UC-8100-LX Software Manual

Second Edition, March 2015

www.moxa.com/product

UC-8100-LX Software Manual

Copyright Notice

Trademarks

Disclaimer

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

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

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

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

Technical Support Contact Information

www.moxa.com/support

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

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

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

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

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

Table of Contents

1

Introduction

This is the programming and software operation manual for the Linux OS models of the UC-8100 series of embedded computers.

The UC-8100 series of computers come in various models that provide a range of operating systems, ARM architecture CPUs, and software enhancements. The following models are covered by this manual:

UC-8131-LX: RISC-based platform with 300 MHz CPU, 2 Ethernet, 1 Serial port, 1 GB SD, USB Port and Debian ARM 7

UC-8132-LX: RISC-based platform with 300 MHz CPU, Mini PCIe socket for cellular, 2 Ethernet, 2 serial ports, 1 GB SD, USB port and Debian ARM 7

UC-8162-LX: RISC-based platform with 600 MHz CPU, Mini PCIe socket for cellular, 2 Ethernet, 2 serial ports, 1 GB SD, USB port and Debian ARM 7

UC-8112-LX: RISC-based platform with 1 GHz CPU, Mini PCIe socket for cellular, 2 Ethernet, 2 serial ports, 1 GB SD, USB port, TPM, Micro SD Socket, and Debian ARM 7

2

Getting Started

This chapter describes how to use configure the UC-8100 basic settings.

The following topics are covered in this chapter:

- **Software Architecture**
- **Software Packages**
- **Connecting to the UC-8100-LX**
  - Connecting through Serial Console
  - SSH Console
- **User Account Management**
  - Switching to the Root Account
- **Creating and Deleting User Accounts**
- **Disabling the Default User Account**
- **Network Settings**
  - Configuring Ethernet Interfaces
  - Connecting to a Cellular Network
- **System Administration**
  - Querying the Firmware Version
  - Adjusting the Time
  - Setting the Time Zone
- **Determining Available Drive Space**
- **Enabling and Disabling Daemons**
- **Package Management**
- **Reboot/Shutdown the UC-8100-LX**

Software Architecture

The Linux operating system that is pre-installed in UC-8100-LX series follows standard Linux architecture, making it easy to accept programs that follow the POSIX standard. This computer uses the Debian ARM 7 distribution so that users can enjoy the full range of Debian software, and benefit from its strong community of developers and documentation. With Debian ARM, the UC-8100-LX supports both native and cross compilation, making programming on the computer more easy and straightforward.

The UC-8100-LX series image is partitioned into bootloader and Linux kernel, backup root file system and root file system. Refer to the following image partition table for reference.

Partition	System Content	Partition Format	Partition Size
1	Bootloader and Linux kernel	W95 FAT32	32 MB
2	Backup root file system	EXT4	128 MB
3	Root file system	EXT4	Rest of the capacity

The default file system format of UC-8100 Series is EXT4. It is a journaling file system for Linux, developed as the successor to EXT3. The journaling file system keeps track of the changes before committing them to the main file system. In the event of a system crash or power failure, journaling file systems are quicker to bring back online and less likely to become corrupted.

NOTE Click on the following links for more information on EXT4.
https://wiki.debian.org/Ext4
https://ext4.wiki.kernel.org/index.php/Ext4_Howto

Software Packages

Please refer to Apendix A to for default installed software packages. Most of the software packages come from Debian community whereas the unique features of UC-8100-LX series such as the diagnostic LED, wireless connection will be supported by Moxa. Please refer to Package Management section for how to manage the software package on UC-8100-LX series.

Connecting to the UC-8100-LX

You will need another computer to connect to the UC-8100-LX and log on to the command line interface. There are two ways to connect: through serial console cable or through Ethernet cable. You may refer to the Hardware Manual to see how to connect them physically.

The default login user and password are:

Login: moxa
Password: moxa

They are the same for all serial console and ssh remote login. Root account login is disabled until you manually create password for the account. User moxa is in the sudo group so you may operate system level commands with this user by sudo command. See more detail in **Sudo Mechanism** section.

ATTENTION

For security reason, we recommend you to disable the default user account and create your own user accounts.

Connecting through Serial Console

This method is particularly useful when using the computer for the first time. The signal is transmitted over a direct serial connection so you do not need to know either of its two IP addresses in order to connect to

UC-8100-LX. To connect through serial console, you need terminal software installed on your PC. Set the following serial connection parameters on your PC terminal software.

Serial Console Port Settings	
Baud rate	115200 bps
Parity	None
Data bits	8
Stop bits:	1
Flow Control	None
Terminal	VT100

Below we will show how to use the terminal software to connect to the UC-8100-LX Series in Linux environment and in Windows environment.

Linux Users

NOTE These steps are done on the Linux PC from which you want to connect to UC-8100-LX, NOT to be done on the UC-8100-LX

For Linux users, you may follow these steps to connect to UC-8100-LX Series from your personal computer.

1. Install **minicom** from the package repository of your operating system.

   For Centos and Fedora:

   ```
   user@PC1:~# yum -y install minicom
   ```

   For Ubuntu and Debian:

   ```
   user@PC2:~# apt-get install minicom
   ```

2. Use **minicom –s** command to enter configuration menu and setup the serial port settings.

   ```
   user@PC1:~# minicom -s
   ```

3. Select **Serial port setup**.

   ```
   +-----[configuration]------+
   | Filenames and paths      |
   | File transfer protocols  |
   | Serial port setup        |
   | Modem and dialing        |
   | Screen and keyboard      |
   | Save setup as dfl        |
   | Save setup as..          |
   | Exit                     |
   | Exit from Minicom        |
   +--------------------------+
   ```

4. Select **A** to change the serial device. Note that you need to know which device node is connected to the UC-8100-LX.

   ```
   +-----------------------------------------------------------------------+
   | A -    Serial Device      : /dev/tty8                                 |
   | B - Lockfile Location     : /var/lock                                 |
   | C -   Callin Program      :                                           |
   | D -  Callout Program      :                                           |
   | E -    Bps/Par/Bits       : 115200 8N1                                |
   | F - Hardware Flow Control : Yes                                       |
   | G - Software Flow Control : No                                        |
   |                                                                       |
   |    Change which setting?                                              |
   +-----------------------------------------------------------------------+
           | Screen and keyboard      |
           | Save setup as dfl        |
           | Save setup as..          |
           | Exit                     |
           | Exit from Minicom        |
           +--------------------------+
   ```

5. Select **E** to configure the port settings according to the **Serial Console Port Settings** table provided.
6. Select **Save setup as dfl** (from the main configuration menu) to save as default value.

7. Select **Exit from minicom** (from the configuration menu) to leave the configuration menu.
8. Execute **minicom** after completing the above configurations.

```
user@PC1:~# minicom
```

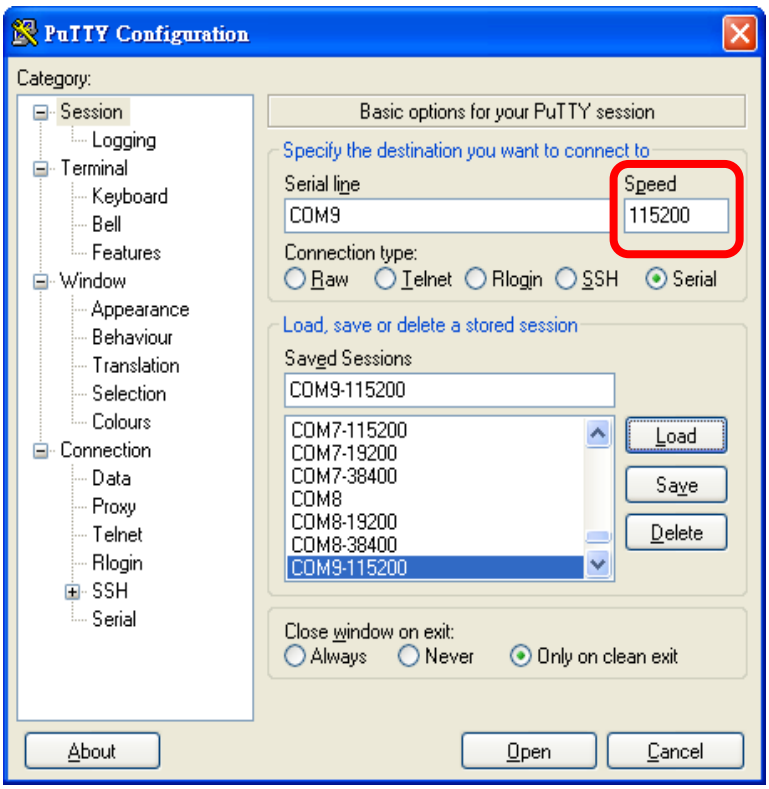
```
Welcome to minicom 2.6.1

OPTIONS: I18n
Compiled on Feb 11 2012, 18:56:01.
Port /dev/tty8

Press CTRL-A Z for help on special keys
```

Windows Users

NOTE These steps are done on the Windows PC from which you want to connect to UC-8100-LX, NOT to be done on the UC-8100-LX

For Windows users, follow these steps.

1. Download PuTTY http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html to set up serial connection withr the UC-8100-LX in Windows environment. The following figure shows a simple example of the configuration that is required.
2. Once the connection is established, the following window will open.



SSH Console

The UC-8100-LX supports SSH connection through Ethernet. Use the following default IP address of UC-8100-LX to connect.

Port	Default IP
LAN 1	192.168.3.127
LAN 2	192.168.4.127

Linux Users

NOTE These steps are done on the Linux PC from which you want to connect to UC-8100-LX, NOT to be done on the UC-8100-LX.

From a Linux computer, use **ssh** command to access from UC-8100-LX LAN1.

```
user@PC1:~ ssh moxa@192.168.3.127
```

Type **yes** to complete the connection.

```
The authenticity of host '192.168.3.127 (192.168.4.127)' can't be established.
RSA key fingerprint is 8b:ee:ff:84:41:25:fc:cd:2a:f2:92:8f:cb:1f:6b:2f.
Are you sure you want to continue connection (yes/no)? yes_
```

ATTENTION

Rekey SSHD regularly

In order to secure your system, we suggest you do SSH-rekey regularly. Refer to the following steps.

```
cd /etc/ssh
sudo rm -rf
ssh_host_dsa_key       ssh_host_ecdsa_key     ssh_host_rsa_key
ssh_host_dsa_key.pub   ssh_host_ecdsa_key.pub  ssh_host_rsa_key.pub
sudo ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
sudo ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key
sudo ssh-keygen -t ecdsa -f. /etc/ssh/ssh_host_ecdsa_key
```

When prompted for a passphrase, leave the passphrase empty and press enter.

```
Restart SSH
moxa@Moxa:~$ sudo /etc/init.d/ssh restart
```

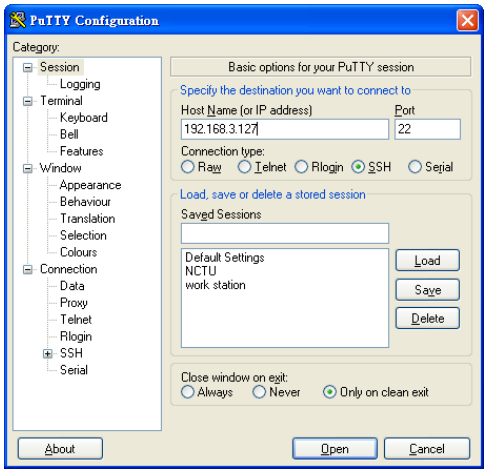
For more information about SSH, refer to the following link.
https://wiki.debian.org/SSH

Windows Users

NOTE These steps are done on the Windows PC from which you want to connect to UC-8100-LX, NOT to be done on the UC-8100-LX

For Windows users, follow these steps.

1. Click on the link http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html to download PuTTY (free software) to set up an SSH console for the UC-8100-LX in a Windows environment. The following figure shows a simple example of the configuration that is required.



User Account Management

Switching to the Root Account

You can switch to root using **sudo -i** (or sudo su). For security reason, it is not suggested to operate all commands by root account.

NOTE Click the following links for more information on sudo.
https://wiki.debian.org/sudo

ATTENTION

You may get **permission denied** while using pipe or redirect behavior with non-root account
You must use 'sudo su -c' to run the command instead of using >, <, >>, << etc.. . Note the quotes around the full command:

Creating and Deleting User Accounts

You may use the command **useradd** and **userdel** to create and delete user accounts. Be sure to reference the man page of these commands to set relevant previledge of the account. Following example shows to create a test1 user in sudo group whose default login shell is bash and has home directory at /home/test1

```
moxa@Moxa:~# sudo useradd -m -G sudo -s /bin/bash test1
```

To edit the password of test1 use **passwd** and enter twice the password to confirm.

```
moxa@Moxa:~# sudo passwd test1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

To delete user test1, you can use the command **userdel**

```
moxa@Moxa:# sudo userdel test1
```

Disabling the Default User Account

ATTENTION

You should first create a user account before you disable the default account.

You can use **passwd** command to lock the default user account so user moxa cannot login.

```
root@Moxa:# passwd -l moxa
```

To unlock the user moxa:

```
root@Moxa:# passwd -u moxa
```

Network Settings

Configuring Ethernet Interfaces

After first login, you may configure the network setting on UC-8100-LX to better fit your application. Note that it is more convenient to manipulate the network interface settings from serial console than from SSH login to avoid reconnections.

Modifying Network Settings via the Serial Console

In this section, we use the serial console to configure network settings of the UC-8100-LX computer.

Follow the instructions given in a previous section to access the Console Utility of the target computer via the serial Console port, and then type **`Moxa:~# cd /etc/network`** to change directories.

```
moxa@Moxa:~$ cd /etc/network/
moxa@Moxa:/etc/network/~$
```

Type **`Moxa:~# sudo vi interfaces`** to edit the network configuration file with vi editor. You can configure Ethernet ports of the UC-8100-LX for **static** or **dynamic** (DHCP) IP addresses.

Static IP address

As shown below, 2 network addresses need to be modified: **address**, **network**, **netmask**, and **broadcast**. The default IP address of the UC-8100-LX is 192.168.3.127 for LAN 1.

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto eth0 eth1 lo
iface lo inet loopback

# embedded ethernet LAN1
#iface eth0 inet dhcp
iface eth0 inet static
       address 192.168.3.127
       network 192.168.3.0
       netmask 255.255.255.0
       broadcast 192.168.3.255

# embedded ethernet LAN2
iface eth1 inet static
       address 192.168.4.127
       network 192.168.4.0
       netmask 255.255.255.0
       broadcast 192.168.4.255~
```

Dynamic IP addresses:

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

Default Setting for LAN1	Dynamic Setting using DHCP
iface eth0 inet static address 192.168.3.127 network: 192.168.3.0 netmask 255.255.255.0 broadcast 192.168.3.255	iface eth0 inet dhcp

```
# embedded ethernet LAN1
iface eth0 inet dhcp
```

Connecting to a Cellular Network

You can install cellular modules on UC-8100-LX series. Refer to Moxa Official Website or the product data for compatible cellular modules.

The cellular connection utitlity is cell_mgmt. Once you have cellular module installed and SIM card inserted, you may use the command to connect.

First, edit the APN name in /etc/qmi-network.conf:

```
moxa@Moxa:~$ echo "APN=internet" | sudo tee /etc/qmi-network.conf
```

And use the following command:

```
moxa@Moxa:~$ sudo cell_mgmt start
```

Please refer to Cellular Module section for more advanced settings.

System Administration

Querying the Firmware Version

To check the firmware version of UC-8100-LX series, type:

```
moxa@Moxa:~$ kversion
UC-8112-LX version 1.0.0
```

Adding the **–a** option will give you the full build version:

```
moxa@Moxa:~$ kversion -a
UC-8112-LX version 1.0.0 Build 14050416
```

Adjusting the Time

The UC-8100-LX has two time settings. One is the system time, and the other is the RTC (Real Time Clock) time kept by the UC-8100-LX hardware. Use the **`#date`** command to query the current system time or set a new system time. Use **`#hwclock`** to query the current RTC time or set a new RTC time.

Use the command **date MMDDhhmmYYYY** to set the system time **:**

MM = Month
DD = Date
hhmm = hour and minute

```
YYYY = Yearmoxa@Moxa:~$ sudo date 071123192014
Mon Jul 11 23:19:00 UTC 2014
```

Use the following command to set the RTC time to system time:

```
moxa@Moxa:~$ sudo hwclock –w
moxa@Moxa:~$ sudo hwclock
Fri 11 Jul 2014 11:19:38 PM UTC  -1.006862 seconds
```

NOTE Click the following links for more information on Datetime.
https://www.debian.org/doc/manuals/system-administrator/ch-sysadmin-time.html
https://wiki.debian.org/DateTime

Setting the Time Zone

There are two ways for supporting the timezone configuration on Moxa embedded computer. One is using the TZ variable. The other is using /etc/localtime.

TZ variable

TZ environment variable format looks like this:

```
TZ=standardHH[:MM[:SS]][daylight[HH[:MM[:SS]]][,start date[/starttime], enddate[/endtime]]]
```

Here are some possible settings for the North American Eastern time zone:

```
TZ=EST5EDT
TZ=EST0EDT
TZ=EST0
```

In the first case, the reference time is GMT and thus stored time values are correct world wide. A simple change of the TZ variable prints local time correctly, anywhere. In the second case, the reference time is Eastern Standard Time and the only conversion performed is for Daylight Saving Time. Therefore, there is no need to adjust the hardware clock for Daylight Saving Time twice per year. In the third case, the reference time is always the time reported. This is suggested if the hardware clock on your machine automatically adjusts for Daylight Saving Time or you insist on manually resetting the hardware time twice a year.

Adding in the file /etc/rc.d/rc.local, the timezone setting will be activated after the computer restarts.

```
moxa@Moxa:~$ export TZ= EST0
```

Following are other possible values for the TZ environment variable:

Hours From Greenwich Mean Time (GMT)	Value	Description
0	GMT	Greenwich Mean Time
+1	ECT	European Central Time
+2	EET	European Eastern Time
+2	ART	
+3	EAT	Saudi Arabia
+3.5	MET	Iran
+4	NET	
+5	PLT	West Asia
+5.5	IST	India
+6	BST	Central Asia
+7	VST	Bangkok
+8	CTT	China
+9	JST	Japan
+9.5	ACT	Central Australia

Hours From Greenwich Mean Time (GMT)	Value	Description
+10	AET	Eastern Australia
+11	SST	Central Pacific
+12	NST	New Zealand
-11	MIT	Samoa
-10	HST	Hawaii
-9	AST	Alaska
-8	PST	Pacific Standard Time
-7	PNT	Arizona
-7	MST	Mountain Standard Time
-6	CST	Central Standard Time
-5	EST	Eastern Standard Time
-5	IET	Indiana East
-4	PRT	Atlantic Standard Time
-3.5	CNT	Newfoundland
-3	AGT	Eastern South America
-3	BET	Eastern South America
-1	CAT	Azores

/etc/localtime

The local timezone is stored in /etc/localtime and is used by GNU Library for C (glibc) if the TZ environment variable is not set. This file is either a copy of /usr/share/zoneinfo/ file or a symbolic link to it. The UC-8100-LX does not provide /usr/share/zoneinfo/ files, so you have to copy a time zone information file to the UC-8100-LX and write over the original local time file.

Determining Available Drive Space

To know the available drive space remaining, use **df** command with the **–h** tag. The system will return the amount of drive space broken down by file system. Check the following example.

```
moxa@Moxa:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          803M  238M  524M  32% /
/dev/root       803M  238M  524M  32% /
tmpfs            25M  188K   25M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs            10M     0   10M   0% /dev
tmpfs            50M     0   50M   0% /run/shm
```

Enabling and Disabling Daemons

Only the following daemons are enabled in UC-8100-LX by default

sftpd SFTP Server / Client daemon

sshd Secure Shell Server daemon

You may manage what services to run in the background by the command **insserv**. Below example shows how to add the apache daemon in current runlevel.

```
moxa@Moxa:~$ sudo insserv –d apache2
```

Apache will not activate in current boot session but will be running in the background from next boot session.

To disable the apache daemon, use the following command.

```
moxa@Moxa:~$ sudo insserv -r apache2
```

You can also write your own daemon and start it in the system init stage.

```
### BEGIN INIT INFO
# Provides:         scriptname
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start daemon at boot time
# Description:       Enable service provided by daemon.
### END INIT INFO

YOUR SCRIPT
```

Linux daemons can be started or stopped in current boot session by using of the scripts in /etc/init.d. To start the apache daemon, use:

```
moxa@Moxa:~$ sudo /etc/init.d/apache2 start
```

To stop the apache daemon, use:

```
moxa@Moxa:~$ sudo /etc/init.d/apache2 stop
```

In comparison to insserv, scripts in /etc/init.d/ will only start or stop the services in the current boot session. Once you reboot the UC-8100-LX, it will go back to the default settings which managed by insserv.

