>
<td>amd64 block device id library</td>
</tr>
<tr>
<td>libboost-iostreams1.49.0</td>
<td>1.49.0-3.2</td>
<td>amd64 Boost.Iostreams Library</td>
</tr>
<tr>
<td>libbz2</td>
<td>2.13-38+deb7u4</td>
<td>amd64 high-quality block-sorting file compressor library - runtime</td>
</tr>
<tr>
<td>libc-bin</td>
<td>2.13-38+deb7u4</td>
<td>amd64 Embedded GNU C Library: Binaries</td>
</tr>
<tr>
<td>libc-dev-bin</td>
<td>2.13-38+deb7u4</td>
<td>amd64 Embedded GNU C Library: Development binaries</td>
</tr>
<tr>
<td>libc6:amd64</td>
<td>2.13-38+deb7u4</td>
<td>amd64 Embedded GNU C Library: Shared libraries</td>
</tr>
<tr>
<td>libc6-dev:amd64</td>
<td>2.13-38+deb7u4</td>
<td>amd64 Embedded GNU C Library: Development Libraries and Header Files</td>
</tr>
<tr>
<td>libcap2:amd64</td>
<td>1:2.22-1.2</td>
<td>amd64 support for getting/setting POSIX.1e capabilities</td>
</tr>
<tr>
<td>libclass-isa-perl</td>
<td>0.36-3</td>
<td>All reports the search path for a class's ISA tree</td>
</tr>
<tr>
<td>libcomerr2:amd64</td>
<td>1.42.5-1.1</td>
<td>amd64 common error description library</td>
</tr>
<tr>
<td>libcwidget3</td>
<td>0.5.16-3.4</td>
<td>amd64 high-level terminal interface library for C++ (runtime files)</td>
</tr>
<tr>
<td>libdb5.1:amd64</td>
<td>5.1.29-5</td>
<td>amd64 Berkeley v5.1 Database Libraries [runtime]</td>
</tr>
<tr>
<td>libdevmapper1.02.1:amd64</td>
<td>2:1.02.74-8</td>
<td>amd64 Linux Kernel Device Mapper userspace library</td>
</tr>
<tr>
<td>libdiscover2</td>
<td>2.1.2-5.2</td>
<td>amd64 hardware identification library</td>
</tr>
<tr>
<td>lddmraid1.0.0.rc16</td>
<td>1.0.0.rc16-4.2</td>
<td>amd64 Device-Mapper Software RAID support tool - shared library</td>
</tr>
<tr>
<td>lddpkg-perl</td>
<td>1.16.15</td>
<td>all Dpkg perl modules</td>
</tr>
<tr>
<td>libedit2:amd64</td>
<td>2.11-20080614-5</td>
<td>amd64 BSD editline and history libraries</td>
</tr>
<tr>
<td>libep1.4.12</td>
<td>1.0.9</td>
<td>amd64 High-level library for managing Debian package information</td>
</tr>
<tr>
<td>libexpat1:amd64</td>
<td>2.1.0-1+deb7u1</td>
<td>amd64 XML parsing C library - runtime library</td>
</tr>
<tr>
<td>libfile-copy-recursivc-perl</td>
<td>0.38-1</td>
<td>All Perl extensions for recursively copying files and directories</td>
</tr>
<tr>
<td>libfile-fcntllock-perl</td>
<td>0.14-2</td>
<td>amd64 Perl module for file locking with fcntl(2)</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>libfreetype6:amd64</td>
<td>2.4.9-1.1</td>
<td>amd64 FreeType 2 font engine, shared library files</td>
</tr>
<tr>
<td>libfuse2:amd64</td>
<td>2.9.0-2+deb7u1</td>
<td>amd64 Filesystem in Userspace (library)</td>
</tr>
<tr>
<td>libgcc1:amd64</td>
<td>1:4.7.2-5</td>
<td>amd64 GCC support library</td>
</tr>
<tr>
<td>libgpg-error0:amd64</td>
<td>1.10-3.1</td>
<td>amd64 library for common error values and messages in GnuPG components</td>
</tr>
<tr>
<td>libgmp10:amd64</td>
<td>4.7.2-5</td>
<td>amd64 GNU Transactional Memory Library</td>
</tr>
<tr>
<td>libgssapi-krb5-2:amd64</td>
<td>1.10.1+dfsg-5+deb7u2</td>
