UC-8540/8580 Series Software User's Manual (Debian 8)

Version 1.0, October 2021

www.moxa.com/product

© 2021 Moxa Inc. All rights reserved.

UC-8540/8580 Series Software User's Manual (Debian 8)

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

Copyright Notice

© 2021 Moxa Inc. All rights reserved.

Trademarks

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

Disclaimer

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

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

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

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

Technical Support Contact Information

www.moxa.com/support

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

Moxa 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

Thank you for purchasing Moxa’s UC-8540/8580 Series Arm-based computer. This is the software operation and programming manual for the Linux model of the UC-8540/8580 computer and covers the use of Linux functions with examples on how to program the UC-8540. In addition, detailed description of the various basic and advanced functions of the Mobile Intelligent Routing Framework (MIRF) 2.0 tool are provided for use in rail applications.

Linux is an open source, scalable operating system that helps you build a wide range of innovative, small footprint devices. Software written for desktop PCs can be easily ported to the embedded computer with a GNU cross compiler and minimum source code modifications. A typical Linux-based device is designed for a specific use and is often not connected to other computers. In some cases, several such devices could be connected to a centralized, front-end host. Examples include enterprise tools such as industrial controllers and communications hubs.

Moxa's MIRF is an open-platform, multiple-WAN management tool that helps provide unbeatable wireless service for train passengers as the train travels through different regions.

The wireless-enablement of the UC-8540/8580 computer also makes it the most suitable choice for Industrial IoT applications.

2

Getting Started

In this chapter, we describe how to configure the basic settings in your UC-8540/8580 computer.

The following topics are covered in this chapter:

- **Software Architecture**
- **Software Packages**
- **Connecting to the UC-8540/8580 Computer**
  - Using the Serial Console
  - Using the SSH Console
- **Sudo Mechanism**
- **Booting Up the UC-8540 for the First Time**
- **User Account Management**
  - Switching to the Root Account
- **Creating and Deleting User Accounts**
- **Disabling the Default User Account**
- **Network Settings**
  - Configuring Ethernet Interfaces
- **System Administration**
  - Querying the Firmware Version
  - Adjusting the Time
  - Setting the Time Zone
- **Determining Available Drive Space**
- **Enabling and Disabling Daemons**
- **Package Management**
- **Rebooting/Shutting Down the Computer**
- **Updating the Firmware Using a USB Disk**

Software Architecture

The Linux operating system that is preinstalled on the UC-8540/8580 computer follows standard Linux architecture, making it easy to run any program that follows the POSIX standard. This computer uses the Debian ARM 8 so that users can enjoy the full range of Debian software, and benefit from its strong community of developers and shared documentation. With Debian ARM, the UC-8540/8580 computer supports both native and cross compilation, making programming on the computer easier and more straightforward.

The UC-8540/8580 computer image is partitioned into Linux kernel, backup root file system, and root file system. Refer to the following image partition table for details:

Partition	System Content	Partition Format	Partition Size
1	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 the UC-8540/8580 computer is EXT4, which is a journaling file system for Linux, developed as the successor to EXT3. A 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 at bringing back the computer online and less likely to get 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

Most of the software packages come from the Debian community, whereas the unique features of the UC-8540/8580 computer, such as the cellular and wireless connections, are supported by Moxa. Refer to *Appendix A* for software packages installed by default and the *Package Management* section for information on managing the software packages installed on your UC-8540/8580 computer.

Connecting to the UC-8540/8580 Computer

You will need access to a notebook computer or a PC to connect to the UC-8540/8580 computer and log on to the command line interface. There are two ways to connect to the UC-8540/8580 computer: through a serial console cable or through an Ethernet cable. Refer to the *UC-8540/8580 Series Hardware User's Manual* for instructions to set up the physical connections for your computer.

The default login username and password are:

Username: moxa
Password: moxa

The username and password are the same for all serial console and SSH remote log in actions. The **root** account login is disabled until you manually create a password for the account. The user **moxa** is in the *sudo group*, which means that this user can use the **sudo** command to run system-level commands. Additional details on using the **sudo** command are available in the *Sudo Mechanism* section.

ATTENTION

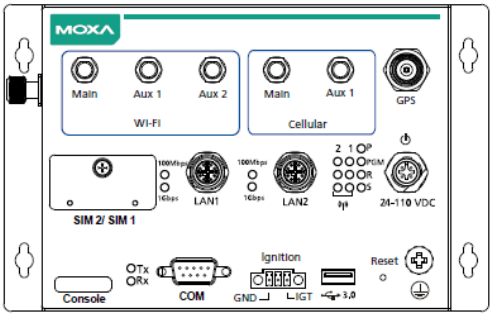
For security reasons, we recommend that you disable the default user account after the initial set up is complete and create new user accounts as per your requirement.
If you want to change the settings manually via the SSH console or serial console, turn off MIRF 2.0. If not, the settings may be restored to MIRF 2.0 settings.
The commands to change MIRF 2.0 state are as follows:
turn on: **`mx-tp-ctl -e 1`**
turn off: **`mx-tp-ctl -e 0`**

Using the Serial Console

This method is particularly useful when you are using the UC-8540/8580 computer for the first time. The signal is transmitted over a direct serial connection without using an IP address.

To connect to the UC-8540/8580 computer using a serial console port, do the following:

1. Open the console port cover on the front panel and connect one end of the cable to it



2. Connect the other end of the serial console cable to your PC.
3. Configure your PC's terminal software with the following settings:

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

The procedure to use the terminal software to connect to the UC-8540/8580 computer in a Linux and Windows environments is described in the following two sections:

Linux Users

WARNING

DO NOT apply these steps to the UC-8540/8580 computer. These steps apply only to the Linux PC that you use to connect to the UC-8540/8580 computer.

Take the following steps to connect to the UC-8540/8580 computer from your Linux PC.

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

   For Centos and Fedora, run the following command:

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

   For Ubuntu and Debian, run the following command:

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

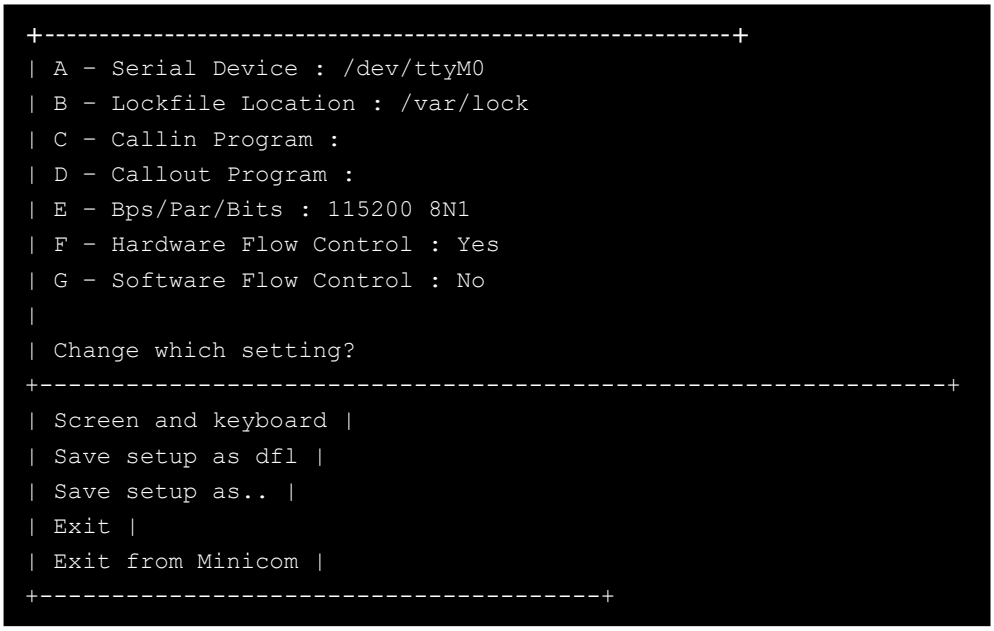
2. Use the **minicom –s** command to enter the configuration menu and set up the serial port settings.

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

3. Select **Serial port setup**.

4. Select **A** to change the serial device setting.

NOTE You need to know which device node is connected to the UC-8540/8580 computer to configure this setting.

5. Select **E** to configure the port settings according to the **Serial Console Port Settings** table provided above.
6. Select **Save setup as dfl** (from the main configuration menu) to use default values.
7. Select **Exit from minicom** (from the configuration menu) to leave the configuration menu.
8. Run the **minicom** tool after completing the abovementioned configuration settings.

   ```
   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
   ```

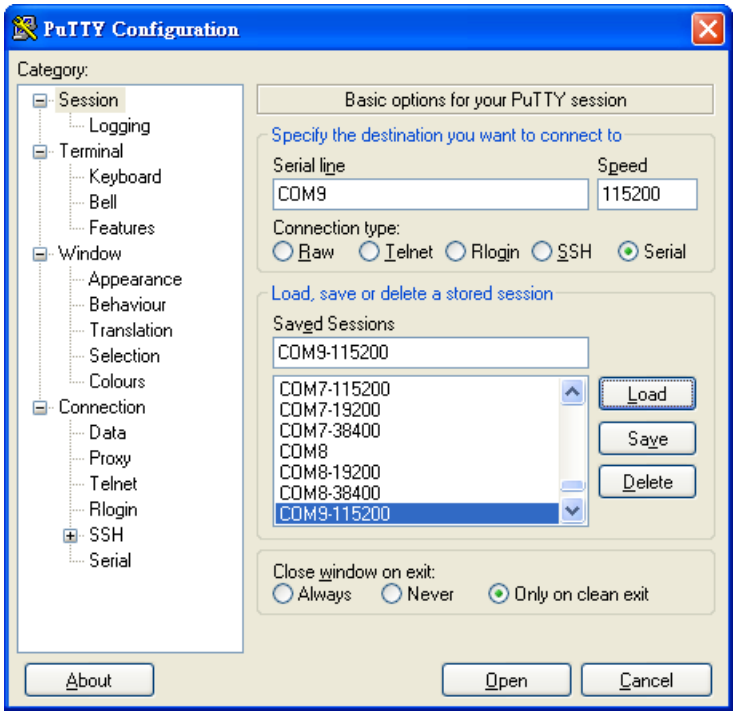
Windows Users

WARNING

DO NOT apply these steps to the UC-8540/8580 computer. These steps are for the Windows PC that you use to connect to the UC-8540/8580 computer.