Package Management

Most of the software packages will be maintained by Debian community through Debian official apt repository while the UC-8100-LX-only features will be maintained by Moxa. By adding Moxa repository to **/etc/apt/sources.list** will allow you to keep your system up to date with the newest UC-8100-LX packages.

```
moxa@Moxa:~$ cat /etc/apt/sources.list
deb http://debian.moxa.com/debian wheezy main

deb http://ftp.us.debian.org/debian/ wheezy main contrib non-free
deb-src http://ftp.us.debian.org/debian/ wheezy main contrib non-free

deb http://ftp.us.debian.org/debian/ wheezy-updates main contrib non-free
deb-src http://ftp.us.debian.org/debian/ wheezy-updates main contrib non-free

deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-free

deb http://ftp.debian.org/debian wheezy-backports main contrib non-free
deb-src http://ftp.debian.org/debian wheezy-backports main contrib non-free
```

Following packages will be maintained through Moxa official repository.

Package Name	Version	Architecture	Description
libssl1.0.0:armhf	1.0.1e-2+deb7u13+uc8100	armhf	SSL shared libraries
openssl	1.0.1e-2+deb7u13+uc8100	armhf	Secure Socket Layer (SSL) binary and related cryptographic tools
uc8100-cellular-utils	1.2.0	armhf	Cellular driver and related utility on uc8100 series.
uc8100-diag	1.0.0	armhf	Self-diagnostic utility on uc8100 series.
uc8100-push-btn	1.0.0	armhf	Push button utility on uc8100 series.
uc8100-setdef	1.0.0	all	Set-to-default utility on uc8100 series.

Package Name	Version	Architecture	Description
uc8100-setinterface	1.0.0	all	Adjust UART mode utility on uc8100 series.
uc8100-snmpd	5.4.3~dfsg-2.7	armhf	SNMP (Simple Network Management Protocol) agents
uc8100-system	1.3.0-2	armhf	System files in uc8100
uc8100-wifi-utils	1.2.0	armhf	WiFi utils on uc8100 series.

Reboot/Shutdown the UC-8100-LX

To shut down a running Debian GNU/Linux system, you must not reboot with the reset switch on the front or back of your computer, or just turn off the computer. Debian GNU/Linux should be shut down in a controlled manner, otherwise files might get lost and/or disk damage might occur. If you run a desktop environment, there is usually an option to "log out" available from the application menu that allows you to shutdown (or reboot) the system.

To reboot the UC-8100-LX, use the following command.

```
moxa@Moxa:~$ sudo reboot –i –f –d
```

To shut down the UC-8100-LX, use the following command.

```
moxa@Moxa:~$ sudo shutdown -h "now"
```

3

Advanced Configurations on Peripherals

This chapter includes more information on the periperials on UC-8100-LX, such as the serial interface, the storage, the diagnostic LEDs, and the cellualar module.

The following topics are covered in this chapter:

Serial Ports

UC-8131-LX has 1 serial ports on /dev/ttyM0, while all other UC-8100-LX models has 2 serial ports at /dec/ttyM0 and /dev/ttyM1. They support RS-232, RS-422, and RS-485 2-wire operation modes with flexible baudrate settings.

The default operation mode is set to RS-422, you can use **setinterface** command to change the operation mode.

Usage: setinterface device-node [interface-no]
Device-node: /dev/ttyMn; n = 0,1,2,...
Interface-no: As in the following table

Interface-no	Operation Mode
None	Display current setting
0	RS-232
1	RS-485 2-wire
2	RS-422 / RS-485 4-wire

For example, to set /dev/ttyM0 to RS-485 2-wire mode, you may use the following command:

```
moxa@Moxa:~# sudo setinterface /dev/ttyM0 1
Now setting is RS485-2W mode
moxa@Moxa:~# sudo setinterface /dev/ttyM0
UART Port#0 is in RS485-2W Mode
```

stty

stty command is used to manipulate the terminal settings. You can view and modify the serial terminal settings with this command. See below for more detail.

Display All Settings

Check the following table for displaying all settings.

```
moxa@Moxa:~$ sudo stty -a -F /dev/ttyM0
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

Configure Serial Settings

The following example will change the baudrate to 115200.

```
moxa@Moxa:~$ sudo stty 115200 -F /dev/ttyM0
```

The baud rate is already switched to 115200.

```
moxa@Moxa:~$ sudo stty -a -F /dev/ttyM0
speed 115200 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

NOTE Click the following link for more information about stty.
http://www.gnu.org/software/coreutils/manual/coreutils.html#stty-invocation

USB Port

The UC-8100-LX series provides a USB slot for storage expansion.

Disable the USB Port

USB ports on the UC-8100-LX can be disabled. This must be done via the bootloader. before booting up. To disable a USB port, follow these steps:

1. After powering on the UC-8100 computer, press <DEL> to enter BIOS configuration setting

```
--------------------------------------------------------------------------------
  Boot Loader Version 1.0.0S12        CPU TYPE: 1GHz
  Build date: May  7 2014 - 15:55:07   Serial Number: MOXATESTSN01
  LAN1 MAC: 00:90:E8:00:00:01         LAN2 MAC: 00:90:E8:00:00:02
--------------------------------------------------------------------------------
(0) TPM Setting                   (1) SD Card Write Protect
(2) Extend USB Port Control       (3) Go To OS
--------------------------------------------------------------------------------
Command>>2
```

2. Enter 2 to enter Extend USB Port Control.

```
Current Extend USB Port is ON.

Change to ,0 - ON, 1 - OFF (0-1,enter for abort):
```

3. Enter 1 to disable the USB port.

```
Change to ,0 - ON, 1 - OFF (0-1,enter for abort): 1
Saving Environment to EEPROM...
```

3. You need to reboot the UC-8100-LX computer to make the change effective. Also, during boot up, you will see below message printed on the console indicating you have successfully disabled the USB port.

```
[60.268951] hub 2-0:1.0: unable to enumerate USB device on port 1
```

ATTENTION

No USB devices may be mounted when the port is disabled

Regardless if the usb device is block storage or a dongle, it can not be mounted.

USB Automount

The UC-8100-LX supports hot plug functions for connecting USB mass storage devices. The UC-8100-LX has an udev automount utility that eases the mount procedure. However, the udev automount utility default only supports mounting one partition automatically.

```
//dev/root on / type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=25432k,mode=755)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev type tmpfs (rw,relatime,size=10240k,mode=755)
/dev/sda1 on /media/usb1 type vfat
(rw,nodev,noexec,noatime,nodiratime,sync,fmask=0022,dmask=0022,codepage=cp437,iocharset=iso
8859-1,shortname=mixed,errors=remount-ro)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=50840k)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620)
/dev/sdb1 on /media/usb0 type vfat
(rw,nodev,noexec,noatime,nodiratime,sync,fmask=0022,dmask=0022,codepage=cp437,iocharset=iso
8859-1,shortname=mixed,errors=remount-ro)
```

ATTENTION

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

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

SD and MicroSD Slot

The SD slot supports SD, SDHC, and SDXC format and is used as the main storage for UC-8100-LX series. UC-8100-LX comes with a pre-installed 1GB SD card but allows user to use larger capacity SD card up to 64GB cards.

In UC-8112-LX, an internal Micro SD slot is provided. Depending on which slot you insert a storage media with a valid boot code, you may have the option to make the SD or MicroSD slot your main storage, and the other as expansion. In either case, the expansion storage will be automounted to `/media/sd-mmcblk1pX` with X to be the partition number.

Read below on how to enable write protection on the SD and Micro SD slot, and on how to prepare a bootable SD with different capacities.

Enabling Write Protection

The SD slot does not support the write protection dip switch on SD cards or MicroSD-SD converter. However, it can be configured to read-only through bootloader. Micro SD slot can also be configured to read-only through bootloader too. Follow below steps to learn how to enable write protection on these slots.

1. Press <DEL> To Enter BIOS configuration Setting after powering on the device
2. Select (1) SD Card Write Protect.

```
--------------------------------------------------------------------------------
  Boot Loader Version 1.0.0S12        CPU TYPE: 1GHz
  Build date: May  7 2014 - 15:55:07   Serial Number: MOXATESTSN01
  LAN1 MAC: 00:90:E8:00:00:01         LAN2 MAC: 00:90:E8:00:00:02
--------------------------------------------------------------------------------
(0) TPM Setting                 (1) SD Card Write Protect
(2) Extend USB Port Control
--------------------------------------------------------------------------------
Command>>1
```

3. Select on which storage you would like to configure write protection on.

```
--------------------------------------------------------------------------------
  Boot Loader Version 1.0.0S12        CPU TYPE: 1GHz
  Build date: May  7 2014 - 15:55:07   Serial Number: MOXATESTSN01
  LAN1 MAC: 00:90:E8:00:00:01         LAN2 MAC: 00:90:E8:00:00:02
--------------------------------------------------------------------------------
(0) Boot Storage Write Protect    (1) Extend Storage Write Protect
--------------------------------------------------------------------------------
Command>>0
```

4. You will first see the current write protection status on the storage, and then you may select Enable or Disable write protect function.

```
Current Boot Storage Write Protect is Disabled.
Change to ,0 - Disabled, 1 - Enabled (0-1,enter for abort):
```

Above steps will mount the partitons on the storage to read-only after booting up. You may alther the read-only status in the OS by remounting the partitions. The command to use is **mount**. For example, to mount root directory to be read-writable, use **mount –o remount,rw /**, and when you don't need to write in the root directory, use **umount** to make it read-only again.

ATTENTION

If you create your own bootable SD or MicroSD card, please do not set the boot storage to be read-only when the system is booted up for the first time. The system is required to configure itself with read-writeable mode on the first boot. You may set up write protection after the first boot.

Preparing a Bootable SD Card

If you want to use SD card with greater capacity or upgrade UC-8100 firmware, you can download the latest UC-8100-LX image from Moxa official website and prepare a bootable SD card by yourself.

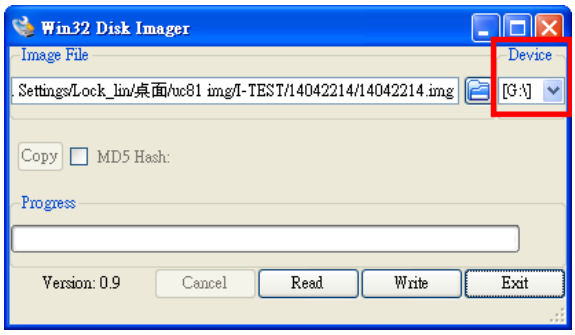
You can to download the image file to a eith Windows or Linux PC, and then transfer the file to SD card. Please refer to the following tutorial.

Creating a Linux System Image using a Windows Platform

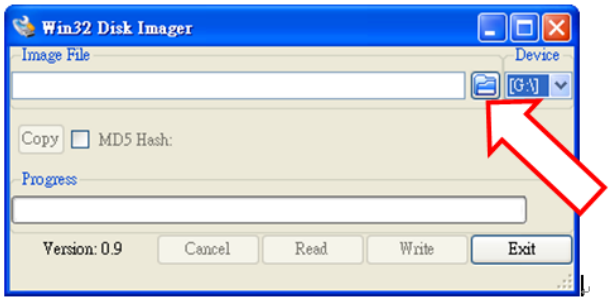
For Windows users, follow these steps.

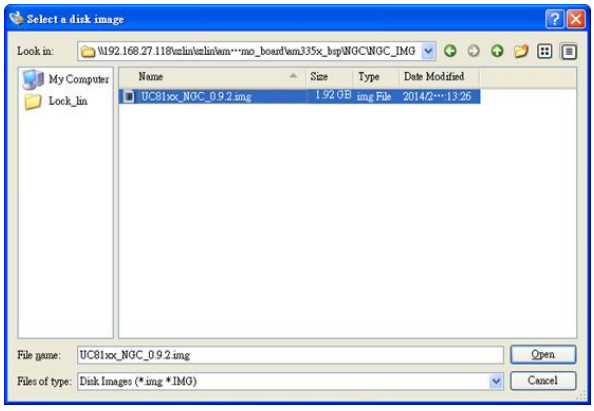
1. Make sure the write protection switch of the SD card is unlocked.

2. Insert the SD card into a Windows PC.
3. Download win32diskimager from following link.
   http://sourceforge.net/projects/win32diskimager/
4. Execute the win32diskimager after installation
5. Make sure the device name is match with the USB device.

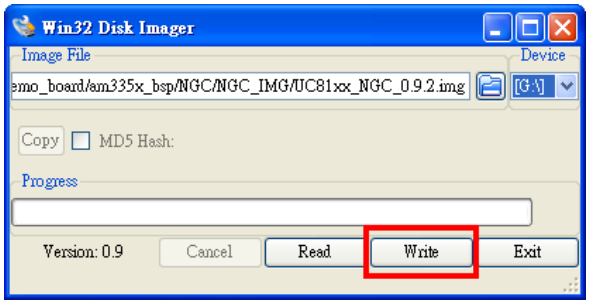


6. Select the image file.

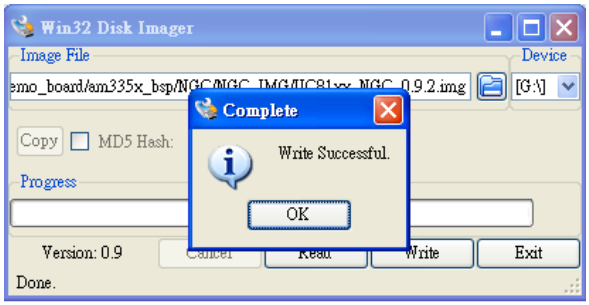




7. Make sure you have selected the correct image file. Click **Write** button.



8. When finished, click **OK**.



Creating a System Image in a Linux Environment

For Linux users, follow these steps.

1. Make sure the write protection switch of the SD card is unlocked.

2. Insert the SD card into a Linux PC.
3. Use **dmesg** command to find out the device node.

```
scsi 25:0:0:0: Direct-Access     TS-RDF5  SD  Transcend    TS35 PQ: 0 ANSI: 6
sd 25:0:0:0: Attached scsi generic sg3 type 0
sd 25:0:0:0: [sdd] 31260672 512-byte logical blocks: (16.0 GB/14.9 GiB)
sd 25:0:0:0: [sdd] Write Protect is off
sd 25:0:0:0: [sdd] Mode Sense: 23 00 00 00
sd 25:0:0:0: [sdd] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
 sdd: unknown partition table
sd 25:0:0:0: [sdd] Attached SCSI removable disk
```

4. Use **dd** command to configure the UC-8100-LX image on the SD card.

```
root@Lock-Lin:/home/work# sudo dd if=./140
42420.img of=/dev/sdd
bs=512k
          1954+0 records in

1954+0 records out
1024458752 bytes (1.0 GB) copied, 119.572 s, 8.6 MB/s
```

NOTE Click the following links for more information on the **dd** command.
http://www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html

Booting Up the UC-8100-LX for the first time

It is suggested to use the serial console to log in for the first time. Please refer to the Hardware Manual to see how to connect serial console.

File system resizing

Connect the UC-8100-LX to 12-24 VDC power source and the computer will immediately boot up. The power LED will be lit up first and then the SD Card LED. You will see messages printing out from the serial console too. Upon the first boot up, you will notice the he root filesystem is being resized and initialized with below message notification.

```
[....] Starting resize2fs_once...It will take some time to finish this action!:resize2fs 1.42.5 (29-Jul-2012)
Filesystem at /dev/root is mounted on /; on-line resizing required
old_desc_blocks = 4, new_desc_blocks = 29
[    9.563018] PHY: 0:10 - Link is Up - 100/Full
[    9.567718] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
Performing an on-line resize of /dev/root to 7550976 (1k) blocks.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
[ ok ] Starting Trusted Computing daemon: tcsd.
```

This can take a few minutes and the actual time required of this procedure will depend on the capacity of the SD card. The diagnostic LED indicators are activated as indicated in below table during resizing. Please wait patiently until the diagnostic LEDs are put out.

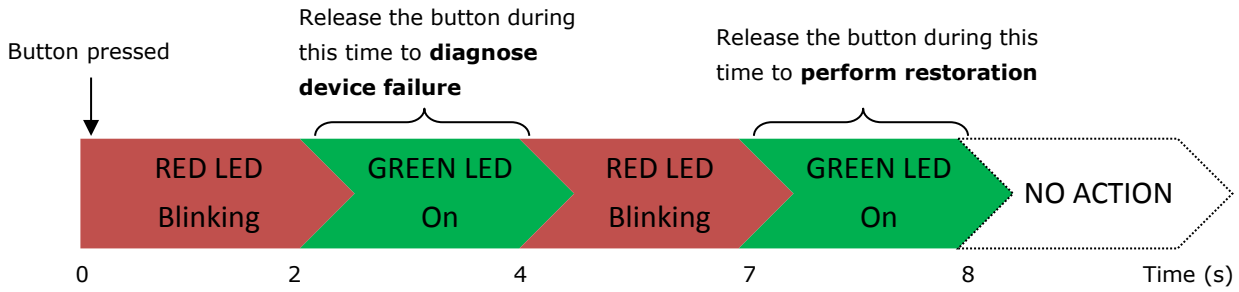
System Status	Diagnostic LED		
	RED	YELLOW	GREEN
Expanding root filesystem	Off	Blink	On

Booting from a MicroSD Card (UC-8112 Model Only)

The default boot up device of UC-8100-LX Series is the SD card. However, since the UC-8112-LX model provides the option to put the image on the internal MircoSD card, the UC-8112-LX will boot up from the MicroSD slot if no boot up code is found from SD card slot.

The Push Button and the LED indicators

The push button is used to diagnose device failure or to perform firmware restoration. Be alert to the indications to release the button so you can enter the correct mode to either diagnose your device or to restore your device back to default. See the figure and description for the indications.



The LED indicators have different behaviors when diagnosing for device failure and for performing firmware restoration. See below table for detail.

Status	Red LED	Yellow LED	Green LED
Execution of diagnostic program	Blink	Off	On
Reseting to default	Blink	Blink	On

Diagnosing Device and Subsystem Failures

The red LED will start blinking once you start to press the push button. Hold the button pressed until green LED is lit for the first time and release. This will enter the diagnostic mode for you to check the peripherals availabilities on UC-8100-LX.When diagnostic program is excuting, the red LED will be blinking.

Status	Red LED	Yellow LED	Green LED
Execution of diagnostic program	Blink	Off	On

Following 2 tables describes the diagnostic results. First table show hardware defects. If you observe any of the hardware issues, contact Moxa for further steps.

Status	Red LED	Yellow LED	Green LED
UART1 device issue	On	On	Off
UART2 device issue (except UC-8131)	On	On	Blink
LAN 1 device issue	On	Off	Off
LAN 2 device issue	On	Off	Blink
Button device issue	On	Blink	Off
TPM device issue	On	Blink	Blink
LED device issue	On	Off	Off

Second table is about system operation. If you observe any of the following issues, check if your UC-8100-LX is being occupied by bad programs.

Status	Red LED	Yellow LED	Green LED
CPU usage (over 90%)	Blink	On	Off
RAM usage (over 90%)	Off	On	Off
Disk usage (over 90 %)	Off	On	Blink
File system corrupted	Blink	On	Blink

Restoring Firmware to Factory Default

Hold the push button pressed until green LED is lit for the second time and release. This will enter the restoration process of UC-8100-LX and set the computer back to factoty default. The green LED will be lit, and the red and yellow LED indicators will be blinking when root filesystem is performing set-to-default function.

Status	Red LED	Yellow LED	Green LED
Reseting to default	Blink	Blink	On

You may also restore to factory default by issuing the command **setdef** in the OS.

```
moxa@Moxa:~$ sudo setdef
```

ATTENTION

Reset-to-default will erase all the data stored on the boot storage

Please backup your files before resetting the system to factory default. On UC-8100-LX series all the data stored in boot storage will be destroyed after reseting to factory default.

Using Cellular Modules

The UC-8100-LX series comes with a mini PCIe socket that a cellular module can be installed. Contact your sales representative for more information about available modules. Read below on to find out how to understand cellular signal strength from signal indicators, how to dial up on UC-8100-LX, and several advanced setting in cellular module.