<td>amd64 MIT Kerberos runtime libraries - krb5 GSS-API Mechanism</td>
</tr>
<tr>
<td>libidn11:amd64</td>
<td>1.25-2</td>
<td>amd64 GNU Libidn library, implementation of IETF IDN specifications</td>
</tr>
<tr>
<td>libitm1:amd64</td>
<td>4.7.2-5</td>
<td>amd64 Linux Key Management Utilities (library)</td>
</tr>
<tr>
<td>libk5crypto3:amd64</td>
<td>1.10.1+dfsg-5+deb7u2</td>
<td>amd64 MIT Kerberos runtime libraries - Crypto Library</td>
</tr>
<tr>
<td>libkmod2:amd64</td>
<td>9-3</td>
<td>amd64 libkmod shared library</td>
</tr>
<tr>
<td>libkrb5-3:amd64</td>
<td>1.10.1+dfsg-5+deb7u2</td>
<td>amd64 MIT Kerberos runtime libraries</td>
</tr>
<tr>
<td>libkrb5support0:amd64</td>
<td>1.10.1+dfsg-5+deb7u2</td>
<td>amd64 MIT Kerberos runtime libraries - Support library</td>
</tr>
<tr>
<td>libldap-2.4-2:amd64</td>
<td>2.4.31-1+nmu2</td>
<td>amd64 OpenLDAP libraries</td>
</tr>
<tr>
<td>liblocale-gettext-perl</td>
<td>1.05-7+b1</td>
<td>amd64 module using libc functions for internationalization in Perl</td>
</tr>
<tr>
<td>libgpm2:amd64</td>
<td>1.20.4-6</td>
<td>amd64 General Purpose Mouse - shared library</td>
</tr>
<tr>
<td>libncurses5:amd64</td>
<td>5.11-2+deb7u3</td>
<td>amd64 File type determination library using &quot;magic&quot; numbers</td>
</tr>
<tr>
<td>libnewt0.52</td>
<td>0.52.14-11.1</td>
<td>amd64 Not Erik's Windowing Toolkit - text mode windowing with slang</td>
</tr>
<tr>
<td>libnfnetlink0</td>
<td>1.0.0-1.1</td>
<td>amd64 Netfilter netlink library</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>libonig2</td>
<td>5.9.1-1</td>
<td>amd64 Oniguruma regular expressions library</td>
</tr>
<tr>
<td>libopencryptoki0</td>
<td>2.3.1+dfsg-3</td>
<td>amd64 PKCS#11 implementation (library)</td>
</tr>
<tr>
<td>libp11-kit:amd64</td>
<td>0.12-3</td>
<td>amd64 Library for loading and coordinating access to PKCS#11 modules - runtime</td>
</tr>
<tr>
<td>libpam-modules:amd64</td>
<td>1.1.3-7.1</td>
<td>amd64 Pluggable Authentication Modules for PAM</td>
</tr>
<tr>
<td>libpam-modules-bin</td>
<td>1.1.3-7.1</td>
<td>amd64 Pluggable Authentication Modules for PAM - helper binaries</td>
</tr>
<tr>
<td>libpam-runtime</td>
<td>1.1.3-7.1</td>
<td>All runtime support for the PAM library</td>
</tr>
<tr>
<td>libpam0g:amd64</td>
<td>1.1.3-7.1</td>
<td>amd64 Pluggable Authentication Modules library</td>
</tr>
<tr>
<td>libpcap0.8:amd64</td>
<td>1.3.0-1</td>
<td>amd64 system interface for user-level packet capture</td>
</tr>
<tr>
<td>libpci3:amd64</td>
<td>1:3.1.9-6</td>
<td>amd64 Linux PCI Utilities (shared library)</td>
</tr>
<tr>
<td>libpcre3:amd64</td>
<td>1:8.30-5</td>
<td>amd64 Perl 5 Compatible Regular Expression Library - runtime files</td>
</tr>
<tr>
<td>libperl-dev</td>
<td>5.14.2-21+deb7u1</td>
<td>amd64 Perl library: development files</td>
</tr>
<tr>
<td>libperl5.14</td>
<td>5.14.2-21+deb7u1</td>
<td>amd64 shared Perl library</td>
</tr>
<tr>
<td>libpipeline1:amd64</td>
<td>1.2.1-1</td>
<td>amd64 pipeline manipulation library</td>
</tr>
<tr>
<td>libpks11-helper1:amd64</td>
<td>1.09-1</td>