Take the following steps to connect to the UC-8540/8580 computer from your Windows PC:

1. Download **PuTTY** (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html), the free SSH and telnet client for Windows.
2. Run the PuTTY application (**`putty.exe`**) on the Windows PC.
3. Enter the details of the serial connection in the configuration window.
   The figure below shows an example of the configuration that is required:



4. Click **Open**.
5. Type in the **`username`** and **`password`** in the console that opens to establish a serial connection with the UC-8540/8580 computer.

Using the SSH Console

The UC-8540/8580 computer supports SSH connections over an Ethernet network. Use the following default IP addresses to connect to the UC-8540/8580 computer:

Port	Default IP
LAN 1	192.168.3.127
LAN 2	192.168.4.127

Linux Users

NOTE Do NOT apply these steps to the UC-8540/8580 computer. The instructions in this section are for the Linux PC that you use to connect to the UC-8540/8580 computer.

Use the **ssh** command to access the UC-8540/8580 computer's LAN1 port from a Linux computer.

```
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

To secure your system, we suggest doing a regular SSH-rekey as shown in 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 as follows:

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

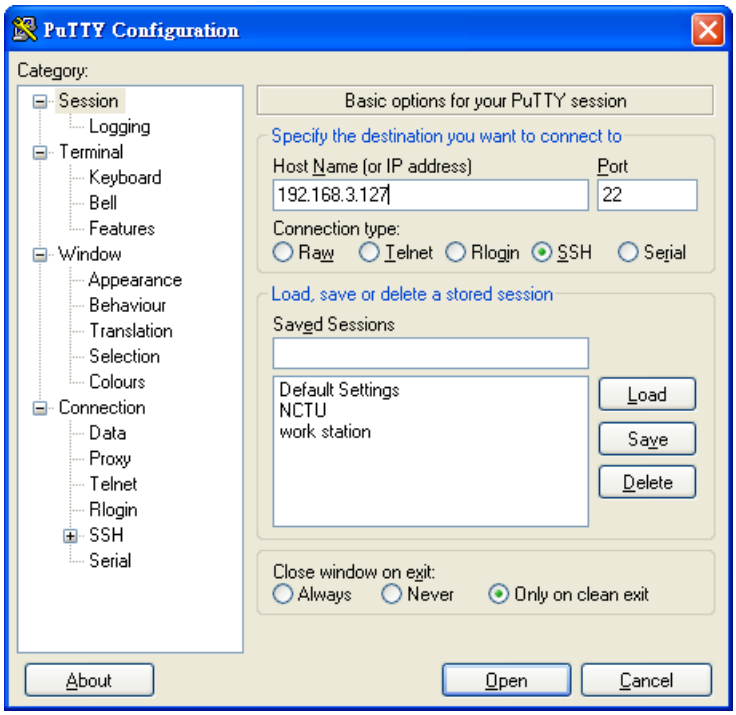
For more information about SSH, refer to the following link:
https://wiki.debian.org/SSH

Windows Users

NOTE Do NOT apply these steps to the UC-8540/8580 computer. These steps are for the Windows PC that you use to connect to the UC-8540/8580 computer.

Take the following steps from your Windows PC.

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-8540/8580 computer in a Windows environment. The following figure shows an example of the configuration that is required:



In the console that opens, type in the **username** and **password** to establish an SSH connection with the UC-8540/8580 computer.

Sudo Mechanism

In the UC-8540-LX, the **root** account login is disabled to ensure a higher level of 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 root privileges as possible to users to enable them to get their work done. Using sudo is better (safer) than opening a session as root for several 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 demonstrates that some system-level commands are not available to the user **moxa**.

```
moxa@Moxa:~$ ifconfig

-bash: ifconfig: command not found

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
```

Booting Up the UC-8540 for the First Time

We suggest using the serial console when you log in for the first time. Once you have connected the UC-8540/8580 computer to a PC, power on the UC-8540. The computer will start the boot-up process immediately. The power LED will light up first, followed by the ready LED. The serial console will display messages that indicate the status of the boot-up process. When the computer boots-up for the first time, the root file system is resized and initialized.

User Account Management

Switching to the Root Account

You can switch to the **root** user account using the **sudo -i** (or **sudo su**) command. For security reasons, do not operate the "**all**" command from the **root** account.

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

ATTENTION

You might get a **permission denied** message when you use pipe or redirect behavior with a non-root account.

You must use **'sudo su -c'** to run the command instead of using >, <, >>, <<, etc.

Note: The single quotes around the full command are required.

Creating and Deleting User Accounts

You can use the commands **useradd** and **userdel** to create and delete user accounts. Refer to the main page of these commands to set relevant access privileges for the account. The following example shows how you can create a user, **test1** in the **sudo** group. The default login shell for the user is **bash** and the home directory is **/home/test1**.

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

To change the password of **test1**, use the **passwd** command and enter the new password twice to confirm the change as shown below:

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

To delete the **test1** user, use the **userdel** command as follows:

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

Disabling the Default User Account

ATTENTION

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

Use the **passwd** command to lock the default user account so the user, **moxa** cannot log in.

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

To unlock the user account **moxa**, use the following command:

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

Network Settings

Configuring Ethernet Interfaces

After the first login, you can configure the UC-8540/8580 computer's network settings to better fit your application. A serial console makes it more convenient for you to manipulate the network interface settings, which can help you to avoid reconnections, when compared to an SSH login.

Modifying Network Settings via the Serial Console

In this section, we use the serial console to configure the UC-8540/8580 computer's network settings. Follow the instructions given in the *Connecting to the UC-8540/8580 computer* section to access the console utility of the target UC-8540/8580 computer via the serial console port, and then type **Moxa:~# cd /etc/network** to change the directory path.

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

Type **Moxa:~# sudo vi interfaces** to edit the network configuration file with the **vi** editor. You can configure the UC-8540/8580 computer's Ethernet ports to use either **static** or **dynamic** (DHCP) IP addresses.

Setting a Static IP Address

To set a static IP address for the UC-8540/8580 computer, use the **iface** command to modify the **address**, **network**, **netmask**, and **broadcast** parameters of the Ethernet interface.

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

# embedded ethernet LAN1
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~
```

ATTENTION

After changing the IP configuration, restart the Ethernet interface.
Example:
ifdown eth0 (turn off eth0)
ifup eth0 (turn on eth0).

Setting a Dynamic IP Address

To configure one or both LAN ports to request an IP address dynamically, use the **dhcp** option in place of the **static** option in the **iface** command as follows:

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
```

System Administration

Querying the Firmware Version

To check the UC-8540/8580 computer's firmware version, type:

```
moxa@Moxa:~$ kversion
UC-8540-LX version 1.1
```

Add the **–a** option to the command to view the build number:

```
moxa@Moxa:~$ kversion -a
UC-8540-LX  version 1.1 Build 18090615
```

Adjusting the Time

The UC-8540/8580 computer has two time settings. One is the system time, and the other is the RTC (Real-Time Clock) time maintained by the UC-8540/8580 Series hardware. Use the **#date** command to query the current system time or set a new system time. Use the **#hwclock** command to query the current RTC time or set a new RTC time.

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

MM = Month
DD = Date
hhmm = hour and minute

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

Use the following command to set the RTC time using the 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 date and time:
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 to configure the Moxa embedded computer's time zone. One is using the **TZ** variable. The other is using **/etc/localtime** file.

Using the TZ Variable

The format of the TZ environment variable format looks like this:

TZ=*<Value>HH[:MM[:SS]][daylight[HH[:MM[:SS]]][,start date[/starttime], enddate[/endtime]]]*

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

1. **TZ=EST5EDT**
2. **TZ=EST0EDT**
3. **TZ=EST0**

In the first case, the reference time is GMT and the stored time values are correct worldwide. A simple change of the TZ variable can print the local time correctly in any time zone.

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. You can use this option if the hardware clock on your machine automatically adjusts the Daylight Saving Time or you would like to manually adjust the hardware time twice a year.

```
moxa@Moxa:~$ TZ= EST5EDT
moxa@Moxa:~$ export TZ
```

You must include the TZ setting in the **/etc/rc.local** file. The timezone setting will be activated when you restart the computer.

The following table lists 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
+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

Using the /etc/localtime File

The local timezone information is stored in the **/etc/localtime** file and is used by the GNU Library for C (glibc) if no value has been set for the TZ environment variable. This file is either a copy of the **/usr/share/zoneinfo/** file or a symbolic link to it. You should find a suitable timezone information file and write over the original local time file in the UC-8540/8580 computer.

Determining Available Drive Space

To determine the amount of available drive space, use the **df** command with the **–h** tag. The system will return the amount of drive space broken down by file system. Here is an example:

```
moxa@Moxa:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       3.3G  600M  2.6G  19% /
devtmpfs        505M     0  505M   0% /dev
tmpfs           505M     0  505M   0% /dev/shm
tmpfs           505M  6.8M  499M   2% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           505M     0  505M   0% /sys/fs/cgroup
```

Enabling and Disabling Daemons

By default, only the following daemons are enabled in the UC-8540/8580 computer:

sshd Secure shell server daemon

You can use the **systemctl** command to manage which services will run in the background. The following example shows how to add the SNMP daemon to the current *run level*.