Cellular Signal Strength

Check the following table for the cellular signal strength and its relation to the signal indicator.

Signal Indicator	Value	RSSI dbm	Condition
3 LEDs on (red, yellow, green)	20 to 30	-73 to -53	Excellent
2 LEDs on (red, yellow)	10 to 19	-93 to -74	Good
1 LED on (red)	2 to 9	-109 to -94	Marginal
No LED on	Else	Else	No signal

Cellular Dial-Up mode

For the 2 modules provided, it is suggested to dial up from QMI interface with QMI commands instead of using AT commands from the AT ports.

Module	MC-7304	MC-7354	MC-9090
Dial Up mode	QMI /dev/cdc-wdm1	QMI /dev/cdc-wdm1	QMI /dev/cdc-wdm0
AT Port	/dev/ttyUSB1	/dev/ttyUSB1	/dev/ttyUSB2
Default mode	7	7	8

Cellular GPS Port

Module	MC-7304	MC-7354	MC-9090
Device node	/dev/ttyUSB0	/dev/ttyUSB0	/dev/ttyUSB1

Configuring MC-7304/ MC-7354 Cellular Modules

MC-7304 and MC-7354 are both cellular modules that can be used on UC-8100-LX serried. Refer to datasheet for their specification. You may use UC-8100-LX cellular connection utitlity **cell_mgmt** to make cellular connection.

Dial-Up Connetions

APN is set manually in /etc/qmi-network.conf. Consult your carrier for the correct APN name and insert it into the configuration file like below: (APN=internet is used in the example, but your APN can be different)

```
moxa@Moxa:~$ echo "APN=internet" | sudo tee /etc/qmi-network.conf
```

To dial up with default configuration, use the following command.

```
moxa@Moxa:~$ sudo /sbin/cell_mgmt start
```

cell_mgmt is a Moxa script setup for your convenience. If you need to alter any options in making the cellular connection, you may use **qmi-network** and **qmi-cli** commands.

```
moxa@Moxa:~$ sudo qmi-network /dev/cdc-wdm0 start
Loading profile...
   APN: internet
Starting network with 'qmicli --device-open-flag-net-802-3 -d /dev/cdc-wdm0
--wds-start-network=internet  --client-no-release-cid'...
Saving state... (CID: 9)
Saving state... (PDH: 1205295888)
Network started successfully
```

Note that you need to manually start dhcp client if you use qmi-network to connect. The default interface of cellular connection is wwan0

```
moxa@Moxa:~$ dhclient wwan0
```

Disconnecting from a Dial-Up Network

Be sure to hang-up the connection if you don't need the service anymore. To disconnect, you many use the following command.

```
moxa@Moxa:~$ sudo /sbin/cell_mgmt stop
```

Alternatively you may use qmi-network too.

```
moxa@Moxa:~$ sudo qmi-network /dev/cdc-wdm0 stop
```

GPS

The GPS function of MC-7304/ MC-7354 is enabled by default.You may get raw GPS data by just listen to the GPS port /dev/ttyUSB1.

```
moxa@Moxa:~# cat /dev/ttyUSB1
```

Switching Between US Carrier Frequency Bands (MC-7354 only)

MC-7354 has pre-setup different profiles for different carriers in the US. If you are using the SIM card from Versizon, AT&T, or Sprint, you need to change to correspondant image for the module to connect. For ISPs other than mentioned carriers, you may chosse the generic profile.

Insert the following command to check which profile is currently used.

```
moxa@Moxa:~# echo -n -e 'ATE0\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -n -e 'AT!entercnd="A710"\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -n -e 'AT!GOBISETIMAGEPREF?\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~$ sudo cat /dev/ttyUSB1
```

Please follow the section corresponds to your carrier to switch profiles.

Verizon

For Verison users, use the following commands to switch to correspondant profile for Version.

```
moxa@Moxa:~# echo -n -e 'ATE0\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -n -e 'AT!entercnd="A710"\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -e -n
'AT!GOBISETIMAGEPREF="05.05.16.02","VZW","VZW_005.012_002"\r\n' | sudo tee
/dev/ttyUSB1
```

For the settings to come into effect, issue the following commands to re-initialize the cellular module.

```
moxa@Moxa:~$ sudo cell_mgmt power_off
moxa@Moxa:~$ sudo cell_mgmt power_on
```

AT&T

For AT&T users, use the following commands to switch to correspondant profile for AT&T.

```
moxa@Moxa:~# echo -n -e 'ATE0\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -n -e 'AT!entercnd="A710"\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -e -n
'AT!GOBISETIMAGEPREF="05.05.16.02","ATT","ATT_005.010_001"\r\n' | sudo tee
/dev/ttyUSB1
```

For the settings to come into effect, issue the following commands to re-initialize the cellular module.

```
moxa@Moxa:~$ sudo cell_mgmt power_off
moxa@Moxa:~$ sudo cell_mgmt power_on
```

Sprint

For Sprint users, use the following commands to switch to correspondant profile for Sprint.

```
moxa@Moxa:~# echo -n -e 'ATE0\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -n -e 'AT!entercnd="A710"\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -e -n 'AT!GOBISETIMAGEPREF="05.05.16.02","SPRINT","
SPRINT_005.011_000"\r\n' | sudo tee /dev/ttyUSB1
```

For the settings to come into effect, issue the following commands to re-initialize the cellular module.

```
moxa@Moxa:~$ sudo cell_mgmt power_off
moxa@Moxa:~$ sudo cell_mgmt power_on
```

Generic

For general users, use the following command.

```
echo -n -e 'ATE0\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -n -e 'AT!entercnd="A710"\r\n' | sudo tee /dev/ttyUSB1
moxa@Moxa:~# echo -e -n 'AT!GOBISETIMAGEPREF="05.05.16.02"," GENNA-UMTS","
GENNA-UMTS_005.009_000"\r\n' | sudo tee /dev/ttyUSB1
```

For the settings to come into effect, issue the following commands to re-initialize the cellular module.

```
moxa@Moxa:~$ sudo cell_mgmt power_off
moxa@Moxa:~$ sudo cell_mgmt power_on
```

Power on/off Module

cell_mgmt offers to re-initialize the module without rebooting UC-8100-LX series.You may issue commands to power off the module:

```
moxa@Moxa:~# sudo cell_mgmt power_off
```

And to re-initialize the cellular module, power on the module:

```
moxa@Moxa:~# sudo cell_mgmt power_on
```

NOTE You may find more information on qmi utilities in the following link.
http://www.freedesktop.org/wiki/Software/libqmi/

Configuring the Wireless LAN

You can configure the Wi-Fi connection on the UC-8100 Wi-Fi connection using a configuration file or the `wpa_supplicant` command (recommended).

NOTE You might encounter compatibility issues if you configure Wi-Fi settings using commands other than `wpa_supplicant`.

You can list the available wireless network IDs using the following command:

#iwlist wlan0 scanning

```
root@Moxa:# iwlist wlan0 scanning
wlan0     Scan completed :
          Cell 01 - Address: 50:67:F0:61:2D:7A
                    Protocol:802.11b/g
                    ESSID:"MIS-WAP-1"
                    Mode:Managed
                    Frequency:2.412 GHz (Channel 1)
                    Quality=81/100  Signal level=-58 dBm  Noise level=-92 dBm
                    Encryption key:on
                    Bit Rates:54 Mb/s
```

Configuring WPA2 Settings

The UC-8100 series computer supports WPA2 security using the **/sbin/wpa_supplicant** program. Refer to the following table to determine the configuration options. The **Key required before joining network?** column describes whether an encryption and/or authentication key must be configured before associating with a network.

Infrastructure mode	Authentication mode	Encryption status	Manual Key required?	IEEE 802.1X enabled?	Key required before joining network?
ESS	Open	None	No	No	No
ESS	Open	WEP	Optional	Optional	Yes
ESS	Shared	None	Yes	No	Yes
ESS	Shared	WEP	Optional	Optional	Yes
ESS	WPA	WEP	No	Yes	No
ESS	WPA	TKIP	No	Yes	No
ESS	WPA	AES	No	Yes	No
ESS	WPA-PSK	WEP	Yes	Yes	No
ESS	WPA-PSK	TKIP	Yes	Yes	No
ESS	WPA-PSK	AES	Yes	Yes	No

Connecting to an AP Using WEP Authentication

1. Edit the /etc/wpa_supplicant.conf file.

```
##### WEP #####
network={
            ssid="MIS-WAP-1"
            bssid=50:67:F0:61:2D:7A
            key mgmt=NONE
            wep key0=CFEE46EED3FA94FAEB92348922
}
################
```

The following table describes the related parameters.

Parameter	Usage	Function
ssid	{Acess Point Name}	Network name (as announced by the access point). An ASCII or hex string enclosed in quotation marks.
bssid	{MAC address of the AP}	Set network bssid, (typically the MAC address of the access point).
key_mgmt	{NONE,WEP,TKIP,AES}	List of acceptable key management protocols;
wep_key0	{wep key}	WEP key in hexadecimal format

2. Type `/usr/sbin/wifi_mgmt start` to enable this function.

   To stop the function, type `/usr/sbin/wifi_mgmt stop`.

NOTE For more information about wpa_supplicant.conf, go to the following websites:

- http://www.daemon-systems.org/man/wpa_supplicant.conf.5.html
- http://linux.die.net/man/5/wpa_supplicant.conf

Connecting to an AP Using WPA/WPA2 PSK Authentication

1. Edit the related parameters in the **/etc/wpa_supplicant.conf** file.

```
##### WPA/WPA2 PSK #####
network={
            ssid="5566"
            proto=WPA WPA2 RSN
            key mgmt=WPA-PSK
            pairwise=TKIP CCMP
            group=TKIP CCMP
            psk="01234567890"
}
######################
```

2. Type `/usr/sbin/wifi_mgmt start` to enable this function.

   To stop the function, type `/usr/sbin/wifi_mgmt stop`.

The following table describes the related parameters.

Parameter	Usage	Function
ssid	{Acess Point Name}	Network name (as announced by the access point). An ASCII or hex string enclosed in quotation marks.
proto	{WPA WPA2 RSN}	List of acceptable protocols; one or more of: WPA (IEEE802.11i/D3.0) and RSN (IEEE 802.11i). WPA2 is another name for RSN. The default value is "WPA RSN".

Parameter	Usage	Function
key_mgmt	{WPA-PSK or WPA-EAP}	List of acceptable key management protocols; one or more of: WPA-PSK (WPA pre-shared key), WPA-EAP (WPA using EAP authentica-tion), IEEE8021X (IEEE 802.1x using EAP authentication and, optionally, dynamically generated WEP keys). The default value is "WPA-PSK WPA-EAP".
pairwise	{TKIP CCMP, or NONE}	List of acceptable pairwise (unicast) ciphers for WPA; one or more of: CCMP (AES in Counter mode with CBC-MAC, RFC 3610, IEEE802.11i/D7.0), TKIP (Temporal Key Integrity Protocol, IEEE802.11i/D7.0), NONE (deprecated). The default value is "CCMP TKIP".
group	{CCMP, TKIP, WEP104, WEP40}	List of acceptable group (multicast) ciphers for WPA; one or more of: CCMP (AES in Counter mode with CBC-MAC, RFC 3610, IEEE802.11i/D7.0), TKIP (Temporal Key Integrity Protocol, IEEE802.11i/D7.0), WEP104 (WEP with 104-bit key), EP40 (WEP with 40-bit key). The default value is "CCMP TKIP WEP104 WEP40".
psk	{preshared key}	WPA preshared key used in WPA-PSK mode. The key is specified as 64 hex digits or as an 8-63 character ASCII passphrase.
mode	# 0 = infrastructure (Managed) mode, i.e., associate with an AP (default) # 1 = IBSS (ad-hoc, peer-to-peer)	IEEE 802.11 operation mode.

Using wpa_cli

`wpa_cli` is a text-based frontend program for interacting with `wpa_supplicant`. You can use the `wpa_cli` command to query the current status, change configuration, trigger events, and request user input.

NOTE Before you use the `wpa_cli` command, you must run the `wpa_supplicant` command.
For more information on wpa_cli, go to http://linux.die.net/man/8/wpa_cli.

Scanning APs and Viewing Scan Results

To scan access points in the area, enter the the command as shown in the figure.

```
root@Moxa:/home# wpa_cli -i wlan0 scan
OK
```

To display AP scan results, enter the command as shown in the figure.

```
root@Moxa:/home# wpa_cli -i wlan0 scan_results
bssid / frequency / signal level / flags / ssid
50:67:f0:61:2d:7a       2412    200     [WEP][ESS]
00:1f:1f:8c:0f:64       2462    210     [WPA2-PSK-CCMP-p
1c:7e:e5:93:ff:2a       2422    222     [WPA-PSK-TKIP+CC
b0:48:7a:a5:9b:70       2427    190     [WPA-PSK-CCMP][W
14:e6:e4:f0:57:5a       2442    182     [WPA-PSK-CCMP][W
54:04:a6:de:ce:dc       2412    186     [WPA2-PSK-CCMP][
c8:6c:87:78:af:7d       2412    174     [WPA2-PSK-TKIP+C
10:6f:3f:4c:af:e3       2462    166     [WPA-PSK-CCMP][E
```

Adding WEP Settings in a Configuration File

The following figure shows the related commands you can enter to add WEP configuration information in the **/etc/wpa_supplicant.conf** file.

```
root@Moxa:/home# wpa cli -i wlan0 add network
0
root@Moxa:/home# wpa_cli -i wlan0 set_network 0 key_mgmt NONE
OK
root@Moxa:/home# wpa_cli -i wlan0 set_network 0 ssid '"MOXA-AP-1"'
OK
root@Moxa:/home# wpa cli -i wlan0 set network 0 bssid 50:67:F0:61:2D:7A
OK
root@Moxa:/home# wpa_cli -i wlan0 set_network 0 wep_key0 AAEE431ED3FVV4FAEB923443C4
OK
root@Moxa:/home# wpa cli -i wlan0 enable network 0
OK
root@Moxa:/home# wpa cli -iwlan0 select network 0
OK
root@Moxa:/home# wpa_cli -i wlan0 save_config
```

Adding WPA/WPA2 Settings in a Configuration File

The following figure shows the related commands you can enter to add WPA/WPA2 configuration information in the **/etc/wpa_supplicant.conf** file.

```
root@Moxa:/home# wpa_cli -i wlan0 add_network
1
root@Moxa:/home# wpa cli -i wlan0 set network 1 ssid '"MOXA-AP"'
OK
root@Moxa:/home# wpa_cli -i wlan0 set_network 1 proto 'WPA WPA2 RSN'
OK
root@Moxa:/home# wpa cli -i wlan0 set network 1 key mgmt 'WPA-PSK'
OK
root@Moxa:/home# wpa cli -i wlan0 set network 1 pairwise 'TKIP CCMP'
OK
root@Moxa:/home# wpa_cli -i wlan0 set_network 1 group 'TKIP CCMP'
OK
root@Moxa:/home# wpa cli -i wlan0 set network 1 psk '"01234567890"'
'SET NETWORK 1 psk "01234567890"' command timed out.
root@Moxa:/home# wpa cli -i wlan0 enable network 1
OK
root@Moxa:/home# wpa_cli -iwlan0 select_network 1
OK
root@Moxa:/home# wpa cli -i wlan0 save config
OK
```

The following table lists the **wpa_cli** commands.

Command	Function
wpa_cli -i wlan0 status	Get current WEP/WPA/EAPOL/EAP status.
wpa_cli -i wlan0 help	Show this usage help.
wpa_cli -i wlan0 terminate	Terminate wpa_supplicant.
wpa_cli -i wlan0 interface	Show interfaces or select an interface.
wpa_cli -i wlan0 list_networks	List configured networks in wpa_supplicant.conf.
wpa_cli -i wlan0 select_network	Set network variables. Network id can be received from the LIST_NETWORKS command output. This command uses the same variables and data formats as the configuration file.
wpa_cli -i wlan0 enable_network	Enable a network. Network id can be received from the LIST_NETWORKS command output.

Command	Function
wpa_cli -i wlan0 disable_network	Disable a network. Network id can be received from the LIST_NETWORKS command output. Special network id "all" can be used to disable all networks.
wpa_cli -i wlan0 remove_network	Remove a network. Network id can be received from the LIST_NETWORKS command output. Special network id "all" can be used to remove all networks.
wpa_cli -i wlan0 reconfigure	Force wpa_supplicant to re-read its configuration file.
wpa_cli -i wlan0 save_config	Save the current configuration. Replace original /etc/wpa_supplicant.conf file.
wpa_cli -i wlan0 scan wpa_cli -i wlan0 scan_results	Scan available networks. Get scanning results.

4

Security On UC-8100-LX

UC-8100-LX series offer better security by introducing Moxa's innovative secure boot feature. Also, the intergration of a Trusted Platform Module gives the user more solid protection to the platform.

The following topics are covered in this chapter:

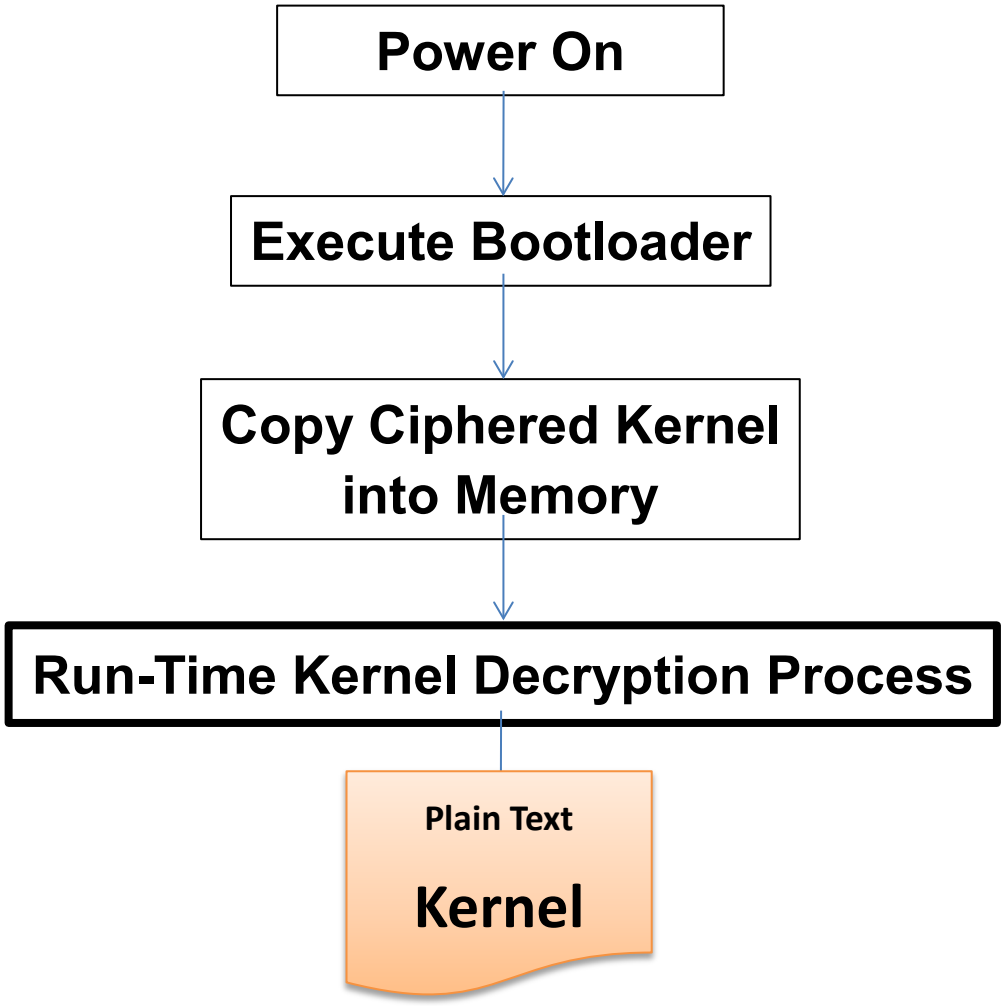
- **Secure Boot**
- **Trusted Platform Module (TPM) and TrouSerS**
  - Enabling TPM via the Bootloader
  - Start TPM Services
  - Initializing the Trusted Platform Module
  - Getting the Public Endorsement Key
  - Sealing/Unsealing Data
- **SUDO Mechanism**

Secure Boot

Secure boot is a novel authentication algorithm developed by Moxa which is proposed to secure platform integration. Only trusted Linux kernel and bootloader could be executed; malicious or un-authenticated kernel will not be able to boot up the UC-8100-LX. All UC-8100-LX series support this feature by default.