<td>amd64 library that simplifies the interaction with PKCS#11</td>
</tr>
<tr>
<td>libpept0:amd64</td>
<td>1.16-7</td>
<td>amd64 lib for parsing cmdline parameters</td>
</tr>
<tr>
<td>libprocpss0:amd64</td>
<td>1:3.3.3-3</td>
<td>amd64 library for accessing process information from /proc</td>
</tr>
<tr>
<td>libqdbm14</td>
<td>1.8.78-2</td>
<td>amd64 QDBM Database Libraries without GDBM wrapper[runtime]</td>
</tr>
<tr>
<td>libquadmath0:amd64</td>
<td>4.7.2-5</td>
<td>amd64 GCC Quad-Precision Math Library</td>
</tr>
<tr>
<td>libreadline6:amd64</td>
<td>6.2+dfsg-0.1</td>
<td>amd64 GNU readline and history libraries, run-time libraries</td>
</tr>
<tr>
<td>libsamplerate0:amd64</td>
<td>0.1.8-5</td>
<td>amd64 Audio sample rate conversion library</td>
</tr>
<tr>
<td>libssl2-2:amd64</td>
<td>2.1.25.dfsg1-6+deb7u1</td>
<td>amd64 Cyrus SASL - authentication abstraction library</td>
</tr>
<tr>
<td>libssl2-modules:amd64</td>
<td>2.1.25.dfsg1-6+deb7u1</td>
<td>amd64 Cyrus SASL - pluggable authentication modules</td>
</tr>
<tr>
<td>libselinux1:amd64</td>
<td>2.1.9-5</td>
<td>amd64 SELinux runtime shared libraries</td>
</tr>
<tr>
<td>libsemanage-common</td>
<td>2.1.6-6</td>
<td>All common files for SELinux policy management libraries</td>
</tr>
<tr>
<td>libsemanage1:amd64</td>
<td>2.1.6-6</td>
<td>amd64 SELinux policy management library</td>
</tr>
<tr>
<td>libsensors4:amd64</td>
<td>1:3.3.2-2+deb7u1</td>
<td>amd64 library to read temperature/voltage/fan sensors</td>
</tr>
<tr>
<td>libsepol1:amd64</td>
<td>2.1.4-3</td>
<td>amd64 SELinux library for manipulating binary security policies</td>
</tr>
<tr>
<td>libsigc++-2.0-0c2a:amd64</td>
<td>2.2.10-0.2</td>
<td>amd64 type-safe Signal Framework for C++ - runtime</td>
</tr>
<tr>
<td>libslang2:amd64</td>
<td>2.2.4-15</td>
<td>amd64 S-Lang programming library - runtime version</td>
</tr>
<tr>
<td>libsqlite3-0:amd64</td>
<td>3.7.13-1+deb7u1</td>
<td>amd64 SQLite 3 shared library</td>
</tr>
<tr>
<td>libsqlite3-dev</td>
<td>3.7.13-1+deb7u1</td>
<td>amd64 SQLite 3 development files</td>
</tr>
<tr>
<td>libss2:amd64</td>
<td>1.42.5-1.1</td>
<td>amd64 command-line interface parsing library</td>
</tr>
<tr>
<td>libssl1.0.0:amd64</td>
<td>1.0.1e-2+deb7u12</td>
<td>amd64 SSL shared libraries</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>libstdc++6:amd64</td>
<td>4.7.2-5</td>
<td>amd64 GNU Standard C++ Library v3</td>
</tr>
<tr>
<td>libstdc++-4.7-dev</td>
<td>4.7.2-5</td>
<td>amd64 GNU Standard C++ Library v3 (development files)</td>
</tr>
<tr>
<td>libswitch-perl</td>
<td>2.16-2</td>
<td>All switch statements for Perl</td>
</tr>
<tr>
<td>libtasn1-3:amd64</td>
<td>2.13-2</td>
<td>amd64 Manage ASN.1 structures (runtime)</td>
</tr>
<tr>
<td>libtext-charwidth-perl</td>
<td>0.04-7+b1</td>
<td>amd64 get display widths of characters on the terminal</td>
</tr>
<tr>
<td>libtext-conv-perl</td>
<td>1.7-5</td>
<td>amd64 converts between character sets in Perl</td>
</tr>
<tr>
<td>libtext-wrap18n-perl</td>
<td>0.06-7</td>
<td>All internationalized substitute of Text::Wrap</td>