```
moxa@Moxa:~$ sudo systemctl enable snmpd
```

The SNMP daemon will not get activated in the current boot session but will be running in the background from the next boot session.

To disable the SNMP daemon, use the following command:

```
moxa@Moxa:~$ sudo systemctl disable snmpd
```

You can also write your own script to start and stop a daemon during 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 a current boot session by using the scripts in the **/etc/init.d** file. To start the SNMP daemon, use:

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

To stop the SNMP daemon, use:

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

In comparison to **systemctl**, scripts in **/etc/init.d/** will only start or stop the services in the current boot session. Once you reboot the UC-8540/8580 computer, it will go back to the default settings managed by **systemctl**.

Package Management

Most of the software Debian packages are maintained by the Debian community in the official Debian **apt** repository. The features that are exclusively supported by the UC-8540/8580 Series are maintained by Moxa. You must add the Moxa repository to the **/etc/apt/sources.list** file to keep your system up to date with the newest UC-8540/8580 Series packages.

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

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

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

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

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

The following packages are maintained in Moxa's official repository.

Package Name	Version	Architecture	Description
libssl1.0.0:armhf	1.0.1k-3+deb8u1+moxa	armhf	Secure Sockets Layer toolkit shared libraries
openssl	1.0.1k-3+deb8u1+moxa	armhf	Secure Socket Layer (SSL) binary
moxa-cellular-utils	1.0.0	armhf	Cellular-related utility on the Moxa computer. (libqmi: v1.12.6)
UC-8540-diag	1.0.0	armhf	Self-diagnostic utility on a UC-8540/8580 Series embedded computer
UC-8540-push-btn	1.0.0	armhf	Push-button utility on a UC-8540/8580 Series embedded computer
UC-8540-setinterface	1.0.0	armhf	Adjust UART mode utility on a UC-8540/8580 Series embedded computer
moxa-snmpd	1.0.0	armhf	SNMP (Simple Network Management Protocol)
UC-8540-system	1.0.0	armhf	System files on a UC-8540/8580 Series embedded computer
moxa-wifi-utils	1.0.0	armhf	Wi-Fi related utility on the Moxa computer.

Rebooting/Shutting Down the Computer

IMPORTANT!

Do NOT use the reset switch on the front panel of the UC-8540/8580 computer to shut down a running Debian GNU/Linux system. Do NOT turn off the UC-8540 when Debian GNU/Linux OS is running on 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, a **log out** option is usually available from the application menu. The **log out** option provides the proper means of shutting down (or rebooting) the system.

To reboot the UC-8540/8580 computer, use the following command:

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

To shut down the UC-8540/8580 computer, use the following command:

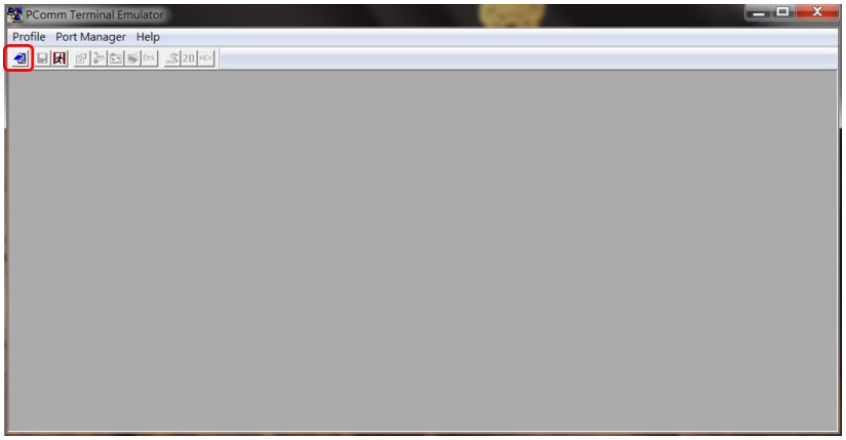
```
moxa@Moxa:~$ sudo shutdown -h "now"
```

Updating the Firmware Using a USB Disk

The firmware of the UC-8540/8580 computer can be updated through an external USB disk. Prepare a USB disk with the firmware image and plug it into USB port of the UC-8540/8580 computer. Power on the computer and take the following steps:

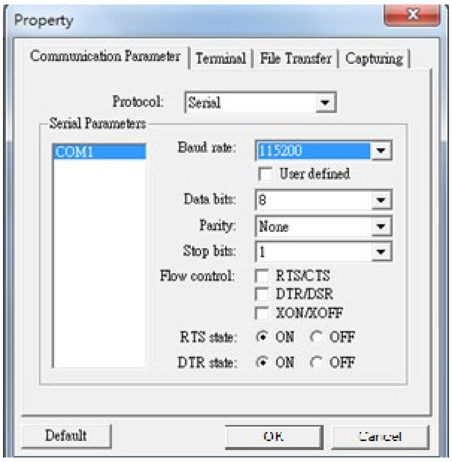
Windows Users:

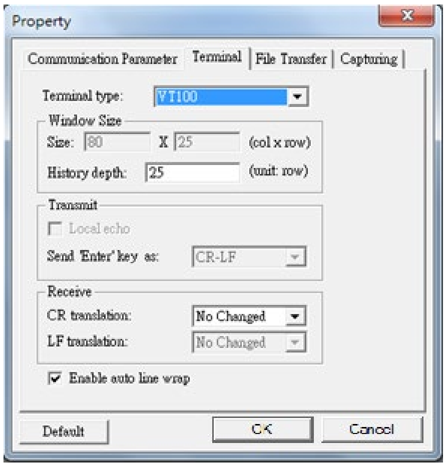
1. Download **PComm Lite** from the following site to set up a telnet client for windows.
   http://www.moxa.com/product/download_pcommlite_info.htm
2. Run the **PComm Terminal Emulator** on your windows computer.
3. Click the **Open** icon located on the upper-left corner of the window.



4. Configure the following properties and connect the UC-8540/8580 computer to the same COM port.

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





Linux Users:

1. After powering on the UC-8540/8580 computer, press DEL to enter the Bootloader configuration settings.

```
----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
----------------------------------------------------------------------------
(0) Fastboot mode                (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
---------------------------------------------------------------------------
Command>>
----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
----------------------------------------------------------------------------
(0) Fastboot mode                (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
---------------------------------------------------------------------------
Command>>
----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
----------------------------------------------------------------------------
(0) Fastboot mode                (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
---------------------------------------------------------------------------
Command>>
----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
----------------------------------------------------------------------------
(0) Fastboot mode                (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
---------------------------------------------------------------------------
Command>>
```

2. Enter **1** and type in the firmware filename.

The system will start the firmware upgrade process.

```
-----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
-----------------------------------------------------------------------------
(0) Fastboot mode               (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
----------------------------------------------------------------------------
Command>>1
Firmware File Name (firmware.img): FWR_UC-8540-LX_V1.0_Build_17021003_bata.img
-----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
-----------------------------------------------------------------------------
(0) Fastboot mode               (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
----------------------------------------------------------------------------
Command>>1
Firmware File Name (firmware.img): FWR_UC-8540-LX_V1.0_Build_17021003_bata.img
-----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
-----------------------------------------------------------------------------
(0) Fastboot mode               (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
----------------------------------------------------------------------------
Command>>1
Firmware File Name (firmware.img): FWR_UC-8540-LX_V1.0_Build_17021003_bata.img
-----------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
-----------------------------------------------------------------------------
(0) Fastboot mode               (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
----------------------------------------------------------------------------
Command>>1
Firmware File Name (firmware.img): FWR_UC-8540-LX_V1.0_Build_17021003_bata.img
```

```
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
201326592 bytes read in 104939 ms (1.8 MiB/s)

MMC write: dev # 0, block # 0, count 393216 ... 393216 blocks written: OK
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
201326592 bytes read in 105312 ms (1.8 MiB/s)

MMC write: dev # 0, block # 393216, count 393216 ... 393216 blocks written: OK
```

```
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
201326592 bytes read in 105762 ms (1.8 MiB/s)

MMC write: dev # 0, block # 786432, count 393216 ... 393216 blocks written: OK
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
25165824 bytes read in 14392 ms (1.7 MiB/s)

MMC write: dev # 0, block # 1179648, count 49152 ... 49152 blocks written: OK
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
201326592 bytes read in 104939 ms (1.8 MiB/s)

MMC write: dev # 0, block # 0, count 393216 ... 393216 blocks written: OK
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
201326592 bytes read in 105312 ms (1.8 MiB/s)

MMC write: dev # 0, block # 393216, count 393216 ... 393216 blocks written: OK
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
201326592 bytes read in 105762 ms (1.8 MiB/s)

MMC write: dev # 0, block # 786432, count 393216 ... 393216 blocks written: OK
switch to partitions #0, OK
mmc0(part 0) is current device
reading fwr_UC-8540-lx_v1.0_build_17021003_bata.img
25165824 bytes read in 14392 ms (1.7 MiB/s)

MMC write: dev # 0, block # 1179648, count 49152 ... 49152 blocks written: OK
```

3. After the firmware upgrade process is complete, unplug the power and reboot the system.

```
--------------------------------------------------------------------------
  Model: UC-8540
  Boot Loader Version 1.0.0S02          Serial Number: BOSTON000004
  LAN1 MAC: 00:90:e8:00:ee:0d        LAN2 MAC: 00:90:e8:00:ee:0e
--------------------------------------------------------------------------
(0) Fastboot mode               (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
--------------------------------------------------------------------------
Command>>
```

4. After rebooting the machine, you can use the following command to check if the firmware is up to date.