For UC-8100-LX, the kernel file will be stored on SD card in cipher text. This is the first protection for a secure platform that whoever copies the kernel file cannot understand or to add malicious code easily.

Next, during boot up, the ciphered kernel will be checked and decrypted into to plain kernel. In case the kernel is being replaced by malicious code, the predefined decryption will not make the code into excutable kernel.



ATTENTION

DO NOT replace kernel or bootloader arbitrary, or the computer will not be able to boot up.

NOTE Secure Boot is only provided with UC-8100-LX standard image. The provided source on Moxa website does not include source for Secure Boot feature.

Trusted Platform Module (TPM) and TrouSerS

TPM is a microcontroller that can securely store artifacts like passwords, certificates, or encryption keys which are used to authenticate the platform. It provides hardware-based protection of data 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). The full TPM specification can be found in the provided references at the end of this section.

This TPM hardware needs a stable software and TrouSerS is the implementation of TCG Software Stack (TSS) which contains the tcsd daemon and the TPM tool for user to access to and communicate with the TPM. These packagages are all installed within UC-8112-LX for user to utilize.

Below is the list of the supported TPM tool commands

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

And the PKCS#11 data management commands of the TPM Tools.

Command	Description
tpmtoken_import	Import an X.590 certificate and/or an RSA key pair into the user's tPM PKCS#11 data store
tpmtoken_init	Initialize the user's TPM PKCS#11 data store
tpmtoken_objects	Display the objects in the user's TPM PKCS#11 data store
tpmtoken_protect	Encrypt or decrypt data using a symmetric key stored in the user's TPM PKCS#11 data store
tpmtoken_setpasswd	Change the passwords associated with the user's TPM PKCS#11 data store

NOTE Click on the 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 TPM via the Bootloader

To start using TPM on UC-8100, you need to first enable TPM function from bootloader, and then start the related services in the OS. Follow these steps to enable the TPM.

1. Press <DEL> To Enter BIOS configuration Setting after powering the device
2. Select (0) TPM Setting

```
--------------------------------------------------------------------------------
  Boot Loader Version 1.0.0S11       CPU TYPE: 1GHz
  Build date: Apr 25 2014 - 15:29:07  Serial Number: IMOXA1234567
  LAN1 MAC: 00:90:e8:00:00:07        LAN2 MAC: 00:90:e8:00:00:08
--------------------------------------------------------------------------------
        (0) TPM Setting                (1) SD Card Write Protect
(2) Extend USB Port Control
--------------------------------------------------------------------------------
Command>>0
```

3. Select (0) TPM Function is Enable

```
--------------------------------------------------------------------------------
  Boot Loader Version 1.0.0S11       CPU TYPE: 1GHz
  Build date: Apr 25 2014 - 15:29:07  Serial Number: IMOXA1234567
  LAN1 MAC: 00:90:e8:00:00:07        LAN2 MAC: 00:90:e8:00:00:08
--------------------------------------------------------------------------------
(0) TPM Function is Enable        (1) TPM Function is Disable
--------------------------------------------------------------------------------
Command>>0
1.2 TPM (Chip Type: SLB9645TT, Device-ID: 0x1a)
TPM status check.........is enabled and activated

### Please reboot the system to complete the operation ###
```

4. After setting, power off and then power on the device.

Start TPM Services

To make the services recongise the TPM hardware, you need to first enable TPM from bootloader. The TPM related tools are **trousers** and **opencryptoki**.

Put **trousers** and **opencryptoki** daemons in default running services.

```
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
```

You may reference reference the section **Enabling and Disabling Daemons** for how to use **insserv** and the **/etc/init.d/** scripts.Next you may check if the TPM is accessible, and get the TPM version like below.

```
moxa@Moxa:~$ sudo tpm_version
  TPM 1.2 Version Info:
  Chip Version:       1.2.133.32
  Spec Level:         2
  Errata Revision:    3
  TPM Vendor ID:      IFX
  Vendor Specific data: 85200050 0074706d 3438ffff ff
  TPM Version:        01010000
  Manufacturer Info:  49465800
```

NOTE Please make sure TPM daemon is running before issuing TPM tool commands, otherwise you will get following error:

Tspi_Context_Connect failed: 0x00003011 - layer=tsp, code=0011 (17), Communication failure

Initializing the Trusted Platform Module

First step to start using TPM is to take its ownership. This can be done through the following command:

```
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 If you encounter the following error : …

`Tspi_TPM_GetPubEndorsementKey failed: 0x00000023 - layer=tpm, code=0023 (35), No EKlease make sure …..`

This is because your TPM does not have an Endorsement Key. Then, launch the following command and wait for it to complete.

`sudo tpm_createek`

ATTENTION

Ownership can only be realized once, any other attempt will fail

Keep your owner/SRK password carefully.

Getting the Public Endorsement Key

The Endorsement Key is typically a 2,048-bit RSA public and private key pair, which is created randomly on the chip at manufacture time and cannot be changed. The private key never leaves the chip, while the public key is used for attestation and for encryption of sensitive data sent to the chip.

```
moxa@Moxa:~$ sudo tpm_getpubek
Tspi_TPM_GetPubEndorsementKey failed: 0x00000008 - layer=tpm, code=0008 (8), The TPM
target command has been disabled
Enter owner password:
Public Endorsement Key:
Version:   01010000
Usage:     0x0002 (Unknown)
Flags:     0x00000000 (!VOLATILE, !MIGRATABLE, !REDIRECTION)
AuthUsage: 0x00 (Never)
Algorithm:        0x00000020 (Unknown)
Encryption Scheme: 0x00000012 (Unknown)
Signature Scheme:  0x00000010 (Unknown)
Public Key:
       b1000e32 269ee2bd f2114775 dd553e8a b9bac458 cfd52496 b6dd590b 776e2fd1
       4a519f1c e1fe6085 d8365f02 261bc6f0 e1e7f2e0 833da920 970cd588 d1e6939e
       3f35a8a8 251f298d 78c46e34 d68ef7cc 7a685d9e baf7f6e5 e3bcc303 163e9e67
       395426dc 39c90b37 9aa17f55 6fbea49e 0a76fc01 cafd9062 772112c4 c9207e6c
       ebda664c 7a6cbda8 301dcc4c 67dc8f03 9ea8993a 1f9068ec 9757ec8e 26b4c6e2
       87e30470 6fbf4ae3 3e32b5b7 dfe55dfc 4da3012d b6a600bb d7eeed48 99c118b5
       4950c2f5 1527c78f 12dfcea7 d9dfdc8a 10cd442a d3f17173 784a69c7 9689c822
       f364af90 2802bfcd 5a1227c2 3c7d02b0 e7e804a3 abe8034b 3584c529 1265a881
```

Sealing/Unsealing Data

tpm_sealdata and **tpm_unsealdata** command are used to seal or unseal data .They are invoked with the following parameters:

```
-i, --infile FILE

            Filename containing key to seal/unseal. Default is STDIN.

-o, --outfile FILE

            Filename to write sealed/unseal key to.  Default is STDOUT.

-p, --pcr NUMBER

            PCR to seal data to.  Default is none.  This option can be specified
multiple times to choose more than one PCR.
```

After invoking the **tpm_sealdata** function, **tpm_sealdata** retrieves random data from the TPM. To do this, the **tpmGetRandom** function invokes the method Tspi_TPM_GetRandom() of the class TPM . Then tpm_sealdata sets the SRK policy using the classes Policy and Context. 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.

Sealing Data

```
moxa@Moxa:~$ tpm_sealdata -i secrect -o secrect.enc -p 12 -p 14
Enter SRK password:
```

Unsealing Data

```
moxa@Moxa:~$ tpm_unsealdata -i secrect.enc -o plain
```

SUDO Mechanism

In the UC-8100-LX, the root account is disabled for better security. Sudo is a program designed to let system administrators allow some users to execute some commands as root (or another user). The basic philosophy is to give as few privileges as possible but still allow people to get their work done. Using sudo is better (safer) than opening a session as root for a number of reasons, including:

Nobody needs to know the root password (sudo prompts for the current user's password). Extra privileges can be granted to individual users temporarily, and then taken away without the need for a password change.

It is easy to run only the commands that require special privileges via sudo; the rest of the time, you work as an unprivileged user, which reduces the damage that mistakes can cause.

The code below shows that some system level command is not accessable to user moxa directly.

```
moxa@Moxa:~$ ifconfig
-bash: ifconfig: command not found

moxa@Moxa:~$ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr 00:90:e8:00:00:07
          inet addr:192.168.3.127  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST ALLMULTI 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)

eth1      Link encap:Ethernet  HWaddr 00:90:e8:00:00:08
          inet addr:192.168.4.127  Bcast:192.168.4.255  Mask:255.255.255.0
          UP BROADCAST ALLMULTI 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)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2592 (2.5 KiB)  TX bytes:2592 (2.5 KiB)
```

5

General Debian Package Usage

This chapter explains how to configure the UC-8100-LX's various functions.

The following topics are covered in this chapter:

- **NTP Client**
- **Execute Scheduled Commands with cron**
  - Updating System Time and RTC
- **Rocket-Fast System for Log Processing: rsyslog**
  - Rsyslog's Configuration File
  - Syntax of the Selector
- **OpenSSL**
  - Ciphers
  - Cryptographic Hash Functions
- **The Apache Web Server**
  - Edit ServerName in Apache Configuration File
- **SFTP**
- **DNS**
- **IPTABLES**
  - Observe and Erase Chain Rules
  - Define Policy for Chain Rules
  - Append or Delete Rules
- **rsync**
  - Using rsync for External Backups
  - Automating rsync Backups
- **NAT**
  - NAT Example
  - Enabling NAT at Bootup
- **NFS (Network File System)**
  - Setting up UC-8100-LX as an NFS Client
- **SNMP**
- **OpenVPN**
  - Static-Key VPN

NTP Client

The UC-8100-LX has a built-in NTP (Network Time Protocol) client that is used to initialize a time request to a remote NTP server. Use **#ntpdate <*this client utility*>** to update the system time.

ntpdate 192.168.1.97
hwclock –w

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

```
  192.168.4.127 - PuTTY
moxa@Moxa:~$ sudo ntpdate 192.168.50.33
 6 May 03:55:10 ntpdate[4511]: step time server 192.168.50.33 offset 78338115.278119
sec
moxa@Moxa:~$ sudo hwclock -w
moxa@Moxa:~$ sudo hwclock
Tue 06 May 2014 03:56:14 AM UTC  -0.846314 seconds
```

NOTE Before using the NTP client utility, check your IP and DNS settings to make sure that an Internet connection is available. Refer to Chapter 2 for instructions on how to configure the Ethernet interface, and see Chapter 4 for DNS setting information.

Execute Scheduled Commands with cron

The cron daemon reads /etc/crontab to retrieve scripts and other commands to be run at regularly scheduled times.

Cron wakes up every minute and checks each command listed in the crontab file to see if it should be run at that time. Whenever cron executes a command, a report is automatically mailed to the owner of the crontab (or to the user named in the MAILTO environment variable in the crontab, if such a user exists).

Modify the file /etc/crontab to schedule an application. Crontab entries follow the format below:

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

For example, issue the following command if you want to launch a program at 8:00 every day:

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

Every column in a crontab entry must be marked with a character. The asterisk indicates "every possible unit," so that setting an asterisk in the day-of-week column will configure cron to run the command on every day of the week. If you wish to run a command "every X minutes" or "every X hours", then use the format */X.

Updating System Time and RTC

You may use cron to update the system time and RTC. Follow these steps.

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

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

2. Reset the access permissions for fixtime.sh

```
moxa@MOXA:~# chmod 755 fixtime.sh
```

3. Modify the /etc/crontab file to run fixtime.sh every 10 minutes (i.e.: */10) by adding this line:

```
*/10 * * * * root /home/fixtime.sh
```

NOTE Click the following link for more information on cron.
http://www.debian-administration.org/articles/56

Rocket-Fast System for Log Processing: rsyslog

Rsyslog is an enhanced, multi-threaded log reporting utility with a focus on security and reliability. It offers support for on-demand disk buffering, log reports and alarms delvered over TCP, SSL, TLS and RELP, writing to databases, and email alerting. It is a drop-in replacement for syslogd.

Rsyslog is installed but disabled by default.

Enable rsyslog manually	/etc/init.d/rsyslog start
Disable rsyslog manually	/etc/init.d/rsyslog stop
Enable rsyslog	insserv -d rsyslog
Disable rsyslog	insserv -r rsyslog

Rsyslog's Configuration File

The syntax of the **/etc/rsyslog.conf** file is detailed in the rsyslog.conf(5) manual page, but there is also HTML documentation available in the rsyslog-doc package **(/usr/share/doc/rsyslog-doc/html/index.html)**. The overall principle is to write "selector" and "action" pairs. The selector defines all relevant messages, and the actions describe how to deal with them.

Each message is associated with an application, called a *facility* in rsyslog documentation	
auth and authpriv	for authentication
cron	comes from task scheduling services, cron and atd
daemon	affects a daemon without any special classification (DNS, NTP, etc.)
ftp	concerns the FTP server
kern	message coming from the kernel
lpr	comes from the printing subsystem
mail	comes from the e-mail subsystem
news	Usenet subsystem message (especially from an NNTP — Network News Transfer Protocol — server that manages newsgroups)
syslog	messages from the syslogd server, itself
user	user messages (generic)
uucp	messages from the UUCP server (Unix to Unix Copy Program, an old protocol notably used to distribute e-mail messages)
local0 to local7	reserved for local use
Each message is also associated with a priority level. Here is the list in decreasing order:	
emerg	**Help!** There's an emergency, the system is probably unusable.
alert	hurry up, any delay can be dangerous, action must be taken immediately
crit	conditions are critical
err	error
warn	warning (potential error)
notice	conditions are normal, but the message is important
info	informative message
debug	debugging message

Syntax of the Selector

The selector is a semicolon-separated list of *subsystem*.*priority* pairs (example: auth.notice;mail.info). An asterisk may represent all subsystems or all priorities (examples: *.alert or mail.*). Several subsystems can be grouped, by separating them with a comma (example: auth,mail.info). The priority indicated also covers

messages of equal or higher priority; thus auth.alert indicates the auth subsystem messages of alert or emerg priority. Prefixed with an exclamation point (!), it indicates the opposite, in other words the strictly lower priorities; auth.!notice, thus, indicates messages issued from auth, with info or debug priority. Prefixed with an equal sign (=), it corresponds to precisely and only the priority indicated (auth.=notice only concerns messages from auth with notice priority).

Each element in the list on the selector overrides previous elements. It is thus possible to restrict a set or to exclude certain elements from it. For example, kern.info;kern.!err means messages from the kernel with priority between info and warn. The none priority indicates the empty set (no priorities), and may serve to exclude a subsystem from a set of messages. Thus, *.crit;kern.none indicates all the messages of priority equal to or higher than crit not coming from the kernel.

NOTE Click the following link for more information on rsyslog.
https://wiki.debian.org/Rsyslog
http://www.rsyslog.com/doc/

OpenSSL

UC8100 supports hardware accelerator with openssl. Type **lsmod** to make sure the **cryptodev** module is loaded.

```
Module                  Size  Used by
cryptodev              30504  1
```

Make sure the version of openssl, it should modified by MOXA

```
moxa@Moxa:~$ dpkg -l | grep openssl
ii  openssl                      1.0.1e-2+deb7u7+uc8100 armhf       Secure Socket
Layer (SSL) binary and related cryptographic tools on MOXA uc8100
```

Before enabling hardware accelerator

```
root@Moxa:/home# openssl speed -evp aes-128-cbc
Doing aes-128-cbc for 3s on 16 size blocks: 5625719 aes-128-cbc's in 2.95s
Doing aes-128-cbc for 3s on 64 size blocks: 1769561 aes-128-cbc's in 2.94s
Doing aes-128-cbc for 3s on 256 size blocks: 498367 aes-128-cbc's in 2.99s
Doing aes-128-cbc for 3s on 1024 size blocks: 125670 aes-128-cbc's in 2.95s
Doing aes-128-cbc for 3s on 8192 size blocks: 16023 aes-128-cbc's in 2.99s
OpenSSL 1.0.1e 11 Feb 2013
built on: Mon Apr  7 03:26:32 UTC 2014
options:bn(64,32) rc4(ptr,char) des(idx,cisc,16,long) aes(partial) idea(int) blowfish(ptr)

compiler: gcc -fPIC -DOPENSSL_PIC -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_
H -DHAVE_CRYPTODEV -DUSE_CRYPTDEV_DIGESTS -march=armv7-a -Wa,--noexecstack -DTERMIO -O3 -W
all -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DAES
_ASM -DGHASH_ASM
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
aes-128-cbc      30512.37k    38521.06k    42669.55k    43622.40k    43899.80k
```

After enabling hardware accelerator

```
moxa@Moxa:~$ sudo openssl speed -evp aes-128-cbc
[sudo] password for moxa:
Doing aes-128-cbc for 3s on 16 size blocks: 261302 aes-128-cbc's in 0.14s
Doing aes-128-cbc for 3s on 64 size blocks: 222033 aes-128-cbc's in 0.13s
Doing aes-128-cbc for 3s on 256 size blocks: 139516 aes-128-cbc's in 0.16s
Doing aes-128-cbc for 3s on 1024 size blocks: 48524 aes-128-cbc's in 0.09s
Doing aes-128-cbc for 3s on 8192 size blocks: 8126 aes-128-cbc's in 0.00s
OpenSSL 1.0.1e 11 Feb 2013
built on: Mon Apr 21 06:14:54 UTC 2014
options:bn(64,32) rc4(ptr,char) des(idx,cisc,16,long) aes(partial) idea(int)
blowfish(ptr)
compiler: gcc -fPIC -DOPENSSL_PIC -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN
-DHAVE_DLFCN_H -DHAVE_CRYPTODEV -DUSE_CRYPTDEV_DIGESTS -march=armv7-a
```

```
-Wa,--noexecstack -DTERMIO -O3 -Wall -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_GF2m
-DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DAES_ASM -DGHASH_ASM
The 'numbers' are in 1000s of bytes per second processed.
type          16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
aes-128-cbc     29863.09k   109308.55k   223225.60k   552095.29k       infk
```

OpenSSL supports a number of different cryptographic algorithms:

Ciphers

Ciphers supports the following cryptographic methods:

AES, Blowfish, Camellia, SEED, CAST-128, DES, IDEA, RC2, RC4, RC5, Triple DES, GOST 28147-89

Cryptographic Hash Functions

MD5, MD4, MD2, SHA-1, SHA-2, RIPEMD-160, MDC-2, GOST R 34.11-94

Public-key cryptography

RSA, DSA, Diffie–Hellman key exchange, Elliptic curve, GOST R 34.10-2001

NOTE Make sure the version of openssl is built by MOXA, or the hardware accelerator function will not work in other version

The Apache Web Server

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

APACHE has been installed but is disabled by default.

Enable apache manually	sudo /etc/init.d/apache2 start
Disable apache manually	sudo /etc/init.d/apache2 stop
Enable apache	insserv -d apache2
Disable apache	insserv -r apache2

Edit ServerName in Apache Configuration File

Edit apache2.conf.

```
moxa@Moxa:~$ sudo vi /etc/apache2/apache2.conf
```

Add servername of this device in apache2.conf.

```
ServerName xxx
```

Restart apache2.

```
moxa@Moxa:~$ sudo /etc/init.d/apache2 restart
```

NOTE Click the following link for more information on apache.
https://wiki.debian.org/Apache
http://httpd.apache.org/

SFTP

The default sftp is started, and the account/password is **moxa/moxa** as the same as the system account/password. You can also configure sftp account with the following steps.

1. Creating a user & group for the sftp access without any shell

```
moxa@Moxa:~$ sudo adduser sftp
[sudo] password for moxa:
Adding user `sftp' ...
Adding new group `sftp' (1003) ...
Adding new user `sftp' (1001) with group `sftp' ...
Creating home directory `/home/sftp' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for sftp
Enter the new value, or press ENTER for the default
       Full Name []:
       Room Number []:
       Work Phone []:
       Home Phone []:
       Other []:
Is the information correct? [Y/n]
```

```
moxa@Moxa:~$ sudo usermod -s /bin/false sftp
```

2. Then a modification to the current user made in the debian installation fase. In this example, we use ftpuser as the user.

```
moxa@Moxa:~$ sudo mkdir /home/sftp/upload/
moxa@Moxa:~$ sudo chown root:root /home/sftp
moxa@Moxa:~$ sudo chown sftp:sftp /home/sftp/upload/
```

3. Adapt sshd configuration in the bottom of /etc/ssh/sshd_config.