</tr>
<tr>
<td>libtime-deate-perl</td>
<td>1.2000-1</td>
<td>All collections of modules to manipulate date/time information</td>
</tr>
<tr>
<td>libtinfo-dev:amd64</td>
<td>5.9-10</td>
<td>amd64 developer's library for the low-level terminfo library</td>
</tr>
<tr>
<td>libtinfo5:amd64</td>
<td>5.9-10</td>
<td>amd64 shared low-level terminfo library for terminal handling</td>
</tr>
<tr>
<td>libtpm-unseal1</td>
<td>1.3.7-1</td>
<td>amd64 Management tools for the TPM hardware (library)</td>
</tr>
<tr>
<td>libtspi1</td>
<td>0.3.9-3+wheezy1</td>
<td>amd64 open-source TCG Software Stack (library)</td>
</tr>
<tr>
<td>libudev0:amd64</td>
<td>175-7.2</td>
<td>amd64 libudev shared library</td>
</tr>
<tr>
<td>libusb-0.1-4:amd64</td>
<td>2:0.1.12-20+nmu1</td>
<td>amd64 userspace USB programming library</td>
</tr>
<tr>
<td>libusb-1.0-0:amd64</td>
<td>2:1.0.11-1</td>
<td>amd64 userspace USB programming library</td>
</tr>
<tr>
<td>libustr-1.0-1:amd64</td>
<td>1.0.4-3</td>
<td>amd64 Micro string library: shared library</td>
</tr>
<tr>
<td>libuuid-perl</td>
<td>0.02-5</td>
<td>amd64 Perl extension for using UUID interfaces as defined in e2fsprogs</td>
</tr>
<tr>
<td>libuuid1:amd64</td>
<td>2.2.0-1-5.3</td>
<td>amd64 Universally Unique ID library</td>
</tr>
<tr>
<td>libwrap0:amd64</td>
<td>7.6-q-24</td>
<td>amd64 Wietse Venema's TCP wrappers library</td>
</tr>
<tr>
<td>libx11-6:amd64</td>
<td>2:1.5.0-1+deb7u1</td>
<td>amd64 X11 client-side library</td>
</tr>
<tr>
<td>libx11-data</td>
<td>2:1.5.0-1+deb7u1</td>
<td>all X11 client-side library</td>
</tr>
<tr>
<td>libx86-1:amd64</td>
<td>1.1+ds1-10</td>
<td>amd64 x86 real-mode library</td>
</tr>
<tr>
<td>libxapian22</td>
<td>1.2.12-2</td>
<td>amd64 Search engine library</td>
</tr>
<tr>
<td>libxau6:amd64</td>
<td>1:1.0.7-1</td>
<td>amd64 X11 authorisation library</td>
</tr>
<tr>
<td>libxcb1:amd64</td>
<td>1.8.1-2+deb7u1</td>
<td>amd64 X C Binding</td>
</tr>
<tr>
<td>libxdmcp6:amd64</td>
<td>1:1.1.1-1</td>
<td>amd64 X11 Display Manager Control Protocol library</td>
</tr>
<tr>
<td>libxext6:amd64</td>
<td>2:1.3.1-2+deb7u1</td>
<td>amd64 X11 miscellaneous extension library</td>
</tr>
<tr>
<td>libxml2:amd64</td>
<td>2.8.0+dfsg1-7+wheezy1</td>
<td>amd64 GNOME XML library</td>
</tr>
<tr>
<td>libxmlu1:amd64</td>
<td>2:1.1.1-1</td>
<td>amd64 X11 miscellaneous micro-utility library</td>
</tr>
<tr>
<td>linux-base</td>
<td>3.5</td>
<td>all Linux image base package</td>
</tr>
<tr>
<td>linux-compiler-gcc-4.6-x86</td>
<td>3.14.15-2~bpo70+1</td>
<td>amd64 Compiler for Linux on x86 (meta-package)</td>
</tr>
<tr>
<td>linux-headers-3.14-0.bpo.2-all- amd64</td>
<td>3.14.15-2~bpo70+1</td>
<td>amd64 All header files for Linux 3.14 (meta-package)</td>
</tr>
<tr>
<td>linux-headers-3.14-0.bpo.2-amd64</td>
<td>3.14.15-2~bpo70+1</td>
<td>amd64 Header files for Linux 3.14-0.bpo.2-amd64</td>
</tr>
<tr>
<td>linux-headers-3.14-0.bpo.2-common</td>
<td>3.14.15-2~bpo70+1</td>
<td>amd64 Common header files for Linux 3.14-0.bpo.2-common</td>
</tr>
<tr>
<td>linux-headers-3.14-0.bpo.2-common-rt</td>