```
moxa@Moxa:~$ kversion -a
UC-8540-LX version 1.0 Build 17021003
```

3

Advanced Configuration of Peripherals

In this chapter, we include more information on the UC-8540/8580 computer's peripherals, such as the serial interface, storage, and cellular module.

The following topics are covered in this chapter:

- **Serial Ports**
  - stty
- **USB Port**
  - USB Automount
- **Restoring the Firmware to Factory Default Settings**
- **Using Cellular Modules**
  - Cellular Signal Strength
  - Cellular Management Utility
  - Dial-Up Connections
  - Disconnecting from a Dial-Up Network
  - Powering On/Off the Cellular Module
- **Configuring the Wireless LAN**
  - Wi-Fi Management Utility

Serial Ports

The serial ports support RS-232, RS-422, and RS-485 2-wire operation modes with flexible baudrate settings.

The default operation mode is set to **RS-232**. Use the **setinterface** command to change the operation mode as follows:

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

Interface Number	Operation Mode
None	Display current setting
0	RS-232
1	RS-485 2-wire
2	RS-422
3	RS-485 4-wire

For example, to set /dev/ttyM0 to RS-485 2-wire (RS485-2W) mode and view the current setting, use the following commands:

```
moxa@Moxa:~# sudo setinterface /dev/ttyM0 1
Now setting is RS-485 2-wire interface
moxa@Moxa:~# sudo setinterface /dev/ttyM0
UART Port#0 is in RS-485 2-wire interface
```

stty

The **stty** command is used to manipulate the serial terminal settings. You can view and modify the serial terminal settings with this command as described below:

Displaying All Serial Terminal Settings

The following text shows how to display all settings:

```
moxa@Moxa:~$ sudo stty -a -F /dev/ttyS0
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
```

Configuring the Serial Terminal Settings

The following example changes the **baudrate** to **115200**.

```
moxa@Moxa:~$ sudo stty 115200 -F /dev/ttyS0
```

After you run this command, the **baudrate** will be changed to **115200**.

```
moxa@Moxa:~$ sudo stty -a -F /dev/ttyS0
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 Refer to the following link for additional details on the **stty** command:
http://www.gnu.org/software/coreutils/manual/coreutils.html#stty-invocation

USB Port

The UC-8540/8580 computer has a USB port that you can use to expand the storage capacity of the computer.

USB Automount

The UC-8540/8580 computer supports the hot plug function for connecting USB mass storage devices. However, by default, the automount utility (udev) only supports automounting of one partition. Use the **mount** command to view details about all partitions.

ATTENTION

Remember to type the **#sync** command before you disconnect the USB mass storage device to prevent loss of data.

Exit the **/media/usb*** directory before you disconnect the storage device. If you stay in this directory, the auto un-mount process for the device will fail. If that happens, you can type **#umount /media/usb*** to unmount the device manually.

Restoring the Firmware to Factory Default Settings

To load the system's factory default settings, press the reset button for at least 5 seconds. While holding the button for the first 5 seconds, the ready LED will blink once each second. After holding the button continuously for more than 5 seconds, the ready LED will switch off, indicating that the factory defaults have been loaded.

You can also use the OS's **setdef** command to restore the computer to factory defaults:

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

ATTENTION

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

Please back up your files before resetting the system to factory defaults. All the data stored in the UC-8540/8580 computer's boot storage will be erased after resetting to factory defaults. Do not turn off the power supply when the Reset-to-default process is in progress.

Using Cellular Modules

The UC-8540/8580 computer has a mini PCIe socket for installing a cellular module. Contact your sales representative for more information about available modules.

Cellular Signal Strength

The following table shows how cellular signal strength is indicated by the signal indicators.

Signal Indicator	RSSI (dbm)	Condition
3 LEDs on (green)	>=-70	Excellent
2 LEDs on (green)	-100 to -71	Fair
1 LED on (green)	-110 to -101	Poor
No LED on	Else	No signal

Cellular Management Utility

Moxa provides a cellular management utility to access UC-8540's cellular modules. It's a script program with a series of command sets to identify, query, configure, dial up or power on/off cellular modules.

Cellular Management Utility (cell_mgmt) Command List

Using **cell_mgmt**:

```
cell_mgmt [-i <module id>] [-s <slot id>] <OPTIONS>
OPTIONS
      -i <module id>
            Module identifier, start from 0 and default to 0
      example: wwan0
      -s <slot id>
            Slot identifier, start from 1 and default value depends
            on module interface
            example: module 0 may in slot 2
```

A complete list of the commands with their descriptions is given below:

Command	Description
modules	Shows module numbers supported
slot	Shows module slot id
interface [interface id]	Switching and checking module interface(s)
start [OPTIONS]	Start network OPTIONS: APN - Access point name PIN - PIN code Phone - Phone number (especially for AT based modules)

Command	Description
	Username Password
stop	Stop network
restart	Restart network
power_on	Power ON.
power_off	Power OFF
power_cycle	Power cycle the module slot
switch_sim <1|2>	Switch SIM slot
gps_on	GPS ON
gps_off	GPS OFF
attach_status	Query network registration status.
status	Query network connection status
signal	Get signal strength
at <'AT_COMMAND'>	Input AT Command. Must use SINGLE QUOTATION to enclose AT Command
sim_status	Query sim card status
unlock_pin <PIN>	Unlock PIN code and save to configuration file
pin_retries	Get PIN code retry remain times
pin_protection <enable|disable> <current PIN>	Set PIN protection in the UIM
set_flight_mode <0|1>	Set module into flight mode (1) or online mode (0)
get_profiles	Get profile list format: <id>,<APN>,<PDP Type>
set_profile <id> [APN [PDP Type]]	Update PDP profile
set_apn <APN>	Set APN to configuration file
check_carrier	Check current carrier
switch_carrier <Verizon|ATT|Sprint|Generic>	Switching between US carrier frequency bands
m_info	Module/SIM information
module_info	Module information
module_ids	Get device IDs (ex: IMEI and/or ESN)
iccid	Get SIM card ID
imsi	Get IMSI (International Mobile Subscriber Identity)
location_info	Get cell location information
operator	Telecommunication operator
vzwauto	Verizon Private Network auto dialup
version	Cellular management version

Dial-Up Connections

You can manually set the APN in the **/etc/moxa-cellular-utils/moxa-cellular-utils.conf** file. Consult your service provider for the correct APN name and insert it into the configuration file. To dial up with the default configuration, use the following command in the following example:

```
moxa@Moxa:~$ sudo cell_mgmt start APN=internet PIN=0000
```

Switch SIM Card for Dial-Up Connections

The cell_mgmt tool can be used to choose specific cellular module and switch specific SIM card to Dial-up connections.

Example 1, switch slot 1 cellular module from SIM slot 1 to SIM slot 2 for Dial-up connection

```
$ sudo cell_mgmt -s 1 switch_sim 2
$ sudo cell_mgmt -s 1 start APN=internet PIN=0000
```

Example 2, read module info of wwan0

```
$ sudo cell_mgmt -i 0 module_info
```

Disconnecting from a Dial-Up Network

Be sure to disconnect the connection if you no longer need the service using the following command:

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

Powering On/Off the Cellular Module

The **cell_mgmt** tool can be used to re-initialize the module without rebooting the UC-8540. Issue the following command to power off the module:

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

Issue the following command to re-initialize and power on the cellular module:

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

NOTE Additional information about **qmi** utilities can be found at the following link:
http://www.freedesktop.org/wiki/Software/libqmi/

Configuring the Wireless LAN

Wi-Fi Management Utility

Moxa provides Wi-Fi management utility to access UC-8540's Wi-Fi modules. It's a script program with a series of command sets to identify, query, configure, connect, or power on/off Wi-Fi modules.

Wi-Fi Management Utility (wifi_mgmt) Command List

Using **wifi_mgmt**:

```
wifi_mgmt [-i <interface id>] [-s <slot id>] [OPTIONS]
OPTIONS
       -i <interface id>
              Interface identifier.
              example: wlan0
       -s <slot id>
              Slot identifier, start from 1 and default value depends
              on module interface.
              example: wlan0 may in slot 3
```

Command	Description
start Type=[type] SSID=[ssid] Password=[password]	Insert an AP information to the managed AP list and then connect to the AP [type] open/wep/wpa/wpa2 [ssid] access point's SSID [password] access point's password
start [network id	Connect to AP by the managed AP list network id
start	Connect to the last time AP that was used
scan	Scan all the access points information
signal	Show the AP's signal
list	Show the managed AP list
insert Type=[type] SSID=[ssid] Password=[password]	Insert a new AP information to the managed AP list [type] open/wep/wpa/wpa2 [ssid] access point's SSID [password] access point's password
delete [network id]	Choose an AP network id to delete which is in the managed AP list
select [network id]	Select an AP network id to connect which is in the managed AP list
stop	Stop network
status	Query network connection status
interfaces	Show interface numbers
interface [num]	Switch to another wlan[num] interface [num] interface number
interface	Get the current setting interface
reconnect	Reconnect to the access point
restart	Stop the **wifi_mgmt** command then start it again
version	Wi-Fi management tool version

4

Programmer's Guide

In this chapter, we briefly introduce the toolchain and teach you how to program the UC-8540/8580 computer. The programming example package can be downloaded from Moxa's website.

The following topics are covered in this chapter:

- **Introduction to the Linux Tool Chain**
  - Native Compilation
  - Cross Compilation
  - Obtaining Help
- **Developing a Test Program—hello.c**
  - Compiling hello.c with Native Compilation
  - Compiling hello.c using Cross Compilation
- **Makefile Example**
- **RTC (Real Time Clock)**
- **WDT (Watch Dog Timer)**
- **Cryptographic Hardware Accelerator**
- **LED Indicators**
- **Power Ignition Function**

Introduction to the Linux Tool Chain

Linux Toolchain contains the necessary libraries and compilers for developing your programs. The UC-8540/8580 computer supports both native compilation and cross compilation of code. Native compiling is more straightforward since all the coding and compilation can be done directly on the UC-8540/8580 computer, but since you will be constrained by the UC-8540/8580 computer's ARM CPU resources, the compilation speed is slower. On the other hand, cross compiling can be done on any Linux machine with the correct toolchain, and the compilation speed is much faster.