```
Subsystem sftp internal-sftp
#Subsystem sftp /usr/lib/openssh/sftp-server
```

```
Match User sftp
ChrootDirectory /home/%u
ForceCommand internal-sftp
```

4. Restart SSHD Daemon:

```
moxa@Moxa:~$ sudo /etc/init.d/sshd restart
```

5. The account and its default path are already done

NOTE Click the following link for more information on SSH.
https://wiki.debian.org/SSH

DNS

The UC-8100-LX supports DNS client (but not DNS server). To set up DNS client, you need to edit three configuration files: **/etc/hosts**, **/etc/resolv.conf**, and **/etc/nsswitch.conf**.

/etc/hosts

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

/etc/resolv.conf

This is the most important file that you need to edit when using DNS for the other programs. For example, before you using #ntpdate time.nist.goc to update the system time, you will need to add the DNS server address to the file. Ask your network administrator which DNS server address you should use. The DNS server's IP address is specified with the "nameserver" command. For example, add the following line to /etc/resolv.conf if the DNS server's IP address is 168.95.1.1:

nameserver 168.95.1.1

```
10.120.53.100 - PuTTY
moxa@Moxa:~$ sudo cat /etc/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
```

/etc/nsswitch.conf

This file defines the sequence to resolve the IP address by using /etc/hosts file or /etc/resolv.conf.

IPTABLES

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

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

The UC-8100-LX supports three types of IPTABLES table: **Filter** tables, **NAT** tables, and **Mangle** tables:

Filter Table—includes three chains:

INPUT chain

OUTPUT chain

FORWARD chain

NAT Table—includes three chains:

PREROUTING chain—transfers the destination IP address (DNAT)

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

OUTPUT chain—produces local packets

sub-tables

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

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

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

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

Mangle Table—includes two chains, and it has three extensions—TTL, MARK, TOS.

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

OUTPUT chain—processes packets after the routing process.

The following figure shows the IPTABLES hierarchy.

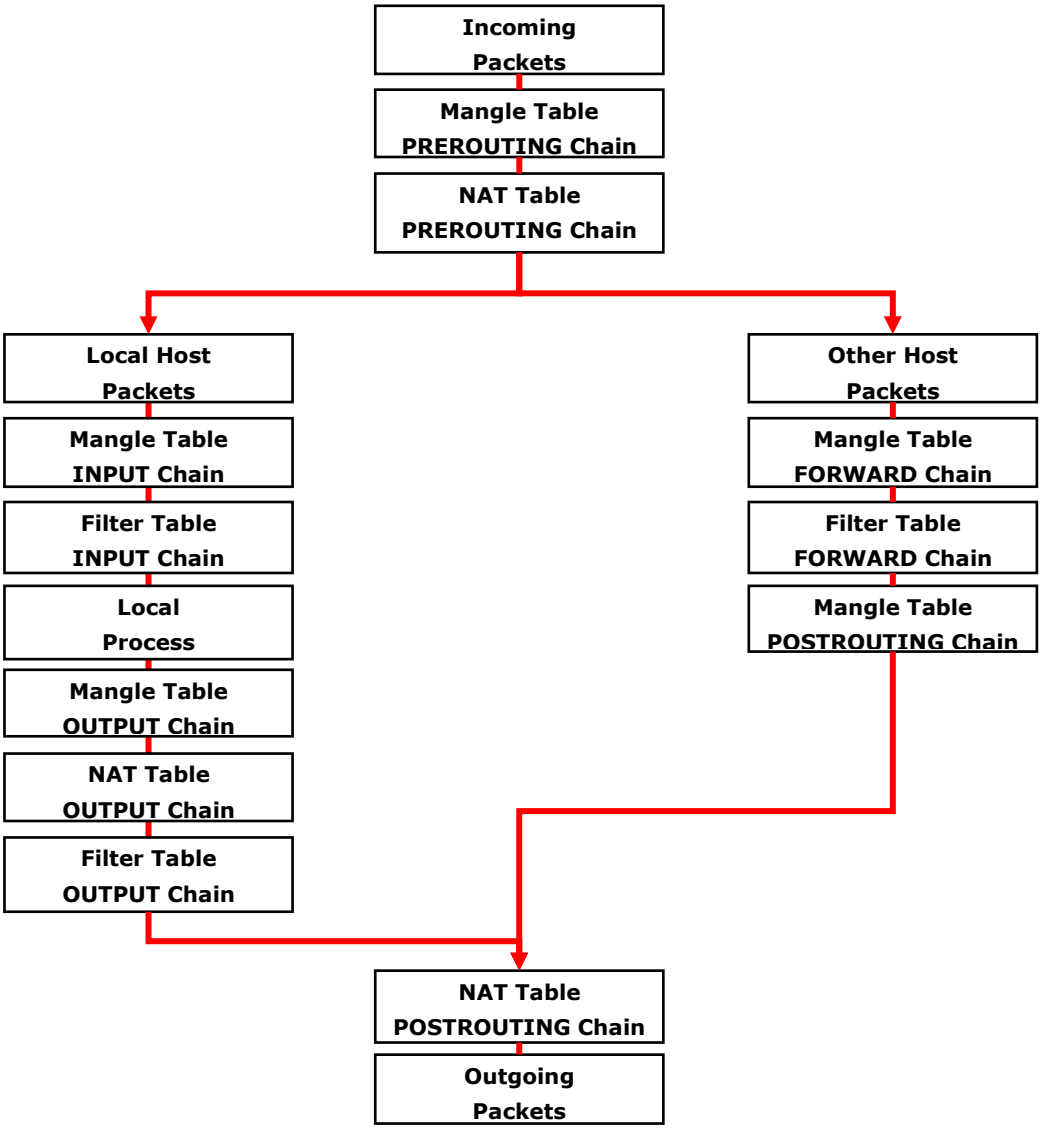
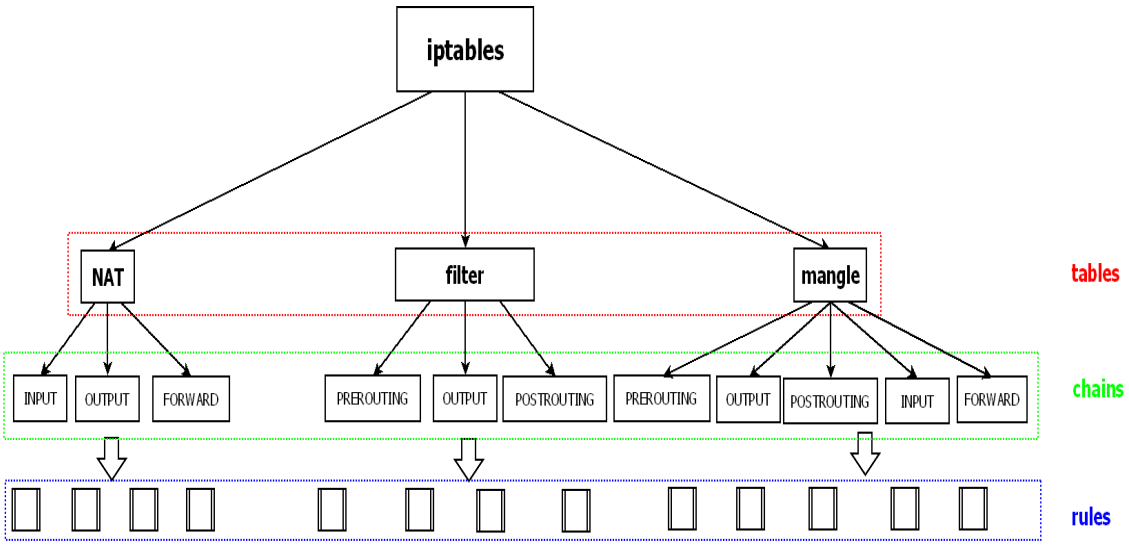


Table	Chain	Rule
NAT (Network translation translation)	PREROUTING	Types of rule • **Policy** • **Self-defined** Targets of rule • **ACCEPT** • **DROP** • **REJECT** • **LOG** • SNAT • DNAT • MASQUERADE
	POSTROUTING	
	OUTPUT	
Filter (Default) (Packet filtering)	INPUT	
	OUTPUT	
	FORWARD	
Mangle (Packet header modification)	PREROUTING	
	INPUT	
	FORWARD	
	OUTPUT	
	POSTROUTING	



The UC-8100-LX supports the following sub-modules. Be sure to use the module that matches your application.

Most of common module is already built-in with the kernel

ip6t_eui64.ko	ip6t_ipv6header.ko	nf_conntrack_ipv6.ko	xfrm4_mode_tunnel.ko
ip6t_rt.ko	ip6t_LOG.ko	xfrm6_mode_beet.ko	ah4.ko
ip6table_security.ko	ip6t_ah.ko	sit.ko	xfrm4_mode_beet.ko
ip6table_filter.ko	ip6_tables.ko	ipv6.ko	xfrm4_mode_transport.ko
ip6t_frag.ko	ip6table_raw.ko	xfrm6_mode_tunnel.ko	esp4.ko
ip6t_hbh.ko	nf_defrag_ipv6.ko	xfrm6_mode_transport.ko	ipcomp.ko
ip6t_REJECT.ko	ip6t_mh.ko	xfrm_ipcomp.ko	tcp_diag.ko
inet_lro.ko	xfrm4_tunnel.ko	inet_diag.ko	

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

Use **`lsmod`** to check if the ip_tables module has already been loaded in the UC-8100-LX series. Use **`modprobe`** to insert and enable the module.

Use the following command to load the modules (iptable_filter, iptable_mangle, iptable_nat):

```
#modprobe iptable_filter
```

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

NOTE IPTABLES plays the role of packet filtering or NAT. Take care when setting up the IPTABLES rules. If the rules are not correct, remote hosts that connect via a LAN or PPP may be denied access. We recommend using the serial console to set up the IPTABLES.
Click on the following links for more information on iptables.
http://www.linuxguruz.com/iptables/
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html
https://wiki.debian.org/DebianFirewall
https://wiki.debian.org/iptables

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

Observe and Erase Chain Rules

Usage:

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

-t tables: Table to manipulate (default: 'filter'); example: nat or filter.

-L [chain]: List List all rules in selected chains. If no chain is selected, all chains are listed.

-n: Numeric output of addresses and ports.

 # iptables [-t tables] [-FXZ]

-F: Flush the selected chain (all the chains in the table if none is listed).

-X: Delete the specified user-defined chain.

-Z: Set the packet and byte counters in all chains to zero.

Examples:

 # iptables -L -n

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

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

Define Policy for Chain Rules

Usage:

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

-P: Set the policy for the chain to the given target.

INPUT: For packets coming into the UC-8100-LX series.

OUTPUT: For locally-generated packets.

FORWARD: For packets routed out through the UC-8100-LX series.

PREROUTING: To alter packets as soon as they come in.

POSTROUTING:To alter packets as they are about to be sent out.

Examples:

#iptables –P INPUT DROP

#iptables –P OUTPUT ACCEPT

#iptables –P FORWARD ACCEPT

 # modprobe iptable_nat

#iptables –t nat –P PREROUTING ACCEPT

#iptables –t nat –P OUTPUT ACCEPT

#iptables -t nat –P POSTROUTING ACCEPT

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

Append or Delete Rules

Usage:

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

-A: Append one or more rules to the end of the selected chain.

-I: Insert one or more rules in the selected chain as the given rule number.

-i: Name of an interface via which a packet is going to be received.

-o: Name of an interface via which a packet is going to be sent.

-p: The protocol of the rule or of the packet to check.

-s: Source address (network name, host name, network IP address, or plain IP address).

--sport: Source port number.

-d: Destination address.

--dport: Destination port number.

-j: Jump target. Specifies the target of the rules; i.e., how to handle matched packets. For example, ACCEPT the packet, DROP the packet, or LOG the packet.

Examples:

Example 1: Accept all packets from 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.

modprobe modprobe xt_tcpudp

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

Example 6: Accept TCP packets from 192.168.0.24 to UC-8100 series's port 137, 138, 139

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

Example 7: Log TCP packets that visit UC-8100 series's port 25.

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

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

modprobe xt_mac

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

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

rsync

rsync is a utility software and network protocol that synchronizes files and directories from one location to another while minimizing data transfer by using delta encoding when appropriate. It also has the option to provide encrypted transfer by use of SSH. SSL encrypted transfer can be done via Stunnel wrapping. rsync uses the 'rsync algorithm' which provides a very fast method for bringing remote files into sync. rsync can copy or display directory contents and copy files, optionally using compression and recursion.

Using rsync for local backups

You can backup your data to secure data via rsync

We could backup our data in rootfilesystem - /Directory1 to expand storage microSD or USB - /Directory2.

```
moxa@Moxa:~$ sudo rsync -avP /Directory1/ /Directory2/
```

-v, --verbose increase verbosity

-a, --archive archive mode; equals -rlptgoD (no -H,-A,-X)

-P --progress show progress during transfer

--partial keep partially transferred files

Using rsync for External Backups

rsync can be configured in several different ways for external backups, but we will go over the most practical (also the easiest and most secure) method of tunneling rsync through SSH. Most servers and even many clients already have SSH, and it can be used for your rsync backups. We will show you the process to get one Linux machine to backup to another on a local network. The process would be the exact same if one host were out on the internet somewhere, just note that port 22 (or whatever port you have SSH configured on), would need to be forwarded on any network equipment on the server's side of things.

Other than installing SSH and rsync on the server, all that really needs to be done is to setup the repositories on the server where you would like the files backed up, and make sure that SSH is locked down. Make sure the user you plan on using has a complex password, and it may also be a good idea to switch the port that SSH listens on (default is 22).

We will run the same command that we did for using rsync on a local computer, but include the necessary additions for tunneling rsync through SSH to a server on my local network. For user "user" connecting to "192.168.1.1" and using the same switches as above (-avP) we will run the following:

```
moxa@Moxa:~$ sudo rsync -avP -e ssh /Directory1/ user@192.168.1.1:/Directory2/
```

Automating rsync Backups

Cron can be used on Linux to automate the execution of commands, such as rsync. Using Cron, we can have our Linux system run nightly backups, or however often you would like them to run.

To edit the cron table file for the user you are logged in as, run:

```
moxa@Moxa:~$ sudo crontab -e
```

You will need to be familiar with vi in order to edit this file. Type "I" for insert, and then begin editing the cron table file.

Cron uses the following syntax: minute of the hour, hour of the day, day of the month, month of the year, day of the week, command.

It can be a little confusing at first, so let me give you an example. The following command will run the rsync command every night at 10 PM:

```
0 22 * * * rsync -avP /Directory1/ /Directory2/
```

The first "0" specifies the minute of the hour, and "22" specifies 10 PM. Since we want this command to run daily, we will leave the rest of the fields with asterisks and then paste the rsync command.

NOTE Click the following link for more information on iptables and rsync.
http://rsync.samba.org/

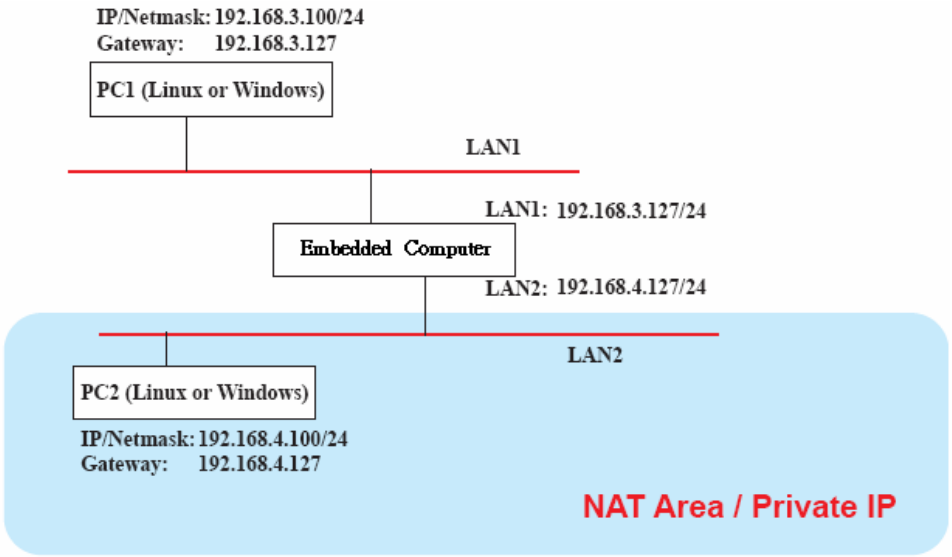
NAT

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-682A-LX connects several devices on a network and maps local inside network addresses to one or more global outside IP addresses, and un-maps the global IP addresses on incoming packets back into local IP addresses.

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

NAT Example

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



Enabling NAT at Bootup

In most real world situations, you will want to use a simple shell script to enable NAT when the DA-682A-LX boots up. The following script is an example.

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

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

# Step 1. Insert modules.

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

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

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

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

# Step 3. Enable IP masquerade.

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

NFS (Network File System)

The Network File System (NFS) is used to mount a disk partition on a remote machine, as if it were on a local hard drive, allowing fast, seamless sharing of files across a network. NFS allows users to develop applications for UC-8100 series, without worrying about the amount of disk space that will be available. UC-8100 series supports NFS protocol for client.

NFS has been installed but disabled by default. Check the following table for details.

Enable nfs manually	sudo /etc/init.d/nfs-common start sudo /etc/init.d/nfs-kernel-server start sudo /etc/init.d/rpcbind start
Disable nfs manually	sudo /etc/init.d/nfs-common stop sudo /etc/init.d/nfs-kernel-server stop sudo /etc/init.d/rpcbind stop
Enable nfs	insserv -d nfs-common insserv -d nfs-kernel-server insserv -d /etc/init.d/rpcbind
Disable nfs	insserv -r nfs-common insserv -r nfs-kernel-server insserv -r /etc/init.d/rpcbind

Setting up UC-8100-LX as an NFS Client

The following procedure is used to mount a remote NFS Server.

Establish a mount point on the NFS Client site.

Mount the remote directory to a local directory.

Steps 1:

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

```
Step 2:
#mount  -t  nfs  NFS_Server(IP):/directory  /mount/point
```

Example

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

NOTE Click the following links for more information on NFS.
http://www.tldp.org/HOWTO/NFS-HOWTO/index.html
http://nfs.sourceforge.net/nfs-howto/client.html
http://nfs.sourceforge.net/nfs-howto/server.html

SNMP

UC-8100 series has built-in SNMP (Simple Network Management Protocol) agent software. It supports RFC1317 RS-232 like group and RFC 1213 MIB-II. SNMP daemon is installed but disabled by default. You may activate the daemon manually or set it to be enabled by default.

You will need to start/stop the service with following command.

Start snmpd manually	sudo /etc/init.d/snmpd start
Stop snmpd manually	sudo /etc/init.d/snmpd stop
Enable snmpd	insserv -d snmpd
Disable snmpd	insserv -r snmpd

The UC-8100-LX has built-in SNMP (Simple Network Management Protocol) agent software.

The following simple example allows you to use an SNMP browser on the host site to query the UC-8100 series, which is the SNMP agent. UC-8100 series will respond.

debian:~# snmpwalk -v 2c -c public -Cc 192.168.27.115

iso.3.6.1.2.1.1.1.0 = STRING: "Linux Moxa 3.2.0_UC81XX #3 Thu Apr 24 10:38:04 CST 2014 armv7l"

iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8691.12.8100

iso.3.6.1.2.1.1.3.0 = Timeticks: (201692) 0:33:36.92

iso.3.6.1.2.1.1.4.0 = STRING: "Moxa Inc., Embedded Computing Business. <www.moxa.com>"

iso.3.6.1.2.1.1.5.0 = STRING: "Moxa"

iso.3.6.1.2.1.1.6.0 = STRING: "Fl.4, No.135, Lane 235, Baoquao Rd., Xindian Dist., New Taipei City, Taiwan, R.O.C.\""

iso.3.6.1.2.1.1.7.0 = INTEGER: 72

iso.3.6.1.2.1.1.8.0 = Timeticks: (4) 0:00:00.04

iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1

iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1

iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1

iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1

iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.2.1.49

iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.4

iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50

iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.16.2.2.1

iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The SNMP Management Architecture MIB."

iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and Dispatching."

iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for the SNMP User-based Security Model."

iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"

iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP implementations"

iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP implementations"

iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"

iso.3.6.1.2.1.1.9.1.3.8 = STRING: "View-based Access Control Model for SNMP."

iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (3) 0:00:00.03

iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (3) 0:00:00.03

iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (3) 0:00:00.03

iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (4) 0:00:00.04

iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (4) 0:00:00.04

iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (4) 0:00:00.04

iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (4) 0:00:00.04

iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (4) 0:00:00.04

iso.3.6.1.2.1.25.1.1.0 = Timeticks: (2866708) 7:57:47.08

iso.3.6.1.2.1.25.1.2.0 = Hex-STRING: 07 DE 05 0D 0A 12 15 00 2B 00 00

iso.3.6.1.2.1.25.1.3.0 = INTEGER: 1536

iso.3.6.1.2.1.25.1.4.0 = STRING: "mac=00:90:e8:00:00:07 sd=0 ver=1.0.0S11 console=ttyO0,115200n8 root=/dev/mmcblk0p2 rootfstype=ext4 rootwait”

iso.3.6.1.2.1.25.1.5.0 = Gauge32: 1

iso.3.6.1.2.1.25.1.6.0 = Gauge32: 58

iso.3.6.1.2.1.25.1.7.0 = INTEGER: 0

iso.3.6.1.2.1.25.1.7.0 = No more variables left in this MIB View (It is past the end of the MIB tree)

NOTE Click the following links for more information on MIB II.
http://www.faqs.org/rfcs/rfc1213.html
https://wiki.debian.org/SNMP

OpenVPN

The OpenVPN package is installed but disable in default. Please type “insserv –d openvpn” to enable after next booting up. You can also type “/etc/init.d/openvpn start” to enable service of openvpn immediately.

OpenVPN support user/pass, pre-shared key, certificates etc. to authenticate users To begin with, check to make sure that the system has a virtual device **/dev/net/tun**.

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.