<td>3.14.15-2~bpo70+1</td>
<td>amd64 Common header files for Linux 3.14-0.bpo.2-rt</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>linux-headers-3.14-0.bpo.2-rt-amd64</td>
<td>3.14.15-2-bpo70+1</td>
<td>amd64 Header files for Linux</td>
</tr>
<tr>
<td></td>
<td>3.14-0.bpo.2-rt-amd64</td>
<td></td>
</tr>
<tr>
<td>linux-image-3.14-0.bpo.2-amd64</td>
<td>3.14.15-2-bpo70+1</td>
<td>amd64 Linux 3.14 for 64-bit PCs</td>
</tr>
<tr>
<td>linux-kbuild-3.14</td>
<td>3.14-1~bpo70+1</td>
<td>amd64 Kbuild infrastructure for Linux 3.14</td>
</tr>
<tr>
<td>linux-libc-dev:amd64</td>
<td>3.2.60-1+deb7u3</td>
<td>amd64 Linux support headers for userspace development</td>
</tr>
<tr>
<td>locales</td>
<td>2.13-38+deb7u4</td>
<td>all Embedded GNU C Library: National Language (locale) data [support]</td>
</tr>
<tr>
<td>lockfile-progs</td>
<td>0.1.17</td>
<td>amd64 Programs for locking and unlocking files and mailboxes</td>
</tr>
<tr>
<td>login</td>
<td>1:4.1.5.1-1</td>
<td>amd64 system login tools</td>
</tr>
<tr>
<td>logrotate</td>
<td>3.8.1-4</td>
<td>amd64 Log rotation utility</td>
</tr>
<tr>
<td>lsb-base</td>
<td>4.1+Debian8+deb7u1</td>
<td>All Linux Standard Base 4.1 init script functionality</td>
</tr>
<tr>
<td>make</td>
<td>3.8i-8.2</td>
<td>amd64 An utility for Directing compilation.</td>
</tr>
<tr>
<td>man-db</td>
<td>2.6.2-1</td>
<td>amd64 on-line manual pager</td>
</tr>
<tr>
<td>manpages</td>
<td>3.44-1</td>
<td>All Manual pages about using a GNU/Linux system</td>
</tr>
<tr>
<td>manpages-dev</td>
<td>3.44-1</td>
<td>All Manual pages about using GNU/Linux for development</td>
</tr>
<tr>
<td>mawk</td>
<td>1.3.3-17</td>
<td>amd64 a pattern scanning and text processing language</td>
</tr>
<tr>
<td>mdadm</td>
<td>3.2.5-5</td>
<td>amd64 tool to administer Linux MD arrays (software RAID)</td>
</tr>
<tr>
<td>mime-support</td>
<td>3.52-1</td>
<td>All MIME files 'mime.types' &amp; 'mailcap', and support programs</td>
</tr>
<tr>
<td>module-init-tools</td>
<td>9-3</td>
<td>All transitional dummy packages (module-init-tools to kmod)</td>
</tr>
<tr>
<td>mount</td>
<td>2.20.1-5.3</td>
<td>amd64 Tools for mounting and manipulating filesystems</td>
</tr>
<tr>
<td>multiarch-support</td>
<td>2.13-38+deb7u4</td>
<td>amd64 Transitional package to ensure multiarch compatibility</td>
</tr>
<tr>
<td>ncurses-base</td>
<td>5.9-10</td>
<td>All basic terminal type definitions</td>
</tr>
<tr>
<td>ncurses-bin</td>
<td>5.9-10</td>
<td>amd64 terminal-related programs and man pages</td>
</tr>
<tr>
<td>ncurses-term</td>
<td>5.9-10</td>
<td>All additional terminal type definitions</td>
</tr>
<tr>
<td>net-tools</td>
<td>1.60-24.2</td>
<td>amd64 The NET-3 networking toolkit</td>
</tr>
<tr>
<td>netbase</td>
<td>5.0</td>
<td>All basic TCP/IP networking systems</td>
</tr>
<tr>
<td>netcat-traditional</td>
<td>1.10-40</td>
<td>amd64 TCP/IP swiss army knife</td>
</tr>
<tr>
<td>ntpdate</td>
<td>1:4.2.6.p5+dfsg-2</td>
<td>amd64 client for setting system time from NTP servers</td>
</tr>
<tr>
<td>openbsd-inetd</td>