Native Compilation

Follow these steps to update the package menu.

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

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

3. Install the 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 the UC-8540/8580 computer, you must ensure that it is compiled and linked to the same libraries that will be present on the UC-8540/8580 computer.

The host tool chain that comes with the UC-8540/8580 computer contains a suite of cross compilers and other tools, as well as the libraries and headers that are necessary to compile applications for the UC-8540/8580 computer. The host environment must be running Linux to install the UC-8540/8580 computer 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, Fedora core 1 to 20, and Debian 4/5/6/7 32-bit/64-bit platforms.

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

After you unzip the package, run the install script, and follow the instructions.

```
user@Linux:~$ sudo ./arm-linux-gnueabihf_4.9_Build_amd64_16053113.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-4.9/lib/x86_64-linux-gnu/
usr/local/arm-linux-gnueabihf-4.9/lib/x86_64-linux-gnu/libexpat.so.1
usr/local/arm-linux-gnueabihf-4.9/lib/x86_64-linux-gnu/libexpat.so.1.6.0
usr/local/arm-linux-gnueabihf-4.9/lib/ld-linux-armhf.so.3
-----------------------------------------------
arm-linux-gnueabihf install complete
Please export these environment variables before using toolchain:
export PATH=$PATH:/usr/local/arm-linux-gnueabihf-4.9/usr/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_4.9_Build_amd64_16053113//bin** to your path and the directory **/usr/local/arm-linux-gnueabihf_4.9_Build_amd64_16053113-20130415//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_4.9_Build_amd64_16053113//bin:$PATH"
#export MANPATH="/usr/local/arm-linux-gnueabihf_4.9_Build_amd64_16053113//man:$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_4.9_Build_amd64_16053113/. This means that the original /usr/local/arm-linux-gnueabihf_4.9_Build_amd64_16053113/ path will be overwritten. If you have installed an old arm-linux toolchain, you will need to rename the original folder before installing the new one.

Cross Compiling Applications and Libraries

To compile a simple C application, 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
```

Obtaining Help

You can use the Linux **man** utility to get help on many of the utilities provided by the tool chain located at /usr/local/arm-linux-gnueabihf-4.9/usr/share/man/man1/. For example, to get help on the **arm-linux-gnueabihf-gcc** compiler, issue the command:

```
user@Linux:~$ man /usr/local/arm-linux-gnueabihf-4.9/usr/share/man/man1/arm-linux-gnueabihf-gcc-ar-4.9.1.gz

/usr/local/arm-linux-gnueabihf-4.9/usr/share/man/man1/arm-linux-gnueabihf-gcc-nm-4.9.1.gz

/usr/local/arm-linux-gnueabihf-4.9/usr/share/man/man1/arm-linux-gnueabihf-gcc-ranlib-4.9.1.gz
```

Developing a Test Program—hello.c

In this section, we use the standard "Hello World" example to illustrate how to develop a program for the UC-8540/8580 computer.

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

The following compiler tools are provided in the UC-8540.

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

Use the following commands for native compilation.

```
apt-get install build-essential
sudo gcc –o hello-release hello.c
sudo strip –s hello-release
```

After compiling the program, issue the following command to run the program.

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

Compiling hello.c using Cross Compilation

Follow these steps for cross compilation.

1. Connect the UC-8540/8580 computer to a Linux PC.
2. Install the 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 the UC-8540/8580 computer 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-8540/8580 computer units if needed).

The CD provided with the UC-8540/8580 Series 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-8540/8580 Series computer.

Uploading and Running the hello.c Program

The program can be uploaded via SFTP, NFS, SCP, or RSYNC.

Use the following command to upload **hello-release** to the UC-8540/8580 computer via SFTP.

From the PC, type:
sftp moxa@192.168.3.127

Use the "put" command to initiate the file transfer:

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

From the UC-8540/8580 computer, type:

 # chmod +x hello-release

 # ./hello-release

The phrase **"Hello World"** is printed on the screen.

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

NOTE Contact Moxa technical support staff if you need help to use the example code.

Makefile Example

The following Makefile is copied from the "Hello World" example in the package provided with the UC-8540/8580 computer. For cross compilation, use the following:

```
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 compilation, make the following changes:

```
CC = gcc
CPP = g++
```

RTC (Real Time Clock)

The device node is located at **/dev/rtc0**. The UC-8540/8580 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	Reads time information from the RTC; returns the value of argument 3.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_RD_TIME, &rtc_tm);
Function	**RTC_SET_TIME**
Description	Sets the RTC time. Argument 3 will be passed to the RTC.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_SET_TIME, &rtc_tm);
Function	**RTC_ALM_SET**
Description	Sets the alarm time.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_ALM_SET, &rtc_tm);
Function	**RTC_ALM_READ**
Description	Reads the alarm time.
Usage	struct rtc_time rtc_tm; ioctl(fd, RTC_ALM_READ, &rtc_tm);

Function	**RTC_IRQP_SET**
Description	Sets the 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	Reads the IRQ rate.
Usage	unsigned long tmp; int ioctl(fd, RTC_IRQP_READ, &tmp);
Function	**RTC_ALM_SET**
Description	Sets the 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 examples in the example package to see how to use these functions.

WDT (Watch Dog Timer)

You can either enable or disable the WDT based on your application. When the WDT is enabled, 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 1 day. The default ack time is 60 seconds and the **nowayout** parameter is enabled by default. You cannot disable the watchdog once it has been started. For this reason, if the watchdog daemon crashes, the system will reboot after the timeout period has passed.

Function	**WDIOC_KEEPALIVE**
Description	Writes to the watchdog device to keep the watchdog alive.
Usage	int ioctl(fd, WDIOC_KEEPALIVE, 0)
Function	**WDIOC_SETTIMEOUT**
Description	Modifies the watchdog timeout Min: 1second. Max: 1day; Default: 60seconds
Usage	int timeout = 60; ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
Function	**WDIOC_GETTIMEOUT**
Description	Queries the current timeout
Usage	int timeout; ioctl(fd, WDIOC_GETTIMEOUT, &timeout);

Function	WDIOC_GETSTATUS
Description	Asks for the current status
Usage	int flags; ioctl(fd, WDIOC_GETSTATUS, &flags);
Function	**WDIOC_GETBOOTSTATUS**
Description	Asks for the status at the last reboot
Usage	int flags; ioctl(fd, WDIOC_GETBOOTSTATUS, &flags);
Function	**WDIOC_GETSUPPORT**
Description	Asks 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 encryption/decryption and compression/decompression tasks from CPU. You can use the cryptographic hardware accelerator when your application needs to do cryptographic calculations. To use it, you need to make sure that the cryptodev driver is loaded.

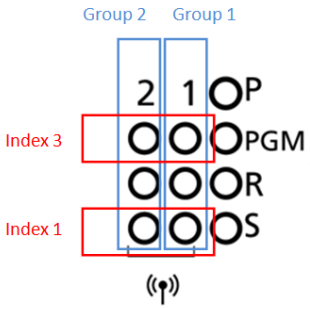
NOTE Click the following link for more information about cryptodev:
http://cryptodev-linux.org/documentation.html

LED Indicators

You can configure and view the status of the UC-8540's LEDs using Moxa's LED control utility. An example code is provided to enable you to use the utility in your applications.

The command **`mx_led_control`** enables you to control UC-8540's 6 Signal LEDs and 1 Programmable (PGM) LED.

Each Signal LED can be identified by its group and index number as shown below:



Use the **mx_led_control –h** command to view instructions and help regarding the command.

```
# mx_led_control -h
Usage:
            /sbin/mx_led_control <-s #sgn_group |-p #prog_group > <-i #led_index
|-r #data >
OPTIONS:
            -s <#led_group>
                        Control signal led
            -p <#led_group>
                        Control programmable led
            -i <#led_index> [on|off|blink]
                        Switch #led_index led on/off/blink
            -r <#data>
                        Switch #led_group leds on/off/blink by raw data
                        data 0 --> led off
                        data 1 --> led on
                        data b --> led blink
            --all-signal <on|off|blink>
                        Switch all signal leds on/off/blink
            --all-programmable <on|off|blink>
                        Switch all programmable leds on/off/blink
```

Some examples are given below:

Example 1: Turning off the Programmable LED.

```
# mx_led_control -p 1 -i 1 off
```

Example 2: Turning on the Signal LED in group 2 with index number 3.

```
# mx_led_control -s 2 -i 3 on
```

Example 3: Turning on/off the signal LEDs in group 2 as given below using user input.

Index 1 --> off
Index 2 --> on
Index 3 --> off
Index 4 --> off

```
# mx_led_control -s 2 -r 0100
```

NOTE The signal LEDs are controlled by UC-8540's firmware and are used to indicate the wireless signal strength. If you need to customize the signal LEDs for other purposes, please contact a Moxa sales representative for customization services.

Power Ignition Function

The Power Ignition function controls the computer's power behavior. This function detects the ignition signal status and allows users to control the on/off delay time setting through Moxa's Power Ignition Software Utility.

The default setting of power ignition function is disabled. You could use Moxa power ignition utility to enable the function.

You can use command **mx_igt –h** for help instructions

```
# mx_igt -h
Moxa power ignition utility
```

```
Usage:
      /sbin/mx_igt [Options]

Options:
-l             , list power ignition configuration
-s [on|off]    , setting power on/off function
-t <time>      , setting delay time(seconds) of power on/off
-e             , enable power ignition
-d             , disable power ignition

Example:
mx_igt -l, power ignition configuration state
mx_igt -s on -t 10, set 10 seconds delay time for power on
mx_igt -d, disable power ignition function
```

To enable the power ignition function, use the following command:

```
# mx_igt -e
Ignition function is ENABLE
```

To list the configurations of current power ignition setting, use the following command:

```
# mx_igt -l
Power ignition configuration:
-------------------------------------------
Status : Disable (val=0x00)
Signal : OFF (val=0x00)
Delay time of power on : 3 (sec)
Delay time of power off : 3 (sec)
```

For example, to set 10 seconds delay time for power on

```
# mx_igt -s on -t 10
```

You will see the delay time of power on is set to 10 seconds:

```
# mx_igt -l
Power ignition configuration:
-------------------------------------------
Status : Disable (val=0x00)
Signal : OFF (val=0x00)
Delay time of power on : 10 (sec)
Delay time of power off : 3 (sec)
```

To disable the power ignition function, use the following command:

```
# mx_igt -d
Ignition function is DISABLE
```

To utilize the power ignition function, you need to use the following command to activate service first

```
# systemctl unmask mx_igt
# reboot
```

After reboot, use the following command to enable ignition function.