```
# modprobe tun
```

On each OpenVPN machine, you should generate a working directory, such as **/etc/openvpn**, where script files and key files reside. Once established, all operations will be performed in that directory.

OpenVPN daemon has been installed but disabled by default.

Enable openvpn manually	sudo /etc/init.d/openvpn start
Disable openvpn manually	sudo /etc/init.d/openvpn stop
Enable openvpn	insserv -d openvpn
Disable openvpn	insserv -r openvpn

Static-Key VPN

In the server's /etc/openvpn directory, run the following command to generate a static key

```
moxa@Moxa:/etc/openvpn$ sudo openvpn --genkey --secret static.key
```

Copy this static key to the clients /etc/openvpn directory using a secure channel like scp or sftp.

On the server, create a new /etc/openvpn/tun0.conf file and add the following:

```
dev tun0
ifconfig 10.9.8.1 10.9.8.2
secret /etc/openvpn/static.key
```

Where 10.9.8.x is your VPN subnetwork, 10.9.8.1 will be IP of the server, 10.9.8.2 is IP of client.

On the client, copy /etc/openvpn/static.key from server and create a new /etc/openvpn/tun0.conf file and add the following:

```
remote your-server.org
dev tun0
ifconfig 10.9.8.2 10.9.8.1
secret /etc/openvpn/static.key
```

Start OpenVPN by hand on both sides with the following command

```
moxa@Moxa:/etc/openvpn$ sudo openvpn --config /etc/openvpn/tun0.conf --verb 6  //
verbose output.
```

ATTENTION

Firewall should create policy for OpenVPN-related application

On the server's firewall, open up UDP 1194 (default port). If you are using ?shorewall, on both devices, add a new VPN zone to represent tun0 and create a default policy for it. This means adding something to the following files in /etc/shorewall:

- zone
- interfaces
- policy

Bear in mind that 90% of all connection problems encountered by new OpenVPN users are firewall-related.

NOTE Click the following links for more information on OpenVPN.
https://wiki.debian.org/OpenVPN
http://openvpn.net/

Package Management

This article explains how quickly you can learn to install, remove, update and search software packages using apt-get and apt-cache commands from the command line. This article provides some useful commands that will help you to handle package management in Debian/Ubuntu based systems.

apt-get

The apt-get utility is a powerful and free package management command line program, that is used to work with Ubuntu's APT (Advanced Packaging Tool) library to perform installation of new software packages, removing existing software packages, upgrading of existing software packages and even used to upgrading the entire operating system.

apt-cache

The apt-cache command line tool is used for searching apt software package cache. In simple words, this tool is used to search software packages, collects information of packages and also used to search for what available packages are ready for installation on Debian or Ubuntu based systems.

List All Available Packages

Use the following command to list all available packages.

```
moxa@Moxa:~$ sudo apt-cache pkgnames
```

Find Out Package Name and Description of Software

To find out the package name and the description, use the 'search' flag. Using "search" with apt-cache will display a list of matched packages with short description. Let's say you would like to find out description of package 'vim', then command would be.

```
moxa@Moxa:~$ sudo apt-cache search vim
```

To find and list down all the packages starting with 'vim', you can use the following command.

```
moxa@Moxa:~$ sudo apt-cache pkgnames vim
```

Check Package Information

If you would like to check information of package along with it short description say (version number, check sums, size, installed size, category etc). Use '**show**' sub command as shown below.

```
moxa@Moxa:~$ sudo apt-cache show vim
```

Check Dependencies for Specific Packages

Use the '**showpkg**' sub command to check the dependencies for particular software packages. whether those dependencies packages are installed or not. For example, use '**showpkg**' command along with package-name.

```
moxa@Moxa:~$ sudo apt-cache showpkg vim
```

Check statistics of Cache

The '**stats**' sub command will display overall statistics about the cache. For example, the following command will display Total package names is the number of packages have found in the cache.

```
moxa@Moxa:~$ sudo apt-cache stats
```

Update System Packages

The '**update**' command is used to resynchronize the package index files from the their sources specified in **/etc/apt/sources.list** file. The updated commands will fetch the packages from their locations and update the packages to newer version.

```
moxa@Moxa:~$ sudo apt-get update
```

Install or Upgrade Specific Packages

The '**install**' sub command is tracked by one or more packages wish for installation or upgrading.

```
moxa@Moxa:~$ sudo apt-get install vim
```

Upgrade All Software Packages

The **upgrade** command is used to upgrade all the currently installed software packages on the system. Under any circumstances currently installed packages are not removed or packages which are not already installed neither retrieved and installed to satisfy upgrade dependencies.

```
moxa@Moxa:~$ sudo apt-get upgrade
```

Install Multiple Packages

You can add more than one package name along with the command in order to install multiple packages at the same time. For example, the following command will install packages 'vim' and 'goaccess'.

```
moxa@Moxa:~$ sudo apt-get install vim   goaccess
```

Install Several Packages using Wildcard

With the help of regular expression you can add several packages with one string. For example, we use ***** wildcard to install several packages that contains the '***name***' string, name would be 'package-name'.

Install Packages without Upgrading

Using sub '**–no-upgrade**' command will prevent the installed packages from upgrading.

```
moxa@Moxa:~$ sudo apt-get install packageName --no-upgrade
```

Upgrade Specific Packages

The '–only-upgrade' command do not install new packages but it only upgrade the already installed packages and disables new installation of packages.

```
moxa@Moxa:~$ sudo apt-get install packageName --only-upgrade
```

Install Specific Package Version

Let's say you wish to install only specific version of packages, simply use the '**=**' with the package-name and append desired version.

```
moxa@Moxa:~$ sudo apt-get install wget=1.13.4-3+deb7u1
```

Remove Packages Without Configuration

To un-install software packages without removing their configuration files (for later re-use the same configuration). Use the **'remove'** command as shown.

```
moxa@Moxa:~$ sudo apt-get remove wget
```

Completely Remove Packages

To remove software packages including their configuration files, use the 'purge' sub command as shown below.

```
moxa@Moxa:~$ sudo apt-get remove --purge wget
```

Clean Up Disk Space

The '**clean**' command is used to free up the disk space by cleaning retrieved (downloaded) **.deb** files (packages) from the local repository.

```
moxa@Moxa:~$ sudo apt-get clean
```

Download Only Source Code of Package

To download only source code of particular package, use the option '**–download-only source**' with 'package-name' as shown.

```
moxa@Moxa:~$ sudo apt-get --download-only source wget
```

Download and Unpack a Package

To download and unpack source code of a package to a specific directory, type the following command.

```
moxa@Moxa:~$ sudo apt-get source wget
```

Download, Unpack and Compile a Package

You can also download, unpack and compile the source code at the same time, using option '–compile' as shown below.

```
moxa@Moxa:~$ sudo apt-get --compile source wget
```

Download a Package Without Installing

Using '**download**' option, you can download any given package without installing it. For example, the following command will only download '**nethogs**' package to current working directory.

```
moxa@Moxa:~$ sudo apt-get download wget
```

Check Change Log of Package

The '**changelog**' flag downloads a package change-log and shows the package version that is installed.

```
moxa@Moxa:~$ sudo apt-get changelog wget
```

Check Broken Dependencies

The '**check**' command is a diagnostic tool. It used to update package cache and checks for broken dependencies.

```
moxa@Moxa:~$ sudo apt-get check
```

Search and Build Dependencies

This 'build-dep' command searches the local repositories in the system and install the build dependencies for package. If the package does not exists in the local repository it will return an error code.

```
moxa@Moxa:~$ sudo apt-get build-dep wget
```

Auto Clean Apt-Get Cache

The '**autoclean**' command deletes all **.deb** files from **/var/cache/apt/archives** to free-up significant volume of disk space.

```
moxa@Moxa:~$ sudo apt-get autoclean
```

Auto Remove Installed Packages

The '**autoremove**' sub command is used to auto remove packages that were certainly installed to satisfy dependencies for other packages and but they were now no longer required. For example, the following command will remove an installed package with its dependencies.

```
moxa@Moxa:~$ sudo apt-get autoremove wget
```

6

Programmer's Guide

This chapter briefely introduces the tool-chain and teaches you how to program with UC-8100-LX. Please go to Moxa official website to download the example pagckage and read through this chapter if you intend to develop your own program on UC-8100-LX.

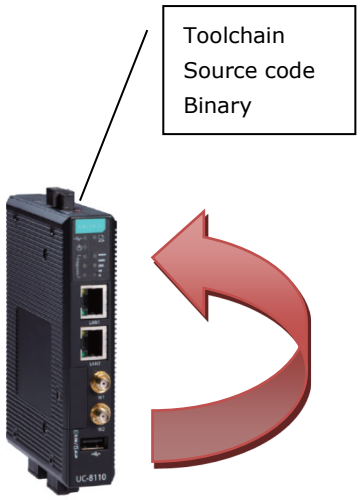
The following topics are covered in this chapter:

- **Linux Tool Chain Introduction**
  - Native Compilation
  - Cross Compilation
  - Obtaining Help
- **Test Program—Developing Hello.c**
  - Compiling Hello.c with Native Compilation
  - Compiling Hello.c with Cross Compilation
- **Modbus**
- **RTC (Real Time Clock)**
- **WDT (Watch Dog Timer)**
- **Cryptographic Hardware Accelerator**
- **Diagnostic LED**
  - Turn on LEDs API
  - Turn off LED API
  - Blink LED API
- **TPM**

Linux Tool Chain Introduction

Linux Tool-Chain contains the necessary libraries and compilers for you to develop your program. UC-8100 series supports both native and cross- compile. Native compile is more straightforward as the all the coding and compilation can be done directly on UC-8100-LX but constrained by the resource of ARM CPU, the speed for compilation is slower. On the other hand, cross compile can be done on any Linux machine with the correct tool-chain and is much faster in terms of compilation speed.

Native Compilation



Follow these steps to update package menu.

1. Make sure network connection is available.
2. Use **apt-get update** to update Debian package list.

```
moxa@Moxa:~$ sudo apt-get update
```

3. Install native compiler and necessary packages

```
moxa@Moxa:~$ sudo apt-get install gcc build-essential flex bison automake
```

Cross Compilation

To ensure that an application will be able to run correctly when installed on UC-8100-LX series, you must ensure that it is compiled and linked to the same libraries that will be present on the UC-8100-LX series. This is particularly true when the RISC Cortex processor architecture of the UC-8100 series differs from the CISC x86 processor architecture of the host system, but it is also true if the processor architecture is the same.

The host tool chain that comes with UC-8100-LX series contains a suite of cross compilers and other tools, as well as the libraries and headers that are necessary to compile applications for UC-8100 series. The host environment must be running Linux to install the UC-8100 series GNU Tool Chain. We have confirmed that the following Linux distributions can be used to install the tool chain:

Redhat 7.3/8.0/9.0, Fefora core 1~20, Debian 4/5/6,7 32/64 bits platform.

The Tool Chain will need about 300 MB of hard disk space on your PC. To install, download the tool-chain file from Moxa official website.

After you untar the package, run the install script and follow its instruction.

```
user@Linux:/home#sh arm-linux-gnueabihf_4.7_Build_130415.sh


Welcome to MOXA ARM Linux platform toolchain installer.
This toolchain built with arm-linux-gnueabihf compiler v4.7.3 and glibc v2.15.
Any problem please contact support@moxa.com

Press the number:
1.Install Linux cross compiler tool.
2.Uninstall Linux cross compiler tool.
3.Exit or CTRL+C
1
usr/local/arm-linux-gnueabihf-4.7-20130415/
usr/local/arm-linux-gnueabihf-4.7-20130415/bin/
usr/local/arm-linux-gnueabihf-4.7-20130415/bin/arm-linux-gnueabihf-gcc-ranlib
usr/local/arm-linux-gnueabihf-4.7-20130415/bin/arm-linux-gnueabihf-ld
usr/local/arm-linux-gnueabihf-4.7-20130415/bin/arm-linux-gnueabihf-objcopy
usr/local/arm-linux-gnueabihf-4.7-20130415/bin/arm-linux-gnueabihf-ld.gold

...
...

usr/local/arm-linux-gnueabihf-4.7-20130415/lib/gcc/arm-linux-gnueabihf/4.7.3/incl
ude/stdbool.h
usr/local/arm-linux-gnueabihf-4.7-20130415/lib/gcc/arm-linux-gnueabihf/4.7.3/incl
ude/mf-runtime.h
usr/local/arm-linux-gnueabihf-4.7-20130415/lib/gcc/arm-linux-gnueabihf/4.7.3/incl
ude/mmintrin.h
usr/local/arm-linux-gnueabihf-4.7-20130415/lib/gcc/arm-linux-gnueabihf/4.7.3/incl
ude/stddef.h
usr/local/arm-linux-gnueabihf-4.7-20130415/20130415-gcc-linaro-arm-linux-gnueabih
f
-----------------------------------------------
arm-linux-gnueabihf install complete
Please export these environment variables before using toolchain:
export PATH=$PATH:/usr/local/arm-linux-gnueabihf-4.7-20130415/bin
```

Wait for a few minutes while the Tool Chain is installed automatically on your Linux PC. Once the host environment has been installed, add the directory **/usr/local/arm-linux-gnueabihf-4.7-20130415//bin** to your path and the directory **/usr/local/arm-linux-gnueabihf-4.7-20130415//man** to your manual path. You can do this temporarily for the current login session by issuing the following commands:

#export PATH="/usr/local/arm-linux-gnueabihf-4.7-20130415//bin:$PATH"

#export MANPATH="/usr/local/arm-linux-gnueabihf-4.7-20130415//man:$MANPATH"

Alternatively, you can add the same commands to **$HOME/.bash_profile** to cause it to take effect for all login sessions initiated by this user.

NOTE The toolchain will be installed at /usr/local/arm-linux-gnueabihf-4.7-20130415/. This means that the original /usr/local/arm-linux-gnueabihf-4.7-20130415/ path will be overwrited. If you have installed an old arm-linux toolchain, you could have to rename the original one before you install the new one.

Obtaining Help

Use the Linux **man** utility to obtain help on many of the utilities provided by the tool chain which is located at /usr/local/arm-linux-gnueabihf-4.7-20130415/share/doc/gcc-linaro-arm-linux-gnueabihf/man/. For example to get help on the **arm-linux-gnueabihf-gcc** compiler, issue the command:

```
moxa@Moxa:~$ man
/usr/local/arm-linux-gnueabihf-4.7-20130415/share/doc/gcc-linaro-arm-linux-gnueab
ihf/man/man1/arm-linux-gnueabihf-gcc.1
```

Cross Compiling Applications and Libraries

To compile a simple C application, just use the cross compiler instead of the regular compiler:

#arm-linux-gnueabihf-gcc –o example –Wall –g –O2 example.c

#arm-linux-gnueabihf-strip –s example

#arm-linux-gnueabihf-gcc -ggdb –o example-debug example.c

Test Program—Developing Hello.c

In this section, we use the standard "Hello" programming example to illustrate how to develop a program for the UC-8100-LX.

```
#include <stdio.h>
int main()
{
       printf("Hello World\n");
       return 0;
}
```

The following compiler tools are provided:

ar	Manage archives (static libraries)
as	Assembler
c++, g++	C++ compiler
cpp	C preprocessor
gcc	C compiler
gdb	Debugger
ld	Linker
nm	Lists symbols from object files
objcopy	Copies and translates object files
objdump	Displays information about object files
ranlib	Generates indexes to archives (static libraries)
readelf	Displays information about ELF files
size	Lists object file section sizes
strings	Prints strings of printable characters from files (usually object files)

strip	Removes symbols and sections from object files (usually debugging information)

Compiling Hello.c with Native Compilation

Follow these steps for native compilation.

1. Install tool-chain via APT server.
2. sudo gcc –o hello-release hello.c
3. sudo strip –s hello-release

After compilation, you can run the executeable file

```
moxa@Moxa:~$ ./hello-release
Hello World
```

Compiling Hello.c with Cross Compilation

Follow these steps for cross compilation.

1. Connect UC-8100-LX series to a Linux PC.
2. Install Tool Chain (GNU Cross Compiler & glibc).
3. Set the cross compiler and glibc environment variables.
4. Code and compile the program.
5. Download the program to UC-8100 series via SFTP/ NFS/ SCP or RSYNC.
6. Debug the program

→ If bugs are found, return to Step 4.

→ If no bugs are found, continue with Step 7

7. Back up the user directory (distribute the program to additional UC-8100 series units if needed).

The package CD contains several example programs. Here we use **Hello.c** as an example to show you how to compile and run your applications. Type the following commands from your PC to copy the files used for this example from the CD to your computer's hard drive:

```
# cd /tmp/
# mkdir example
# cp –r /mnt/cdrom/example/* /tmp/example
```

To compile the program, go to the **Hello** subdirectory and issue the following commands:

```
#cd example/hello
#make
```

You should receive the following response:

```
[root@localhost hello]# make
arm-linux-gnueabihf-gcc -o hello-release  hello.c
arm-linux-gnueabihf-strip -s hello-release
```

hello-release—an ARM platform execution file (created specifically to run on the UC-8100 series)

Uploading and Running the "Hello" Program

The program could upload via SFTP/ NFS/ SCP or RSYNC

Use the following command to upload **hello-release** to the UC-8100 series via SFTP.

From the PC, type:
#ftp 192.168.3.127

Use "put" command to initiate the file transfer:

```
sftp> put hello-release
Uploading hello-release to /home/moxa/hello-release
hello-release
```

From the UC-8100-LX, type:

 # chmod +x hello-release

 # ./hello-release

The word **Hello** will be printed on the screen.

```
moxa@Moxa:~$ ./hello-release
Hello World
```

Makefile Example

The following Makefile is copied from the Hello example on the UC-8100-LX's example package. It is used for cross compile.