<td>0.20091229-2</td>
<td>amd64 OpenBSD Internet Superserver</td>
</tr>
<tr>
<td>openssh-blacklist</td>
<td>0.4.1+nmu1</td>
<td>All lists of default blacklisted OpenSSH RSA and DSA keys</td>
</tr>
<tr>
<td>openssh-blacklist-extra</td>
<td>0.4.1+nmu1</td>
<td>All lists of non-default blacklisted OpenSSH RSA and DSA keys</td>
</tr>
<tr>
<td>openssh-client</td>
<td>1:6.0p1-4+deb7u2</td>
<td>amd64 secure shell (SSH) client, for secure access to remote machines</td>
</tr>
<tr>
<td>openssh-server</td>
<td>1:6.0p1-4+deb7u2</td>
<td>amd64 secure shell (SSH) server, for secure access from remote machines</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>openssl</td>
<td>1.0.1e-2+deb7u12</td>
<td>amd64 Secure Socket Layer (SSL) binary and related cryptographic tools</td>
</tr>
<tr>
<td>openvpn</td>
<td>2.2.1-8+deb7u2</td>
<td>amd64 virtual private network daemon</td>
</tr>
<tr>
<td>os-prober</td>
<td>1.58</td>
<td>amd64 utility to detect other OSes on a set of drives</td>
</tr>
<tr>
<td>passwd</td>
<td>1:4.1.5.1-1</td>
<td>amd64 change and administer password and group data</td>
</tr>
<tr>
<td>patch</td>
<td>2.6.1-3</td>
<td>amd64 Apply a diff file to an original</td>
</tr>
<tr>
<td>pciutils</td>
<td>1:3.1.9-6</td>
<td>amd64 Linux PCI Utilities</td>
</tr>
<tr>
<td>perl</td>
<td>5.14.2-21+deb7u1</td>
<td>amd64 Larry Wall's Practical Extraction and Report Language</td>
</tr>
<tr>
<td>perl-base</td>
<td>5.14.2-21+deb7u1</td>
<td>amd64 minimal Perl system</td>
</tr>
<tr>
<td>perl-modules</td>
<td>5.14.2-21+deb7u1</td>
<td>All core Perl modules</td>
</tr>
<tr>
<td>php5</td>
<td>5.4.4-14+deb7u12</td>
<td>All server-side, HTML-embedded scripting language (metapackage)</td>
</tr>
<tr>
<td>php5-cli</td>
<td>5.4.4-14+deb7u12</td>
<td>amd64 command-line interpreter for the php5 scripting language</td>
</tr>
<tr>
<td>php5-common</td>
<td>5.4.4-14+deb7u12</td>
<td>amd64 Common files for packages built from the php5 source</td>
</tr>
<tr>
<td>pm-utils</td>
<td>1.4.1-9</td>
<td>All utilities and scripts for power management</td>
</tr>
<tr>
<td>pmount</td>
<td>0.9.23-2</td>
<td>amd64 mount removable devices as normal user</td>
</tr>
<tr>
<td>powermgmt-base</td>
<td>1.31</td>
<td>amd64 Common utils and configs for power management</td>
</tr>
<tr>
<td>ppp</td>
<td>2.4.5-5.1+b1</td>
<td>amd64 Point-to-Point Protocol (PPP) - daemon</td>
</tr>
<tr>
<td>pppconfig</td>
<td>2.3.18+nmu4</td>
<td>All A text menu based utility for configuring ppp</td>
</tr>
<tr>
<td>pppoe</td>
<td>3.8-3</td>
<td>amd64 PPP over Ethernet driver</td>
</tr>
<tr>
<td>pppoeconf</td>
<td>1.20</td>
<td>All PPPoE/ADSL connections</td>
</tr>
<tr>
<td>procs</td>
<td>1:3.3.3-3</td>
<td>amd64 /proc file system utilities</td>
</tr>
<tr>
<td>proftpd-basic</td>
<td>1.3.4a-5+deb7u1</td>
<td>amd64 Versatile, virtual-hosting FTP daemon - binaries</td>
</tr>
<tr>
<td>proftpd-mod-vroot</td>
<td>0.9.2-2+b2</td>
<td>amd64 ProFTPD module mod_vroot</td>
</tr>
<tr>
<td>psmisc</td>
<td>22.19-1+deb7u1</td>
<td>amd64 utilities that use the proc file system</td>