```
# mx_igt -e
Ignition function is ENABLE
```

A

Using the General Debian Package

In this chapter, we explain how to use the general Debian Package on the UC-8540/8580 computer.

The following topics are covered in this appendix:

- **NTP Client**
- **Executing Scheduled Commands with cron**
  - Updating System Time and RTC
- **Log Processing using rsyslog**
  - Rsyslog's Configuration File
  - Using Selectors
- **OpenSSL**
  - Ciphers
  - Cryptographic Hash Functions
  - Public-Key Cryptography
- **The Apache Web Server**
  - Editing the ServerName in the Apache Configuration File
- **SFTP**
- **DNS**
  - /etc/hosts
  - /etc/resolv.conf
  - /etc/nsswitch.conf
- **iptables**
  - Observing and Erasing Chain Rules
  - Defining a Policy for Chain Rules
  - Appending or Deleting Rules
- **NAT**
  - NAT Example
  - Enabling NAT at Bootup
- **rsync**
  - Using rsync for External Backups
  - Automating rsync Backups
- **NFS (Network File System)**
  - Setting Up the UC-8540/8580 computer as an NFS Client
- **SNMP**
- **OpenVPN**
  - Static-Key VPN
- **Package Management**
  - apt-get
  - apt-cache
  - Listing All Available Packages
  - Finding the Package Name and Software Description
  - Checking Package Information
  - Checking Dependencies for Specific Packages
  - Checking the Cache Statistics
  - Updating System Packages
  - Installing or Upgrading Specific Packages
  - Upgrading All Software Packages
  - Installing Multiple Packages
  - Installing Packages Without Upgrading
  - Upgrading Specific Packages
  - Installing Specific Package Version
  - Removing Packages
  - Completely Removing Packages
  - Cleaning Up Disk Space
  - Downloading Only the Source Code of a Package
  - Downloading and Unpacking a Package
  - Downloading, Unpacking, and Compiling a Package
  - Download a Package Without Installing the Package
  - Checking the Change Log of a Package
  - Checking Broken Dependencies
  - Searching and Building Dependencies
  - Cleaning Apt-Get Cache
- **Removing Installed Packages**

NTP Client

The UC-8540/8580 computer has a built-in NTP (Network Time Protocol) client that is used to initialize a time request to a remote NTP server. Use **#ntpdate <*IP address of the server*>** 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 10.128.8.1
8 Mar 13:13:01 ntpdate[1758]: step time server 10.128.8.1 offset 3.792556 sec
moxa@Moxa:~$ hwclock -w
moxa@Moxa:~$ hwclock
2017-03-08 13:14:14.497444+0800
```

NOTE Before using the NTP client utility, check your IP and DNS settings to make sure that an Internet connection is available. For details on configuring the Ethernet interface, refer to "Configuring Ethernet Interface" section under "Network Setting" in Chapter 2.

Executing Scheduled Commands with cron

The cron daemon reads **/etc/crontab** to retrieve scripts and other commands to be run at regularly scheduled times. The cron daemon wakes up every minute and checks each command listed in the crontab file to see if it should be run at that time.

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

Take the following steps to use cron to update the system time and RTC:

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

Log Processing using 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 delivered over TCP, SSL, TLS, and RELP, writing to databases, and email alerting. Rsyslog replaces syslogd.

Rsyslog is installed but disabled by default.

Start rsyslog manually	systemctl start rsyslog
Stop rsyslog manually	systemctl stop rsyslog
Enable rsyslog	systemctl enable rsyslog
Disable rsyslog	systemctl disable rsyslog

Rsyslog's Configuration File

The syntax of the **/etc/rsyslog.conf** file is detailed in the rsyslog.conf(5) manual page. The overall principle is to write "selector" and "action" pairs. The selector defines all relevant messages, and the action describes 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

Using Selectors

The selector is a semicolon-separated list of *subsystem*.*priority* pairs (example: **auth.notice;mail.info**). An asterisk represents 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 emergency 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 (e.g., **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 serves to exclude a subsystem from a set of messages. Thus, ***.crit;kern.none** indicates all the messages of priority equal to or higher than critical, not originating from the kernel.

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

OpenSSL

The UC-8540/8580 computer supports hardware accelerator with openssl. Check the version of openssl; it should indicate that it was modified by Moxa.

```
moxa@Moxa:~$ dpkg -l | grep openssl
ii  openssl                          1.0.2j-1~bpo8+1+moxa       armhf       Secure
Sockets Layer toolkit - cryptographic utility
```

Before enabling the hardware accelerator:

```
root@Moxa:/home/moxa# openssl speed -evp aes-128-cbc
Doing aes-128-cbc for 3s on 16 size blocks: 3647090 aes-128-cbc's in 3.01s
Doing aes-128-cbc for 3s on 64 size blocks: 1117858 aes-128-cbc's in 3.01s
Doing aes-128-cbc for 3s on 256 size blocks: 299016 aes-128-cbc's in 3.01s
Doing aes-128-cbc for 3s on 1024 size blocks: 69587 aes-128-cbc's in 2.88s
Doing aes-128-cbc for 3s on 8192 size blocks: 9482 aes-128-cbc's in 3.01s
OpenSSL 1.0.1t  3 May 2016
built on: Fri Jan 27 00:26:25 2017
options:bn(64,32) rc4(ptr,char) des(idx,cisc,16,long) aes(partial) blowfish(ptr)
compiler: gcc -I. -I.. -I../include  -fPIC -DOPENSSL_PIC -DOPENSSL_THREADS -
D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -DL_ENDIAN -DTERMIO -g -O2 -fstack-protector-
strong -Wformat -Werror=format-security -D_FORTIFY_SOURCE=2 -Wl,-z,relro -Wa,--
noexecstack -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      19386.52k    23768.41k    25431.26k    24742.04k    25806.16k
```

After enabling the 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: 100640 aes-128-cbc's in 0.18s
Doing aes-128-cbc for 3s on 64 size blocks: 111456 aes-128-cbc's in 0.11s
Doing aes-128-cbc for 3s on 256 size blocks: 119786 aes-128-cbc's in 0.16s
Doing aes-128-cbc for 3s on 1024 size blocks: 107512 aes-128-cbc's in 0.08s
Doing aes-128-cbc for 3s on 8192 size blocks: 47739 aes-128-cbc's in 0.06s
OpenSSL 1.0.2j  26 Sep 2016
built on: reproducible build, date unspecified
options:bn(64,32) rc4(ptr,char) des(idx,cisc,16,long) aes(partial) blowfish(ptr)
compiler: gcc -I. -I.. -I../include  -fPIC -DOPENSSL_PIC -DOPENSSL_THREADS -
D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -DHAVE_CRYPTODEV -DUSE_CRYPTDEV_DIGESTS -
DL_ENDIAN -g -O2 -fstack-protector-strong -Wformat -Werror=format-security -
D_FORTIFY_SOURCE=2 -Wl,-z,relro -Wa,--noexecstack -Wall -DOPENSSL_BN_ASM_MONT -
DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DAES_ASM -DBSAES_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      8945.78k    64847.13k   191657.60k  1376153.60k  6517964.80k
```

OpenSSL supports several different cryptographic algorithms, described in the following subsections.

Ciphers

Ciphers support 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 was built by Moxa, or the hardware accelerator function will not work with other versions.

SFTP

The default SFTP daemon will start when the system boots up. The login and password used are the same as the system login and password (**moxa/moxa).** You can also configure the SFTP account using the following steps.

1. Create a user & group for SFTP access, without a 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]
```

To block the user account "sftp" from being used for normal Linux logins, and only available for sftp programs, we need to assign a special shell for the account. In the example shown below, we assign the shell "/bin/false" to the user account "sftp" and change account's folder and owner.

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

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

2. Execute the file "/etc/ssh/sshd_config" to append SSHD-related configuration**.**

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

Add the following commands at the end of the config file:

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

3. Restart SSHD Daemon:

```
moxa@Moxa:~$ sudo systemctl restart ssh
```

4. At this point, the account and its default path should be configured.

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

DNS

The UC-8540/8580 computer 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 using **#ntpdate time.nist.gov** 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 file if the DNS server's IP address is 168.95.1.1:

nameserver 168.95.1.1

```
10.120.53.100 - PuTTY
moxa@Moxa:~$ cat /etc/resolv.conf
#nameserver 192.168.3.1
nameserver 168.95.1.1
```

/etc/nsswitch.conf

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

iptables

ATTENTION

All the **iptables** settings configured in this section are temporary and will be reset to the settings configured by the RESTful API once the computer has been restarted.