```
CC = arm-linux-gnueabihf-gcc

CPP = arm-linux-gnueabihf-g++

SOURCES = hello.c

OBJS =  $(SOURCES:.c=.o)

all:   hello

hello: $(OBJS)

   $(CC) -o $@ $^ $(LDFLAGS) $(LIBS)

clean:

   rm -f $(OBJS) hello core *.gdb
```

For Native compile, change

```
CC = gcc

CPP = g++
```

Modbus

Modbus Protocol is a messaging structure which is used to establish master-slave/client-server communication between intelligent devices. It is a de facto standard, truly open and the most widely used network protocol in the industrial manufacturing environment. It has been implemented by hundreds of vendors on thousands of different devices to transfer discrete/analog I/O and register data between control devices. It's a lingua franca or common denominator between different manufacturers. One report called it the "de facto standard in multi-vendor integration".

The libmodbus version in Debian 7 is v3.03

We use libmodbus as our modbus package

Please download the source and example code from the following link.

https://github.com/downloads/stephane/libmodbus/libmodbus-3.0.3.tar.gz

NOTE Click the following link for more information on libmodbus.
http://libmodbus.org/

RTC (Real Time Clock)

The device node is located at **/dev/rtc0**. UC-8100-LX series supports Linux standard simple RTC control. You must **include <linux/rtc.h>** in your program to use the following functions.

Function	**RTC_RD_TIME**
Description	Read time information from RTC. It will return the value on argument 3.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_RD_TIME, &rtc_tm);
Function	**RTC_SET_TIME**
Description	Set RTC time. Argument 3 will be passed to RTC.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_SET_TIME, &rtc_tm);
Function	**RTC_ALM_SET**
Description	Set alarm time.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_ALM_SET, &rtc_tm);
Function	**RTC_ALM_READ**
Description	Read alarm time.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_ALM_READ, &rtc_tm);
Function	**RTC_IRQP_SET**
Description	Set IRQ rate

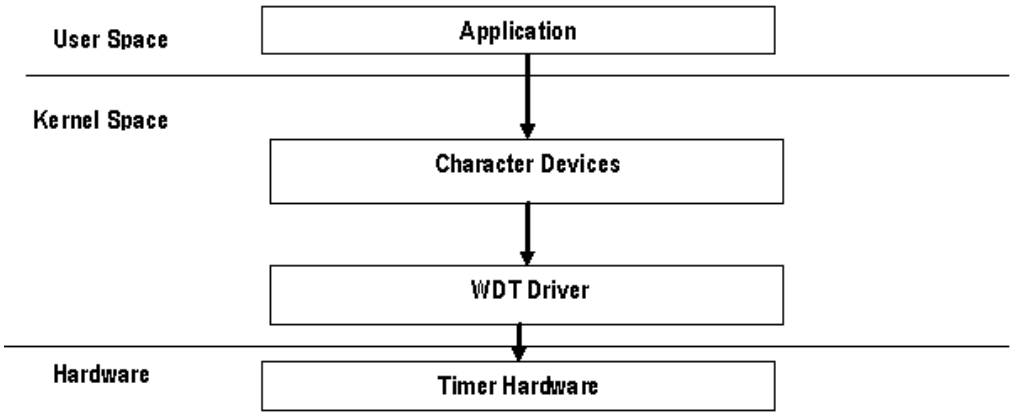
Usage	unsigned long tmp = 2;
	int ioctl(fd, RTC_IRQP_SET, tmp);
	value : {2, 4, 8, 16, 32,64}Hz

Function	**RTC_IRQP_READ**
Description	Read IRQ rate.
Usage	unsigned long tmp; int ioctl(fd, RTC_IRQP_READ, &tmp);
Function	**RTC_ALM_SET**
Description	Set alarm time.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_ALM_SET, &rtc_tm);
Function	**RTC_PIE_ON**
Description	Periodic int. enable on
Usage	int ioctl(fd, RTC_PIE_ON, 0);
Function	**RTC_PIE_OFF**
Description	Periodic int. enable off.
Usage	int ioctl(fd, RTC_PIE_OFF, 0);
Function	**RTC_UIE_ON**
Description	Update int. enable on.
Usage	int ioctl(fd, RTC_UIE_ON, 0);
Function	**RTC_UIE_OFF**
Description	Update int. enable off
Usage	int ioctl(fd, RTC_UIE_OFF, 0);
Function	**RTC_AIE_ON**
Description	Periodic int. enable on
Usage	int ioctl (fd, RTC_AIE_ON, 0);
Function	**RTC_AIE_OFF**
Description	Alarm int. enable off
Usage	int ioctl (fd, RTC_AIE_OFF, 0);

Refer to the example in example package on how to use these functions

WDT (Watch Dog Timer)

The WDT works like a watch dog function. You can enable it or disable it. When the user enables WDT but the application does not acknowledge it, the system will reboot. You can set the ack time from a minimum of 1 sec to a maximum of 1days. The default timer is 60seconds and the NO WAY OUT is enable in default, there is no way of disabling the watchdog once it has been started. So, if the watchdog daemon crashes, the system will reboot after the timeout has passed.



Function	**WDIOC_KEEPALIVE**
Description	Write to the watchdog device to keep watchdog alive.
Usage	int ioctl(fd, WDIOC_KEEPALIVE, 0)
Function	**WDIOC_SETTIMEOUT**
Description	Modify the watchdog timeout Min: 1second. Max: 1day Default: 60seconds
Usage	int timeout = 60; ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
Function	**WDIOC_GETTIMEOUT**
Description	Query the current timeout
Usage	int timeout; ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
Function	**WDIOC_GETSTATUS**
Description	Ask for the current status
Usage	int flags; ioctl(fd, WDIOC_GETSTATUS, &flags);
Function	**WDIOC_GETBOOTSTATUS**
Description	Ask for the status at the last reboot
Usage	int flags; ioctl(fd, WDIOC_GETBOOTSTATUS, &flags);
Function	**WDIOC_GETSUPPORT**
Description	Ask what the device can do
Usage	struct watchdog_info ident; ioctl(fd, WDIOC_GETSUPPORT, &ident);

Cryptographic Hardware Accelerator

The purpose of cryptographic hardware accelerator is to load off the intensive tasks of encryption/decryption and compression/decompression from CPU. You may take advantage of the cytpyographic hardware accelerator when your application needs to do cytpyographic calutations. To use it, you need to make sure that cryptodev driver is loaded.

Moxa provides examples to show how to use this crytpyographic accelerator. Go to *example/cryptodev/* folder and find out more.

NOTE Click the following link for more information on cryptodev.
http://cryptodev-linux.org/documentation.html/

Diagnostic LED

We provide diagnostic LEDs library which name is libmx_led.so to show the status of device, but we also provide diagnostic LED API to let your own application be able to use these LEDs

Turn on LEDs API

Return code: 0 for OK, else for error

Turn on GREEN LED	onoff_led ("GREEN", 1);
Turn on YELLOW LED	onoff_led ("YELLOW", 1);
Turn on RED LED	onoff_led ("RED", 1);
Turn on all LED	on_all_led();

Turn off LED API

Return code: 0 for OK, else for error

Turn off GREEN LED	onoff_led ("GREEN", 0);
Turn off YELLOW LED	onoff_led ("YELLOW", 0);
Turn off RED LED	onoff_led ("RED", 0);
Turn off ALL LED	off_all_led();

Blink LED API

Return code: 0 for OK, else for error

Blink GREEN LED	blink_led ("GREEN");
Blink YELLOW LED	blink_led ("YELLOW");
Blink RED LED	blink_led ("RED");
Blink all LED	blink_all_led();

ATTENTION

Be careful to use diagnostic LEDs

Do not use diagnotstic function while running your own application with diagnostic LED.

Example to set the baud rate

```
#include <termio.h>
#include <fcntl.h>
#include <err.h>
#include <linux/serial.h>
…
struct termios options;
        struct serial_struct serinfo;
int fd;
int speed = 0;
static int rate_to_constant(int baudrate) {
#define B(x) case x: return B##x
   switch(baudrate) {
      B(50);    B(75);    B(110);   B(134);   B(150);
      B(200);   B(300);   B(600);   B(1200);  B(1800);
      B(2400);  B(4800);  B(9600);  B(19200); B(38400);
      B(57600); B(115200);
   default: return 0;
   }
#undef B
}
…
/* Open and configure serial port */
      if ((fd = open(device,O_RDWR|O_NOCTTY)) == -1)
            return -1;

   fcntl(fd, F_SETFL, 0);
   tcgetattr(fd, &options);
   cfsetispeed(&options, speed ?: B115200);
   cfsetospeed(&options, speed ?: B115200);
   cfmakeraw(&options);
   options.c_cflag |= (CLOCAL | CREAD);
   options.c_cflag &= ~CRTSCTS;
   if (tcsetattr(fd, TCSANOW, &options) != 0)
return -1;
```

TPM

TCG Software Stack (TSS) API

TPM services provided through the TSS API are:

1. RSA key pair generation
2. RSA encryption and decryption using PKCS v1.5 and OAEP padding
3. RSA sign/verify
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

More detailed please refer to the example code or trousers web site.

A

Default Installed Package List

This appendix lists the default packages that have been installed on the UC-8100 series.

Name	Version	Architecture	Description
adduser	3.113+nmu3	all	add and remove users and groups
apache2	2.2.22-13+deb7u1	armhf	Apache HTTP Server metapackage
apache2-mpm-prefork	2.2.22-13+deb7u1	armhf	Apache HTTP Server - traditional non-threade
apache2-utils	2.2.22-13+deb7u1	armhf	utility programs for webservers
apache2.2-bin	2.2.22-13+deb7u1	armhf	Apache HTTP Server common binary files
apache2.2-common	2.2.22-13+deb7u1	armhf	Apache HTTP Server common files
apt	0.9.7.9+deb7u2	armhf	commandline package manager
apt-utils	0.9.7.9+deb7u2	armhf	package managment related utility programs
base-files	7.1wheezy5	armhf	Debian base system miscellaneous files
base-passwd	3.5.26	armhf	Debian base system master password and group
bash	4.2+dfsg-0.1	armhf	GNU Bourne Again SHell
bsdmainutils	9.0.3	armhf	collection of more utilities from FreeBSD
bsdutils	1:2.20.1-5.3	armhf	Basic utilities from 4.4BSD-Lite
bzip2	1.0.6-4	armhf	high-quality block-sorting file compressor -
coreutils	8.13-3.5	armhf	GNU core utilities
cpio	2.11+dfsg-0.1	armhf	GNU cpio -- a program to manage archives of
cron	3.0pl1-124	armhf	process scheduling daemon
dash	0.5.7-3	armhf	POSIX-compliant shell
dbus	1.6.8-1+deb7u1	armhf	simple interprocess messaging system (daemon
debconf	1.5.49	all	Debian configuration management system
debconf-i18n	1.5.49	all	full internationalization support for debcon
debian-archive-keyring	2012.4	all	GnuPG archive keys of the Debian archive
debianutils	4.3.2	armhf	Miscellaneous utilities specific to Debian
diffutils	1:3.2-6	armhf	File comparison utilities
dmsetup	2:1.02.74-8	armhf	Linux Kernel Device Mapper userspace library

Name	Version	Architecture	Description
dpkg	1.16.15	armhf	Debian package management system
e2fslibs:armhf	1.42.5-1.1	armhf	ext2/ext3/ext4 file system libraries
e2fsprogs	1.42.5-1.1	armhf	ext2/ext3/ext4 file system utilities
file	5.11-2+deb7u3	armhf	Determines file type using "magic" numbers
findutils	4.4.2-4	armhf	utilities for finding files--find, xargs
gawk	1:4.0.1+dfsg-2.1	armhf	GNU awk, a pattern scanning and processing l
gcc-4.7-base:armhf	4.7.2-5	armhf	GCC, the GNU Compiler Collection (base packa
gnupg	1.4.12-7+deb7u4	armhf	GNU privacy guard - a free PGP replacement
gpgv	1.4.12-7+deb7u4	armhf	GNU privacy guard - signature verification t
grep	2.12-2	armhf	GNU grep, egrep and fgrep
groff-base	1.21-9	armhf	GNU troff text-formatting system (base syste
gzip	1.5-1.1	armhf	GNU compression utilities
hostname	3.11	armhf	utility to set/show the host name or domain
ifupdown	0.7.8	armhf	high level tools to configure network interf
initscripts	2.88dsf-41+deb7u1	armhf	scripts for initializing and shutting down t
insserv	1.14.0-5	armhf	boot sequence organizer using LSB init.d scr
iproute	20120521-3+b3	armhf	networking and traffic control tools
iptables	1.4.14-3.1	armhf	administration tools for packet filtering an
iputils-ping	3:20101006-1+b2	armhf	Tools to test the reachability of network ho
isc-dhcp-client	4.2.2.dfsg.1-5+deb70u6	armhf	ISC DHCP client
isc-dhcp-common	4.2.2.dfsg.1-5+deb70u6	armhf	common files used by all the isc-dhcp* packa
kmod	9 月 3 日	armhf	tools for managing Linux kernel modules
krb5-locales	1.10.1+dfsg-5+deb7u1	all	Internationalization support for MIT Kerbero
libacl1:armhf	2.2.51-8	armhf	Access control list shared library
libapache2-mod-php5	5.4.4-14+deb7u11	armhf	server-side, HTML-embedded scripting languag
libapr1	1.4.6-3+deb7u1	armhf	Apache Portable Runtime Library
libaprutil1	1.4.1-3	armhf	Apache Portable Runtime Utility Library
libaprutil1-dbd-sqlite3	1.4.1-3	armhf	Apache Portable Runtime Utility Library - SQ
libaprutil1-ldap	1.4.1-3	armhf	Apache Portable Runtime Utility Library - LD

Name	Version	Architecture	Description
libapt-inst1.5:armhf	0.9.7.9+deb7u2	armhf	deb package format runtime library
libapt-pkg4.12:armhf	0.9.7.9+deb7u2	armhf	package managment runtime library
libattr1:armhf	1:2.4.46-8	armhf	Extended attribute shared library
libblkid1:armhf	2.20.1-5.3	armhf	block device id library
libbsd0:armhf	0.4.2-1	armhf	utility functions from BSD systems - shared
libbz2-1.0:armhf	1.0.6-4	armhf	high-quality block-sorting file compressor l
libc-bin	2.13-38+deb7u1	armhf	Embedded GNU C Library: Binaries
libc6:armhf	2.13-38+deb7u1	armhf	Embedded GNU C Library: Shared libraries
libcap2:armhf	1:2.22-1.2	armhf	support for getting/setting POSIX.1e capabil
libclass-isa-perl	0.36-3	all	report the search path for a class's ISA tre
libcomerr2:armhf	1.42.5-1.1	armhf	common error description library
libdb5.1:armhf	5.1.29-5	armhf	Berkeley v5.1 Database Libraries [runtime]
libdbus-1-3:armhf	1.6.8-1+deb7u1	armhf	simple interprocess messaging system (librar
libdevmapper1.02.1:armhf	2:1.02.74-8	armhf	Linux Kernel Device Mapper userspace library
libedit2:armhf	2.11-20080614-5	armhf	BSD editline and history libraries
libevent-2.0-5:armhf	2.0.19-stable-3	armhf	Asynchronous event notification library
libexpat1:armhf	2.1.0-1+deb7u1	armhf	XML parsing C library - runtime library
libffi5:armhf	3.0.10-3+b1	armhf	Foreign Function Interface library runtime
libfribidi0:armhf	0.19.2-3	armhf	Free Implementation of the Unicode BiDi algo
libgcc1:armhf	1:4.7.2-5	armhf	GCC support library
libgcrypt11:armhf	1.5.0-5+deb7u1	armhf	LGPL Crypto library - runtime library
libgdbm3:armhf	1.8.3-11	armhf	GNU dbm database routines (runtime version)
libglib2.0-0:armhf	2.33.12+really2.32.4-5	armhf	GLib library of C routines
libglib2.0-data	2.33.12+really2.32.4-5	all	Common files for GLib library
libgnutls26:armhf	2.12.20-8+deb7u2	armhf	GNU TLS library - runtime library
libgpg-error0:armhf	1.10-3.1	armhf	library for common error values and messages
libgpm2:armhf	1.20.4-6	armhf	General Purpose Mouse - shared library
libgssapi-krb5-2:armhf	1.10.1+dfsg-5+deb7u1	armhf	MIT Kerberos runtime libraries - krb5 GSS-AP
libgssglue1:armhf	0.4-2	armhf	mechanism-switch gssapi library
libidn11:armhf	1.25-2	armhf	GNU Libidn library, implementation of IETF I
libiw30:armhf	30~pre9-8	armhf	Wireless tools - library

Name	Version	Architecture	Description
libk5crypto3:armhf	1.10.1+dfsg-5+deb7u1	armhf	MIT Kerberos runtime libraries - Crypto Libr
libkeyutils1:armhf	1.5.5-3	armhf	Linux Key Management Utilities (library)
libkmod2:armhf	9月3日	armhf	libkmod shared library
libkrb5-3:armhf	1.10.1+dfsg-5+deb7u1	armhf	MIT Kerberos runtime libraries
libkrb5support0:armhf	1.10.1+dfsg-5+deb7u1	armhf	MIT Kerberos runtime libraries - Support lib
libldap-2.4-2:armhf	2.4.31-1+nmu2	armhf	OpenLDAP libraries
liblocale-gettext-perl	1.05-7+b3	armhf	module using libc functions for internationa
liblockfile-bin	1.09-5	armhf	support binaries for and cli utilities based
liblockfile1:armhf	1.09-5	armhf	NFS-safe locking library
liblzma5:armhf	5.1.1alpha+20120614-2	armhf	XZ-format compression library
liblzo2-2:armhf	2.06-1	armhf	data compression library
libmagic1:armhf	5.11-2+deb7u3	armhf	File type determination library using "magic
libmodbus-dev	3.0.3-1	armhf	development files for the Modbus protocol li
libmodbus5	3.0.3-1	armhf	library for the Modbus protocol
libmount1	2.20.1-5.3	armhf	block device id library
libmysqlclient18:armhf	5.5.37-0+wheezy1	armhf	MySQL database client library
libncurses5:armhf	5.9-10	armhf	shared libraries for terminal handling
libncursesw5:armhf	5.9-10	armhf	shared libraries for terminal handling (wide
libnewt0.52	0.52.14-11.1	armhf	Not Erik's Windowing Toolkit - text mode win
libnfnetlink0	1.0.0-1.1	armhf	Netfilter netlink library
libnfsidmap2:armhf	0.25-4	armhf	NFS idmapping library
libnl-3-200:armhf	3.2.7-4	armhf	library for dealing with netlink sockets
libnl-genl-3-200:armhf	3.2.7-4	armhf	library for dealing with netlink sockets - g
libonig2	5.9.1-1	armhf	Oniguruma regular expressions library
libopencryptoki0	2.3.1+dfsg-3	armhf	PKCS#11 implementation (library)
libp11-kit0:armhf	0.12-3	armhf	Library for loading and coordinating access
libpam-modules:armhf	1.1.3-7.1	armhf	Pluggable Authentication Modules for PAM
libpam-modules-bin	1.1.3-7.1	armhf	Pluggable Authentication Modules for PAM - h
libpam-runtime	1.1.3-7.1	all	Runtime support for the PAM library
libpam0g:armhf	1.1.3-7.1	armhf	Pluggable Authentication Modules library
libparted0debian1:armhf	2.3-12	armhf	disk partition manipulator - shared library

Name	Version	Architecture	Description
libpcap0.8:armhf	1.3.0-1	armhf	system interface for user-level packet captu
libpcre3:armhf	1:8.30-5	armhf	Perl 5 Compatible Regular Expression Library
libpcsclite1:armhf	1.8.4-1+deb7u1	armhf	Middleware to access a smart card using PC/S
libperl5.14	5.14.2-21+deb7u1	armhf	shared Perl library
libpipeline1:armhf	1.2.1-1	armhf	pipeline manipulation library
libpkcs11-helper1:armhf	1.09-1	armhf	library that simplifies the interaction with
libpopt0:armhf	1.16-7	armhf	lib for parsing cmdline parameters
libprocps0:armhf	1:3.3.3-3	armhf	library for accessing process information fr
libqdbm14	1.8.78-2	armhf	QDBM Database Libraries without GDBM wrapper
libreadline5:armhf	5.2+dfsg-2~deb7u1	armhf	GNU readline and history libraries, run-time
libreadline6:armhf	6.2+dfsg-0.1	armhf	GNU readline and history libraries, run-time
libsasl2-2:armhf	2.1.25.dfsg1-6+deb7u1	armhf	Cyrus SASL - authentication abstraction libr
libsasl2-modules:armhf	2.1.25.dfsg1-6+deb7u1	armhf	Cyrus SASL - pluggable authentication module
libselinux1:armhf	2.1.9-5	armhf	SELinux runtime shared libraries
libsemanage-common	2.1.6-6	all	Common files for SELinux policy management l
libsemanage1:armhf	2.1.6-6	armhf	SELinux policy management library
libsensors4:armhf	1:3.3.2-2+deb7u1	armhf	library to read temperature/voltage/fan sens
libsepol1:armhf	2.1.4-3	armhf	SELinux library for manipulating binary secu
libsigc++-2.0-0c2a:armhf	2.2.10-0.2	armhf	type-safe Signal Framework for C++ - runtime
libsigsegv2	2.9-4+b1	armhf	Library for handling page faults in a portab
libslang2:armhf	2.2.4-15	armhf	S-Lang programming library - runtime version
libsnmp-base	5.4.3~dfsg-2.8	all	SNMP (Simple Network Management Protocol) MI
libsnmp15	5.4.3~dfsg-2.8	armhf	SNMP (Simple Network Management Protocol) li
libsqlite3-0:armhf	3.7.13-1+deb7u1	armhf	SQLite 3 shared library
libss2:armhf	1.42.5-1.1	armhf	command-line interface parsing library
libssl1.0.0:armhf	1.0.1e-2+deb7u11+uc8100	armhf	SSL shared libraries
libstdc++6:armhf	4.7.2-5	armhf	GNU Standard C++ Library v3
libswitch-perl	2.16-2	all	switch statement for Perl
libsystemd-login0:armhf	44-11+deb7u4	armhf	systemd login utility library
libtasn1-3:armhf	2.13-2	armhf	Manage ASN.1 structures (runtime)

Name	Version	Architecture	Description
libtext-charwidth-perl	0.04-7+b2	armhf	get display widths of characters on the term
libtext-iconv-perl	1.7-5	armhf	converts between character sets in Perl
libtext-wrapi18n-perl	0.06-7	all	internationalized substitute of Text::Wrap
libtinfo5:armhf	5.9-10	armhf	shared low-level terminfo library for termin
libtirpc1:armhf	0.2.2-5	armhf	transport-independent RPC library
libtpm-unseal1	1.3.7-1	armhf	Management tools for the TPM hardware (libra
libtspi1	0.3.9-3+wheezy1	armhf	open-source TCG Software Stack (library)
libudev0:armhf	175-7.2	armhf	libudev shared library
libuniconf4.6	4.6.1-5	armhf	C++ network libraries for rapid application
libusb-0.1-4:armhf	2:0.1.12-20+nmu1	armhf	userspace USB programming library
libustr-1.0-1:armhf	1.0.4-3	armhf	Micro string library: shared library
libuuid1:armhf	2.20.1-5.3	armhf	Universally Unique ID library
libwrap0:armhf	7.6.q-24	armhf	Wietse Venema's TCP wrappers library
libwvstreams4.6-base	4.6.1-5	armhf	C++ network libraries for rapid application
libwvstreams4.6-extras	4.6.1-5	armhf	C++ network libraries for rapid application
libxml2:armhf	2.8.0+dfsg1-7+nmu3	armhf	GNOME XML library
locales	2.13-38+deb7u1	all	Embedded GNU C Library: National Language (l
lockfile-progs	0.1.17	armhf	Programs for locking and unlocking files and
login	1:4.1.5.1-1	armhf	system login tools
logrotate	3.8.1-4	armhf	Log rotation utility
lsb-base	4.1+Debian8+deb7u1	all	Linux Standard Base 4.1 init script function
man-db	2.6.2-1	armhf	on-line manual pager
mawk	1.3.3-17	armhf	a pattern scanning and text processing langu
mime-support	3.52-1	all	MIME files 'mime.types' & 'mailcap', and sup
module-assistant	0.11.4	all	tool to make module package creation easier
mount	2.20.1-5.3	armhf	Tools for mounting and manipulating filesyst
multiarch-support	2.13-38+deb7u1	armhf	Transitional package to ensure multiarch com
mysql-common	5.5.37-0+wheezy1	all	MySQL database common files, e.g. /etc/mysql
nano	2.2.6-1+b2	armhf	small, friendly text editor inspired by Pico

Name	Version	Architecture	Description
ncurses-base	5.9-10	all	basic terminal type definitions
ncurses-bin	5.9-10	armhf	terminal-related programs and man pages
ncurses-term	5.9-10	all	additional terminal type definitions
net-tools	1.60-24.2	armhf	The NET-3 networking toolkit
netbase	5	all	Basic TCP/IP networking system
netcat-traditional	1.10-40	armhf	TCP/IP swiss army knife
nfs-common	1:1.2.6-4	armhf	NFS support files common to client and serve
nfs-kernel-server	1:1.2.6-4	armhf	support for NFS kernel server
ntpdate	1:4.2.6.p5+dfsg-2	armhf	client for setting system time from NTP serv
opencryptoki	2.3.1+dfsg-3	armhf	PKCS#11 implementation (daemon)
openssh-blacklist	0.4.1+nmu1	all	list of default blacklisted OpenSSH RSA and
openssh-blacklist-extra	0.4.1+nmu1	all	list of non-default blacklisted OpenSSH RSA
openssh-client	1:6.0p1-4+deb7u1	armhf	secure shell (SSH) client, for secure access
openssh-server	1:6.0p1-4+deb7u1	armhf	secure shell (SSH) server, for secure access
openssl	1.0.1e-2+deb7u11+uc8100	armhf	Secure Socket Layer (SSL) binary and related
openvpn	2.2.1-8+deb7u2	armhf	virtual private network daemon
parted	2.3-12	armhf	disk partition manipulator
passwd	1:4.1.5.1-1	armhf	change and administer password and group dat
perl	5.14.2-21+deb7u1	armhf	Larry Wall's Practical Extraction and Report
perl-base	5.14.2-21+deb7u1	armhf	minimal Perl system
perl-modules	5.14.2-21+deb7u1	all	Core Perl modules
php5	5.4.4-14+deb7u11	all	server-side, HTML-embedded scripting languag
php5-cli	5.4.4-14+deb7u11	armhf	command-line interpreter for the php5 script
php5-common	5.4.4-14+deb7u11	armhf	Common files for packages built from the php
php5-mysql	5.4.4-14+deb7u11	armhf	MySQL module for php5
pkg-config	0.26-1+b1	armhf	manage compile and link flags for libraries
pmount	0.9.23-2	armhf	mount removable devices as normal user
ppp	2.4.5-5.1+b1	armhf	Point-to-Point Protocol (PPP) - daemon
procps	1:3.3.3-3	armhf	/proc file system utilities
psmisc	22.19-1+deb7u1	armhf	utilities that use the proc file system
python	2.7.3-4+deb7u1	all	interactive high-level object-oriented langu

Name	Version	Architecture	Description
python-minimal	2.7.3-4+deb7u1	all	minimal subset of the Python language (defau
python2.7	2.7.3-6+deb7u2	armhf	Interactive high-level object-oriented langu
python2.7-minimal	2.7.3-6+deb7u2	armhf	Minimal subset of the Python language (versi
readline-common	6.2+dfsg-0.1	all	GNU readline and history libraries, common f
rpcbind	0.2.0-8	armhf	converts RPC program numbers into universal
rsync	3.0.9-4	armhf	fast, versatile, remote (and local) file-cop
rsyslog	5.8.11-3	armhf	reliable system and kernel logging daemon
sed	4.2.1-10	armhf	The GNU sed stream editor
sensible-utils	0.0.7	all	Utilities for sensible alternative selection
sgml-base	1.26+nmu4	all	SGML infrastructure and SGML catalog file su
snmp	5.4.3~dfsg-2.8	armhf	SNMP (Simple Network Management Protocol) ap
ssh	1:6.0p1-4+deb7u1	all	secure shell client and server (metapackage)
sudo	1.8.5p2-1+nmu1	armhf	Provide limited super user privileges to spe
sysv-rc	2.88dsf-41+deb7u1	all	System-V-like runlevel change mechanism
sysvinit	2.88dsf-41+deb7u1	armhf	System-V-like init utilities
sysvinit-utils	2.88dsf-41+deb7u1	armhf	System-V-like utilities
tar	1.26+dfsg-0.1	armhf	GNU version of the tar archiving utility
tcpd	7.6.q-24	armhf	Wietse Venema's TCP wrapper utilities
tpm-tools	1.3.7-1	armhf	Management tools for the TPM hardware (tools
traceroute	1:2.0.18-3	armhf	Traces the route taken by packets over an IP
trousers	0.3.9-3+wheezy1	armhf	open-source TCG Software Stack (daemon)
tzdata	2014e-0wheezy1	all	time zone and daylight-saving time data
uc8100-cellular-utils	1.0.0	armhf	Cellular driver and related utility on uc810
uc8100-diag	1.0.0	armhf	Self-diagnostic utility on uc8100 series.
uc8100-push-btn	1.0.0	armhf	Push button utility on uc8100 series.
uc8100-setdef	1.0.0	all	Set-to-default utility on uc8100 series.
uc8100-setinterface	1.0.0	all	Adjust UART mode utility on uc8100 series.
uc8100-snmpd	5.4.3~dfsg-2.7	armhf	SNMP (Simple Network Management Protocol) ag

Name	Version	Architecture	Description
uc8100-system	1.0.0	armhf	System files in uc8100
uc8100-wifi-utils	1.0.0	armhf	WiFi utils on uc8100 series.
ucf	3.0025+nmu3	all	Update Configuration File: preserve user cha
udev	175-7.2	armhf	/dev/ and hotplug management daemon
util-linux	2.20.1-5.3	armhf	Miscellaneous system utilities
vim	2:7.3.547-7	armhf	Vi IMproved - enhanced vi editor
vim-common	2:7.3.547-7	armhf	Vi IMproved - Common files
vim-runtime	2:7.3.547-7	all	Vi IMproved - Runtime files
watchdog	5.12-1	armhf	system health checker and software/hardware
wget	1.13.4-3+deb7u1	armhf	retrieves files from the web
whiptail	0.52.14-11.1	armhf	Displays user-friendly dialog boxes from she
wireless-tools	30~pre9-8	armhf	Tools for manipulating Linux Wireless Extens
wpasupplicant	1.0-3+b1	armhf	client support for WPA and WPA2 (IEEE 802.11
wvdial	1.61-4.1	armhf	intelligent Point-to-Point Protocol dialer
xml-core	0.13+nmu2	all	XML infrastructure and XML catalog file supp
xz-utils	5.1.1alpha+20120614-2	armhf	XZ-format compression utilities
zlib1g:armhf	1:1.2.7.dfsg-13	armhf	compression library - runtime

B

Extending the Lifetime of the SD Card

This appendix describes how to extend the lifetime of the SD card.

The following topics are covered in this appendix:

- **Overview**
  - SD Flash Types
- **Tips for Running GNU/Linux on an SD Card**
  - Use SLC SD Card
  - Use an SD Card with Larger Capacity
  - Tweak GNU/Linux to Write to RAM Instead of the SD card.
  - Set the SD Card to Read-only Mode

Overview

The UC-8100-LX comes with an SD socket that can provide storage expansion, and you can even store the operating system in the SD card. Choosing an ideal SD card for the UC-8100-LX has become crucial. Here are some general ideas on the SD cards that can be purchased in the current market.

SD Flash Types

Single-level-cell (SLC)

Single-level-cell (SLC) has the simplest operation of all the flash type: there is only one bit per cell, and the firmware does not need to negotiate with the datas in different levels and states. SLC can provide longer lifetime than other flash types.

Multi-level cell (MLC)

Multi-level cell (MLC), as its name suggests, can store multiple bits per cell. The primary benefit of MLC flash memory is its lower cost per unit of storage due to the higher data density.

Triple-level cell (TLC)

TLC flash (triple level cell flash) is a type of solid-state NAND flash memory that stores three bits of data per cell. TLC flash is less expensive than single-level cell (SLC) and multi-level cell (MLC) solid-state flash memory, and is commonly used in various cosumer devices that use solid-state storage.

Comparison Table for Flash Types

Flash type	SLC, Single Level Cell (1 bit)	MLC, Multilevel Cell (2 bits)	TLC, Triple Level Cell (3 bits)
Bits per cell	1	2	3
Program/Erase cycles	Generally 100000 write/erase cycles	Anywhere from 3000 to 15000 write/erase cycles	Anywhere from 1000 to 5000 write/erase cycles
Erase time	Erase time: 1.5-2 ms	Erase time: 2.5-3.5ms	Erase time: 4-5ms
Operation scenario	Industrial	Commercial	Commercial

Check again the type of the SD card before deploying on the UC-8100-LX. The SD cards in the current market usually use TLC as their flash type due to price issue. We strongly recommend you use the SLC SD card in the UC-8100-LX computer.

Tips for Running GNU/Linux on an SD Card

Use SLC SD Card

We strongly recommend you use the SLC flash type SD card for the UC-8100-LX computer, as it provides longer lifetime cycle.

Use an SD Card with Larger Capacity

Using an SD card with larger capacity can provide more space for writing and reading data, so the larger the card, the less chance of writing over the same area for multiple times. Most GNU/Linux distributions on the UC-8100-LX can fit on a 4 GB card, but 8 GB and even 16 GB cards are more advisable.

Tweak GNU/Linux to Write to RAM Instead of the SD card.

This uses a feature called "tmpfs", a usedful function of GNU/Linux. Tmpfs can write to RAM as if it was an ordinary filesystem. It's fast, efficient, and easy to use.

tmpfs can write to RAM instead of the local disk (in this case, the SD card). All that needs to be done is add an entry to the /etc/fstab file (to mount the folder you wish to have written to RAM) and reboot (so that each mount is cleanly mounted before services start writing files).

The kernel will do the rest for you by managing the writes to the RAM on this virtual filesystem. The really neat part about this is that the kernel will only use the amount of RAM required for writing files, not the entire size of the mount. So, for example, say we add this line to the /etc/fstab file:

```
tmpfs   /var/log    tmpfs    defaults,noatime,nosuid,mode=0755,size=100m    0 0
```

The kernel will mount /var/log to RAM, however it will not use any RAM until files are written to /var/log. When files are written to /var/log, the kernel will save them to RAM and only use space to save the files. When files are removed from /var/log, the associated RAM to store them is freed up.

This means that it only uses what RAM it needs to in order to store the files, which makes it very efficient.

In /etc/fstab, you can also specify the total size to allocate for each mount. In the example above, we set "size=100m" so that /var/log can use up to 100 MB of space and no more. This avoids a filesystem from using up all of the RAM which can cause the system to slow down or even crash. By running the "mount" command, we can see in the example above that /var/log is mounted as a tmpfs volume to RAM, 100 MB in size.

```
Filesystem          Size  Used Avail Use% Mounted on

tmpfs               100M  596K  100M   1% /var/log
```

There are a variety of locations that GNU/Linux likes to make frequent writes. This is a list of entries below that I use as a starting point that should fit for most distributions.

```
tmpfs    /tmp    tmpfs    defaults,noatime,nosuid,size=100m    0 0

tmpfs    /var/tmp    tmpfs    defaults,noatime,nosuid,size=30m    0 0

tmpfs    /var/log    tmpfs    defaults,noatime,nosuid,mode=0755,size=100m    0 0

tmpfs    /var/run    tmpfs    defaults,noatime,nosuid,mode=0755,size=2m    0 0

tmpfs    /var/spool/mqueue    tmpfs    defaults,noatime,nosuid,mode=0700,gid=12,size=30m   0 0
```

Use "size=" parameter to avoid using up huge amounts of RAM in case something tries to save a huge amount of data. The "noatime" and "nosuid" parameters are also recommended for security and performance, and "mode=" along with "gid=" matches the permissions and group of the original filesystem to what was located on the SD card originally.

In addition, tmpfs can also handle permissions. As usual, entries in /etc/fstab mount over the top of what is on the SD card, as standard Unix/Linux types do. So if for some reason the mounts fail, writes will still work to the SD card.

One additional point to keep in mind is that anything mounted with tmpfs will be lost on a reboot. So, logs in /var/log in the example above will be wiped out if the computer is shut down or rebooted. So you will not want to save any files with tmpfs that need to be persistent among reboots.

Set the SD Card to Read-only Mode

This essentially makes GNU/Linux run in read-only mode, similar to how it works booting from a Live CD. This avoids any writing to the SD card and in theory can extend its life. There are some drawbacks to this though. First, it takes a bit of work to set up, which is out of the scope of this article. Second, changes that are made will be lost when the system is rebooted because they are not written to the SD card. To me, running GNU/Linux in read-only mode is overkill and I don't recommend going to this extreme.

NOTE Click the following links for more information on minicom.
http://www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html

C

Copying Images on an SD/MicroSD Card

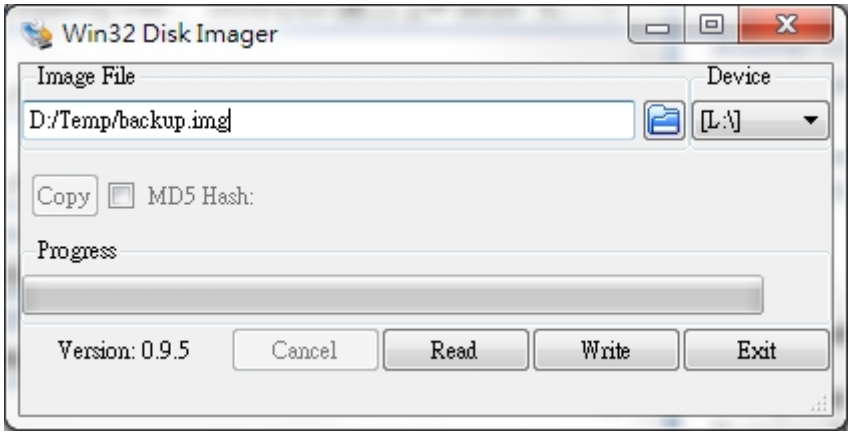
This appendix shows you how to copy an image from an SD or MicroSD card.

Using Win32 Disk Imager

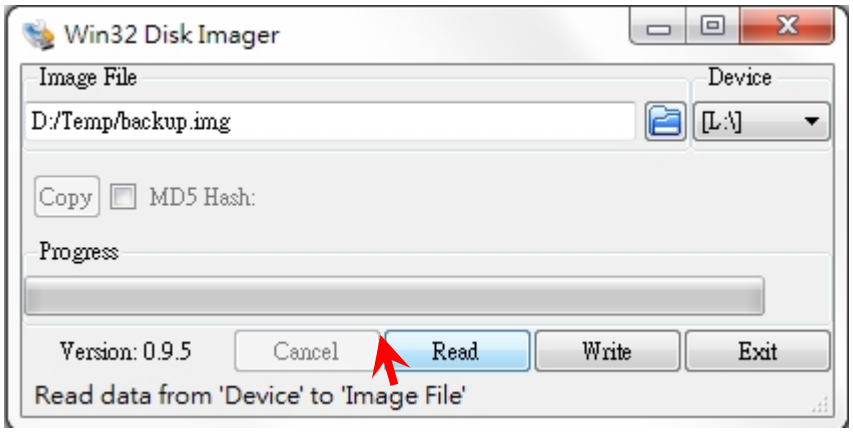
1. Remove the SD or microSD card from the UC-8100 and insert it into another computer.
2. Start Win32 Disk imager and complete the following steps:

   a. From the **Device** drop-down list, select the drive letter for the SD card (for example, L:\).

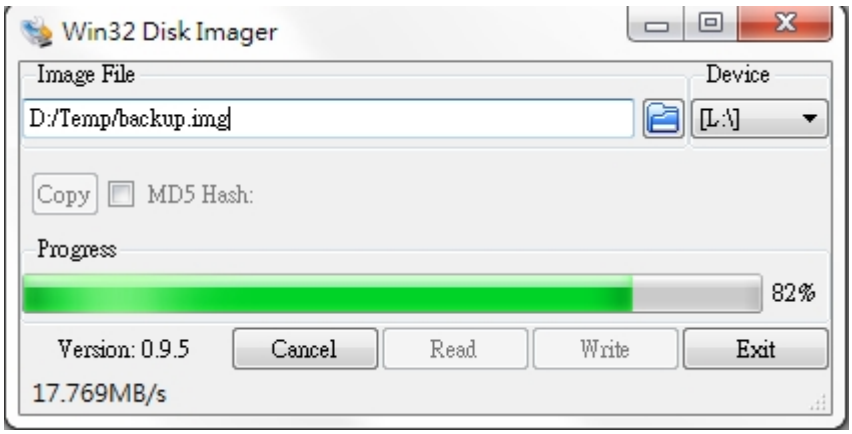
   b. In the **Image File** field, enter the location for the image file. Althernatively, you can click the folder icon to locate the image file (for example, D:\Temp\backup.imp).

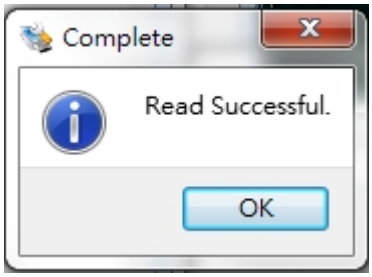


3. Click **Read**.



4. Wait until the file copy process is complete. Click **OK** to close the window.





5. Click **Exit** to exit the utility.

6. Eject the SD card from the computer. You can start using the image.

Using the dd command

1. Insert the SD or microSD card into another computer.
2. Check the device folder for the SD or microSD card (for example, /dev/sdd) and the directory and file name of the image file (for example, /home/backup.img).
3. Run the `dd` command. The following shows an example.

```
#dd if=/dev/sdd of=/home/backup.img bs=512k
```