</tr>
<tr>
<td>python</td>
<td>2.7.3-4+deb7u1</td>
<td>All interactive high-level object-oriented language (default version)</td>
</tr>
<tr>
<td>python-apt</td>
<td>0.8.8.2</td>
<td>amd64 Python interface to libapt-pkg</td>
</tr>
<tr>
<td>python-apt-common</td>
<td>0.8.8.2</td>
<td>All Python interfaces to libapt-pkg (locales)</td>
</tr>
<tr>
<td>python-chardet</td>
<td>2.0.1-2</td>
<td>All universal character encoding detectors</td>
</tr>
<tr>
<td>python-debian</td>
<td>0.1.21</td>
<td>All Python modules to work with Debian-related data formats</td>
</tr>
<tr>
<td>python-debianbts</td>
<td>1.11</td>
<td>All Python interfaces to Debian’s Bug Tracking System</td>
</tr>
<tr>
<td>python-fpconst</td>
<td>0.7.2-5</td>
<td>All utilities for handling IEEE 754 floating point special values</td>
</tr>
<tr>
<td>python-minimal</td>
<td>2.7.3-4+deb7u1</td>
<td>All minimal subsets of the Python language (default version)</td>
</tr>
<tr>
<td>python-reportbug</td>
<td>6.4.4+deb7u1</td>
<td>All Python modules for interacting with bug tracking systems</td>
</tr>
<tr>
<td>python-soappy</td>
<td>0.12.0-4</td>
<td>All SOAP support for Python</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>python-support</td>
<td>1.0.15</td>
<td>All automated rebuilding support for Python modules</td>
</tr>
<tr>
<td>python2.7</td>
<td>2.7.3-6+deb7u2</td>
<td>amd64 Interactive high-level object-oriented language (version 2.7)</td>
</tr>
<tr>
<td>python2.7-minimal</td>
<td>2.7.3-6+deb7u2</td>
<td>amd64 Minimal subset of the Python language (version 2.7)</td>
</tr>
<tr>
<td>readline-common</td>
<td>6.2+dfsg-0.1</td>
<td>All GNU readline and history libraries, common files</td>
</tr>
<tr>
<td>rsyslog</td>
<td>5.8.11-3</td>
<td>amd64 reliable system and kernel logging daemon</td>
</tr>
<tr>
<td>sed</td>
<td>4.2.1-10</td>
<td>amd64 The GNU sed stream editor</td>
</tr>
<tr>
<td>sensible-utils</td>
<td>0.0.7</td>
<td>All utilities for sensible alternative selection</td>
</tr>
<tr>
<td>sgml-base</td>
<td>1.26+nmu4</td>
<td>All SGML infrastructure and SGML catalog file support</td>
</tr>
<tr>
<td>sqlite3</td>
<td>3.7.13-1+deb7u1</td>
<td>amd64 Command line interface for SQLite 3</td>
</tr>
<tr>
<td>ssh</td>
<td>1:6.0p1-4+deb7u2</td>
<td>All secure shell client and server (metapackage)</td>
</tr>
<tr>
<td>ssl-cert</td>
<td>1.0.32</td>
<td>All simple debconf wrapper for OpenSSL</td>
</tr>
<tr>
<td>sudo</td>
<td>1.8.5p2-1+nmu1</td>
<td>amd64 Provide limited super user privileges to specific users</td>
</tr>
<tr>
<td>sysstat</td>
<td>10.0.5-1</td>
<td>amd64 system performance tools for Linux</td>
</tr>
<tr>
<td>sysv-rc</td>
<td>2.88dsf-41+deb7u1</td>
<td>All System-V-like runlevel change mechanism</td>
</tr>
<tr>
<td>sysvinit</td>
<td>2.88dsf-41+deb7u1</td>
<td>amd64 System-V-like init utilities</td>
</tr>
<tr>
<td>sysvinit-utils</td>
<td>2.88dsf-41+deb7u1</td>
<td>amd64 System-V-like utilities</td>
</tr>
<tr>
<td>tar</td>
<td>1.26+dfsg-0.1</td>
<td>amd64 GNU version of the tar archiving utility</td>
</tr>
<tr>