The tool **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-8540/8580 computer supports three types of **iptables**: **Filter** tables, **NAT** tables, and **Mangle** tables:

Filter Table—includes three chains:

INPUT chain

OUTPUT chain

FORWARD chain

NAT Table—includes three chains:

PREROUTING chain—transfers the destination IP address (DNAT)

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

OUTPUT chain—produces local packets

sub-tables

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

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

MASQUERADE—a special form for SNAT. If one host can connect to 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 rules • **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-8540/8580 computer supports the following sub-modules. Be sure to use the module that matches your application. The most common modules are already built into 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-8540/8580 computer.
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, and iptables-save** commands to maintain the database.

NOTE The **iptables** tool 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 might 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, we have divided our discussion of the various rules into the following three categories to illustrate the **iptables** syntax: *Observing and Erasing Chain Rules*, *Defining a Policy for Chain Rules*, and *Appending or Deleting Rules*.

Observing and Erasing Chain Rules

Usage:

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

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

#iptables [-t tables] [-FXZ]

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

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

Defining a 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 destined for the UC-8540/8580 computer sockets.
OUTPUT: For locally-generated packets.
FORWARD: For packets routed out through the UC-8540/8580 computer.
PREROUTING: To alter packets as soon as they come in.
POSTROUTING: For altering incoming packets before routing them.

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.

Appending or Deleting 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-8540/8580 computer's ports, 137, 138, and 139

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

Example 7: Log TCP packets that are received by UC-8540/8580 computer'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**.

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 UC-8540 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. For additional information on NAT, visit http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

NAT Example

In this 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 UC-8540 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.
#echo 1 > /proc/sys/net/ipv4/ip_forward
#modprobe ipt_MASQUERADE
#iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE
```

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 the *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.

The **rsync** command can be used to back up data to the destination location with encryption. The following example illustrates how to back up data from directory1 to 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 that can be used for your rsync backups. We will show you the process to get one Linux machine to back up to another on a local network. The process would be the same if one host was somewhere on the Internet; 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 set up 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. You might also want to switch the port (default port is 22) that SSH listens on for added security.

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
```

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.

For additional information on **iptables** and **rsync**, visit http://rsync.samba.org/

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 the UC-8540/8580 computer, without worrying about the amount of disk space that will be available. The UC-8540/8580 computer supports NFS protocol for client.

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

Start nfs manually	sudo systemctl start nfs-common sudo systemctl start nfs-kernel-server sudo systemctl start rpcbind
Stop nfs manually	sudo systemctl stop nfs-common sudo systemctl stop nfs-kernel-server sudo systemctl stop rpcbind
Enable nfs	systemctl enable nfs-common systemctl enable nfs-kernel-server systemctl enable rpcbind
Disable nfs	systemctl disable nfs-common systemctl disable nfs-kernel-server systemctl disable rpcbind

Setting Up the UC-8540/8580 Computer as an NFS Client

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

Step 1: Create a folder to link a mount point on the NFS Client site.
`#mkdir –p /home/nfs/public`

Step 2: Mount the remote directory to a local directory.
`#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

The UC-8540/8580 computer has a SNMP (Simple Network Management Protocol) agent software built in, which supports RFC1317 RS-232 like group and RFC 1213 MIB-II. The SNMP daemon is installed but disabled by default. You can activate the daemon manually or set it to be enabled by default.

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

To	Run the command
Start snmpd manually	`systemctl start snmpd`
Stop snmpd manually	`systemctl stop snmpd`
Enable snmpd	`systemctl enable snmpd`
Disable snmpd	`systemctl disable snmpd`

The following simple example shows the use of an SNMP browser on the host site to query the UC-8540/8580 computer, which is the SNMP agent. The UC-8540/8580 computer responds with the following:

```
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.3.6.1.4.1.8691.12.8410
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 disabled by default. Use the **`insserv –d openvpn`** command to enable OpenVPN package at the next bootup. To enable the OpenVPN package with immediate effect, you can use the **`/etc/init.d/openvpn start`** command.

OpenVPN supports 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. Type the following command to load driver "tun".

`# 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.

The OpenVPN daemon is installed but disabled by default.

To	Run the command
Start openvpn manually	`systemctl start openvpn`
Stop openvpn manually	`systemctl stop openvpn`
Enable openvpn	`systemctl enable openvpn`
Disable openvpn	`systemctl disable 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
```

This is where 10.9.8.x is your VPN subnetwork, 10.9.8.1 is the IP of the server, and 10.9.8.2 the IP of the client.

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

```
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

When using an OpenVPN-related application, you need to create a firewall policy.

On the server's firewall, open 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

In this section, we explain how you can quickly learn to install, remove, update, and search for software packages using the **apt-get** and **apt-cache** commands from the command line. Some useful commands that will help you handle package management in Debian/Ubuntu based systems are listed in this section.

apt-get

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

apt-cache

The **apt-cache** command line tool is used to search for apt software package cache. That is, the tool is used to search for software packages, collect package information, and search for which available packages are ready for installation on Debian or Ubuntu based systems.

Listing All Available Packages

Use the following command to list all available packages:

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

Finding the Package Name and Software Description

To find the package name and description, use the "search" flag. Using "search" with apt-cache will display a list of matched packages with short descriptions. For example, if you would like to find the description of package "vim", use the following command:

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

To find and list all packages starting with "vim", use the following command:

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

Checking Package Information

To get more detailed package information (e.g., version number, check sums, size, installed size, category) along with the short description, use the **show** sub-command, as shown below:

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

Checking Dependencies for Specific Packages

Use the **showpkg** sub command to check the dependencies on specific software packages, and whether those dependent packages are installed or not. For example, use the **showpkg** command along with the package name as shown below:

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

Checking the Cache Statistics

The **stats** sub command displays the overall statistics of the cache. For example, the following command will show the complete package information of all packages found in the cache:

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

Updating System Packages

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

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

Installing or Upgrading Specific Packages

Use the **install** sub command to install or upgrade one or more packages.

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

Upgrading All Software Packages

The **upgrade** command is used to upgrade all software packages currently installed on the system.

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

Installing Multiple Packages

You can add more than one package name along with the command 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
```

Installing Packages Without Upgrading

Use the **–no-upgrade** sub command to prevent the installed packages from being upgraded.

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

Upgrading Specific Packages

Use the **–only-upgrade** sub command to NOT install new packages, but only upgrade already installed packages.

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

Installing Specific Package Version

To install a specific version of a package, use "**=**" with the package name and the version as shown below:

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

Removing Packages

To un-install software packages without deleting the configuration files (for reusing the same configuration later), use the **remove** command:

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

Completely Removing Packages

To remove software packages along with their configuration files, use the **purge** sub command:

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

Cleaning Up Disk Space

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

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

Downloading Only the Source Code of a Package

To download only the source code of a particular package, use the **–download-only source** option along with the package name as shown below:

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

Downloading and Unpacking a Package

To download and unpack the source code of a package, type the following command:

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

Downloading, Unpacking, and Compiling a Package

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

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

Download a Package Without Installing the Package

Use the **download** option to download any given package without installing it. For example, the following command will only download the "nethogs" package to the current working directory.

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

Checking the Change Log of a Package

The **changelog** flag downloads a package's change log and displays the version information of the package that is installed:

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

Checking Broken Dependencies

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

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

Searching and Building Dependencies

The **build-dep** command searches the local repositories in the system and installs the build dependencies for a package. If the package does not exist in the local repository, it will return an error code.

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

Cleaning Apt-Get Cache

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

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

Removing Installed Packages

The **autoremove** sub command is used to automatically remove packages that were installed to satisfy dependencies on other packages but are no longer required. For example, the following command will remove the installed package wget, including all its dependent packages:

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

B

Firmware Upgrade

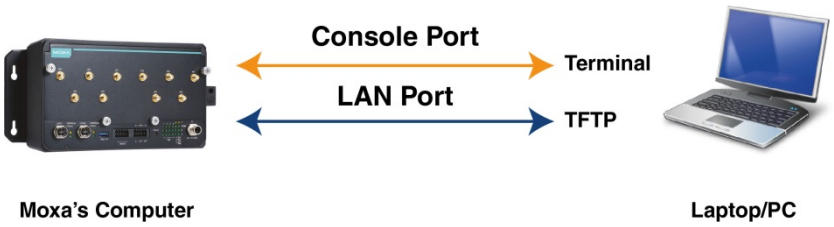
The following topics are covered in this appendix:

- **Overview**
  - A. Connecting to the UC-8540/8580 Computer
  - B. Download and Launch the TFTP Program
  - C. Downloading and Upgrading the Firmware Through the Serial Port

Overview

Moxa provides a boot loader utility for firmware upgrade or recovery. You will need the following items to use this utility.

1. The embedded computer that you would like to upgrade or recover.
2. A PC or a laptop computer.
3. A console port cable for connecting to the UC-8540/8580 Series computer.
4. A cross-over Ethernet cable for upgrading the firmware through a TFTP server and LAN port.
5. The firmware for the embedded computer.



There are three steps in the recovery process, as follows:

A. Connect to the UC-8540/8580 Series computer

B. Download and launch the TFTP program.

C. Download the firmware and upgrade through serial port.

If you are familiar with Moxa embedded computers and the firmware upgrade procedure, you may skip to Step C. However, we suggest that you go through all three steps to ensure the firmware upgrades properly.