<td>tasksel</td>
<td>3.14.1</td>
<td>All Tool for selecting tasks for installation on Debian systems</td>
</tr>
<tr>
<td>tasksel-data</td>
<td>3.14.1</td>
<td>All Official tasks used for installation of Debian systems</td>
</tr>
<tr>
<td>tcpd</td>
<td>7.6.q-24</td>
<td>amd64 Wietse Venema’s TCP wrapper utilities</td>
</tr>
<tr>
<td>telnet</td>
<td>0.17-36</td>
<td>amd64 The telnet client</td>
</tr>
<tr>
<td>telnetd</td>
<td>0.17-36</td>
<td>amd64 The telnet server</td>
</tr>
<tr>
<td>tftp</td>
<td>0.17-18</td>
<td>amd64 Trivial file transfer protocol server</td>
</tr>
<tr>
<td>tpm-tools</td>
<td>1.3.7-1</td>
<td>amd64 Management tools for the TPM hardware (tools)</td>
</tr>
<tr>
<td>traceroute</td>
<td>1:2.0.18-3</td>
<td>amd64 Traces the route taken by packets over an IPv4/IPv6 network</td>
</tr>
<tr>
<td>trousers</td>
<td>0.3.9-3+wheezy1</td>
<td>amd64 open-source TCG Software Stack (daemon)</td>
</tr>
<tr>
<td>tzdata</td>
<td>2014e-0wheezy1</td>
<td>All time zone and daylight-saving time data</td>
</tr>
<tr>
<td>ucf</td>
<td>3.0025+nmu3</td>
<td>All Update Configuration File: preserve user changes to config files.</td>
</tr>
<tr>
<td>udev</td>
<td>175-7.2</td>
<td>amd64 /dev/ and hotplug management daemon</td>
</tr>
<tr>
<td>update-inetd</td>
<td>4.43</td>
<td>All inetd configuration file updater</td>
</tr>
<tr>
<td>usbmount</td>
<td>0.0.22</td>
<td>All automatically mount and unmount USB mass storage devices</td>
</tr>
<tr>
<td>usbutils</td>
<td>1:005-3</td>
<td>amd64 Linux USB utilities</td>
</tr>
<tr>
<td>util-linux</td>
<td>2.20.1-5.3</td>
<td>amd64 Miscellaneous system utilities</td>
</tr>
<tr>
<td>vbetool</td>
<td>1.1-2</td>
<td>amd64 run real-mode video BIOS code to alter hardware state</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vim</td>
<td>2:7.3.547-7</td>
<td>amd64 Vi IMproved - enhanced vi editor</td>
</tr>
<tr>
<td>vim-common</td>
<td>2:7.3.547-7</td>
<td>amd64 Vi IMproved - Common files</td>
</tr>
<tr>
<td>vim-runtime</td>
<td>2:7.3.547-7</td>
<td>all Vi IMproved - Runtime files</td>
</tr>
<tr>
<td>vim-tiny</td>
<td>2:7.3.547-7</td>
<td>amd64 Vi IMproved - enhanced vi editor - compact version</td>
</tr>
<tr>
<td>watchdog</td>
<td>5.12-1</td>
<td>amd64 system health checker and software/hardware watchdog handler</td>
</tr>
<tr>
<td>wget</td>
<td>1.13.4-3+deb7u1</td>
<td>amd64 retrieves files from the web</td>
</tr>
<tr>
<td>whiptail</td>
<td>0.52.14-11.1</td>
<td>amd64 Displays user-friendly dialog boxes from shell scripts</td>
</tr>
<tr>
<td>xauth</td>
<td>1:1.0.7-1</td>
<td>amd64 X authentication utility</td>
</tr>
<tr>
<td>xkb-data</td>
<td>2.5.1-3</td>
<td>All X Keyboard Extension (XKB) configuration data</td>
</tr>
<tr>
<td>xml-core</td>
<td>0.13+nmu2</td>
<td>All XML infrastructure and XML catalog file support</td>
</tr>
<tr>
<td>xz-utils</td>
<td>5.1.1alpha+20120614-2</td>
<td>amd64 XZ-format compression utilities</td>
</tr>
<tr>
<td>zlib1g:amd64</td>
<td>1:1.2.7.dfsg-13</td>
<td>amd64 compression library - runtime</td>
</tr>
</tbody>
</table>