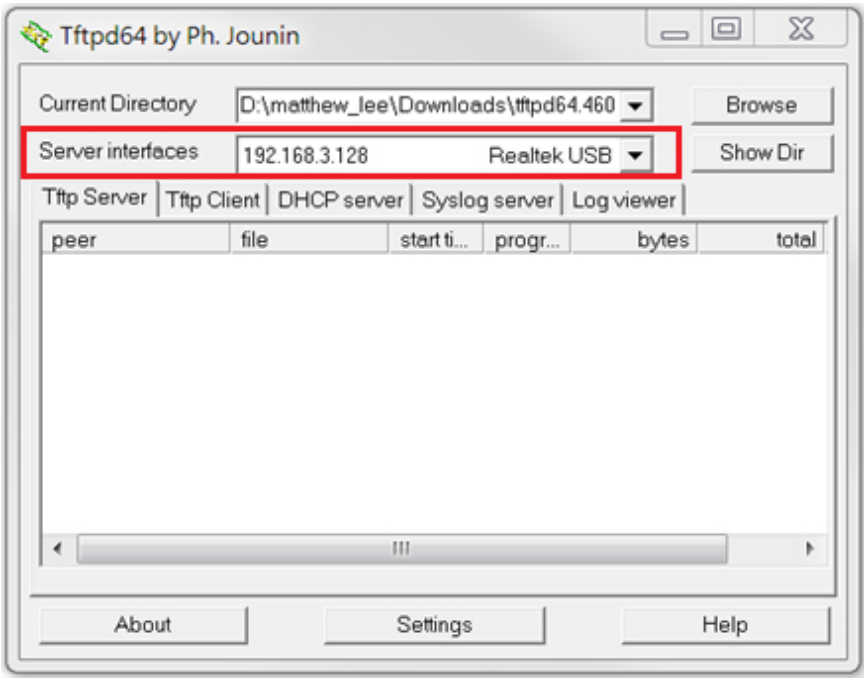
A. Connecting to the UC-8540/8580 Computer

Refer to Connecting to the UC-8540/8580 computer section in Chapter 2 Getting Started.

B. Download and Launch the TFTP Program

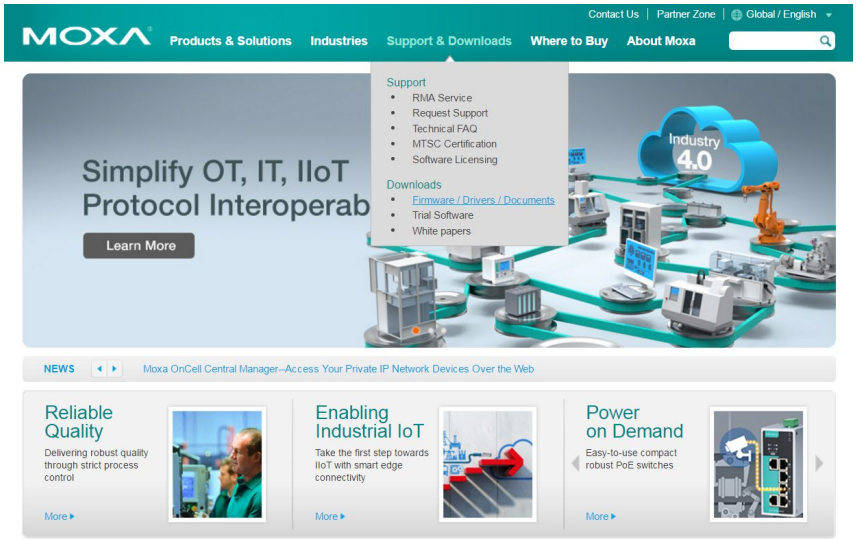
1. You will need to download a free TFTP server package to upgrade the firmware for the boot loader utility. Link to the following URL to download:
2. http://tftpd32.jounin.net/
3. Download the latest version of the TFTP program.
4. When finished downloading, extract the files to your PC.
5. To start the TFPF server, double-click on the tftpd32 icon to launch the TFTP server.

6. In the TFTP server configuration window, choose the corresponding server interface that is connected to the UC-8540/8580 computer.

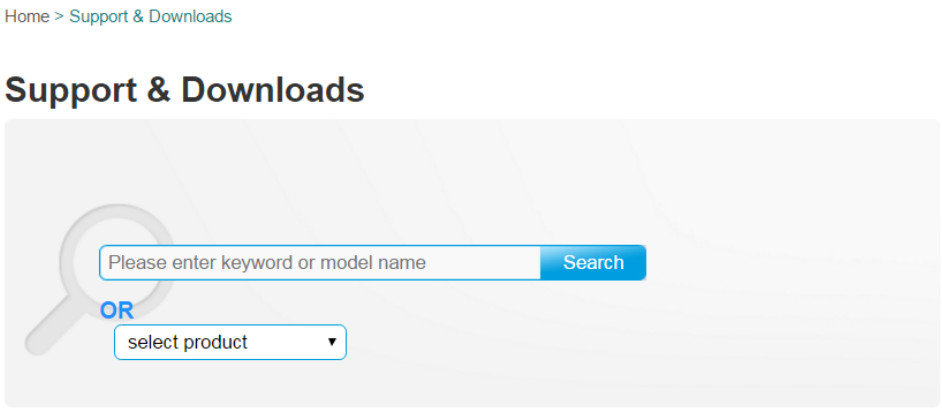


C. Downloading and Upgrading the Firmware Through the Serial Port

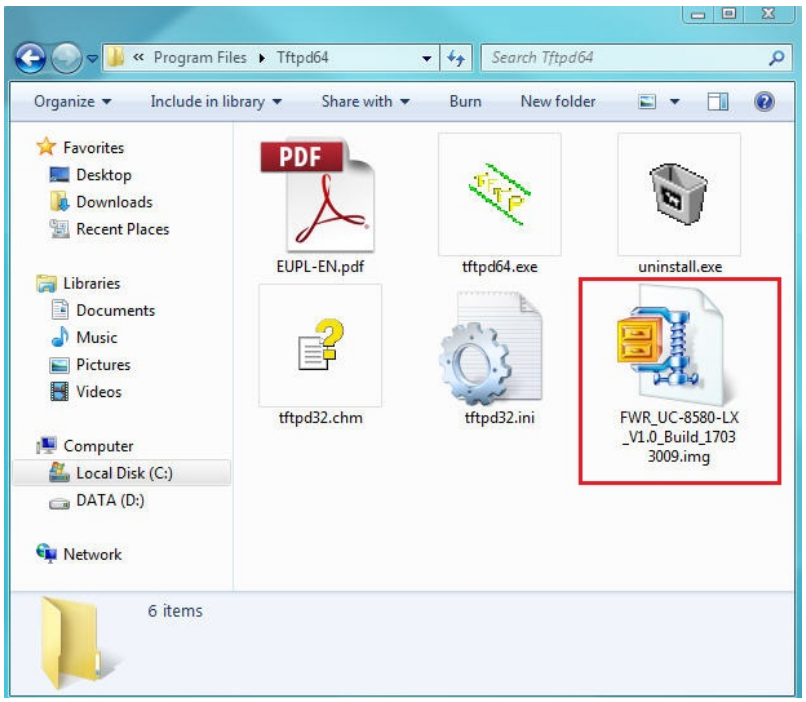
1. Connect to Moxa's website at http://www.moxa.com, and then select Firmware/Drivers/Documents from the Support & Downloads drop-down menu.



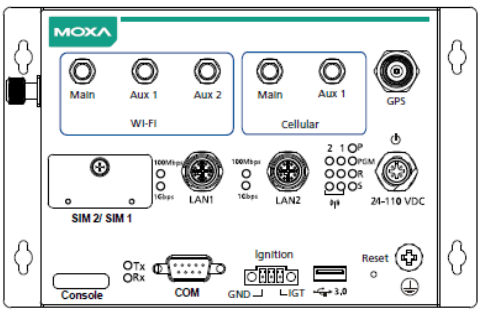
2. Enter the model name"UC-8540/8580 Series" or select the specific product model from the down-down list. Click on **Search** to continue.



3. Select Firmware from the Software column. Choose the appropriate OS and then click the download icon to start downloading the new firmware. **Note: Check the filename, it may differ from the filename shown below.**
4. Put the latest firmware icon in the same directory as the TFTP files.



5. Next, connect **LAN1** of the computer to your PC using a cross-over Ethernet cable.



6. Press and hold down the **DEL** key on your PC and power on the embedded computer at the same time. You will be guided to the boot loader utility menu, as show below.

```
--------------------------------------------------------------------------
 Model: UC-8540
 Boot Loader Version 1.0.1B01          Serial Number: IMOXA1234567
```

```
  LAN1 MAC: 00:90:e8:00:00:47          LAN2 MAC: 00:90:e8:00:00:57
--------------------------------------------------------------------------
(0) Fastboot mode               (1) Firmware Update by USB Disk
(2) Firmware Update by Tftp
--------------------------------------------------------------------------
Command>>
```

7. In the boot loader utility, select **[2] Firmware Update by Tftp**, and then **[1] Set IP Address** if you want to reconfigure IP addresses.

```
Current IP Address

Local IP Address : ipaddr=192.168.31.148

Server IP Address : serverip=192.168.31.141

Do you want to change the ip address?

0 - No, 1 - Yes (0-1,enter for abort): 1
```

8. You will need to enter the IP address of the embedded computer and your PC. Please make sure the IP address of your computer and PC are in the same network domain. For example, if the server IP address is 192.168.3.128, you can choose a local IP address between 192.168.3.1 to 192.168.3.254.

```

Local IP Address : 192.168.3.127

Server IP Address : 192.168.3.128
```

9. Enter the firmware file name which locate in the same folder with TFTP server

```
Saving Environment to SPI Flash...
SF: Detected S25FL164K with page size 256 Bytes, erase size 64 KiB, total 8
MiB
Erasing SPI flash...Writing to SPI flash...done
Valid environment: 1

Firmware File Name (firmware.img): FWR_UC-8540-LX_V1.0_Build_17033009.img
```

10. The firmware upgrade will then start to run.
11. It will take several minutes for the firmware files to be written to your computer. Do not power off your computer.
12. When you see Update OK, the firmware upgrade is finished. At this point, you may reboot the computer to complete the firmware upgrade or recovery from the boot loader utility.

If you cannot reboot your embedded computer (after following all the steps above), contact Moxa's technical support staff for further assistance.